CLASSIFICATION OF COMPLAINT NARRATIVES INTO ISSUE CATEGORIES USING LOCALITY-SENSITIVE HASHING, NAÏVE BAYES AND ANN

Aditya Joseph James, Yash Pratap Solanky

ABSTRACT

Classification of text has been the focus of many real-world problems. This has gained recent traction with the advent of fake news. Another application of text classification is for increasing service effectiveness. In this project we look to classify complaint narratives into the issue categories to streamline the complaint resolution process.

1. INTRODUCTION

Text classification has been the object of interest for a very long time. One of the earliest applications of this was to tag the parts of speech for text. This scope has increased further to include other use cases such as spam detection, sentiment analysis, news categorization etc [1]. Classification of text data has recently become even more important with the emergence of fake news online. Google estimated that in 2020 alone their systems found 40 billion spam pages every day [2]. This volume is clearly massive and with each day this only keeps multiplying exponentially.

Clearly, the classification of textual data is very important, and many researchers have tried to come up with innovative solutions to tackle this. Some of the most used techniques are Naïve Bayes, Hidden Markov models, SVM [3]. We have tried to use some of these algorithms to come up to solve another important but highly relevant problem that plagues the service industry.

Almost all of us have used a service in which we have found some fault or issue. The first step that we take to try and resolve the problem will be to drop a service request to the company from which we have bought the service. We then wait anxiously for a call back from the company so that the issue can be resolved. From a company standpoint the main challenge is to ensure that the complaints are funneled through proper channels and assigned to the right team so that corrective measures might be taken to resolve the problem. If the company delays taking action on a complaint raised by a complaint it can face a lot of negative publicity. A study by finances online indicates that 91% of the customers leave without warning as a reaction to bad customer service. The study also indicated that 40% would also recommend their friends not to support the company [4]. Both these insights clearly indicate why the turnaround time is very important from the company standpoint. Our study aims to carry out classification analysis on complaints received by the Consumer Financial Protection Bureau, a sample of which has been shared on Kaggle [5]. We have tried to carry out the classification of issues based on the complaint narrative so that the complaint can be funneled to the right team for complaint resolution. We have tried using Naïve Bayes, LSH and neural networks for our classification problem. It is important to note that in our dataset a specific issue could be linked to multiple products which means that the narratives could be very different for the same issue type. For example, the issue type of "Incorrect information on your report" could be tagged to a mortgage or a student loan or a credit card or a savings product. Each narrative might hence be different as the product is different.

2. METHODS

2.1. Locality-Sensitive Hashing (LSH)

[6] "Locality-sensitive hashing (LSH) is an algorithmic technique that hashes similar input items into the same "buckets" with high probability. (The number of buckets is much smaller than the universe of possible input items.) Since similar items end up in the same buckets, this technique can be used for data clustering and nearest neighbor search. It differs from conventional hashing techniques in that hash collisions are maximized, not minimized. Alternatively, the technique can be seen as a way to reduce the dimensionality of high-dimensional data; high-dimensional input items can be reduced to low-dimensional versions while preserving relative distances between items.

Hashing-based approximate nearest neighbor search algorithms generally use one of two main categories of hashing methods: either data-independent methods, such as locality-sensitive hashing (LSH); or data-dependent methods, such as locality-preserving hashing (LPH)".

One of the initial formulations of this was by Andrei Z Broder [7] who found that he could use resemblance which is a mathematical notion to find similar documents. The method looks to find documents which have the same information but different grammar and other formatting. The provides a value between 0 and 1 with 0 meaning that the documents are dissimilar and 1 implying that they are very similar.

[7] “To compute the resemblance of two documents it suffices to keep for each document a “sketch” of a few (three to eight) hundred bytes consisting of a collection of fingerprints of “shingles” (contiguous subsequences of words, sometimes called “q-grams”). The sketches can be computed fairly fast (linear in the size of the documents) and given two sketches the resemblance of the corresponding documents can be computed in linear time in the size of the sketches. Furthermore, clustering a collection of m documents into sets of closely resembling documents can be done in time proportional to m log m rather than m^2.”

We wanted to try and implement the minhash LSH algorithm as it was an interesting algorithm that we were taught in class. Our idea was that when a customer raises a complaint which falls in a particular category then they should highlight the same key points in the narrative. This might be true for some cases but the accuracy would largely depend on how structured the company or organization is in tagging the complains to a category and how similar the customers are in terms of how they raise the complaint as every individual might raise the same complaint differently.

2.2. Naive Bayes

In statistics, naive Bayes classifiers are a family of simple "probabilistic classifiers" based on applying Bayes' theorem with strong (naïve) independence assumptions between the features (see Bayes classifier). They are among the simplest Bayesian network models, but coupled with kernel density estimation, they can achieve higher accuracy levels.[8]

Text classifiers often don’t use any kind of deep representation about language: often a document is represented as a bag of words. (A bag is like a set that allows repeating elements.) This is an extremely simple representation: it only knows which words are included in the document (and how many times each word occurs), and throws away the word order! [16]

We implemented Naïve Bayes in order to predict the Issue from customer complaints using MultinomialNB and ComplementNB functions from sklearn library in python. We used the default model as that gave us the highest accuracy

MultinomialNB implements the naive Bayes algorithm for multinomial distributed data, and is one of the two classic naive Bayes variants used in text classification. The distribution is parametrized by vectors for each class where the number of features (in text classification, the size of the vocabulary) and is the probability of features appearing in a sample belonging to class.

ComplementNB implements the complement naive Bayes (CNB) algorithm. CNB is an adaptation of the standard multinomial naive Bayes (MNB) algorithm that is particularly suited for imbalanced data sets. Specifically, CNB uses statistics from the complement of each class to compute the model’s weights. The inventors of CNB show empirically that the parameter estimates for CNB are more stable than those for MNB. Further, CNB regularly outperforms MNB (often by a considerable margin) on text classification tasks. [9]

2.3. Support Vector Machine (SVM)

In machine learning, support-vector machines (SVMs, also support-vector networks) are supervised learning models with associated learning algorithms that analyze data for classification and regression analysis. Developed at AT&T Bell Laboratories SVMs are one of the most robust prediction methods, being based on statistical learning frameworks Given a set of training examples, each marked as belonging to one of two categories, an SVM training algorithm builds a model that assigns new examples to one category or the other, making it a non-probabilistic binary linear classifier [10]

Some algorithms like the SVM are by default binary classifiers. Therefore, if we have a problem with more than two classes, we need to construct as many classifiers as there are classes [one versus all strategy). However, it is not fair to compare a single multi-class naive Bayes (or kNN) classifier to n SVM classifiers (for n classes). [17]

We used the LinearSVC function of the sklearn library to implement SVM. We again, used the default model as that gave us the highest accuracy

2.4. Artificial Neural Network (ANN)

An ANN is based on a collection of connected units or nodes called artificial neurons, which loosely model the neurons in a biological brain. Each connection, like the synapses in a biological brain, can transmit a signal to other neurons. An artificial neuron receives a signal then processes it and can signal neurons connected to it. The "signal" at a connection is a real number, and the output of each neuron is computed by some non-linear function of the sum of its inputs. The connections are called edges. Neurons and edges typically have a weight that adjusts as learning proceeds. The weight increases or decreases the strength of the signal at a connection. Neurons may have a threshold such that a signal is sent only if the aggregate signal crosses that threshold. Typically, neurons are aggregated into layers. Different layers may perform different transformations on their inputs. Signals travel from the first layer (the input layer), to the last layer (the output layer), possibly after traversing the layers multiple times.[11]

We implemented an artificial neural network for our problem using Keras library. We tried a lot of different models and

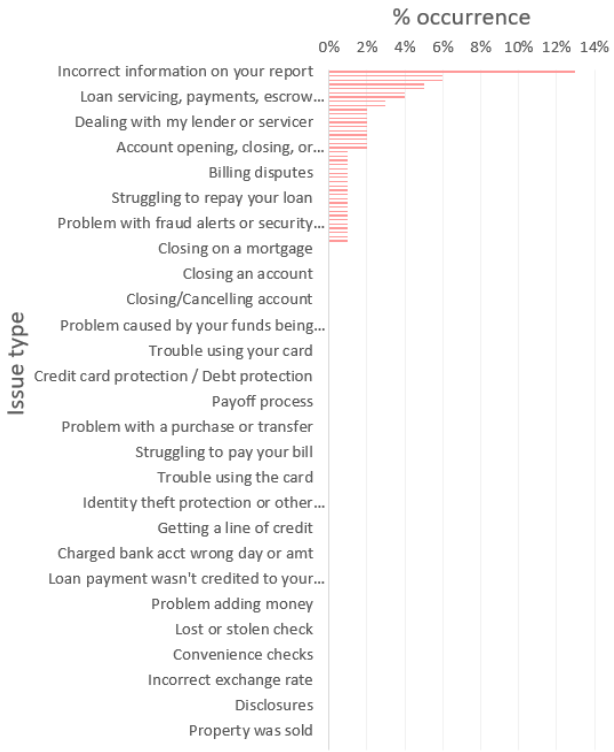
hyperparameter tuning but we ended up using a model with the following layers:

- 512 Node Dense input layer with Relu activation
- Dropout layer with 0.5 parameter
- 256 Node Dense hidden layer with Relu activation
- Dropout layer with 0.5 parameter
- 9 output Dense layer with sigmoid activation

We trained our model using the binary cross entropy loss. This is because the labels are not disjoint. For a given abstract, we may have multiple categories. So, we will divide the prediction task into a series of multiple binary classification problems. This is also why we kept the activation function of the classification layer in our model to sigmoid. [12]

3. DATA EXPLORATION

The data that we chose to carry out our study was from Kaggle, but it was originally taken from the consumer financial protection bureau [13]. The data had over 1.28 million complaints which were categorized across 161 different issue types. The below chart shows the distribution of the various categories.



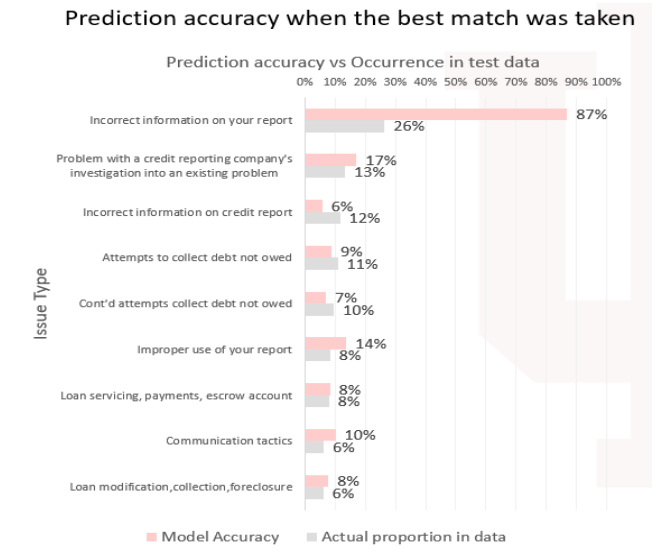
As observed from the above chart there is a long tail where some categories have very low counts of complaint narratives. For this reason, we chose to filter the records to have only those records where the number of occurrences in the issue category was at least 10,000. This resulted in us narrowing the number of issue types from 161 to 9 primary issue types. One of the potential problems that we foresaw was that some of the issue types seemed similar and hence there was a high chance that there might be similarity in the narratives. The graph on the adjacent side would illustrate the issue that we have raised here.

Given the above illustration it was clear that we might have a certain number of issues being tagged incorrectly due to similarity of the complaints between different issue types. We still wanted to try and implement LSH as it was a new and very different technique as compared to the others that are regularly used.

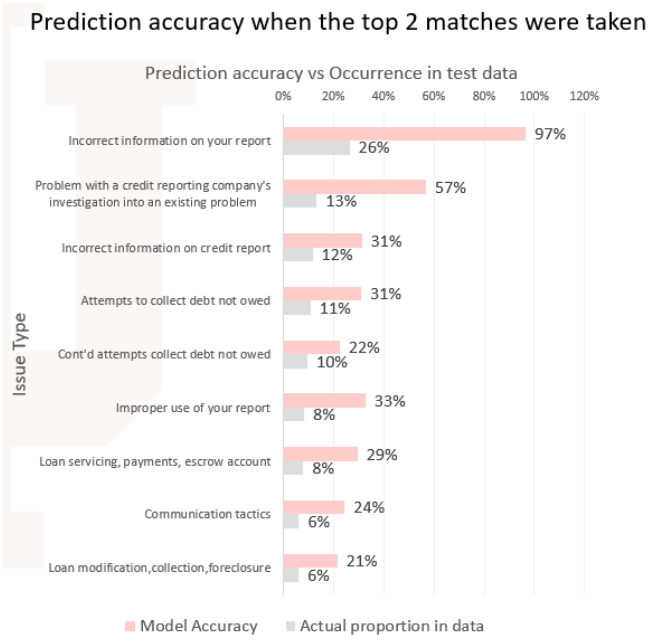
4. RESULTS

4.1. Locality-Sensitive Hashing (LSH)

We implemented the LSH as given by the MinhashLSHforest method in the datasketch package [14]. This implementation allows the user to extract the top k recommendations from for each test data that is supplied. Given that LSH provides an exact match we wanted to use an ensemble of the results. We chose to extract the top 50 matches issues for each narrative in the test set and then took the highest occurring issue and also the top 2 occurring issue type. We then proceeded to compare it to the distribution of issues in the test dataset. The reason we did this was because the probability of randomly picking out an issue from the test set would be the ratio of the occurrence of the issue over the total number of issues. The below chart illustrates the comparison when we took out the highest occurring issue for each test complaint narrative. We obtained an accuracy of 30% for the top issue prediction and an accuracy of 50% for the top two issues prediction.



We find that apart from the highest occurring issue the model does not do a very good job at predicting the issue type. Note that we had seen in the exploratory analysis that there were multiple issue types which seemed very similar and that there was a good chance that this might lead to some misclassification. We also tried to take the top 2 occurring issue types by proportion to understand if there was any improvement in the model performance. The chart depicting the results is shown below.



We clearly see an improvement in the allocation of an issue type with the increase in prediction accuracy across all categories. Although this is not a very effective method to predict the issue type we chose this due to the similarity of the issue tags across the data. The true effectiveness of this technique can be tested by using very clearly defined categories where no such overlap is present.

4.2. Naive Bayes

The measure that we chose to check the effectiveness of our model is Accuracy. Summarized below are the results that we were able to obtain with selecting just 1 issue (Hard Boundary) and selecting the top two issues with the highest probability (Soft boundary). It is to be noted that there is a noticeable increase in accuracy with soft boundaries because the consumer narratives in our dataset are often very similar to one another, but they belong to different Issue labels.

The Naïve Bayes model was trained on a sample of 25,000 rows of the dataset due to memory limitations. It was then tested on a separately sampled dataset of 10,000 rows.

	MultinomialNB	ComplementNB
Top Result Accuracy	58.67%	55.35%
Top 2 results Accuracy	81.55%	77.47%

4.3. Support Vector Machines (SVM)

The SVM was observed to perform poorly as compared to the Naïve Bayes. We believe this is because of the high overlap between the Issues and the decision boundary created by the SVM was not able to classify correctly. The SVM model was trained and tested on the same sample of data as the Naive Bayes model.

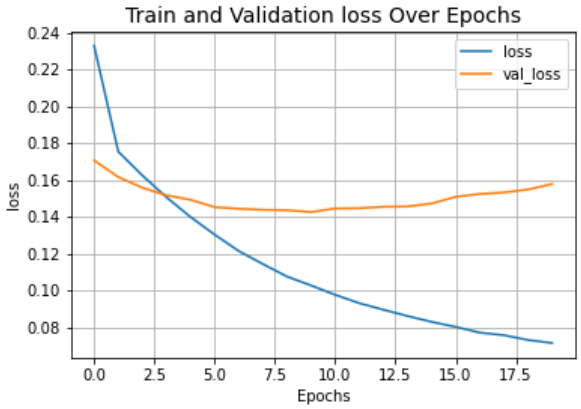
	LinearSVC
Accuracy	58.69%

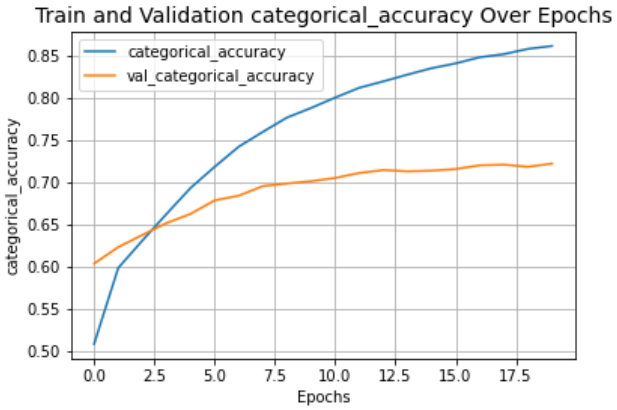
4.4. Artificial Neural Network

The ANN was trained on a total of 165914 rows, validated on 9218 rows and tested on 9218 rows as well. The Following is the result that we were able to obtain.

	ANN	Epochs
Accuracy	70.89%	20

The neural network was able to learn the subtle differences in different Issues much better than Naïve bayes or SVM.





The model was overtraining at first, that is why we used Dropout layers with such a high parameter. Even now, after about 12 epochs we started to see a rise in the validation loss whereas categorical accuracy more or less remained the same while training accuracy increased.

5. DISCUSSION

In this study we aimed to predict the issue category based on the narrative of the complaint filed by the customer. The results of the study brought out some key insights which need to be carefully understood so that text categorization can be more accurate.

We noticed that in the data the issue categories were not distinct and were overlapped in a few cases. This resulted in some errors in the prediction of the issue categories. Since this is a supervised technique, it is very important that the tagging be accurate. This is very important for companies that want to deliver great customer service as the complaints are a pain point for a customer and any delay and inaction would result in the negative perception of the brand. Some of the steps that the organization could take is to have a human overwrite the categorization in cases where they deemed it.

One of the assumptions of the Naïve Bayes classifier is that the features in the dataset are mutually independent [15]. In this example we saw that there were instances where this was not the case. This was further strengthened by observing the classification of the similar issues as the issue with the second highest occurrence for a particular complaint narrative. Hence, one needs to be careful while choosing the model when classifying textual data. Having said that, the model still performs well when predicting the top two issue categories.

For the implementation of LSH we observed that the model performs satisfactorily even though the model carries out matching using shingles and similarity. However, we must point out that several narratives can have the same words in complaint data since there tends to be a section where the customer might establish the relationship between them and the company which might be the same for most of the narratives. Apart from this we also think that since the issue categories in this dataset seemed to overlap there is a chance that there was some misclassification. Hence, the classification with LSH needs to be done keeping these issues in mind.

Finally, we found that ANN was able to carry out the classification better than the other methods. According to us the main driver for this is because it was able to pick out the subtle differences between the narratives that were mapped to issues which the other methods were not able to pick out.

6. PROJECT TAKEAWAYS / FUTURE WORK

We believe that using the above methods are good for classification of large corpus of textual data owing to efficient algorithms such as LSH. However, in order to get high accuracy, we also learnt that it is very important that the training data have accurate classification as the algorithms such as LSH tend to perform exact matches for calculation of similarity. In other algorithms such as Naïve Bayes the independence of the features is an important assumption failing which the model's performance would deteriorate.

For further test out the accuracies of the above models for classification we would recommend that the study be done on more accurately tagged data. This would help in providing insights on the effectiveness of the modelling techniques used. But the results obtained from a neural network are promising even on such data.

7. KEY WORDS

Text classification, LSH, Locality Sensitive Hashing, Naïve Bayes, ANN, Artificial Neural Network, Keras

9. REFERENCES

[1] Minaee, S., Kalchbrenner, N., Cambria, E., Nikzad, N., Chenaghlu, M., & Gao, J. (2021). Deep learning--based text classification. ACM Computing Surveys, 54(3), 1. https://doi.org/10.1145/3439726

[2] Google. (n.d.). Retrieved from https://www.google.com/search/howsearchworks/how-search-works/detecting-spam/.

[3] P.p.1; Minaee et al; Deep Learning Based Text Classification: A Comprehensive Review , https://arxiv.org/pdf/2004.03705.pdf

[4] Anthony, J. (2021, April 6). 106 Customer Service Statistics You Must See: 2020/2021 Data & Analysis. Financesonline.com. 1. Retrieved from https://financesonline.com/customer-service-statistics/.

[5] Reyes, S. (2019, May 13). Consumer Complaint Database. Kaggle. Retrieved from https://www.kaggle.com/selener/consumer-complaint-database.

[6] Locality-Sensitive Hashing . Wikipedia. (2021, November 30). Retrieved from https://en.wikipedia.org/wiki/Locality-sensitive_hashing.

[7] Broder, A. Z. (n.d.). Identifying and filtering Near-Duplicate Documents. Retrieved from https://cs.brown.edu/courses/cs253/papers/nearduplicate.pdf

[8] Naive Bayes Classifier. Wikipedia. (2021, November 5). Retrieved from https://en.wikipedia.org/wiki/Naive_Bayes_classifier.

[9] Naive Bayes. scikit. (n.d.). Retrieved from https://scikit-learn.org/stable/modules/naive_bayes.html.

[10] Support-Vector Machine. Wikipedia. (2021, December 6). Retrieved from https://en.wikipedia.org/wiki/Support-vector_machine.

[11] Artificial Neural Network. Wikipedia. (2021, December 2). Retrieved from https://en.wikipedia.org/wiki/Artificial_neural_network.

[12] Paul, S., & Rakshit, S. (2020, September 25). Keras documentation: Large-scale multi-label text classification. Keras. Retrieved from https://keras.io/examples/nlp/multi_label_classification/.

[13] Consumer Financial Protection Bureau. (n.d.). Retrieved from https://www.consumerfinance.gov/.

[14] Datasketch. MinHash LSH Forest - datasketch 1.0.0 documentation. (n.d.). 2. Retrieved from http://ekzhu.com/datasketch/lshforest.html.

[15] Raschka, S. (2014, October 4). Introduction and Theory. 3. Retrieved from https://arxiv.org/pdf/1410.5329.pdf.

[16] Shimodaira, H. (2014). Text classification using naive bayes. *Learning and Data Note*, *7*, 1-9. https://www.inf.ed.ac.uk/teaching/courses/inf2b/learnnotes/inf2b-learn07-notes-nup.pdf

[17] Colas, F., & Brazdil, P. (2006, August). Comparison of SVM and some older classification algorithms in text classification tasks. In IFIP International Conference on Artificial Intelligence in Theory and Practice (pp. 169-178). Springer, Boston, MA. https://link.springer.com/content/pdf/10.1007%2F978-0-387-34747-9_18.pdf