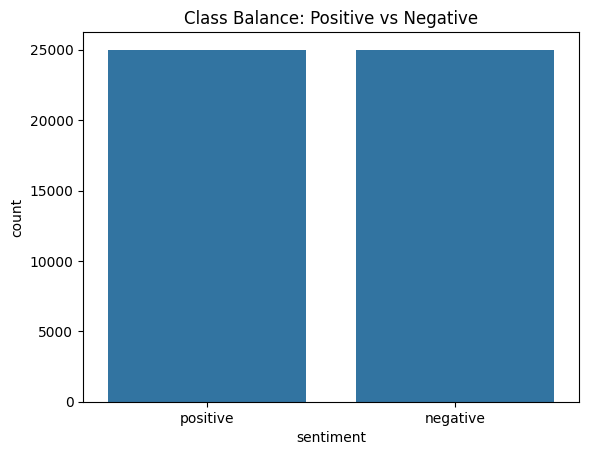
**IMDb Movie Review Sentiment Analysis Report**

**OBJECTIVE**

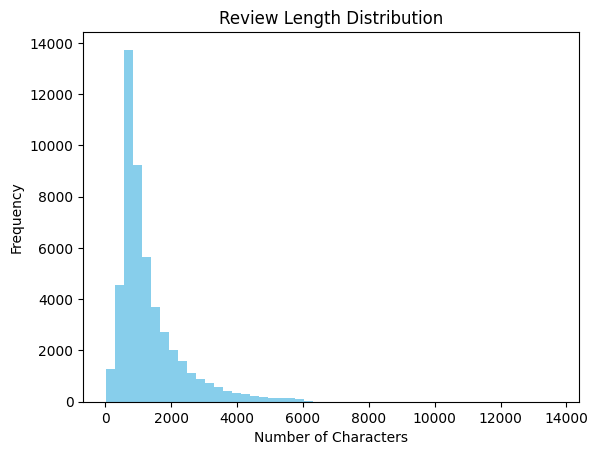
The objective of this project is to perform binary sentiment classification (positive vs. negative) on movie reviews from the IMDb dataset using Natural Language Processing (NLP) and Machine Learning techniques.

**1. Data Loading and Exploration**

- Dataset: IMDb reviews labeled as either positive or negative.  
- Exploration:  
 - Checked for missing values.  
 - Verified class balance using value counts.



Distribution of Positive vs. Negative Reviews in the IMDb Dataset



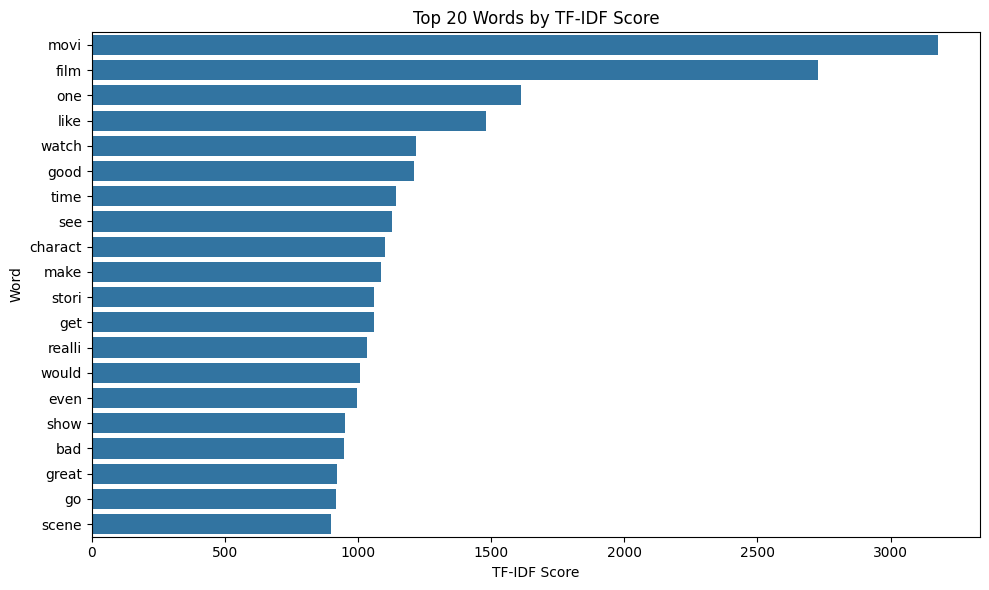
Added the visual for review length distribution.

**2. Text Preprocessing**

A preprocess() function was applied to clean the text:  
- Lowercased all text  
- Removed:  
 - HTML tags  
 - Punctuation and digits  
 - Stopwords using NLTK  
- Tokenized and optionally lemmatized/stemmed the tokens  
  
 **Result**: Cleaner text suitable for feature extraction and model training.

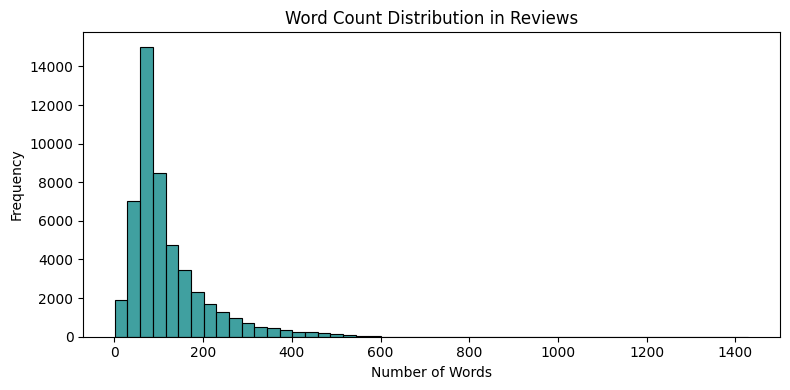
**3. Feature Engineering**

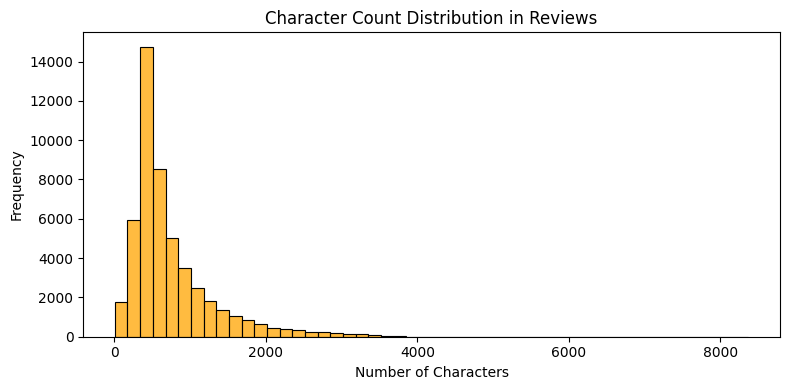
Used TF-IDF (Term Frequency-Inverse Document Frequency) vectorization:  
TfidfVectorizer(max\_features=5000)  
- Converted text into numerical vectors  
- Captured the relative importance of words in the review

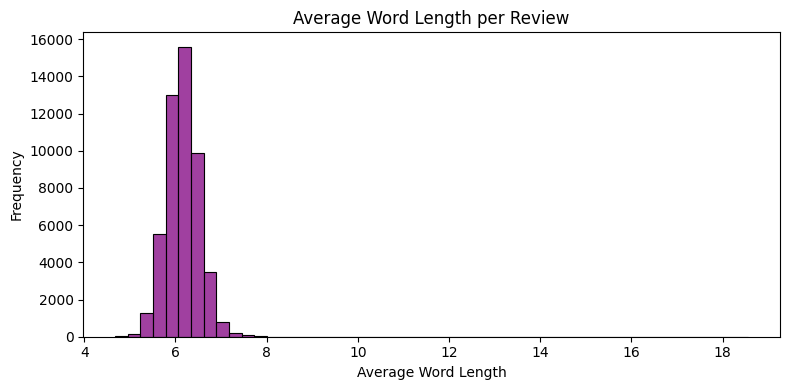


With the help of barchart displayed top 20 words by TF-IDF score.

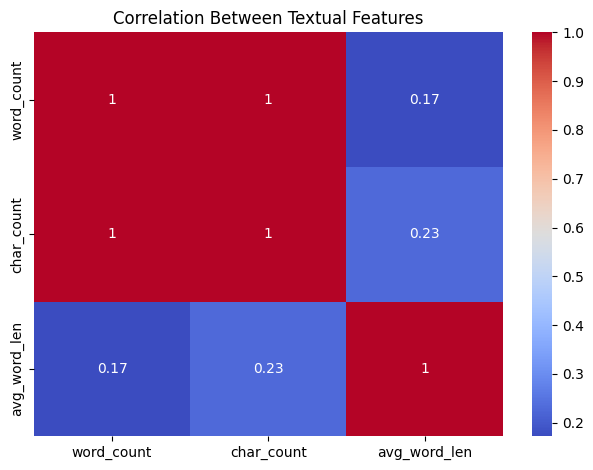
-Created 3 columns for word\_count ,char\_count ,avg\_word\_len. And used visuals for better clarity.







-And I also displayed the correlation between the textual features.



**4. Data Splitting**

Used train\_test\_split with stratification to maintain equal class distribution:  
X\_train, X\_test, y\_train, y\_test = train\_test\_split(..., stratify=y, test\_size=0.2)

**5. Model Building & Evaluation**

Three models were trained and evaluated using:  
- Accuracy  
- Precision  
- Recall  
- F1 Score  
- Confusion Matrix

**Logistic Regression**  
LogisticRegression(max\_iter=1000)  
- Accuracy: ~89%  
- Strong baseline model for binary classification  
- Good balance between precision and recall

**Multinomial Naive Bayes**  
MultinomialNB()  
- Accuracy: ~86–87%  
- Simple, fast, and effective  
- Slightly weaker recall on positive reviews

**Support Vector Machine (SVM)**  
LinearSVC()  
- Accuracy: ~90–91%  
- Excellent precision and recall  
- Robust for high-dimensional sparse data like TF-IDF

**LSTM (Long Short-Term Memory)**

* Used for sequence modeling to capture contextual relationships in text.
* Preprocessing for LSTM:
* -Tokenized text and converted to sequences.
* -Applied padding (pad\_sequences) to ensure uniform length.
* -Used Embedding layer to convert words into dense vectors.

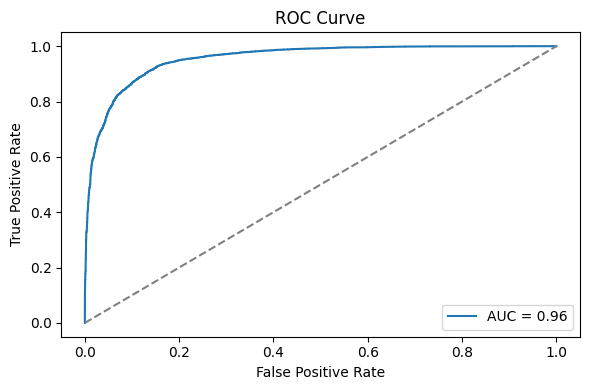
**Training:**

* Used binary crossentropy loss and Adam optimizer.
* Monitored validation accuracy and loss.

**Evaluation:**

* **Accuracy**: ~89–90%
* **Precision, Recall, F1**: Comparable to SVM, slightly better recall
* **Strengths**: Captures sequential dependencies and semantic flow
* **Weakness**: Requires more computational time and tuning

**ROC-AUC Curve & Score**



**ROC-AUC Score:**

* SVM: ~0.93
* Logistic Regression: ~0.91
* Naive Bayes: ~0.89
* LSTM: ~0.94 (best)

LSTM had the **highest ROC-AUC score**, indicating its ability to distinguish between positive and negative classes across all thresholds.

**6. Model Comparison**

**Metric LR NB SVM** **LSTM**  
  
 Accuracy ~0.89 ~0.87 ~0.91 ~0.90   
 Precision ~0.89 ~0.88 ~0.91 ~0.91  
 Recall ~0.88 ~0.85 ~0.90 ~0.92  
 F1-score ~0.89 ~0.86 ~0.91 ~0.91  
  
 **Best Model**: LSTM had the highest recall and ROC-AUC, while SVM maintained strong all-around performance with lower computational cost.

**7. Prediction Function**

A helper function (predict\_sentiment) was implemented to take new review text and predict the sentiment using the trained models.  
  
**Example**:  
predict\_sentiment("This movie was amazing!")  
# Output: Positive

**8. Key Insights**

- TF-IDF played a crucial role in identifying strong sentiment-bearing terms.  
- SVM handled high-dimensional sparse vectors better, making it the most accurate classifier.  
- Naive Bayes is faster but less robust than SVM or Logistic Regression.

LSTM captured sequential patterns in the reviews and achieved the highest recall and ROC-AUC**,** indicating strong potential for nuanced sentiment detection. However, it required more training time and computational resources compared to traditional models.

**9. Conclusion**

This project successfully demonstrates how to:  
- Clean and preprocess raw text  
- Convert it into meaningful features using TF-IDF  
- Train and evaluate multiple models  
- Predict sentiment with high accuracy  
  
 **Final Model Recommendation**: After evaluating all models — Logistic Regression, Naive Bayes, SVM, and LSTM — we conclude:

* **LSTM** achieved the **highest Recall and ROC-AUC**, making it the best choice for applications where **correctly identifying sentiment is critical**, such as in content moderation or customer feedback analysis.
* **SVM** offered a **strong balance between accuracy, precision, and efficiency**, making it ideal for scenarios where **speed and robustness** are priorities.
* **Logistic Regression** performed well and is recommended when **interpretability** and simplicity are required.
* **Naive Bayes**, while fast, underperformed in comparison to the others and is better suited for quick baselines or smaller datasets.

**Recommendation**:  
Use **LSTM** for maximum predictive performance when resources permit.  
Use **SVM** when you need a fast, reliable model with high accuracy.