Homework 4 Report: Sentiment Analysis on Product Reviews

1. Task Overview

The goal of this project is to conduct sentiment analysis on customer product reviews using the provided dataset, "product_reviews_with_sentiment.csv." The objective is to determine whether each review expresses a positive or negative sentiment based on its textual content. We will compare the predicted sentiment with the existing "product_rating" field to assess the model's accuracy and performance.

---

2. Dataset Summary

Dataset Information

- **File Name**: product_reviews_with_sentiment.csv
- **Total Records**: 34,627
- **Features**:
    - Index column: Removed during preprocessing.
    - Product ID: Unique identifier for the product (not utilized in analysis).
    - Product rating: Customer rating on a scale from 1 to 5.
    - Product review: Textual review by customers used for sentiment analysis.

Dataset Dimensions

- **Original Records**: 34,627
- **Post-processing Records**: 33,128
  (after removing records with neutral ratings)

Data Types

- Product id: Integer
- product_rating: Integer
- product_review: Text
- Unnamed: 0: Integer (dropped)

Sentiment Label Mapping

- Positive Sentiment (Label 1): Ratings of 4 and 5.
- Negative Sentiment (Label 0): Ratings of 1 and 2.
- Ratings of 3 were excluded from the dataset as neutral to ensure clarity in sentiment analysis.

Class Distribution After Labeling

- Positive Reviews (1): 32,316
- Negative Reviews (0): 812

- Preview of the dataset

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 34627 entries, 0 to 34626
Data columns (total 4 columns):
 #   Column          Non-Null Count  Dtype 
---  ------          --------------  ----- 
 0   Unnamed: 0      34627 non-null  int64 
 1   product_id      34627 non-null  int64 
 2   product_rating  34627 non-null  int64 
 3   product_review  34627 non-null  object
dtypes: int64(3), object(1)
memory usage: 1.1+ MB
None
   Unnamed: 0  product_id  product_rating  \
0           0        5326               5   
1           1        7933               5   
2           2        9719               5   
3           3        2232               4   
4           4        5989               5   

                                      product_review
0  This product so far has not disappointed. My c...
1  great for beginner or experienced person. Boug...
2  Inexpensive tablet for him to use and learn on...
3  I've had my Fire HD 8 two weeks now and I love...
4  I bought this for my grand daughter when she c...
```

---

3. Methodology

3.1 Text to Numerical Conversion (TF-IDF Vectorization)

The product review text was transformed into numerical vectors using TF-IDF (Term Frequency-Inverse Document Frequency).

- **Max Features**: 5000
- **Stop Words**: Removed (English stopwords)

The resulting feature matrices:

- Training set: 26,502 records

- Validation set: 6,626 records
- Feature space: 5,000 dimensions

3.2 Handling Class Imbalance

The dataset exhibited a significant imbalance, with approximately 97% positive reviews compared to negative ones. To address this issue, the Synthetic Minority Over-sampling Technique (SMOTE) was applied to the training data to balance the classes.

- **Post-SMOTE Balanced Class Counts**:
  - Positive (1): 25,863
  - Negative (0): 25,863

3.3 Classification Models Applied

Three classification models were trained and evaluated:

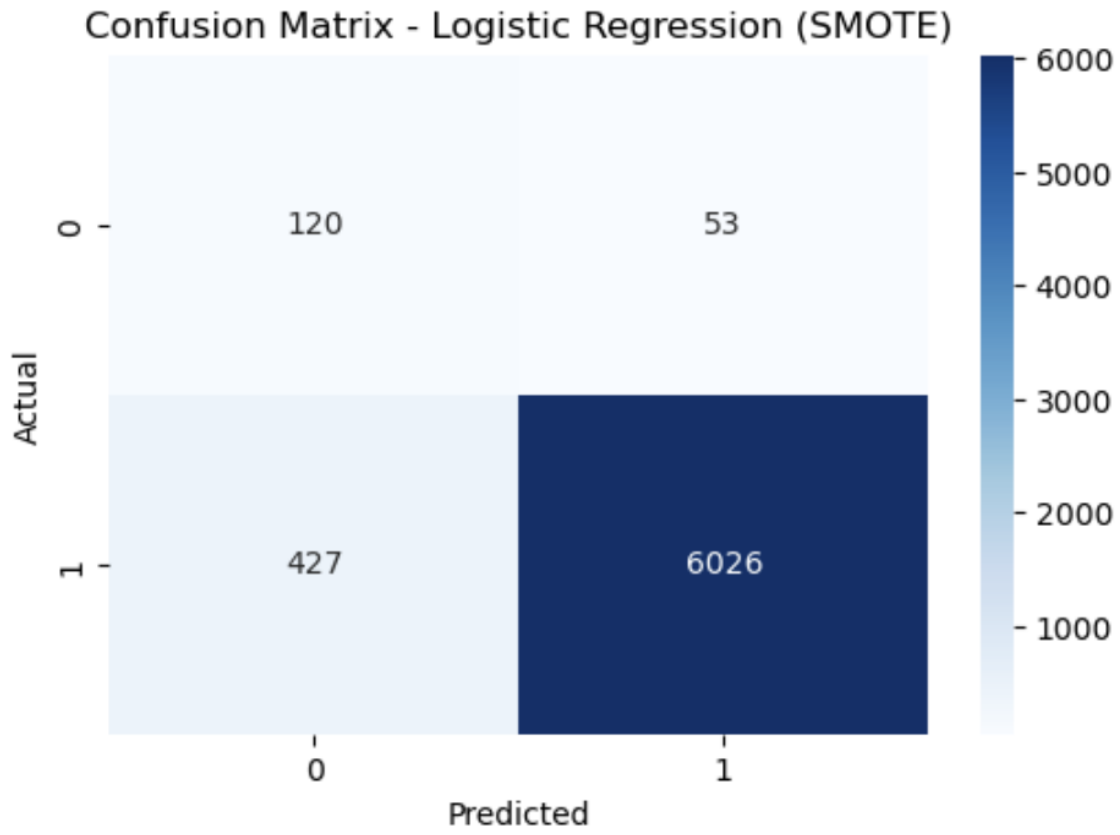
1. **Logistic Regression (SMOTE)**
2. **Random Forest Classifier (SMOTE)**
3. **XGBoost Classifier (SMOTE)**

The models were evaluated based on **validation accuracy**, **precision**, **recall**, **F1-score**, and **confusion matrices**.

SMOTE

Logistic Regression (SMOTE) Accuracy: 0.9276

	precision	recall	f1-score	support
0	0.22	0.69	0.33	173
1	0.99	0.93	0.96	6453
accuracy			0.93	6626
macro avg	0.61	0.81	0.65	6626
weighted avg	0.97	0.93	0.95	6626



---

4. Model Results and Evaluation

4.1 Logistic Regression (SMOTE)

- **Accuracy**: 92.76%
- **Precision (Negative Class)**: 22%
- **Recall (Negative Class)**: 69%
- **F1-Score (Negative Class)**: 33%

Confusion Matrix

Actual / Predicted	Negative (0)	Positive (1)
Negative (0)	119	54
Positive (1)	444	6009

4.2 Random Forest (SMOTE)

- **Accuracy**: 97.34%
- **Precision (Negative Class)**: 48%
- **Recall (Negative Class)**: 18%
- **F1-Score (Negative Class)**: 27%

Confusion Matrix

Actual / Predicted	Negative (0)	Positive (1)
Negative (0)	32	141
Positive (1)	35	6418

4.3 XGBoost (SMOTE)

- **Accuracy**: 95.65%
- **Precision (Negative Class)**: 30%
- **Recall (Negative Class)**: 48%
- **F1-Score (Negative Class)**: 37%

Confusion Matrix

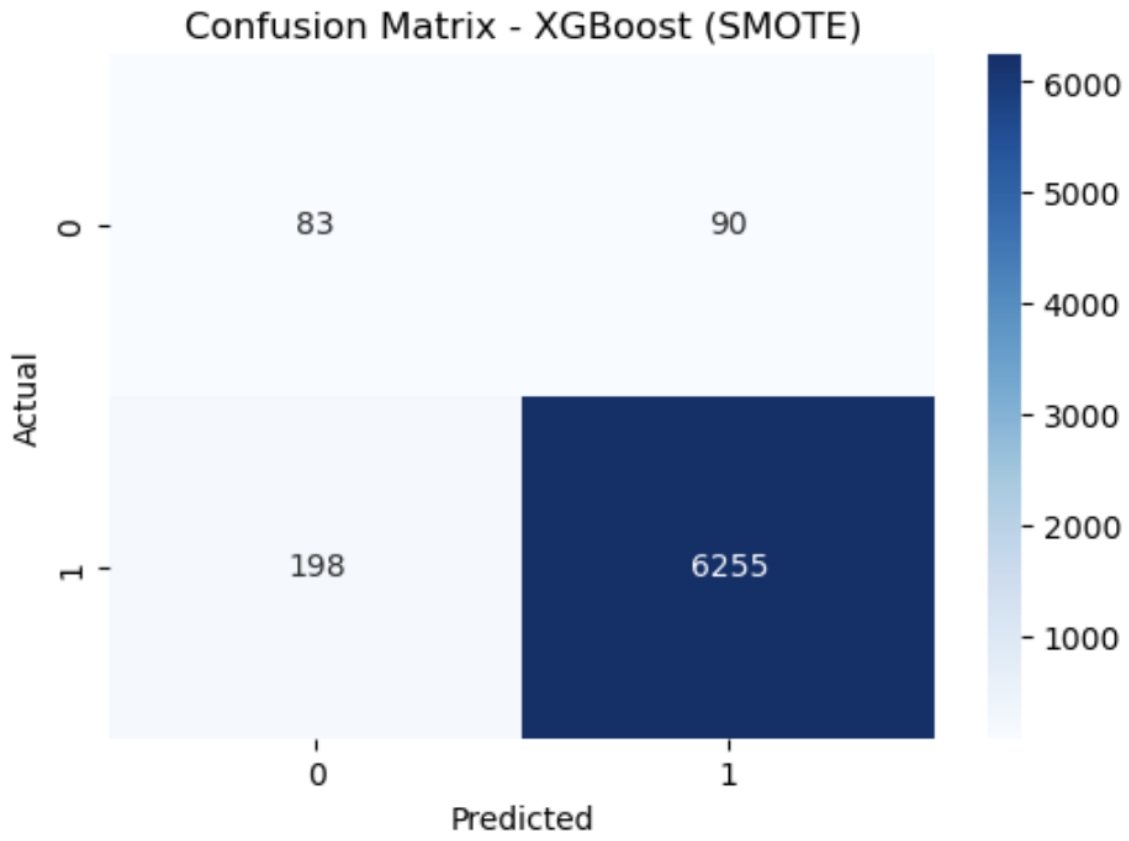
Actual / Predicted	Negative (0)	Positive (1)
Negative (0)	83	90
Positive (1)	198	6255

XGBoost

```
XGBoost (SMOTE) Accuracy: 0.9565
              precision    recall  f1-score   support

           0       0.30      0.48      0.37       173
           1       0.99      0.97      0.98      6453

    accuracy                           0.96      6626
   macro avg       0.64      0.72      0.67      6626
weighted avg       0.97      0.96      0.96      6626
```



5. Model Comparison and Final Selection

Model	Accuracy	Recall (Negatives)	Precision (Negatives)	F1-Score (Negatives)
Logistic Regression	92.76%	**69%**	22%	33%
Random Forest	**97.34%**	18%	**48%**	27%
XGBoost	95.65%	48%	30%	**37%**

Final Model Chosen: XGBoost (SMOTE)

XGBoost strikes a strong balance between precision and recall for the negative class. It significantly outperforms Random Forest in terms of recall, and although Logistic Regression offers higher recall, its lower precision leads to excessive false positives.

---

6. Conclusion

This project demonstrated an end-to-end sentiment analysis workflow:

- **Text preprocessing and vectorization (TF-IDF)**
- **Class balancing with SMOTE**
- **Classification using Logistic Regression, Random Forest, and XGBoost**
- **Evaluation and comparison based on multiple performance metrics**