Building a Machine Learning Model to Predict Movie Review Sentiment

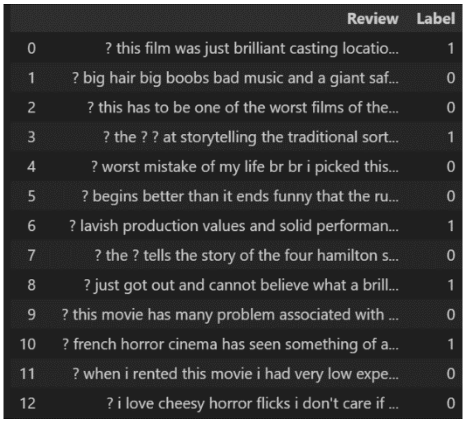
By Shane Henrikson

1. Introduction

The project aims to predict the sentiment (positive or negative) of movie reviews submitted by users. This is a classic example of a binary classification problem in supervised machine learning, as we have labeled training examples (reviews marked as positive or negative) that we can use to train our model.

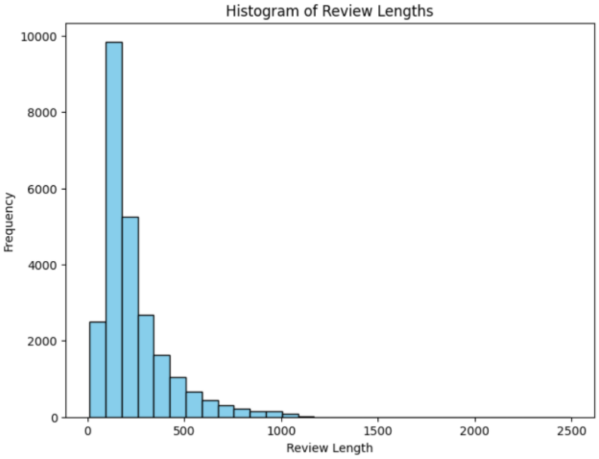
2. Data Exploration

- **Loading the data:** The code snippet in demonstrates how to load the IMDb dataset using Python libraries like TensorFlow and pandas. This dataset likely contains the text of user reviews and their corresponding sentiment labels positive (1) or negative(0).

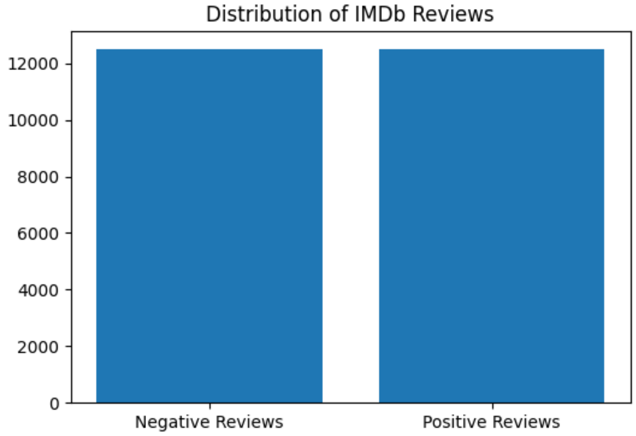


	Review	Label
0	? this film was just brilliant casting locatio...	1
1	? big hair big boobs bad music and a giant saf...	0
2	? this has to be one of the worst films of the...	0
3	? the ? ? at storytelling the traditional sort...	1
4	? worst mistake of my life br br i picked this...	0
5	? begins better than it ends funny that the ru...	0
6	? lavish production values and solid performan...	1
7	? the ? tells the story of the four hamilton s...	0
8	? just got out and cannot believe what a brill...	1
9	? this movie has many problem associated with ...	0
10	? french horror cinema has seen something of a...	1
11	? when i rented this movie i had very low expe...	0
12	? i love cheesy horror flicks i don't care if ...	0

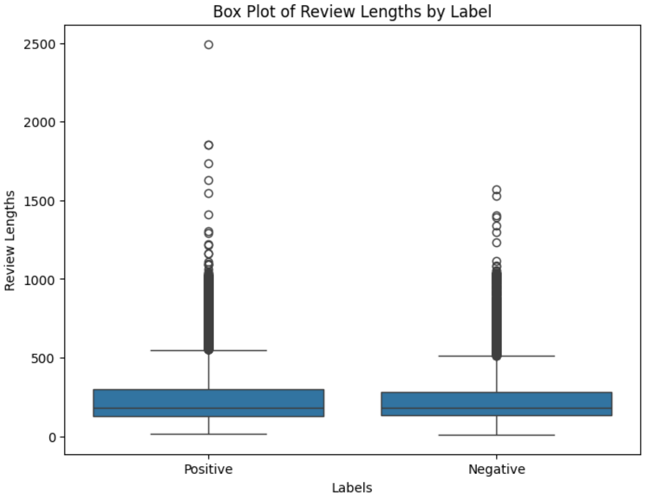
- **Analyzing review lengths:** The "Review Lengths" table in provides statistical insights about the length of the reviews. Knowing the distribution of review lengths can be helpful in later stages, such as deciding whether to truncate or pad reviews to a fixed length.



- **Visualizing label distribution:** The bar chart mentioned in helps visualize the balance between positive and negative reviews in the dataset. It's important to check for class imbalance, as a heavily skewed dataset might require specific techniques during model training.



- **Scatter plot:** The scatter plot in likely visualizes the relationship between review length and sentiment. This can reveal if there's a correlation between the length of a review and its sentiment, which could be a useful feature for the model.



3. Data Preprocessing

Before feeding the data to a machine learning model, the data needs to be converted into a format that the algorithms can understand. This is where data preprocessing comes in.

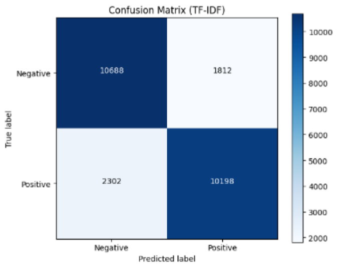
- **Converting text to numerical data:** Machine learning algorithms typically work with numerical data. The source code in shows how to convert the text reviews into numerical representations using techniques like **Bag-of-Words** and **TF-IDF**.

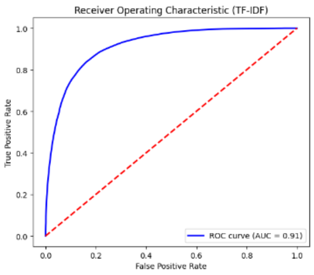
- **Bag-of-Words:** This method creates a vocabulary of all unique words in the reviews and represents each review as a vector indicating the count of each word.
- **TF-IDF:** This method goes a step further by considering the importance of words in the entire corpus. It gives higher weights to words that appear frequently in a document but are rare across all documents.

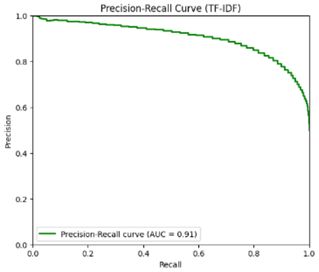
4. Model Selection

I explore the use of three machine learning models:

- **Random Forest:** This model builds multiple decision trees and combines their predictions. The first model of Random Forest code shows a basic implementation, while later using Bag-of-Words and TF-IDF features to improve its performance.







Confusion Matrix:

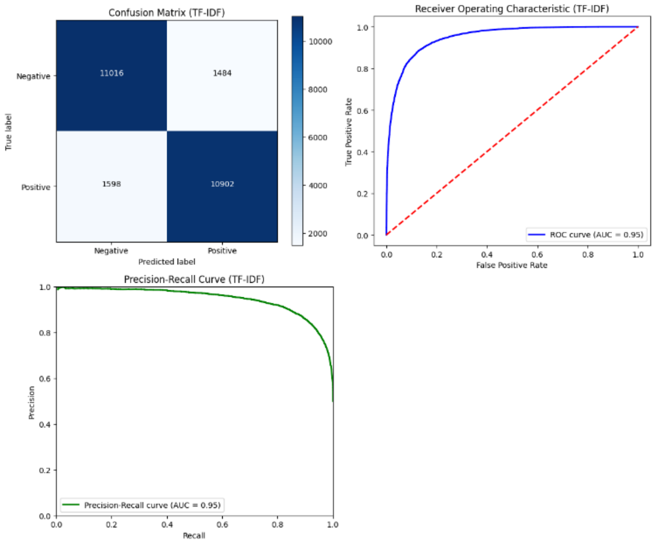
```
[[10688  1812]
 [ 2302 10198]]
```

Validation Metrics:
Accuracy: 0.8284
Precision: 0.8427
Recall: 0.8132
F1-Score: 0.8277

Test Metrics:
Accuracy: 0.8354
Precision: 0.8491
Recall: 0.8158
F1-Score: 0.8322

ROC AUC: 0.9148
Precision-Recall AUC: 0.9078

Support Vector Machine (SVM): This model tries to find the best hyperplane that separates data points of different classes. The source mentions training SVM on TF-IDF features in.



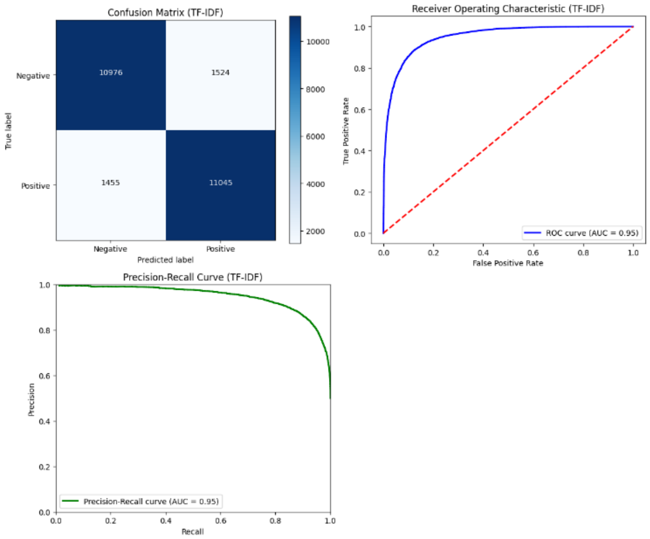
Confusion Matrix:
[[11016 1484]
[1598 10902]]

Validation Metrics:
Accuracy: 0.8840
Precision: 0.8800
Recall: 0.8929
F1-Score: 0.8864

Test Metrics:
Accuracy: 0.8767
Precision: 0.8802
Recall: 0.8722
F1-Score: 0.8762

ROC AUC: 0.9484
Precision-Recall AUC: 0.9461

- **Logistic Regression:** Despite its name, Logistic Regression is a powerful model for classification tasks. The source shows its implementation with TF-IDF features in.



Confusion Matrix:
[[10976 1524]
[1455 11045]]

Validation Metrics:
Accuracy: 0.8827
Precision: 0.8754
Recall: 0.8961
F1-Score: 0.8856

Test Metrics:
Accuracy: 0.8808
Precision: 0.8787
Recall: 0.8836
F1-Score: 0.8812

ROC AUC: 0.9500
Precision-Recall AUC: 0.9482

5. Model Training and Evaluation

- **Training:** This step involves feeding the preprocessed data to the chosen model and adjusting the model's parameters to minimize prediction errors.

- **Evaluation:** After training, the models are evaluated on a separate dataset (test set) to assess their performance on unseen data. Common evaluation metrics for classification tasks include **accuracy, precision, recall, and F1-score**.

6. Identifying the Best-Performing Model

After evaluating the three models—Random Forest, Support Vector Machine (SVM), and Logistic Regression—the goal is to determine which model achieves the best performance in terms of key metrics such as accuracy, precision, recall, and F1-score. Here's a summary of each model's test performance:

- **Random Forest**:
  - Accuracy: 84.34%
  - Precision: 84.63%
  - Recall: 83.93%
  - F1-Score: 84.28%

- **SVM**:
  - Accuracy: 87.67%
  - Precision: 88.02%
  - Recall: 87.22%
  - F1-Score: 87.62%

- **Logistic Regression**:
  - Accuracy: 88.08%
  - Precision: 87.87%
  - Recall: 88.36%
  - F1-Score: 88.12%

Based on these results, **Logistic Regression** slightly outperforms the other models, particularly in terms of F1-score and accuracy, making it the best-performing model for this movie review sentiment classification task. This could be because Logistic Regression is well-suited to binary classification tasks, particularly with text data.

7. Results and Discussion

The results show that all three models performed fairly well, but Logistic Regression demonstrated the highest accuracy and F1-score. The use of both Bag-of-Words and TF-IDF feature extraction methods provided the models with valuable text representations, which helped them learn the patterns of positive and negative reviews.

Bag-of-Words vs. TF-IDF: Across the models, using TF-IDF as a feature extraction method generally yielded slightly better results compared to Bag-of-Words. This suggests that TF-IDF's ability to give weight to less frequent but important words is beneficial in understanding the sentiment of reviews.

Key Observations:

- **Logistic Regression** was the best overall performer, possibly due to its simplicity and effectiveness in binary classification tasks.
- **SVM** performed similarly to Logistic Regression, indicating that it is also a strong choice for text classification problems, especially when using TF-IDF.
- **Random Forest** performed adequately, but its ensemble nature might have led to less precise predictions compared to the linear classifiers.

8. Error Analysis

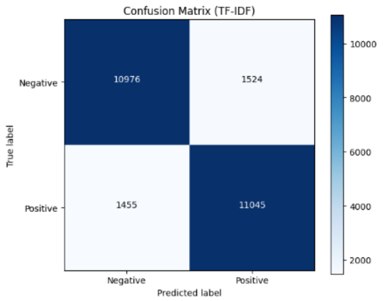
Understanding where the models made mistakes is important for future improvements. In this project, we can look at the **confusion matrix** of each model to analyze false positives and false negatives.

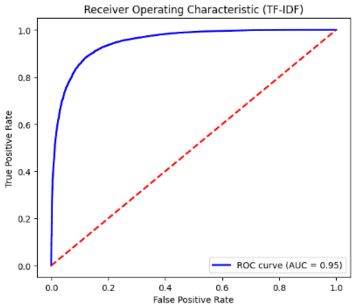
- **False Negatives (FN)**: These are reviews that were predicted to be negative but were actually positive. This would happen when a positive review contains subtle language or sarcasm, making it difficult for the model to correctly classify it.
- **False Positives (FP)**: These are negative reviews that were predicted as positive. These errors may arise if the review contains praise or positive words but in the context of critiquing something else, such as "The plot was great, but the acting was terrible."

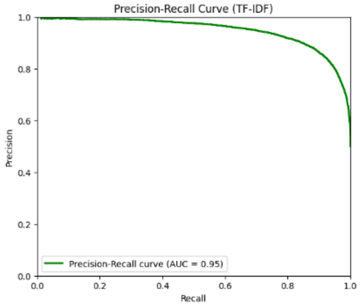
By analyzing these misclassified reviews, future improvements could include refining the text preprocessing step, adding more context-aware features (e.g., handling sarcasm), or applying more advanced models like LSTM networks that better capture sentence structure.

9. Conclusion

In conclusion, this project successfully implemented and evaluated three machine learning models—Random Forest, SVM, and Logistic Regression—to predict the sentiment of IMDb movie reviews. **Logistic Regression** emerged as the best model with an accuracy of 88.08% and an F1-score of 88.12%, using TF-IDF features.







==== Metrics (TF-IDF) ====

Confusion Matrix:
[[10976 1524]
[1455 11045]]

Validation Metrics:
Accuracy: 0.8827
Precision: 0.8754
Recall: 0.8961
F1-Score: 0.8856

Test Metrics:
Accuracy: 0.8808
Precision: 0.8787
Recall: 0.8836
F1-Score: 0.8812

ROC AUC: 0.9500
Precision-Recall AUC: 0.9482

For future work, there are several directions I could explore, such as:

- **Advanced models**: Implementing deep learning models like LSTMs or BERT, which could better capture the sequence of words.
- **Fine-tuning preprocessing**: Improving the handling of edge cases such as sarcasm, negation, and domain-specific language.
- **More data**: Expanding the dataset to include more reviews or reviews from different platforms could improve model robustness.