**TITLE:** FRAUD DETECTION BY USING TEXT CLASSIFICATION

**Team Members**

1. Venkata Mani Sahith Srungarapu
2. Lokesh Naidu Bavigadda
3. Srikanth Karni
4. Pasumarthi Raghu Ram

**Motivation**

The increasing popularity of online shopping has led to a rise in fraudulent activities such as fake reviews, which can mislead customers and harm the reputation of businesses. Detecting fraudulent reviews manually is time taking and a difficult task, importantly for businesses with a large reviews count. Therefore, developing an automated system to detect fraudulent reviews can be a valuable tool for businesses to maintain the integrity of their reviews and protect their reputation.

**Objective**

* To create a system that can properly identify reviews that are phony or fraudulent, as doing so can assist to increase the credibility of reviews.
* Businesses may safeguard their reputations and keep customers' trust by recognizing and eliminating bogus or fraudulent reviews.
* Manually spotting fake reviews is a difficult and time-consuming task. By automatically identifying bogus reviews, an automated system can help organizations save time and resources.
* Businesses can enhance their general performance and boost client loyalty by delivering honest reviews and safeguarding their reputation.

**Significant**

1.The capacity of a text classification-based fraud detection project to precisely identify fraudulent emails, transactions, and messages by examining their content sets it apart from other programs on the market.

2. Compared to rival initiatives, this one has a number of advantages, including flexibility to meet business objectives, real-time detection to allow prompt action, and the use of hybrid models, which mix several machine learning techniques and data sources to provide a more potent system.

3. In addition, the results of this project can be explained, resulting in more system transparency and trust.

4. The project is a useful tool for companies looking to combat fraud because of its capacity to adapt to new fraud types and continuously enhance its performance utilizing active learning and transfer learning methodologies.

**Related Work**

* To look into books, articles, and papers about detecting fake reviews. This step will assist us in comprehending the cutting-edge strategies and tactics applied in the industry.
* we are research so many online websites and the books some of the references are mentioned below.
* We're looking into the pertinent data sources available on various internet distribution channels, like e-commerce websites, social media networks, and review websites.
* Utilize a variety of metrics to assess the model's performance, including precision, recall, F1-score, and ROC-AUC. This phase is essential to ensuring that the model is precise and trustworthy in identifying fake reviews.
* we are looking into so many models, and we are selected the best models to get the perfect accuracy.

**Dataset**

The dataset consists of reviews from a popular e-commerce website and includes the review text, corresponding rating, and usefulness score. The review text, related rating, and usefulness score from reviews on a well-known e-commerce website were all included in the dataset utilized for this study.

user\_id: The user id consisting of the combination of character and the numbers, and it is the unique

Stars: the dataset consists of the stars which means the review of product defines from 1 to 5 and this is a number.

Review\_id: after giving the review the review id is generated which is also a unique, this is combination of the characters and the numbers.

date: the date is created this is the review date.

**Pre-processing**

Preprocessing In the preprocessing step, we perform the following operations on the text data: ● Remove HTML tags: We use BeautifulSoup library to remove HTML tags from the text data.

● Remove punctuation: We remove all the special characters and punctuations using regular expressions.

● Convert text to lowercase: We convert all the text data to lowercase so that the model treats "Great" and "great" as the same word.

● Remove stop words: We remove stop words from the text data. Stop words have examples such as "the", “and” "a", etc.

**Features and implementation**

In the feature engineering step, we create numerical features from the text data using CountVectorizer and TfidfVectorizer, which transforms the text data into a matrix of token counts. We saw that the results from TfidfVectorizer worked better and hence used it to convert the preprocessed text data into numerical features. The text input is transformed into a token count matrix by CountVectorizer, where each row was representing one review and with a column showing words from the dictionary. The value of each cell shows the count of a specific word.

TfidfVectorizer is a text feature extraction method that is similar to CountVectorizer, which converts text data into numerical features. However, unlike CountVectorizer, which simply counts the occurrence of each word in a document, TfidfVectorizer assigns a word-based weight based on its count and its inverse frequency in the whole dataset.

The term "tf-idf" stands for "term frequency-inverse document frequency". The "term frequency" component refers to the count of the word, while the "inverse document frequency" component refers to how rare the word is in the entire corpus of documents.

The TfidfVectorizer calculates the tf idf score for each word in each document, which is a measure of how important the word is to that document relative to the rest of the corpus.

**Model Selection and Evaluation**

In the model selection and evaluation step, we train and evaluate several machine learning models on the numerical features created by the TfidfVectorizer. We use various metrics to evaluate each model's performance, including F1 and accuracy. We also use a confusion matrix and ROC curve to visualize and compare the performance of the models.

The models we used in this project include:

● Support Vector Machine: A model that finds the hyperplane that maximally separates the data into different classes.

● Random Forest: A model based on using several decision trees to model the target variable. ● Logistic Regression: A model using a logistic function to calculate the target variable probability. We evaluate each model's performance based on a cross validation technique to avoid overfitting the model to the training data. Finally, we select the best performing model based on the evaluation metrics and test its performance on real-life reviews.

**Project Management**

**Workflow**

Implementation status report

Work completed: In progress.

Description:

* We start by performing some initial data cleaning, such as removing HTML tags, removing special characters and punctuation, and converting the text to lowercase. We also remove stop words to improve the efficiency of our model.
* We create numerical features from the text data using CountVectorizer and TfidfVectorizer, which transforms the text data into a matrix of token counts. We saw that the results from TfidfVectorizer worked better and hence used it.
* We train and evaluate several machine learning models, including Support Vector Machine, Logistic Regression, and Random Forest. We evaluate the models based on a variety of metrics, including accuracy and F1 score. We also use a confusion matrix and ROC curve to visualise and compare the performance of the models.
* Responsibility:
  + - Venkata Manisahith: Implementing machine learning techniques.
    - Lokesh Naidu Bavigadda: To review the code, implementing machine learning techniques.
    - Srikanth Karni: Research by using previous work, Analyze the features of the dataset we have taken.
    - Pasumarthi Raghu Ram: Analyze the features of the dataset we have taken, Background and related work, dataset.
* Contributions:
  + - Venkata Manisahith: 25%
    - Lokesh Naidu Bavigadda: 25%
    - Srikanth Karni: 25%
    - Pasumarthi Raghu Ram: 25%

Work to be completed:

Description: At this moment, we are doing these models: multinomial naïve Bayes, logistic regression, random forest, and we are research the other models to get the better accurate for the reviews and we try to improve our project more efficiently.

**Analysis and Preliminary Results**

Chart, bar chart

Description automatically generated

Multinomial Naive Bayes:

Accuracy: 0.7135

Recall: 1.0000

Precision: 0.7128

F1: 0.8323

Logistic Regression:

Accuracy: 0.7105

Recall: 0.9198

Precision: 0.7377

F1: 0.8188

Random Forest:

Accuracy: 0.7145

Recall: 0.9578

Precision: 0.7272

F1: 0.8267

**Testing on Real Life Data:**

**Text

Description automatically generated**

In conclusion, we have implemented a fraud detection system using text classification techniques. We started with preprocessing the data by removing stop words, performing stemming, and converting text to lowercase. We then used TfidfVectorizer to convert the preprocessed text into numerical features. Finally, we trained and evaluated several machines learning algorithmic models, including Support Vector Machines, Random Forest, and Logistic Regression to predict whether a review is fraudulent or not.

Our results show that Random Forest model has received the best accuracy, recall, precision, and f1 score among all the models. This suggests that it is a good choice for this particular problem.

Overall, our fraud detection system can be used to identify fraudulent reviews based on their text content. This can be helpful for businesses and consumers who rely on online reviews to make informed decisions.

**References/Bibliography**

* Zhang, X., Li, Q., Huang, X., & Wu, X. (2018). A deep learning approach for financial fraud detection based on textual data. Expert Systems with Applications, 107, 12-21.
* Abbasi, S. A., & Chen, H. (2008). Cybercrime classification: A motivational model. Communications of the ACM, 51(3), 88-93
* Wang, Z., Sun, Y., & Gao, X. (2020). A novel credit card fraud detection model based on word embedding and deep learning. Expert Systems with Applications, 156, 113397.
* Kumar, N., Agarwal, A., & Choudhary, A. (2020). Fraud Detection in E-commerce: A Comprehensive Review. Journal of Ambient Intelligence and Humanized Computing, 11(10), 4235-4261

**GitHub Link:** [**https://github.com/manisahith54321/Fraud-Detection-using-Text-Classification**](https://github.com/manisahith54321/Fraud-Detection-using-Text-Classification)