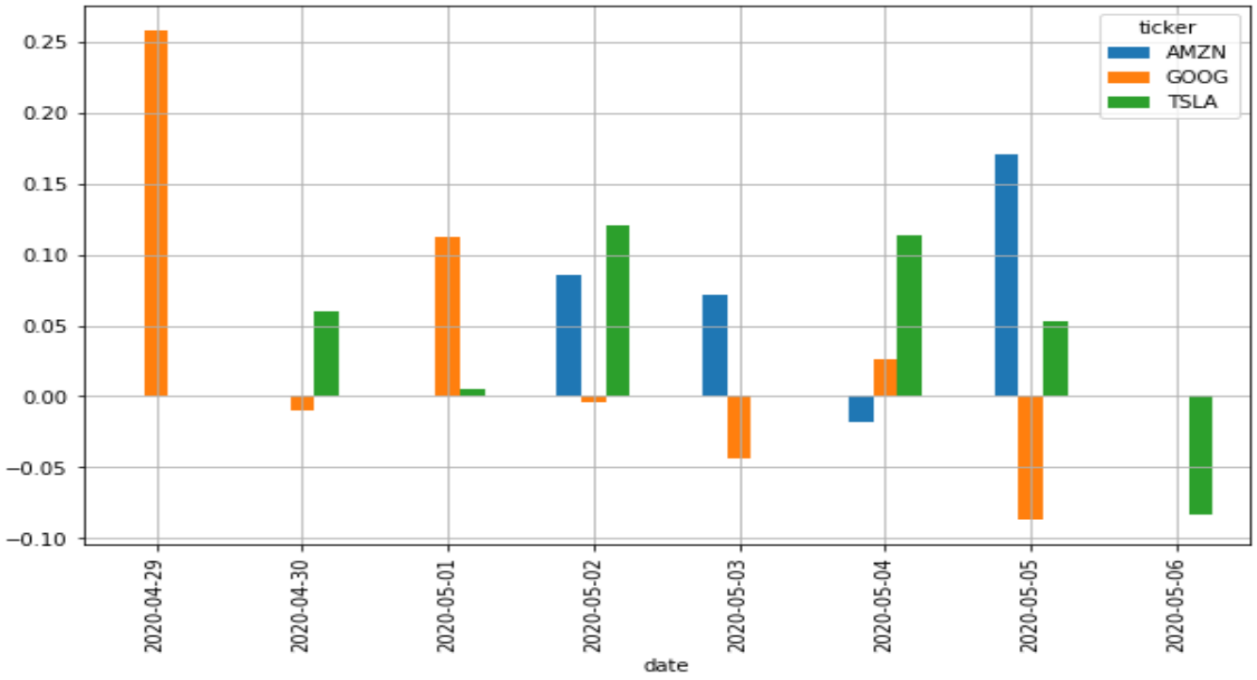
STOCK-SENTIMENT ANALYSIS

Hariom paramr(2017kucp1060)

CSE,Final Year

Sentiment analysis combines the understanding of semantics and symbolic representations of language. The algorithm will learn from labeled data and predict the label of new/unseen data points. This approach is called supervised learning, as we train our model with a corpus of labeled news.



METHODOLOGY:

- All required libraries include

```
## import all required libraries
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
from sklearn.metrics import classification_report,confusion_matrix,accuracy_score
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.svm import SVC
```

- Then I include dataset file data.csv
- File will contain date,news and target variable **'label'**

```
## include the dataset
dataset = pd.read_csv("Data.csv")
dataset.shape
```

```
(4101, 27)
```

```
dataset.head()
```

- Then split the dataset into tranning and testing phase on the basis of **date.**
- Tranning data train the model and model perform on testing data

```
#split the train and test data according to date
#from sklearn.model_selection import train_test_split
#train,test = train_test_split(dataset,test_size=0.1,random_state=0)
train = dataset[dataset['Date'] < '20150101']
test = dataset[dataset['Date'] > '20141231']
train.shape
```

- After that clean the dataset (remove everything apart letter from a-z or A-Z).
- Change the column name and all words change to lower words for **bow** model

```
# cleaning the data
data=train.iloc[:,2:27]
data.replace("[^a-zA-Z]"," ",regex=True, inplace=True)
# change the column names
list1= [i for i in range(25)]
new_Index=[str(i) for i in list1]
data.columns= new_Index
data.head()
# Convertng headlines to lower case
for index in new_Index:
    data[index]=data[index].str.lower()
data.head()
```

- Combine all the news headlines in one paragraph for indivisual date.

```
headlines = []
for row in range(0,len(data.index)):
    headlines.append(' '.join(str(x) for x in data.iloc[row,0:25]))
```

```
headlines[1]
```

- Then implement bag of words model using countvectorizer and put all **unique** words

In the bag. N_gram range represent what type of ngram used in the bag of words model.

- We use **SVM** classifier to classify the data into two category. **Label-0**

  Describe decrasing of stock price and **Label-1** describe increasing of stock price.

```
## implement BAG OF WORDS
countvector=CountVectorizer(ngram_range=(2,2))
traindataset=countvector.fit_transform(headlines)
# implement SVM Classifier
svclassifier = SVC(kernel='linear')
svclassifier.fit(traindataset,train['Label'])
```

- Predict the test dataset According to the trained model by the use of bow and svm classifier into two classes.

```
##Predict for the Test Dataset
test_transform= []
for row in range(0,len(test.index)):
    test_transform.append(' '.join(str(x) for x in test.iloc[row,2:27]))
test_dataset = countvector.transform(test_transform)
predictions = svclassifier.predict(test_dataset)
print(predictions)
```

- Evaluation of model
- After creation of model we find the peformance of model Ex. Accuracy,F1_measure and report.Accuracy of this model is 86%.

```
#find accuracy of model
matrix=confusion_matrix(test['Label'],predictions)
print(matrix)
score=accuracy_score(test['Label'],predictions)
print(score)
report=classification_report(test['Label'],predictions)
print(report)
```

Instructions to Run:

- Some libraries such as numpy , sklearn etc. Required for run the stock_sentiment.ipynb file.
- Open the file in jupyter notebook
- After Run all the code step by step and find output.
- For more details pls refer to https://github.com/harry-v528/Stock_sentiment-analysis.