ASSIGNMENT – 2 REPORT

1. Data Set

The dataset is taken from the **Enron Email corpus**, a widely used dataset for email classification. There are a total of 11,172 emails, of which **6,000 are spam** and the remaining **5,172 are ham** emails.

The dataset has a slightly imbalanced distribution, with a slight majority of emails labeled as spam. This imbalance is manageable and provides a realistic environment for training classification models without requiring significant adjustments for class distribution.

Each email is stored as a separate text file, and mostly contain standard email headers (e.g., Subject, Date, From) along with the body text, enabling diverse feature extraction approaches.

2. Data Preprocessing and Feature Extraction:

To prepare the text data for feature extraction, preprocessing steps were applied to each email to clean and standardize the input.

1. **Lowercasing**: All text was converted to lowercase to ensure uniformity.
2. **Removing Punctuation and Special Characters**: Non-alphanumeric characters were removed to remove noise.
3. **Stopword Removal**: Common words (like “and”, “the”) that do not provide discriminatory information were removed.
4. **Stemming/ Lemmatization**: Words were reduced to their root form, ensured that variations of a word are treated as the same feature (e.g. “running” and “run”).

These preprocessing step helped refine the text data by removing irrelevant components, leading to a reduced and more informative vocabulary.

After the initial data preprocessing, feature extraction step is taken into consideration. For representing the text data as numerical features, **TF-IDF(Term frequency-Inverse document frequency) Vectorizer** was employed as this vectorizer was chosen for its ability to highlight uncommon but important terms across emails. Both unigrams and bigrams were extracted to capture common word sequences found in spam (e.g., “limited offer”).

The dataset, being slightly imbalanced (6,000 spam vs. 5,172 ham emails), was divided into balanced training and validation datasets (using **stratified sampling**).

The final feature matrix incorporated all engineered features. Numerical features (e.g., email length, punctuation count) were scaled to ensure compatibility with the TF-IDF text features, providing a comprehensive, normalized feature set ready for model training.

I have chosen **“accuracy_score” as my metric** for judging the performance of my training and validation process. Since, the dataset is though imbalanced, but we have the balanced training and validation dataset obtained using stratified sampling, and accuracy score will fairly reflect the performance across both classes and can be seen a reliable measure of model effectiveness without need for complex metrics.

3. Algorithms Implemented:

I have first implemented **K-Nearest Neighbours** algorithm from scratch, and I tried to do cross-validation for tuning the k-value but, it was computationally time consuming given the dataset-size, so I used k-value as 5, and got an accuracy score of **96.2%** on validation set.

Then, I implemented **Naïve-Bayes classifier** from scratch, but it couldn’t give a good accuracy, ending up to around **47% accuracy**, showing that classifying the mails based on the probabilistic distribution and Bayesian approach couldn’t give a good prediction.

Followed by **Logistic Regression**, which gave a very good accuracy score of 89.4%, which is appreciable given the simplistic nature of logistic function and its decision boundary.

Also, I have tried **SVM versions of linear, polynomial, rbf and sigmoid kernel**, where all the kernels were giving almost similar accuracy score of around 97%, while SVM with rbf kernel gave the highest one (97.9%), since rbf(radial basis function) kernel captures the information present in all the degree of the features involved in input data given).

Finally, perceptron algorithm has also been implemented from scratch, and it also gave a good result, (outperforming naïve-bayes classifier) with an accuracy score of 96.4% on the validation set.

4. Conclusion:

Considering the best one among all the classifiers built, SVM(with RBF Kernel) is the best one with the highest accuracy score of 97.9%, I built my final function that picks the 'test' dataset in the current directory, and goes through each of the email text files, extracts the text and does the essential preprocessing, feature extraction, etc., (like what is done during the training phase), and predictions are made using previously trained SVM-RBF model.

Note: Since, I have trained my model on google colab, I kindly request you to do change the address of the input zip file which I used when you train on any other IDE, and similarly, change your 'test' folder address accordingly in the final function which I coded, so that the entire process is smooth and efficient.