Report - Machine Learning Developer Test Assignment

This report gives information about how to reproduce the experiment and describes the steps taken in the development process.

How to reproduce the experiment:

- First and foremost it is necessary to create a virtual environment python.
- Installation of the libraries in the “requiriments.txt” file.
- Select the kernel with the virtual environment.
- Run the Jupyter Notebook file.

After running the experiment the following files will be generated: (classification report of knn cosine, classification report of knn euclidean, results.txt and ROC_curve.png)

Steps taken to develop the experiment:

1. Cleaning and data rearrangement.

   The first step was to take the given dataset and transform the data, performing a rearrangement. The most important information was selected, such as “syndromes ids” and the “320x1 encoding” of each image.

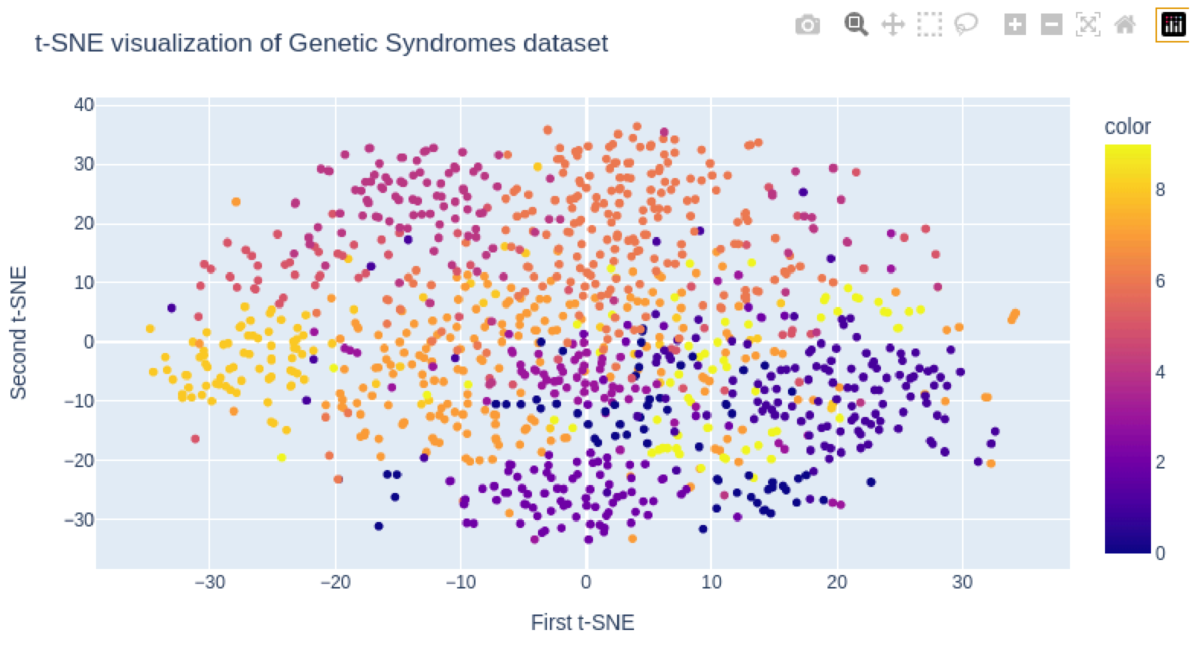
   The following process was to create a new dataframe relating each image encoding to a syndrome. Then, the “320x1 encoding” of each image was transformed into 320 features, then a new dataframe was created, containing 320 columns representing the features, and the last column containing the “syndrome id” related to the features.

2. Plotting t-SNE.

   The second step was the plotting of the t-SNE tridimensional graph. Through the graph we have a good representation of the data information, and a visualization of how the features relate to each other.

   We can see clusters forming. The features closer to each other tend to represent a specific “syndrome id”.

   The color column on the far right represents the “syndromes ids”, which are the target variables of the dataset.



3. Cross validation.

   In this step cross validation was performed with the intent to find the best value for K which was used for the K-Nearest Neighbor Classifier. The Grid Search method was the technique choice. Both KNN with the cosine distance and euclidean distance was run under the Grid Search. Below we see the best value for K in both KNN models:

KNN Cosine distance Top-k	KNN Euclidean distance top-k
12	24

4. The next step was to develop the classification of the two models. The dataset was divided in the training and test set through the train_test_split of the sklearn. The values chosen were 75% for the training set and 25% for the test set.

   The performance of both models were evaluated through the metrics (precision, recall, f1-score) generated through the classification report of the sklearn library.

   The results are shown below:

KNN Cosine distance model classification report

```
        precision    recall  f1-score   support

   0       0.59       0.56      0.57        18
   1       0.75       0.87      0.81        31
   2       0.88       0.92      0.90        25
   3       0.60       0.35      0.44        17
```

```
          4       0.79      0.73      0.76        26
          5       0.80      0.63      0.71        19
          6       0.76      0.93      0.84        55
          7       0.70      0.81      0.75        47
          8       0.83      0.83      0.83        23
          9       0.71      0.28      0.40        18

   accuracy                           0.75       279
  macro avg       0.74      0.69      0.70       279
weighted avg       0.75      0.75      0.74       279
```

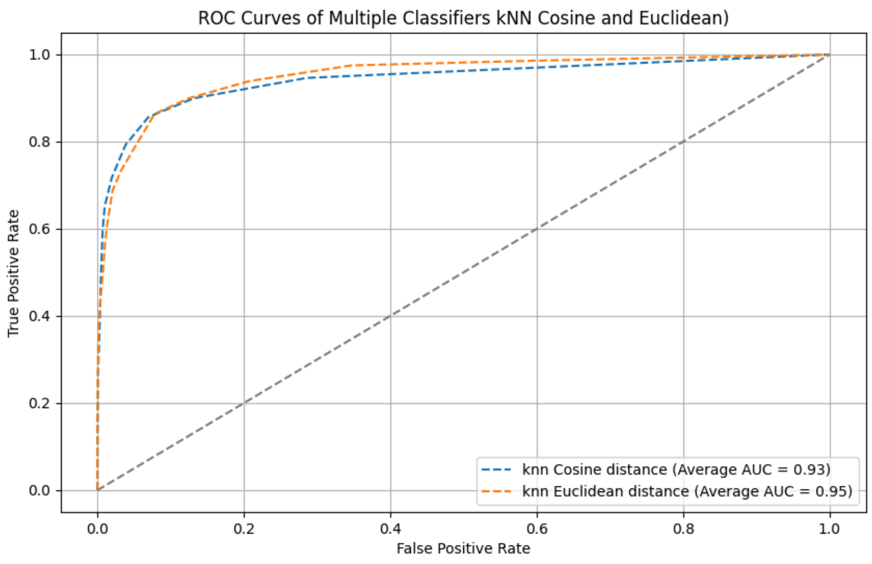
```
KNN Euclidean distance model classification report
            precision    recall  f1-score   support

          0       0.67      0.56      0.61        18
          1       0.72      0.84      0.78        31
          2       0.91      0.84      0.88        25
          3       0.53      0.59      0.56        17
          4       0.78      0.69      0.73        26
          5       1.00      0.42      0.59        19
          6       0.71      0.93      0.80        55
          7       0.66      0.81      0.72        47
          8       0.90      0.83      0.86        23
          9       0.75      0.17      0.27        18

    accuracy                           0.73       279
  macro avg       0.76      0.67      0.68       279
weighted avg       0.75      0.73      0.72       279
```

5. The last step was the generation of the AUC scores of both models. In order to generate the ROC AUC graph some strategies were studied given the fact the dataset is multiclass.

   The best two features that represent each model were chosen, then the graph was generated.


ROC Curves of Multiple Classifiers kNN Cosine and Euclidean)
True Positive Rate
False Positive Rate
knn Cosine distance (Average AUC = 0.93)
knn Euclidean distance (Average AUC = 0.95)