UNIVERSITA DEGLI STUDI DI MODENA E REGGIO EMILIA

CORSO DI LAUREA IN INGEGNERIA INFORMATICA

TESI DI LAUREA MAGISTRALE

MACHINE LEARNING E ANALISI PREDITTIVE: UNA SOLUZIONE SOFTWARE PER IL MARKETING AUTOMATION

Candidato:
Kuicheu Tony Wilson

Relatore:
Chiar.ma Prof. Sonia Bergamaschi

Tesi illustrativa del tirocinio aziendale svolto in

Hopenly S.r.l.

Anno accademico 2015/2016

“Come si nasce bambino e si diventa adulto, per fare grandi cose, bisogna iniziare dalle piccole cose. Guardare sempre in alto e mirare sempre all’obbiettivo carico come Jack Bauer.”

Tony Wilson Kuicheu

Abstract

Oggigiorno, il marketing è diventato fondamentale per le aziende, innanzitutto per farsi conoscere dalla clientela, per valorizzare i propri prodotti e per avere visibilità su un mercato sempre più competitivo. Quindi nasce il bisogno di ottimizzare l'efficacia le campagne marketing . La previsione del futuro risponde a questo bisogno dando loro la possibilità anticipare gli avvenimenti futuri e fare delle scelte sempre più giuste.

In questa relazione di tesi, ci si pone l'obbiettivo di realizzare una soluzione software che sfrutta il machine learning e le analisi predittive per misurare l'impatto delle campagne marketing sulle attività economiche per dare un supporto al decision making. Il software da realizzare è nel senso generale un software di marketing automation. In questa tesi, illustriamo come misurare l'impatto delle campagne marketing grazie alle tecniche di machine learning e diamo l'architettura e le tecnologie che stanno alla base delle soluzione software adottata.

Indice

Capitolo 1

Introduzione

Questa relazione di tesi, illustra l'attività svolta presso l'azienda "Hopenly SRL" inerente alla realizzazione di una soluzione software per il marketing automation. Il software da realizzazione si vede componente di un software proprietario e vasto progetto chiamato "Smarkety".

La soluzione software ha lo scopo principale di fornire una misura o una stima dell'impatto delle campagne marketing sulle attività economiche e si pone l'obbiettivo di dare alle aziende un supporto al decision making per ottimizzare le campagne marketing. La soluzione software vuole usare analisi predittiva e machine learning per raggiungere il suo obbiettivo. In questa relazione di tesi, parliamo di un software di marketing automation in generale in quanto la soluzione si focalizza principalmente sul supporto al marketing.

In questa relazione di tesi, andremo a dare la definizione della soluzione del software e illustreremo come il machine learning e gli analisi predittive possono portare al raggiungimento dei suoi obbiettivi.

L'elaborato di tesi è strutturato come segue:

- Il capitolo 2 raccoglie gli elementi di teoria necessari per comprendere l capitoli successivi.
- Il capitolo 3 analizza due dataset per dare una soluzione in risposta agli obbiettivi da raggiungere.
- Il capitolo 4 presenta la soluzione software, la sua architettura e le tecnologie alla base della sua realizzazione.
- Il capitolo 5: conclude l'elaborato.

Capitolo 2

Teoria

In questo capitolo, diamo una introduzione ai concetti che stanno alla base di questo lavoro di tesi. In particolare, diamo un introduzione alle serie storiche e a le loro previsioni; una definizione di machine learning e poi approfondiremo alcune delle sue tecniche; e infine diamo una introduzione al concetto marketing.

2.1 Machine Learning

Il machine learning è un metodo di analisi dei dati che consente di automatizzare la creazione di un modello analitico. L'apprendimento automatico consente ai computer di trovare intuizioni nascoste senza essere esplicitamente programmato per sapere dove guardare. Altri definiscono il machine learning come metodo che consente ai computer di agire come nei film di fantascienza.

I compiti dell'apprendimento automatico vengono tipicamente classificati in tre ampie categorie. Queste categorie, anche dette paradigmi sono:

- Apprendimento supervisionato: Al computer vengono forniti degli esempi nella forma di possibili input e i rispettivi output desiderati e l'obiettivo è quello di estrarre una regola generale che associ l'input all'output corretto.
- Apprendimento non supervisionato: il computer ha lo scopo di trovare una struttura negli input forniti, senza che gli input vengano etichettati in alcun modo.
- Apprendimento per rinforzo: Il computer interagisce con un ambiente dinamico nel quale cerca di raggiungere un obiettivo (per esempio guidare un veicolo), avendo un insegnante che gli dice solo se ha raggiunto l'obiettivo. Un altro esempio è quello di imparare a giocare un gioco giocando contro un avversario.

A metà strada tra l'apprendimento supervisionato e quello non supervisionato c'è l'apprendimento_semi_supervisionato, nel quale si fornisce un dataset incompleto per l'allenamento, cioè un insieme di dati per l'allenamento tra i quali ci sono dati senza il rispettivo output desiderato.

Considerando output del compito di apprendimento automatico, si ha:

- Nella classificazione, gli input sono divisi in due o più classi e il sistema di apprendimento deve produrre un modello che assegni gli input non ancora visti a una o più di queste. Questo viene affrontato solitamente in maniera supervisionata. Il filtraggio anti-spam è un esempio di classificazione, dove gli input sono le email e le classi sono "spam" e “non spam".

- Nella regressione, che è anch'essa un problema supervisionato, l'output e il modello utilizzati sono continui. Un esempio di regressione è la determinazione della quantità di olio presente in un oleodotto, avendo le misurazioni dell'attenuazione dei raggi gamma che passano attraverso il condotto. Un altro esempio è la predizione del valore del tasso di cambio di una valuta nel futuro, dati i suoi valori in tempi recenti.

- Nel clustering un insieme di input viene diviso in gruppi. Diversamente da quanto accade per la classificazione, i gruppi non sono noti prima, rendendolo tipicamente un compito non supervisionato.

Un algoritmo di machine è un insiemi di passi elementare che portano alla stima di un modello. Un modello di machine learning è una costruzione matematica che contiene parametri che devono essere stimati grazie all’algoritmo.

Attività di economiche(Vendite, ecc.) entrano nella classe delle serie storiche in quanto sono fenomeni che variano con tempo.

La previsione delle serie storiche è un classico caso di regressione, in cui si deve stimare o prevedere l’andamento futura della stessa in funzione delle osservazioni passate e/o in funzione di variabili esplicative o features.

2.2 Introduzione alla previsione delle Serie Temporale

Una serie storica (o temporale) è definita come un insieme di variabili casuali ordinate rispetto al tempo, ed esprime la dinamica di un certo fenomeno nel tempo.
Una serie storica può presentare 3 pattern:

- un trend (o tendenza) quando presenta un andamento crescente o decrescente a lungo termine.
- una stagionalità se influenzata da fattori stagionali come il giorno della settimana o festività.
- dei cicli cioè l'alternanza di fluttuazioni di segno diverso intorno al trend.
- Il rumore bianco: Il rumore bianco si può definire come una distribuzione normale con media zero e deviazione standard pari a uno.

Una serie temporale si dice stazionaria se le sue proprietà non dipendono dal tempo in cui la serie è osservata. Le serie temporali con trend, oppure con stagionalità, non sono stazionarie - il trend e stagionalità influenzeranno il valore della serie storica diverse volte. Una serie rumore (Figura 1) è stazionaria non importa quando si osserva, dovrebbe apparire molto simile in qualsiasi periodo di tempo.

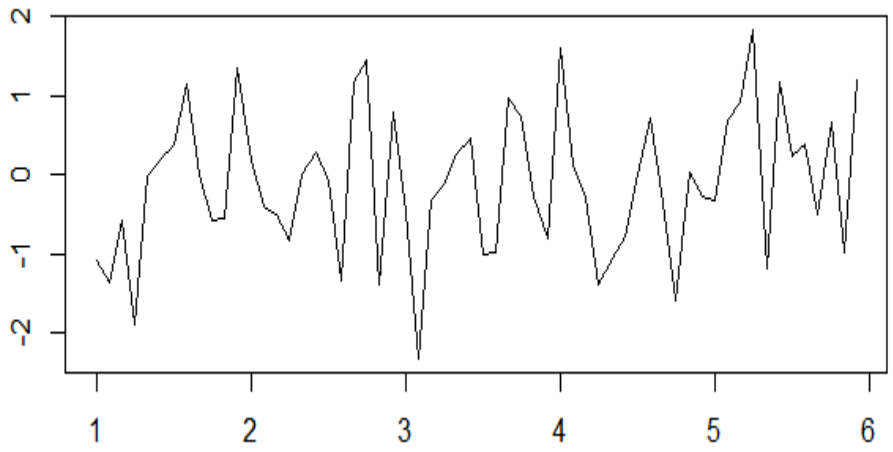

Figura 1 Serie storica rumore bianco

Le serie temporali sono utili quando si prevedono fenomeni che stanno cambiando nel tempo (ad esempio i dati di vendita, profitti, ecc). Esempi di dati di serie temporali includono:

- prezzi giornalieri delle azioni IBM
- precipitazioni mensili
- risultati di vendita trimestrali

Prenderemo in considerazione solo le serie temporali che vengono osservate ad intervalli regolari di tempo (ad esempio, ogni ora, giornaliera, settimanale, mensile, trimestrale, annuale). Serie temporali irregolarmente distanziate possono ovviamente verificarsi, ma non sono oggetto di questa tesi.

La Previsione è il compito di prevedere il futuro nel modo più accurato possibile, dato tutte le informazioni disponibili, compresi i dati storici e la conoscenza degli eventi futuri che potrebbero avere un impatto le previsioni. Non bisogno confondere previsione con obbiettivo e pianificazione. L'obiettivo è ciò che si vuole raggiungere e la pianificazione è una risposta alle previsioni e gli obiettivi e consiste nel determinare le azioni appropriate che sono necessari per rendere le vostre previsioni corrispondono ai suoi obiettivi. Le previsioni possono essere di 3 tipi: previsione a breve termine, previsioni a medio termine e previsioni a lungo termine.

La figura seguente mostra la produzione di birra australiana trimestrale dal 1992 al terzo trimestre del 2008(Figura 2)

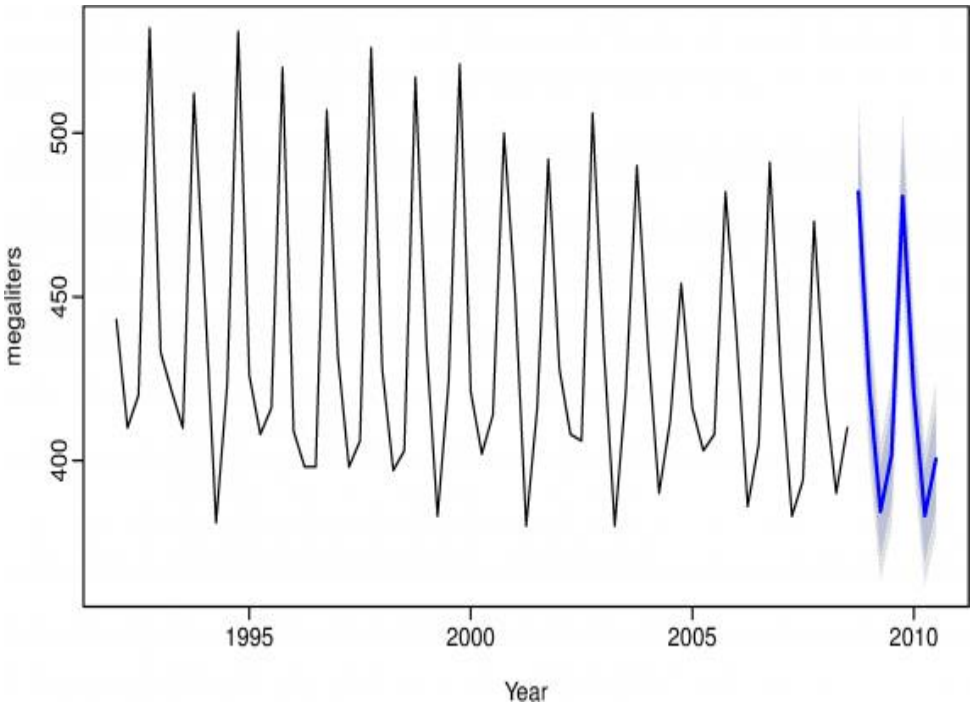

Figura 2: produzione di birra trimestrale australiana: 1992Q1--2008Q3, con due anni di previsioni.

Le linee blu indicano le previsioni per i prossimi due anni. Si noti come le previsioni hanno catturato l'andamento stagionale visto nei dati storici e replicati per i prossimi due anni. Il buio ombreggiato regione mostra intervalli di 80% di previsione. Cioè, ogni valore futuro si prevede risiedere nella regione blu scuro con una probabilità del 80%. La luce ombreggiata regione mostra intervalli di 95% di previsione. Questi intervalli di previsione sono un modo molto utile di visualizzare l'incertezza nelle previsioni. In questo caso, le previsioni dovrebbero essere molto precise, quindi tutti gli intervalli di previsione sono abbastanza stretti.

La previsione di serie temporali utilizza solo le informazioni sulla variabile da prevedere, e non fa alcun tentativo di scoprire i fattori che influenzano il suo comportamento. Pertanto saprà estrapolare tendenze e modelli stagionali, ma ignora tutte le altre informazioni quali iniziative di marketing, attività concorrente, cambiamenti delle condizioni economiche, e così via.

Variabili predittive possono essere utilizzate anche in previsione delle serie temporali. Ad esempio, supponiamo di voler prevedere le prenotazioni di un albergo (A) di una stazione balneare durante il periodo estivo. Un modello con variabili predittive potrebbe essere della forma:

A=f(temperatura attuale, la forza dell'economia, la popolazione, ora del giorno, giorno della settimana, promozioni, errore).

Un compito di previsione di solito comporta cinque fasi fondamentali.

- *Definizione del problema:* Spesso questa è la parte più difficile della previsione. Definire il problema richiede attenzione alla comprensione del modo in cui vengono utilizzate le previsioni e a come la funzione di previsione si inserisce all'interno dell'organizzazione che richiede le previsioni.
- *Raccolta di informazioni:* Ci sono sempre almeno due tipi di informazioni richieste: (a) dati statistici, e (b) l'esperienza accumulata delle persone che raccolgono i dati e utilizzano le previsioni. Spesso, sarà difficile ottenere dati storici sufficienti per poter scegliere un buon modello.
- *Preliminare (esplorativo) analisi:* Spesso, si inizia dalla visualizzazione grafica dei dati. In seguito si cerca se ci sono modelli coerenti, se c'è una tendenza

significativa, se è importante la stagionalità, se c'è evidenza di presenza di cicli economici, se ci sono dei valori anomali nei dati che hanno bisogno di essere spiegato da chi ha conoscenze specifiche, e quanto sono forti le relazioni tra le variabili disponibili per l'analisi. Vari strumenti sono stati sviluppati per aiutare con questa analisi come **R** un linguaggio di programmazione e un ambiente di sviluppo specifico per l'analisi statistica dei dati.

- *Scegliere e fittare il modello:* Il modello migliore da utilizzare dipende dalla disponibilità di dati storici, l'intensità delle relazioni tra la variabile del tempo e le eventuali variabili esplicative, e il modo in cui le previsioni sono da utilizzare. E' comune dovere confrontare due o tre modelli possibili. Ogni modello è di per sé una costruzione artificiale che si basa su una serie di ipotesi (esplicite ed implicite) e di solito comporta uno o più parametri che devono essere stimati utilizzando i dati storici noti.
- *Uso e la valutazione di un modello di previsione:* Una volta che un modello è stato selezionato ed i suoi parametri stimati, il modello è usato per fare previsioni. Le prestazioni del modello può essere adeguatamente valutata solo dopo che i dati per il periodo di previsione sono resi disponibili. Un certo numero di metodi sono stati sviluppati per aiutare a valutare la precisione delle previsioni tra cui il RMSE (Root Mean Squared Error) e il MAE (Mean Absolute Error).

la regolazione dei dati storici può spesso portare a un modello di previsione più semplice. Ad esempio, trasformazioni matematiche, aggiustamenti di calendario, le regolazioni della popolazione e rivalutazione. Lo scopo di tutte queste trasformazioni e aggiustamenti è quello di semplificare i modelli nei dati storici, eliminando le fonti conosciute di variazione o rendendo il modello più coerente in tutta l'intera serie di dati. Le trasformazioni come logaritmi possono aiutare a stabilizzare la varianza di una serie temporale. Invece, la differenziazione può contribuire a stabilizzare la media della serie rimuovendo cambiamenti nel livello, e quindi eliminando trend e stagionalità, con la conseguenza notevole di rendere la serie temporale stazionaria.

La differenziazione(differencing) è l'operazione di calcolare le differenze tra osservazioni consecutive.

$$y'_t = y_t - y_{t-1}$$

La differenziazione di secondo ordine:

$$y''_t = y'_t - y'_{t-1}$$

Ci sono tantissime metriche per valutare la previsione di una serie storica tra cui:

- La media degli errori assoluti (MAE, *Mean Absolute Error*). E' una grandezza utilizzata per misurare quanto le previsioni o predizioni sono vicini alle eventuali esiti. L'errore medio assoluto è dato da:

  (1) $$\mathrm{MAE} = \frac{1}{n}\sum_{i=1}^{n}|f_i - y_i| = \frac{1}{n}\sum_{i=1}^{n}|e_i|\,.$$

  Come suggerisce il nome, l'errore medio assoluto è una media degli errori assoluti $|\,\mathbf{f_i} - \mathbf{y_i}\,|$, dove $\mathbf{f_i}$ è la previsione e $\mathbf{y_i}$ il vero valore. L'errore medio assoluto utilizza la stessa scala dei dati misurati. Questo è noto come una misura di precisione scala-dipendente e pertanto non può essere utilizzato per fare confronti tra serie utilizzando diverse scale.

- La media dei quadrati degli errori (MSE, *Mean Squared Error*): MSE misura la qualità di un predittore. MSE del predittore può essere stimata da:

  (2) $$\mathrm{MSE} = \frac{1}{n}\sum_{i=1}^{n}(\hat{Y}_i - Y_i)^2$$

  Vale a dire che il MSE è la *media* del *quadrato degli errori*.

  Questa è una quantità facilmente calcolabile per un particolare campione (e quindi è dipendente dal campione).

- La radice quadrata del precedente (RMSE, *Root Mean Squared Error)**:*** RMSE è la radice quadrata delle MSE.

In questa sezione si introduce gli algoritmi o modelli di machine learning che sono state studiati per la previsione delle serie storiche: modelli Arima, Modelli a livellamento esponenziale, XGboost (Extreme Gradient Boosting), Modelli di reti neurali.

2.2.1 Modelli ARIMA

Un modello auto regressivo integrato a media mobile (ARIMA) è uno modello specifico per dati di serie storiche ed è utile sia per capire meglio i dati o per prevedere i punti futuri nella serie (previsione). Modelli ARIMA sono applicati in alcuni casi in cui i dati mostrano evidenza di non stazionarietà, dove una fase differencing iniziale (corrispondente alla parte "integrata" del modello) può essere applicato una o più volte per eliminare le non stazionarietà.

La parte AR (Auto Regressive) di ARIMA indica che la variabile di interesse è regredita su propri valori ritardati (cioè anteriori). La parte MA (Moving Average) indica che l'errore di regressione è una combinazione lineare di termini d'errore cui valori sono verificati contemporaneamente e in diversi momenti del passato. I (per "integrata") indica che i valori dei dati sono state sostituiti con la differenza tra i loro valori e i valori precedenti (e questo processo di differenziazione può essere eseguito più di una volta). Lo scopo di ciascuna di queste caratteristiche è quello di permettere al modello di adattare i dati nel miglior modo possibile.

Modelli ARIMA non stagionali sono generalmente indicati ARIMA (p, d, q) dove i parametri p, d, e q sono numeri interi non negativi, p è l'ordine (numero di sfasamenti temporali) del modello auto regressivo, d è il grado di differenziazione (il numero di volte in cui i dati hanno avuto valori passati sottratti), e q è l'ordine del modello a media mobile. Modelli. I modelli stagionali ARIMA si denotano usualmente ARIMA (p, d, q) (P, D, Q) m, dove m indica il numero di periodi in ogni stagione, e le lettere maiuscole P, D, Q si

riferiscono alla auto regressione, differenziazione, e lo spostamento dei termini medi per la parte stagionale del modello ARIMA.

Quando due dei tre termini sono zero, il modello può essere definito sulla base del parametro non-zero, lasciando cadere "AR", "I" o "MA" dall'acronimo descrivendo il modello. Ad esempio, ARIMA (1,0,0) è AR (1), ARIMA (0,1,0) è I (1), e ARIMA (0,0,1) viene MA (1).

Data una serie temporale di dati X_t cui t è un indice intero e X_t sono numeri reali, un modello ARMA(p, q) è dato da:

(3) $$X_t - \alpha_1 X_{t-1} - \cdots - \alpha_{p'} X_{t-p'} = \varepsilon_t + \theta_1 \varepsilon_{t-1} + \cdots + \theta_q \varepsilon_{t-q},$$

equivalente ad:

(4) $$\left(1 - \sum_{i=1}^{p'} \alpha_i L^i\right) X_t = \left(1 + \sum_{i=1}^{q} \theta_i L^i\right) \varepsilon_t$$

L è l'operatore ritardo, α_i sono i parametri della parte auto regressiva del modello, θ_i sono i parametri della parte media mobile e la sono termini di errore. I termini di errore sono generalmente presume essere indipendenti e identicamente distribuite, variabili campionate da una distribuzione normale con media zero.

Supponiamo ora che il polinomio $\left(1 - \sum_{i=1}^{p'} \alpha_i L^i\right)$ ha una radice unitaria (un fattore $(1 - L)$) con molteplicità *d*. Si può riscrivere il polinomio come:

(5) $$\left(1 - \sum_{i=1}^{p'} \alpha_i L^i\right) = \left(1 - \sum_{i=1}^{p'-d} \phi_i L^i\right) (1 - L)^d.$$

Un processo ARIMA (*p*, *d*, *q*) esprime questa struttura di fattorizzazione polinomiale con *p* = *p'*-*d*, ed è data da:

$$(6) \qquad \left(1 - \sum_{i=1}^{p} \phi_i L^i\right)(1 - L)^d X_t = \left(1 + \sum_{i=1}^{q} \theta_i L^i\right) \varepsilon_t$$

e quindi può essere pensato come un caso particolare di un ARMA (p + d, q) avente il polinomio auto regressivo con d radici unitarie.

Nel nostro caso, per estendere il modello ARIMA consentendo altri variabili esplicative, si considera una semplice combinazione di modelli di regressione e modelli ARIMA per dare la regressione con errori ARIMA. Questi sono poi estesi nella classe generale di modelli di regressione dinamici.

Si considera un modello di regressione lineare data da:

$$(7) \qquad y_t = \beta_0 + \beta_1 x_{1,t} + \cdots + \beta_k x_{k,t} + e_t,$$

dove y_t è una funzione lineare di k variabili $(x_{1,t}, x_{2,t}, \ldots, x_{k,t})$ e e_t è l'errore che si assume non correlato cioè un rumore bianco. Se si suppone che l'errore segue un modello ARIMA(1,1,1), allora il modello completo sarà data da:

$$(8) \qquad y_t = \beta_0 + \beta_1 x_{1,t} + \cdots + \beta_k x_{k,t} + e_t,$$

$$(9) \qquad (1 - \phi_1 B)(1 - B) e_t = (1 + \theta_1 B)\varepsilon_t$$

dove ε_t è il rumore bianco.

2.2.2 Modelli a livellamento esponenziale(Exponential Smoothing)

Exponential Smoothing è stato proposto alla fine del 1950 e ha motivato alcuni dei metodi di previsione di maggior successo. Previsioni ottenuti con i metodi di livellamento esponenziale sono medie ponderate di osservazioni passate, con i pesi che decrescono in

modo esponenziale quando le osservazioni invecchiano. In altre parole, più recente è l'osservazione maggiore è il peso associato .

Un esempio semplice di modello a livellamento esponenziale è il livellamento esponenziale semplice (SSE) che si applica maggiormente sulle serie temporali senza tendenze e stagionalità. La previsione è ottenuta la media ponderata delle osservazioni passate con pesi che decrescono in modo esponenziale dalle osservazioni più recenti a quelle più vecchie:

(10) $$\hat{y}_{T+1|T} = \alpha y_T + \alpha(1-\alpha)y_{T-1} + \alpha(1-\alpha)^2 y_{T-2} + \cdots,$$

Questo modello (SSE) può essere rappresentato mediante 2 equazioni:

(11) $$\hat{y}_{t+1|t} = \ell_t$$

(12) $$\ell_t = \alpha y_t + (1-\alpha)\ell_{t-1},$$

dove ℓ_t è il livello (o il valore livellato) della serie al momento t. L'equazione di previsione (11) mostra che il valore previsto al tempo t + 1 è il livello previsto al momento t. L'equazione smoothing (12) per il livello (solitamente indicato come equazione livello) indica il livello stimato della serie ad ogni tempo t.

Il livellamento esponenziale semplice ha una funzione di previsione "piatta", e quindi costante per gli orizzonti di previsione più lunghi:

(13) $$\hat{y}_{T+h|T} = \hat{y}_{T+1|T} = \ell_T, \qquad h = 2, 3, \ldots.$$

Ci sono modelli a livellamento esponenziale che estendono il livellamento esponenziale semplice per consentire la previsione di dati con una tendenza e/o una stagionalità come quelli di Holt-Winters..

Considerando variazioni nella combinazione di tendenza e componenti stagionali, quindici metodi di livellamento esponenziale sono possibili, elencati nella Tabella 1 . Ogni metodo è etichettato con una coppia di lettere (T, S) che definiscono il tipo di 'Trend'

e componenti 'stagionali'. Ad esempio, (A, M) è il metodo con andamento additivo e di stagionalità moltiplicativa;(M, N) è il metodo con tendenza moltiplicativo e non stagionalità; e così via.

Stagionalità	N(Nessuna)	A(Additivo)	M(Moltiplicativo)
Trend			
N (Nessuno)	(N, N)	(N, A)	(N, M)
A (additivo)	(A, N)	(A, A)	(A, M)
A_d (additivo smorzata)	(A_d, N)	(A_d, A)	(A_d , M)
M (moltiplicativo)	(M, N)	(M, A)	(M, M)
M_d (smorzata moltiplicativo)	(M_d, N)	(M_d, A)	(M_d , M)

Tabella 1: Metodi di livellamento esponenziale.

La Figura 3 fornisce le formule ricorsive per l'applicazione di tutte le possibili quindici metodi di livellamento esponenziale. Ciascuna cella comprende l'equazione previsione per generare h previsioni -step-ahead e le equazioni di livellamento per l'applicazione del metodo.

Trend	**Seasonal**		
	N	**A**	**M**
N	$\hat{y}_{t+h\|t} = \ell_t$ $\ell_t = \alpha y_t + (1-\alpha)\ell_{t-1}$	$\hat{y}_{t+h\|t} = \ell_t + s_{t-m+h_m^+}$ $\ell_t = \alpha(y_t - s_{t-m}) + (1-\alpha)\ell_{t-1}$ $s_t = \gamma(y_t - \ell_{t-1}) + (1-\gamma)s_{t-m}$	$\hat{y}_{t+h\|t} = \ell_t s_{t-m+h_m^+}$ $\ell_t = \alpha(y_t/s_{t-m}) + (1-\alpha)\ell_{t-1}$ $s_t = \gamma(y_t/\ell_{t-1}) + (1-\gamma)s_{t-m}$
A	$\hat{y}_{t+h\|t} = \ell_t + hb_t$ $\ell_t = \alpha y_t + (1-\alpha)(\ell_{t-1} + b_{t-1})$ $b_t = \beta^*(\ell_t - \ell_{t-1}) + (1-\beta^*)b_{t-1}$	$\hat{y}_{t+h\|t} = \ell_t + hb_t + s_{t-m+h_m^+}$ $\ell_t = \alpha(y_t - s_{t-m}) + (1-\alpha)(\ell_{t-1} + b_{t-1})$ $b_t = \beta^*(\ell_t - \ell_{t-1}) + (1-\beta^*)b_{t-1}$ $s_t = \gamma(y_t - \ell_{t-1} - b_{t-1}) + (1-\gamma)s_{t-m}$	$\hat{y}_{t+h\|t} = (\ell_t + hb_t)s_{t-m+h_m^+}$ $\ell_t = \alpha(y_t/s_{t-m}) + (1-\alpha)(\ell_{t-1} + b_{t-1})$ $b_t = \beta^*(\ell_t - \ell_{t-1}) + (1-\beta^*)b_{t-1}$ $s_t = \gamma(y_t/(\ell_{t-1} + b_{t-1})) + (1-\gamma)s_{t-m}$
$\mathbf{A_d}$	$\hat{y}_{t+h\|t} = \ell_t + \phi_h b_t$ $\ell_t = \alpha y_t + (1-\alpha)(\ell_{t-1} + \phi b_{t-1})$ $b_t = \beta^*(\ell_t - \ell_{t-1}) + (1-\beta^*)\phi b_{t-1}$	$\hat{y}_{t+h\|t} = \ell_t + \phi_h b_t + s_{t-m+h_m^+}$ $\ell_t = \alpha(y_t - s_{t-m}) + (1-\alpha)(\ell_{t-1} + \phi b_{t-1})$ $b_t = \beta^*(\ell_t - \ell_{t-1}) + (1-\beta^*)\phi b_{t-1}$ $s_t = \gamma(y_t - \ell_{t-1} - \phi b_{t-1}) + (1-\gamma)s_{t-m}$	$\hat{y}_{t+h\|t} = (\ell_t + \phi_h b_t)s_{t-m+h_m^+}$ $\ell_t = \alpha(y_t/s_{t-m}) + (1-\alpha)(\ell_{t-1} + \phi b_{t-1})$ $b_t = \beta^*(\ell_t - \ell_{t-1}) + (1-\beta^*)\phi b_{t-1}$ $s_t = \gamma(y_t/(\ell_{t-1} + \phi b_{t-1})) + (1-\gamma)s_{t-m}$
M	$\hat{y}_{t+h\|t} = \ell_t b_t^h$ $\ell_t = \alpha y_t + (1-\alpha)\ell_{t-1}b_{t-1}$ $b_t = \beta^*(\ell_t/\ell_{t-1}) + (1-\beta^*)b_{t-1}$	$\hat{y}_{t+h\|t} = \ell_t b_t^h + s_{t-m+h_m^+}$ $\ell_t = \alpha(y_t - s_{t-m}) + (1-\alpha)\ell_{t-1}b_{t-1}$ $b_t = \beta^*(\ell_t/\ell_{t-1}) + (1-\beta^*)b_{t-1}$ $s_t = \gamma(y_t - \ell_{t-1}b_{t-1}) + (1-\gamma)s_{t-m}$	$\hat{y}_{t+h\|t} = \ell_t b_t^h s_{t-m+h_m^+}$ $\ell_t = \alpha(y_t/s_{t-m}) + (1-\alpha)\ell_{t-1}b_{t-1}$ $b_t = \beta^*(\ell_t/\ell_{t-1}) + (1-\beta^*)b_{t-1}$ $s_t = \gamma(y_t/(\ell_{t-1}b_{t-1})) + (1-\gamma)s_{t-m}$
$\mathbf{M_d}$	$\hat{y}_{t+h\|t} = \ell_t b_t^{\phi_h}$ $\ell_t = \alpha y_t + (1-\alpha)\ell_{t-1}b_{t-1}^{\phi}$ $b_t = \beta^*(\ell_t/\ell_{t-1}) + (1-\beta^*)b_{t-1}^{\phi}$	$\hat{y}_{t+h\|t} = \ell_t b_t^{\phi_h} + s_{t-m+h_m^+}$ $\ell_t = \alpha(y_t - s_{t-m}) + (1-\alpha)\ell_{t-1}b_{t-1}^{\phi}$ $b_t = \beta^*(\ell_t/\ell_{t-1}) + (1-\beta^*)b_{t-1}^{\phi}$ $s_t = \gamma(y_t - \ell_{t-1}b_{t-1}^{\phi}) + (1-\gamma)s_{t-m}$	$\hat{y}_{t+h\|t} = \ell_t b_t^{\phi_h} s_{t-m+h_m^+}$ $\ell_t = \alpha(y_t/s_{t-m}) + (1-\alpha)\ell_{t-1}b_{t-1}^{\phi}$ $b_t = \beta^*(\ell_t/\ell_{t-1}) + (1-\beta^*)b_{t-1}^{\phi}$ $s_t = \gamma(y_t/(\ell_{t-1}b_{t-1}^{\phi})) + (1-\gamma)s_{t-m}$

Figura 3 : Formule ricorsive dei metodi a livellamento esponenziale.

l_t denota il livello della serie al momento t, b_t denota la pendenza nel tempo t, s_t denota

la componente stagionale della serie al momento t e m indica il numero di stagioni in un anno; $\gamma, \alpha, \phi, \beta^*$ sono parametri del livellamento.

2.2.3.Il modello XGboost

XGboost è una libreria software open-source che fornisce il Framework Gradient Boosting per C ++, Java, Python, R, e Julia. Funziona su Linux, di Windows, e MacOS. Oltre all'esecuzione su una singola macchina, supporta anche i Framework di elaborazione distribuita Apache Hadoop, Apache Spark, e Apache Flink. Ha guadagnato molta popolarità e attenzione di recente per quanto è stato l'algoritmo scelto per molte squadre vincitrici di numerosi concorsi di machine learning.
XGboost si basa sul modello originale proposto da "*Friedman nella sua pubblicazione: Greedy Function Approximation: A Gradient Boosting Machine".*

XGboost viene utilizzato per problemi di apprendimento supervisionato.
XGboost è un insieme di alberi: un insieme di alberi di classificazione e di alberi di regressione.
Un esempio di albero di regressione è illustrato in Figura 4 che classifica se a qualcuno piacerà giochi per computer.

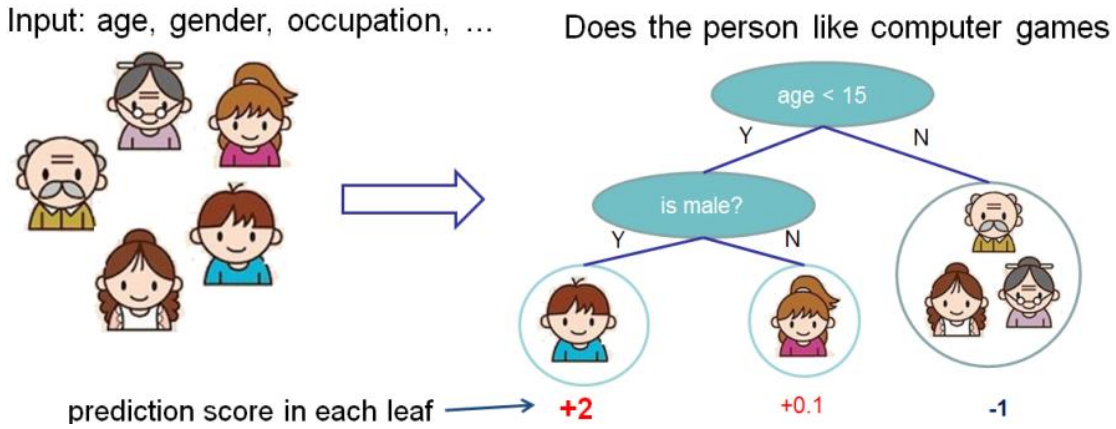

Figura 4 : Albero che classifica gli amanti dei giochi per computer

In questo esempio(Figura 4), si classifica i membri di una famiglia in diverse foglie, e si assegna a loro il punteggio sul corrispondente foglia. Di solito, un solo albero non è abbastanza forte per essere utilizzato in pratica. Ciò che è effettivamente utilizzato è il cosiddetto modello insieme d'alberi(es. Figura 5), che riassume la previsione di più alberi.

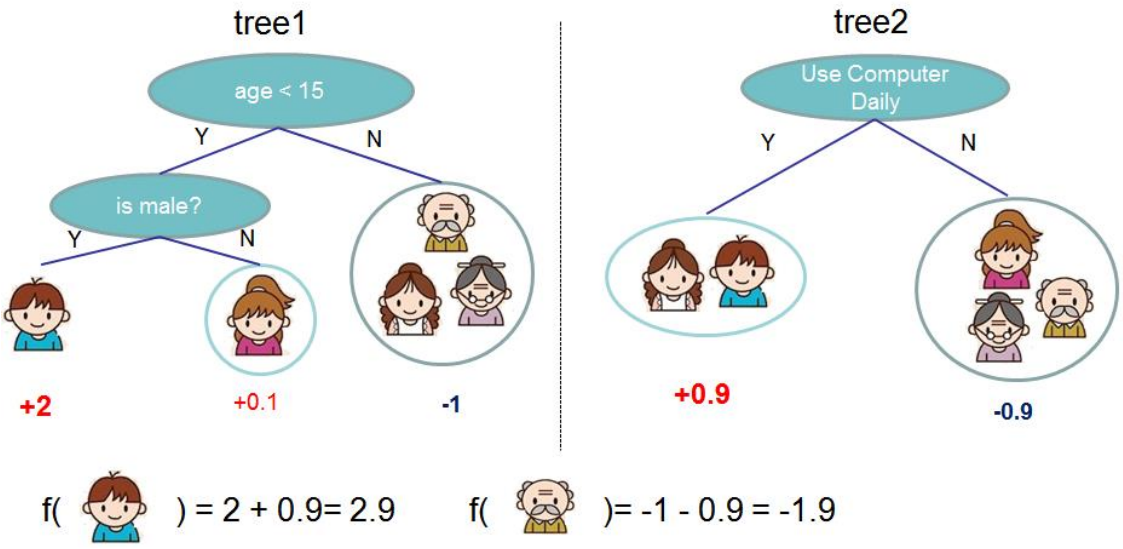

Figura 5: Due alberi di regressione per classificare amanti di giochi al computer.

I valori predetti da ogni singolo albero vengono sommati per ottenere il punteggio finale Come illustrato poco fa, L'algoritmo XGboost produce un grosso numero di questi alberi e si distingue dagli altri(es. Random Forest) nella sua fase di training.

2.2.4 Modelli di reti neurali

Una rete neurale può essere pensato come una rete di "neuroni" organizzati in strati. I predittori (o input) formano lo strato inferiore, e le previsioni (o uscite) formano lo strato superiore. Ci possono essere strati intermedi che contengono "neuroni nascosti".

Le reti molto più semplici non contengono livelli nascosti e sono equivalenti a regressione lineare. Figura 6 mostra la versione rete neurale di una regressione lineare con quattro predittori. I coefficienti legati a queste predittori sono chiamati "pesi". Le previsioni sono

ottenuti da una combinazione lineare degli ingressi. I pesi sono scelti nell'ambito rete neurale utilizzando un "algoritmo di apprendimento" che minimizza una "funzione di costo", come MSE.

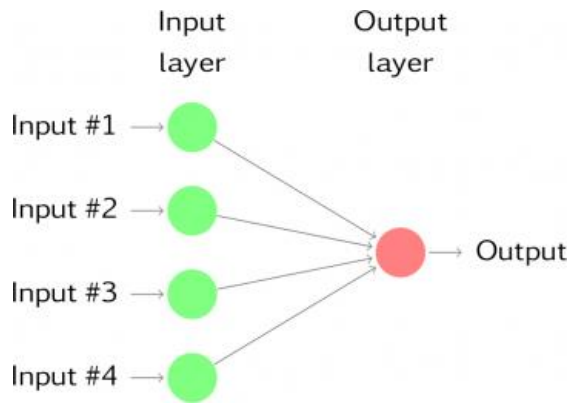

Figura 6: Una semplice rete neurale equivalente ad una regressione lineare

Quando si aggiunge uno strato intermedio con neuroni nascosti, la rete neurale diventa non lineare. Un semplice esempio è mostrato nella Figura 7.

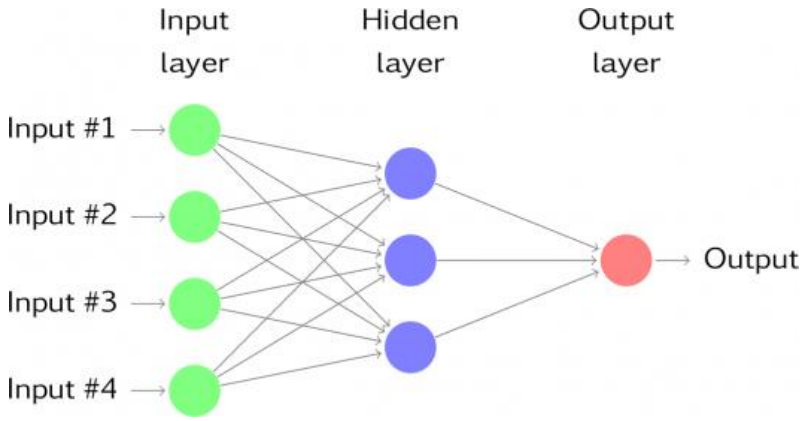

Figura 7:Una rete neurale con quattro ingressi e uno strato nascosto con tre neuroni nascosti.

Questo è noto come una *rete di feed-forward multistrato* dove ogni strato di nodi riceve input dagli strati precedenti. Le uscite di nodi in uno strato sono ingressi allo strato successivo. Gli ingressi per ogni nodo sono combinati utilizzando una combinazione

lineare ponderata. Il risultato viene poi modificato da una funzione non lineare prima dell'invio.

Con dati di serie temporali, i valori delle osservazioni passate possono essere utilizzati come ingressi ad una rete neurale.
Si considera le reti feed-forward con uno strato nascosto, utilizziamo la notazione NNAR(p, k) per indicare che ci sono p ingressi ritardati e k nodi nello strato nascosto. Ad esempio, un NNAR(9, 5) è un modello di rete neurale con le ultime nove osservazioni utilizzate come ingressi per prevedere l'uscita con cinque neuroni nello strato nascosto.

Inoltre, con dati stagionali, è utile aggiungere anche gli ultimi valori osservati dal medesimo stagione come ingressi. Ad esempio, un modello NNAR (3, 1, 2)12 ha 4 ingressi e due neuroni nello strato nascosto. Più in generale, un modello
NNAR(p, P, k)m ha p + P ingressi (y_{t-1} , y_{t-2}, y_{t-p}, y_{t-1m}, y_{t-2m}, y_{t-Pm}) e k neuroni nello strato nascosto.

2.2.5 Scelta del Modello e Tradeoff Bias-Variance

Il trade off bias-varianza è il problema di minimizzare contemporaneamente due fonti di errore che impediscono che algoritmi di apprendimento supervisionato generalizzano oltre il loro training set:

- *Bias:* diciamo che un modello ha un elevato bias se non è in grado di utilizzare pienamente le informazioni nei dati. Fa troppo affidamento sulle informazioni generali, come ad esempio il caso più frequente, la media della risposta, o alcune features potenti. Il Bias può essere generato da presupposti sbagliati, per esempio assumendo che le variabili abbiano una distribuzione normale o che il modello sia lineare. Un elevato Bias può causare una perdita dei rapporti rilevanti tra le feature e le uscite del bersaglio (underfitting).
- *La varianza:* Diciamo che un modello ha un'alta varianza se sta usando troppe informazioni dai dati. Essa si basa su informazioni che sono rilevanti solo nel set di training che è stato utilizzato ad esso e non generalizza abbastanza bene. In genere, il modello cambierà molto se si cambia il set di training, da cui il nome "alto contrasto".

Alta varianza può causare overfitting: modellazione del rumore casuale nei dati di addestramento, piuttosto che le uscite previste.

Il compromesso Bias-varianza è un problema centrale per l'apprendimento supervisionato. Idealmente, si vuole scegliere un modello che cattura sia accuratamente le regolarità nei suoi dati di training, ma generalizza anche bene con i dati non visti. Purtroppo, è in genere impossibile fare entrambe le cose contemporaneamente.

Per la scelta del modello nella previsione, ci si può affidare alle metriche come il MAE, RMSE, MSE per valutare quanto bene predice un modello.

L 'accuratezza delle previsioni può essere determinata solo considerando quanto bene un modello esegue su nuovi dati che non sono stati utilizzati durante il training del modello. Al momento di scegliere i modelli, è comune utilizzare una parte dei dati disponibili per il. training, e poi i dati di prova(i rimanenti) possono essere utilizzati per misurare quanto bene il modello prevede su nuovi dati.

Una versione più sofisticata di training e test è la cross validation. Per le serie storiche, la cross validation consiste a selezionare un tempo di osservazione T in cui si suppone che l'algoritmo considerato produca un modello con un'accuratezza di previsione accettabile sui dati di training set(tutte le osservazioni con tempo di osservazione P inferiore a T) e si procede come segue:

1. Selezionare l'osservazione al tempo T + i. come set di prova, e utilizzare le osservazioni rimanenti fino al tempo T + i - 1 come training set. Calcolare l'errore sull'osservazione di prova.
2. Ripetere la stessa operazione per i=1,2…,T-N dove N è il numero totale di osservazioni.
3. Calcolare l'accuratezza delle previsioni con gli errori ottenuti.

Questa procedura è spesso conosciuta come il rolling forecasting origin.

Per questa tesi, usiamo la variante che prevede un nuova stima del modello ad ogni passo come descritto in precedenza.

Un altro metodo per verificare la bontà di un modello è l'analisi dei residui. Un residuo è la differenza tra il valore osservato e quello predetto. I residui di un buon modello di previsione deve avere le proprietà:

- I residui non correlati. Se ci sono correlazioni tra residui, allora ci sono informazioni lasciato nei residui che devono essere utilizzate nella previsione.
- I residui con media zero. Se i residui hanno una media diversa da zero, allora le previsioni sono distorti.

2.3.Marketing

Una delle definizioni di marketing più adottate è quella data da Philip Kotler nel 1967: "Il marketing è quel processo sociale e manageriale diretto a soddisfare bisogni ed esigenze attraverso processi di creazione e scambio di prodotti e valori. È l'arte e la scienza di individuare, creare e fornire valore per soddisfare le esigenze di un mercato di riferimento, realizzando un profitto."

Altra definizione, data dalla American Marketing Association, è la seguente: Il marketing è il processo che pianifica e realizza la progettazione, la politica dei prezzi, la promozione e la distribuzione di idee, beni e servizi volti a creare mercato e a soddisfare obiettivi di singoli individui e organizzazioni.

Vengono riconosciuti tre tipi di marketing:

- marketing analitico: studio del mercato, della clientela, dei concorrenti e della propria realtà aziendale;
- marketing strategico: è un'attività di pianificazione, tradotta in pratica da un'impresa, per ottenere, pur privilegiando il cliente, la fedeltà e la collaborazione da parte di tutti gli attori del mercato.
- marketing operativo: attiene invece a tutte quelle scelte che l'azienda pone in essere per raggiungere i suoi obiettivi strategici

Più formalmente, Il marketing analitico può essere definito come un insieme di tecniche e strumenti utilizzati per la raccolta e l'analisi di informazioni rilevanti sul business dell'impresa, il cui fine è quello di fornire un supporto nei processi decisionali aziendali.

Si tratta di attività che, favorendo una migliore comprensione dei trend di mercato, creano le condizioni affinché l'impresa possa definire la strategia più adatta per raggiungere gli obiettivi prefissati e, di conseguenza, orientare le politiche da attuare nei confronti della clientela. Le attività di marketing analitico presuppongono l'impiego di un marketing information system per la raccolta e l'analisi di dati di mercato e per lo sviluppo di modelli comportamentali della clientela che aiutano ad esaminare il profilo, i comportamenti e gli atteggiamenti di specifici segmenti di mercato che si intende raggiungere.

Le tecniche di machine learning possono servire a misurare il ROI(Return of Investments) sul marketing operativo e più specificamente sulle campagne marketing.
Qui, parliamo di misurare l'effetto delle campagne marketing o dare una previsione dello stesso.
Per marketing automation intendiamo l'uso del software per automatizzare processi di marketing come la segmentazione dei clienti, l'integrazione dei customer data e il campaign management. Il marketing automation permette alle aziende di centralizzare l'esecuzione di funzionalità come l'email marketing, il web analytics, la creazione di landing pages, la segmentazione, il list management, il lead nurturing,il lead scoring e le campagne multi canale.
Il processo di misurazione e di previsione dell'effetto degli investimenti sulle campagne marketing agevola il.decision.making permettendo alle aziende di ottimizzare le azione marketing e quindi incrementare i ricavi.
In questa tesi, diamo la definizione di una soluzione software di supporto al marketing.

Capitolo 3

Analisi Predittive per le Serie Storiche E Attività di Marketing

Partendo da un problema di marketing per ottimizzare le campagne e valutare il rendimento, si vuole trovare un modello di serie storiche che predica le vendite sulla base delle promozioni, campagne pubblicitarie, meteo ed eventi di calendario come giorni festivi ed eventi culturali o sportivi. Si suppone che le campagne di marketing da analizzare si sono già svolte nel passato almeno una volta e che si abbia dati al riguardo.

Nel costruire il modello, si allena la serie storica seguendo ogni modello qui sopra indicato e scegliere il migliore. Nello scoprire il modello, capiremo l'importanza delle campagne marketing su una determinata attività economica. Nell'utilizzare il modello, l'utente vorrebbe sapere quale e quando effettuare un'attività di marketing.

Il modello corretto dovrà produrre previsioni il più accurato possibile. Non si fa nessun ipotesi sui dati in ingresso, ma mediante un processo di backtracking si va a scegliere il modello migliore tra tutti i modelli considerati sulla basse dei valori ottenuti dalla valutazione di accuratezza dei modelli. Questa scelta analitica risponde bene ai le caratteristiche di ciò che vogliamo realizzare cioè un software in quanto non prevede una intervenzione umana nel processo di analisi.

Si usa R per svolgere le analisi a seguire. R è un linguaggio di programmazione open-source e un ambiente per la computazione statistica e la grafica.

Consideriamo due dataset di prova, il primo non contiene dati di campagne marketing ma illustra e presenta le librerie e funzioni R che si usano, il secondo invece contiene dati campagne marketing nello specifico promozioni. IL primo dataset è il dataset delle vendite settimanali delle sigarette in un supermercato negli stati uniti e il secondo il dataset delle vendite settimanali delle luci di Natale in un supermercato negli stati uniti.

Valuteremo modelli di machine learning introdotte in precedenza quindi modelli Arima, modelli a livellamento esponenziale, modelli Xgboost e modelli di reti neurali e sceglieremo il modello migliore in due casi: nel caso di un 2 step ahead forecasts cioè previsione di due settimane delle vendite e nel caso 8 step ahead forecasts cioè una previsione di 8 settimane delle vendite. Inoltre, si introduce anche le funzioni e le librerie R che si usano. Dividiamo il dataset in 90% di training set e 10% di test set.

Per ogni modello, in primis, valuteremo il modello con la cross validation e poi faremo le previsioni con il modello. L'implementazione degli algoritmi in R prevedono parametri per allenare e ottimizzare il modello, in questa tesi ci limitiamo solo a presentare i risultati.

3.1 Analisi delle vendite delle sigarette

Si considera il dataset delle vendite delle sigarette in un supermercato negli stati uniti(Figura 8).

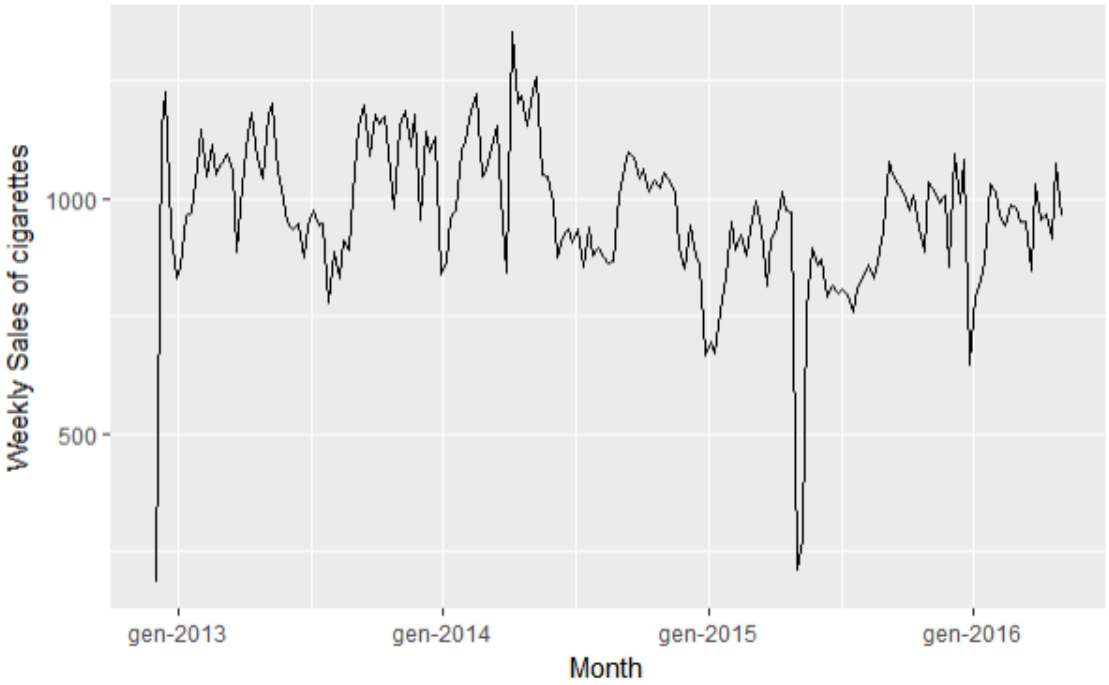

Figura 8: Numero di vendite settimanali di sigarett

IL dataset contiene gli attributi

- "Weekly_Sales" il numero di vendite settimanali di sigarette in pacchetto venduto.
- "Sales". L'ammontare delle vendite in dollari($).
- "Date": le date settimanali delle osservazioni.
- "Rain": un flag booleano che indica se durante la settimana c'è piovuto.
- "Snow": un flag booleano che indica se durante la settimana è nevicato.
- "Temp": le temperature medie settimanali in grado celsius.
- "IsHoliday": flag booleano indice se la settimana considerata era una settimana di vacanza.

Date	IsHoliday	Promo	Rain	Snow	Temp	Sales	Weekly_Sales
2012-12-01	0	1	1	0	3	1179.77	185
2012-12-08	0	1	1	0	8	7448.10	1151
2012-12-15	0	1	1	0	6	7895.57	1226
2012-12-22	0	1	1	0	7	5979.46	930
2012-12-29	1	1	1	1	2	5339.47	831
2013-01-05	1	1	1	0	1	5558.69	857
2013-01-12	0	1	1	0	5	6267.33	963
2013-01-19	0	1	1	0	4	6256.66	971
2013-01-26	0	1	0	1	-3	6910.34	1071
2013-02-02	0	1	1	1	3	7388.74	1148
2013-02-09	0	1	1	1	0	6729.06	1045
2013-02-16	1	1	1	1	4	7197.48	1114
2013-02-23	1	1	1	1	0	6751.70	1050
2013-03-02	0	1	1	1	4	6858.20	1076
2013-03-09	0	1	1	1	3	7072.94	1096
2013-03-16	0	1	1	0	6	6820.88	1057
2013-03-23	0	1	1	1	3	5688.92	886
2013-03-30	0	1	1	1	5	6616.86	1028
2013-04-06	1	1	1	0	6	7329.15	1140
2013-04-13	0	1	1	0	17	7542.99	1184
2013-04-20	0	1	1	0	15	7016.56	1087
2013-04-27	0	1	1	0	11	6643.88	1044

Figura 9: dataset delle vendite delle sigarette.

Le osservazioni nel dataset vanno dal 1-12-2012 fino al 30-04-2016, sono in totale 179. L'andamento delle vendite nella Figura 8 non sembra presentare un trend particolare e non sembra presentare una stagionalità. Ci sono picchi anomali verso il basso come alla prima osservazione e nel mese di aprile 2015. Questi picchi(outlier) possono essere dovuti a degli errori nella rilevazione delle vendite .

La decomposizione della serie storica invece mostra la presenza di un trend decrescente a scala e presenta una stagionalità molto variabile.(Figura 10).

Figura 10: Decomposizione delle vendite di sigarette in trend e stagionalità

Un' analisi della distribuzione delle vendite (Figura 11) mostra una sua variazione tra un minimo di 600 e un massimo di 1400 con un picco tra 900 e 1000. Inoltre, si nota un piccolo numero isolato di osservazioni con vendite tra 100 e 300. Questi sono considerati anomali(outlier) perché avvengono con poco frequenza.

La Figura 12 mostra le vendite dopo l'eliminazione degli outlier cioè dopo la sostituzione del loro valore con la media delle vendite.

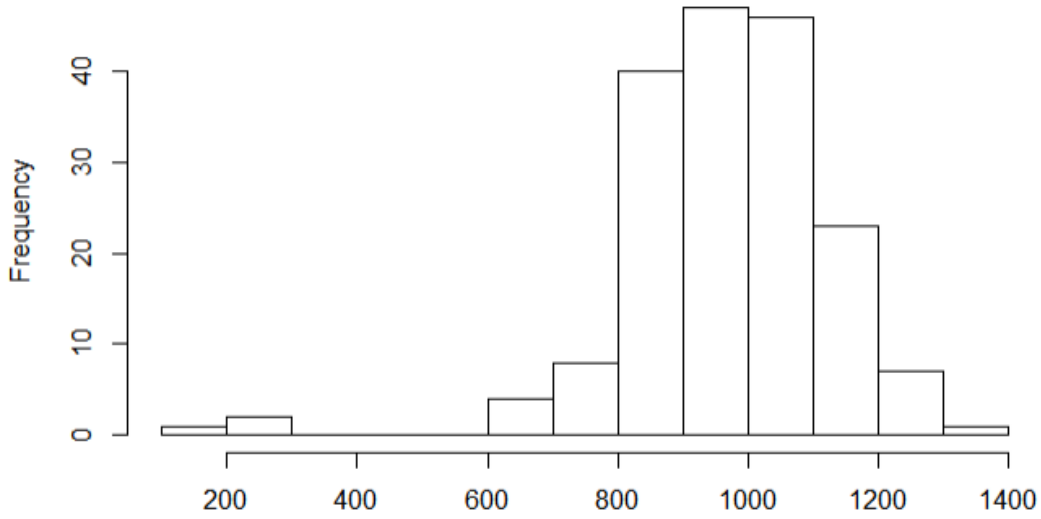

Figura 11: distribuzione delle vendite delle sigarette.

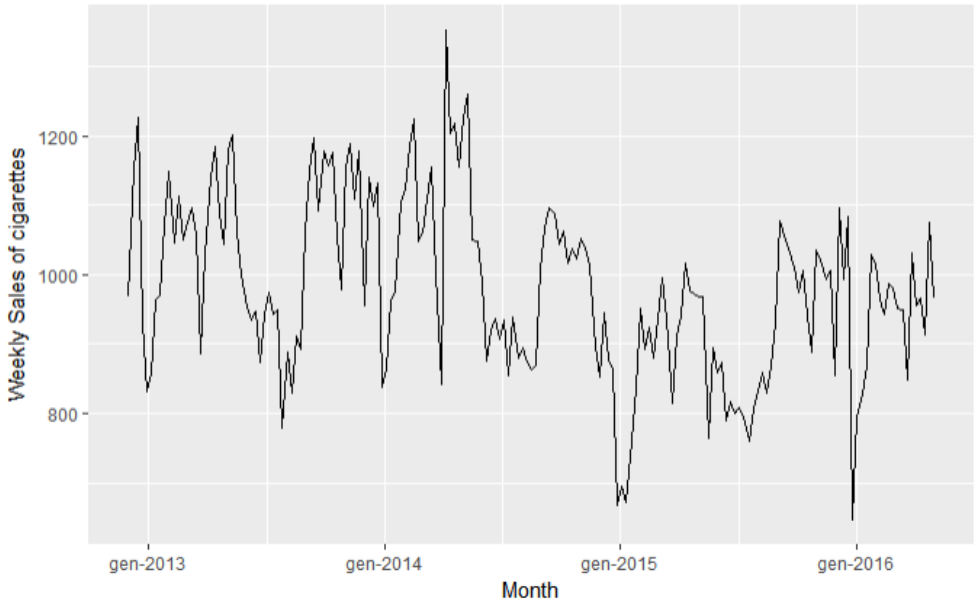

Figura 12: Vendite di sigarette dopo l'eliminazione dei valori anormali

Il dataset di training con 90% delle osservazioni cioè 162 osservazioni e il test set 17 osservazioni. Le osservazioni di test sono le ultime osservazioni della serie temporale.

3.1.1 Analisi con modelli ARIMA

In R, si usa la funzione *auto.arima* dello package *forecast*. *auto.arima* stima il modello Arima migliore della serie temporale sulla base in generale dell'AIC(Akaike's Information Criterion). L'AIC è un metodo per la valutazione e il confronto tra modelli statistici, è basato sul concetto di entropia dell'informazione e offre una misura relativa di informazioni perse quando un dato modello è usato per descrivere la realtà.
La regola è quella di preferire i modelli con l'AIC più basso.

Nel caso delle serie storiche, la cross validation(CV) si può definire come un h step ahead forecast with re-estimation. Quindi la cross-validation dà un'idea di come prevede i modelli nel futuro.
La Figura 13 e Figura 14 mostra l'andamento dell'errore con la cross validation::

- 2 step ahead forecasts

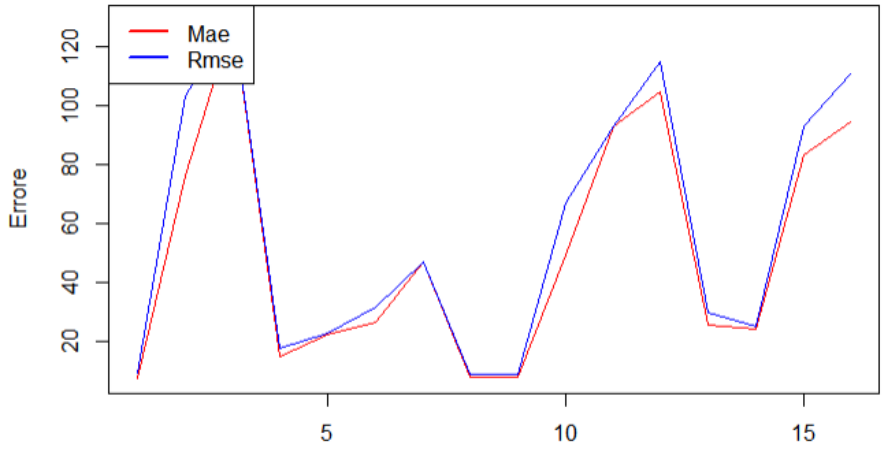

Figura 13: 2-step ahead forecasts error for Arima (CV).

L'andamento dell'errore è molto variabile tra casi di successo e casi di non successo. Il MAE ha un minimo di 7.505 e un massimo di 129.278, invece L'RMSE ha un minimo di 8.7 e un minimo di 129.7.

- 8 step ahead forecasts

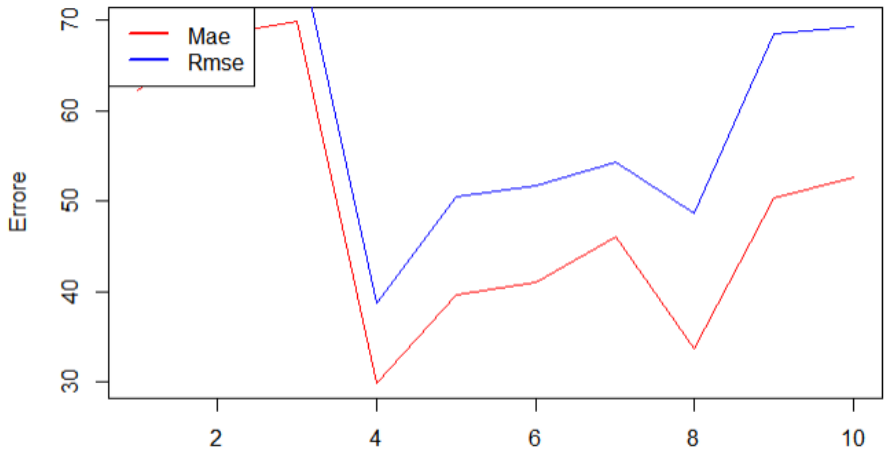

Figura 14: 8-step ahead forecasts error for Arima(CV)

Arima sbaglia molto all'inizio ma migliora. Il MAE ha un minimo di 29.6 e un massimo di 69.86 e l' RMSE ha un minimo di 38 e un massimo di 80.79. Si nota che l'errore massimo in questo caso è minore rispetto al caso 2 step ahead forecasts

La Tabella 2 riassume le media del MAE e RMSE ottenute:

Tabella 2: Sintesi errore Arima(CV)

ARIMA	2 step forecast	8 step forecast
MAE	50.954	48,21
RMSE	57.115	68.44

Mediamente Arima prevede con una differenza di 50.95 e 48,21 dal valore osservato rispettivamente per il 2 step e 8 step forecasting.

La Figura 15 e la Figura 16 mostrano le previsioni rispettivamente su di 2 settimane e 8 settimane. Si nota subito un intervallo di previsione molto ampio, cioè una alta incertezza nella previsione. Abbiamo ottenuto un modello ARIMA(1,1,1). Questa modello ci informa che la serie storica non è stagionale e non stazionaria.

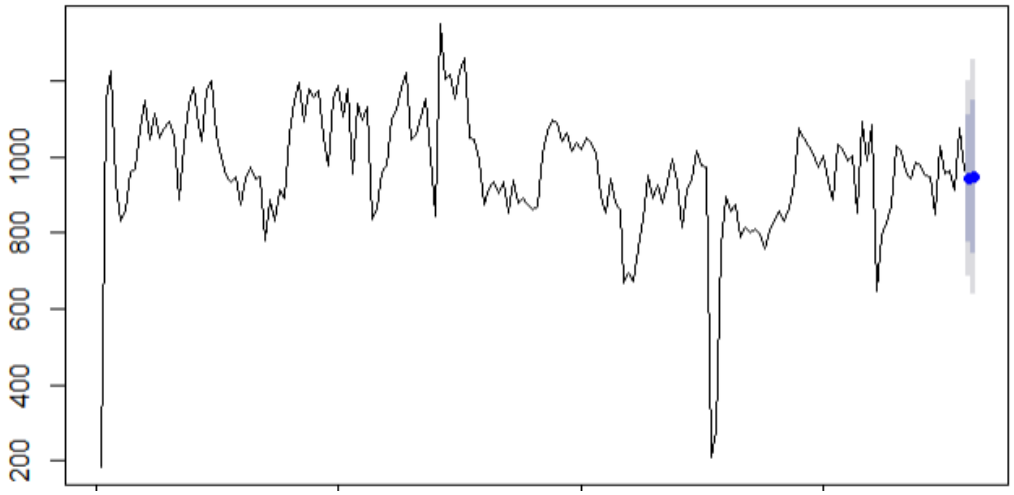

Figura 15: 2 step ahead forecasts con Arima delle vendite delle sigarette

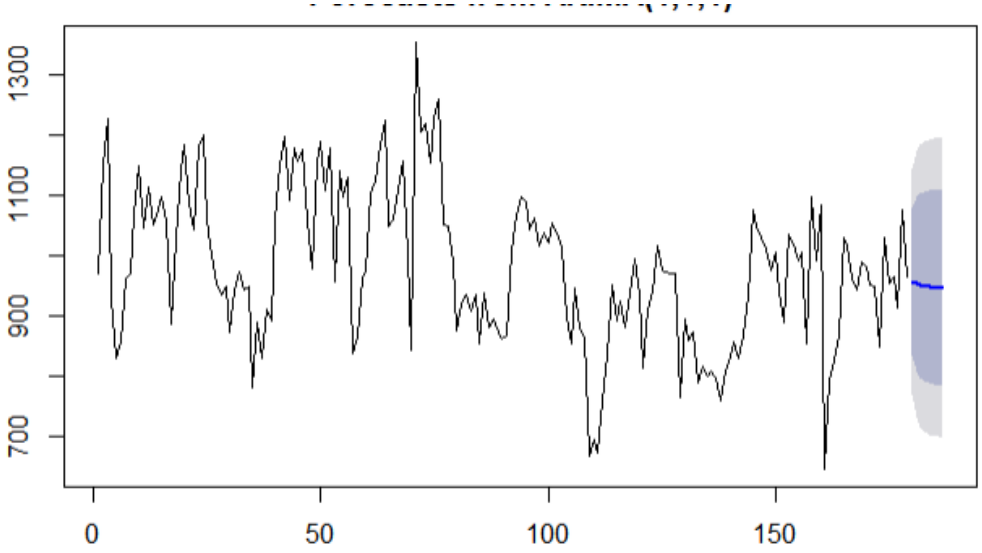

Figura 16: 8 step ahead forecasts con Arima delle vendite delle sigarette.

Non verrà allenato un modello di regressione lineare con errori Arima in quanto una verifica del coefficiente R squared tra le vendite e le features: “Rain”, “Snow”, “IsHoliday”, “Temp” mostra la poco relazione lineare tra le features e le vendite. Il coefficiente R squared è pari a 0.0146.

3.1.2 Analisi con modelli a livellamento esponenziale

In R, si usa la funzione *ETS* della libreria *forecast*. Come nel caso di Arima *ETS* ricava il metodo a livellamento esponenziale più adatto alla serie temporale. *ETS* no prevede l'uso di variabili esplicative per la stima del modello.

La Figura 17 e la Figura 18 mostrano l'andamento dell'errore con la cross validation.

- 2 step Forecast

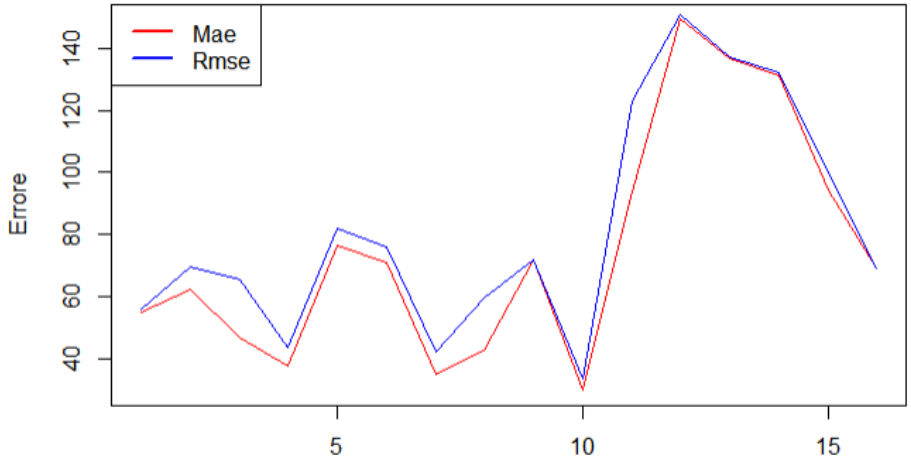

Figura 17: 2 step ahead forecast error for Exponential Smoothing(CV)

Il MAE varia tra un minimo di 29,71 e un massimo 149,49 e l'RMSE invece va da un minimo di 33.51 ad un massimo di 105.87.

- 8 step forecast

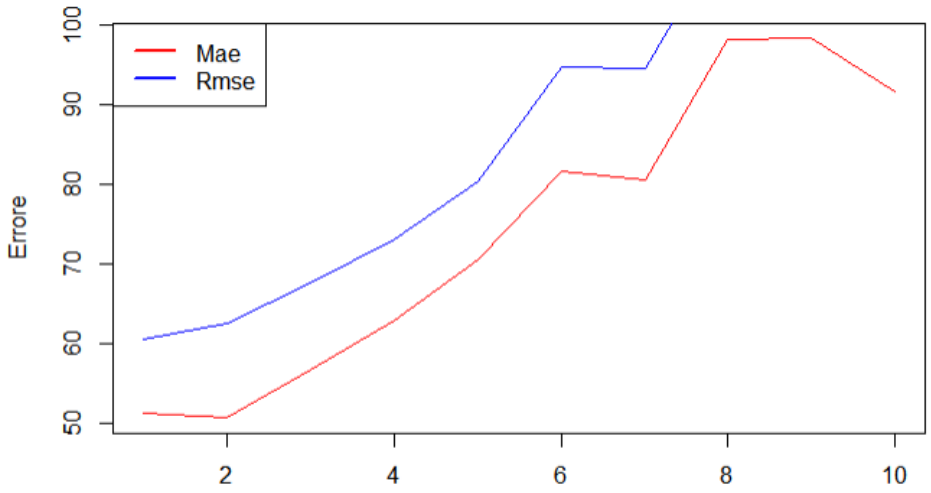

Figura 18: 8 step ahead forecast for Exponential smoothing(CV)

Il MAE varia tra un minimo di 35 e un massimo 95,70 . L'RMSE invece va da un minimo di 48 ad un massimo di 107,33.

La Tabella 3 riassume la media del MAE e RMSE ottenuto:

Tabella 3: Sintesi errore Exponential Smoothing(CV)

Exponential Smoothing	2 step Forecast	8 step forecast
MAE	69,93	74,28
RMSE	70,84	85.99

In media, il livellamento esponenziale sbaglia di più rispetto ad 'Arima.

La Figura 19 e la Figura 20 si vede l'andamento delle previsione con il modello a livellamento esponenziale ottenuto. Le previsioni catturano l'andamento della generale della serie storica e gli intervalli di previsioni sono piuttosto stretti. Il modello ottenuto è un modello a livellamento esponenziale semplice.

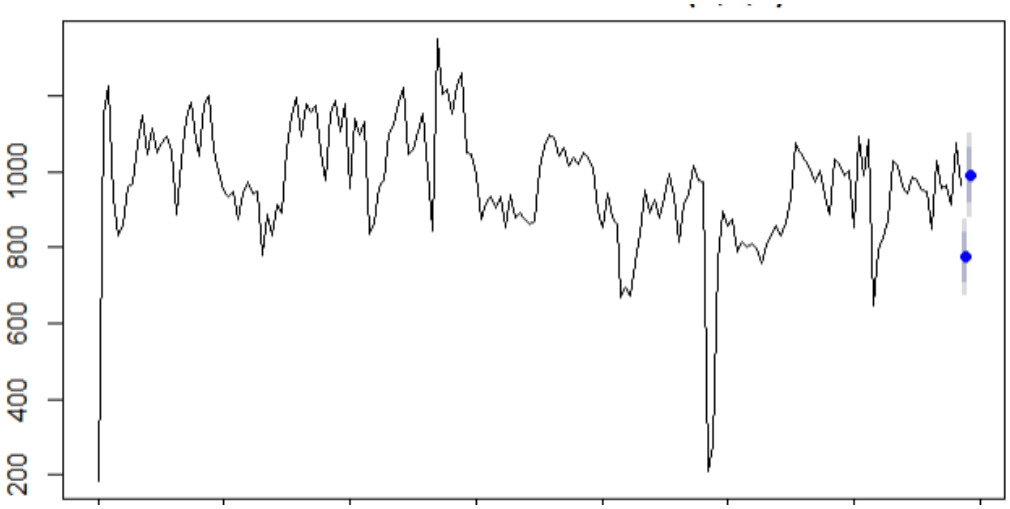

Figura 19: Previsione di 2 settimane delle vendite di sigarette con Exponential Smoothing.

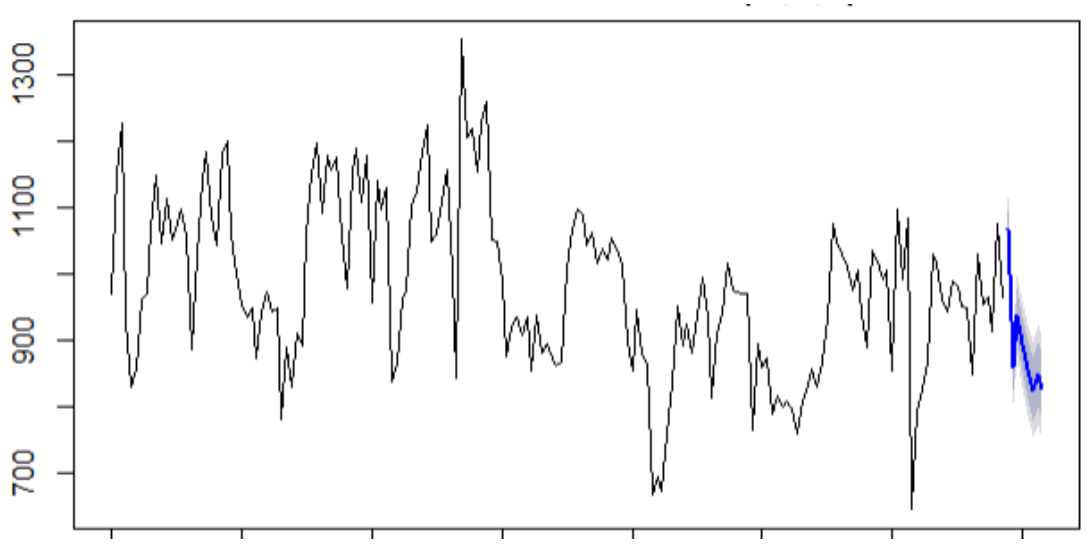

Figura 20: 8 step ahead forecast con il livellamento esponenziale.

3.1.3 Analisi con modelli Xgboost

In R, la libreria *xgboost* rende disponibile le funzioni necessarie per allenare un modello Xgboost. Xgboost è molto utile perché diversamente da Arima e l'Exponential Smoothing, usa un insieme ottimizzato di alberi e quindi identifica relazioni non lineari nei dati. Per allenare un modello Xgboost per le serie temporali, si ricava alcuni pattern importanti nella data di osservazione come il mese, il giorno del mese, il giorno del l'anno e se li usa come feature. Questi pattern hanno lo scopo di catturare il trend e la stagionalità nei dati. In questo esempio, si usa il mese e il giorno del mese come variabili esplicative. Il mese varia tra 1 e 12 invece il giorno del mese varia tra 1 e 31 a seconda del mese. Per allenare la serie temporale, abbiamo quindi le seguenti features: "mese", "giorno del mese", "IsHoliday", "Rain", "Snow", "Temp".

Gli errori ottenuti dalla cross validation sono i seguenti:

- 2 step Forecast

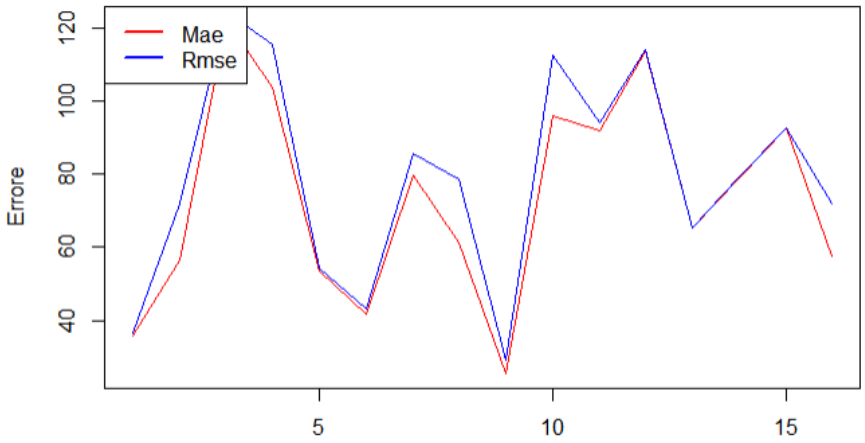

Figura 21: 2 step ahead forecast error for Xgboost(CV)

Il MAE varia tra un minimo di 25 ad un massimo di 122,21. L'RMSE invece varia tra un 28,91 e un massimo di 124,53.

- 8 step forecast

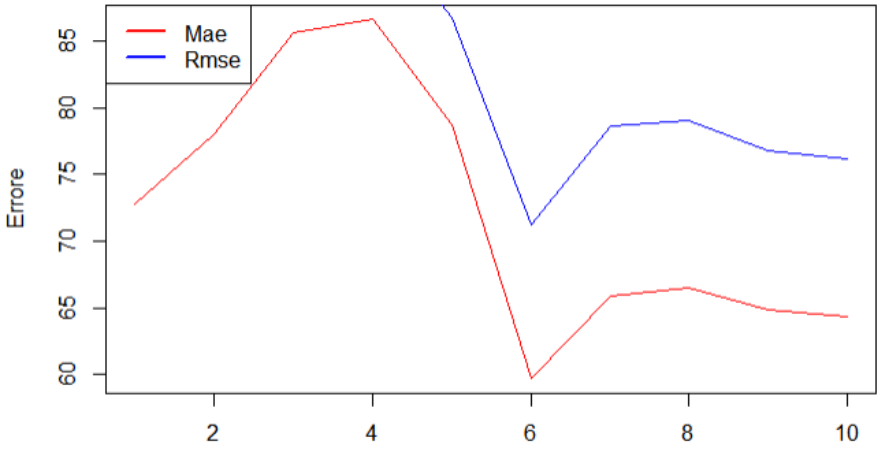

Figura 22: 8 step ahead forecast for Xgboost(CV)

Il MAE varie tra un minimo di 59 ad un massimo di 86.60 e l'RMSE tra 71,26 e un massimo di 94,33.

La Tabella 4 riassume la media del MAE e RMSE ottenuto:

Tabella 4: Sintesi errori con Xgboost(CV)

Xgboost	2 step forecast	8 step forecast
MAE	73,47	72,30
RMSE	79,35	82,28

L’errore medio è più elevato rispetto a Arima ma non molto differente rispetto al livellamento esponenziale. La libreria xgboost fornisce una matrice di importanza il quale mostra l’importanza di ogni feature nella stima del modello. In questo caso, la temperatura è la più importante.(Figura 23).

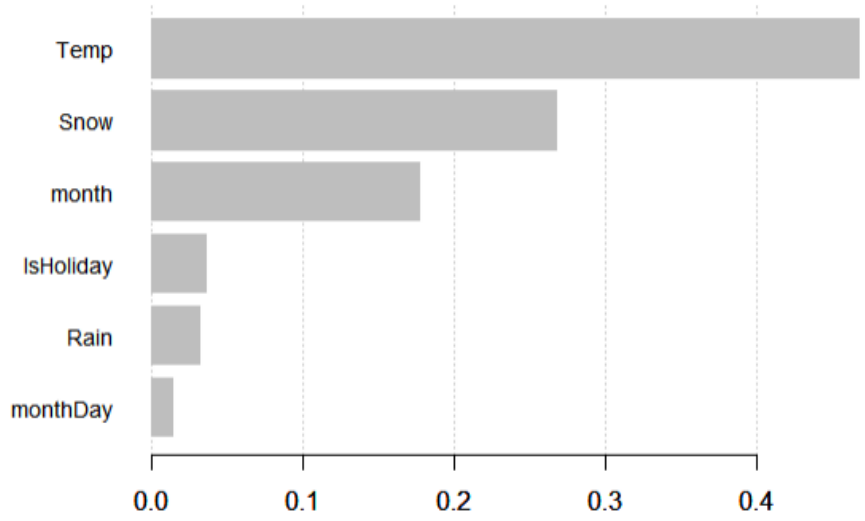

Figura 23: Matrice di importanza Xgboost nelle vendite delle sigarette.

La Figura 24 mostra la previsione 2 step ahead forecasts con il modello Xgboost. Notiamo che Xgboost prevede un andamento verso il basso come nel caso a livellamento esponenziale.

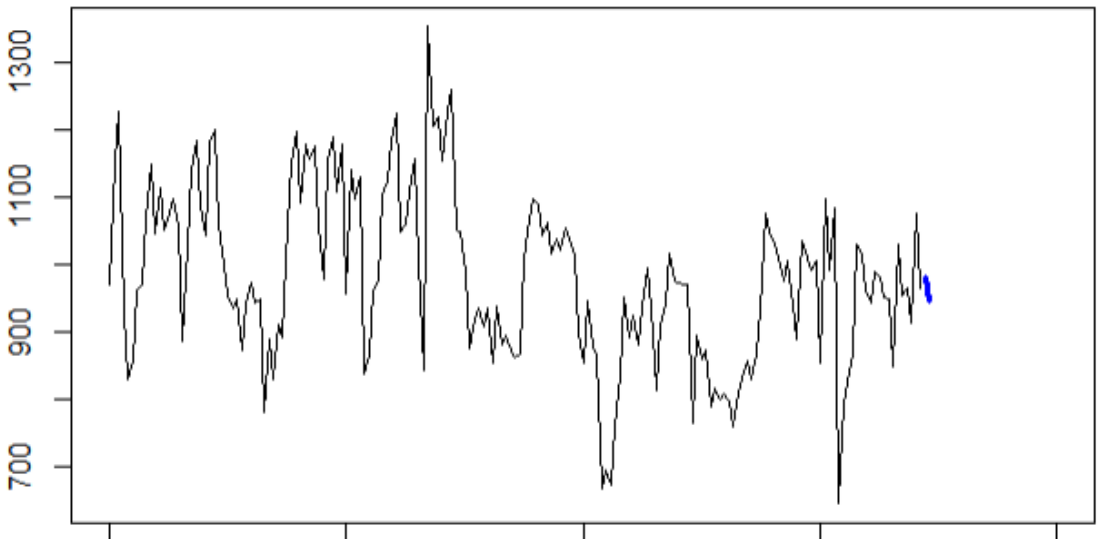

Figura 24: 2 step ahead forecasts con Xgboost delle vendite delle sigarette.

Nella previsione di 8 settimane, si vede che Xgboost riesce a catturare l'andamento generale della serie storica.

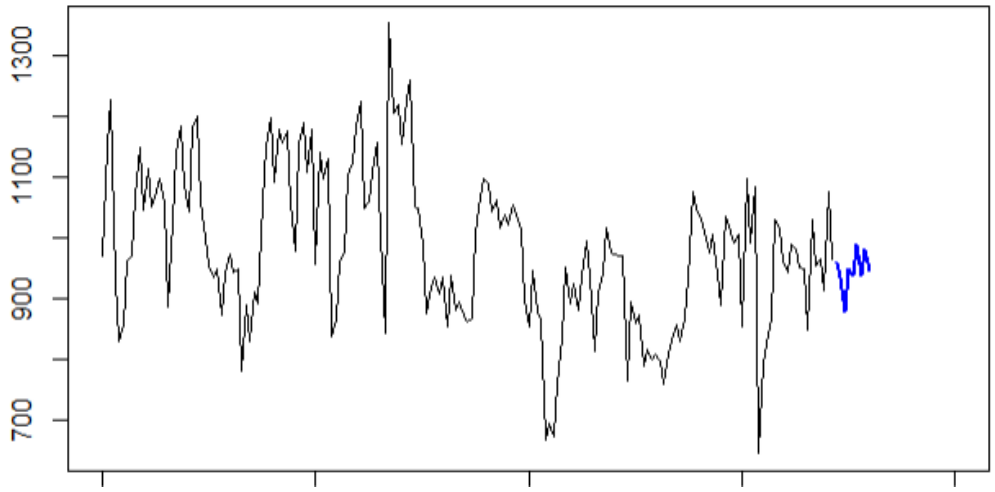

Figura 25: 8 step ahead forecast con Xgboost delle vendite delle sigarette.

3.1.4 Analisi con modelli di reti Neurali Auto regressive

In R, si usa la funzione *nnetar*. La funzione *nnetar* usando i varie parametri che gli vengono passati, sceglie un modello NNAR per la serie temporale. In generale anche *nnetar* sceglie il modello sulla base dell'AIC.

Gli errori ottenuti durante la cross validation sono i seguenti:

- 2 step ahead forecast

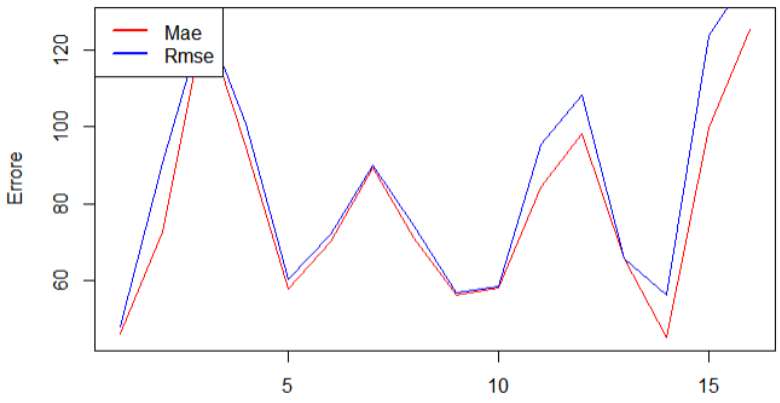

Figura 26: 2 step ahead forecast error NNAR(CV)

Il MAE varia tra 56 e 77 e l'RMSE varia tra 61 e 98. Come negli altri casi, l'errore nel futuro è molto variabile. Tutti hanno casi di successo e non successo.

- 8 step forecast

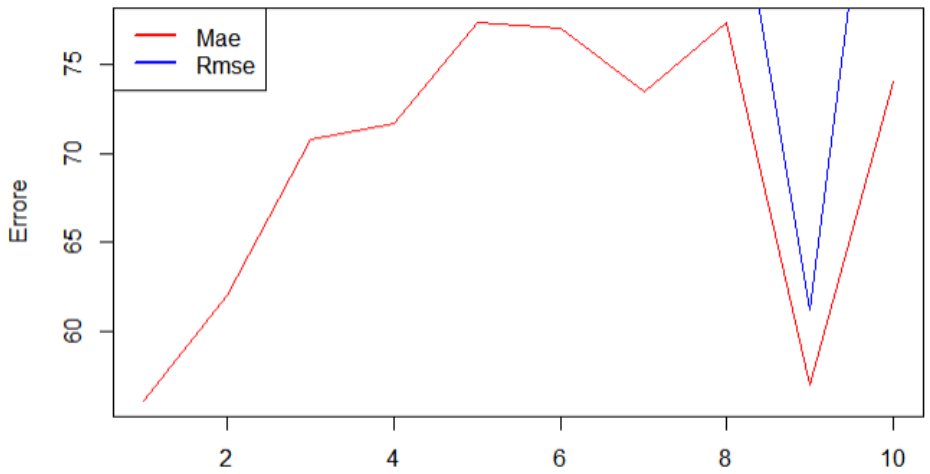

Figura 27: 8 step forecast error NNAR(CV).

Il MAE varia tra 43,27 e 127,77 e l'RMSE varia tra 48,29 e 139,89

La Tabella 5 riassume la media del MAE e RMSE ottenuta:

Tabella 5: Sintesi errore per NNAR(CV)

NNAR	2 step Forecast	8 step Forecast
MAE	72	72
RMSE	88	82,38

In media, NNAR è migliore rispetto a Xgboost ma non in maniere significativa. NNAR prevede su 8 settimane quanto su 2 settimane.

Abbiamo ottenuto un modello NNAR(5,1,4)[52] con 6 ingressi e 4 neuroni nello strato nascosto. Prevede un andamento in basso per la previsione di 2 settimane di vendite come Xgboost e il livellamento esponenziale(Figura 28). Nel caso di 8 settimane di previsione, si nota aumenti e diminuzioni nelle vendite. NNAR cattura l'andamento generale della serie storica.(Figura 29).

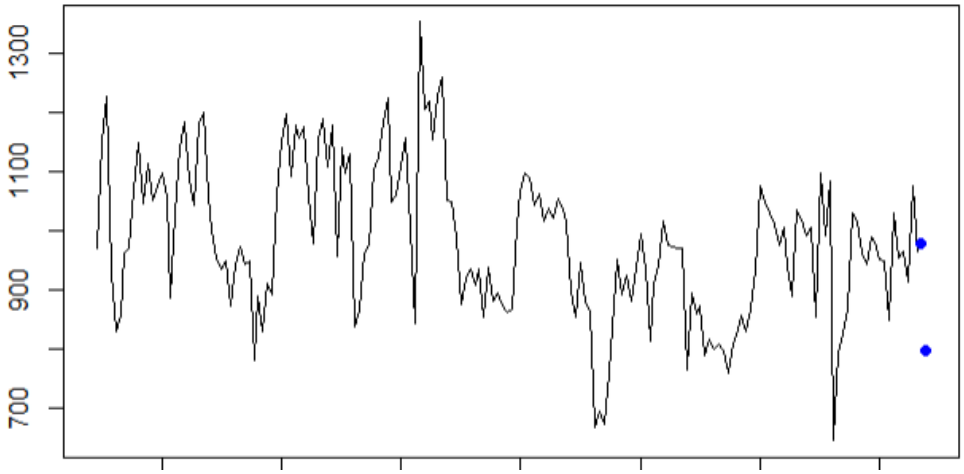

Figura 28: 2 step ahead forecasts con NNAR delle vendite delle sigarette.

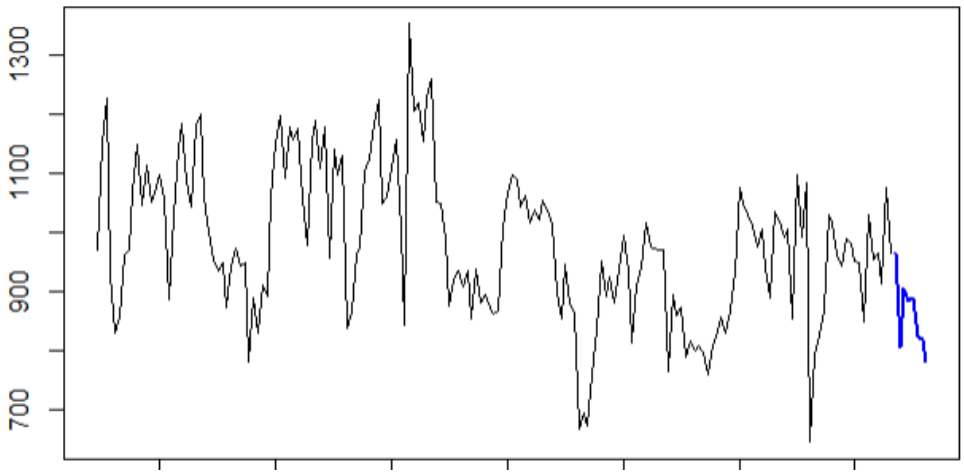

Figura 29: 8 step ahead forecasts con NNAR delle vendite delle sigarette.

3.1.5 Scelta del Modello

La *Tabella 6* mostra una sintesi delle valutazioni fatte in precedenza.

Tabella 6: Sintesi accuratezza dei modelli per le vendite delle sigarette.

2 step ahead forecasts	Arima	E. Smoothing	Xgboost	NNAR
MAE	50,95	75,13	73,47	72
RMSE	57,11	82,00	79,35	88

8 step ahead forecasts				
MAE	48,21	68	72,30	72
RMSE	68,44	81	82,28	82,5

Considerando il MAE e L'RMSE in tabella, Arima è il modello migliore per questa serie storica sia nel caso delle previsioni su 2 settimane che su 8 settimane(quasi 2 mesi).

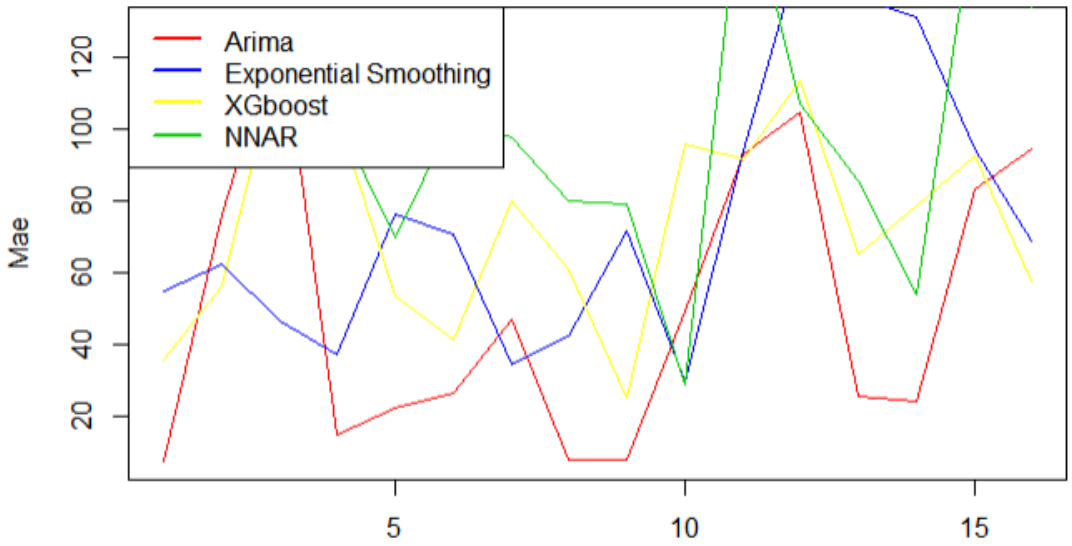

Figura 30: Andamento del MAE per ogni modello(2 step ahead forecast).

3.2 Analisi delle vendite delle luci di natale.

Consideriamo il dataset delle vendite delle luci di natale(Figura 31).

IL dataset contiene gli attributi

- "Weekly_Sales" il numero di vendite settimanali di sigarette in pacchetto venduto.
- "Sales". L'ammontare delle vendite in dollari($).
- "Date": le date settimanali delle osservazioni.
- "Rain": un flag booleano che indica se durante la settimana c'è piovuto.
- "Snow": un flag booleano che indica se durante la settimana è nevicato.
- "Temp": le temperature medie settimanali in grado celsius.

- “IsHoliday”: flag booleano indice se la settimana considerata era una settimana di vacanza.
- “Promo”: flag booleano che indica se la settimana c’è stato una promozione.

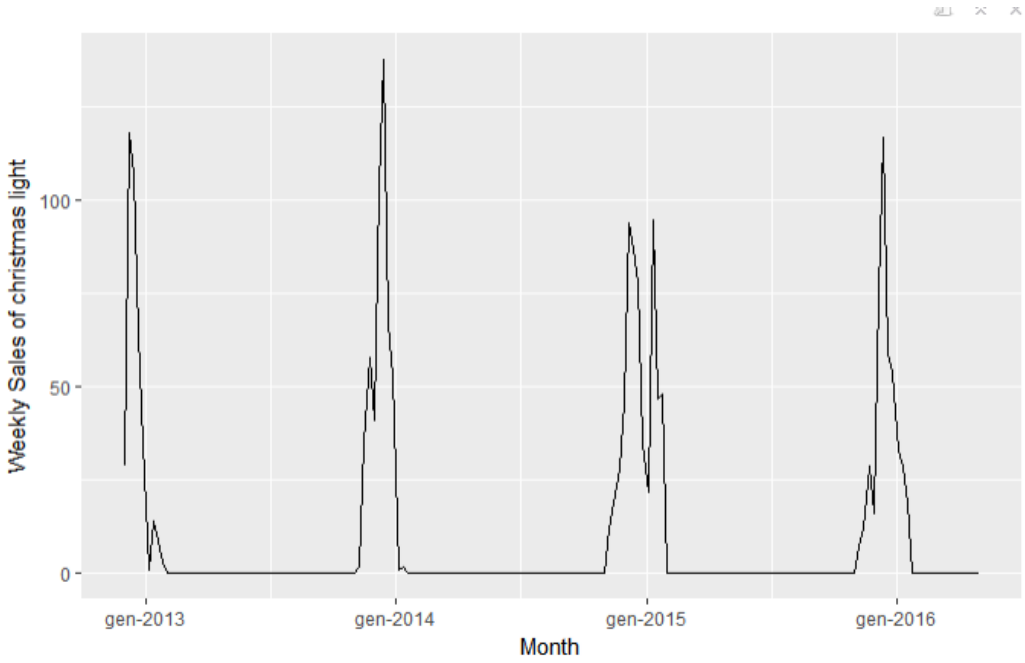

Figura 31: Vendite delle luci di natale.

L’andamento delle vendite è stagionale tra i mesi di dicembre e gennaio. Non sembra presentare un trend né particolari valori anomali. Le osservazioni nel dataset vanno dal 1-12-2012 fino al 30-04-2016, sono in totale 179.()Figura 32).

Date	IsHoliday	Promo	Rain	Snow	Temp	Sales	Weekly_Sales
2012-12-01	0	1	1	0	3	117.72	29
2012-12-08	0	1	1	0	8	547.28	118
2012-12-15	0	1	1	0	6	525.83	107
2012-12-22	0	1	1	0	7	344.59	64
2012-12-29	1	1	1	1	2	148.32	35
2013-01-05	1	1	1	0	1	3.99	1
2013-01-12	0	1	1	0	5	35.36	14
2013-01-19	0	1	1	0	4	24.66	9
2013-01-26	0	1	0	1	-3	9.72	3
2013-02-02	0	0	1	1	3	0.00	0
2013-02-09	0	0	1	1	0	0.00	0
2013-02-16	1	0	1	1	4	0.00	0
2013-02-23	1	0	1	1	0	0.00	0
2013-03-02	0	0	1	1	4	0.00	0
2013-03-09	0	0	1	1	3	0.00	0

Figura 32: Sample del dataset delle vendite delle luci di natale.

La decomposizione delle serie storica evidenzia la sua stagionalità.()

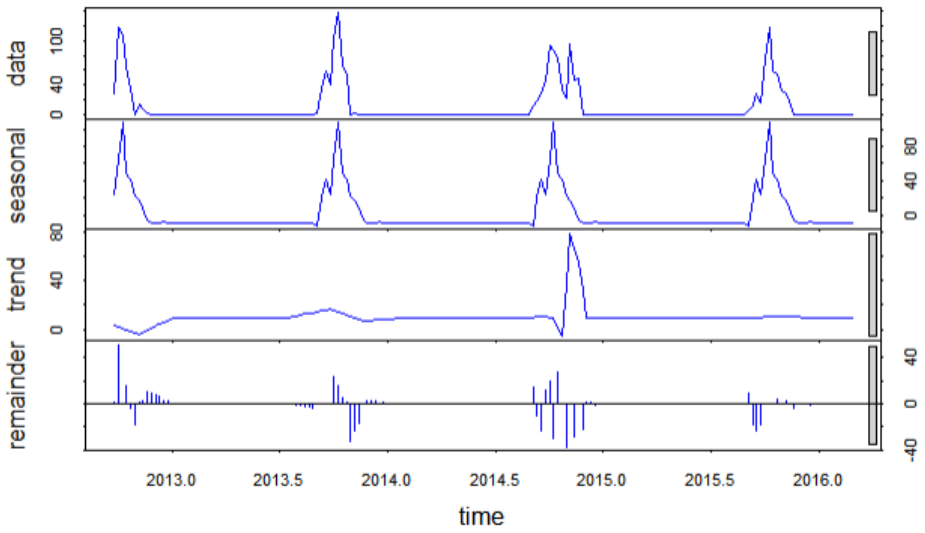

Figura 33: Decomposizione delle vendite delle luci di natale.

In questo esempio, vogliamo misurare l'effetto delle promozioni sulle vendite, quindi noi valuteremo i modelli Xgboost, Regressione lineare con errori Arima e NNAR.

Supponiamo che la misura dell'effetto delle promozioni è sempre positiva e quindi valori negativi sono considerati come zero(nessun effetto).

Dopo valutazione delle serie usando la cross validation con un 2 step ahead forecast , otteniamo i risultati nella .tabella.

	Regressione Lineare con errori Arima.	Xgboost	NNAR
MAE	13,796	2,12	2,62
RMSE	14,42	2,39	2,94

La cross validation è stata effettuata su training set di 162 osservazioni(90% del dataset) e 17 osservazioni(le ultime) come test set.

Xgboost è il migliore in questo caso. In seguito, usiamo Xgboost per misurare l'effetto delle promozioni. Per illustrare quanto detto, dividiamo il dataset in 127 osservazioni di training e 52 osservazioni di test set(le ultime) per avere una visione su un anno delle previsione e cosi catturare la stagione natalizia delle vendite e le promozioni.

Abbiamo allenato Xgboost con le features: "mese", "giorno del mese", "Promo", "IsHoliday", "Snow", "Rain" e "Temp" e abbiamo Xgboost ha ricavato la seguente matrice di importanza in Figura 34.

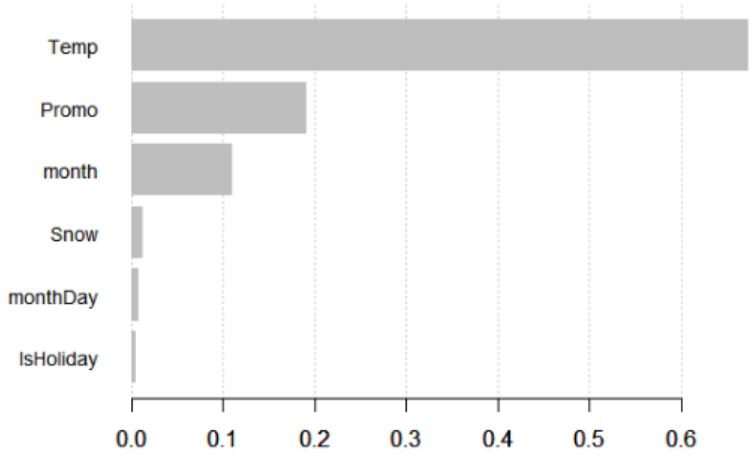

Figura 34: Matrice di importanza Xgboost per le vendite delle luci di natale

Xgboost ritiene le temperature e le promozioni come le più importanti.

La Figura 35 mostra le previsioni su 1 anno delle vendite delle luci di natale.

Xgboost funziona molto bene su questa serie storica, cattura la sua stagionalità e il suo andamento. In blue, la previsione di Xgboost e in Nero le osservazioni reali.

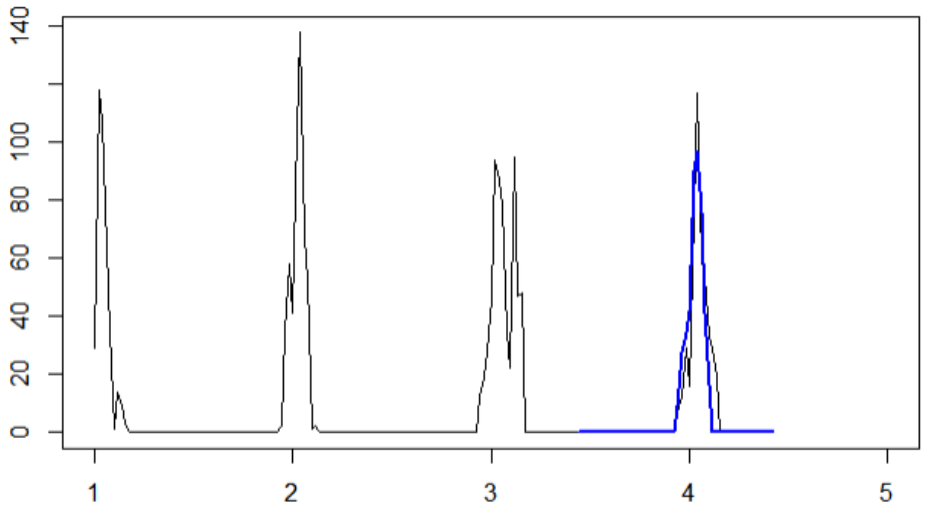

Figura 35: 52 step ahead forecasts con Xgboost delle vendite delle luci di natale.

Supponiamo ora che le promozioni nel test set sono erano 0 che non sono state fatte. Con questa ipotesi, le previsioni di Xgboost si possono vedere in Figura 36.

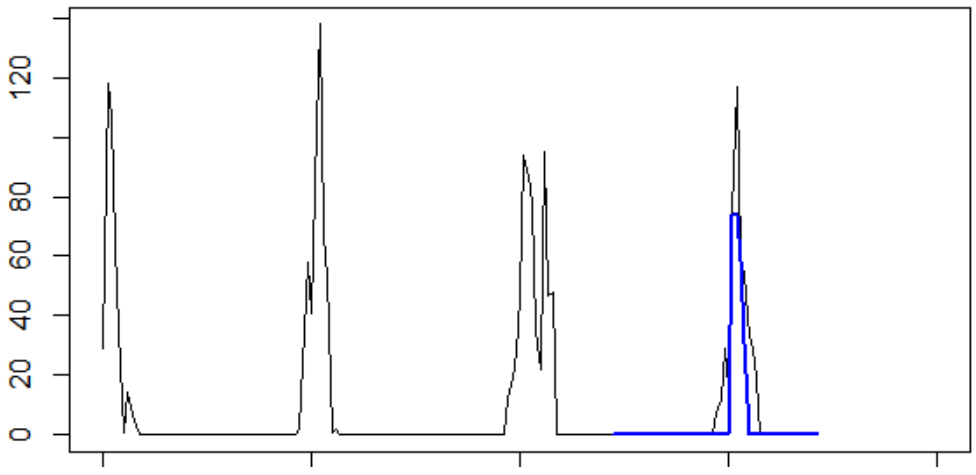

Figura 36: 52 step ahead forecast con Xgboost per le vendite delle luci di natale

Si vede che le vendite senza promozioni hanno una forma meno fine rispetto alle vendite con promozioni.

La misura dell'effetto delle promozioni è la differenza tra le vendite con promozioni e senza promozioni.

In Figura 37, vediamo la misura dell'effetto delle promozioni. Notiamo che le promozioni hanno portato fino a 40 vendite.

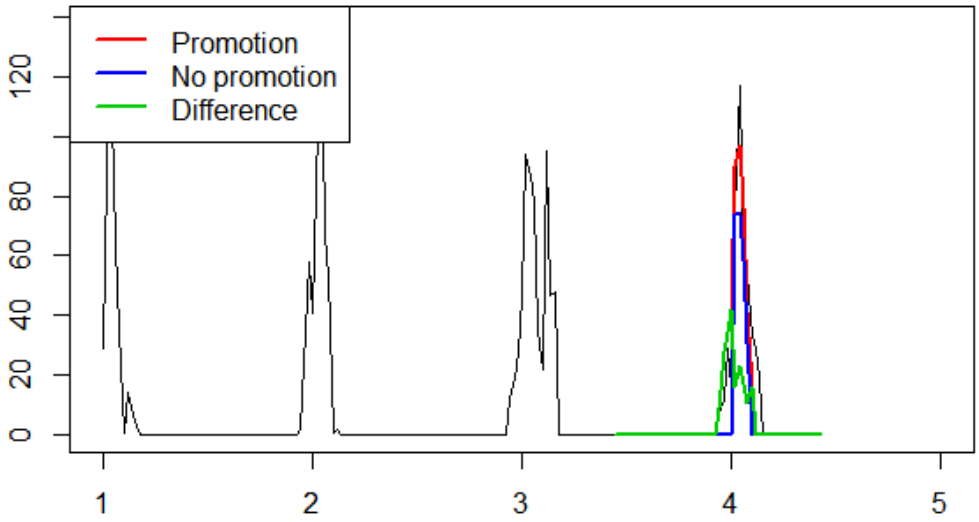

Figura 37: Effetto delle promozioni sulle vendite delle luci di natale misurato con Xgboost

Capitolo 4

Architettura e Tecnologie della soluzione software

Dai capitoli precedenti, siamo arrivati alla conclusione che grazie al machine learning e l'analisi predittiva è possibile dai dati rilevare informazioni utili per il supporto al marketing. In questo capitolo, presentiamo l'architettura e le tecnologie della soluzione software.

4.1 La soluzione software e le sue caratteristiche

La soluzione software scelta è una applicazione web fruibile in modalità SaaS (Software as a Services) tramite un browser.
Per usufruire del software l'utente deve caricare dati nel sistema. L'acquisizione e il caricamento dei dati richiedono assistenza dell'utente in quanto l'utente dovrà fornire informazioni necessarie a svolgere un compito di ETL per poter caricare i dati nel sistema.
Un ETL(Extract Trasform Load) è un'espressione che si riferisce al processo di estrazione, trasformazione e caricamento dei dati in un sistema di sintesi. I dati vengono estratti da sistemi sorgenti quali database transazionali (OLTP), comuni file di testo o altri e vengono trasformati(es join, selezione di alcuni campi) e infine vengono memorizzate nelle tabelle del sistema.
Il sistema integra dati provenienti da file CSV, XLS, XLSX, database, Web Api.
L'utente deve fornire almeno dati di vendite e campagne marketing come promozioni.
Il software si interfaccia con vari servizi di rete per acquisire informazioni quali il meteo, le festività e eventi particolari.

4.2 Architettura

La Figura 38 mostra l'architettura del software.

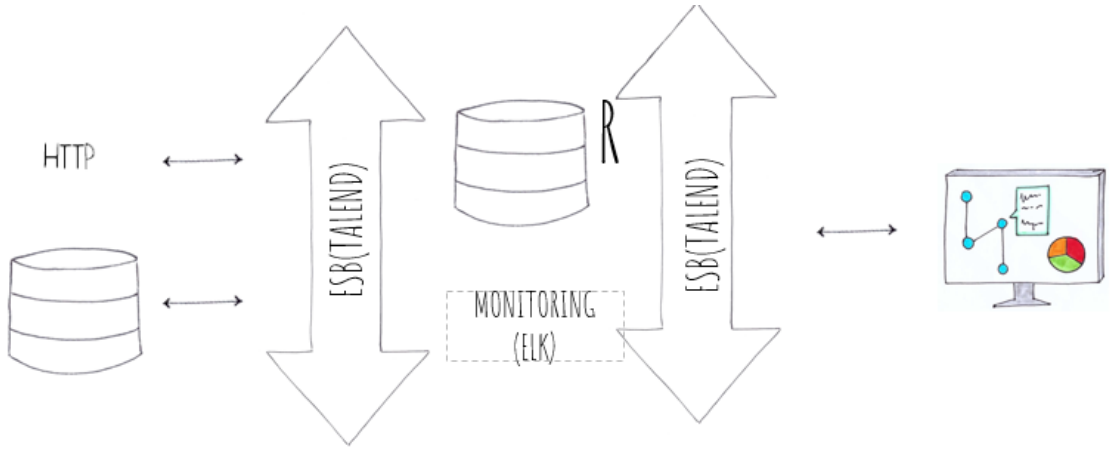

Figura 38: Architettura della soluzione software.

Questa architettura è composta da:

- Un ESB(Enterprise service bus): L' Enterprise service bus è un'infrastruttura software che fornisce servizi di supporto a service-oriented architecture. Si basa su sistemi disparati, interconnessi con tecnologie eterogenee, e fornisce in maniera consistente servizi di coordinamento, sicurezza, messaggistica, instradamento intelligente e trasformazioni, agendo come una dorsale attraverso la quale viaggiano servizi software e componenti applicativi.

- Un sistema di monitoring: Un sistema di monitoring è un sistema software utilizzato per monitorare le risorse e le prestazioni del sistema.

- Un web service: Un servizio web è un sistema software progettato per supportare l'interoperabilità tra diversi elaboratori su una medesima rete ovvero in un contesto distribuito. Questo servizio web espone i modelli di analisi dati come servizio o risorsa nel sistema.

- Una web application: Implementa l'interfaccia utente del sistema.

- Un DBMS: Un database management system per la gestione dei dati.

L'enterprise service bus consuma servizi di rete e integra sorgente dati distribuite in rete. Inoltre implementa servizi che si interfacciano con la web application e il servizio web di analisi dati. In più offre tool ETL(Extract Trasform Load) per il caricamento dei dati in ingresso nel sistema.
Il funzionamento dell'sistema può essere riassunto come segue: L'utente carica i dati nel sistema grazie a una collaborazione tra la web application e ESB. I dati vengono inseriti in un database. In seguito, su richiesta dell'utente, la web application sollecita l' ESB che s'interfaccia con il servizio web per gli analisi e ritorna i risultati a l'interfaccia utente.

Questa architettura è un esempio di architettura SOA. Una SOA(Service oriented architetture) indica un'architettura software adatta a supportare l'uso di servizi Web per garantire l'interoperabilità tra diversi sistemi così da consentire l'utilizzo delle singole applicazioni come componenti del processo di business e soddisfare le richieste degli utenti in modo integrato e trasparente. Infatti, tutte le componenti di questa architettura sono sistemi software(servizi) che comunicano e s'integrano tra di loro per fornire un o più servizi.
Le architetture SOA portano ad una serie di vantaggi quali:

- **Software as a servizio**: al contrario del software tradizionale, un Web Service può essere consegnato ed utilizzato come un canale di comunicazione accessibile, in modo obliquo, da qualsiasi piattaforma. I servizi di rete consentono l'incapsulamento cioè i componenti possono essere isolati in modo tale che solo lo strato relativo al servizio vero e proprio sia esposto all'esterno. Ciò comporta due vantaggi fondamentali: indipendenza dall'implementazione e sicurezza del sistema interno.
- **Interoperabilità**: la logica applicativa incapsulata all'interno dei servii di rete è completamente decentralizzata ed accessibile attraverso Internet da piattaforme, dispositivi e linguaggi di programmazione differenti.
- **Semplicità di sviluppo e di rilascio**: sviluppare un insieme di servizi di rete, intorno ad uno strato di software esistente, è un'operazione semplice che non dovrebbe richiedere cambiamenti nel codice originale dell'applicazione. Lo

sviluppo incrementale dei WS avviene in modo semplice e naturale. Inoltre, rilasciare un WS significa solo esporlo al Web.

- **Standard**: concetti fondamentali che stanno dietro ai Web Service sono regolati da specifiche universalmente riconosciute e da standard approvati dalle più grandi ed importanti società d'Information Technology al mondo.

Inoltre, permette a più figure professionali di lavorare in modo svincolato sullo stesso software. Una SOA è progettata per il collegamento a richiesta di risorse computazionali come applicazioni e dati.

Le tecnologie software alla base del software sono:

- Talend Open Studio for ESB: implementa le funzionalità dell'ESB.
  Talend Open studio è gratuito, stabile e s'integra con numerose sorgenti dati.

- Lo Stack ELK(Elasticsearch Logstash Kibana): implementa il sistema di monitoraggio del software. ELK è molto potente e è nato per analizzare file di log e quindi supporta(Logstash) tanti plugin per estrarre velocemente informazioni in file di log.

- PostgreSQl: è il database management system del software per la gestione dei dati.

Il Web Service per l'analisi dati e l'interfaccia utente sono pezzi software in sviluppo in Hopenly.
Nelle sezioni a seguire, presentiamo Talend open studio for ESB e diamo una breve descrizione dello Stack ELK.

4.3 Talend Open Studio for ESB

Talend ESB è un'applicazione software che permette di integrare sia applicazione che dati. le soluzioni open source di Talend ESB offrono funzionalità di routing dei messaggi

e di mediazione dinamici basati su regole personalizzate e modelli di integrazione aziendale. Con le soluzioni Talend ESB, si può rapidamente ed efficacemente costruire servizi web in una gamma di modalità tra cui SOAP su HTTP, XML su JMS, servizi RESTful, e altro ancora. Il sistema può essere esteso per incorporare in modo incrementale funzioni avanzate come la crittografia dei dati, la gestione delle identità, e la scoperta di servizio remoto.

La Figura 39 mostra le soluzioni Talend ESB.

	Open Studio for ESB	ESB	Data Services Platform
Licenza	Open Source Apache License v2	Abbonamento	Abbonamento
Modellazione e sviluppo	Più di 900 componenti e connettori	+ repository e gestione delle identità	+ repository e gestione delle identità
Ampie funzionalità di integrazione	Creazione di servizi, integrazione dati	+ supporto per EIP	+ ETL / ELT, mappatura visuale
Maggiore fiducia grazie alla qualità dei dati	—	—	Profiling, matching e standardizzazione
Deployment più rapido	ESB e servizi Web	+ repository condiviso	+ gestione repository, API
Collaborazione migliore e gestione ottimizzata	Monitoraggio dei servizi, plug-in	+ Talend Administration Center	+ esecuzione pianificata, alta disponibilità
Utilizzo più rapido dei dati	—	+ operazionalizz azione dei dati preparati	+ operazionalizza zione dei dati preparati
Support	TalendForge Community, Help Center access	+ Guaranteed Response Times, Web & Email Support	+ Phone support, faster response, optional 24/7
Supporto	Community TalendForge, accesso all'Help Center	+ tempi di risposta garantiti, supporto web ed e-mail	+ supporto telefonico, risposta rapida, 24/7 opzionale
Indennizzo / garanzia	—	✓	✓

Figura 39: Soluzioni Talend ESB

Talend Open Studio for ESB è la versione è un'interfaccia grafica utente (GUI) di supporto ai prodotti open source ESB di Talend, utilizzata dagli sviluppatori per creare in modo rapido processi di integrazione. Grazie alla funzione drag and drop, gli utenti possono spostare componenti di integrazione e connettori in un ambiente di lavoro grafico, creare connessioni e rapporti tra di loro, e determinarne caratteristiche specifiche. Talend Open studio for ESB comprende

IL Talend Open studio for ESB permette anche la creazione di servizi di integrazione dati Talend facilita operazioni di ETL e s'integra con numerosi tipi di sorgente dati.

Alcuni termini ricorrenti che è utile conoscere quando si lavora con questo software sono:

- Repository: è una locazione di storage che Talend Open Studio utilizza per raccogliere i dati relativi a tutti gli oggetti tecnici che si utilizzano nel design di un Job o di un Business Model.

- Progetto: è una collezione strutturata di oggetti tecnici e dei metadati associati ad essi. I Jobs sono organizzati in progetti

- Workspace: è una directory dove vengono salvate tutte le cartelle del progetto.

- Job: è una rappresentazione grafica di uno o più componenti connessi assieme, che ci permette di creare ed eseguire processi di gestione dei flussi di dati. Traduce i bisogni del business in codice e programmi. Un job gestisce tutte le sorgenti e le destinazioni di cui si ha bisogno nel processo di data integration

- Componente: è un connettore preconfigurato usato per realizzare una specifica operazione di data integration. Un componente aiuta a minimizzare il quantitativo di codice necessario a lavorare sui dati proveniente da sorgenti diverse.

Un Esempio di Job Talend(Figura 40).

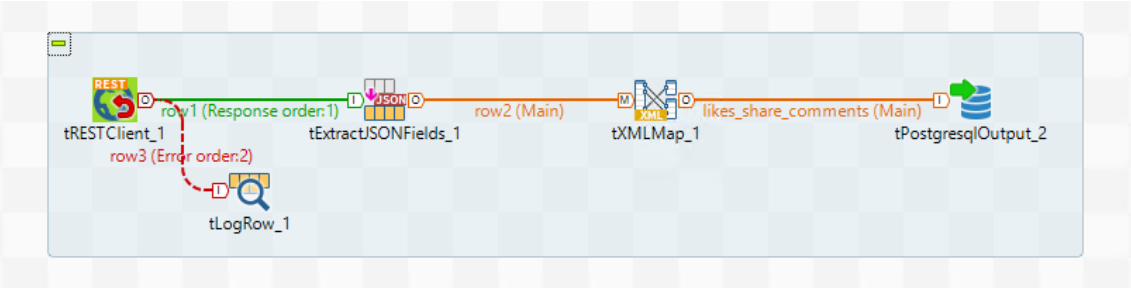

Figura 40: Esempio di Job Talend

Il job in Figura 40 integra una REST API e memorizza i dati in database PostgresQl.
Il job è composto da 4 componenti:

- Il componente tRESTClient_1 è il componente ESB che serve a consumare una web API REST.
- Il componente tExtractJSONFields_1 permette di estrarre informazioni in una struttura dati Json.

- Il componente tXMLMap_1 permette di mappare e trasformare dati input. In questo caso permette trasforma i campi estratti dal Json ritornato dalla web API e lo mappa nei campi della tabella di sintesi memorizzata nella base di dati PostgreSQl.

- Il componente tPostgresqlOutput_2: permette di inserire, aggiornare dati in una tabella contenuta in una base di dati PostgreSQl.

La soluzione Talend Open studio For ESB offre il Talend Runtime Container. Il Runtime Container è basato sull'application server Karaf, permette di esporre job Talend in produzione. Il Runtime Container può essere installato come servizio nei sistemi operativi Windows e Unix/Linux.

4.4 Lo Stack ELK

ELK sta per Elasticsearch Logstash e Kibana, sono tecnologie per la creazione di visualizzazioni da dati grezzi. ELK è specializzato per l'analisi dei file di log.

Elasticsearch è un motore di ricerca avanzato e super veloce. Con Elasticsearch, è possibile cercare e filtrare tutti i tipi di dati tramite una semplice API. L'API è RESTful, quindi non si può utilizzare solo per analizzare i dati, ma anche utilizzarlo nella produzione per le applicazioni web-based.

Logstash è uno strumento destinato per l'organizzazione e la ricerca di file di log. Ma può essere utilizzato anche per la pulizia e lo streaming di dati di grandi dimensioni provenienti da tutti i tipi di fonti in un database. Logstash ha anche un adattatore per Elasticsearch, per cui questi due giocano molto bene insieme. Lavorare con Logstash richiede un elemento di ingresso e di uscita, e opzionalmente un filtro. I plugin di ingresso consumano i dati da una fonte, i plugin di filtro modificano i dati e i plugin di uscita scrivono i dati ad una destinazione.(Figura 41).

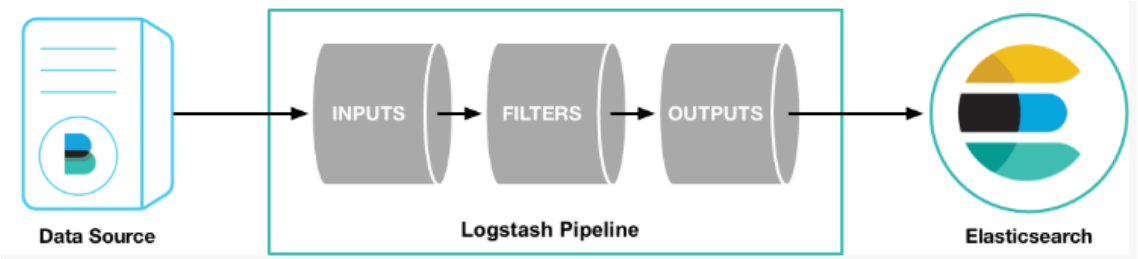

Figura 41: Pipeline Logstash.

Kibana è un'interfaccia visiva per Elasticsearch, funziona nel browser. E 'abbastanza bravo a visualizzare i dati memorizzati in Elasticsearch e non richiede competenze di programmazione in quanto le visualizzazioni sono configurate completamente attraverso la sua interfaccia.

Figura 42: Una vista di Kibana

Capitolo 5

Conclusioni

Nell'ambito di questo relazione di tesi, si è illustrato l'attività svolta presso l'azienda "Hopenly SRL"[] inerente alla realizzazione di software di supporto al marketing. In particolare, si è partito da un problema di machine learning per arrivare alla definizione di un software che sfrutta modelli di machine learning per fare analisi di campagne marketing per supportare il decision making. Il software ha lo scopo principale di fornire una misura del ROI(effetto o impatto) delle campagne marketing sulle attività economiche.

Inizialmente, si è studiato modelli di machine learning capaci di dare una previsione delle l'attività economica, con previsione a breve , medio e lungo termine. Cosi, ci siamo focalizzati sui modelli Arima, modelli a livellamento esponenziale, modelli Xgboost e modelli di reti neurali auto regressive. Con questi modelli abbiamo adottato un approccio semplice il quale era di allenare e scegliere il modello che meglio modella la realtà dell'attività economica e quindi sfruttare quel modello per dare misure dell'impatto marketing. Con questo approccio, abbiamo potuto misurare l'impatto delle promozioni sulla vendita di luci di natale. Abbiamo ottenuto buoni risultati, ma c'è ancora tanto da fare , soprattutto tanti modelli da sperimentare oltre a una ottimizzazione degli algoritmi considerati. Si sottolinea che questi lavori di analisi predittive stanno proseguendo presso "Hopenly SRL".

Successivamente, abbiamo dato una soluzione architetturale del software del software da realizzare. I 'architettura del software si basa su una SOA ed è costituita da componenti logicamente separate. Quest'architettura ha tanti vantaggi tra cui la sua modularità, l'interoperabilità e la sua semplicità di sviluppo. Uno dei problemi che doveva risolvere questo software era l'integrazione con sorgente dati degli utenti. Il suddetto problema si risolve grazie alla potenza di Talend ESB che offre numerose soluzioni di integrazione con database, file, servizi web, REST API etc.

Il software è in sviluppo presso Hopenly SRL.

Come sviluppo futuri, il software potrebbe integrare anche soluzioni di sentiment analytics e social graph mining per poter dare una risposta al target marketing e market segmentation.

Bibliografia

[1] Hopenly SrL http://www.hopenly.com

[2] Rob J Hyndman e George Athanasopoulos, Forecasting: principles and practice, https://www.otexts.org/fpp.

[3] Mean Absolute Error, https://en.wikipedia.org/wiki/Mean_absolute_error.

[4] Mean Squared Error, https://en.wikipedia.org/wiki/Mean_squared_error.

[5] ARIMA, https://en.wikipedia.org/wiki/Autoregressive_integrated_moving_average.

[6] Exponential Smoothing, https://en.wikipedia.org/wiki/Exponential_smoothing.

[7] Xgboost, https://en.wikipedia.org/wiki/Xgboost.

[8] Introduction to Boosted Trees, http://xgboost.readthedocs.io/en/latest/model.html.

[9] Bias–variance tradeoff,
https://en.wikipedia.org/wiki/Bias%E2%80%93variance_tradeoff.

[10] Brown 1959, 1957 e Holt Winters 1960 sono fondamentali lavori pionieristici.

[11] http://www.riccardoperini.com/marketing-definizione.php

[12] https://it.wikipedia.org/wiki/Test_di_verifica_delle_informazioni_di_Akaike

[13] https://www.elastic.co/

[14] https://help.talend.com/display/HOME/Talend+Open+Studio+for+ESB

[15] https://it.wikipedia.org/wiki/Service-oriented_architecture

Ringraziamenti

Vorrei ringraziare principalmente la professoressa Sonia Bergamaschi per i preziosi insegnamenti, per la sua gentilezza, per la sua guida durante la stesura di questa tesi e per la grande disponibilità dimostrata.

Vorrei ringraziare l'azienda Hopenly S.r.l per l'opportunità e l'esperienza che ho acquisito durante il mio tirocinio. In particolare, vorrei ringraziare il CEO Barbara Vecchi, il CDO Roberto Grassi, Alessandra Coppetta, Marco Michelangeli, Pierpaolo Basile.

Desidero ringraziare con affetto la mia famiglia "Kuicheu": i miei genitori ,i miei fratelli e sorelle per il sostegno morale e soprattutto economico, che mi hanno dato lungo questi anni di studi.

Un grazie speciale a Dodiane Carole.

Un grazie particolare a Gerard le Maire, Rouxel Kamgaing, Valery, "les gars" per le feste, il casino e le serate da pazzi che hanno animato la mia vita universitaria.

Infine ringrazio tutti coloro che mi hanno sostenuto in maniera diretta e indiretta durante questa bellissima esperienza che mi hanno consentito di maturare sotto ogni tipo di aspetto.

Grazie di cuore a tutti...