MLBA Assignment 1 Documentation

Group 44: Anurag Sista(2020495), Debjit Pramanik(2020504), Nikita Kapoor(2020531)

We have used amino acid compositions to generate features for the peptide strings that subsequently feed the dataset to our machine learning model. The final file of predictions is included in the zip folder submitted for this assignment.

Below are the steps that were used to arrive at our final results in Assignment 1:

1. Feature Engineering

   Feature engineering was responsible for the feature generation of our dataset that contained the peptide sequences. The amino acide composition of the given sequences is what was used in generating the features for the same.
   The input was the dataframe that contained the peptide sequences. It was passed by our function defined as ‘feature_engineering’ to return a dataframe that contained the new features, or amino acid composition that corresponds to our input data.
   The function defined as composition finds the composition of a peptide sequences by measuring its frequency of the amino acid and subsequently dividing it by the length of that sequence. It further appended the amino acid composition to its respective list along the amino acid compositions of the other sequences.
   The function takes a peptide sequence as its arguement.

2. Machine Learning Model

   We experimented with several models before finally using the Stacking Classifier to stack the Random Forest CLassifier along with the Logisitc Regression Classifier. The Logistic Regression Classifier was used to combine the Random Forest base estimator. For this, the function in our code is define as ‘machine_learning_model()’ and it returns a variable ‘model’.

   - Further Details:

     For the Stacking Classifier the base estimators were RandomForestClassifier and Logistic Regression and our final estimator was also Logistic Regression. Other parameters were set to their default settings. In the RandomForestClassifier, the number of trees used were 700 and for LogisticRegression which was our final estimator, the settings for the parameters were left at default.

3. Evaluating and fitting the model

   The function defined as ‘evaluate_and_fit_model’ takes the train features and the labels as the parameters. The train features refer to the trained amino acid composition features and training labels as separate dataframes. It gets the machine learning model and performs kfold cross validation on the model using accuracy as the scoring criteria and further displays the cross validation results for the same. It then fits the model on the whole of the training data such that it can be used to make predictions on the test dataset. The function returns the final fitted model.

4. Predictions

   After the final model has been fit again, we save it in a variable and use the predict function to give us our final values. The final predictions have been saved in a csv file and are included in the zip file that has been submitted for this assignment.

5. Final Score

   Our final score for the predicted values on the Kaggle leaderboard is 0.75844

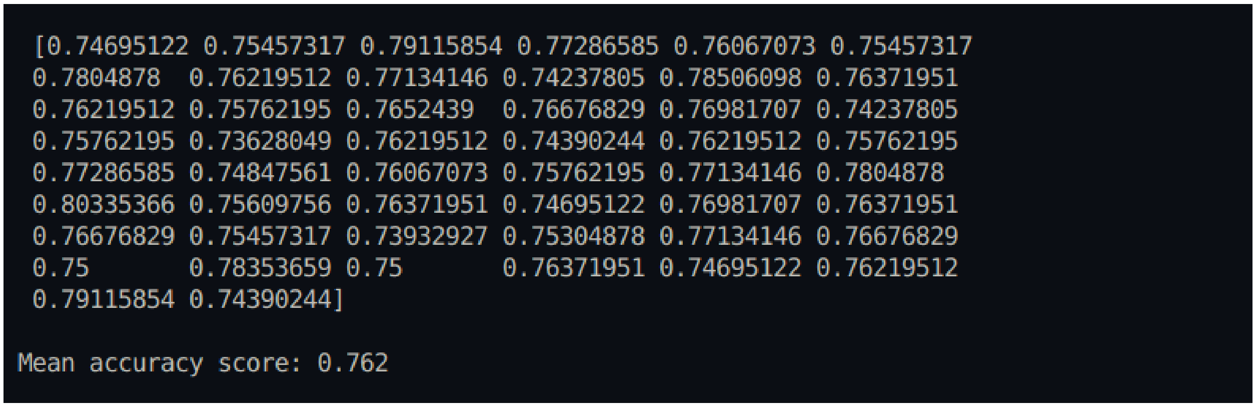
6. Libraries needed to run the python code

   - Numpy
   - Pandas
   - Re
   - Collections
   - Sklearn

7. How to run the code

   - Run the command - python3 Grp-44.py
   - The program will ask to input the train and test data file path. Please provide a path to the csv file trained dataset which contanis the peptide sequences.
   - Once the code is finished running, a CSV output file will be generated which contains the predictions that correspond to the input files.

Sample output on the terminal:



```
 [0.74695122 0.75457317 0.79115854 0.77286585 0.76067073 0.75457317
 0.7804878  0.76219512 0.77134146 0.74237805 0.78506098 0.76371951
 0.76219512 0.75762195 0.7652439  0.76676829 0.76981707 0.74237805
 0.75762195 0.73628049 0.76219512 0.74390244 0.76219512 0.75762195
 0.77286585 0.74847561 0.76067073 0.75762195 0.77134146 0.7804878
 0.80335366 0.75609756 0.76371951 0.74695122 0.76981707 0.76371951
 0.76676829 0.75457317 0.73932927 0.75304878 0.77134146 0.76676829
 0.75       0.78353659 0.75       0.76371951 0.74695122 0.76219512
 0.79115854 0.74390244]

Mean accuracy score: 0.762
```

About the Python program:

- **Input Files**
  The program takes paths to the training data as the user input. We have used the peptide sequences' files provided on kaggle as our training and testing data to find our predictions file
  Group_44_Predictions_Output.csv
  For training data: train.csv
  For testing data: test.csv

- **Output Files**
  The program outputs a csv file Group_44_Predictions_Output.csv containing the predicted
  class probabilities for the testing data.
  Prediction file submitted on Kaggle: Group_44_Predictions_Output.csv
  We have also included the submitted prediction file Group_44_Predictions_Output.csv in the zip folder.