ALMA MATER STUDIORUM - UNIVERSITÀ DEGLI STUDI DI BOLOGNA
CAMPUS DI CESENA
SCUOLA DI INGEGNERIA E ARCHITETTURA

CORSO DI LAUREA SPECIALISTICA IN INGEGNERIA INFORMATICA

PREDICTIVE TEXT MINING: METODI DI PREVISIONE DI INDICI DI BORSA BASATI SU TWITTER

Tesi in

Sistemi Informativi Distribuiti LS

Relatore
Gianluca Moro

Co-Relatore
Giacomo Domeniconi

Presentata da
Denis Di Paolo

Sessione II
Anno Accademico 2012/13

PAROLE CHIAVE

Metodi di Previsione

Opinion Mining & Sentiment Analysis

Text classification

Twitter

Indici di Borsa

Indice

Introduzione

Il problema relativo alla predizione, la ricerca di pattern predittivi all'interno dei dati, è stato studiato ampiamente. Molte metodologie robuste ed efficienti sono state sviluppate, procedimenti che si basano sull'analisi di informazioni numeriche strutturate. Quella testuale, d'altro canto, è una tipologia di informazione fortemente destrutturata. Quindi, una immediata conclusione porterebbe a pensare che per l'analisi predittiva su dati testuali sia necessario sviluppare metodi completamente diversi da quelli ben noti dalle tecniche di data mining. Un problema di predizione può essere risolto utilizzando invece gli stessi metodi : dati testuali e documenti possono essere trasformati in valori numerici, considerando per esempio l'assenza o la presenza di termini, rendendo di fatto possibile una utilizzazione efficiente delle tecniche già sviluppate. Il text mining abilita la congiunzione di concetti da campi di applicazione estremamente eterogenei. Con l'immensa quantità di dati testuali presenti, basti pensare, sul World Wide Web, ed in continua crescita a causa dell'utilizzo pervasivo di smartphones e computers, i campi di applicazione delle analisi di tipo testuale divengono innumerevoli.

L'avvento e la diffusione dei social networks e della pratica di micro blogging abilita le persone alla condivisione di opinioni e stati d'animo, creando un corpus testuale di dimensioni incalcolabili aggiornato giornalmente. Le nuove tecniche di Sentiment Analysis, o Opinion Mining, si occupano di analizzare lo stato emotivo o la tipologia di opinione espressa all'interno di un documento testuale. Esse sono discipline attraverso le quali, per esempio, estrarre indicatori dello stato d'animo di un individuo, oppure di un insieme di individui, creando una rappresentazione dello stato emotivo sociale.

L’andamento dello stato emotivo sociale può condizionare macroscopicamente l’evolvere di eventi globali? Studi in campo di Economia e Finanza Comportamentale assicurano un legame fra stato emotivo, capacità nel prendere decisioni ed indicatori economici. Grazie alle tecniche disponibili ed alla mole di dati testuali continuamente aggiornati riguardanti lo stato d’animo di milioni di individui diviene possibile analizzare tali correlazioni.

In questo studio viene costruito un sistema per la previsione delle variazioni di indici di borsa, basandosi su dati testuali estratti dalla piattaforma di microblogging Twitter, sotto forma di tweets pubblici; tale sistema include tecniche di miglioramento della previsione basate sullo studio di similarità dei testi, categorizzandone il contributo effettivo alla previsione.

Nel *capitolo 1* viene illustrata una panoramica del data mining e del text mining, fornendo un dettagliato studio dei più importanti metodi di classificazione.
Nel *capitolo 2* viene affrontata la disciplina emergente della sentiment analysis, illustrandone caratteristiche principali, applicazioni e metodi; viene poi analizzato nel dettaglio lo studio di Bollen [46], che sfrutta la sentiment analysis per effettuare una previsione dell’indice Dow Jones Industrial Average di chiusura, sull’anno 2008.
Nel *capitolo 3* vengono descritti gli strumenti utilizzati per la costruzione ed il testing del sistema proposto.
Nel *capitolo 4* viene presentata l’idea fulcro della tesi, riguardante la costruzione di un modello dei dati adatto alla previsione, nonchè di tecniche migliorative che permettono un incremento nell’accuratezza rispetto a metodi di classificazione standard.
Nel *capitolo 5* viene illustrata l’architettura implementativa del sistema utilizzato.
Il *capitolo 6* mostra i risultati ottenuti nei vari test del sistema, analizzando nel dettaglio ogni modello performante ottenuto.

Capitolo 1
Text Mining come metodo multidisciplinare

1.1 Data Mining

Il Data Mining può essere definito come il processo di estrazione di informazioni implicite, precedentemente sconosciute e potenzialmente utili, dai dati; o ancora, come un procedimento di esplorazione ed analisi, per mezzo di sistemi automatici e semi-automatici, di grandi quantità di dati al fine di scoprire pattern significativi. Questa disciplina nasce per sopperire ai limiti delle tradizionali tecniche di analisi, che falliscono sul trattamento di elevate quantità di dati eterogenei, caratterizzati da dimensionalità alte; molte delle informazioni presenti sui dati non sono direttamente evidenti e le analisi guidate dagli uomini possono richiedere settimane per scoprire indicazioni utili. Tipicamente, le attività di un sistema di data mining sono di due tipologie: *predizione* di variabili, che consiste nell'utilizzare alcune variabili per predire il valore incognito o futuro di altre variabili; *descrizione*, intesa come il procedimento di identificazione di pattern interpretabili dall'uomo in grado di descrivere i dati. Un pattern è una rappresentazione sintetica e ricca di semantica di un insieme di dati; esso esprime, in genere, un modello ricorrente nei dati stessi, ma può anche esprimere un modello eccezionale. Un pattern deve essere:

- Valido sui dati con un certo grado di confidenza
- Comprensibile dal punto di vista sintattico e semantico, affinché l'utente lo possa interpretare
- Precedentemente sconosciuto e potenzialmente utile, affinché l'utente possa intraprendere azioni di conseguenza

Il tipico processo di estrazione di conoscenza è raffigurato in figura 1, tratta da [1].

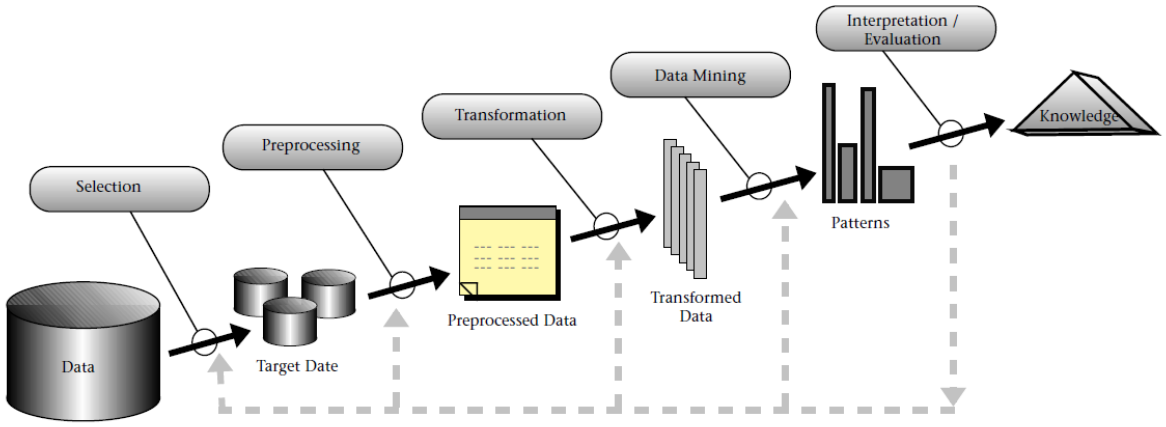

Figura 1 : Processo di estrazione di conoscenza

I dati *Data* rappresentano la conoscenza globale sul dominio applicativo; su di essi viene effettuata una selezione (*Selection*) focalizzandosi su di un sottoinsieme significativo dei dati, ottenendo *Target Date*. Questi vengono poi sottoposti ad una fase di *Preprocessing* o preprocessamento: in questo step è utile rimuovere dati rumore presente sui *Target Date*, gestire la mancanza di date e più in generale pulire e filtrare quelli disponibili. I dati preprocessati ottenuti, *Preprocessed Data*, vengono trasformati allo scopo di ridurre il numero di variabili da considerare nelle conseguenti analisi. I dati trasformati ottenuti, *Transformed Data*, vengono sottoposti al processo di *Data Mining* vero e proprio, selezionando una tipologia di estrazione di conoscenza e quindi un preciso algoritmo. I *Pattern* estratti possono essere così valutati ed interpretati ricavando la conoscenza *Knowledge*.

1.1.1 Classificazione

La classificazione rappresenta una tipica attività del Data Mining; essa rappresenta il compito di assegnare oggetti ad una fra diverse predefinite categorie, dette classi. L'input al

processo di classificazione è un insieme di records; ognuno di questi records, detto anche istanza o esempio, è caratterizzato da una tupla (x, y), dove x è l'insieme degli attributi ed y è un attributo speciale, detto attributo (etichetta) classe. L'attributo classe, anche detto categoria o attributo target, a differenza degli altri attributi del set, deve assumere unicamente valori discreti: ciò distingue la classificazione dalla regressione, dove la predizione viene effettuata su di una y a valori continui.
Formalmente, il processo di classificazione prevede la costruzione di una funzione target f, che associ ad ogni insieme di attributi x ad una delle predefinite classi y. La funzione target è anche chiamata, più informalmente, modello di classificazione. Un modello di classificazione può essere utilizzato per predire il valore dell'attributo classe di istanze per il quale la classe di appartenenza è sconosciuta.
Nelle prossime sezioni il processo di classificazione viene illustrato ed approfondito per modelli di classificazione che si occupano di attributi classe binari o nominali.

1.1.2 Approccio generale alla risoluzione di un problema di classificazione

Una tecnica di classificazione (o classificatore) è un approccio sistematico volto alla costruzione di modelli di classificazione a partire da un data set di input; esempi includono classificatori ad alberi decisionali, a regole,utilizzanti reti neurali, support vector machines e classificatori naive Bayes. Ogni tecnica incorpora un algoritmo di learning il cui obiettivo è quello di identificare un modello di classificazione che spieghi nella maniera migliore la relazione fra il set degli attributi e la classe dei dati in input. Il modello così generato ha come scopo quindi quello di esprimere correttamente le relazioni dei dati in input così come di predire correttamente la classe di appartenenza di istanze mai esaminate in precedenza. Un approccio generale alla risoluzione di un problema di classificazione è mostrato in figura 2.

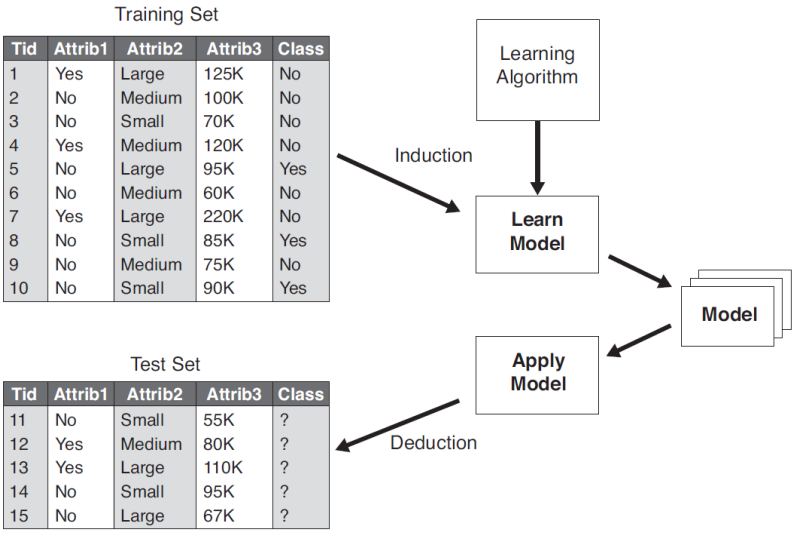

Figura 2 : Approccio generale ad un problema di classificazione

L'insieme di *training* è composto da istanze per le quali è conosciuta la classe di appartenenza: esso rappresenta una entità fondamentale, poichè è utilizzato per produrre il

modello di classificazione. Tale modello viene in seguito applicato su di un insieme detto di *test*, che consiste di istanze per le quali l'attributo classe è sconosciuto al modello di classificazione, il quale produce, per ognuna di tali istanze, la classe di appartenenza predetta. La valutazione delle performance di un modello di classificazione è basata fondamentalmente sul conteggio delle istanze di test per le quali l'attributo classe predetto si rivela corretto e per le quali invece la previsione risulta errata. Questi conteggi vengono raccolti in una tabella detta *matrice di confusione*; un esempio di matrice di confusione per un problema di classificazione binaria è riportato in tabella 1.

		Classe Predetta	
		Positive	Negative
Classe Reale	Positive	*TP*	*FP*
	Negative	*FN*	*TN*

Tabella 1 : Matrice di confusione per un problema di classificazione binaria

L'elemento *TP, True Positive*, denota gli elementi di classe Positive classificati correttamente; l'elemento *TN, True Negative*, rappresenta gli elementi di classe Negative classificati correttamente. L'elemento *FP, False Positive*, rappresenta invece gli elementi di classe Negative classificati erroneamente come di classe Positive; l'elemento *FN, False Negative*, dualmente, rappresenta gli elementi di classe Positive classificati erroneamente come di classe Negative.
Basandosi sugli elementi di una matrice di confusione, il numero totale di predizioni corrette effettuate dal modello è $TP + TN$ mentre le predizioni scorrette effettuate ammontano a $FP + FN$.
La matrice di confusione riporta le informazioni necessarie al fine di valutare le performance di un modello di classificazione; diviene comunque la definizione di indici numerici singoli in grado di riassumere tale contenuto

informativo, di modo da rendere la comparazione delle performance di molteplici modelli più immediata e conveniente.
In tal senso, una delle metriche di valutazione del modello a valore singolo più utilizzata è l'accuratezza, definita come il rapporto fra il numero delle predizioni corrette e il numero totale di predizioni effettuate.

$$Accuratezza = \frac{Numero\ delle\ predizioni\ corrette}{Numero\ delle\ predizioni\ totali} = \frac{TP + TN}{TP + TN + FP + FN}$$

In maniera equivalente, le informazioni contenute in una matrice di confusione possono essere sintetizzate utilizzando come metrica la frequenza dell'errore, definita come il rapporto fra il numero di predizioni sbagliate e il numero totale di predizioni effettuate.

$$Frequenza\ dell'errore = \frac{Numero\ delle\ predizioni\ sbagliate}{Numero\ delle\ predizioni\ totali} = \\ = \frac{FP + FN}{TP + TN + FP + FN}$$

L'accuratezza non rappresenta una metrica adeguata di sintesi delle performance nel caso in cui le classi contengano un numero fortemente diverso di record; nel caso di problemi di classificazione binaria la classe la classe più "rara", ossia che contiene meno record, è anche chiamata classe Positiva, mentre la classe che include la maggioranza dei record è chiamata classe Negativa.
In tal contesto vengono introdotte le misure di *precision*, che misura la frazione di record risultati effettivamente positivi tra tutti quelli che erano stati classificati come tali, e *recall*, che misura la frazione di record positivi correttamente classificati.

$$Precision, p = \frac{TP}{TP + FP}$$

$$Recall, r = \frac{TP}{TP + FN}$$

Valori elevati di precision indicano che pochi record della classe negativa sono stati erroneamente classificati come positivi; mentre invece valori elevati di recall indicano che pochi record della classe positiva sono stati erroneamente classificati come negativi.
Una metrica che riassume ed unifica i valori di *precision* e di *recall* è denominata F-measure; essa rappresenta la media armonica tra precision e recall: la media armonica tra due numeri x e y tende a essere vicina al più piccolo dei due numeri. Quindi se la media armonica è elevata significa che sia *precision*, sia *recall* lo sono.

$$F - measure, F = \frac{2rp}{r + p} = \frac{2 \times TP}{2 \times TP + FP + FN}$$

1.1.3 Tecniche di classificazione

Di seguito una lista delle più rilevanti tecniche di classificazione [11].

- **Alberi decisionali o Decision trees.** Essi rappresentano una delle tecniche di classificazione maggiormente utilizzate che permette di rappresentare con una struttura ad albero gerarchica un insieme di regole; tale struttura consiste di un insieme di nodi, correlati da archi (rami) orientati ed "etichettati“ di classificazione. L’albero possiede 3 diverse tipologie di nodi:
  1. un nodo radice, il quale è caratterizzato da nessun arco entrante e zero o più archi uscenti;
  2. nodi interni, i quali sono caratterizzati da precisamente un arco entrante e 2 o più archi uscenti;
  3. foglie o nodi terminali, ognuno dei quali possiede precisamente un arco entrante e zero archi uscenti.

In un albero decisionale, ogni nodo terminale viene associato ad una classe definita. I nodi non di tipo terminale, ossia il nodo radice e gli altri nodi interni all'albero, contengono condizioni di test sugli attributi, al fine di separare istanze aventi caratteristiche differenti.
Una volta ottenuto l'albero, il procedimento di classificazione diviene semplice ed immediato; partendo dal nodo radice si applicano le condizioni di test relative ad ogni nodo e si seguono gli archi corrispondenti al risultato di tali test. Questo porta l'esecuzione al raggiungimento di un nuovo nodo: nel caso sia un nodo interno all'albero, si valuterà una nuova condizione di test, procedendo come già detto; nel caso sia un nodo terminale la classe associata al nodo stesso viene associata all'istanza da classificare.

- **Classificatori basati su regole.** Classificano i record utilizzando insiemi di regole del tipo "if-then"; ogni regola assume la forma di (Condizione)->y , dove Condizione è una congiunzione di predicati logici sugli attributi dell'istanza da classificare, mentre y è l'etichetta di classe che ne consegue; la Condizione viene anche detta antecedente della regola, mentre l'etichetta della classe y è detta anche conseguente. Il modello di classificazione viene quindi costruito dalla identificazione di una serie di regole. Metriche di valutazione di un classificatore basato su regole sono la Copertura, che rappresenta la frazione dei record che soddisfano l'antecedente della regola, e l'Accuratezza, definita in termini della frazione dei record che, soddisfando l'antecedente, soddisfano anche il conseguente della regola.
- **Classificatori Nearest Neighbor.** Classificano le istanze in base alla loro somiglianza con elementi del training set; sono detti di tipo lazy, ossia pigro, poichè non costruiscono modelli: essi utilizzano i k punti "più vicini" (nearest neighbors) per effettuare la classificazione. E' necessario definire, oltre ad un insieme di training, una metrica attraverso la quale

calcolare una distanza fra i records ed il numero k di istanze "vicine" da utilizzare nella comparazione. Il processo di classificazione prevede il calcolo iniziale della distanza fra l'istanza da classificare, in input, ed i record presenti nel training set, attraverso la metrica prescelta; identifica quindi i k nearest neighbors ed infine utilizza le etichette di classe dei vicini così identificati per determinare la classe sconosciuta dell'istanza in input (per esempio, semplicemente scegliendo quella che compare con maggiore frequenza fra i vicini).

- **Classificatori Bayesiani.** Rappresentano un approccio probabilistico alla risoluzione di problemi di classificazione. In una grande quantità di applicazioni reali la relazione tra i valori assunti dagli attributi delle istanze e quello della classe non è deterministica; ciò è dovuto a possibile rumore sui dati, alla presenza di caratteristiche insite nel fenomeno ma non modellate a dovere dagli attributi oppure ancora a difficoltà nel quantificare operativamente certi aspetti del fenomeno stesso. Questo introduce talvolta incertezza sull'esito della previsione: i classificatori Bayesiani modellano relazioni probabilistiche tra gli attributi e l'attributo di classificazione per superare tali insicurezze. Essi si basano sul teorema di Bayes descritto in seguito, adattandolo al problema della classificazione.
  Sia dato il vettore $A = (A_1, A_2, \dots, A_n)$ che descrive il set di attributi e sia C la variabile di classe: se C è legata in modo non deterministico ai valori assunti da A possiamo trattare le due variabili come variabili casuali e catturare le loro relazioni probabilistiche utilizzando $P(C|A)$, ossia la probabilità che si verifichi l'evento C sapendo che si è verificato l'evento A. Durante la fase di training si imparano i legami probabilistici $P(C|A)$ per ogni combinazione di valori assunti da A e C; conoscendo queste probabilità, una istanza di test a può essere classificata trovando la label di classe c che massimizza la probabilità a posteriori $P(c|a)$.
  Calcolare $P(C|A)$ per ogni possibile valore di C e A

richiede un training set molto grande anche per un numero ridotto di attributi: il teorema di Bayes è utile in questo caso poiché permette di esprimere la probabilità a posteriori $P(C|A)$ in termini di $P(A|C)$, $P(C)$ e $P(A)$ come segue:

$$P(C|A_1, A_2, \dots, A_n) = \frac{P(A_1, A_2, \dots, A_n|C)\, P(C)}{P(A_1, A_2, \dots, A_n)}$$

Visto che $P(A)$ è costante in questa formula il problema di massimizzare la probabilità a posteriori equivale a scegliere il valore di C che massimizzi $P(A_1, A_2, \dots, A_n|C)\, P(C)$.

- **Classificatori a reti neurali.** Una rete neurale artificiale definisce un modello matematico per la simulazione di una rete di neuroni biologici. Una rete di neuroni biologici è costituita da un insieme di cellule nervose (i neuroni) collegati tramite fibre nervose. Tali classificatori sfruttano come unità base una entità definita come neurone, ispirata al neurone biologico: è in grado di ricevere informazioni di input e trasferirle al proprio interno; può trasferire le informazioni immagazzinate verso l'esterno; può trasferire e ricevere informazioni verso o da altri neuroni (equivalentemente al processo sinaptico di trasferimento di segnali tramite processo elettrochimico).
  Nelle reti neurali artificiali, ogni neurone è associato ad un insieme di pesi A, che rappresentano una misura della conoscenza accumulata dal singolo neurone; tali pesi vengono utilizzati nel processo di classificazione. Identificando con X_i il vettore delle features in ingresso alla rete, i pesi vengono combinati con i valori assunti da tale features attraverso, per esempio, la semplice funzione lineare $A * X_i$. Il risultato della classificazione sfrutta tale combinazione come input di una funzione di attivazione che simula il comportamento del neurone postsinaptico, ossia ricevente le informazioni dai layers precedenti di neuroni.
  Il modello di apprendimento di tali reti prevede di

aggiustare i pesi *A* al fine di ridurre l'errore fra il valore di output generato dalla rete neurale e l'output corretto, identificato da un training set.
Le reti neurali ad un solo strato di neuroni hanno un algoritmo di apprendimento efficiente, ma sono utili soltanto nel caso di dati linearmente separabili.
Viceversa, le reti neurali multistrato possono rappresentare funzioni non lineari, ma sono difficili da addestrare a causa dell'alto numero di dimensioni dello spazio dei pesi.

- **Classificatori a Support Vector Machines.** In italiano vengono dette Macchine a Vettori di Supporto, oppure macchine kernel; sono un insieme di metodi di apprendimento supervisionato per la regressione e la classificazione di pattern, sviluppati negli anni '90 da Vladimir Vapnik [12] ed il suo team presso i laboratori Bell AT&T.
I modelli SVM furono originariamente definiti per la classificazione di classi di oggetti lineramente separabili. Per ogni gruppo di oggetti divisi in due classi una SVM identifica l'iperpiano avente il massimo margine di separazione.

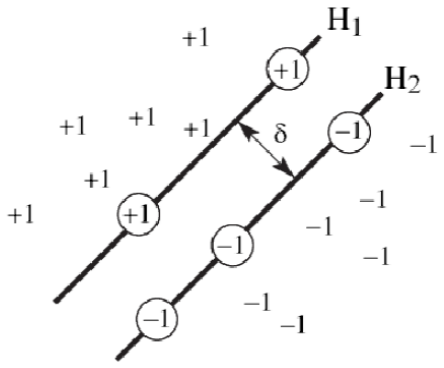

Figura 3: Esempio di separazione classi con SVM

In figura 3 l'iperpiano H_1 definisce il bordo della classe i cui oggetti sono rappresentati dai "+1" mentre l'iperpiano H_2 quello degli oggetti rappresentati dai "-1". E' quindi possibile notare che due oggetti della classe "+1" servono a definire H_1 (sono quelli cerchiati)

e ne servono tre della classe "-1" per definire H_2; questi oggetti vengono chiamati "support vectors", quindi il problema di identificare la miglior separazione tra le due classi è risolto individuando i vettori di supporto che determinano il massimo margine δ tra i due iperpiani.
Ovviamente le SVM possono essere usate per separare classi che non potrebbero essere separate con un classificatore lineare, altrimenti la loro applicazione a casi di reale interesse non sarebbe possibile. In questi casi le coordinate degli oggetti sono mappate in uno spazio detto "feature space" utilizzando funzioni non lineari, chiamate "feature function" Φ [14]. Il feature space è uno spazio fortemente multidimensionale in cui le due classi possono essere separate con un classificatore lineare, come mostrato con un semplice esempio grafico in figura 4.

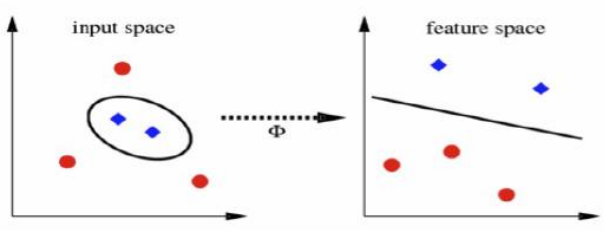

Figura 4 : Trasformazione in feature space

Questo metodo, che sta alla base della teoria delle SVM, consiste nel mappare i dati iniziali in uno spazio di dimensione superiore. Presupponendo quindi $m > n$, per la mappa si utilizza una funzione:

$$\Phi: \mathbb{R}^n \rightarrow \mathbb{R}^m$$

Come si può notare dall'esempio in figura 4, le due classi nello spazio di input non sono linearmente separabili, ma attraverso la funzione Φ i dati vengono mappati in uno spazio in cui diventano linearmente separabili e in cui sarà possibile trovare un iperpiano

che li separi.
La funzione Φ combina quindi lo spazio iniziale (le caratteristiche originali degli oggetti) nello spazio delle features che potrebbe in linea di principio avere anche dimensione infinita. A causa del fatto che questo spazio ha molte dimensioni non sarebbe pratico utilizzare una funzione generica per trovare l'iperpiano di separazione, quindi vengono usate delle funzioni dette "kernel" e si identifica la funzione Φ tramite una combinazione di funzioni di kernel.
Una efficace ottimizzazione dei kernel SVM è rappresentata dall'algoritmo SMO (*Sequential Minimal Optimization*), che si propone di risolvere i problemi di ottimizzazione presenti durante il processo di training degli SVM; ciò viene fatto suddividendo i problemi in una serie di sottoproblemi di dimensione minore possibile, i quali vengono poi risolti in maniera analitica.

1.2 Text Mining

Nel campo del Data Mining gli ultimi anni sono stati denotati da ingenti miglioramenti nelle tecniche e nei risultati, dovuti agli avanzamenti tecnologici sia per l'hardware che per il software, causanti la generazione e la conseguente disponibilità di diversi ed elevate quantità di dati. Questo è particolarmente vero per quanto riguarda i dati testuali, per la generazione dei quali lo sviluppo di piattaforme hardware e software volte al sostenimento di un rapido e portabile accesso al web ed ai social networks ha reso possibile una rapida creazione di archivi digitali contenenti una grandissima varietà di dati. In particolare, il web funge da tecnologia abilitante la creazione di contenuto testuale da parte di una varietà di tipologie di utenza molto vasta, in una forma semplice da immagazzinare e processare. La crescita in termini di disponibilità di tali dati testuali, resi disponibili da diverse applicazioni, ha creato il bisogno di una progettazione più avanzata degli algoritmi di analisi degli

stessi dati, dai quali occorre estrarre pattern interessanti in maniera dinamica e scalabile.
Dati di tipo strutturato vengono solitamente e storicamente organizzati e contenuti all'interno di database; per quanto riguarda il testo, invece, i dati vengono tipicamente sfruttati attraverso motori di ricerca, a causa della mancanza di strutture adeguate [5]. Un motore di ricerca abilita l'utente al recupero di informazioni utili da una collezione di dati, inserendo una query basata su di una (o più) parola chiave; come migliorare l'efficienza e la efficacia di un motore di ricerca è stato un tema centrale di ricerca, nel campo dell'Information Retrieval [13,3].
La ricerca nel campo del recupero di informazioni è stata tradizionalmente focalizzata più sul facilitare l'accesso alle informazioni stesse, piuttosto che all'analisi dei dati al fine di scoprire patterns interessanti, quale è l'obiettivo primario della disciplina del text mining. L'obiettivo, invece, derivante dalla necessità di accedere ad una informazione, tradizionalmente, è quello di connettere l'informazione esatta con gli utenti che ne fanno richiesta in un tempo accettabile, senza porre alcuna enfasi sulla trasformazione o sul processamento dei dati testuali, ossia: recuperare ciò che è stato richiesto, senza alcuna manipolazione. Il text mining, d'altro canto, può essere riconosciuta come una pratica che si colloca oltre al semplice accesso dei dati, incorporando come scopo principe quello di analizzare, concentrare ed apprendere informazioni e facilitare così la fase decisionale, legata a tali informazioni, dell'utenza interessata. Se quindi un tradizionale motore di ricerca si occupa di effettuare associazioni triviali fra chiavi di ricerca e possibili risultati, con il text mining si 'scava' (mining) nei dati testuali allo scopo di ricavare informazioni importanti e di interesse, estraendo conoscenza da grandi repository non strutturate.

1.2.1 Caratteristiche ed importanza dei dati testuali

Un numero elevato di caratteristiche chiave distinguono i dati testuali da altre forme di dati, come per esempio quelli relazionali. Questo naturalmente incide fortemente sulle tecniche di mining che possono essere sfruttate su tali tipologie di dati.
La caratteristica principale è quella di essere dati non strutturati, ossia dati privi di un modello/schema che li descriva o permetta di attribuire ad essi una semantica ben precisa. L'importanza di tale tipologia di dati è in continua crescita: il successo della tecnologia web e dei motori di ricerca, tramite i quali recuperare tradizionalmente dati testuali non strutturati, conferma il ruolo dei dati testuali, e della rilevanza della loro analisi. Secondo Gartner Group, una società multinazionale leader mondiale nella consulenza strategica, ricerca e analisi nel campo dell'Information Technology, l'80% dei sistemi di business vengono condotti sulla base di dati non strutturati; inoltre la quantità totale di dati testuali non strutturati raddoppia ogni 3 mesi.
Un'altra caratteristica importante relativa ai dati testuali è che la sua rappresentazione risulta in modelli sparsi e di dimensionalità molto elevate: per esempio, se disponiamo di un vocabolario di 100,000 termini e vogliamo rappresentare un insieme di documenti, ognuno di questi conterrà una percentuale dei termini totali, probabilmente qualche centinaio di parole in tutto; perciò un insieme di documenti testuali potrà essere rappresentato da una *matrice sparsa termine-documento* di dimensioni $n \times d$, dove n è il numero di documenti rappresentati e d il numero di termini del vocabolario; l'elemento (i,j) della matrice rappresenterebbe in questo contesto la frequenza (normalizzata) del termine j-esimo all'interno del documento i.

1.2.2 Applicazioni del Text Mining

Nel contesto del Text Mining emergono una grande numerosità di possibili applicazioni [15], e correlati problemi di analisi e modellazione. Sorto con la spinta della ricerca nel campo del Data Mining, esistono una grande varietà di comunità scientifiche provenienti da diversi campi di studio che ne collaborano per il progresso e la ricerca, includendo applicazioni in processamento di linguaggio naturale (natural language processing), recupero di informazioni (Information Retrieval), apprendimento automatico, intelligenza artificiale e che riguardano domini fra i più disparati, dal World Wide Web alle scienze biomediche.

- **Estrazione di informazioni da dati testuali.** E' una delle applicazioni chiave riguardanti il text mining, la quale assume il ruolo di punto di partenza per molti algoritmi. Per esempio, l'estrazione di entità e delle relazioni fra di esse da un testo può essere in grado di rivelare informazioni semantiche molto più ricche della mera considerazione dei termini utilizzati all'interno di un documento, ed è di importanza strategica fondamentale allo scopo di inferire conoscenza nascosta all'interno delle strutture sintattiche/semantiche.
- **Sintetizzazione di testi.** Un'altra funzione comune richiesta da molte applicazioni di text mining è quella di sintetizzare documenti testuali al fine di ottenere un riassunto o una panoramica di documenti di testo molto lunghi oppure di un insieme di documenti appartenenti allo stesso argomento. Il bisogno di questo tipo di trattamento dei dati è naturale conseguenza della vasta disponibilità di dati non strutturati testuali dovuta all'incremento delle nuove tecnologie abilitanti. Le tecniche di sintetizzazione generalmente fanno parte di due categorie: nella sintetizzazione per estrazione (*extractive summarization*) il riassunto viene costruito estraendo unità di informazione testuale estratte direttamente dal testo originale; nella sintetizzazione

per astrazione (*abstractive summarization*) il riassunto potrebbe contenere invece anche unità informative testuali costruite ad hoc, ossia non facenti parte del testo originale.

- **Metodi di apprendimento non supervisionati su dati testuali.** I metodi di apprendimento non supervisionati non richiedono nessun insieme di training per costruire il modello di classificazione, e quindi possono essere applicati a qualsiasi tipologia di dato testuale senza richiedere sforzi manuali. Le due principali tipologie di metodi di apprendimento non supervisionati usate comunamente nel contesto dei dati testuali sono il *clustering* ed il *topic modeling*.
  Il problema che si pone un metodo di clustering è quello di segmentare una raccolta di documenti in partizioni, ognuna delle quali corrispondente ad un cluster facente riferimento un certo topic, o argomento in senso generico. Clustering e topic modeling sono correlate strettamente: nel topic modeling viene utilizzato un modello probabilistico allo scopo di determinare l'appartenenza di un documento ad un certo cluster;ciò che risulta viene detto soft clustering, nel quale viene associata ad ogni documento una probabilità di appartenenza ad un certo cluster, in maniera diversa dall'hard clustering, o clustering tradizionale.
- **Metodi di apprendimento supervisionati su dati testuali.** I metodi di apprendimento supervisionato sono metodi generali di apprendimento automatico che sfruttano un insieme di dati di training, od addestramento per addestrare un classificatore e produrre un modello di classificazione che può essere utilizzato per computare predizioni su dati nuovi. Esiste una grande gamma di problematiche esprimibili attraverso metodi di apprendimento supervisionati. Molti dei metodi tradizionali in apprendimento automatico sono stati estesi al fine di risolvere problemi in ambito di text mining. Questi includono metodi come quelli basati su classificatori a regole, alberi decisionali, classificatori nearest neighbor, che verrano trattati più

nello specifico, essendo parte fondante del presente elaborato, nel sottoparagrafo 1.2.4.

- **Transfer learning con dati testuali.** Basti pensare ad un problema di mining che sfrutta dati testuali espressi in diverse lingue per portare l'attenzione sulla possibile eterogeneità di dati da trattare. L'obiettivo del transfer learning è quello di trasferire conoscenza acquisita da un dominio ad un altro: questo è di fondamentale importanza quando, per esempio in metodi di apprendimento supervisionato, insieme di training ed insieme di test sono costruiti su differenti insiemi di features. Tornando all'esempio riguardante la possibilità di trasferire conoscenza da un dominio fondato su dati testuali espressi in una data lingua ad un altro invece connotato dall'utilizzo di una seconda lingua, un trasferimento di conoscenza eseguito con accuratezza assume rilevanza molto elevata, soprattutto qualora vi sia penuria di dati in uno dei due domini. Altri scenari in cui il trasferimento di conoscenza diviene un problema emergente riguardano la disponibilità di dati eterogenei fra dati testuali e dati multimediali; spesso questo è il caso di applicazioni web come Flickr,Youtube o altri siti di condivisione di contenuti multimediali in genere.
- **Mining Text streams.** Molte recenti applicazioni nate sul web creano flussi ingenti di dati testuali; in particolare applicazioni come i social networks rendono possibile l'immissione simultanea di testo da parte di una varietà molto ampia di utenti e possono risultare perciò in un flusso continuo di informazioni testuali dai volumi elevati. In maniera simile, servizi di recupero di notizie come Reuters oppure aggregatori come Google news creano flussi di testi dal volume talmente consistente da poter effettuare mining in maniera continuativa. In questo contesto l'analisi di tali tipologie di flussi continui è di stimolante elaborazione, poichè sorge la necessità di processare i testi nel contesto di un vincolo one-pass: questo significa che è difficile immagazzinare i dati necessari per un trattamento tradizionale off-line, e che quindi il compito

di mining debba essere eseguito continuamente, non appena i dati siano disponibili.

- **Text mining multi-lingua.** Già accennato in precedenza, è diventato particolarmente utile, a fronte delle nuove applicazioni web based globali, considerare l'applicazione di tecniche di mining su testi di diversa lingua, oppure di trasferire la conoscenza sviluppata su di documenti di una certa lingua in un dominio caratterizzato da una diversa. Per esempio, potrebbe essere desiderabile, in una applicazione di clustering multi-lingua, considerare documenti in diverse lingue, così che documenti espressi in linguaggi differenti ma caratterizzati da argomenti simili possano essere collocati nel medesimo cluster.
- **Text Mining in social media.** Una delle fonti più comune di dati testuali presenti sul web è dovuta alla presenza di applicazioni social media, che abilitano gli esseri umani ad esprimersi in maniera veloce e completamente libera su di un vasto range di diversi argomenti. Il processo di mining testuale in ambito social media richiede l'abilità di elaborare dati estremamente dinamici nei contenuti, nonchè caratterizzati da vocabolari non standard.
  Dati testuali espressi in tale ambito possono essere ulteriormente analizzati tramite il meccanismo generato da social networks collegati fra di essi: per esempio, metodi che sfruttano sia il contenuto testuale che i vari collegamenti ottengono risultati più efficaci di metodi che scartano una delle due entità.
- **Opinion Mining da dati testuali.** Un ammontare considerevole di dati testuali presente su siti web riguarda l'espressione di opinioni o nel contesto di recensione di prodotti da parte di differenti utenti. Analizzare tali testi con tecniche di mining rivelando e riassumendo le opinioni a proposito di un certo argomento ha applicazioni universali, come per esempio nel supportare consumatori per ottimizzare le decisioni ed in ambito di business intelligence. Le problematiche in tale campo di applicazioni sono numerose: determinare quali porzioni di documento contengano o

no opinioni personali non è semplice, così come inferire il grado di soggettività di frasi oppure discernere dalle espressioni di sarcasmo o di spam.

- **Text Mining su dati biomedici.** Le tecniche di text mining assumono un ruolo primario nel consentire a ricercatori in ambito biomedico di accedere efficacemente ed in maniera efficiente a conoscenza letteralmente seppellita in archivi digitali contenenti una quantità di letteratura scientifica esorbitante. Al fine di facilitare e velocizzare scoperte in campo biomedico, tecniche di mining divengono utili nell'analizzare dati biomedici come sequenze genomiche e strutture di proteine.

1.2.3 Text preprocessing

Per effettuare mining su collezioni di documenti di grandi dimensioni diventa quindi necessario effettuare un preprocessing dei dati testuali ed immagazzinare le informazioni estratte in strutture di dati adeguate.
I dati testuali possono essere quindi analizzati a differenti livelli di rappresentazione:

- Come una *bag of words*, in cui un documento è rappresentato da un vettore di parole, in cui a ognuna è associata o la presenza/assenza, nel caso binomiale, o la frequenza dell'occorrenza, nel caso multinomiale, all'interno del documento stesso.
  Le frequenze vengono usualmente e convenientemente espresse tramite la funzione di peso *tf-idf* (term frequency-inverse document frequency), utilizzata in Information Retrieval per misurare l'importanza di un termine rispetto ad un documento o ad una collezione di documenti.
  La funzione può essere scomposta in due fattori: Il primo fattore della funzione è il numero dei termini presenti nel documento (*tf*). In genere questo numero viene diviso per la lunghezza del documento stesso per evitare che siano privilegiati i documenti più lunghi.

$$tf_{i,j} = \frac{n_{i,j}}{|d_j|}$$

dove $n_{i,j}$ è il numero di occorrenze del termine t_i nel documento d_j, mentre il denominatore è semplicemente la dimensione, espressa in numero di termini, del documento d_j.
L'altro fattore della funzione indica l'importanza generale del termine nella collezione:

$$idf_{i,D} = \log \frac{|D|}{|\{d \in D: t_i \in d\}|}$$

dove $|D|$, cardinalità di D, è il numero totale di documenti nella collezione, mentre al denominatore $|\{d \in D: t_i \in d\}|$ rappresenta il numero dei documenti dove il termine t_i compare. Infine, il valore finale del termine composto viene calcolato come

$$tfidf_{i,j,D} = tf_{i,j} \times idf_{i,D}$$

- Mantenuti direttamente come stringhe di testo, ogni documento è una sequenza di parole.

In molte applicazioni sarebbe desiderabile rappresentare le informazioni testuali in maniera semantica, di modo da poter effettuare su di esse analisi più significative, risultanti in un text mining più esaustivo. Per esempio, essere in grado di rappresentare un testo con un livello di strutturazione che renda possibile distinguere entità come persone, organizzazioni, località e le relative relazioni intercorrenti potrebbe con grande probabilità essere utile al fine di scoprire patterns più interessanti, piuttosto che utilizzare una tradizionale rappresentazione a bag of words.
Sfortunatamente le tecniche allo stato dell'arte attuale nel campo del natural language processing non sono robuste abbastanza da essere utilizzate in domini testuali non ristretti, allo scopo di generare rappresentazioni semantiche accurate del testo. Per questo motivo moltissimi approcci text mining fanno utilizzo di rappresentazioni dei dati basate

su approcci a bag of words che, nonostante perdano le informazioni circa la posizione dei termini all'interno del documento originario, sono generalmente più semplici da manipolare da un punto di vista algoritmico rispetto alle controparti basate su stringhe di testo.

Qui di seguito vengono analizzati, prendendo in considerazione la rappresentazione a bag of words, alcuni step di preprocessing del testo, che divengono necessari al fine di produrre la struttura adatta e preparare i dati alle analisi di mining.

- **Divisione in tokens.** Per ottenere tutte le parole utilizzate all'interno di un testo è necessario utilizzare un processo di tokenization: questo consiste nell'estrarre da un documento testuale un flusso di parole rimuovendo la punteggiatura e sostituendo i tabs ed altri caratteri non testuali con spazi bianchi singoli. La rappresentazione così ottenuta viene utilizzata per altri step di preprocessing. L'insieme delle differenti parole ottenute unificando tutti i documenti testuali di una collezione viene chiamata dizionario della collezione di documenti.
- **Filtraggio dei termini.** I metodi di filtraggio (filtering) rimuovono parole dal dizionario e quindi dai documenti stessi. Un classico metodo è quello di rimozione delle stop words; l'idea alla base di questo approccio è quella di rimuovere parole che portano con sè poca o nulla informazione contestuale, come articoli, congiunzioni, preposizioni ecc. Una analisi più accurata porta alla conclusione che anche le parole estremamente frequenti possono essere considerate come contenenti informazione molto bassa al fine di distinguere un documento da un altro, così come termini che raramente vengono ritrovati possono essere eliminati dal dizionario.
- **Lemmatizzazione.** Rappresenta il processo di riduzione di una forma flessa di una parola alla sua forma canonica (non marcata), detta lemma; in pratica i metodi di lemmatizzazione si propongono di mappare le forme

verbali alla loro forma infinita, e sostantivi alla loro forma singolare. Per ottenere ciò è necessario che la forma di ogni parola sia nota, quindi che, per ogni termine, sia conosciuta la parte del discorso associata, come per esempio 'verbo' oppure 'sostantivo' eccetera. Siccome il procedimento di etichettamento delle parti del discorso è normalmente oneroso sia per quanto riguarda il tempo di esecuzione sia per quanto concerne la risoluzione degli errori associati (frequenti), vengono solitamente applicati metodi di stemming.

- **Stemming.** I metodi di stemming cercando di ottenere le forme base delle parole, per esempio, considerando la lingua inglese, eliminando la 's' finale dai nome, il suffisso 'ing' dai verbi ecc. Uno stem è un gruppo naturale di parole con significato uguale, o molto simile. In seguito ad un processo di stemming, ogni parola viene rappresentata dal proprio stem. Un algoritmo di stemming basato su regole e di notorietà elevata è stato originariamente proposto da Porter [4], ed è ampiamente utilizzato: definì un insieme di regole di produzione per trasformare iterativamente parole inglesi nei propri stem.

Altri metodi si propongono di effettuare una selezione dei termini da includere nel dizionario più oculata, di modo da ridurre la dimensionalità delle features da considerare; tali metodi si propongono di selezionare, fra tutti i termini utilizzati in una collezione di documenti, quelli che rappresentano un contenuto informativo più elevato, nel contesto di un processo di classificazione specifico. Esistono numerosi metodi di *feature selection* in letteratura, volti al Text Mining [16]; la caratteristica più rilevante considerata al fine di misurare la qualità di una feature selection è data dalla capacità di favorire la selezione di feature comuni e di considerare le caratteristiche del dominio sul quale si agisce e dell'algoritmo.

- **Index Term Selection.** Al fine di diminuire ulteriormente il numero di termini che debbano essere inseriti nel dizionario è possibile sfruttare algoritmi di

indexing o di selezione di parole chiave (keyword selection). In tal caso, solo le parole selezionate andranno a far parte del dizionario finale utilizzato per descrivere i documenti. Un metodo semplice per estrarre le parole chiave da utilizzare è quello di selezionarle in base alla loro entropia. Data P_i la probabilità globale della classe i, e $p_i(w)$ la probabilità che il documento appartenga alla classe i considerato il fatto che contiene la parola *w*, si definisce $F(w)$ la frazione dei documenti contenenti la parola *w*. La misura dell'entropia, o information gain, è definita come:

$$I(w) = -\sum_{i=1}^{k} P_i \cdot \log(P_i) + F(w) \cdot \sum_{i=1}^{k} p_i(w) \cdot \log(p_i(w)) +$$

$$+(1 - F(w)) \cdot \sum_{i=1}^{k} 1 - p_i(w) \cdot \log(1 - p_i(w))$$

Ciò che risulta dall'equazione di cui sopra indica che più elevato è il valore assunto da $I(w)$ maggiore è il potere di discriminazione di *w*; parole che sono contenute in molti documenti avranno una entropia bassa.
Per ottenere un numero fissato di termini nel dizionario che coprano opportunamente i documenti, può essere applicata una semplice strategia greedy, che aggiunge la soluzione migliore ad ogni passo: partendo dal primo documento nella collezione si sceglie il termine caratterizzato dal valore di entropia più alto e si marcano tutti i documenti contenenti tale termine; si continua con il primo documento non marcato selezionando il termine a maggiore entropia e si marcano nuovamente tutti i documenti contenenti tale termine e così via, continuando il processo fino a quando tutti i documenti risulteranno marcati; arrivati a questo punto è possibile eliminare tutti i marchi e riniziare da capo, sino ad ottenere il numero di termini prestabilito.

- **Gini index.** E' uno dei metodi più comuni per quantificare il livello di discriminazione di una feature. Utilizza una misura detta gini-index, o coefficiente di gini, introdotta dallo statistico italiano Corrado Gini. Dati $p_1(w) \dots p_k(w)$ frazioni della presenza nelle k diverse classi per la parola w, ovvero $p_i(w)$ è la probabilità condizionata che un documento appartenga alla classe i considerato il fatto che contiene la parola w;è possibile constatare quindi che:

$$\sum_{i=1}^{k} p_i(w) = 1$$

  Allora il gini-index per la parola w, denotato con $G(w)$ viene definito come segue:

$$G(w) = \sum_{i=1}^{k} p_i^2(w)$$

  $G(w)$ indica il potere discriminativo della parola w: più è alto, maggiore è la discriminazione. Il problema di questo approccio è che inzialmente la distribuzione delle classi non è accurata e può non riflettere correttamente la reale potenza di discriminazione delle parole. Una possibile modifica per ovviare parzialmente a questo problema si ha inserendo una normalizzazione nelle $p_i(w)$.

Durante la feature selection si può decidere che tipo di vocabolario utilizzare, il metodo standard consiste nel considerare ogni singola parola (unigramma) e valutare in base all'occorrenza della parola stessa nei documenti il proprio apporto discriminitivo.
Un metodo diverso consiste invece nel valutare anche più parole occorrenti consecutivamente, portando ad avere vocabolari n-gram, dove n è il numero massimo di parole consecutive considerate. In questo modo si possono carpire semantiche discriminative impossibili da valutare con semplici unigrammi, ad esempio tramite un vocabolario 3-gram si può valutare l'occorrenza di "world wide web",

parole che singolarmente non sarebbero molto discriminative, ma che insieme esprimono un concetto potenzialmente utile alla classificazione.

1.2.4 Tecniche di classificazione per il Text Mining

La classificazione di testi ha come obiettivo quello di assegnare classi a documenti testuali. Un esempio potrebbe essere quello in cui l'obiettivo sia di etichettare automaticamente ogni notizia ottenuta da un aggregatore web con un argomento, come "sport", "politica" oppure "arte". Qualsiasi sia lo specifico metodo utilizzato, un processo di mining fa utilizzo di un insieme di training $D = (d_1, \dots, d_n)$ composto da documenti per i quali la classe di appartenenza $C \in \mathbb{C}$ è conosciuta; l'obiettivo è quindi quello di determinare un modello di classificazione $f: D \to \mathbb{C}$, $f(d) = C$ che sia in grado di assegnare la classe corretta ad un nuovo documento d del dominio di interesse.
Nel caso si utilizzi una modellazione VSM, *Vector Space Modeling*, ogni documento esaminato è un vettore, le cui componenti sono relative alle parole scelte per la sua rappresentazione, che può essere normalizzato in vettore unitario; lo spazio vettoriale che ne consegue è ad elevata dimensionalità e sparsità:

- L'alta dimensionalità è diretta conseguenza della necessità della scelta di un numero di termini rappresentativi di un documento inevitabilmente elevato
- L'alta sparsità è dovuta alla scelta di un feature set comune a tutti i documenti dell'insieme D; non tutti gli elementi verranno per questo rappresentati attraverso tutte le parole presenti nel feature set

Quindi il training set, essendo un insieme di documenti, diviene un insieme di punti in uno spazio vettoriale. L'obiettivo è quello di trovare buone separazioni spaziali fra i punti nello spazio così costruito (vettori/documenti) appartenenti a classi differenti.

- **Alberi decisionali o decision trees per Text Mining.** Un albero di decisione è essenzialmente una decomposizione gerarchica dello spazio dei dati (di training) dove un predicato, o condizione, sugli attributi è usato per dividere lo spazio gerarchicamente; nei problemi di text-mining generalmente questa condizione riguarda la presenza o meno di una o più parole nel documento. I tipi di split con cui dividere lo spazio dei dati possono essere:
  - Single attribute splits: in questo caso si usa la presenza o l'assenza di una parola in un particolare nodo dell'albero per effettuare lo split; a ogni livello, viene utilizzata la parola che discrimina maggiormente le classi, misurata ad esempio con il Gini-index.
  - Similarity-based multi-attribute split: si usano le parole clusterizzate per la similarità tra documenti. Discriminant-based multi-attribute split: si sceglie un cluster di parole che discrimini maggiormente le diferenti classi.

  Un'implementazione molto utilizzata in letteratura dei Decision Trees è il C4.5 [2] e C5, che usa il single-attribute split. I Decision Tree sono spesso usati insieme a tecniche di boosting, una tecnica adattiva che può essere usata per aumentare l'accuratezza della classificazione usando n classificatori, con l'n-esimo classificatore che viene costruito esaminando gli errori dell'(n-1)-esimo.

- **Classificatori testuali basati su regole.** Come già detto,lo spazio dei dati in tali tipologie di classificatori è modellato come un'insieme di regole, che partendo da condizioni sul feature-set indirizza una label; queste regole sono generalmente espresse come semplici congiunzioni di condizioni sulla presenza dei termini. Le condizioni maggiormente usate nella generazione delle regole,dal training-set, sono:
  - Supporto: quantifica il numero assoluto di istanze nel training-set rilevanti per la regola, in pratica

quantifica il volume statistico che è associato alla regola. Data una associazione $A \rightarrow B$ è la proporzione di transazioni che contengono sia A e B.
    - Confidenza: quantifica l'accuratezza dell'associazione. Consiste nella probabilità condizionata che le transazioni che contengono A, contengano anche B.

  Nella fase di training vengono costruite tutte le regole, cercando per ogni istanza tutte le regole rilevanti. Un interessante classificatore rule-based per dati testuali prevede l'utilizzo di una metodologia iterativa, tramite la quale viene determinata la singola regola migliore per ogni classe nel training-set in termini della confidenza della regola [7]; in tal senso seguono alla fase di training due step: dapprima uno step di rule induction, nel quale vengono individuate le regole di decisione che siano in grado di distinguere una categoria (classe) dalle altre, e poi uno step di evaluation, dove la miglior regola, fra quelle generate dallo step precedente, viene selezionata.
  Un ulteriore tecnica è rappresentata dall'algoritmo RIPPER [8], in grado di determinare le combinazioni frequenti di parole che sono in relazione con una particolare classe. Questo algoritmo utilizza i documenti rappresentati direttamente come lista di tokens, in maniera diretta; ma ciò che è più interessante è la capacità di considerare il contesto in cui un termine appare all'interno di un documento: il contesto quindi di una parola w influenza come l'assenza o la presenza di w in un documento contribuisce alla classificazione.

- **Classificatori Bayesiani per Text Mining.** Se in 1.1.3 è stato affrontato il caso da un punto di vista generico di Data Mining, qui si analizzano le caratteristiche per la classificazione di dati testuali attraverso un modello probabilistico. L'obiettivo diventa quindi quello di determinare la categoria/classe $c_j \in C$ più probabile di un documento d in base ai termini x_i presenti. Quindi,

applicando la regola finale costruita in precedenza, il problema può essere definito con la formula

$$C_{MAP} = argmax\, P(x_1, x_2, \dots, x_n|c_j)P(c_j)\ ,\ c_j \in C$$

Dove MAP è la massima probabilità a posteriori, che identifica la categoria (classe) più probabile.
In questo contesto esistono principalmente tre modelli che definiscono quali features considerare per costruire le rappresentazioni dei documenti da classificare; essi sono:

1. Modello Bernoulli/Binomiale Multivariato; si usa la presenza/assenza delle parole nel testo. Viene costruita una feature X_w per ogni parola del dizionario; $X_w = 1$ se la parola appare nel documento, $X_w = 0$ altrimenti. Assunzione fondamentale : data la categoria di un documento, la presenza di ogni parola nel documento è considerata indipendente dalle altre.
2. Modello Multinomiale; si utilizza la posizione delle parole all'interno del documento. Viene costruita una feature X_i per ogni posizione di parola del documento; $X_i =$ alla parola trovata in posizione i-esima all'interno del documento. Assunzione fondamentale : data la categoria di un documento, la posizione di ogni parola nel documento è considerata indipendente dalle altre.
3. Variante Multinomiale; considera la frequenza delle parole nel testo. Viene costruita una feature T_j per ogni parola del dizionario; $T_j =$ alla frequenza della parola all'interno del documento. Valgono le precedenti assunzioni per i precedenti modelli.

Il processo di classificazione combina quindi uno di questi modelli con delle regole di selezione.

- **Classificatori a Support Vector Machines con dati testuali.** I dati testuali sono molto adatti ai classificatori SVM, in quanto l'alta sparsità e dimensionalità dei dati può risultare in una più facile

separazione lineare delle classi di appartenenza [9]. Uno degli algoritmi più diffusi per il training di classificatori SVM è conosciuto come Sequential Minimal Optimization, ed è implementato in nel popolare tool LIBSVM; considerando un problema di classificazione binaria, con dati $(x_1, y_1) \dots (x_n, y_n)$, dove x_i è un vettore di input, quindi rappresentante un documento, e $y_i \in \{-1, +1\}$ è l'etichetta binaria rappresentante la classe di appartenenza corrispondente all'elemento x_i . Quindi, formalmente, un classificatore SVM viene addestrato risolvendo il seguente problema quadratico:

$$\max_{\alpha} \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i=1}^{n} \sum_{j=1}^{n} y_i \, y_j K(x_i, x_j) \alpha_i \alpha_j$$

con $0 \leq \alpha_i \leq C$, per $i = 1,2, \dots, n$ e $\sum_{i=1}^{n} y_i \alpha_i = 0$; dove C è un parametro in ingresso alla SVM, $K(x_i, x_j)$ è la funzione kernel (anch'essa selezionata dall'utente) e le variabili α_i sono moltiplicatori di Lagrange.
L'algoritmo SMO è di tipo iterativo e viene utilizzato per risolvere il problema quadratico appena descritto; L'idea di questo algoritmo è di risolvere i problemi di ottimizzazione presenti durante il processo di training degli SVM, ciò viene fatto dividendo i problemi in una serie di sottoproblemi più piccoli possibile, i quali vengono poi risolti in maniera analitica. I passi dell'algoritmo sono:

1. Seleziona una coppia di variabili α_1e α_2.
2. Congela tutte le variabili eccetto α_1e α_2.
3. Risolvi il problema considerando solo α_1e α_2.
4. Ripeti fino alla convergenza.

Il numero di passi necessari per arrivare alla convergenza è fortemente dipendente dal metodo di selezione utilizzato nel passo 1.

- **Classificatori Nearest Neighbor per dati testuali.**
  Invece che costruire modelli espliciti per le differenti classi possiamo selezionare i documenti dal training set che sono "simili" al documento di cui vogliamo analizzare la classe di appartenenza; in questo modo, la classificazione è immediata, inferendo dai documenti più "simili". Se vengono considerati k documenti simili per ogni procedimento di classificazione, l'approccio viene anche detto k-nearest neighbor classification.
  Esistono un gran numero di misure di similarità nell'ambito del text mining. Una possibilità semplice è quella di contare il numero di parole in comune fra due documenti (certamente una normalizzazione è indispensabile in questo caso, per considerare documenti di lunghezza differente); questo approccio risulta essere eccessivamente semplicistico, in quanto il contenuto informativo delle parole non è costante e varia in uno spettro molto ampio.
  Un metodo standard in ambito text mining è quello di calcolare la similarità in termini di cosine similarity; essendo i documenti rappresentati come vettori, è possibile confrontarli calcolando il coseno dell'angolo compreso fra di essi. Con A e B vettori, è possibile scrivere

$$S(A,B) = \cos\theta = \frac{A \cdot B}{\|A\|\|B\|}$$

  come misura di similarità. I risultati possono assumere valori da 1, per uguaglianza, 0 per indipendenza (usualmente) e -1 per completa disuguaglianza: da notare che, nel caso del text mining e quindi di vettori a valori positivi, i risultati saranno sempre appartenenti all'intervallo [0,1].
  Per decidere qualora un documento d_i appartenga o meno ad una certa classe C, viene calcolata la similarità $S(d_i, d_j)$ con tutti i documenti d_j nell'insieme di training; quindi, i k elementi con similarità più elevata (neighbors) vengono selezionati. La proporzione di

neighbors facenti parte della stessa classe può essere presa in considerazione come una stima della probabilità di appartenenza alla classe stessa, e la classe con la più larga proporzione è infine assegnata a d_i . Il valore ottimale k dei neighbors da considerare può essere stimato mediante cross-validation sull'insieme di training.

- **Reti neurali e text mining.** Questa tipologia di classificatori si basa su di una unità base detta neurone, il quale raccoglie un insieme di input X_i, nel nostro caso rappresentanti la frequenza dei termini nel documento i-esimo. Ogni neurone è associato a un insieme di pesi A, che viene utilizzato nella funzione di classificazione; un esempio di tipica funzione lineare è $p_i = A * X_i$. L'idea è quella di partire con pesi scelti in maniera casuale, o 0, e gradualmente aggiornarli ogni volta che si riscontra un errore di classificazione, applicando la funzione corrente dell'esempio di training con una potenza di aggiornamento regolata da un parametro μ (learning-rate). La potenza di questa tecnica risiede nella possibilità di separare classi non separabili linearmente tramite l'utilizzo di strati multipli di neuroni; il prezzo da pagare è però la complessità del processo di training e che l'errore deve essere propagato a ritroso lungo gli strati.
  Alcune osservazioni e test [10] mostrano che i benefici di classificatori non lineari rispetto ai lineari non pagano, in termini di efficienza ed efficacia, il prezzo computazionale speso per tale implementazione.

Capitolo 2
Public mood ed indicatori economici

Per **public mood** si intende lo stato emotivo che, da prerogativa di un singolo essere umano, si propaga a stato sociale, come caratteristica della totalità degli individui. Una parte importante del procedimento di estrazione di informazioni generate da esseri umani, principalmente sottoforma di testo e quindi dati non strutturati, ha come scopo quello di inferire cosa le persone stesse pensino, o vogliano esprimere, in termini di opinione o di sentimento. Uno dei fattori che causa la diffusione delle discipline che studiano il public mood, così come per la le tecniche del text mining, è il crescente interesse verso la grandissima quantità di dati testuali disponibili; questa volta, in particolare, di tipo soggettivo. In questo contesto nasce quindi la disciplina della Sentiment Analysis.
Studi in economia e finanza comportamentale assicurano un legame fra stato emotivo, capacità nel prendere decisioni ed indicatori economici; diversi studi sono stati condotti nel campo, con risultati sorprendenti ed analizzati nel corso di questo capitolo.

2.1 Sentiment analysis

Cosa le persone pensino e quale sia il loro stato emotivo rappresenta un contenuto informativo di grande importanza nell'atto del decision-making; basti pensare a quanto recensioni e valutazioni influenzino ogni essere umano nella scelta, per esempio, di un prodotto [17]. Oggi, chiunque voglia acquistare un prodotto, online e non, tipicamente ricerca recensioni ed opinioni sul prodotto stesso, scritte da altre persone.
L'anno 2001 rappresenta l'inizio della presa di coscienza da parte del mondo scientifico delle opportunità che la ricerca

sui temi quali Sentiment Analysis (ed Opinion Mining, usati spesso interscambiabilmente) potrebbe potenzialmente creare; in seguito sono innumerevoli le pubblicazioni scientifiche a riguardo. La Sentiment analysis è una delle aree di ricerca più interessate degli ultimi anni, con più di 7000 pubblicazioni scientifiche a riguardo (ad Aprile 2013). I fattori di questa esplosione di interesse possono essere ricondotti a:

- Il miglioramento dei metodi di apprendimento automatico per quanto riguarda natural language processing e information retrieval.
- La grande disponibilità di dati sui quali poter effettivamente addestrare calcolatori.
- Il fascino intellettuale (e commerciale) che lo sviluppo di applicazioni in tale area offre.

La Sentiment Analysis è quindi una disciplina che, sfruttando il Natural Language Processing, analisi del testo e linguistica computazionale, si pone come scopo quello di identificare ed estrarre contenuto informativo soggettivo, ossia associabile ad un autore che ne incorpora il significato, da dati solitamente testuali.
Diversi aspetti rendono il trattamento di testo con tecniche di Sentiment Analysis diverso dallo studio effettuato tramite Text Mining.
Tradizionalmente, la classificazione testuale mira ad associare ad un dato documento un certo argomento, o topic; in tal senso, si può avere a che fare con poche o centinaia di topics. Nella classificazione del sentimento, invece, abbiamo spesso poche classi (es. “positivo” “negativo”) che generalizzano su di diversi domini ed utenti.
Concetti come “forza del sentimento espresso” oppure “grado di soggettività” ed altri sono tipici della Sentiment Analysis.

2.1.1 Applicazioni della Sentiment analysis

E' comune classificare frasi in due principali categorie, riguardanti il loro grado di soggettività: frasi oggettive, che contengono informazioni basate su fatti, e frasi soggettive, che contengono credenze, opinioni, sentimenti e punti di vista riguardanti entità specifiche; la Sentiment Analysis si concentra nel riconoscimento di tali categorie, sfruttandone i contenuti di conseguenza.
I campi di applicazione della disciplina sono numerosi; alcuni possono essere raccolti nelle seguenti categorie.

- **Applicazioni a Website relativi a recensioni.**
  L'utilizzo di aggregatori automatici di recensioni potrebbe essere permesso grazie all'utilizzo di tali nuove tecniche; a riguardo invece dei siti tradizionali, che sollecitano l'utente alla redazione di recensioni, la Sentiment analysis potrebbe collaborare nel creare riassunti automatizzati delle opinioni degli utenti, come verificarne la veridicibilità (per esempio, quando un voto basso viene associato ad una recensione riconosciuta come positiva, identificando errori [18]).
- **Come una tecnologia ausiliaria.** Sentiment analysis assume un ruolo importante anche nelle vesti di tecnologia abilitante per altri sistemi.
  Una possibilità è quella di potenziare i sistemi di raccomandazione (per esempio, relativi a film o libri), eliminando i prodotti o servizi rilevati come negativi [19].
  Nei sistemi online che visualizzano pubblicità (ads) nelle barre laterali, può essere importante determinare qualora le pagine contengano contenuti informativi sensibili, e quindi inappropriati al fine dell'accoppiamento con segnali pubblicitari [20]; per sistemi più sofisticati potrebbe essere utile mostrare pubblicità quando sentimenti positivi sono rilevati, respingerle in caso opposto.
  E' stato anche discusso come l'estrazione di informazione possa essere migliorata eliminando i

contenuti trovati in frasi soggettive, e quindi probabilmente legate ad una opinione personale [21]. Uno studio interessante riguarda le citazioni nell'ambito di pubblicazioni scientifiche: lo scopo, in questo caso, sarebbe quello di determinare qualora l'autore citi un lavoro come supporto positivo o se ne contesti i contenuti [22].

- **Applicazioni Business Intelligence.** L'aspetto insito nella disciplina riguardante le applicazioni di intelligence rende la Sentiment Analysis predisposta a tali categorie di utilizzo. Attività di Business Intelligence includono per esempio la scoperta dei fattori che causano la vendita di un prodotto; tecnologie nate per estrarre opioni da documenti non strutturati redatti da esseri umani rappresentano strumenti eccellenti a tale scopo.
  Twitter e Facebook rappresentano dei punti focali di molte applicazioni nel campo della Sentiment Analysis: in questo contesto, la possibilità di monitorare la fama di un brand o di un prodotto rappresenta un comune obiettivo di molte implementazioni [23].
- **Applicazioni di analisi di mercati finanziari.** Lo studio del legame fra sentimenti espressi ed indicatori economici rappresenta una interessante applicazione. Esistono numerosi articoli, blog e tweets riguardanti ogni compagnia pubblica. Un sistema di sentiment analysis può quindi utilizzare queste fonti alla ricerca di articoli che discutano le compagnie e aggregando il sentimento verso di esse in un singolo punteggio, utilizzabile per svolgere analisi. Un sistema di questo tipo è per esempio The Stock Sonar [24]; il sistema visualizza graficamente il sentimento giornaliero positivo o negativo relativo ad ogni indice, insieme al grafico dell'andamento del prezzo dell'indice stesso.

2.1.2 Tecniche e strumenti della Sentiment Analysis

In figura 5 si riporta l'architettura generale di un sistema per la Sentiment Analysis.
L'input è rappresentato da documenti testuali, il Corpus, in un qualsiasi formato (PDF,XML,HTML,Word e molti altri).
I documenti vengono poi convertiti in testo e così processati utilizzando tecniche e strumenti linguistici, come lo stemming, l'etichettamento delle parti del discorso, la riduzione in token, ecc.

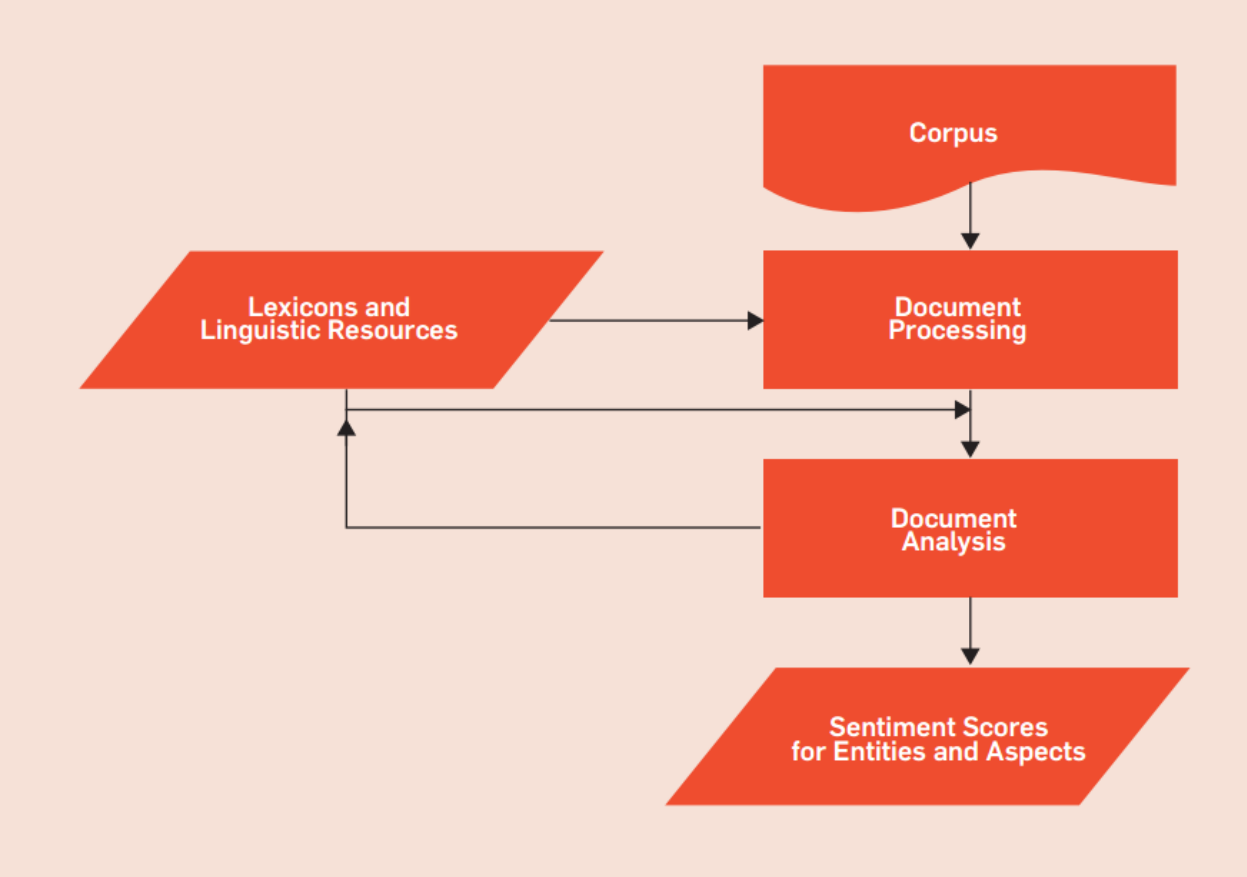

Figura 5 : Architettura generale di un sistema per la Sentiment Analysis

Il sistema, spesso, fa uso di vocabolari e risorse linguistiche, al fine di discriminare termini od co-occorrenze di termini per i quali sia verificato un livello di importanza o di

soggettività particolare.
La parte centrale del sistema è il modulo relativo all'analisi del documento preprocessato, *Document Analysis*, che si occupa di annotare sul documento in ingresso stesso le informazioni relative all'analisi del sentimento, facendo uso dei vocabolari e delle risorse linguistiche del blocco precedente. Tali annotazioni possono essere relative al documento nella sua interezza, analisi *document-level*, alle frasi prese individualmente, analisi *sentence-level*, o ad aspetti specifici o entità preselezionate, analisi *aspect-based*.
I documenti così annotati rappresentano il prodotto del sistema, che potrà poi essere analizzato ed utilizzato in innumerevoli contesti.
Di seguito vengono analizzate le 3 forme di Sentiment Analysis citate, più altre 2 di importanza rilevante.

- **Analisi Document-level.** Rappresenta la forma più semplice di analisi del sentimento, e si basa sull'assunzione che il documento contenga una opinione su di una entità principale, espressa dall'autore del documento stesso. Esistono due approcci principali al fine di condurre tale analisi: addestramento supervisionato oppure non supervisionato.
  L'approccio supervisionato assume l'esistenza di un insieme finito di classi entro le quali il documento debba essere classificato, ed un insieme di training è disponibile per ogni classe. Il caso più semplice ricade nella classificazione binaria (per esempio, sentimento positivo o negativo); semplici estensioni aggiungono una terza classe (neutro) oppure possiedono una scala numerica discreta entro la quale il documento debba essere posizionato (es. il sistema a 5 stelle utilizzato nelle recensioni di Amazon). La classificazione ricalca un classico processo di mining. Varie metodologie sfruttano anche vocabolari relativi al sentimento, tagging di parti del discorso e così via.
  Gli approcci non supervisionati sono basati sul determinare l'orientamento semantico *SO* (semantic orientation) di frasi specifiche contenute all'interno del documento; se il *SO* medio di queste frasi supera una

certa soglia allora il documento viene classificato come positivo, altrimenti negativo. La selezione delle frasi avviene attraverso uno di due principali approcci: un insieme di pattern di etichettamenti di parti del discorso può essere utilizzato, scegliendo le frasi che facciano match con tali tagging; oppure viene utilizzato un vocabolario contente parole (o addirittura frasi o parti di frasi) relative ad una espressione di sentimento, selezionando parti del documenti contenenti tali strutture.

Un metodo classico per determinare il *SO* di una certa parola oppure frase è di calcolare la differenza fra il *PMI* (Pointwise Mutual Information) della frase e quello di due termini esplicanti un sentimento ben preciso [31]. $PMI(P,W)$ misura la dipendenza statistica fra la frase P e la parola W basandosi sulla loro co-occorrenza in un corpus dato oppure sul Web (utilizzando ricerche Web). Le due parole utilizzate nel lavoro di Turney [31] sono 'excellent' e 'poor'. Il SO restituisce una misura di quanto P sia vicino in termini di significato alla parola positiva ('excellent') o alla parola negativa ('poor').

- **Analisi Sentence-level.** Un singolo documento può contenere diverse opinioni riguardanti anche le stesse entità. Se lo scopo è quello di ottenere una visione più fine a riguardo delle diverse opinioni espresse a proposito delle entità di interesse è necessario svolgere una analisi sentence-level.

  Assumendo di conoscere l'identità delle entità discusse nella frase in esame, ed assumendo che per ogni frase sia contenuta una opinione singola (vincolo rilassato dal poter dividere le frasi ulteriormente), diviene necessario determinare il livello di soggettività delle frasi stesse; solo le frasi contenenti informazioni soggettive verrano poi analizzate in termini di recupero della polarità del sentimento espresso (alcuni approcci analizzano anche frasi oggettive, di complessità crescente).

  La maggior parte dei metodi utilizzano approcci supervisionati al fine di classificare le frasi binariamente [32]; un approccio unico basato su tagli minimi dei grafi è stato proposto da Pang e Lee [33]. La

premessa fondamentale del loro approccio è che frasi vicine fra di esse dovrebbero essere caratterizzate dalla stessa classificazione in ambito di livello di soggettività.
Dopo aver evidenziato e selezionato le frasi soggettive è possibile procedere alla classificazione delle stesse in, per esempio, "positive" e "negative", secondo approcci supervisionati e non (gli ultimi simili a quelli visti in [31]).
Recenti ricerche [34] mostrano come sia consigliabile trattare tipi diversi di frasi attraverso diverse strategie; tali particolari frasi sono quelle condizionali, interrogative e sarcastiche.

- **Analisi aspect-based.** I due precedenti approcci risultano performanti ed efficienti quando l'intero documento o ogni frase si riferisce ad una singola entità; in molti casi le persone discutono a proposito di entità che possiedono molteplici aspetti (attributi) e sostengono diverse opinioni a proposito di ognuno di questi. Questo capita spesso nelle recensioni di prodotti o in forum di discussioni dedicati a categorie di prodotti (come per esempio automobili, smartphones). Un esempio comune è la recensione di un prodotto tecnologico: diversi possono essere gli attributi dell'oggetto, come la velocità computazionale, il design, la durata della batteria; l'utente esprime diverse opinioni su di ognuno di questi. Classificare quindi una recensione di questo tipo in maniera binaria eliminerebbe molto del contenuto informativo presente.
  L'analisi aspect-based (conosciuta anche come feature-based) è il problema di ricerca che focalizza l'attenzione sul riconoscimento di tutte le espressioni di sentimento all'interno di un certo documento e gli aspetti verso i quali si riferiscono.
  Un approccio classico, utilizzato da molte compagnie commerciali, al fine di identificare tutti gli aspetti toccati in una collezione di recensioni, è quello di estrarre tutte le noun phrases *NPs* (ossia frasi che iniziano con un nome o pronome indefinito, o che svolgono la stessa funzione grammaticale di tali frasi) e

mantere solo le *NPs* la cui frequenza supera una soglia determinata sperimentalmente [35].
Un altro approccio si propone ridurre il rumore nelle NPs trovate [36]: l'idea principale è quella di misurare per ogni NPs candidata il PMI con frasi strettamente correlate alla categoria del prodotto in esame. Solo quelle caratterizzate da PMI sopra una certa sogli vengono mantenute.
Gli approcci elencati si propongono di ritrovare gli aspetti definiti esplicitamente nel testo; esistono però aspetti non espressi in maniera diretta, detti impliciti, ma che possono essere inferiti dalle espressioni di sentimento che li menzionano in maniera implicita (come ad esempio il peso di un telefono nella frase "il telefono è leggero").
Una modalità attraverso la quale inferire aspetti impliciti è suggerita da Liu [37], dove un approccio di mining che sfrutta regole associative è utilizzato per accoppiare aspetti impliciti (espressioni di sentimento) con aspetti espliciti.
Ottenuti gli insiemi contenenti frasi riguardanti aspetti espliciti ed impliciti è possibile utilizzare un semplice algoritmo the determina la polarità di ogni espressione, basandosi su di un vocabolario per la sentiment analysis, considerando termini che esprimono negazione e congiunzioni avversative; la polarità finale relativa ad ognuno degli aspetti è determinata da una media pesata delle polarità di tutte le espressioni pesate in maniera inversa dalla distanza fra l'aspetto e l'espressione stessa.

- **Comparative sentiment analysis.** Una possibile traduzione è Analisi del Sentimento Comparativo: tale analisi fa riferimento ai casi in cui gli utenti non sviluppino una opinione diretta su di una entità, bensì delle espressioni comparative fra l'entità stessa ed altre. L'obiettivo di questi sistemi di analisi è quello di identificare le frasi che contengono opinioni comparative, ed estrarre le entità che emergono positivamente in ogni opinione.
  Prime ricerche in tale ambito sono state fatte da Jindal e

Liu [38]; in questo lavoro è stato verificato come usare un relativamente ristretto insieme di parole sia in grado di coprire il 98% di tutte le opinioni comparative espresse in genere. Le parole sono:

- Aggettivi ed avverbi comparativi come: 'more','less', e parole terminanti con il suffisso –er (per esempio, 'lighter')
- Aggettivi ed avverbi superlativi come:'most','least', e parole terminanti con il suffisso –est (per esempio, 'finest')
- Frasi e termini addizionali come 'favor', 'exceed', 'outperform', 'prefer', 'than', 'superior', 'inferior', 'number one', 'up against'.

Dal momento in cui tali parole portano ad un recall molto alto, ma una precisione molto bassa, un classificatore naive Bayes viene utilizzato per eliminare le frasi che non contengono opinioni comparative.
Un semplice algoritmo utilizzato per identificare le entità preferite basandosi sul tipo di struttura sintattica comparativa utilizzata e la presenza di negazione è descritto da Ding et al in [39].

- **Acquisizione di un vocabolario per la Sentiment analysis.** Il vocabolario utilizzato nelle varie fasi di analisi è la risorsa più importante per la grande maggioranza di algoritmi di sentiment analysis. Esistono 3 opzioni attraverso le quali ottenere un vocabolario:
  - Approcci manuali, per i quali i termini vengono elencati a mano
  - Basati su dizionari pre esistenti, per i quali un insieme di parole viene espanso utilizzando risorse come WordNet. L'insieme di parole di partenza viene costruito includendo termini adatti all'analisi nel dominio di interesse; l'espansione viene poi realizzata sfruttando, per esempio, i sinonimi e contrari estratti da WordNet.
    Un algoritmo elegante è proposto da Kamp et al [40]; il metodo definisce la distanza $d(t_1, t_2)$ fra due termini t_1 e t_2 come la lunghezza del cammino più

breve fra t_1 e t_2 in WordNet. L'orientamento, in termini di sentimento, di t è definito come $SO(t) = \frac{d(t,bad)-d(t,good)}{d(good,bad)}$. Con $|SO(t)|$ si identifica l'intensità del sentimento espresso da t: a $SO(t) > 0$ consegue che t è positivo, negativo per il caso opposto.
Lo svantaggio principale degli algoritmi basati su dizionari è che il vocabolario costruito è indipendente dal dominio di interesse, e perciò non cattura le peculiarità di un dominio specifico.

- Basati su corpus di documenti, per i quali un insieme di parole viene espanso utilizzando una collezione di testi facenti parti di un preciso dominio; essi sono utilizzati al fine di creare vocabolari specifici per un certo dominio di interesse.
  Un approccio classico [41] in questo ambito introduce il concetto di sentiment consistency che permette di identificare aggettivi addizionali che hanno una polarità consistente e di utilizzarli come insieme di termini iniziali, da espandere. Un insieme di connettori linguistici (AND,OR,NEITHER-NOR,EITHER-OR) vengono utilizzati per cercare aggettivi che sono connessi ad altri per cui la polarità è nota. Per eliminare rumore residuo, l'algoritmo crea un grafo di aggettivi utilizzando connessioni indotte dal corpus di documenti e dopo uno step di clustering vengono formati i gruppi di termini positivi e negativi.

2.2 Stock market prediction

Per ovvie ragioni, la capacità di predire l'andamento degli indici di borsa attrae storicamente interesse sia dall'accademia che dagli azionisti.
Prime ricerche scientifiche erano basate su random walk theory ed Efficient Market Hypothesis (EMH) [25]:

- Con random walk theory si identifica la teoria finanziaria che sostiene come i valori del campo azionario evolvano seguendo un cammino casuale (random walk), e perciò non possano essere predetti.
- La EMH asserisce nella definizione di Fama [25] (1970) che un mercato finanziario è efficiente se in ogni istante il prezzo delle attività scambiate riflette pienamente le informazioni rilevanti disponibili per cui non sono possibili ulteriori operazioni di arbitraggio: la concorrenza garantisce che i rendimenti delle attività siano ai loro livelli di equilibrio (eguaglianza tra domanda e offerta). In un mercato finanziario siffatto né l'analisi tecnica (previsione dei prezzi futuri basata sullo studio dei prezzi passati) né l'analisi fondamentale (studiando l'andamento del valore delle imprese attraverso l'analisi della redditività si tenta di capire se esistono nuove prospettive sul valore delle azioni) possono consentire ad un investitore di conseguire profitti maggiori di quelli che un altro investitore otterrebbe detenendo un portafoglio di titoli scelti a caso, con il medesimo grado di rischio

La conseguente crescita di studi di ricerca sul settore viene incorporata da discipline più recenti quali Economia e Finanza Comportamentale, ed in teorie come la Socionomic Theory of Finance (STF), che vanno ad esaminare criticamente la teoria dell'EMH.

2.2.1 Economia comportamentale

La finanza comportamentale e l'economia comportamentale sono campi di studio strettamente legati, che applicano la ricerca scientifica nell'ambito della psicologia cognitiva alla comprensione delle decisioni economiche e come queste si riflettano nei prezzi di mercato e nell'allocazione delle risorse.
Numerosi trattati evidenziano come i valori degli indici dei mercati azionari non seguano un cammino completamente

casuale, e possano per cui essere predetti con un certo livello di affidabilità [26,27,28,29]. Tali variazioni venivano classicamente associate prevalentemente al sovvenire di nuove informazioni, per natura imprevedibili; grazie ai social network, al microblogging e quindi agli online social media in genere diventa possibile estrarre indicatori relativi al sopraggiungere di news, includendo perciò, perlomeno in parte, la capacità di predirne la presentazione (per esempio, le query di ricerca Google sono state analizzate recuperando indicatori preventivi relativi a diffusione di malattie [30]).

Allo stesso modo, lo stato emotivo globale (public mood) può giocare un ruolo equiparabile alle notizie nell'atto della previsione degli andamenti degli indici di borsa. Da ricerche di carattere psicologico è affermato come le emozioni, congiunte alle informazioni, vestano un ruolo di primaria importanza nel procedimento decisionale dell'essere umano [42,43,44]; di conseguenza, studi in Finanza Comportamentale comprovano come lo stato emotivo guida decisioni in ambito finanziario [45]; in tal senso sono 2 le conclusioni che derivano dallo studio:

- Il social mood determina la tipologia di decisioni prese da consumatori, investitori e manager di corporazioni
- Dal momento che le caratteristiche delle attività di business seguono l’andamento del social mood, le variazioni del mercato azionario divengono utili nel predire future attività economiche e finanziarie

Diventa quindi ragionevole sostenere la teoria secondo la quale lo stato emotivo globale possa guidare le variazioni in ambito borsistico tanto quanto l'incombere di notizie ed avvenimenti mediatici.

2.3 Analisi di “Twitter mood predicts the stock market”

Le fondamenta del lavoro presentato nella pubblicazione scientifica dal titolo “Twitter mood predicts the stock

market” [46] risiedono negli studi riguardanti economia comportamentale, che spiegano come le emozioni possano incidere fortemente nel decision-making individuale, con particolare interesse alle scelte di compra vendita in campo finanziario.

Lo scopo dello studio è quello di investigare la possibilità tramite la quale sia possibile estendere tale assunto, da una prerogativa posseduta da un singolo individuo ad una caratteristica della società intesa come insieme di persone, tramite l'analisi dello stato emotivo delle collettività, intese come nuovo soggetto; viene esplorata quindi la correlazione fra indicatori di stato emotivo della società nel suo insieme ed andamento dell'indice economico DJIA (Dow Jones Industrial Average) nel tempo, nonchè le capacità predittive dei primi sul secondo.

2.3.1 Strumenti per la raccolta di informazioni

Essendo lo scopo quello di studiare come lo stato emotivo pubblico influenzi l'andamento degli indici di borsa, diviene di fondamentale importanza la capacità di recupero di tale informazione in tempi brevi e con frequenza elevata; è quindi necessario disporre di strumenti che permettano la raccolta e l'estrazione di indicatori dello stato emotivo pubblico da entità disponibili gratuitamente e che contengano una grande quantità di dati, in grado di ricoprire campioni di pubblico di elevata densità. Gli ultimi 6 anni di ricerca, nell'ambito dell'individuazione di tali indicatori e della ricostruzione quindi di rappresentazioni del public mood, sono stati caratterizzati da progressi significativi per quanto concerne le tecniche facenti utilizzo di social media come oggetto da cui estrarre i dati utili alla rappresentazione dello stato emotivo pubblico: in tale contesto vengono in particolare sfruttati blog e, principalmente, feed Twitter di grandi dimensioni. Sebbene un solo tweet contenga al più 140 caratteri testuali, l'aggregazione di milioni di tweets, pubblicati a qualsiasi orario, permette di ottenere un corpus

di dati testuali in grado di rappresentare stato emotivo e sentimento in maniera globale.

La collezione di tweets utilizzata nell'ambito dell'esperimento in analisi raccoglie 9.853.498 tweets pubblici, pubblicati da circa 2.7 milioni di individui durante il periodo che va dal 28 Febbraio 2008 al 19 Dicembre 2008; essa contiene dati testuali di varia provenienza geografica, quindi espressi in lingua eterogenea, prevalentemente anglofona. Per ogni tweet contenuto nel corpus vengono riportati un identificatore numero della specifica istanza, l'orario e la data di immissione e, naturalmente, il contenuto testuale per esteso.

I dati grezzi così ottenuti necessitano di una fase di preparazione all'analisi, o preprocessing; in primo luogo viene rimossa qualsiasi tipologia di punteggiatura e tutti i termini rinvenuti all'interno di una lista di generiche stop-words, il cui contenuto non è ulteriormente specificato; dopo questo filtraggio preliminare i tweets vengono raggruppati per giorno di inserimento. Nel contesto di cui trattiamo, lo scopo è quello di estrarre indicatori di stato emotivo da tale collezione di testi; vengono considerati perciò unicamente i tweets di lingua inglese per i quali sia possibile sostenere un contenuto rivolto alla esternazione del sentimento o stato emotivo proprio dell'autore. Perciò vengono mantenuti in maniera esclusiva i tweets all'interno dei quali siano rinvenute le seguenti espressioni: “i feel”,”i am feeling”,”i'm feeling”,”i dont feel”,”I'm”,”Im”,”I am” e “makes me”; tale lista è esaustiva. Dall'insieme ridotto così ottenuto, allo scopo di filtrare elementi di disturbo di tipologia pubblicitaria o facenti riferimento a spamming, vengono eliminati anche i tweets contenenti le espressioni regolari “http:” oppure “www.”.

2.3.2 Analisi del public mood

La collezione di tweets ottenuta e preparata all'analisi contiene quindi, per ogni giornata del periodo di studio,

l'insieme dei tweets pubblicati e coerenti alle operazioni di filtraggio.

La possibilità di descrivere lo stato emotivo globale viene esaudita dall'analisi di tale contenuto testuale, con lo scopo di costruire serie temporali in grado di associare ad ogni giornata per la quale si disponga di contenuto informativo un valore numerico rappresentativo dello stato emotivo globale, in termini generici di positività/negatività oppure facente riferimento ad una specifica sfumatura dello stato emotivo umano, come per esempio ansia o calma.

Il corpus ottenuto come risultato del preprocessing viene sottoposto all'analisi di due strumenti, costruiti per lo studio e la rappresentazione dello stato emotivo: *OpinionFinder*, il quale misura il mood come un rapporto fra termini positivi e termini negativi utilizzati, ed *GPOMS*, un algoritmo sviluppato dagli autori stessi dell'esperimento, che invece misura il mood estraendo indicatori capaci di costruirne 6 diverse serie temporali relative a 6 diversi suoi aspetti (calm, alert, vital, sure, kind e happy).

OpinionFinder, conosciuto anche con l'acronimo *OF*, è uno strumento, disponibile gratuitamente, utilizzato per effettuare sentiment analysis su testi di lingua inglese. Può essere utilizzato per studiare il livello di soggettività delle espressioni rinvenute all'interno di una frase, così come la polarità delle emozioni attribuite all'autore della stessa, in termini di positività o negatività generica del sentimento esternato [48]. Per fare ciò *OF* sfrutta un lessico interno, ossia un dizionario all'interno del quale vengono elencati i termini ai quali sia possibile attribuire un livello di soggettività ed una polarità, assegnabili di rimando all'individuo che ne fa uso esprimendosi testualmente. Oltre al presente esperimento, *OF* è stato utilizzato nell'ambito dell'analisi di collezioni di tweets di grandi dimensioni in altri contesti: uno dei suoi primi impieghi è descritto in “From Tweets to polls: linking text sentiment to public opinion time series” [47], esperimento che andava a verificare la similarità dei risultati ottenuti tramite sondaggi riguardanti lo stato emotivo globale e le serie temporali del

mood costruite analizzando collezioni di tweets, utilizzando *OpinionFinder*. In tale contesto viene sfruttato direttamente il lessico di *OF*, utilizzato in lavori precedenti [48,49,50]: per ogni giornata viene determinato il rapporto fra numero di tweets considerati positivi e numero di tweets considerati negativi, ottenendo così una serie temporale. Per la previsione dell'andamento dell'indice di borsa viene sfruttato come segue: vengono selezionati dal lessico i termini positivi e negativi marcati indistintamente come "weak" e "strong", ottenendo così una lista di 2718 termini positivi e 4912 termini negativi; ogni tweet viene controllato al fine di rilevare il numero di termini positivi o negativi contenuti in esso e facenti parte del lessico; ad ogni occorrenza di tali termini viene incrementato il conteggio dei messaggi positivi o negativi di una unità e calcolato il rapporto fra di essi (positivi/negativi) per ogni giornata, costruendo i valori della serie temporale. Il limite espresso da modelli unidimensionali di ricostruzione del mood come OF è quello di operare una distinzione binaria fra sentimento positivo o negativo [51], ignorando la struttura multidimensionale del sentimento umano, e quindi informazioni potenzialmente utili.

GPOMS è uno strumento creato appositamente dagli autori della ricerca al fine di catturare sfaccettature del sentimento umano non ottenibili tramite mezzi convenzionali. Esso è in grado di misurare lo stato emotivo umano in termini di 6 differenti dimensioni, nominate *calm, alert, vital, sure, kind* e *happy*; tali 6 fattori derivano direttamente dall'analisi di uno strumento psicometrico denominato *POMS* , Profile of mood states, un metodo semplice e rapido per identificare e quantificare stati affettivi particolari. Esso è un test che misura 6 stati dell'umore umano, dalla tensione ansiosa alla depressione e al senso di disorientamento; i risultati si dimostrano particolarmente utili per valutare pazienti con disturbi nevrotici o da stress, e per prevederne le risposte a vari approcci terapeutici. Ne esistono diverse tipologie e versioni: quella più recente ed interessante è il *POMS bipolar* [52,53], sulla quale viene costruito *GPOMS*, applicabile a soggetti privi di disturbi particolari. In tale

versione, il test consiste di 72 aggettivi che contribuiscono a definire i 6 fattori dello stato emotivo: i soggetti che vi si sottopongono debbono scegliere l'intensità con la quale hanno risentito di quel particolare stato dell'umore, solitamente attribuendo un valore numerico intero compreso fra 0 e 4; il valore di ogni fattore viene calcolato utilizzando tali valutazioni, che compongono le variabili di 6 diverse equazioni (scoring keys) la cui soluzione ne permette il calcolo. Per rendere tale questionario applicabile allo studio dei termini testuali utilizzati nei tweets, il lessico di 72 aggettivi viene ampliato a 964 termini ad essi associati analizzando le co-occorrenze dei termini in una collezione di 2.5 miliardi di 4-grams e 5-grams, elaborata e costruita da Google nell'anno 2006 manipolando circa mille miliardi di parole osservate in pagine pubbliche presenti sul web [54,55]. Il lessico così espanso permette di catturare una varietà straordinariamente ampia di espressioni, ricoprendo gran parte dei termini utilizzati abitualmente in linguaggio naturale; è così possibile effettuare una ricerca testuale che, trovando un termine appartenente al lessico dei 964 vocaboli del GPOMS, sia in grado di associarlo ad una delle 6 dimensioni del mood. In particolare, ogni termine utilizzato nei tweets che sia contenuto in un n-gram è collegato logicamente ad uno dei 72 termini originali del POMS, e collabora nella costruzione della rispettiva dimensione del mood, attraverso la scoring key, attraverso un peso, derivato dall'analisi della co-occorenza del termine stesso con il termine originale. Il valore di ogni dimensione è quindi determinato dalla somma pesata dei pesi di co-occorrenza dei termini rinvenuti nei tweets e presenti nel lessico GPOMS.

2.3.3 OF vs GPOMS

Al fine di comparare le serie temporali ottenute tramite *OpinionFinder* e *Google Profile Of Mood States*, viene effettuata una standardizzazione dei dati a *z*-scores: il procedimento riconduce una variabile aleatoria distribuita secondo una media μ e varianza σ^2, ad una variabile aleatoria

con distribuzione "standard", ossia di media zero e varianza pari a 1; prevede di sottrarre alla variabile aleatoria la sua media locale e dividere il tutto per la deviazione standard, all'interno di una finestra temporale di k giorni prima e dopo la data particolare del calcolo. Per esempio, lo z-score della serie temporale X_t, denominato Z_{X_t}è quindi definito come:

$$Z_{X_t} = \frac{X_t - \bar{x}(X_{t\pm k})}{\sigma(X_{t\pm k})}$$

dove $\bar{x}(X_{t\pm k})$ ed $\sigma(X_{t\pm k})$ rappresentano rispettivamente media e deviazione standard della serie temporale all'interno del periodo $[t-k, t+k]$. Grazie a tale standardizzazione è immediato comparare le serie temporali derivanti dall'utilizzo di *OF* ed *GPOMS*, poichè utilizzanti la medesima scala.

E' necessario verificare la capacità delle serie così costruite di catturare aspetti interessanti dello stato emotivo pubblico; per fare ciò esse vengono costruite applicando *OF* e *GPOMS* nel periodo di due mesi fra il 5 Ottobre 2008 ed il 5 Dicembre 2008. Tale periodo non è frutto di una scelta causale; esso racchiude eventi di grande interesse socio culturale e quindi di impatto emotivo non trascurabile, come le elezioni presidenziali degli Stati Uniti del 4 Novembre 2008 ed il giorno del Ringraziamento del 27 Novembre 2008. Ciò che è auspicabile è che le serie temporali riflettano in qualche modo tali significativi avvenimenti, in quanto la risposta emotiva globale aspettata è naturalmente degna di nota. I risultati grafici, espressi in z-scores, sono riportati in figura 6.

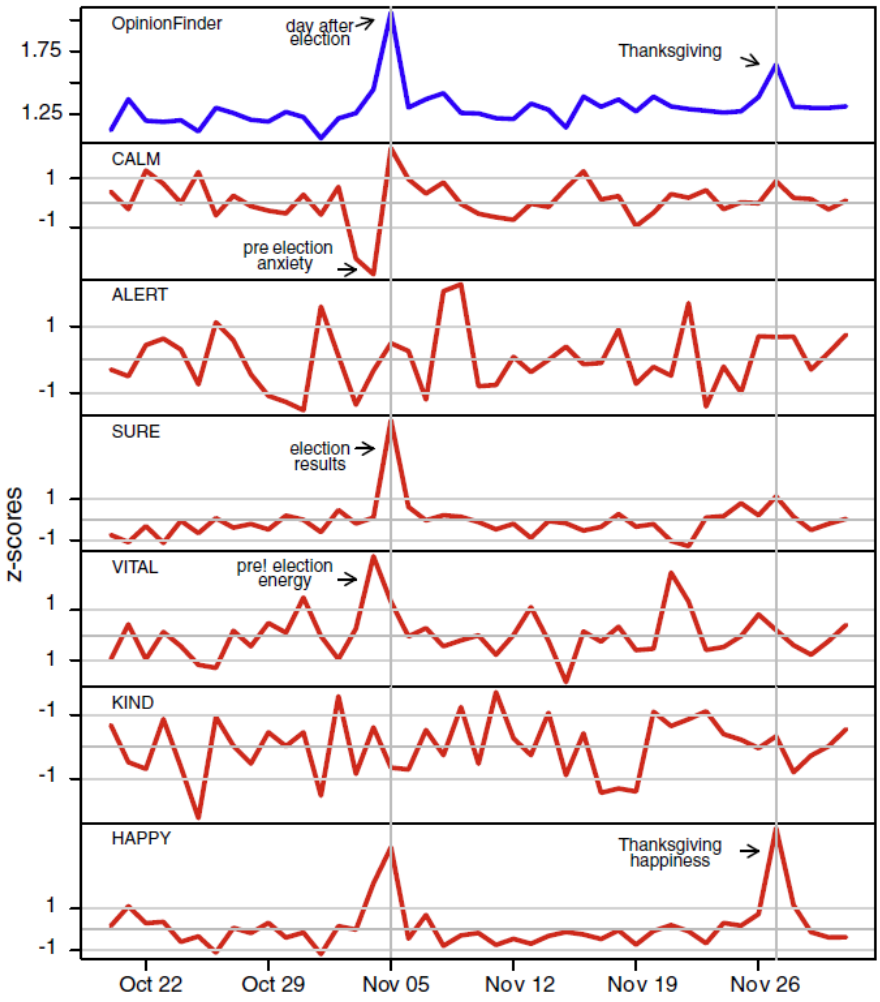

Figura 6 : Grafici delle serie temporali del mood da Ottobre, 2008 a Dicembre, 2008 e risposte emotive relative ad elezioni presidenziali ed il Giorno del Ringraziamento

La prima serie temporale, partendo dall'alto, è quella costruita utilizzando il lessico dell'OpinionFinder. Come evidenziato, tramite l'andamento della serie è possibile verificare una risposta emotiva ai due avvenimenti citati in precedenza: ciò che si nota è un picco di breve durata in corrispondenza delle elezioni e del giorno del Ringraziamento.
Le 6 serie temporali ottenute applicando il *GPOMS* restituiscono risultati variegati ed eterogenei, in quanto, come auspicato, ricoprono diverse sfaccettature dello stato emotivo umano. Prendendo come punto focale le elezioni

presidenziali, si nota una risposta emotiva diversa nei giorni adiacenti la data del 4 Novembre: in corrispondenza del 3 di Novembre vi è un calo significativo nei valori della dimensione *Calm*, il che riflette livelli alti di ansia pre-voto; il giorno delle elezioni è caratterizzato da una inversione di tendenza per quanto riguarda la dimensione *Calm*, evidenziando una riduzione dell'ansia generale, come da una crescita sostanziale per quanto riguarda i valori di *Vital*, *Happy* e *Kind*. A partire dal 5 Novembre in poi, i valori sopra la norma tendono a ristabilirsi. L'analisi delle reazioni al giorno del Ringraziamento permettono di osservare come lo stato emotivo reagisca a festività tipicamente gioiose: ciò che viene verificato è infatti un valore molto alto della dimensione *Happy*, sebbene limitato ad una sola giornata in quanto non si notano risposte significative nelle giornate precedenti ed antecedenti il 27 Novembre.

Il solo confronto grafico permette di sostenere come l'andamento dei valori della dimensione *Happy* sia similare a quello provveduto dalla serie costruita tramite *OpinionFinder*; per determinare quantitativamente le relazioni fra le 6 serie temporali costruite con *GPOMS* e quella con *OF* viene testata la correlazione fra di esse utilizzando una analisi di regressione multipla.
Il metodo della regressione (semplice) può essere esteso dal caso in cui si considera la variabilità della risposta della funzione Y in relazione ad una sola variabile indipendente X ad una situazione più generale in cui le variabili indipendenti siano più di una: il metodo è così detto regressione multipla ed è uno degli strumenti statistici più largamente utilizzati.

L'elaborazione eseguita secondo il metodo della regressione consente di adattare ai dati un'equazione lineare della forma:

$$Y_{OF} = \alpha + \sum_{i=1}^{N} \beta_i X_i + \varepsilon_t$$

In senso geometrico l'equazione rappresenta un iperpiano nello spazio multidimensionale. I dati da includere nel modello sono:

- $N = 6$, ed $X_1, X_2, X_3, X_4, X_5, X_6$ rappresentano rispettivamente le 6 serie temporali *GPOMS Calm*, *Alert*, *Sure*, *Vital*, *Kind* e *Happy*.
- α è l'intercetta, ossia il valore atteso di Y_{OF} qualora ogni elemento X_i sia uguale a zero.
- β_i sono i coefficienti di regressione multipla; essi misurano la variazione media di Y_{OF} quando X_i varia di una sola unità, e tutte le altre X sono tenute costanti. In virtù di questo significato i coefficienti β_i sono anche chiamati coefficienti di regressione parziale, per rimarcare la differenza nei confronti del coefficiente di regressione semplice lineare, che viene indicato come coefficiente di regressione totale.
- ε_t è l'errore statistico associato alla costruzione del modello.

I risultati relativi al test sono riportati in tabella 2.

Parameters	Coeff.	Std. Err.	*t*	*P*
Calm (X_1)	1.731	1.348	1.284	0.205
Alert (X_2)	0.199	2.319	0.086	0.932
Sure (X_3)	3.897	0.613	6.356	**4.25e−08*****
Vital (X_4)	1.763	0.595	2.965	0.004*
Kind (X_5)	1.687	1.377	1.226	0.226
Happy (X_6)	2.770	0.578	4.790	**1.30e−05****

Summary	Residual Std. Err.	Adj. R^2	$F_{6,55}$	*p*
	0.078	0.683	22.93	2.382e−13

* $p < 0.1$.
** $p < 0.05$.
*** $p < 0.001$.

Tabella 2: Risultati del confronto tramite regressione multipla fra serie OpinionFinder e le 6 dimensioni del mood generate da GPOMS

Sono riportati i valori dei coefficienti di regressione multipla, *Parameters*, di ognuna delle variabili indipendenti del modello, accompagnate dalla dicitura che ne rappresenta

la dimensione particolare sviluppata tramite GPOMS; sono in aggiunta inclusi numerosi indicatori che permettono di verificare la bontà di adattamento del modello e la significatività dei parametri stimati dall'esperimento:

- $Std.Err.$, errore standard, rappresenta la stima dello scarto quadratico medio dell'errore teorico del modello. Questa statistica riassuntiva misura l'esattezza o la qualità generale del modello multiplo valutata in termini di media/variabilità standardizzata non spiegata nella variabile dipendente che può essere dovuta a errori riconducibili alla costruzione delle serie temporali X_i. Fondamentalmente, quando tali errori sono contenuti, il valore dell'errore standard è piccolo, tendente a zero, e quindi il modello risulta utilizzabile; in caso contrario, con errori elevati, l'errore standard tende ad infinito, ed il modello non è utilizzabile.
- t rappresenta i risultati dei singoli t-tests effettuati su di ognuno dei coefficienti parziali ottenuti; mostra qualora esista o meno una relazione lineare significativa fra la variabile X_i ed Y_{OF}.
- p elenca i p-values, valori di significatività statistica, relativi ad ognuno dei coefficienti di regressione.
- $Adj.R^2$, o R^2 corretto, è una misura della bontà dell'adattamento (closeness of fit) del piano di regressione ai punti osservati. Vale a dire, più prossimo a 1 è il valore di R^2, più contenuta è la dispersione dei punti intorno al piano di regressione e migliore l'adattamento. Mostra la proporzione di variabilità di Y_{OF} spiegata da tutte le variabili indipendenti X_i, corretta per il numero di variabili di X_i utilizzate.
- $F_{6,55}$, è il risultato del così detto F-test, utilizzato al fine di calcolare la significatività globale del modello: mostra qualora sia presente un rapporto lineare fra tutte le variabili X_i collettivamente e Y_{OF}.

Dal momento che siamo in presenza di più di una variabile esplicativa, l'ipotesi nulla e quella alternativa sono:

1. I coefficienti β_i sono nulli; in tal caso non vi è una relazione lineare tra la variabile dipendente e le variabili esplicative.
2. Almeno un coefficiente $\beta_i$risulta diverso da zero; vi è una relazione lineare tra la variabile dipendente e almeno una delle variabili esplicative.

Il risultato dell' F-test, congiunto al p-value globale risultante minore di 0.05, risolve il problema di verifica di tali ipotesi, permettendo di rifiutare l'ipotesi 1.
L'analisi dei risultati indica come Y_{OF} sia significativamente correlata alle dimensioni $X_3(Sure)$, $X_4(Vital)$ ed $X_6(Happy)$, ma non ad $X_1(Calm)$, $X_2(Alert)$ ed $X_5(Kind)$, che invece risultano scorrelate e quindi contenenti informazioni differenti: in conclusione, alcune dimensioni del mood costruite tramite *GPOMS* si sovrappongono parzialmente ai valori dello stato emotivo pubblico ottenuti tramite l'analisi con *OpinionFinder*, ma esse non sono necessariamente tutte quelle utili all'evidenziare risposte emotive globali, come il raffronto precedente rispetto alle elezioni presidenziali ha dimostrato. L'algoritmo *GPOMS* permette quindi la costruzione di serie temporali che collaborano nell'ottenere una prospettiva unica di osservazione del mood globale, non catturata da metodi unidimensionali come, per esempio, *OF*.

2.3.4 Causalità di Granger del mood pubblico vs valori DJIA

Dopo aver verificato come le serie temporali costruite siano in grado di rispondere ad eventi sociali significativi, e quindi di riflettere lo stato emotivo globale, è necessario provare come queste siano correlate ai cambiamenti temporali degli indici di borsa, in particolare in relazione alle variazioni nei valori di chiusura dell'indice Dow Jones Industrial Average.

Il metodo applicato al fine di ottenere indicatori matematici di tale correlazione è quello della analisi di causalità di Granger. Esso incorpora una tecnica econometrica che fonda

sull'assunto secondo il quale se una variabile *X* causa una variabile *Y*, allora i cambiamenti in *X* debbono sistematicamente avvenire prima dei cambiamenti in *Y*. Il concetto, sviluppato da Clive Granger nel 1969 [54], mira a determinare in maniera statistica una causalità tra variabili espresse in un modello VAR (Vector Autoregression) che fondamentalmente incorpora un sistema di equazioni, rappresentanti modelli di regressione.
Formalmente una serie storica $\{x_t\}_t$ causa (nel senso di Granger) una serie storica $\{y_t\}_t$ se condizionando rispetto ai valori passati di x_t l'errore quadratico medio di previsione della $y+t$ risulta ridotto rispetto al caso in cui l'informazione relativa ai valori passati di x_t sia ignorata.

Quella che viene testata in questo contesto non è una vera e proprio causalità; bensì viene verificato qualora una serie temporale contenga o meno informazioni di carattere predittivo su di una seconda, in maniera simile a [57].

La serie temporale contenente i valori storici di DJIA, chiamata D_t , viene definita con lo scopo di riflettere le variazioni giornaliere dell'indice Dow Jones Industrial Average, al suo valore di chiusura: i suoi valori rappresentano quindi una variazione fra l'indice di chiusura di una data *t* e l'indice di chiusura della giornata precedente in data *t-1*: $D_t = DJIA_t - DJIA_{t-1}$. Tali valori storici vengono recuperati dal sito Yahoo! Finance, che permette di ottenere le statistiche relative alla borsa per qualsiasi anno desiderato. Si noti che non è sempre possibile calcolare una variazione di indice di chiusura fra una data e quella precedente, poichè naturalmente tali valori risultano disponibili unicamente in corrispondenza dei giorni di apertura della borsa stessa: la serie temporale costruita in questo contesto non contiene valori per i weekends così come in occasione di festività (vedi giorno del Ringraziamento): tali gap non vengono colmati estrapolando linearmente l'andamento dell'indice.

Viene comparata la varianza spiegata tramite i due modelli lineari seguenti:

$$L_1: D_t = \alpha + \sum_{i=1}^{n} \beta_i D_{t-i} + \varepsilon_t$$

$$L_2: D_t = \alpha + \sum_{i=1}^{n} \beta_i D_{t-i} + \sum_{i=1}^{n} \gamma_i X_{t-i} + \varepsilon_t$$

Il primo modello, L_1 , esprime la serie temporale utilizzando come variabili indipendenti unicamente i valori stessi di D_t ritardati temporalmente (ossia $D_{t-1}, \ldots, D_{t-n}$) per effettuare la previsione; il secondo modello L_2, invece, sfrutta sia i valori di D_t ritardati temporalmente che i valori delle serie temporali costruite tramite *OF* ed *GPOMS* (X_{t-i}).
Ciò che viene effettuato, applicando una analisi di causalità di Granger a tali modelli, è il confronto fra l'errore quadratico medio di previsione di L_1(che contiene solo informazioni storiche riguardanti i valori dell'indice di borsa) ed L_2 (che contiene anche informazioni sul mood pubblico, attraverso le serie costruite): una riduzione di tale errore in L_2 andrebbe a confermare l'ipotesi di correlazione tra la serie storica delle variazioni dell'indice DJIA nel tempo e gli indicatori dello stato emotivo pubblico generati da *OF* e *GPOMS*.
Il test viene effettuato utilizzando come periodo di analisi quello che va dal 28 Febbraio al 3 Novembre 2008, allo scopo di escludere eventi eccezionali quali le elezioni presidenziali ed il giorno del Ringraziamento, che manifestano risposte straordinarie da parte degli indicatori del mood pubblico: in tale periodo la serie storica delle variazioni dell'indice DJIA include unicamente 64 giornate/valori.

I risultati dell'analisi, mostrati in tabella 3, in termini di significatività della correlazione (p-values), mostrano come l'ipotesi nulla secondo la quale i coefficienti β_i siano eguali a zero possa essere rifiutata con un alto livello di confidenza per la serie temporale X_1(dimensione del mood Calm), in quanto presenta valori di $p<0.05$.

Lag	OF	Calm	Alert	Sure	Vital	Kind	Happy
1 Day	**0.085***	0.272	0.952	0.648	0.120	0.848	0.388
2 Days	0.268	**0.013****	0.973	0.811	0.369	0.991	0.7061
3 Days	0.436	**0.022****	0.981	0.349	0.418	0.991	0.723
4 Days	0.218	**0.030****	0.998	0.415	0.475	0.989	0.750
5 Days	0.300	**0.036****	0.989	0.544	0.553	0.996	0.173
6 Days	0.446	**0.065***	0.996	0.691	0.682	0.994	**0.081***
7 Days	0.620	0.157	0.999	0.381	0.713	0.999	0.150

* $p < 0.1$.
** $p < 0.05$.

Tabella 3 : Risultati dell'analisi di causalità Granger, in termini di p-values, fra dimensioni del mood ed andamento dell'indice DJIA, fra 28 Febbraio, 2008 e 3 Novembre, 2008

Quindi viene osservato come *Calm* sia relazionata alle variazioni dell'indice DJIA con la più alta causalità di Granger, per scostamenti temporali da 2 a 6 giorni. Le altre 5 dimensioni costruite tramite *GPOMS*, così come la singola generata da *OF*, non risultano correlate con le variazioni dell'indice.

2.3.5 Correlazione tra Calm ed DJIA

Per meglio visualizzare la correlazione tra la dimensione dello stato emotivo X_1 descritta come *Calm* e l'andamento delle variazioni del valore di chiusura dell'indice DJIA, si standardizzano i valori delle due serie temporali secondo z-scores (come già fatto in precedenza) e si visualizzano quindi su grafici utilizzanti la stessa scala, raffigurati in figura 7: i grafici riportati in figura spiegano l'andamento delle due serie in un periodo che va dal 1 Agosto 2008 al 30 Ottobre 2008.

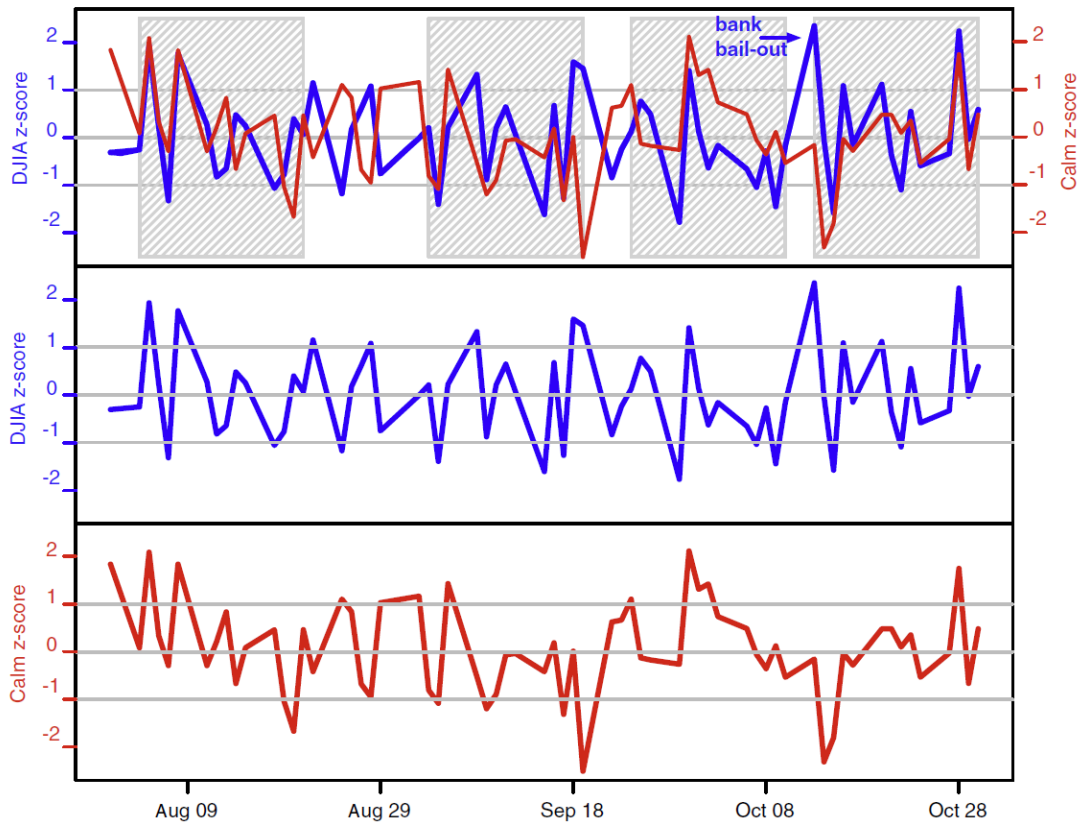

Figura 7: Grafici relativi all'andamento delle serie Clam e dell'indice DJIA, denotando periodi di correlazione

Entrambe le serie temporali coincidono frequentemente oppure hanno direzioni di diminuzione crescita dei valori riportati molto simili: nel primo grafico la serie temporale *Calm* è traslata in avanti di 3 giorni, con lo scopo di mostrare come valori passati della serie dello stato emotivo $X_1(t-3)$ siano in grado di predire andamenti similari nei valori di variazione dell'indice DJIA $(t-0)$. Nel secondo grafico vengono descritti i valori storici della serie delle variazioni DJIA, mentre nel terzo viene riportato l'andamento della serie *Calm* non ritardata.

La dimensione del mood denominata come *Calm* contiene quindi informazioni utili al fine della previsione dell'andamento dell'indice DJIA: nel periodo utilizzato nella comparazione in figura il p-value scende addirittura ad un valore di 0.009, con il ritardo di 3 giorni.

I casi in cui la serie ritardata fallisce nell'aderire alle variazioni dell'indice di borsa fanno emergere informazioni molto interessanti sul modello: in particolare, in corrispondenza del 13 Ottobre 2008, si nota una discrepanza significativa fra i due grafici, dove la serie temporale *Calm* rimane essenzialmente piatta, mentre le variazioni dell'indice di borsa subiscono un forte rialzo. In tale occasione la causa è altamente correlata ad un annuncio della Federal Reserve, evidenziando come le notizie non aspettate (e non prevedibili) giochino un ruolo talvolta fondamentale nell'atto predittivo, a maggior ragione se riguardante il mercato borsistico.

2.3.6 Un modello non lineare per la predizione

L'analisi di causalità Granger suggerisce una relazione predittiva fra alcune dimensioni del mood e l'andamento dell'indice DJIA nel tempo; tale analisi si basa però su modelli di regressione lineari, mentre la relazione fra il social mood ed i valori del mercato azionario è quasi certamente di tipo non lineare. Per meglio considerare questi effetti non lineari vengono esaminate le performance di un modello utilizzante reti neurali, in particolare Self-organizing Fuzzy Neural Network (SOFNN) [58], il quale è in grado di predire i valori DJIA sulla base di due insiemi di input:

- I valori della serie temporale DJIA per i 3 giorni precedenti al giorno della previsione
- I valori della serie temporale DJIA per i 3 giorni precedenti al giorno della previsione combinati a varie permutazioni fra le mood time series prodotte

L'utilizzo di reti neurali per la previsione di serie temporali non lineari che descrivono l'andamento degli indici del mercato azionario è stato verificato in studi precedenti [59,60]; le SOFNN sono progettate in maniera specifica per compiti di regressione, approssimazione di funzioni e problemi di analisi di serie temporali. Tale

modello richiede il settaggio di diversi parametri che influenzano le performance della previsione, ossia:

- $\delta = 0.04$, tolleranza all'errore; nei criteri che decidono quando aggiungere o meno un neurone alla rete, questo parametro sceglie la soglia dell'errore del modello, calcolato come la differenza fra l'output desiderato e quello attuale
- $\sigma_0 = 0.01$, pesi iniziali dei neuroni
- $k_{rmse} = 0.05$, errore quadratico medio aspettato sull'insieme di training
- $k_d(i),(i = 1,\ldots,r) = 0.1$, dove r è la dimensione delle variabili di input

Per valutare l'abilità del modello SOFNN nel predire i valori giornalieri della serie DJIA viene utilizzato come periodo quello che intercorre fra il 28 Febbraio 2008 ed il 19 Dicembre 2008, per l'insieme di training e l'insieme di test:

- L'insieme di training considera come periodo dal 28 Febbraio al 28 Novembre
- L'insieme di test dal 1 Dicembre al 19 Dicembre
- Il periodo di test è stato appositamente scelto poichè caratterizzato da una fondamentale stabilizzazione della serie DJIA, il cui andamento è raffigurato in figura 8, e per l'assenza di eventi socio culturali eccezionali

Figura 8 : Valori giornalieri dell'indice DJIA fra 28 Febbraio, 2008 e 19 Dicembre, 2008

L'analisi causale di Granger indica che solo la serie *Calm* (e, meno precisamente, *Happy*) è causa dei valori DJIA; questo non esclude che anche le altre dimensioni del mood possano contenere contenuto informativo di carattere predittivo se combinate con la serie *Calm* stessa. Per esempio, la serie *Happy* può non essere predittiva, ossia legata linearmente alle variazioni della serie DJIA, ma potrebbe in ogni caso migliorare le capacità predittive del modello SOFNN quando combinata con la serie *Calm*.
Per testare queste ipotesi, vengono considerata sette permutazioni delle variabili in input al modello SOFNN, la prima delle quali, I_0, rappresenta un modello di base addestrato unicamente utilizzando i valori storici dell'indice DJIA per i giorni $t-3$, $t-2$ e $t-1$.

$$I_0 = \{DJIA_{t-3,2,1}\}$$

$$I_1 = \{DJIA_{t-3,2,1}, X_{1,\ t-3,2,1}\}$$

$$I_{1,2} = \{DJIA_{t-3,2,1}, X_{1,\ t-3,2,1}, X_{2,\ t-3,2,1}\}$$

$$I_{1,3} = \{DJIA_{t-3,2,1}, X_{1,\ t-3,2,1}, X_{3,\ t-3,2,1}\}$$

...

$DJIA_{t-3,2,1}$ rappresenta i valori della serie temporale DJIA per i giorni $t-1, t-2$ e $t-1$, mentre $X_{1,\ t-3,2,1}$ rappresenta i valori

della dimensione 1 del mood generata dal GPOMS ($Calm$) per i giorni $t-1, t-2$ e $t-1$. Seguendo la stessa notazione $I_{1,2}$, $I_{1,3}$, $I_{1,4}$, $I_{1,5}$ ed $I_{1,6}$ rappresentano una combinazione dei valori storici di DJIA con la dimensione 1 del mood e la dimensione, rispettivamente, 2,3,4,5 o 6 insieme, per i giorni $t-1, t-2$ e $t-1$.
Viene anche considerata la combinazione di input che sfrutta i valori della serie del mood generata tramite $OpinionFinder$, allo scopo di confrontarne l'esito con le serie temporali costruite tramite $GPOMS$:

$$I_{OF} = \{DJIA_{t-3,2,1}, X_{OF,\ t-3,2,1}\}$$

L'accuratezza della previsione viene misurata in termini del Mean Absolute Percentage Error (MAPE) e della direzione, crescente o calante, della previsione stessa sul periodo di test; i risultati sono riportati in tabella 4.

Evaluation	I_{OF}	I_0	I_1	$I_{1,2}$	$I_{1,3}$	$I_{1,4}$	$I_{1,5}$	$I_{1,6}$
MAPE (%)	1.95	1.94	1.83	2.03	2.13	2.05	1.85	**1.79***
Direction (%)	73.3	73.3	**86.7**	60.0	46.7	60.0	73.3	80.0

Tabella 4: Previsione giornaliera dell'indice DJIA tramite SOFNN

L'accuratezza migliore della predizione, 86.7%, viene ottenuta utilizzando unicamente la serie I_1 come input. Considerando la numerosità del test set è possibile calcolare l'intervallo di confidenza dell'accuratezza ottenuta: questo ci indica con una confidenza scelta l'intervallo entro il quale la reale accuratezza del modello si trova. Per il periodo da 1 Dicembre, 2008 a 19 Dicembre, 2008 , sono disponibili 15 giornate di apertura del mercato di borsa, e quindi effettuabili al più 15 previsioni. L'intervallo risultante, per garantire una confidenza del 95%, è:

$$Intervallo\ di\ confidenza_{Calm} = [70.5\%, 87.9\%]$$

Capitolo 3
Strumenti

3.1 Sorgenti dati

I dati utilizzati dal sistema proposto sono:

- Una collezione di tweets, eterogenei per provenienza geografica e lingua utilizzata, pubblicati nel periodo da 1 Gennaio, 2008 a 19 Dicembre, 2008; in totale sono presenti 7861865 tweets, la cui distribuizione giornaliera è rappresentata in figura 9.

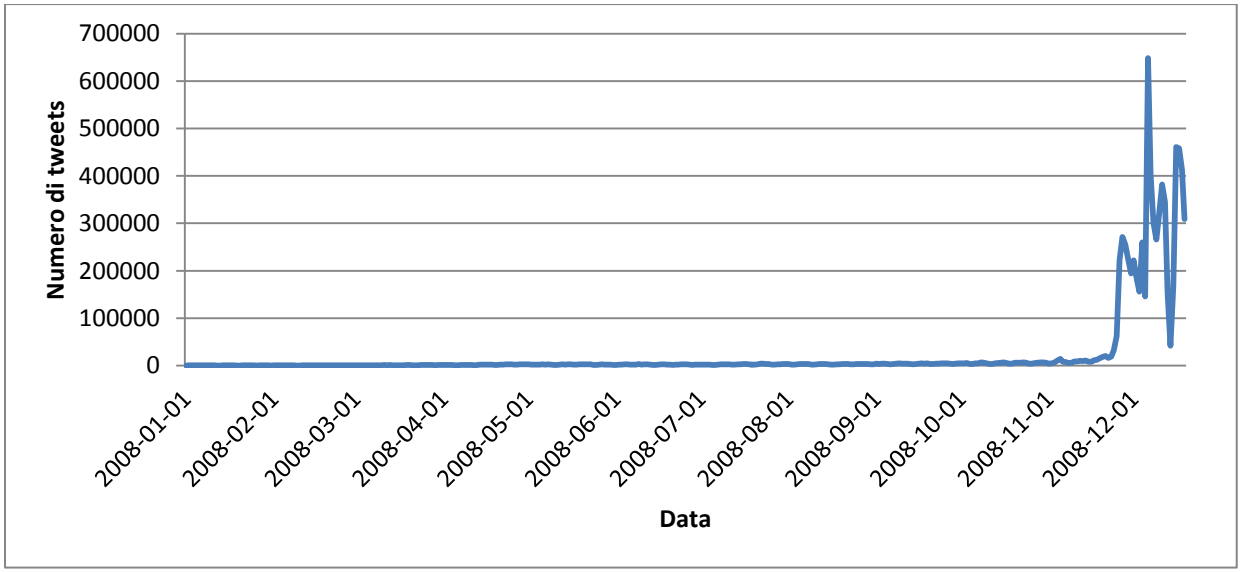

Figura 9 : Distribuzione giornaliera dei tweets nella collezione

Per ogni tweet sono disponibili le seguenti informazioni:

- Un identificatore univoco numerico del tweet stesso
- La data e l'orario di immissione relativo al fuso orario GMT+0

    - Il contenuto testuale, limitato per caratteristiche di Twitter a 140 caratteri

- I valori dell'indice Dow Jones Industrial Average giornalieri; questi vengono estratti da Yahoo! Finance, che consente di ottenere le informazioni relative alle quote degli indici desiderati su di un periodo storico, in formato csv. Quello che interessa è il valore giornaliero di chiusura dell'indice DJIA, più in particolare la sua variazione nel tempo; viene quindi costruita una serie temporale D_t i cui valori sono la differenza fra $DJIA_t$ e $DJIA_{t-1}$. I valori dell'indice non sono naturalmente disponibili nei giorni di chiusura del mercato di borsa: questi non vengono linearmente estrapolati, ma considerati costanti su tali periodi. Questo significa che la serie D_t conterrà valori per ogni giorno t di apertura della borsa, ossia per cui l'indice di chiusura sia disponibile; la variazione verrà calcolata al primo giorno precedente per il quale il valore dell'indice sia disponibile.

3.2 Weka

Weka (Waikato Environment for Knowledge Analysis) [56] è un software open source rilasciato con licenza GNU (General Public License) sviluppato presso l'università di Waikato in Nuova Zelanda. Essendo sviluppato completamente in Java questo software è utilizzabile su qualsiasi sistema operativo dotato di una Java Virtual Machine. Weka consiste in una collezione di algoritmi di machine learning (ovvero apprendimento automatico) nell'ambito di data mining, in particolare fornisce tramite interfaccia grafica vari tools per l'analisi dei dati e la creazione di modelli predittivi.
Questo sistema fornisce diversi strumenti di data mining come pre-processamento, classificazione, regressione, visualizzazione e selezione delle features. Tutte queste tecniche sono applicabili a dati in formato flat, ovvero ogni dato del data set è descritto da un numero fisso di attributi,

numerici o categorici. Le funzionalità rese disponibili da Weka sono utilizzabili sia tramite interfaccia grafica, sia richiamabili in progetti esterni java tramite l'utilizzo del file .jar .
Weka consente inoltre, grazie al JDBC (Java DataBase Connectivity), l'interfacciamento a database SQL, permettendo il processamento del risultato di una query su un database.

Sono disponibili quattro diversi ambienti operativi grafici:

- Explorer: permette l'analisi dei dati e l'applicazione di tecniche di Data Mining.
- Experimenter: versione batch di esplorer, offre la possibilità di eseguire esperimenti e test per l'analisi statistica.
- Knowledge Flow: offre la possibilità di automatizzare i processi di mining, definendo un determinato workflow per l'esecuzione di alcune funzionalità (es. caricamento di file, applicazione di filtri, etc).
- Simple CLI: utilizzo di Weka da linea di comando.

Capitolo 4
Framework concettuale

In questo capitolo verranno esposti i metodi sviluppati e le scelte fatte nell'ambito della creazione di un sistema per la previsione degli andamenti dell'indice di borsa Dow Jones Industrial Average di chiusura analizzando il contenuto testuale di tweets generici ed eterogenei; in particolare verrà descritto un approccio alternativo rispetto alla risoluzione del problema proposta da Bollen [1], ed alcune tecniche per la selezione e filtraggio dei tweets da utilizzare per la costruzione dei modelli di previsione.
Come descritto in precedenza, la classificazione testuale è il processo che approssima la funzione target f attraverso la costruzione induttiva di un classificatore di un dato data set. Fatto ciò, si assegnano documenti ignoti al modello utilizzando la funzione approssimata f; nel caso in esame i documenti a cui si fa riferimento sono costituiti da tweets, raggruppati secondo diversi criteri descritti dettagliatamente in 4.1.
La prima fase è chiamata apprendimento, la seconda classificazione.
Come di consueto nei processi di classificazione, si inserisce una fase preliminare, necessaria al trattamento ed alla preparazione dei dati, che necessitano di essere rappresentati in una maniera consona alle elaborazioni successive:

1. Pre-processing: viene creato un mapping del contenuto di ogni documento in una logical view, ovvero una rappresentazione degli stessi, che poi può essere utilizzata nell'algoritmo di classificazione. Varie operazioni testuali e statistiche sono utilizzate per estrarre il contenuto più importante di ogni documento.
2. Apprendimento/Classificazione: basato sulla rappresentazione dei documenti, rappresenta il vero algoritmo di apprendimento tramite un insieme di training di documenti e la successiva classificazione di documenti test.

Nell'ambito del framework sviluppato, la fase di pre-processamento dei dati, illustrata in figura 10, viene logicamente ulteriormente suddivisa in tre sotto sezioni:

1. Un processo di *preparazione e filtraggio dei dati*, ossia della collezione di tweets da analizzare, al fine di mantenerne i più interessanti coerentemente allo studio di correlazione da effettuare
2. Uno step di costruzione della *logical view testuale*, ancora non classificabile, che prevede il raggruppamento dei tweets su base giornaliera e multi-giorno, secondo diversi metodi descritti dettagliatamente
3. La costruzione delle bag-of-words a partire dalla *logical view testuale* del passo precedente; si opera quindi *term extraction*, ovvero l'estrazione di tutti i termini potenzialmente utili ai fini della rappresentazione finale dei raggruppamenti di tweets, che produce come artefatto la *logical view* finale, utilizzata nella fase di classificazione

Figura 10 : Suddivisione del preprocessing della collezione di tweets

Il data set ottenuto può essere ora utilizzato per la classificazione.
Nella metodologia proposta viene aggiunta una fase preliminare alla classificazione: in questa vengono sfruttate alcune tecniche di miglioramento del data set ottenuto, che hanno come scopo quello di rimuovere tweets dalle aggregazioni che contribuiscono in maniera errata alla predizione oppure di eliminare intere istanze, dagli insiemi utilizzati per l'addestramento degli algoritmi di classificazione, che causano la generazione di un modello di

classificazione non performante. Per questo il data set risulta suddiso in 3 ulteriori sottoinsiemi:

- *Training set*, primo periodo temporale, le cui istanze hanno classe conosciuta;
- *Test set*, secondo periodo di più breve durata, le cui istanze hanno classe conosciuta;
- *Validation set*, ultimo periodo dell'anno, sul quale viene effettuata la previsione finale, le cui istanze hanno classe sconosciuta sulla quale si valuta l'affidabilità della previsione

L'insieme t*raining set* viene utilizzato per addestrare il classificatore attraverso il quale analizzare *test set*, le cui istanze possiedono classe conosciuta, producendo le previsioni ed i relativi errori per ogni entry. Tali artefatti così prodotti verranno sfruttati secondo alcune metodologie qui sviluppate, al fine di migliorare la composizione dei tre insiemi sopra citati: analizzando gli errori di previsione vengono costituiti gruppi di istanze 'buone', ossia classificate correttamente, e 'cattive', ossia classificate erroneamente, che verranno poi comparate a *training*, *validation* e *test set*.
Infine, la classificazione finale viene effettuata addestrando un ultimo classificatore sfruttando *training* e *validation set* congiunti, analizzando le istanze di *test set*.

4.1 Preparazione e filtaggio dei dati

In questo contesto ciò che assume ruolo di primaria importanza è considerare il contenuto testuale dei tweets da un punto di vista semantico, ossia inferire quali messaggi possano o meno rappresentare uno stato emotivo, un sentimento posseduto dall'autore del testo stesso; tali saranno i tweets che in seguito verranno raggruppati giornalmente, o su più giorni, e poi sottoposti alla term extraction e quindi utilizzati per la costruzione della logical view, oggetto della conseguente classificazione.

Il data set contenente i tweets utilizzati nel procedimento, come già detto, contiene testi di provenienza geografica e linguistica differente; naturalmente la lingua inglese assume il ruolo principale, e viene utilizzata come riferimento per la totalità delle analisi testuali successive. Seguendo le indicazioni in tal senso di Bollen [46] , vengono in un primo step di preprocessing testuale mantenuti unicamente i tweets contenenti le seguenti espressioni regolari, facenti riferimento alla volontà, insita nell'autore del tweet stesso, di esternare opinioni, pensieri personali o stati emotivi:

- *i feel*
- *i am feeling*
- *i'm feeling*
- *i dont feel*
- *I'm*
- *Im*
- *I am*
- *makes me*

Vengono anche rimossi dal data set:

- I tweets contenenti riferimenti web esterni, onde evitare la considerazione di messaggi pubblicitari o di spam; vengono quindi rimossi quelli contenenti le espressioni regolari "*http:*" o "*www.*".
- Le espressioni regolari del tipo @<*user*> che rappresentano l'indirizzamento di un particolare contenuto verso un utente *user* del sistema di microblogging.

Al fine di considerare il maggior numero possibile di tweets di carattere soggettivo, e quindi contenenti indicazioni sullo stato emotivo dell'autore, viene considerato nel processo di selezione di cui sopra, come aspetto di novità nel lavoro qui proposto, il concetto delle *emoticons* (o *smileys*).
Le emoticons sono un insieme di simboli, prevalentemente di punteggiatura, ai quali vengono associati stati emotivi; essi vengono costruiti approssimando una espressione facciale relativa ad una certa emozione. Quando l'autore di un tweet utilizza una emoticon, egli sta annotando direttamente sul

proprio testo uno stato emotivo: in tal senso, un *sorriso* rappresenta generalmente uno stato emotivo positivo, mentre invece una espressione *triste* rappresenta uno stato emotivo negativo, mentre ancora il simbolo del cuore, <3, può rappresentare una ulteriore sfaccettatura dello stato emotivo. Vengono quindi considerate in questo contesto tre categorie di *emoticons*, le cui componenti specifiche sono elencate in tabella 5, le quali vengono individuate all'interno del corpus testuale e sostituite da tre keywords, che permangono oltre il filtraggio della punteggiatura e vanno ad etichettare ogni tweets con una delle tre categorie (liste).
Quindi i termini *emotHappy*, *emotSad* ed *emotHeart* vanno ad aggiungersi alle espressioni di cui sopra, onde mantenere i tweets che li contengono.

keyword	emoticons
emotHappy	:) :-) :D :-D (: (-:
emotSad	:(:-(:'(:'-( ):)-:)':)-':
emotHeart	<3

Tabella 5: Emoticons considerate e relative keywords

Infine, vengono filtrati tutti i caratteri non-testuali, quali punteggiatura, caratteri numerici o di alfabeto non inglese; in particolare viene utilizzato il codice ASCII di ogni termine, eliminando tutti quelli non compresi fra [a-z,A-Z].

I dati così preparati sono pronti ai raggruppamenti di cui il prossimo paragrafo.

4.2 Costruzione della logical view testuale

I tweets ottenuti al passo precedente vengono raggruppati sulla base della data di pubblicazione: tali raggruppamenti

rappresentano quindi il contenuto informativo testuale relativo ad una certa giornata, ed assumono il ruolo di unità fondamentale al processo di classificazione. A tali entità si attribuisce quindi la capacità di predire le variazioni future dell'indice DJIA, quindi successive alla data nella quale i tweets sono stati effettivamente pubblicati.
Come evidenziato nell'ambito dell'esperimento di Bollen et al, la correlazione fra andamento dell'indice DJIA e public mood, costruito tramite analisi del contenuto testuale dei tweets, è alta quando esiste un certo ritardo nella previsione: in particolare e in tal contesto la serie Calm approssima l'andamento del DJIA con maggiore accuratezza quando traslata temporalmente di 4 giorni (riferimento al paragrafo); inoltre, la previsione finale viene attuata addestrando il classificatore non considerando semplicemente il giorno precedente alla data in cui la previsione stessa fa riferimento, bensì utilizzando più giornate, in particolare 3.

Considerato ciò, diviene interessante considerare diversi modelli di aggregazione multi-giorno dei tweets raggruppati giornalmente, al fine di testarne le differenti capacità predittive nel diverso contesto qui esaminato di una classificazione puramente testuale. I modelli qui sviluppati considerano anche la disponibilità dei tweets della collezione raffrontata alla presenza o meno invece di un valore di variazione dell'indice DJIA, vincolato dall'apertura chiusura del mercato azionario; vengono esplorati diversi approcci, taluni che puntano a mantenere una popolazione elevata di istanze all'interno della logical view, altri che invece ne riducono la numerosità al minimo.
Per esempio un modello embrionale, costituente una possibile logical view, potrebbe considerare per la previsione dell'andamento dell'indice DJIA alla data *d* i soli tweets pubblicati in data *d-1*, oppure aggregando più giornate, considerando a ritroso anche le giornate *d-2*, *d-3* eccetera. Se venissero considerate tutte le giornate dell'anno per le quali sono disponibili tweets, la data *d* potrebbe essere di chiusura del mercato di borsa, e quindi non contenere un valore di variazione dell'indice DJIA; in tal caso può comunque essere

utile considerare all'interno della logical view tali elementi, etichettati propriamente con una diversa categoria.

Sullo sviluppo di tali considerazioni sono stati realizzati 3 modelli di raggruppamento dei tweets; le istanze facenti parte questi modelli sono connotate da una data *forecastDate* sulla quale avviene la previsione e sulla quale quindi fa riferimento la classe di appartenenza:

- *positive* se la variazione dell'indice DJIA risulta positiva sul periodo entro il quale i tweets vengono aggregati
- *negative* se la variazione dell'indice DJIA risulta negativa sul periodo entro il quale i tweets vengono aggregati

neutral se la variazione dell'indice DJIA non risulta disponibile, ossia la data per la quale si effettua la previsione è una giornata di chiusura del mercato di borsa (disponibile per il solo modello di priorità ai tweets, si veda oltre).

Ogni modello raggruppa i tweets su più giornate precedenti la data di previsione; questo viene effettuato impostando oppurtunamente il parametro *paramAggr* in ingresso all'algoritmo di produzione del modello stesso: con *paramAggr=0* verranno considerati per la previsione unicamente i tweets pubblicati nella giornata precedente *forecastDate* (sempre che questa possa essere inclusa dal particolare modello scelto), con *paramAggr=1* verrano considerati i tweets pubblicati nelle giornate *forecastDate-1* ed *forecastDate-2*, con *paramAggr=2* verrano considerati i tweets pubblicati nelle giornate *forecastDate-1*, *forecastDate-2* ed *forecastDate-3*, e così via. L'intervallo di aggregazione diviene quindi:

$$[forecastDate-1-paramAggr\ ,\ forecastDate-1]$$

L'aggregazione dei tweets su più date di pubblicazione non deve obbligatoriamente avvenire a partire da *forecastDate-1*: a tal scopo viene aggiunto il parametro *paramLag*, che opera al fine di “traslare” temporalmente i raggruppamenti relativi

alla previsione effettuata su *forecastDate*. La selezione delle giornate dalle quali attingere per recuperare i raggruppamenti di tweets da includere avviene quindi sull'intervallo

[*forecastDate-1-paramAggr-paramLag , forecastDate-1-paramLag*]

Tali modelli sono:

Modello di priorità ai tweets. Il primo modello inserisce nella logical view tutte le giornate incluse fra 1 Gennaio, 2008 e 19 Dicembre, 2008; ciò significa che vengono incluse istanze recanti date di chiusura del mercato di borsa, e conseguentemente classe *neutral*.

Modello di priorità all'apertura della borsa. Il secondo modello non inserisce nella logical view le istanze facenti riferimento, in termini di previsione, a giornate di chiusura del mercato di borsa; conseguentemente tutti gli elementi della logical view assumeranno classe *positive* oppure *negative*.

Modello di sola apertura borsa. Nell'ultimo modello tutte le istanze faranno riferimento a giornate di apertura del mercato di borsa e non verranno considerate istanze includenti tweets pubblicati in giornate di chiusura della borsa stessa. Per esempio, nel caso in cui il raggruppamento avvenga unicamente sul giorno precedente la data di previsione e non vi sia traslazione temporale non esisteranno istanze relative ai Lunedì.

Ogni istanza facente parte dei suddetti modelli viene etichettata inoltre con un attributo binario *weekend* che evidenzia qualora, all'interno dell'aggregamento multi-giorno effettuato, vengano considerati tweets pubblicati nelle giornate di Sabato e Domenica, assumendo un valore *true*, oppure no, assumendo dualmente un valore *false*.

4.3 Costruzione delle bag-of-words

I modelli ottenuti secondo le direttive esposte nel paragrafo precedente (aggregazioni di raggruppamenti di tweets giornalieri) vengono poi sottoposti ad un procedimento di trasformazione in Bag-of-Words, al fine di costruirne la *logical view* finale utilizzata nella fase di classificazione.

Nella categorizzazione testuale, come già esposto nei capitoli precedenti, uno dei maggiori problemi è l'alta dimensionalità delle feature tramite le quali viene creata la classificazione, ovvero tutte le differenti parole occorrenti nella collezione di documenti. Una riduzione di questa dimensionalità è necesarria per varie ragioni, la prima e più ovvia delle quali è per le performance del classificatore, in quanto un così elavato numero di feature all'interno del modello renderebbe il processo impraticabile in termini di complessità sia temporale che spaziale. Inoltre una buona riduzione della dimensionalità risulta vantaggiosa anche in termini di riduzione dell'overfitting [40], ovvero il fenomeno per il quale il classificatore viene sintonizzato più sui documenti specifici di training che sulle caratteristiche semantiche reali delle categorie, ed in termini di riduzione del rumore, ovvero la selezione di feature utili alla corretta classificazione.

La selezione dei termini (features) e il peso assegnato ad essi per ciascuna aggregazione di raggruppamenti giornalieri di tweets, viene effettuata sfruttando le funzionalità dei filtri Weka, in particolare del filtro *StringToWordVector*. Tale filtro effettua automaticamente la term extraction sulla base di alcuni parametri, scegliendo come features i termini più rappresentativi; permette di selezionare il numero di features da produrre, nonché scegliere come rappresentarne i pesi, secondo due opportunità:

- semplicemente considerando la presenza/assenza di un termine su di ogni aggreggazione di raggruppamenti giornalieri di tweets
- calcolando il *tfidf* per ogni termine

Al fine di effettuare un ulteriore filtraggio dei termini, il filtro rende possibile l'utilizzo di una Stop-word list, ossia una lista di termini superflui o generalmente inutili, che non vengono considerati nel procedimento di term extraction.
Un ulteriore filtro classico di preprocessamento utilizzabile nel contesto della trasformazione a bag-of-words è lo *Stemming*. Si tratta di un processo utilizzato per ridurre parole flesse al loro tema, il quale non deve necessariamente coincidere con la radice morfologica della parola: l'importante è che parole con una semantica strettamente correlata vengano mappate sullo stesso tema; i termini risultanti, quindi, possono essere troncati in maniera 'scorretta' da un punto di vista linguistico.
Le tecniche di stemming sono note in informatica dagli anni '60, Porter nel 1980 creò un raffinato metodo che si impose come metodo standard per lo stemming in inglese. In questo contesto l'algoritmo di stemming considerato si basa sul Lovins Stemming, sviluppato da Julie Beth Lovins nel 1968, il quale rappresenta il primo algoritmo di stemming per il quale fu pubblicata la descrizione in ambito scientifico. E' di tipologia affix removal, ossia rimozione dell'affisso, poiché appunto applica una serie di trasformazioni ad ogni termine cercando di rimuoverne prefissi e suffissi conosciuti; lo svantaggio principale relativo a questa tipologia di algoritmi di stemming è il prerequisito rappresentato dalla conoscenza a priori delle caratteristiche morfologiche della lingua attraverso la quale i termini sono espressi. Tale conoscenza viene espressa in Lovins tramite 294 suffissi, ognuno collegato ad una di 29 condizioni, e 35 regole di trasformazione; effettuando lo stemming di una parola viene cercato e rimosso un suffisso che soddisfa una certa condizione. Per esempio effettuando lo stemming del termine "nationally" si considerano due suffissi plausibili: "ationally" con la condizione "lo stem deve essere composto da più di 3 simboli" ed "ionally", privo di restrizioni; quindi viene prodotto lo stem "nat". In seguito viene applicata una regola di trasformazione, che ha lo scopo di trattare i casi in cui lo stem termina con consonanti doppie, oppure i termini con plurali irregolari eccetera.

4.4 Estrazione ed analisi dei gruppi di bontà dei tweets

L'innovazione proposta nel framework oggetto di questo lavoro di tesi sta nei metodi e gli algoritmi che, analizzando i risultati di classificazione (con classificatore addestrato su *training set*) emergenti dall'analisi del *test set*, hanno come scopo quello di filtrare i tweets sull'intero data set, discernendo fra tweets 'utili' alla classificazione oppure tweets considerati 'inutili'.
Per effettuare la classificazione/previsione occorre distinguere il procedimento seguito per modelli generati con priorità ai tweets, per i quali le istanze vengono etichettate sulla base delle 3 classi *positive*, *negative* e *neutral;* in questo caso è quindi una classificazione multi-classe.
Per questa tipologia di modelli viene utilizzata la strategia detta One-versus-all (OvA), attraverso quale ogni singolo classificatore viene addestrato per classe, al fine di distinguere quella classe specifica dalle altre; in questo contesto solo due classificatori vengono addestrati, in quanto non interessa classificare la classe *neutral*. La previsione viene quindi effettuata utilizzando ogni classificatore binario e scegliendo la classe predetta con probabilità/confidenza più elevata. In pratica vengono costruiti due classificatori:

- Il primo viene utilizzato per classificare istanze di classe *positive*; le istanze di classe *neutral* assumono un nuovo valore di classe *negative*
- Il secondo viene utilizzato per classificare istanze di classe *negative*; le istanze di classe *neutral* assumono un nuovo valore di classe *positive*

I risultati di classificazione sopra menzionati riportano in particolare per ogni istanza la classe dell'istanza stessa, ossia la variazione dell'indice DJIA relativo ad una certa data di predizione, e la classe predetta, ossia la classe calcolata dall'algoritmo analizzando tutti gli altri attributi, ossia aggregazioni di tweets rappresentate dai termini

ottenuti nella trasformazione a bag-of-words e tag di *weekend*; in tale contesto le coppie classe-predizione possono essere di 4 tipologie:

- *TruePositive*, identifica le istanze per cui l'andamento dell'indice DJIA cresce e che vengono classificate correttamente
- *TrueNegative*, identifica le istanze per cui l'andamento dell'indice DJIA è in ribasso e che vengono classificate correttamente
- *FalsePositive*, identifica le istanze per cui l'andamento dell'indice DJIA è in ribasso ma che vengono classificate erroneamente come *positive*
- *FalseNegative*, identifica le istanze per cui l'andamento dell'indice DJIA è in crescita ma che vengono classificate erroneamente come *negative*

Raggruppando istanze specifiche di tali previsione secondo le 4 categorie sopra descritte otteniamo 4 gruppi di bontà dei tweets, ognuno dei quali raccoglie aggregazioni di tweets che collaborano, nel contesto di uno specifico algoritmo di classificazione, all'identificazione di una certa coppia classe-predizione. In particolare, Le istanze facenti parte delle categorie *TruePositive* e *TrueNegative* rappresentano i tweets utili alla predizione, quindi 'buoni' ; le istanze facenti parte delle categorie *FalsePositive* e *FalseNegative* rappresentano i tweets che vengono male interpretati dall'algoritmo, e quindi 'cattivi'.
Ogni aggregazione di tweets è qui rappresentata come un vettore di dimensionalità equivalente al numero di features (numero di termini estratti ed attributi nominali); quindi le istanze facenti parte di tali 4 gruppi sono in realtà vettori, e quindi geometricamente confrontabili. Allo scopo di ottenere una misura della similarità fra aggregazioni di tweets rappresentate da vettori, oppure fra tweet singoli, rappresentati anch'essi naturalmente da vettori, ed aggregazioni, viene sfruttata la tecnica del *coseno di similitudine*.
Come già presentato nel paragrafo 1.2.4, relativo alle tecniche di classificazione per il Text Mining, un metodo

standard al fine di ottenere una misura di similarità tra documenti è quello di calcolarla in termini di cosine similarity, o coseno di similitudine; essendo le aggregazioni di tweets rappresentate nella bag-of-words come vettori, i cui pesi sono i *tfidf* dei termini rinvenuti oppure a valori binari, valutando semplicemente la presenza/assenza del termine nell'aggregazione, è possibile confrontarli calcolando il coseno dell'angolo compreso fra di essi.
Tale similarità viene sfruttata ampiamente in ambito di categorizzazione di testi, o di tweets, al fine di inferire se e quando il contenuto di essi riguardi un certo tema od argomento; in questo contesto viene utilizzata in maniera differente, sperimentando qualora essa possa dare una indicazione su di una proprietà comune, incarnabile nella capacità o meno di migliorare la classificazione.

I metodi che seguono hanno come scopo quello di migliorare il data set filtrando tweets considerati inutili o puntando a mantenere tweets considerati utili, confrontando tweets singoli o aggregazioni con le istanze facenti parte i 4 gruppi sopra descritti; possono perciò essere suddivisi in due categorie:

- **Metodi che confrontano tweets singoli con i 4 gruppi di bontà.** Il confronto avviene qui riconducendo ogni tweet singolo testuale alla sua rappresentazione vettoriale, conforme al *feature set* tramite il quale sono rappresentate le istanze facenti parte i 4 gruppi di bontà. Il vettore così ottenuto viene confrontato con ogni istanza appartenente ai 4 gruppi, anch'esse vettori, calcolandone il coseno dell'angolo compreso; naturalmente tweets singoli che non possiedono nessun termine facente parte di tale *feature set* verranno presentati come vettori composti unicamente da zeri. A seconda dei valori di similarità, rappresentati quindi dal coseno dell'angolo compreso fra i vettori, il tweet verrà incluso o meno nel raggruppamento; i criteri che regolano questa scelta, chiamati qui regole, vengono esaminati secondo due tipologie:

    - **RULETRUE.** Vengono mantenuti unicamente i tweets che manifestino una similarità media verso le istanze *TruePositive* e *TrueNegative* maggiore di una soglia *threshold*.
    - **RULEFALSE.** Vengono mantenuti unicamente i tweets che manifestino una similarità media verso le istanze *FalsePositive* e *FalseNegative* minore di una soglia *threshold*.
- **Metodi che confrontano aggregazioni di tweets con i 4 gruppi di bontà.** Il confronto avviene qui invece direttamente fra istanze rappresentanti aggregazioni di tweets in forma vettoriale, calcolandone il coseno dell'angolo compreso.

4.5 Previsione dell'indice DJIA

La classificazione/previsione finale viene effettuata utilizzando i data set migliorati al passo precedente, addestrando un classificatore sfruttando *training* e *test set* congiunti ed analizzando *validation set*. Il classificatore è qui necessariamente lo stesso utilizzato per la costruzione dei 4 gruppi di bontà, onde perseverare la linearità del metodo.

Capitolo 5
Architettura del sistema

In questo capitolo verrà descritta l'implementazione pratica del sistema, la quale ha permesso di effettuare le simulazioni e i test che verranno elencati nel capitolo successivo.
Un diagramma riassuntivo del comportamento del sistema, descritto nel capitolo precedente, è dato in figura 11.

5.1 Architettura

Un primo preprocessamento della collezione di tweets anonimizzati viene effettuato direttamente da linea di comando o con l'ausilio di tools per il text processing, quali Sublime Text 2, vista la dimensionalità della raccolta; in particolare i procedimenti descritti in paragrafo 4.1, ossia sostituzione di emoticon con parole chiave, filtraggio dei tweets sulla base delle espressioni personali ed eliminazione di elementi inpuri.

Il sistema è stato implementato attraverso il linguaggio Java, utilizzando come IDE Eclipse, facendo utilizzo delle API Weka per la classificazione e gestione di tutti i data sets. Le classi sviluppate sono state suddivise in differenti packages ricalcanti le varie fasi del metodo.

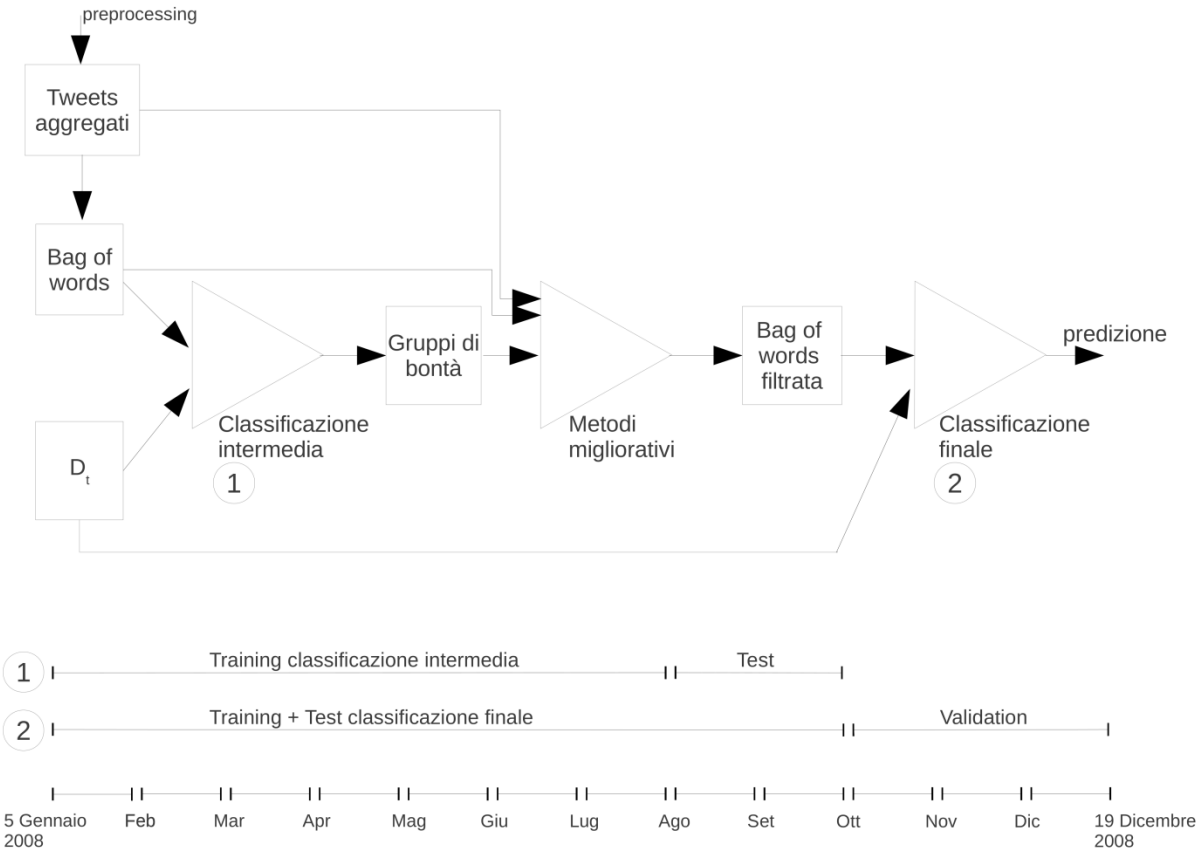

Figura 11 : Overview della metodologia e suddivisione temporale dei data set per le fasi di classificazione

Figura 4.1 evidenzia le fasi della metodologia successive al preprocessing dei dati che produce la Bag-of-words.

Package smp.datamodel. Contiene le classi utilizzate per la costruzione della *logical view testuale* e delle bag-of-words; riguarda il modello dei dati del sistema.

- DailyTweetsGrouper è la classe utilizzata per il raggruppamento dei tweets su base giornaliera e produce quindi una istanza per ogni data disponibile, contenente i tweets raggruppati
- Aggregator è la classe che si occupa di aggreggare le istanze ottenute dalla classe precedente sulla base delle 3 metodologie proposte in 4.2, costruendo la prima *logical view testuale*; sfrutta la serie temporale delle

variazioni del DJIA per etichettare le istanze secondo la previsione di variazione in crescita, classe *positive*, o in calo, classe *negative*. Permette di specificare i parametri che comandano la costruzione dei tre modelli: i già citati *paramAggr*, *paramLag*.
- BagOfWordsFiller si occupa della preparazione ed applicazione del filtro Weka StringToWordVector sulle istanze prodotte da Aggregator, producendo le bag-of-words e quindi la *logical view* utilizzata in ambito di classificazione. E' una classe fortemente parametrica, permettendo di specificare il numero di termini da estrarre dal testo, qualora effettuare stemming o meno su tali termini, la tipologia del peso associato a tali termini (*tfidf* oppure presenza/assenza).

Package smp.extr. Contiene le classi per l'estrazione dei 4 gruppi di bontà descritti in precedenza.

- GoodBadExtractor si occupa di estrarre i 4 gruppi di bontà, specificando le istanze rappresentanti la *logical view* da classificare, il classificatore da utilizzare per tale scopo ed una soglia sull'affidabilità di classificazione, utile per la selezione di istanze per le quali il classificatore classifica con probabilità più alta.

Package smp.discr. Contiene le classi utilizzate per effettuare il confronto fra tweets singoli ed aggregazioni di tweets con i 4 gruppi di bontà ottenuti in precedenza.

- TweetChooser confronta i vettori in ingresso, siano essi tweets singoli o aggregazioni di tweets raggruppati giornalmente, con i 4 gruppi di bontà, mettendo in pratica una regola di filtaggio. Rende possibile anche ottenere le distanze verso uno o più gruppi specifici.
- Discriminator utilizza TweetChooser per ripulire una *logical view*, secondo diverse strategie, adattando il metodo a seconda del tipo di aggregazione multi-giorno effettuata, le quali richiedono un diverso recupero, per esempio, dei tweet singoli utilizzati.

Package smp.cls. Rende disponibili diverse tipologie di classificazione finale, sia binaria che multi-classe, parametrizzando e selezionando una lista di classificatori onde operare test massicci.

Package smp.util. Fornisce operazioni accessorie, per il caricamento/salvataggio agile di file .arff, nonché la divisione di istanze su periodi e la gestione del parsing delle date temporali.

5.2 Preprocessamento dei dati

Il preprocessing testuale dei dati viene svolto filtrando le espressioni di carattere soggettivo e la presenza delle parole chiave relative alle emoticons considerate, nonché eliminando gli aspetti di disturbo descritti in precedenza.

Operando un filtraggio dei tweets disponibili sulla base delle espressioni soggettive considerate, che ricalca alla perfezione quella effettuata da Bollen [1], restituisce una distribuzione giornaliera migliorabile, illustrata in figura 12: considerando la collezione in esame, i primi mesi dell'anno sono caratterizzati da un numero di tweets molto basso ed il differenziale fra questi ed il mese di Dicembre permane esorbitante.

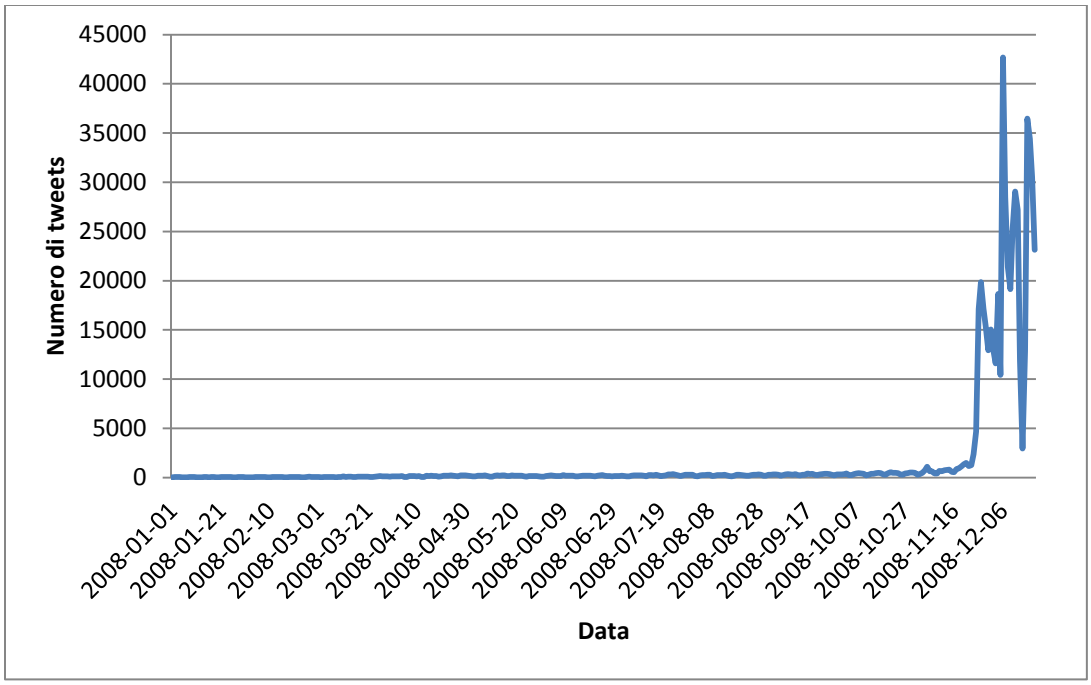

Figura 12 : Distribuzione dei tweets giornaliera su tutto il 2008, filtrati utilizzando le espressioni soggettive

Nel grafico che segue di figura 13 viene denotata in particolare la numerosità dei tweets giornalieri sul periodo da 1 Gennaio, 2008 a 30 Settembre, 2008, ottenuta con il filtraggio delle espressioni soggettive.

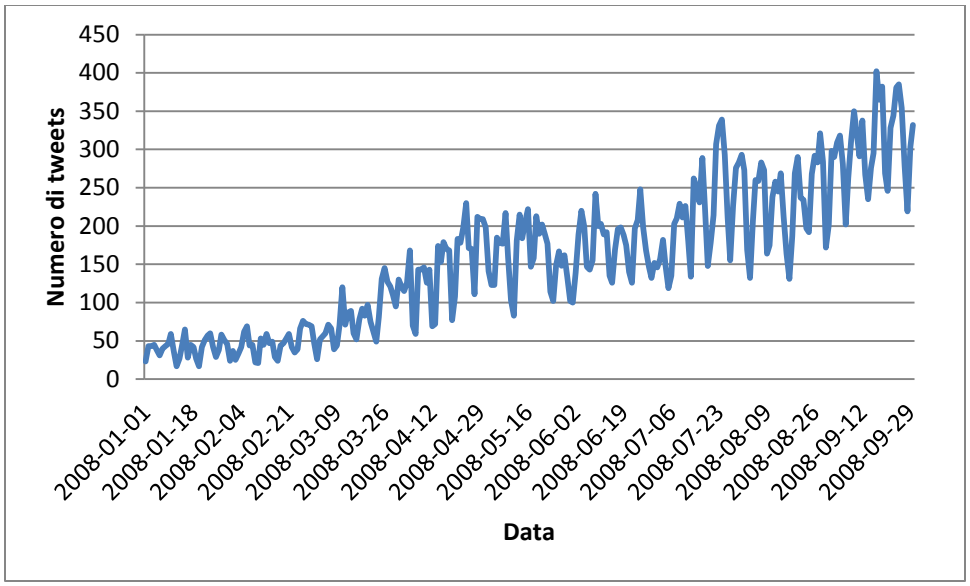

Figura 13 : Distribuzione dei tweets giornaliera fra 1 Gennaio, 2008 e 30 Settembre, 2008, filtrati utilizzando le espressioni soggettive

Per aumentare il numero di tweets giornalieri disponibili vengono aggiunte alle espressioni soggettive le emoticons. Con tale nuovo filtraggio la distribuzione giornaliera dei tweets risultante porta ad un aumento sensibile del numero di tweets; si riporta in figura 14 la distribuzione dei tweets sul periodo da 1 Gennaio, 2008 a 30 Settembre, 2008 sia effettuando il solo filtraggio delle espressioni personali, in blu, sia aggiungendo le emoticon, in arancione.

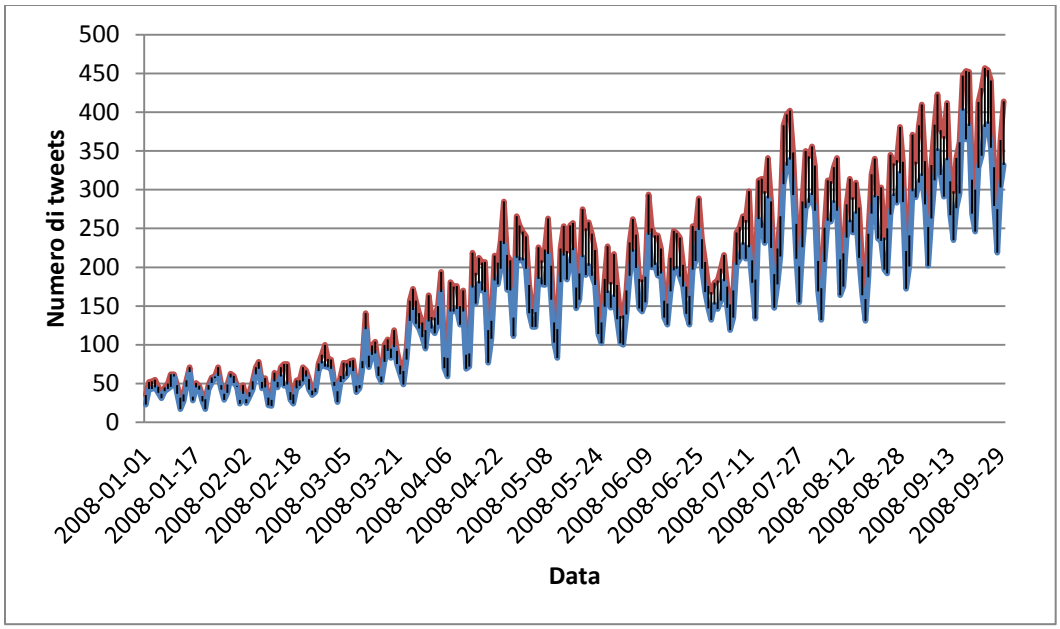

Figura 14 : Distribuzione dei tweets giornaliera fra 1 Gennaio, 2008 e 30 Settembre, 2008, filtrati utilizzando le sole espressioni soggettive (in blu) ed utilizzando espressioni soggettive ed emoticons (in rosso)

Raggiunto un numero accettabile minimo di tweets giornalieri, viene considerato il problema della differenza in termini di numero di tweets fra i primi mesi dell'anno e gli ultimi; diviene necessaria una linearizzazione delle quantità di tweets giornaliere, al fine di effettuare una analisi testuale consistente. Essendo lo scopo fondamentale quello di utilizzare l'intero data set disponibile per l'anno 2008 e di non consentire differenze in termini di quantità di tweets considerati per giorno, vengono campionati giornalmente 300 tweets, successivamente al filtraggio descritto sopra.

5.3 Costruzione della logical view testuale

Come detto la *logical view testuale* viene costruita secondo tre metodi, che creano 3 diversi modelli di rappresentazione dei dati. Tali modelli sono:

- **Modello di priorità ai tweets, nominato TWMOD nei tests.** Il primo modello inserisce nella logical view tutte le giornate incluse fra 1 Gennaio, 2008 e 19 Dicembre, 2008; ciò significa che vengono incluse istanze recanti date di chiusura del mercato di borsa, e conseguentemente classe *neutral*.

- **Modello di priorità all'apertura della borsa, nominato DJMOD nei tests.** Il secondo modello non inserisce nella logical view le istanze facenti riferimento, in termini di previsione, a giornate di chiusura del mercato di borsa; conseguentemente tutti gli elementi della logical view assumeranno classe *positive* oppure *negative*.

- **Modello di sola apertura borsa, nominato STRICKTDJMOD nei tests.** Nell'ultimo modello tutte le istanze faranno riferimento a giornate di apertura del mercato di borsa e non verranno considerate istanze includenti tweets pubblicati in giornate di chiusura della borsa stessa. Per esempio, nel caso in cui il raggruppamento avvenga unicamente sul giorno precedente la data di previsione e non vi sia traslazione temporale non esisteranno istanze relative ai Lunedì.

Al fine di sperimentare un ventaglio di modelli dalle caratteristiche differenti, i suddetti modelli vengono prodotti con:

- *paramAgg* variabile nell'intervallo [0,3]
- *paramLag* variabile nell'intervallo [0,2]

In totale vengono quindi generati 36 modelli testuali.

5.4 Costruzione delle bag-of-words

La costruzione delle bag-of-words viene effettuata sfruttando le funzionalità del filtro Weka StringToWordVector, attraverso la classe java BagOfWordsFiller. Ognuno dei 36 modelli prodotti viene quindi trasformato in una bag-of-words, secondo le specifiche:

- Mantenendo un numero di termini variabile fra 500, 1000 e 2000
- Rappresentando i pesi di occorrenza dei termini come presenza/assenza o calcolandone il *tfidf*
- Effettuando, o meno, LovinsStemming

Generando quindi 432 bag-of-words differenti.

5.5 Estrazione ed analisi dei gruppi di bontà dei tweets

La costruzione dei 4 gruppi di bontà avviene analizzando gli errori di classificazione prodotti classificando su *test set*, addestrando su *training set*. I risultati e gli errori generati da tale processo per ogni istanza (aggregazione di tweets) classificata riportano i seguenti parametri:

- Classe di appartenenza dell'istanza
- Classe predetta del classificatore
- Margine di previsione, che rappresenta in linguaggio Weka la confidenza con la quale il classificatore effettua una certa classificazione

Sulla base del margine di previsione e le 4 coppie classe di appartenenza-classe predetta che generano i 4 gruppi di bontà descritti, è possibile selezionare le istanze per le quali:

- La classificazione è avvenuta correttamente con un'alta confidenza

- La classificazione, seppure con alta confidenza, è risultata errata

Ciò permette di ottenere aggregazioni di tweets molto utili alla previsione, nel primo caso, oppure dualmente molto fastidiosi, nel secondo.

Attraverso la classe Discriminator sono state implementate diverse strategie di analisi e filtraggio del data set; ognuna di queste fa utilizzo della classe TweetChooser che gestisce il calcolo delle similarità con i 4 gruppi di bontà attraverso l'istituzione di una regola. Tale regola rappresenta l'aspetto fondamentale dell'atto di filtraggio: è una espressione booleana che identifica qualora il vettore per il quale si siano calcolate le similarità con i 4 gruppi debba essere mantenuto o meno nel data set. In generale sono state considerate due tipologie di regole:

- Regole che puntano a mantenere tweets utili, quindi simili a $TruePositive$ e $TrueNegative$
- Regole che puntano ad eliminare tweets inutili, quindi simili a $FalsePositive$ e $FalseNegative$

Le strategie implementate ed analizzate all'interno della classe Discriminator possono essere suddivise fra confronti singolo tweet-gruppi di bontà oppure confronti aggregazioni di tweets-gruppi di bontà.

5.5.1 Confronti fra singoli tweets e gruppi di bontà

Per ottenere una rappresentazione vettoriale di ogni singolo tweet compatibile con le istanze generate automaticamente da Weka, in particolare dal filtro StringToWordVector, i pesi relativi ad un termine i rappresentati tramite $tfidf$ vengono calcolati come il prodotto fra le grandezze $TFTransform$ ed $IDFTransform$, definite come segue:

$$TFTransform = ln(1 + f_{ij})$$

$$IDFTransform = ln(< Numero\ di\ documenti > / < Numero\ di\ documenti\ contenenti\ il\ termine\ i >)$$

Dove con f_{ij} si intende la frequenza di apparizione del termine i nel documento j.

Le strategie implementate in questo contesto sono:

- Utilizzando unicamente i tweets aggregati generati dalla costruzione della *logical view* diviene possibile filtrarli attraverso la regola di TweetChooser, decidendo se mantenerli nel data set oppure rimuoverli. Questo metodo può essere applicato unicamente sul *validation set*, oppure sull'intero data set.
- Espandendo la ricerca di tweets sull'intera collezione disponibile diviene possibile mantenere i soli tweets considerati utili (o molto utili) oppure dualmente eliminare i tweets inutili (o molto inutili); a differenza del punto precedente è possibile mantenere la stessa quantità di tweets per ogni aggregazione, sostituendo i tweets eliminati con altri migliori, esclusi in prima analisi per dovere di linearizzazione.
- Un approccio Greedy è stato esplorato per la costruzione di istanze considerate buone; tale algoritmo seleziona dapprima il 'miglior' tweet disponibile per una certa aggregazione, ossia il più simile a *TrueNegative* e/o *TruePositive*, per poi proseguire aggiungendo tweets che migliorino la similitudine dell'aggregazione che via via si va costruendo.

5.5.2 Confronti fra tweets aggregati e gruppi di bontà

Le strategie implementate in questo contesto sono:

- Una strategia inversamente Greedy è stata investigata; a partire da una aggregazione di tweets che manifesti similarità non sufficiente con i gruppi positivi (oppure troppo alta con i negativi) si tenta di rimuovere tweets al fine di far crescere tale similarità (oppure calare). In

questo contesto vengono tolti iterativamente tweets e testate le nuove similarità: se in crescita (calo) i tweet vengono definitivamente rimossi, altrimenti reinseriti.
- Un approccio vede etichettare il data set con le distanze ai 4 gruppi di bontà, facendo rientrare i parametri numerici in classificazione. Attraverso tale metodo è possibile analizzare sottogruppi di aggregazioni che manifestano simili gradi di similarità con i gruppi *TruePositive* e *TrueNegative*.

5.5 Classificazione finale

Il processo di classificazione finale sfrutta il data set ristrutturato ottenuto al passo precedente per classificare il *validation set*, sfruttando il medesimo algoritmo utilizzato per la produzione dei 4 gruppi di bontà.
Gli algoritmi di classificazione considerati in questo sistema sono:

- **RandomForest.** L'algoritmo consiste in una collezione di classificatori strutturati ad albero. RandomForest classifica un nuovo oggetto in input sfruttando tutti gli alberi presenti; ogni albero esprime un voto unitario di classificazione, risultando nella classe che, in tutta la 'foresta', ha ottenuto il numero maggiore di preferenze.
- **J48.** L'algoritmo *J48* è l'implementazione *Weka* dell'albero di decisione C4.5.
- **SMO.** Implementazione dell'algoritmo di Sequential Minimal Optimization in Weka.

Capitolo 6
Esperimenti e risultati

Il funzionamento del sistema in esame dipende fortemente dalla scelta dei valori di numerosi parametri, nonchè dalla modalità di rappresentazione e raggruppamento dei dati disponibili secondo i tre modelli di aggregazione proposti. Per questi motivi sono stati effettuati numerosi test, i cui risultati sono descritti nei paragrafi successivi, al variare dei diversi parametri che configurano il funzionamento del sistema.
Il data set disponibile riguarda i tweets pubblicati nell'anno 2008; più precisamente si dispone di tweets dalla data del 1 Gennaio, 2008 al 19 Dicembre, 2008. Ogni istanza della *logical view* costruita rappresenta una previsione su di una speficifica data; tale istanza potrà contenere tweets raggruppati per giorno ed aggregati su più giornate, in questi test sino a 4. Detto ciò il data set della *logical view* viene suddiviso in 3 insiemi:

- Training set
- Test set
- Validation set

Verrà inizialmente valutato il miglior metodo di aggregazione, fra i tre proposti, testandone l'efficacia della predizione classificando su validation set, addestrando utilizzando training e test set congiunti, senza effettuare alcuno step di miglioramento del modello. Questo procedimento identifica la miglior tripla composta da:

- modello di aggregazione
- tipologia di bag-of-words
- algoritmo di classificazione

I risultati ottenuti rappresenteranno un fondamentale elemento di comparazione per la fase di miglioramento del metodo.

Verrano estratti i 4 gruppi di bontà dei tweets, facendo riferimento al miglior modello di aggregazione risultato dello step precedente. L'estrazione avviene come detto addestrando un algoritmo di classificazione utilizzando training set ed analizzando test set. Per la scelta dell'algoritmo di classificazione vengono analizzate due alternative:

- Utilizzare l'algoritmo con le performance migliori ottenute nello step precedente
- Utilizzare una SVM con accuratezza elevata, in grado di garantire per ipotesi una buona separazione spaziale delle istanze da comparare

In ultima analisi verrano sperimentate alcune tecniche migliorative, presentando i risultati più efficaci.

6.1 Classificazione standard

I seguenti esperimenti vengono effettuati addestrando gli algoritmi di classificazione utilizzando gli insiemi di *training* e *test* congiunti, effettuando previsioni sul *validation set*.
Per comodità di lettura vengono suddivisi i test per modello di aggregazione scelto ed ulteriormente per algoritmo di classificazione utilizzato; i parametri variabili degli algoritmi vengono illustrati nelle tabelle 6, 7 e 8 che fungono da legenda, a seconda della tipologia utilizzata.

RandomForest	
I	Numero di alberi da costruire
K	Numero di feature da considerare

S	Seed per la generazione di numeri casuali

Tabella 6 : Legenda dei parametri dell'algoritmo RandomForest

J48	
C	Fattore di confidenza; determina il valore da utilizzare per effettuare pruning (rimozione dei rami che non portano guadagno in termini di accuratezza statistica del modello)
M	Numero minimo di istanze per foglia

Tabella 7 : Legenda dei parametri dell'algoritmo J48

SMO	
C	Parametro di complessità sulla base del quale costruire l'iperpiano; controlla quante istanze debbano essere usate come *support vectors*

Tabella 8 : Legenda dei parametri dell'algoritmo SMO

Il kernel utilizzato per l'algoritmo SMO è quello polinomiale.

Per ogni test vengono riportati i parametri relativi alla tipologia di aggregazione effettuata e alle scelte relative alla

costruzione della bag-of-words secondo la seguente legenda, in tabella 9.

paramAggr	Indica il numero di giorni da aggregare per effettuare la previsione
paramLag	Indica la traslazione temporale di aggregamento rispetto alla data di previsione
W	Numero di feature (termini,words) estratte dal testo per rappresentare ogni aggregazione, ossia per costruire la bag-of-words
Stemming	Indica l'utilizzo (*true*) o meno (*false*) dell'algoritmo LovinsStemming sui termini
tfidf	Indica qualora i pesi dei termini siano rappresentati attraverso *tfidf* (*true*) o semplicemente per presenza/assenza (*false*)

Tabella 9 : Legenda contenente i parametri relativi alla tipologia di aggregazione effettuata ed alle scelte relative alla costruzione della bag-of-words

Infine i risultati vengono riportati secondo un ordine di efficacia decrescente, in base al valore di fMeasure ottenuto.

6.1.1 Modello TWMOD

Viste le scarse performance del modello vengono riportati, brevemente, solo i migliori 3 esperimenti effettuati a seconda della tipologia del classificatore utilizzato.

- **RandomForest.**

paramAggr	*paramLag*	W	*Stemming*	*tfidf*	I	K	fMeasure	*Num.test set*
3	0	1000	false	true	100	200	0.5420560748	45

- **J48.**

paramAggr	*paramLag*	W	*Stemming*	*tfidf*	M	fMeasure	*Num.test set*
3	0	500	false	false	11	0.798961039	45

- **SMO**

paramAggr	*paramLag*	W	*Stemming*	*tfidf*	C	fMeasure	*Num.test set*
0	2	500	true	true	1.0	0.6129032258	45

6.1.2 Modello DJMOD

- **RandomForest.**

paramAggr	*paramLag*	W	*Stemming*	*tfidf*	I	K	fMeasure	*Num.test set*
3	1	1000	false	true	100	200	0.7139870582	45
3	1	1000	false	true	200	200	0.7111111111	45
3	1	1000	false	true	100	100	0.7079663533	45
3	2	1000	false	true	100	200	0.6901185771	45

3	2	1000	false	false	200	200	0.6696832579	45
1	2	2000	false	false	200	100	0.648	45
3	1	2000	false	true	100	200	0.6344827586	45
2	2	2000	true	false	100	200	0.6279202279	45
2	2	1000	false	false	200	100	0.6268457136	45
2	2	1000	false	false	100	200	0.6268457136	45

- **J48.**

paramAggr	*paramLag*	W	*Stemming*	*tfidf*	M	fMeasure	*Num.test set*
3	**0**	**500**	**false**	**true**	**5**	**0.798961039**	**45**
3	1	2000	false	true	5	0.736	45
3	1	2000	false	true	2	0.736	45
3	0	1000	false	true	11	0.7000546747	45
3	0	1000	false	true	8	0.6689757546	45
0	2	500	false	false	8	0.6680265171	45
3	0	1000	false	true	2	0.6680265171	45
0	2	2000	false	false	5	0.6602870813	45
2	2	500	false	true	5	0.6567114651	45
3	2	2000	true	true	2	0.6533653846	45

- **SMO**

paramAggr	*paramLag*	W	*Stemming*	*tfidf*	C	fMeasure	*Num.test set*
2	1	1000	true	true	0.01	0.6824074074	45
1	2	2000	false	false	0.01	0.6680265171	45
1	2	2000	false	false	0.1	0.6649350649	45
1	2	2000	false	false	1.0	0.6649350649	45
1	2	2000	false	false	10.0	0.6649350649	45

2	2	1000	true	false	0.1	0.6494826811	45
2	2	1000	true	false	1.0	0.6494826811	45
2	2	1000	true	false	10.0	0.6494826811	45
1	2	2000	true	false	0.1	0.642687747	45
1	2	2000	true	true	1.0	0.642687747	45

6.1.3 Modello STRICKTDJMOD

- **RandomForest.**

paramAggr	*paramLag*	W	*Stemming*	*tfidf*	I	K	fMeasure	*Num.test set*
1	2	2000	true	false	100	200	0.7375102599	34
1	1	500	true	false	100	100	0.6826080026	34
1	2	2000	false	false	200	200	0.6791792066	34
1	2	2000	true	false	100	100	0.6544891641	34
1	2	2000	true	false	200	200	0.6522301228	34
1	0	500	false	false	200	200	0.6470588235	34
1	1	500	true	false	100	200	0.6260987153	34
1	2	2000	true	false	200	100	0.6249003667	34
1	2	1000	false	false	200	100	0.6246474178	34
1	2	2000	false	true	100	100	0.6206264324	34

- **J48.**

paramAggr	*paramLag*	W	*Stemming*	*tfidf*	M	fMeasure	*Num.test set*
2	0	2000	false	false	2	0.7836990596	22
3	1	500	false	true	2	0.7828282828	11
2	0	1000	false	true	2	0.7658402204	22

2	1	2000	true	true	5	0.7658402204	22
2	1	2000	true	true	2	0.7658402204	22
3	2	1000	false	false	2	0.7376623377	11
3	2	1000	true	true	8	0.7376623377	11
3	2	2000	true	true	8	0.7376623377	11
3	2	500	true	true	8	0.7376623377	11
2	0	2000	false	true	2	0.7272727273	22

- **SMO.**

paramAggr	*paramLag*	W	*Stemming*	*tfidf*	C	fMeasure	*Num.test set*
1	2	1000	true	false	0.01	0.7403156385	34
1	2	2000	false	false	0.1	0.7403156385	34
1	2	2000	false	false	1.0	0.7403156385	34
1	2	2000	false	false	10.0	0.7403156385	34
1	2	500	true	false	0.01	0.7403156385	34
1	2	1000	true	true	0.1	0.7248346147	34
1	2	1000	true	true	1.0	0.7248346147	34
1	2	1000	true	true	10.0	0.7248346147	34
1	2	2000	false	false	0.01	0.7121848739	34
1	2	1000	true	false	0.1	0.710191769	34

6.1.4 Analisi del risultato migliore per la classificazione standard

Il risultato di classificazione migliore viene restituito dall'esperimento denotato dalle seguenti caratteristiche:

- Modello di aggregazione DJMOD, con *paramAggr=3* e *paramLag=0*. Ogni istanza di previsione viene costruita quindi considerando i tweets pubblicati nei 4 giorni precedenti la data

di previsione stessa. Questo risulta in, al più, 1200 tweets per ogni istanza;
- Il numero di termini estratti per la costruzione del *feature set* è di 500;
- Non fa utilizzo di algoritmi di Stemming per la selezione delle feature;
- Rappresenta i pesi dei termini estratti per ogni istanza con *tfidf;*
- Utilizza come algoritmo di classificazione J48, con minimo 5 istanze per foglia.

Il modello così costruito mette in luce una correlazione fra variazione dell'indice DJIA relativa ad una data d e contenuto testuale dei tweets, sottoposti ad una selezione iniziale, pubblicati nelle giornate $d-1$, $d-2$, $d-3$ e $d-4$; la previsione effettuata sui 45 giorni che compongono il *validation* set finale è caratterizzata da una accuratezza dell'80%.

Si riportano per esteso i risultati ottenuti dal modello.

Correctly Classified Instances	36	80 %
Incorrectly Classified Instances	9	20 %

Kappa statistic	0.5794
Mean absolute error	0.2964
Root mean squared error	0.4623
Relative absolute error	59.4382 %
Root relative squared error	92.7068 %
Coverage of cases (0.95 level)	84.4444 %
Mean rel. region size (0.95 level)	68.8889 %
Total Number of Instances	45

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
	0.722	0.148	0.765	0.722	0.743	0.580	0.765	0.675	positive
	0.852	0.278	0.821	0.852	0.836	0.580	0.765	0.756	negative
Weighted Avg.	0.800	0.226	0.799	0.800	0.799	0.580	0.765	0.723	

6.2 Classificazione con metodi migliorativi

La miglior combinazione modello di aggregazione, tipologia di bag-of-words ed algoritmo di classificazione risultante dai test di cui il paragrafo precedente risulta essere quella che utilizza:

- Metodo di aggregazione DJMOD con *paramAggr*=3 ed *paramLag*=0
- Bag-of-words costruita tramite il filtro StringToWordVector di Weka, estrando 500 features (termini) dal testo, senza effettuare stemming sugli stessi, rappresentandone i pesi con *tfidf*
- Algoritmo di classificazione J48, con parametri C=0.25 ed M=5

Lo scopo di questo ultimo step è quello di migliorare la fMeasure della classificazione descritta in 6.1.4, ristrutturando i data set utilizzati ed utilizzando il medesimo algoritmo di classificazione; questo procedimento si svolge analizzando le similarità fra i tweets del data set e le istanze, rappresentanti gli aggregamenti di tweets, appartenenti ad i 4 gruppi di bontà descritti nei capitoli precedenti.
Verrà descritta la costruzione dei 4 gruppi di bontà ed una valutazione delle similarità fra le istanze appartenenti al medesimo gruppo ed a gruppi differenti, utilizzando come algoritmo di classificazione J48 ed SMO:

- J48 viene utilizzato per mantenere la scelta dell'algoritmo migliore, relativamente al modello da migliorare.
- SMO viene utilizzato allo scopo di ottenere una divisione delle istanze classificate più netta da un punto di vista vettoriale, verificando l'ipotesi che questo approccio possa contribuire positivamente al metodo.

Dai risultati ottenuti è possibile sostenere la possibilità di confrontare quindi tweet singoli od aggregati rispetto ad i 4 gruppi e valutarne le similarità maggiori come indicazione di appartenenza e, quindi, come parametro attraverso il quale poter selezionare gli elementi da includere o meno nel data set utilizzato per la predizione finale.

Il primo gruppo di metodi proposti riguarda il confronto fra singoli tweets e gruppi di bontà. I procedimenti ammettono unicamente nelle aggregazioni rappresentanti le istanze del data set i tweets che soddisfano un certo criterio di similarità, detta regola. Considerando il modello di aggregazione utilizzato, il parametro *paramAggr=3* ed il numero giornaliero di tweets (300), ogni metodo almeno analizzerà per ogni aggregamento 1200 tweets, rimuovendo gli elementi che non soddisfano la regola selezionata. Vengono testate due tipologie di regole:

- **RULETRUE.** Vengono mantenuti unicamente i tweets che manifestino una similarità media verso le istanze *TruePositive* e *TrueNegative* maggiore di una soglia *threshold*.
- **RULEFALSE.** Per la seconda vengono mantenuti unicamente i tweets che manifestino una similarità media verso le istanze *FalsePositive* e *FalseNegative* minore di una soglia *threshold*.

Il settaggio della soglia *threshold* iniziale viene effettuato valutando empiricamente le similarità medie di tutti i tweets con i gruppi di bontà; essa viene poi fatta variare in un intorno che permetta di valutare la qualità della classificazione finale utilizzando, al limite, una regola molto restrittiva e dualmente una regola largamente ammissiva, che

porta all'accettazione di tutti i tweets di partenza e quindi al medesimo risultato riscontrato in 6.1.4.

6.2.1 Estrazione G.J48: gruppi di bontà dei tweets utilizzando J48

L'obiettivo è quello di estrarre dal validation set originario gruppi di istanze che portino caratteristiche comuni utili ai fini della classificazione; in particolare ciò che serve è un insieme di tweet 'buoni', ossia che rendano possibile effettuare una corretta classificazione, e 'cattivi', per i quali invece la classificazione non avvenga correttamente.

A tal scopo diviene necessaria una restrutturazione del procedimento di classificazione descritto in 5.1, addestrando l'algoritmo rivelatosi migliore su training set ed effettuando la classificazione su validation set, così suddivisi temporalmente:

- Training set, dal 4 Gennaio, 2008 al 31 Luglio,2008
- Validation set, dal 1 Agosto,2008 al 30 Settembre,2008

Si riportano i risultati di tale classificazione, e relativi errori, in tabella, includendo la data della particolare istanza-predizione, il *prediction margin,* che identifica l'affidabilità della previsione (dal punto di vista del classificatore), la classe prevista e quindi la classe reale.

data	prediction margin	classe prevista	classe
2008-08-01	1	positive	positive
2008-08-05	-1	negative	positive
2008-08-06	-1	negative	positive
2008-08-07	-1	negative	positive

2008-08-08	-1	negative	positive
2008-08-12	-1	positive	negative
2008-08-13	-1	positive	negative
2008-08-14	-1	positive	negative
2008-08-15	-1	positive	negative
2008-08-19	-1	positive	negative
2008-08-20	0.842105	negative	negative
2008-08-21	0.842105	negative	negative
2008-08-22	-0.842105	negative	positive
2008-08-26	0.714286	positive	positive
2008-08-27	0.714286	positive	positive
2008-08-28	0.714286	positive	positive
2008-08-29	1	positive	positive
2008-09-03	0.714286	positive	positive
2008-09-04	-0.714286	positive	negative
2008-09-05	0.842105	negative	negative
2008-09-09	-0.714286	positive	negative
2008-09-10	1	negative	negative
2008-09-11	1	negative	negative
2008-09-12	1	negative	negative
2008-09-16	0.714286	positive	positive
2008-09-17	1	negative	negative

2008-09-18	1	positive	positive
2008-09-19	1	positive	positive
2008-09-23	-0.714286	positive	negative
2008-09-24	-0.714286	positive	negative
2008-09-25	0.714286	positive	positive
2008-09-26	0.714286	positive	positive
2008-09-30	0.714286	positive	positive

L'accuratezza della classificazione che porta alla generazione delle istanze, come si può notare dalla tabella considerando il numero di predizioni giuste ed errate, risulta essere del 57.57% ; ridurre l'insieme di addestramento da training e set congiunti al solo insieme di training abbassa notevolmente le performance del modello.

Valori positivi del *prediction margin* identificano classificazioni effettuate correttamente; in maniera duale, valori negativi identificano classificazioni errate. In particolare, siamo interessati ai valori unitari del *prediction margin*, che evidenziano classificazioni per le quali vi era una buona affidabilità da parte dell'algoritmo: valori unitari positivi identificano classificazioni giuste con alto grado di affidabilità, mentre invece valori unitari negativi identificano classificazioni errate, nonostante l'algoritmo fosse sicuro della previsione.
Mantenendo quindi le sole istanze relative a valori unitari del *prediction margin* vengono creati 4 gruppi di istanze come segue:

- *Gruppo TruePositive*, ossia di istanze classificate correttamente come *positive*; conta 4 istanze
- *Gruppo TrueNegative*, ossia di istanze classificate correttamente come *negative*; conta 4 istanze

- *Gruppo FalsePositive*, ossia di istanze classificate come *positive*, ma che in realtà sono *negative*; conta 5 istanze
- *Gruppo FalseNegative*, ossia di istanze classificate come *negative*, ma che in realtà sono *positive*; conta 4 istanze

Al fine di valutare la bontà dei raggruppamenti effettuati e delle ipotesi sostenute, vengono calcolate le similarità fra istanze sia appartenenti allo stesso gruppo e sia appartenenti a gruppi diversi; ciò che ci si aspetta è che le istanze appartenenti al medesimo raggruppamento espongano una alta similarità, mentre invece possano comparire aspetti di dissimilarità comparando istanze di raggruppamenti differenti. Sono riportati i risultati di tali confronti in tabella.

	TruePositive	*TrueNegative*	*FalsePositive*	*FalseNegative*
TruePositive	0.8193593991547234	0.8278441425440165	0.7786811433441484	0.7723943878911239
TrueNegative	0.8278441425440165	0.9141146464403017	0.7760305763985135	0.7382468039429315
FalsePositive	0.7786811433441484	0.7760305763985135	0.8479150997984135	0.770381300567943
FalseNegative	0.7723943878911239	0.7382468039429315	0.770381300567943	0.912532322202225

Sulla diagonale principale si trovano i confronti fra istanze appartenenti al medesimo gruppo di bontà; tali similarità sono significativamente maggiori rispetto agli altri confronti. Questo significa che istanze sulle quali viene effettuata una certa previsione sono simili, e sostiene l'ipotesi alla base dello sviluppo dei metodi migliorativi proposti in seguito.

Si noti come le similarità fra istanze appartenenti a diversi gruppi 'buoni' siano alte, mentre invece fra istanze appartenenti a differenti gruppi "cattivi" siano più basse; le similarità esposte fra istanze appartenenti a gruppi 'buoni' e gruppi 'cattivi' sono anch'esse minori di quelle calcolate per istanze appartenenti al medesimo gruppo.

Nelle prossime sezioni ci si riferirà ad i gruppi di bontà generati tramite J48 con il termine G.J48.

6.2.2 Estrazione G.SMO: gruppi di bontà dei tweets utilizzando SMO

SMO viene utilizzato allo scopo di ottenere una divisione delle istanze classificate più netta da un punto di vista vettoriale, verificando l'ipotesi che questo approccio possa contribuire positivamente al metodo; l'approccio in questo caso è quello di identificare una SVM che sia in grado di classificare le istanze di test set, addestrando su training set, con accuratezza elevata.

SMO, che fallisce nella classificazione dell'intero validation set rispetto a J48, aumenta molto le performance sul solo test set; l'accuratezza è in questo caso dell'81.82%. Si riportano i risultati di tale classificazione, e relativi errori, in tabella, includendo la data della particolare istanza-predizione, il *prediction margin,* che identifica l'affidabilità della previsione (dal punto di vista del classificatore), la classe prevista e quindi la classe reale.

data	prediction margin	classe prevista	classe
2008-08-01	1	positive	positive
2008-08-05	1	positive	positive
2008-08-06	1	positive	positive

2008-08-07	1	positive	positive
2008-08-08	1	positive	positive
2008-08-12	1	negative	negative
2008-08-13	1	negative	negative
2008-08-14	1	negative	negative
2008-08-15	1	negative	negative
2008-08-19	1	negative	negative
2008-08-20	1	negative	negative
2008-08-21	1	negative	negative
2008-08-22	1	positive	positive
2008-08-26	1	positive	positive
2008-08-27	1	positive	positive
2008-08-28	1	positive	positive
2008-08-29	1	positive	positive
2008-09-03	-1	negative	positive
2008-09-04	1	negative	negative
2008-09-05	1	negative	negative
2008-09-09	1	negative	negative
2008-09-10	1	negative	negative
2008-09-11	1	negative	negative
2008-09-12	1	negative	negative
2008-09-16	-1	negative	positive

2008-09-17	1	negative	negative
2008-09-18	-1	negative	positive
2008-09-19	-1	negative	positive
2008-09-23	-1	positive	negative
2008-09-24	1	negative	negative
2008-09-25	1	positive	positive
2008-09-26	1	positive	positive
2008-09-30	-1	negative	positive

A differenza del metodo relativo alla sezione precedente, pur sempre mantenendo quindi le sole istanze relative a valori unitari del *prediction margin*, vengono creati solamente 2 gruppi di istanze come segue:

- *Gruppo TruePositive*, ossia di istanze classificate correttamente come *positive*; conta 12 istanze
- *Gruppo TrueNegative*, ossia di istanze classificate correttamente come *negative*; conta 15 istanze

Il *Gruppo FalsePositive*, ossia di istanze classificate come *positive*, ma che in realtà sono *negative*, conterebbe unicamente 1 istanza, mentre il *Gruppo FalseNegative*, ossia di istanze classificate come negative, ma che in realtà sono positive, ne conterebbe 5 istanze; vengono utilizzati per i confronti unicamente i raggruppamenti *TruePositive* e *TrueNegative*, molto più numerosi.

Anche per i gruppi ottenuti con SMO, al fine di valutarne la bontà, vengono calcolate le similarità fra istanze sia appartenenti allo stesso gruppo e sia appartenenti a gruppi diversi; ciò che ci si aspetta è che le istanze appartenenti al medesimo raggruppamento espongano una alta similarità, mentre invece possano comparire aspetti di dissimilarità

comparando istanze di raggruppamenti differenti. Sono riportati i risultati di tali confronti in tabella.

	TruePositive	*TrueNegative*
TruePositive	0.7938310475565953	0.7711354017203641
TrueNegative	0.7711354017203641	0.8000437277992063

Sulla diagonale principale si trovano i confronti fra istanze appartenenti al medesimo gruppo di bontà; tali similarità sono significativamente maggiori rispetto agli altri confronti. Per SMO, a differenza dei gruppi ottenuti con J48, la differenza fra similarità fra istanze appartenenti allo stesso gruppo e similarità fra istanze appartenenti a gruppi differenti non è così marcata.

Nelle prossime sezioni ci si riferirà ad i gruppi di bontà generati tramite SMO con il termine G.SMO.

6.2.3 Rimozione dei singoli tweets dal test set – G.J48

Un primo metodo si propone di riesaminare le aggregazioni effettuate per la costruzione delle istanze del test set, utilizzate per la classificazione descritta in 5.1, ammettendo unicamente i tweets che soddisfano la regola corrente. Il metodo viene testato utilizzando le regole RULETRUE e RULEFALSE descritte in precedenza, facendo variare la soglia *threshold*; l'intervallo di variazione della soglia è stato valutato empiricamente, risultando ai due estremi in regole molto restrittive e molto permissive (ossia si ottiene il risultato precedente, in quanto tutti i tweets di partenza vengono mantenuti). I risultati, applicando la regola RULETRUE, sono mostrati nel grafico che segue, in figura 15.

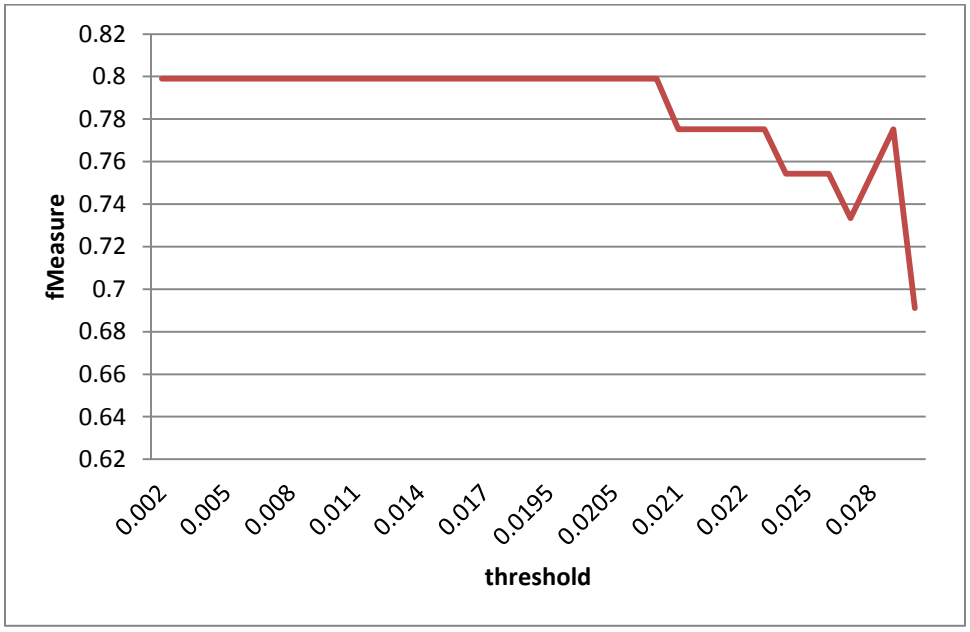

Figura 15 : Andamento della fMeasure effettuando filtraggio tweets singoli con RULETRUE utilizzando G.J48

La linea rossa approssima l'andamento della fMeasure, calcolata per la classificazione finale naturalmente con J48, al variare della soglia threshold.

La regola RULETRUE si propone di mantenere unicamente i tweets che manifestino una similarità verso i gruppi *TruePositive* e *TrueNegative* maggiore della soglia *threshold*: una soglia bassa ammette la totalità dei tweets, mentre una soglia alta ammette un sottoinsieme dei tweets disponibili, plausibilmente più simili ai set 'buoni' quindi più utili alla classificazione. I risultati però non riflettono questa ipotesi ma un calo nelle performance. E' interessante però notare come per, al limite, un valore di threshold di 0.0209 il numero di tweets eliminati dalle aggregazioni sia 9391, identificabili quindi come inutili alla classificazione.

I risultati, applicando la regola RULEFALSE, sono mostrati nel grafico che segue di figura 16.

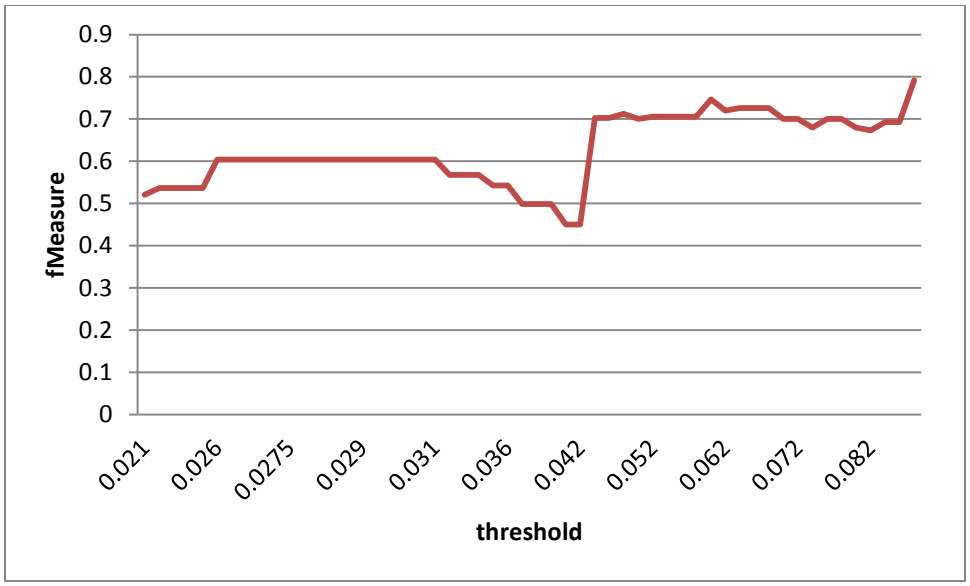

Figura 16 : Andamento della fMeasure effettuando filtraggio tweets singoli con RULETRUE utilizzando G.J48

Nel secondo caso la regola RULEFALSE si propone di eliminare i tweets che manifestino una similarità maggiore della soglia *threshold* verso i gruppi *FalseNegative* e *FalsePositive*: un valore basso di soglia elimina il maggior numero di tweets dal data set, mentre dualmente un valore elevato ammette ogni elemento. Anche in questo caso l'ipotesi non è confermata: analizzando singolarmente le similarità dei tweets verso i raggruppamenti ricavati con G.J48 non riflette la reale utilità ai fini di una corretta classificazione.

6.2.4 Rimozione dei singoli tweets dal test set – G.SMO

Viene qui applicata la regola RULETRUE, che si propone di mantenere unicamente i tweets che manifestino una similarità verso i gruppi *TruePositive* e *TrueNegative* maggiore della soglia *threshold*. I risultati della successiva classificazione, che analizza il test set filtrato, sono mostrati nel grafico che segue, in figura 17.

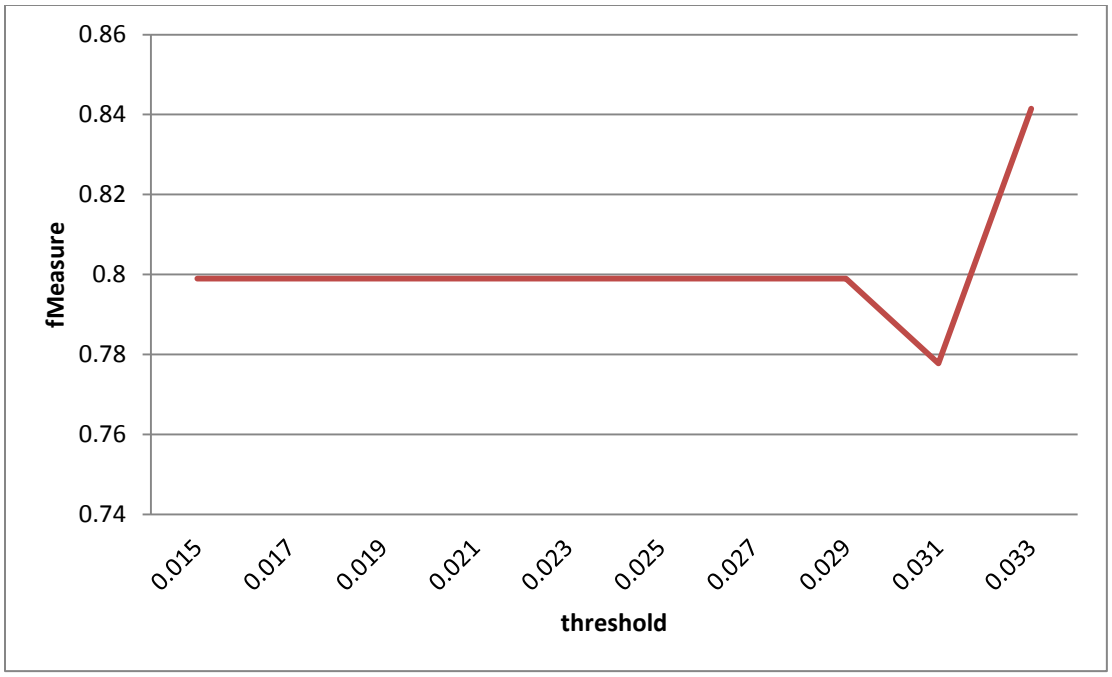

Figura 17 : Andamento della fMeasure effettuando filtraggio tweets singoli con RULETRUE utilizzando G.SMO

Utilizzando come termini di paragone G.SMO, per una *threshold* di 0.033 la fMeasure della classificazione finale aumenta sino ad 0.8415; in corrispondenza di tale valore i tweets rimossi sono 15998. Si riportano per esteso i risultati ottenuti da questo modello migliorativo.

Correctly Classified Instances	38	84.4444 %
Incorrectly Classified Instances	7	15.5556 %

Kappa statistic	0.6667
Mean absolute error	0.2595
Root mean squared error	0.4066

Relative absolute error	52.0315 %
Root relative squared error	81.5342 %
Coverage of cases (0.95 level)	91.1111 %
Mean rel. region size (0.95 level)	73.3333 %
Total Number of Instances	45

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
	0.722	0.074	0.867	0.722	0.788	0.674	0.808	0.755	positive
	0.926	0.278	0.833	0.926	0.877	0.674	0.808	0.793	negative
Weighted Avg.	0.844	0.196	0.847	0.844	0.841	0.674	0.808	0.778	

6.2.5 Sostituzione singoli tweets dal test set

Il secondo metodo sperimentato ricalca il procedimento del primo, descritto in 6.2.3 e 6.2.4, ma non effettua una rimozione dei tweets, bensì sostituisce i tweets considerati dannosi con nuovi tweets, recuperati dalla collezione filtrata iniziale e non considerati inizialmente, che soddisfino la regola corrente. In entrambi i casi, utilizzando G.J48 e G.SMO, le performance della classificazione calano drasticamente, in quanto anche per soglie molto restrittive vengono sostituiti molti tweets con nuovi i cui termini utilizzati sono sconosciuti al classificatore.

6.2.6 Rimozione singoli tweets dall'intero data set – G.SMO

Questo metodo espande la discriminazione di tweets 'buoni' o 'cattivi' all'intero data set rappresentante la *logical view*, costruendo quindi un nuovo modello di classificazione: esso rimuove tutti i tweet rinvenuti che non soddisfano la regola corrente.

Essendo verificato positivo il filtraggio utilizzando G.SMO e RULETRUE sul test set viene effettuato anche sull'intero data set, con riferimento ad un intorno del valore di *threshold* migliore ottenuta dall'esperimento di paragrafo 6.2.4.

Utilizzando quindi un valore di *threshold* di 0.033 e filtrando l'intero data set, discriminando la scelta dei tweets da mantenere attraverso RULETRUE, l'accuratezza finale risultante dalla classificazione con J48 scende sino al 60%: osservando l'albero risultante dal modello di classificazione si notano notevoli differenze, dovute alla ristrutturazione delle aggregazioni, che non fanno emergere il carattere predittivo del contenuto testuale dei tweets.

6.2.7 Filtraggio istanze training e test set – G.J48

Una opportunità è quella di analizzare le istanze del data set utilizzato per la classificazione di 5.1 in termini della similarità fra di esse ed i 4 gruppi di bontà costruiti. A questo scopo vengono aggiunti ad ogni istanza 4 attributi, frutto del calcolo delle similarità fra l'istanza stessa ed i gruppi *TruePositive, TrueNegative, FalsePositive* e *FalseNegative*. Tali similarità vengono poi confrontate con una soglia *threshold*, allo scopo di eliminare dal data set:

- Istanze per le quali la similarità con *TruePositive* e *TrueNegative* sia minore della soglia *threshold*

- E che esibiscano una similarità simile verso i gruppi *FalsePositive* e *FalseNegative*

I risultati relativi alla classificazione su set filtrati dal metodo descritto sono riportati nel grafico seguente di figura 18, al variare della soglia *threshold*.

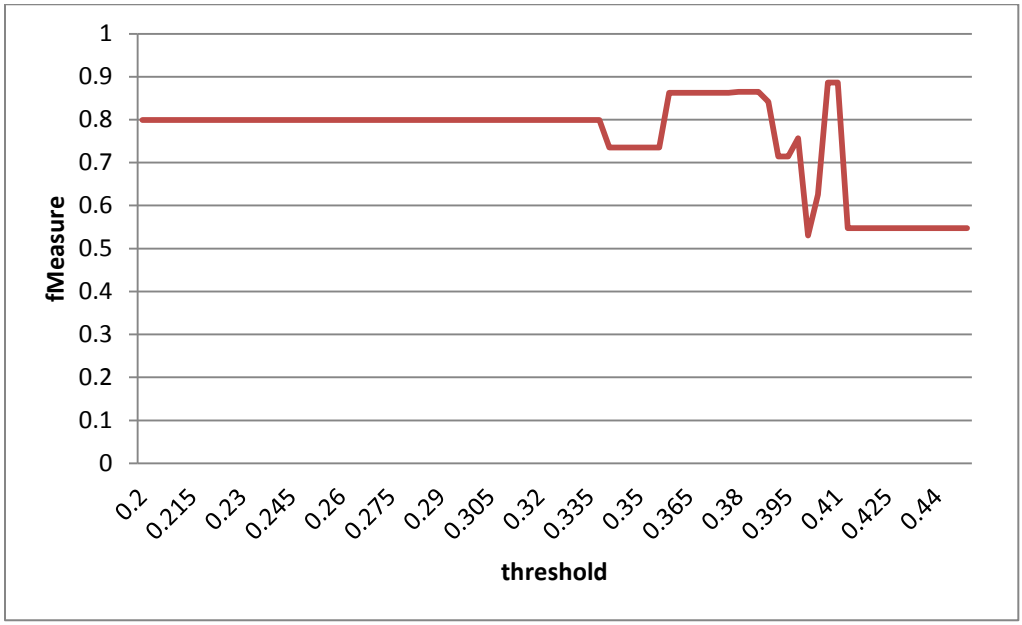

Figura 18 : Andamento della fMeasure classificando su data set filtrato con analisi similarità istanze training e test set con G.J48

I primi tentativi di filtraggio, corrisponenti ad una soglia *threshold* bassa, eliminano istanze senza influire sul comportamento del classificatore; per una soglia di 0.347 le performance calano: questo può risiedere nella semplicità dei criteri di rimozione binari implementati.
Per una soglia di 0.407 le performance crescono oltre i risultati migliori descritti in 5.1.4: il modello che ne risulta è privo di 25 istanze rimosse nel periodo Gennaio-Febbraio. Questo dimostra che nonostante la scarsa disponibilità di tweets giornalieri nel periodo sia possibile, considerandone l'aggregazione, contribuire positivamente al modello finale mantenendone certi raggruppamenti che manifestano particolari caratteristiche. L'affidabilità di classificazione

ottenuta operando il filtraggio relativo a *threshold*=0.407 è dell'88.8%, per una fMeasure=0.887; i risultati ottenuti per esteso sono riportati in seguito.

Correctly Classified Instances	40	88.8889 %
Incorrectly Classified Instances	5	11.1111 %

Kappa statistic	0.7619
Mean absolute error	0.2108
Root mean squared error	0.345
Relative absolute error	42.0906 %
Root relative squared error	68.9005 %
Coverage of cases (0.95 level)	97.7778 %
Mean rel. region size (0.95 level)	84.4444 %
Total Number of Instances	45

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
	0.778	0.037	0.933	0.778	0.848	0.770	0.878	0.830	positive
	0.963	0.222	0.867	0.963	0.912	0.770	0.878	0.866	negative
Weighted Avg.	0.889	0.148	0.893	0.889	0.887	0.770	0.878	0.851	

6.2.8 Filtraggio istanze training e test set – G.SMO

L'analisi della sezione precedente viene svolta anche utilizzando G.SMO, considerando naturalmente le sole distanze fra istanze del data set e gruppi *TruePositive* e *TrueNegative*.

Tali similarità vengono poi confrontate con una soglia *threshold*, allo scopo di eliminare dal data set:

- Istanze per le quali la similarità con *TruePositive* e *TrueNegative* sia minore della soglia *threshold*

I risultati relativi alla classificazione su set filtrati dal metodo descritto sono riportati nel grafico seguente di figura 19, al variare della soglia threshold.

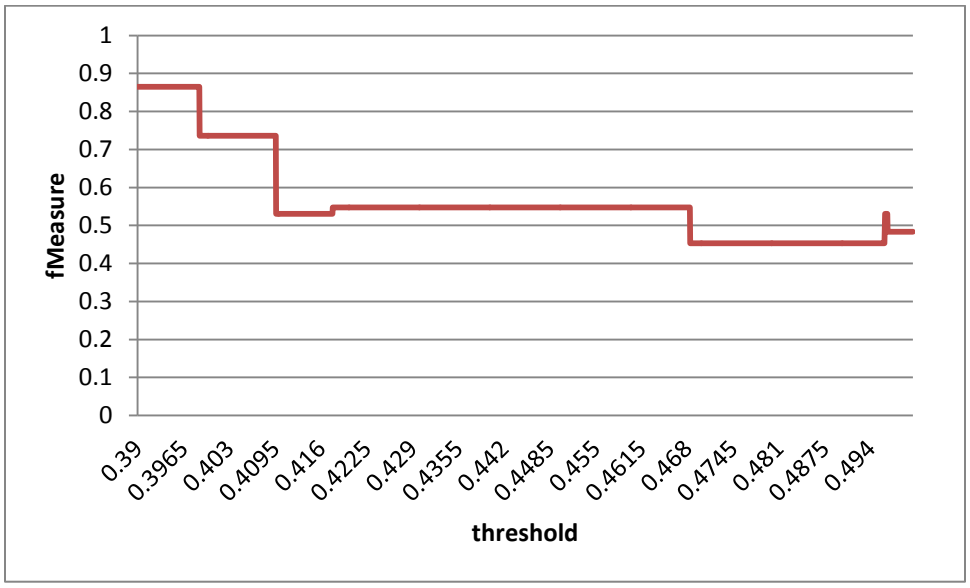

Figura 19 : Andamento della fMeasure classificando su data set filtrato con analisi similarità istanze training e test set con G.SMO

In corrispondenza del valore iniziale di *threshold* di 0.39 l'accuratezza ottenuta con classificazione standard viene qui migliorata: per tale valore vengono rimosse 3 istanze dai training set e test set, aumentando la fMeasure per la

classificazione finale ad 0.8651. Per un valore di *threshold* di 0.39875 vengono rimosse 12 istanze, mantenendo la fMeasure costante a 0.8651; per valori maggiori di soglia questa decresce.

I risultati del modello migliorativo sono riportati per esteso.

Correctly Classified Instances	39	86.6667 %
Incorrectly Classified Instances	6	13.3333 %

Kappa statistic	0.717
Mean absolute error	0.212
Root mean squared error	0.363
Relative absolute error	42.5731 %
Root relative squared error	72.903 %
Coverage of cases (0.95 level)	97.7778 %
Mean rel. region size (0.95 level)	72.2222 %
Total Number of Instances	45

	TP Rate	FP Rate	Precision	Recall	F-Measure	MCC	ROC Area	PRC Area	Class
	0.778	0.074	0.875	0.778	0.824	0.720	0.893	0.844	positive
	0.926	0.222	0.862	0.926	0.893	0.720	0.893	0.906	negative
Weighted Avg.	0.867	0.163	0.867	0.867	0.865	0.720	0.893	0.881	

6.3 Valutazione dei modelli trattati

Allo scopo di confrontare i risultati ottenuti per la classificazione effettuata in 6.1.4, senza utilizzare filtraggi relativi a relazioni di similarità, e nei modelli migliorativi ottenuti in 6.2.4, 6.2.7 e 6.2.6 effettuando confronti di similarità fra istanze del data set ed i 4 gruppi di bontà generati, vengono valutati gli intervalli di confidenza dell'accuratezza risultante per entrambi i casi. Lo studio ha lo scopo di verificare la reale accuratezza dei modelli. Per mantenere linearità nei confronti verso il modello di Bollen [46] vengono considerate le accuratezze percentuali approssimate sino alla prima cifra decimale.
La probabilità di correttezza (ossia confidenza) che si intende garantire è del 95%; l'intervallo di confidenza rigurdante il modello non filtrato, quindi classificazione standard, di accuratezza 80% relativo a 6.1.4 è:

$$Intervallo\ di\ confidenza_{6.1.4} = [71.79\% ; 83.49\%]$$

L'intervallo di confidenza risultante per il modello di accuratezza 84.4% risultato del filtraggio descritto in 6.2.4 è:

$$Intervallo\ di\ confidenza_{6.2.4} = [76.32\% ; 87.07\%]$$

L'intervallo di confidenza risultante per il modello di accuratezza 86.7% risultato del filtraggio descritto in 6.2.7 è:

$$Intervallo\ di\ confidenza_{6.2.7} = [78.74\% ; 88.89\%]$$

L'intervallo di confidenza risultante per il modello migliore, con accuratezza 88.9%, risultato del filtraggio descritto in 6.2.6 è:

$$Intervallo\ di\ confidenza_{6.2.6} = [81.08\% ; 90.59\%]$$

Infine, come metro di paragone, si riporta l'intervallo di confidenza risultante dal metodo di Bollen [46], di accuratezza 86.7%:

$$Intervallo\ di\ confidenza_{Calm} = [70.51\%, 87.92\%]$$

L'ampiezza degli intervalli di confidenza dei metodi proposti è minore di quello derivante dallo studio [46]: questo è da ricondurre alla numerosità dell'insieme analizzato per la predizione finale che, nel caso dei metodi proposti, conta 45 istanze, mentre per [46] ne conta unicamente 15.
Riportandoci nella condizione dell'esperimento di Bollen [46], quindi analizzando unicamente le variazioni dell'indice DJIA per il mese di Dicembre, otteniamo un'accuratezza del 100%, anche con il modello a classificazione standard di cui 6.1.4.

Conclusioni

In questa tesi si è sviluppato un sistema per la previsione delle variazioni dell'indice Dow Jones Industrial Average di chiusura analizzando il contenuto testuale di tweets di carattere soggettivo, esprimenti stati d'animo o condizioni psicologiche.
Il contributo del lavoro svolto si riconduce alle tecniche migliorative costruite ed analizzate, che grazie ad uno studio di similarità condotto sulla base dei risultati di una classificazione intermedia consentono di aumentare l'accuratezza ed individuare gruppi di tweets inadatti alla previsione. I miglioramenti sono stati riscontrati sia eliminando tweets testuali singoli riconosciuti come non idonei al modello di previsione, sia intere istanze facenti parte degli insiemi utilizzati per l'addestramento dell'algoritmo di classificazione.
Queste tecniche assumono una rilevanza più generale del solo caso in esame: esse permettono di inferire qualora i dati raccolti siano in realtà non idonei allo scopo predittivo di interesse.
Seguendo ed adattando le indicazioni di Bollen [46] sul preprocessamento e la selezione dei dati, sono stati implementati e testati diversi modelli di aggregazione e di rappresentazione dei tweets, sui quali applicare le tecniche migliorative descritte ed effettuare la previsione finale. Questa, nel modello migliore costruito, ha un'accuratezza dell'88.9% nell'effettuare previsioni sulle variazioni giornaliere dell'indice.
Essendo lo studio globale scorrelato dall'andamento del DJIA, il sistema è utilizzabile per effettuare previsioni su qualsiasi indice di borsa scelto: in tal senso, attraverso un differente preprocessamento dei dati, è possibile selezionare tweets riguardanti aspetti d'interesse per il nuovo indice selezionato, costruendo un modello di categoria semantica differente.

Appendice A Confronti fra l'accuratezza dei modelli proposti

Nonostante le numerosità del validation set siano in generale non elevate, vengono qui applicati alcuni metodi statistici per confrontare le accuratezze dei modelli considerati.
Per determinare quale modello sia migliore, quindi valutare qualora la differenza fra le accuratezze restituite da due modelli sia statisticamente significativa, occore considerare l'errore e che li caratterizza; questo risulta approssimabile, per modelli testati su data set con numerosità $N > 30$, ad una Normale di media μ e deviazione standard σ:

$$e \sim N(\mu, \sigma)$$

La cui varianza approssimata è:

$$\hat{\sigma}^2 = \frac{e(1-e)}{n}$$

Con n numerosità del data set su cui è stato testato il modello in analisi. Per verificare se la differenza d dell'accuratezza tra due modelli sia statisticamente significativa si definisce $d = e_1 - e_2$ come la differenza fra l'errore di un modello *1* e l'errore di un modello *2*;
$d \sim N(d_t, \sigma_t)$, dove d_t è la reale differenza cercata.
La varianza ${\sigma_t}^2$ è data da ${\sigma_t}^2 = {\sigma_1}^2 + {\sigma_2}^2$, approssimabile a ${\hat{\sigma}_1}^2 + {\hat{\sigma}_2}^2 = \frac{e_1(1-e_1)}{n_1} + \frac{e_2(1-e_2)}{n_2}$.

Infine d_t(per confidenza 95%) è data da $d_t = d \pm 1.96\,\hat{\sigma}_t$

Confrontiamo quindi il modello migliore, con accuratezza 88.89%, risultato del filtraggio descritto in 6.2.6, con gli altri modelli ottenuti, allo scopo di ricavare il livello di confidenza necessario per rigettare l'ipotesi che la differenza non sia statisticamente significativa.

- **Modello 6.1.4, accuratezza 80%.** La differenza non è statisticamente significativa per confidenza inferiore a 0.762

- **Modello 6.2.4, accuratezza 84.44%.** La differenza non è statisticamente significativa per confidenza inferiore a 0.847
- **Modello 6.2.7, accuratezza 86.67%.** La differenza non è statisticamente significativa per confidenza inferiore a 0.98

Bibliografia

[1] U. Fayyad, G. Piatetsky-Shapiro, P. Smyth. From Data Mining to Knowledge Discovery in Databases

[2] J. R. Quinlan. Induction of Decision Trees, Machine Learning, 1(1), pp 81-106

[3] R. Baeza-Yates, B. Ribeiro-Neto. Modern Information Retrieval : the concepts and technology behind search

[4] M. Porter. An algorithm for suffix stripping. *Program*, pages 130–137

[5] W. Bruce Croft, D. Metzler, T. Strohman. Search engines, Information Retrevial in practice

[6] K. E. Lochbaum, L. A. Streeter. Combining and comparing the effectiveness of latent semantic indexing and the ordinary vector space model for information retrieval. *Information Processing and Management*, 25(6):665–676

[7] C. Apte, F. Damerau, S. Weiss. Automated Learning of Decision Rules for Text Categorization, ACM Transactions on Information Systems, 12(3), pp. 233-251

[8] W. W. Cohen, Y. Singer. Context-Sensitive Learning Methods for Text Categorization

[9] T. Joachims. Text Categorization with Support Vector Machines: Learning with Many Relevant Features

[10] H. Schutze, D. Hull, J. Pedersen. A comparison of classifiers and document representations for the routing problem.ACM SIGIR Conference

[11] J. Han, M. Kamber. Data Mining: Concepts and Techniques, Second Edition

[12] B. E. Boser, I. Guyon, V. Vapnik . A training algorithm for optimal margin classifiers

[13] K. S. Jones, P. W., Morgan Kaufmann. Readings in information retrieval

[14] M. Hearst. Support Vector Machines, IEEE Intelligent Systems (1998)

[15] C. C. Aggarwal, Chengxiang Zhai. Mining Text Data, Chapter 1

[16] C. C. Aggarwal, Chengxiang Zhai. Mining Text Data, Chapter 6

[17] comScore/the Kelsey group. Online consumer-generated reviews have significant impact on offline purchase behavior. Press Release, November 2007.

[18] L. Cabral, A. Hortacsu. The dynamics of seller reputation: Theory and evidence from eBay. Working paper, downloaded version revised in March, 2006.

[19] J. Tatemura. Virtual reviewers for collaborative exploration of movie reviews. In Proceedings of Intelligent User Interfaces (IUI), pages 272–275, 2000.

[20] X. Jin, Y. Li, T. Mah, J. Tong. Sensitive webpage classification for content advertising. In Proceedings of the International Workshop on Data Mining and Audience Intelligence for Advertising, 2007.

[21] E. Riloff, J. Wiebe. Learning extraction patterns for subjective expressions. In Proceedings of the Conference on

Empirical Methods in Natural Language Processing (EMNLP), 2003.

[22] S. Piao, S. Ananiadou, Y. Tsuruoka, Y. Sasaki, J. McNaught. Mining opinion polarity relations of citations. In International Workshop on Computational Semantics 84 (IWCS), pages 366–371, 2007. Short paper.

[23] G. Mishne, N. Glance. Predicting movie sales from blogger sentiment. In AAAI Symposium on Computational Approaches to Analysing Weblogs (AAAI-CAAW), pages 155–158, 2006.

[24] R. Feldman, B. Rosenfeld, R. Bar-Haim. Fresko M. The Stock Sonar—Sentiment Analysis of Stocks Based on a Hybrid Approach. *IAAI-12* (2011), 1642–1647.

[25] E.F. Fama. The behavior of stock-market prices, The Journal of Business 38 (1) (1965) 34–105, http://dx.doi.org/10.2307/2350752.

[26] K. C. Butler, S. J. Malaikah. Efficiency and inefficiency in thinly traded stock markets: Kuwait and Saudi Arabia, Journal of Banking & Finance 16 (1) (1992) 197–210.

[27] R. Narayanan, B. Liu, A. Choudhary. Sentiment analysis of conditional sentences. In Proceedings ofthe 2009 Conference on Empirical Methods in Natural Language Processing (Singapore, 2009). Association for Computational Linguistics, 180–189.

[28] L.A. Gallagher, M.P. Taylor. Permanent and temporary components of stock prices: evidence from assessing macroeconomic shocks, Southern Economic Journal 69 (2) (2002) 345–362, http://www.jstor.org/stable/1061676.

[29] B. Qian, K. Rasheed, Stock market prediction with multiple classifiers, Applied Intelligence 26 (February (1)) (2007) 25–33, http://dx.doi.org/10.1007/s10489-006-0001-7.

[30] H. Choi, H. Varian. Predicting the Present with Google Trends, Tech. rep., Google,2009.

[31] P. Turney. Thumbs up or thumbs down? Semanticorientation applied to unsupervised classificationof reviews. In Proceedings of the Association for Computational Linguistics (2002), 417–424

[32] H.Yu, V. Hatzivassiloglou, Towards answering opinion questions: Separating facts from opinions and identifying the polarity of opinion sentences. In Proceedings of the Conference on Empirical Methods in Natural Language Processing (2003).

[33] B. Pang, L. Lee. A Sentimental Education: Sentiment Analysis using Subjectivity Summarization based on minimum cuts. In Proceedings of the Association for Computational Linguistics (2004), 271–278.

[34] Narayanan, R., Liu, B. and Choudhary, A. Sentiment analysis of conditional sentences. In Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing (Singapore, 2009). Association for Computational Linguistics, 180–189.

[35] M. Hu, B. Liu. Mining and summarizing customer reviews. In Proceedings of the ACM SIGKDD Conference on Knowledge Discovery and Data Mining (2004), 168–177.

[36] A.-M. Popescu, O. Etzioni. Extracting product features and opinions from reviews. In Proceedings of Conference on Empirical Methods in Natural Language Processing (2005).

[37] Z. Hai, K. Chang, J.-j. Kim. Implicit feature identification via co-occurrence association rule mining. Computational Linguistics and Intelligent Text Processing (2011), 393–404.

[38] N. Jindal, B. Liu. Identifying comparative sentences in text documents. In Proceedings of ACM SIGIR Conf. on Research and Development in Information Retrieval (2006).

[39] X. Ding, B. Liu, L. Zhang. Entity discovery and assignment for opinion mining applications. In Proceedings of ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (2009).

[40] J. Kamps, M. Marx, R.J. Mokken, M. de Rijke. Using WordNet to measure semantic orientation of adjectives. LREC, 2004.

[41] V. Hatzivassiloglou, K. McKeown. Predicting the semantic orientation of adjectives. In Proceedings of the Joint ACL/EACL Conference (1997), 174–181.

[42] R.J. Dolan, Emotion cognition, and behavior, Science 298 (5596) (2002) 1191–1194, http://www.sciencemag.org/cgi/content/abstract/298/5596/1191.

[43] A.R. Damasio. Descartes' Error: Emotion Reason, and the Human Brain, Putnam, 1994.

[44] D. Kahneman, A. Tversky. Prospect theory: an analysis of decision under risk, Econometrica 47 (2) (1979) 263–291.

[45] J.R. Nofsinger. Social mood and financial economics, Journal of Behaviour Finance 6 (3) (2005) 144–160.

[46] J. Bollen, H. Mao, X. Zeng. Twitter mood predicts the stock market, Journal of Computational Science.

[47] B. O'Connory, R. Balasubramanyan, B. R. Routledgex, N. A. Smithy. From Tweets to Polls: Linking Text Sentiment to Public Opinion Time Series.

[48] T. Wilson, J. Wiebe, P. Hofimann. Recognizing contextual polarity in phrase-level sentiment analysis, in: Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing—HLT '05 (October), 2005, pp. 347–354.

[49] E. Riloff, J. Wiebe. Learning extraction patterns for subjective expressions, in: Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Morristown, NJ, 2003, pp. 105–112.

[50] E. Riloff, J. Wiebe, T. Wilson. Learning subjective nouns using extraction pattern bootstrapping, in: Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003, Association for Computational Linguistics, Morristown, NJ, 2003, pp. 25–32.

[51] B. Pang, L. Lee. Opinion mining and sentiment analysis, Foundations and Trends in Information Retrieval 2 (1–2) (2008) 1–135.

[52] J.C. Norcross, E. Guadagnoli, J.O. Prochaska. Factor structure of the profile of mood states (POMS): two partial replications, Journal of Clinical Psychology 40 (5) (2006) 1270–1277.

[53] D.M. McNair, J.W.P. Heuchert, E. Shilony. Profile of Mood States. Bibliography 1964–2002, Multi-Health Systems, 2003, https://www.mhs.com/ecom/TechBrochures/POMSBibliography.pdf.

[54] T. Brants, A. Franz. Web 1T 5-gram Version 1, Tech. rep., Linguistic Data Consortium, Philadelphia, 2006.

[55] S. Bergsma, L. Dekang, R. Goebel. Web-scale N-gram models for lexical disambiguation, in: Proceedings of the Twenty-first International Joint Conference on Artificial Intelligence (IJCAI-09), Pasadena, CA, 2009, pp. 1507–1512.

[56] Machine Learning Group at University of Waikato. Weka 3: Data Mining Software in Java, http://www.cs.waikato.ac.nz/ml/weka/.

[57] E. Gilbert, K. Karahalios. Widespread worry and the stock market, in: Fourth International AAAI Conference on Weblogs and Social Media, Washington, DC, 2010, pp. 58–65, http://www.aaai.org/ocs/index.php/ICWSM/ICWSM10/paper/download/1513/1833.

[58] G. Leng, G. Prasad, T.M. McGinnity. An on-line algorithm for creating selforganizing fuzzy neural networks, Neural Networks: The Official Journal of the International Neural Network Society 17 (December (10)) (2004) 1477–1493,

[59] X. Zhu, H. Wang, L. Xu, H. Li. Predicting stock index increments by neural networks: the role of trading volume under different horizons, Expert Systems with Applications 34 (4) (2008) 3043–3054.

[60] T. Kimoto, K. Asakawa, M. Yoda, M. Takeoka. Stock market prediction system with modular neural networks, in: Proceedings of the International Joint Conference on Neural Networks, IEEE, San Diego, CA, 1990, pp. 1–6.