e1

May 30, 2021

1 Resposta ao Exercício 1

Grupo:

Isaque Elcio de Souza — RA: 225310

Matheus Vinicius Correa — RA: 22524

Thiago Bruschi Martins — RA: 120212

[1]:
```
import pandas as pd
```

[2]:
```
cols = ['class', 'l_spot', 'spot_d','act', 'evol', 'prev_act', 'hist_complex',
        'new_complex', 'area', 'max_area', 'C_flares','M_class','X_class']

types = {'class':'category', 'l_spot':'category',
         'spot_d':'category'}

df = pd.read_csv('solar-flare.csv', sep=' ', skiprows=1, names=cols,
                 dtype=types)
```

[3]:
```
df.head()
```

[3]:

	class	l_spot	spot_d	act	evol	prev_act	hist_complex	new_complex	area	max_area	C_flares	M_class	X_class
0	H	A	X	1	3	1	1	1	1	1	0	0	0
1	D	R	O	1	3	1	1	2	1	1	0	0	0
2	C	S	O	1	3	1	1	2	1	1	0	0	0
3	H	R	X	1	2	1	1	1	1	1	0	0	0
4	H	S	X	1	1	1	1	2	1	1	0	0	0

[4]:
```
df.tail()
```

[4]:
```
     class l_spot spot_d  act  evol  prev_act  hist_complex  new_complex  \
1061     H      S      X    1     2         1             1            1
1062     H      S      X    2     2         1             1            2
1063     C      S      O    1     2         1             2            2
1064     H      R      X    1     2         1             1            2
1065     B      X      O    1     1         1             1            2

      area  max_area  C_flares  M_class  X_class
1061     1         1         0        0        0
1062     1         1         0        0        0
1063     1         1         0        0        0
1064     1         1         0        0        0
1065     1         1         0        0        0
```

[5]:
```
df.dtypes
```

[5]:
```
class           category
l_spot          category
spot_d          category
act                int64
evol               int64
prev_act           int64
hist_complex       int64
new_complex        int64
area               int64
max_area           int64
C_flares           int64
M_class            int64
X_class            int64
dtype: object
```

2 Encode input data

[6]:
```
df_dummies = pd.get_dummies(df)
```

[7]:
```
df_dummies.head()
```

[7]:
```
   act  evol  prev_act  hist_complex  new_complex  area  max_area  C_flares  \
0    1     3         1             1            1     1         1         0
1    1     3         1             1            2     1         1         0
2    1     3         1             1            2     1         1         0
3    1     2         1             1            1     1         1         0
4    1     1         1             1            2     1         1         0

   M_class  X_class  …  l_spot_A  l_spot_H  l_spot_K  l_spot_R  l_spot_S  \
0        0        0  …         1         0         0         0         0
```

```
1        0        0  …         0         0         0         1         0
2        0        0  …         0         0         0         0         1
3        0        0  …         0         0         0         1         0
4        0        0  …         0         0         0         0         1

   l_spot_X  spot_d_C  spot_d_I  spot_d_O  spot_d_X
0         0         0         0         0         1
1         0         0         0         1         0
2         0         0         0         1         0
3         0         0         0         0         1
4         0         0         0         0         1

[5 rows x 26 columns]
```

[8]:
```
df_dummies.tail()
```

[8]:
```
      act  evol  prev_act  hist_complex  new_complex  area  max_area  \
1061    1     2         1             1            1     1         1
1062    2     2         1             1            2     1         1
1063    1     2         1             2            2     1         1
1064    1     2         1             1            2     1         1
1065    1     1         1             1            2     1         1

      C_flares  M_class  X_class  …  l_spot_A  l_spot_H  l_spot_K  l_spot_R  \
1061         0        0        0  …         0         0         0         0
1062         0        0        0  …         0         0         0         0
1063         0        0        0  …         0         0         0         0
1064         0        0        0  …         0         0         0         1
1065         0        0        0  …         0         0         0         0

      l_spot_S  l_spot_X  spot_d_C  spot_d_I  spot_d_O  spot_d_X
1061         1         0         0         0         0         1
1062         1         0         0         0         0         1
1063         1         0         0         0         1         0
1064         0         0         0         0         0         1
1065         0         1         0         0         1         0

[5 rows x 26 columns]
```

3 Scalling and Centering

[9]:
```
targets = ['C_flares', 'M_class', 'X_class']
input_data = df_dummies.drop(targets, axis=1)
```

[10]:
```
from sklearn.preprocessing import StandardScaler
```

```
scaller = StandardScaler()
centered = scaller.fit_transform(input_data)
```

[11]:
```
centered
```

[11]:
```
array([[-0.42640143,  0.96486532, -0.18458572, …, -0.5143262 ,
        -0.89991511,  1.49014892],
       [-0.42640143,  0.96486532, -0.18458572, …, -0.5143262 ,
         1.11121593, -0.67107387],
       [-0.42640143,  0.96486532, -0.18458572, …, -0.5143262 ,
         1.11121593, -0.67107387],
       …,
       [-0.42640143, -0.64727642, -0.18458572, …, -0.5143262 ,
         1.11121593, -0.67107387],
       [-0.42640143, -0.64727642, -0.18458572, …, -0.5143262 ,
        -0.89991511,  1.49014892],
       [-0.42640143, -2.25941816, -0.18458572, …, -0.5143262 ,
         1.11121593, -0.67107387]])
```

4 PCA

[12]:
```
from sklearn.decomposition import PCA

pca = PCA(n_components=0.9)
pca.fit(centered)
pca.n_components_
```

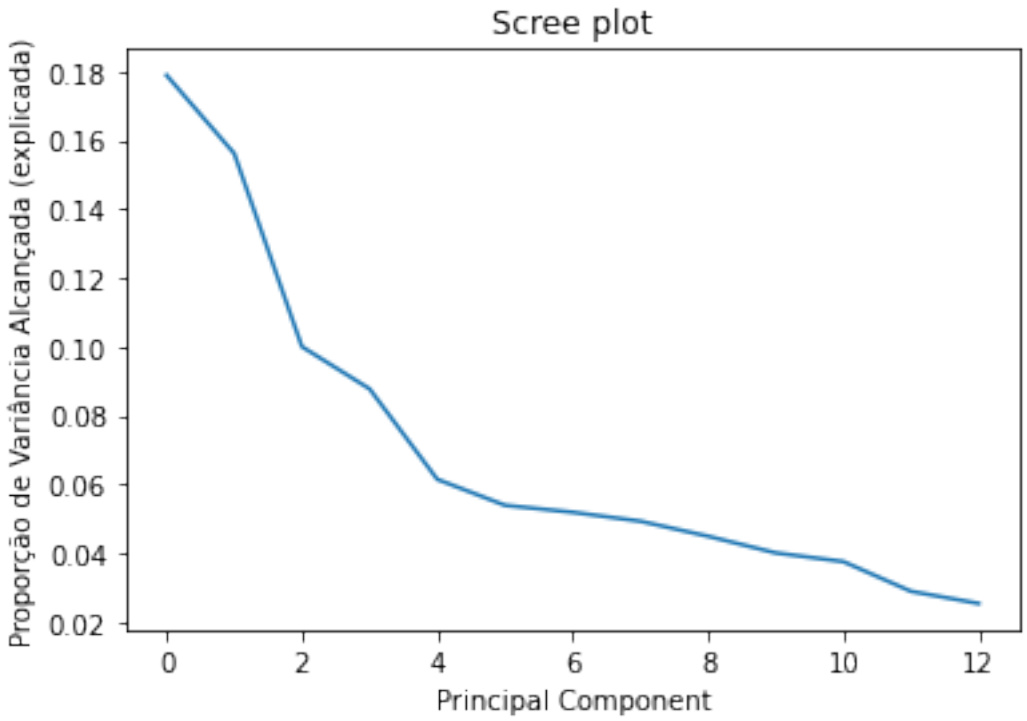
[12]:
```
13
```

5 Scree Plot

[13]:
```
variance_ratio = pca.explained_variance_ratio_
```

[14]:
```
import matplotlib.pyplot as plt

plt.plot(range(len(variance_ratio)), variance_ratio)
plt.title('Scree plot')
plt.xlabel('Principal Component')
plt.ylabel('Proporção de Variância Alcançada (explicada)')
plt.show()
```



6 PCA 90

[15]:
```
pca_90 = PCA(n_components=pca.n_components_)
x = pca_90.fit_transform(centered)
```

7 Validação cruzada

[16]:
```
import numpy as np
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import ShuffleSplit
from sklearn.model_selection import cross_val_score
```

[17]:
```
scores = {}

for feature in df_dummies[targets]:  # iterates over each column of outcomes
    y = df_dummies[targets][feature]
    ss = ShuffleSplit(n_splits=5, test_size=0.3, random_state=42)
    rmse_scores = -np.round(cross_val_score(LinearRegression(), x,  y, cv=ss,␣
 ↪scoring='neg_mean_squared_error'),3)
    mae_scores = -np.round(cross_val_score(LinearRegression(), x,  y, cv=ss,␣
 ↪scoring='neg_mean_absolute_error'),3)
```

```
    scores[feature] = {'RMSE':rmse_scores, 'MAE':mae_scores}

print('Resultado de 5 repetições cruzadas sobre cada uma das saídas')
for x in scores:
    print (x)
    for y in scores[x]:
        print ('\t',y,':',scores[x][y])
```

```
Resultado de 5 repetições cruzadas sobre cada uma das saídas
C_flares
         RMSE : [0.583 0.54  0.558 0.729 0.541]
         MAE : [0.429 0.427 0.4   0.445 0.389]
M_class
         RMSE : [0.079 0.039 0.056 0.068 0.045]
         MAE : [0.092 0.094 0.089 0.088 0.084]
X_class
         RMSE : [0.001 0.005 0.001 0.002 0.007]
         MAE : [0.011 0.024 0.011 0.015 0.02 ]
```

[18]:
```
avg_scores = {}

for feature in df_dummies[targets]:
    avg_scores[feature] = {'RMSE':round(np.mean(scores[feature]['RMSE']),3),
                           'MAE':round(np.mean(scores[feature]['MAE']),3)}

print('Média do RMSE e do MAE')
for x in avg_scores:
    print (x)
    for y in avg_scores[x]:
        print ('\t',y,':',avg_scores[x][y])
```

```
Média do RMSE e do MAE
C_flares
         RMSE : 0.59
         MAE : 0.418
M_class
         RMSE : 0.057
         MAE : 0.089
X_class
         RMSE : 0.003
         MAE : 0.016
```