JETIR.ORG ISSN: 2349-5162 | ESTD Year : 2014 | Monthly Issue
JOURNAL OF EMERGING TECHNOLOGIES AND INNOVATIVE RESEARCH (JETIR)
An International Scholarly Open Access, Peer-reviewed, Refereed Journal

FAKE ONLINE REVIEWS DETECTION USING MACHINE LEARNING

[1]Santhosh Voruganti , [2]U.Sairam

[1]Assistant Professor, [2]Assistant professor
[1,2] Information Technology
[1,2]CBIT , Hyderabad, India

Abstract : Online reviews have great impact on today's business and commerce. Decision making for purchase of online products mostly depends on reviews given by the users. Hence, opportunistic individuals or groups try to manipulate product reviews for their own interests.

We make some classification approaches for detecting fake online reviews. we use Label Propagation algorithm. Random Forest, K nearest neighbours are used as classifiers in our research work to improve the performance of classification. We have mainly focused on the content of the review-based approaches. As feature we have used word frequency count, sentiment polarity and length of review.

***Index Terms* - Label Propagation algorithm. Random Forest, K nearest neighbors**

I. INTRODUCTION

Almost, every one of us checks out reviews before purchasing some products or services. Hence, online reviews have become a great source of reputation for the companies. Also, they have large impact on advertisement and promotion of products and services. With the spread of online marketplace, fake online reviews are becoming great matter of concern. People can make false reviews for promotion of their own products that harms the actual users. Also, competitive companies can try to damage each other's reputation by providing fake negative reviews.

Online reviews have great impact on today's business and commerce. Decision making for purchase of online products mostly depends on reviews given by the users. Hence, opportunistic individuals or groups try to manipulate product reviews for their own interests.

Companies can increase the sales of their products by write fake positive reviews and giving highest rating for their products. And also, companies can decrease the sales of other company's products by writing fake negative reviews and giving lowest rating for products. So by using our project one can find out the fake reviews.

The impact of online reviews on businesses has grown significantly during last years, being crucial to determine business success in a wide array of sectors, ranging from restaurants, hotels to e-commerce. Unfortunately, some users use unethical means to improve their online reputation by writing fake reviews of their businesses or competitors. Previous research has addressed fake review detection in a number of domains, such as product or business reviews in restaurants and hotels.

We make some classification approaches for detecting fake online reviews. we use Label Propagation algorithm. Random Forest , K nearest neighbors are used as classifiers in our research work to improve the performance of classification. We have mainly focused on the content of the review-based approaches. As feature we have used word frequency count, sentiment polarity and length of review.

In this paper we will be developing ML model to detect fake online reviews. The model uses a labelled database of hotel reviews. The dataset contains 1600 reviews out of which 800 are real reviews and 800 are fake reviews. We have mainly focused on the content of the review based approaches.

The system is very fast and effective due to semi-supervised and supervised learning. Focused on the content of the review based approaches. As feature we have used word frequency count, sentiment polarity and length of review.

Companies can increase the sales of their products by write fake positive reviews and giving highest rating for their products. And also, companies can decrease the sales of other company's products by writing fake negative reviews and giving lowest rating for products. So by using our project one can find out the fake reviews.

II. SYSTEM DESIGN

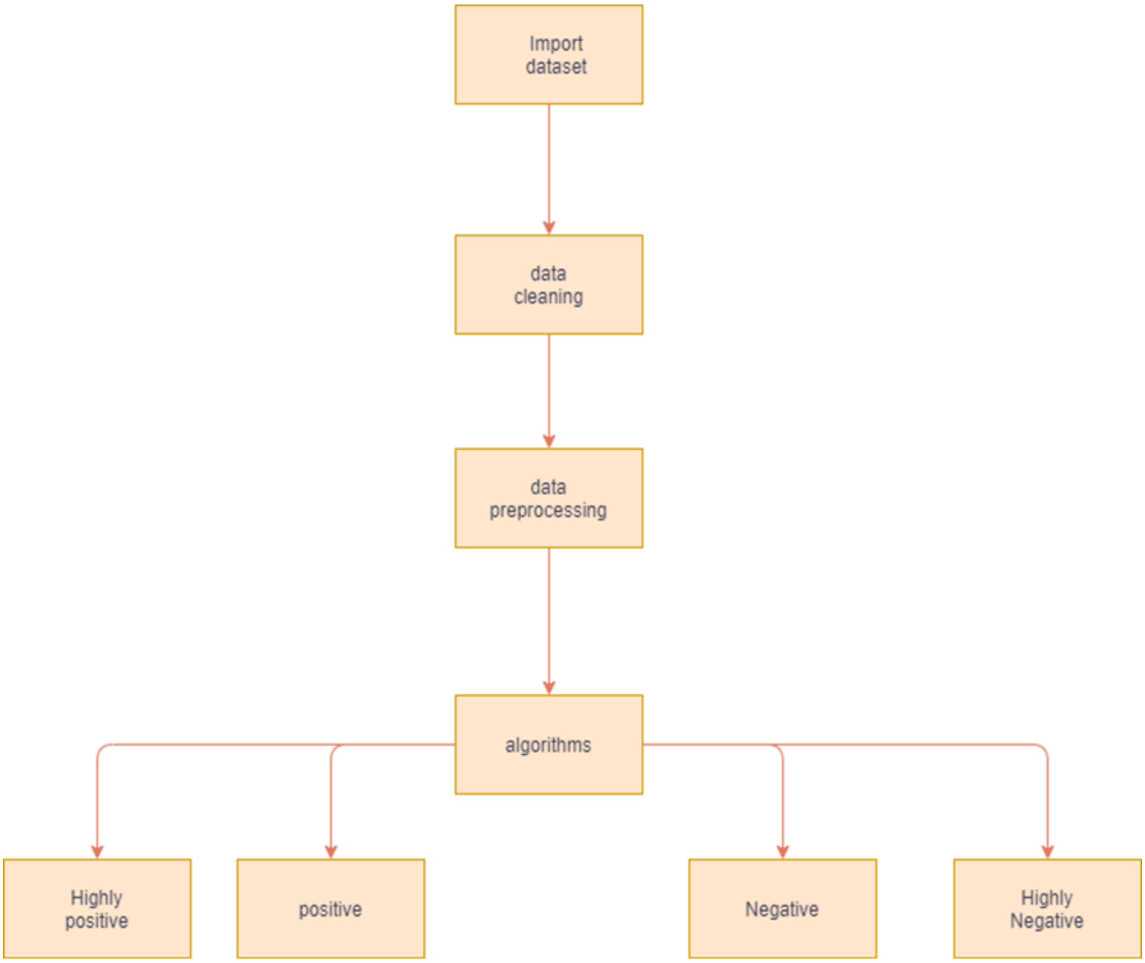

Figure 2.1: Proposed System

A. Dataset

We have taken a hotel dataset which has 1600 hotel reviews. The dataset consist of true reviews and fake reviews. The number of true reviews are 800 and the number of fake reviews are 800. The reviews are further divided into positive reviews and negative reviews.

The true reviews are further divided into positive and negative reviews. The number of reviews which are positive and true are 400, the number of reviews which are negative and true are 400.

The fake reviews are also further divided into positive and negative. The number of reviews which are positive and fake are 400. The number of reviews which are negative and fake are 400.

B.Data Cleaning

Data cleaning is a critically important step in any machine learning project. In tabular data, there are many different statistical analysis and data visualization techniques you can use to explore your data in order to identify data cleaning operations you may want to perform. Before jumping to the sophisticated methods, there are some very basic data cleaning operations that you probably should perform on every single machine learning project. These are so basic that they are often overlooked by seasoned machine learning practitioners, yet are so critical that if skipped, models may break or report overly optimistic performance results

In this paper we removed all the stop words and used the lemmatization process to clean the data. Stop words are the words which does not give any semantic meaning to the data. Lemmatization in linguistics is the process of grouping together the inflected forms of a word so they can be analyzed as a single item, identified by the word's lemma, or dictionary form.

C. Data pre-processing

Before applying any machine learning algorithm data should be pre-processed and noisy data should be removed. If the irrelevant features are used for training of machine learning models then models may suffer from the under fitting problem. We also need to normalize all the input features by scaling. We will be dividing the dataset into 70 % as training dataset and 30 % as testing dataset.

D. Prediction

In this paper we use different algorithms such as random forest, k Nearest Neighbors, label propagation and MLP, we predict the reviews whether they are absolutely positive, positive, negative and absolutely negative. We even find the accuracy of the algorithms and compare them.

III.IMPLEMENTATION

Semi-supervised learning is an approach to machine learning that combines a small amount of labelled data with a large amount of unlabeled data during training. Semi-supervised learning falls between unsupervised learning (with no labelled training data) and supervised learning (with only labelled training data).

Supervised learning is a process of providing input data as well as correct output data to the machine learning model. The aim of a supervised learning algorithm is to find a mapping function to map the input variable(x) with the output variable(y).

E. Random Forest

Random forest is a flexible, easy to use machine learning algorithm that produces, even without hyper-parameter tuning, a great result most of the time. It is also one of the most used algorithms, because of its simplicity and diversity (it can be used for both classification and regression tasks). In this post we'll learn how the random forest algorithm works, how it differs from other algorithms and how to use it.

Random forest is a supervised learning algorithm. The "forest" it builds, is an ensemble of decision trees, usually trained with the “bagging” method. The general idea of the bagging method is that a combination of learning models increases the overall result.

One big advantage of random forest is that it can be used for both classification and regression problems, which form the majority of current machine learning systems. Let's look at random forest in classification, since classification is sometimes considered the building block of machine learning.

```
def rfc():
    text.delete('1.0', END)
    global rfc,rfc_acc
    rfc = RandomForestClassifier(min_samples_leaf= 1, min_samples_split= 2, n_estimators= 100, verbose=1,
    rfc.fit(X_train, y_train)
    y_pred=rfc.predict(X_test)
    rfc_acc = accuracy_score(y_test,y_pred)
    text.insert(END,"accuracy of RandomForest: "+str(rfc_acc)+"\n")
    predict(rfc,"RandomForest")
```

Figure 3.1 : RFC code

The most important parameter is the number of random features to sample at each split point (max_features).

You could try a range of integer values, such as 1 to 20, or 1 to half the number of input features.

•max_features [1 to 20]

Alternately, you could try a suite of different default value calculators.

•max_features in [‘sqrt’, ‘log2’]

Another important parameter for random forest is the number of trees (n_estimators).

Ideally, this should be increased until no further improvement is seen in the model.

Good values might be a log scale from 10 to 1,000.

•estimators in [10, 100, 1000]

F. K Nearest Neighbors

K-Nearest Neighbor is one of the simplest Machine Learning algorithms based on Supervised Learning technique. K-NN algorithm assumes the similarity between the new case/data and available cases and put the new case into the category that is most similar to the available categories.

K-NN algorithm stores all the available data and classifies a new data point based on the similarity. This means when new data appears then it can be easily classified into a well suite category by using K- NN algorithm. K-NN algorithm can be used for Regression as well as for Classification but mostly it is used for the Classification problems.

K-NN is a non-parametric algorithm, which means it does not make any assumption on underlying data. It is also called a lazy learner algorithm because it does not learn from the training set immediately instead it stores the dataset and at the time of classification, it performs an action on the dataset. KNN algorithm at the training phase just stores the dataset and when it gets new data, then it classifies that data into a category that is much similar to the new data.

Suppose there are two categories, i.e., Category A and Category B, and we have a new data point x1, so this data point will lie in which of these categories. To solve this type of problem, we need a K-NN algorithm. With the help of K-NN, we can easily identify the category or class of a particular dataset.

```
def knn():
    global knn,knn_acc
    knn = KNeighborsClassifier(n_neighbors=21)
    knn.fit(X_train,y_train)
    y_pred=knn.predict(X_test)
    knn_acc = accuracy_score(y_test,y_pred)
    text.insert(END,"accuracy of KNN: "+str(knn_acc)+"\n")
    predict(knn,"KNN")
```

Figure 3.2: KNN Code

The most important hyper parameter for KNN is the number of neighbors (neighbors).

Test values between at least 1 and 21, perhaps just the odd numbers.

• neighbors in [1 to 21]

It may also be interesting to test different distance metrics (metric) for choosing the composition of the neighborhood.

• metric in [‘euclidean’, ‘manhattan’, ‘minkowski’]

It may also be interesting to test the contribution of members of the neighborhood via different weightings (weights).

• weights in [‘uniform’, ‘distance’]

G. Label Propagation Algorithm

Label propagation is a semi-supervised machine learning algorithm that assigns labels to previously unlabelled data points. At the start of the algorithm, a subset of the data points have labels. These labels are propagated to the unlabelled points throughout the course of the algorithm.

Within complex networks, real networks tend to have community structure. Label propagation is an algorithm for finding communities. In comparison with other algorithms label propagation has advantages in its running time and amount of a priori information needed about the network structure (no parameter is required to be known beforehand). The disadvantage is that it produces no unique solution, but an aggregate of many solutions.

At initial condition, the nodes carry a label that denotes the community they belong to. Membership in a community changes based on the labels that the neighboring nodes possess. This change is subject to the maximum number of labels within one degree of the nodes. Every node is initialized with a unique label, then the labels diffuse through the network. Consequently, densely connected groups reach a common label quickly. When many such dense (consensus) groups are created throughout the network, they continue to expand outwards until it is impossible to do so

```
def semilp():
    global lpm,lpm_acc
    lpm = LabelPropagation()
    rng = np.random.RandomState(42)
    random_unlabeled_points = rng.rand(len(y_train)) < 0.3
    labels = np.copy(y_train)
    labels[random_unlabeled_points] = -1
    lpm.fit(X_train.toarray(), labels)
    y_pred=lpm.predict(X_test)
    lpm_acc = accuracy_score(y_test,y_pred)
    text.insert(END,"accuracy of SEMI-LP: "+str(lpm_acc)+"\n")
    predict(lpm,"LabelPropagation")
```

Figure 3.3: Label Propagation Code

H. MLP

A multilayer perceptron (MLP) is a class of feedforward artificial neural network (ANN). The term MLP is used ambiguously, sometimes loosely to any feedforward ANN, sometimes strictly to refer to networks composed of multiple layers of perceptron's (with threshold activation); see Multilayer perceptron's are sometimes colloquially referred to as "vanilla" neural networks, especially when they have a single hidden layer.

An MLP consists of at least three layers of nodes: an input layer, a hidden layer and an output layer. Except for the input nodes, each node is a neuron that uses a nonlinear activation function. MLP utilizes a supervised learning technique called backpropagation for training. Its multiple layers and non-linear activation distinguish MLP from a linear perceptron. It can distinguish data that is not linearly separable Subsequent work with multilayer perceptron's has shown that they are capable of approximating an XOR operator as well as many other non-linear functions.

Just as Rosenblatt based the perceptron on a McCulloch-Pitts neuron, conceived in 1943, so too, perceptron's themselves are building blocks that only prove to be useful in such larger functions as multilayer perceptron's.

The multilayer perceptron is the hello world of deep learning: a good place to start when you are learning about deep learning.

A multilayer perceptron (MLP) is a deep, artificial neural network. It is composed of more than one perceptron. They are composed of an input layer to receive the signal, an output layer that makes a decision or prediction about the input, and in between those two, an arbitrary number of hidden layers that are the true computational engine of the MLP. MLPs with one hidden layer are capable of approximating any continuous function.

```
def mlp():
    global mlp,mlp_acc
    mlp = MLPClassifier()
    mlp.fit(X_train,y_train)
    y_pred=mlp.predict(X_test)
    mlp_acc = accuracy_score(y_test,y_pred)
    text.insert(END,"accuracy of MLP: "+str(mlp_acc)+"\n")
    predict(mlp,"MLP")
```

Figure 3.4: MLP Code

I. Graph

In this module we are going to plot a graph comparing all the accuracies of the algorithms by using plt.bar() and plt.xticks() methods.

Plt.bar()

A bar plot or bar chart is a graph that represents the category of data with rectangular bars with lengths and heights that is proportional to the values which they represent. The bar plots can be plotted horizontally or vertically. A bar chart describes the comparisons between the discrete categories. One of the axis of the plot represents the specific categories being compared, while the other axis represents the measured values corresponding to those categories.

The matplotlib API in Python provides the bar() function which can be used in MATLAB style use or as an object-oriented API. The function creates a bar plot bounded with a rectangle depending on the given parameters.

```
def graph():
    acc = [rfc_acc,knn_acc,mlp_acc,lpm_acc]
    bars = ('RF','KNN','MLP','LB')
    y_pos = np.arange(len(bars))
    plt.bar(y_pos, acc)
    plt.xticks(y_pos, bars)


    plt.show()
```

Figure 3.5: Graph Code

IV.RESULTS

In this paper we have used tkinter as the frontend GUI. Tkinter is a Python binding to the Tk GUI toolkit. It is the standard Python interface to the Tk GUI toolkit, and is Python's standard GUI.

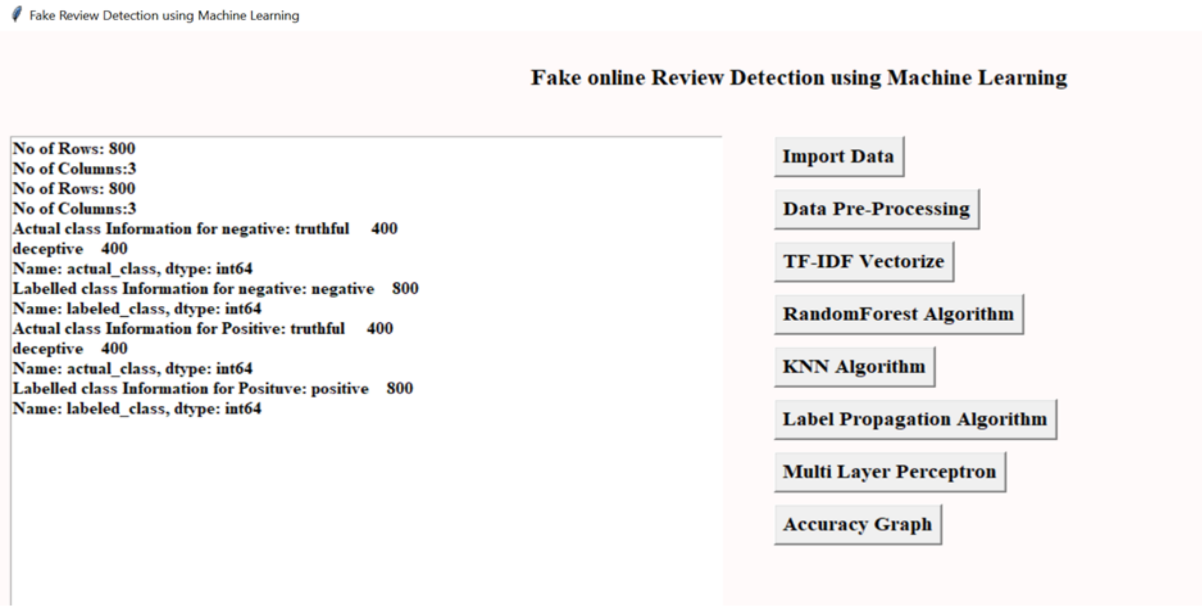

Figure 4.1: Output of the Import Data

In the above figure dataset information is displayed. In this paper we have used a dataset of 1600 reviews. The dataset is further divided into true reviews and fake reviews. The number of true reviews are 800 and the number of fake reviews are 800. The true and fake reviews are further divided into positive and Negative reviews which are 400 each.

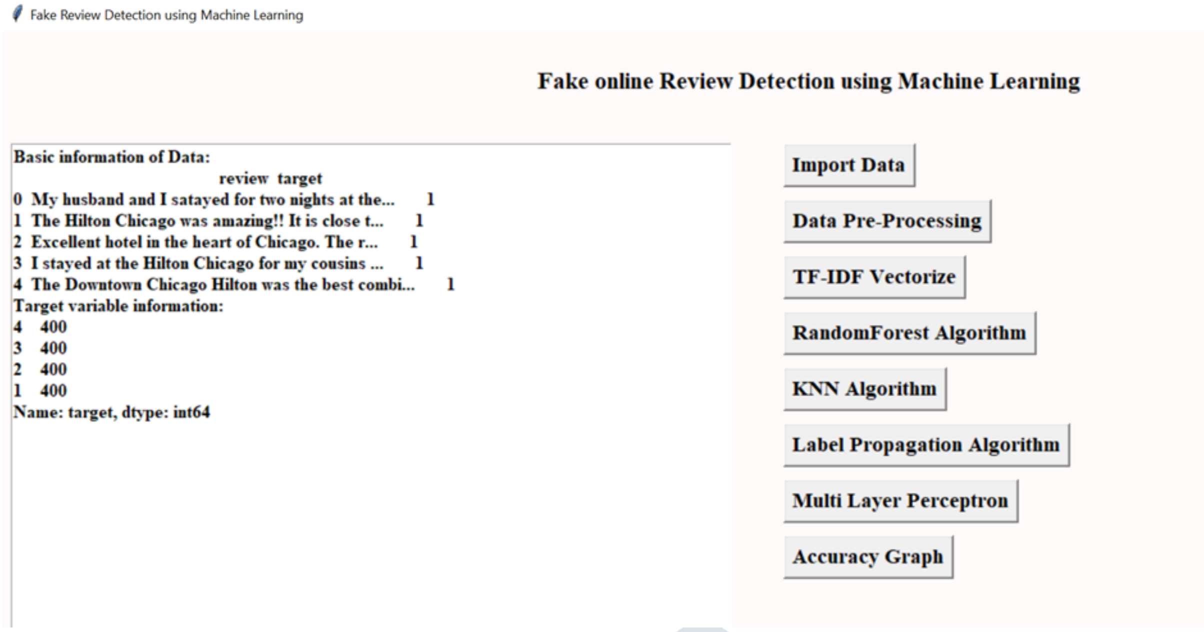

Figure 4.2: Data Preprocessing Output

In the above figure the data preprocessing output is printed. In this project the reviews are divided into four types. The first type are the reviews which are positive and true. The second type of review are positive and fake. The third type of reviews are negative and true. The fourth type of reviews are negative and fake.

In the below figure the TF-IDF output is printed. TF-IDF is an abbreviation for Term Frequency Inverse Document Frequency. This is very common algorithm to transform text into a meaningful representation of numbers which is used to fit machine algorithm for prediction. In this project we have calculated TF-IDF score for three thousand words.

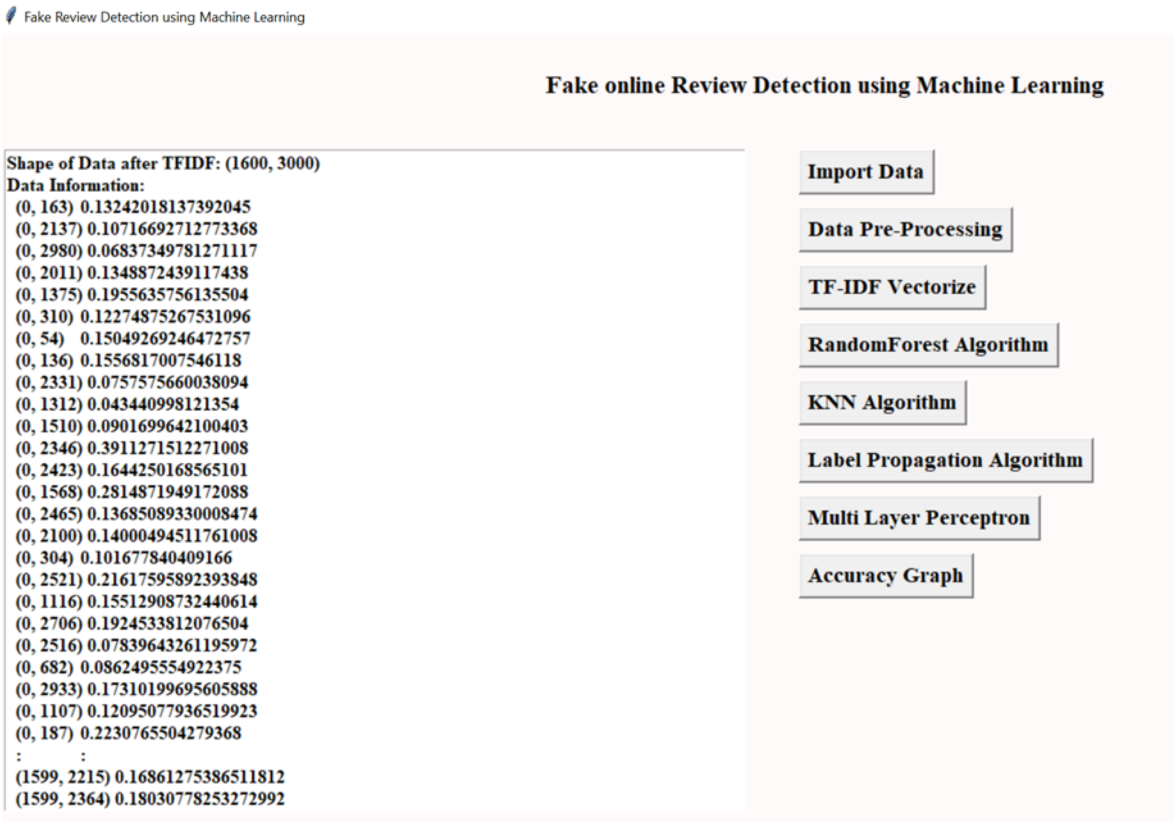

Figure 4.3: TF-IDF Output

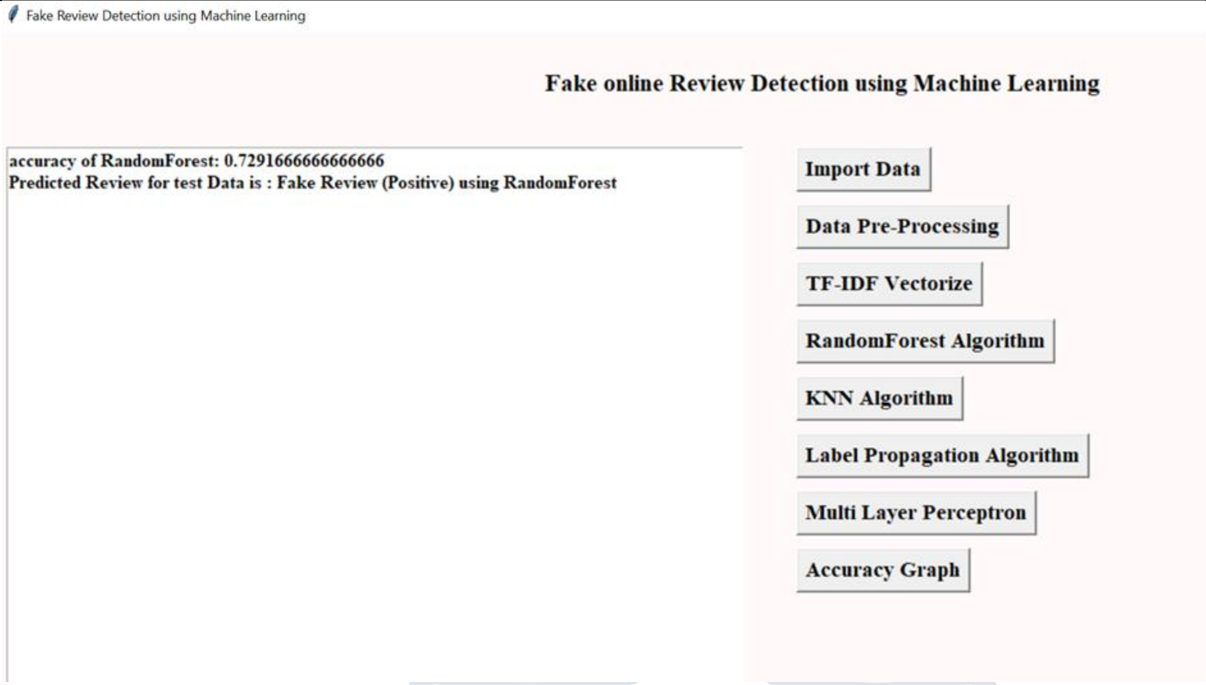

Figure 4.4:RFC Output

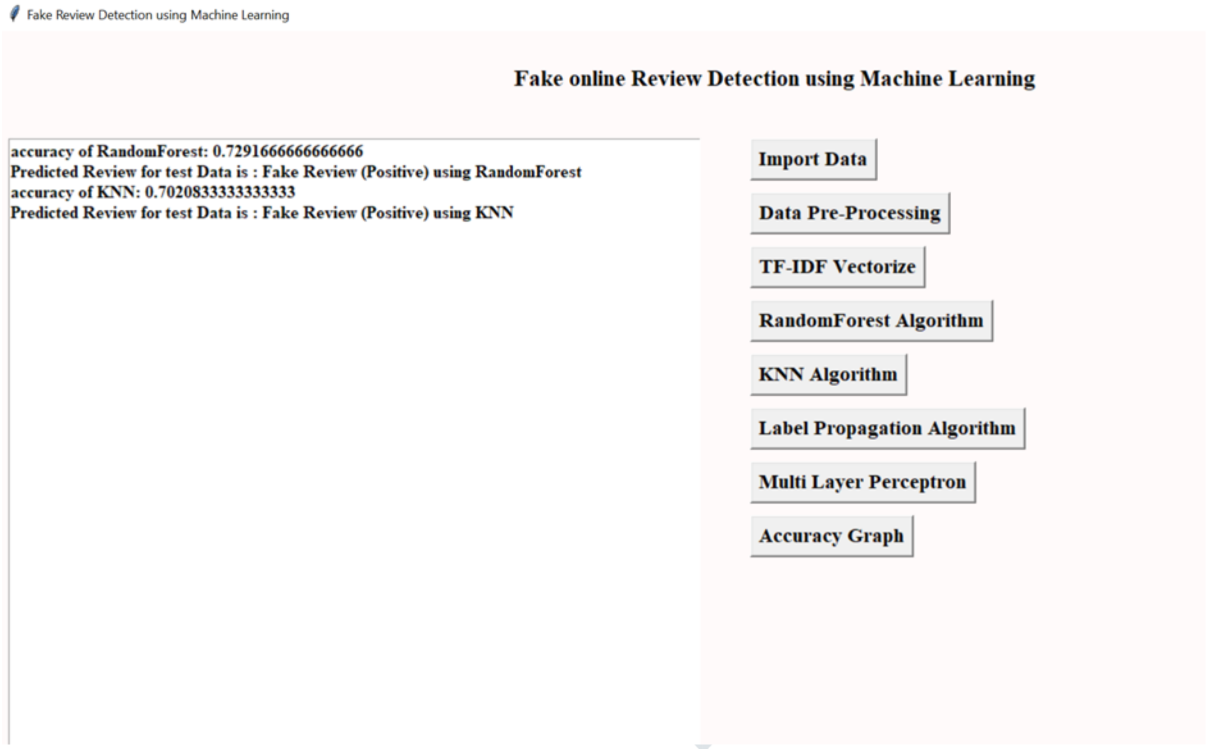

Figure 4.5: KNN Output

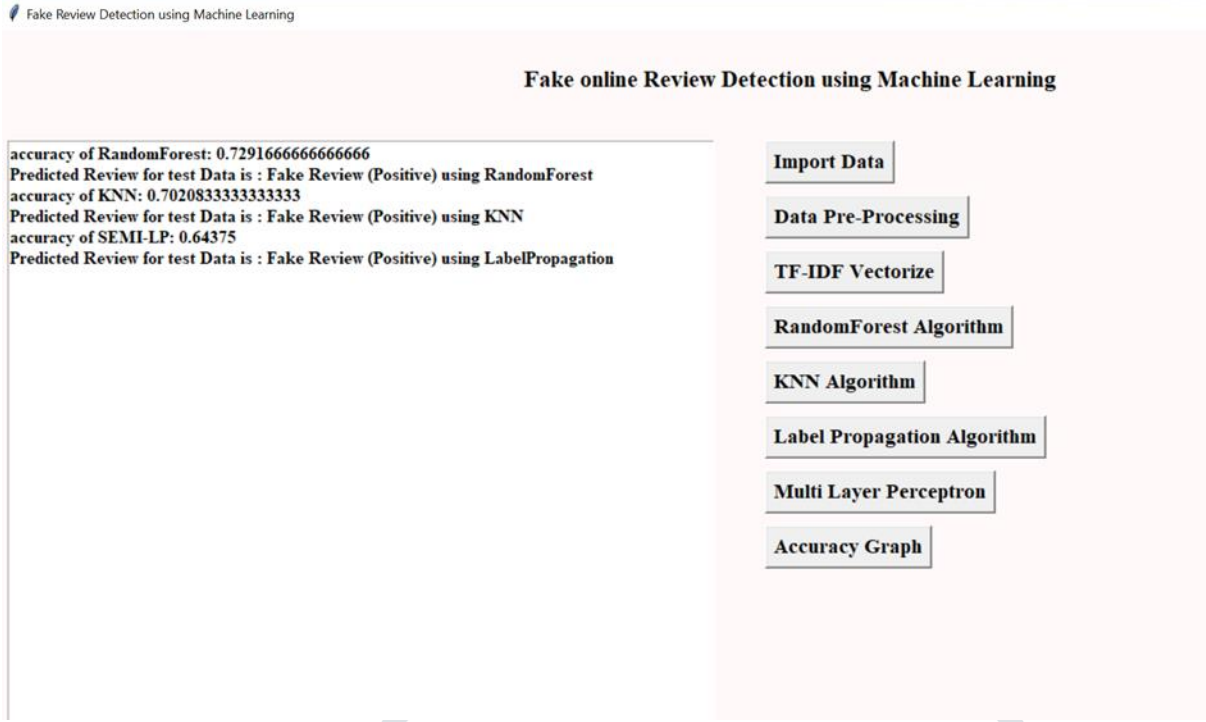

Figure 4.6: Label Propagation Output

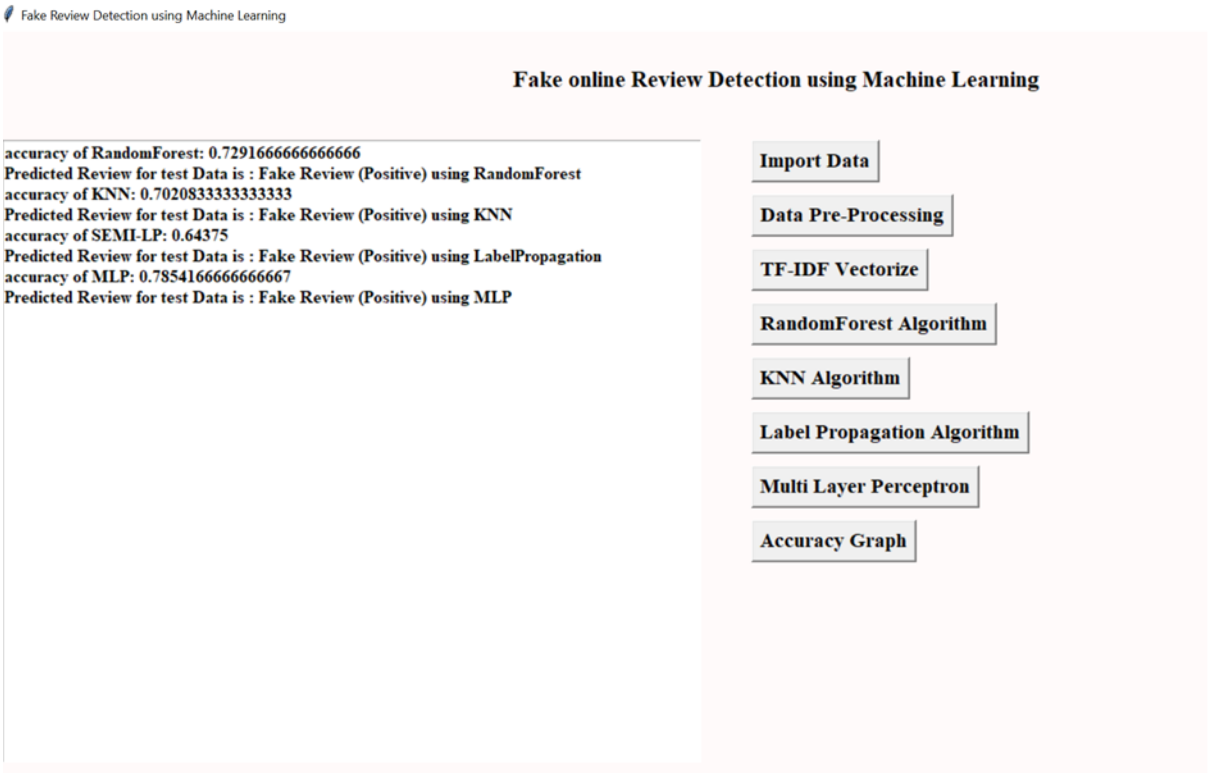

Figure 4.7: MLP Output

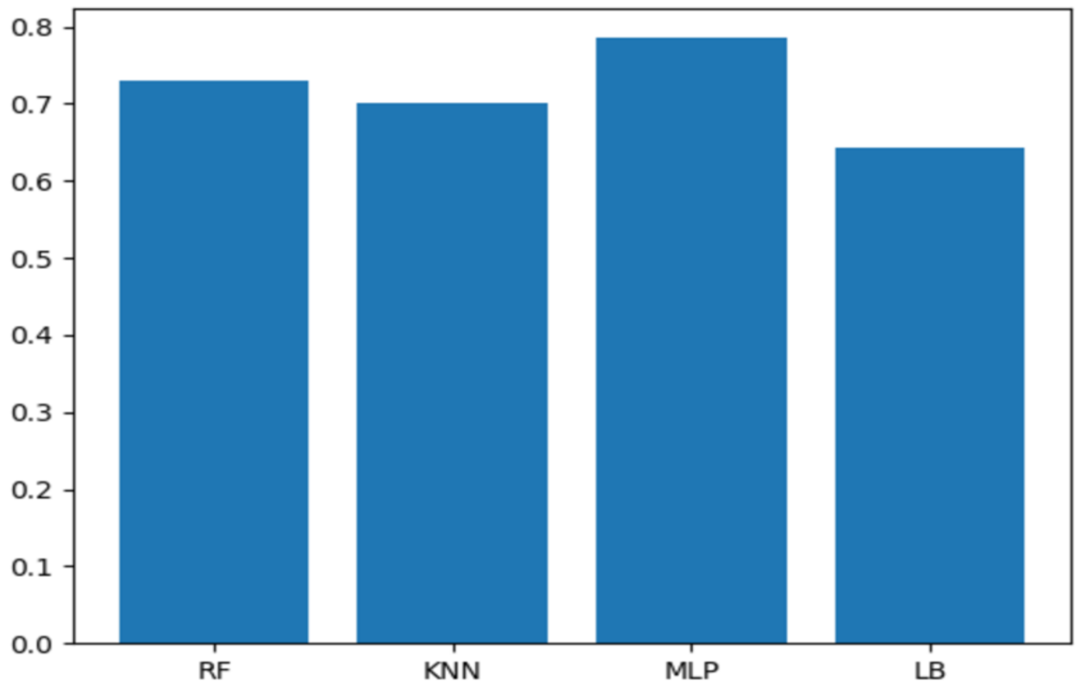

Figure 4.8: Graph Output

In the above figure we compare the accuracies of different algorithms. The accuracy of RFC is 72.9%. The accuracy of KNN is 70.2%. The accuracy of Label Propagation is 64.4%. The accuracy of MLP is 78.5%. In this project we got highest accuracy for MLP algorithm.

V.CONCLUSION AND FUTURE SCOPE

This study was conducted to predict the fake online reviews using different algorithms and comparing their accuracies. In this study we not only predicted if the review is fake or true but also whether the review is positive or negative. In this study we achieved the highest accuracy of 78% by using MLP classifier.

In future we would like to implement this model in real time by taking the live data from online. We would also be optimizing the model and increase the accuracy of the model. Advanced pre-processing tools for tokenization can be used to make the dataset more precise. Evaluation of the effectiveness of the proposed methodology can be done for a larger data set. This research work is being done only for English reviews. It can be done for several other languages. Future work may consider including other behavioral features such as features that depend on the frequent times the reviewers do the reviews, the time reviewers take to complete reviews, and how frequent they are submitting positive or negative reviews. It is highly expected that considering more behavioral features will enhance the performance of the presented fake reviews detection approach.

VI.REFERENCES

[1] Chengai Sun, Qiaolin Du and Gang Tian, "Exploiting Product Related Review Features for Fake Review Detection," Mathematical Problems in Engineering, 2016.
[2] A. Heydari, M. A. Tavakoli, N. Salim, and Z. Heydari, "Detection of review spam: a survey", Expert Systems with Applications, vol. 42, no. 7, pp. 3634–3642, 2015.
[3] M. Ott, Y. Choi, C. Cardie, and J. T. Hancock, "Finding deceptive opinion spam by any stretch of the imagination," in Proceedings of the 49th Annual Meeting of the Association for Computational Linguis- tics: Human Language Technologies (ACL-HLT), vol. 1, pp. 309–319, Association for Computational Linguistics, Portland, Ore, USA, June 2011.
[4] J. W. Pennebaker, M. E. Francis, and R. J. Booth, "Linguistic Inquiry and Word Count: Liwc," vol. 71, 2001.
[5] S. Feng, R. Banerjee, and Y. Choi, "Syntactic stylometry for deception detection," in Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Short Papers, Vol. 2, 2012.
[6] J. Li, M. Ott, C. Cardie, and E. Hovy, "Towards a general rule for identifying deceptive opinion spam," in Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (ACL), 2014.
[7] E.P.Lim,V.-A.Nguyen,N.Jindal,B.Liu,andH.W.Lauw,"Detecting product review spammers using rating behaviors," in Proceedings of the 19th ACM International Conference on Information and Knowledge Management (CIKM), 2010.

[8] J. K. Rout, A. Dalmia, and K.-K. R. Choo, “Revisiting semi-supervised learning for online deceptive review detection,” IEEE Access, Vol. 5, pp. 1319–1327, 2017.

[9].M V Bhanu Prakash U Sairam, Feature Prospect of the VAST Applications of Machine Learning, Research Review international Journal of Multidisciplinary,volume 4 and issue 4 in April 2019.

[10] Y. Adepu, V. R. Boga and S. U, "Interviewee Performance Analyzer Using Facial Emotion Recognition and Speech Fluency Recognition," 2020 IEEE International Conference for Innovation in Technology (INOCON), 2020, pp. 1-5, doi: 10.1109/INOCON50539.2020.9298427.

[11] V. Kunta, C. Tuniki and U. Sairam, "Multi-Functional Blind Stick for Visually Impaired People," 2020 5th International Conference on Communication andElectronics Systems (ICCES), 2020, pp. 895-899, doi: 10.1109/ICCES48766.2020.9137870.

[12] Santhosh Voruganti Enhanced Rating Prediction Based On Location And Friend Set published in JETIR May 2019 volume 6 issue 5 ISSN-2349-5162.

[13] Santhosh Voruganti Local Security Enhancement and Intrusion Prevention in Android Devices published in International Research Journal of Engineering and Technology Volume: 07 Issue: 01 January 2020 e-ISSN: 2395-0056 p-ISSN: 2395-0072.

[14] Santhosh Voruganti Map reduce A programming model for cloud computing based on hadoop ecosystem published in International Journal of Computer Science and Information Technologies, Vol. 5 (3), 2014, 3794-3799.

[15] Santhosh Voruganti Survey on Data-intensive Applications, Tools and Techniques for Mining Unstructured Data. International Journal of Computer Applications (0975 – 8887), volume 146-No.12, July 2016.

[16] Santhosh Voruganti Comparative Analysis of Dimensionality Reduction Techniques for Machine Learning IJSRST Volume 4 Issue 8 Print ISSN: 2395-6011Online ISSN: 2395-602X Themed Section: Science and Technology June 2018.

[17] Santhosh Voruganti EFFECTIVE IOT TECHNIQUES TO MONITOR THE LEVELS OF GARBAGE IN SMART DUSTBINS published in International Research Journal of Engineering and Technology Volume: 07 Issue: 06 June 2020 e-ISSN: 2395-0056 p-ISSN: 2395-0072.

[18] U.Sairam, Santhosh Voruganti, M V Bhanu Prakash, R Govardhan Reddy ,A Study on IoT Applications Towards Impact of Loss of Data, Proceedings of the Fifth International Conference on Trends in Electronics and Informatics (ICOEI). IEEE Xplore Part Number:CFP21J32-ART; ISBN:978-1-6654-1571-2/21,DOI:10.1109/ICOEI51242.2021.9452935.

[19] Santhosh Voruganti, U. Sairam,Breast Cancer Prediction using CNN and Machine Learning Algorithms with Comparative Analysis International Journal for Research in Applied Science & Engineering Technology (IJRASET) ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429 Volume 9 Issue VI Jun 2021.

[20] Santhosh Voruganti, U. Sairam Digital Image Watermarking using Chaotic Encryption and Arnold Transform , International Journal for Research in AppliedScience & Engineering Technology (IJRASET) ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429 Volume 9 Issue VI Jun 2021.

[21] Santhosh Voruganti, U Sairam, S Meghana , M Sravanthi, Visual Question Answering with External Knowledge International Conference on Smart Data Intelligence(ICSMDI 2021), Electronic copy available at: https://ssrn.com/abstract=3853031

[22] U. Sairam, Santhosh Voruganti, Mental Health Prediction Using Deep Learning, International Journal for Research in Applied Science & Engineering Technology (IJRASET) ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.538,Volume 10 Issue II Feb 2022- Available at www.ijraset.com