Avaliação de Aprendizagem de Máquina em Data Science/Data Analytics

Deverá ser enviado um arquivo texto contendo os gráficos, resultados e comentários requeridos em cada item.
Obs. Antes de cada um dos exercícios, normalize os dados de entrada.

Questão 1. Considere os dados do arquivo "ex1data1.txt". Plote os dados em um gráfico (com a biblioteca matplotlib) colocando a primeira coluna dos dados no eixo x e a segunda coluna no eixo y. Baseado na visualização obtida, um modelo de regressão linear seria adequado para representar esses dados?

Para este conjunto de dados, considere a primeira coluna dos dados como o único atributo dos indivíduos (X) e a segunda coluna é o rótulo dos mesmos (y).

Questão 2. - Carregue os dados contidos no arquivo "ex2data1.txt".

O arquivo contem 100 linhas e 3 colunas de dados. Cada coluna se refere a uma variável. Neste problema, deve-se desenvolver um modelo de classificação capaz de reproduzir as classes apresentadas na terceira coluna dos dados.

O problema consiste em um sistema de admissão de alunos em uma universidade. Os dados das colunas 1 e 2 representam as notas de cada aluno em dois testes. A coluna 3 indica se este aluno foi ou não admitido na universidade.

Os dados apresentados são dados históricos de alunos aceitos ou não. Deseja-se fazer um sistema que faça a avaliação dos alunos automaticamente.

Apresentar: Figura com os dados. Para a figura, utilize um gráfico em duas dimensões, cada uma contendo uma nota, e diferenciando aprovação e reprovação pela cor dos dados no gráfico.

- Divida o conjunto de dados entre treino e teste.
- Utilize a Regressão Logística para efetuar a classificação nos dados de treino e calcule a acurácia (score) obtida para o conjunto de teste.

Questão 3. - Carregue os dados contidos no Dataset de Câncer (breast cancer) do scikit-learn.

- Divida o conjunto de dados entre treino e teste.
- Utilize o Classificador do SVM (SVC) para realizar a classificação dos dados carregados.
- Varie o parâmetro C de 0.1 a 1 e mostre um gráfico contendo a diferença entre os scores obtidos no conjunto de **treino** para cada valor do parâmetro C.
- Varie o parâmetro C de 0.1 a 1 e mostre um gráfico contendo a diferença entre os scores obtidos no conjunto de **teste** para cada valor do parâmetro C.

Justifique as diferenças apresentadas em nos gráficos do treino e do teste para cada valor de C.

Questão 4. – Carregue os dados contidos no arquivo "fruit_data_with_colors_miss.txt".

- Utilize a estratégia de imputar os dados faltantes utilizando a média dos demais dados.
- Divida o conjunto de dados entre treino e teste.
- Utilize o Classificador do KNN para realizar a classificação dos dados carregados, com o conjunto de atributos sendo constituído das colunas "mass", "width", "height" e "color_score" e o rótulo sendo constituído pela coluna "fruit_label".
- Varie o parâmetro n_neighbors de 1 a 10 e mostre um gráfico contendo a diferença entre os scores obtidos no conjunto de **treino** para cada valor do parâmetro.
- Varie o parâmetro n_neighbors de 1 a 10 e mostre um gráfico contendo a diferença entre os scores obtidos no conjunto de **teste** para cada valor do parâmetro.

Justifique as diferenças apresentadas em nos gráficos do treino e do teste para cada valor de n_neighbors.

Questão 5.

– Carregue os dados contidos no arquivo "fruit_data_with_colors_miss.txt".

- Utilize a estratégia de imputar os dados faltantes utilizando a média dos demais dados.

- Utilize o PCA para diminuir a dimensionalidade dos atributos para n_components=2

- Divida o conjunto de dados entre treino e teste.

- Utilize o Classificador do KNN para realizar a classificação dos dados carregados, com o conjunto de atributos sendo constituído pela saída do PCA sobre as colunas “mass”, “width”, “height” e “color_score” e o rótulo sendo constituído pela coluna “fruit_label”.

- Varie o parâmetro n_neighbors de 1 a 10 e mostre um gráfico contendo a diferença entre os scores obtidos no conjunto de **treino** para cada valor do parâmetro.

- Varie o parâmetro n_neighbors de 1 a 10 e mostre um gráfico contendo a diferença entre os scores obtidos no conjunto de **teste** para cada valor do parâmetro.

Justifique as diferenças apresentadas nos gráficos da questão anterior em relação aos obtidos nesta questão.

Questão 6. - Carregue os dados contidos no Dataset de Iris do scikit-learn.

- Divida o conjunto de dados entre treino e teste.

- Utilize a Random Forest para realizar a classificação dos dados carregados.

- Varie o parâmetro n_estimators de 1 a 100 (de 10 em 10) e mostre um gráfico contendo a diferença entre os scores obtidos no conjunto de **treino** para cada valor do parâmetro n_estimators.

- Varie o parâmetro n_estimators de 1 a 100 (de 10 em 10) e mostre um gráfico contendo a diferença entre os scores obtidos no conjunto de **teste** para cada valor do parâmetro n_estimators.

Justifique as diferenças apresentadas em nos gráficos do treino e do teste para cada valor de n_estimators.

Apresente a matriz de confusão dos dados de teste.

Questão 7. - Carregue os dados contidos no Dataset de Iris do scikit-learn.

- Divida o conjunto de dados entre treino e teste.

- Utilize a Gradient Boosted Decision Tree (GradientBoostingClassifier) para realizar a classificação dos dados carregados.

- Varie o parâmetro n_estimators de 1 a 100 (de 10 em 10) e mostre um gráfico contendo a diferença entre os scores obtidos no conjunto de **treino** para cada valor do parâmetro n_estimators.
- Varie o parâmetro n_estimators de 1 a 100 (de 10 em 10) e mostre um gráfico contendo a diferença entre os scores obtidos no conjunto de **teste** para cada valor do parâmetro n_estimators.

Justifique as diferenças apresentadas em nos gráficos do treino e do teste para cada valor de n_estimators.

Apresente a matriz de confusão dos dados de teste.

Compare os resultados obtidos pela Random Forest e a Gradient Boosted Decision Tree, considerando acurácia e matriz de confusão, e diga qual deveria ser utilizada.

Questão 8. - Carregue os dados contidos no Dataset de Iris do scikit-learn.

- Divida o conjunto de dados entre treino e teste.
- Utilize o Classificador baseado em Redes Neurais (MLPClassifier) para realizar a classificação dos dados carregados.
- Varie o parâmetro hidden_layer_sizes de 10 a 100 (de 10 em 10) e mostre um gráfico contendo a diferença entre os scores obtidos no conjunto de **treino** para cada valor do parâmetro.
- Varie o parâmetro hidden_layer_sizes de 10 a 100 (de 10 em 10) e mostre um gráfico contendo a diferença entre os scores obtidos no conjunto de **teste** para cada valor do parâmetro.

Justifique as diferenças apresentadas em nos gráficos do treino e do teste para cada valor de hidden_layer_sizes.

Altere o parâmetro learning_rate para ‘adaptive’ e repita os experimentos realizados mostrando os mesmos gráficos.

Questão 9. - Carregue os dados contidos no Dataset de Câncer (breast cancer) do scikit-learn.

- Divida o conjunto de dados entre treino e teste.
- Utilize o Classificador baseado em Redes Neurais (MLPClassifier) para realizar a classificação dos dados carregados.
- Varie o parâmetro hidden_layer_sizes para [10,10], [25,50], [50,25] e [50,50] e mostre um gráfico contendo a diferença entre os scores obtidos no conjunto de **treino** para cada valor do parâmetro.

- Varie o parâmetro hidden_layer_sizes para [10,10], [25,50], [50,25], e [50,50] e mostre um gráfico contendo a diferença entre os scores obtidos no conjunto de **teste** para cada valor do parâmetro.

Justifique as diferenças apresentadas em nos gráficos do treino e do teste para cada valor de hidden_layer_sizes.

Altere o parâmetro alpha para 0.0001, 0.001 e 0.01 e repita os experimentos realizados mostrando os mesmos gráficos. Justifique os resultados obtidos.

Questão 10. - Carregue os dados contidos no Dataset de Câncer (breast cancer) do scikit-learn.

- Utilize o modelo K-means para encontrar os grupos dos dados carregados.
- Varie o parâmetro n_clusters para 2, 5 e 10.
- Calcule a quantidade de elementos em cada cluster para cada valor do parâmetro n_clusters. Para obter a quantidade de elementos que ficaram no cluster i, utilize o código km.labels_[km.labels_ == i].shape[0], considerando que km é a variável que contém o fit sobre o KMeans.

Baseado nos resultados das divisões dos dados entre os cluster obtidos na etapa anterior, justifique qual valor você escolheria para n_clusters.