MOVIE REVIEW CLASSIFIER

### Project Details:

The main goal of the project is to develop a system that can automatically analyse and classify the sentiment expressed in movie reviews (positive and negative).

### Work Flow:

1. **Importing Libraries:**
   * Necessary libraries are imported for data manipulation, machine learning, and evaluation.
2. **Loading Dataset:**
   * The IMDB dataset is loaded into a pandas Data Frame. The head() method displays the first few rows of the dataset.
3. **Encoding Sentiment Labels:**
   * The Label Encoder is used to convert the sentiment labels (positive/negative) into numerical form (1/0).
4. **Checking Distribution of Encoded Sentiments:**
   * This line displays the count of each sentiment class in the dataset to check for class balance.
5. **Creating a Pipeline:**
   * A pipeline is created to streamline the process of vectorizing text data and training a classifier. The pipeline consists of two steps:
     + **CountVectorizer:** Converts text data into a matrix of token counts.
     + **RandomForestClassifier:** A random forest classifier with 5 trees and using entropy as the criterion for splitting.
6. **Splitting the Data:**
   * The dataset is split into training and testing sets. 80% of the data is used for training, and 20% is used for testing.
7. **Training the Model:**
   * The pipeline is trained using the training data (X\_train and y\_train). The fit method applies the CountVectorizer to the text data and then trains the RandomForestClassifier on the resulting numerical data.
8. **Making Predictions:**
   * The trained model is used to predict the sentiment labels for the test data (X\_test).
9. **Evaluating the Model:**
   * The accuracy of the model is calculated by comparing the predicted labels (y\_pred) with the true labels (y\_test). The accuracy score is printed.
10. **Importing Tkinter and Modules:**

* Tkinter is imported to create a graphical user interface (GUI) for user interaction.

1. **Creating the Main Application Window:**

* The main window of the application is set up.

1. **Creating Widgets (GUI Elements):**

* Various widgets such as labels, entry boxes, and buttons are created.

1. **Defining the Prediction Function:**

* process\_input() function retrieves the text entered by the user (entry.get()).
* Checks if there's any input. If not, it displays a warning message using messagebox.showwarning().
* If there is input:
* Uses a machine learning model (clf) to predict the sentiment (positive or negative) of the review text.
* Updates result\_label with the prediction result (Prediction: Positive or Prediction: Negative) and changes the text color (fg) based on the sentiment (green for positive, red for negative).

1. **Creating the Predict Button:**

* A button is created to trigger the prediction function.

1. **Running the Application:**

* The application is started with the main loop.

### 

### Challenges & Solutions:

 **Feature Extraction:**

* **Challenge:** Converting text data into a numerical format while retaining the contextual meaning.
* **Solution:** Use CountVectorizer for a Bag of Words model.

 **Model Overfitting:**

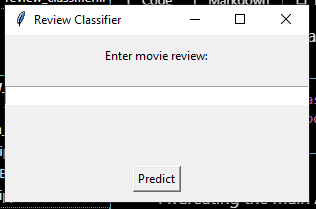
* **Challenge:** The model may perform well on training data but poorly on unseen data.
* **Solution:** Implement techniques such as cross-validation, pruning, or regularization to improve generalization.

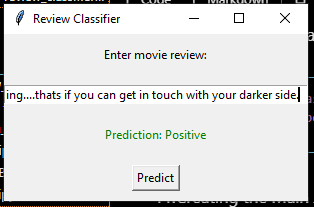
**Team:**

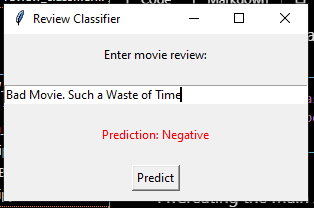
Vani Sugovind S.R

Siddharth

**Visuals:**







**References**

DATASET:

https://www.kaggle.com/datasets/lakshmi25npathi/imdb-dataset-of-50k-movie-reviews