Universidade Federal da Paraíba

Discente: Nathan Carlos de Macena Gomes

Matrícula: 20180040907

Prof. Thaís Gaudencio

Prova 2 Inteligência Artificial

SUMÁRIO

QUESTÃO 1 CLASSIFICAÇÃO

Nesta questão foi fornecido um dataset com informações referentes a músicas com frequências, etc… Desejando-se predizer as categorias de cada música.

Os algoritmos escolhidos foram KNN e uma rede neural fornecida da biblioteca do Keras chamada Sequential. A explicação do porquê escolhido cada algoritmo será dita no decorrer do do trabalho.

1. Análise da base de dados

Foi feita a análise de dados duplicados e incompletos mas não foi obtido nenhum dado subsequente. Também foi feita a análise de correlação em que as correlações que foram obtidas maiores que 0,7 foram deletadas. A normalização foi feita logo no começo para diminuir o número de outliers e além disso foi feita a conversão dos dados qualitativos em quantitativos.

As músicas precisam ser divididas em 6 categorias onde uma pode ter mais de uma categoria. A solução para isso foi colocar os dados em uma única coluna para poder dividir cada uma no momento da classificação como foi feito no algoritmo da FIGURA 1

E então, foi notado posteriormente que as classes não ficaram balanceadas como mostrado na FIGURA 2, e por isso foi necessário ser feito o oversampling.

2. KNN

KNN é um algoritmo de classificação que faz o cálculo com as K vizinhanças dependendo do K. E por ser simples, foi é um bom método para classificação e por isso foi um

dos algoritmos de aprendizagem escolhidos. Os resultados foram obtidos juntamente com a matriz de confusão onde foi feita as medidas de desempenho que foram calculadas de acordo com a matriz de confusão na FIGURA 3

3. Rede neural

A rede neural “Sequential” fornecida pelo Keras é bem robusta e completa para tratar modelos de classificação, por isso foi uma boa escolha para ser usada. A rede, no entanto, precisa de um modelo para ser definido, foi usado um modelo com 2 camadas de ativação. A primeira com dropout de 0.5 e 38 neurônios com função linear chamada “relu”. A segunda com dropout de 0.3 e 64 neurônios também com função “relu”. No final com a ativação“softmax” com 14 neurônios. E por fim, com o número de épocas 1500 foi gerado uma acurácia como mostra na FIGURA 4 obtendo uma acurácia de `66.52%`% segue o gráfico da acurácia e validação

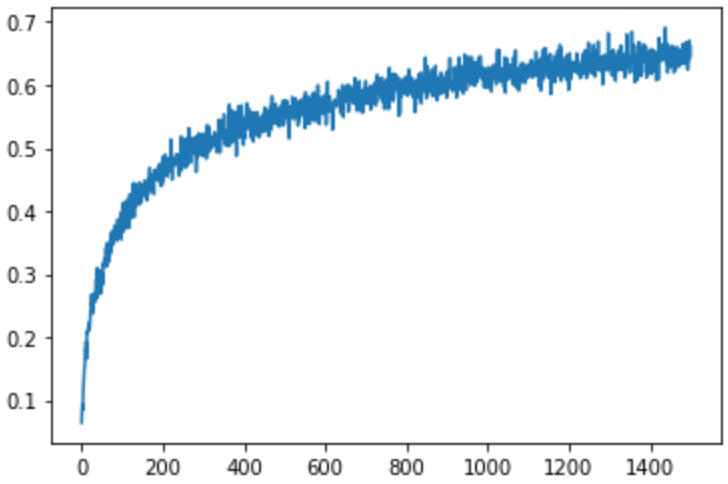


Gráfico em relação à acurácia

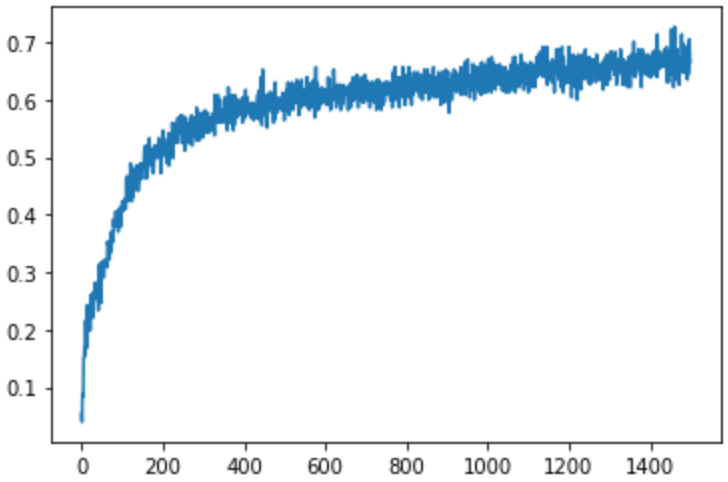


Gráfico em relação à validação

Foi calculado as métricas de desempenho e obtido os resultados de especificidade, medida f1, sensibilidade na FIGURA 14 com o auxílio da biblioteca “classification_report”

4. Conclusão

O algorítmo de KNN obteve pouca taxa de precisão, por se tratar de um algoritmo mais simples,. Contudo, as taxas de desempenho foram altas. Por outro lado, a rede neural obteve mais precisão por ter medidas de validação, e além disso obteve taxas de desempenho razoáveis

QUESTÃO 2 REGRESSÃO

Nesta questão foi fornecido um dataset com informações referentes ao clima de um determinado local onde se deseja predizer, por meio de uma aprendizagem de máquina, o melhor tráfego baseado nas informações do tempo.

1. Pré-processamento

Foi feito alguns testes de correlação e verificado se contém algum atributo vazio, mas o dataset já está bem ajustado e só foi preciso converter os dados de qualitativos para quantitativos.

2. Kfold

Para fazer a validação cruzada foi utilizado o processo de K Fold com splits de 10 com treinamento de 70% e teste 30% com o auxílio da função “train test split” e também gerando os atributos aleatoriamente

3. Naive bayes

Foi utilizado a predição por Naive bayes para predizer os atributos em y como mostra na FIGURA 6

Foi predito dentro do for onde ocorre os splits do Kfold e obtido os resultados na FIGURA 7 e com o resultado final de 4807 incorretos de 4820, ou seja, 13 predições. Foi feito mais um teste pegando mais uma vez as instâncias aleatórias e obtido 4811 incorretos de 4820 ou seja, 9 predições. Logo, em média o algoritmo de Baye obteve 11 resultados preditas.

4. Árvore de decisão

A segunda aprendizagem foi utilizada com árvore de decisão. Os mesmos passos foi utilizado para árvore de decisão, foi colocado a função de predição no for dos splits do KFold e geraram os resultados como mostra na FIGURA 8 com resultado final de 4820 resultados incorretos de 4035. Ou seja, 5 resultados preditos. Foi executado mais uma vez e obtidos 4820

resultados incorretos de 4031. Ou seja, 11 resultados preditos. Logo, em média o algoritmo de árvore de decisão teve 8 resultados preditos.

5. Conclusão

Os dois algoritmo obtiveram poucas taxas de predições, porém o que mais teve predições foi o algoritmo de bayes com 11 predições em média. Já o algoritmo de árvore de decisão obteve 8 predições. Talvez se fosse tratados melhor as correlações, por exemplo, excluído todas as correlações maiores que 0.7 talvez as taxas de predições fossem mais altas. Porém, no caso, só foi excluído uma instância com índice quase 1.

QUESTÃO 3 CLUSTERIZAÇÃO

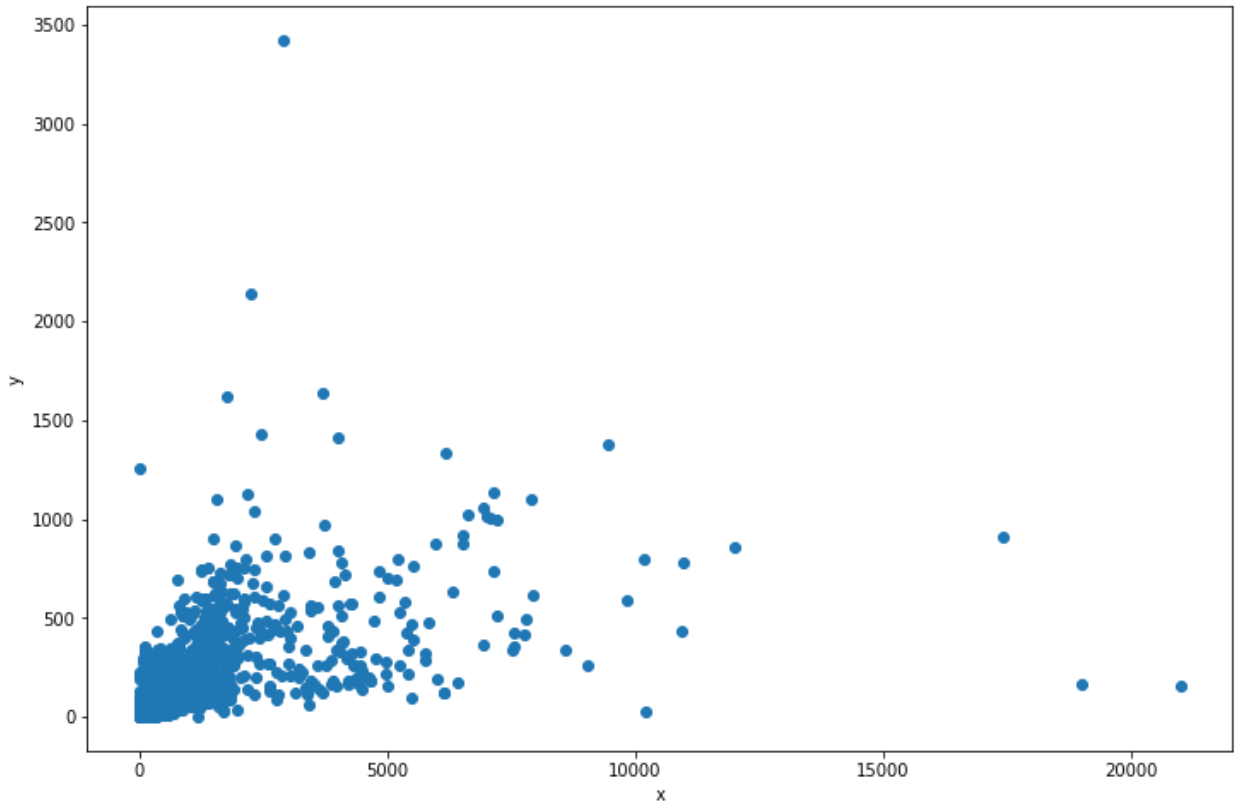
Nesta questão foi fornecido um dataset com informações de números referentes a posts de uma determinada empresa.

1. Pré-processamento

Logo de início, foi notado que algumas colunas constavam vazias e por esse motivo foram retiradas. Foi feito logo depois um mapa de correlação e notou-se que uma das instâncias têm alto índice de correlação (quase 1), foram as instâncias de “num_reactions” e “num_likes”, portanto, foi retirada por convenção a instância “num_reactions”.

Foi feito uma análise de outliers, porém, como os números podem variar muito por tratar-se de números referentes à curtidas de posts, existem muitos pontos fora do aglomerado, por esse motivo fica muito difícil saber qual o dado real ou não, portanto não foi preciso tratar.

Segue o exemplo feito entre as instâncias “num_coments” e “num_shares” da figura abaixo



Exemplo do plot referente às instâncias “num_coments” e “num_shares”.

Foi feito também a conversão de tipos qualitativos em quantitativos e também para tipo float, para melhor precisão na hora dos cálculos de divisão.

2. K-means

De início foi um pouco difícil saber o número de clusters que seria feito. Porém, depois de testar vários k’s dentro de um for indo no máximo até 12, já que constam 11 objetos no problema, conforme a FIGURA 9.

Colocando os índices de cada cluster em uma tabela foi concluído que o número de instâncias que melhor se distribuiu foi 4 conforme o resultado do print contendo a soma dos números referentes a cada cluster (FIGURA 10)

3. Hierárquico complete

Aplicando o algoritmo AgglomerativeClustering com os parâmetros “complete” e o número de clusters igual a 4 (pois foi o que melhor se distribuiu no k-means), foi obtido a distribuição dos clusters que segue nas figuras FIGURA 11 FIGURA 12

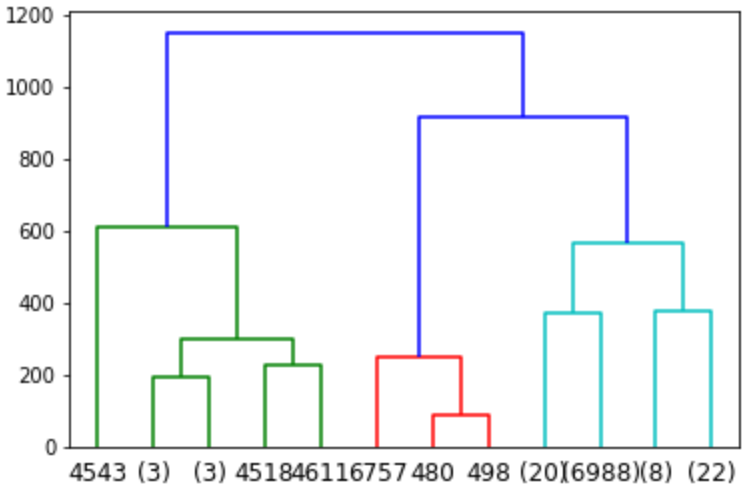


Imagem referente à hierarquia do agrupamento complete

4. Hierárquico single

Foi feito os mesmos passos do algoritmo hierárquico complete para o single, obtendo os resultados de clusters (FIGURA 13)

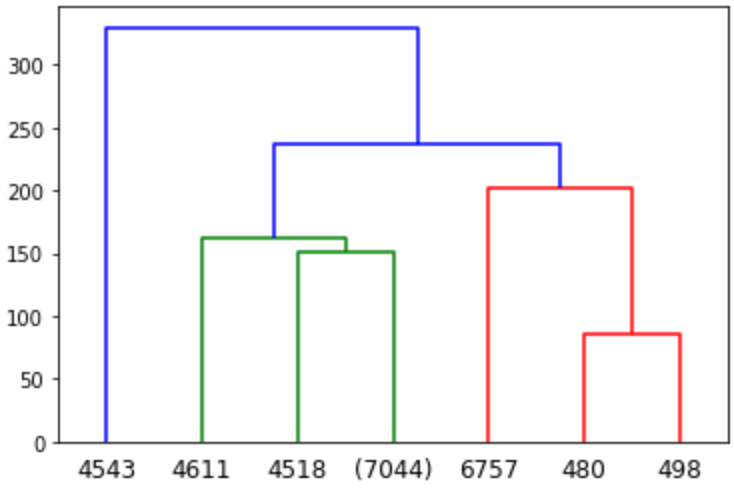


Imagem referente às hierarquias do agrupamento single

5. Conclusão

Foi concluído que o algoritmo de k-means teve a distribuição de cada instância do problema, o hierárquico complete teve uma distribuição razoável, enquanto o single agrupou a maioria das instâncias em apenas 1 cluster

LISTA DE FIGURAS

FIGURA 1

Imagem referente ao algoritmo feito às categorias de músicas

FIGURA 2

suprisedaggresive	81
happyrelaxing	74
aggresive	72
relaxingquietsad	67
relaxing	42
suprisedhappy	38
quietsad	37
relaxingquiet	30
relaxingsad	25
suprised	24
happy	23
sadaggresive	12
sad	12
suprisedhappyrelaxing	11
suprisedhappyaggresive	7
happyrelaxingquiet	6
suprisedsad	6
happyaggresive	5
quiet	5
suprisedsadaggresive	4
relaxingaggresive	3
relaxingsadaggresive	3
suprisedrelaxing	2
relaxingquietaggresive	1
happyquiet	1
happysad	1
quietsadaggresive	1

Figura referente às classes desbalanceadas

FIGURA 3

```
 ######## K-NN #########

Matriz de Confusão:
 [[3 1 0 0 1 0 0 0 3 3 0 1 1]
 [0 0 1 0 0 0 0 0 0 0 0 0 0]
 [0 1 2 0 0 1 2 0 0 0 0 2 0]
 [0 0 0 2 0 0 1 0 0 0 0 0 0]
 [0 0 0 0 0 0 2 0 0 0 0 0 1]
 [0 0 0 1 0 1 2 2 0 0 0 0 1]
 [0 0 0 0 0 0 0 0 0 0 0 0 0]
 [0 0 0 0 0 0 0 0 0 0 0 0 0]
 [0 0 0 0 0 0 0 0 0 0 0 0 0]
 [0 0 0 0 0 0 0 0 0 0 0 0 0]
 [2 5 0 0 1 0 1 0 2 5 0 3 0]
 [2 0 0 0 0 0 0 1 0 0 0 0 0]
 [0 0 0 0 0 0 0 0 0 0 0 0 0]]
Precisão:  0.75
Medida F1:  0.25
Sensibilidade: 0.14
Especificidade:  0.75
```

Figura referente ao resultado obtido do KNN da matriz de confusão

FIGURA 4

```
1 # evaluate loaded model on test data
2 loaded_model.compile(loss='sparse_categorical_crossentropy', optimizer='adam', metrics=['accuracy'])
3 # evaluate the model
4 scores = model.evaluate(x_test, y_test, verbose=0)
5 print("%s: %.2f%%" % (model.metrics_names[1], scores[1]*100))
```

```
acc: 70.93%
```

Figura referente ao resultado obtido do modelo da rede neural

FIGURA 5

```
kf = KFold(n_splits=10, random_state=0, shuffle=True)
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, train_size=0.7, random_state=Non
```

Figura referente a função utilizada para dividir os kfolds aleatoriamente

FIGURA 6

```
for train_index, test_index in kf.split(X):
  X_train, X_test = X.iloc[train_index], X.iloc[test_index]
  y_train, y_test = y.iloc[train_index], y.iloc[test_index]

  y_pred = gnb.fit(X_train, y_train).predict(X_test)

  print("Número de pontos com rótulos incorretos: %d em um total de %d pontos" % (X_test.shape[0], (y_tes
```

Figura referente a predição feita com Naive bayes

FIGURA 7

:

```
Numero de pontos incorretos de um total 4821 pontos : 4810
Numero de pontos incorretos de um total 4821 pontos : 4816
Numero de pontos incorretos de um total 4821 pontos : 4813
Numero de pontos incorretos de um total 4821 pontos : 4811
Numero de pontos incorretos de um total 4820 pontos : 4808
Numero de pontos incorretos de um total 4820 pontos : 4807
Numero de pontos incorretos de um total 4820 pontos : 4813
Numero de pontos incorretos de um total 4820 pontos : 4803
Numero de pontos incorretos de um total 4820 pontos : 4807
Numero de pontos incorretos de um total 4820 pontos : 4807
```

Resultado referente aos resultados feitos com Naive Bayes

FIGURA 8

```
Numero de pontos incorretos de um total 4821 pontos : 4073
Numero de pontos incorretos de um total 4821 pontos : 4096
Numero de pontos incorretos de um total 4821 pontos : 4186
Numero de pontos incorretos de um total 4821 pontos : 4263
Numero de pontos incorretos de um total 4820 pontos : 4130
Numero de pontos incorretos de um total 4820 pontos : 4061
Numero de pontos incorretos de um total 4820 pontos : 4156
Numero de pontos incorretos de um total 4820 pontos : 4137
Numero de pontos incorretos de um total 4820 pontos : 4320
Numero de pontos incorretos de um total 4820 pontos : 4035
```

Resultados referentes à aprendizagem por árvores de decisão

FIGURA 9

```
[ ]  #K-MEANS
     results = pd.DataFrame() #dataframe para colocar os indices de
     for k in range(1,12): #for com o numero maximo de objetos (12)
       kmeans = KMeans(n_clusters=k, random_state=True).fit(X)#Calcu
       X_clustered = kmeans.fit_predict(X) #Calcula os centros de cl
       results['Cluster', k-1] = X_clustered
```

Figura referente ao problema de K-means para cálculo com no máximo 11 objetos.

FIGURA 10

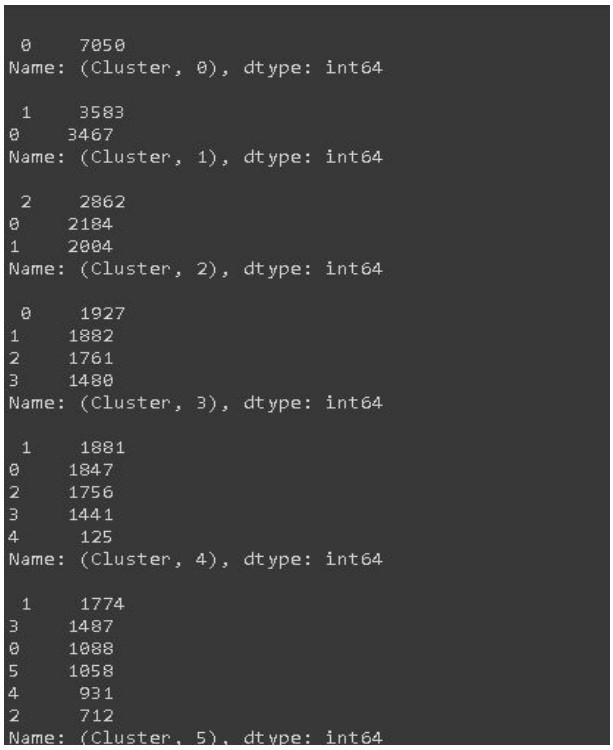


Figura referente ao número de índices de cada cluster

FIGURA 11

```
clustering = AgglomerativeClustering(n_clusters=4, linkage='complete').fit(X)
labels = clustering.labels_
```

Figura referente a função usada para calcular o agrupamento hierárquico complete

FIGURA 12

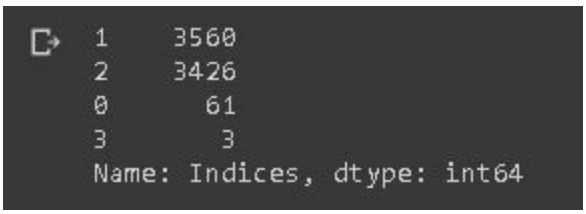


Imagem referente a soma de instâncias de cada cluster

FIGURA 13

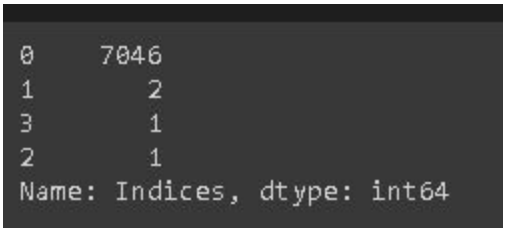


Imagem referente a soma de instâncias de cada cluster

FIGURA 14

	precision	recall	f1-score	support
0	0.61	0.89	0.72	19
1	1.00	0.93	0.97	15
2	1.00	0.75	0.86	16
3	0.93	0.93	0.93	15
4	0.94	1.00	0.97	16
5	1.00	1.00	1.00	17
6	0.93	0.93	0.93	14
7	0.95	1.00	0.98	20
8	1.00	0.71	0.83	7
9	0.94	1.00	0.97	17
10	0.95	1.00	0.97	19
11	1.00	0.75	0.86	20
12	1.00	0.71	0.83	14
13	0.86	1.00	0.92	18
accuracy			0.91	227
macro avg	0.94	0.90	0.91	227
weighted avg	0.93	0.91	0.91	227

Figura referente às medidas de desempenho da rede neural