# Apollo ML Project

## Exploratory Data Analysis

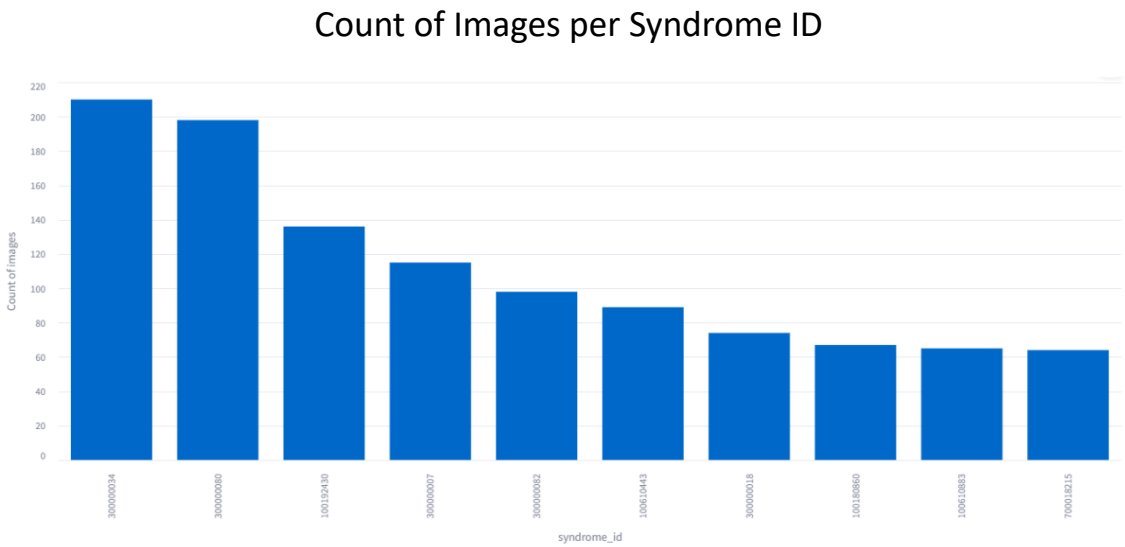
Lets start by going over the distribution of the syndromes and do some validation check (e.g. null values).

Missing Values

None ✅

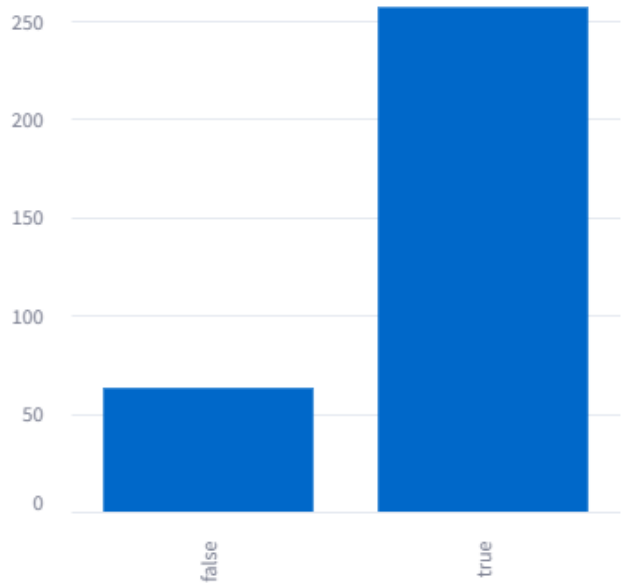
## Distribution of Syndrome

The distribution of the syndromes is as follows:




As we can see the distribution of syndromes in the dataset is not heavily imbalanced.

## Checking the normality of the embeddings using Shapiro Test




| Normality | Percentage of Gaussian Data |
|---|---|
| ☑ | 0.8031 |
| ☐ | 0.1969 |

## Exploring Dimensional Reduction with T-SNE

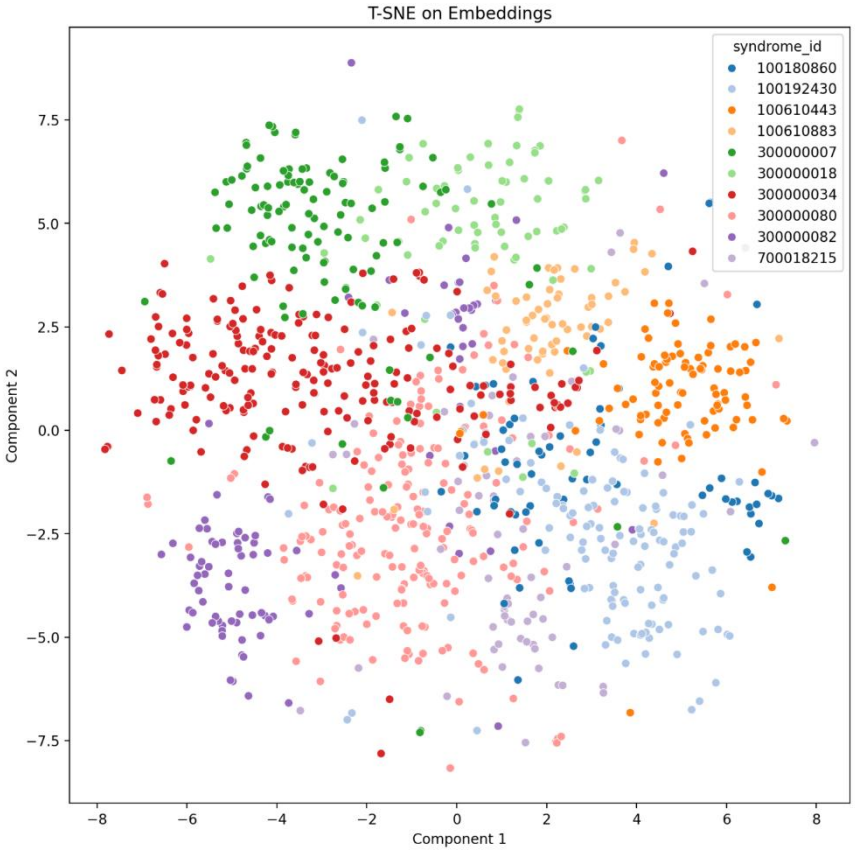
t-SNE is useful for capturing relationships among high-dimensional data points while reducing dimensionality, preserving their proximity.

## Scalling Data

First, the embedding data was scaled to a standard distribution with a mean of zero and a standard deviation of one.

| | d_1 | d_2 | d_3 | d_4 | d_5 | d_6 | d_7 | d_8 | d_9 | d_10 | d_11 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| count | 1116 | 1116 | 1116 | 1116 | 1116 | 1116 | 1116 | 1116 | 1116 | 1116 | 1116 |
| mean | 00000000 | 00000000 | 00000000 | 00000000 | 00000000 | 0 | 00000000 | 00000000 | 00000000 | 00000000 | 00000000 |
| std | 1.0004 | 1.0004 | 1.0004 | 1.0004 | 1.0004 | 1.0004 | 1.0004 | 1.0004 | 1.0004 | 1.0004 | 1.0004 |
| min | -2.9511 | -3.2277 | -3.3985 | -4.2162 | -3.5218 | -3.8216 | -3.789 | -3.4505 | -3.5642 | -2.9015 | -3.5065 |
| 25% | -0.7133 | -0.6776 | -0.7372 | -0.6293 | -0.6216 | -0.6691 | -0.6525 | -0.6909 | -0.7183 | -0.7337 | -0.6449 |
| 50% | -0.0139 | 0.0112 | -0.0595 | -0.007 | -0.0334 | -0.0145 | -0.0048 | -0.0158 | 0.0395 | 0.0247 | 0.0158 |
| 75% | 0.6668 | 0.6372 | 0.6888 | 0.6616 | 0.6504 | 0.6622 | 0.6808 | 0.6936 | 0.6458 | 0.6788 | 0.7035 |
| max | 3.4374 | 3.8668 | 3.2208 | 4.1715 | 3.6972 | 3.001 | 3.4698 | 2.6717 | 3.5537 | 3.0771 | 2.897 |

After the reduction of dimensionality to 2-D it is possible to plot the datapoints and visualize possible correlation of the embeddings of the same syndrome.



The proximity of data points within the same syndrome suggests a correlation between the embedding and the syndrome category. Despite the presence of outliers, this further confirms that the data distribution exhibits long-tailed characteristics.

## K- Nearest Neighbors Classifier (KNN)

To the task of predicting the syndrome using the embeded image there is a model

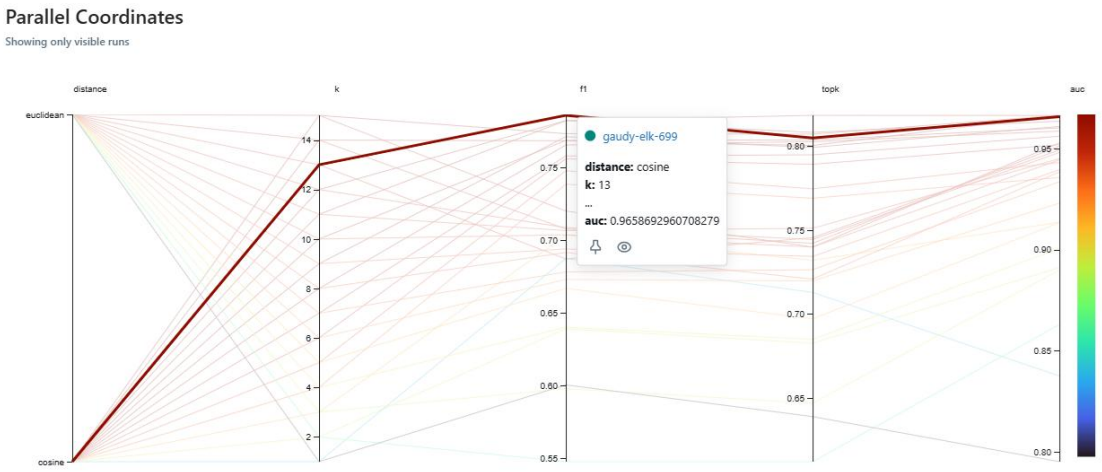
To optimize the KNN Classifier we can try out the combination of hyperparameters such as:
- Metric of distance: Euclidean and Cosine.
- Number of neighbors: 1 to 15.

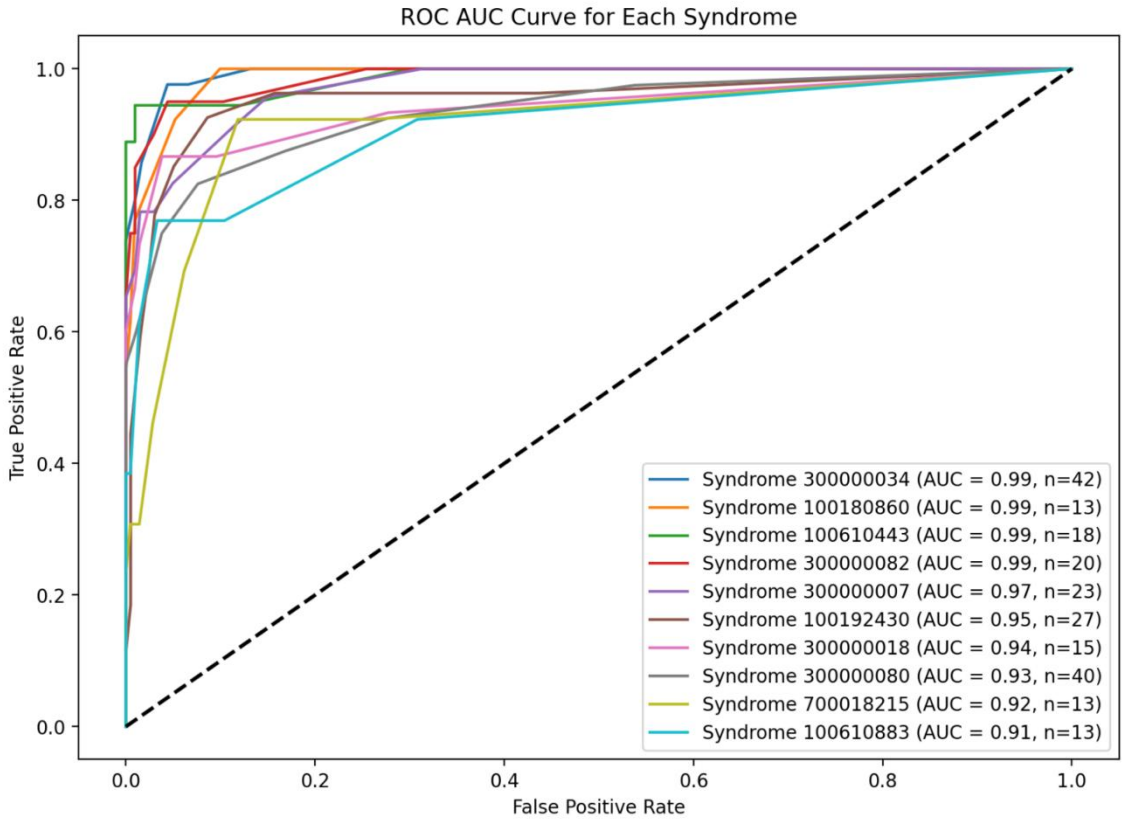
The distance change how the magnitude and dimensionality of the data impacts the model. Euclidean distance are very sensitive to scale and high-dimensionality of the data. Cosine distance is more robust since it focus on the angle of the vectors. Moreover, there is another import hyperparameter for KNN model.
The number of neighbors is inversionally proportional to the overfitting of the model.
The smaller K number, more prone to high variance (overfit), leading the model to capture every noise in the data, memorizing the training data.

Let us see the result of the different combinations.

| | distance | k | ↓ f1 | topk | auc |
|---|---|---|---|---|---|
| 27 | cosine | 13 | 0.7857 | 0.8049 | 0.9659 |
| 25 | cosine | 11 | 0.7821 | 0.8082 | 0.9636 |
| 26 | cosine | 12 | 0.782 | 0.8183 | 0.9664 |
| 29 | cosine | 15 | 0.7731 | 0.8071 | 0.9665 |
| 23 | cosine | 9 | 0.771 | 0.7992 | 0.961 |
| 24 | cosine | 10 | 0.7684 | 0.8037 | 0.9603 |
| 28 | cosine | 14 | 0.7682 | 0.8004 | 0.9664 |
| 22 | cosine | 8 | 0.7653 | 0.8026 | 0.9587 |
| 21 | cosine | 7 | 0.7578 | 0.7948 | 0.9561 |
| 20 | cosine | 6 | 0.7558 | 0.7892 | 0.952 |

The best model was with k=13 and cosine distance, achieving the following performance metrics:
Overall:
F1-score: 0.786
Top 1-k accuracy: 0.805
ROC AUC: 0.966
As expected, the Cosine Distance and higher k performed better. The model was trained with 80 percent of the data, stratified by the syndrome label to achieve higher accuracy for the smaller classes. The remaining data will be used for final testing, validating the model and helping us visualize the ROC curve for each syndrome.




## Conclusion

To conclude, since the embedding probably was extracted using CNN pre-trained model such as RESNET, the semantic information in the vectors helped to improve the model using cosine distance to achieve acceptable results.

What can be further exploring is to try out a pre-trained model with a output layer for classification task.

Moreover, there are other types of classifiers that are also robust against high-dimensionality such as LightGBM and XGBoost.