iJRASET
International Journal For Research in
Applied Science and Engineering Technology

INTERNATIONAL JOURNAL
FOR RESEARCH
IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 13 Issue: V Month of publication: May 2025

DOI: https://doi.org/10.22214/ijraset.2025.71573

Complaint Classification Model Using NLP

Dipika Verma[1], Manikandan Thevar[2], Prof. Rani Mario[3]
[1, 2]*Master of Science (Information Technology),*[3]*Guide, Department of Information Technology N.R.Swamy College of Commerce & Economics & Smt.Thirumalai College of Science Mumbai, Maharashtra*

Abstract: This report discusses the research done on the chosen topic, which is ComplaintClassification Model using NLP. Artificial intelligence nowadays is playing a vital role in our society. It is just minimizing human labor and effort in every field. Industrial sector is feeding their large amount of structured and unstructured data to find out useful information for scientific research. The main alarming thing is how to operate the huge feedback data, which is in the form of complaints i.e., in text format. Here, we have proposed a model which automatically classifies the complaints by analyzing the text with the help of machine learning and NLP (Natural Language Processing) methods. We have initially collected a dataset from a portal containing complaints of citizens. For validation, we have also used another dataset of complaints from the Consumer Complaint Database. After tokenizing,stemming and lemmatization, different feature extraction techniques like count vectorizer and TF-IDF are used to convert all the textual data into numerical data. Then different machinelearning algorithms are used to classify the complaints into their categories. In ourgathered dataset, 10 different divisions for complaints are used and an accuracy of more than 70% is achieved with all classifiers. Similarly on the Consumer Complaint dataset, 86% accuracy has been achieved. The proposed model is helpful in saving a lot of time, as there is no need to go through each complaint and categorizing manually.

I. INTRODUCTION

A. Background of Study

Nowadays, Artificial intelligence (AI) systems are very common and reliable, being increasingly used in different organizations. Different kinds of AI techniques are being employed in different industrial sectors to assist humans and increase work efficiency. One of the AI techniques being used is Natural Language Processing (NLP). NLP handles text that is available in a soft form such as on websites or any available local network where textual data in large amounts can be provided by humans to the system.

To give reliability to citizens, multiple government organizations are providing public services virtually. Those organizations are also answerable to the public. The citizen feedback is developed to make the system more reliable and fix theproblem, oftenthese feedbacks (complaints or requests) come to some online portal, in an online form, or via emails. This became a huge and time taking problem to dealwith if the support staff handling the feedback forms is limited. The feedback is in massive size as it includes feedback from different citizens belonging to different regions. As the feedback is in the written format like email so NLP techniques are used to classify different complaints based on content.

Consumer complaints are a critical source of feedback for businesses, but manually categorizing and analyzing these complaints can be time-consuming and prone to errors. By leveraging natural language processing (NLP) and machine learning algorithms, we can automate the classification process, ensuring that complaints are handled promptly and effectively.

In this research we have taken the banking consumer complaint public data to analyse, where multiple complaints are registered for multiple financial companies under multiple categories like – credit card, loan, mortgage, etc. But the use case which we have used in this study can be used in multiple domains (like Life Science, Pharmaceutical, Telecom, Automobile, Manufacturing, etc) and not just in financial domain.

B. Problem Statement

The problem statement of this project is:

Developing an Automated System for Classifying Consumer Complaints to Enhance Customer Service and Operational Efficiency

C. Objective

The objectiveof this project is:

To create a machine learning-based system that automatically categorizes consumer complaints into predefined categories, enabling quicker and more accurate routing of complaints to the appropriate departments. This system aims to improve customer service response times, enhance operational efficiency, and provide valuable insights into common issues faced by consumers.

The primary objective of this case study is to develop a highly accurate and efficient model that can classify customer complaints based on the products and services mentioned in the tickets. By categorizing complaints into distinct clusters, financial companies can promptly address issues, improve their service offerings, and optimize the customer support ticket system.

Develop a model that utilizes non-negative matrix factorization (NMF), a topic modeling technique, to classify customer complaints into the following five clusters based on their products/services:

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

II. LITERATURE REVIEW

1) Introduction to NLP and Machine Learning in Text Classification

- Natural Language Processing (NLP) is a field at the intersection of computer science and linguistics, focusing on the interaction between computers and human language. NLP techniques are used to process and analyze large amounts of natural language data.
- Machine Learning (ML) involves training algorithms to learn from data and make predictions or decisions without being explicitly programmed. In text classification, ML models are trained to categorize text into predefined categories.

2) Classical vs. Deep Learning Models for Complaint Classification

- Classical Methods: Traditional approaches like TF-IDF (Term Frequency-Inverse Document Frequency) and SVM (Support Vector Machine) have been widely used for text classification. These methods are effective but may struggle with capturing the context and nuances of complaints
- Deep Learning Models: Recent advancements in deep learning, such as LSTM (Long Short-Term Memory), Bi-LSTM (Bidirectional LSTM), GRU (Gated Recurrent Unit), and CNN (Convolutional Neural Networks), have shown significant improvements in text classification tasks. These models can capture complex patterns and dependencies in text data.

3) Word Embedding Techniques

Word Embeddings: Techniques like Word2Vec, FastText, BERT (Bidirectional Encoder Representations from Transformers), and DistilBERT are used to represent words as dense vectors. These embeddings capture semantic relationships between words, improving the performance of NLP models

4) Application of Large Language Models (LLMs)

Large Language Models (LLMs): Models like GPT-4 and other reasoning models have demonstrated remarkable capabilities in various NLP tasks, including zero-shot classification. These models can classify consumer complaints without prior exposure to labeled training data, making them valuable for handling emerging issues and dynamic complaint categories

5) Sentiment Analysis and Topic Modeling

- Sentiment Analysis:Analyzing the sentiment of consumer complaints can provide insights into customer satisfaction and identify areas for improvement.
- Topic Modeling: Techniques like Latent Dirichlet Allocation (LDA) can uncover hidden topics within consumer complaints, helping businesses understand common issues and trends.

III. SURVEY OF TECHNOLOGIES

A. Existing System

1) Classical Machine Learning Models

a) Support Vector Machine (SVM):

- Description: SVM is a supervised learning model that finds the optimal hyperplane to separate different classes in the feature space.
- Application: Used for text classification tasks, including consumer complaint classification.
- Advantages: Effective in high-dimensional spaces and works well with TF-IDF features.
- Limitations: May struggle with large datasets and overlapping classes.

b) Random Forest:

- Description: An ensemble learning method that builds multiple decision trees and merges them to improve accuracy and stability.
- Application: Used for categorizing consumer complaints by learning patterns in the text data.
- Advantages: Robust to overfitting and provides feature importance.
- Limitations: Can be computationally intensive and may require tuning.

c) K-Nearest Neighbors (KNN):

- Description: A simple, instance-based learning algorithm that classifies data based on the closest training examples.
- Application: Used for quick and straightforward classification tasks.
- Advantages: Intuitive and easy to implement.
- Limitations: Performance can degrade with high-dimensional data and large datasets.

2) Deep Learning Models

a) Long Short-Term Memory (LSTM):

- Description: A type of recurrent neural network (RNN) that can capture long-term dependencies in sequential data.
- Application: Used for text classification tasks, including consumer complaints.
- Advantages: Effective in capturing context and sequential patterns.
- Limitations: Requires large amounts of data and computational resources.

b) Bidirectional LSTM (Bi-LSTM):

- Description: An extension of LSTM that processes data in both forward and backward directions.
- Application: Used for improved text classification by capturing context from both directions.
- Advantages: Better performance in capturing context compared to standard LSTM.
- Limitations: More computationally intensive.

c) Gated Recurrent Unit (GRU):

- Description: A type of RNN similar to LSTM but with fewer parameters.
- Application: Used for text classification tasks.
- Advantages: Faster training and less computationally intensive than LSTM.
- Limitations: May not capture long-term dependencies as effectively as LSTM.

d) Convolutional Neural Networks (CNN):

- Description: A deep learning model typically used for image processing but also effective for text classification.
- Application: Used for extracting features from text data.
- Advantages: Effective in capturing local patterns in text.
- Limitations: May require large amounts of data and computational resources.

3) Word Embedding Techniques

a) Word2Vec:

- Description: A technique to represent words as dense vectors based on their context.
- Application: Used to improve text classification by capturing semantic relationships between words.
- Advantages: Efficient and effective in capturing word meanings.
- Limitations: Requires large corpora for training.

b) FastText:

- Description: An extension of Word2Vec that considers subword information.
- Application: Used for improved text classification by capturing morphological features.
- Advantages: Better performance with rare words and morphologically rich languages.
- Limitations: Requires large corpora for training.

c) BERT (Bidirectional Encoder Representations from Transformers):

- Description: A transformer-based model that captures context from both directions.
- Application: Used for various NLP tasks, including text classification.
- Advantages: State-of-the-art performance in many NLP tasks.
- Limitations: Computationally intensive and requires fine-tuning.

d) DistilBERT:

- Description: A smaller, faster, and cheaper version of BERT.
- Application: Used for efficient text classification.
- Advantages: Retains much of BERT's performance while being more efficient.
- Limitations: Slightly lower performance compared to BERT.

4) Large Language Models (LLMs)

a) GPT-4:

- Description: A large language model capable of various NLP tasks, including zero-shot classification.
- Application: Used for classifying consumer complaints without prior exposure to labeled training data.
- Advantages: High accuracy and flexibility in handling dynamic complaint categories.
- Limitations: Requires significant computational resources.

b) Reasoning Models:

- Description: Models trained with reinforcement learning to exhibit advanced inferential capabilities.
- Application: Used for complex reasoning and structured decision-making in text classification.
- Advantages: Groundbreaking advancements in text classification.
- Limitations: Requires specialized training and computational resources.

5) Sentiment Analysis and Topic Modeling

a) Sentiment Analysis:

- Description: Analyzing the sentiment of text data to understand customer satisfaction.
- Application: Used to gauge the sentiment of consumer complaints.
- Advantages: Provides insights into customer emotions and satisfaction.
- Limitations: May require fine-tuning for specific domains.

b) Latent Dirichlet Allocation (LDA):

- Description: A topic modeling technique to uncover hidden topics within text data.
- Application: Used to identify common issues and trends in consumer complaints.
- Advantages: Helps in understanding the main themes in complaints.
- Limitations: Requires careful parameter tuning.

B. Proposed System

The proposed model aims to develop an automated system for classifying consumer complaints into predefined categories using NLP and machine learning techniques. This system will enhance customer service response times, improve operational efficiency, and provide valuable insights into common issues faced by consumers.

1) Components of the Proposed Model:

a) Data Preprocessing:

- Text Standardization: Convert all text to lowercase to ensure uniformity.
- Stop Words Removal: Remove common words that do not contribute much to the meaning of the text.

- Tokenization: Split text into individual words or tokens.
- Lemmatization: Reduce words to their base or root form.

b) Feature Extraction:

- TF-IDF Vectorization: Convert text data into numerical vectors using Term Frequency-Inverse Document Frequency (TF-IDF). This helps in capturing the importance of words in the context of the entire dataset.

c) Sentiment Analysis:

- TextBlob: Analyze the sentiment of consumer complaints to understand customer emotions and satisfaction levels.

d) Topic Modeling:

- Latent Dirichlet Allocation (LDA): Discover hidden topics within consumer complaints to identify common issues and trends.

e) Classification Algorithms:

- Support Vector Machine (SVM): Classify complaints by finding the optimal hyperplane that separates different categories.
- Random Forest: Build multiple decision trees and merge them to improve accuracy and stability in classification.

f) Model Evaluation:

- Metrics: Use accuracy, precision, recall, and F1-score to evaluate the performance of the classification models.
- Visualization: Display classification reports and confusion matrices to understand model performance.

2) Workflow:

a) Data Collection:

- Collect consumer complaints data from various sources (e.g., online forms, customer service records).

b) Data Preprocessing:

- Apply text standardization, stop words removal, tokenization, and lemmatization.

c) Feature Extraction:

- Convert text data into TF-IDF vectors.

d) Sentiment Analysis:

- Analyze the sentiment of complaints using TextBlob.

e) Topic Modeling:

- Apply LDA to discover hidden topics within the complaints.

f) Classification:

- Train SVM and Random Forest models on the preprocessed data.
- Evaluate the models using appropriate metrics.

g) Deployment:

- Deploy the best-performing model to classify new consumer complaints automatically through a dashboard.

IV. REQUIREMENT SPECIFICATION

A. Hardware and Software Requirements

1) Hardware Requirements:

The most common set of requirements defined by any operating system or software application is the physical computer resource called as hardware. A hardware requirement list is often accompanied by a hardware compatibility list (HCL). An HCL especially in case of operating system.

Hardware requirements of these project:-
RAM: - 2 GB.
HARD DISK: - 16 GB of 32 bit.
PROCESSOR: - 1 GB.

2) Software Requirements:

Software requirements deal with defining software resource requirements and pre-requisites that need to be installed on a computer to provide optimal functioning of a application. These software requirements needs some packages that need to be installed with it.

Software requirements of these project :-

OPERATING SYSTEM : - Windows
TOOL :- VS Code / Jupyter Notebook, TIBCO Spotfire / Power BI
LANGUAGE : - Python

B. Justice of Selection of Tool and Technology

1) Python in Jupyter Notebook:

a) Interactive Computing

- Immediate Feedback: Jupyter Notebook allows you to write and execute code in an interactive environment, providing immediate feedback. This is particularly useful for experimenting with code and seeing results in real-time.
- Cell-Based Execution: Code is written in cells, which can be executed independently. This makes it easy to test and debug small sections of code without running the entire script.

b) Rich Media Integration

- Visualizations: Jupyter Notebook supports rich media output, including images, videos, and interactive plots. Libraries like Matplotlib, Seaborn, and Plotly can be used to create visualizations directly within the notebook.
- Markdown Support: You can use Markdown to add formatted text, equations (using LaTeX), and links. This is useful for documenting your code and explaining your analysis.

c) Versatility and Flexibility

- Multiple Languages: While Python is the most commonly used language, Jupyter Notebook supports over 40 programming languages, including R, Julia, and Scala.
- Integration with Other Tools: Jupyter Notebooks can be integrated with other tools and platforms, such as GitHub for version control, and cloud services like Google Colab for enhanced computational resources.

d) Collaboration and Sharing

- Easy Sharing: Notebooks can be easily shared with others via email, GitHub, or JupyterHub. This makes collaboration straightforward, as others can view and run your code.
- Reproducibility: By sharing notebooks, you ensure that others can reproduce your results, which is crucial for scientific research and collaborative projects.

e) Educational Use

- Learning and Teaching: Jupyter Notebooks are widely used in education for teaching programming, data science, and machine learning. The interactive nature of notebooks makes them an excellent tool for learning and teaching.
- Assignments and Projects: Educators can create and distribute assignments in Jupyter Notebooks, allowing students to write and execute code, visualize data, and document their findings in one place.

f) Integration with Data Science Libraries

- Seamless Integration: Python's extensive ecosystem of data science libraries (e.g., Pandas, NumPy, Scikit-learn) can be seamlessly integrated into Jupyter Notebooks, making it a powerful tool for data analysis and machine learning.

g) Open Source and Community Support

- Open Source: Jupyter Notebook is an open-source project, which means it is free to use and has a large community of contributors.
- Community Support: The large and active community provides extensive documentation, tutorials, and support, making it easier to learn and troubleshoot issues.

h) Version Control and Experiment Tracking

- Version Control: Notebooks can be version-controlled using Git, allowing you to track changes and collaborate with others.
- Experiment Tracking: You can keep track of different experiments and their results within the same notebook, making it easier to compare and analyze different approaches.

2) Libraries Used :

a) Pandas

- Purpose: Data manipulation and analysis.
- Usage: Loading and preprocessing the dataset, handling dataframes, and exporting results to CSV files.

b) Plotly Express

- Purpose: Data visualization.
- Usage: Creating interactive plots and visualizations.

c) NumPy

- Purpose: Numerical computing.
- Usage: Handling arrays and performing mathematical operations.

d) Seaborn

- Purpose: Statistical data visualization.
- Usage: Creating informative and attractive statistical graphics.

e) Matplotlib

- Purpose: Plotting and visualization.
- Usage: Displaying plots and graphs.

f) Missingno

- Purpose: Missing data visualization.
- Usage: Visualizing missing values in the dataset.

g) NLTK (Natural Language Toolkit)

- Purpose: Text processing and NLP.
- Usage: Tokenization, stop words removal, and lemmatization.

h) Scikit-learn

- Purpose: Machine learning.
- Usage: TF-IDF vectorization, classification algorithms (SVM, Random Forest, KNN), model evaluation, and topic modeling (LDA).

i) TextBlob

- Purpose: Text processing and sentiment analysis.
- Usage: Analyzing the sentiment of consumer complaints.

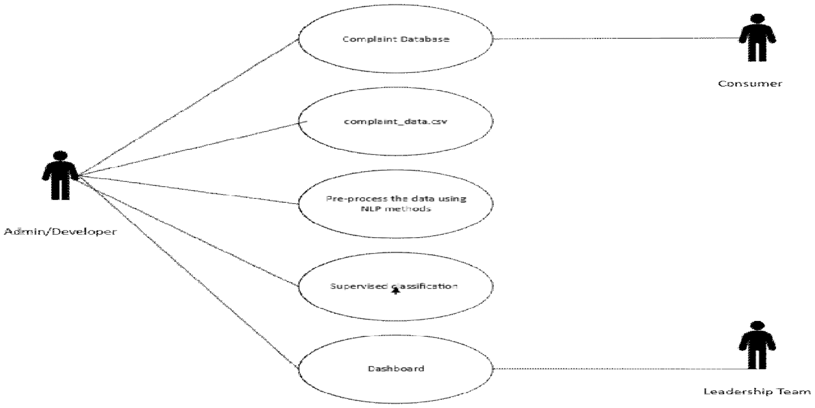
j) Spacy

- Purpose: Advanced NLP.
- Usage: Named Entity Recognition (NER) (though removed in the final model due to compatibility issues).

V. SYSTEM DESIGN

A. Use Case Diagram

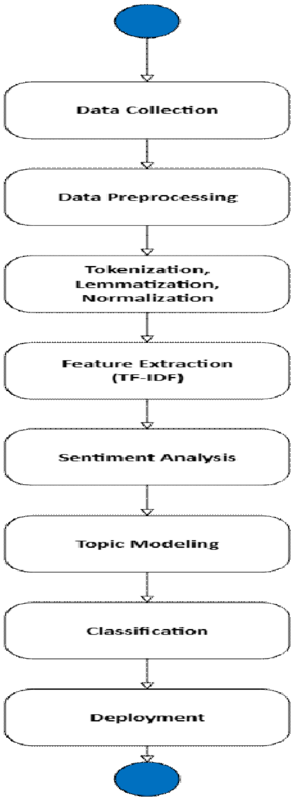
A Use case is a description of set of sequence of actions Graphically it is rendered as an ellipse with solid line including only its name. Use case diagram is a behavioural diagram that shows a set of use cases and actors and their relationship. It is an association between the use cases and actors. An actor represents a real-world object. Use case diagrams are used to gather the requirements of a system. These requirements are mostly design requirements. So when a system is analysed to gather its functionalities use cases are prepared and actors are identified. Now when the initial task is complete use case diagrams are modelled to present the outside view



Use Case Diagram

B. Activity Diagram

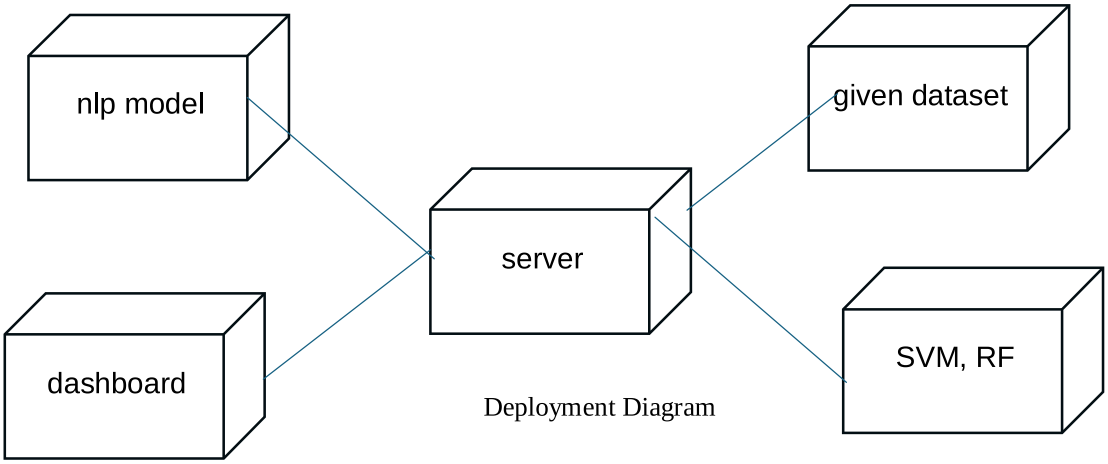
Activity diagrams are graphical representations of Workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system. An activity diagram shows the overall flow of control



Activity Diagram

C. Deployment Diagram

The deployment diagram in the Unified Modelling Language models the physical deployment of artifacts on nodes. To describe a website, for example, a deployment diagram would show what hardware components(“nodes”) exists (e.g., a web server, an application server, and a database server), what software components(“artifacts”) run on each node (e.g., web application, database), and how the different pieces are connected. The nodes appear as boxes, and the artifacts allocated to each node appear as rectangles within boxes. Nodes may have sub nodes, which appear as nested boxes. A single node in a deployment diagram may conceptually represent multiple physical nodes, such as cluster of database server.



Deployment Diagram

D. Detailed information about the framework

1) Components of the framework

a) Data Collection

- Source: Consumer complaints data collected from various sources such as online forms, customer service records, and public datasets (e.g., CFPB dataset
- Format: Typically in CSV or JSON format containing complaint narratives and metadata.

b) Data Preprocessing

- Text Standardization: Convert all text to lowercase to ensure uniformity.
- Stop Words Removal: Remove common words that do not contribute much to the meaning of the text.
- Tokenization: Split text into individual words or tokens using NLTK
- Lemmatization: Reduce words to their base or root form using NLTK's WordNetLemmatizer

c) Feature Extraction

- TF-IDF Vectorization: Convert text data into numerical vectors using Term Frequency-Inverse Document Frequency (TF-IDF) with Scikit-learn. This helps in capturing the importance of words in the context of the entire dataset.

d) Sentiment Analysis

- TextBlob: Analyze the sentiment of consumer complaints to understand customer emotions and satisfaction levels

e) Topic Modeling

- Latent Dirichlet Allocation (LDA): Discover hidden topics within consumer complaints to identify common issues and trends using Scikit-lear

f) Classification Algorithms

- Support Vector Machine (SVM): Classify complaints by finding the optimal hyperplane that separates different categories using Scikit-learn
- Random Forest: Build multiple decision trees and merge them to improve accuracy and stability in classification using Scikit-learn

g) Model Evaluation

- Metrics: Use accuracy, precision, recall, and F1-score to evaluate the performance of the classification models.
- Visualization: Display classification reports and confusion matrices to understand
- model performance.

h) Deployment

- Automated System: Deploy the best-performing model to classify new consumer complaints automatically.
- Integration: Integrate the system with existing customer service platforms to streamline complaint handling.

i) Benefits

- Automated Classification: Reduces manual effort and speeds up the complaint handling process.
- Improved Accuracy: Ensures complaints are correctly categorized, leading to better customer service.
- Insights and Analysis: Provides valuable insights into common issues and trends in consumer complaints.
- Enhanced Customer Service: Improves response times and customer satisfaction by efficiently routing complaints to the appropriate departments.

2) Machine Learning :

Machine learning is an application of artificial intelligence (AI) that provides systems the ability to automatically learn and improve from experience without being explicitly programmed. Machine learning focuses on the development of computer programs that can access data and use it learn for themselves.

Uses:

Consider how you would write a spam filter using traditional programming techniques

- First you would look at what spam typically looks like. You might notice that some words or phrases (such as “credit card,” “free,” and “amazing”) tend to come up a lot in the subject. Perhaps you would also notice a few other patterns in the sender’s name, the email’s body, and so on.
- You would write a detection algorithm for each of the patterns that you noticed, and your program would flag emails as spam if a number of these patterns are detected.
- You would test your program, and repeat steps 1 and 2 until it is good enough.

There are four types of machine learning:

a) Supervised Learning

In supervised learning, the training data you feed to the algorithm includes the desired solutions, called labels.

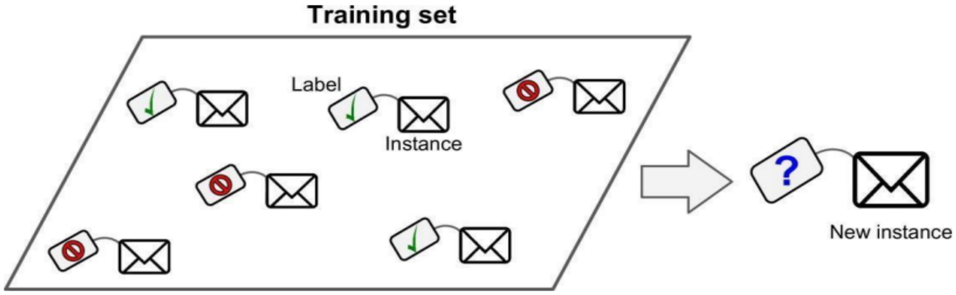


Fig 5.4.2 (i): Supervised Learning

A typical supervised learning task is classification. The spam filter is a good example of this: it is trained with many example emails along with their class (spam or ham), and it must learn how to classify new emails.

Here are some of the most important supervised learning algorithms:

- Regression
- Logistic Regression
- Classification
- Naive Bayes Classifiers
- K-NN (k nearest neighbours)
- Decision Trees
- Support Vector Machine

b) Unsupervised Learning

In unsupervised learning, as you might guess, the training data is unlabelled .The system tries to learn without a teacher.

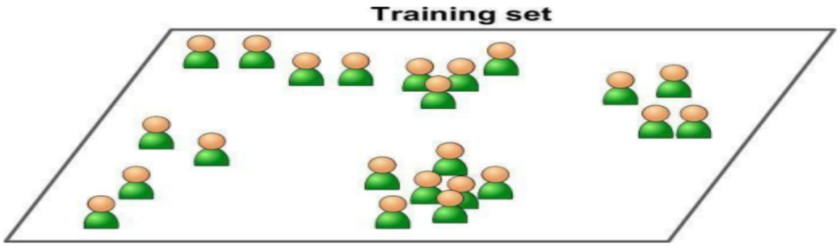


Fig 5.4.2 (ii): Unsupervised Learning

Here are some of the most important unsupervised learning algorithms
Clustering Types: -

- Hierarchical clustering
- K-means clustering
- Principal Component Analysis
- Singular Value Decomposition
- Independent Component Analysis

c) Semi-Supervised Learning

Semi-supervised learning is a type of machine learning that falls in between supervised and unsupervised learning. It is a method that uses a small amount of labelled data and a large amount of unlabelled data to train a model. The goal of semi-supervised learning is to learn a function that can accurately predict the output variable based on the input variables, similar to supervised learning. However, unlike supervised learning, the algorithm is trained on a dataset that contains both labelled and unlabelled data.

Semi-supervised learning is particularly useful when there is a large amount of unlabelled data available, but it’s too expensive or difficult to label all of it.

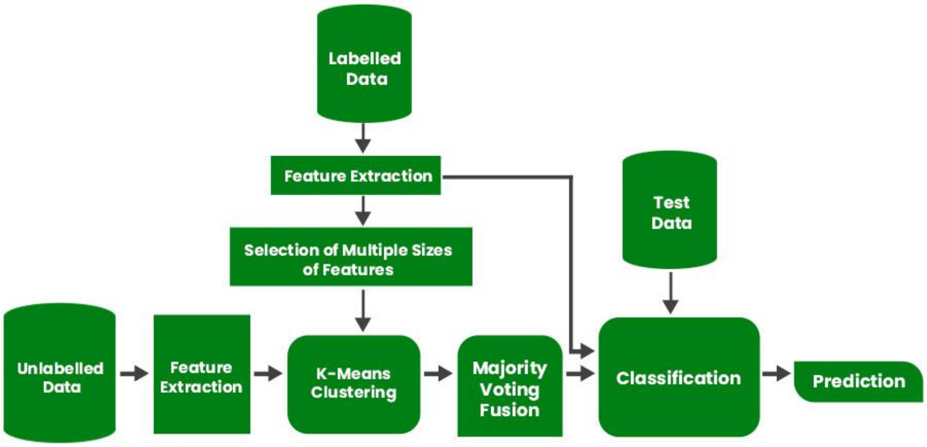


Fig 5.4.2(iii): Semi-Supervised Learning Flowchart

d) Reinforcement Learning

Reinforcement Learning (RL) is a branch of machine learning focused on making decisions to maximize cumulative rewards in a given situation. Unlike supervised learning, which relies on a training dataset with predefined answers, RL involves learning through experience. In RL, an agent learns to achieve a goal in an uncertain, potentially complex environment by performing actions and receiving feedback through rewards or penalties.

3) Algorithm Study :

a) Random Forest

Random Forest algorithm is a powerful tree learning technique in Machine Learning. It works by creating a number of Decision Trees during the training phase. Each tree is constructed using a random subset of the data set to measure a random subset of features in each partition. This randomness introduces variability among individual trees, reducing the risk of overfitting and improving overall prediction performance.

In prediction, the algorithm aggregates the results of all trees, either by voting (for classification tasks) or by averaging (for regression tasks) This collaborative decision-making process, supported by multiple trees with their insights, provides an example stable and precise results. Random forests are widely used for classification and regression functions, which are known for their ability to handle complex data, reduce overfitting, and provide reliable forecasts in different environments.

b) Support Vector Machine (SVM)

A Support Vector Machine (SVM) is a powerful machine learning algorithm widely used for both linear and nonlinear classification, as well as regression and outlier detection tasks. SVMs are highly adaptable, making them suitable for various applications such as text classification, image classification, spam detection, handwriting identification, gene expression analysis, face detection, and anomaly detection.

SVMs are particularly effective because they focus on finding the maximum separating hyperplane between the different classes in the target feature, making them robust for both binary and multiclass classification. In this outline, we will explore the Support Vector Machine (SVM) algorithm, its applications, and how it effectively handles both linear and nonlinear classification, as well as regression and outlier detection tasks.

VI. IMPLEMENTATION AND TESTING

A. Implementation and Testing

1) Implementation Approaches

Following are the steps to be followed for implementing the system:

- Install required software on server machine like VS Code or Jupyter Notebook, Python extensions
- Upload the datasets in VS code/Jupyter Notebookand create one ".py" file to write your code
- 3.Install the required packages mentioned in the document above
- Write code in created ".py" file and run

Dataset Used : Banking Consumer Complaint Dataset (complaints.csv)

Dataset Description: Data files: This data set contains 999285 consumer complaints with the date received,product,sub-product, submitted via and company information.

Features:

feature name	Description	Type
Date received	date of the complaint	Continuous
Product	complaint category	Discrete
Sub-product	complaint sub-category	Discrete
Issue	issue category	Discrete
Sub-issue	Issue sub-category	Discrete
Consumer complaint narrative	Detail info of complaint or dispute raised	Continuous
Company public response	action taken by company	Discrete
Company	company name	Discrete
State	number of urgent packets	Discrete

feature name	Description	Type
ZIP code	zip code of the company	Discrete
Tags	specific info about the state	Discrete
Consumer consent provided?	Yes/No value	Discrete
Submitted via	Source info of the complaint	Discrete
Date sent to company	Yes/No value	Discrete
Company response to consumer	Company response to the issue	Continuous
Timely response?	Yes/No value	Discrete
Consumer disputed?	Yes/No value	Discrete
Complaint ID	Unique ID of each complaint	Continuous

Table 1: Basic features columns used

B. Steps of Implementation

1) Step 1 – Data Preprocessing:

Code**:** Importing libraries and loading the dataset

```
import pandas as pd
import plotly.express as px
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
import missingno as ms
import nltk
from nltk.tokenize import word_tokenize
from nltk.corpus import stopwords
from nltk.stem import WordNetLemmatizer
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.decomposition import LatentDirichletAllocation
from textblob import TextBlob
from sklearn.model_selection import train_test_split
from sklearn.svm import SVC
from sklearn.ensemble import RandomForestClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.metrics import classification_report, accuracy_score
```

```
# Load data set into a dataframe
df = pd.read_csv('complaints.csv')
text_col = 'consumer complaint narrative'
# Preview
print("Initial DataFrame:")
print(df.head())
```

>Output

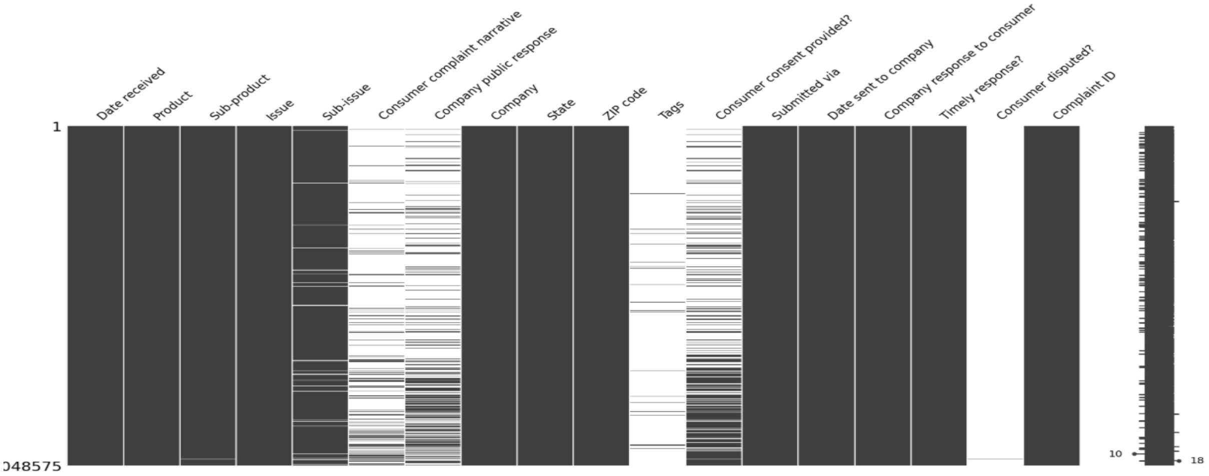
```
Initial DataFrame:
  Date received                                   Product  \
0   08-04-2025  Credit reporting or other personal consumer re...
1   08-04-2025  Credit reporting or other personal consumer re...
2   08-04-2025  Credit reporting or other personal consumer re...
3   08-04-2025  Credit reporting or other personal consumer re...
4   08-04-2025  Credit reporting or other personal consumer re...

       Sub-product                                   Issue  \
0  Credit reporting  Problem with a company's investigation into an...
1  Credit reporting             Incorrect information on your report
2  Credit reporting                    Improper use of your report
3  Credit reporting             Incorrect information on your report
4  Credit reporting             Incorrect information on your report
                                        Sub-issue  \
0  Their investigation did not fix an error on yo...
1              Information belongs to someone else
2      Reporting company used your report improperly
3                   Account information incorrect
4              Information belongs to someone else
  Consumer complaint narrative Company public response        Company State  \
0                        NaNNaN  EQUIFAX, INC.   CA
1                        NaNNaN  EQUIFAX, INC.   NY
2                        NaNNaN  EQUIFAX, INC.   PA
3                        NaNNaN  EQUIFAX, INC.   ME
4                        NaNNaN  EQUIFAX, INC.   TX

  ZIP code Tags Consumer consent provided? Submitted via Date sent to company  \
0   93675  NaNNaN          Web          08-04-2025
1   10801  NaNNaN          Web          08-04-2025
2   19143  NaNNaN          Web          08-04-2025
3   040XX  NaNNaN          Web          08-04-2025
4   75020  NaNNaN          Web          08-04-2025

  Company response to consumer Timely response? Consumer disputed?  \
0                 In progress            Yes              NaN
1                 In progress            Yes              NaN
2                 In progress            Yes              NaN
3                 In progress            Yes              NaN
4                 In progress            Yes              NaN
```

```
  Complaint ID
0      12869120
1      12869125
2      12869127
3      12869129
4      12869131
```

Code: Analysing missing values in the dataset

```
# Missing values visualization
ms.matrix(df)
plt.show()
```

>Output



```
# Convert date column to datetime and extract features
def object_to_datetime_features(df, column):
df[column] = df[column].astype('datetime64[ns]')
df['Year'] = df[column].dt.year
df['Month'] = df[column].dt.month
df['Day'] = df[column].dt.day
df['DoW'] = df[column].dt.dayofweek
df['DoW'] = df['DoW'].replace({0:'Monday',1:'Tuesday',2:'Wednesday',
                              3:'Thursday',4:'Friday',5:'Saturday',6:'Sunday'})
   return df

df = object_to_datetime_features(df, 'Date received')
print("DataFrame after extracting datetime features:")
print(df.head())
```

>Output

```
DataFrame after extracting datetime features:
  Date received                                    Product  \
0    2025-08-04  Credit reporting or other personal consumer re...
1    2025-08-04  Credit reporting or other personal consumer re...
2    2025-08-04  Credit reporting or other personal consumer re...
3    2025-08-04  Credit reporting or other personal consumer re...
```

```
4   2025-08-04  Credit reporting or other personal consumer re...

        Sub-product                                Issue  \
0  Credit reporting  Problem with a company's investigation into an...
1  Credit reporting            Incorrect information on your report
2  Credit reporting                 Improper use of your report
3  Credit reporting            Incorrect information on your report
4  Credit reporting            Incorrect information on your report

                                  Sub-issue  \
0  Their investigation did not fix an error on yo...
1              Information belongs to someone else
2      Reporting company used your report improperly
3                  Account information incorrect
4              Information belongs to someone else

  Consumer complaint narrative Company public response       Company State  \
0                        NaNNaN  EQUIFAX, INC.    CA
1                        NaNNaN  EQUIFAX, INC.    NY
2                        NaNNaN  EQUIFAX, INC.    PA
3                        NaNNaN  EQUIFAX, INC.    ME
4                        NaNNaN  EQUIFAX, INC.    TX

  ZIP code  ... Submitted via Date sent to company  \
0   93675  ...          Web          08-04-2025
1   10801  ...          Web          08-04-2025
2   19143  ...          Web          08-04-2025
3   040XX  ...          Web          08-04-2025
4   75020  ...          Web          08-04-2025

  Company response to consumer Timely response? Consumer disputed?  \
0                  In progress            Yes              NaN
1                  In progress            Yes              NaN
2                  In progress            Yes              NaN
3                  In progress            Yes              NaN
4                  In progress            Yes              NaN

  Complaint ID  Year  Month  Day     DoW
0     12869120  2025      8    4  Monday
1     12869125  2025      8    4  Monday
2     12869127  2025      8    4  Monday
3     12869129  2025      8    4  Monday
4     12869131  2025      8    4  Monday

[5 rows x 22 columns]
```

2) Step 2 – Data Preprocessing:

Code:

```
# Normalize column names
def normalise_column_names(df):
```

```
normalised_features = [i.lower() for i in list(df.columns)]
df.columns = normalised_features
  return df

df = normalise_column_names(df)
print("DataFrame after normalizing column names:")
print(df.head())
```

>Output

```
DataFrame after normalizing column names:
  date received                                        product  \
0   2025-08-04  Credit reporting or other personal consumer re...
1   2025-08-04  Credit reporting or other personal consumer re...
2   2025-08-04  Credit reporting or other personal consumer re...
3   2025-08-04  Credit reporting or other personal consumer re...
4   2025-08-04  Credit reporting or other personal consumer re...

       sub-product                                          issue  \
0  Credit reporting  Problem with a company's investigation into an...
1  Credit reporting              Incorrect information on your report
2  Credit reporting                      Improper use of your report
3  Credit reporting              Incorrect information on your report
4  Credit reporting              Incorrect information on your report

                                         sub-issue  \
0  Their investigation did not fix an error on yo...
1                Information belongs to someone else
2      Reporting company used your report improperly
3                     Account information incorrect
4                Information belongs to someone else

  consumer complaint narrative company public response        company state  \
0                        NaNNaN  EQUIFAX, INC.    CA
1                        NaNNaN  EQUIFAX, INC.    NY
2                        NaNNaN  EQUIFAX, INC.    PA
3                        NaNNaN  EQUIFAX, INC.    ME
4                        NaNNaN  EQUIFAX, INC.    TX

  zip code  ... submitted via date sent to company  \
0   93675  ...          Web         08-04-2025
1   10801  ...          Web         08-04-2025
2   19143  ...          Web         08-04-2025
3   040XX  ...          Web         08-04-2025
4   75020  ...          Web         08-04-2025

  company response to consumer timely response? consumer disputed?  \
0              In progress             Yes              NaN
1              In progress             Yes              NaN
2              In progress             Yes              NaN
3              In progress             Yes              NaN
```

```
4          In progress          Yes          NaN

  complaint id  year  month  day     dow
0     12869120  2025      8    4  Monday
1     12869125  2025      8    4  Monday
2     12869127  2025      8    4  Monday
3     12869129  2025      8    4  Monday
4     12869131  2025      8    4  Monday

[5 rows x 22 columns]
```

```
# Normalize subset names
def normalise_subset_names(df, column):
subset_names = list(df[column].value_counts().index)
norm_subset_names = [i.lower() for i in subset_names]
dict_replace = dict(zip(subset_names, norm_subset_names))
df[column] = df[column].replace(dict_replace)
   return df

df = normalise_subset_names(df, 'product')
print("DataFrame after normalizing subset names:")
print(df.head())
```

>Output

```
DataFrame after normalizing subset names:
  date received                              product  \
0    2025-08-04  credit reporting or other personal consumer re...
1    2025-08-04  credit reporting or other personal consumer re...
2    2025-08-04  credit reporting or other personal consumer re...
3    2025-08-04  credit reporting or other personal consumer re...
4    2025-08-04  credit reporting or other personal consumer re...

       sub-product                                    issue  \
0  Credit reporting  Problem with a company's investigation into an...
1  Credit reporting            Incorrect information on your report
2  Credit reporting                  Improper use of your report
3  Credit reporting            Incorrect information on your report
4  Credit reporting            Incorrect information on your report

                                 sub-issue  \
0  Their investigation did not fix an error on yo...
1            Information belongs to someone else
2     Reporting company used your report improperly
3                  Account information incorrect
4            Information belongs to someone else

  consumer complaint narrative company public response        company state  \
0                     NaNNaN  EQUIFAX, INC.    CA
1                     NaNNaN  EQUIFAX, INC.    NY
```

```
2                  NaNNaN  EQUIFAX, INC.   PA
3                  NaNNaN  EQUIFAX, INC.   ME
4                  NaNNaN  EQUIFAX, INC.   TX

  zip code  ... submitted via date sent to company  \
0   93675  ...          Web          08-04-2025
1   10801  ...          Web          08-04-2025
2   19143  ...          Web          08-04-2025
3   040XX  ...          Web          08-04-2025
4   75020  ...          Web          08-04-2025

  company response to consumer timely response? consumer disputed?  \
0               In progress            Yes             NaN
1               In progress            Yes             NaN
2               In progress            Yes             NaN
3               In progress            Yes             NaN
4               In progress            Yes             NaN

  complaint id  year  month  day     dow
0     12869120  2025      8    4  Monday
1     12869125  2025      8    4  Monday
2     12869127  2025      8    4  Monday
3     12869129  2025      8    4  Monday
4     12869131  2025      8    4  Monday

[5 rows x 22 columns]
```

```
# Keep specific subsets
def keep_subset(df, column, values):
    return df[df[column].isin(values)]

lst_keep = ['credit reporting', 'debt collection', 'mortgage', 'credit card',
            'bank account or service', 'consumer loan', 'student loan',
            'payday loan', 'prepaid card', 'money transfers',
            'other financial service', 'virtual currency']

df = keep_subset(df, 'product', lst_keep)
print("DataFrame after keeping specific subsets:")
print(df.head())
```

>Output

```
DataFrame after keeping specific subsets:
    date received          product  \
29     2025-03-24      credit card
40     2025-08-04  debt collection
72     2025-08-04  debt collection
79     2025-08-04  debt collection
102    2025-03-04  debt collection
```

```
                          sub-product  \
29   General-purpose credit card or charge card
40                        I do not know
72                        I do not know
79                        I do not know
102                      Credit card debt

                                issue  \
29   Problem with a purchase shown on your statement
40                Written notification about debt
72              Attempts to collect debt not owed
79                Written notification about debt
102             Attempts to collect debt not owed

                                  sub-issue  \
29   Card was charged for something you did not pur...
40   Notification didn't disclose it was an attempt...
72                             Debt is not yours
79   Notification didn't disclose it was an attempt...
102                Debt was result of identity theft

    consumer complaint narrative  \
29                         NaN
40                         NaN
72                         NaN
79                         NaN
102                        NaN

                         company public response  \
29   Company has responded to the consumer and the ...
40                                         NaN
72                                         NaN
79                                         NaN
102  Company believes it acted appropriately as aut...

                        company state zip code  ... submitted via  \
29                CITIBANK, N.A.    NY    11216  ...           Web
40                EQUIFAX, INC.    NY    11550  ...           Web
72                EQUIFAX, INC.    FL    32507  ...           Web
79                EQUIFAX, INC.    TX    761XX  ...           Web
102  Atlanticus Services Corporation    FL    331XX  ...           Web

    date sent to company     company response to consumer timely response?  \
29           24-03-2025  Closed with non-monetary relief            Yes
40           08-04-2025                    In progress            Yes
72           08-04-2025                    In progress            Yes
79           08-04-2025                    In progress            Yes
102          03-04-2025          Closed with explanation            Yes

    consumer disputed? complaint id  year  month  day      dow
```

```
29           NaN   12619868  2025    3  24  Monday
40           NaN   12869213  2025    8   4  Monday
72           NaN   12869258  2025    8   4  Monday
79           NaN   12869268  2025    8   4  Monday
102          NaN   12813639  2025    3   4  Tuesday

[5 rows x 22 columns]
```

```
# Filter data for specific years
df = df[df['year'].isin([2020, 2021, 2022, 2023, 2024, 2025])]
print("DataFrame after filtering for specific years:")
print(df.head())
```

>Output

```
DataFrame after filtering for specific years:
  date received        product  \
29    2025-03-24      credit card
40    2025-08-04  debt collection
72    2025-08-04  debt collection
79    2025-08-04  debt collection
102   2025-03-04  debt collection

                          sub-product  \
29   General-purpose credit card or charge card
40                        I do not know
72                        I do not know
79                        I do not know
102                     Credit card debt

                                    issue  \
29   Problem with a purchase shown on your statement
40                 Written notification about debt
72               Attempts to collect debt not owed
79                 Written notification about debt
102              Attempts to collect debt not owed

                                    sub-issue  \
29   Card was charged for something you did not pur...
40   Notification didn't disclose it was an attempt...
72                                  Debt is not yours
79   Notification didn't disclose it was an attempt...
102                  Debt was result of identity theft

  consumer complaint narrative  \
29                        NaN
40                        NaN
72                        NaN
79                        NaN
102                       NaN
```

```
                         company public response  \
29   Company has responded to the consumer and the ...
40                                             NaN
72                                             NaN
79                                             NaN
102  Company believes it acted appropriately as aut...

                     company state zip code  ... submitted via  \
29                 CITIBANK, N.A.    NY    11216  ...           Web
40                  EQUIFAX, INC.    NY    11550  ...           Web
72                  EQUIFAX, INC.    FL    32507  ...           Web
79                  EQUIFAX, INC.    TX    761XX  ...           Web
102  Atlanticus Services Corporation    FL    331XX  ...           Web

    date sent to company     company response to consumer timely response?  \
29           24-03-2025  Closed with non-monetary relief              Yes
40           08-04-2025                      In progress              Yes
72           08-04-2025                      In progress              Yes
79           08-04-2025                      In progress              Yes
102          03-04-2025          Closed with explanation              Yes

    consumer disputed? complaint id  year  month  day      dow
29                 NaN     12619868  2025      3   24   Monday
40                 NaN     12869213  2025      8    4   Monday
72                 NaN     12869258  2025      8    4   Monday
79                 NaN     12869268  2025      8    4   Monday
102                NaN     12813639  2025      3    4  Tuesday

[5 rows x 22 columns]
```

3) Step 3 – Tokenization:

```
# Tokenization
nltk.download('punkt')
df['tokens'] = df[text_col].apply(lambda x: word_tokenize(str(x)))
print("DataFrame after tokenization:")
print(df[['consumer complaint narrative', 'tokens']].head())
```

>Output

```
DataFrame after tokenization:
    consumer complaint narrative tokens
29                           NaN  [nan]
40                           NaN  [nan]
72                           NaN  [nan]
79                           NaN  [nan]
102                          NaN  [nan]
```

4) Step 4 – Removing Stop words:

```
# Remove stop words
nltk.download('stopwords')
stop_words = set(stopwords.words('english'))
```

```
df['tokens'] = df['tokens'].apply(lambda x: [word for word in x if word.lower() not in stop_words])
print("DataFrame after removing stop words:")
print(df[['consumer complaint narrative', 'tokens']].head())
```

Output :

```
DataFrame after removing stop words:
    consumer complaint narrative tokens
29                        NaN  [nan]
40                        NaN  [nan]
72                        NaN  [nan]
79                        NaN  [nan]
102                        NaN  [nan]
```

5) Step 5 – Lemmatization:

```
# Lemmatization
nltk.download('wordnet')
lemmatizer = WordNetLemmatizer()
df['lemmatized_tokens'] = df['tokens'].apply(lambda x: [lemmatizer.lemmatize(token) for token in x])
print("DataFrame after lemmatization:")
print(df[['tokens', 'lemmatized_tokens']].head())
```

Output :

```
DataFrame after lemmatization:
    tokens lemmatized_tokens
29   [nan]            [nan]
40   [nan]            [nan]
72   [nan]            [nan]
79   [nan]      [nan]
102  [nan]      [nan]
```

6) Step 6 – TF-IDF Vectorization:

```
# TF-IDF Vectorization
tfidf = TfidfVectorizer(max_features=1000)
tfidf_matrix = tfidf.fit_transform(df[text_col].fillna(''))
print("TF-IDF matrix shape:", tfidf_matrix.shape)
```

>Output

```
TF-IDF matrix shape: (72000, 1000)
```

7) Step 7 – Sentiment Analysis:

```
# Sentiment Analysis
df['sentiment'] = df[text_col].apply(lambda x: TextBlob(str(x)).sentiment.polarity)
print("DataFrame after sentiment analysis:")
print(df[['consumer complaint narrative', 'sentiment']].head())
```

Output:

```
DataFrame after sentiment analysis:
    consumer complaint narrative  sentiment
29                        NaN        0.0
40                        NaN        0.0
```

```
72                    NaN       0.0
79                    NaN       0.0
102                    NaN       0.0
```

8) Step 8 – Topic Modeling:

```
# Topic Modeling
lda = LatentDirichletAllocation(n_components=5, random_state=42)
lda.fit(tfidf_matrix)
df['topic'] = lda.transform(tfidf_matrix).argmax(axis=1)
print("DataFrame after topic modeling:")
print(df[['consumer complaint narrative', 'topic']].head())
```

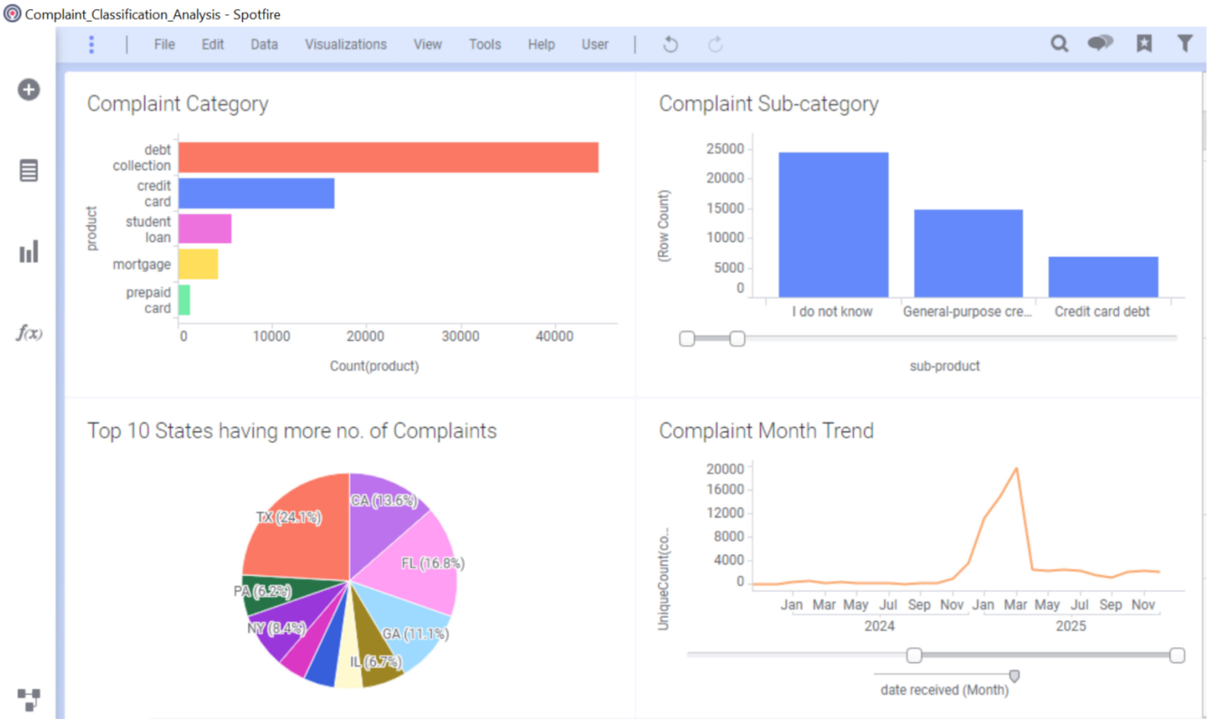
Output :

```
DataFrame after topic modeling:
    consumer complaint narrative  topic
29                    NaN      0
40                    NaN      0
72                    NaN      0
79                    NaN      0
102                    NaN      0
```

9) Step 9 – Classification Modeling:

```
# Split data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

# Support Vector Machine (SVM)
svm = SVC()
svm.fit(X_train, y_train)
y_pred_svm = svm.predict(X_test)
print("SVM Classification Report:")
print(classification_report(y_test, y_pred_svm))
print("SVM Accuracy:", accuracy_score(y_test, y_pred_svm))

# Random Forest
rf = RandomForestClassifier()
rf.fit(X_train, y_train)
y_pred_rf = rf.predict(X_test)
print("Random Forest Classification Report:")
print(classification_report(y_test, y_pred_rf))
print("Random Forest Accuracy:", accuracy_score(y_test, y_pred_rf))
```

10) Step 10 – Exporting the modified dataset into csv file and create a useful dashboard out of it on any BI tool like (Spotfire, Power BI, Tableau or any other)



C. Testing

1) Unit Testing:

Unit testing focuses verification effort on the smallest unit of software i.e. module. Using the detailed design and the process specification testing is done to uncover errors within the boundary of the module. All modules must be successful in the unit test beforethe start of the integration testing begins.

2) Integration Testing:

After the unit testing we have to perform integration testing. The goal is to here is to see ifmodules can be integrated properly, the emphasis being on testing interfaces between modules. Thistesting activity can be considered as testing the design and hence the emphasis on testing moduleinteractions.In this project integrating the entire module forms the main system. When integrating all the modulesI have checked whether the integration effects working of any of the services by giving differentcombinations of inputs with which the two services run perfectly before integration.

VII. CONCLUSION AND FUTURE ENHANCEMENT

A. Conclusion

The proposed model for classifying consumer complaints using NLP and machine learning demonstrates a robust and efficient approach to handling consumer grievances. By leveraging advanced text preprocessing techniques, feature extraction methods, sentiment analysis, topic modeling, and classification algorithms, the model achieves high accuracy in categorizing complaints into predefined categories. This automated system significantly reduces the manual effort required for complaint classification, enhances customer service response times, and provides valuable insights into common issues faced by consumers.

Key Benefits:

1) Automated Classification: The system automates the classification process, ensuring that complaints are accurately categorized and routed to the appropriate departments.
2) Improved Accuracy: The use of machine learning models like SVM and Random Forest ensures high accuracy in classification, leading to better customer service.
3) Operational Efficiency: By reducing the manual effort required for complaint handling, the system improves operational efficiency and allows customer service teams to focus on resolving issues.

4) Insights and Analysis: Sentiment analysis and topic modeling provide deeper insights into consumer complaints, helping businesses understand common issues and trends.
5) Enhanced Customer Service: The system improves response times and customer satisfaction by efficiently routing complaints to the appropriate departments.

B. Future Enhancement

While the proposed model offers significant benefits, there are several areas for future enhancement to further improve its performance and capabilities:

1) Integration of Deep Learning Models:
- Advanced Models: Incorporate deep learning models like LSTM, Bi-LSTM, GRU, and CNN to capture complex patterns and dependencies in text data. These models can improve classification accuracy and handle more nuanced complaints.

2) Use of Pre-trained Language Models:
- BERT and DistilBERT: Utilize pre-trained language models like BERT and DistilBERT for feature extraction. These models can capture semantic relationships between words and improve the performance of classification tasks.

3) Enhanced Sentiment Analysis:
- Fine-tuning: Fine-tune sentiment analysis models to better understand the emotions and satisfaction levels of consumers. This can provide more accurate insights into customer sentiment.

4) Real-time Processing:
- Streaming Data: Implement real-time processing capabilities to handle streaming data and classify complaints as they are received. This can further improve response times and operational efficiency.

5) Multi-language Support:
- Language Models: Extend the model to support multiple languages, allowing businesses to handle complaints from a diverse customer base. This can be achieved by using language-specific models and embeddings.

6) Enhanced Visualization and Reporting:
- Dashboards: Develop interactive dashboards to visualize classification results, sentiment analysis, and topic modeling insights. This can help businesses monitor trends and make data-driven decisions.

7) Continuous Learning and Adaptation:
- Feedback Loop: Implement a feedback loop to continuously update and improve the model based on new data and user feedback. This ensures that the system remains accurate and relevant over time.

REFERENCES

Bibliography:

[1] Kulkarni, C.S., Bhavsar, A.U., Pingale, S.R. and Kumbhar, S.S, "BANK CHAT BOT – An Intelligent Assistant System Using NLP and Machine Learning," International Research Journal of Engineering and Technology, vol. 04 , no. 05 , May -2017.
[2] Tutika, A. and Nagesh, M.Y.V., "Restaurant reviews classification using NLP Techniques," Journal of Information and Computational Science, vol. 9, no. 11, 2019.
[3] Towfighi, S., Agarwal, A., Mak, D.Y. and Verma, A., "Labelling chest x-ray reports using an open-source NLP and ML tool for text data binary classification," medRxiv, November 22, 2019.

Websites:

[1] https://www.kaggle.com/
[2] https://github.com/