# The Automated Ticket Router System

by

Ramavtar

A thesis submitted in conformity with the requirements for
the degree of Diploma in Applied Artificial Intelligence

in the
Faculty of Computer Science
School of Computer and Information Science

Supervisor: Prof. Brahmma Reddy

on

10/03/2022

# Declaration

I, Author, declare that this thesis “The Automated Ticket Router System” is my own work, that it has not been submitted before for any degree or assessment at any other university, and that all the sources I have used or quoted have been indicated and acknowledged by means of complete references.

[illegible]

Signature:. . . . . . . . . . . . . . . . . . . . . . . . Date:

*“We’re making this analogy that AI is the new electricity. Electricity transformed industries: agriculture, transportation, communication, manufacturing.”*

Andrew Ng

## Abstract

We know that amongst all the companies, the routing of tickets raised by both internal users and external customers across departments has been a major challenge for years. Over the years, due to rising number of complaints, this has only added to the complexity of matter. This project takes consumer court complaints as one of examples to analyze the gravity of such problem and provide the AI based solution to expedite the routing of tickets across departments. We aim to reduce human effort and errors involving in handling the routing of consumer tickets, also minimizing the delay. The solution tries multiple machine learning models ranging from linear models like logistic regression, linear SVC to complex ensembling based techniques like Random Forest and LightGBM classifier based on multiple error metrices like precision, recall and F1 score. We then consider the best performing model and create a WebAPI based solution around it which is built using Flask and deploy the same on AWS.

## Keywords

WebAPI, Machine learning, Automated ticket routing, Dimensionality reduction, Data aquisition, Exploratory data anaylysis, High dimensionality data visualization, Hyper paramater tunning, Model productionization.

# Acknowledgements

I am heartfully thankful for Mr. Brahmma Reddy for his continued assistance provided during the various stages of this project.
It was an honour to work under him as he gave the much required exposure to do the research in this field.
Without his continued help and encouragement, this project would not have materialized.

# Contents

# List of Figures

# Abbreviations

| | |
|---|---|
| **PCA** | **P**rinciple **C**omponent **A**nalysis |
| **TSNE** | **T** **D**istributed **S**tochastic **N**eighborhood **E**mbedding |
| **LR** | **L**ogistic **R**egression |
| **NB** | **N**aive **B**ayes |
| **SVM** | **S**upport **V**ector **M**achine |
| **RF** | **R**andom **F**orest |
| **LGBM** | **L**ight **G**radient **Boosting** **M**odel |
| **OHE** | **O**ne **H**ot **E**ncoding |
| **LE** | **L**abel **E**ncoding |
| **EDA** | **E**xploratory **D**data **A**nalysis |
| **NLP** | **N**atural **L**anguage **P**rocessing |

# Chapter 1

# Introduction

The ticket routing problem is not new to us and is prevalent across all sorts of organizations. Many such examples are there starting from simple incident tickets or service requests in an IT firm to large number of pending cases in consumer courts. For this project, we consider the problem of delay in consumer court. The Problem At Hand. Millions of customers, thousands of companies, thousands of products and services and millions of pending consumer cases and months and sometimes years of delays in justice. This is not some sort of fancy tag line but the hard truth that customers face. No business can claim 100% customer satisfaction on their products or services. This can often lead to disputes among companies and their customers. To get the justice, customers reach out to consumer court. However court has its own procedure and no promised timelines to resolve the consumer complaints. The tickets line up for days and keep on rotating across departments before actual action is initiated. Sometimes, the same ticket gets rotated and stuck in wrong departments for days adding to further delay. This leads to customers dissatisfaction and loss of trust in justice. Over the years, the consumer complaints have risen exponentially and so are the delays.

## 1.1 The Problem at Hand

Millions of customers, thousands of companies, thousands of products and services and millions of pending consumer cases and months and sometimes years of delays in justice. This is not some sort of fancy tag line but the hard truth that customers face. No business can claim 100% customer satisfaction on their products or services. This can often lead to disputes among companies and their customers.

To get the justice, customers reach out to consumer court. However court has its own procedure and no promised timelines to resolve the consumer complaints. The tickets line up for days and keep on rotating across departments before actual action is initiated. Sometimes, the same ticket gets rotated and stuck in wrong departments for days adding to further delay. This leads to customers dissatisfaction and loss of trust in justice. Over the years, the consumer complaints have risen exponentially and so are the delays.

In early days, we had manual experts involved who would read the complaints and route them to different departments. There are multiple problems associated with this approach.

- Experts newly hired may not have enough experience to understand the context of problem.
- The tickets may not have enough context itself, because customer may not have provided them.
- There may be large number of tickets across the location which makes it humanly impossible.
- The cost of man power involved to route the ticket to appropriate department may not be sufficient enough.

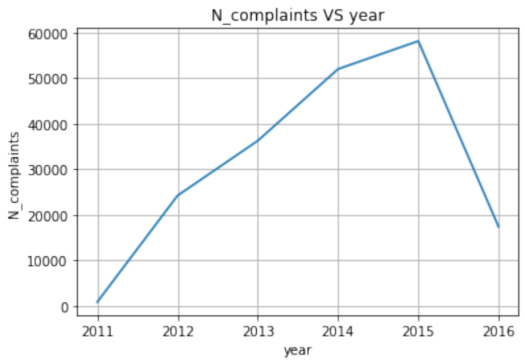


FIGURE 1.1: Rising number of consumer court cases

## 1.2 Inspiration

This leads us to below question.

- How can we handle the large volume of user tickets?
- How can we expedite the ticket routing across departments?
- How can we reduce human errors and intentional or unintentional delays?
- How can we design a solution with minimal cost?

This inspired us to come up with AI driven solution to design an Automated ticket router system to route the ticket to appropriate department keeping the above constraints in mind. This solution will have below advantage over the regular manual approach to filter the tickets across departments.

- No need of human involvement for handling the routing of tickets.
- Cost saving.
- Operates 24X7.
- Number of tickets not a problem anymore.
- Speedy routing of tickets.
- More correct routing of tickets to any department as compared to human.
- WebAPI based solution provides easy to understand user interface.
- Builds customer relation with company if they take their customers seriously because the quicker their resolution, the better the relation becomes.
- Restore and retains the faith of people in justice system.

# Chapter 2

# Literature survey

There are many incredible works done by the researchers to handle the ticket routing problem across the globe. These are primarily based on AI based solutions involving classical machine learning and deep learning based techniques.

Aixin Sun et al. [1] divide the problem into initial group assignment and inter-group transfer. They explore interrelationships between the group assignment and group transfer problems using UFTR, a Unified Framework for Ticket Routing using four types of features (derived from tickets, groups, and their interactions).

Paramesh et al. [2] take IT service desk data and first mine the unstructured historical data. Then they use traditional ML approach to find the relationships among the features and target department. To further improve the score, they use Bagging, Boosting and Voting ensemble techniques to further improve the accuracy.

Shimpi et al. [3] makes use of domain knowledge based feature generation and information retrieval based techniques for devising issues from the content of customer tickets. She also uses clustering approach to find the underlying hidden issues from system and user tickets

Potharaju et al. [4] analyzes the network issue tickets, troubleshooting activities on them and proposes NetSieve which makes use of NLP techniques and Ontology modelling to infer the problem symptoms.

S. Roy et al [5] takes user description, combines them into multiple clusters and then use these clusters to categorize the user tickets. These user tickets have fixed and flexible fields and author proposes Jaccard and cosine distances respectively for each of them and then combines these 2 distances to create altogether a new distance metric which measures the distance between tickets. Kmeans clustering is applied on these distances to group the ticket into different ticket clusters. Our goal in this paper is to capture meaning from human natural language through NLP and provide an automated solution for aiding the process of service ticket solving, through the intelligent classification of tickets, pattern recognition and similarities between texts. The difficulty of this task lies in translating the human language Our goal in this paper is to capture meaning from human natural language through NLP and provide an automated solution for aiding the process of service ticket solving, through the intelligent classification of tickets, pattern recognition and similarities between texts. The difficulty of this task lies in translating the human language Our goal in this paper is to capture meaning from human natural language through NLP and provide an automated solution for aiding the process of service ticket solving, through the intelligent classification of tickets, pattern recognition and similarities between texts. The difficulty of this task lies in translating the human language

Mati Cristian et al. [6] use NLP based approach to understand the relation between the various texts to classify the IT tickets as service request or incident requests.

Qihong Shao et al. [7] proposes a solution called easyTicket which works on Markov modelling technique, where first the model learns or records the set of already passed on groups with respect to each other, i.e. in the past from which group the ticket has been passed to which group. The model learns the sequence of interconnected groups (Social networking). Then these learnt behaviors are then applied altogether to new data to predict the best possible group.

Li and Jain et al. [8], takes individual vs combination of base classifiers and the concept of document classification to handle the ticket routing problem.

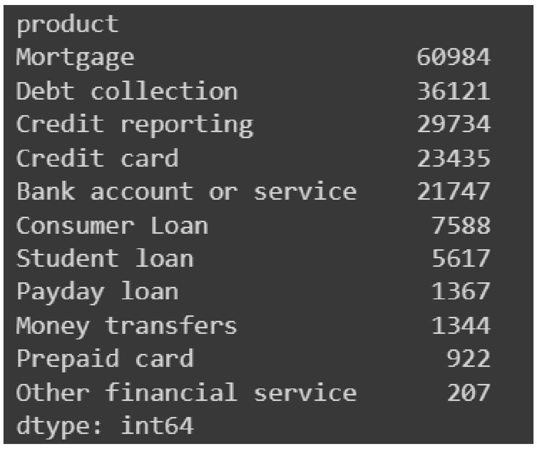
Ikonomakis et al. [9] describes the various stages in text classification process e.g. data preprecossing, feature selection and transformation etc. and then they use the various classification models.

A.Khan et al [10] explains various machine learning techniques for text document classification. Also text mining techniques like feature extraction and selection are discussed in detail as well.

Chapter 3

# Data acquisition and Pre-analysis

We have used freely available public dataset "consumer comaplaints" from Kaggle which is of size 5M and considered 0.2 M data points sampled randomly. The data consists of 18 columns, almost every column containing null value. Majority of columns are categorical except consumer_narrative which is text based column, only 12 % of which contains not null value. We have considered those columns which are relevant while raising the consumer ticket. We used the most frequent category in every column for filling out the missing values. The challenging task we are trying to do here is classify the tickets into multiple departments (considering each product in product column representing the unique department). The dataset has lot of imbalance.



```
product
Mortgage                   60984
Debt collection            36121
Credit reporting           29734
Credit card                23435
Bank account or service    21747
Consumer Loan               7588
Student loan                5617
Payday loan                 1367
Money transfers             1344
Prepaid card                 922
Other financial service      207
dtype: int64
```

FIGURE 3.1: Data imbalance

We chose multiclass precision and recall and also F1 score(Micro averaged) as business metrics to assess how well our models are performing with the multiclass logistic loss being the technical metric for the models to converge to the best point possible. The reason for choosing them as business metric is that all classes are equally important and hence precision, recall and F1 score simultaneously become important metric here. With them, we can choose a threshold of our choice and simultaneously multiple metrics like FNR, FPR and TPR are automatically taken care. In this way, we ensure that we are sufficiently testing our models before forming any conclusions. We have used confusion matrix, precision and recall metrices to showcase the misclassification % across different classes. while applying the models we need to ensure the below business constraints as well.

- Generating wrong prediction of class should be avoided. If there are 3 classes, C1, C2 and C3 then C1 should not be misclassified as C2 or C3 (reduce FN). At the same time, C2 and C3 should not misclassified as C1(reduce FP).
- Latency to a certain extent is OK. Since it takes consumer complaints to solve. Immediately routing the tickets to concerned departments can take few seconds of time.
- Interpretability as to why particular feature is important.

Please note that we have sourced data from Dataset `https://www.kaggle.com/sebastienverpile/consumercomplaintsdata`

# Chapter 4

# Proposed methodology and workflow

We have chosen 0.2 M samples and split it into train and test. We dropped the unnecessary columns from each. We used one hot encoding to encode categorical variables for each of train and test data. For the consumer_narrative column, we preprocessed each data point by removing the html tags, non english alphabets. For zipcode column, which contains few thousand categories, we replaced it with region_code column which is 10 category variable by considering to divide US into 10 different regions. One of the trickiest problem we faced was with the company_name variable which against consist of many unique values. So we did preprocessing on these company names and replaced it with k category categorical variable where k is the number of best possible clusters generated by KMeans algorithm which we used to cluster various companies into k clusters.

Then we combined all of these features into one. We have tried different machine learning models like Naïve bayes, LR, Random forest, LGBMClassifier and measured them on micro weighted average recall, precision and F1 score and compared them simultaneously and found the best working model.

This model was deployed on AWS, we built a front end where the user needs to enter the necessary ticket details. The model will predict the probability for each class and also the name of best predicted department.

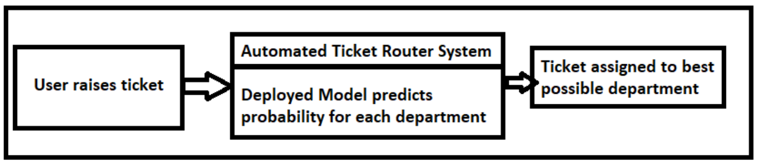

User raises ticket
Automated Ticket Router System
Deployed Model predicts probability for each department
Ticket assigned to best possible department

Figure 4.1: Overall workflow

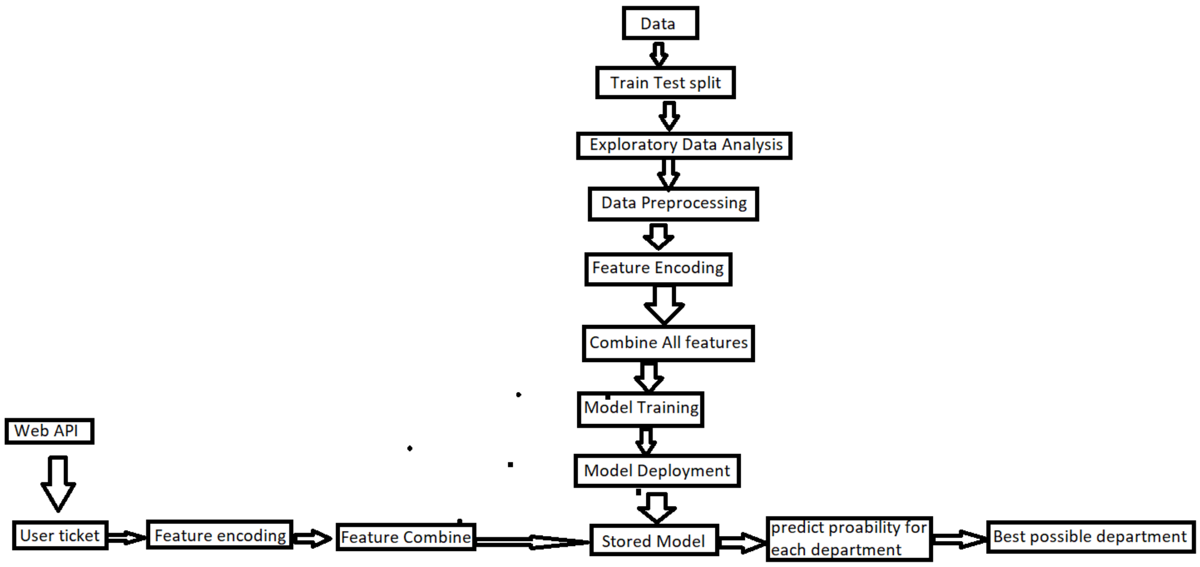


Figure 4.2: Complete workflow

Chapter 5

# EDA and Feature extraction

We consider EDA (exploratory data analysis) as first hand approach to understand the various aspects of the data. This could be a visualization or graph based approach, numerical analysis using quantiles or descriptive analysis. We have considered a mix of these approaches.

## 5.1 Exploratory data analysis

EDA is data visualization based technique used to analyze data by trying to find some patterns around the different categorical or numerical features. Here can perform plot the distribution plot, box plot, bar graphs in case of categorical features etc. We can see if there are any outliers using the box plot or violin plot. EDA are easy visual representation of data. We can compare the bar graphs of different features and do a comparative analysis of individual categories. However since its graphical approach we can not know the exact data related details since on graph, the plotting of data to very granular level could be very difficult specially if data size is very large. Besides we can see if there are outliers but we can't find the exact data point which is an outlier.

## 5.2 Numerical analysis

While EDA is a visualization technique it does not necessarily tell the exact value of mean or outliers. We need to use percentile or quantile information to analyze the data in the more granular way. So that we can filter out the exact outlier.

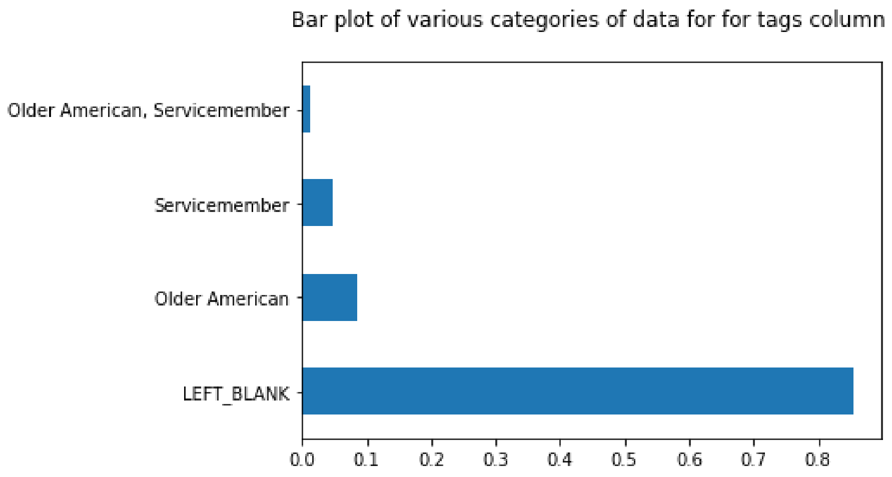


Figure 5.1: Tags Category distribution

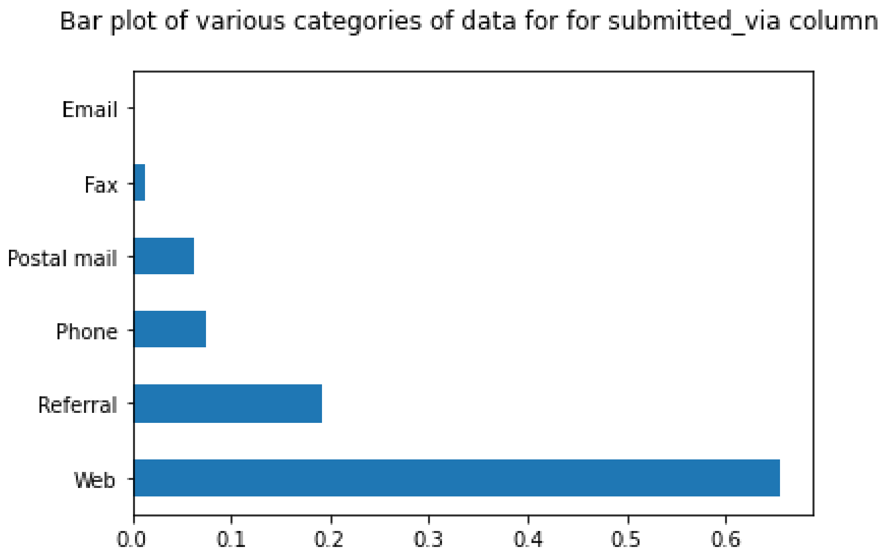
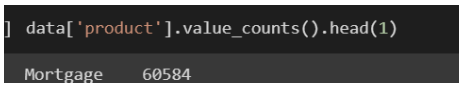


Figure 5.2: Submitted Via Category distribution

## 5.3 Descriptive analysis

We can analyze the data straight away asking important questions about the data from user's perspective, totally based on what we want from the data. e.g.

Q. Which product the complaints are most frequent in nature.

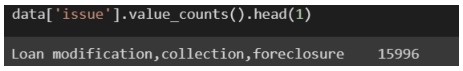


```
] data['product'].value_counts().head(1)

  Mortgage    60584
```

Figure 5.3: Most frequent product

Q. What type of complaints is very general in nature? [issue, sub-issue]?

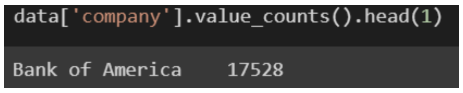


```
data['issue'].value_counts().head(1)

Loan modification,collection,foreclosure    15996
```

FIGURE 5.4: Most frequent issue

Q. Which companies have low reputation in general?

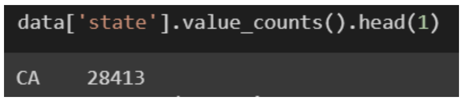


```
data['company'].value_counts().head(1)

Bank of America    17528
```

FIGURE 5.5: Most frequent company

Q. Which state has most complaints?

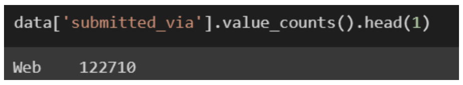


```
data['state'].value_counts().head(1)

CA    28413
```

FIGURE 5.6: Most frequent state

Q. Which means is more frequent way of complaints to be filed?



```
data['submitted_via'].value_counts().head(1)

Web    122710
```

FIGURE 5.7: Most frequent way of complaint raise

Please note that above stated methods are applicable for both univariate and multivariate analysis.

## 5.4 Encoding methods

While we have plethora of options to encode categorical features using say one hot encoding, ordinal or label encoding, probabilistic encoding etc. we have considered one hot encoding technique to create one hot encoded numeric vector representation given any categorical feature. Plus side is that it does not consider any precedence of category whereas on the downside It increases the dimensionality of the data.

## 5.5 Advanced Encoding techniques

For text featurization, we have used TFIDFVectorizer to create numerical representation of text doing a unigram and bigram as well. Plus side is that it is simple to understand as it is frequency based. It creates sparse matrices so save a lot of space. However the downside is that It does not take care of semanticity of words. Due to large number of features time complexity is high.

## 5.6 Special handling of zipcode and company_name feature

Due to large number of zipcode categories, we had to replace them with the region_code feature which is a 10 class categorical feature since we logically US has been grouped into 10 different region codes ranging from 0 to 9.

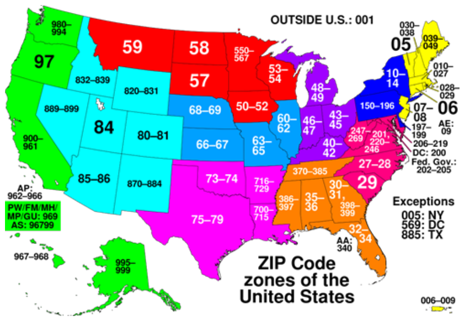


FIGURE 5.8: Color coded region code representation of US.

On the other hand company name feature had multiple text based categories. We tried out a unique approach of clustering these names into k different clusters.

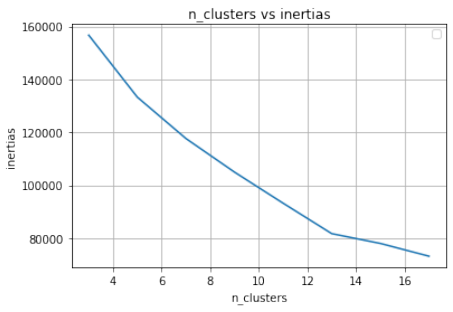


FIGURE 5.9: Best number of clusters using kmeans clustering.

## 5.7 High dimensional data visualization

We can use PCA or TSNE to see how any higher dimensional data appears on 2D plane. This gives us a feeling whether product categories are separable with respect to each other. For our case, we considered PCA, which tries to find the eigen vector by maximizing the variance across different axes. Advantage of using PCA is that it maintains the global structure of the data and takes less time to execute as compared to other dimensionality reduction techniques. Whereas on the other side It breaks the local structure of the data. There is much information that gets lost while data is being projected along maximum variance axis. While TSNE improves upon the lag point of PCA by keeping the local and global structures intact, time complexity of TSNE is huge.

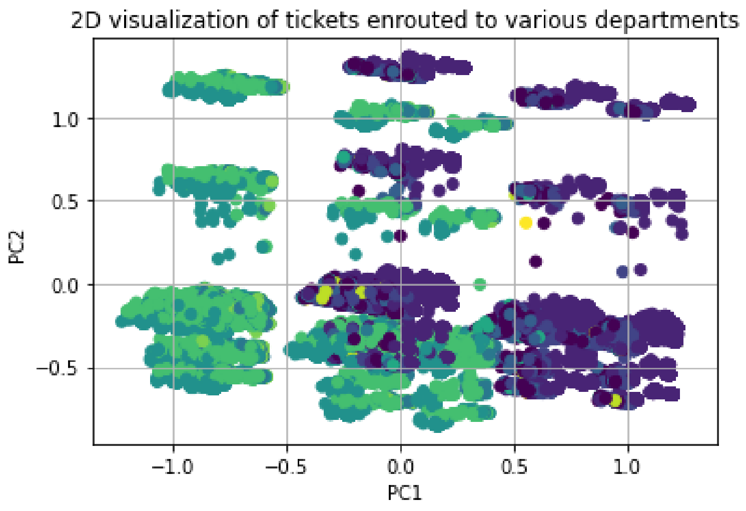


Figure 5.10: Tickets distribution for various department on 2D using PCA.

Chapter 6

# Modelling and Error Analysis

## 6.1 Metrics

Our model should be able to remember the classes predicted correctly. So if a complaint is about mortgage it should be classifier as mortgage. Also other product like loan should not be classified as mortgage. Which means both precision and recall are important and hence F1 score. Along with that, for our model should be able to converge to the best possible loss value, that is why we chose multiclass log loss or categorical cross entropy as error metric. By choosing precision, recall we are able to take care of False negative rate, False positive rate and True positive rate.

Recall will fail when we want to predict both the classes equally well. It should not be used where the correct identification of the negative class does not play a role. Precision should not be used when positive class has very few observations.

Any problem in which one class is extremely important involves using recall. Cancer patients prediction problem where any cancer patient should not be predicted to have no cancer or the fraud transaction problem where the fraud transactions are extremely important.

Precision can be used in the problems like youtube song recommendations where generation of too many suggestions is OK but not the wrong one. Other case where we want to look for good restaurants so too many of them is fine but not the wrong one.

False positives should be minimum in case of precision and False Negative should be minimum in case of recall.

We have considered, micro average based recall, precision and F1 scores. We also considered confusion matrix to see how the misclassifications are happening and how well the models are performing. Since it is a multiclass classification problem so we think that an average of all the classes will be necessary to get overall score. We have macro and micro options in the sklearn library. 'Micro' option seemed to be the go to options since classes are imbalanced. Therefore we considered it.

```
def cross_entropy(y_pred, y_true, epsilon=1e-12):
    """
    Computes cross entropy between y_true (encoded as one-hot
   vectors)
    and y_pred.
    Input: y_pred (N, k) ndarray
           y_true (N, k) ndarray
    Returns: scalar
    """
    y_pred = np.clip(y_pred, epsilon, 1. - epsilon)
    N = y_pred.shape[0]
    cr_ent = -np.sum(y_true*np.log(y_pred+1e-9))/N
    return cr_ent

def precision(y_pred, y_true):
        precision = (y_pred * y_true).sum() / y_pred.sum()
        return precision

def recall(y_pred, y_true):
        recall = np.sum(y_true * y_pred))/np.sum(y_true)
        return recall

def f1_score(y_pred, y_true):
        precision = (y_pred * y_true).sum() / y_pred.sum()
        recall = np.sum(y_true * y_pred)/np.sum(y_true)
        f1_score = 2*precision*recall/(precision+recall)
        return f1_score
```

## 6.2 Models

We have used models ranging from simple models like naïve bayes and logistic regression to complex tree based models like random forest and lightgbm.

### 6.2.1 Naïve bayes

Naïve bayes is a simple frequency based model which has been found to working great for the text based features. Sometimes it can beat deep learning models as well. It works on the principle of Bayes theorem which is based on conditional probability. It assumes that features are conditionally independent. It has been found practically that even though the features are partially independent, naive bayes still works very well.

### 6.2.2 SVM

SVM is an ML model which works on margin maximization principle. There are both linear and non linear versions of SVM but we tried only linear SVM through SGDClassifier hinge loss method. The result was pretty good.

### 6.2.3 Logistic regression

Logistic regression is a linear model which tries to find the best separating hyperplane. We used this, because there are were too many features which makes the dataset linearly separable in higher dimensions. .

### 6.2.4 Random Forest

It's a bagging technique to reduce overfitting caused by individual decision trees. We wanted to try tree based model and see how well it performs on this dataset. Like other models it performed very well giving a very high recall, precision and f1 score. Time to train and predict the data is more than linear models. The hyper parameter tunning stage was the most time taking part. So from time complexity perspective, it was not that a great option.

### 6.2.5 Lightgbm

Light gradient boosting is boosting based technique to improve the bias score by averaging out the contributions of underfit models such as shallow decision trees. It tries to retrain the residuals at multiple stages and reevaluating them until they stop changing. It performed very well as well, however as one can guess, time to train and predict the results was huge so probably it is not that a great choice, the most cumbersome stage being obviously the hyper parameter tunning.

### 6.2.6 KMeans

KMeans is a supervised learning based technique, which tries to create equal size cluster based on the similarity of data points with respect to centroids. The intention was to group so many company categories into few. By doing the hit and trial and computing the sum of intra cluster distances (called inertia), We found that 11 was the best number of clusters in which the companies can be grouped together.

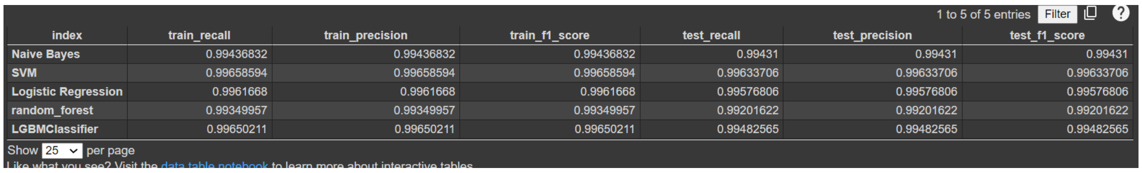
## 6.3 Hyperparameter tunning

We have used Grid search for performing hyper parameter tunning. Grid search takes a parameter grid of different values for a parameter and the model is trained using all of these values with the chosen cross validation value. We have plotted the graphs for train and test scores for different values of hyper parameters for each of the models to get the best score.

Like Grid search, we have other options e.g. RandomSearch which provides an option to choose random combinations of different parameters. This will reduce the time complexities of Grid search which uses every combination of parameters.

## 6.4 Result comparison

Added below is the comparison of all the machine learning models on the basis of business metrices and latency in training and testing. Compare using best models (model with appropriate hyper parameters).



1 to 5 of 5 entries Filter

| index | train_recall | train_precision | train_f1_score | test_recall | test_precision | test_f1_score |
|---|---|---|---|---|---|---|
| Naive Bayes | 0.99436832 | 0.99436832 | 0.99436832 | 0.99431 | 0.99431 | 0.99431 |
| SVM | 0.99658594 | 0.99658594 | 0.99658594 | 0.99633706 | 0.99633706 | 0.99633706 |
| Logistic Regression | 0.9961668 | 0.9961668 | 0.9961668 | 0.99576806 | 0.99576806 | 0.99576806 |
| random_forest | 0.99349957 | 0.99349957 | 0.99349957 | 0.99201622 | 0.99201622 | 0.99201622 |
| LGBMClassifier | 0.99650211 | 0.99650211 | 0.99650211 | 0.99482565 | 0.99482565 | 0.99482565 |

Show 25 per page

Like what you see? Visit the data table notebook to learn more about interactive tables.

FIGURE 6.1: Tickets distribution for various department on 2D using PCA.

## 6.5 Error analysis

We have used confusion matrices, precision and recall matrices for each of the models mentioned above to see among which classes the models are getting confused. We found that the confusion was due to reason that they looked almost similar e.g. Fraud and scam v/s unauthorized transactions. Other reason could be less number of data points for the issues.

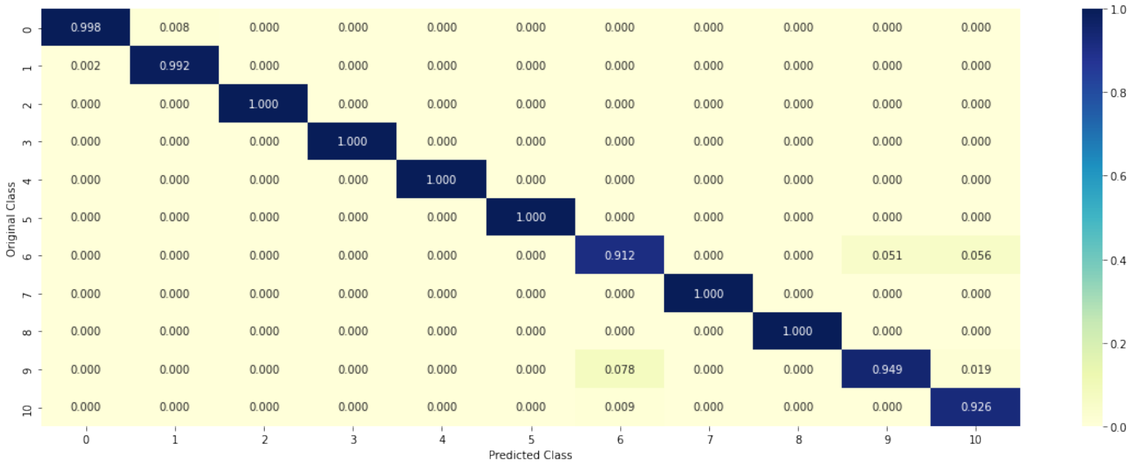
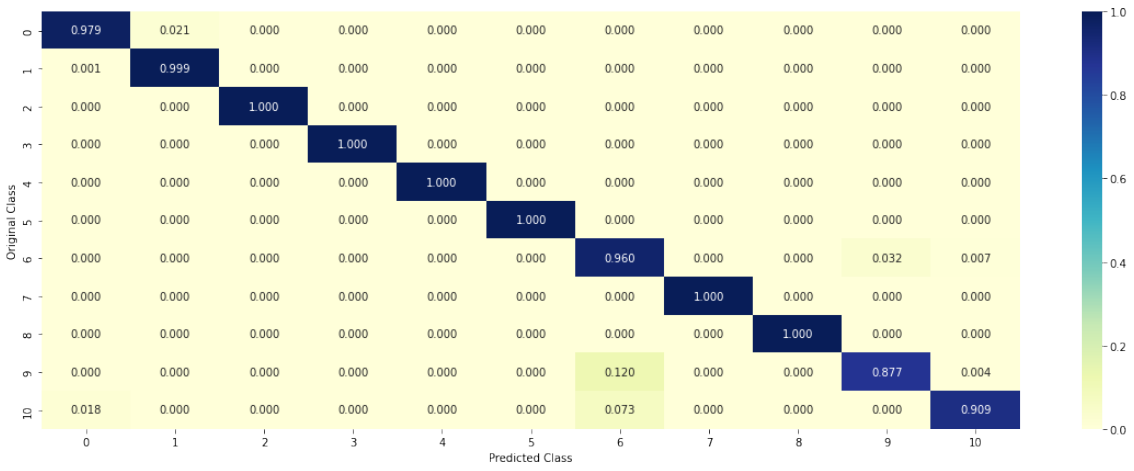


FIGURE 6.2: precision matrix

## 6.6 Result

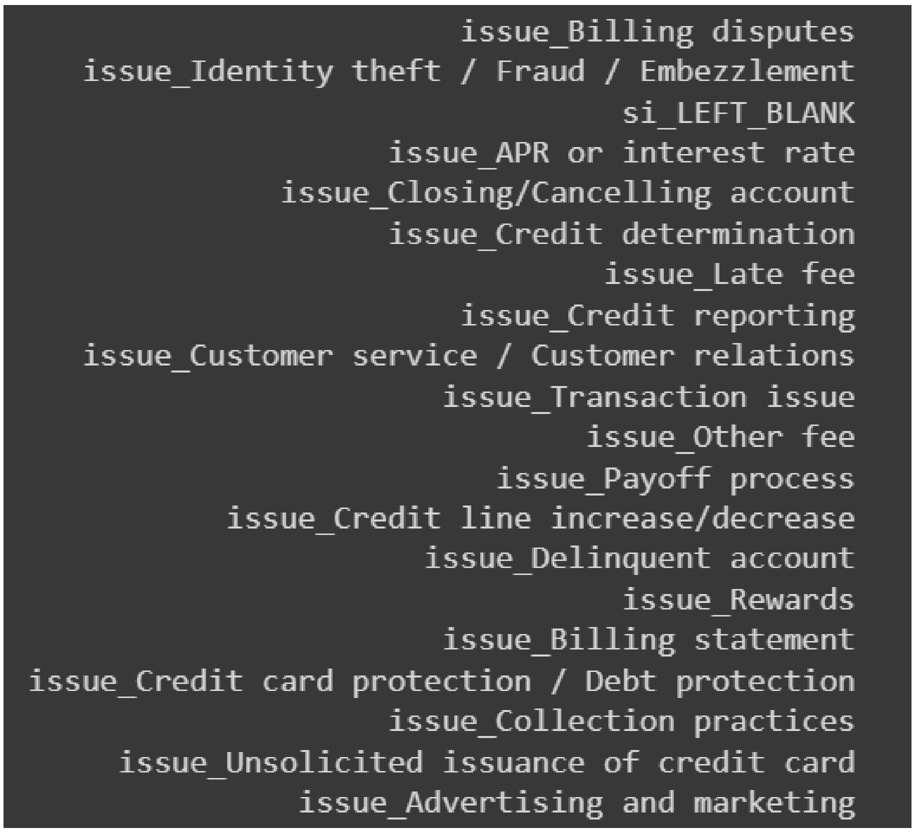
With the above comparison, we found that SVM outperformed all of the models with the best score as. We further improved the score with hyper parameter tunning using Grid search. Please note that the model used here is multiclass logistic regression model, so weight vector received is a multiclass array, each vector representing separate class weight vector.

We found that issue and sub_issue were the most important features amongst all which seemed very logical to us.



| Original Class \ Predicted Class | 0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 0.979 | 0.021 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| 1 | 0.001 | 0.999 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| 2 | 0.000 | 0.000 | 1.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| 3 | 0.000 | 0.000 | 0.000 | 1.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| 4 | 0.000 | 0.000 | 0.000 | 0.000 | 1.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| 5 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 1.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 |
| 6 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.960 | 0.000 | 0.000 | 0.032 | 0.007 |
| 7 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 1.000 | 0.000 | 0.000 | 0.000 |
| 8 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 1.000 | 0.000 | 0.000 |
| 9 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.120 | 0.000 | 0.000 | 0.877 | 0.004 |
| 10 | 0.018 | 0.000 | 0.000 | 0.000 | 0.000 | 0.000 | 0.073 | 0.000 | 0.000 | 0.000 | 0.909 |

FIGURE 6.3: recall matrix



```
                              issue_Billing disputes
     issue_Identity theft / Fraud / Embezzlement
                                   si_LEFT_BLANK
                        issue_APR or interest rate
                   issue_Closing/Cancelling account
                        issue_Credit determination
                                    issue_Late fee
                           issue_Credit reporting
     issue_Customer service / Customer relations
                           issue_Transaction issue
                                   issue_Other fee
                              issue_Payoff process
                issue_Credit line increase/decrease
                          issue_Delinquent account
                                     issue_Rewards
                           issue_Billing statement
  issue_Credit card protection / Debt protection
                        issue_Collection practices
       issue_Unsolicited issuance of credit card
                  issue_Advertising and marketing
```

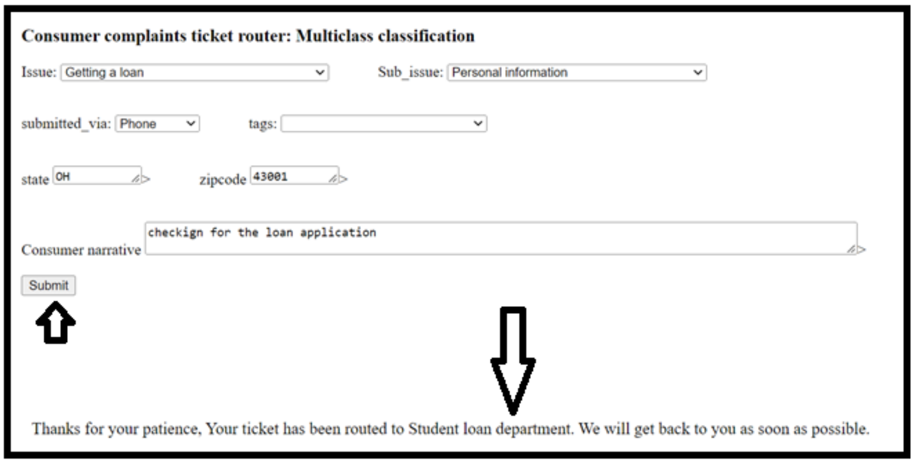
FIGURE 6.4: Feature importance

Then we further investigated on that trying out different features one by one (forward feature engineering method) and checking on the contribution of each feature towards the final result. And as our earlier model suggested, issue and sub-issue contributed the maximum towards results. So we decided to simplify our solution and limit the features to only issue and sub_issue.

Chapter 7

# Productionization and model testing

While its important that user tickets are routed fast, its important as well to provide an easy interface to the user to enter the details of their ticket as well. We therefore, propose a webAPI which is built using Flask. We then deployed it on AWS EC2 instance. The webAPI when it receives the user input, it converts the entered input internally into numeric vectors and combine them to create the test data point. This data point is received by the deployed model which predicts a best possible department.



**Consumer complaints ticket router: Multiclass classification**

| | | | |
|---|---|---|---|
| Issue: | Getting a loan | Sub_issue: | Personal information |
| submitted_via: | Phone | tags: | |
| state | OH | zipcode | 43001 |
| Consumer narrative | checkign for the loan application | | |

Thanks for your patience, Your ticket has been routed to Student loan department. We will get back to you as soon as possible.

Figure 7.1: WebApp

## 7.1 Testing the model and its results

We have used multiple business metrices like precision, recall and F1score to assess the working models. Plotted confusion, precision and recall metrices to locate the weaker zones where our models do not work properly. Its never that bad to add further checks to test the strength and endurance that our model can exhibit by doing few or all of below.

### 7.1.1 Outlier handling

Outliers are not just the points very far from the main stream of data point region but also the points which the least frequent as well. In this case, data imbalance is very severe. For each column there are multiple categories and some of the categories have extremely few entries as compared to large data volume. E.g. for company column, there are some companies against which the there are very few complaints (say less than 5) as compared to large number of complaints against many other companies. Same pattern can be found for other columns such as 'issue', 'sub_issue', 'tags' etc. conceptually we can think of them as outliers.

Presence of outlier creates missing data issue between train and test data. We have used handle_unknown='ignore' option in the one hot encoder. We could have tried to handle these as separate case but the frequency of such points are is less as compared to total large volume of the data. With this, we are able to get a good train and test score.

### 7.1.2 Handling missing values

We are going to explain variable wise.

- Issue: No blank/missing values.
- Sub-Issue: Created another category 'LEFT_BLANK'.
- Tags: Created another category 'LEFT_BLANK'.
- Submitted_via: Replacing the missing values with the most frequent category 'Web'.
- State: Replacing the missing values with the most frequent category 'CA.

- Zipcode: To be dropped.
- Region_code: To handle too many zip code categories, we have created region code which is a 10 category categorical variable. For missing region code, we have replaced with most frequent category 3.

### 7.1.3 Noise handling

We added noise to the data and recomputed the performance with our final model and there was no change in the performance of the model.

### 7.1.4 Data imbalance

- We can certainly try to do up-sampling for the minority classes but in our case there are many minority classes and ratio between majority and minority class data points is extremely huge. If there are few classes and the difference between their count is not much (still showing data imbalance), then we can try upsampling. Also if we do up-sampling, we are simply repeating the same data points over and over. This may theoretically seem correct to do that however, we do have a section of removal of duplicate data. While on one side we are deduping the data, on the other side we are duplicating it by doing up-sampling. Even if we take the approach of generating too many data points and changing it here and there manually, it wont be feasible humanly to do so.

# Appendix A

Appendix A

## Logistic Regression

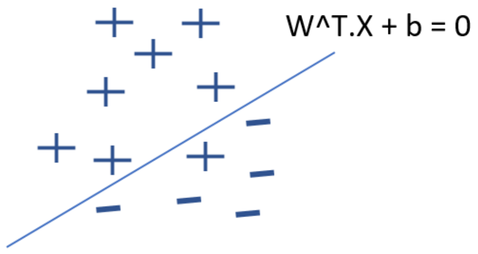


Figure A.1: Logistic regression

If d_i is the distance for any point from the hyperplane then sum of such distances should be maximum. Optimization equation becomes

W* $= \text{argmax} \sum_{i=1}^{n} z_i \quad \ni z_i >= 0 \quad where\ z_i = y_i(W^T.X_i) >= 0$

But this expression can't deal with the outlier since they have large negative values. So we would need a function which can generate smaller values (say between 0 to 1) for even large negative weights. One of such function is sigmoid. which is defined as

$$f(x) = \frac{1}{1+e^{-x}}$$

Then this optimization equation becomes

W* = argmin $\sum_{i=1}^{n}(1 + e^{-W^T x_i}) \qquad \ni\ y_i(W^T.X_i) >= 0$

Since logarithm is monotonic function we can re write above expression as

W* = argmin $\sum_{i=1}^{n}\log(1 + e^{-W^T x_i}) \qquad \ni\ y_i(W^T.X_i) >= 0$

One problem with this expression is that if there are large negative weights then this expression becomes 0 so we should add a regularizer term.

W* = argmin $\sum_{i=1}^{n}\log(1 + e^{-W^T x_i})$ + $\lambda\|W^T W\| \qquad \ni\ y_i(W^T.X_i) >= 0$

$\lambda$ being the hyperparameter which will control overfitting. Higher $\lambda$ means more underfitting and vice verse.
We thought that since we are considering too many dimensions (generated out of tfidf factorization of consumer narrative feature, so linear model like logistic regression can certainly perform well on the large dimension and medium size dataset with 0.1 M data points, since the data becomes linear in high dimensional space.
PROS:

- Easy to implement and train.
- Can give probabilistic interpretation of class distribution.
- Can handle high dimensionality and large size data.
- It is known to work exceptionally well for text and categorical variables.

CONS:

- Assumes linear relationship between the dependent data variable and independent class label. Can not handle the complex nonlinear relationship between the dependent and independent data variable.

REAL WORLD EXAMPLE:

- Given weight and exercise based data what is the probability of a person getting a heart attack.

# Appendix B

## Principle Component Analysis

PCA is dimensionality reduction technique which tries to find axes along which the variance is maximum. The basic concept here is that the axes along which the maximum variance of data is there contains maximum data.
In the below diagram, let $x_i$ be the data point. u be the maximum variance axis unit vector. Lets try to project the the data point $x_i$ and projected data point be $x_i^{'}$.

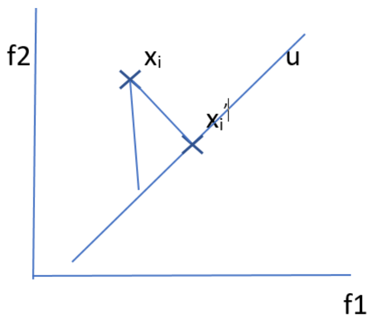


FIGURE A.2: PCA

consider the below equations.

$$\mathrm{x}_i^{'} = \mathrm{proj}_u x_i$$

$$\mathrm{x}_i^{'} = \frac{u.x_i}{\|u^2\|}$$

$$\mathrm{x}_i^{'} = \mathrm{u}^T.x_i \ni \|u\| = 1$$

(A.2)

The constraint optimization equation thus tried to maximize the variance.

$$var\{u^T.x_i\}_{i=1}^n = max\frac{1}{n}\sum_{i=1}^{n}(u^T.x_i - u^T.\bar{x})^2 \quad \ni u^T u = 1 \tag{A.3}$$

This can be solved using below equation.

$$\lambda V = SV \tag{A.4}$$

where S is called covariance matrix.

$$\underset{d\times d}{\mathrm{S}} = \underset{d\times n}{X^T} \times \underset{n\times d}{X} \tag{A.5}$$

Eigen values represent the amount of information present across different eigen vector axes.

$$\begin{bmatrix} \lambda_1 & & & \\ & \lambda_2 & & \\ & & \ddots & \\ & & & \lambda_n \end{bmatrix}$$

# Appendix C

## Naive Bayes

It works on the principle of Bayes theorem which is based on conditional probability. It assumes that features are conditionally independent. It has been found practically that even though the features are partially independent, naive bayes still works very well.

$$p(C_k \mid \mathbf{x}) = \frac{p(C_k)\, p(\mathbf{x} \mid C_k)}{p(\mathbf{x})}$$

$$\begin{aligned} p(C_k, x_1, \dots, x_n) &= p(x_1, \dots, x_n, C_k) \\ &= p(x_1 \mid x_2, \dots, x_n, C_k)\, p(x_2, \dots, x_n, C_k) \\ &= p(x_1 \mid x_2, \dots, x_n, C_k)\, p(x_2 \mid x_3, \dots, x_n, C_k)\, p(x_3, \dots, x_n, C_k) \\ &= \cdots \\ &= p(x_1 \mid x_2, \dots, x_n, C_k)\, p(x_2 \mid x_3, \dots, x_n, C_k) \cdots p(x_{n-1} \mid x_n, C_k)\, p(x_n \mid C_k)\, p(C_k) \end{aligned}$$

$$\begin{aligned} p(C_k \mid x_1, \dots, x_n) &\propto p(C_k, x_1, \dots, x_n) \\ &\propto p(C_k)\, p(x_1 \mid C_k)\, p(x_2 \mid C_k)\, p(x_3 \mid C_k)\, \cdots \\ &\propto p(C_k) \prod_{i=1}^{n} p(x_i \mid C_k)\,, \end{aligned}$$

Figure A.3: Naive Bayes conditional independence[9]

PROS:

- It works very fast and is based on principal of counting.
- It does not assume anything about the relationship between independent variable and dependent variable.
- It provides support for multi-class prediction problems.
- It is known to work exceptionally well for text and categorical variables.

CONS:

- Assumes the conditional independence between different variables in the data.

REAL WORLD EXAMPLE:

- Mark an email as spam or not.

# Appendix D

## Support Vector Machines

If we assume no error points or misclassification then this is called Hard margin SVM. d denotes the Margin width. To avoid any misclassification, this width must be maximum.

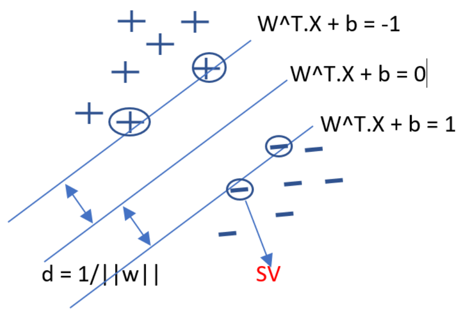


FIGURE A.4: Hard Margin SVM

Which leads us to below optimization problem.
W*, b* = argmax $2/\|W\| \quad \ni y_i(W^T.X_i + b) >= 1$
On the other hand, we have soft margin SVM where misclassification is allowed.

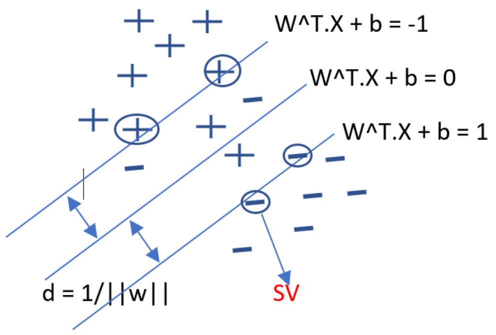


FIGURE A.5: Soft Margin SVM

Note that zita is the misclassification error, the farther it away from the dividing plane the more the misclassification error will be. Also C is the hyper parameter which we can control between overfitting and underfitting. The higher the Value of C the more the overfitting will be and vice versa. Here the first term acts like regularization and second term the total cost error(misclasssification error).
W*, b* = argmax $2/\|W\| + C * 1/n * \sum_{i=1}^{n} \xi_i \quad \ni y_i(W^T.X_i + b) >= 1 - \xi_i$
——1—— ——2——
SVM works on the principle of reducing the hinge loss which is defined as

$$f(z) = max(z_i, 0) where z_i = y_i(W^T.X_i + b) \; i.e. \; z_i >= 1. \tag{A.6}$$

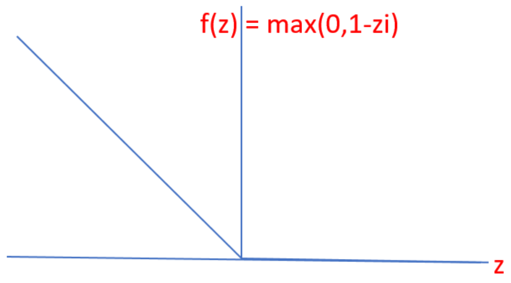


Figure A.6: hinge loss

PROS:

- Linear SVM works very well if there is linear relation between features and target.
- works well in case of high dimensionality in the data.
- it can handle nonlinear data as well with appropriate kernel.

CONS:

- Not very interpretable. Time complexity changes according to kernel.
- finding right kerel is tough.

REAL WORLD EXAMPLE:

- Face detection
- Text and hypertext categorization
- Classification of images

# Conclusions

Consumers around the globe face delays in getting their tickets resolved across businesses and their different departments. Consumer court is one such example with millions of pending cases. Most delays are caused by the wrong routing of tickets in wrong departments. To expedite the ticket routing process with maximum accuracy, we propose automated ticket router system based on AI along with flask based webAPI to receive the user input. The train data and test data was split, Necessary columns were kept and text based columns were preprocessed. Too many category features were replaced with few category based column. Null values were replaced with their most frequent values. Clustering technique was used to replace feature with large number of categories. All of these one hot encoded or TFIDFvectorization based features were combined and fed to multiple models like Logistic regression, Naïve Bayes, Random forest and LGBMClassifier. We did hyper parameter tunning for each to come up with the best possible recall, precision and F1 scores. These metrices were chosen thoughtfully to assess the models. Out of comparison amongst all the models, Logistic regression outperformed all. After getting the best features, we found that "issue" and "sub_issue" were the most prominent features and therefore we decided to consider only these two to simplify the solution. We retrained the data using logistic regression and stored the best possible weights. The webAPI solution was deployed on AWS. Our model was simple yet powerful with little or no latency.

# Future Work

While our model is very simple yet very effective, there is always chance to improve.

- We plan to acquire more data from across multiple domains and analyze the same.
- On top of this, we plan to build deep learning based multiple model architectures combining CNN and LSTM to process categorical and text data based features.
- To further add a flavor, the user data can be designed to come live stream and on the next level, a distributed computation platform like Spark can be incorporated to handle the load balancing and parallel processing of the data.
- The webapp version can be converted into mobile app based system to provide more and more flexibility to user, some of these are more of software engineering ideas, than analytics ones.

# Bibliography

[1] Jianglei Han; Aixin Sun. DeepRouting: A Deep Neural Network Approach for Ticket Routing in Expert Network.
[2] Paramesh S P and Shreedhara K S Automated IT Service Desk Systems Using Machine Learning Techniques: Proceedings of DAL 2018.
[3] V. Shimpi, M. Natu, V. Sadaphal, et al., Problem identification by mining trouble tickets
[4] R. Potharaju, N. Jain, C. Nita-Rotaru, Juggling the jigsaw: Towards automated problem inference from network trouble tickets.
[5] S. Roy, D. P. Muni, J. Y. T. Yan, N. Budhiraja, F. Ceiler, ”Clustering and labeling IT maintenance tickets”.
[6] Mati Cristian, Tolciu Tudor and Sacarea Christian in A Study in the Automation of Service Ticket Recognition using Natural Language Processing.
[7] Qihong Shao, Yi Chen, Shu Tao, Xifeng Yan, Nikos Anerousis. EasyTicket: a ticket routing recommendation engine for enterprise problem resolution
[8] Li. and Jain. Classification of Text Documents.
[9] Paramesh S.P, Ramya C, Dr. Shreedhara K.S. Classifying the Unstructured IT Service Desk Tickets Using Ensemble of Classifiers.
[10] A. Khan, B. Baharudin, L.H. Lee, Kh. Khan, A Review of Machine Learning Algorithms for Text-Documents Classification, Journal of Advances in Information Technology.