REPORT

Q1. Simple Data Input and Pre-Processing

→ A number of functions have been implemented, such as pre_process and parse_dt_line. Using the parse_dt_line function, a second function is called convert_lbl, which further transforms the label categories into binary classes of 'REAL' and 'FAKE.

Q2. Basic Feature Extraction

→ Incorporated into feature_vector. Each token receives uniform features, which means weight equivalent to 1 is assigned to each word in the token, and global_feature_dict is updated with the total feature counts.

Q3. Cross-validation on the training dataset

→ This part implements a cross validation function that takes a dataset as its parameter and folds it. In this function, the step size is derived from the folds and used to loop through the dataset using the range method. The function is then improved further such that it can record each fold run's accuracy, precision, recall, and f1 score. The average of these recorded figures for the class "FAKE" is then returned. We found that the average accuracy was 0.56

Additionally, there were other indicators including f1score, accuracy, and precision in the range of 0.51

Q4.Error Analysis

→ We applied the model to the first fold of the Cross-Validation and got an accuracy of 0.57 and a recall of after some poor precision of 0.49 followed by recall of 0.50. We can infer from the first three false positives and negatives that the model lacks any morphological traits that would allow us to distinguish between "FAKE" and "REAL" more accurately.
Currently, the model was trained with a constant weight of 1 for each feature. This allows us to develop features to help the model distinguish between these two groups more effectively, but even with this basic model, it serves as a decent starting point.

Q5. Optimizing Pre-processing and feature extraction

→ The split and preprocess data in this maintains track of frequency and adds all the tokens from sentences. After determining the best value through trial and error, a threshold was determined. To improve outcomes, a TF-IDF Approach is utilised to determine a word's importance and use that number as a weight in cross validation.

Q6.Using Other Metadata in the File

→ In addition to integrating several characteristics into identifiers rather than statements and labels, the final code in the notebook demonstrates the TF-IDF methodology. To incorporate the new functionality, only a few functions are modified.
There are five characteristics: (1) total barely true counts, (2) total false counts, (3) total half-true-counts, (4) total pants-on-fire-counts, and (5) total mostly-true-counts.
According to the data in the notebook, the accuracy significantly improves. On the other hand, several features did not significantly improve the classifier.

The precision calculated was 0.60
Recall was 0.60
F1 score is approximately 0.61
Accuracy is 0.65-0.66