

**ENUNCIADO do Trabalho Prático n. 2 - Análise de
Dados e Aplicação de Machine Learning Avançado
usando ferramentas e linguagens de programação**

Índice

1. Enquadramento.....	1
2. Contexto e exemplos base	1
2.2 Tarefas.....	2
Anexos	5
A.1. Visualização de Estatística de Dados	5
A.2. Scripts em python da aplicabilidade das Árvores de Decisão	6
A.3. Scripts em python da aplicabilidade da Regressão Linear	7
A.4. Scripts em python da aplicabilidade das Redes Neurais Artificiais.....	8

1. Enquadramento

Este documento tem como objetivo descrever o enunciado do 2.º trabalho prático de avaliação referente à Análise de Dados usando ferramentas e linguagens de programação (python) e aplicação de abordagens de (Machine Learning) Aprendizagem Supervisionada (Regressão Linear, Árvores de Decisão e Redes Neurais Artificiais) e Não Supervisionada (clustering).

2. Contexto e exemplos base

Com base no exemplo apresentado nas aulas baseado na base de dados “*Insol.csv*”, disponível na página da disciplina no moodle que corresponde à recolha de informação referente a dados de insolvências de empresas. Esta base de dados contém 2738 registos de empresas e 33 variáveis.

O exemplo dado disponibiliza implementações **já disponíveis em python** para as seguintes tarefas:

Tarefa 1 -Realização do pré-processamento do conjunto de dados: *Insol.csv* e análise estatística dos dados através da análise do tipo de Atributos: Contínuos e Discretos e analisar Estatísticas simples:

1. n - número de elementos da amostra;
2. Mínimo (min) - valor menor da amostra;
3. Máximo (max) - valor maior da amostra;
4. Moda - valor mais comum da amostra;
5. Mediana - valor central da amostra (ordenar valores, se n é impar a é o valor do meio, se é par, é o valor médio das 2 observações centrais);

ENUNCIADO do Trabalho Prático n. 2 - Análise de
Dados e Aplicação de Machine Learning Avançado
usando ferramentas e linguagens de programação

6. Média - valor esperado de cada atributo;
7. 1º/3º Quartil - 25% dos valores da amostra estão abaixo/acima deste valor;
8. Desvio padrão;
9. Analisar a Distribuição Normal (ou gaussiana);
10. Outras distribuições: Binomial - utilizada quando o resultado final de um fenómeno é do tipo binário: verdadeiro/falso;
11. Uniforme - utilizada quando existe a mesma probabilidade de ocorrência;
12. Poisson - deveras utilizada em simulações de tempos, por exemplo passagem de automóveis numa estrada;
13. T-student (utilizada quando não se sabe qual o valor do desvio padrão da população original).
14. Intervalo de Confiança: Um intervalo de confiança representa uma gama de valores que sustentam um parâmetro desconhecido para a população;
15. Correlação de valores entre atributos (heatmap);
16. Apresentação da Scatterplot Matrix para cada atributo face ao atributo classe;
17. Outras estatísticas que poderá analisar no relacionamento entre atributos;

Tarefa 2: Utilização de algoritmos, que implementam a abordagem supervisionada para a criação de modelos de previsão/classificação através das árvores de decisão, regressão logística e redes neurais artificiais com a linguagem de programação *Python* (com a *framework scikit-learn* ou outra) ;

Tarefa 3: Exploração dos algoritmos a utilizar, da biblioteca *scikit-learn*, visualizando diferentes métodos de avaliação do modelo e realizando uma previsão para um novo registo;

2.2 Tarefas

Seguindo o exemplo do dataset insolvências, **deverá usar um outro dataset** (que já o utilizou num trabalho anterior) para seguir

Tarefa 1: Explorar o Dataset a nível de estatísticas de dados: deverá seguir a checklist de métricas a analisar no dataset (mediana, distribuições, correlação de atributos (heat map) e outras funções estatísticas., por exemplo em poderá consultar algumas em python neste link: <https://www.kaggle.com/code/benhamner/python-data-visualizations/notebook>

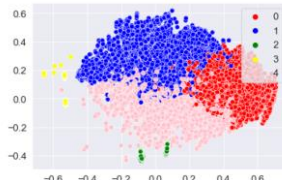
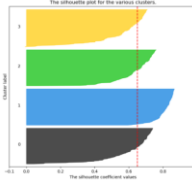
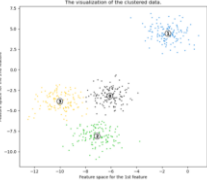
Tarefa 2: Aplicar abordagens de Machine Learning:

- **Aprendizagem Supervisionada:** O dataset tem de ter uma classe/atributo para se criar modelos de classificação. Pretende-se encontrar o melhor modelo de previsão: Com a Aprendizagem

**ENUNCIADO do Trabalho Prático n. 2 - Análise de
Dados e Aplicação de Machine Learning Avançado
usando ferramentas e linguagens de programação**

supervisionada e o código em python já disponibilizado com a implementação das árvores de decisão, regressão logística e redes neurais artificiais pretende-se que

- **Tarefa 2.1:** altere o código para processar vários modelos mudando os parâmetros do split/cross-validation k-folds e imprima no ecrã o máximo de métricas disponibilizadas pela framework (matriz da confusão, recall, Precision, f1score, etc). Por cada iteração grave a sequência num ficheiro JSON.
 - Ao fim de N iterações (N>20) deverá encontrar o melhor modelo de previsão tanto para as árvores de decisão, redes neurais artificiais e regressão linear.
 - Como output terá um ficheiro JSON com os registos das iterações, o tipo de separação do dataset de treino e teste e as métricas de avaliação dos modelos assim como os vários modelos gravados em ficheiro;
 - Com base no modelo com melhores resultados deverá “consumir/usar” o modelo para dado um novo caso a prever, seja submetido a modelo e seja devolvida a previsão.
-
- **Aprendizagem não supervisionada: tarefa 2.2:** Pretende-se encontrar padrões através da implementação do objetivo da Inteligência Artificial (clustering) implementando o algoritmo k-means (*).
 - Esta abordagem tem tido bastante sucesso quando aplicada a sistemas de recomendações (ex. clicks de páginas, user experience, etc)
 - (*) <https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html>
 NOTA: O agrupamento K-means, é um algoritmo de aprendizagem não supervisionada que classifica a entrada do conjunto de dados em vários clusters com base na distância da representação de cada variável de entrada. Para calcular a distância entre eles, diferentes métricas podem ser especificadas no algoritmo para tentar agrupar os dados de entrada em vários clusters. Os pontos são agrupados em torno de pontos centrais chamados centróides. Exemplo de Gráfico de Clustering:




- As etapas a serem seguidas para implementar o clustering K-means são as seguintes:
 1. Calcular a distribuição de intensidade das intensidades;
 2. Escolher k centroides de forma aleatória;
 3. Repetir as etapas a seguir até que o cluster não mude mais;
 4. Agrupar os pontos com base na distância de suas intensidades das intensidades do centroide;
 5. Calcular o novo centroide ou ponto médio para cada cluster

ENUNCIADO do Trabalho Prático n. 2 - Análise de
Dados e Aplicação de Machine Learning Avançado
usando ferramentas e linguagens de programação

- Complementando a informação das aulas de inteligência Artificial e de Sistemas de Suporte à Decisão (opção II – Aprendizagem Organizacional), poderá complementar a informação de exemplos de aplicabilidade deste algoritmo, por exemplo neste link:
 - <https://towardsdatascience.com/k-means-clustering-algorithm-applications-evaluation-methods-and-drawbacks-aa03e644b48a>

Como documentação final deverá criar um powerpoint de registo das tarefas e juntar as scripts usadas, compactando num ficheiro ZIP para submeter no moodle na página da unidade curricular.

ENUNCIADO do Trabalho Prático n. 2 - Análise de Dados e Aplicação de Machine Learning Avançado usando ferramentas e linguagens de programação

Anexos

A.1. Visualização de Estatística de Dados

Siga as scripts python disponibilizadas nas aulas e disponíveis na página da unidade curricular

```

← → ↺ kaggle.com/code/jribeiro2018/2022-insolv-viewdata/edit
2022-Insolv-ViewData Draft saved
File Edit View Run Add-ons Help
+ + Code - Draft Session Off (run a cell to start)
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import scipy.stats as st

# Configuring matplotlib to display and save images with a white background
plt.rcParams['figure.facecolor'] = 'white'

# Reading the data from the csv file
raw_data = pd.read_csv("../input/insolvi/Insol.csv", decimal=",", sep=";")
np.set_printoptions(suppress=True)

# Defining some regular bins
# Bin for variation coefficient
bins_var = [-1, -0.8, -0.6, -0.4, -0.2, 0, 0.2, 0.4, 0.6, 0.8, 1]

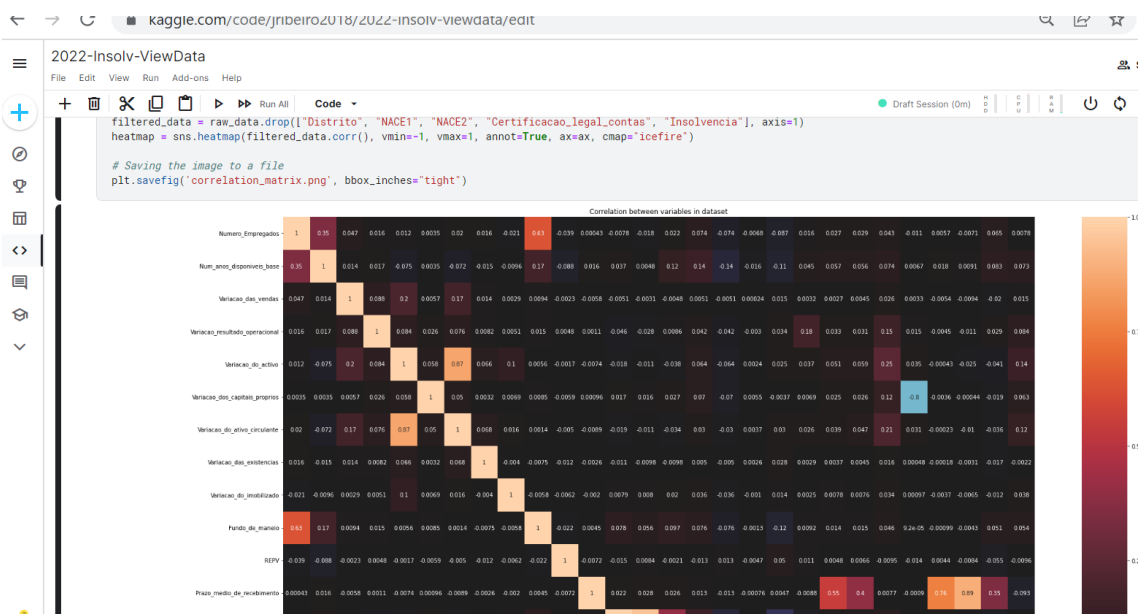
# Bin for values between 0 and 1
bins_from_zero = [0, 0.2, 0.4, 0.6, 0.8, 1]

[ ]:
curColumn = raw_data['Distrito']

fPorDistrito = curColumn[raw_data['Insolvencia'] == 1]
empresasFalidasPorDistrito = pd.DataFrame(fPorDistrito.value_counts().sort_index())
empresasFalidasPorDistrito.columns = ['Nº de empresas falidas']

colSummary = curColumn.describe()
colSummary.loc['mode'] = curColumn.mode().tolist()[0]
dfSummary = pd.DataFrame(colSummary)
dfSummary.columns = ['Valor']
display(dfSummary)

```



ENUNCIADO do Trabalho Prático n. 2 - Análise de Dados e Aplicação de Machine Learning Avançado usando ferramentas e linguagens de programação

A.2. Scripts em python da aplicabilidade das Árvores de Decisão

2022-Insolv-ArvoresDeDecisao-Previsao

```

## import
import numpy as np
import pandas as pd
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier
from sklearn import metrics

## carregar o ficheiro Insol para um dataframe do pandas dInsol
dInsol = pd.read_csv('../input/insolvencias-pontos/Insol_3.csv', sep=';', header=0)

dInsol.head(2) #Mostra as primeiras n linhas do conjunto de dados
dInsol.tail(5) #Mostra as últimas n linhas do conjunto de dados

```

	Distrito	NACE1	NACE2	Numero_Empregados	Num_anos_disponiveis_base	Variacao_das_vendas	Variacao_resultado_operacional	Variacao_do_ativo	Variacao_dos_capitais_proprios	Variacao_do_ativo_circulante	...	Rendibilidade_operaci
2733	5	14	141	35	7	0.027582	-0.274997	-0.176842	-0.062334	-0.028360
2734	5	14	141	4	4	-0.404304	-12.181545	-0.352800	-3.845692	-0.461129
2735	1	14	141	11	4	0.162294	3.474090	0.034278	0.975000	0.040681
2736	1	13	139	12	9	-0.165809	7.954233	-0.109300	0.018465	-0.108231
2737	1	13	139	4	4	-0.402254	-1.167466	0.095151	-1.435204	0.099516

5 rows x 33 columns

2022-Insolv-ArvoresDeDecisao-Previsao

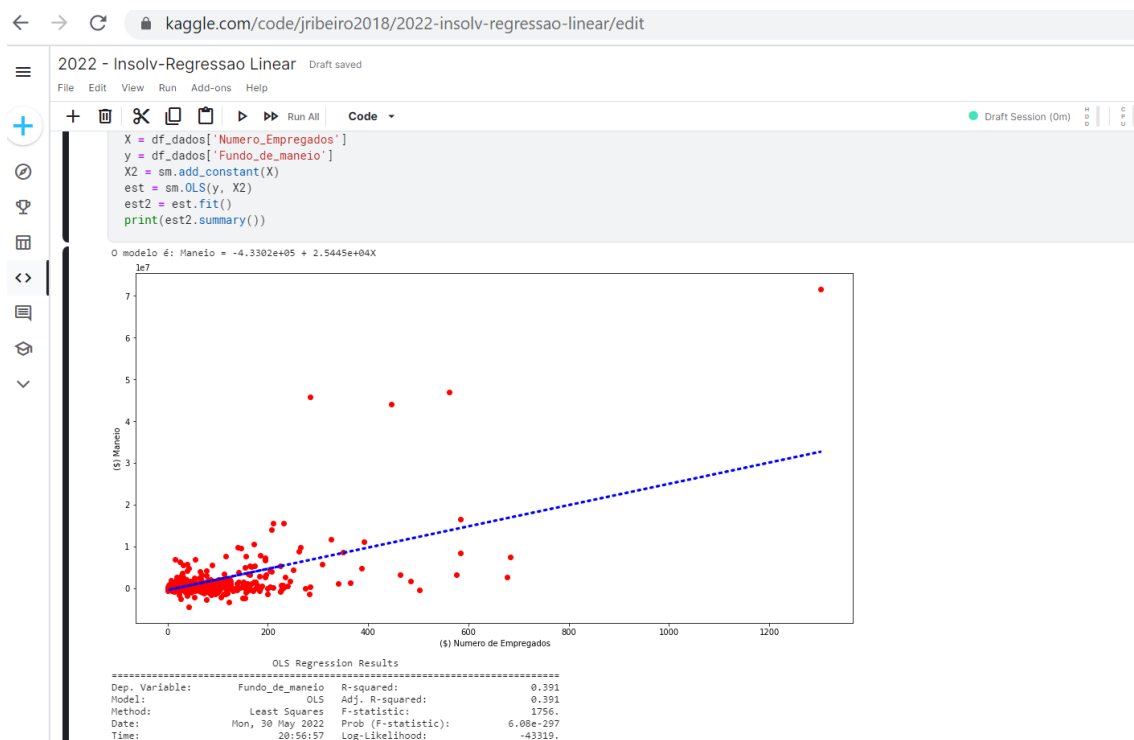
```

[68]:

```

ENUNCIADO do Trabalho Prático n. 2 - Análise de Dados e Aplicação de Machine Learning Avançado usando ferramentas e linguagens de programação

A.3. Scripts em python da aplicabilidade da Regressão Linear



ENUNCIADO do Trabalho Prático n. 2 - Análise de
Dados e Aplicação de Machine Learning Avançado
usando ferramentas e linguagens de programação

A.4. Scripts em python da aplicabilidade das Redes Neurais Artificiais

```
#Dividir em conjunto de treino e conjunto de teste
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.30, random_state=18)

#-----Cirar a Rede Neuronal Artificial-----
import keras
from keras.models import Sequential
from keras.layers import Dense
from numpy import loadtxt

model = Sequential()

#Adicionar a Input Layer e a primeira hidden layers
model.add(Dense(12, input_dim=32, activation='relu'))
model.add(Dense(32, activation='relu'))
model.add(Dense(1, activation='sigmoid'))

#Compiling the ANN
model.compile(loss='binary_crossentropy', optimizer='adam', metrics=['accuracy'])

#classifier.fit(X_train, y_train, batch_size = 10, epochs= 100)
model.fit(X_train, y_train, epochs=150, batch_size=10)

accuracy = model.evaluate(X, y, verbose=0)
print(accuracy)
```

```
Epoch 1/150
192/192 [=====] - 1s 2ms/step - loss: 322.3193 - accuracy: 0.7479
Epoch 2/150
192/192 [=====] - 0s 2ms/step - loss: 185.6676 - accuracy: 0.7615
Epoch 3/150
192/192 [=====] - 0s 2ms/step - loss: 292.5523 - accuracy: 0.7505
Epoch 4/150
192/192 [=====] - 0s 2ms/step - loss: 251.5001 - accuracy: 0.7771
Epoch 5/150
192/192 [=====] - 0s 2ms/step - loss: 567.8483 - accuracy: 0.7693
Epoch 6/150
192/192 [=====] - 0s 2ms/step - loss: 377.4080 - accuracy: 0.7563
Epoch 7/150
192/192 [=====] - 0s 2ms/step - loss: 320.8524 - accuracy: 0.7599
Epoch 8/150
192/192 [=====] - 0s 1ms/step - loss: 391.9887 - accuracy: 0.7521
Epoch 9/150
192/192 [=====] - 0s 2ms/step - loss: 1160.8945 - accuracy: 0.7985
Epoch 10/150
192/192 [=====] - 0s 2ms/step - loss: 643.1241 - accuracy: 0.8064
Epoch 11/150
192/192 [=====] - 0s 2ms/step - loss: 159.9599 - accuracy: 0.7620
Epoch 12/150
192/192 [=====] - 0s 2ms/step - loss: 145.1798 - accuracy: 0.7636
Epoch 13/150
192/192 [=====] - 0s 2ms/step - loss: 140.7422 - accuracy: 0.7761
Epoch 14/150
192/192 [=====] - 0s 2ms/step - loss: 124.3159 - accuracy: 0.7610
Epoch 15/150
192/192 [=====] - 0s 2ms/step - loss: 133.2973 - accuracy: 0.7516
Epoch 16/150
192/192 [=====] - 0s 2ms/step - loss: 78.9035 - accuracy: 0.7818
Epoch 17/150
192/192 [=====] - 0s 2ms/step - loss: 257.8298 - accuracy: 0.7683
```


**ENUNCIADO do Trabalho Prático n. 2 - Análise de
Dados e Aplicação de Machine Learning Avançado
usando ferramentas e linguagens de programação**

```
#Grafico da rede neuronal
!pip3 install ann_visualizer
!pip install graphviz

from ann_visualizer.visualize import ann_viz;
ann_viz(model, title="Visualização da Rede Neuronal Insolvência")
```

WARNING: Retrying (Retry(total=4, connect=None, read=None, redirect=None, status=None)) after connection broken by 'NewConnectionError('<pip._vendor 61346d0>: Failed to establish a new connection: [Errno -3] Temporary failure in name resolution')': /simple/ann-visualizer/
WARNING: Retrying (Retry(total=3, connect=None, read=None, redirect=None, status=None)) after connection broken by 'NewConnectionError('<pip._vendor 6134bd0>: Failed to establish a new connection: [Errno -3] Temporary failure in name resolution')': /simple/ann-visualizer/
WARNING: Retrying (Retry(total=2, connect=None, read=None, redirect=None, status=None)) after connection broken by 'NewConnectionError('<pip._vendor 6134190>: Failed to establish a new connection: [Errno -3] Temporary failure in name resolution')': /simple/ann-visualizer/
WARNING: Retrying (Retry(total=1, connect=None, read=None, redirect=None, status=None)) after connection broken by 'NewConnectionError('<pip._vendor 6134290>: Failed to establish a new connection: [Errno -3] Temporary failure in name resolution')': /simple/ann-visualizer/
WARNING: Retrying (Retry(total=0, connect=None, read=None, redirect=None, status=None)) after connection broken by 'NewConnectionError('<pip._vendor

