Mid Point Project Report

Explorations in Data Science

Project Objective: In today's world where information is readily available and at your fingertips, we think and believe that the paradigm of how stocks rise and fall based on information has changed in a way where we can use data science to use available data to decide if the outcome will cause a positive or negative impact.

Project Approach: The Project is twitter sentiment analysis. Sentiment Analysis is the automated process of analyzing text data and sorting it into sentiments positive, negative, or neutral. Performing Sentiment Analysis on data from Twitter using machine learning can help companies understand how people are talking about their brand. Monitoring Twitter allows companies to understand their audience, keep on top of what's being said about their brand and their competitors, and discover new trends in the industry. Are users talking positively or negatively about a product? Well, that's exactly what sentiment analysis determines.

Our first step towards this project is to fetch the tweets from Kaggle which are pre-processed twitter data and then perform sentiment analysis on the same. Similarly, get the stock prices for the respective company for the dates we fetched the tweets. We considered the percentage change in stock values.

Team Structure

- Shashank Shekhar
- Likhitha vanga

We started this project by initially dividing our work into 2 halves, so that we can contribute equally to the project. Below are the individual duties:

Shashank:

- Prepared data for processing by deleting unwanted columns and rows.
- Selecting the algorithms and analyzing why the chosen is better than other.
- Implementing an algorithm.
- Documentation.

Likhitha:

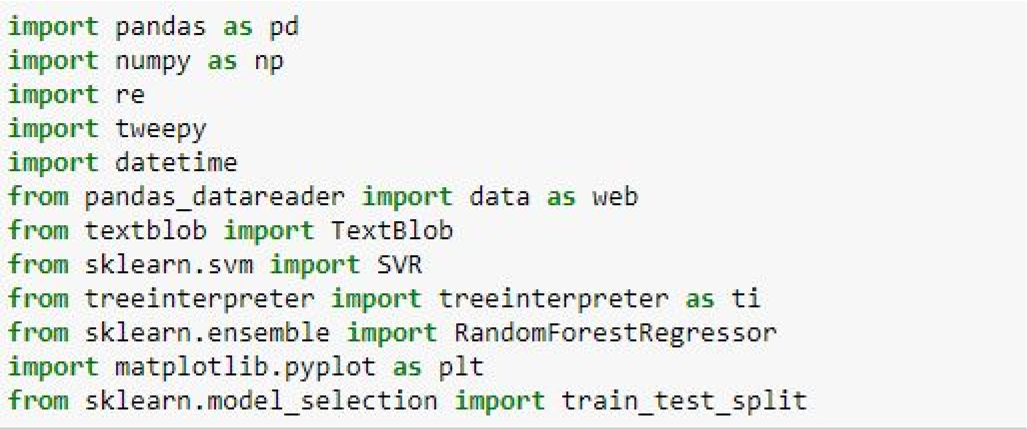
- Fetched data from Kaggle.

- Understand the algorithms we are using for this project.
- Implementing an algorithm.
- Documentation.

Project Milestones

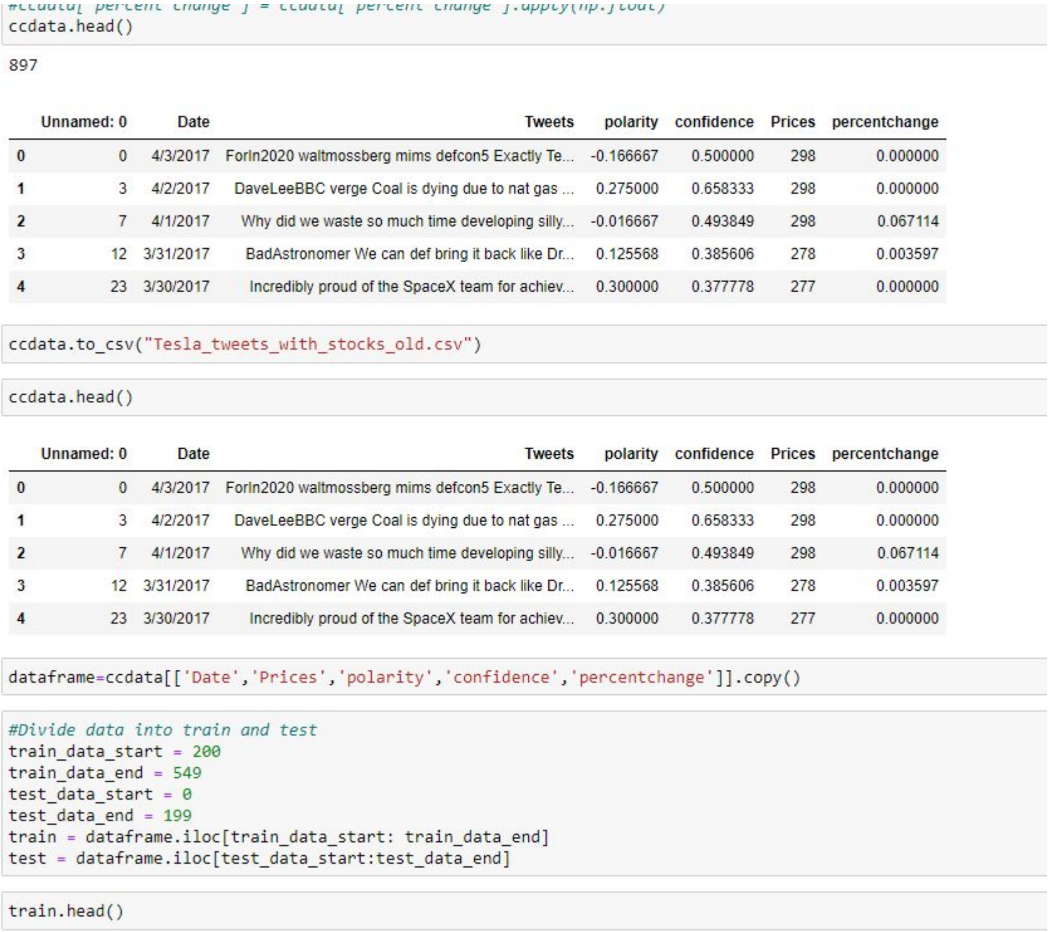
Accomplished:

- Started the project by a basic understanding about tweepy and understanding about algorithms.
- Made an outline of how our reports want to look like.
- Able to read the csv file and process the data using python.
- Trained data on random forest and linear regressor and analyzed the algorithms.
- We got the environment set up for the project.
- Imported necessary libraries.



```
import pandas as pd
import numpy as np
import re
import tweepy
import datetime
from pandas_datareader import data as web
from textblob import TextBlob
from sklearn.svm import SVR
from treeinterpreter import treeinterpreter as ti
from sklearn.ensemble import RandomForestRegressor
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
```

- Able to fetch and read data.



```
#ccdata['percent change'] = ccdata['percent change'].apply(np.float)
ccdata.head()
```

```
897
```

	Unnamed: 0	Date	Tweets	polarity	confidence	Prices	percentchange
0	0	4/3/2017	ForIn2020 waltmossberg mims defcon5 Exactly Te...	-0.166667	0.500000	298	0.000000
1	3	4/2/2017	DaveLeeBBC verge Coal is dying due to nat gas ...	0.275000	0.658333	298	0.000000
2	7	4/1/2017	Why did we waste so much time developing silly...	-0.016667	0.493849	298	0.067114
3	12	3/31/2017	BadAstronomer We can def bring it back like Dr...	0.125568	0.385606	278	0.003597
4	23	3/30/2017	Incredibly proud of the SpaceX team for achiev...	0.300000	0.377778	277	0.000000

```
ccdata.to_csv("Tesla_tweets_with_stocks_old.csv")
```

```
ccdata.head()
```

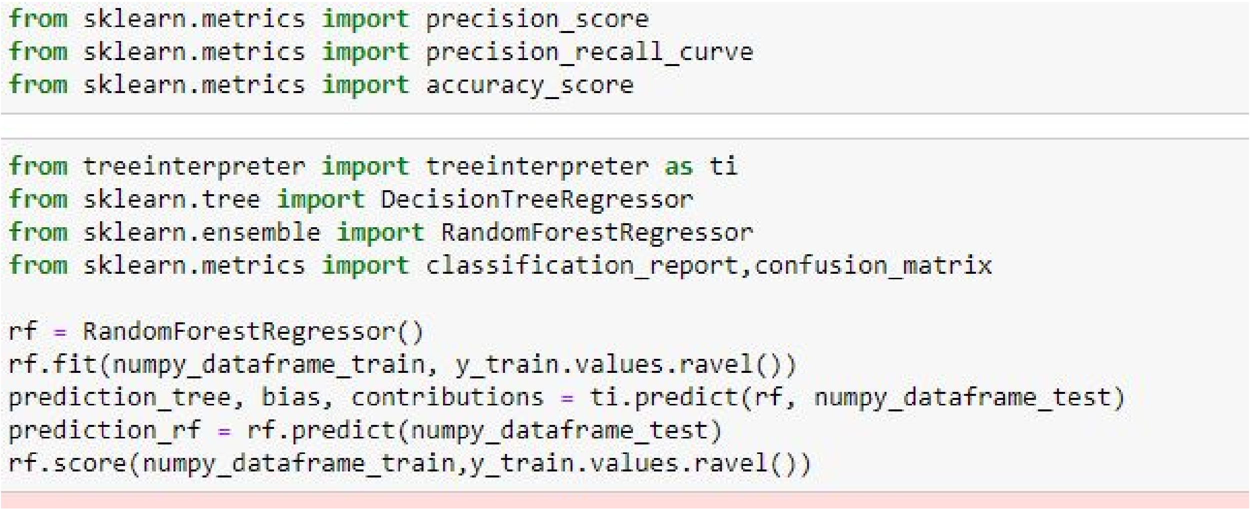
	Unnamed: 0	Date	Tweets	polarity	confidence	Prices	percentchange
0	0	4/3/2017	ForIn2020 waltmossberg mims defcon5 Exactly Te...	-0.166667	0.500000	298	0.000000
1	3	4/2/2017	DaveLeeBBC verge Coal is dying due to nat gas ...	0.275000	0.658333	298	0.000000
2	7	4/1/2017	Why did we waste so much time developing silly...	-0.016667	0.493849	298	0.067114
3	12	3/31/2017	BadAstronomer We can def bring it back like Dr...	0.125568	0.385606	278	0.003597
4	23	3/30/2017	Incredibly proud of the SpaceX team for achiev...	0.300000	0.377778	277	0.000000

```
dataframe=ccdata[['Date','Prices','polarity','confidence','percentchange']].copy()
```

```
#Divide data into train and test
train_data_start = 200
train_data_end = 549
test_data_start = 0
test_data_end = 199
train = dataframe.iloc[train_data_start: train_data_end]
test = dataframe.iloc[test_data_start:test_data_end]
```

```
train.head()
```

- Constructed the random forest algorithm with respective libraries.

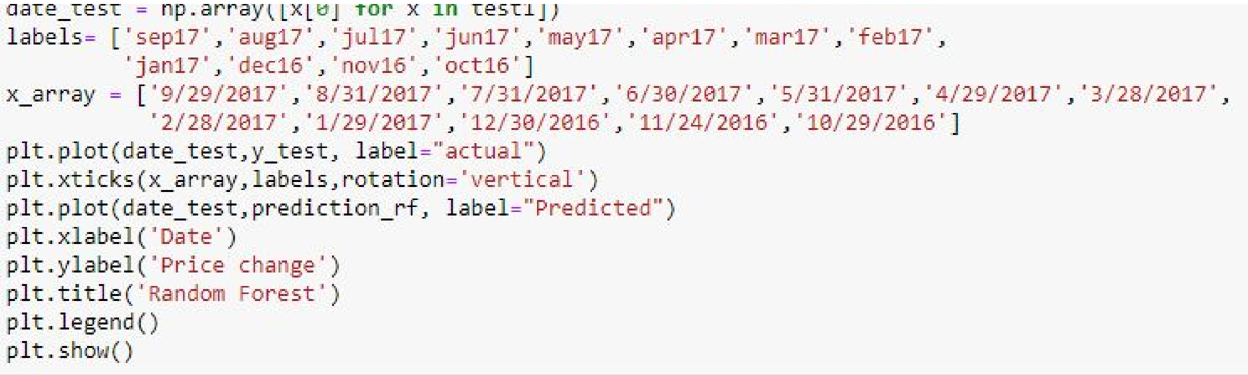


```
from sklearn.metrics import precision_score
from sklearn.metrics import precision_recall_curve
from sklearn.metrics import accuracy_score
```

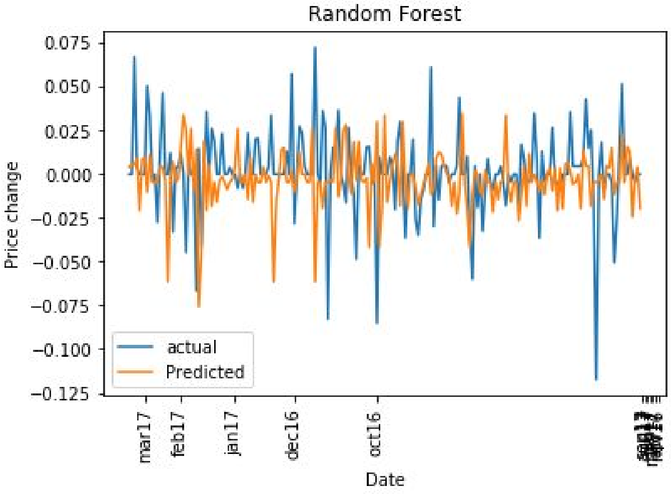
```
from treeinterpreter import treeinterpreter as ti
from sklearn.tree import DecisionTreeRegressor
from sklearn.ensemble import RandomForestRegressor
from sklearn.metrics import classification_report,confusion_matrix

rf = RandomForestRegressor()
rf.fit(numpy_dataframe_train, y_train.values.ravel())
prediction_tree, bias, contributions = ti.predict(rf, numpy_dataframe_test)
prediction_rf = rf.predict(numpy_dataframe_test)
rf.score(numpy_dataframe_train,y_train.values.ravel())
```

- Deciding on how our reports and graphs should look like.



```
date_test = np.array([x[0] for x in test1])
labels= ['sep17','aug17','jul17','jun17','may17','apr17','mar17','feb17',
         'jan17','dec16','nov16','oct16']
x_array = ['9/29/2017','8/31/2017','7/31/2017','6/30/2017','5/31/2017','4/29/2017','3/28/2017',
           '2/28/2017','1/29/2017','12/30/2016','11/24/2016','10/29/2016']
plt.plot(date_test,y_test, label="actual")
plt.xticks(x_array,labels,rotation='vertical')
plt.plot(date_test,prediction_rf, label="Predicted")
plt.xlabel('Date')
plt.ylabel('Price change')
plt.title('Random Forest')
plt.legend()
plt.show()
```



To Be Accomplished:

- Have to consider the factors which algorithm is better.
- Testing the scalability of data.
- Being able to create a csv file with tweepy without doing it manually.

Scheduled time of Meeting: Thursday, 1:20 PM(July 23,2020)