

Bootcamp: Analista de Machine Learning

Desafio do módulo

Módulo 1	Fundamentos de Aprendizado de Máquina
-----------------	--

Objetivos

Exercitar os seguintes conceitos trabalhados no Módulo:

- ✓ Análise exploratória dos dados (EDA - *Exploratory Data Analysis*).
- ✓ Preparação dos dados.
- ✓ Comparação e ajuste de modelos de classificação.

Enunciado

Neste desafio, serão abordados todos os conceitos apresentados durante o módulo Fundamentos de Aprendizado de Máquina. Utilizaremos uma versão modificada do dataset “Wine Quality” disponível no **UCI Machine Learning Repository** (<https://archive.ics.uci.edu/ml/datasets/wine>). Esse dataset contém um conjunto de atributos (dados de sensores) sobre o processo de fabricação de vinhos (tinto e branco). Esses dados são utilizados para classificar, ao final do processo, a qualidade do vinho obtido. Existem informações como o teor alcoólico e nível de acidez. Para este desafio, é necessário baixar o arquivo “winequality-red.data” presente no seguinte link:

<https://drive.google.com/file/d/13jSMzdwO3nZDr-n62--fO4jrE-olG8cX/view?usp=sharing>

Atividades

Os alunos deverão desempenhar as seguintes atividades:

1. Acessar o Google Colaboratory.

2. Realizar o upload do dataset “winequality-red.data” presente no link:

<https://drive.google.com/open?id=13jSMzdwO3nZDr-n62--fO4jrE-olG8cX>

3. Para a implementação dos algoritmos, utilize as definições abaixo:

Algoritmo KNN:

```
clf_KNN = KNeighborsClassifier(n_neighbors=5)
```

Algoritmo Árvore de Decisão:

```
clf_arvore = DecisionTreeClassifier()
```

Algoritmo Floresta Randômica:

```
clf_floresta = RandomForestClassifier(max_depth=10, random_state=1)
```

Algoritmo SVM:

```
clf_svm=SVC(gamma='auto',kernel='rbf')
```

Algoritmo Rede MLP:

```
clf_mlp = MLPClassifier( alpha=1e-5, hidden_layer_sizes=(5, 5), random_state=1)
```

Obs.:

1. Quando for realizar a leitura do arquivo “**winequality-red.csv**” com a função `pandas.read_csv()`, é necessário utilizar o atributo “ **sep=';** ” para que as colunas sejam reconhecidas.
2. Para a divisão dos dados de treinamento e teste dos algoritmos, utilize o valor de “**random_state=1**” e a proporção de **70%** para **treinamento** e **30%** para **teste**.
3. Utilize a normalização dos dados utilizando o `MinMaxScaler` para todos os algoritmos.
4. Utilize a variável “quality” como saída e as demais como entrada do modelo.

5. Para a última questão, considere a realização das mesmas etapas desenvolvidas (MinMaxScaler, train_test_split etc.).