Consumer Complaints Classification

Name:	**Akanksha Singh**
Registration No./Roll No.:	19022
Institute/University Name:	IISER Bhopal
Program/Stream:	DSE
Problem Release date:	August 15, 2022
Date of Submission:	September 29, 2022

1 Introduction

Businesses can better understand consumer concerns and experiences by investing in customer assistance. Understanding your customers' complaints and treating them as extremely useful feedback to incorporate into your customer service plan in order to enhance the brand experience is the best method to guarantee your company's growth.

The data used for this project is collected from the Consumer Financial Protection Bureau (CFPB), which is a federal U.S. agency that acts as a mediator when disputes arise between financial institutions and consumers. The data was downloaded directly from the CFPB website for training and testing the model. It included one year's worth of data (March 2020 to March 2021). Retail banking, credit cards, debt collection, mortgages and loans, and credit reporting are the five categories for the complaints given. The training dataset categories are visualized in the bar graph (fig.1) and pie chart (fig.2) provided below. Word cloud shows the complaints for credit reporting (fig.3).

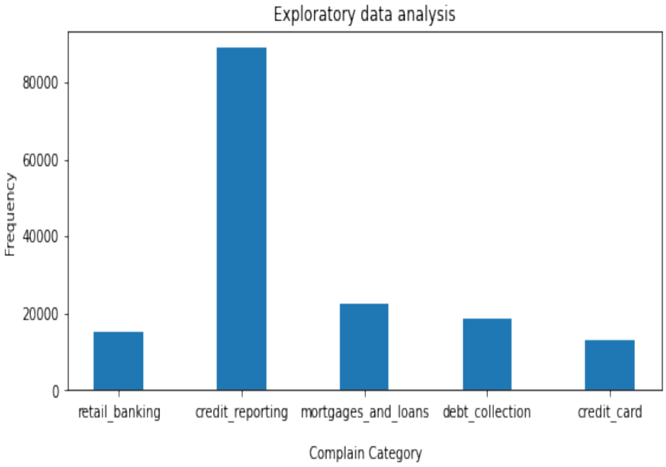

Figure 1: Overview of categories in training dataset in bar graph

2 Methods

2.1 Data Cleaning

The training data used for this project has 158360 rows and two features, 'Complaint' and 'Category'. The 'Complaint' contains all the complaints made by customers in the english language, whereas 'Category' represents predefined classes like retail banking, debt collection, etc. Since the 'Complaint'

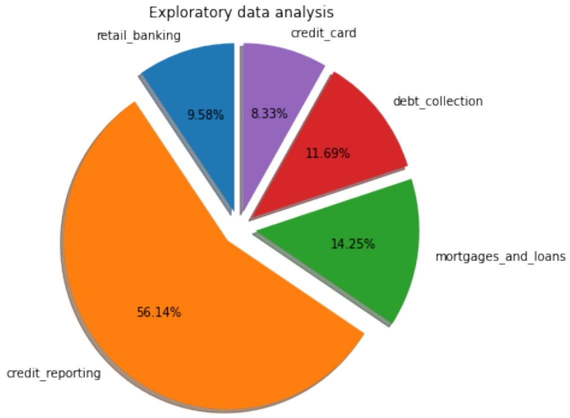

Figure 2: Overview of categories in training dataset in pie chart

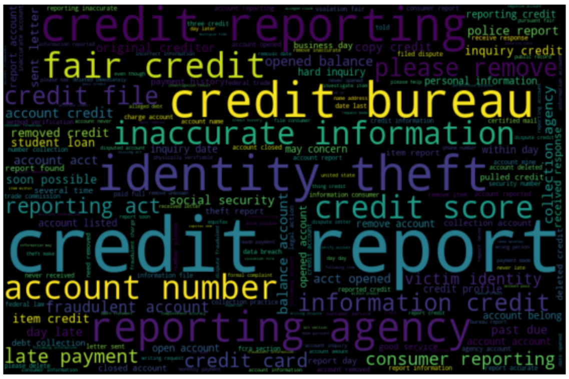

Figure 3: Word cloud of credit reporting complaints

section in the training data contains ten null values; therefore, the data is cleaned by eliminating those rows. It would not make a huge impact on the model training since the dataset is already large.

2.2 Text Pre-processing

(i) **Removal of Punctuation:** Punctuation is taken out of every text in 'Complaint' since it doesn't convey any important information and is irrelevant.

(ii) **Conversion to lowercase:** The unpunctuated data is then passed as an input to lowercase, which aided in pre-processing and later parsing stages of the NLP application.

(iii) **Removing stopwords:** Stopwords simply don't provide any useful information. Hence they are eliminated from the corpus.

(iv) **Tokenization:** We used sentence tokenization to break our data.

(v) **Stemming:** We employ stemming from reducing words to their basic form or stem, which may or may not be legitimate words in the language.

(vi) **Lemmatization:** Lemmatization, as compared to stemming, reduces the words to a single word from the language. This aids in giving the word its correct root form.

2.3 Data Manipulation

(i) **Feature Extraction:**

(a) **Bag of Words:** We used Countvectorizer for prediction with hyper-parameter tuning, which vectorizes text in 'Complaint' from the dataset based on the count of each word occurring.

Classifier	Precision	Recall	F1-score	support
credit_card	0.81	0.75	0.78	3035
credit_reporting	0.88	0.94	0.91	17779
debt_collection	0.79	0.67	0.72	4514
mortgages_and_loans	0.86	0.80	0.83	3703
retail_banking	0.87	0.86	0.86	2639
accuracy			0.86	31670
macro avg	0.84	0.80	0.82	31670
weighted avg	0.86	0.86	0.86	31670

Table 1: Performance Of Logistic Regression

The parameters used are min_df, max_df, and n-gram range set to (1,3), which denotes unigram, bigram, and trigram.

(b) **TF-IDF:** TF-IDF (term frequency and inverse term frequency) TfidfVectorizer vectorizes 'Complaint' based on frequency of words. We used TfidfVectorizer for predictions in two cases, with and without hyperparameter tuning.

For both bag of words and TF-IDF, parameters min_df is set to 5, i.e., a bare minimum of documents that should include this functionality. Hence, we only list the words that appear in at least 5 documents. max_df feature is set to 0.7, i.e. we include only those words that appear in a maximum of 70% of the documents.

(ii) **Feature Selection:** Chi-square feature selection method was used and mutual information gain was discarded. The first trial iteration of the code with chi-square was done for top 10 features only, the second iteration with top 5000 features that we couldn't run till results. We expect the accuracy to increase with larger number of features as the vectorized data has more than 5 lacs of features. Mutual information gain threw errors of discrete, continuous data and class labels that we couldn't resolve.

2.4 Data Classification

We employed five classification models: Random Forest, Logistic, KNN Regression, SVM, and Multinomial Naive Bayes. With these models, we got various outcomes.

(i) **Hyper-parameter tuning:** We did hyper-parameter tuning for each of the five classifiers to get a set of optimal hyper-parameters for each classifier.

(ii) **Pipeline:** We used Pipeline for vectorization, feature selection, and classification.

(iii) **GridSearchCV:** This allow us to pass our specific estimator, our grid of parameters, and our chosen number of cross validation folds. GridSearchCV permutes vectorizer, selector, and classifier and selects the best set. Ten-fold cross-validation was performed for each classifier.

3 Evaluation Criteria

For evaluation, we considered four criteria accuracy, precision, recall, and F1 score. We found that logistic regression works best among all the classifiers for all three cases namely, preprocessed data and hyper-parameter tuning, preprocessed data and no hyper-parameter tuning, unprocessed data and no hyper-parameter tuning. Classification report for using pre-processed data and hyper-parameter tuning is shown in table 1.

Classifier	Precision	Recall	F1-score	support
credit_card	0.79	0.77	0.78	842
credit_reporting	0.90	0.93	0.91	5139
debt_collection	0.78	0.69	0.73	1281
mortgages_and_loans	0.83	0.81	0.82	995
retail_banking	0.86	0.87	0.86	767
accuracy			0.86	9024
macro avg	0.83	0.81	0.82	9024
weighted avg	0.86	0.86	0.86	9024

Table 2: Performance of model without text pre-processing

Classifier	Precision	Recall	F1-score	support
credit_card	0.79	0.77	0.78	842
credit_reporting	0.90	0.93	0.91	5139
debt_collection	0.78	0.69	0.74	1281
mortgages_and_loans	0.83	0.82	0.83	995
retail_banking	0.86	0.87	0.86	767
accuracy			0.86	9024
macro avg	0.83	0.82	0.82	9024
weighted avg	0.86	0.86	0.86	9024

Table 3: Performance of model without hyper-parameter

4 Analysis of Results

Logistic Regression turned out to be the best classifier for our dataset, with an accuracy 85.9% for pre-processed data and hyper-parameter tuning(table1). For each classifier, we contrasted the model without text pre-processing and without hyper-parameter. The best classifier found is Logistic Regression with 86.3% accuracy, and table 2. Additionally, we contrasted the model with text pre-processing and without modifying the hyper-parameters. The outcomes of the best classifier are logistic regression with text pre-processing and without hyper-parameter with an accuracy 86.4% are shown in Table 3. Overall, logistic regression is the best classifier for our text classification problem.

5 Discussions and Conclusion

We got the best results by using pre-processed text and without hyper-parameter tuning for logistic regression with accuracy 86.4%. Since 'Credit Reporting' has 56% weightage in the training data, therefore our model will be more biased towards it. To understand the true value that comes from complaints, we need to change the way we think about them. Given that there are some significant benefits to customer complaints, we should support them. They significantly affect the company. So, by identifying the advantages of complaints, you not only enhance the reputation of your brand but also boost team output. This project gave us the opportunity to learn linux, tmux, remote server access. It surely wasn't easy, but we are happy that we learnt new things on the way.

6 Contribution

I did data pre-processing, prediction with pre-processed data and hyper-parameter tuning, test labels predictions.

Directory path - `\DATA1\NLP\akanksha_19022\akanksha_19022(venv)`