International Research Journal of Engineering and Technology (IRJET) e-ISSN: 2395-0056
Volume: 07 Issue: 05 | May 2020 www.irjet.net p-ISSN: 2395-0072

STOCK MOVEMENT PREDICTION USING DAILY BROADCAST TRENDS

M. Dhana Raju[1], Dr.K.Kranthi Kumar[2], Ruhi Sania[3], V. Shiva Prasad[4], D.V.S.Mythili[5],

Ch. Sushmitha[6]

[1,]*Assistant Professor, Department of Information Technology, Sreenidhi Institute of Science and Technology, Telangana, India*

[2]*Associate Professor, Department of Information Technology, Sreenidhi Institute of Science and Technology, Telangana, India*

[3,4,,5,6]*B. Tech Student, Department of Information Technology, Sreenidhi Institute of Science and Technology, Telangana, India*

Abstract - *Stock market or Share market is one of the most complicated and sophisticated way to do business. Small ownerships, brokerage corporations, banking sector, all depend on this very body to make revenue and divide risks, a very complicated model. This is project is about predicting stock movement based on the daily trends published on a particular day. We shall see how this simple implementation will bring acceptable results. The DJIA stock dataset is collected from a Kaggle which are voted by reddit users and top 25 trends are taken based on the votes casted by the users. Sentiment Analysis is used to analyze the polarity of the sentence and Various techniques of Natural language processing and machine learning are used to predict the model. Scikit learn library is used for feature extraction and it also provides supervised learning algorithms to match output.*

Key Words: **Logistic regression, SGD, Random Forest, AdaBoost, KNN Algorithm.**

1.INTRODUCTION

Stock market is one of the oldest methods where an ordinary person can trade stocks, make deposits, and gain some money on this platform from businesses that sell a portion of themselves. But the prices and profitability of this network are unpredictable and that's where we need technologies to help us out. Machine learning is one of those resources that help us to accomplish what we want. Machine learning is one of those resources that help us to accomplish what we want. Machine learning is one of the hottest study subjects in computation and engineering that is relevant in several disciplines. It offers a range of algorithms, methods and techniques that can integrate any type of intelligence into machines. The strength of ML is the cognitive tools accessible that can be utilized, and can be learned in a learning process, through data collection representing a particular question and reacting to related unknown data in a different way. Machine learning has played a significant role in the identification of photos, reorganization, natural order of expression, drug suggestion and medical diagnosis over the past years. The new framework for machine learning allows us to enhance disaster alerts, public safety and make medical advances. The machine learning platform also promises better customer support and more stable networks for vehicles.

1.1 Logistic regression

This is a supervised method for learning classification which is used to estimate the probability of a goal variable. The essence of the explanatory or dependent variable is dichotomous, meaning there can be just two types. The dependent variable is simply binary in nature, with either 1 (stand for success / yes) or 0 (stand for failure / no) coding of results.

1.2 SGD

Suppose you have a massive data set containing millions of samples, and if you're using a regular Gradient Descent optimization process, you'll have to use all one million samples to complete one iteration with each iteration until the minimum is reached. Hence, the numerical efficiency is rather difficult. The problem is solved by Stochastic Gradient Descent. In SGD, performing each iteration needs only one sample, that is, a batch size of one. The sample is randomly shuffled and picked for the outcome of its iteration.

1.3 Random Forest

This algorithm is a collection of a number of individual decision trees as one. Classes are predicted by each random tree and class with high votes is our model of prediction. So random forest works on the principle which is called wisdom of crowds.

1.3 Decision Tree

This algorithm is a simple data structure designed to model decision rules on a particular problem. One function is selected at each node to differentiate the decisions. If the leaf node has fewer data points optimally we can avoid splitting. All such nodes on the leaf then give us insight into the final result.

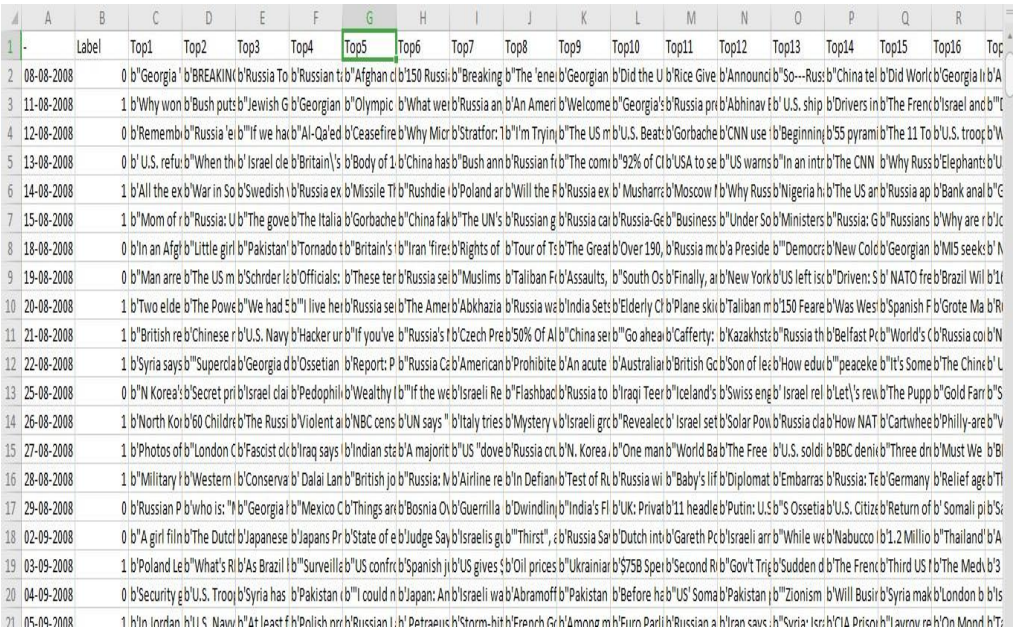
1.4 AdaBoost

It is better used to increase the usefulness of decision trees on binary classification issues. It can be called as discrete AdaBoost, as it is used for classification rather than regression. AdaBoost can be used to improve performance and is better spent on poor learners. Adaboost is the most feasible and comfortable to use compared to decision tree.

1.5 KNN

A supervised classification algorithm is the algorithm k-nearest to neighbors. It searches the points which are closest neighbors and access them to vote. The name for the new point is what most neighbors have with every mark. Here "k" is the number of neighbors in K-Nearest Neighbors that it looks for. It is overseen because you are attempting to classify a point based on the other points that are classified as known.

2. EXPERIMENT ANALYSIS



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	
1	-	Label	Top1	Top2	Top3	Top4	Top5	Top6	Top7	Top8	Top9	Top10	Top11	Top12	Top13	Top14	Top15	Top16	Top
2	08-08-2008	0	b"Georgia '	b'BREAKIN(	b'Russia To	b'Russian t	b"Afghan cl	b'150 Russi	b"Breaking	b"The 'ene	b'Georgian	b'Did the U	b'Rice Give	b'Announci	b"So---Russ	b'China tel	b'Did Worl	b'Georgia l	b'A
3	11-08-2008	1	b'Why won	b'Bush puts	b'Jewish G	b'Georgian	b"Olympic	b'What we	b'Russia an	b'An Ameri	b'Welcome	b"Georgia's	b'Russia pr	b'Abhinav E	b' U.S. ship	b'Drivers in	b'The Frenc	b'Israel and	b"'L
4	12-08-2008	0	b'Rememb	b"Russia 'e	b'"If we ha	b'Al-Qa'ed	b'Ceasefire	b'Why Micr	b'Stratfor:	b"I'm Tryin	b'The US m	b'U.S. Beat	b'Gorbache	b'CNN use	b'Beginnin	b'55 pyrami	b'The 11 To	b'U.S. troo	b'W
5	13-08-2008	0	b' U.S. refu	b"When th	b' Israel cle	b'Britain\'s	b'Body of 1	b'China has	b"Bush ann	b'Russian fi	b"The com	b"92% of CI	b'USA to se	b"US warns	b"In an intr	b'The CNN	b'Why Russ	b'Elephant	b'U
6	14-08-2008	1	b'All the ex	b'War in So	b'Swedish	b'Russia ex	b'Missile Th	b"Rushdie	b'Poland ar	b'Will the F	b'Russia ex	b' Musharra	b'Moscow f	b'Why Russ	b'Nigeria h	b'The US ar	b'Russia ap	b'Bank anal	b"'G
7	15-08-2008	1	b"Mom of r	b"Russia: U	b"The gove	b'The Italia	b'Gorbache	b"China fak	b"The UN's	b'Russian g	b'Russia ca	b'Russia-Ge	b"Business	b"Under So	b'Ministers	b"Russia: G	b'Russians	b'Why are r	b'Jc
8	18-08-2008	0	b'In an Afgl	b"Little girl	b'Pakistan'	b'Tornado t	b"Britain's	b"Iran 'fire	b'Rights of	b'Tour of Tc	b'The Great	b'Over 190,	b'Russia mc	b'a Preside	b"'Democr	b'New Cold	b'Georgian	b'MI5 seek	b' N
9	19-08-2008	0	b"Man arre	b'The US m	b'Schrder l	b'Officials:	b'These ter	b'Russia sei	b"Muslims	b'Taliban Fc	b'Assaults,	b"South Os	b'Finally, ar	b'New York	b'US left isc	b"Driven: S	b' NATO fre	b'Brazil Wil	b'1(
10	20-08-2008	1	b'Two elde	b'The Powe	b'We had S	b'"I live her	b'Russia se	b'The Amer	b'Abkhazia	b'Russia we	b'India Set	b'Elderly Cl	b'Plane ski	b'Taliban m	b'150 Feare	b'Was West	b'Spanish F	b'Grote Ma	b'R
11	21-08-2008	1	b"British re	b'Chinese r	b'U.S. Navy	b'Hacker ur	b'"If you've	b"Russia's I	b'Czech Pre	b'50% Of Al	b'China se	b'"Go ahea	b'Cafferty:	b'Kazakhst	b'Russia th	b'Belfast Pc	b"World's C	b'Russia co	b'N
12	22-08-2008	1	b'Syria says	b'"Superclu	b'Georgia d	b'Ossetian	b'Report: P	b"Russia Ca	b'American	b'Prohibite	b'An acute	b'Australian	b'British Gc	b'Son of le	b'How edu	b'"peaceke	b"It's Some	b'The Chine	b' U
13	25-08-2008	0	b"N Korea'	b'Secret pri	b'Israel clai	b'Pedophil	b'Wealthy I	b'"If the we	b'Israeli Re	b"Flashbac	b'Russia to	b'Iraqi Teer	b"Iceland's	b'Swiss eng	b' Israel rel	b'Let\'s rew	b'The Pupp	b"Gold Farr	b"'S
14	26-08-2008	1	b'North Kor	b'60 Childr	b'The Russi	b'Violent ai	b'NBC cens	b'UN says "	b'Italy tries	b'Mystery v	b'Israeli gro	b'Revealec	b' Israel set	b'Solar Pow	b'Russia cla	b'How NAT	b'Cartwhee	b'Philly-are	b"'V
15	27-08-2008	1	b'Photos of	b"London C	b'Fascist cl	b'Iraq says	b'Indian sta	b'A majorit	b"US 'dove	b'Russia cru	b'N. Korea	b"One man	b"World Ba	b'The Free	b'U.S. soldi	b'BBC deni	b'Three dr	b'Must We	b'BI
16	28-08-2008	1	b"Military I	b'Western	b'Conserva	b' Dalai Lan	b"British jo	b"Russia: N	b'Airline re	b'In Defian	b'Test of Ru	b'Russia wi	b"Baby's lif	b'Diplomat	b'Embarras	b'Russia: Te	b'Germany	b'Relief age	b'TI
17	29-08-2008	0	b'Russian P	b'who is: "I	b"Georgia I	b"Mexico C	b'Things are	b'Bosnia Ov	b'Guerrilla	b'Dwindlin	b'India's Fl	b'UK: Privat	b'11 headle	b'Putin: U.S	b"'S Ossetia	b'U.S. Citiz	b'Return of	b' Somali pi	b'Sa
18	02-09-2008	0	b"A girl filn	b'The Dutc	b'Japanese	b'Japans Pr	b'State of e	b'Judge Say	b'Israelis gu	b"'Thirst",	b'Russia Sar	b'Dutch int	b'Gareth Pc	b'Israeli arr	b"While we	b'Nabucco	b'1.2 Millio	b"Thailand'	b'A
19	03-09-2008	1	b'Poland Le	b"What's Ri	b'As Brazil	b'"Surveille	b"US confrc	b'Spanish ju	b'US gives $	b'Oil prices	b'Ukrainiar	b'$75B Spe	b'Second Ri	b"Gov't Trig	b'Sudden d	b'The Frenc	b'Third US f	b'The Medv	b'3
20	04-09-2008	0	b'Security g	b'U.S. Troo	b'Syria has	b' Pakistan	b'"I could n	b'Japan: An	b'Israeli wa	b'Abramoff	b'Pakistan	b'Before ha	b"US' Soma	b'Pakistan j	b'"Zionism	b'Will Busir	b'Syria mak	b'London b	b'Is
21	05-09-2008	1	b'In Jordan	b'U.S. Navy	b"At least f	b'Polish pro	b'Russian L	b' Petraeus	b'Storm-hit	b'French Gc	b'Among m	b'Euro Parl	b'Russian a	b'Iran says	b"Syria: Isr	b'CIA Prisor	b"Lavrov re	b'On Mond	b'T

Fig 2.1: Dataset

This is the dataset that we will use to forecast the movement of stocks. It includes eight years of regular trends, as well as dates and top 25 trends. Mark is an attribute with a value of either 1 or 0. 1 Suggests a rise in the stock price or stayed the same. 0 Suggests a fall in stock price.

```
In [2]: import pandas as pd
        from sklearn.feature_extraction.text import CountVectorizer, TfidfVectorizer
        from sklearn.linear_model import LogisticRegression
        from sklearn.metrics import accuracy_score, classification_report, confusion_matrix
        from sklearn.ensemble import RandomForestClassifier

        news= pd.read_csv('/Users/shiva prasad/Desktop/Combined_News_DJIA.csv')
        print(news.head())
```

```
         Date  Label                                               Top1  \
0  2008-08-08      0  b"Georgia 'downs two Russian warplanes' as cou...
1  2008-08-11      1  b'Why wont America and Nato help us? If they w...
2  2008-08-12      0  b'Remember that adorable 9-year-old who sang a...
3  2008-08-13      0  b' U.S. refuses Israel weapons to attack Iran:...
4  2008-08-14      1  b'All the experts admit that we should legalis...

                                                Top2  \
0           b'BREAKING: Musharraf to be impeached.'
1       b'Bush puts foot down on Georgian conflict'
2               b"Russia 'ends Georgia operation'"
3  b"When the president ordered to attack Tskhinv...
4  b'War in South Osetia - 89 pictures made by a ...

                                                Top3  \
0  b'Russia Today: Columns of troops roll into So...
1  b"Jewish Georgian minister: Thanks to Israeli ...
2  b'"If we had no sexual harassment we would hav...
3  b' Israel clears troops who killed Reuters cam...
```

Fig 2.2: Displaying the data frame values

We will be reading the Excel sheets by using the pandas and we will convert the Excel sheets into data frames and display the data frames.

```
: from sklearn.metrics import confusion_matrix
train_news = news[news['Date'] < '2014-07-15']
test_news = news[news['Date'] > '2014-07-14']

train_news_list = []
for row in range(0,len(train_news.index)):
    train_news_list.append(' '.join(str(k) for k in train_news.iloc[row,2:27]))

vectorize= CountVectorizer(min_df=0.01, max_df=0.8)
news_vector = vectorize.fit_transform(train_news_list)

test_news_list = []
for row in range(0,len(test_news.index)):
    test_news_list.append(' '.join(str(x) for x in test_news.iloc[row,2:27]))
test_vector = vectorize.transform(test_news_list)

nvectorize = TfidfVectorizer(min_df=0.05, max_df=0.85,ngram_range=(2,2))
news_nvector = nvectorize.fit_transform(train_news_list)

nmodel = lr.fit(news_nvector, train_news["Label"])

test_news_list = []
for row in range(0,len(test_news.index)):
    test_news_list.append(' '.join(str(x) for x in test_news.iloc[row,2:27]))
ntest_vector = nvectorize.transform(test_news_list)
npredictions = nmodel.predict(ntest_vector)

pd.crosstab(test_news["Label"], npredictions, rownames=["Actual"], colnames=["Predicted"])

accuracy2=accuracy_score(test_news['Label'], npredictions)
print(" Logistics Regression ",accuracy2)
```

```
accuracy2=accuracy_score(test_news['Label'], npredictions)
print(" Logistics Regression ",accuracy2)
print(confusion_matrix(test_news['Label'], npredictions))
```

```
C:\Users\shiva prasad\anaconda\lib\site-packages\sklearn\linear_model\logistic.py:432: Future
anged to 'lbfgs' in 0.22. Specify a solver to silence this warning.
  FutureWarning)

 Logistics Regression  0.5311871227364185
[[ 82 158]
 [ 75 182]]
```

Fig 2.3: Logistic Regression Result

The accuracy of the model is determined after constructing the logistic regression model. Logistic regression model output is assessed

```
35]: from sklearn.linear_model import SGDClassifier
     sgd=SGDClassifier(random_state=19)
     sgd =sgd.fit(news_nvector, train_news["Label"])
     sgdPrediction = sgd.predict(ntest_vector)
     sgdAccuracy=accuracy_score(test_news['Label'], sgdPrediction.round())
     print("SGD Classifier Accuracy:",sgdAccuracy)
     print("Confusion Matrix:")
     print(confusion_matrix(test_news['Label'], sgdPrediction))
```

```
SGD Classifier Accuracy: 0.5311871227364185
Confusion Matrix:
[[123 117]
 [116 141]]
```

Fig 2.4: SGD classifier Result

The model accuracy is calculated after the build of the SGD classifier. Using the confusion matrix the efficiency of the SGD classifier model is assessed.

```
[137]: from sklearn.ensemble import RandomForestClassifier
       RandomForest=RandomForestClassifier(random_state=19)
       RandomForest=RandomForest.fit(news_nvector,train_news["Label"])
       RandomForestPrediction=RandomForest.predict(ntest_vector)
       RandomForestAccuracy=accuracy_score(test_news["Label"],RandomForestPrediction)
       print("Random Forest Accuracy:",RandomForestAccuracy)
       print("Confusion Matrix:")
       print(confusion_matrix(test_news['Label'], RandomForestPrediction))
```

```
Random Forest Accuracy: 0.5412474849094567
Confusion Matrix:
[[159  81]
 [147 110]]
```

Fig 2.5: Random Forest Model Result

The model's accuracy is calculated after the build of the Random Forest classifier model. The performance of the model Random Forest classifier is assessed using the confusion matrix.

```
141]: from sklearn.neighbors import KNeighborsClassifier
knn=KNeighborsClassifier(n_neighbors=42)
knn=knn.fit(news_nvector,train_news["Label"])
knnprediction=knn.predict(ntest_vector)
knnaccuracy=accuracy_score(test_news["Label"],knnprediction)
print("K Nearest Neighbor Accuracy:",knnaccuracy)
print("Confusion Matrix:")
print(confusion_matrix(test_news['Label'], knnprediction))

K Nearest Neighbor Accuracy: 0.5513078470824949
Confusion Matrix:
[[ 99 141]
 [ 82 175]]
```

Fig 2.6: KNN Classifier Result

The model's accuracy is calculated after the build of the KNN classifier model. The performance of the KNN model classifier is assessed using the confusion matrix.

```
In [136]: from sklearn.tree import DecisionTreeClassifier
DecisionTree = DecisionTreeClassifier(max_depth=9)
DecisionTree=DecisionTree.fit(news_nvector,train_news["Label"])
DecisionTreePrediction=DecisionTree.predict(ntest_vector)
DecisionTreeAccuracy=accuracy_score(test_news["Label"],DecisionTreePrediction)
print("Decision Tree Accuracy:",DecisionTreeAccuracy)
print("Confusion Matrix:")
print(confusion_matrix(test_news['Label'], DecisionTreePrediction))

Decision Tree Accuracy: 0.5130784708249497
Confusion Matrix:
[[ 30 210]
 [ 32 225]]
```

Fig2.7: Decision Tree Classifier Result

The model's accuracy is after the creation of the Decision Tree classifier algorithm. The output of the classifier model for Decision Tree is evaluated using the confusion matrix.

```
[137]: from sklearn.ensemble import RandomForestClassifier
RandomForest=RandomForestClassifier(random_state=19)
RandomForest=RandomForest.fit(news_nvector,train_news["Label"])
RandomForestPrediction=RandomForest.predict(ntest_vector)
RandomForestAccuracy=accuracy_score(test_news["Label"],RandomForestPrediction)
print("Random Forest Accuracy:",RandomForestAccuracy)
print("Confusion Matrix:")
print(confusion_matrix(test_news['Label'], RandomForestPrediction))

Random Forest Accuracy: 0.5412474849094567
Confusion Matrix:
[[159  81]
 [147 110]]
```

Fig 2.8: Random Forest Model Result

The model's accuracy is calculated after the build of the Random Forest classifier model. The performance of the model Random Forest classifier is assessed using the confusion matrix.

```
141]: from sklearn.neighbors import KNeighborsClassifier
knn=KNeighborsClassifier(n_neighbors=42)
knn=knn.fit(news_nvector,train_news["Label"])
knnprediction=knn.predict(ntest_vector)
knnaccuracy=accuracy_score(test_news["Label"],knnprediction)
print("K Nearest Neighbor Accuracy:",knnaccuracy)
print("Confusion Matrix:")
print(confusion_matrix(test_news['Label'], knnprediction))

K Nearest Neighbor Accuracy: 0.5513078470824949
Confusion Matrix:
[[ 99 141]
 [ 82 175]]
```

Fig 2.9 KNN Classifier Result

The model's accuracy is calculated after the build of the KNN classifier model. The performance of the KNN model classifier is assessed using the confusion matrix.

```
3]: from sklearn.ensemble import AdaBoostClassifier
AdaBoost=AdaBoostClassifier(n_estimators=10)
AdaBoost =AdaBoost.fit(news_nvector, train_news["Label"])
AdaBoostPrediction = AdaBoost.predict(ntest_vector)
AdaBoostAccuracy=accuracy_score(test_news['Label'], AdaBoostPrediction)
print("Ada Boost Accuracy:",AdaBoostAccuracy)
print("Confusion Matrix:")
print(confusion_matrix(test_news['Label'], AdaBoostPrediction))

Ada Boost Accuracy: 0.5633802816901409
Confusion Matrix:
[[125 115]
 [102 155]]
```

Fig 3.1 AdaBoost Classifier Result

After building the AdaBoost classifier model, the accuracy of the model is calculated. The performance of the AdaBoost classifier model is evaluated using the confusion matrix.

Table -1: Results

S.No.	ML MODEL	ACCURACY
1	Decision Tree Algorithm	51.1%
2	SGD Algorithm	53.11%
3	Logistic Regression	53%
4	Random Forest Classifier	54.2%
5	KNN Classifier	55.2%
6	AdaBoost Classifier	56.3%

3. CONCLUSION

Using only textual datasets, the proposed system with various algorithms to predict stocks while comparing them with each other. Classification with the numerical data has been done using linear regression. In our project, Adaboost classifier gives higher accuracy of 56.3% compared to other algorithms

REFERENCES

[1] https://webfocusinfocenter.informationbuilders.com/wfappent/TLs/TL_rstat/ source/LogisticRegression43.htm
[2] http://dsbyprateekg.blogspot.com/2017/09/machine-learning-decision-trees- and.html
[3] https://www.geeksforgeeks.org/ml-stochastic-gradient-descent-sgd/
[4] https://towardsdatascience.com/boosting-algorithm-adaboost-b6737a9ee60c
[5] https://www.kaggle.com/aaron7sun/stocknews
[6] https://www.tutorialspoint.com/machine_learning_with_python/machine_lear ning_with_python_knn_algorithm_finding_nearest_neighbors.htm