Final Project Report

[Manish Patel April- 2024]

Automated Customer Complaint Classification

For financial companies, customer complaints can provide valuable insights into the areas where their products and services may be lacking. By responding to these complaints promptly and effectively, companies can mitigate customer dissatisfaction and foster greater loyalty. Additionally, addressing customer feedback can help companies to refine their offerings and appeal to new customers. Overall, prioritizing customer complaints is a key aspect of maintaining a successful financial business.

Customer complaints are usually unstructured text data, which traditionally require companies to assign multiple support employees to evaluate and allocate each ticket to the relevant department. As the company grows and acquires a larger customer base, this process becomes tedious. Therefore, the company aims to automate its customer support ticket system. Being a financial company, the firm offers numerous products and services, including credit cards, banking, and mortgages/loans.

Problem Statement :

The task is to build a model that can efficiently classify customer complaints based on the products and services offered by the company. These complaints are unstructured text data, and the company wants to automate the process of evaluating and assigning each ticket to the relevant department.

- Ticket Classification: Develop a model that can accurately categorize customer complaints into one of the following five clusters:
  - **Credit card / Prepaid card**
  - **Bank account services**
  - **Theft/Dispute reporting**
  - **Mortgages/loans**
  - **Others**
- Quick Resolution: By segregating the clusters, the company aims to expedite the resolution process for customer issues. Efficiently handling complaints can lead to higher customer satisfaction and stronger loyalty.
- Unlabeled Data: The provided data is in JSON format and lacks accurate labels. We need to apply unsupervised techniques to analyze patterns and create meaningful clusters.

The solution should enable the company to streamline its customer support ticket system, improve service quality, and enhance customer experience.

Data Wrangling :

We utilized a dataset named 'Automated Customer Complaint Classification,' which is accessible on Kaggle [Dataset] in json format. The dataset contains 78313 complaints, and each complaint contains 22 distinct features. These features include complaint Id, issue, date, company response, company, complaint what happened, sub-issue, and several others. Initially, we categorized each feature as categorical or numerical to establish the domain of each variable. Out of the 22 features, 21 were categorical variables, while one was a numerical variable. We found that feature names begin with '-' and 'source'. We removed 'source' without any impact on their functionality, to make the names easier to read and use.

To achieve our desired outcome, we need to extract information from the 'complaint_what_happened' column. This column likely contains feedback or reply provided by users or customers. By analyzing this data, we can gather insights and draw conclusions that can help us make informed decisions. Therefore, it is crucial that we effectively obtain and interpret the data from this column.

Missing values:-

As part of our thorough analysis of the dataset, we conducted a detailed examination of the 'complaint_what_happened' column. We found that this column did not have any instances of null values, indicating that all the cells in the column contained some form of data. However, upon closer inspection, we discovered that the column did contain occurrences of empty strings. In order to maintain the integrity and consistency of the data, we have made the decision to remove any row that contains an empty string in the 'complaint_what_happened' column. This will allow us to ensure that only complete and accurate data is retained in the dataset, which will in turn enable us to generate reliable insights and make informed decisions based on the data. As a result, the dataset has been reduced in size and we now have a total of 21072 rows and 22 columns to work with. This adjustment will allow us to move forward with greater accuracy and confidence in our analysis.

Prepare the text for topic modeling:-

Once we have removed all the blank complaints, we need to:

- Make the text lowercase
- Remove text in square brackets
- Remove punctuation
- Remove words containing numbers

Once we have done these cleaning operations we need to perform the following:

- Lemmatize the texts - Lemmatization reduces words to their base or dictionary form (e.g., “running” becomes “run,” “better” becomes “good”).
- Use POS tags to get relevant words from the texts.

The graph below(figure-1) displays how long each complaint is in terms of character count.

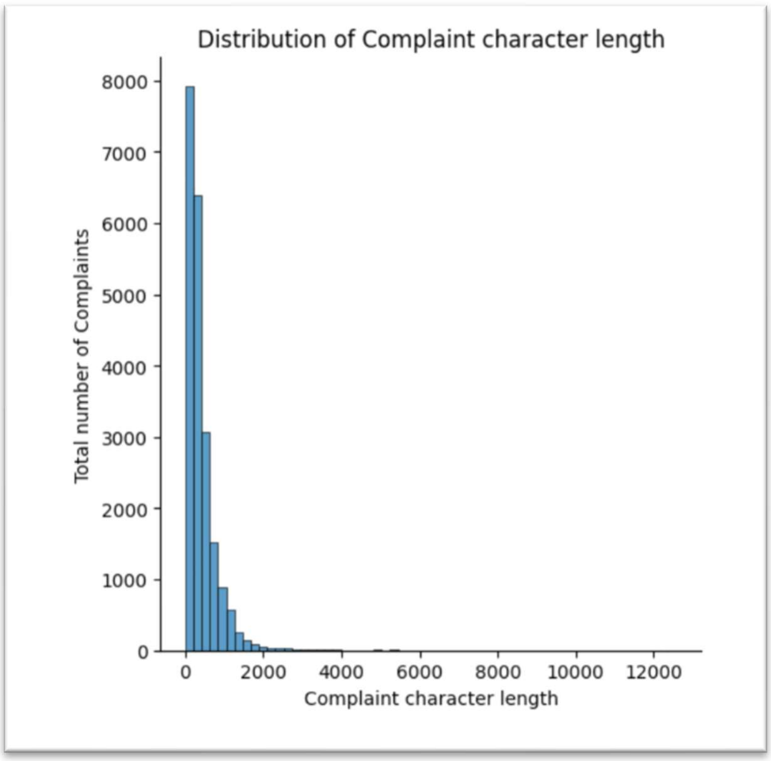


Figure 1

Using a word cloud found the top 60 words(figure-2) by frequency among all the articles after processing the text.

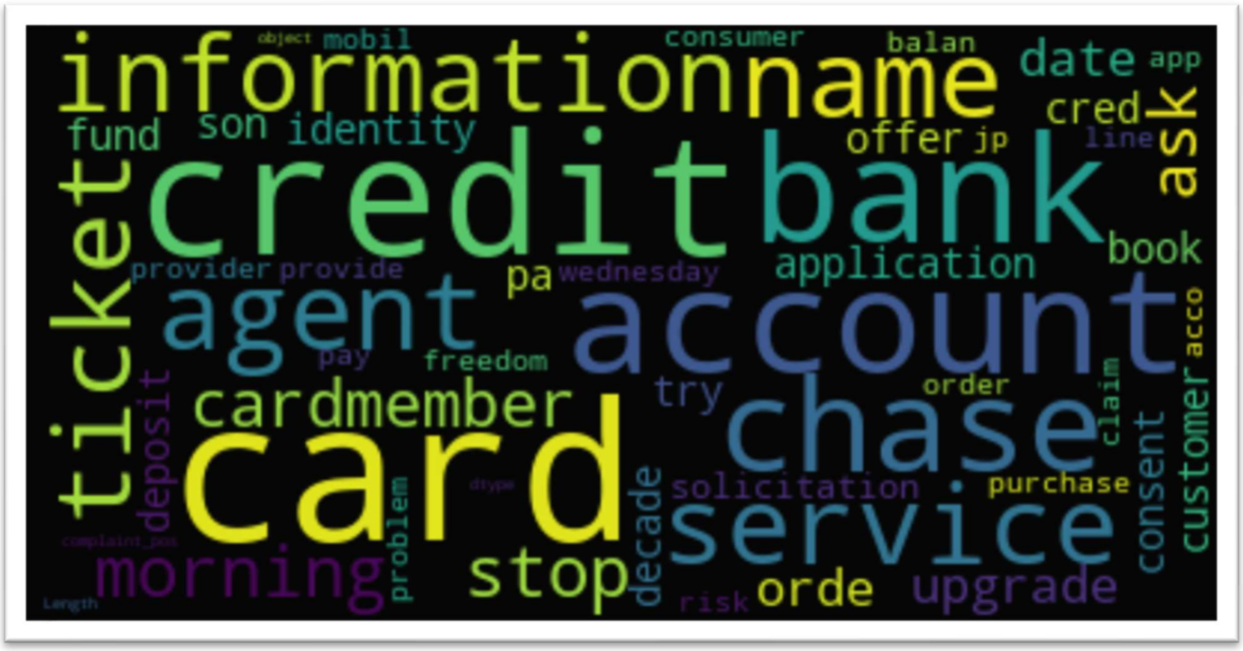
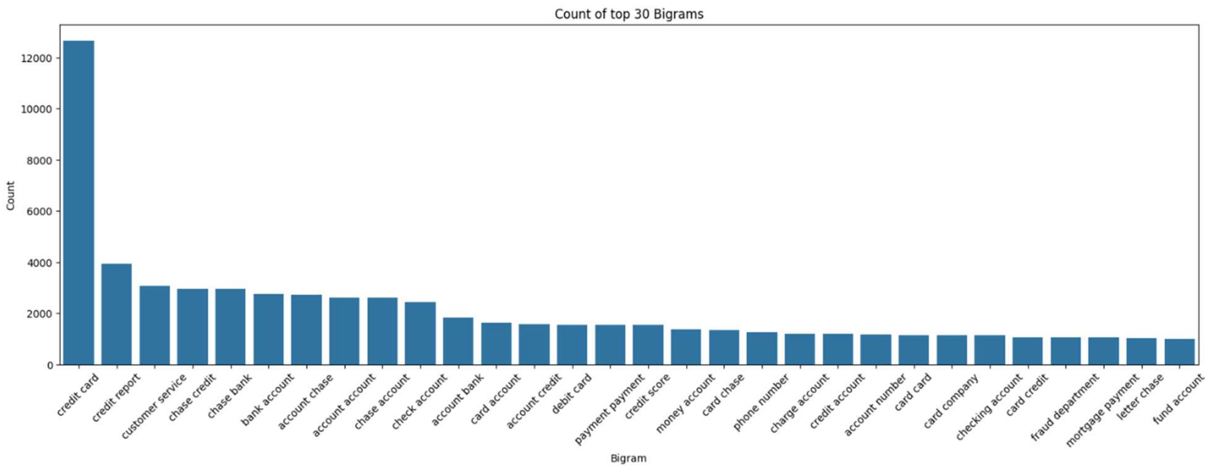
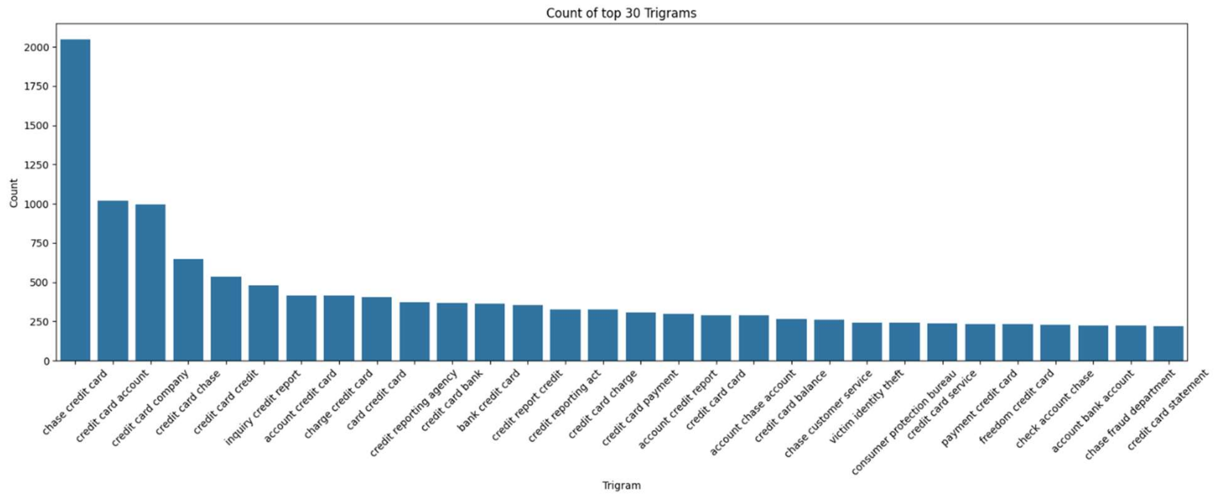


Figure 2

Bigram and Trigram:

Using bigram and trigram analysis techniques, we have conducted a comprehensive examination of the data at hand. As a result of this analysis, we have been able to identify the frequency with which specific words are used in conjunction with others. Our findings are presented in *figure-3*, which provides a clear and concise representation of the results of our research.





Topic Modeling:

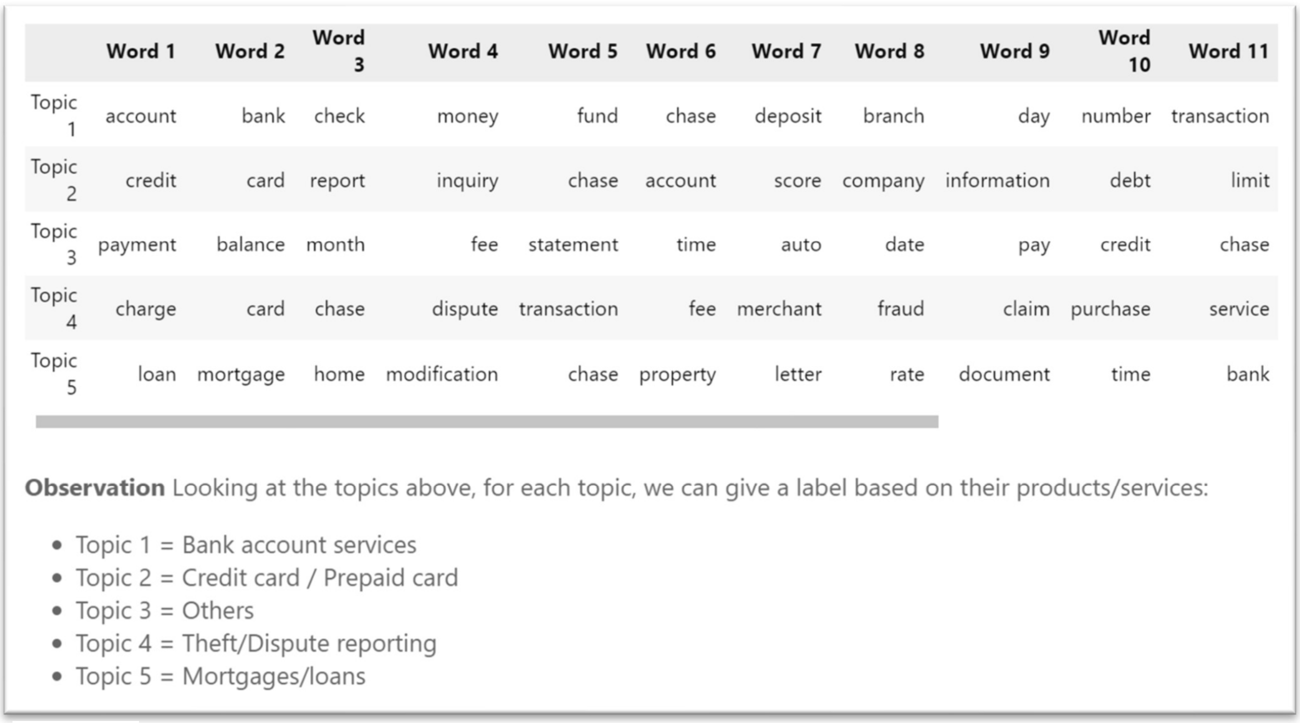
As we continue our project, we are currently executing a series of steps to ensure its successful completion. These steps include various tasks and processes that we need to undertake in order to achieve our desired outcome. We are carefully and methodically working through each step, paying close attention to detail and ensuring that we are meeting all of the necessary requirements. Our ultimate goal is to deliver a high-quality result that exceeds expectations.

- **TFIDF**:- During the preprocessing step, we applied a technique called TFIDF (Term Frequency-Inverse Document Frequency) on our feature column. TFIDF is a powerful incantation from the orbit of Natural Language Processing (NLP) that helps in determining the importance of a word in a document. This technique creates a magical weight for each word by taking into account its local importance within a document as well as its global rarity across all documents in the corpus. The weight assigned to each word is proportional to its frequency in the current document and inversely proportional to the number of documents in the corpus that contain that word. In other words, the more frequently a word appears in a document and the rarer it is across all documents, the higher its TFIDF weight will be. This approach is widely used in text mining, information retrieval, and other NLP tasks to identify the most important words in a document or corpus.

- **DTM:-** The Document-Term Matrix (DTM) is a powerful tool used in natural language processing that provides a detailed representation of the frequency of terms or words in each document within a collection or corpus. In other words, it is a matrix in which the rows represent the individual documents and the columns represent the terms or words found within those documents. Each cell in the matrix holds the count of how many times a specific term appears in a specific document. This matrix can be used to extract valuable insights from a large corpus of text, such as identifying common themes or topics, and can be further analyzed using various statistical techniques.

- **Non-Negative Matrix Factorization (NMF)**: NMF is a powerful technique used to uncover latent topics within documents or text data. It works by breaking down a large matrix of data into smaller, more manageable matrices that can be analyzed for patterns and relationships. By identifying which words or phrases frequently co-occur across different documents, NMF can help us to understand the underlying themes or topics that are present within a large corpus of text data. This information can be incredibly valuable for a wide range of applications, from improving search algorithms to developing more effective marketing strategies.

Following the application of three different techniques, namely TFIDF, DTM, and NMF to our dataset, we were able to obtain a comprehensive data frame(*figure-4*) that showcases five distinct topics that were discovered from the data. We were able to successfully identify all five of these topics and gain a deeper understanding of the underlying patterns



	Word 1	Word 2	Word 3	Word 4	Word 5	Word 6	Word 7	Word 8	Word 9	Word 10	Word 11
Topic 1	account	bank	check	money	fund	chase	deposit	branch	day	number	transaction
Topic 2	credit	card	report	inquiry	chase	account	score	company	information	debt	limit
Topic 3	payment	balance	month	fee	statement	time	auto	date	pay	credit	chase
Topic 4	charge	card	chase	dispute	transaction	fee	merchant	fraud	claim	purchase	service
Topic 5	loan	mortgage	home	modification	chase	property	letter	rate	document	time	bank

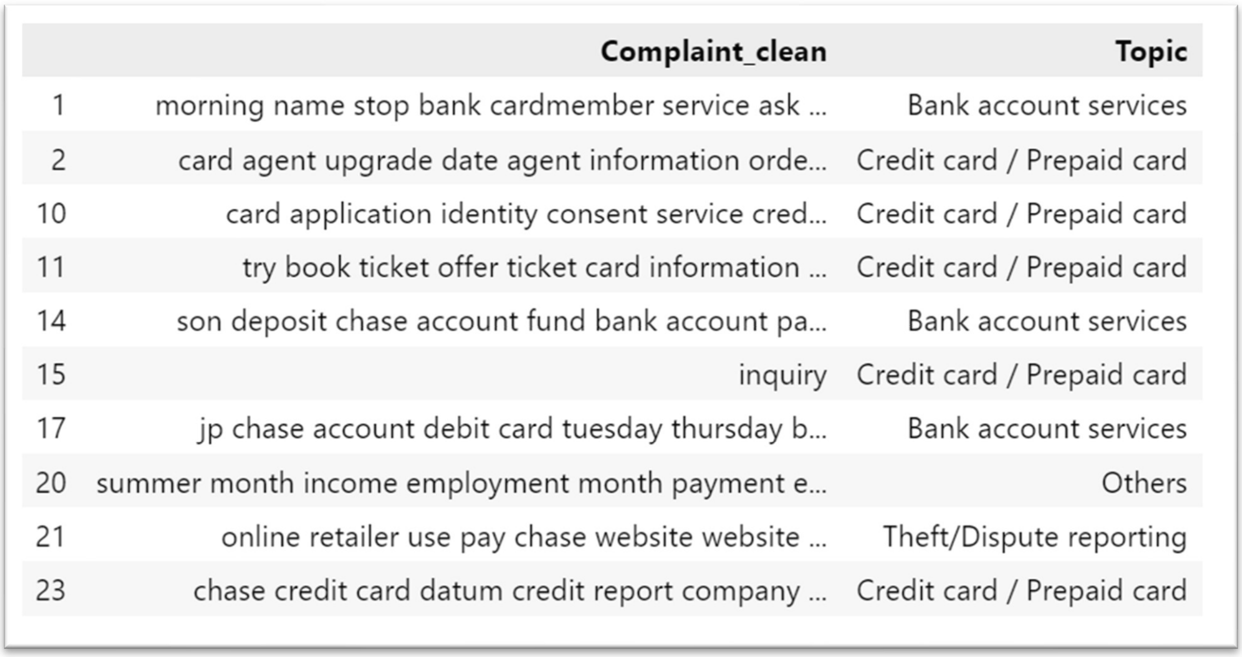
Observation Looking at the topics above, for each topic, we can give a label based on their products/services:

- Topic 1 = Bank account services
- Topic 2 = Credit card / Prepaid card
- Topic 3 = Others
- Topic 4 = Theft/Dispute reporting
- Topic 5 = Mortgages/loans

Figure 3

and trends present within our dataset. Ultimately leading to valuable insights that can be used to inform our decision-making processes moving forward.

After performing topic modeling on our customer complaint dataset, we were able to identify various topics that emerged from the data. To facilitate further analysis and modeling, we decided to create a new variable in our dataset called 'Topic', which will serve as our target variable. This variable captures the essence of the complaint and represents the main issue that the customer was facing. By creating this new variable, we aim to improve the accuracy and effectiveness of our models in predicting customer complaints and finding solutions to address their concerns. The data set includes a variable called "Topic", which is represented in *figure-5.*



	Complaint_clean	Topic
1	morning name stop bank cardmember service ask ...	Bank account services
2	card agent upgrade date agent information orde...	Credit card / Prepaid card
10	card application identity consent service cred...	Credit card / Prepaid card
11	try book ticket offer ticket card information ...	Credit card / Prepaid card
14	son deposit chase account fund bank account pa...	Bank account services
15	inquiry	Credit card / Prepaid card
17	jp chase account debit card tuesday thursday b...	Bank account services
20	summer month income employment month payment e...	Others
21	online retailer use pay chase website website ...	Theft/Dispute reporting
23	chase credit card datum credit report company ...	Credit card / Prepaid card

Figure 4

Modeling:-

- **CountVectorizer**:

  -In the process of developing a machine learning model, we first created a training dataset that consisted of two variables – ‘complaint_what_happened ' and 'Topic'. Once the dataset was ready, we applied a technique called CountVectorizer to it. CountVectorizer is a great tool provided by the sci-kit-learn library in Python it converts a collection of text documents to a matrix of token counts, which can then be used to train a machine-learning model. This process helps in identifying the frequency of each word in the dataset and creating a numerical representation of the text data. By applying CountVectorizer to our dataset, we were able to prepare it for further analysis and machine-learning model development.

- **TfidfTransformer:-**

  The concept of term frequency-inverse document frequency (tf-idf) is widely used in natural language processing(NLP) and information retrieval. The aim of this technique is to reduce the weight of frequently occurring words or tokens in a given corpus, which may not be as informative as the words that appear rarely in the text. By scaling down the impact of such common words, tf-idf helps to extract the most relevant features or keywords from the text, which can be used for various applications such as text classification, clustering, and summarization. The tf-idf score for a term is calculated by multiplying its frequency in a document by the inverse frequency of the term in the corpus. This way, the words that occur more frequently in a document but less frequently in the corpus will get a higher weight in the tf-idf calculation, indicating their importance in that particular document.

- **Training and Testing**:- "To ensure accurate results, we have taken the necessary step of dividing our dataset into two parts - 75% for training and 25% for testing. This will allow us to train our model effectively on the majority of the dataset while also ensuring that we have a separate set of data to test the efficacy of our model once it is trained."

<u>Models we used for modeling:</u>

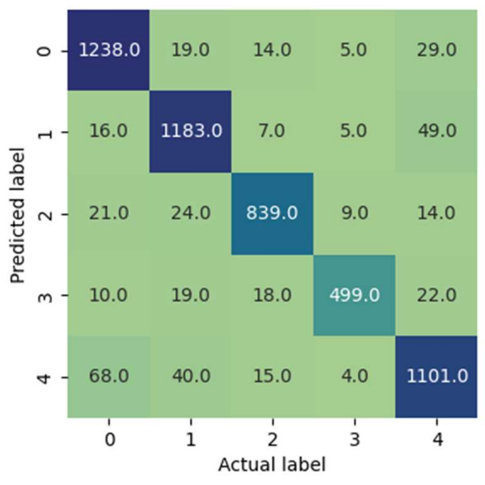
Model 1 - Logistic Regression

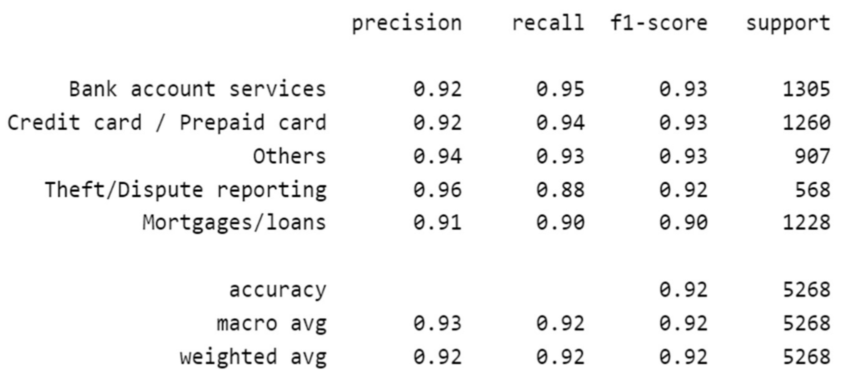
Model 2 - Decision Tree Classifier

Model 3 - Random Forest Classifier

Logistic Regression:-

After applying logistic regression on both the training and testing data, we obtained an accuracy score of 91.91%. However, after tuning the hyperparameters, we were able to improve the score to 92.25%. Additionally, we achieved good precision scores, recall, and f1-scores for all of the topics. The results can be seen in *figure-6.*



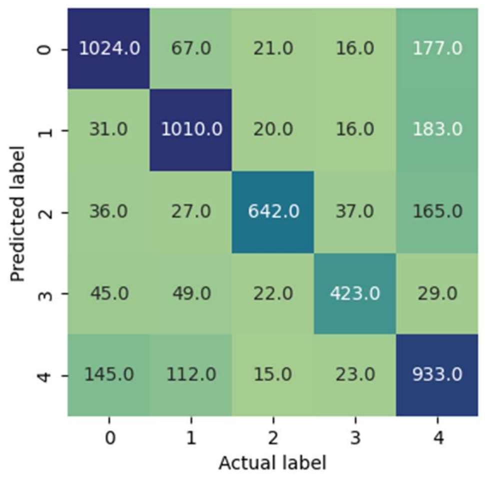


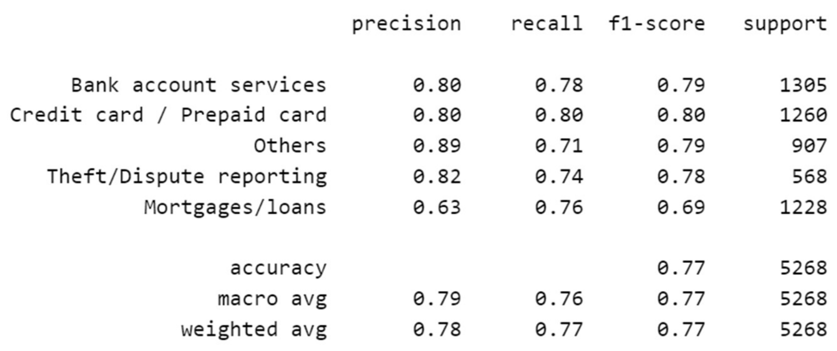
```
                           precision    recall  f1-score   support

     Bank account services       0.92      0.95      0.93      1305
Credit card / Prepaid card       0.92      0.94      0.93      1260
                    Others       0.94      0.93      0.93       907
  Theft/Dispute reporting        0.96      0.88      0.92       568
           Mortgages/loans       0.91      0.90      0.90      1228

                  accuracy                           0.92      5268
                 macro avg       0.93      0.92      0.92      5268
              weighted avg       0.92      0.92      0.92      5268
```

Figure 5

Decision Tree Classifier:-

We applied Decision Tree Classifier to our training and testing data and achieved an accuracy score of 77%. Despite tuning the hyperparameters, we were unable to improve the score. Unfortunately, we did not achieve satisfactory precision scores, recall, and f1-scores for all topics. The result is displayed in *figure-7*.



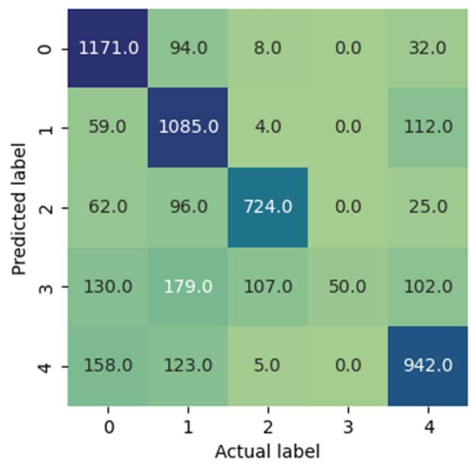


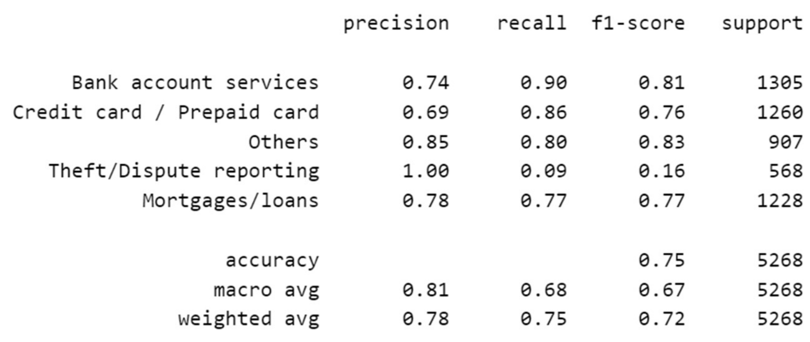
```
                         precision    recall  f1-score   support

    Bank account services     0.80      0.78      0.79      1305
Credit card / Prepaid card     0.80      0.80      0.80      1260
                   Others     0.89      0.71      0.79       907
  Theft/Dispute reporting     0.82      0.74      0.78       568
          Mortgages/loans     0.63      0.76      0.69      1228

                 accuracy                         0.77      5268
                macro avg     0.79      0.76      0.77      5268
             weighted avg     0.78      0.77      0.77      5268
```

Figure 6

Random Forest Classifier:-

We achieved an 77.02% accuracy score using logistic regression on our training and testing data. Despite our best efforts in tuning the hyperparameters, we were unable to improve the score. Unfortunately, we encountered challenges in achieving satisfactory precision scores, recall, and f1-scores for all topics. A visual representation of our results can be found in *figure-8*.





```
                         precision    recall  f1-score   support

    Bank account services     0.74      0.90      0.81      1305
Credit card / Prepaid card     0.69      0.86      0.76      1260
                   Others     0.85      0.80      0.83       907
  Theft/Dispute reporting     1.00      0.09      0.16       568
          Mortgages/loans     0.78      0.77      0.77      1228

                 accuracy                         0.75      5268
                macro avg     0.81      0.68      0.67      5268
             weighted avg     0.78      0.75      0.72      5268
```

Figure 7

Concluding the modeling process:

After a thorough modeling process, we have tested the effectiveness of three different models, namely 'Logistic Regression', 'Decision Tree Classifier' and 'Random Forest Classifier'. We evaluated the performance of these models and found that 'DecisionTreeClassifier' produced an accuracy score of 77% and 'RandomForestClassifier' produced an accuracy score of 75%. However, the 'LogisticRegression' model outperformed the other two models by producing a much higher accuracy score of 92%.

The 'LogisticRegression' model, which demonstrated the highest precision, recall, and f1-score for all topics, has proved to be the most effective algorithm for the given dataset. This model has shown superior performance in terms of accuracy and precision compared to the other models tested.

Overall, our findings suggest that the 'LogisticRegression' model is the most effective algorithm for the given dataset. This model's high accuracy score and superior performance in precision, recall, and f1-score for all topics highlight its suitability for the given dataset. Therefore, we can conclude that the 'LogisticRegression' model is the best option for modeling the given dataset.

Reference :

[1] Customer Complaint Classification Data: Kaggel