Sentiment Polarity Classification

Nandini Thakur
Roll number -2101128

OBJECTIVE AND DATASET

Objective: Build a binary classifier to categorize movie reviews as positive or negative.
Dataset: Rotten Tomatoes movie review dataset (10,662 reviews: 5,331 positive, 5,331 negative).

PREPROCESSING AND APPROACH

Downloaded and extracted the dataset.
Analyzed the structure of the data.
Loaded reviews into a DataFrame and labeled them:
Training: Positive reviews as 1
Validation: Negative reviews as 0
Data Splitting:
Training: 4,000 positive & 4,000 negative reviews.
Validation: 500 positive & 500 negative reviews.
Test: 831 positive & 831 negative reviews.
Preprocessing:TF-IDF vectorization to convert reviews into numerical format.

MODELS USED:

I have used Three Models for the sentiment analysis:

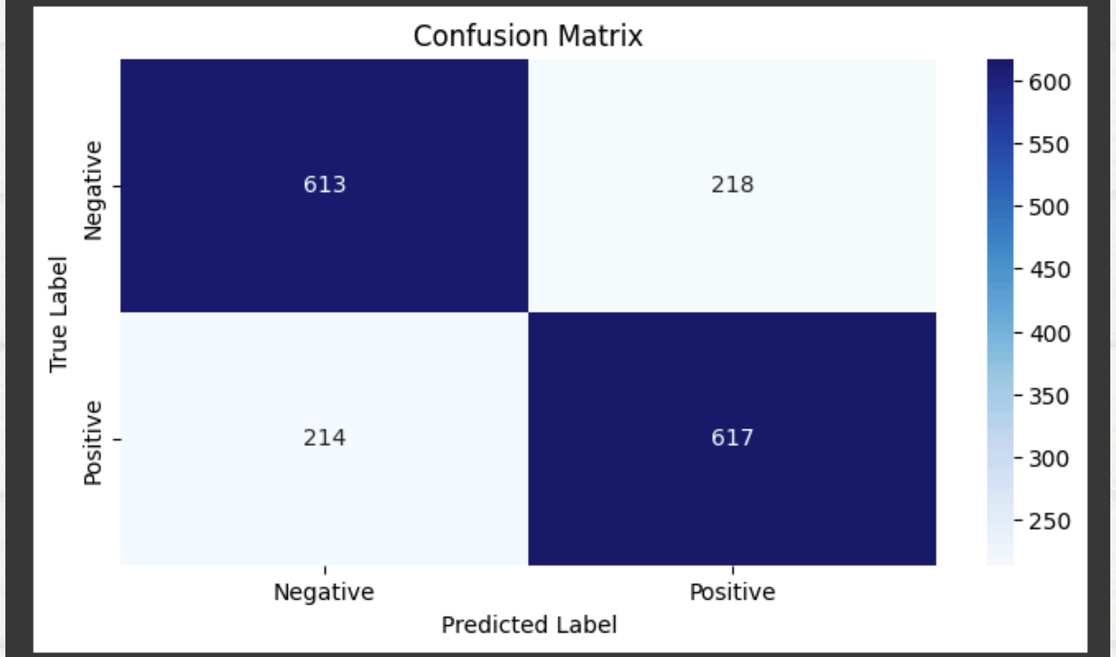
01. LOGISTIC REGRESSION

02. RANDOM FOREST

03. SVM

EVALUATION OF MODEL NAIVE BAYES :

```
Validation Precision: 0.7859, Recall: 0.7340, F1-Score: 0.7590
Test Precision: 0.7389, Recall: 0.7425, F1-Score: 0.7407
True Positives: 617
True Negatives: 613
False Positives: 218
False Negatives: 214

Classification Report for Logistic Rehression:
              precision    recall  f1-score   support

           0       0.74      0.74      0.74       831
           1       0.74      0.74      0.74       831

    accuracy                           0.74      1662
   macro avg       0.74      0.74      0.74      1662
weighted avg       0.74      0.74      0.74      1662
```



EVALUATION OF MODEL RANDOM FOREST :

```
Validation Precision (Random Forest): 0.7432, Recall: 0.6540, F1-Score: 0.6957
Test Precision (Random Forest): 0.7163, Recall: 0.6594, F1-Score: 0.6867
```

EVALUATION OF MODEL RANDOM FOREST :

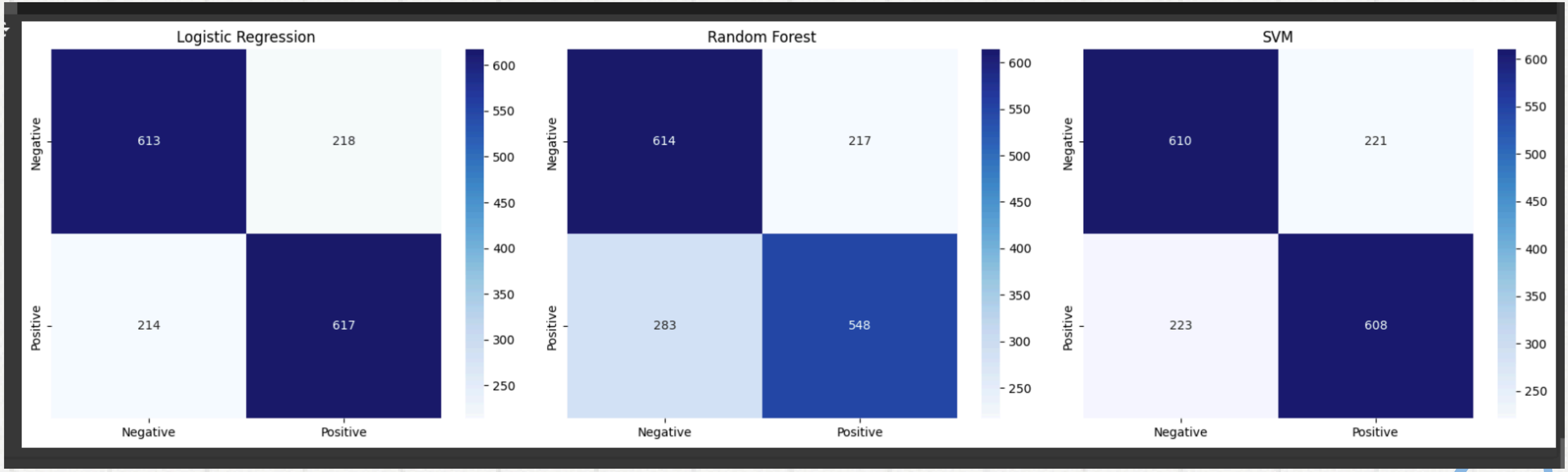
```
Validation Precision (SVM): 0.7868, Recall: 0.7380, F1-Score: 0.7616
Test Precision (SVM): 0.7334, Recall: 0.7316, F1-Score: 0.7325
```

MODEL COMPARISON

Model	Precision	Recall	F1-Score
Logistic Regression	0.7389	0.7425	0.7407
Random Forest	0.7163	0.6594	0.6867
SVM	0.7334	0.7316	0.7325

```
Classification Report for Logistic Regression:
              precision    recall  f1-score   support

           0       0.74      0.74      0.74       831
           1       0.74      0.74      0.74       831

    accuracy                           0.74      1662
   macro avg       0.74      0.74      0.74      1662
weighted avg       0.74      0.74      0.74      1662


Classification Report for Random Forest:
              precision    recall  f1-score   support

           0       0.68      0.74      0.71       831
           1       0.72      0.66      0.69       831

    accuracy                           0.70      1662
   macro avg       0.70      0.70      0.70      1662
weighted avg       0.70      0.70      0.70      1662


Classification Report for SVM:
              precision    recall  f1-score   support

           0       0.73      0.73      0.73       831
           1       0.73      0.73      0.73       831

    accuracy                           0.73      1662
   macro avg       0.73      0.73      0.73      1662
weighted avg       0.73      0.73      0.73      1662
```

CONFUSION MATRIX COMPARISON



BEST MODEL

Here are the key reasons why Naive Bayes might have performed best :

1. **Assumption of Conditional Independence**: Naive Bayes assumes that all features are conditionally independent given the target class. This assumption simplifies the model but works surprisingly well in many practical scenarios, especially when the features are independent or only weakly correlated.
2. **Works Well with High-Dimensional Data**: Naive Bayes performs well in scenarios with high-dimensional data (many features) because it computes the probability of each feature individually. This reduces the complexity of model training.
3. **Handles Small Datasets Effectively**: Naive Bayes is a good fit when the dataset is small. Since it estimates the likelihood of features individually, it doesn't need as much data to learn interactions between features, unlike models like Random Forest or SVM.
4. **Fast Training and Prediction**: The algorithm is computationally efficient because it doesn't require iterative learning as in some other models (e.g., logistic regression, SVM). This leads to faster training and inference, especially useful for large datasets.
5. **Robust to Irrelevant Features**: Naive Bayes is less sensitive to irrelevant features. Even if there are redundant or less useful features in the data, they don't significantly harm the performance because each feature contributes independently to the classification decision.

LINK TO GITHUB

https://github.com/purple0608/Sentiment_Analysis