

Bootcamp: Desenvolvedor(a) Python

Trabalho Prático

Módulo 4: Python Avançado

Objetivos de Ensino

Exercitar os seguintes conceitos trabalhados no Módulo:

1. Python para aprendizado de Máquina (RNA do tipo MLP) para resolver problema de Regressão Linear.
2. Ajuste de modelos de aprendizado de Máquina.

Enunciado

A linguagem Python é bastante utilizada no meio da Ciência de Dados e do Aprendizado de Máquina. Outras áreas do conhecimento, como a Matemática, a Física e a Probabilidade, também a empregam bastante para ajudar a simular e a resolver problemas através da programação. Boa parte da popularidade dessa linguagem está relacionada à forma como o código é construído e à sua facilidade de leitura e de interpretação.

Atividades

Os alunos deverão desempenhar as seguintes atividades:

1. Para acessar o banco de dados, deve-se realizar a importação através da biblioteca sklearn.

```
from sklearn.datasets import load_boston
bostonData = load_boston()

X = bostonData.data
y = bostonData.target
```

Todos os dados da tabela deverão ser considerados para uso, a título de conhecimento os dados são descritos conforme a seguir:

```
df = pd.DataFrame(data=bostonData.data, columns=bostonData.feature_names)
df.head()
```

	CRIM	ZN	INDUS	CHAS	NOX	RM	AGE	DIS	RAD	TAX	PTRATIO	B	LSTAT
0	0.00632	18.0	2.31	0.0	0.538	6.575	65.2	4.0900	1.0	296.0	15.3	396.90	4.98
1	0.02731	0.0	7.07	0.0	0.469	6.421	78.9	4.9671	2.0	242.0	17.8	396.90	9.14
2	0.02729	0.0	7.07	0.0	0.469	7.185	61.1	4.9671	2.0	242.0	17.8	392.83	4.03
3	0.03237	0.0	2.18	0.0	0.458	6.998	45.8	6.0622	3.0	222.0	18.7	394.63	2.94
4	0.06905	0.0	2.18	0.0	0.458	7.147	54.2	6.0622	3.0	222.0	18.7	396.90	5.33

Onde,

- CRIM - taxa de criminalidade per capita por cidade.
- ZN - proporção de terrenos residenciais zoneados para lotes acima de 25.000 m².
- INDUS - proporção de acres de negócios não varejistas por cidade.
- CHAS - Variável fictícia Charles River (1 se o trato limita o rio; 0 caso contrário).
- NOX - concentração de óxidos nítricos (partes por 10 milhões).
- RM - número médio de cômodos por domicílio.
- IDADE - proporção de unidades ocupadas pelos proprietários construídas antes de 1940.
- DIS - distâncias ponderadas para cinco centros de emprego de Boston.

- RAD - índice de acessibilidade às rodovias radiais.
- IMPOSTO - taxa de imposto de propriedade de valor total por \$ 10.000.
- PTRATIO - relação aluno-professor por cidade.
- B - $1000(B_k - 0,63)^2$ onde B_k é a proporção de negros por cidade.
- LSTAT - % menor status da população.
- MEDV - Valor médio das casas ocupadas pelos proprietários em \$ 1.000.

2. Para a separação dos dados de treino e teste considere usar o módulo *train_test_split*

```
from sklearn.model_selection import train_test_split  
  
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

Para criar sua rede neural, utilize a arquitetura abaixo, observando o ajuste da quantidade de neurônios, em *units* conforme desejado. Observe que a camada de saída terá apenas um neurônio.

Nota: Substitua 'P' por um número inteiro que deseja ter de neurônios em sua camada.

```
model_MLP.add(keras.layers.Dense(units=P, activation='relu', input_shape= X_train.shape[1:]))  
model_MLP.add(keras.layers.Dense(units=1, activation='linear'))
```

Teste diferentes modelos de redes neurais e veja qual modelo obteve melhores resultados de erros e acurácia.