**Machine Learning Programming**

**PROG8245 - Winter 2025 - Section 1**

**Sentiment Analysis on Consumer Financial Complaints Using NLP**

**Instructor**: **Islam Mahmoud**

**Submitted on : 17/04/2025**

**GROUP 5**

**Team Members:**

**Daya Meenakshi Bala Subbu (8964370)**

**Abdul Bari Mohammed (8985877)**

**Bismi Poulose (8981107)**

**Table of Contents**

|  |  |  |
| --- | --- | --- |
| **1** | **Introduction** | **3** |
| **2** | **Dataset and Data Collection** | **3** |
| **3** | **Step-by-Step Methodology**  **- Step 1: Data Loading and Cleaning**  **- Step 2: Text Preprocessing**  **- Step 3: Sentiment Analysis using TextBlob**  **- Step 4: Exploratory Data Analysis (EDA)**  **- Step 5: Feature Extraction & Model Comparison** | **4** |
| **4** | **Results & Visualizations** | **9** |
| **5** | **Challenges and Learnings** | **11** |
| **6** | **Demo** | **11** |
| **7** | **Future Work** | **12** |
| **8** | **Conclusion** | **12** |
| **9** | **Appendix** | **12** |
| **10** | **References** | **13** |

**1. Introduction**

Natural Language Processing (NLP) is a powerful domain that focuses on teaching machines to interpret and generate human language. In this project, we apply NLP techniques to **analyze customer sentiment** from financial product complaints.

**Objective**

The goal of this project was to build a sentiment analysis system to classify customer complaints into **positive**, **negative**, or **neutral** sentiments using Natural Language Processing techniques. This was done by:

* Cleaning and preprocessing real-world complaint texts
* Extracting meaningful features using various embeddings
* Comparing performance of traditional ML and deep learning models
* Visualizing the insights and finalizing the best model

**2. Dataset and Data Collection**

These complaints were submitted by users about issues faced with banks, credit cards, loans, mortgages, etc. Our goal was to analyze the sentiment of these complaints (positive, negative, neutral) and build models that can automatically classify new complaint texts.

* **Source:** Kaggle - Consumer Complaints on Financial Products
* **Original Records:** 670,598
* **Used for Analysis:** First 60,000 complaints with valid narratives
* **Key columns:**
  + Consumer complaint narrative → complaint text
  + Product → e.g., Credit Card, Loan
  + Company → e.g., Citibank, Capital One
  + State → U.S. state where complaint originated
  + Submitted via → method of submission (Web, Phone, etc.)

We filtered rows where Consumer complaint narrative is not null and retained the essential columns.

**3. Step-by-Step Methodology**

**Step 1: Data Loading and Cleaning**

* We loaded the raw CSV file and retained rows where the Consumer complaint narrative was not null.
* Renamed the column to text.
* Extracted 5 relevant columns for analysis.
* Computed and saved the complaint text lengths.

A black screen with white text

AI-generated content may be incorrect.

**Step 2: Text Preprocessing**

* Converted all text to lowercase.
* Expanded contractions using contractions library.
* Removed emojis using emoji package.
* Removed punctuation and non-alphabetic characters using regex.
* Tokenized using TreebankWordTokenizer.
* Removed stopwords and applied lemmatization.
* Saved cleaned dataset as final\_cleaned\_60000.csv

A screenshot of a computer

AI-generated content may be incorrect.

**Step 3: Sentiment Analysis using TextBlob**

* Used the polarity score to assign one of three classes: positive, negative, or neutral.
* Saved the results in sentiment\_textblob\_60000.csv.

A screen shot of a computer

AI-generated content may be incorrect.

**Step 4: Exploratory Data Analysis (EDA)**

* Plotted sentiment distribution bar chart.
* Generated bar plots of top states, products, and companies by complaint count.
* Created a word cloud from all cleaned complaints.
* Analyzed sentiment distribution across products and companies.

A screen shot of a computer

AI-generated content may be incorrect.

A graph showing different colored squares

AI-generated content may be incorrect.

A graph showing the top states

AI-generated content may be incorrect.

A bar graph with different colored bars

AI-generated content may be incorrect.

A chart of a company's company

AI-generated content may be incorrect.

A close-up of a text

AI-generated content may be incorrect.

A graph of different colored bars

AI-generated content may be incorrect.

**Step 5: Feature Extraction & Model Comparison**

We experimented with three major approaches to embedding and classification:

**1. TF-IDF + LinearSVC**

* Achieved 92.1% accuracy.
* Best performing model overall.

**2. Bag of Words + Multinomial Naive Bayes**

* Achieved 67.7% accuracy.
* Quick baseline model with lower performance.

**3. Word2Vec + LSTM**

* Embedded complaint vectors using trained Word2Vec.
* Achieved 78.3% accuracy.
* Strong deep learning alternative.

**4. Results & Visualizations**

**Accuracy Comparison**

|  |  |
| --- | --- |
| **Model** | **Accuracy** |
| **TF-IDF + LinearSVC** | **0.92** |
| **BoW + Naive Bayes** | **0.68** |
| **Word2Vec + LSTM** | **0.78** |

* **TF-IDF with LinearSVC outperformed others.**
* **LSTM provided good results, but training was resource intensive.**
* **BoW worked well as a simple benchmark.**

**A graph with different colored squares

AI-generated content may be incorrect.**

**Confusion Matrix (LSTM)**

* Clear distinction between positive and neutral/negative classes.
* Misclassifications occurred mostly between neutral and negative.

A blue squares with numbers and a white background

AI-generated content may be incorrect.

**Word Cloud**

* Words like “account”, “payment”, “credit”, “report”, “capital one” appeared frequently.

**Sentiment Distribution**

* 29187 - positive
* 22681 - negative
* 8132 - neutral

**5. Challenges and Learnings**

* Installing and managing NLTK/VADER/TextBlob
* Large RAM usage during Word2Vec + LSTM
* Model drift due to imbalance in some classes

**Learnings**:

* Difference between TF-IDF vs Embeddings
* How context improves sentiment detection
* Deploying prediction-ready LSTM with Keras

**6. Live Demo Example (Test Cases)**

demo\_reviews = [

"I am extremely disappointed with the service. I will never use this bank again.",

"The staff was very helpful and resolved my issue quickly. Great experience!",

"The transaction process was okay, but took longer than expected."

]

**Predicted Sentiments:**

* 1 → Neutral
* 2 → Positive
* 3 → Neutral

**7. Future Work**

* Fine-tune BERT transformer models for even higher accuracy.
* Add topic modeling or keyword extraction.
* Integrate with web app for real-time sentiment feedback.
* Use multi-label classification for complaints with mixed sentiments.

**8. Conclusion**

We successfully built an end-to-end NLP pipeline for sentiment classification. Using pre-trained and self-trained embeddings and comparing ML vs DL models provided deeper insights into text analytics.

Even though the TF-IDF + Logistic Regression model achieved the highest accuracy (92%), the Word2Vec + LSTM model was selected as the best performing model for the final deployment and use case.

We successfully built and evaluated multiple NLP pipelines using classic machine learning (TF-IDF, BoW) and deep learning (Word2Vec + LSTM). While TF-IDF + Logistic Regression had the highest numeric accuracy, the Word2Vec + LSTM model provided superior contextual understanding and generalizability. This makes it the most practical choice for deploying in sentiment analysis tools aimed at consumer complaint tracking, where understanding the tone and structure of language is critical.

**Best Performing Model**: Word2Vec + LSTM

**Use Case Readiness**: Suitable for consumer sentiment tracking tools

**9. Appendix**

* Dataset source: <https://www.kaggle.com/datasets/ashwinik/consumer-complaints-financial-products>
* Libraries used: pandas, gensim, tensorflow, sklearn, seaborn, matplotlib, wordcloud , textblob, transformers, etc.
* Total runtime: Approximately 6 hours including training

**10. References**

*Consumer Complaints - Financial products*. (2020, August 2). Kaggle. <https://www.kaggle.com/datasets/ashwinik/consumer-complaints-financial-products>

*TextBlob: Simplified Text Processing — TextBlob 0.19.0 documentation*. (n.d.). <https://textblob.readthedocs.io/en/dev/>

*Hugging Face – The AI community building the future.* (n.d.). <https://huggingface.co/>