A Stock Market Prediction System based on Learning from Historical Prices and Customer Reviews

Supervised By

Dr. Reda El-Basiony

Team Members

Abdallah Ibrahim El-Badry

Abdelrahman Mohamed Emam

Aya Mohsen El-Kilany

Habiba Khaled Ibrahim

Ibrahim Mohamed Yousef

Mona Mardy El-Araby

Reem Emad Abdelaziz

Outlines

01 Problem & Solution

02 System Components And Architecture

03 Front End Development

04 Back End Development

05 Machine Learning For Time Series Forecasting

06 Conclusions & Future Work

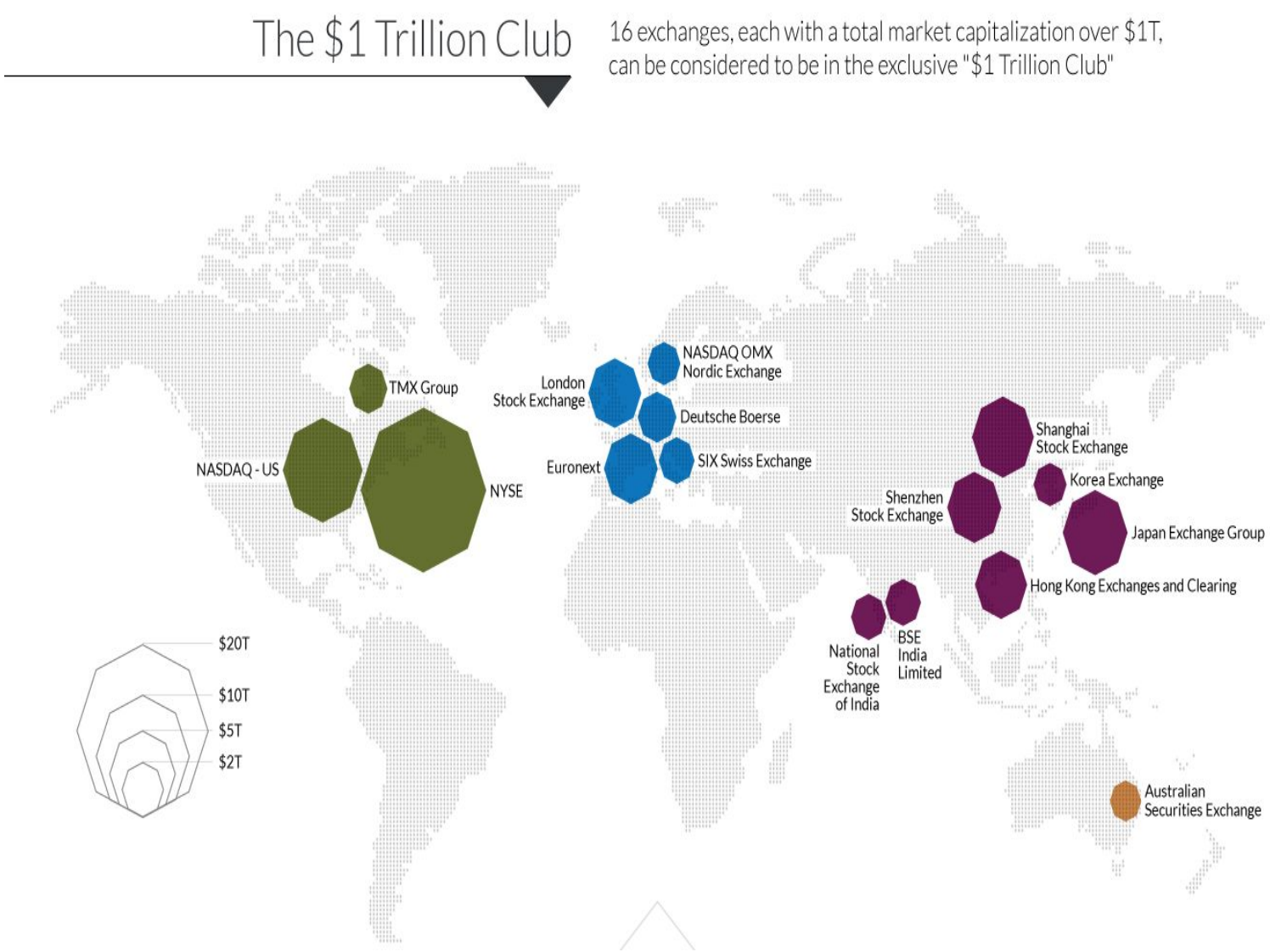
Problem Statement

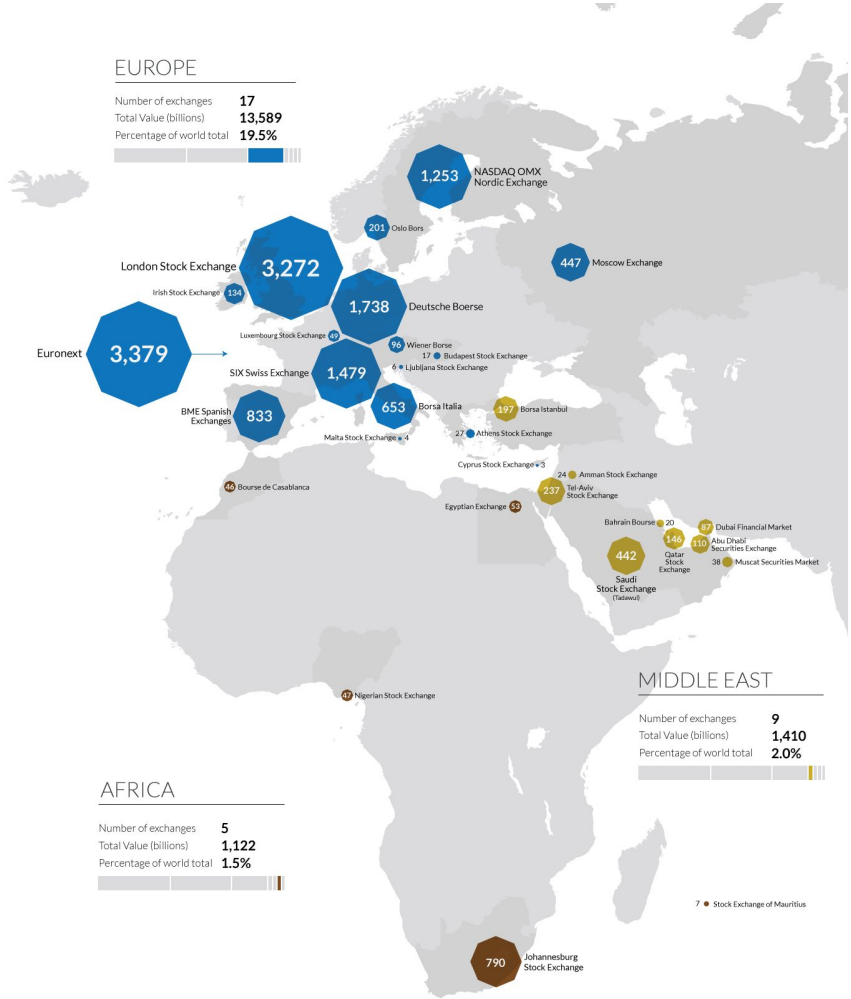
- **How Stocks Begin!?**
- **Modern stock markets**
- **Future of stock market**

- Prediction of stocks is generally believed to be a very difficult task. In fact, stock market price prediction is very fluctuating in nature. They depend on various factors like the previous stock prices, current market trends, financial news, competitor’s performance etc.

- Financial data of stock market is of complex nature, which makes it difficult to predict or forecast the stock market behavior.

Problem Statement





Motivation

- The stock market play a pivotal role in the growth of the industry and commerce.
- The government, industry and central banks of the country keep a close watch on the happenings of the stock market
- For successful investment, many investors are interested in knowing about the future situation of the market.
- Effective prediction systems indirectly help traders by providing supportive information such as the future market direction.

Stock Market Applications

- **Yahoo! Finance**
  Allows users to sync portfolios and quotes across multiple devices
  Tracking stocks, currencies, commodities
  Notifications allow you to stay current on breaking news

- **Stock Market Simulator**
  Users can play a simulated version of the real U.S. stock market and invest virtual funds without taking on any real risk.
  Shouldn't be used as a decision-making tool for actual trading
  A great way for first-time investors to get their feet wet and learn how the stock
  market works without losing any actual money.

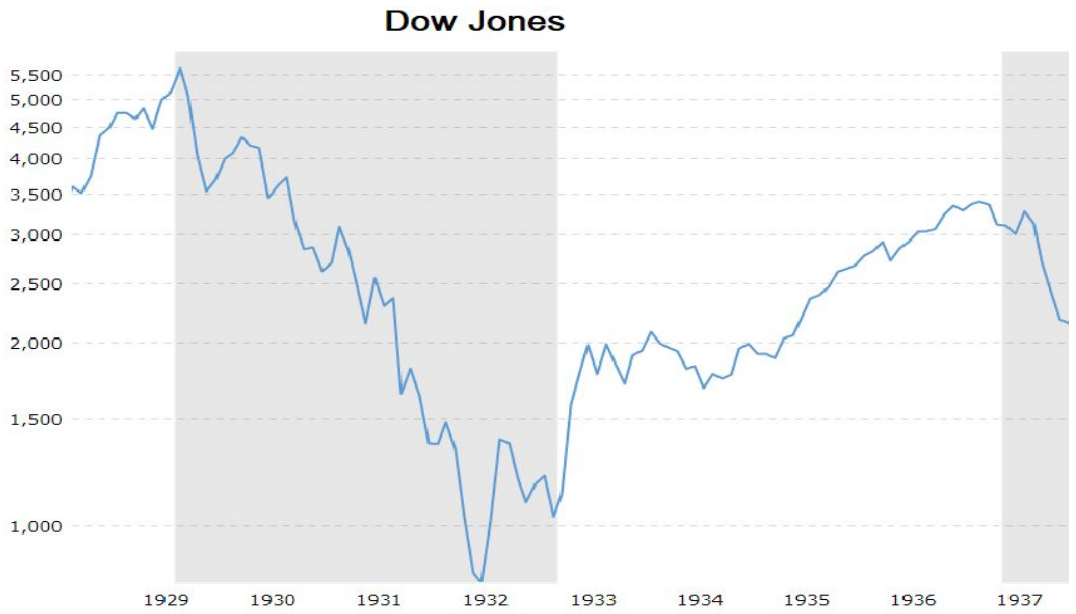
Proposed Solution

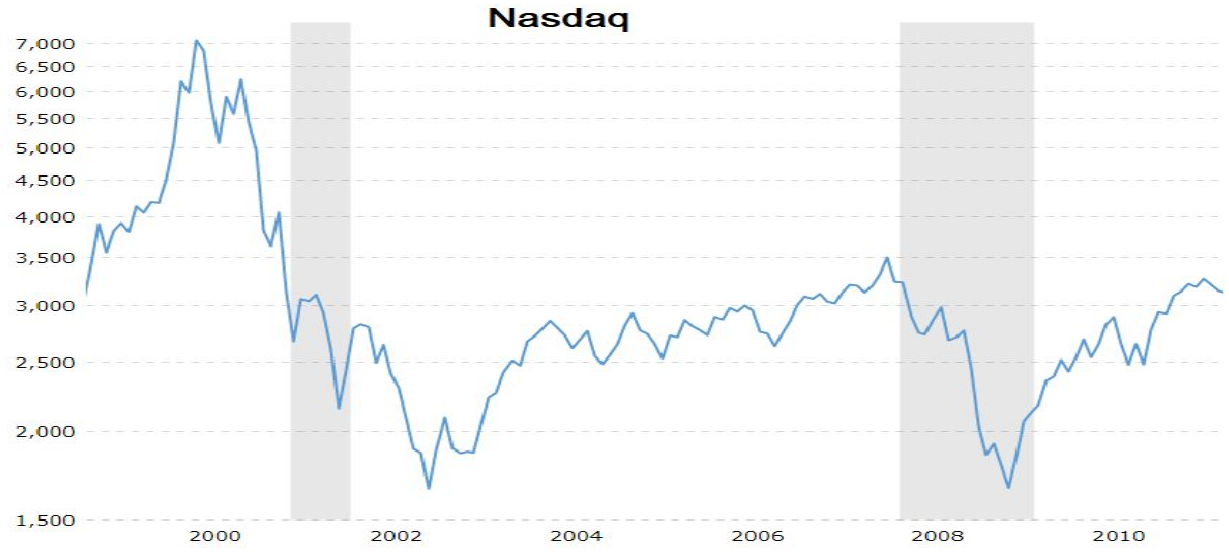
- Study and analyze the historical information.
- Sentiment analysis of Tweets.
- Construct an effective model to predict stock market future trends with small error ratio and improve the accuracy of prediction.
- This model is based on sentiment analysis of financial news from twitter and historical stock market prices that provides better accuracy results considering multiple types of news related to market and company with historical stock prices.

Proposed Solution

Historical Data

- Using stock market historical data and studying specific examples from the stock market past, make it easier to put current market moves in context, while making informed decision.
- notice how the Dow and NASDAQ index charts move in lockstep through their respective decades. The 1929 stock market crash reflected in Dow index overlays almost perfectly against the 2000 NASDAQ index break.





Proposed Solution

Tweets

- Several studies show why. Blake spoor, Miller, and White (2014) suggest that Twitter influences the trading volume within the market.
- Another study showed that sentiment of tweets (including non-company published tweets) are associated with abnormal returns.

Outlines

01 Introduction

02 System Components And Architecture

03 Front End Development

04 Back End Development

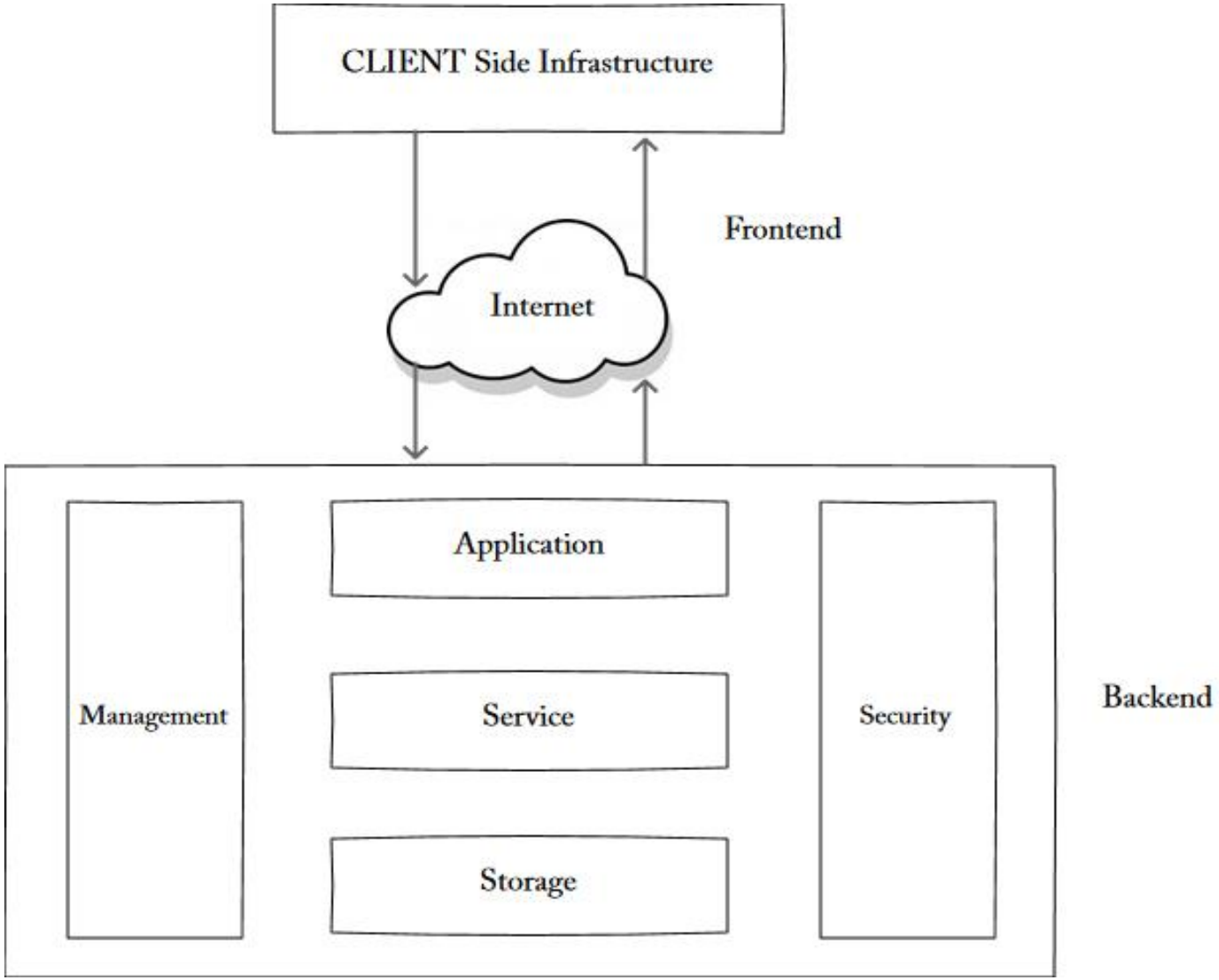
05 Machine Learning For Time Series Forecasting

06 Conclusions & Future Work

System Model

Goals:

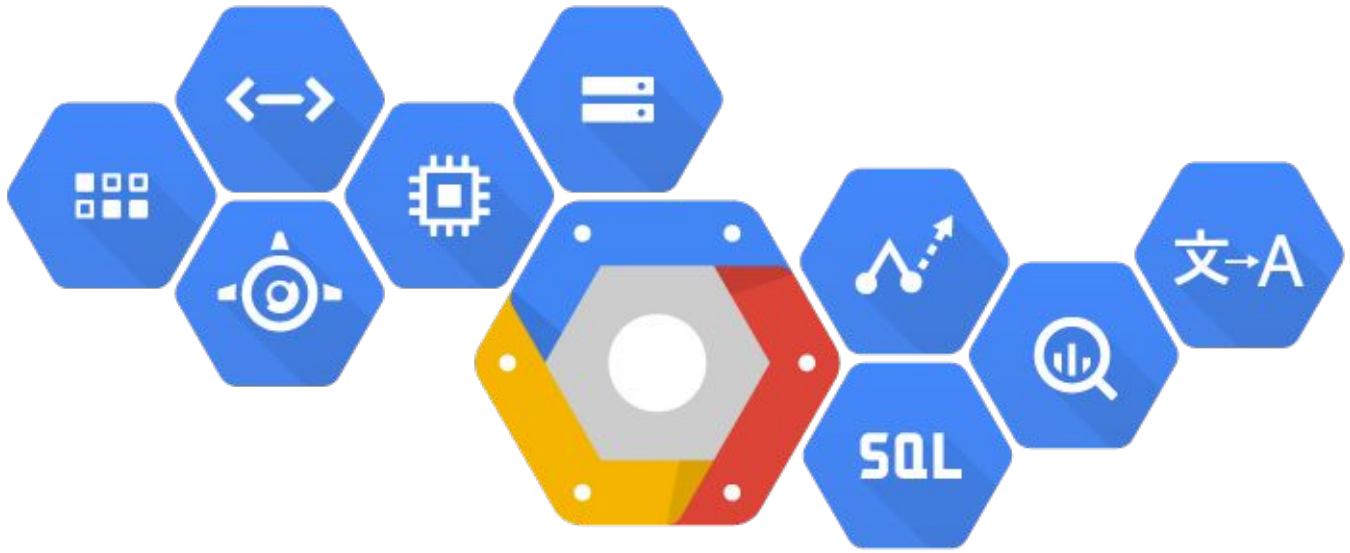
- Facilize the communication between the system components
- Organize the workflow between services
- Facilize data transformation between machine learning model and web application
- Increase the response time of the system to provide customers with updated predictions in time.



System Model

This project built on Google Cloud infrastructure..

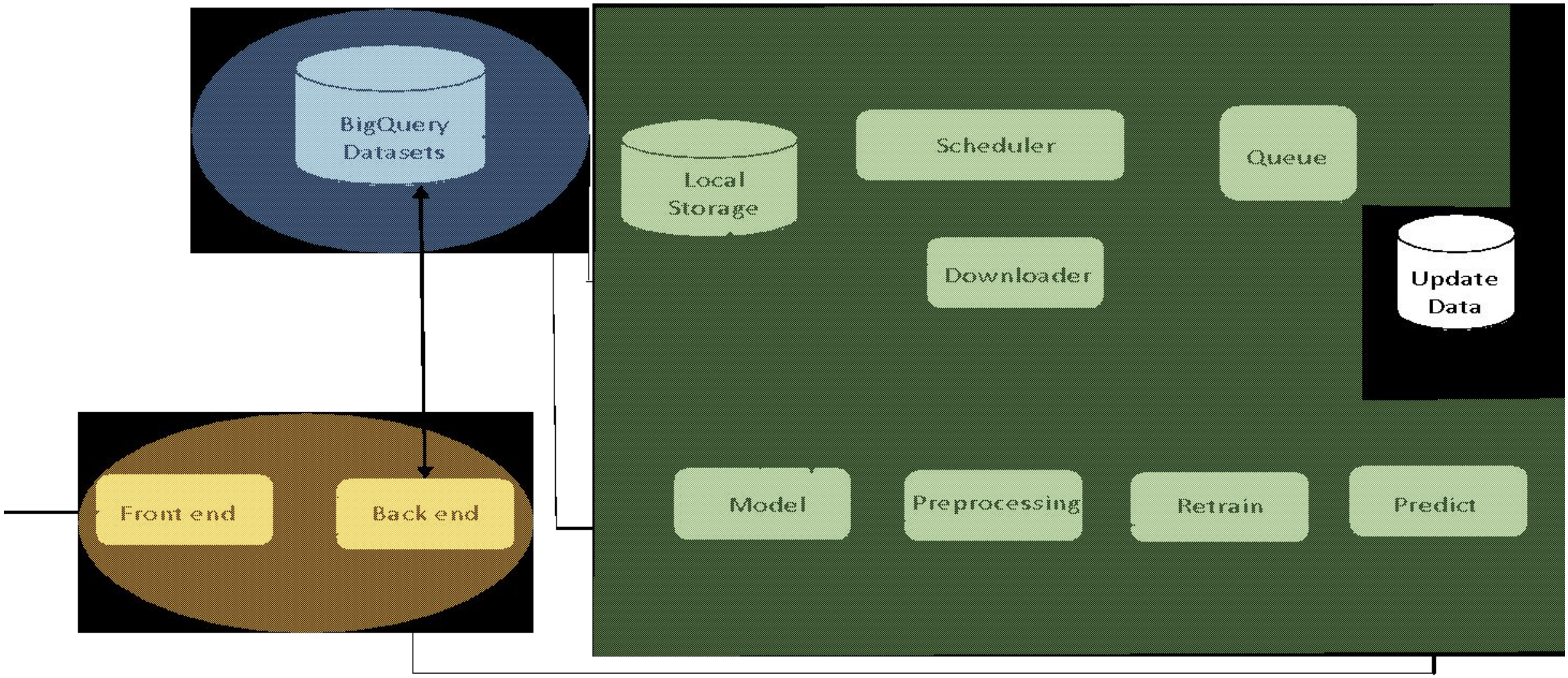
Why Cloud!?
1. Cost efficiency
2. Private Global Fiber + Tiered Network
3. Increase number of users
4. Improved Performance
5. More Security
6. Continued Expansion
7. Scalability
8. Flexibility



What is Google Cloud Platform!?
Google Cloud Platform is a suite of public cloud computing services offered by Google

Google Cloud Platform

System Architecture



Outlines

01 Introduction

02 System Components And Architecture

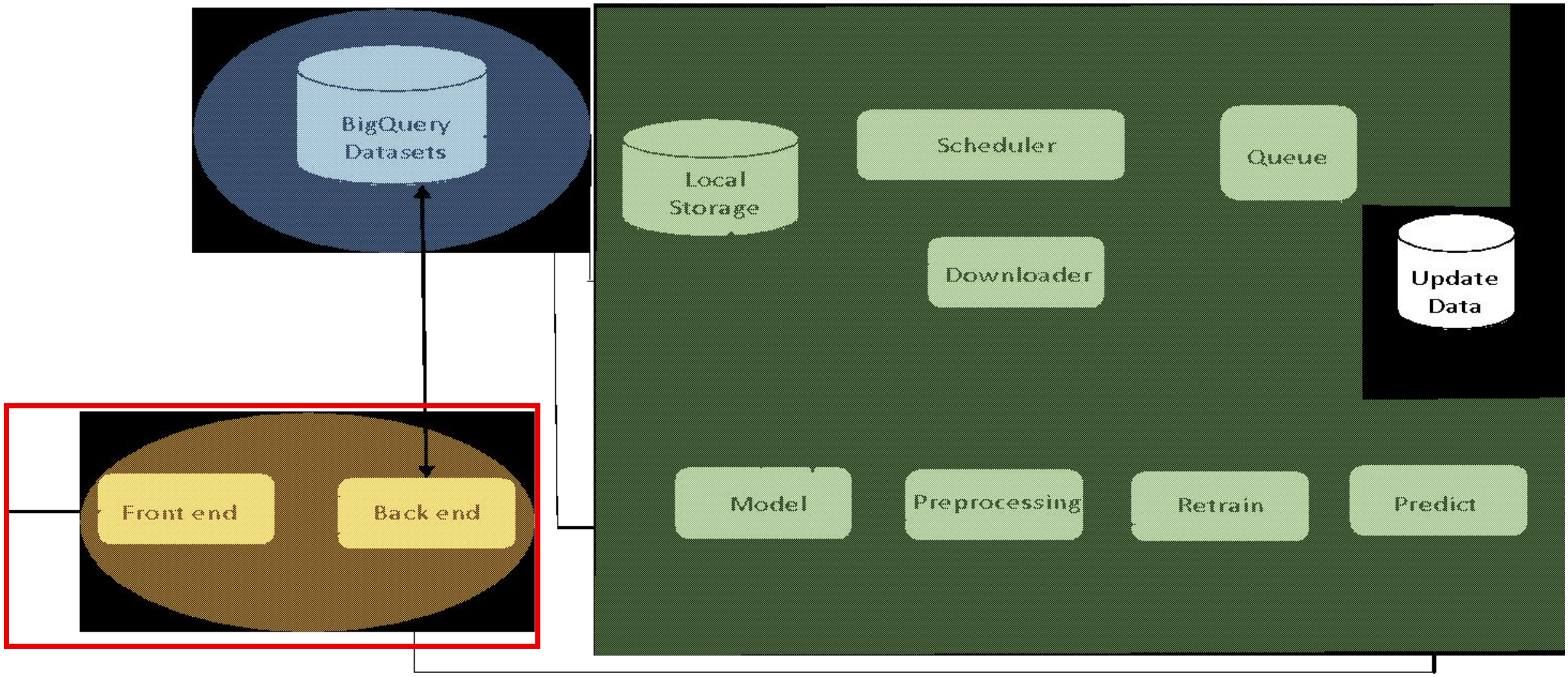
03 Front End Development

04 Back End Development

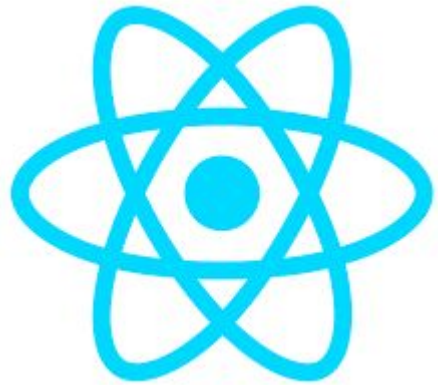
05 Machine Learning For Time Series Forecasting

06 Conclusions & Future Work

System Architecture



Html, css and..React js!



What is React js?

React (also known as React.js or ReactJS) is a JavaScript library for building user interfaces.

React allows developers to create large web applications which can change data, without reloading the page.

Why React js?

- It’s about using reusable components.
- Small Learning Curve, much less than other frameworks.
- React has a huge community of support.

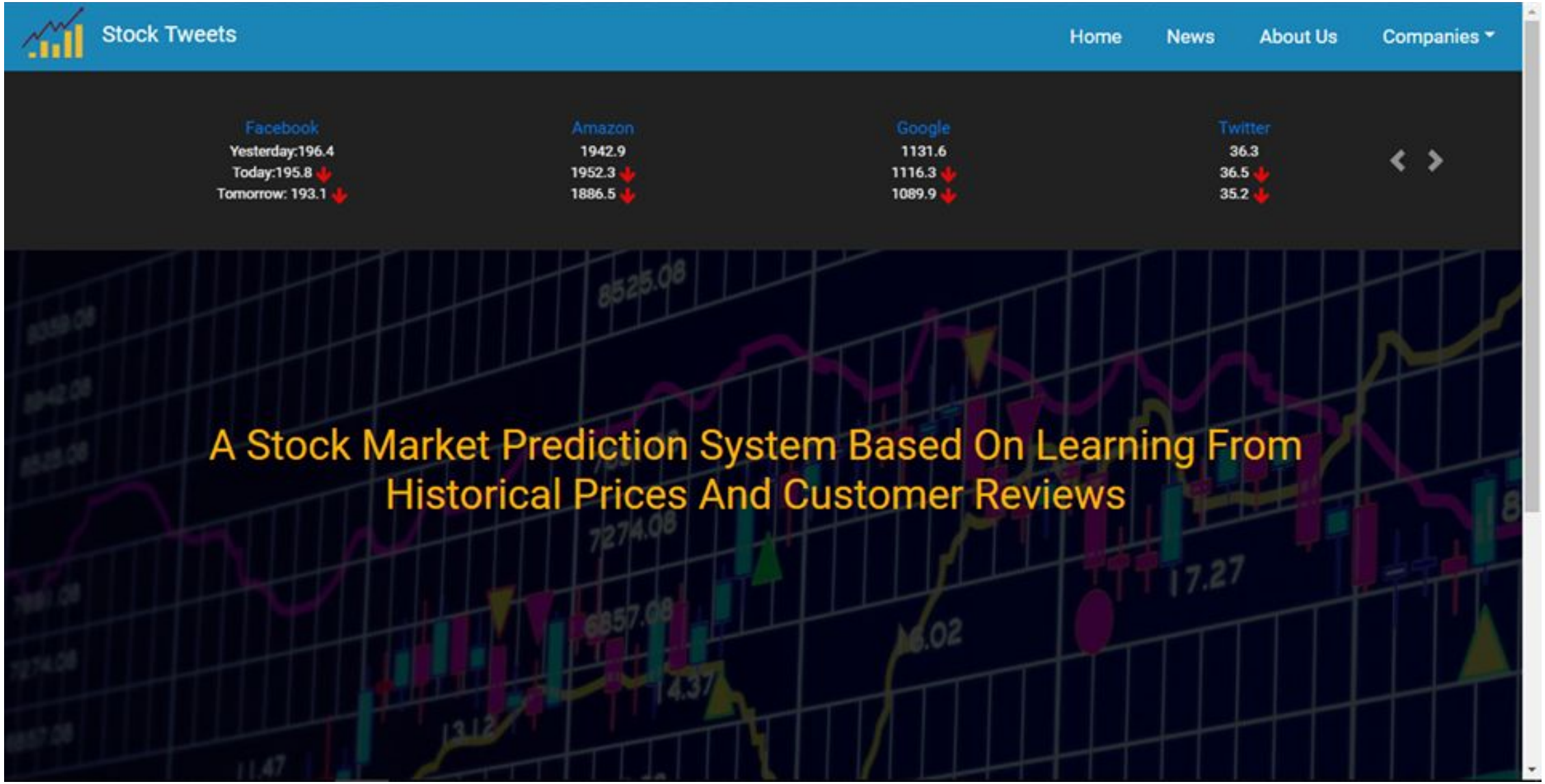
Data Visualization

- Is representing data in a visual form. This visual form can be a chart, graphs, lists. This representation helps people to understand the magnitude of the data.
- To communicate information clearly and efficiently to users.
- Helps data analytics to get better insights.
- Data visualization and analytics will draw the conclusions about the datasets.

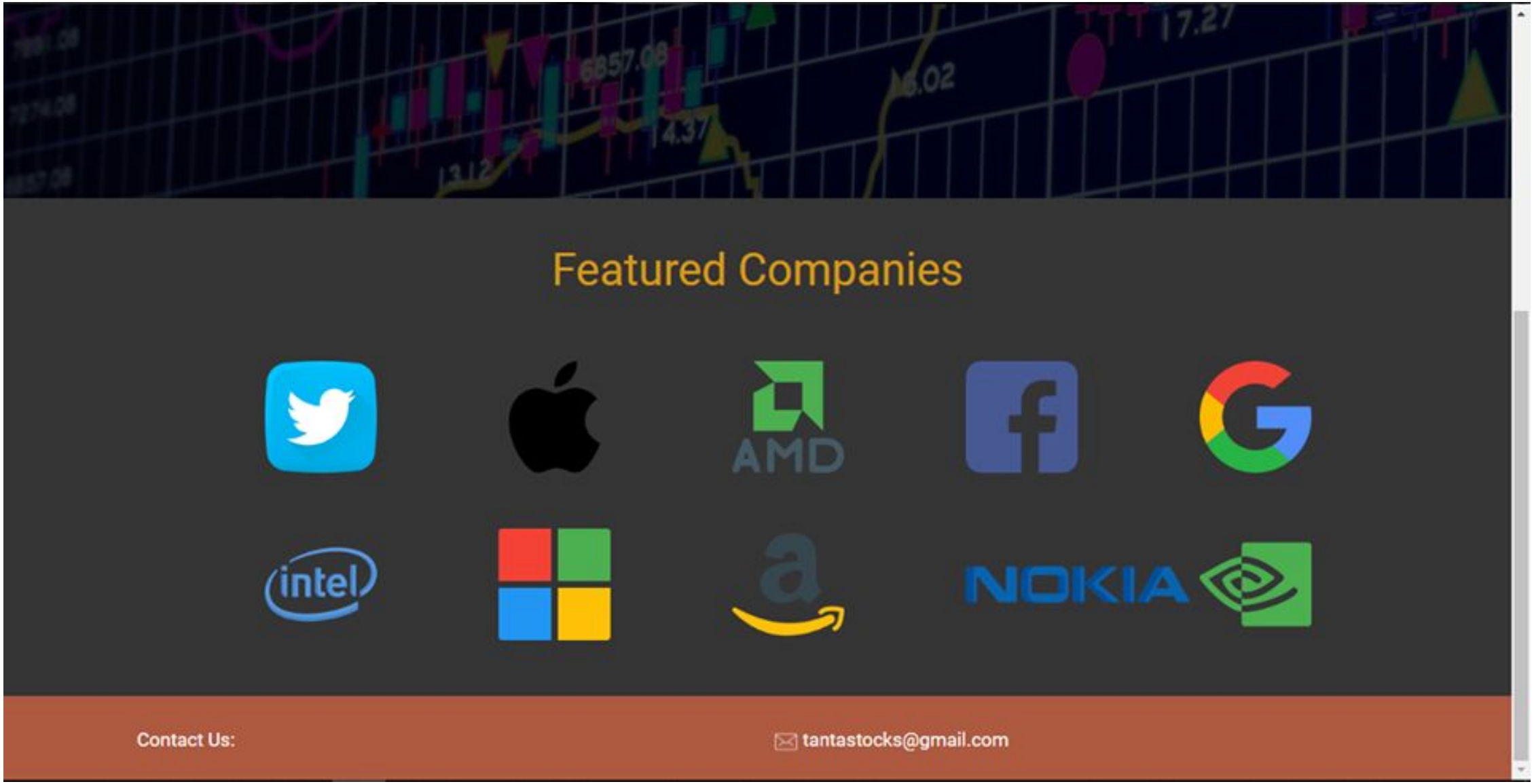
The Tool We Used:

React Stockcharts: creates highly customizable stock charts built with React js and d3.

Home page


Stock Tweets
Home
News
About Us
Companies
Facebook
Yesterday:196.4
Today:195.8
Tomorrow: 193.1
Amazon
1942.9
1952.3
1886.5
Google
1131.6
1116.3
1089.9
Twitter
36.3
36.5
35.2
A Stock Market Prediction System Based On Learning From Historical Prices And Customer Reviews

Featured Companies

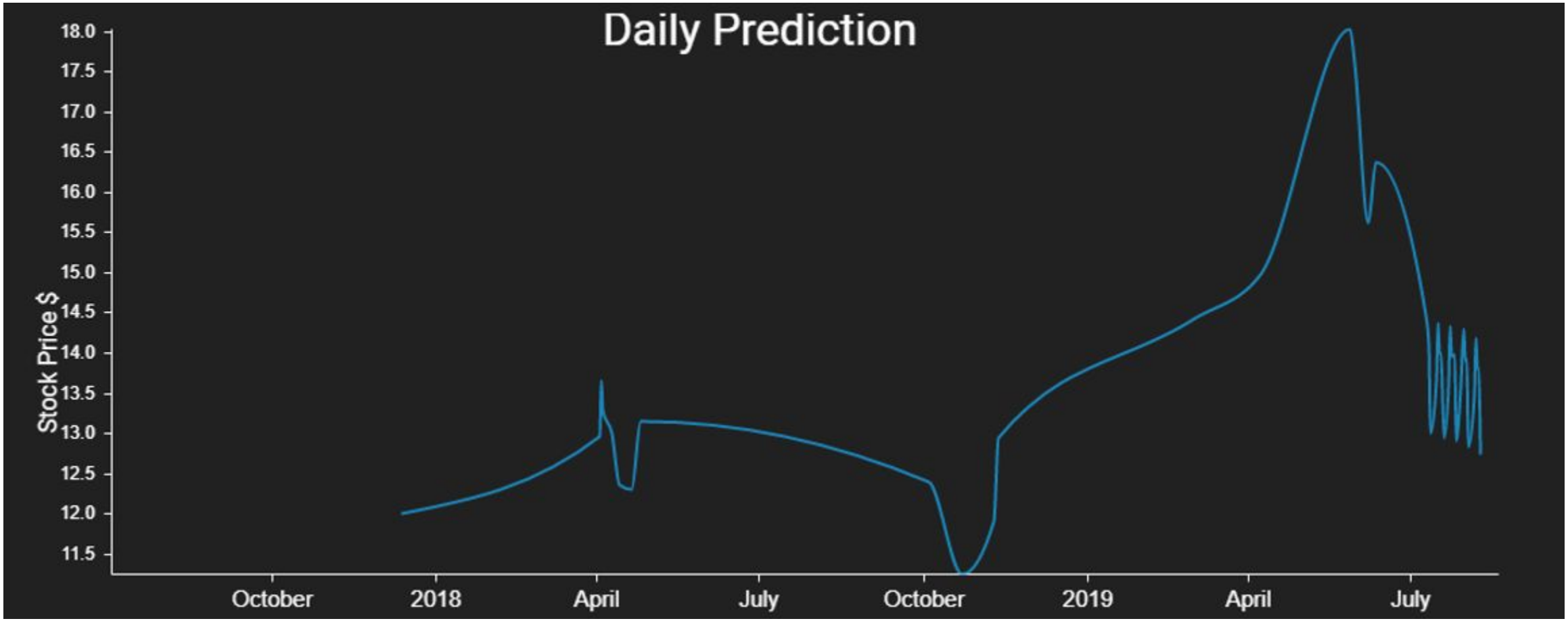


Company Page

Historical Prices

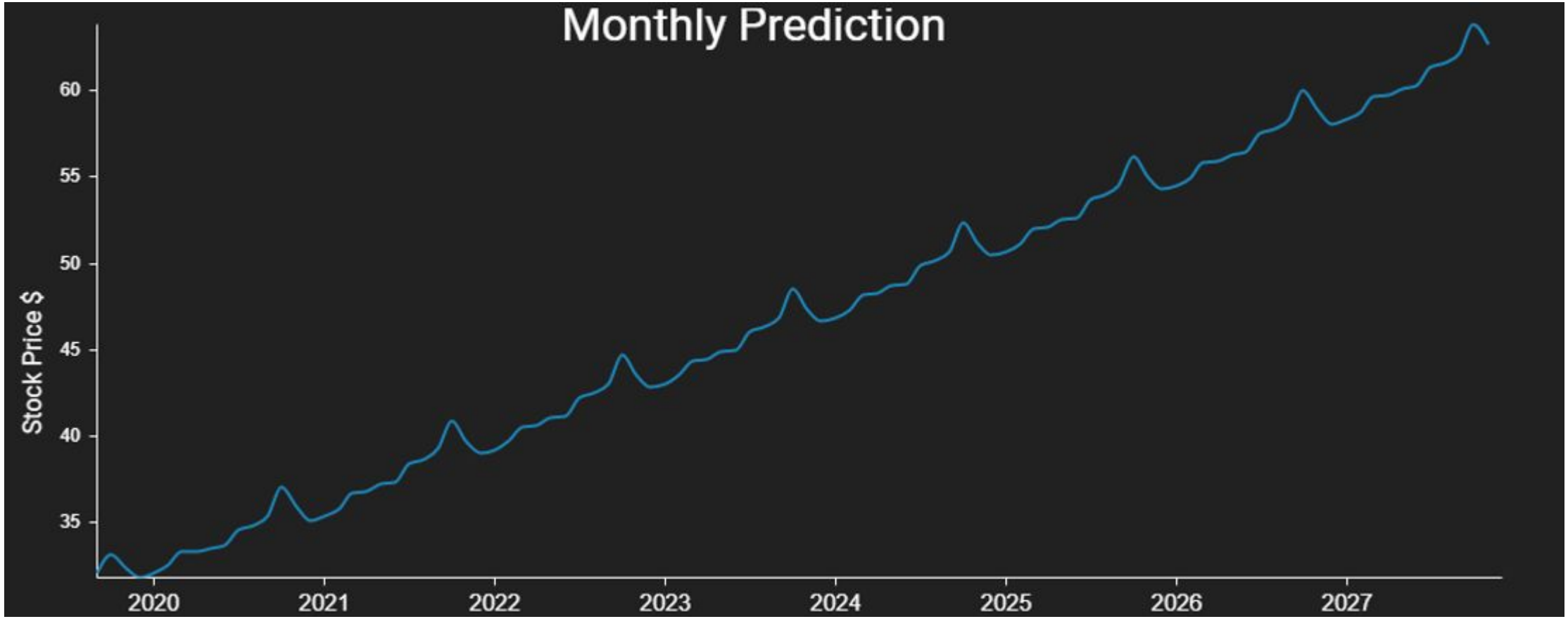
Company Page

Daily Prediction



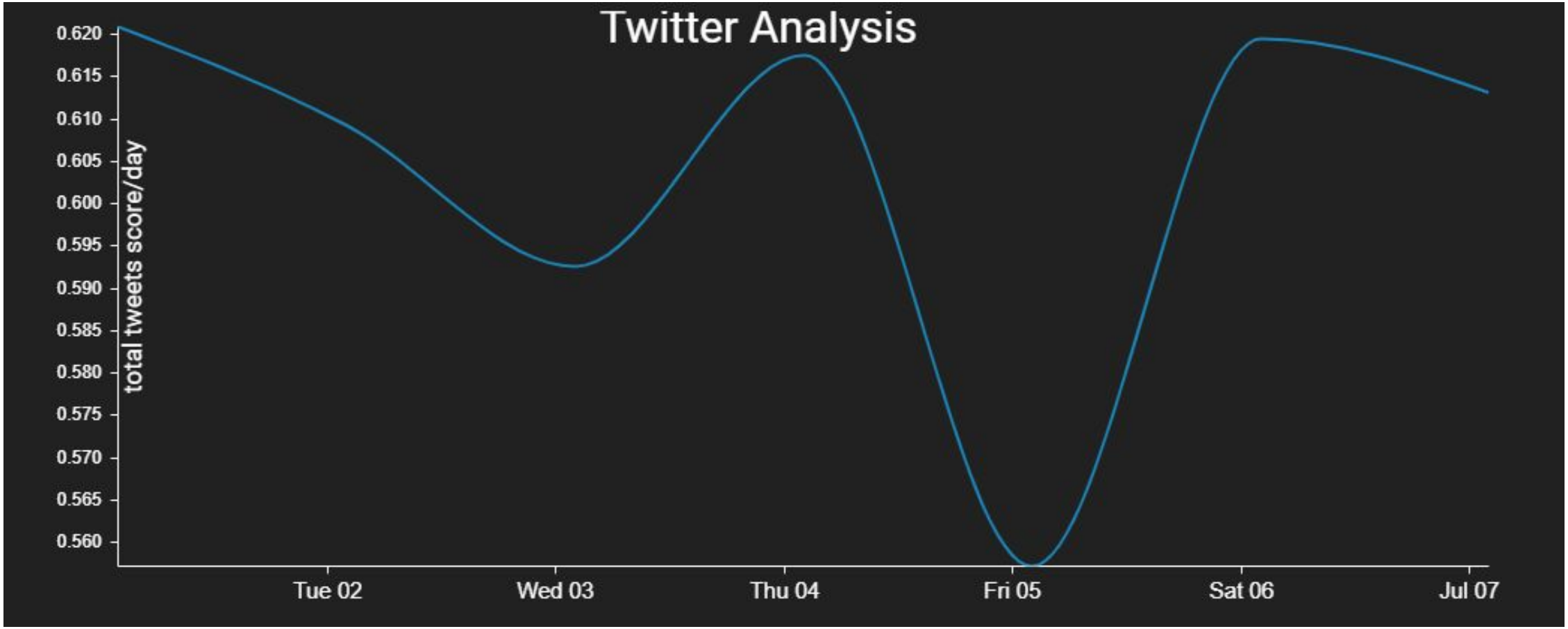
Company Page

Monthly Prediction



Company Page

Twitter Analysis



News(Twitter)

News(Financial News)

Top News

Amazon lays out constellation service goals, deployment and deorbit plans to FCC - SpaceNews

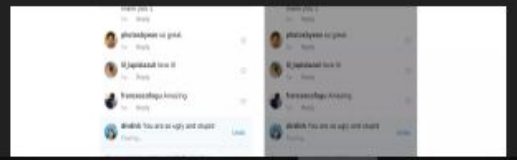
Amazon released more details on its plan to deploy 3,236 broadband satellites, telling U.S. telecom regulators the constellation can start service in limited areas with less than a fifth of the total constellation.

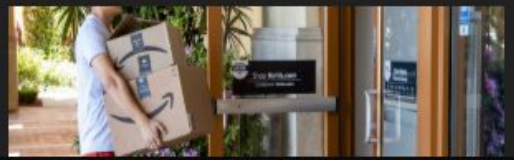
Apple 'fundamental deterioration' to last into 2020, analyst predicts - MarketWatch

Apple Inc.'s iPhone business could disappoint in the next upgrade cycle, following a massive flop for the iPhone XS over the past few months.

In a scene reminiscent of the financial crisis, axed Deutsche Bank workers leave with belongings - CNBC

Carrying boxes and envelopes containing personal effects and A4 forms, many Deutsche workers started their work week by emptying their desks.





Outlines

01 Introduction

02 System Components And Architecture

03 Front End Development

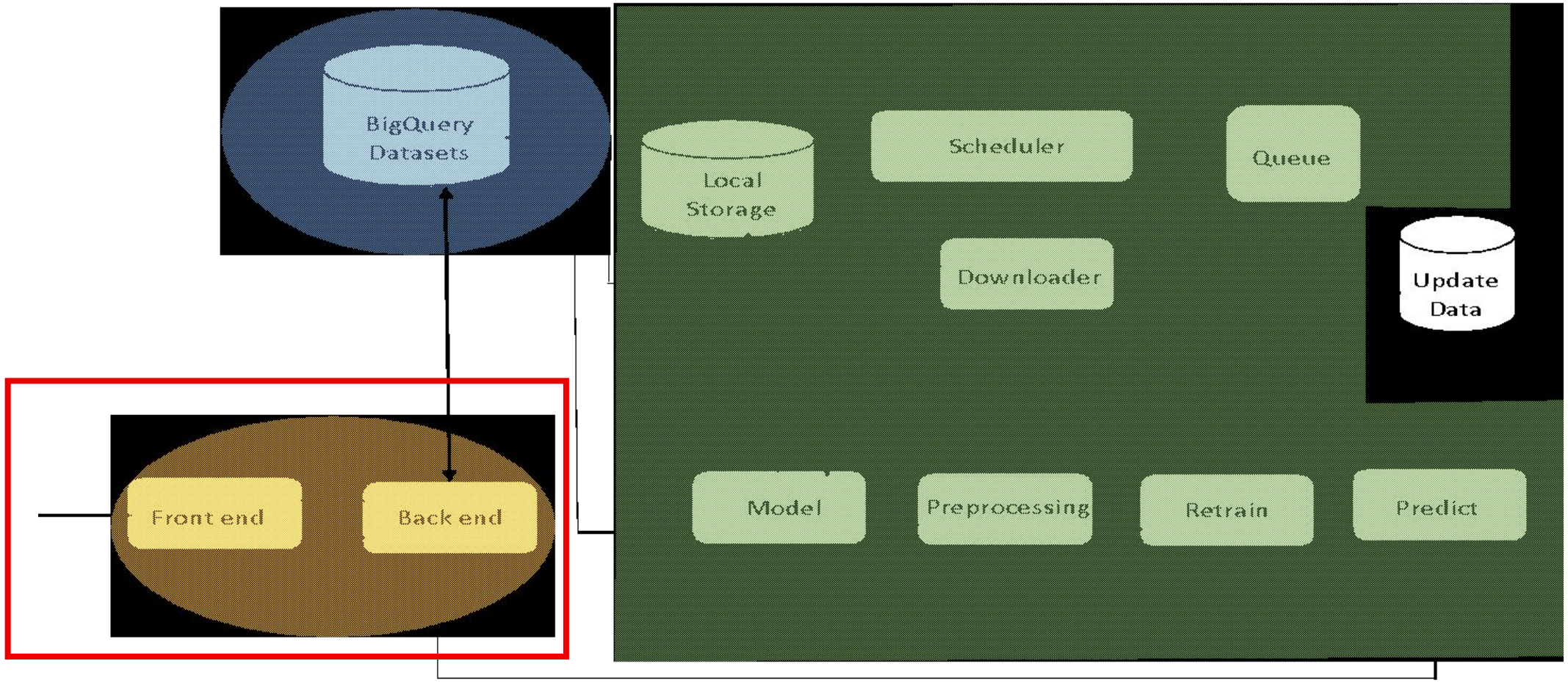
04 Back End Development

05 Machine Learning For Time Series Forecasting

06 Conclusions & Future Work

Back End Development

- System Architecture

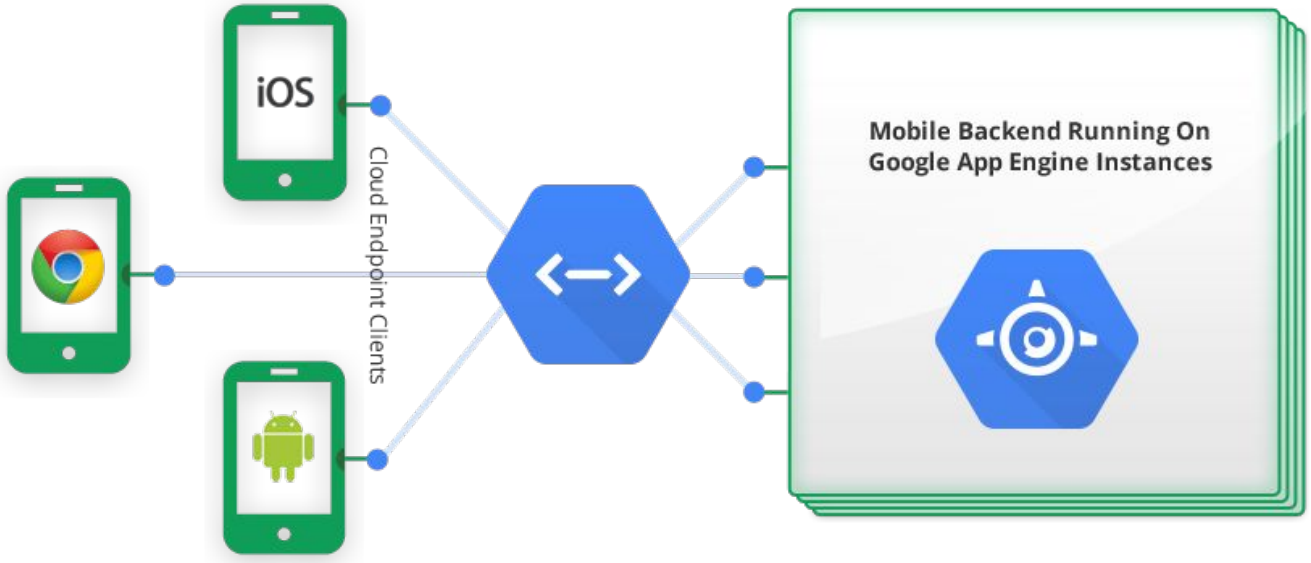


Back End Development

- App Engine

What is app engine & features?

- Google App Engine is a web framework and cloud computing platform for developing and hosting web applications in Google-managed data centers.



Back End Development

- App engine features

- ❑ **Salable**
- ❑ **Runtimes**
- ❑ **Data Storage**
- ❑ **Infrastructure**
- ❑ **Security**
- ❑ **Faster time**
- ❑ **Easy to use and maintain.**
- ❑ **communication between product.**
- ❑ **SDK**

by download SDK, we can start the project.

Back End Development

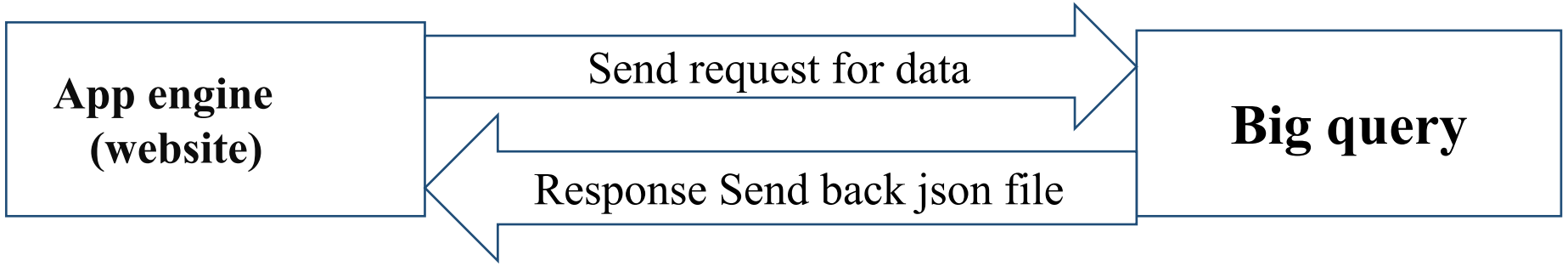
- What is Nodejs ?

- Node.js is a cross-platform runtime environment and library for running JavaScript applications outside the browser. It is used for creating server-side and networking web applications.

- Fast Deployment, Automatic scaling and Reliable Security

Back End Development

- Storage



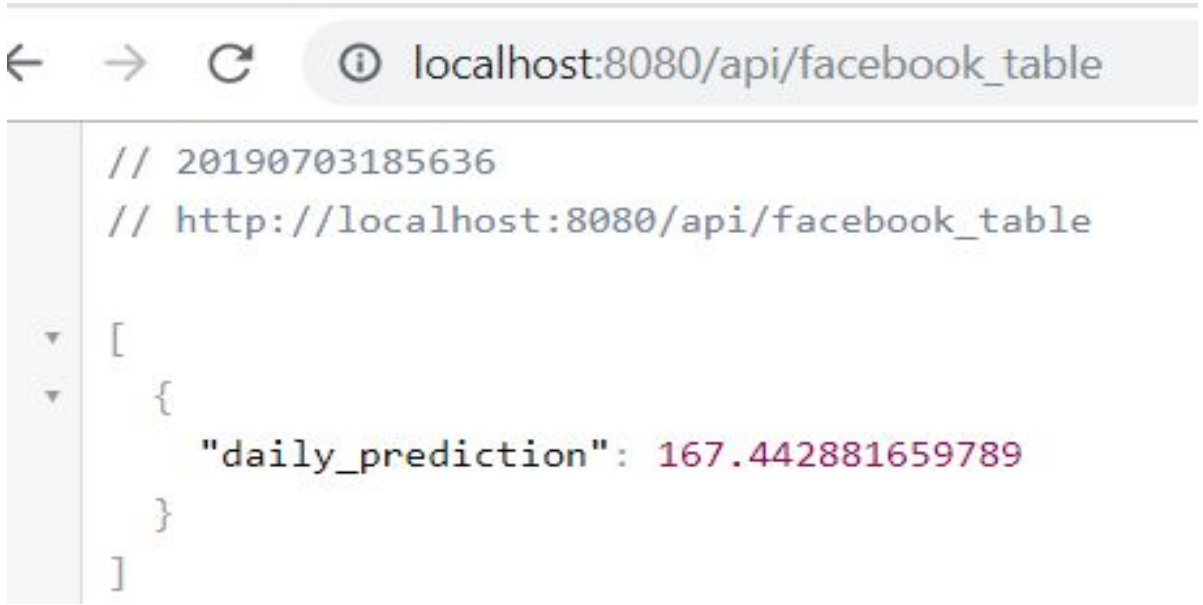
Back End Development

- Connection example

This API call for Facebook stocks prediction.
Call query from bigquery , the result json file

```
app.use('/api/facebook_table', facebook_table);
```

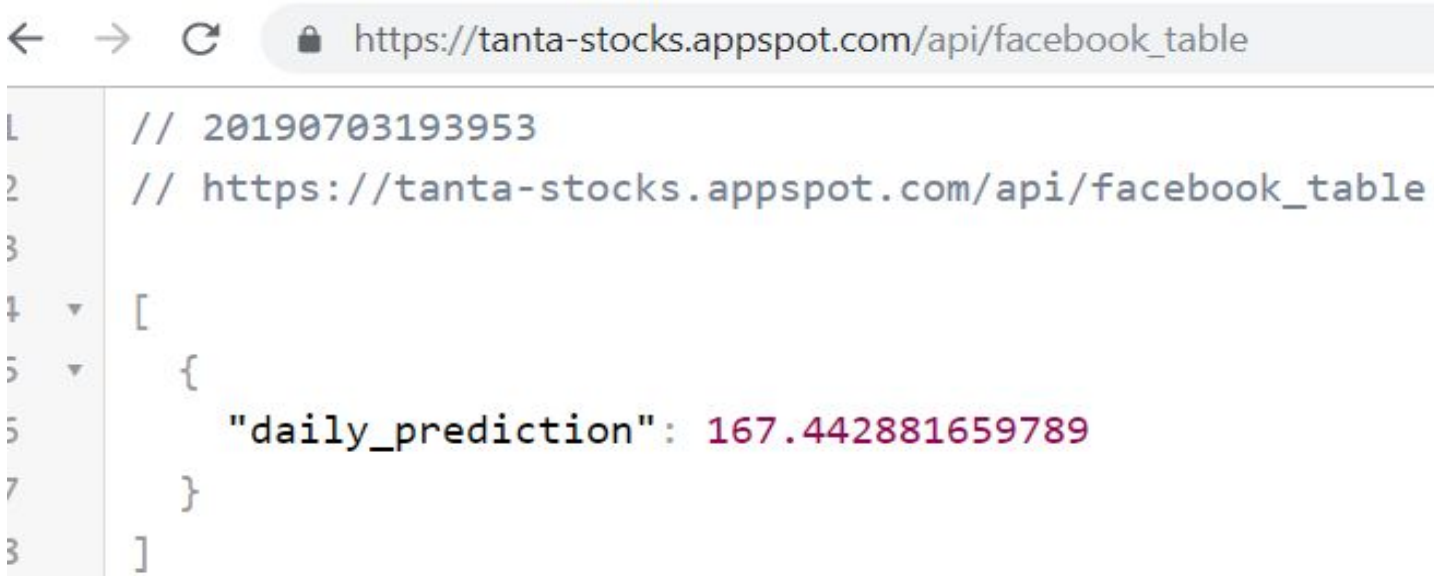
This API call could be local like this



Back End Development

- Connection example

Then after deploy the backend folder on google cloud:



Back End Development

- Connection example

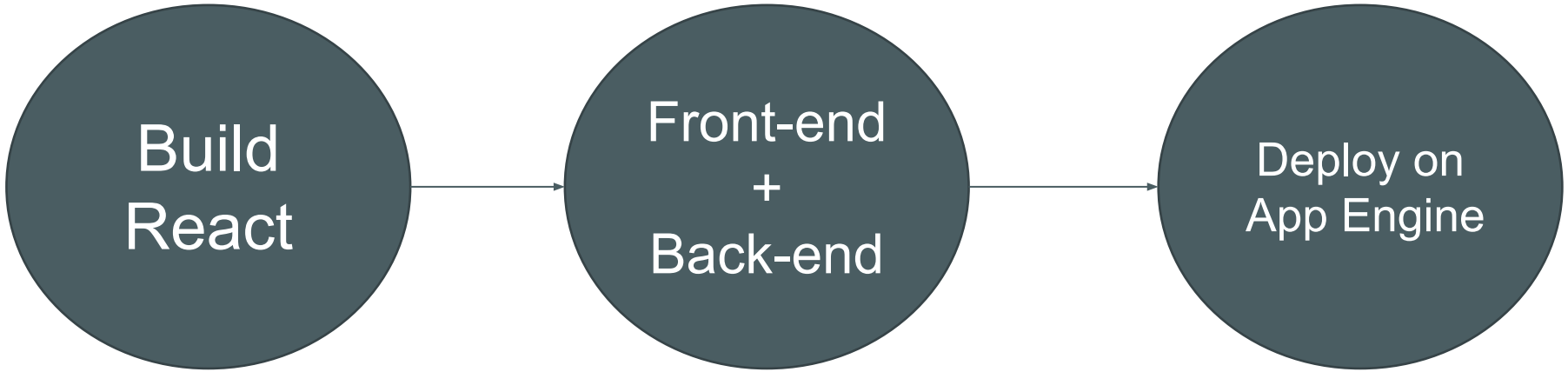
In Front-End Use Axios to call API

Axios is a JavaScript library used to make HTTP requests

```
    axios.get("https://tanta-stocks.appspot.com/api/facebook_table")
    .then(res=>
{
  const predfb= res.data ;
  this.setState({predfb}) ;
  //console.log(this.state.predtw)
}
    )
```

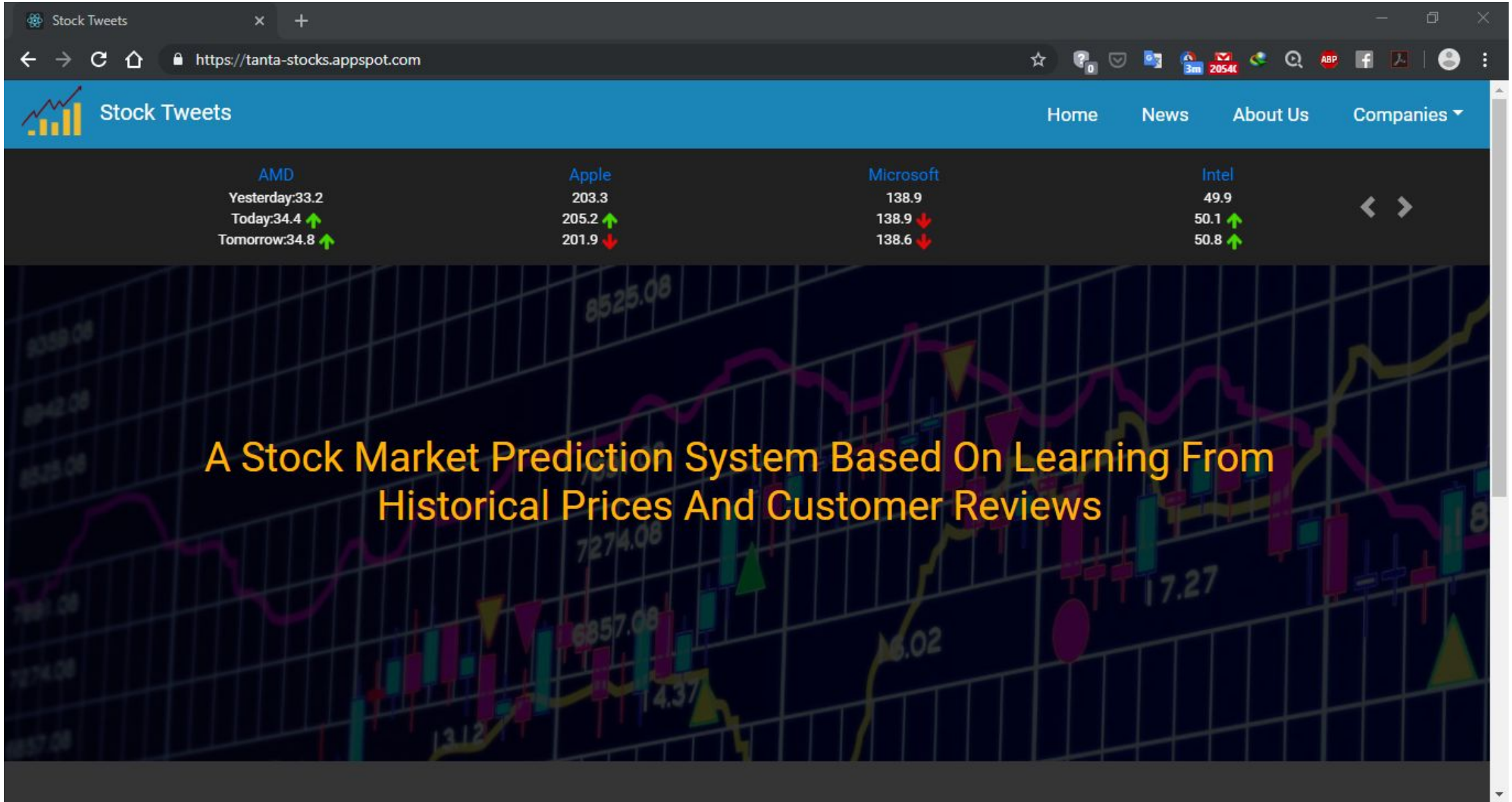
Back End Development

- Deployment



http://www.tanta-stocks.appspot.com

Home page


Stock Tweets
https://tanta-stocks.appspot.com
Stock Tweets
Home
News
About Us
Companies
AMD
Yesterday:33.2
Today:34.4
Tomorrow:34.8
Apple
203.3
205.2
201.9
Microsoft
138.9
138.9
138.6
Intel
49.9
50.1
50.8
A Stock Market Prediction System Based On Learning From Historical Prices And Customer Reviews

Outlines

01 Introduction

02 System Components And Architecture

03 Front End Development

04 Back End Development

05 Machine Learning For Time Series Forecasting

06 Conclusions & Future Work

Machine Learning For Time Series Forecasting

Phase1
Historical Data

Phase2
Social media Data(Twitter)

Phase3
Historical + Twitter Data

Machine Learning For Time Series Forecasting

Phase1
Historical Data

Phase2
Social media Data(Twitter)

Phase3
Historical + Twitter Data

Phase 1

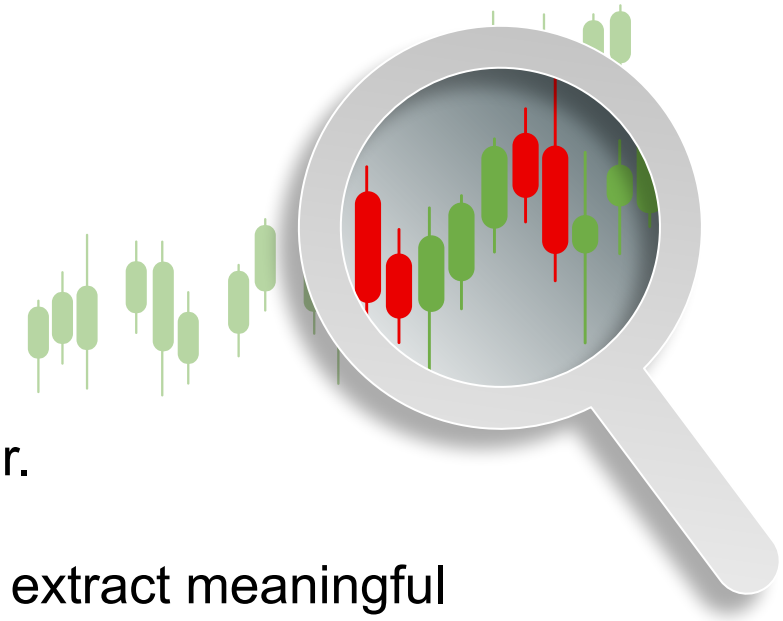
ARIMA Model

For Long term Prediction

FBProphet

For Short term Prediction

Time Series Data



Time-series data is a sequence of observations stored in time order.

Time series analysis is an approach to analyze time series data to extract meaningful characteristics of data and generate other useful insights applied in business situation.
Time series analysis helps understand time-based patterns of a set of metric data points which is critical for any business.

ARIMA Model

ARIMA stands for Auto Regressive Integrated Moving Average.

The AutoRegressive (AR) model
specifies that the output variable depends linearity on its own previous values.

$$Yt = \beta 0 + \beta 1\, Yt\text{-}1 + \beta 2\, Yt\text{-}2 + .. + \beta p Yt\text{-}p$$

Moving average Model
In statistics, a moving average is a calculation to analyze data points by creating a series of averages of different subsets of the full data set.

$$MA=(A1+A2+\ldots+An)/n$$

ARIMA is a combination of **Auto Regression Model** and **moving Average** Model with **difference** to make data stationary.
The parameters of the ARIMA model are defined as follows:

- **p** is the number of autoregressive terms(AR Model)
- **q** is the number of lagged forecast errors in the prediction equation(MA Model)
- **d** is the number of nonseasonal differences needed for stationarity(Integration level)

Historical Data API

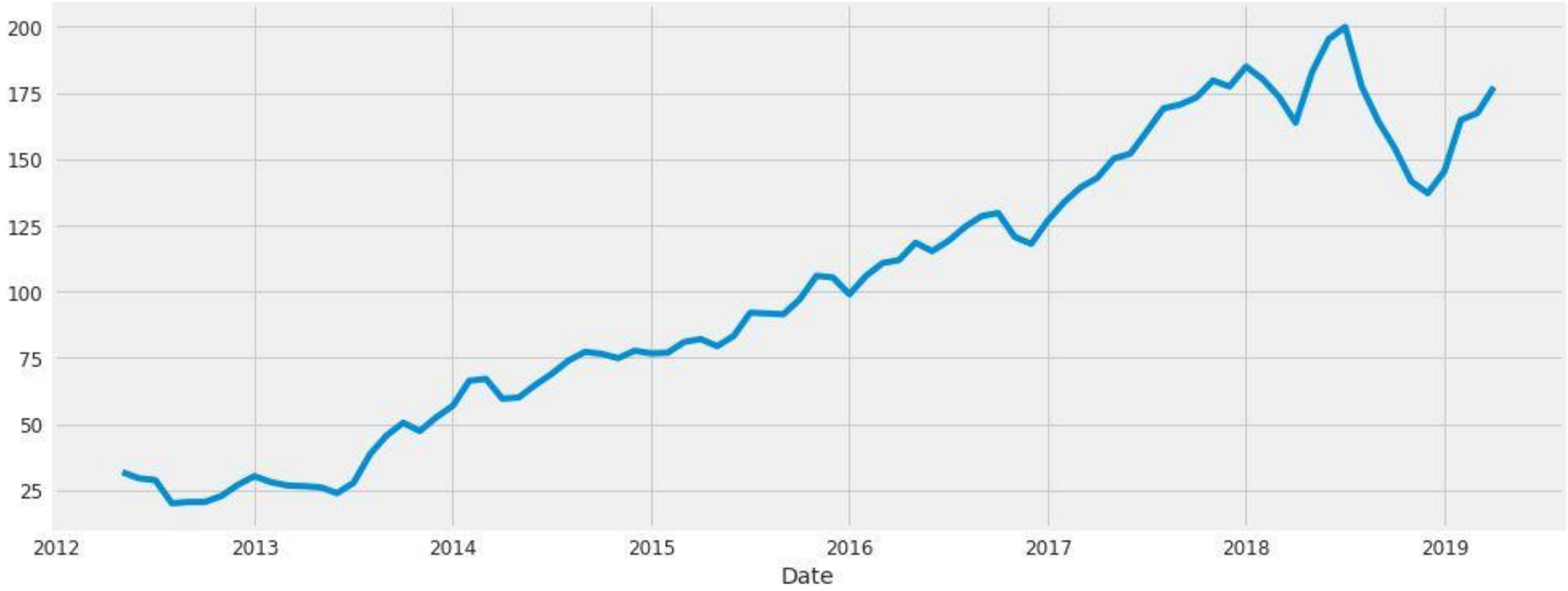
- Yahoo Finance is our source of stock prices data.
- Yahoo Finance APIs in python is available in different ways and sources. And in this point, we choose pandas-datareader to become our API to crewel data from Yahoo Finance.
- Our data considers 10 main corporations.

Data Schema

Schema | Details | Preview

Field name	Type
Date	TIMESTAMP
High	FLOAT
Low	FLOAT
Open	FLOAT
Close	FLOAT
Volume	INTEGER
AdjClose	FLOAT

Visualizing Close time series data



Validating forecasts

To help us understand the accuracy of our forecasts, we compare predicted Close to real close price of the time series

Producing and visualizing forecasts

FBprophet Model

Fbprophet is an open source released by Facebook in order to provide some useful guidance for producing forecast at scale.

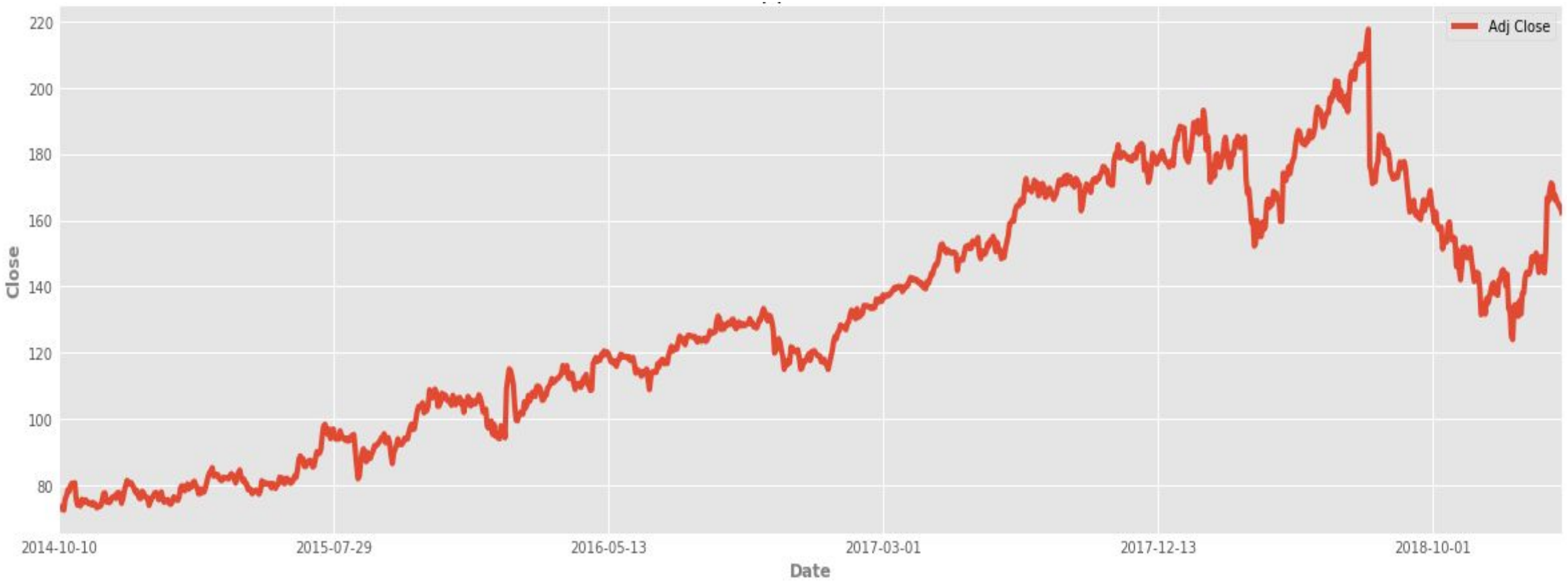
fbprophet package in Python or R can help us look deeper into the hidden cycles in stock market.

The Prophet uses a decomposable time series model with three main model components: **trend, seasonality**, and **holidays**. They are combined in the following equation:
y(t)= g(t) + s(t) + h(t) + εt

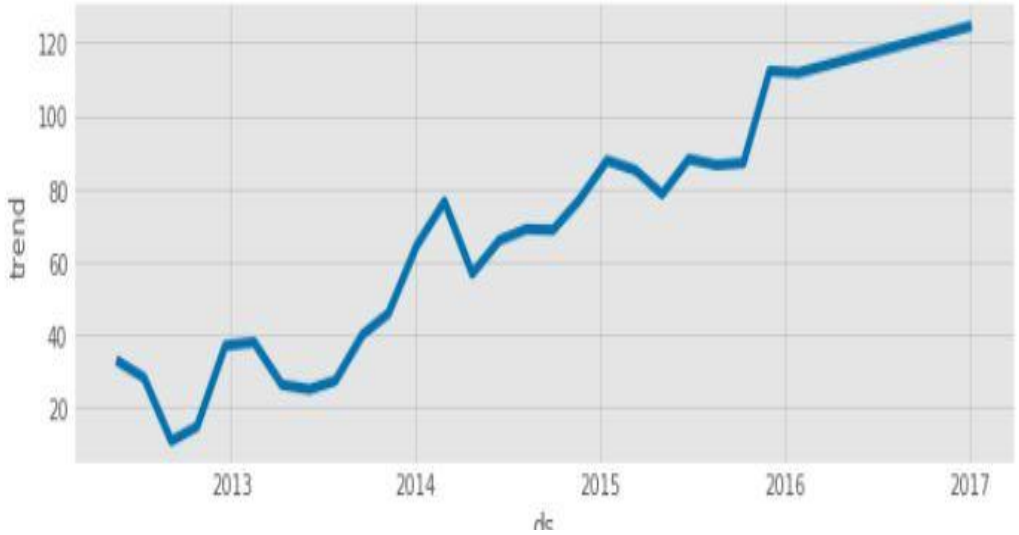
- **g(t):** piecewise linear or logistic growth curve for modeling non-periodic changes in time series.
- **s(t):** periodic changes (e.g. weekly/yearly seasonality).
- **h(t):** effects of holidays (user provided) with irregular schedules.
- **εt:** error term accounts for any unusual changes not accommodated by the model.

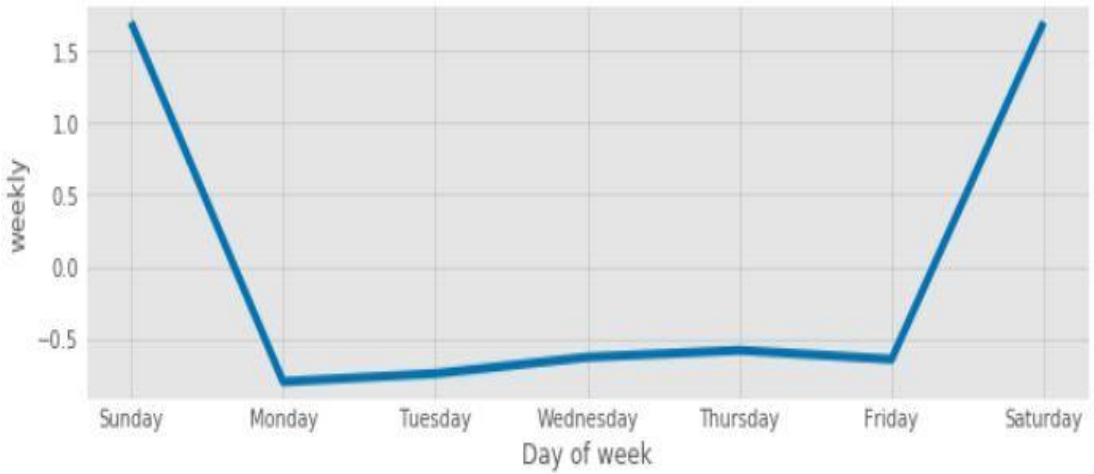
The important idea in Prophet is that by doing a better job of fitting the trend component very flexibly, we more accurately model seasonality and the result is a more accurate forecast

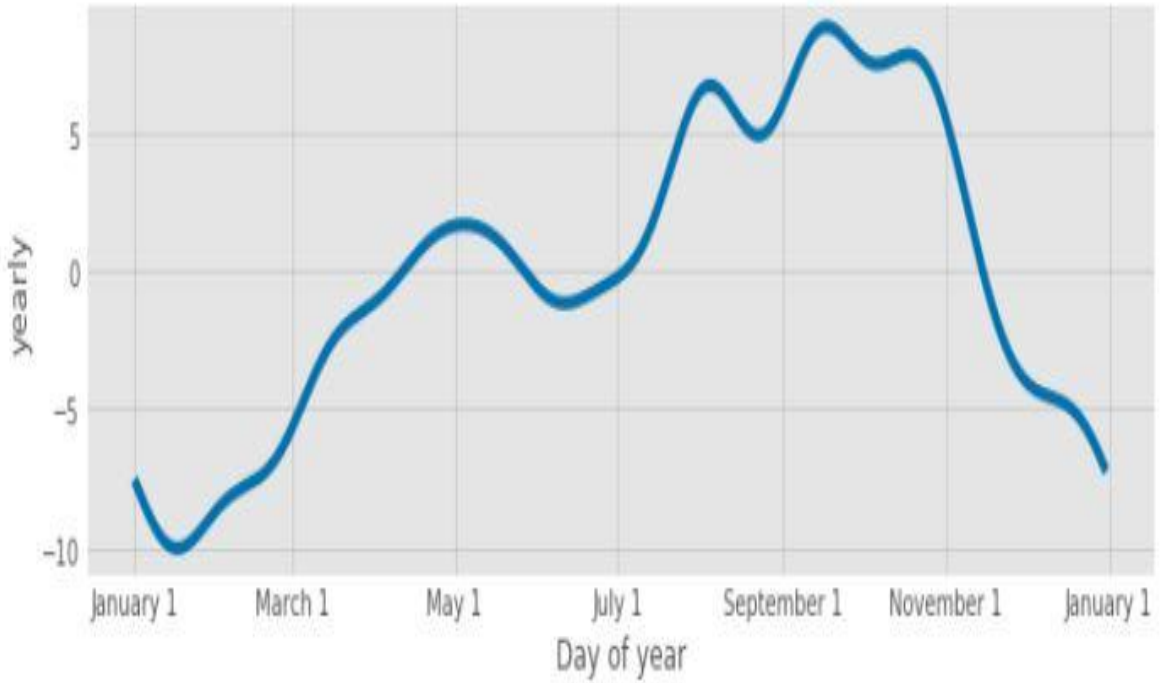
Apple stock price from 2014/10/10 to 2019/2/15



We can also visualize our data according to different components of FBprophet model: weekly, yearly, trend and user-defined cycle

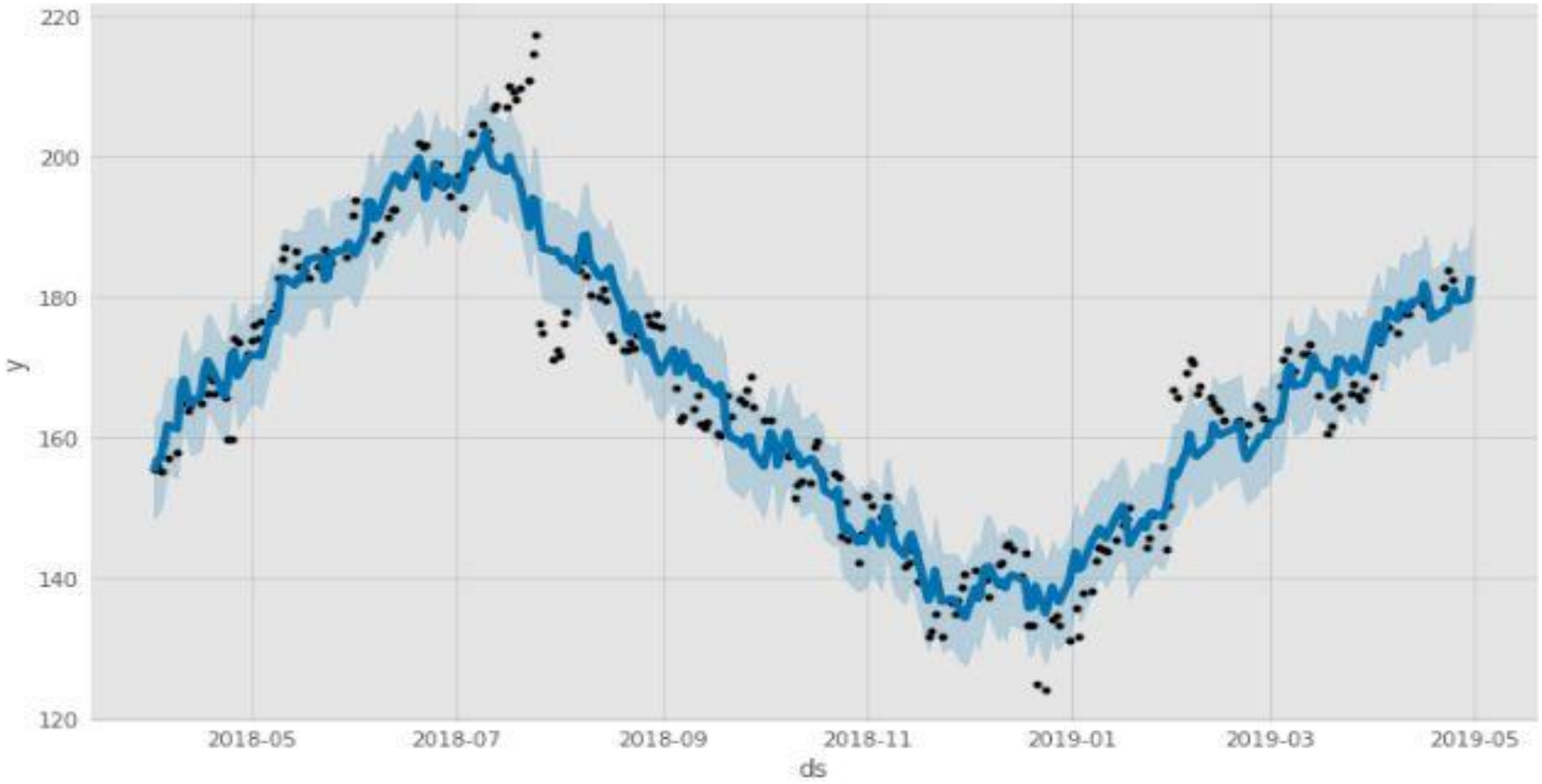






Model validation

In our model we try to use last 12-month data to predict the next 10 days.



Machine Learning For Time Series Forecasting

Phase1
Historical Data

Phase2
Social media Data(Twitter)

Phase3
Historical + Twitter Data

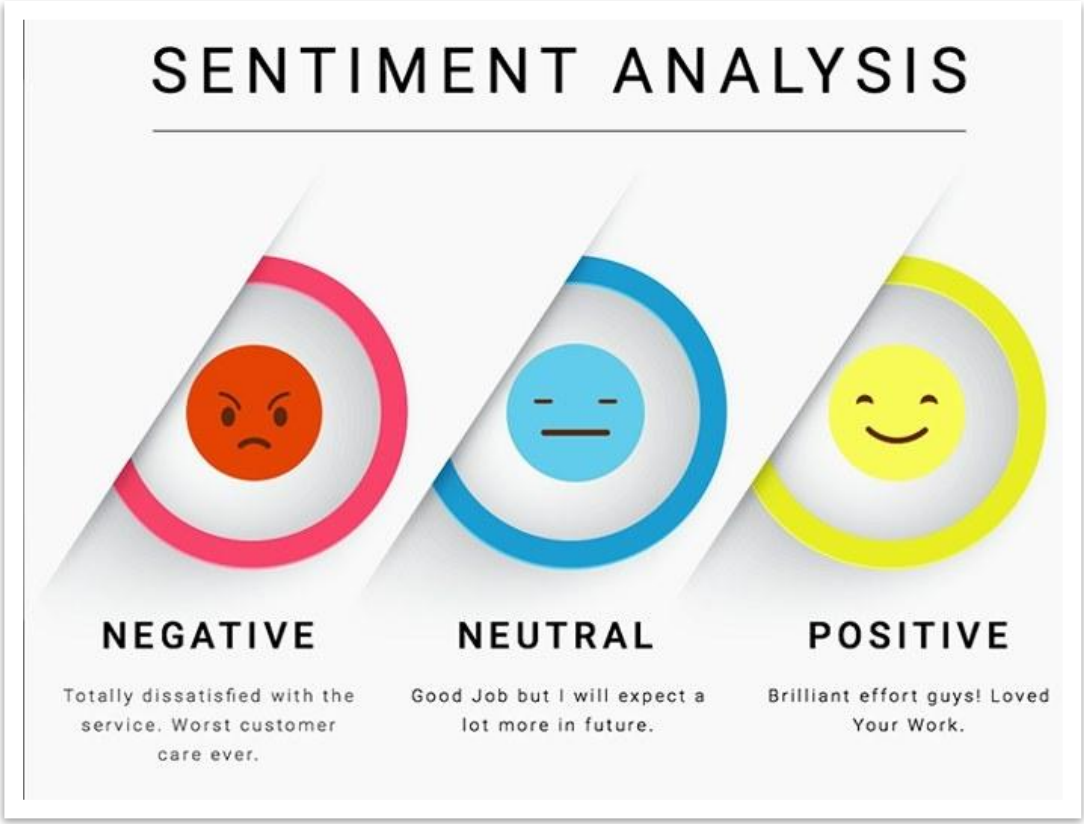
Phase 2

Sentimental Analysis

Nature Language Processing

Sentiment Analysis

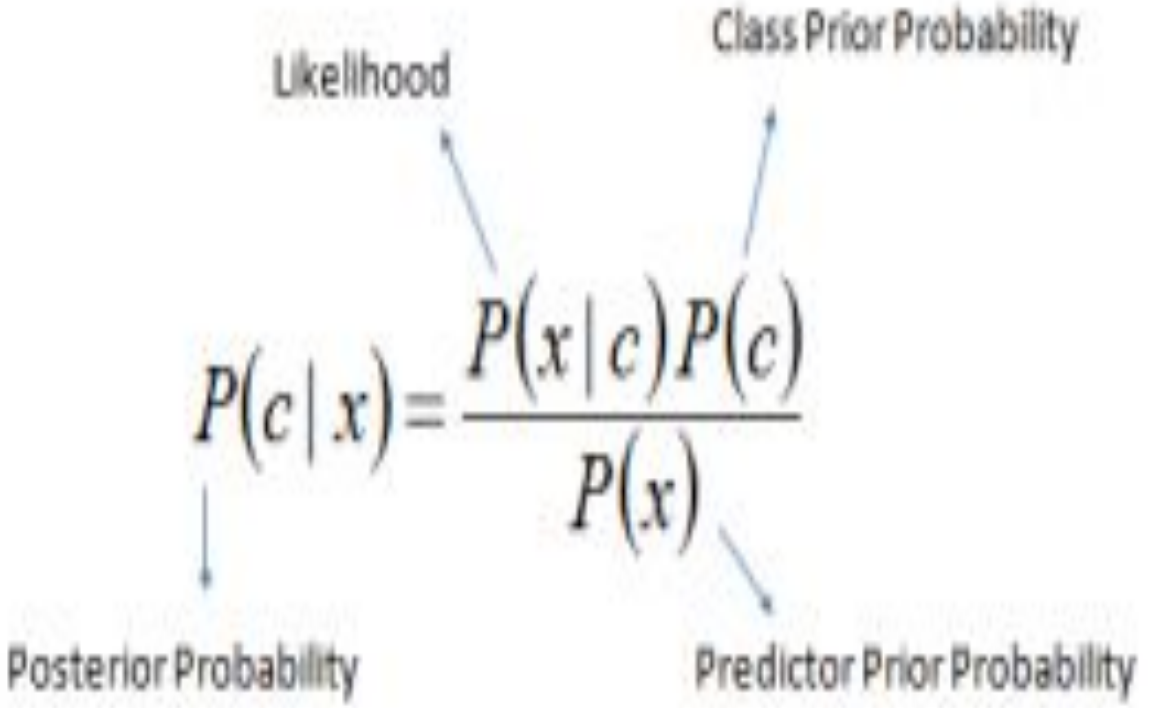
Sentiment Analysis is the most common text classification tool that analyses an incoming message and tells whether the underlying sentiment is positive, negative our neutral.



Naïve Bayes

Bayes theorem calculations :

- P(c|x) is the probability of class (c, target) given predictor (x, attributes).
- P(c) is the probability of class.
- P(x|c) is the probability of predictor given class.
- P(x) is the prior probability of predictor.



$$P(c|X) = P(x_1|c) \times P(x_2|c) \times \cdots \times P(x_n|c) \times P(c)$$

Data Set Sample

We used training dataset from sentiment140.com. The dataset comes with prelabeled columns containing one of two binary targets. Zero(0) for Negative sentiment and Four(4) for Positive sentiment. The data is made up of about 1.6 million random tweets with corresponding sentiment labels.

Out[24]:

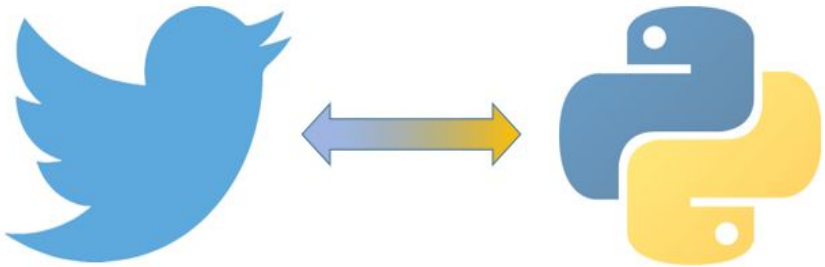
	label	text
0	0	is so sad for my APL friend.....
1	0	I missed the New Moon trailer...
2	1	omg its already 7:30 :O
3	0	.. Omgaga. Im sooo im gunna CRy. I've been at this dentist since 11.. I was suposed 2 just get a crown put on (30mins)...
4	0	i think mi bf is cheating on me!!! T_T
5	0	or i just worry too much?
6	1	Juuuuuuuuuuuuuuuuussssst Chillin!!
7	0	Sunny Again Work Tomorrow :-| TV Tonight
8	1	handed in my uniform today . i miss you already
9	1	hmmmm.... i wonder how she my number @-)

Twitter API

Tweets

- To get the most effective influence of tweets, we were in need to get historical tweets and up-to-date tweets.

- To work with twitter API to get up-to-date data we install some libraries like **tweepy**.



Training Data Preprocessing

Pre-processing steps:

- Convert @username to AT_USER.
- HTML special entities (e.g. & amp;).
- Make all the text in lower case.
- Remove:
    - hyperlinks.
    - Remove hashtags.
    - Remove Punctuation and split 's, 't, 've with a space for filter.
    - Remove words with 2 or fewer letters.
    - Remove whitespace (including new line characters).
    - Remove single space remaining at the front of the.
    - tickers.

Feature Extraction

We will transform data into tokens and then tokens into features. This is done by Term Frequency, Inverse Document Frequency (TF-IDF).

We'll convert each message which is represented by a list of tokens into a vector that a machine learning model can understand.

Feature Extraction steps:

1. Count how many times does a word occur in each message (Known as term frequency TF which measures how frequently a term occurs in a document)

2. Weigh the counts, so that frequent tokens get lower weight (inverse document frequency IDF which measures how important a term is)

- TF(t) = (Number of times term appears in a document) / (Total number of terms in the document).

- IDF(t) = loge (Total number of documents / Number of documents with term t in it)

Naïve Bayes Training

- The recommended method for training a good model is to first cross-validate using a portion of the training set itself to check if you have used a model with too much capacity (i.e. if the model is overfitting the data).
- We use the KFold strategy to split the model into random parts to find out if it's generalizing well or if it's overfitting.
- For our case, there are 4 + 2 + 2=8 parameter combinations to test and 10 KFold validations, so the model will be trained and tested on the validation set 8 x 10 = 80 times.

```
Fitting 10 folds for each of 8 candidates, totalling 80 fits
[Parallel(n_jobs=1)]: Done  80 out of  80 | elapsed: 73.0min finished

Best Model: 0.752827 using {'bow__ngram_range': (1, 1), 'classifier__alpha': 0.01, 'tfidf__use_idf': False}

Mean: 0.740987 Stdev:(0.000974) with: {'bow__ngram_range': (1, 1), 'classifier__alpha': 0.01, 'tfidf__use_idf': True}
Mean: 0.752827 Stdev:(0.000977) with: {'bow__ngram_range': (1, 1), 'classifier__alpha': 0.01, 'tfidf__use_idf': False}
Mean: 0.740697 Stdev:(0.000966) with: {'bow__ngram_range': (1, 1), 'classifier__alpha': 0.001, 'tfidf__use_idf': True}
Mean: 0.750797 Stdev:(0.001050) with: {'bow__ngram_range': (1, 1), 'classifier__alpha': 0.001, 'tfidf__use_idf': False}
Mean: 0.728533 Stdev:(0.000813) with: {'bow__ngram_range': (1, 2), 'classifier__alpha': 0.01, 'tfidf__use_idf': True}
Mean: 0.745505 Stdev:(0.000643) with: {'bow__ngram_range': (1, 2), 'classifier__alpha': 0.01, 'tfidf__use_idf': False}
Mean: 0.719832 Stdev:(0.000970) with: {'bow__ngram_range': (1, 2), 'classifier__alpha': 0.001, 'tfidf__use_idf': True}
Mean: 0.732427 Stdev:(0.000923) with: {'bow__ngram_range': (1, 2), 'classifier__alpha': 0.001, 'tfidf__use_idf': False}

Out[94]: ['twitter_sentiment.pkl']
```

Naïve Bayes score

- **Confusion matrix**
- **Precision**
- **Recall**
- **F1 -score**

```
accuracy score:  0.7531167007235139


confusion matrix:
 [[116466  35394]
 [ 38789 109829]]


              precision    recall  f1-score   support

           0       0.75      0.77      0.76    151860
           4       0.76      0.74      0.75    148618

avg / total       0.75      0.75      0.75    300478
```

Phase 2

Sentimental Analysis

Nature Language Processing

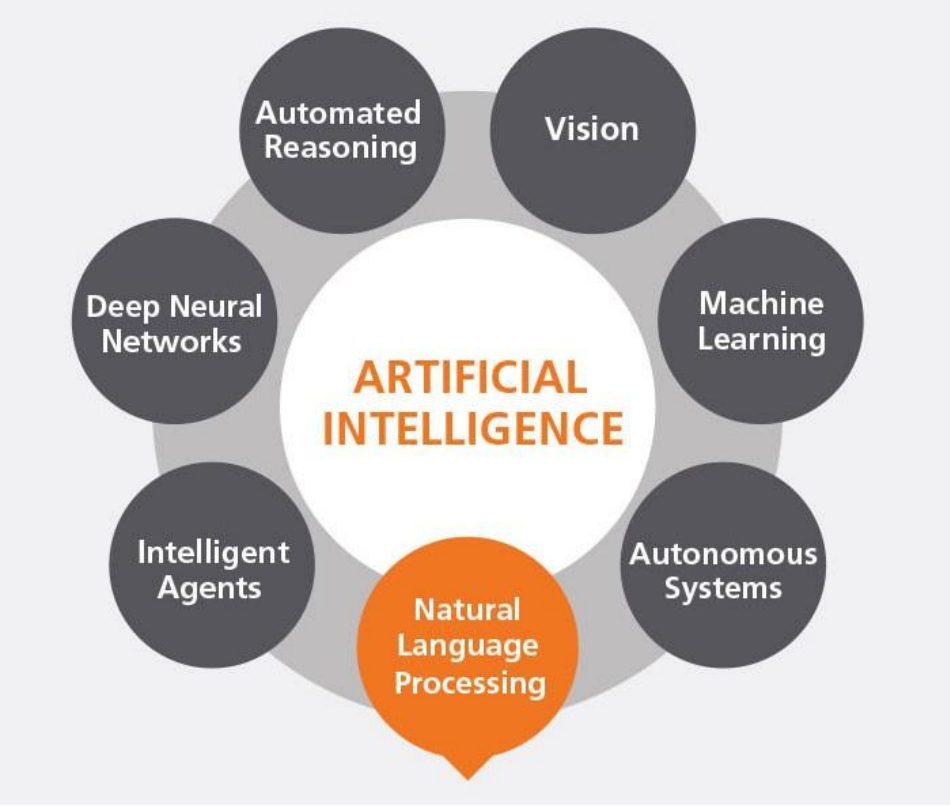
What is deep learning ?

Deep Learning is :

- subset of machine learning
- deep learning was inspired by artificial neural networks, that composed of three stages :
    - *input layer .
    - *hidden layer.
    - *output layer .

NLP

Natural Language Processing (NLP) is the field of Artificial Intelligence concerned with the processing and understanding of human language.
Since its inception during the 1950s, machine understanding of language has played a pivotal role in translation, topic modeling, document indexing, information retrieval, and extraction.

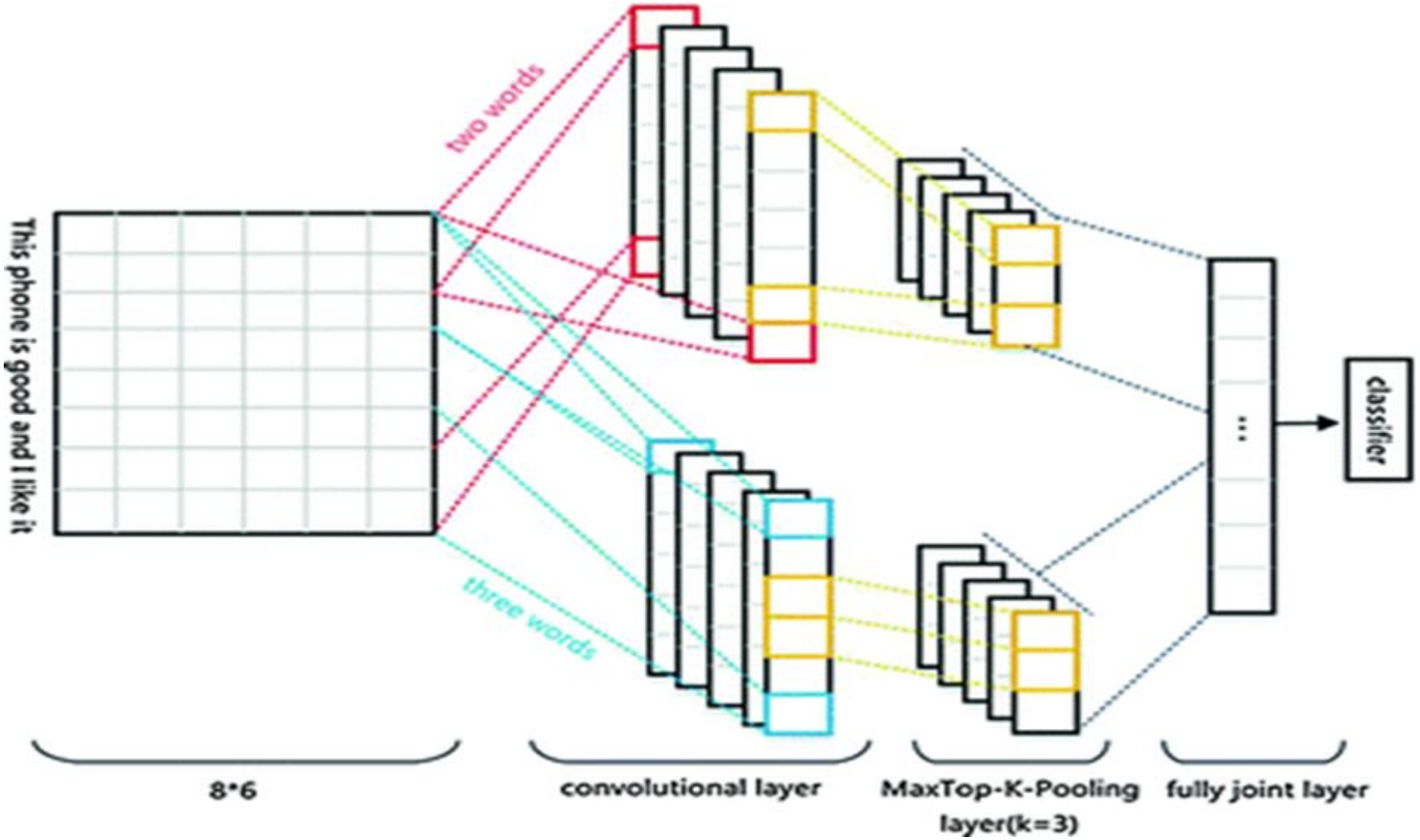


NLP using Deep learning

CNN can extract an area of features from global information, and it is able to consider the relationship among these features. The above solution can achieve a higher accuracy in analysis and classification. For natural language processing, texts data features also can be extracted piece by piece and to consider the relationship among these features, but without the consideration of context or whole sentence, the sentiment might be understood wrong.

CNN has a convolutional layer to extract information by a larger piece of text, so we work for sentiment analysis with convolutional neural network

Model Architecture



Model Architecture

Layer (type)	Output Shape	Param #
embedding_15 (Embedding)	(None, 49, 100)	2000000
conv1d_43 (Conv1D)	(None, 49, 300)	90300
max_pooling1d_43 (MaxPooling	(None, 16, 300)	0
flatten_12 (Flatten)	(None, 4800)	0
dense_27 (Dense)	(None, 100)	480100
dense_28 (Dense)	(None, 1)	101

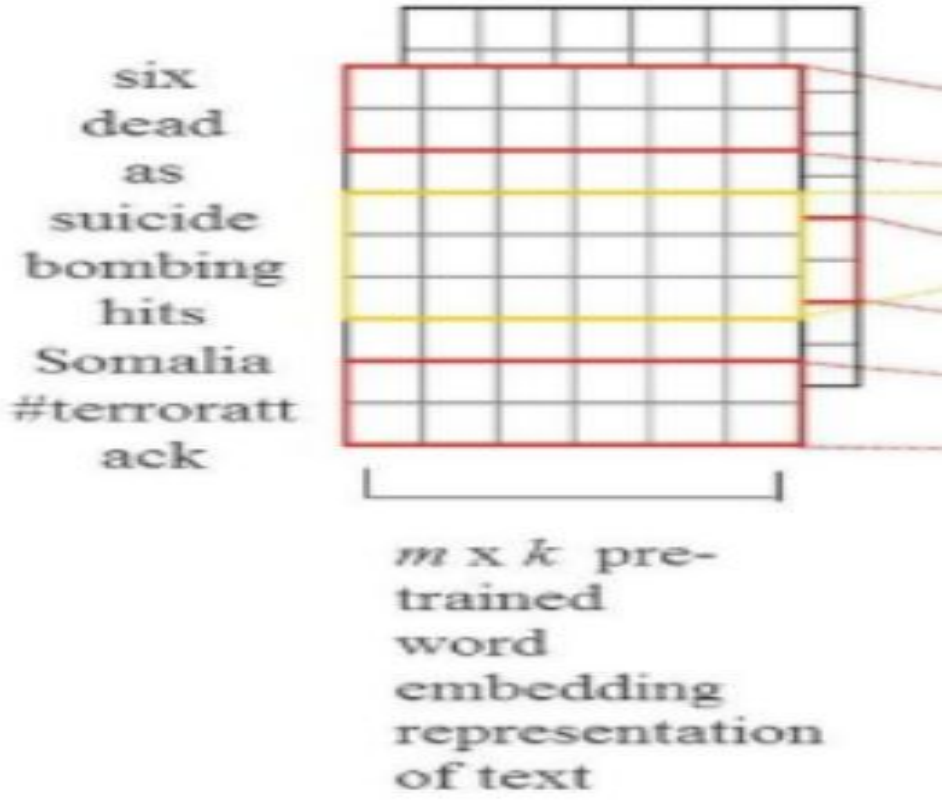
Total params: 2,570,501
Trainable params: 570,501
Non-trainable params: 2,000,000

Model Layer description:

1- Embedding Layer:

we can think of the Embedding layer as a dictionary
that maps an index assigned to a word to a word vector.
This layer is very flexible and can be used in a few ways:
a-The embedding layer can be used at the start of
a larger deep learning model.

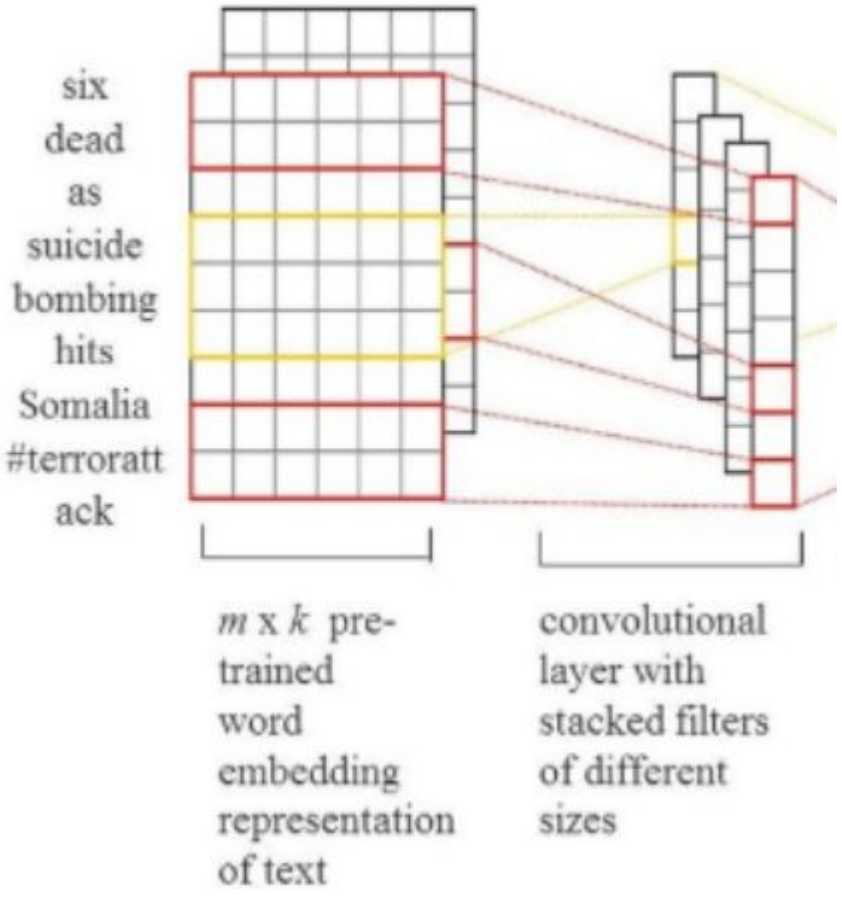
b-Also, we could load pre-train word embeddings into
the embedding layer when we create our model.



Model Layer description:

2- Convolution layer:

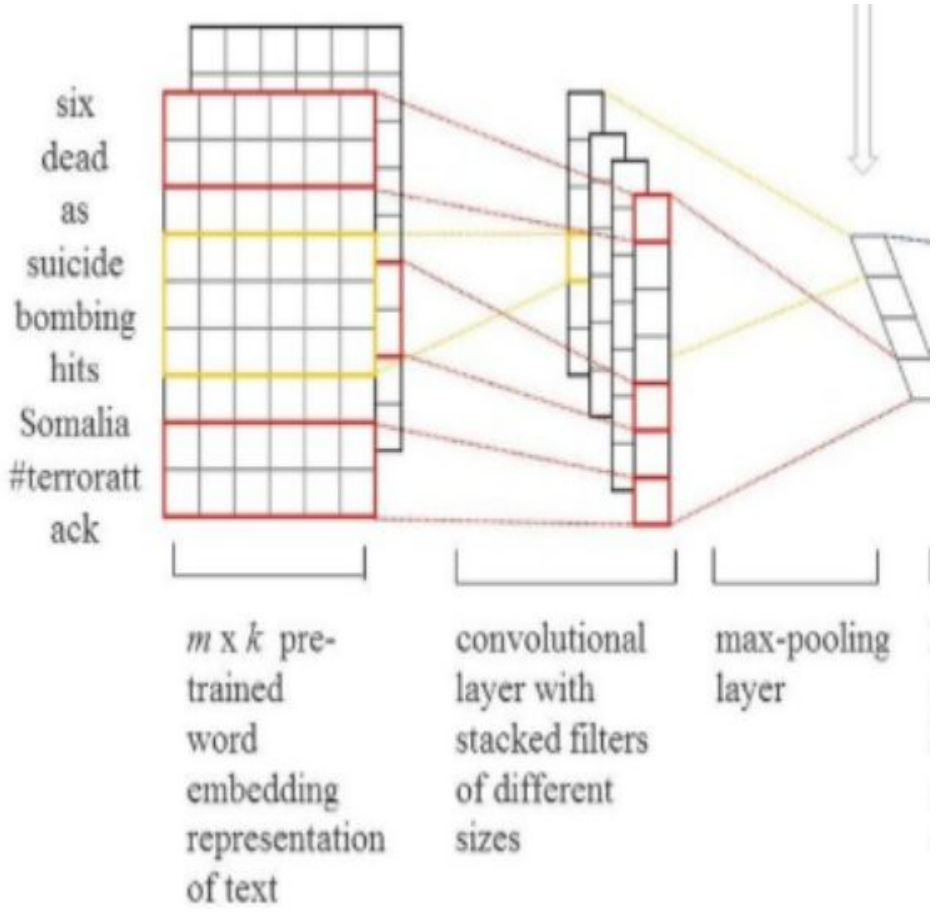
After we come out of embedding layer, we want to get feature descriptor for whole of text, so we use neural network over this embedding. To analyze n-grams in our text, we just convolve the word vectors that are near, you can think of it as sliding window walk on our word vector matrix. You need to track many n-grams, that's why you need a lot of convolutional filters. And these filters are called 1D convolutions because we slide the window only in one direction.



Model Layer description:

3- Maxpooling layer:

is a sample-based discretization process. The objective is to down-sample an input representation (image, hidden-**layer** output matrix, etc.), reducing its dimensionality , we do maximum pooling overtime and, we apply multiple filter with different length and finally we got vector is a kind of embedding of our input sequence.



Model Layer description:

4- Flatten layer:
Does not affect the batch size. The purpose of this argument is to preserve weight ordering when switching a model from one data format to another

5- Dense layer :
Two dense layer are used in our architecture one with relu activation function , another one with sigmoid

6- dropout layer:

This layer prevent overfitting .

Data Cleaning

Text preprocessing is technique that transforms raw data into an understandable format. Raw data(real world data) is always incomplete and that data cannot be sent through a model. That would cause certain errors. That is why we need to preprocess data before sending through a model.

Preprocessing steps:
1- lowering words.
2- remove punctuation.
3- remove hyperlinks.
4-remove unused fields.

Feature Extraction

We will transform tokens(words) into features. This is done by word embedding. Word Embedding is the texts converted into numbers and there may be different numerical representations of the same text. many Machine Learning algorithms and almost all Deep Learning Architectures are incapable of processing *strings* or *plain text* in their raw form They require numbers as inputs to perform any sort of job, be it classification, regression etc. A Word Embedding format generally tries to map a word using a dictionary to a vector.

Feature Extraction steps:

1- Tokenize sentence into words or tokens. Each word take an integer id . Start indexing from1.
2- Convert sentence into vector of numbers by replacing each word by its integer id .(To_sequence_function)
3- padding sequence mean when there are multiple sentence with different length ,we are in need to make them equal in length vector.

```
array([  22,  328,   33,  287,  140,   19,  629,    5, 1456,    0,    0,
            0,    0,    0,    0,    0,    0,    0,    0,    0,    0,    0,
            0,    0,    0,    0,    0,    0,    0,    0,    0,    0,    0,
            0,    0,    0,    0,    0,    0,    0,    0,    0,    0,    0,
            0,    0,    0,    0,    0])
```

Model Training

```
_________________________________________________________________
Train on 170000 samples, validate on 29999 samples
Epoch 1/10
170000/170000 [==============================] - 236s 1ms/step - loss: 0.5489 - acc: 0.7128 - val_loss: 0.5064 - val_acc: 0.751
1
Epoch 2/10
170000/170000 [==============================] - 248s 1ms/step - loss: 0.4768 - acc: 0.7686 - val_loss: 0.5204 - val_acc: 0.740
1
Epoch 3/10
170000/170000 [==============================] - 227s 1ms/step - loss: 0.4453 - acc: 0.7882 - val_loss: 0.5009 - val_acc: 0.753
8
Epoch 4/10
170000/170000 [==============================] - 227s 1ms/step - loss: 0.4233 - acc: 0.8015 - val_loss: 0.4651 - val_acc: 0.780
1
Epoch 5/10
170000/170000 [==============================] - 239s 1ms/step - loss: 0.4031 - acc: 0.8139 - val_loss: 0.4719 - val_acc: 0.772
4
Epoch 6/10
170000/170000 [==============================] - 232s 1ms/step - loss: 0.3852 - acc: 0.8231 - val_loss: 0.4575 - val_acc: 0.783
9
Epoch 7/10
170000/170000 [==============================] - 238s 1ms/step - loss: 0.3646 - acc: 0.8357 - val_loss: 0.4695 - val_acc: 0.783
6
Epoch 8/10
170000/170000 [==============================] - 247s 1ms/step - loss: 0.3509 - acc: 0.8427 - val_loss: 0.5311 - val_acc: 0.755
2
Epoch 9/10
170000/170000 [==============================] - 236s 1ms/step - loss: 0.3311 - acc: 0.8528 - val_loss: 0.4794 - val_acc: 0.780
6
Epoch 10/10
170000/170000 [==============================] - 223s 1ms/step - loss: 0.3154 - acc: 0.8616 - val_loss: 0.4963 - val_acc: 0.782
2
29999/29999 [==============================] - 19s 642us/step
test_loss: 0.50%
 test_acc: 78.22%
```

Comparison

	sentiment analysis (with machine learning)	**Nlp using deep learning**
Data dependency	Machine learning algorithms often work well even if the dataset is small	Deep learning is *Data Hungry* the more data you have, the better it is likely to perform
computation power	Traditional machine learning algorithm can be implemented on a CPU	For the deep learning network training, you need a graphical processing unit (GPU) which have thousands of cores compared to a CPU that has very minimal cores.
Time consuming	Whereas traditional machine learning algorithms often train very fast ranging from few minutes to a couple of hours, but during the test time, some algorithms can also take quite a bit of time.	The training time of a deep learning network can range from anywhere between a few hours to months

Machine Learning For Time Series Forecasting

Phase1
Historical Data

Phase2
Social media Data(Twitter)

Phase3
Historical + Twitter Data

Phase 3

Sell or buy model

Polynomial model

CNN model

1- Sell/Buy Model

- We extracted tweets from API, processed them for Natural Language Processing (NLP) and sentiment analysis.
- After predicting sentiment, we extracted historical data from Yahoo Finance.
- We then developed a model for stock market prediction using stock price data and sentiment score to predict the change in stock market.
- The proposed methodology for predicting the stock market movement through sentiment analysis showed in the following figure:

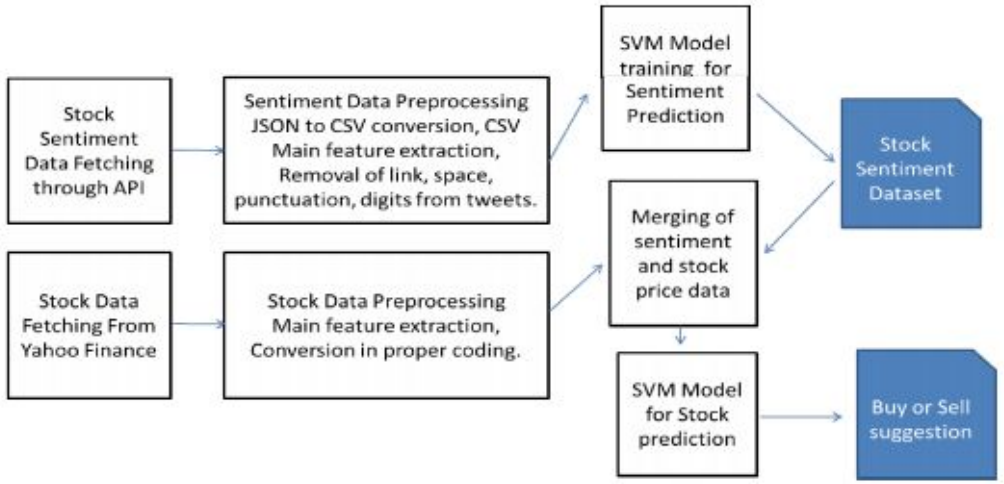
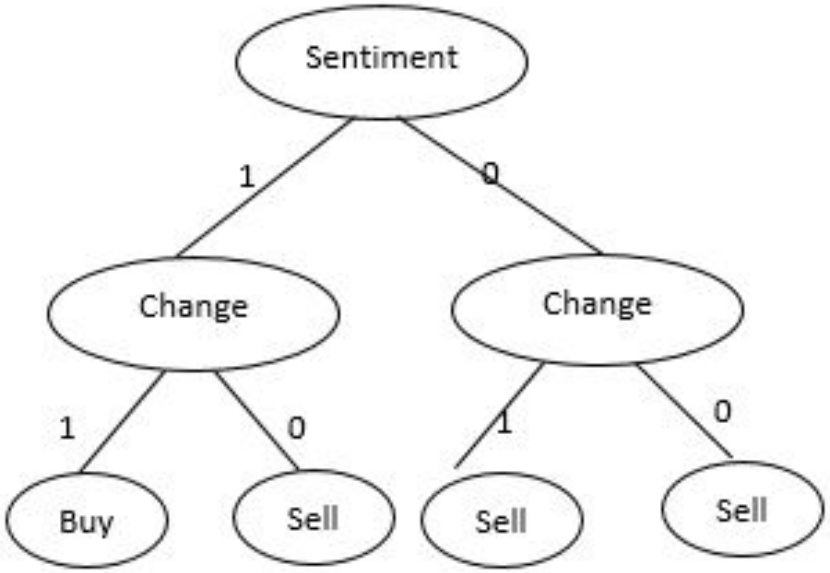


Figure 2: Flowchart of proposed Methodology

SVC Model

We used SVM model to suggest whether a person should buy or sell a share. We applied some preprocessing on the data as we created new columns to present the class if it is 1 or 0 (buy or sell) so we can train the model. We achieved 98% accuracy



2- Polynomial Regression Model

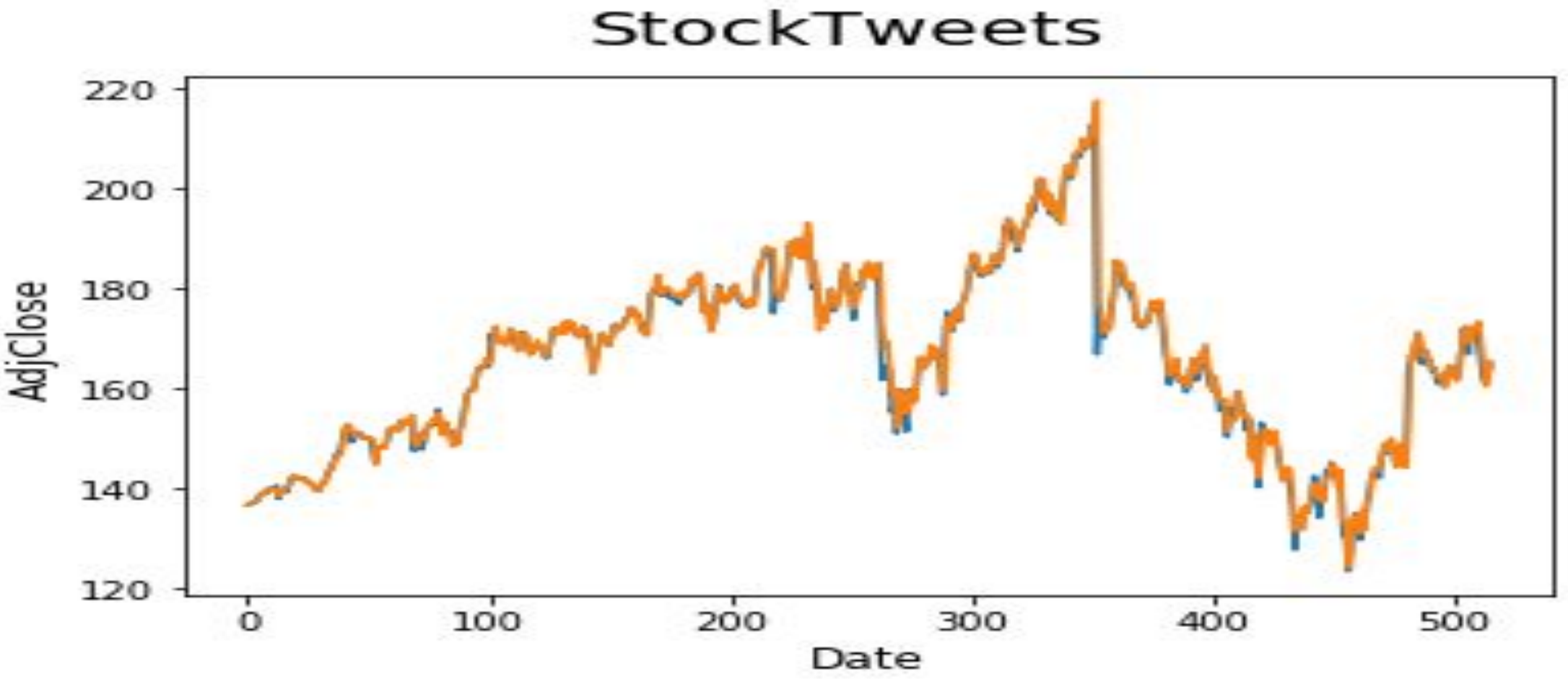
In statistics, polynomial regression is a form of regression analysis in which the relationship between the independent variable *x* and the dependent variable *y* is modelled as an *n*th degree polynomial in *x*. Polynomial regression fits a nonlinear relationship between the value of *x* and the corresponding conditional mean of y, denoted E(*y* | *x*). Polynomial regression is considered to be a special case of multiple linear regression.

This is the general equation of a polynomial regression is:

$$\mathbf{Y=\theta o + \theta_1 X + \theta_2 X^2 + \ldots + \theta \square X^m + residual\ error}$$

We developed a polynomial regression model to enhance our daily prediction using the score we get from NLP analysis of tweets and customer reviews. We used a second-degree model to fit our data. We used 70% of the data for training the model and 30% for the evaluation.

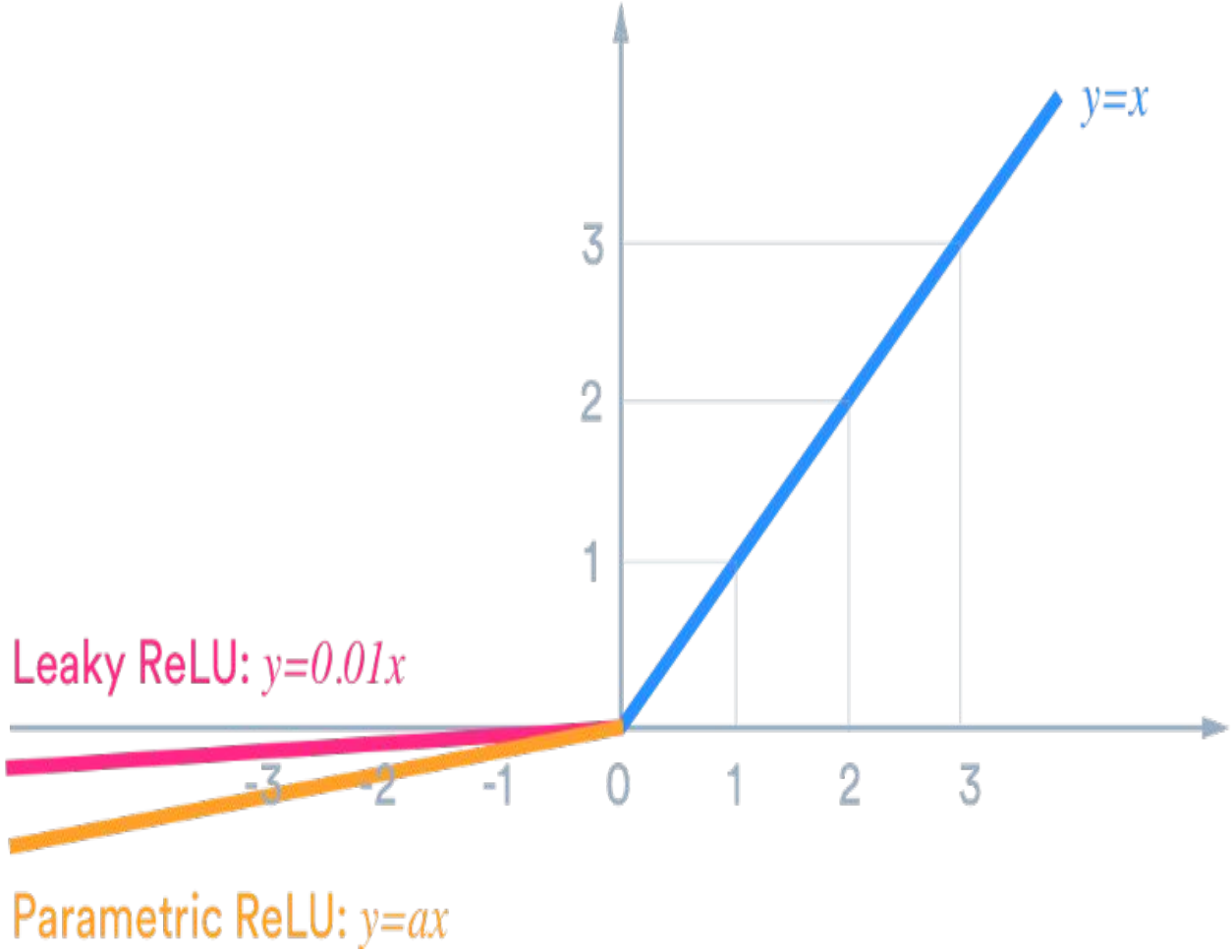
2- Polynomial Regression Validation



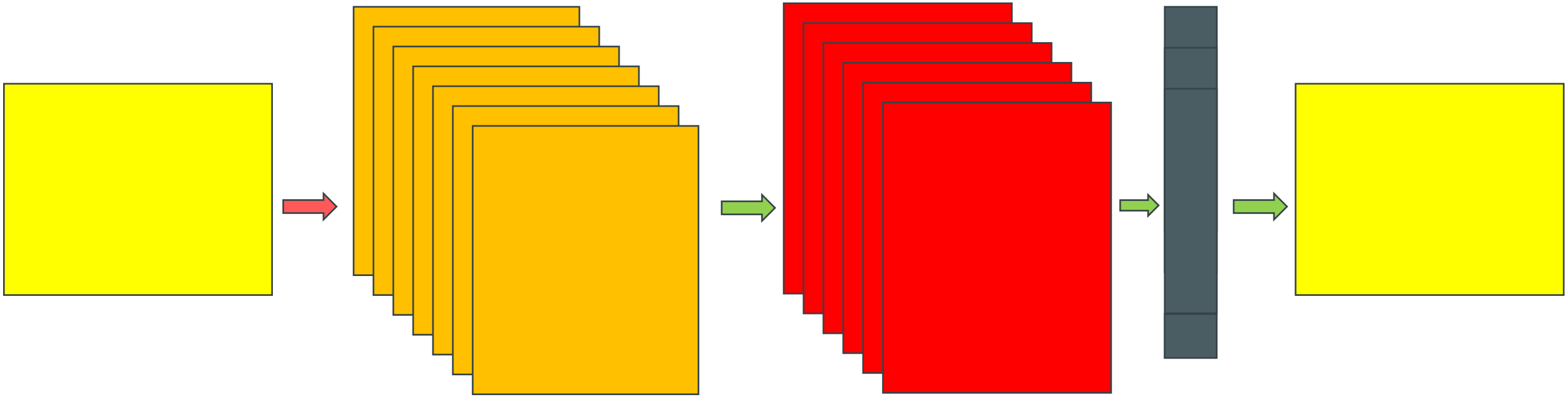
3- CNN Model

We use 3 hidden layers (Conv2D, MaxPooling2D and Dense). And input of size (1,7,7). We use RELU function for all of them.

- **Conv2D**
  Conv2D of 64 neurons.
- **MaxPooling2D**
  Pool size = (1, 2).
- **Dense**
  Dense of 30 neurons



3- CNN Model Architecture



Input Layer
49 neurons

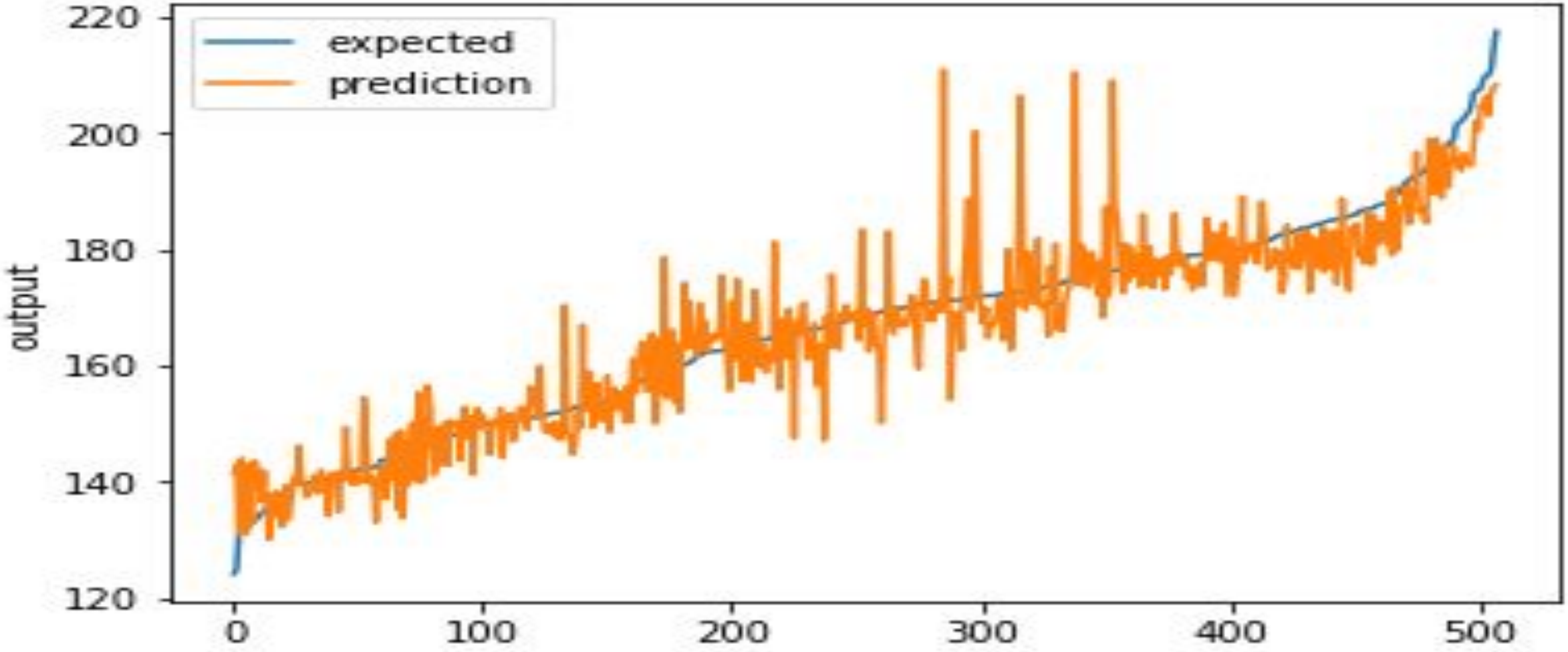
Conv2D
7*7*64

MaxPooling2D
1*2*64

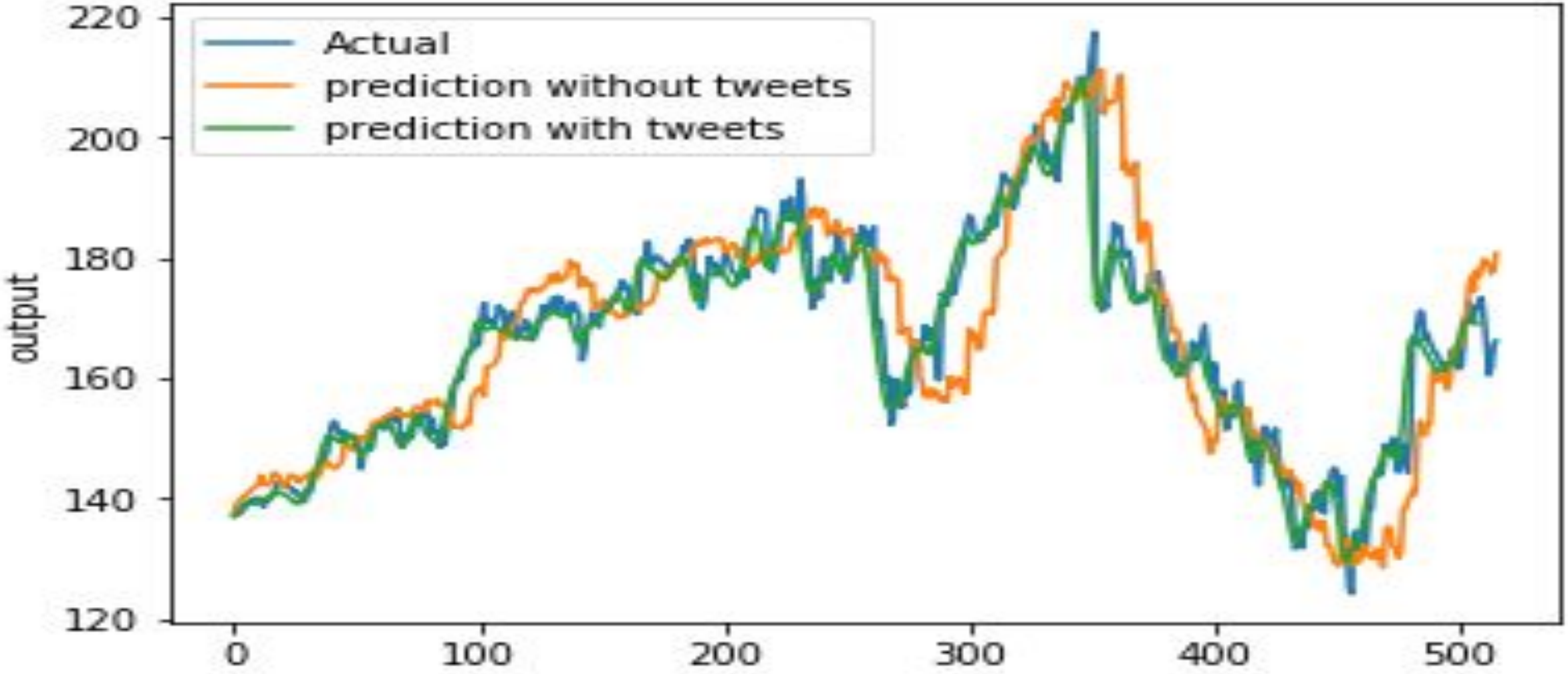
Dense
7*7*32

Output Layer

3- CNN Model Architecture



3- CNN Model Validation



Evaluation

1. Sell/Buy Model

Accuracy = $\frac{TP+TN}{TP+TN+FP+FN}$

We achieved 98.8% accuracy for the SVC model we used to merge the customer reviews with the stock predictions.

Actual / predicted	True	False
True	35	1
False	0	63

Evaluation

Model	MSE	RMSE	EVS	R2
CNN	4.567077	6.52544	0.86388	0.86266
Polynomial	8.4125	2.9	0.9731	0.9724
FBprophet	6.6753	9.219508	0.73108	0.73107

Outlines

01 Introduction

02 System Components And Architecture

03 Front End Development

04 Back End Development

05 Machine Learning For Time Series Forecasting

06 Conclusions & Future Work

Conclusions

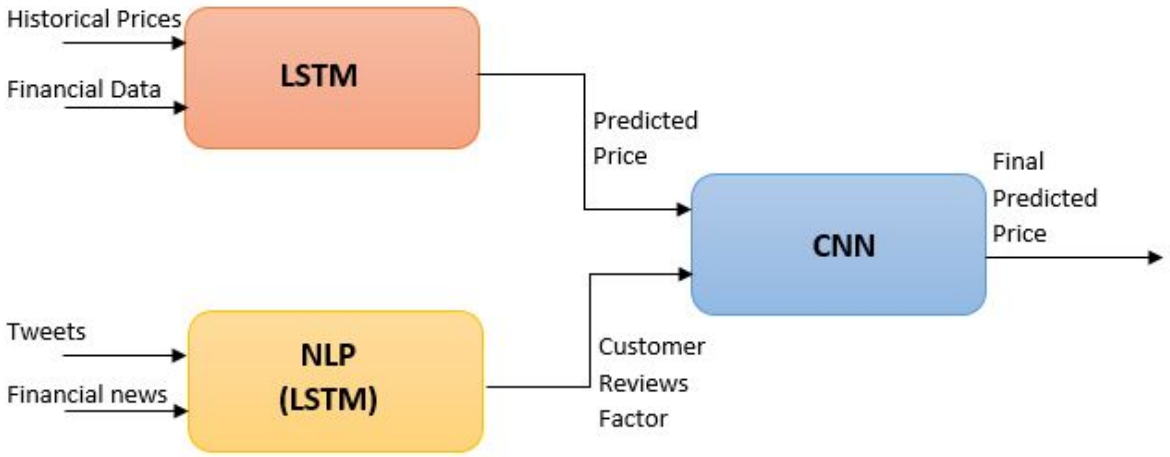
- We developed two models based on historical data for long-term and short-term predictions
- We used the customer reviews to make our predictions more accurate , So we developed a NLP model to analysis the customer review
- Finally , We developed three models to combine the analysis of customer reviews with our predictions so we can get better predictions and more accurate system with higher performance

Future Work

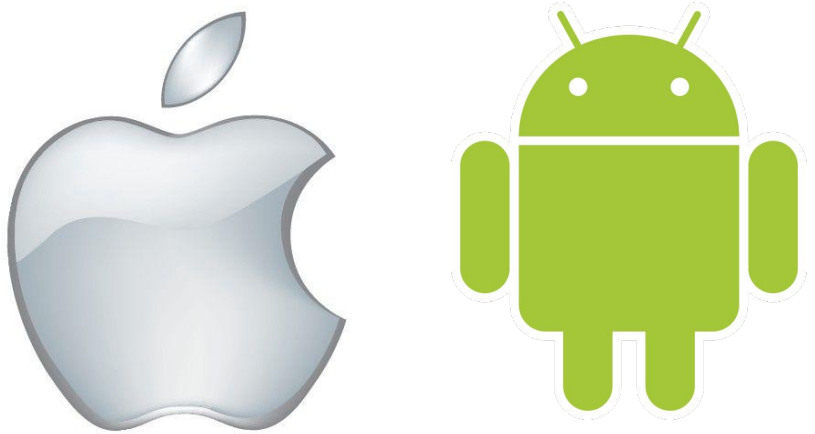
- Data

FINANCIAL NEWS + +

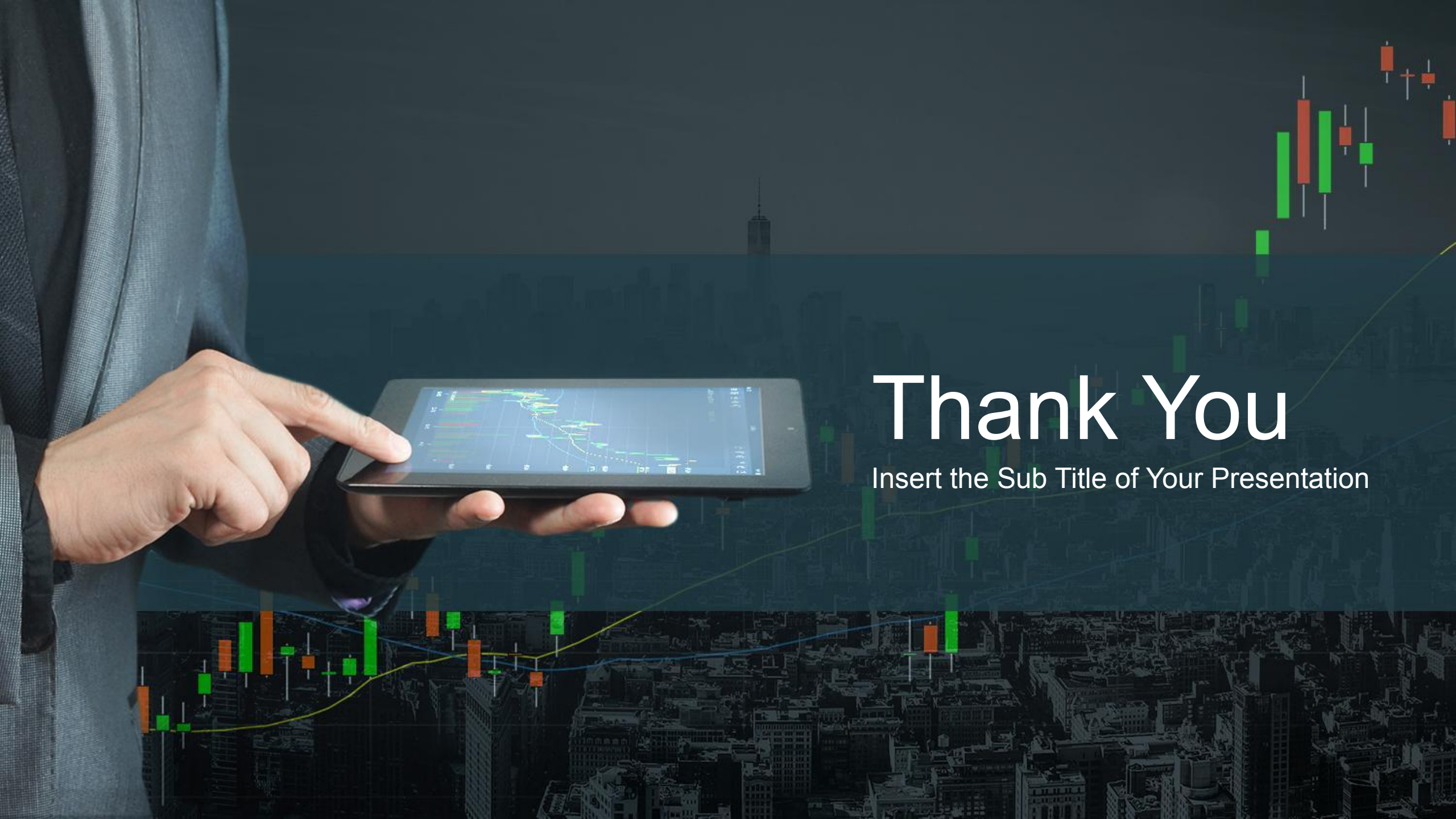
- Prediction



- Application



Demo



Thank You

Insert the Sub Title of Your Presentation