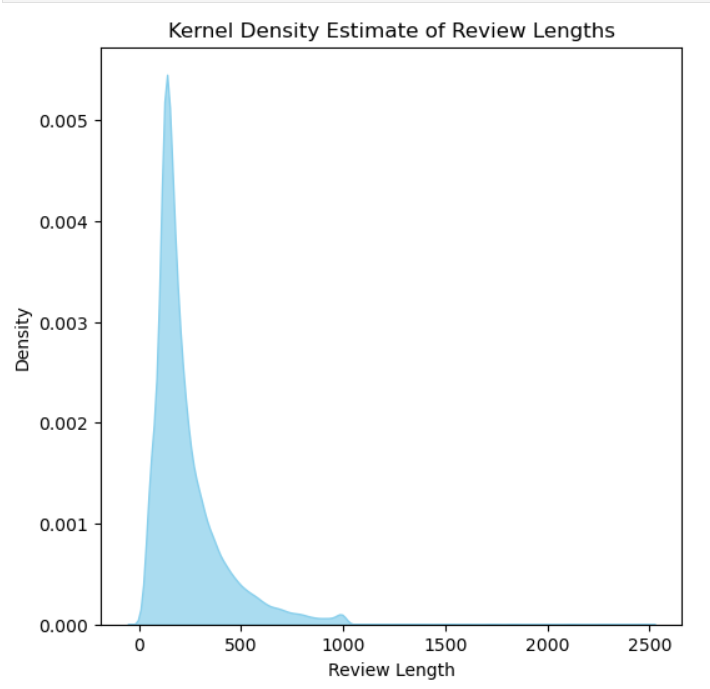
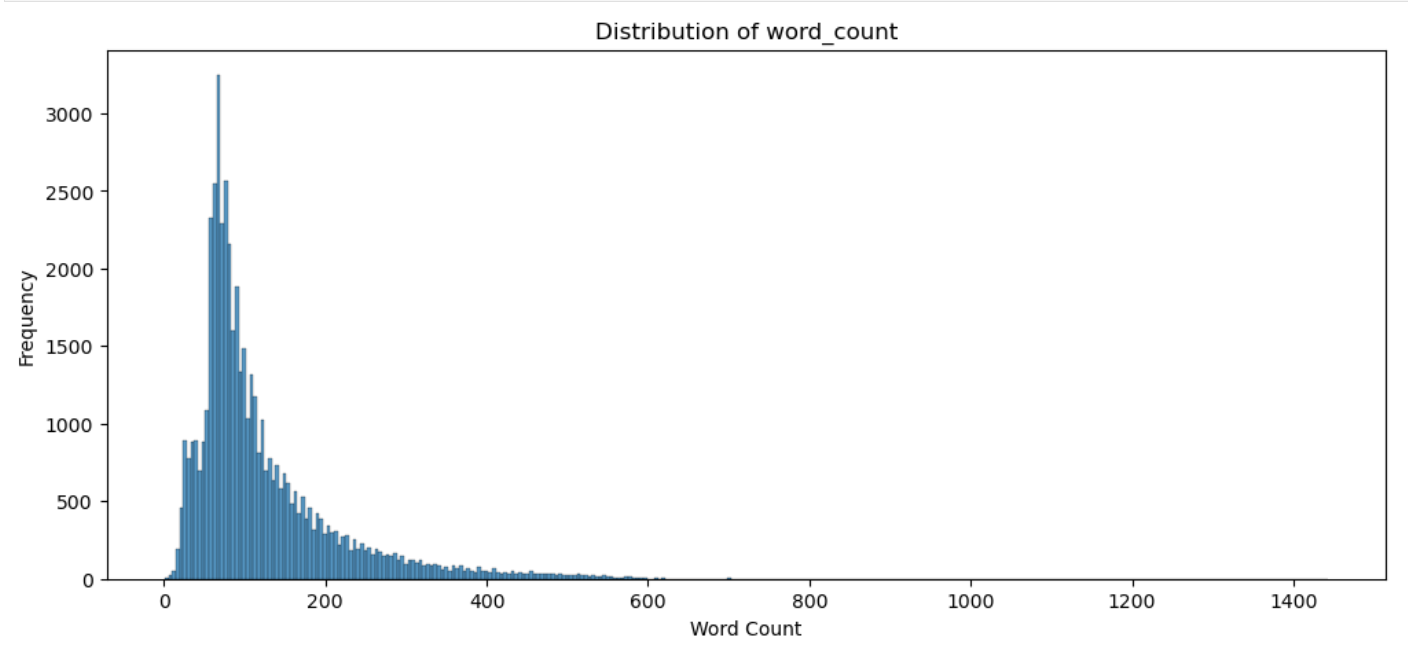
# Final Report on Part A: IMDb Movie Review Sentiment Analysis

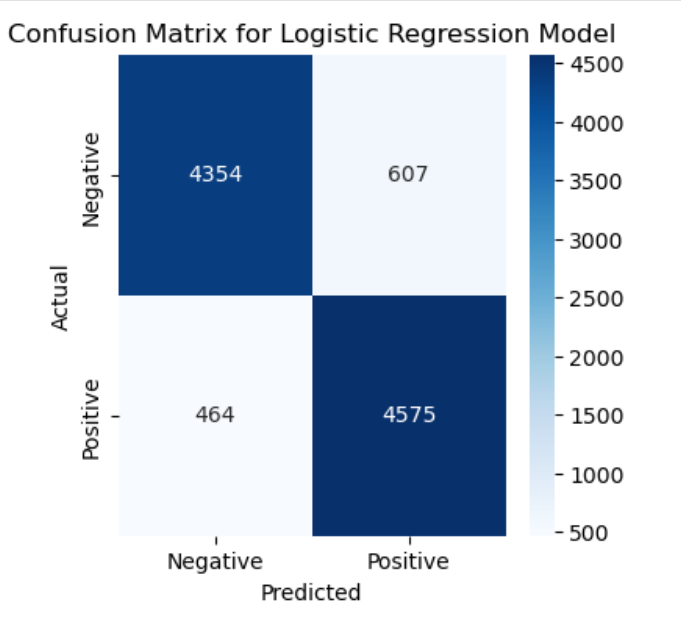
* Step 1: Import Libraries
* Initially imported all the required Libraries in Jupyter Notebook.
* Step 2: Data Exploration
* Then loaded dataset and created dataframe.
* When analyzed the dataset for missing values, found that there are no missing values.
* Dataset is a balanced dataset as number of positive and negative reviews are same.
* Created new column called review length.
* Have used Kernel Density Estimation plot for analysing review length column.



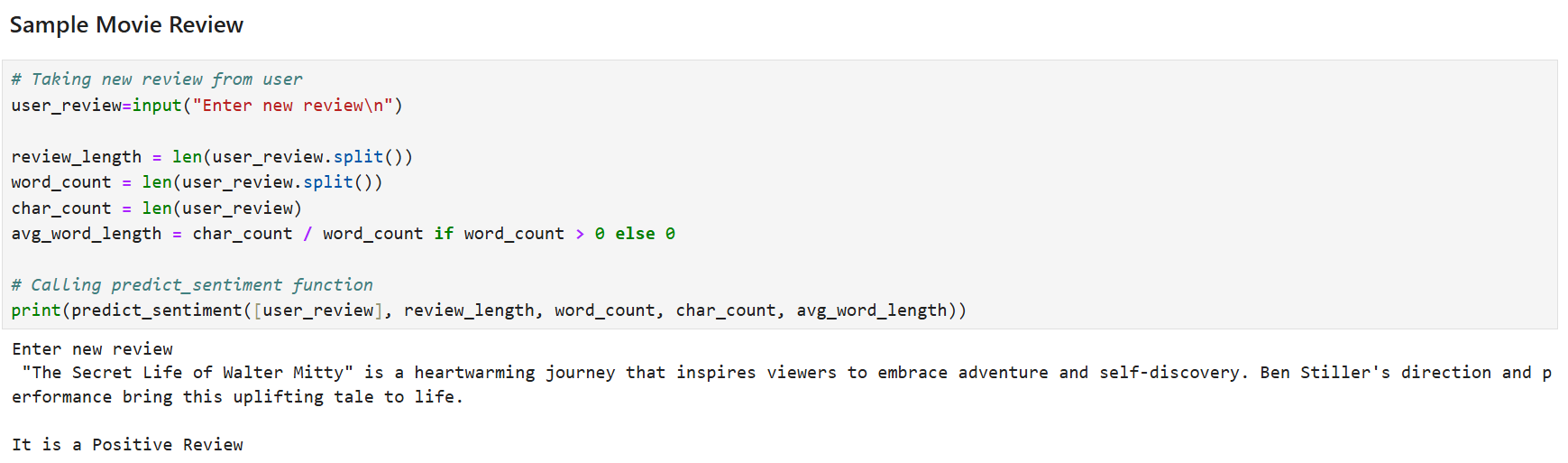
* Most of the reviews are of length between 50 to 450.
* Very few reviews have length above 500.
* The distribution is skewed to the right and shows positive skewness in the distribution of data.
* Step 3: Data Preprocessing
* Firstly, converted reviews to lowercase.
* Then created functions for removing punctuation and special characters from review column.
* Have created function to tokenize review column.
* Removed "not" word from stopword list. As it might have impact on negative review.
* Have created function to remove stopwords and extra spaces from review column.
* Then created function to lemmatize tokenised review. Because meaning of the words in the review column is crucial in this scenario.
* Finally joined the cleaned text into single string.
* Have applied all the above functions to the dataframe called imdb\_df.
* Step 4: Feature Extraction
* Applied label encoder to the sentiment column to get 1 for positive review and 0 for negative review.
* Created textual features like word count, character count and average word length using review column.
* Using histogram plot for word count column.



* Most of the word counts are between 20 to 250.
* Very few words have count above 400.
* The distribution is skewed to the right and shows positive skewness in the distribution of data.
* Fitted and Transformed review column using TF-IDF Vectorizer as we need to know the importance of the word in it.
* Step 5: Model Development
* Combined TF-IDF features with numerical features to create final dataset.
* Have split the dataset into 80% as train set and 20% as test set.
* With Default values, Logistic Regression Model performed well on test data with an accuracy of 86.57%.
* After hyperparameter tuning accuracy increased from 86.57% to 88.86%. We can consider this model during model evaluation as the accuracy value is highest among all other models.
* With Default values, Random Forest Model is performed well on test data with an accuracy of 84.98%.
* After hyperparameter tuning for Random Forest model, accuracy decreased from 84.98% to 84.58%. Since the accuracy value is less than logistic regression model, not considering this model during model evaluation.
* With Default values, SVM Model is performing very poorly on test data with an accuracy of 52.77%. So not performing hyperparameter tuning and not considering this model during model evaluation as the accuracy value is lowest among all other models.
* Step 6: Model Evaluation
* Among all the models, Logistic regression is giving highest accuracy. So, considering this model for evaluating the model performance.
* Accuracy is improved from 88.86% to 89.29% on unseen data when hyperparameter tuned values are considered. So, we can consider this model for predicting sentiment of a review in IMDB dataset.
* Step 7: Visualisation
* Generated confusion matrix on the predicted values of logistic regression model.



* True positive and True negative values are very close the total actual positive and total actual negative values.
* Out of 4961 Actual negative reviews, the model has predicted 4354 negative reviews correctly, indicating that the model is performing well on unseen negative reviews.
* Out of 5039 Actual positive reviews, the model has predicted 4575 positive reviews correctly, indicating that the model is performing well on unseen positive reviews.
* As the TP and TN values are high, it states that the model is reliable and can make very good prediction on future movie reviews.
* So, IMDB can rely on this model for its future movie reviews.
* Step 8: Prediction on user input
* Created pre-processing text function to preprocess the new review given by user.
* Created predict sentiment function to return the sentiment of the review.
* When user inputs new movie review, the sentiment of the review will be displayed.
* Example: Sample Positive Review



* Final Insights:
* 1. IMDB movie review dataset is a large dataset with 50k records of reviews and sentiment for each review.
* 2. Have trained the dataset with Logistic Regression, Random Forest and SVM algorithms.
* 3. Among these algorithms, Logistic Regression Algorithm works better as it has accuracy value of 88.86 percent.
* 4. This indicates that the model is able to predict the new or unseen data correctly 88.86 of times.
* 5. From confusion matrix, we observe a high value for true positives and true negatives. This implies that the model is correctly predicting positive review and negative review of IMDB.
* 6. So, we can completely rely on this model for future or unseen movie reviews of IMDB.