PROJECT REPORT

Professor Name

Dr. Vimal Bhatia

Project Name

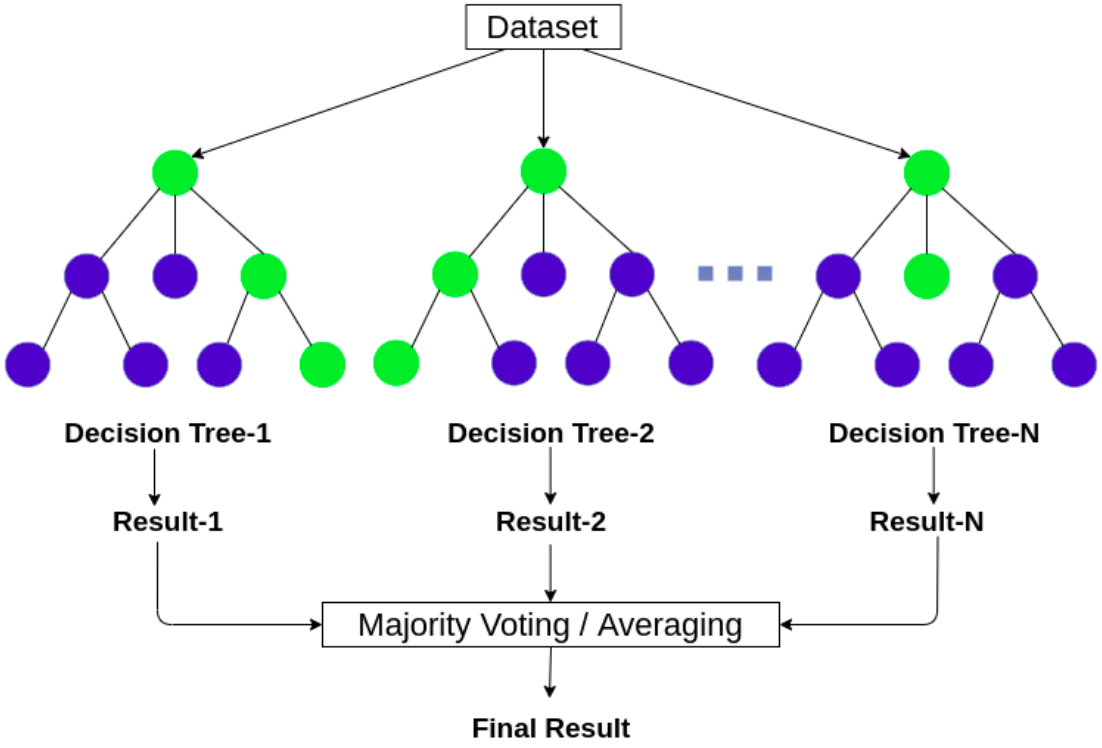
Stock Prices Prediction Using Machine Learning

Student Name

Rupesh Gelal

1-PROBLEM

Stock market are almost unpredictable. The goal of this project is to predict stock market price of next few days using news headlines.



2- SOLUTION

In this project, I will build machine learning model using random forests algorithm to predict the stock price.

I will be importing random forest classifier from the sklearn library rather than coding it from the scratch.

Before, passing data to the classifier I will be doing necessary pre-processing steps to clean the data like removing special characters, converting to lowercase to name a few.

2.1 ALGORITHM

Random forest is an ensemble machine learning technique that averages several decision trees on different parts of the same training set. It is used for both classification and regression problem statements. The best part of the algorithm is that there are very few assumptions attached to it so data preparation is less challenging which results in time-saving. The steps that are included while performing the random forest algorithm are as follows:

Step-1: Pick K random records from the dataset having a total of N records.

Step-2: Build and train a decision tree model on these K records.

Step-3: Choose the number of trees you want in your algorithm and repeat steps 1 and 2.

Step-4: In the case of a regression problem, for an unseen data point, each tree in the forest predicts a value for output. The final value can be calculated by taking the mean or average of all the values predicted by all the trees in the forest.

and, in the case of a classification problem, each tree in the forest predicts the class to which the new data point belongs. Finally, the new data point is assigned to the class that has the maximum votes among them i.e, wins the majority vote.

3- DEMO:

-FIRST, Importing necessary library and reading csv file.

```
import pandas as pd
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import classification_report,confusion_matrix,accuracy_score
```

```
df = pd.read_csv('/Stock_Data .csv', encoding = "ISO-8859-1")
```

-SECOND, Creating training and testing data set and doing necessary preprocessing.

```
train = df[df['Date'] < '20150101']
test = df[df['Date'] > '20141231']
```

```
data=train.iloc[:,2:27]
data.replace("[^a-zA-Z]"," ",regex=True, inplace=True)
```

```
list1= [i for i in range(25)]
new_Index=[str(i) for i in list1]
data.columns= new_Index
```

```
for index in new_Index:
    data[index] = data[index].str.lower()
data.head(1)
```

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

-THIRD, Converting word2vec and training and testing.

```
countvector=CountVectorizer(ngram_range=(2,2))
traindataset=countvector.fit_transform(headlines)
```

```
randomclassifier=RandomForestClassifier(n_estimators=200,criterion='entropy')
randomclassifier.fit(traindataset,train['Label'])
```

```
RandomForestClassifier(bootstrap=True, ccp_alpha=0.0, class_weight=None,
                       criterion='entropy', max_depth=None, max_features='auto',
                       max_leaf_nodes=None, max_samples=None,
                       min_impurity_decrease=0.0, min_impurity_split=None,
                       min_samples_leaf=1, min_samples_split=2,
                       min_weight_fraction_leaf=0.0, n_estimators=200,
                       n_jobs=None, oob_score=False, random_state=None,
                       verbose=0, warm_start=False)
```

```
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 = randomclassifier.predict(test_dataset)
```

4-ACCURACY:

```
matrix = confusion_matrix(test['Label'],predictions)
print(matrix)
score = accuracy_score(test['Label'],predictions)
print(score)
report = classification_report(test['Label'],predictions)
print(report)
```

```
[[135  51]
 [  7 185]]
0.8465608465608465
              precision    recall  f1-score   support

           0       0.95      0.73      0.82       186
           1       0.78      0.96      0.86       192

    accuracy                           0.85       378
   macro avg       0.87      0.84      0.84       378
weighted avg       0.87      0.85      0.84       378
```