Consumer Complaint Resolution Analysis

BinaryBrains
Diptesh Saha
email: diptesh953@gmail.com
Samapan Kar
email: karsamapan0702@gmail.com

May 1, 2024

Abstract

Consumer complaints are a critical aspect of business operations, reflecting customer satisfaction levels and highlighting areas for improvement. Traditional methods of complaint resolution often suffer from inefficiencies, resulting in delayed responses and heightened consumer frustration. To address this issue, we propose a machine learning-driven approach for Consumer Complaint Resolution Analysis. Our project focuses on automating the process of complaint classification and prioritization, leveraging natural language processing (NLP) techniques to extract meaningful insights from complaint texts. By collecting and preprocessing complaint data from various sources, including consumer forums and social media platforms, we aim to develop robust models capable of categorizing complaints based on product/service type and issue severity. Through feature engineering and model training, we intend to deploy a scalable system capable of real-time complaint processing and resolution prioritization. This project seeks to enhance the efficiency and effectiveness of complaint resolution processes, ultimately improving customer satisfaction and fostering stronger consumer-company relationships.

1 Introduction

1.1 Description

Our project aims to utilize machine learning techniques to analyze consumer complaints and enhance the resolution process for both consumers and companies.

1.2 What?

This project leverages Python libraries like Pandas for data manipulation, Seaborn, and Matplotlib for visualization, and Sklearn for model building. We aim to discern customer disputes using machine learning algorithms. The best model will predict disputes in the test dataset, with results saved for integration into business processes. Our focus on accuracy enables proactive issue resolution, enhancing customer satisfaction and loyalty. By employing Python tools and machine learning, we offer a scalable solution to this vital business challenge.

1.3 Why?

Product review is the most basic function/factor in resolving customer issues and increasing the sales growth of any product. We can understand their mindset toward our service without asking each customer.

When consumers are unhappy with some aspect of a business, they reach out to customer service and might raise a complaint. Companies try their best to resolve the complaints that they receive. However, it might not always be possible to appease every customer.

So Here, we will analyze data, and with the help of different algorithms, we are finding the best classification of customer category so that we can predict our test data.

1.4 How?

1.4.1 Data Preprocessing:

- Data Reading and Type Check: Read from Excel and check data types for train and test sets.
- Handling Missing Values: Analyze missing data, dropping columns with >25% missing.
- Date Extraction and Calculation: Extract day, month, year, and calculate days held.
- Storing Disputed Data: Store disputed data for future use.
- Data Visualization: Plot various dispute-related bar graphs.
- Data Cleaning: Convert negative days held, drop unnecessary columns, impute "State" nulls, create 'Week_Received', change dispute labels, create dummy variables, and scale data.

1.4.2 Feature Engineering:

- Text Pre-processing: Tokenize the Issues of the customers using Natural Language Processing for classification purpose.
- Feature Selection: Scaling the datasets and make feature selection using Principal Component Analysis up to 80% of the information.

1.4.3 Model Building & Evaluation:

- Splitting & Modeling: Split data, build models (e.g., Logistic Regression, Decision Tree, etc.).
- Model Evaluation: Assess model accuracy, select best-performing for test file prediction.

2 Literature review

- Pramod Kumar Naik *et al.* [4] utilized Multinomial Naive Bayes model with 79.82% accuracy, Decision Tree with 71.51% accuracy, Linear SVM with 83.62% accuracy, Logistic Regression with 83.62% accuracy and K-Nearest Neighbours with 74.82% accuracy.

  They are planning to develop an automatic financial complaint classification system that automatically deals with the customer complaints by segregating the data & routing it to the right department. They are planning to develop the system by using Natural Language Processing (NLP), Artificial Intelligence (AI), Machine Learning (ML) & Deep Learning (DL) concepts and implement using Python, Jupyter Notebook,.etc. The end product will be a webbased application system where customer can register their complaints without having to worry about sending it to right department.

  Product review is the most basic function/factor in resolving customer issues and increasing the sales growth of any product. We can understand their mindset toward our service without asking each customer. When consumers are unhappy with some aspect of a business, they reach out to customer service and might raise a complaint. Companies try their best to resolve the complaints that they receive. However, it might not always be possible to appease every customer. So Here, we

will analyze data, and with the help of different algorithms, we are finding the best classification of customer category so that we can predict our test data. Use Python libraries such as Pandas for data operations, Seaborn and Matplotlib for data visualization and EDA tasks, Sklearn for model building and performance visualization, and based on the best model, make a prediction for the test file and save the output. The main objective is to predict whether our customer is disputed or not with the help of given data.

3 Proposed methodology

The project involves gathering complaint data from diverse sources, cleaning it, and handling missing values. Relevant features will be extracted, and categorical variables will be transformed into numerical representations for analysis. Various algorithms will be explored, and models will be trained and optimized for performance using techniques such as hyperparameter tuning. Model performance will be assessed using metrics like accuracy, precision, and recall. Cross-validation will be employed, and the data will be split into training, validation, and test sets to ensure robustness. Ensemble techniques will be utilized to further improve model performance, with hyperparameters optimized accordingly. Finally, the trained models will be deployed and integrated into existing systems.

After the Data Pre-processing the Text Pre-Processing has done using **Natural Language Processing**. Natural Language Processing (NLP) is a branch of artificial intelligence concerned with the interaction between computers and humans through natural language. It enables computers to understand, interpret, and generate human language in a way that is both meaningful and contextually relevant.One fundamental process within NLP is **Tokenization**, which involves breaking down a text into smaller units, typically words or sentences. This step forms the foundational units upon which further analysis can be conducted. Imagine a book being dissected into individual words or sentences, each serving as a distinct entity for analysis.**Lemmatization** and **Stemming** are two techniques used to reduce words to their base or root form, aiding in the normalization of text. Lemmatization maps words to their dictionary form (lemma), ensuring consistency in semantic meaning. For instance, "running" and "ran" would be lemmatized to "run". Stemming, on the other hand, involves removing prefixes or suffixes to obtain the root form of a word. While stemming might be more aggressive than lemmatization, it is useful in scenarios where computational efficiency is crucial and a less precise approach suffices.In essence, tokenization, lemmatization, and stemming are essential methods in NLP, enabling computers to process and understand human language by breaking it down into manageable units, normalizing variations, and facilitating analysis and interpretation.

After implementing NLP the **TF-IDF** values of the consumer complaints are calculated. **Term Frequency-Inverse Document Frequency (TF-IDF)** is a numerical statistic used in natural language processing and information retrieval to evaluate the importance of a term in a document relative to a collection of documents, often a corpus. TF-IDF is calculated by multiplying two values: Term Frequency (TF) and Inverse Document Frequency (IDF). Term Frequency (TF) measures how frequently a term appears in a document. It is calculated by dividing the number of times a term occurs in a document by the total number of terms in that document. This normalization helps to account for the varying lengths of documents and prevents bias towards longer documents. Inverse Document Frequency (IDF) quantifies the importance of a term across a collection of documents. It is calculated by dividing the total number of documents in the corpus by the number of documents containing the term, and then taking the logarithm of this ratio. This logarithmic scaling ensures that highly frequent terms are penalized, while rare terms are given more weight. The final TF-IDF score for a term in a document is obtained by multiplying its Term Frequency (TF) by its Inverse Document Frequency (IDF). This calculation results in a numerical representation of the importance of a term within a specific document relative to the entire corpus, with higher scores indicating greater significance. TF-IDF is widely used in various NLP tasks such as text classification, information retrieval, and document clustering, aiding in the identification of relevant and meaningful terms within a document. The complaints are now replaced by Vectorized complaints.

Now the scaling of the modified dataset has done. Then the feature selection has done using **Principal**

Component Analysis. Principal Component Analysis (PCA) is a widely used technique in data analysis and dimensionality reduction. It is particularly useful for simplifying complex datasets while preserving the most important information. PCA works by transforming the original features of a dataset into a new set of orthogonal components, known as principal components. These components are ordered by the amount of variance they explain in the data, with the first principal component capturing the maximum variance and each subsequent component capturing as much of the remaining variance as possible. The transformation is achieved through eigendecomposition or singular value decomposition (SVD) of the covariance matrix of the dataset. By retaining only the top principal components that explain the majority of the variance, PCA allows for the reduction of the dimensionality of the data while minimizing information loss. This is particularly beneficial for datasets with a large number of features, as it can help in visualizing and interpreting the data more effectively, as well as in speeding up subsequent machine learning algorithms. PCA finds applications in various fields such as image processing, signal processing, genetics, and finance, where it is used for tasks such as feature extraction, noise reduction, and visualization of high-dimensional data. Its ability to uncover the underlying structure of complex datasets makes it a valuable tool in exploratory data analysis and pattern recognition. So the features which are covering top 80% of the information have been selected and based on those the datasets are splitted by the dependent and independent variables.

Next part comes with Model Building. Some machine learning models are implemented to measure their accuracy values:

- **Logistic Regression:** Logistic regression is a linear model for binary classification, estimating the probability of an instance belonging to a class via the logistic function. It learns parameters, typically through maximum likelihood estimation or gradient-based optimization, minimizing a loss function like cross-entropy. Regularization techniques such as L1 or L2 regularization can mitigate overfitting. It's interpretable, computationally efficient, and applicable to problems with linearly separable classes or when feature-target relationships are relatively simple. Despite its simplicity, logistic regression is a powerful tool, often used as a baseline model for more complex classification tasks, facilitating quick prototyping and interpretation of results.

- **Decision Tree Classifier:** The Decision Tree Classifier is a non-parametric supervised learning algorithm used for classification tasks. It recursively splits the dataset into subsets based on the most discriminative features, optimizing impurity measures like Gini impurity or entropy at each step. This hierarchical structure forms a tree-like model where internal nodes represent features, branches denote decision rules, and leaf nodes represent class labels. Decision trees are interpretable, robust to outliers, and capable of handling non-linear relationships. However, they are prone to overfitting, which can be mitigated by techniques like pruning or ensemble methods such as Random Forests.

- **Random Forest Classifier:** The Random Forest Classifier is an ensemble learning method based on decision trees. It constructs multiple decision trees by bootstrapping the dataset and selecting a random subset of features for each tree. During training, each tree independently votes for the class label, and the final prediction is determined by aggregating the votes. Random forests mitigate overfitting and improve generalization by averaging predictions across multiple trees. They excel in handling high-dimensional data, capturing non-linear relationships, and are robust to noisy or missing features. However, they may be computationally expensive and less interpretable compared to individual decision trees.

- **AdaBoost Classifier:** The AdaBoost Classifier is an ensemble learning technique that builds a strong classifier by combining multiple weak classifiers. It iteratively trains a sequence of weak learners on weighted versions of the dataset, focusing on instances that were previously misclassified. Each weak learner contributes to the final model based on its performance, with more weight given to classifiers that classify difficult instances correctly. The final prediction is determined by a weighted sum of individual learner predictions. AdaBoost is robust, adaptive, and effective in handling complex datasets. However, it may be sensitive to noisy data and outliers, requiring careful parameter tuning for optimal performance.

- **Gradient Boosting Classifier:** The Gradient Boosting Classifier is an ensemble learning method that builds a predictive model by sequentially adding weak learners, usually decision trees, to minimize a differentiable loss function. Each new learner is trained on the residuals of the previous predictions, focusing on the errors made by the existing model. By iteratively improving upon the shortcomings of the preceding models, Gradient Boosting creates a strong ensemble model. It provides high predictive accuracy, handles complex interactions between features, and is robust to overfitting when properly tuned. However, it can be computationally intensive and sensitive to noisy data or outliers, necessitating careful regularization.

- **KNeighbours Classifer:** The K-Nearest Neighbors (KNN) Classifier is a non-parametric supervised learning algorithm used for classification tasks. It classifies instances based on the majority class of their K nearest neighbors in the feature space, determined by a distance metric such as Euclidean or Manhattan distance. During training, the algorithm stores the entire dataset, making it memory-intensive but computationally inexpensive during prediction. KNN's performance heavily relies on the choice of K and the distance metric. It's robust to noisy data, handles multi-class classification, and can capture complex decision boundaries. However, it may suffer from the curse of dimensionality and requires careful preprocessing of the data.

- **XGB Classifier:** The XGBoost (Extreme Gradient Boosting) Classifier is an advanced implementation of gradient boosting algorithm known for its speed and performance. It builds an ensemble of decision trees sequentially, optimizing a differentiable loss function by minimizing gradients. XGBoost introduces regularization terms to prevent overfitting and incorporates advanced techniques like tree pruning and parallel processing to enhance efficiency. It supports various objective functions and evaluation metrics, making it highly customizable for different tasks. XGBoost is widely used in Kaggle competitions and real-world applications due to its superior predictive accuracy, scalability, and robustness against overfitting. However, it requires careful parameter tuning for optimal performance.

After implementing all the models, according to their accuracy values the best models are taken and using Hyperparameter tuning the optimal values of the model parameters are chosen. Hyperparameter tuning involves systematically searching for the best combination of hyperparameters to maximize the model's performance on a validation dataset. By adjusting hyperparameters, such as learning rate, regularization strength, or tree depth, practitioners can fine-tune the model to achieve better generalization, improve accuracy, or reduce overfitting.

4 Experimental result

The dataset used for this project can be accessed through the given link:
Consumer Complaint Resolution Dataset

- **Dataset Description: Dispute:** This is our target variable based on train data; we have two groups, one with a dispute with the bank and another don't have any issue with the bank. **Date received:** The day complaint was received. **Product:** different products offered by the bank (credit cards, debit cards, different types of transaction methods, accounts, locker services, and money-related). **Sub-product:** loan, insurance, other mortgage options. **Issue:** Complaint of customers. **Company public response:** Company's response to consumer complaint. **Company:** Company name. **State:** State where the customer lives (different state of USA). **ZIP code:** Where the customer lives. **Submitted via:** Register complaints via different platforms (online web, phone, referral, fax, post mail). **Date sent to company:** The day complaint was registered. **Timely response?:** Yes/no. **Consumer disputed?:** yes/no (target variable). **Complaint ID:** unique to each consumer.

- **Experimental settings:** After Data pre-processing such as missing value analysis, deleting unrequired columns, Text pre-processing using NLP and feature Selection using PCA the final dataset

comes with 201 columns in the training dataset. In the Consumer Disputed column the 'No' values are changed to 0 and 'Yes' values are changed to 1. Also the dataset is splitted into training and test dataset by 75% and 25% respectively.

- **Results:** After implementing all the machine learning models and collecting the accuracy values the results are coming as:

Model	Accuracy
Logistic Regression	0.7879223
Decision Tree Classifier	0.7750019
Random Forest Classifier	0.774511
AdaBoost Classifier	0.7879112
Gradient Boosting Classifier	0.787905
KNN Classifier	0.755515
XGB Classifier	0.787130

Here the Logistic Regression model and AdaBoost Classifier models are giving the highest accuracy values among all the models. Also after the hyperparameter tuning of the models there are no significant changes in the accuracy values of the models.

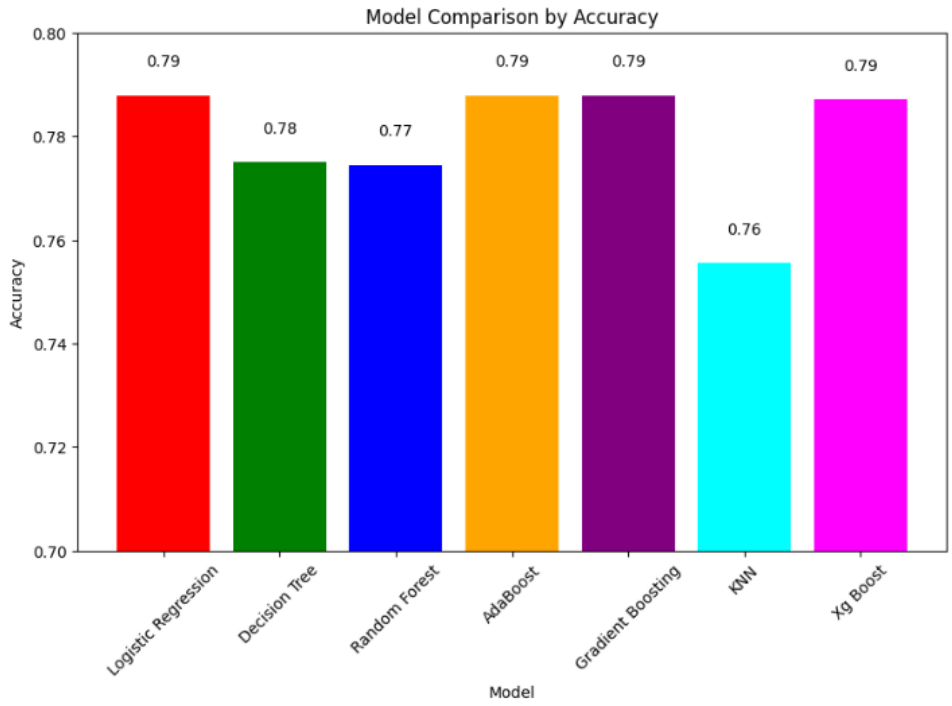

Figure 1: Barplot of Accuracy values

5 Summary

Product reviews play a crucial role in understanding customer satisfaction and can significantly impact the sales and reputation of any business. Often, customers express their discontent through complaints, which provides valuable insights into the areas needing improvement.

The primary goal of this project is to utilize advanced data analytics techniques to classify customer complaints and predict customer disputes effectively. By leveraging Python libraries such as Pandas for data handling, Seaborn and Matplotlib for data visualization and exploratory data analysis (EDA), and Scikit-learn for machine learning model development, the project aims to identify whether a customer will dispute a service or not.

We used various kinds of classification models while doing this project. They were Logistic Regression, Decision Tree Classifier, Random Forest Classifier, AdaBoost Classifier, Gradient Boosting Classifier, XGBoost Classifier etc. Among them we got the best accuracy from **Logistic Regression** and **Ada Boost Classifier** with accuracy of 79%.

We faced many challenges while doing this project.

PCA was crucial for managing high-dimensional data and reducing computational load without significantly sacrificing information.

The use of diverse models allowed for a comprehensive evaluation across different algorithmic approaches, highlighting their unique strengths and weaknesses in handling classification tasks.

The project underscored the importance of examining Type I and Type II errors, providing deeper insights into model performance beyond mere accuracy.

- **Conclusion:** This project effectively demonstrated the application of various machine learning techniques from preprocessing through to model evaluation. It highlighted the importance of using a systematic approach to model comparison and provided valuable insights into the strengths and weaknesses of different classifiers in a practical, real-world scenario.

6 References

Some references related to the project are given below:

References

[1] Nitin Indurkhya and Fred J Damerau. *Handbook of natural language processing.* Chapman and Hall/CRC, 2nd edition, 2010.

[2] Daniel Jurafsky and James H. Martin. *Speech and Language Processing.* Prentice Hall PTR, 2nd edition, 2008.

[3] Mehryar Mohri, Afshin Rostamizadeh, and Ameet Talwalkar. *Foundations of Machine Learning.* The MIT Press, 2nd edition, 2018.

[4] Pramod Kumar Naik, Sandesh Balan, et al. Consumer complaints classification using machine learning & deep learning. *International Research Journal on Advanced Science Hub*, 5(05S):116–122, 2023.