APOLLO SERVIÇOS EM TECNOLOGIA DE INFORMAÇÃO E INTELIGÊNCIA ARTIFICIAL LTDA

MARIA ELISA FERNANDES OCTAVIANO

MACHINE LEARNING DEVELOPER TEST ASSIGNMENT

2024

SUMÁRIO

1. INTRODUÇÃO

O presente relatório tem o objetivo de trazer algumas considerações gerais bem como discussões a respeito do *Machine Learning Developer Test Assignment* que consiste no seguinte:

Goal

The purpose of this exercise is to simulate a real-life problem that a Machine Learning Developer will handle while working with our team.

We want the candidate to “feel” one of the research domains the company is exploring, and we want to be able to assess the technical/coding skills of the candidate.

Guidelines

- The output for this exercise should be Python file/s (.py) + .pdf report that summarizes the work with the different steps taken, insights and instructions on how to reproduce the experiment and visualization (if any is relevant)
- You have 5 days to deliver the test via e-mail with both files.
- Feel free to email guilherme.marchini@apollosolutions.dev for any questions you have during the exercise.

Exercise

- You will be provided with a pickle file that contains all the necessary data. They are embeddings from a classification model.
- The classification task was to classify the genetic syndromes (syndrome_id) of a given image.
- The structure of the dictionary saved in the pickle file:
- {'syndrome_id': { 'subject_id': {'image_id': [320x1 encoding]}}}
- If you get the "numpy.core._multiarray_umath" error when loading the pickle file, please upgrade your numpy package.

- The steps to perform:

a) Plotting tSNE of the inputs, explaining the statistics and the data

b) Do a 10-fold cross validation for the following steps:
   - Calculate cosine distance from each test set vector to the gallery vectors
   - Calculate euclidean distance from each test set vector to the gallery vectors
   - Classify each image (vector) or each subject to syndrome Ids based on KNN algorithm for both cosine and euclidean distances.

c) Create automatic tables in a txt / pdf file for both algorithms, to enable comparison (please specify top-k, AUC etc.)

d) Create an ROC AUC graph comparing both algorithms (2 outputs in the same graph, averaged across gallery / test splits)

Bonus: Create 1-2 simple unit tests to your choice (preferably with pytest or similar).

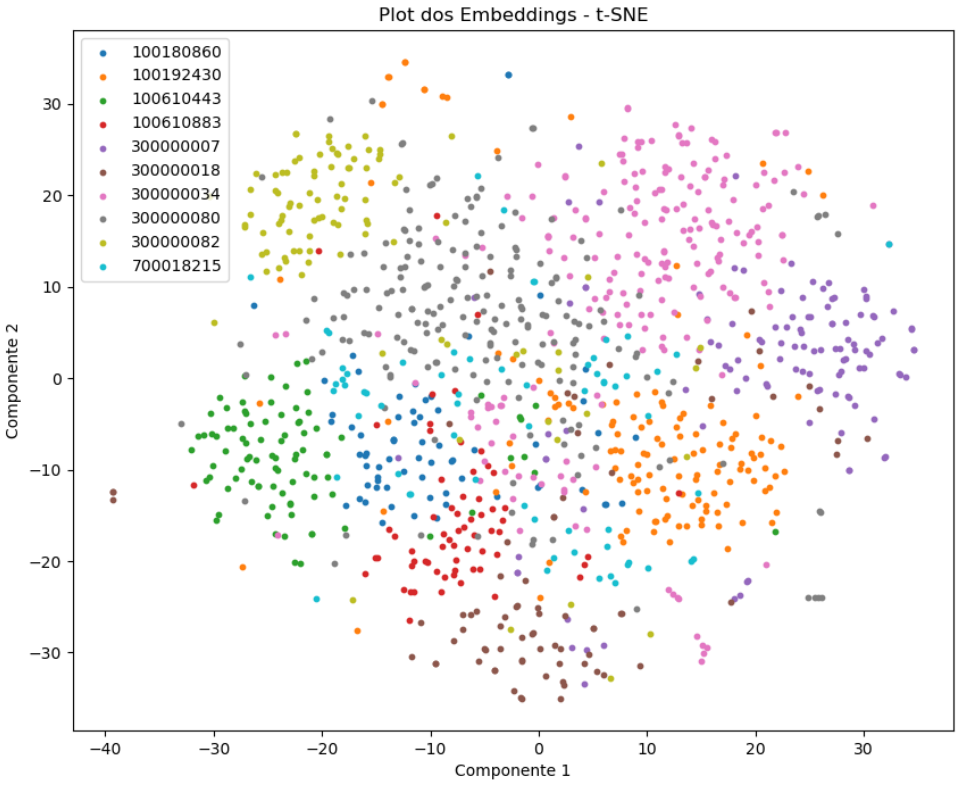
Please delete any local data copy (if any) after sending the solution.

Good luck!

2. EXERCÍCIO A

Neste exercício, foi solicitado um gráfico de dispersão resultante de uma redução de dimensionalidade usando o método t-SNE (t-*distributed Stochastic Neighbor Embedding*). No contexto de classificação de síndromes genéticas a partir de imagens, a técnica foi utilizada como uma forma de compactar informações, melhorando sua visualização. O resultado da implementação é mostrado na Figura 1, que será discutida a seguir.

Figura 1: Plot dos embeddings usando o método t-SNE



Diante da observação da Figura 1, pode-se concluir que, não há clusters distintamente separados e que são imediatamente visíveis. Isso porque os pontos estão espalhados de forma relativamente uniforme por todo o espaço.

No contexto, a falta de clusters pode indicar que as síndromes possuem características visuais muito semelhantes ou que os embeddings não estejam capturando bem as diferenças entre elas.

Além disso, ressalta-se que os eixos "Componente 1" e "Componente 2" são as duas principais dimensões resultantes da redução de dimensionalidade e que eles não têm uma unidade de medida específica, mas representam direções de maior variação nos dados originais.

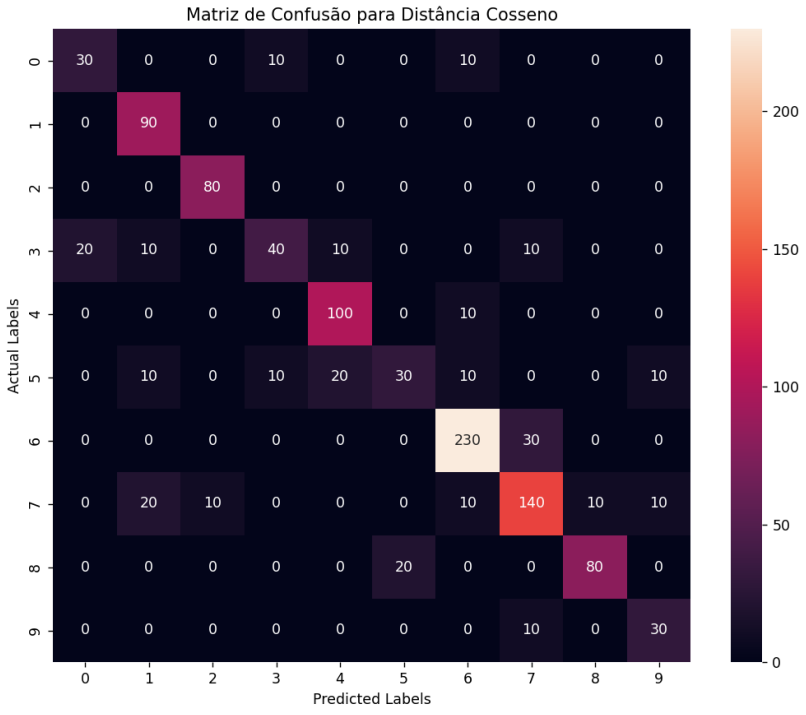
Conclui-se que a atual distribuição dos pontos pode afetar a capacidade de um modelo de machine learning em classificar corretamente as síndromes genéticas e, por isso, são sugeridas investigações adicionais como a utilização de outras técnicas de redução de dimensionalidade (como PCA ou UMAP), ou explorar métodos de clusterização (como K-means ou DBSCAN) para ver se algum padrão emerge.

3. EXERCÍCIO B

Neste exercício foi solicitada a implementação de uma validação cruzada de 10 folds que calcula as distâncias cosseno e euclidiana entre vetores de teste e vetores de uma galeria, e classifica cada imagem ou sujeito para os IDs de síndrome usando KNN.

Para melhor visualização dos resultados da classificação, a matriz de confusão foi plotada. Assim, pode-se avaliar mais facilmente a precisão da classificação para cada classe. Nas Figuras 2 e 3, são mostradas as matrizes de confusão resultantes da classificação utilizando distância cosseno e distância euclidiana, respectivamente.

Figura 2: Matriz de Confusão para a Distância Cosseno

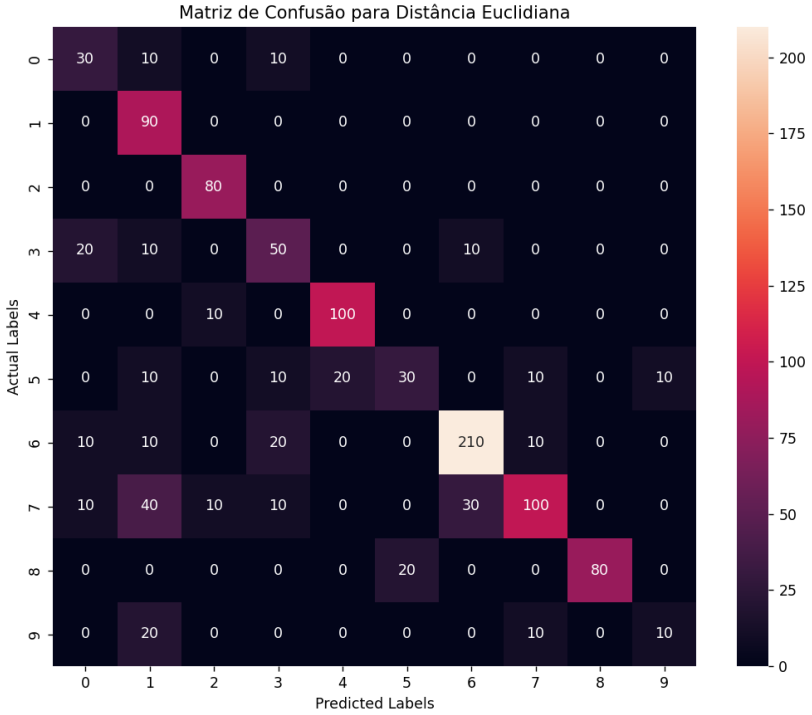


Diante da observação das figuras e sabendo-se que as matrizes de confusão mostram o número de previsões corretas e incorretas feitas por um classificador, quanto mais altos os valores e mais concentrados eles estiverem na diagonal, melhor o classificador está em prever corretamente cada classe. Em contrapartida, valores mais baixos fora da diagonal indicam menos erros de classificação para outras classes.

Tendo em vista que as matrizes de confusão são de vários folds de validação cruzada, para melhor avaliação do modelo, é recomendado calcular as métricas de

desempenho como precisão, recall e F1-score e assim comparar o desempenho geral entre as métricas de distância.

Figura 3: Matriz de Confusão para a Distância Euclidiana



4. EXERCÍCIO C

Neste exercício foi solicitada a criação de tabelas automáticas que resumem as métricas de desempenho de classificação e exportação das informações para um arquivo .txt ou .pdf.

Para isso, foram calculadas as métricas de interesse como acurácia, precisão, recall e F1-score. Como resultado, foram obtidas as informações mostradas na Tabela 1.

Tabela 1: Desempenho das distâncias cosseno e euclidiana

Accuracy	Precision	Recall	F1-Score
0.7657657657657657	0.7601558373488199	0.7657657657657657	0.7543206220290162
0.7027027027027027	0.7306062800799644	0.7027027027027027	0.6964975387374095

Na implementação realizada, ressalta-se que a primeira linha da tabela corresponde à distância cosseno e a segunda linha corresponde à distância euclidiana.

Comparando as duas linhas de números, percebe-se que o modelo utilizando a distância cosseno superou o modelo utilizando a distância euclidiana em todas as métricas.

Portanto, para este conjunto de dados e tarefa específicos, a distância cosseno pode ser a métrica de distância mais adequada para o modelo de classificação de síndromes genéticas.

5. EXERCÍCIO D

Neste exercício foi solicitada a implementação de um gráfico que comparasse as implementações das distâncias cosseno e euclidiana. O gráfico é mostrado na Figura 4. As áreas sobre as curvas de cada modelo foram, respectivamente, 0.94 e 0.92.

Nesta implementação, em conformidade com os resultados anteriores, pode-se perceber que a distância cosseno performou ligeiramente melhor que a distância euclidiana.

Figura 4: Matriz de Confusão para a Distância Euclidi