Project Report

Calvin Wong | Mohammed Aaquib Hussain

CSC 4850

**Data Classification and Predictions**

The objective of classification is to identify which category a new observation belongs to, based on a training dataset. Using 5 data training sets and 5 label sets given to us, we trained two models to predict a label set for 5 data testing sets. For this project we chose to use jupyter notebook it has a vast amount of tools to handle large data sets such as pandas and numpy.

Some of the data sets had missing values in the place of 1.000000e+99. In order to produce accurate results these values needed to be replaced. We used tools in jupyter to replace 1.000000e+99 with nan, calculated the percentage nan values, then replaced nan with values using KNN to input data with k=2.

* Data set 1 had 1.99% missing data
* Data set 3 had 2.13% missing data

To build and evaluate our machine learning solution, we selected two classification algorithms, SVM and KNN. These models were trained on a portion of the dataset and then evaluated on a separate test portion. To determine which model performed better, we compared their performance using four key evaluation metrics.

* Accuracy: This measures the percentage of correctly predicted labels out of all predictions.
* F1 Score: This combines precision and recall into a single metric, balancing the trade-off between false positives and false negatives, especially in cases of imbalanced datasets.
* Precision: This measures how many of the predicted positive labels are actually correct
* Recall: This evaluates how many of the actual positive labels were correctly identified by the model.

Based on these metrics, we selected the model that provided the best overall performance to predict the labels for the test dataset. By comparing multiple metrics rather than relying on just one, we ensured a more robust evaluation of the models' capabilities, particularly in situations where one metric alone might not give a complete picture of performance.

* For data set 1 we used SVM
* For data set 2 we used SVM
* For data set 3 we used KNN
* For data set 4 we used SVM
* For data set 5 we used KNN

After figuring out which model worked best with our data, we used it to predict the labels for the test data. This test data was new to the model, so running these predictions helped us see how well the model could handle new information and make accurate guesses.

**Missing Data Value Estimations**

The objective of missing data value estimations is to estimate missing values in the Microarray Data. Given 3 Datasets with missing values with Dataset 1 missing an estimated amount of 4%, Dataset 2 missing an estimated amount of 10%, and finally Dataset 3 having an estimated amount of 82% of missing data. To accomplish our goal of replacing the missing values we first have the data frames converted to csv to then be read on to as a starting point reference of what the data looked like before we modified and adjusted. We then imported tools that would be beneficial in aiding in resolving the missing data values. The standard imports of pandas and numpy followed that would be the usage of KNN imputer which would help in filling the missing data later down. The next process was to gather the exact number of NaN entries found in each file and the exact percentage. Then using the KNN imputer with 2 of its nearest neighbors to fill in each of the missing values and produce an estimated value of what the value could be based on their neighbors. We set the value of nearest neighbors to 2 as it produced the best results and was the best possible method for filling in the data as each dataset had more data missing than the last. This ensured that we had all missing NaN values filled in and that no lapse due to NaN nearby would cause an error when processing the new values. After we then had the data fitted and transformed into the dataset before converting it back into the data frame. Finally, we printed the results of the new data frame after the KNN imputation.