***Project Report***

Introduction

Sentiment Analysis is a machine learning tool which analyses texts and classifying the data into different classes based on the emotion that data has, by training machine learning tools with examples of emotions in text, machines automatically learn how to detect sentiment without human input. The problem statement that is being solved using machine learning is Stock Sentiment Analysis using News Headlines.

Description of Dataset-

The dataset has been taken from Kaggle, the dataset contains Top 25 news headlines about a specific company which are ranging between the year 2000 to 2016, and a label which determines whether the stock price will have a positive or a negative impact, 0 indicating stock price will have negative effect and 1 indicating positive effect. One news headline constitutes each observation.

Description of application -

Stock market data analysis can be understood in a better way with the help of artificial intelligence and data mining techniques. The volatility of stock prices depends on gains or losses of certain companies. News articles are one of the most important factors which influence the stock market. This project basically shows the effect of emotion classification of financial news to the prediction of stock market prices. The application of this project is to predict the sentiment of the stock price whether the News will have a positive impact or a negative impact on the stock price.

Methodology

1. Importing /Cleaning/preprocessing the dataset

Data cleaning is the process of fixing or removing incorrect, incomplete and unnecessary data which will be in no way useful in solving the problem. Data Cleaning or Preprocessing is an important step before model building because If data is incorrect, outcomes and algorithms are unreliable and inaccurate, even though they may look correct. There is no specific way to define the steps that should be included in data cleaning as it depends on the complexity and necessity of our model, the steps in data cleaning differ from dataset to dataset.  
1. The news headlines data is stored in a CSV file which is imported using pandas, usage of encoding -"ISO-8859-1" is necessary to import this particular dataset because there are many special characters that are present in the dataset which are imported with the help of this encoding.  
2. Dataset is split into training and testing set, headlines before a specific date is training and headlines after that is testing set. here the data is not being split randomly for training and testing because whenever time in included in the problem, older data can be used for training and newer data for testing, after splitting of data.  
3. Punctuations and special characters are removed because they are not required in sentiment analysis, also renaming the column names with '1,2,3...' for ease of access and converting the text to lower case. here removing of stop words is not a good option because in use case which are news headlines, each and every word plays an important role, removing them might change the meaning of sentence.

2. Converting the text into Vectors

1. Machines cannot understand characters and words. So, when dealing with text data we need to represent it in numbers to be understood by the machine. Countvectorizer is a method to convert text to numerical data.  
Before using countvectorizer we need to combine text of all headlines into a one paragraph/sentence so that it can be converted into vectors, a similar process is done to combine all headlines of all records in the dataset for converting them into vectors.  
2. Countvectorizer helps us to create bag of words, The bag-of-words (BOW) model is a representation that turns arbitrary text into fixed-length vectors by counting how many times each word appears. One important parameter while implementing bag of words is Ngram, an N-gram means a sequence of N words, we complete this process by fitting and transforming our independent feature.

3. Implementing Models

1. Random Forest classifier

Random forest consists of a large number of individual decision trees that operate as an ensemble. Each individual tree in the random forest spits out a class prediction and the class with the most votes becomes our model’s prediction, in our model we are using two hyperparameters which are n\_estimators - which define the number of trees in the forest, and criterion -which is the function to measure the quality of a split. Supported criteria are “gini” for the Gini impurity and “log\_loss” and “entropy”. After hyperparameter tuning we fit the model on X and y data.   
Once the model is trained, the same preprocessing steps are performed for the test data set.  
We are going to implement the same code using TFIDF vectorizer, and perform random forest classification, we observe that the accuracy has increased by 1% in this case.

2. Naive Bayes

A Naive Bayes classifier is a probabilistic machine learning model that’s used for classification tasks. The crux of the classifier is based on the Bayes theorem. There are 3 types of naive bayes- Multinomial, Bernoulli, Gaussian.  
In our model we are implementing Multinomial Naive Bayes - This is mostly used for document classification problem, i.e. whether a document belongs to the category of sports, politics, technology etc. The features/predictors used by the classifier are the frequency of the words present in the document; as we are dealing with sentiment analysis in our problem, Multinomial NB is a good choice.

3. Gradient Boosting Classifier

Gradient boosting is an example of Ensemble Learning, In Ensemble Learning, instead of using a single predictor, multiple predictors and training in the data and their results are aggregated, usually giving a better score than using a single model. Boosting is a special type of Ensemble Learning technique that works by combining several weak learners (predictors with poor accuracy) into a strong learner (a model with strong accuracy). This works with each model paying attention to its predecessor’s mistakes. In Gradient Boosting, each predictor tries to improve on its predecessor by reducing the errors. But the fascinating idea behind Gradient Boosting is that instead of fitting a predictor on the data at each iteration, it actually fits a new predictor to the residual errors made by the previous predictor.

4. Adaboosting Classifier

Adaboosting, similar to Gradient boosting, is an ensemble technique. AdaBoost is similar to Random Forest in that they both tally up the predictions made by each decision trees within the forest to decide on the final classification. There are, however, some subtle differences. For instance, in AdaBoost, the decision trees have a depth of 1 (i.e., 2 leaves). In addition, the predictions made by each decision tree have varying impact on the final prediction made by the model.

5.logistic regression

Logistic Regression is a Supervised Machine Learning Technique, In Supervised Learning, the machine learns under supervision. It contains a model that is able to predict with the help of a labeled dataset. A labeled dataset is one where you already know the target answer. Logistic Regression is used when the dependent variable(target) is categorical.

6.K nearest Neighbour

The KNN algorithm assumes that similar things exist in close proximity. In other words, similar things are near to each other. KNN captures the idea of similarity (sometimes called distance, proximity, or closeness) with some mathematics — calculating the distance between points on a graph. There are other ways of calculating distance, and one way might be preferable depending on the problem we are solving. However, the straight-line distance (also called the Euclidean distance) is a popular and familiar choice.

7. Multi-layer Perceptron classifier

Multi-layer perception is also known as MLP. It is fully connected dense layers, which transform any input dimension to the desired dimension. A multi-layer perception is a neural network that has multiple layers. To create a neural network, we combine neurons together so that the outputs of some neurons are inputs of other neurons. A multi-layer perceptron has one input layer and for each input, there is one neuron (or node), it has one output layer with a single node for each output and it can have any number of hidden layers and each hidden layer can have any number of nodes

Results

1. Random Forest classifier (Count vectorizer)

a.) Accuracy of model - 0.8624338624338624   
 b.) Confusion Matrix -

[[140 46]   
 [ 6 186]]

2. Random Forest classifier (TFIDF vectorizer)

a.) Accuracy of model - 0.8492063492063492  
 b.) Confusion Matrix -

[[144 42]   
 [ 15 177]]

3. Naive bayes classifier

a.) Accuracy of model - 0.8518518518518519  
 b.) Confusion Matrix -

[[130 56]   
 [ 0 192]]

4. Gradient boosting Classifier

a.) Accuracy of model - 0.7619047619047619  
 b.) Confusion Matrix -

[[132 54]   
 [ 36 156]]

5. Adaboosting. Classifier

a.) Accuracy of model - 0.7275132275132276  
 b.) Confusion Matrix -

[[131 55]   
 [ 48 144]]

6. logistic Regression

a.) Accuracy of model - 0.8544973544973545  
 b.) Confusion Matrix -

[[131 55]   
 [ 0 192]]

7. K nearest Neighbours

a.) Accuracy of model - 0.8444973544973545  
 b.) Confusion Matrix -

[[131 54]   
 [ 0 193]]

8. Multi-layer Perceptron classifier

a.) Accuracy of model - 0.8465608465608465  
 b.) Confusion Matrix -

[[148 38]  
 [ 20 172]]