TICKET CLASSIFICATION DATA REPORT.

GROUP 2

AUTHORS;
Vincent Mutuku, Western Onzere, Felix Mwendwa, Kennedy Kariuki, Michelle Kavetza, Zena Weru

Executive Summary

This project focuses on developing an automated system to classify customer complaints for a financial institution using Natural Language Processing (NLP) techniques. The project aims to transform unstructured text complaints into actionable insights that can be routed efficiently to the appropriate department, thereby reducing manual review and improving service quality.

Problem Statement

As the company scales, the current **manual process** of categorizing customer complaints becomes inefficient, leading to **delays in issue resolution and increased costs**. To **automate and streamline** this process, **Natural Language Processing (NLP) techniques** will be used to build a model that can automatically classify complaints into predefined categories.

Key objectives include:

Streamlining Complaint Routing: Automatically categorizing complaints into predefined groups such as Credit Card/Prepaid Card, Bank Account Services, Theft/Dispute Reporting, Mortgages/Loans, and Others.

Reducing Resolution Times: Ensure that complaints are addressed promptly.

Enhancing Customer Experience: Through improved response times and targeted resolutions.

Business Understanding and Objective

Business Context

Financial institutions rely on customer feedback to gauge service quality. Customer complaints are a critical indicator of areas requiring attention, as unresolved issues can lead to dissatisfaction, higher churn rates, and reputational damage. Currently, the institution processes complaints that are submitted as unstructured text, which then requires manual review and categorization, a time-consuming and error-prone process.

Business Objective

The primary goal is to automate the classification of customer complaints. By leveraging NLP techniques, our system will:

- Automatically assign complaints to one of the five predefined product or service categories.
- Improve routing efficiency by reducing manual intervention.
- Reduce resolution time and enhance overall customer satisfaction.

Data Understanding & Preparation

Data Loading and Initial Setup

Our data is stored in a JSON file. The first step in our process was to load the JSON file using Python's json and pandas libraries. Once the file was loaded, we converted it into a DataFrame for easier manipulation and analysis. This step ensured that the data was in a tabular format, which is ideal for subsequent analysis.

Correcting Data Structure

The raw JSON contained a column named '_source' with nested dictionaries in string format. To make the data analysis-ready:

Conversion: We used 'ast.literal_eval' to convert these string representations into actual Python dictionaries.

Normalization: The nested dictionary was then normalized (or “flattened”) so that each key became an individual column.

Cleanup: Redundant columns such as ‘_index’, ‘_id’, ‘zip_code’, and complaint_id were dropped to reduce clutter and focus on relevant information. This step is crucial because it standardizes the data and makes it easier to apply further cleaning and analysis.

Data Outlook and Type Correction

After structural corrections:

Inspection: We used commands like ‘df.info()’, ‘df.head()’, and ‘df.describe()’ to inspect the DataFrame. This gave us an overview of the dataset’s dimensions, column data types, and sample records.

Date Conversion:

Columns containing dates (e.g., date_received and date_sent_to_company) were converted into datetime objects. This conversion allows for time-based analyses, such as tracking complaint trends over time and calculating the duration between complaint receipt and response.

Data Cleaning Procedures

Our data-cleaning process included several important steps:

Handling Missing Values:
We identified columns with a high percentage of missing data. Columns with more than 50% missing values were dropped. For the remaining columns (e.g., state), missing values were imputed using KNN imputation. This method estimates missing entries by considering the values from similar records.

Standardizing Categorical Values:
For fields like consumer_consent_provided, inconsistent entries (e.g., "N/A", "Other", or null values) were standardized to a single consistent value ("Consent not provided"). This ensures that our analysis does not misinterpret variations in the same category.
Outlier Detection and Removal:
Outliers in numerical columns such as duration (the time between complaint receipt and

company response) were detected using the Interquartile Range (IQR) method. Visual tools such as box plots were used to identify and confirm the presence of outliers. Subsequently, extreme values that could distort the overall analysis were removed.

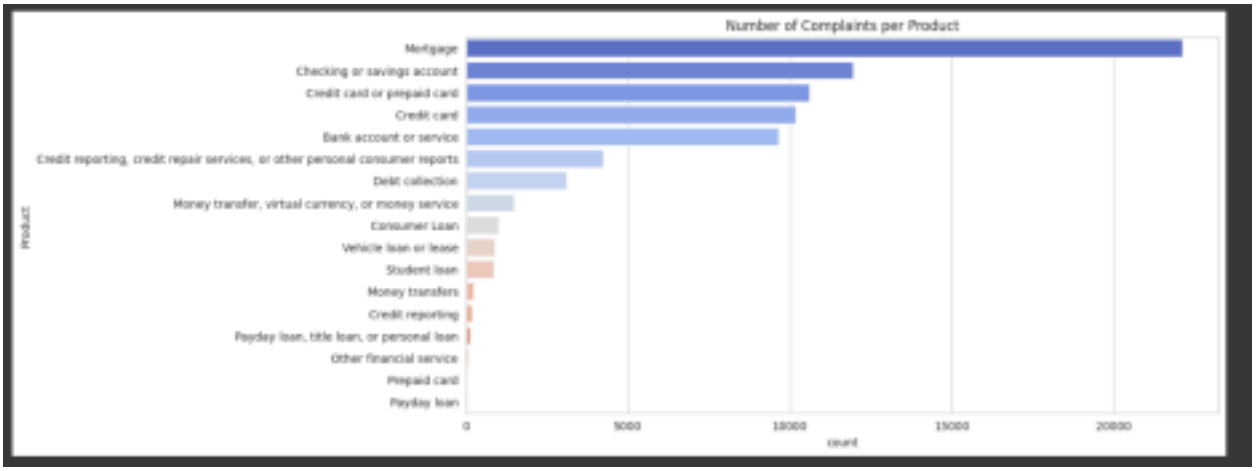
Exploratory Data Analysis (EDA)

Univariate Analysis

Analyzing Complaint Frequencies

Complaints per Product

We plotted a count plot showing the number of complaints received for each financial product.

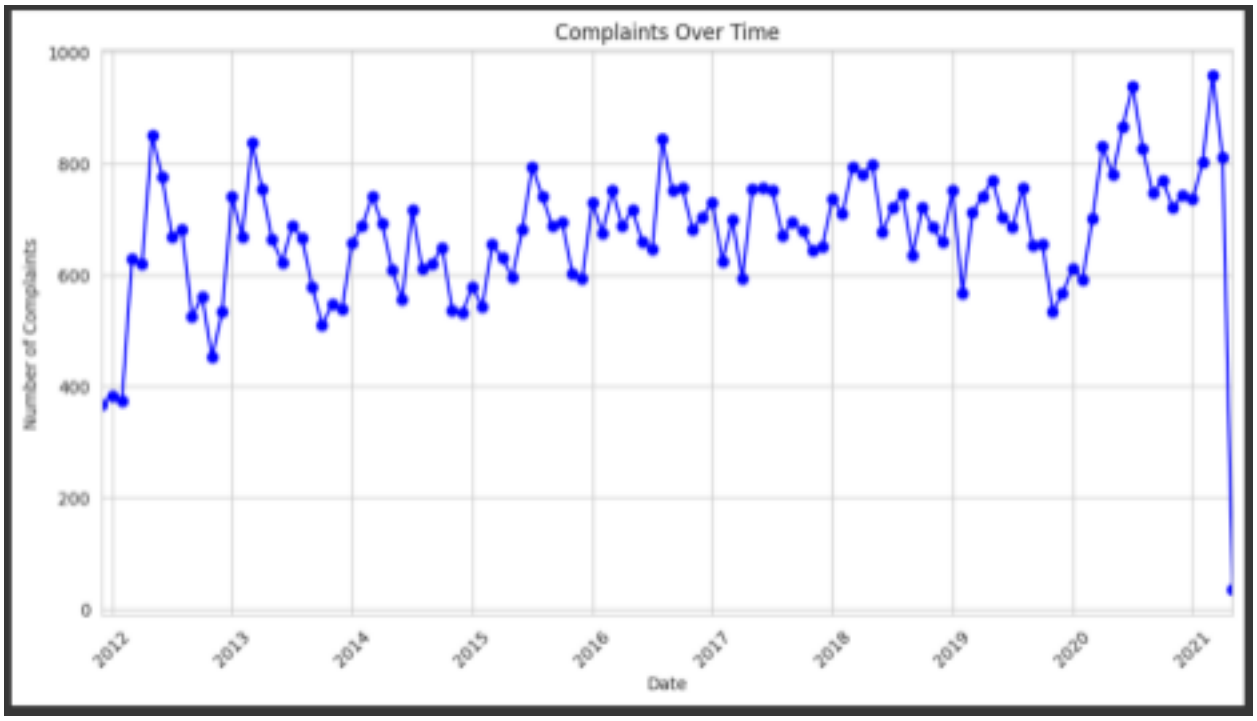


This visualization helps to identify which products (e.g., Credit Cards, Bank Accounts, Mortgages) receive the highest volume of complaints.

Temporal Analysis of Complaints

Complaints Over Time

By converting the ‘complaint_received_date' to a datetime format, we created a line chart that tracks the number of complaints over time.

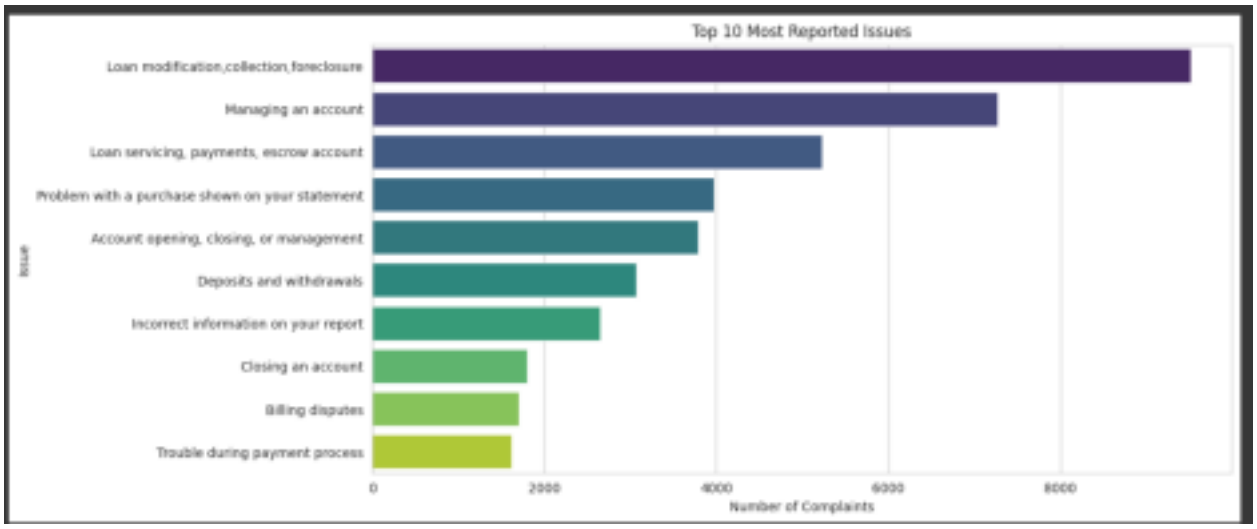


This analysis reveals trends such as seasonal spikes or periods of high complaint volume.

Analysis of Top Reported Issues

Identifying Frequent Issues

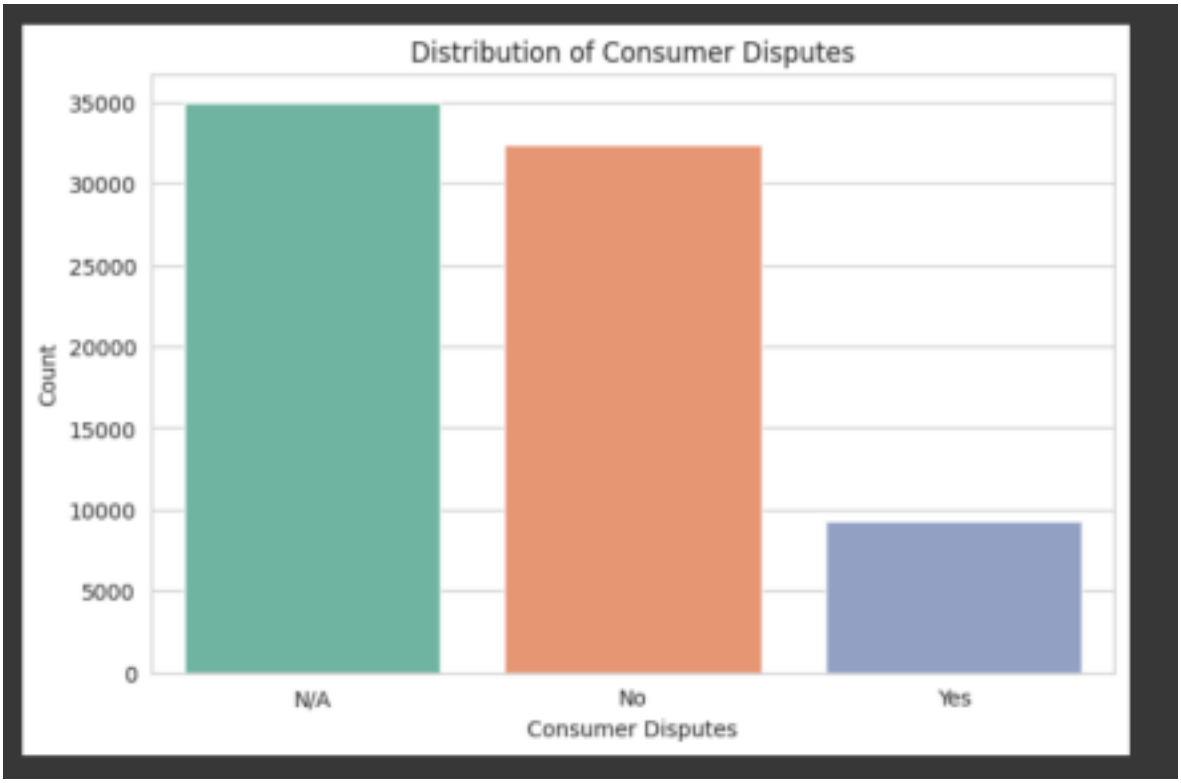
A bar chart was used to identify the top 10 most frequently reported issues. This step helps in pinpointing the most common problems faced by customers.



Consumer Dispute Distribution

We examined the distribution of consumer disputes. This analysis helps us understand how many

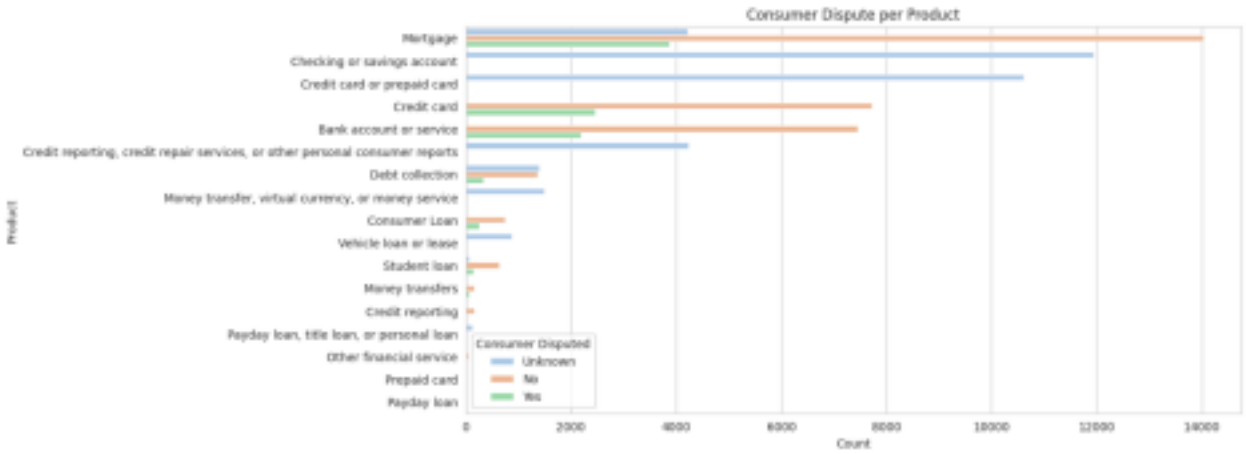
complaints involve a dispute from the consumer's side. A higher number of disputes can indicate areas where customers are less satisfied with the response or resolution.



Bivariate Analysis

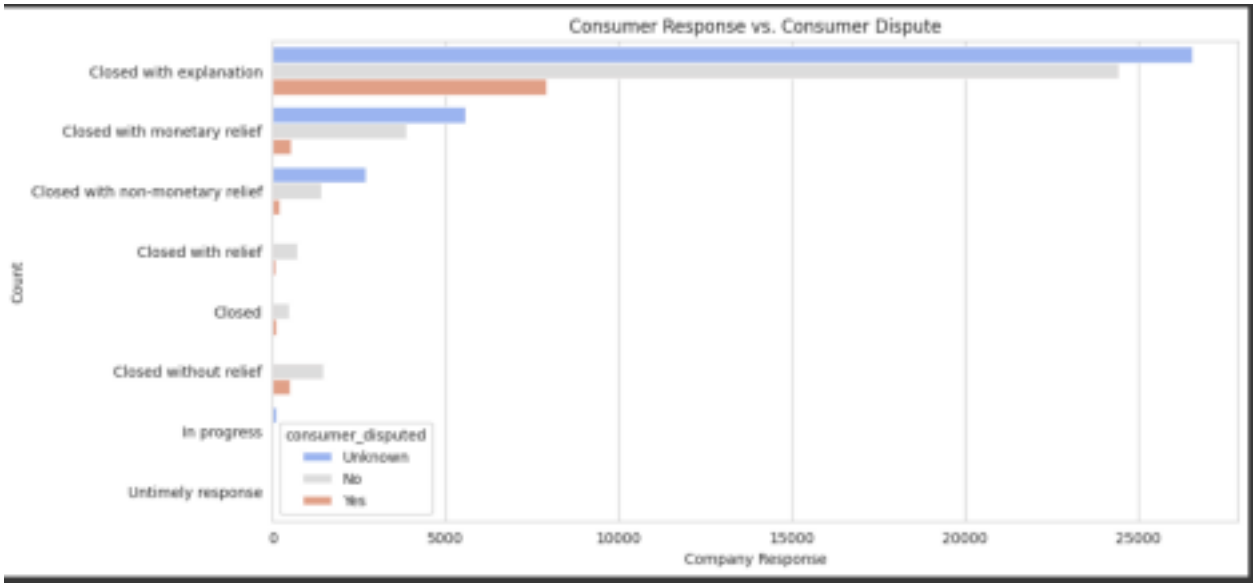
Consumer Dispute by Product:

We compared the frequency of consumer disputes across different products using a count plot. This analysis shows whether certain products experience a higher rate of disputes. The visualization provides insight into whether products with higher complaint volumes also have a higher incidence of disputes, suggesting potential dissatisfaction.



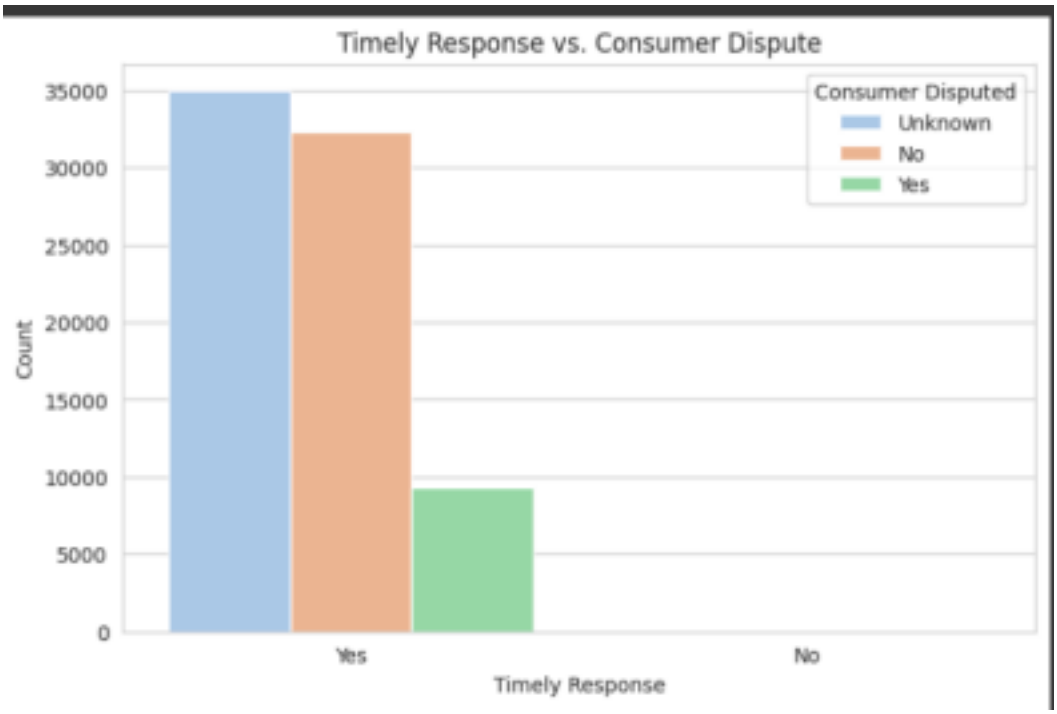
Company Response vs. Consumer Dispute:

We plotted another count plot to examine the relationship between the type of company response and the occurrence of consumer disputes. The count plot highlights how effective different response strategies are. For instance, "Closed with Explanation" may correlate with fewer disputes, suggesting that clear communication is key.



Timely Response vs. Consumer Dispute:

We also analyzed how the timeliness of a response affects consumer disputes. By comparing the frequency of disputes between timely and untimely responses, we can assess the impact of response speed on customer satisfaction. The findings indicate that while most responses are timely, even a small number of delayed responses can lead to a disproportionate number of disputes.



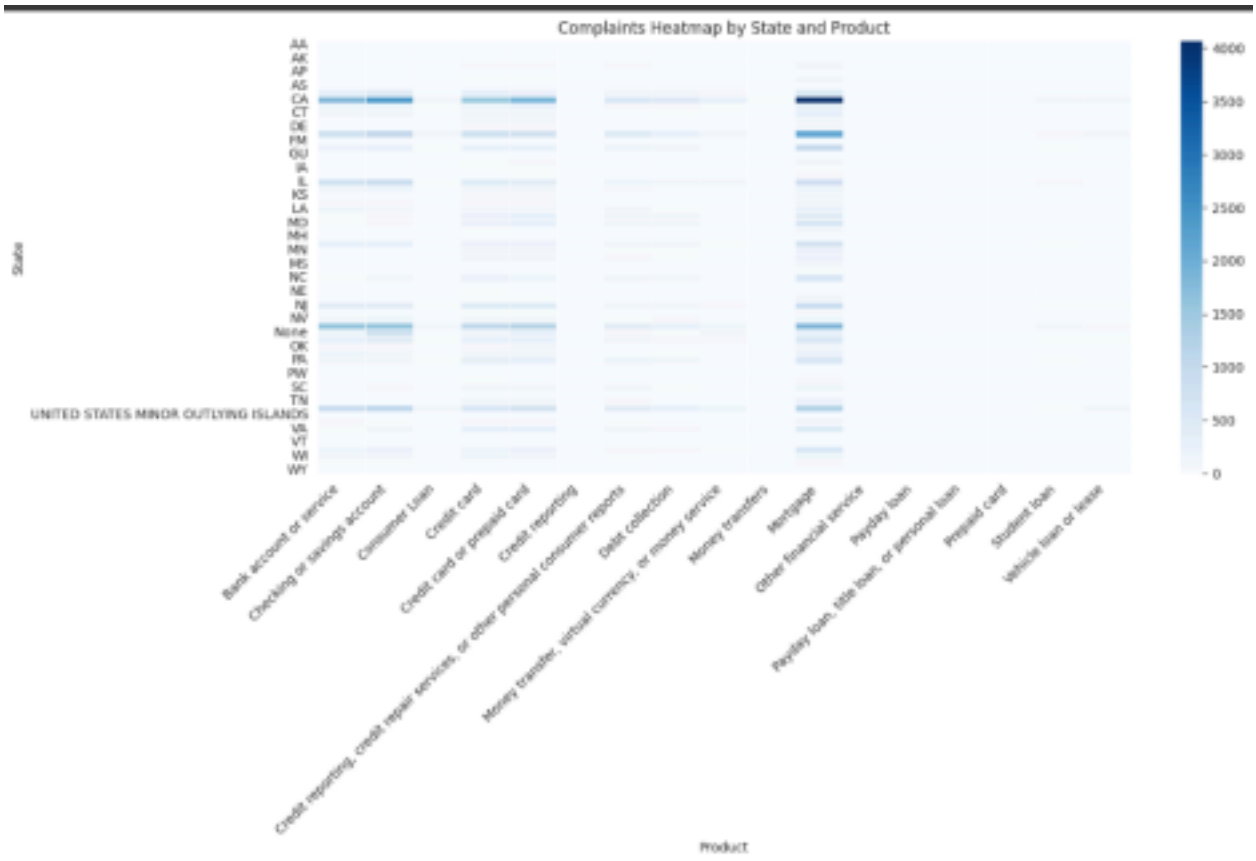
Multivariate Analysis

Complaints by Product, State, and Timely Response:

Using faceted categorical plots, we examined how complaints vary by product and state and how timely responses differ across these categories. This visualization helps identify geographic trends, such as specific states that may have higher complaint volumes, and reveals if timely responses vary significantly by region or product.

Heatmap of Complaints by State and Product:

We also plotted a heat map to illustrate the density of complaints across different states and products. Darker areas in the heatmap indicate higher complaint volumes. This helps the institution pinpoint hotspots where customer dissatisfaction is most pronounced and where regulatory attention might be necessary.



Natural Language Processing

Text Preprocessing

Converting and Cleaning Text

Our initial step was to ensure that all complaint texts were in a string format, as text operations require consistent data types. We then defined a cleaning function that uses regular expressions to:

- **Convert to Lowercase:**

  This standardizes the text, ensuring that words like "Bank" and "bank" are treated equally.
- **Remove Bracketed Content:**
  By stripping out any text contained within square brackets, we removed extraneous details that do not contribute to our analysis.

- **Eliminate Punctuation:**

Punctuation is removed because it rarely provides value in topic modeling or frequency analysis.

- **Remove Words Containing Numbers:**
  Numbers often appear as noise (e.g., ticket IDs or dates) and can skew frequency counts.

After cleaning, the resulting text is stored in a new column. We then filtered out any rows that became empty after cleaning, ensuring our dataset retained only meaningful content.

Tokenization

Once the text is cleaned, the next step is tokenization. We applied the NLTK tokenizer to split each complaint into individual words (tokens). This segmentation is fundamental for any subsequent analysis, such as frequency counting or further linguistic processing.

Lemmatization

After tokenization, we lemmatized the tokens using the WordNetLemmatizer. Lemmatization reduces words to their base or dictionary form (e.g., "running" becomes "run"), which helps group similar words and reduce the overall vocabulary size. The lemmatized tokens are then re-joined into a single string per complaint, creating a new column that represents the simplified form of the text.

Parts-of-Speech (POS) Tagging

Extraction of Nouns

For our topic modeling, we focused on extracting nouns words that often carry significant semantic meaning in a complaint. We processed each lemmatized complaint and filtered out only singular and plural nouns. This step reduces noise by excluding less informative parts of speech and retaining words that are more likely to be relevant to topics.

Word Frequency Analysis and Visualization

WordCloud Generation

To gain insight into the most common terms in the complaints, we aggregated all the extracted nouns into a single string and calculated word frequencies. A WordCloud is generated using these frequencies, visually emphasizing more frequent words with larger fonts. This offers an intuitive overview of the dominant terms in the dataset.

Removing Masked Words
We also apply an additional cleaning function to remove any masked words (e.g., patterns with three or more consecutive 'x' characters) that might not contribute useful information. This ensures the word frequency analysis remains focused on meaningful content.

N-Gram Analysis

Besides unigrams, we analyzed bigrams and trigrams to capture common phrases that may indicate specific issues or contexts in complaints. Using a vectorizer, we extracted and counted the top n-grams.

Feature Extraction

Term Frequency-Inverse Document Frequency

What is TF-IDF?

TF-IDF (Term Frequency-Inverse Document Frequency) is a numerical statistic intended to reflect the importance of a word within a document relative to a corpus. It assigns higher scores to words that appear frequently in a document but infrequently in the corpus, effectively highlighting words that are more discriminative.

- **max_df Parameter:**
  Terms that appear in more than 95% of the documents are considered too common and are excluded.

- **min_df Parameter:**
  Terms that appear in fewer than 2 documents are considered too rare and are also excluded.

Creating the Document-Term Matrix (DTM)

Using the TfidfVectorizer initialized with the parameters above, we transformed our cleaned and POS-filtered text into a document-term matrix. This sparse matrix represents each complaint as a vector of TF-IDF scores corresponding to each term in our vocabulary.

Topic Modeling with NMF

NMF for Topic Extraction

We applied Non-Negative Matrix Factorization (NMF) on the document-term matrix to uncover latent topics within the complaints. NMF decomposes the DTM into two matrices:

- **W Matrix:** Represents the topic distribution for each document.
- **H Matrix:** Represents the weight of each word in each topic.

Extracting and Interpreting Topics

For each of the 5 topics, we extracted the top 20 words based on the weights in the H matrix. These top words are then presented in a table where each row corresponds to a topic. Based on these words, we manually assigned meaningful labels (e.g., Bank Account Services, Credit Card / Prepaid Card, Mortgages/Loans, Theft/Dispute Reporting, and Other).

Topic Allocation

We then assigned each complaint to a topic by taking the argument with the highest weight from the topic distribution. We mapped these numeric topic assignments to their corresponding labels using a dictionary, ensuring that the topics were interpretable from a business perspective.

Latent Dirichlet Allocation (LDA) for Topic Modeling

Overview and Objective

After developing our topic model using Non-Negative Matrix Factorization (NMF), we pursued a complementary approach with Latent Dirichlet Allocation (LDA). LDA offers a probabilistic framework that not only identifies latent topics but also provides deeper insights into topic distributions across our complaint corpus. This method helps validate and enhance the themes discovered via NMF.

Document-Term Matrix Creation

We began by initializing a CountVectorizer with parameters to filter out extremely common and extremely rare words while also excluding English stop words. This vectorizer transforms our

preprocessed and POS-filtered complaint texts into a Document-Term Matrix (DTM).

Applying LDA

Next, we initialized the LDA model to extract five topics from the DTM. The LDA model is then fitted on this matrix. By decomposing the DTM, LDA assigns a probability distribution over topics for each document.

Extracting and Interpreting Topics

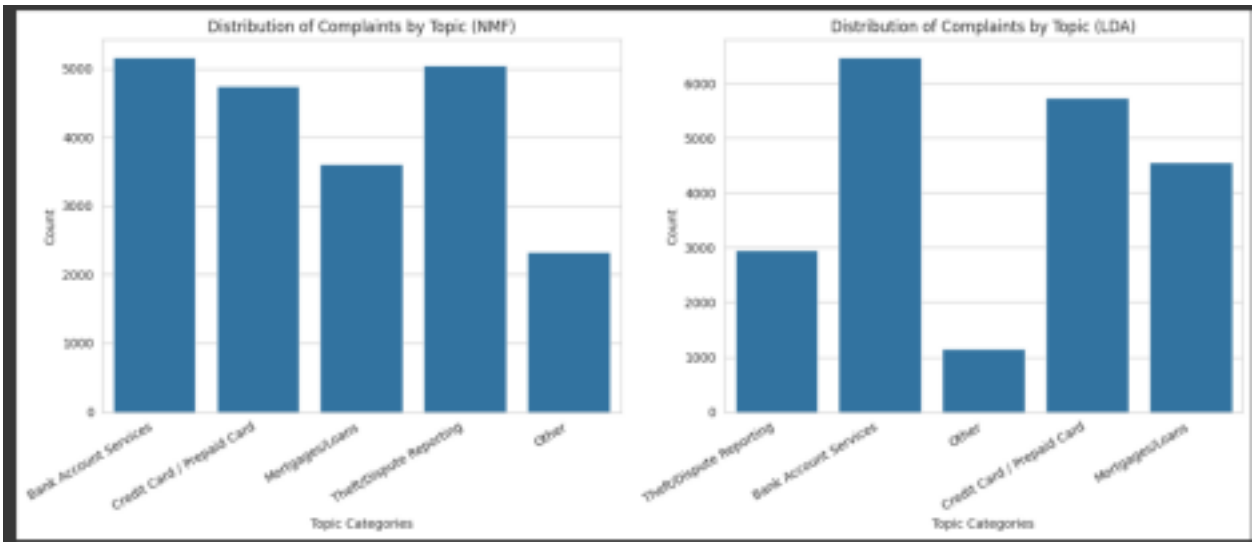
Once the LDA model was fitted, we extracted the top 20 words for each topic from the model's components. These words, ranked by their contribution to the topic, are compiled into a table. Wevthen reviewed these lists to assign meaningful labels based on key financial service themes.

For example, we may observe:

- **Topic 1:** Dominated by terms like "charge," "dispute," "fraud," and "claim," which we interpret as relating to Theft/Dispute Reporting.
- **Topic 2:** Containing words such as "account," "bank," "deposit," and "transaction," suggestive of Bank Account Services.
- **Topic 3:** Featuring terms like "fee," "bonus," and "offer," which could be grouped as Other.
- **Topic 4:** Words like "credit," "card," "payment," and "balance," indicating Credit Card/Prepaid Card issues.
- **Topic 5:** Incorporating terms such as "loan," "mortgage," and "property," clearly pointing to Mortgages/Loans.

Topic Allocation and Comparison with NMF

For each complaint, the LDA model produced a probability distribution over topics. We assigned the complaint to the topic with the highest probability (using argmax). Our DataFrame is then updated with an LDA topic column. To evaluate consistency between models, we create side-by-side count plots comparing the distribution of topics derived from NMF and LDA.



From the two distribution plots, we can observe some consistencies despite the differences in the topic modeling approaches:

- **Bank Account Services** – This topic is the most dominant in both NMF and LDA, suggesting strong clustering around banking-related complaints.
- **Credit Card / Prepaid Card** – This category also appears as a significant topic in both models, showing consistent identification of credit card-related issues.
- **Mortgages/Loans** – Both models assign a comparable number of complaints to this topic, reflecting a shared pattern in complaint distribution.
- **Theft/Dispute Reporting** – While there is some variation in frequency, this category is clearly identified in both models.
- **Other Category** – The "Other" category differs in size between the models, but it still represents a smaller portion of the dataset in both cases.

Predictive Modeling with Deep Learning

Overview and Objective

With topics now assigned to each complaint through our unsupervised methods, our next goal is to build a supervised predictive model. We aimed to train a Bidirectional Long Short-Term Memory (BiLSTM) network that can automatically classify new complaint texts into the appropriate topic category. This approach not only validates our topic modeling but also enables real-time classification.

Data Preparation for Deep Learning

Text Tokenization and Sequence Preparation

The cleaned complaint texts are further preprocessed for deep learning:

- **Tokenization:** We used TensorFlow's Tokenizer to convert texts into sequences of integers, limiting the vocabulary to a maximum number of words (e.g., 10,000 or 25,000).

- **Padding:** To ensure uniform input size, sequences are padded to a fixed maximum length.

- **Label Encoding:** The topic labels (derived from NMF) are encoded as categorical variables using Label Encoder and then converted to one-hot vectors suitable for multiclass classification.

Train-Test Split

We divided the dataset into training and testing sets to validate model performance. This split ensures that the model is evaluated on unseen data to gauge its generalizability.

BiLSTM Model Architecture

We constructed a deep learning model with the following layers:

- **Embedding Layer:** Transforms token sequences into dense vector representations.

- **Bidirectional LSTM Layer:** Captures contextual information in both directions (past and future) to understand the text better.

- **Global Average Pooling:** Aggregates sequence outputs into a fixed-length vector.

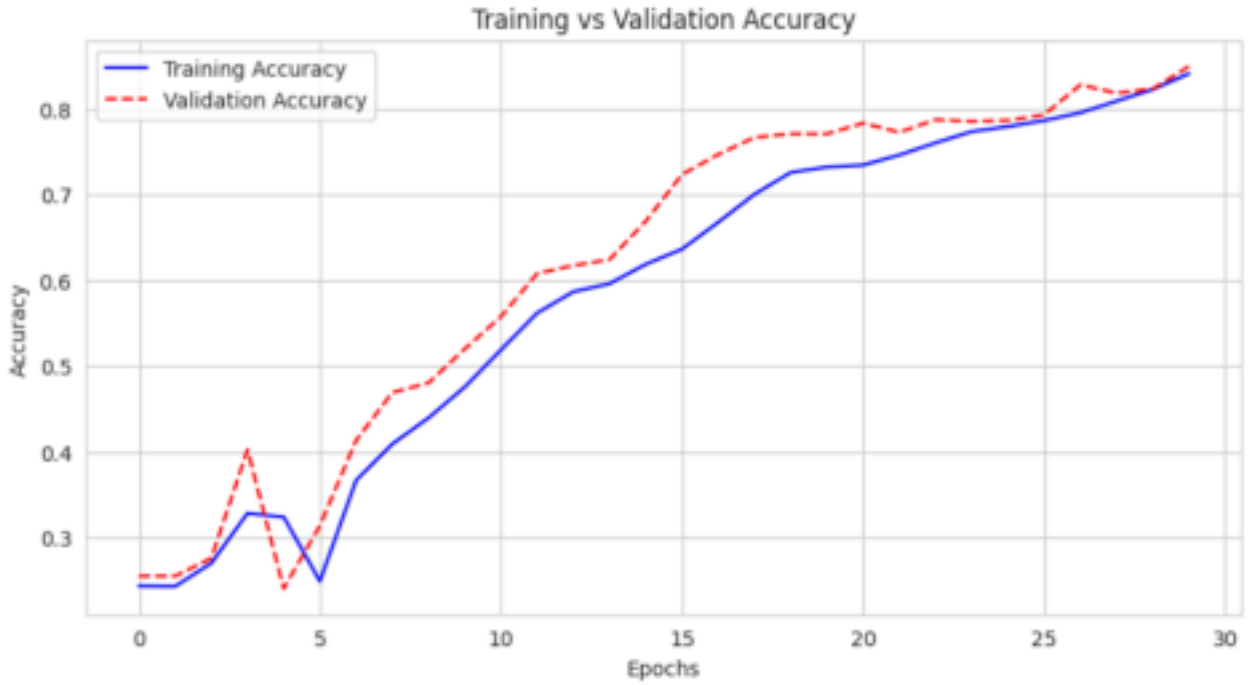
- **Dense Layers:** Several fully connected layers with ReLU activation learn complex patterns.
- **Dropout Layer:** Prevents overfitting by randomly disabling neurons during training.

- **Output Layer:** A softmax activation layer provides the probability distribution for each topic class.

Training and Evaluation

Model Training

The model is compiled with the Adam optimizer and categorical crossentropy loss, suitable for multiclass classification. It is then trained for a fixed number of epochs (e.g., 30 epochs) using a defined batch size. The training process logs metrics such as training and validation accuracy and loss.



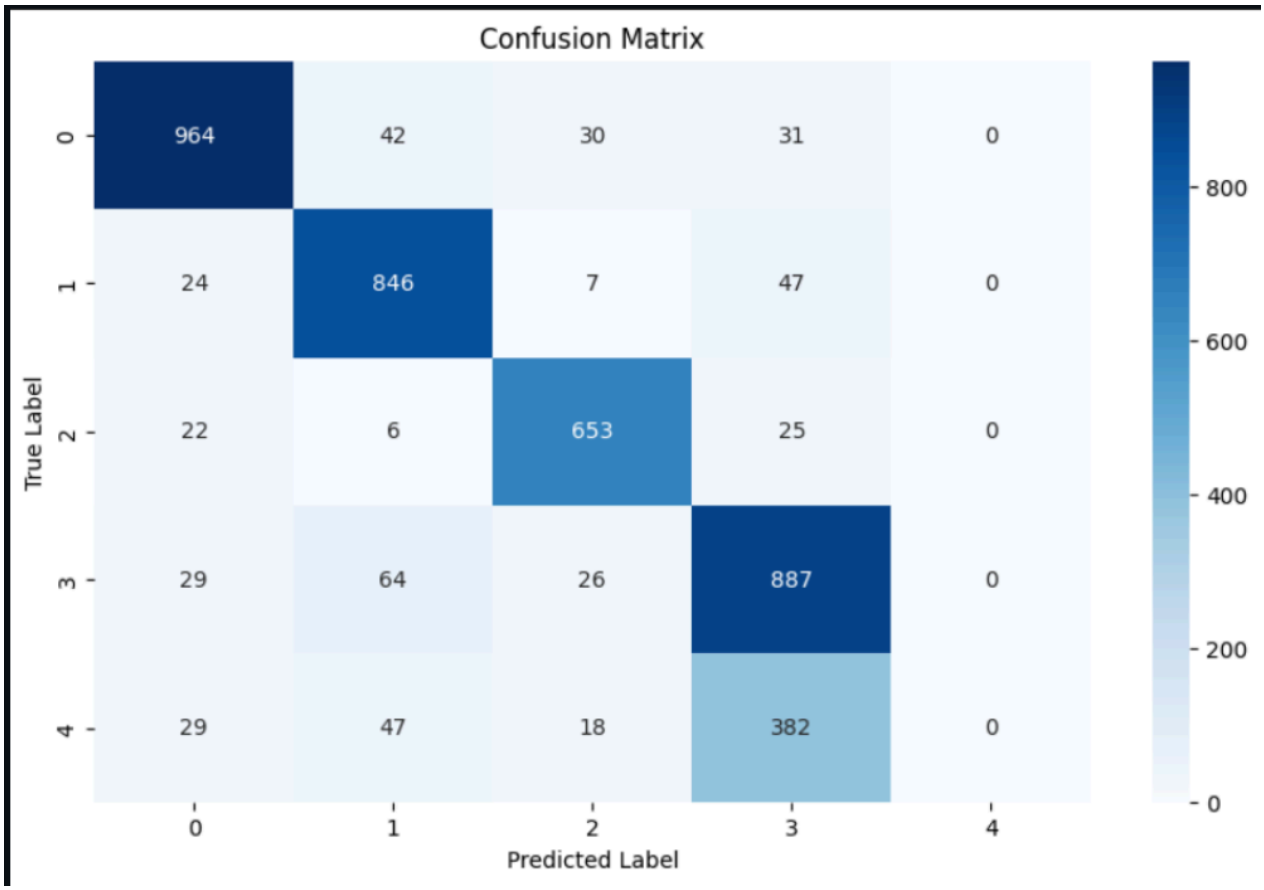
Model Evaluation

After training, the model is evaluated on the test set:

- **Test Accuracy:** The overall accuracy of the model on unseen data is computed.
- **Confusion Matrix:** A confusion matrix is generated to illustrate the performance across different topic classes.
- **Classification Report:** Detailed metrics (precision, recall, and F1 score) are provided for each class.

Classification Report:

	precision	recall	f1-score	support
0	0.90	0.90	0.90	1067
1	0.84	0.92	0.88	924
2	0.89	0.92	0.91	706
3	0.65	0.88	0.75	1006
4	1.00	0.00	0.00	476
accuracy			0.80	4179
macro avg	0.86	0.73	0.69	4179
weighted avg	0.84	0.80	0.76	4179



Conclusion

Topic Modeling Insights

- **Consistency Between LDA and NMF:**
  Both LDA and NMF identify similar topics such as Bank Account Services, Credit Card/Prepaid Card, Mortgages/Loans, and Theft/Dispute Reporting. The "Other" category, although slightly different in size, remains the least dominant.

- **Interpretability:**
  The extraction of top words for each topic allows us to manually assign labels that align with business expectations. This step is critical for translating unsupervised model outputs into actionable insights.

Predictive Modeling Insights

- **Model Performance:**

  The BiLSTM model, trained on tokenized and padded complaint texts, achieves high test

accuracy (approximately 86%). The learning curves indicate steady convergence, and the confusion matrix shows that most topics are well distinguished.

- **Business Impact:**

  An automated ticket classification model based on this approach can significantly reduce manual workload, speed up complaint routing, and ultimately improve customer satisfaction.