# SIADS 696 Milestone II Project Report
# Machine Learning on Duty:
# Solving Consumer Complaints and Improving Satisfaction!

*Chih-Han Yeh (yehch), Yi-Hsin Chien (yihsinc), Weiming Chen (weimingc)*

## INTRODUCTION

Our project aims to tackle the escalating volume of consumer complaints and the complexities involved in addressing them efficiently within the finance area. Specifically, we seek to predict which of the complaints are likely to be disputed by consumers. This predictive capability is crucial for business and the government, as it can significantly help to enhance the efficiency of complaint handling processes, ultimately leading to higher consumer satisfaction. Besides, with the power of data, we intend to unveil concealed patterns or clusters within the narrative of these complaints, which can provide valuable insights into the factors contributing to consumer disputes.

The successful handling of this problem can have significantly positive impacts on different stakeholders. For consumers, it brings a more effective process for resolving their complaints, leading to increased consumer satisfaction. For businesses, they will benefit from a better understanding of which complaints are more likely to escalate into disputes. The understanding can allow them to allocate resources more efficiently and improve their customer service, leading to increased customer loyalty. For the government, they are more capable of identifying potential companies to have higher dispute rates and take necessary actions, ultimately contributing to a fairer marketplace. In summary, we believe the project's success can help to foster a more transparent and efficient consumer complaint resolution ecosystem.

Our motivation to work on this project stems from the increasing importance of consumer complaints resolution in today's business environment. We are inspired by the power of data and dedicated to making a positive impact on both consumers and businesses by developing powerful predictive models that can optimize complaint handling processes. Ultimately, our motivation is driven by the desire to contribute to a more efficient, transparent, and data-driven business environment.

Summary of project’s supervised and unsupervised methods and novel contributions:
In this project, we employed six distinct supervised learning algorithms, including Logistics Regression, Naïve Bayesian, LightGBM, XGBoost, CatBoost, and Artificial Neural Network (ANN), to tackle the task of predicting potential customer disputes based on customer complaint data. We employed a 5-fold cross validation to evaluate the model performance. We then utilized the top-performing model determined by f1-score, the CatBoost, to do further hyperparameter tuning, feature ablation analysis, sensitivity analysis, and failure analysis.

For unsupervised methods, we employed various techniques. We include Latent Dirichlet Allocation (LDA) for topic modeling of consumer complaints and K-means clustering for grouping companies based on their complaints in this report. LDA was used to identify the major consumer complaint topics, the concerns or issues that consumers have. K-means clustering helped group companies with similar operational issues.

The novel contributions of this project lie in the dual approach we applied, which combines both supervised and unsupervised methods. While related projects primarily focused on supervised learning methods for customer complaints analysis, our project takes a more comprehensive approach by integrating both methodologies to investigate into a deeper understanding of the consumer complaint. Furthermore, our project incorporated advanced feature engineering, such as company complaints and disputes within the past 90 days and the dispute-to-complaint ratio, to enhance the model’s predictive capabilities. These novel dimensions in addressing customer complaints represent unique contributions to the field.

Main findings for supervised and unsupervised learning:
In our project, we found that CatBoost outperformed other algorithms, achieving an F1-score of 0.82. This model demonstrated a good performance of predicting potential customer disputes. The analysis also revealed the sensitivity of certain hyperparameter settings, and the positive impact of SMOTE in

addressing class imbalances. For unsupervised learning aspects, we identified five major consumer complaint topics, providing insights into the main area of concerns among customers. Besides, clustering of companies using K-means based on TF-IDF representations of complaints helps to reveal distinct clusters with similar complaint patterns. This approach enables targeted actions for each company cluster.

In summary, the project's findings enhance the understanding of customer complaints and provide valuable insights for addressing potential disputes, thereby improving customer satisfaction.

## RELATED WORKS

A close example is Rahul Rao's master thesis, “Prediction of Complaints through Machine Learning and Story Modeling.[1]”(Rao, 2018). In this thesis, CSS Insurance Inc. sought to enhance customer satisfaction rankings by training a supervised learning model to forecast customer complaints based on prior interactions. However, there exists a notable distinction between our undertaking and this example. Our project intends to employ a dual approach encompassing supervised and unsupervised methods, while in "Prediction of Complaints through Machine Learning and Story Modeling.", they primarily examined whether the sequential or random handling of interactions yielded divergent results.

Another study that is close to our project is “Analysis of Customer Complaints Data using Data Mining Techniques [2]” (Ghazzawi & Alharbi, 2019, 62-69). This project focuses on analyzing customer complaints data from Metropolitan Transportation Authority with supervised classification techniques to help companies to improve quality of services and identify factors that may contribute to customer satisfaction outcome. While our project shares the main objective of analyzing customer complaints, we take a broader approach by developing predictive models to predict which complaints are likely to escalate into disputes. This predictive capability adds a proactive dimension to addressing higher levels of customer satisfaction issues, which is not explicitly addressed in the existing work.

The third close example is “Consumer Complaints Classification Using Machine Learning & Deep Learning [3]” (Naik & T, 2023). This project aims to develop an automatic financial complaint classification system using supervised Machine Learning and Deep Learning techniques to handle customer complaints by routing them to the appropriate department. While our project shares the same goal of automating complaint handling, our approach distinguishes itself through the integration of advanced feature engineering, incorporating factors such as company complaints and disputes within the past 90 days and the dispute-to-complaint ratio, which can augment the predictive capabilities of our models. Additionally, we harness unsupervised learning techniques to enhance our understanding of the factors contributing to consumer disputes.

## DATA SOURCE[4]

Our dataset is sourced from the CFPB Consumer Complaint Database, available for download at the link: https://drive.google.com/file/d/132-ovmSDerlIcbTZCIH-dmcKjuYWxobo/view?usp=sharing.

This dataset, presented in a CSV format, spans from December 1, 2011, to September 22, 2023, and encapsulates an impressive 4,028,530 entries, roughly amounting to 2.57 GB. The pivotal variables encompassed include Product, Sub-product, Issue, Sub-issue, Consumer complaint narrative, Company public response, Company, State, Zip code, Company response to Consumer, and Consumer disputed. An in-depth data cleansing approach was adopted to harness this data for predicting potential consumer disputes. From the initial 4 million rows, we sieved out rows with missing values in 'narrative' and 'dispute,' narrowing it down to 160,000 entries. Subsequent optimization involved pruning several columns, including 'date received' and 'date sent to the company,' 'consumer complaint narrative' and 'narrative_clean,' and 'complaint ID.' Columns like 'timely response?', 'consumer consent provided?', 'submitted via,' 'zip code,' and 'consumer disputed?' were also eliminated for reasons ranging from skewed distribution, redundancy, and lack of variability to model complexity concerns. This data refinement accentuates our commitment to deriving precise and actionable insights.

## FEATURE ENGINEERING

In our pursuit to hone our dataset's predictive power, we embarked on methodical refinements. Recognizing the pitfalls of using aggregate complaint data, we constructed salient features to sidestep potential data leakage. Specifically, we tallied company complaints and disputes over the most recent 90 days and evaluated the dispute-to-complaint ratio, serving as a proxy of a company’s recent performance indicator, hypothesizing that it could foreshadow future dispute trends. We gauged the length of complaint narratives, surmising that more detailed accounts might suggest a heightened risk of disputes. Additionally, we examined the time lag between initiating a complaint and the company's ensuing response, positing that longer waits might amplify dispute chances.

For the categorical features in our dataset, we applied One-Hot Encoding, transforming them into a format more suitable for our machine learning algorithms while preserving the intrinsic information. Anchoring these features in temporally relevant historical data was essential, reflecting a company's performance trajectory and precluding data leakage. Evaluations showed standalone 5,000-dimension TF-IDF features outperformed Word2Vec[5] or combined models. However, high dimensionality caused computational burdens. To balance performance and efficiency, we transformed TF-IDF narratives into mean TF-IDF weighted Word2Vec vectors, retaining TF-IDF's predictive power while reducing the feature space. This TF-IDF weighted Word2Vec approach leveraged TF-IDF's strong performance with Word2Vec's computational savings.

We refined the dataset's columns for utmost clarity and relevance following our feature engineering. Through intensive examination, columns with robust predictive significance were retained, whereas those seen as superfluous, intricate, or devoid of valuable variability were excluded. This rigorous approach, detailed further in the "Data source" segment, underscores our steadfast dedication to accuracy, affirming the trustworthiness and effectiveness of our developed models. The final features included for supervised and unsupervised learning are listed in **Table 1**.

**Table 1. Features Details** [6]

| Column Name | Description | Data Type | Data Source |
|---|---|---|---|
| Product | The type of product the consumer identified in the complaint. | Categorical (one-hot encoded) | From Complaints dataset |
| Sub-Product | The sub-product the consumer identified in the complaint. | Categorical (one-hot encoded) | From Complaints dataset |
| Issue | The issue the consumer identified in the complaint. | Categorical (one-hot encoded) | From Complaints dataset |
| Sub-issue | The sub-issue the consumer identified in the complaint. | Categorical (one-hot encoded) | From Complaints dataset |
| Company public response | The company’s optional, public-facing response to a consumer’s complaint. | Categorical (one-hot encoded) | From Complaints dataset |
| Company | The complaint is about this company. | Categorical (one-hot encoded) | From Complaints dataset |
| State | The state of the mailing address provided by the consumer. | Categorical (one-hot encoded) | From Complaints dataset |

| | | | |
|---|---|---|---|
| Tags | Data that supports easier searching and sorting of complaints submitted by or on behalf of consumers. | Categorical (one-hot encoded) | From Complaints dataset |
| Company response to consumer | This is how the company responded. | Categorical (one-hot encoded) | From Complaints dataset |
| Days_between | Days between the date received and the date sent to the company | Numerical | Engineered from Complaints dataset |
| Narrative_word_count | Word count of each complaint narrative | Numerical | Engineered from Complaints dataset |
| Disputed_count_last_90_days | Number of the company's last 90 days' disputes | Numerical | Engineered from Complaints dataset |
| Complaints_last_90_days | Number of the company's last 90 days' complaints | Numerical | Engineered from Complaints dataset |
| Disputed_ratio_last_90_days | The ratio of the company's last 90 days disputes to the company's last 90 days total complaints | Numerical | Engineered from Complaints dataset |
| Word2vec features | 300 features generated from word2vec on complaint narratives | Categorical (one-hot encoded) | Engineered from Complaints dataset |

## PART A. SUPERVISED LEARNING

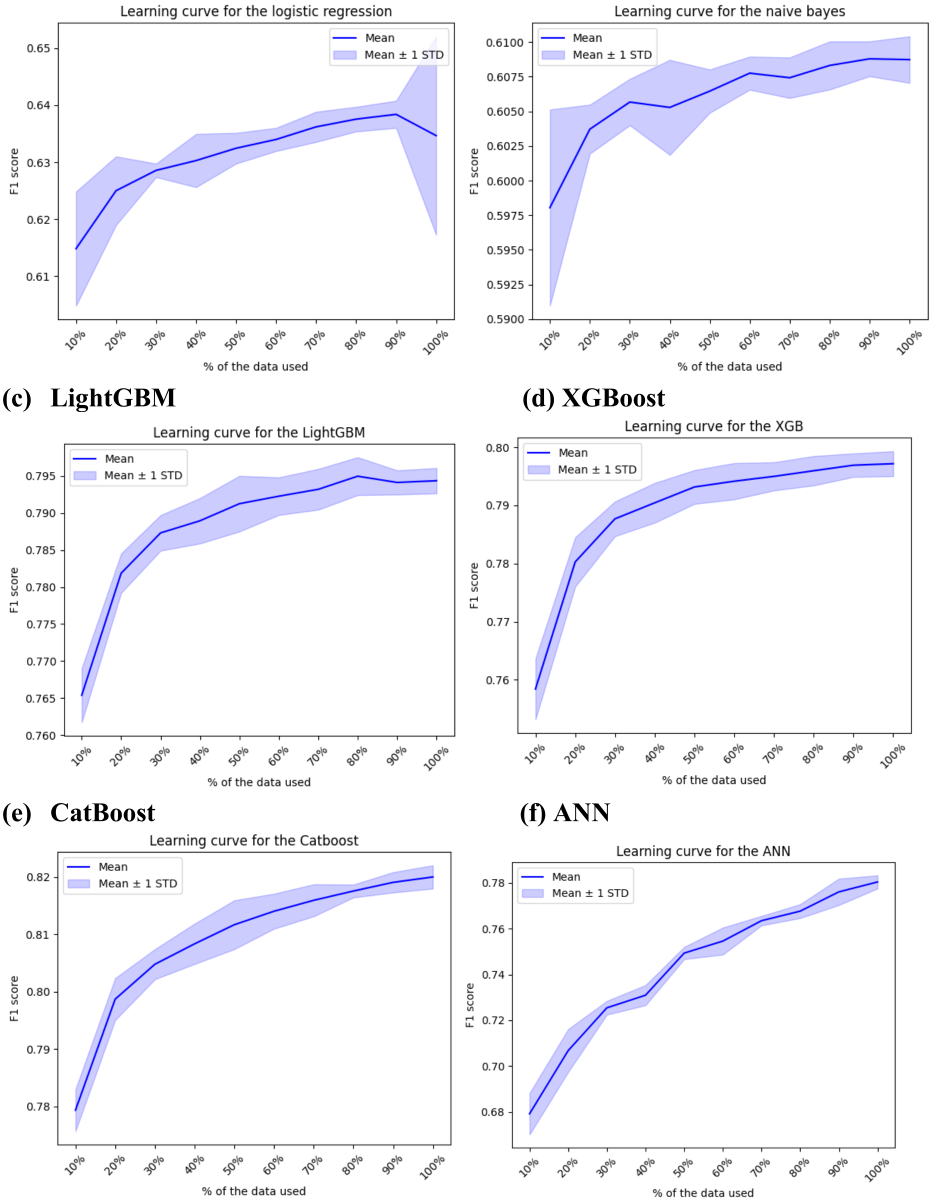
### METHODS DESCRIPTION

In our modeling endeavor, we harnessed six distinct supervised learning algorithms to develop predictive models. These algorithms spanned:

1. **Logistic Regression**
2. **Naive Bayes**
3. **Light GBM**
4. **XGBoost**
5. **CatBoost**
6. **ANN (Artificial Neural Network)**

Our selection of the optimal model was anchored in considerations of accuracy, efficiency, and the model's learning curves—critical factors given our expansive dataset. We employed a 5-fold cross-validation using the sci-kit-learn library on our training dataset to bolster our model's robustness further and manage computational constraints. Initially, we aimed for a 10-fold cross-validation; however, we adopted the 5-fold approach due to computational limitations, ensuring rigorous training without overburdening our computational resources.

**Figure 1. Learning Curve Plots for Six Selected Supervised Models**



(a) Logistic Regression

(b) Naive Bayesian

(c) LightGBM

(d) XGBoost

(e) CatBoost

(f) ANN

Our initial approach to model selection was grounded in the analysis of learning curves. By varying the portion of the dataset and monitoring the corresponding F1-score, we gauged the ability of different models to generalize as more data became available. This method was instrumental in helping us shortlist the models that demonstrated consistent and superior performance across different dataset sizes.

With our top-performing models identified, the next challenge was to optimize their hyperparameters. For this, we turned to GridSearchCV, a powerful method that explores a predefined grid of hyperparameters. We used a 5-fold cross-validation approach, ensuring we robustly tested the models. Within this grid, we examined a range of learning rates, starting from 0.01 and going up to 0.1 in increments of 0.01. Simultaneously, we considered tree depths from 1 through to 10. This exhaustive search culminated in identifying the best hyperparameters, ensuring our chosen model fits the training data well and has strong generalization capabilities.

## SUPERVISED EVALUATION

For evaluating our models, we zeroed in on the F1-score macro. This metric is especially potent for datasets that might be imbalanced or involve multi-class classification. The F1-score offers a well-rounded evaluation by cohesively integrating precision and recall, proving invaluable when skewed class distributions. In contexts involving multiple classes, the macro F1-score evaluates each class on its merit, ensuring a holistic representation of performance without one class overshadowing the rest. Such a metric becomes instrumental in methodically addressing varied errors, facilitating model evaluation and optimization, and counteracting potential biases from imbalances.

The learning curve plots indicate that Catboost, XGB, LightGBM, and ANN outperform Naive Bayesian and Logistic Regression (**Figure 1.**). Interestingly, the logistic regression model peaks in the F1-score at 90% data utilization, only to dip with full data use (**Table 2.**). Their scores for Naive Bayesian, LightGBM, and XGB stabilize after being provided with a certain amount of data, suggesting minimal improvements with additional data. Catboost and ANN are the most promising for subsequent supervised tasks. However, a notable distinction is the processing time: ANN requires about 3 hours for the same amount of data that Catboost processes in roughly 1 hour. Additionally, Catboost attains an F1-score of 0.82, superior to ANN's 0.78. Given these considerations, our team will adopt Catboost as our primary model for the supervised task. We'll further fine-tune this choice through sensitivity testing, hyperparameter tuning, and failure analysis.

**Table 2. F1-score at 90% data utilization**

| Model | Logistic Regression | Naïve Bayesian | LightGBM | XGBoost | CatBoost | ANN |
|---|---|---|---|---|---|---|
| F1-Score | 0.638±0.002 | 0.609±0.001 | 0.794±0.002 | 0.797±0.002 | 0.819±0.002 | 0.776±0.006 |

- **Deeper Analysis of CatBoost Model**

➢ **Feature Importance and Ablation Analysis**

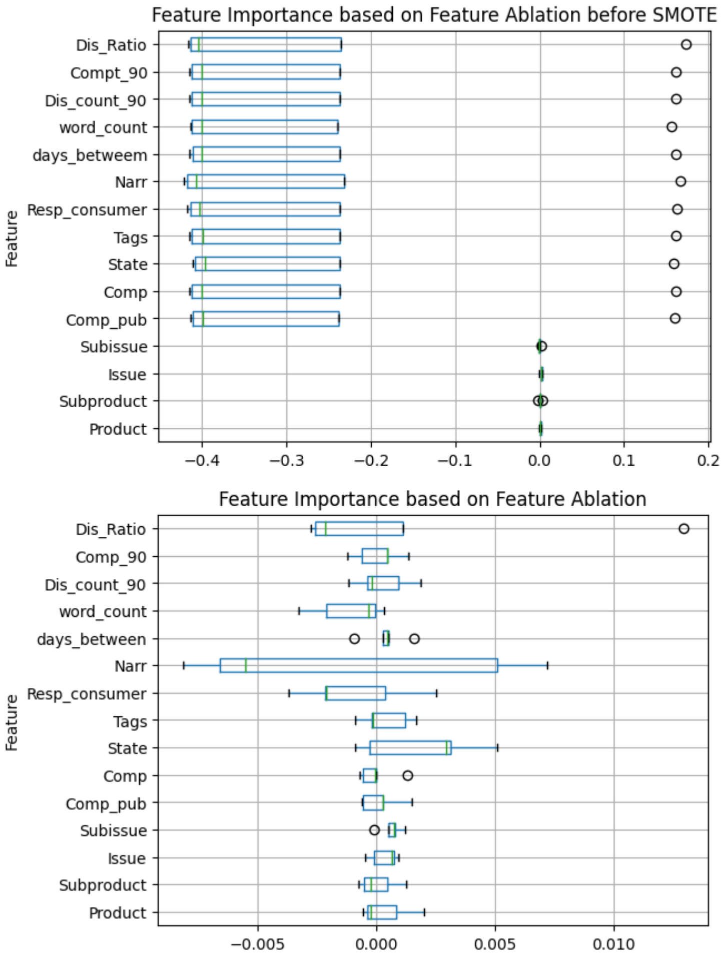
1. **Model with All Features ("Full")**: The "Full" model using all features gives the reference performance. Upon applying SMOTE, there's a significant increase in performance across all folds. This indicates that SMOTE effectively addresses class imbalance and enhances the model's predictive accuracy.
2. **Feature Ablation Analysis**: The performance change when each feature is removed in the pre-SMOTE condition, compared to the full feature set, tells us about the feature's importance:
    a. **"Comp_pub," "Comp," "State," "Tags," "Resp_consumer," "Narr," "days_between," "word_count," "Dis_count_90", "Comp_90",** and **"Dis_Ratio"**: Removing any of these features substantially increase the performance in the second to fifth folds, while slightly decreasing in the first fold. This implies that the choice of features is sensitive to the class distribution and can greatly influence the model performance.
    b. **"Product," "Subproduct", "Issue",** and **"Subissue"**: The model performance does not vary drastically when these features are removed. While they provide some information, they may not be the most critical drivers for predictive power.
3. **Impact of SMOTE**: After SMOTE's application, it seems to have a consistently positive effect on model performance across almost all feature combinations:
    a. Post-SMOTE models register improved performance across folds, hinting that SMOTE benefits the data distribution in these folds.
    b. There's a uniform increase in performance across most folds post-SMOTE, suggesting that it's addressing underlying class imbalances effectively.

**Summary**:

- **Influential Features**: "Comp_pub" and related features like "Narr" and "Resp_consumer" appear to be crucial. Their removal tends to change model performance substantially in the pre-SMOTE condition, implying potential imbalance could have significant influence for predictions.
- **Effect of SMOTE**:

- ○ After applying SMOTE, it's observed that all features have a relatively uniform impact on performance, and there is a significant reduction in the variation of cross-validation scores. We then conclude that after SMOTE, each feature's importance becomes more consistent in terms of their impact on performance.
- ○ SMOTE consistently improves model performance across the board, suggesting it effectively handles class imbalances in the initial dataset. Given its universal positive effect, we decide that it is a worthy inclusion for this dataset.

**Figure 2. Feature Importance and Ablation Analysis:This plot shows, when removing the feature, how the model F1 score varied accordingly.**



Feature Importance based on Feature Ablation before SMOTE

Feature Importance based on Feature Ablation

## ➢ <u>Sensitivity Analysis</u> (Figure 3.)

### ❖ Depth Sensitivity

The model's performance is affected by tree depth. A depth of 1 yields moderate results with scores between 0.6772 and 0.7099. As the depth is increased to 3, there's a noticeable improvement with scores from 0.7349 to 0.7588. The performance remains somewhat stable at a depth of 5, suggesting diminishing returns with scores between 0.7431 and 0.7619. A depth of 7 sees varied outcomes, with scores ranging from 0.7466 to a peak of 0.8209, hinting that other factors, like learning rate, might influence performance. However, at depth 9, scores marginally drop between 0.7401 and 0.7424, potentially suggesting overfitting.

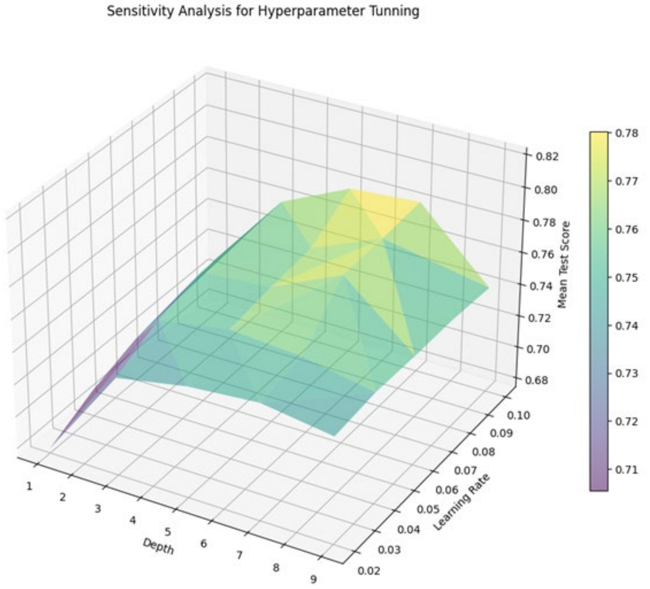
### ❖ Learning Rate Sensitivity

The model's response to the learning rate could be more straightforward. At a depth of 1, there's an ascending trend in performance from a learning rate of 0.02 (score of 0.6772) to 0.1 (score of 0.7099). At depth 3, the score peaks at 0.7588 with a learning rate 0.1. Yet, for deeper trees, such as depths 5 and 7, the ideal learning rate fluctuates, with depth 5 reaching its best score of 0.7619 at a learning rate of 0.08, while depth 7 achieves high scores at various learning rates.

❖ Summary

The CatBoost model's performance shows discernible sensitivity to depth and learning rate. Tree depth provides clear improvements up to a certain threshold, after which the benefits level out or slightly regress. The learning rate's influence varies across depths, indicating its effects are interdependent on the tree depth. The model's peak performance is attained with a depth of 7 and a learning rate of 0.1, though other configurations also deliver competitive outcomes.

**Figure 3. Sensitivity Analysis on Learning Rate and Depth**



➢ <u>**Tradeoff**</u>

The evaluation results reveal a tradeoff between model performance and computational efficiency. Converting the TF-IDF narratives into mean TF-IDF weighted word2vec vectors enabled faster execution compared to using the full 5000-dimensional TF-IDF features. This speed advantage came at a small cost in accuracy - the TF-IDF model achieved an F1-score of 0.86, while the streamlined model scored 0.82. Nonetheless, the computational efficiency granted broader analytical capabilities, particularly extensive cross-validation and sensitivity analysis. Overall, the model optimization prioritized computational speed and analytical breadth over maximizing F1-score, illustrating the inherent tradeoffs between performance metrics, efficiency, and analysis capabilities.

➢ <u>**Failure Analysis**</u>

From **Figure 4**, we see 24,219 True Positives (Actual 1, Predicted 1), 28,536 True Negatives (Actual 0, Predicted 0), 3,372 False Positives (Actual 0, Predicted 1), and 7,825 False Negatives (Actual 1, Predicted 0).

**Categories of Failure:**

There are three distinct categories of failure identifiable from the matrix:

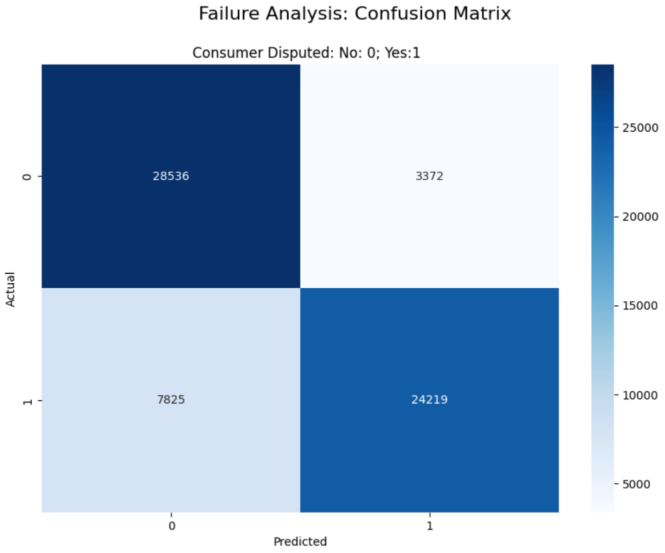
1. **False Negatives (Type II Error)**: With 7,825 instances where class 1 was misclassified as class 0, there's evidence that the model might struggle to distinguish certain characteristics of class 1 or that it's biased towards predicting class 0.
2. **False Positives (Type I Error)**: 3,372 instances were wrongly predicted as class 1 when they were class 0, suggesting the model might be too sensitive to certain features or overfitting some patterns in the training data.
3. **Difference in Magnitude of Errors**: The disparity between Type I and Type II errors indicates that the model might have difficulty consistently classifying Class 1 due to potential feature variability.

**How to Improve:**

1. **Feature Engineering**: Revisiting and enhancing the features can aid in more accurate classification.

2. **Adjusting Model Complexity**: Tweaking the model's complexity can help it better capture data intricacies or prevent overfitting.
3. **Regularization**: Implementing regularization techniques can diminish overfitting, especially if the model is over-sensitive to training data noise.
4. **Ensemble Methods**: Employing ensemble approaches, like bagging or boosting, can amalgamate the strengths of multiple models, refining overall performance.

**Figure 4. Failure Analysis: Confusion Matrix**



## PART B. UNSUPERVISED LEARNING

For unsupervised methods, we employed various techniques, including Latent Dirichlet Allocation (LDA) topic modeling, K-means clustering, Association Rule Mining, and Finding Similar Complaints with Word2Vec and Cosine Similarity. All the methods provide valuable insights, and we include LDA for topic modeling of consumer complaints and K-means clustering for grouping companies based on their complaints in this report, and details of the other unsupervised techniques and findings can be found in the Appendix due to page limits.

### METHODS DESCRIPTION

1. **Learning Method 1: LDA Topic Modeling on Consumer Complaints Narrative**
   LDA topic modeling is a probabilistic method, which assumes that documents are mixtures of topics and that topics have a probability distribution over words.

   - **Workflow & Feature Representation**
     - Selected only those rows where the 'Consumer complaint narrative' was available.
     - Preprocessed the narratives by:
       - Converting to lowercase.
       - Removing masking patterns like XX, XXX, XXXX, …
       - Removing punctuation, numbers, and stopwords.
       - Removing extra spaces.
     - Converted the cleaned narratives into a bag-of-words representation.
     - Used the LDA method to identify topics within these narratives.
     - Visualized the top topics using word clouds.
   - **Justification**
     LDA is a popular probabilistic technique for topic modeling which helps in understanding the hidden thematic structure in a large collection of texts. Here in our project, it helps us in understanding the major concerns or issues that consumers have.

- **Hyperparameter tuning and Exploration**
  For LDA topic modeling, the number of topics is a key hyperparameter that can greatly impact the resulting topics. The coherence score metric was used to tune the number of topics. Models were trained with different numbers of topics (5, 10, 15, 20). The one with the highest coherence score was selected, which was 5 topics.

2. **Learning Method 2: Clustering Companies based on Complaints with TF-IDF and K-means**
   K-means clustering is an instance-based method in which we are directly working with individual instances (complaints or companies) to discover patterns or clusters.

   - **Workflow & Feature Representation**
     - Combined all narratives for each company into a single text.
     - Converted these combined narratives into a TF-IDF matrix.
     - Used the elbow method to determine the optimal number of clusters.
     - Used K-means clustering to cluster the companies based on their TF-IDF representations.
   - **Justification**
     K-means clustering helps us in grouping data points (in this case, companies) that have similar features. By using TF-IDF representations, we are clustering companies that receive similar types of complaints. This helps in identifying companies with similar operational issues. It enables targeted actions for each company cluster instead of a one-size-fits-all approach. It focuses improvement efforts on clusters with the most severe or frequent complaint types.
   - **Hyperparameter tuning and Exploration**
     For K-Means clustering, the elbow method was used to tune the number of clusters k. K-Means models were trained with different k values ranging from 1 to 30 clusters. The inertia was plotted for each k, and the "elbow point" was selected where inertia did not decrease much with more clusters. This led to selecting 14 clusters.

## UNSUPERVISED EVALUATION

➢ **Evaluation Metrics and Justification**

1. **LDA Topic Modeling**
   - Coherence score: Evaluates semantic interpretability of topics. It helps tune the number of topics.
   - Topic distributions: Useful for exploring relative prevalence of topics.
   - Justification: Coherence is a standard metric for topic model evaluation. Topic distributions give insights into data.

2. **K-Means Clustering**
   - Inertia: Sum of squared distances between data points and cluster centroids. Lower is better.
   - Cluster counts: Number of data points assigned to each cluster. Indicates relative sizes. Cluster counts give insights into sizes and to check if there is any outlier.
   - Justification: Inertia helps determine the optimal number of clusters. Inertia measures compactness of clustering. The Elbow Method finds the "sweet spot" where adding more clusters does not significantly improve inertia. It balances between too few and too many clusters.

➢ **Overall summary of results**

The summary of results that compares the best model from each family we used is shown in Table 3.

**Table 3. Unsupervised Learning results**

| Method | Model Details | Key Insight |
|---|---|---|
| Topic Modeling (LDA) | 5 topics, tuned by coherence score | Main consumer complaint topics identified with human-in-the-loop check (some topics mined are: credit report, debt collection, loan etc) |
| Clustering (K-Means) | 10 clusters, tuned by elbow method | Companies clustered by complaint patterns. E.g Cluster 1 are mostly banks related, while Cluster 5 covers a large portion of automotive companies. |

## UNSUPERVISED EVALUATION

### 1. LDA Topic Modeling

**Figure 5** illustrates the relationship between the number of topics chosen for LDA topic modeling and the coherence score for those topics. There's a noticeable initial increase in coherence as the number of topics goes up, peaking at 5 topics, followed by a subsequent decline. This suggests that adding more topics beyond 6 results in less coherent or overlapping topics.

**Figure 5. Coherence Scores for Different Numbers of Topics**

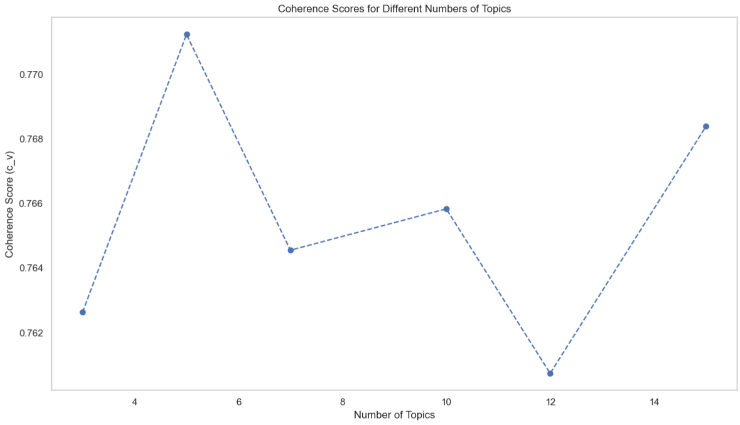
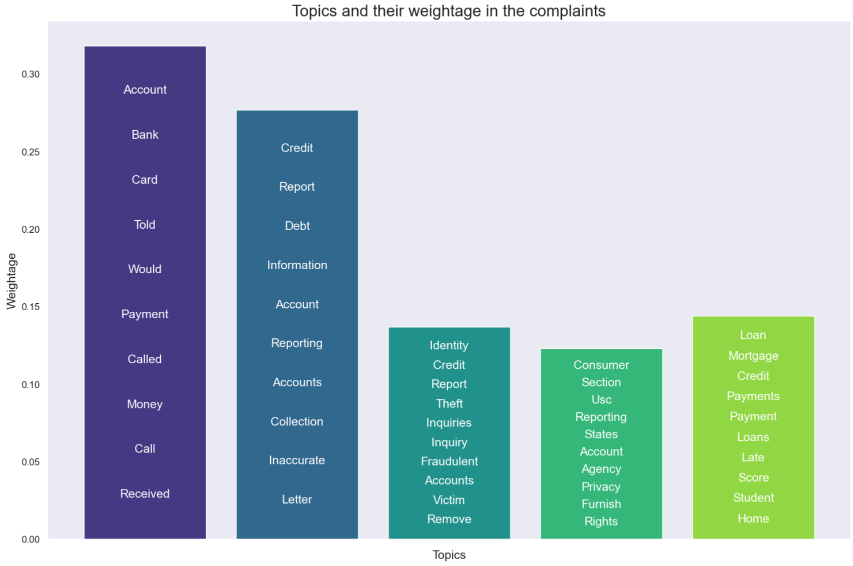


**Figure 6.** presents the distribution of topics and their corresponding weightage in the dataset of complaints. The terms within each topic offer insights into the nature of clusters within the complaints.

**Figure 6. Topics and their weightage in the companies**



### 2. K-Means Clustering

**Figure 7.** presents an "elbow plot" showing the relationship between the number of clusters (k) and the corresponding inertia. The key observation from such plots is to find the "elbow point", which is where the rate of decrease in inertia sharply changes, we think the sweet spot is between 10-20 in this case.

**Figure 7. Inertia vs. k**

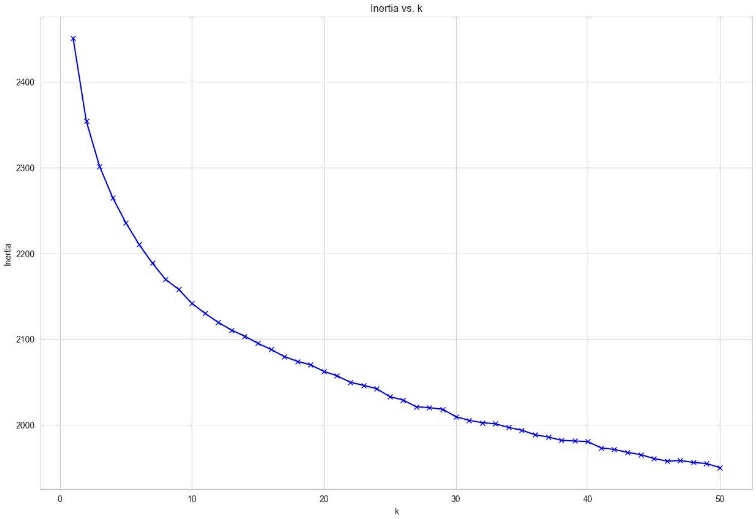
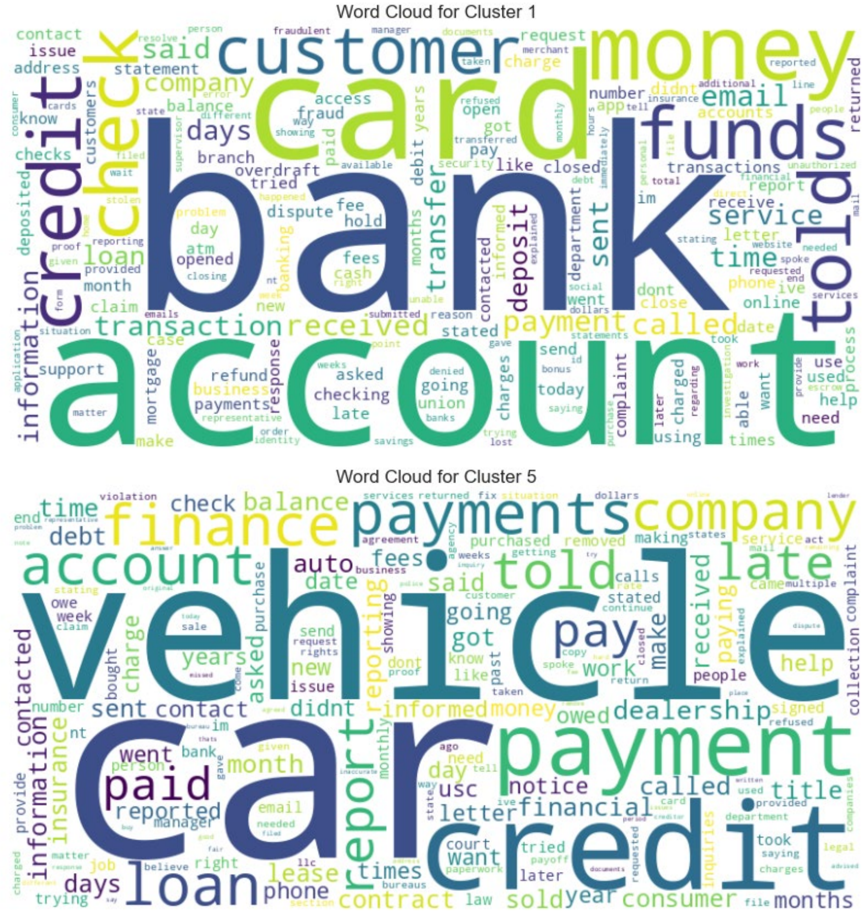


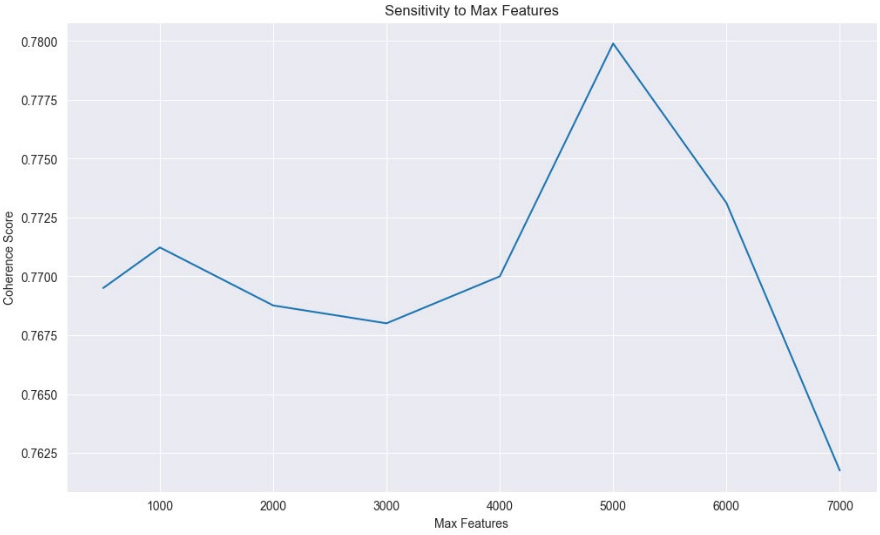
**Figure 8.** provides word clouds for two distinct clusters, Cluster 1 and Cluster 5. It helps in alignment of the meaningfulness of the clusters with human perception.

**Figure 8. WordCloud for Cluster 1 and Cluster 5**



## SENSITIVITY ANALYSIS

**Figure 9. Sensitivity to Max Features**



We will do the sensitivity analysis on LDA topic modeling. As seen in **Figure 9.**, the relationship between the "max_features" parameter of a TF-IDF vectorizer and the coherence score related to LDA topic modeling.

- General Trend: The coherence score seems to vary as the "max_features" parameter changes. This indicates that the number of features considered by the vectorizer affects the quality or coherence of the topics generated by LDA.
- Initial Stability: For values of "max_features" from about 1000 to around 4000, the coherence score remains relatively stable with minor fluctuations. This suggests that within this range, the topics' quality doesn't drastically change.
- Peak Coherence: Around "max_features" value of 5000, there is a sharp peak in the coherence score. This indicates that the LDA model achieved the best topic coherence with a vectorizer set to consider 5000 features. It suggests that this might be an optimal value for modeling, as the topics derived from the data are most coherent (or meaningful) with this setting.
- Decrease after Peak: there's a rapid decrease in coherence as "max_features" increases. This suggests that after a certain point, adding more features (terms) to the model starts to degrade the quality of the topics. It might indicate noise being introduced, which can dilute the significance of more relevant terms.

Sensitivity: The model appears to be most sensitive to the "max_features" parameter around the 5000 mark. Before and after this point, the results either remain stable or degrade rapidly. It underscores the importance of parameter tuning in achieving optimal results.

## DISCUSSION

### PART A: Supervised Learning

The surprising aspect of our results lies in the great influence of addressing data imbalance on model performance. Before applying SMOTE (Synthetic Minority Oversampling Technique), each feature had varying degrees of influence on our models, and the variations of cross-validation scores are large. Upon applying SMOTE, a significant increase in predictive power is observed across all cross-validation folds. This indicates that SMOTE can efficiently address class imbalance and improve model performance across the board. Additionally, after the application of SMOTE, it becomes evident that all features show a relatively uniform impact on model performance, leading to a notable reduction in the variation of cross-validation scores. We learn that the choice of features is highly sensitive to the class distribution and can greatly affect model performance. This phenomenon highlights the importance of engineer features while considering the inherent class distribution within the dataset carefully.

The primary challenges we encountered in this project lie in dealing with the large dataset and balancing feature engineering within computational limits. The computational limits make it challenging to conduct extensive feature engineering and explore the full range of hyperparameters. To address the challenges, we implemented strategies such as using HalvingSearchCV instead of GridSearchCV to save time and enhance efficiency. While this approach allows us to manage computational limits, it also meant that we might not have identified the best and optimized hyperparameters. Additionally, due to time and resource limitations, we are forced to focus only on adjusting key hyperparameters such as depth and learning rate, which are crucial for model performance. These show the trade-off between computational resources and hyperparameter tuning.

With more time and resources, we could extend the scope of our solution in several aspects. First, we could explore the impact of different feature engineering techniques such as to the level of sub-product or product granularities as the data is imbalanced itself. This fine-grained approach would potentially give us more specific insights from the data. Additionally, we could conduct a more extensive search for the optimized hyperparameters to further enhance our model performance. Moreover, we also would like to employ a wider range of NLP techniques to enhance the effectiveness of our narrative features. We would also consider the distributed computing approach such as Apache Spark to parallelize data processing and model training. This would enable us to work with larger datasets. Lastly, we could explore ensemble modeling to combine multiple diverse models, which could potentially further enhance predictive performance and mitigate model bias.

### Part B: Unsupervised Learning

We learned that topic modeling is useful for discovering hidden themes in unstructured text data. However, tuning the number of topics is tricky - coherence scores help but some subjective judgment is still needed. Also, data cleaning and preprocessing are critical for text mining. Cleaning the narratives helped the models learn better representations.

There are several surprising findings in our results. First, the LDA topics seemed reasonably coherent - we wouldn't expect it to extract such clear topics from free-form complaint narratives without much tuning. Secondly, some company clusters aligned very closely with business types (e.g., cluster 1 with banks). This demonstrates how unsupervised learning can recover meaningful industry groups.

We encountered some challenges including that performance and memory issues arose when creating the word vectors for all complaints. We had to subsample the dataset. Also, interpreting the clusters required manually inspecting the contents. An objective/automated way to summarize clusters would have been helpful. Lastly, many parameters needed tuning through trial-and-error, which was time consuming. More principled hyperparameter optimization would be better.

With more time and resources, we could make some extensions to our project. First, we could incorporate product category, company, location, and other metadata to refine the analyses and identify trends. Secondly, we could handle larger datasets with distributed computing like Spark to scale up the workflow. Thirdly, we could also apply more advanced NLP like named entity recognition to extract structured facts from complaints as additional features. Additionally, we could try to build a user interface for browsing, querying and recommender systems for the complaints based on topics, associations, similarities, and clusters.

## ETHICAL CONSIDERATIONS

### Part A: Supervised Learning

Bias in the training data presents a significant ethical concern. If certain groups are disproportionately likely to file complaints, or if complaints from specific groups are treated differently due to implicit biases, the training data can inadvertently reflect these biases, culminating in biased models. To mitigate this, it's crucial to regularly audit the training data for any signs of systemic biases. Employing strategies such as dataset rebalancing, leveraging fairness-enhancing interventions, or using adversarial training can help to reduce these biases. Moreover, when automated systems utilize this data to classify or act upon complaints, stakeholders understandably demand clarity in the decision-making process. As a remedy, it's essential to use models that offer interpretability or integrate tools that elucidate the decisions made by intricate models. Utilizing Explainable AI (XAI) tools can be instrumental in this endeavor.

### Part B: Unsupervised Learning

Unsupervised learning, such as clustering, might inadvertently detect patterns in data that disclose sensitive information about certain groups or specific business methodologies. The potential to unintentionally harm individuals or unjustly mar company reputations is a genuine concern. To navigate this, it's vital to meticulously review the outputs of unsupervised models, weighing the ramifications of any discerned patterns. Exercising discretion in how these findings are employed or shared is essential. Another inherent challenge with unsupervised learning is the frequent lack of a 'ground truth' to validate model outputs. This absence can occasionally result in incorrect insights or potential misinterpretations. To counteract this, it's crucial to ensure that the results from unsupervised models are juxtaposed with domain expertise. While the absence of definitive "correct" labels, as found in supervised learning, is felt, the risk of drawing false conclusions can be substantially minimized by closely collaborating with experts who are deeply familiar with the nuances of the complaints and the broader financial sector.

## STATEMENT OF WORK

| **Member** | Chih-Han Yeh | Yi-Hsin Chien | Weiming Chen |
|---|---|---|---|
| **Data Collection and Preprocessing** | Examine working details and assure the quality | Dive deep in the EDA, present concerns, and prompt discussion on solutions. | Setup the structure and workflow for preprocessing task |
| **Supervised Learning Model Development** | Testing models with small amounts of data to get preliminary results. | Final conclude on the supervised models with learning curve, sensitivity analysis, and failure analysis | Conduct exploratory supervised learning models and guide the directions |
| **Unsupervised Learning** | Conduct evaluation on the unsupervised learning models | Examine working details and assure the quality | Conduct exploratory unsupervised learning models, guide the directions, and final conclude unsupervised tasks |
| **Documentation and Project Report** | Document data collection, preprocessing, and transformations. | Present methodologies and summarize the supervised learning part | Conclude with recommendations for further research and pilot projects before implementation. |
| **Stand-ups** | Video Recordings | Provide Jupyter notebook visualization and narratives | Handle modeling and evaluation |

## REFERENCES

1. Rao, R. (2018, May 24). *Prediction of Complaints through Machine Learning and Story Modeling*. Lucerne University of Applied Sciences and Arts. https://www.zhaw.ch/storage/engineering/institute-zentren/cai/MSc18_Complaint_Prediction_Rao.pdf
2. Ghazzawi, A., & Alharbi, B. (2019). Analysis of Customer Complaints Data using Data Mining Techniques. *Procedia Computer Science*, *163*, 62-69. https://www.sciencedirect.com/science/article/pii/S187705091932126X
3. Naik, P. K., et al. (2023). Consumer Complaints Classification Using Machine Learning & Deep Learning. *International Research Journal on Advanced Science Hub*, *05*(05S). https://rspsciencehub.com/article_23795_3d7d1d4dfc63d072724eafdb92a5bcbf.pdf
4. Consumer complaint database. Consumer Financial Protection Bureau. (n.d.). https://www.consumerfinance.gov/data-research/consumer-complaints/
5. Mikolov, T., et al. (2013, July 30). *Google Word2Vec Project*. Google. https://code.google.com/archive/p/word2vec/
6. How we share complaint data. Consumer Financial Protection Bureau. (n.d.-b). https://www.consumerfinance.gov/complaint/data-use

## APPENDIX

In addition to LDA topic modeling and K-means clustering, our team also implemented some other valuable unsupervised learning methods, including (1) Association rule mining, association rules between the product and issue, and also (2) Finding Similar Complaints with Word2Vec and Cosine Similarity. We supplement the details of the two methods as below in the appendix.

### METHODS DESCRIPTION

**(1) Association rule mining, association rules between the product and issue:**
Instance-based Methods: Association rule mining as an Instance-based Method, where we are directly working with individual instances (complaints or companies) to discover patterns or clusters.

Workflow:

- Focused on the 'Product' and 'Issue' columns.
- One-hot encoded the columns to prepare the data for association rule mining.
- Used the Apriori algorithm to find frequent itemsets.
- Generated association rules based on the frequent itemsets.
- Visualized the top association rules using network graphs.

**Justification:**

Association rule mining, especially the Apriori algorithm, is suitable for finding interesting relationships or associations between variables. Here, it helps in understanding which products are associated with which issue, uncovers relationships between products and issues that may not be immediately apparent. It allows organizations to spur investigations into why certain product-issue combinations are frequent.

**(2) Finding Similar Complaints with Word2Vec and Cosine Similarity:**

Workflow:

- Preprocessed the complaint narratives.
- Loaded a pre-trained Google Word2Vec model.
- Computed the average Word2Vec vector for each complaint narrative.
- Project each narrative onto 300D vector space.
- Computed the cosine similarity between these vectors to find similar complaints.

**Justification:**
Word2Vec captures semantic information about words. By computing the average Word2Vec vector for each complaint, we capture the semantic essence of each complaint narrative. Cosine similarity then helps in finding complaints that are semantically similar. The implementation matches new entries to similar past complaints for quick resolution. It reduces manual effort in searching past records for similar cases, thus improves customer satisfaction by providing consistent responses.

## UNSUPERVISED EVALUATION

**(1) Apriori Association Rules**

- Support: Frequency of occurrence of a rule. Filters rare rules.
- Confidence: Reliability of a rule. Filters spurious rules.
- Lift: Interestingness of a rule, ratios between expected and observed support. Prioritizes useful rules.

Justification: Support, confidence and lift are standard metrics for filtering and ranking association rules.

**(2) Word2Vec Similarity**

- Cosine similarity: Measures how similar two mean vectors are based on their orientation. Gives semantic similarity of text.

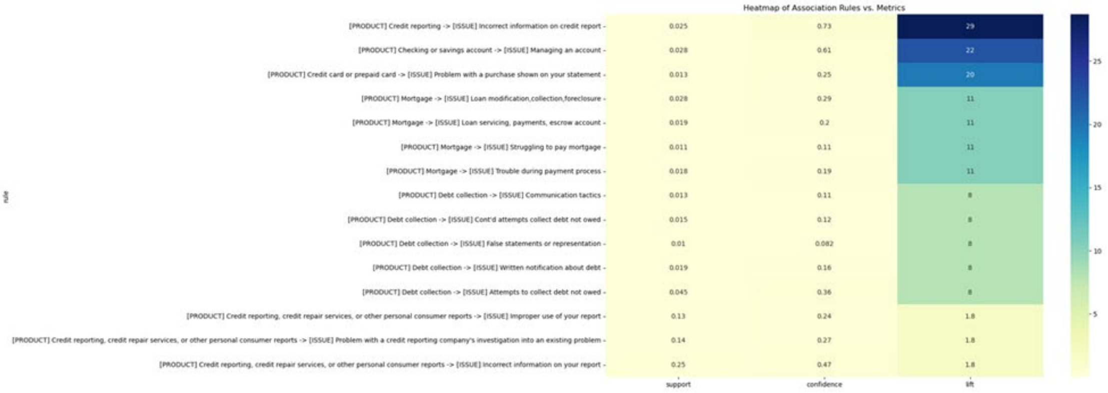
Justification: Cosine similarity is the standard metric used with Word2Vec models for comparing word/document vectors.

## Summary of Results

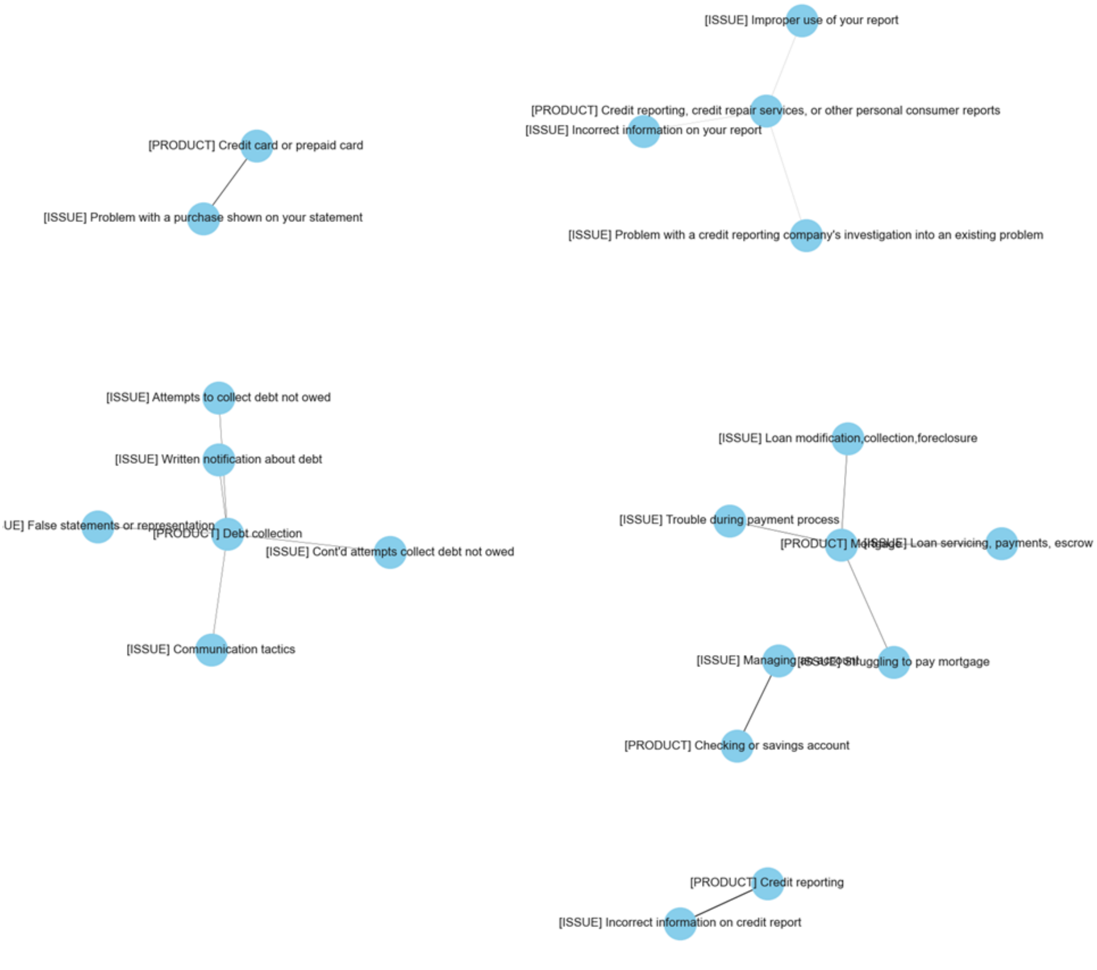
| Method | Model Details | Key Insight |
|---|---|---|
| Association Rule Mining (Apriori) | Lift >= 8 | Strong associations between products and issues uncovered. We focus on strong positive association between two entities. The top ten are ranging from 8 – 28 |
| Document Embedding (Word2Vec) | Google News pre-trained embeddings | Semantic similarity between complaints calculated. Identified narrative meaning with human-in-the-loop check. |

## Visualizations

(1) Association rule

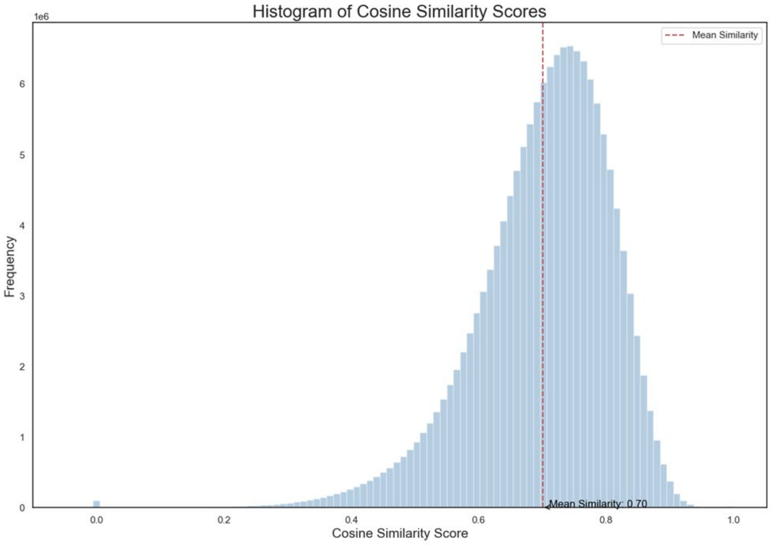


The visualization above presents a heatmap that depicts the association rules between various financial products and the issues related to them. The heatmap's colors correspond to the lift values, with darker shades indicating higher lift values. Lift values represent the likelihood of the association between the product and the issue.

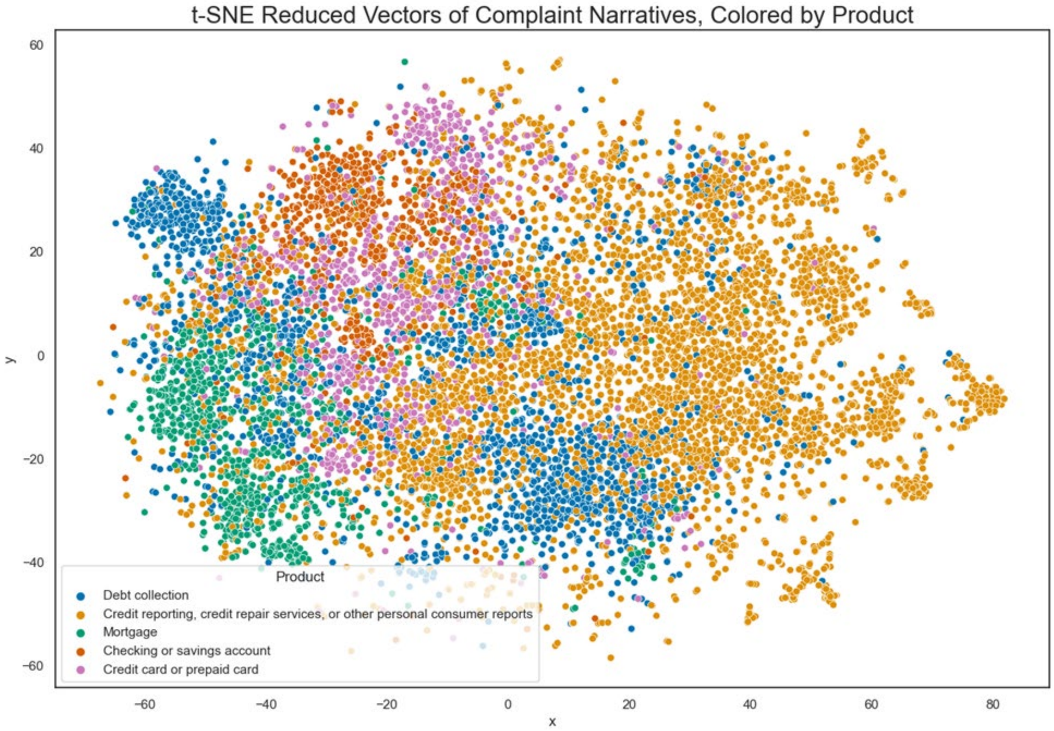

[ISSUE] Improper use of your report
[PRODUCT] Credit reporting, credit repair services, or other personal consumer reports
[ISSUE] Incorrect information on your report
[PRODUCT] Credit card or prepaid card
[ISSUE] Problem with a purchase shown on your statement
[ISSUE] Problem with a credit reporting company's investigation into an existing problem
[ISSUE] Attempts to collect debt not owed
[ISSUE] Written notification about debt
UE] False statements or representation
[PRODUCT] Debt collection
[ISSUE] Cont'd attempts collect debt not owed
[ISSUE] Communication tactics
[ISSUE] Loan modification,collection,foreclosure
[ISSUE] Trouble during payment process
[PRODUCT] Mortgage
[ISSUE] Loan servicing, payments, escrow
[ISSUE] Managing an account
[ISSUE] Struggling to pay mortgage
[PRODUCT] Checking or savings account
[PRODUCT] Credit reporting
[ISSUE] Incorrect information on credit report

The visualization provides a network of associations between various products and the issues associated with them. The boldness of the edge line represents the degree of lift metric. Certain products, such as those related to credit reporting and debt collection, have multiple issues associated with them, highlighting the complexity and range of challenges consumers might face in these areas.

(2) Word2Vec Similarity



The visualization is a histogram that depicts the distribution of Cosine Similarity Scores with every possible pair of complaint narratives. In essence, the histogram shows that the majority of the data has high cosine similarity scores, centering around a mean value of 0.70.



The visualization presents a scatter plot of t-SNE reduced vectors of complaint narratives. These points are colored based on the top 5 products associated with each complaint.

## Discussion

What we learned:

- Association rules are great for finding combinations of items/events that commonly co-occur. Visualizing the rules makes them more interpretable.

- The importance of data cleaning and preprocessing for text mining. Cleaning the narratives helped the models learn better representations.

What surprised us:

- The Word2Vec similarity was able to find very relevant similar complaints even without any labeling or supervision. This shows the power of word embeddings.