Consumer Complaints Classification

Anshul Shandilya
anshul.shandilya@sjsu.edu
San Jose State University,
1 Washington Sq, San Jose, CA
95192

Ganesh Nehru
ganesh.nehru@sjsu.edu
San Jose State University,
1 Washington Sq, San Jose, CA
95192

Ramkumar Sivakumar
ramkumar.sivakumar@sjsu.edu
San Jose State University,
1 Washington Sq, San Jose, CA
95192

Mohammed Tuba Ahmed
mohammedtuba.ahmed@sjsu.edu
San Jose State University,
1 Washington Sq, San Jose, CA
95192

Abstract - **Customer satisfaction plays a crucial role in any business. In fact, it is shown that a company that has a high level of customer satisfaction yields a higher retention rate of customers, as well as having a positive brand reputation, and overall greater lifetime value. Customer satisfaction also plays an important role in driving customer loyalty and boosting acquisition, as well as reflecting on the business team's performance. With the digitization of most businesses, managing customer satisfaction in the digital space has become exceedingly important. While some industries have a rigorous focus on customer outcomes through government regularization, other businesses will benefit from introducing this aspect of process improvement into their workflow..**

1. INTRODUCTION

In the corporate sector, customer complaints are a constant occurrence, with numerous departments and individuals getting complaints often. It's crucial to correctly classify these complaints into the appropriate categories in order to handle and address them in an efficient manner. This makes it possible to take the proper action and to forward concerns to the right person or department for resolution.

Manual labor is used in traditional categorization techniques, which can be time-consuming and error-prone. Additionally, given the huge amount of complaints that many firms get every day, manual categorization might not be able to keep up. As a result, a more effective and precise approach to complaint categorization is required.

Text categorization problems, among others, have been successfully completed using machine learning approaches. In this article, we suggest using machine learning techniques to categorize client complaints, offering a faster and more precise way of complaint classification.

In this project, we will be constructing classification models of consumer complaints based on text mining and natural language processing techniques to produce insights into consumer behavior, which may benefit businesses in taking actionable steps to improve customer satisfaction and retention. Our approach to this solution is two-fold. First, we aim to use standard sentiment analysis techniques to classify the complaints into standard categories. Second, we plan to apply topic modeling using LDA (Latent Dirichlet Allocation) to classify the given complaints and identify potential new categories**.**

2. DATA CLEANING AND PREPROCESSING

The consumer complaints dataset we obtained had a lot of redundant features that were not relevant to our problem. We performed feature extraction where only the 'Product' and 'Consumer complaints narrative' features were extracted to create a data frame. Upon inspecting the data frame, it was discovered that there were a total of eighteen products–most of which were of similar categories. This caused the dataset to be imbalanced as most of the complaints belonged to the 'Credit Reporting' category, while other data points were divided into various other categories, some of which were similar, as shown in Figure 2.1. To clean up the data, we grouped and relabeled the product titles that were similar. In doing so, this resulted in the eighteen products mentioned previously, being condensed down to a total of seven products, as depicted in Figure 2.2.

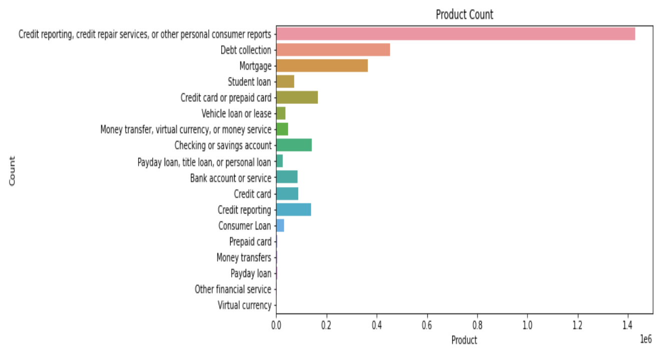


FIGURE 2.1
INITIAL DATASET SHOWN TO BE IMBALANCED, WITH REDUNDANT CLASSES/CATEGORIES

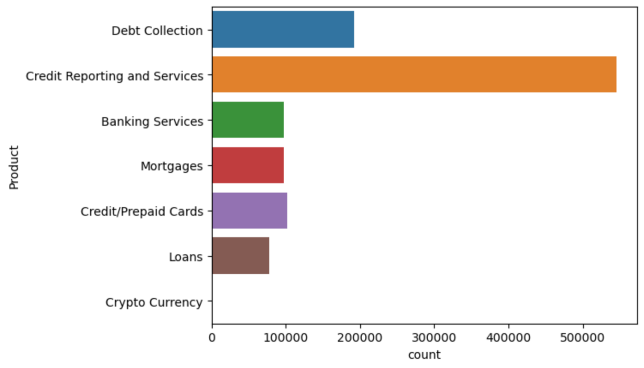


FIGURE 2.2
PREPROCESSED DATASET AFTER COMBINING PRODUCTS WITH SIMILAR TITLES

Upon further analysis, we noticed that there were a total of 3,096,756 data points; of which, 1,984,336 points had null values for their respective ‘narrative’ column. These data points were dropped, which resulted in a condensed data frame consisting of 1,112,420 data points.

Finally, to complete the preprocessing step, we removed stop words, as well as tokenized the text, and performed lemmatization on the complaint narratives. This process took nearly five to six hours to complete. Upon completion, the preprocessed complaints dataset was exported, which will be used for constructing our models.

2.1 Do we have sufficient data?

There can be several approaches to determine if the dataset obtained is sufficient for our learning problem.In our case, two approaches were taken. One of them is to simply create train-test sets from the dataset with 80-20 splits and see how well the model performed on the test set. If the model performed well, then we can safely assume that we had enough data.

This process of creating test sets and evaluating model performance was done for each method that was applied to the dataset and using F1 scores as an evaluation metric, the scores on the test set were recorded, as done in section 4 of this paper. We discovered that we do not have enough data to learn and evaluate for the category “Virtual Currency.”

Later, we analyzed the consumer complaint narratives to check if the data is learnable. On the preprocessed data, we extracted the top 20 most frequent words for each category to generate a word cloud. We observed that the frequent words for each category differed. Hence, comparing the word clouds visually, we inferred that the data is learnable.

3. METHODS

We obtained the pre-processed dataset of customer complaints, Which was obtained using the data cleaning and pre-processing techniques mentioned in the section above. The dataset included text complaints and the categorization labels that went with each complaint. This processed dataset had 1,111,240 data points. This presented us with a computing resource problem, were performing various techniques like Tf-iDF and also training some models would take up more resources like memory, than what was available, and thus fail to execute. So instead of using the full 1,111,240 data points for our problem, it was decided that we would randomly select 100,000 samples from the dataset to apply our methods on the models that took a longer time to run. Each method was performed on individually selected 100,000 random samples except for CuDNNLSTM and Naive Bayes due to its efficiency. This would ensure that all methods would successfully train, without facing any resource deficiency. Using these data points, the following classification models were constructed:

- LSTM Model
- SVM Model
- Multinomial Naive Bayes Model
- KNN Model

We also experimented with LDA - Latent Dirichlett Allocation for topic modeling. This was done to potentially identify new product classes from the complaint narratives.

3.1 - LSTM Model

Long Short Term Memory (LSTM) is a type of Recurrent Neural Network with feedback connections that has been widely used in the past for sequential data. The LSTM is trained by passing words as input sequentially until there are no more words. This input is mapped to a target variable that the model predicts.

The LSTM network was built using the NLTK and Keras library. The complaint narratives were tokenized and padded to a maximum sequence length of 50. The dataset was split into train and test sets with an 80-20 split. During training, 10% of the test set was used for validation. The estimated time for the LSTM model to finish one epoch was over an hour. Therefore CuDNNLSTM was used to speed up the training. The model was trained for 10 epochs for 10 minutes with the default settings of Adam optimizer. Figure 3.1.1 compares E_{in} to E_{out}. The validation loss did not improve as the model started to overfit after 2 epochs.

The performance of the trained model was evaluated on the test dataset. An F1 score of 0.86 with a classification accuracy of 86% was obtained on the test dataset (Figure 3.1.2). The evaluation metrics were poor for category 6 due to insufficient training samples. Hence, learning is not feasible for the category "Crypto Currency."

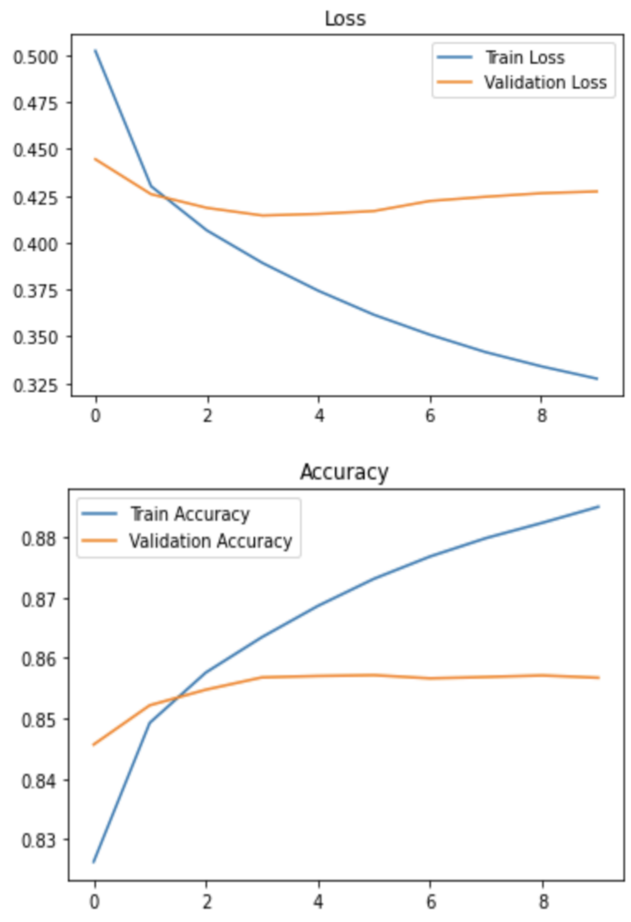
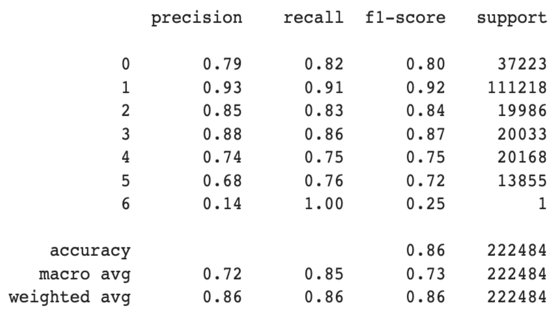


FIGURE 3.1.1
(LSTM TRAINING)



```
              precision    recall  f1-score   support

           0       0.79      0.82      0.80     37223
           1       0.93      0.91      0.92    111218
           2       0.85      0.83      0.84     19986
           3       0.88      0.86      0.87     20033
           4       0.74      0.75      0.75     20168
           5       0.68      0.76      0.72     13855
           6       0.14      1.00      0.25         1

    accuracy                           0.86    222484
   macro avg       0.72      0.85      0.73    222484
weighted avg       0.86      0.86      0.86    222484
```

FIGURE 3.1.2
(LSTM EVALUATION ON THE TEST DATA)

3.2 - SVM Model (linear)

One of the models that were selected for approaching our complaints classification problem was the Support Vector Machines (SVM) model. This was selected as an approach because they are effective for text classification as they can handle nonlinear complex data like texts quite well.

The processed dataset had a 'complaints' or 'narrative' column which contained the processed text for each complaint, and the 'product' column contained the labels for which category/classes the complaints belonged to. So, the first step taken was to extract the 'product' column and encode them as labels using sklearn's LabelEncoder. The next step was to create a validation set from the 100,000 sample data points extracted from the main processed dataset. This was done by splitting the data into 80% training data and 20% test data along with their labels. This was done so that the model could train on the 80% training set, and then it would be validated using the 20% test set.

Before training the model, **TF-iDF** (Term frequency-inverse document frequency) was performed on both the training and test sets to transform the strings to numerical vectors for training. The following parameters were used for **TF-iDF**:

- min_df = 3
- strip_accents = 'unicode'
- analyzer = 'word'
- token_pattern = r'w{1,}'
- ngram_range = (1,3)
- use_idf = 1
- smooth_idf = 1
- sublinear_tf = 1
- stop_words = 'english'

Applying TF-iDF increased the dimension of the dataset, so the transformed dataset was reduced to 250 components using **TruncatedSVD**. After reducing the components, to ensure maximum performance from SVM, we normalized the obtained data using **StandardScaler**.

The SVM model was trained using the above dataset with probability = True and all other parameters left as default, and the model was tested using the validation test set created above. As an evaluation metric, all methods used F1 scoring metric. For the training set, the F1 score was 0.8858307 and on the test set, the score was 0.8568457. Figure 3.2.1 shows the classification report for the SVM model and Figure 3.2.2 shows the confusion matrix for the test set.

```
              precision    recall  f1-score   support

           0       0.84      0.87      0.86      1760
           1       0.89      0.92      0.90      9785
           2       0.80      0.76      0.78      1867
           3       0.00      0.00      0.00         1
           4       0.80      0.77      0.79      3458
           5       0.79      0.70      0.74      1368
           6       0.92      0.90      0.91      1760

    accuracy                           0.86     19999
   macro avg       0.72      0.70      0.71     19999
weighted avg       0.86      0.86      0.86     19999
```

Figure 3.2.1

Classification Report on test set (SVM)

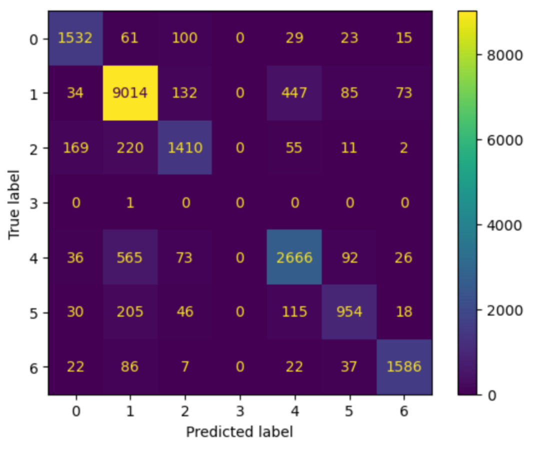


Figure 3.2.2

Confusion matrix for the train set (SVM)

We can see from the confusion matrix that for category 1, which is ‘credit_reporting’, the number of true predictions were the highest while there were none for 3, which is ‘mortgage’. This is for the reason that sampling of 100,000 elements was not able to select samples from that particular class. Since there was no data collected, we can safely assume that learning is not feasible for ‘mortgage’ in this particular case.

One limitation of this model and the scaling done for this method is that on the full dataset with more than 1 million data points, the method requires a lot of memory to allocate the arrays created by it, and also the time required to train the SVM model for the full dataset was more than approx. 20 hours. So for this, we decided to take a sample of 100,000 datasets as mentioned above for the classification problem.

To determine if we used enough data for SVM and that the model was able to learn, we can look at the accuracy metric obtained on the test set. The model was trained on the training set and then tested on data points that the model had not seen before. The accuracy on the test set tells us that for unseen samples, the model was able to accurately predict 86% of all classes correctly. If we say that we have an error margin of 15%, then we can safely assume that the SVM model both had enough data, and that the model was able to learn from it.

3.3 - Multinomial Naive Bayes Model

Although there exists a family of Naive Bayes statistical algorithms, we decided to use the Multinomial Naive Bayes classifier since it is very useful in applications specific to text classification. Using the Multinomial Naive Bayes classifier, we are trying to calculate the probability of each tag for a given narrative text in a consumer complaint.

For this model, we split the data into the training and test set, where the training set consisted of 80% of the data and the test set contained 20% of the data. The resulting sizes of the training and test set were 889,871 and 222,468, respectively. The model produced the following metrics, classification report, and confusion matrix:

```
Metrics:

   Precision:  0.8494
   Recall:  0.8514
   F1-Score:  0.8486
   Accuracy:  0.8514


Classification Report:

              precision    recall  f1-score   support

           0       0.80      0.75      0.78     38161
           1       0.88      0.93      0.90    108891
           2       0.85      0.87      0.86     19693
           3       0.87      0.93      0.90     19533
           4       0.78      0.74      0.76     20568
           5       0.84      0.61      0.71     15619
           6       0.00      0.00      0.00         3

    accuracy                           0.85    222468
   macro avg       0.72      0.69      0.70    222468
weighted avg       0.85      0.85      0.85    222468
```

Figure 3.3.1

Metrics and Classification Report for Naive Bayes

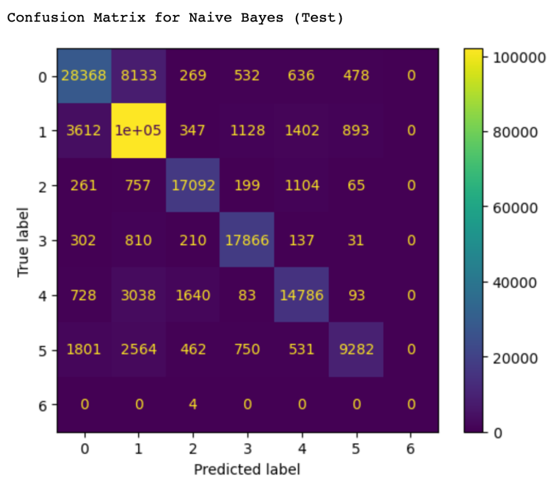


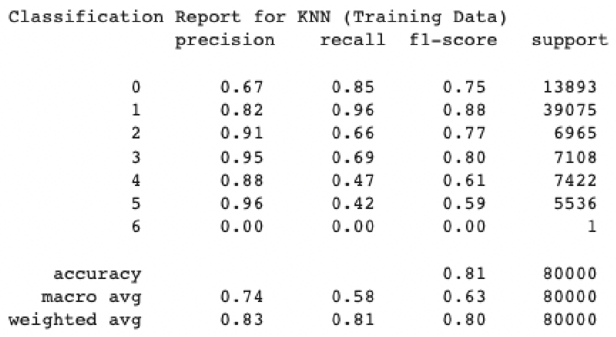
Figure 3.3.2
Confusion Matrix for Naive Bayes

From Figure 3.3.1, we can see that our model produces an accuracy of 85%. Furthermore, we can see from the confusion matrix, the number of products that have been correctly predicted, as well as falsely predicted. Product 1, which is the “Credit Reporting & Services” seems to have the highest number of true predictions. From the classification report, we can also see that Product 1 had the highest F1-score of 0.90, which signifies that the model was very accurate in classifying this product. On the other hand, Product 6, which is the “Crypto Currency” had the lowest number of true predictions of 0, in the confusion matrix. This is due to the lack of training samples of this product. From the classification report, we can also observe that the F1-score of Product 6 is 0.0. Hence, this indicates that learning is not feasible for the product, “Crypto Currency.”

3.4 - KNN Model

KNN - K-Nearest Neighbor is a popular multi-class supervised machine learning classification algorithm. For this project, we used KNN to classify the complaints to its respective product class.

Once the pre-processing was done and the TF-IDF transformation was complete, we proceeded to encode the product categories into numerical values and build a KNNClassifier model on top of this. We used sklearn library’s implementation for the same. As a group, we decided to limit the sample size to 100,000. To enhance performance, We tried implementing the model using the RAPIDS accelerator. While this showed considerable improvement in the processing time, we decided to limit the sample size to standardize across implementations and minimize the performance factor and varying machine resources.

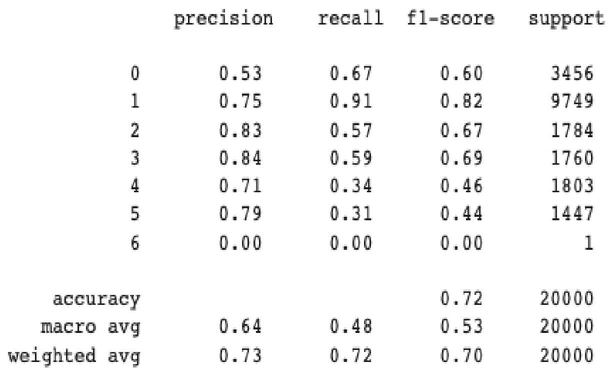


```
Classification Report for KNN (Training Data)
              precision    recall  f1-score   support

           0       0.67      0.85      0.75     13893
           1       0.82      0.96      0.88     39075
           2       0.91      0.66      0.77      6965
           3       0.95      0.69      0.80      7108
           4       0.88      0.47      0.61      7422
           5       0.96      0.42      0.59      5536
           6       0.00      0.00      0.00         1

    accuracy                           0.81     80000
   macro avg       0.74      0.58      0.63     80000
weighted avg       0.83      0.81      0.80     80000
```

Figure 3.4.1
Classification report for KNN - Training Data



```
              precision    recall  f1-score   support

           0       0.53      0.67      0.60      3456
           1       0.75      0.91      0.82      9749
           2       0.83      0.57      0.67      1784
           3       0.84      0.59      0.69      1760
           4       0.71      0.34      0.46      1803
           5       0.79      0.31      0.44      1447
           6       0.00      0.00      0.00         1

    accuracy                           0.72     20000
   macro avg       0.64      0.48      0.53     20000
weighted avg       0.73      0.72      0.70     20000
```

Figure 3.4.2
Classification report for KNN - Test Data

From the classification reports, we can note that KNN performs worst for this particular problem. It doesn’t scale well with the size of the dataset and the number of dimensions. Since it's a distance-based algorithm, the cost of calculating distance for each point is too performance intensive.

3.5 - LDA Topic Modeling

Latent Dirichlet Allocation (LDA) is a statistical topic modeling technique that helps in discovering abstract topics in a collection of text documents. For this project, we decided to experiment with this technique to potentially identify new product classes and subclasses.

While the initial pre-processing steps will remain the same, there were some steps taken exclusively for LDA. The lemmatization was performed using the spacy package ‘en_core_web_md’. Nouns and adjectives were preferred since they usually capture the essence of a sentence. We then created a dictionary based on these tokenized complaints.

We made use of the 'gensim' library to build and train the LDA model.

The output obtained was in terms of 10 abstract topics. For example, one of the topics had keywords like 'account', 'bank', 'money', 'deposit', 'fraud', and 'claim'. The documents categorized under this topic were indeed composed of these keywords. The benefit here is that such keywords can give us insight into new product classes. For instance, consider that 'Bank Fraud' could be one such product class to which the complaint might be redirected.

4. COMPARISONS

The precision, recall, and F1 scores in table 4.1 are macro averages. The LSTM model had a better Recall, F1, and Accuracy score over the other models. KNN had a better precision score. Moreover, the time to train the LSTM model

Method	Precision	Recall	F1 Score	Accuracy
LSTM	0.72	0.85	0.73	0.86
SVM	0.72	0.70	0.71	0.86
Naive-Bayes	0.72	0.69	0.70	0.85
KNN	0.73	0.72	0.70	0.72

TABLE 4.1

over ten epochs was around 10 minutes. Therefore, though SVM had the same accuracy as LSTM, the latter performed better with respect to Recall, F1, Accuracy, and Efficiency.

But considering all the metrics, if we look at all of the models LSTM, apart from Precision, all the other metrics are relatively higher than the other models. So, with this, we selected the LSTM as the best-performing model for our classification problem.

5. CONCLUSION

In the end, we were able to build models that classify the complaints with an accuracy of 86% and help streamline the consumer complaint filing process. We overcame the target variable ambiguity by renaming certain categories from the information we inferred in the data exploration steps. However, we did not have enough data to correctly classify complaint narratives of type "Crypt/Virtual Currency."

Our future work would be focused on improving classification accuracy using novel methods and additional features. During data exploration, we identified that the feature "Submitted-via", the mode of submission of complaints by a consumer, had some influence on the target variable. Hence, the accuracy can be improved by incorporating more such features.

REFERENCE

[1] Consumer Finance Gov - Dataset https://www.consumerfinance.gov/data-research/consumer-complaints/
[2] https://www.nltk.org/
[3] LSTM - CuDNN https://www.tensorflow.org/api_docs/python/tf/compat/v1/keras/layers/CuDNNLSTM