Level-1-Data Analytics and Predictive Modeling Project Documentation

Project Title: *Sentiment Analysis for Product Reviews*

Team Members:

- Madhavendra Singh Naruka

Project Overview:

The goal of this project is to analyse product reviews to determine the sentiment expressed by customers. By leveraging sentiment analysis, the project aims to provide insights into customer satisfaction, identify areas for product improvement, and enhance marketing strategies.

Project Structure

I have organized my project into three Jupyter notebook files to ensure modularity and clarity:

1. **Data Extraction and Sample Saving**:
    - This notebook focuses on extracting relevant data and saving representative samples.
    - It covers data collection, preprocessing, and storage.
2. **Exploratory Data Analysis (EDA)**:
    - In this notebook, I explore the dataset, visualize key features, and identify patterns.
    - EDA helps us understand the data distribution, correlations, and potential insights.
3. **Model Development**:
    - The third notebook is dedicated to building and evaluating machine learning models.
    - It includes feature engineering, model selection, hyperparameter tuning, and performance assessment.

By maintaining this structure, I aim to enhance readability, facilitate collaboration, and streamline the development process. 🚀

1. Introduction

- **Purpose**: The goal of this project is to analyse product reviews to determine the sentiment expressed by customers and to develop models for future Sentiment Analysis.
- **Scope**: I will start with an EDA to understand the relationships in the available data and then make models.

2. Data Collection

- **Data Sources**: I have downloaded Amazon Health and Personal Care Products data, which consists of approximately 500,000 rows and 4.9 million data points. https://drive.google.com/drive/folders/1KVzFcaLU0t7r8sArm6xK4eRSGGjsZaN?usp=sharing
  considering limited capabilities of my system I took a sample of 10,000 rows which I divided into 20-80, train-test dataset.
- **Data Acquisition**: The data was pre-cleaned and in json format.

3. Data Preprocessing

Initial Data

	rating	title	text	images	asin	parent_asin	user_id	timestamp	helpful_vote	verified_purchase
128517	5	Best price	They're Kleenex. Puffs with lotion is much sof...	[]	B07HSF5HTX	B0BSRPX53Z	AFWZZQ4UTCNPISFNY62T67XJLW2Q	2022-05-17 18:17:54.442	0	True

1. Text Processing

- **Combined title and text columns into review.**
- **Applied preprocessing function (preprocess_text) to tokenize and clean text data.**
- **preprocess_text is a custom function that performs various functions like tokenization, lowercasing, removing stopwords & punctuation, lemmatization using spaCy, and filtering out single-letter words and spaces.**

2. Timestamp Conversion

- **Converted timestamp column to datetime format.**

3. Sentiment Analysis

- **Calculated polarity and subjectivity scores using TextBlob.**

4. Data Cleaning

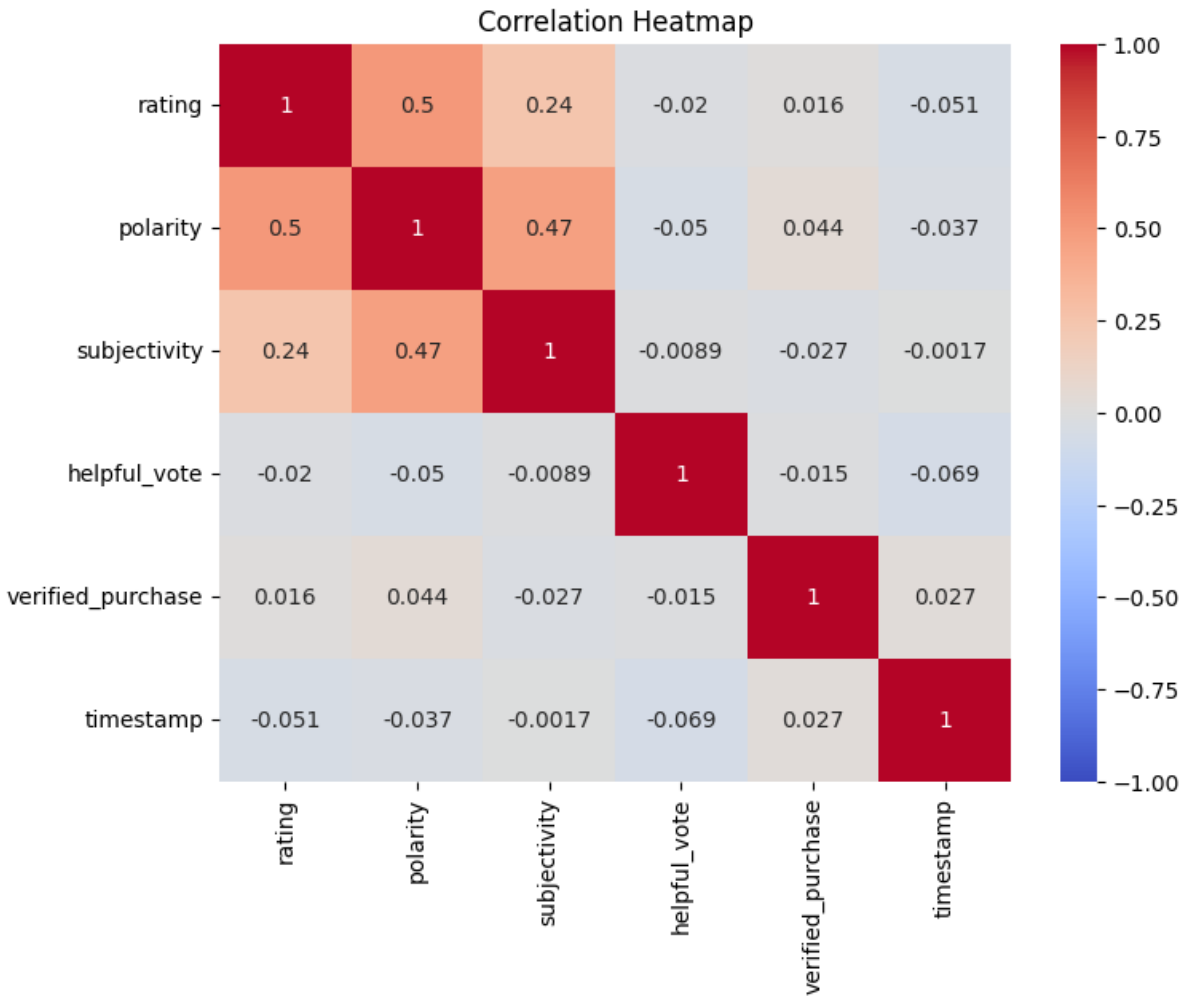
- **Removed unnecessary columns: title, text, images, asin, parent_asin, user_id.**

5. Exporting Data

- **Saved processed data to sample.csv for easier and faster access in future.**

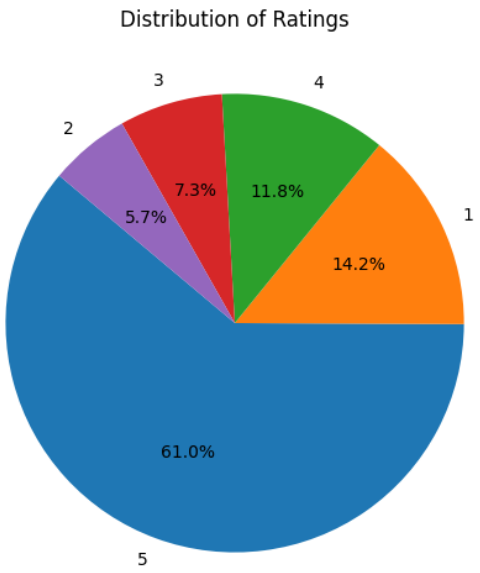
4. Exploratory Data Analysis (EDA)

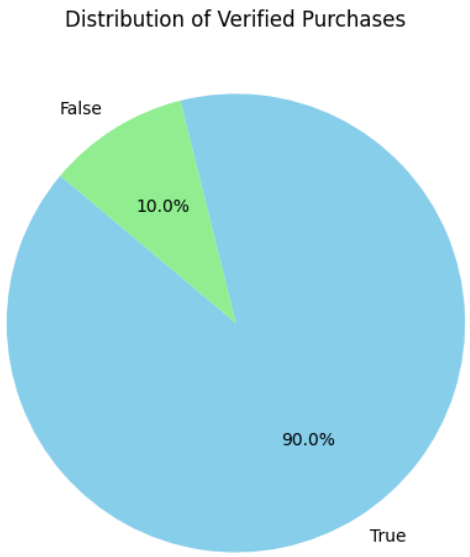
- **Correlation Heatmap**



Significant Findings: Ratings and polarity, have a moderate positive correlation, indicating that more positive reviews tend to have higher ratings.

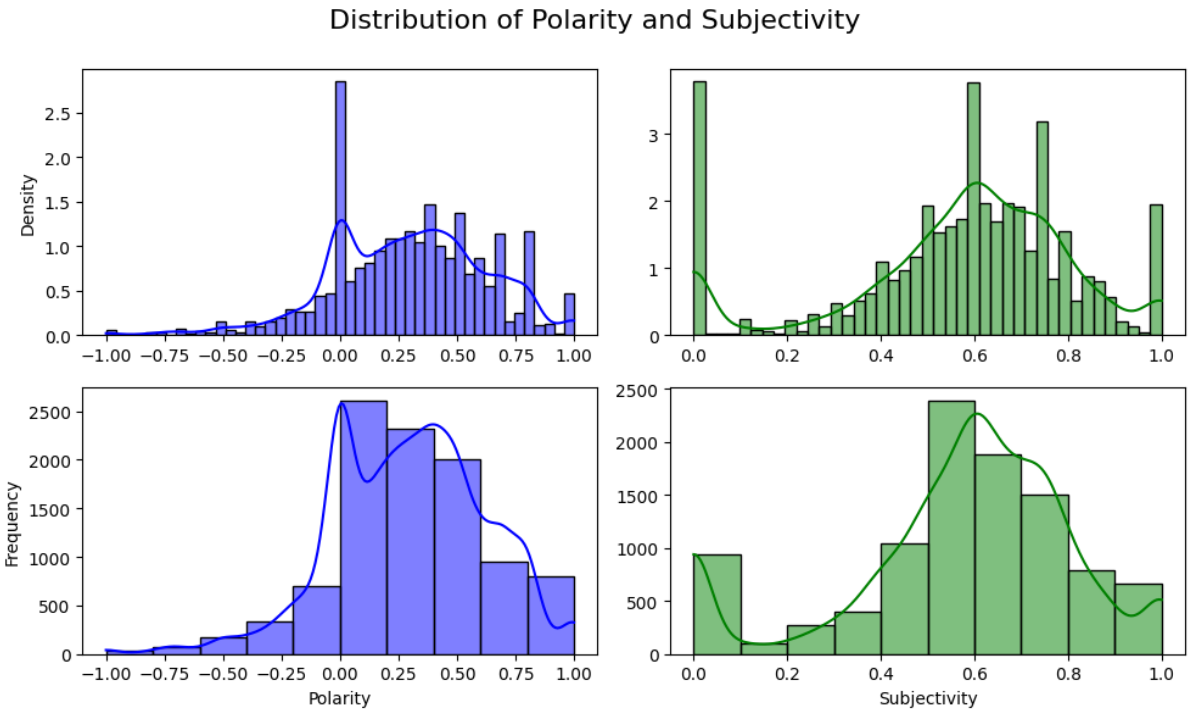
I converted timestamp to unix-timestamp to make corelation matrix





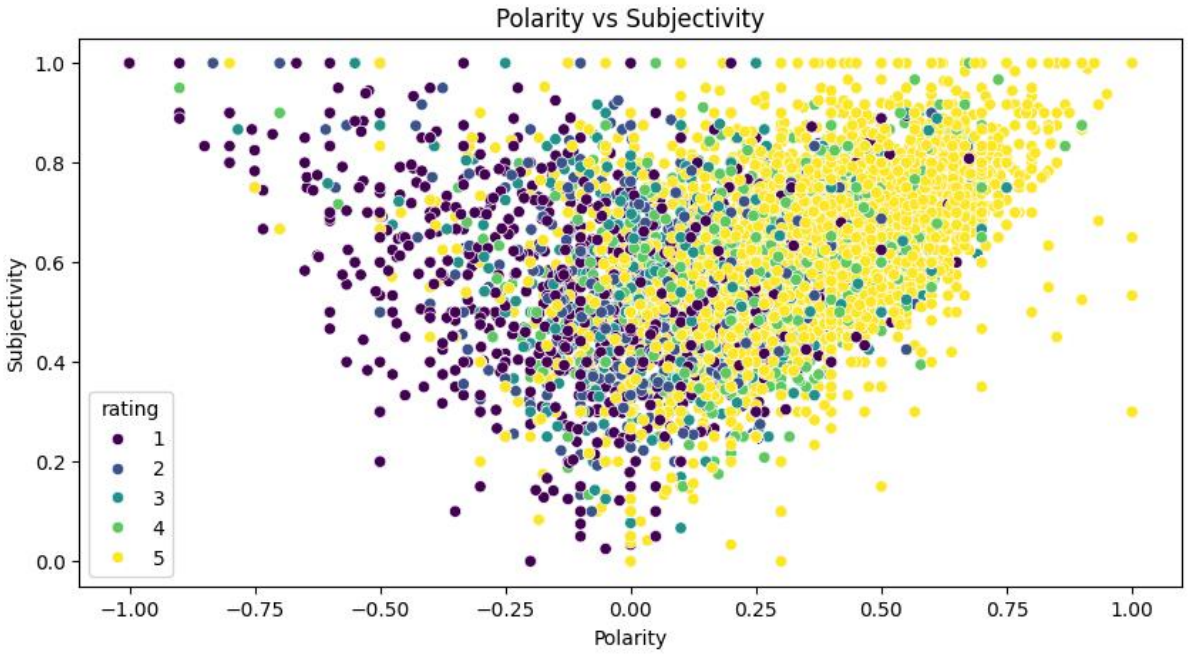
Significant Findings:

- **Positive Customer Sentiment**: More than 70% of reviews express positive feedback, indicating a high level of customer satisfaction with the product/service.
- **Verified User Base**: Approximately 90% of reviews are contributed by verified users, enhancing the credibility and trustworthiness of the feedback.



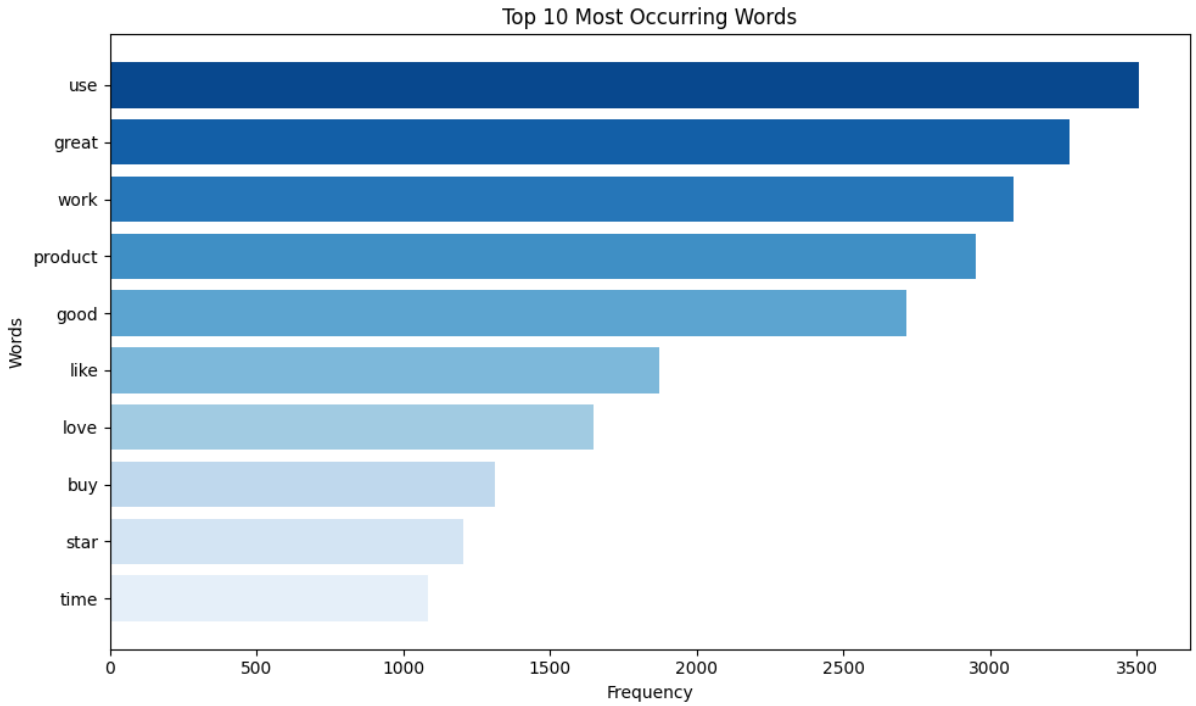
Significant Findings:

1. **Polarity**:
    - There's a slight lean towards positive sentiment(slightly right of zero).
2. **Subjectivity**:
    - A significant peak around 0.5 indicates moderate subjectivity.
    - Entries are evenly distributed between objective and subjective.



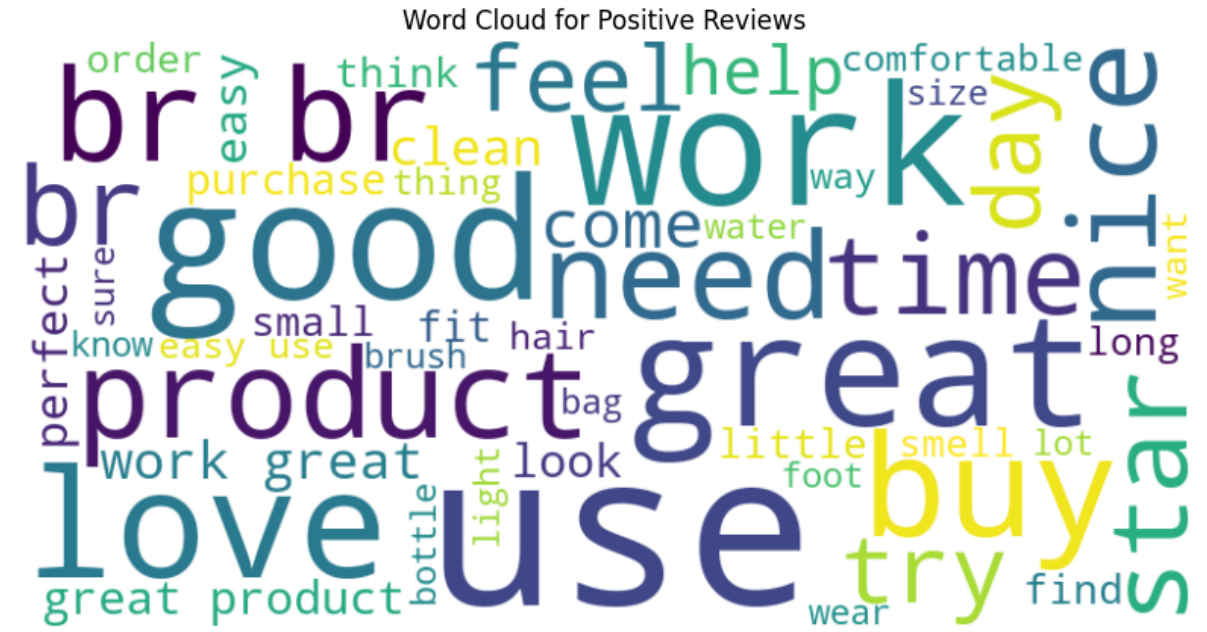
Significant Findings:

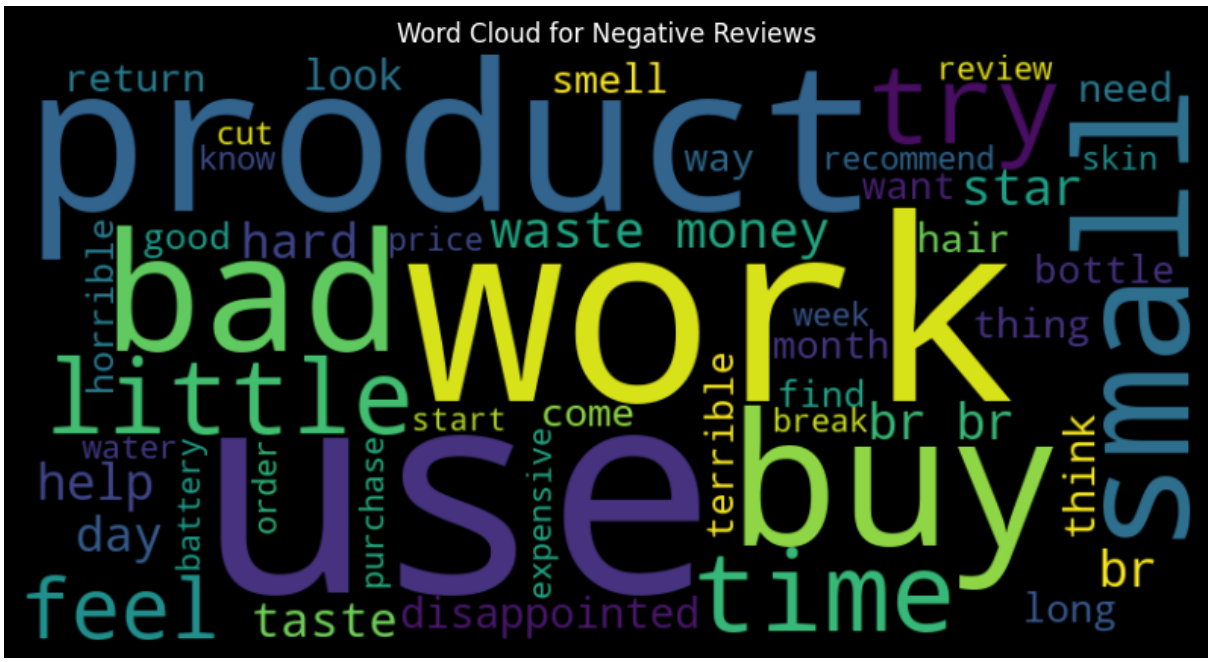
- As positive sentiment increases, subjectivity tends to increase as well. Expressive language and emotional tone are more common in positive reviews
- As positive sentiment **decreases**, subjectivity tends to **decrease** as well. Objective language and factual tone are more common in negative reviews.



Significant Findings:

- Frequent occurrence of ‘use’ and work suggest that Users are likely more concerned about how their products works rather than its design.
- **Positive Words (“Great”, “Good”, “Love”, “like”),**suggest general satisfaction
- The word “time” suggests that delivery time matters.





Significant Findings:

“small” and “little” suggest, customer dissatisfaction regarding size or quantity of product.

Customers don’t seem satisfied with “smell”, ”taste”

Words like “week”, ”month” suggests that customers might have been dissatisfied with the delivery time or product performance within a month or week.

Significant Findings:

- I have not put up all month's graph but august significantly has better rating than other months, maybe a nicer weather affects user ratings(further analysis is required), upon confirmation suitable marketing strategies can be adopted.
- Also, overall ratings seems to improve and stabilize over time, most recent decline was seen 2016 onwards, probably due to demonetization product quality was affected or people simply expected more value for amount paid

5. Feature Engineering

Feature Selection:

- I have used Bag of Words (BoW), Term Frequency-Inverse Document Frequency (TF-IDF) as firstly they were mentioned in project guidelines and also because upon researching I found them to be useful to implement traditional machine learning models like Logistic Regression, Naive Bayes, and Support Vector Machines.
- I have decided to use traditional models as a firm step towards understanding the concepts better.

- **Feature Creation**: I used normalized rating and polarity to create a score to classify sentiments to train model

```
def classify_sentiment(normalized_rating,polarity):
    score=(normalized_rating*0.5+polarity*0.5)
    if score < -0.5:
        return 'strongly negative'
    elif -0.5 <= score < 0:
        return 'negative'
    elif 0 <= score < 0.5:
        return 'positive'
    elif score >= 0.5:
        return 'strongly positive'
    else:
        return 'neutral'
```

6. Model Development

Model Selection:

I opted for traditional models, including Logistic Regression, Naive Bayes, and Support Vector Machines (SVM). These choices align with the project guidelines and are compatible with Bag of Words (BoW) and TF-IDF representations, which I found easy to work with.

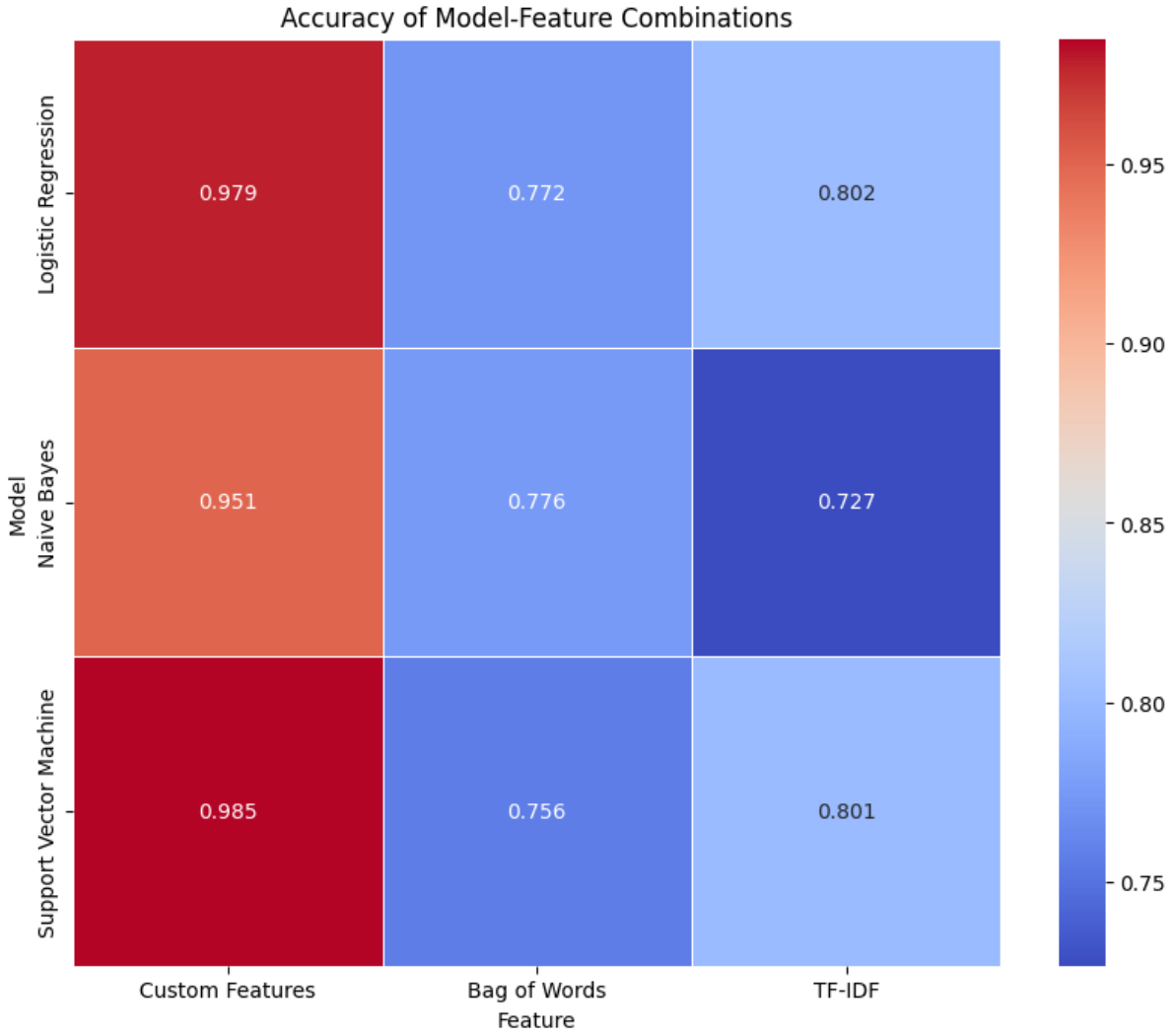
Model Training:

1. **Library Usage:**
    - **I leveraged the pre-built model library in Python's scikit-learn (sklearn). This library provides all three models (Logistic Regression, Naive Bayes, and SVM) and simplifies feature engineering for BoW and TF-IDF.**
2. **Model Exploration:**
    - **I experimented with three models:**
        - **Logistic Regression**
        - **Naive Bayes**
        - **Support Vector Machines (SVM)**
    - **For each model, I explored different feature representations:**
        - **BoW**
        - **TF-IDF**
        - **Custom features**
3. **Custom Features:**
    - **My custom features consist of a subjectivity score and a sentiment score.**
    - **The sentiment score is derived as half the sum of normalized ratings and polarity.**

```
X = df[[ 'subjectivity', 'score']]
```

4. **Target Column:**
    - **The target column for all models was the sentiment label, which can take one of five values:**
        - **Positive**
        - **Strongly positive**
        - **Neutral**
        - **Negative**
        - **Strongly negative**

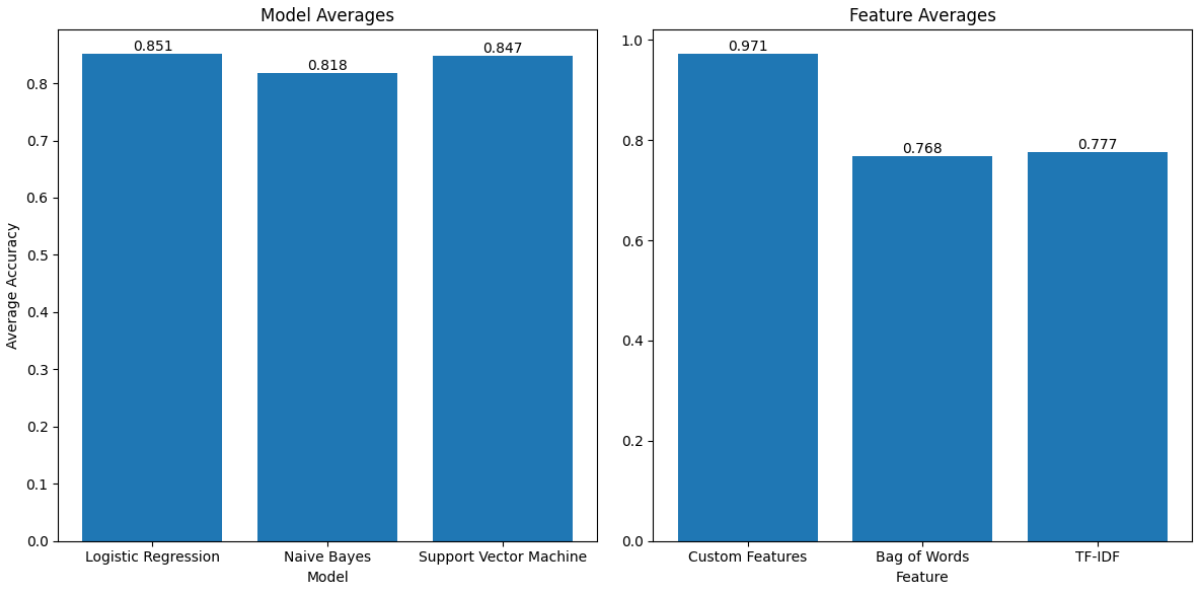
- **Model Evaluation**:



While SVM with Custom Features is the most accurate(98.5%) model the time it takes is far more than Logistic Regression with Custom Features(97.9%), with such a minor difference in accuracies and a huge difference in time later is more suitable for large scale projects.

7. Model Interpretation

- **Feature Importance**:
  - **Custom Features**: Most simple and accurate feature.
  - **BoW(Bag of Words)**: it is simply a matrix of word count across different reviews. Proved to be least accurate.
  - **TF-IDF:** It is a matrix of product of term frequency (TF) and inverse document frequency (IDF). Better than Naïve Bayes as considers the importance of words based on their frequency across documents.



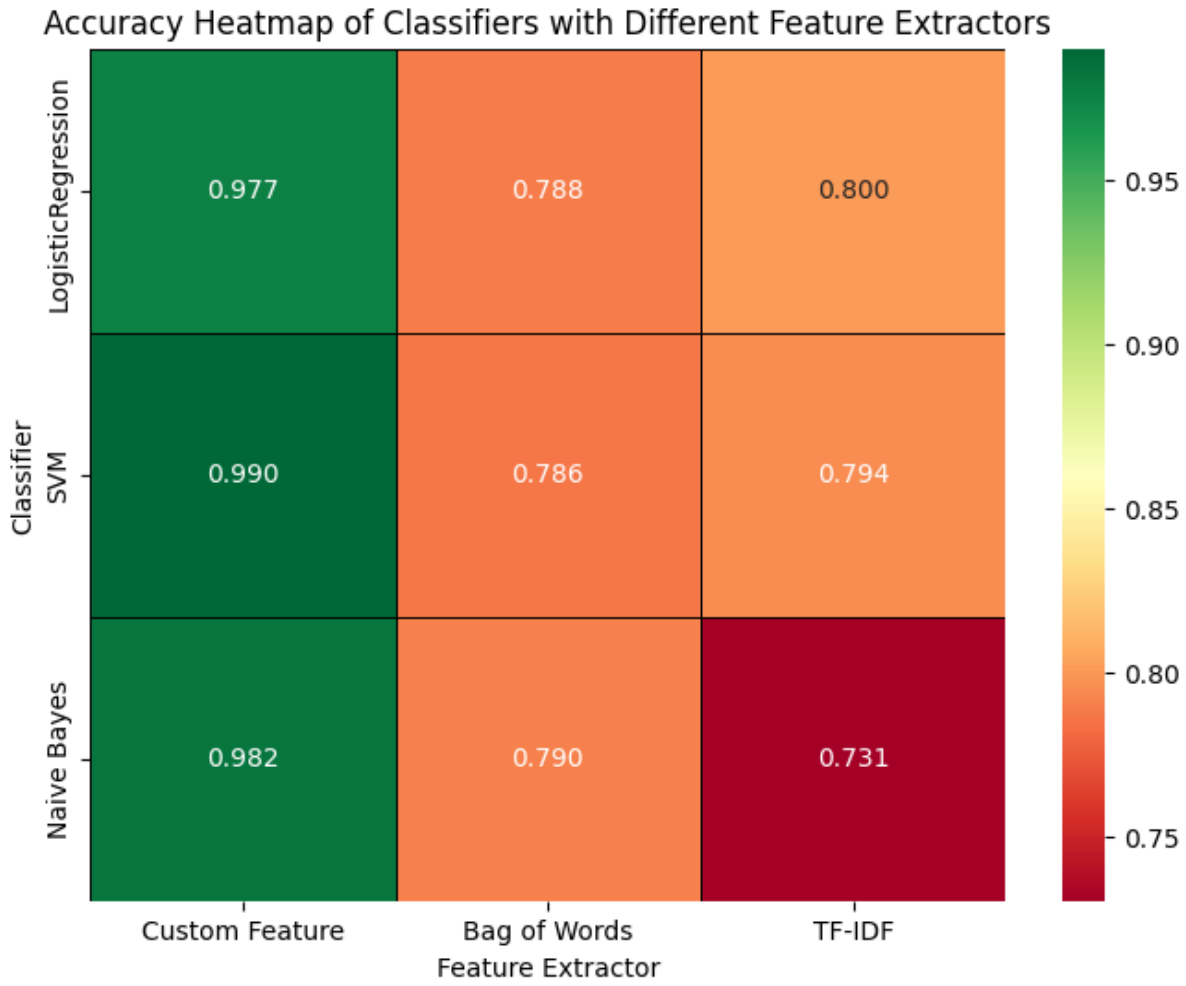
- **Model Insights**: Logistic Regression is simple yet most effective for Sentiment Analysis among traditional algorithms.
- Using 2 Features like Bag of Words and TF-IDF together in a pipeline averages out their accuracy scores.

PIPELINES:

- I have also experimented with pipelines and have found them to be better to manage.
- I created a common function to make pipelines

```
def train_and_evaluate_pipeline(X_train, y_train, X_test, y_test, feature_extractor, classifier):
```

- The lines of code decreased greatly while modularity and manageability increased significantly.
- The results have changed here with average accuracy increasing while Naïve Bayes has become the most accurate model here



Models implemented with pipelines

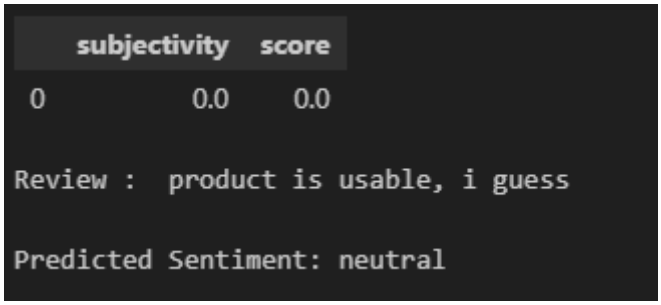
8. Model Deployment

Deployment Plan:

- I chose to implement Logistic Regression with Custom Features where I used polarity instead of score and subjectivity to predict sentiment
- To implement this I developed a function capable of creating and saving a model with a `.pkl` extension using joblib.

```
def train_evaluate_and_save_pipeline(X_train, y_train, X_test, y_test, feature_extractor,classifier, f_name):
```

- Then I made another Jupyter notebook file, deployment.ipynb, here I imported the model using joblib, took an input review from user and predicted its sentiment.



- The model can be scaled to analyse reviews in bulk in an excel file and predicting their sentiment.

9. Monitoring and Maintenance:

- **Data Usage and User Feedback:** Monitoring involves using data analytics to track model performance and user feedback to assess sentiment accuracy and user satisfaction.
- **Model Enhancement:** To enhance the model:
  - I propose training it on a larger dataset using robust computing resources.
  - Regularly retraining the model with new data will ensure its accuracy and relevance over time.

10. Conclusion

Summary:

- This project has sparked a curiosity and deeper understanding of Machine Learning.
- It has enabled me to gain practical experience with traditional models and features like Bag of Words (BoW) and TF-IDF.
- I learned to create and tweak custom features based on model accuracy.
- I implemented my first pipeline, gaining a basic understanding of its functionality.
- I discovered that combining multiple features doesn't necessarily improve a model's performance; it often brings the accuracy closer to their average.

Challenges:

- My previous experience with Machine Learning was in Knime, where models are created by dragging and dropping components. Transitioning to using Jupyter notebooks for ML work was a different experience.
- To complete this project before my exams starting on 28 June, 2024 I had to work more than 9-10 hours a day.

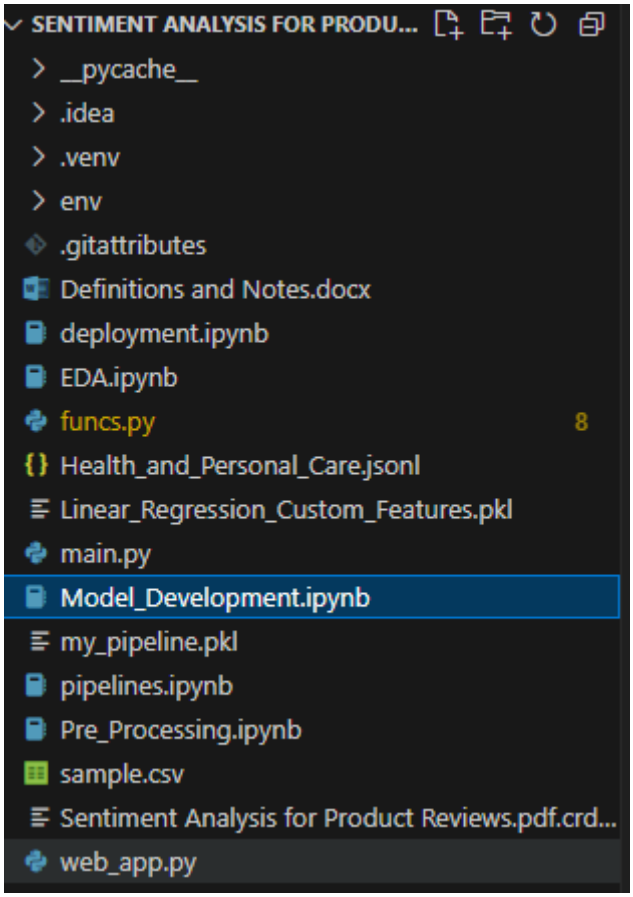
Future Work:

- This project can be scaled further to analyse bulk reviews.
- A website-based service could be developed to generate revenue.
- APIs can be created to contribute to open-source projects.
- More advanced and suitable models can be developed to enhance performance and accuracy.

10. Appendices

- **Additional Visualizations**:

File Structure



- **References**

  - ChatGPT
  - Windows CoPilot
  - BlackBox.ai
  - YouTube