Stock price prediction and Portfolio generation using financial news

Jivjot Singh, Bikramdeep Singh and Manpreet Singh

Department of Computing Science, Simon Fraser University, Burnaby, BC

***Abstract*— This is final project report for describing the experiments done on Stock market data. Project resolves around the analysis of stock market predicting the change in the movement of stock prices in using financial news. News can have huge impact on the prices of the stock in both positive and negative way. Keeping this in mind ,we analyse the news forecast data to predict the stock prices. Last but not the least we try to generate a portfolio for the customer for intra-day trading scenario, to get maximum profit in the market using the predicted stocks.**

I. Motivation and Background

Capital Markets is a very huge and dynamic industry which governs the economy of the stock market. Total Stock Market is close to 55 Trillion USD and 2 billion stocks are traded over NASDAQ daily which is humongous amount of data. The amount of transactions and data under consideration is huge and scope of data science application is vast in applying predictive analytics and earning money.
Due to large scale of money and risk involved in stock market, it is subjected to many studies, experiments and hypothesis. Over the last 50 years many papers and research have been applied to analyze the trends in the stock market. But two popular theories have gained acceptance by majority of economists and researchers. One is Random walk theory [1] which states that movement in prices of various companies is completely random and no statistical tools or predictions can estimate the future value of a stock. It is based on the assumption that stock market are highly efficient that is; any new information (including insider information) which can influence stock prices is already reflected in its price. Change is instantaneous and there is no way to exploit the new information to generate profits.

Another theory is Capital Asset Pricing Model [2] which states that daily return of a stock is determined by the formula.

$$r_t^s = \beta^s r_t^m + a_t^s + \Delta\epsilon_t^s$$

$$where, r_t^s = daily\ return\ of\ stock(s)\ at\ a\ day(t)$$

$$r_t^m = daily\ return\ of\ market(m)\ on\ a\ day(t),$$

$$\beta^s = degree\ of\ correlation\ with\ market,$$

$$a_t^s = independent\ movement\ of\ a\ stock(s)\ on\ a\ day(t),$$

$$\epsilon_t^s = random\ noise\ in\ prediction\ for\ a\ stock(s)\ on\ day(t)$$

This theory extrapolates that daily price movement of any stock is derived as a function of the movement of the market and the independent movement of the company. This is not an accurate model and is subject to some random error. Many researchers have tried to exploit this model to predict ts. As ts is independent from other markets and only depends on a single company, many papers and research have tried to use this model in their prediction such as Twitter [3] sentiments of the people to predict daily movements of a stock.
However, the degree of prediction and accuracy can never be accurate and degree of random noise varies from stock to stock, and could be very high in some cases where patterns cannot be determined. Moreover, stocks market are designed to be efficient and any model which performs very well eventually market will adapt to it and become more efficient. Accuracy rate is limited but even if a small percentage of the pattern is detected it can be used to create a better trading algorithm.

II. Problem Statement

This report answers different questions on which stocks to buy depending on the movement of stock price in the market, considering historical news and latest trends in the market. Depending upon the Portfolio creation for a user to predict, decide on which stocks to buy and sell to earn maximum profit. Also, using the historical data, predict how much accurate is the model and the performance of the model compared to SP index. Last but not the least goal is to perform Sentiment analysis on stock price prediction, using financial news sources and determine the impact.

We want to discover if textual information from news can be used to predict future stock trends. If yes then can this information can be exploited to make money in intra-day trading algorithms. Challenge in this problem, is that the input information is not structured. Also not all articles are relevant to a price movement. Working on textual information is hard due to many different meanings of different words and financial lingo can have different meanings for same words. Also we have collected daily news articles for each company for last 5 years, this is a lot of data and this could lead to scalability issues.

Abbreviations and Acronym **D3 - Data driven documents**

III. Data Processing Pipeline

Data Cleaning → Feature Extraction → Model Training → Portfolio generation

A. Collection

We have created a shell script to download the historical stock data from the sources like Google finance api. Aggregate news from multiple sources like New York Times, Wall street journal, google finance news etc. Stock data of top 100 companies publicly traded on NASDAQ. Data was collected from 2010 to 2016. We did not use a preprocessed collected data set, because it did not match our requirements. We collected our own datasets using the google finance api and yahoo finance api. We first created a list of top 100 traded companies on NASDAQ. Then we wrote a shell script to download the financial news from google finance. Google finance api provides us with news aggregated over different news media, it takes company name, date as input and returns the corresponding news articles, in rss/xml format. As we needed news for each single day for last five years, we needed to query for each day, for each query using the shell script. Google finance has a maximum request limit, so we used a halt and maintained the current progress so that we could continue the download after 1 hour halt. Data downloaded from yahoo finance was historical OHLC data (Open High Low Close) in csv format. Again we used shell script to download the data.

B. Cleaning

Historical news needs to extracted using web api using rss feed, Data available here is in XML format, we would need to extract the news from the html tags, we used shell script to download the html contents and use it to clean and strip html information into a structured format. Data downloaded from Yahoo finance api is in csv format and was directly used. But on the other hand news data from google finance needed to be preprocessed before feature extraction. We parsed data from xml format, as data is aggregated from various news sources it contains html tags and css information, which we pruned to get the cleaned data.
Training / Test Data split: We partition our dataset into training and test dataset. We labelled any news before 2015 as Train set and after 2015(including) as Test Set. We aim to plot the price movement and their predictions over a period of time in order to facilitate this continuous time series data was used as Test set.

IV. Feature Extractions

A. Dimensionality reduction

Only necessary dimension affiliated with any stock price were taken into consideration like Date,Open,High,Low,Close,Volume,Adj Close etc. This also contains redundant and extra information. Dimensionality reduction also helped to reduce the volume of data and pull on required information. For the scope of the this project we filtered the and used fields such as date, Adj Close, Price Movement(Adj Closed of next day - Adj Closed of Previous day).

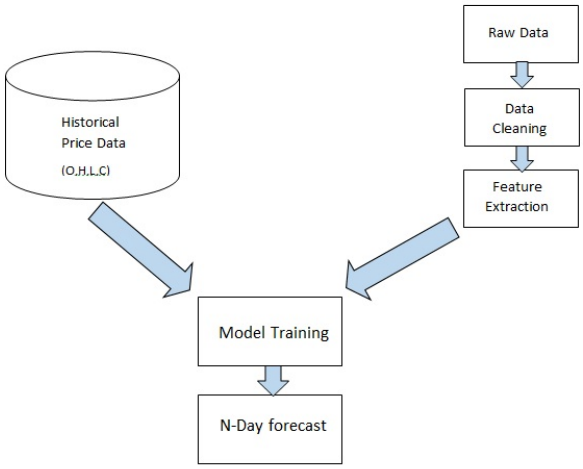

Fig. 1. Stock prediction Pipeline

B. Stop Words

We did sentiment analysis on news sources, therefore it was necessary to remove stop words. This was accomplished using pythons nltk library.We wanted to convert the textual information to numerical feature vector. We used word2vec for this process, We trained a word2vec model using the training data described above. For each day and for each company we have several news articles, we aim to represent each article as word2vec vector. We achieved this by first removing the stopwords from each article, then each article feature vector is calculated as average of word2vec vectors of the remaining words in the article. This is only the initial phase of the feature extraction, this method varies according to the underlying model.

- Window based Feature aggregation: As described above we now have word2vec feature of each article. We use this information to get word2vec representation of single day of each company. This is achieved by the averaging the feature vector of that particular day. Once we have feature vector of single day, we use a day window(30 days) to form an aggregated feature vector which contains the information of past 30 days.
- Window based Feature aggregation with OHLC: OHLC(Open High Low Close) data is downloaded from yahoo finance is joined with the widow aggregation described above. Thus now for each day(each company) we have past 30 days new features and past 30 days OHLC.

C. Model Creation

We use daily percentage change as our input to portfolio optimizer. So our target variable is daily percentage price change of a stock. In order to achieve better results we tried different methodologies and features vectors as described below:

- *Clustering*: First we analyzed the training data and divided the target variable into 6 classes. We decided

upon the classes by observing the histogram of the target variable. We tried many cluster sizes, in order to find a majority cluster configuration. A Majority cluster configuration is where major chunk of each member belongs to a particular class. But unfortunately no such configuration was observed.

- *Word2vec Regression*: Using 30 day window described before, these features were extracted and used for regression. In the image we can observed the regression trend for few companies follows that of original values.
- *Word2vec + OHLC Regression*: Here we add the 30 day OHLC to the data described above. Here we observe that RMSE error decrease and graph displays an average trend of a stock.

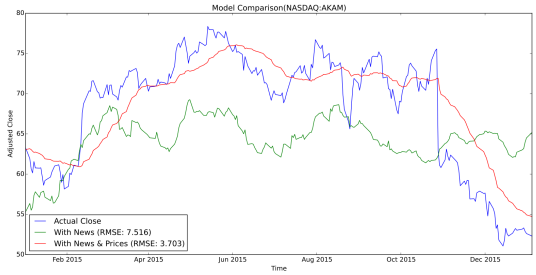

Fig. 2. Prediction Values for different models (NASDAQ:AKAM)

D. Portfolio Creation

As shown in the figure, we use the model results to produce a portfolio. The portfolio created is targeted to an INTRADAY trader. INTRADAY trading is where any stock you buy on a day needs to be sold on the same day, so you make profit/loss on a stock depending upon its daily movement. We assign an utility score to each stock daily which depends upon its Model predicted value, standard deviation and a user defined risk factor.

If R is more which means we penalize higher std deviation leading to a safe portfolio. Each day we choose the top 5 utility score stocks

E. Portfolio Simulation

Using the above forecasting and portfolio we simulate trading on algorithm on test dataset, using 3 different Risk factors R = 1,3,10. Also we inorder to verify the results we compare the portfolio performance with respect to SNP500. SNP500 is an index of top 500 companies on NASDAQ. We also added an random walk example where we used random number in the range of -1 to 1 as expected value and observed their performance. We also added another simulation where we used previous day returns as input and observed its performance.

F. Integration

Data downloaded from various resources was filtered and saved in csv files. Spark was used to apply word2vec to transform the news data as vector representation for sentiment analysis and results were stored in pickle files.

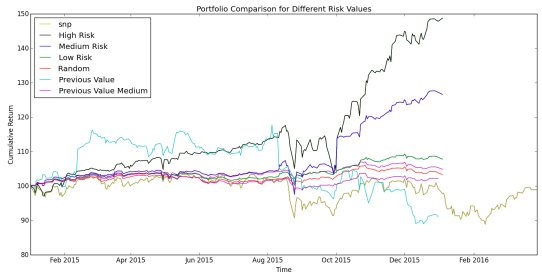

Fig. 3. Final Results for Portfolio Simulation

V. Methodology

There are various tools and technologies used in this project. To name a few Python, Spark, Hadoop, MySQL, AngularJS and D3.js. Python and spark for machine learning and regression analysis on stock prices. Linear regression was used to predict the stock prices which was expected to be inaccurate, so we tried to perform clustering on direction/movement of the prices. K-means was used to perform the clustering and aim was to form clusters which consisted of scales of price movement defined as bins in histogram. Finally, we will compare the predicted accuracy with and without taking the sentiment factor.

We used shell scripting to download the data, as it was more convenient and easy to program. We chose spark for data processing and model creation, because data set was huge and we were not able to process in local environment. Due to massiveness and scalability issues spark was a natural choice. For portfolio market simulation we used python and pyplot to visualize the portfolio performance. We also delivered a web/mobile UI for a user to visualize its portfolio performance for the past one year. Here we also displayed portfolios for each day. It also enabled user to visualize the news based trends vs actual price movements.

We learned that news information can be used to predict future trends in stock prices. RMSE in this project is on an higher scale compared to other machine learning applications, ,but still partial accuracy can be exploited to create good portfolios. Different companies have different relationship with stock prices and news. Some move independent of news movements, whereas some are highly dependent on news movements.

Summary: Word2vec was used to extract features from the historical news. 30 day word2vec feature of daily news and OHLC data were combined to form an aggregated feature vector. This was used for regression and patterns were used to observe if trends match the actual price movement in test data. Future daily returns from the model was used as input to portfolio creator for intraday trading. 3 different portfolio were created for users, for 3 levels of Risk. For better comparison and understanding the portfolio simulation was compared with SNP500 and random walk models.

VI. Observations

- Different models were trained both with and without the news and it was observed that the model with data of prices of the stock was more accurate.
- Results were shown on mobile and desktop application named smartStock which is needs user authorization and creates a user-profile. Different pages were generated for stock prediction and portfolio generation.

VII. Portfolio Generation

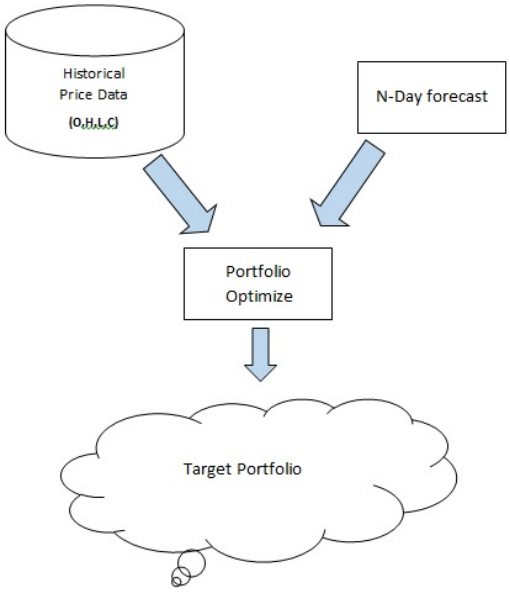

Fig. 4. Portfolio Pipeline

Generation of stock portfolio from the prediction made for intraday trading. Intraday trader refers to a stock trader who opens and closes a position in a security in the same trading day. This can be buying and selling to capitalize on a potential rise in a security's value or shorting and covering the short to capitalize on a potential drop in value.This also is requires minimum brokerage. In this project we designed the portfolio from the prediction of the stocks from the linear regression model.

Various parameters like risk index(how much risk a customer is willing to take - (high, medium and low) and standard deviation are taken into consideration. Standard deviation was calculated from closing price change from each day. To evaluate which stock would be better in terms of financial gains Utility score was calculated for each date.

$$U_s^t = EV_s^t - R\sigma_s^2;$$

$$where, U_s^t = utility\ score\ on\ day\ t\ for\ stock$$

$$EV_s^t = daily\ return\ of\ stock\ s\ on\ day\ t$$

$$R = risk\ factor$$

$$\sigma_s = standard\ deviation\ daily\ returns\ of\ stock\ s$$

This utility score is calculated for each date of the month and those

VIII. Data Product

The final data product is a web an android application named *"SmartStocks"* where users can visualize the predicted trends in the stock market by providing company name as an input. In addition to that users are also able to view portfolio and get best recommendations for investment. The app is having two main functionalities:

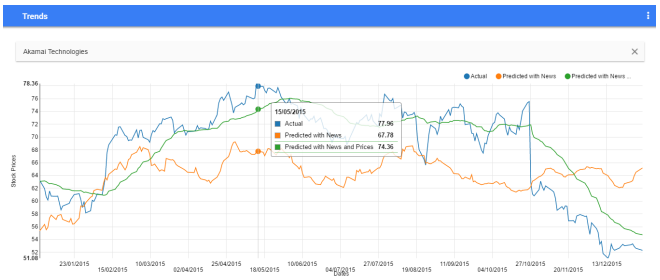

Fig. 5. Trends Screen in App

1) Trends: This page consists of an auto-suggest input section, where users can type any company (from the list of Top 100 publicly traded companies on NASDAQ) and the application will show the actual vs predicted stock prices for that particular company. The predicted prices are shown for different models i.e first with only news and second with news are historic stock price data. Users are also given the functionality of turning off various models and seeing prices for a particular day by hovering mouse on top of it.

Fig. 6. Portfolio Screen in App

2) Portfolio: This page contains stock recommendations for users based on three risk categories i.e. High, Low and Medium. The portfolio operates on intra-day trading model i.e. all the transactions must be completed on that day itself. When the user opens the portfolio section, three plots will be shown where x-axis correspond to dates and y-axis correspond to High, Medium and Low risk factors. Whenever mouse is hovered on a particular day, tickers for best investment options are shown to the user along with the expected return value.

A. Architecture

Fig. 7 represents the web app architecture and consists of following components:

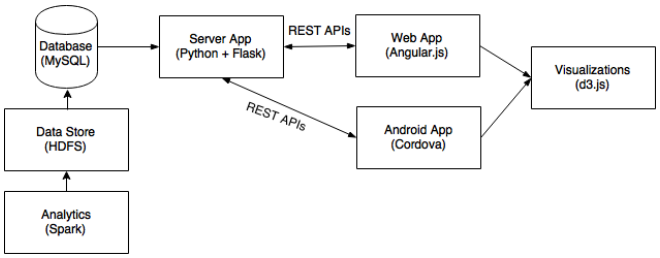

Fig. 7. Web App Architecture

- The output data from Spark jobs are dumped on HDFS, which is then fed into a MySQL database with the help of a python script.
- The MySQL database which now contains the predicted values from the trained model is exposed as a service using REST APIs.
- The data will now be consumed by Web and mobile clients via REST calls based on the user input.
- Finally, the data is visualized and rendered in the form of interactive graphs using the d3.js library.

IX. Lessons Learnt

- There is a relationship between financial news and stock prices Financial news have the direct impact over the stock prices in the market. The relation was attempted with percentage change in the close price of the stock.
- Some stock prices are more sensitive to news than others Some of the stocks are more sensitive to the news which displays a higher correlation of the stock price with the news whereas some of the stocks depicted mild correlation.
- Better results are achieved by combining historical stock prices with news data When the model is combined with the stock prices along with the trend of the news the model exhibits better accuracy with improved RMSE as compared to the model with only news data.

X. Summary

Project pipeline consisted of various stages of data science starting from collecting data to visualizing the results. Data was collected from different sources and filtered in multiple stages. Required features were extracted from data necessary for evaluation. Various machine learning algorithms were applied to do predictions on the data. The end results were visualized using a mobile and desktop application.

XI. Future Enhancements

Following are some features that we would like to add in the project in near future:

- In the portfolio recommendation feature we would like to add cumulative averages of the resulting recommended stocks to provide more accurate recommendations.
- We would also like to add trading cost while simulating the portfolio and see the resulting affect.
- We would also like to try predicting stock prices using different algorithms such as Bayesian & Random Forest and perform ensembling techniques to improve accuracy.

References are important to the reader; therefore, each citation must be complete and correct. If at all possible, references should be commonly available publications.

References

[1] Random walks in stock Market Prices: $http://bit.ly/22mIJhU$
[2] Capital Asset pricing model: $http://www-personal.umich.edu/\ kathrynd/JEP.Perold.pdf$
[3] Twitter mood predicts the stock market: $http://arxiv.org/pdf/1010.3003.pdf$
[4] Brentani,Butterworth-Heinemann, Portfolio Management in Practice (Book style with paper title and editor),Published on May 2014.
[5] Spark MLib: $https://spark.apache.org/docs/1.1.0/mllib-guide.html$
[6] Spark: Cluster Computing with Working Sets $https://www.cs.berkeley.edu/\ matei/papers/2010/hotcloud_spark.pdf$
[7] Coursera: Computational Investing (course on working of stock market)
[8] Spark clustering:$http://spark.apache.org/docs/latest/mllib-clustering.html$
[9] D3 visualization: $https://github.com/mbostock/d3/wiki/Gallery$