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

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.

```
Dataset Shape: (670598, 18)

Columns:
 ['Date received', 'Product', 'Sub-product', 'Issue', 'Sub-issue', 'Consumer complaint narrative', 'Company public response', 'Company', 'State', 'ZIP code', 'Tags', 'Consum

Sample rows:
                                             text         Product  \
0  Received Capital One charge card offer XXXX. A...      Credit card
1  I do n't know how they got my cell number. I t...  Debt collection
2  I 'm a longtime member of Charter One Bank/RBS...      Credit card

                        Company State Submitted via
0                   Capital One    OH           Web
1    CCS Financial Services, Inc.    AR           Web
2  Citizens Financial Group, Inc.    MI           Web

Complaint length stats:
count    114704.000000
mean       1052.132070
std         918.395711
min          10.000000
25%         396.000000
50%         746.000000
75%        1402.000000
max        5153.000000
Name: text_length, dtype: float64
```

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

```
Sample Before Cleaning:
                                                text
0  Received Capital One charge card offer XXXX. A...
1  I do n't know how they got my cell number. I t...

Cleaning text (please wait)...

Cleaned Text Preview:
                                                text  \
0  Received Capital One charge card offer XXXX. A...
1  I do n't know how they got my cell number. I t...
2  I 'm a longtime member of Charter One Bank/RBS...

                                        cleaned_text
0  receive capital one charge card offer xxxx app...
1  n know get cell number tell would deal onlybwi...
2  longtime member charter one bank rbs citizens ...

Final cleaned dataset saved to:
C:/BISMI/COURSE/AI/Machine Learning Programming/Final Project/dataset/final_cleaned_60000.csv
```

Step 3: Sentiment Analysis using TextBlob

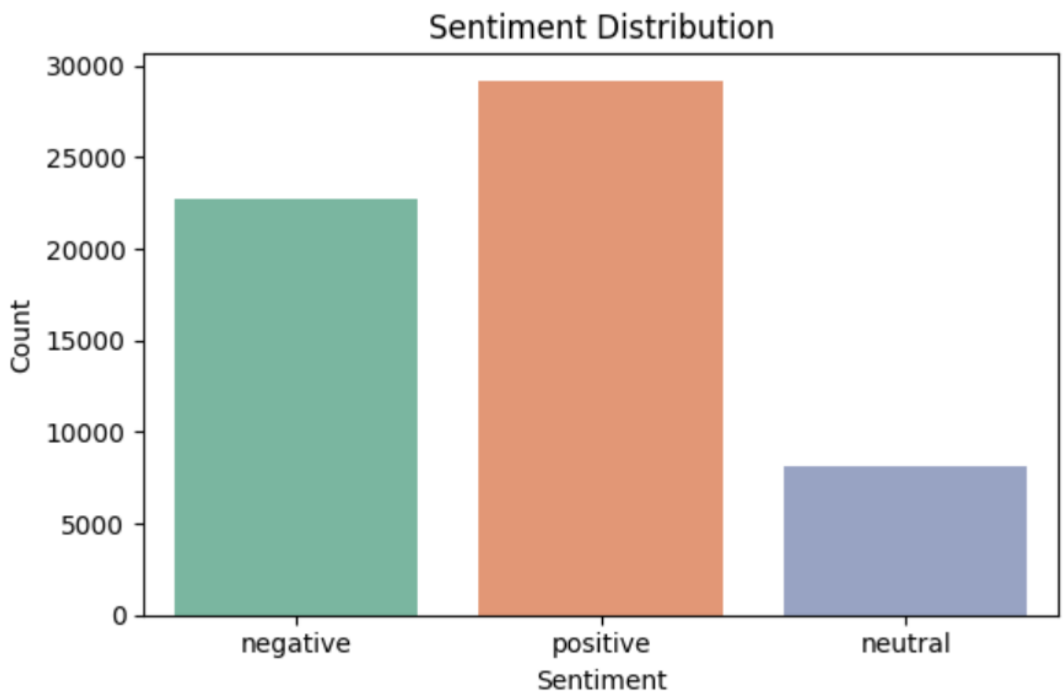
- Used the polarity score to assign one of three classes: positive, negative, or neutral.
- Saved the results in sentiment_textblob_60000.csv.

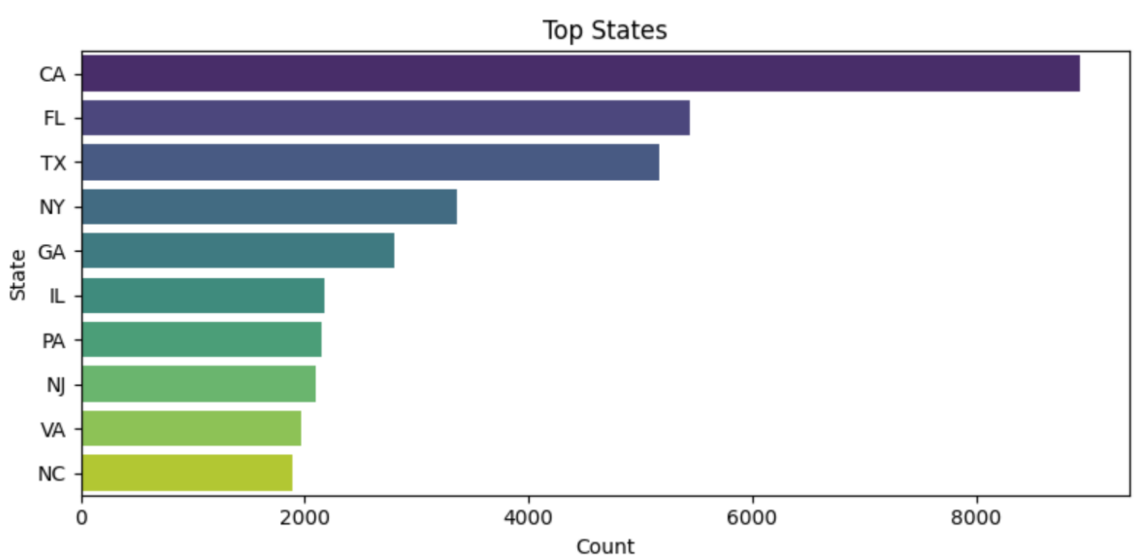
```
Running TextBlob sentiment analysis...

Sample with Sentiment:
                                        cleaned_text sentiment
0  receive capital one charge card offer xxxx app...  negative
1  n know get cell number tell would deal onlybwi...  positive
2  longtime member charter one bank rbs citizens ...  positive
3  look credit report saw collection account belo...  negative
4  receive call xxxx xxxx xxxx xxxx ext xxxx stat...  positive

Sentiment-added file saved to:
C:/BISMI/COURSE/AI/Machine Learning Programming/Final Project/dataset/sentiment_textblob_60000.csv
```

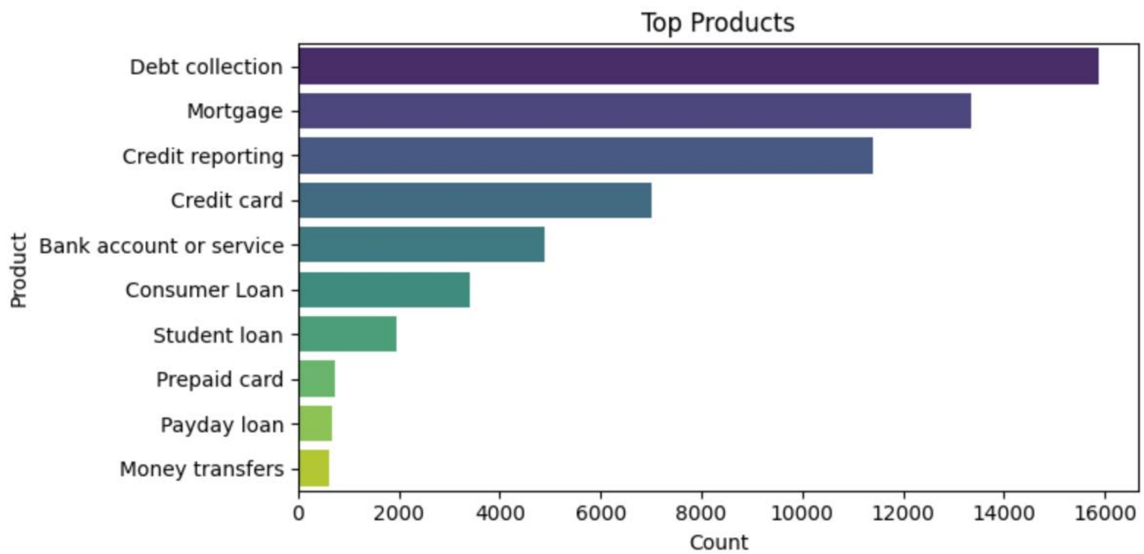
Step 4: Exploratory Data Analysis (EDA)

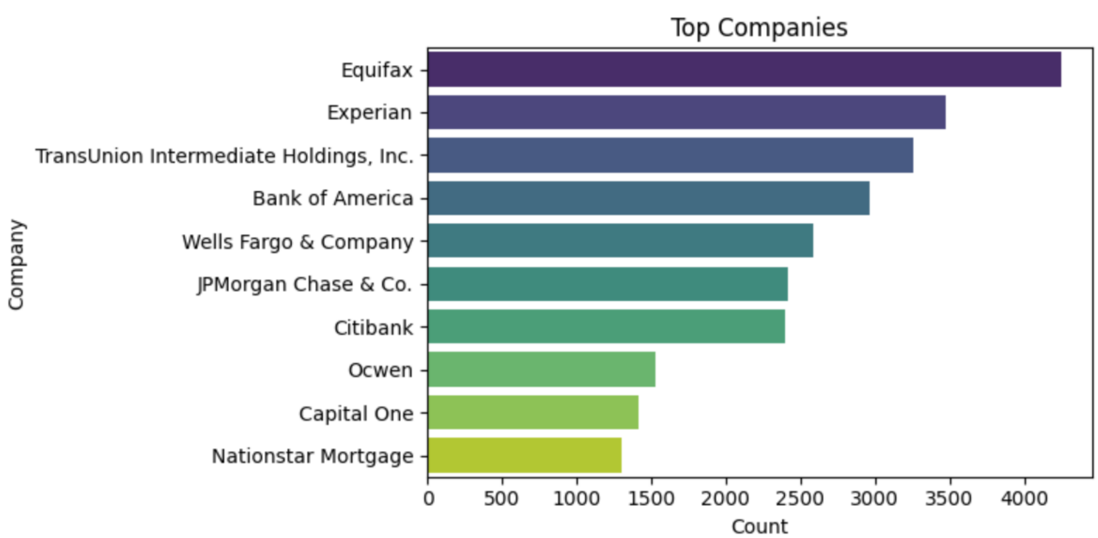
- Plotted sentiment distribution bar chart.
- Generated bar plots of top states, products, and companies by complaint count.
- Created a word cloud from all cleaned complaints.
- Analyzed sentiment distribution across products and companies.

```
 Shape of data: (60000, 8)

 Columns: ['text', 'Product', 'Company', 'State', 'Submitted via', 'text_length', 'cleaned_text', 'sentiment']

 Sentiment distribution:
 sentiment
positive    29187
negative    22681
neutral      8132
Name: count, dtype: int64
```

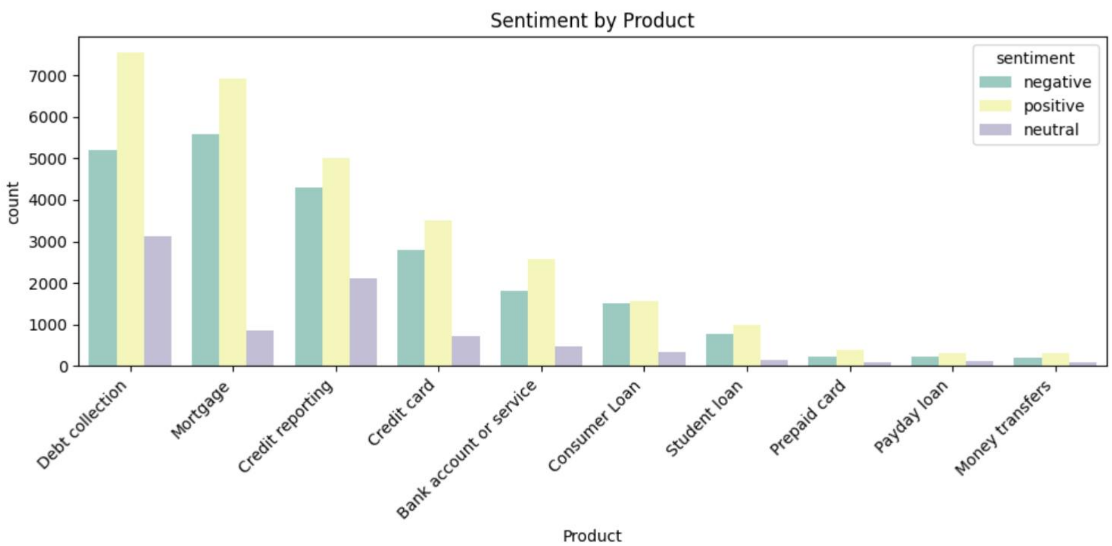


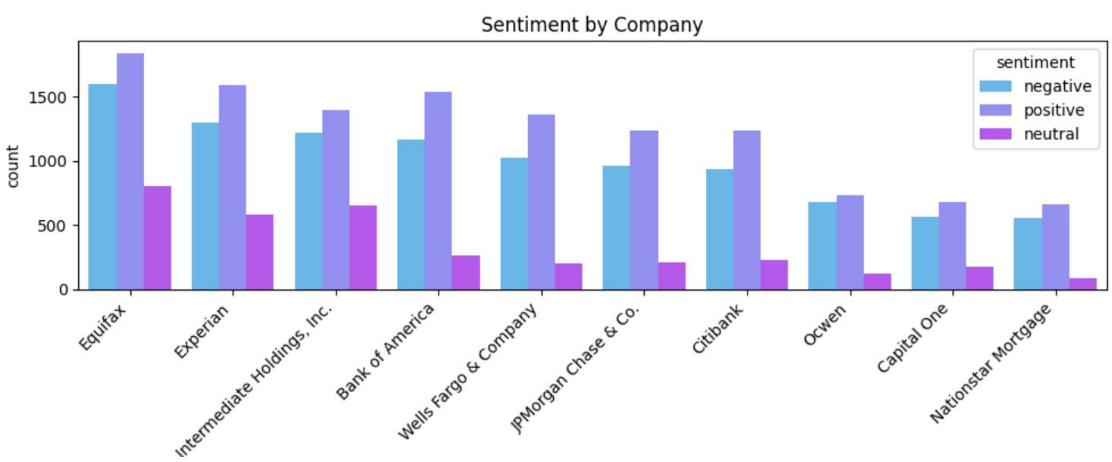



Top Products
Debt collection
Mortgage
Credit reporting
Credit card
Bank account or service
Consumer Loan
Student loan
Prepaid card
Payday loan
Money transfers
Product
0
2000
4000
6000
8000
10000
12000
14000
16000
Count


Top Companies
Equifax
Experian
TransUnion Intermediate Holdings, Inc.
Bank of America
Wells Fargo & Company
JPMorgan Chase & Co.
Citibank
Ocwen
Capital One
Nationstar Mortgage
Company
0
500
1000
1500
2000
2500
3000
3500
4000
Count


Common Words in Complaints
make payment
collection agency
customer service
credit bureau
account xxxx
call xxxx
well fargo
check account
receive letter
phone call
credit score
credit report
credit card
bank america
loan modification
late fee
xxxx account
xxxx payment
interest rate
tell
say


Sentiment by Product
sentiment
negative
positive
neutral
count
0
1000
2000
3000
4000
5000
6000
7000
Debt collection
Mortgage
Credit reporting
Credit card
Bank account or service
Consumer Loan
Student loan
Prepaid card
Payday loan
Money transfers
Product


Sentiment by Company
sentiment
negative
positive
neutral
count
0
500
1000
1500
Equifax
Experian
Intermediate Holdings, Inc.
Bank of America
Wells Fargo & Company
JPMorgan Chase & Co.
Citibank
Ocwen
Capital One
Nationstar Mortgage

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

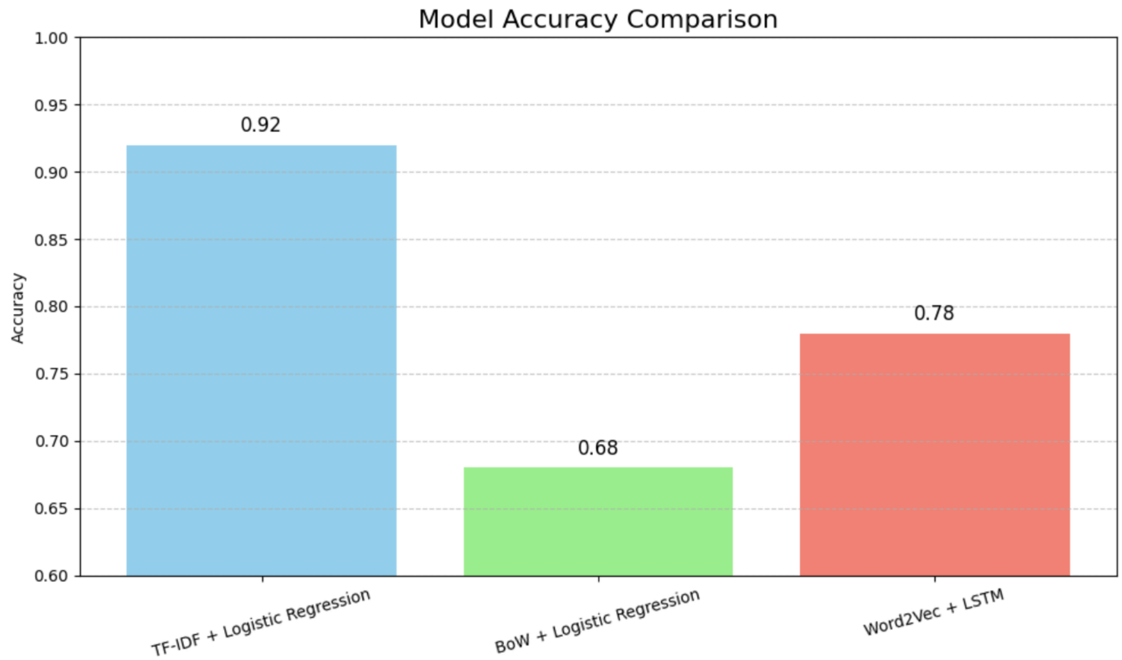
- Embedded complaint vectors using trained Word2Vec.
- Achieved 78.3% accuracy.
- Strong deep learning alternative.

4. Results & Visualizations

Accuracy Comparison

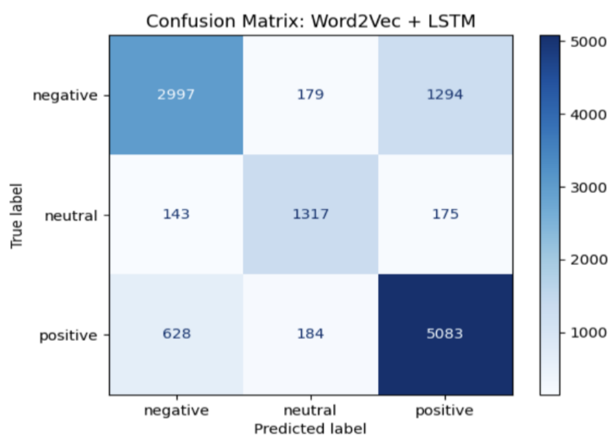
Model	Accuracy
TF-IDF + LinearSVC	**0.92**
BoW + Naive Bayes	**0.68**
Word2Vec + LSTM	**0.78**

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



Confusion Matrix (LSTM)

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



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/