**Increament\_02**

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

**Team Members**

1. Venkata Manisahith
2. Lokesh Naidu Bavigadda
3. Srikanth Karni
4. Pasumarthi Raghu Ram

**Introduction:**

Here, in this project our aim is to design a model by using machine learning algorithms with the help of text content like customer reviews and ratings of the customers to accurately identify the fake reviews. As of now for this project, we have taken customer reviews and customer ratings from well-known websites and used them in our project’s data set.

After the confirmation of dataset, we have done required pre-processing activities so that model can easily do it’s process accurately in less time, pre-processing techniques involves removing stop words from reviews and changing into lower case letters.

Immediately after pre-processing of data we converted the reviews and ratings into numerical features by using the vectorization technique done. Here for this we have used TfidfVectorizer model. Then we focused on our goal to determine whether the review is fraudulent or not, to do this we have used decision trees, logistic regression, and random forests machine learning.

To decide the best model among those three we have taken some metrics like accuracy, recall, precision and F1 score.

Those above metrics are used to determine the best model to detect the fraud reviews, with addition to that metrics we have drawn ROC curve for each model and also confusion matrix to compare the best among the above mentioned three machine learning models.

This project is used to protect the belief of the people those who trust online reviews to trust many places, products and brands without directly going to them to save time by detecting the fake reviews and to make the people experience that reviews posted in online are worthy as mentioned in online.

**Background**

**o Related work for your topic with linked references:**

**1.**The project named as **FRAUD-DETECTION USING TEXT MINING,** it is done by Shashank Chinchilla, it detects the frauds in electronic payment system using machine learning libraries like scikit-learn and NLTK along text mining techniques.

GIT HUB LINK: <https://github.com/shashankchinchli/Fraud-Detection-using-Text-Mining>

2.The project named as **ENSEMBLE METHOD FOR FRAUD DETECTION,** it is done by Pavan Kumar.

Here, it is used to detect the frauds in financial statements. Here, this code uses Naïve Bayes and SVM algorithms to design the project.

GIT HUB LINK: <https://github.com/pavankumarallu/Ensemble-Method-for-Fraud-Detection>

3.The project named as **DETECTING INSIDER TRADING USING TEXT CLASSIFICATION,** it done by Rachel Koh it is used to detect frauds in trading using topic modelling and sentiment analysis techniques.

GIT HUB LINK: <https://github.com/rachelkoh/Detecting-Insider-Trading-Using-Text-Classification>

Overall, these studies highlight the potential of using machine learning and deep learning techniques in detecting various types of fraud, including financial fraud and cybercrime. These models can help organizations and law enforcement agencies prevent and detect fraudulent activities, ultimately improving security and safety for individuals and businesses.

**References**

1. 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.
2. Abbasi, S. A., & Chen, H. (2008). Cybercrime classification: A motivational model. Communications of the ACM, 51(3), 88-93
3. 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

**Architecture diagram:**

**+----------------------+**

**| Data Preprocessing |**

**+----------------------+**

**|**

**v**

**+----------------------+----------------------+**

**| Feature Engineering / Feature Selection |**

**+----------------------+----------------------+**

**|**

**v**

**+------------------------+-----------------------+**

**| Model Training / Hyperparameter Tuning |**

**+------------------------+-----------------------+**

**|**

**v**

**+------------------------+**

**| Model Evaluation |**

**+------------------------+**

**|**

**v**

**+----------------------+----------------------+**

**| Model Deployment |**

**+----------------------+----------------------+**

**|**

**v**

**+------------------------+**

**| Output |**

**+------------------------+**

**Explanation:**

The many steps of our model development pipeline are represented in the architectural diagram.

Data preparation is the first step, during which the raw data is cleaned, normalized, and converted into a structured format appropriate for additional analysis.

The following stage involves selecting the most pertinent features for our model and performing feature engineering.

To improve the model's performance, the model training stage entails choosing the right machine learning algorithms and hyperparameter tuning.

The model evaluation stage evaluates the model's performance on a test dataset and identifies any potential improvement areas.

The model can be used to identify fake reviews in real-time after it has been trained and evaluated.

Finally, the model's output can be applied to mark doubtful reviews for investigation.

**Workflow Diagram:**

+---------------------+

| Input: Reviews |

+---------------------+

|

v

+------------------+

| Data Preprocessing |

+------------------+

|

v

+------------------------+

| Feature Engineering / Feature Selection |

+------------------------+

|

v

+--------------------------+

| Model Training / Hyperparameter Tuning |

+--------------------------+

|

v

+------------------+

| Model Evaluation |

+------------------+

|

v

+----------------+

| Model Deployment |

+----------------+

|

v

+--------------+

| Output: Fraudulent Reviews |

**Explanation:**

The workflow diagram illustrates the steps involved in applying our approach to find fake reviews.

A set of reviews is the model's input.

The data from the reviews is first cleaned and normalized during preprocessing.

To find the most pertinent features for the model, feature engineering and feature selection are carried out.

Hyperparameters are adjusted to maximize performance once the model has been trained using machine learning methods.

To make sure the model is accurately recognizing bogus reviews, it is tested on a test dataset.

The model can be used to identify fake reviews in real-time after it has been trained and evaluated.

A set of reviews marked as possibly fake for additional investigation is the model's output.

**Dataset:**

The dataset used in this project is a collection of reviews from various e-commerce websites. The dataset contains both genuine and fake reviews. here we are using the 20% of the data for the test part and the 80% percentage of the data for the training part.

The following features were used in the project:

1. Date: This feature contains date of the review.

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

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

4. 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.

5. text: here the actual review of the product and this is a text data.

6. type: here the text type is review as default.

here the dataset contains the 10000 reviews with each different review\_id, user\_id. The review text, rating, reviewer's profile, review date, and product information are used as input features to the machine learning model. The model analyzes these features to determine whether the review is genuine or fake.

The diagram below shows the design of features used in the project:

+----------------------+

| Review Text |

+----------------------+

| Rating (0 to 5) |

+----------------------+

| Reviewer's Profile |

+----------------------+

| Review Date |

+----------------------+

| Product Information |

+----------------------+

**Graph Model:**

Based on their accuracy, recall, precision, and F1 score, the three models are contrasted in this graph. The various measurements are displayed on the x-axis, while the score value is displayed on the y-axis. The Multinomial Naive Bayes model is represented by the blue bars, the Logistic Regression model by the orange bars, and the Random Forest model by the green bars.

The Multinomial Naive Bayes model, as seen in the graph, has the best recall score, which indicates that it has the fewest false negatives, but the lowest accuracy score, which indicates that it has the greatest number of false positives. The Random Forest model has the lowest number of false positives and the highest precision score, but it has a slightly lower recall score compared to the other models. an analysis of the Multinomial Naive Bayes model. In terms of precision and recall scores, the Logistic Regression model lies in between the other two models.

Overall, the graph demonstrates that while the accuracy and F1 scores of all three models are comparable, their performance in terms of precision and recall varies. Which model to use ultimately comes down to the requirements and objectives of the project.

Chart, bar chart

Description automatically generated

**Algorithms / Pseudocode:**

The code employs the Multinomial Naive Bayes, Logistic Regression, and Random Forest machine learning techniques. The brief descriptions of each algorithm and its pseudocode are as follows:

Naive Multinomial Bayes:

Naive Bayes is a probabilistic algorithm that determines the likelihood of each class given the features by applying Bayes' theorem.

A variation of Naive Bayes that performs well with discrete characteristics, like word counts, is multinomial Naive Bayes.

Using the Bayes theorem, the algorithm determines the likelihood of each feature given each class before determining the probability of each class given the features.

The output is then predicted to be the class with the highest likelihood.

Pseudocode:

Every class i:

Calculate P(C=i), the prior probability.

Determine the probability P(X|C=i) for each X feature.

Each fresh occurrence of X:

Using Bayes' theorem, determine the probability for each class P(C=i|X).

Pick the class that has the best chance of happening.

Logistic Regression:

Logistic Regression is a linear algorithm that works well with binary classification problems.

The algorithm works by fitting a logistic function to the data, which maps the features to a probability value between 0 and 1.

The logistic function is then used to predict the class of each instance.

Pseudocode:

Initialize the model parameters (weights) to random values.

For each iteration:

Calculate the predicted probability for each instance using the logistic function.

Calculate the error between the predicted probability and the actual class.

Update the weights using gradient descent to minimize the error.

Predict the class of each instance using the logistic function.

Random Forest:

The technique generates predictions by merging the outputs of several decision trees that have been trained on random subsets of the data.

To further avoid overfitting, each tree is trained using a subset of the features.

Pseudocode:

Each tree in i:

Choose a random subset of both the data and the features.

Train a decision tree using the features and data you've chosen.

Each fresh occurrence of X:

Determine each tree's class by using X.

Make a final prediction by combining the tree results (for example, by majority voting).

**Explanation of Implementation:**

The implementation is divided into three parts:

A. Data Pre-processing

B. Training and evaluation of model

C. Deployment of model

A) Pre-processing of data: Here we have used the data set from famous customer reviews website called as yelp review website to create a data frame using the pandas library

2. After we have added an extra column in that data frame and named it as FAULT to confirm whether the review is fault or not.

3.The text in the customer’s review is cleaned by using regex library wordnet lemmatizer to remove non- alphabetic words and converting into lower case, removing stop words also lemmatizing the words using lemmatizers.

B) Training the machine learning models and evaluating them:

1.At first we have to create intuition function for the three models we have taken to drive them

2.Next, training the three models, before we have to split our data set as two parts for training and testing then we have to train the models using that training dataset.

3.Next we have test the model using the test data set.

4.After that we have to evaluate the model based on metrics mentioned above.

C) Model deployment:

1. After checking the evaluation metrics we have got the conclusion that Random forest model is the best to detect the fraud in reviews

A function is decided to proceed to the next step.

**Results**

Diagrams for results with detailed explanation

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

Based on the evaluation metrics provided, the Multinomial Naive Bayes model has the highest accuracy (71.35%), while the Logistic Regression model has the highest recall (91.98%). Precision is highest for the Logistic Regression model (73.77%), and the F1 score is highest for the Multinomial Naive Bayes model (83.23%).

It's important to note that the choice of which evaluation metric to prioritize depends on the specific use case and the relative importance of correctly identifying true positives (recall) versus avoiding false positives (precision). The F1 score is a balanced measure that takes both recall and precision into account.

Overall, it seems like all three models have similar performance, and the choice of which one to use may depend on other factors such as interpretability, computational efficiency, and ease of implementation.

**Project Management:**

Implementation status report

Work completed: completed.

Description:

* After completing the increment 1 we are trying to perform the different model as we mentioned in the increment 1 but the result is satisfied with the models, we performed we got the best accuracy level for this project. we are attached the results screenshot in the result section.
* we referred some of the books and the online as I mentioned in the references for the models.
* we distributed the tasks equally and we discuss the machine learning techniques models so we selected these models as I mentioned in the model selection and evaluation selection in the increment 1.

Responsibility (Task, Person)

* + - Venkata Manisahith: Implementing machine learning techniques, documentation.
    - Lokesh Naidu Bavigadda: To review the code, implementing machine learning techniques, documentation.
    - Srikanth Karni: Research by using previous work, Analyze the features of the dataset we have taken, project management.
    - Pasumarthi Raghu Ram: Analyze the features of the dataset we have taken, Background and related work, dataset.

Contributions (members/percentage)

* + - Venkata Manisahith: 25%
    - Lokesh Naidu Bavigadda: 25%
    - Srikanth Karni: 25%
    - Pasumarthi Raghu Ram: 25%

Issues/Concerns:

Data completeness and quality: The accuracy of any machine learning model is greatly influenced by the completeness and quality of the training data. The performance of the model can be greatly impacted by data problems, such as missing values or erroneous labels.

Interpretability of the models is vital to take into account, even while the accuracy, recall, precision, and F1 scores are useful indicators of how effectively the models are working. Some models, like the random forest, are difficult to interpret and can make it challenging to comprehend how they made their predictions.

Generalizability: It's crucial to consider how well the model will function with brand-new, untested data. If a model is too closely fitted to It might not perform well in the actual world and might not generalize well to new data from the training data. Cross-validation helps lessen this problem, but it is still crucial to carefully assess how well the model performs on a holdout test set.

**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.

Link :[Reference](https://www.google.com/search?q=%E2%80%A2+Zhang%2C+X.%2C+Li%2C+Q.%2C+Huang%2C+X.%2C+%26+Wu%2C+X.+(2018).+A+deep+learning+approach+for+financial+fraud+detection+based+on+textual+data.+Expert+Systems+with+Applications%2C+107%2C+12-21.&rlz=1C5MACD_enUS1046US1047&oq=%E2%80%A2%09Zhang%2C+X.%2C+Li%2C+Q.%2C+Huang%2C+X.%2C+%26+Wu%2C+X.+(2018).+A+deep+learning+approach+for+financial+fraud+detection+based+on+textual+data.+Expert+Systems+with+Applications%2C+107%2C+12-21.&aqs=chrome.0.69i59.803j0j7&sourceid=chrome&ie=UTF-8)

* Abbasi, S. A., & Chen, H. (2008). Cybercrime classification: A motivational model. Communications of the ACM, 51(3), 88-93

Link: [Reference](https://www.google.com/search?q=%E2%80%A2+Abbasi%2C+S.+A.%2C+%26+Chen%2C+H.+(2008).+Cybercrime+classification%3A+A+motivational+model.+Communications+of+the+ACM%2C+51(3)%2C+88-93&rlz=1C5MACD_enUS1046US1047&oq=%E2%80%A2%09Abbasi%2C+S.+A.%2C+%26+Chen%2C+H.+(2008).+Cybercrime+classification%3A+A+motivational+model.+Communications+of+the+ACM%2C+51(3)%2C+88-93&aqs=chrome..69i57.2134j0j9&sourceid=chrome&ie=UTF-8)

* 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.

Link: [Reference](https://www.google.com/search?q=%E2%80%A2+Wang%2C+Z.%2C+Sun%2C+Y.%2C+%26+Gao%2C+X.+(2020).+A+novel+credit+card+fraud+detection+model+based+on+word+embedding+and+deep+learning.+Expert+Systems+with+Applications%2C+156%2C+113397.&rlz=1C5MACD_enUS1046US1047&oq=%E2%80%A2%09Wang%2C+Z.%2C+Sun%2C+Y.%2C+%26+Gao%2C+X.+(2020).+A+novel+credit+card+fraud+detection+model+based+on+word+embedding+and+deep+learning.+Expert+Systems+with+Applications%2C+156%2C+113397.&aqs=chrome..69i57.1865j0j9&sourceid=chrome&ie=UTF-8)

* Kumar, N., Agarwal, A., & Choudhary, A. (2020). Fraud Detection in E-commerce: A Comprehensive Review. Journal of Ambient Intelligence and Humanized Computing.

Link: [Reference](https://www.google.com/search?q=%E2%80%A2+Kumar%2C+N.%2C+Agarwal%2C+A.%2C+%26+Choudhary%2C+A.+(2020).+Fraud+Detection+in+E-commerce%3A+A+Comprehensive+Review.+Journal+of+Ambient+Intelligence+and+Humanized+Computing.&rlz=1C5MACD_enUS1046US1047&oq=%E2%80%A2%09Kumar%2C+N.%2C+Agarwal%2C+A.%2C+%26+Choudhary%2C+A.+(2020).+Fraud+Detection+in+E-commerce%3A+A+Comprehensive+Review.+Journal+of+Ambient+Intelligence+and+Humanized+Computing.&aqs=chrome..69i57.1813j0j9&sourceid=chrome&ie=UTF-8)

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