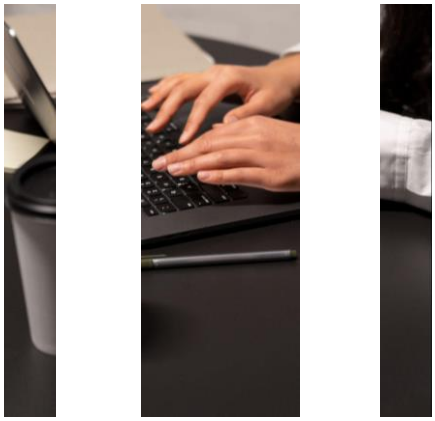
Extração, Tratamento e Carregamento de Dados

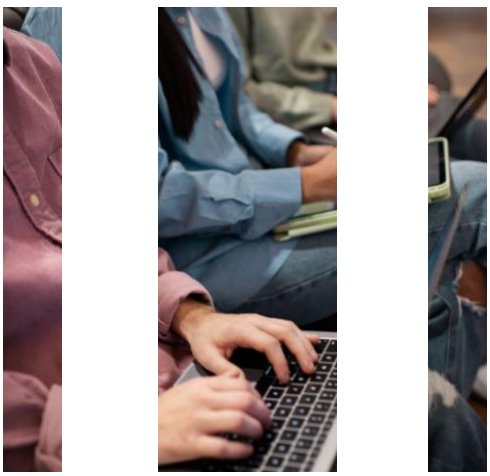


1

05

Módulo 5

Extração, Tratamento e Carregamento de Dados



Balduíno Mateus

2

Random projection

Em matemática e estatística, a projeção aleatória é uma técnica usada para reduzir a dimensionalidade de um conjunto de pontos que se encontram no espaço euclidiano.

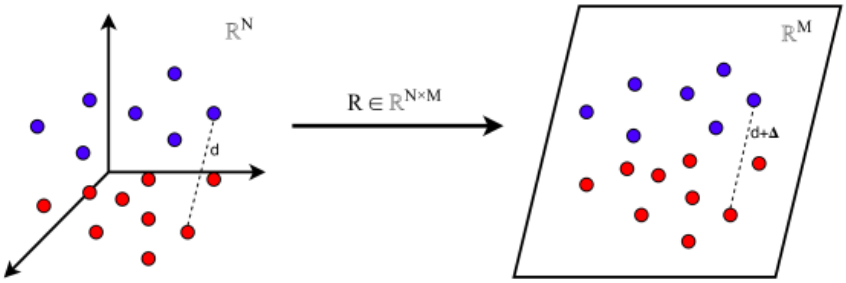
" Projeção " significa que a técnica projeta os dados de um espaço de alta dimensão para um espaço de dimensão inferior, e " Aleatório " significa que a matriz de projeção é gerada aleatoriamente.

3

Random projection

A matriz de projeção na projeção aleatória pode ser gerada por:

- Distribuição gaussiana, que é chamada de projeção aleatória gaussiana;
- Matriz esparsa, que é chamada de projeção aleatória esparsa.



4

Random projection

Bibliotecas:

- from sklearn.random_projection import GaussianRandomProjection
- from sklearn.random_projection import SparseRandomProjection

projeção aleatória gaussiana:

transformer = GaussianRandomProjection(n_components=4)

df_ron = transformer.fit_transform(df)

Projeção aleatória esparsa:

- srp = SparseRandomProjection(n_components=4)
- df_nov = srp.fit_transform(df)

5

Random projection

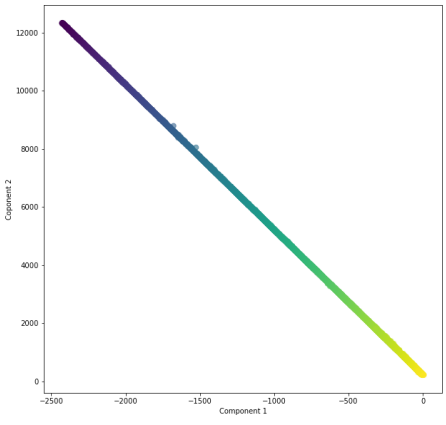
Shape (53940, 7)							
	Quilate	**Profundidade**	**Tabela**	**Preço**	**x**	**y**	**z**
0	0.23	61.5	55.0	326	3.95	3.98	2.43
1	0.21	59.8	61.0	326	3.89	3.84	2.31
2	0.23	56.9	65.0	327	4.05	4.07	2.31

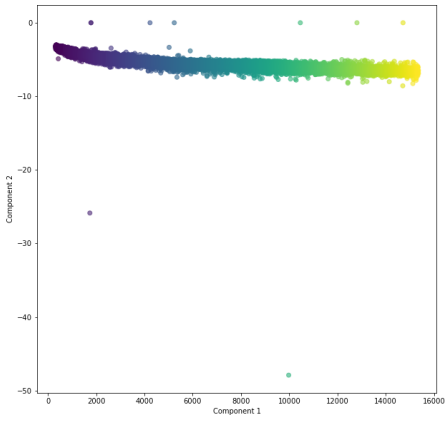
Shape (53940, 4)				
	0	**1**	**2**	**3**
0	44.730855	269.605065	-97.984972	47.967743
1	49.610585	269.540002	-101.368251	52.733612
2	52.863738	270.670473	-102.449925	56.173822

6

Random Projection

Aplicando algoritmo temos como resultado um novo arranjo geometrico do dataframe.





7

Principal Component Analysis (PCA)

- O PCA consiste em uma técnica estatística que utiliza uma transformação linear para representar um conjunto de atributos em um conjunto menor de atributos não correlacionados linearmente, mantendo boa parte das informações contidas nos dados originais.

	Quilate	Corte	Cor	Clareza	Profundidade	Tabela	Preco	x	y	z
0	0.23	Ideal	E	SI2	61.5	55.0	326	3.95	3.98	2.43
1	0.21	Premium	E	SI1	59.8	61.0	326	3.89	3.84	2.31
2	0.23	Good	E	VS1	56.9	65.0	327	4.05	4.07	2.31
3	0.29	Premium	I	VS2	62.4	58.0	334	4.20	4.23	2.63
4	0.31	Good	J	SI2	63.3	58.0	335	4.34	4.35	2.75

	comp1	comp2	comp3	comp4
0	-1.293237	-1.199584	-0.438451	-0.069697
1	1.221157	-2.363267	-0.227023	-0.343502
2	3.136450	-3.171666	-1.383979	-0.205007
3	-0.162971	-1.283409	0.855324	0.100230
4	-0.233340	-1.049009	1.470914	0.139177

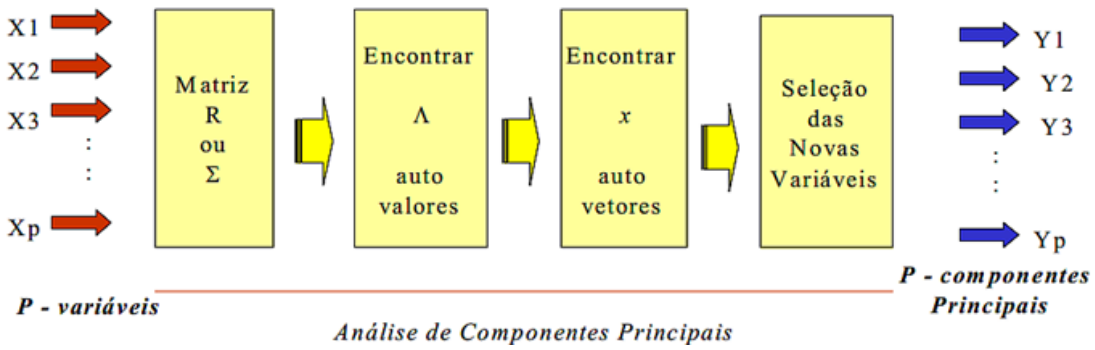
8

Principal Component Analysis (PCA)

- É a técnica mais conhecida da estatística multivariada;
- Pode ser utilizada para geração de índices e agrupamento de indivíduos;
- Cada componente principal é uma combinação linear de todas as variáveis originais;
- São independentes entre si;

9

Principal Component Analysis (PCA)



10

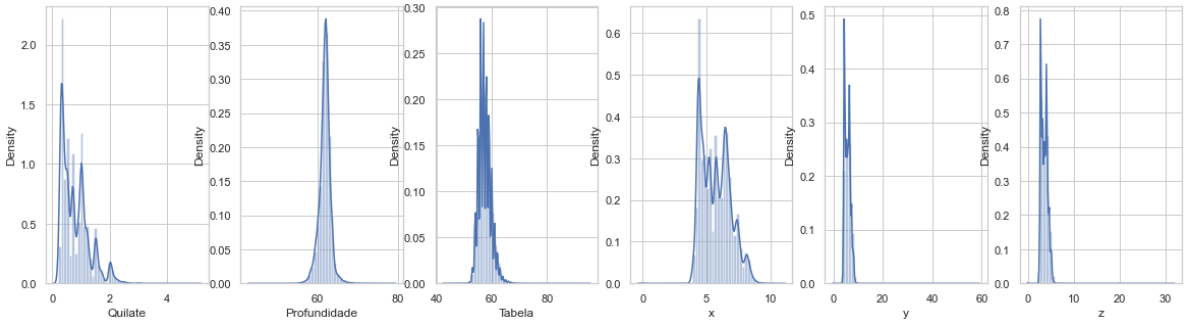
PCA em Python

- Para aplicar o PCA nos nossos dados é necessário pré caraterização dos dados, como parâmetros estatísticos, comportamento das variáveis e possível identificação de outliers.

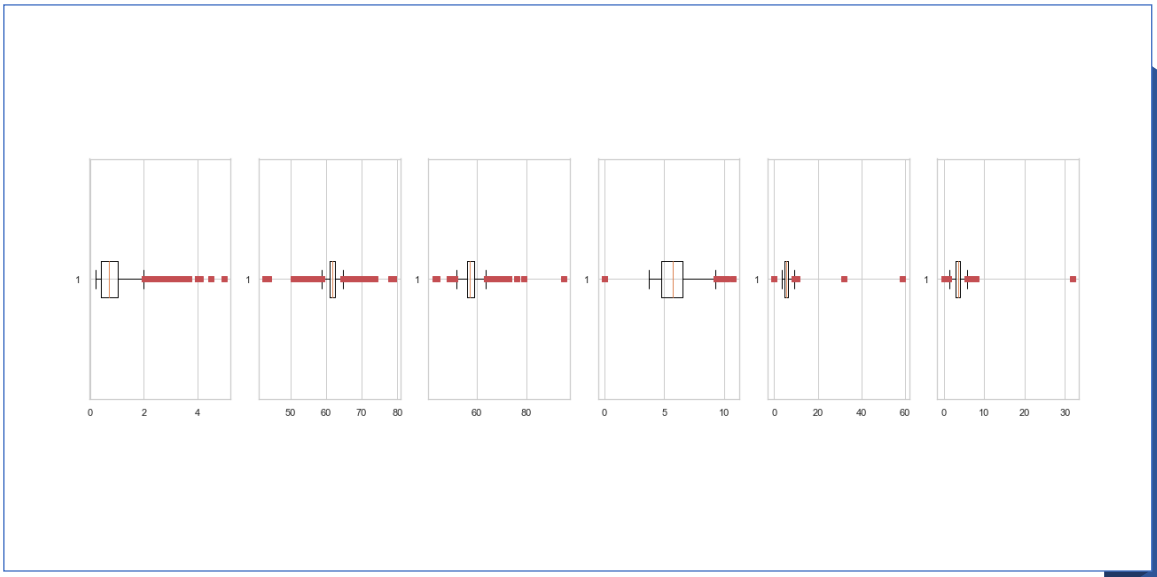
	count	mean	std	min	25%	50%	75%	max
Quilate	53940.0	0.797940	0.474011	0.2	0.40	0.70	1.04	5.01
Profundidade	53940.0	61.749405	1.432621	43.0	61.00	61.80	62.50	79.00
Tabela	53940.0	57.457184	2.234491	43.0	56.00	57.00	59.00	95.00
x	53940.0	5.731157	1.121761	0.0	4.71	5.70	6.54	10.74
y	53940.0	5.734526	1.142135	0.0	4.72	5.71	6.54	58.90
z	53940.0	3.538734	0.705699	0.0	2.91	3.53	4.04	31.80

11

PCA no Python



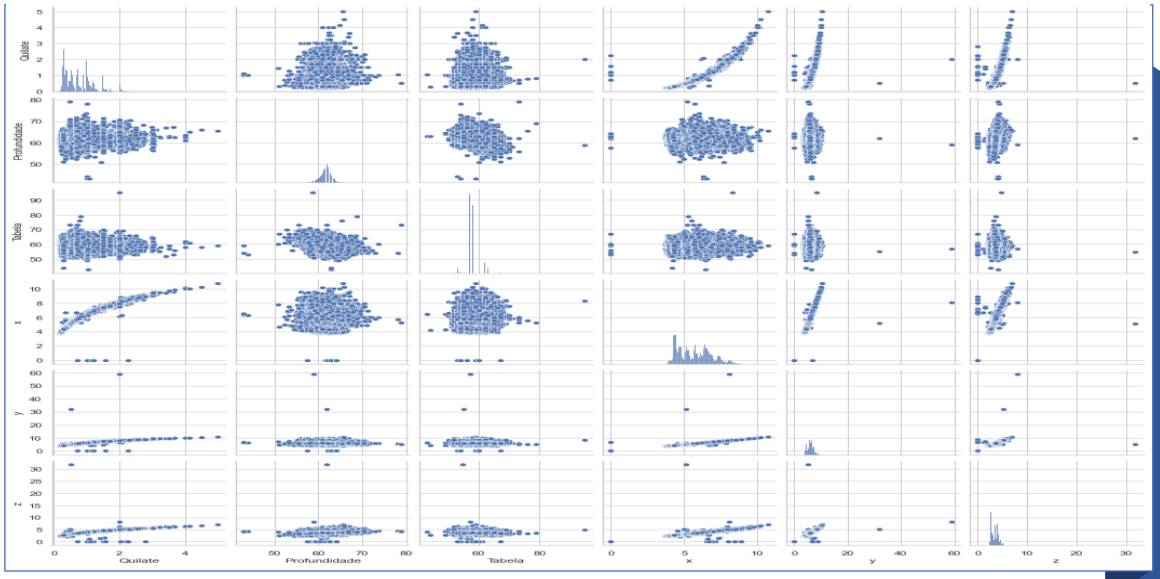
12


1
0
2
4
50
60
70
80
60
80
0
5
10
0
20
40
60
0
10
20
30

13


Qualidade
Profundidade
Tabela
x
y
1.000
0.028
0.182
0.975
0.952
0.953
0.028
1.000
-0.296
-0.025
-0.029
0.095
0.182
-0.296
1.000
0.195
0.184
0.151
0.975
-0.025
0.195
1.000
0.975
0.971
0.952
-0.029
0.184
0.975
1.000
0.952
1.0
0.8
0.6
0.4
0.2
0.0

14



15

PCA no Python

Para a realização do PCA seguiu-se os seguintes passos:

- Obter número da coluna do dataFame;
- Inicializar o parâmetro de números de componentes principais;
- Transformar dataframe em array;
- Aplicação do PCA;
- Transformação dos dados da componente principal em dataframe;
- Visualização das importâncias das componentes principais no gráfico do cotovelo.

16

Passos

Obter número da coluna do dataFame;

- In []: columns_df = list(df_n.columns)
- Out []: ['Quilate', 'Profundidade', 'Tabela', 'x', 'y', 'z']

Inicializar o parâmetro de números de componentes principais;

- In[]: pca = PCA(n_components=4, whiten=True)
- Out[]: PCA(n_components=4, whiten=True)

17

Passos

Transformar dataframe em array;

- In[]: dados_arry = df_n[columns_df].values
- Out[]: array([[0.23, 61.5 , 55. , 3.95, 3.98, 2.43],
  [0.21, 59.8 , 61. , 3.89, 3.84, 2.31],
  [0.23, 56.9 , 65. , 4.05, 4.07, 2.31],
  ...,
  [0.7 , 62.8 , 60. , 5.66, 5.68, 3.56],
  [0.86, 61. , 58. , 6.15, 6.12, 3.74],
  [0.75, 62.2 , 55. , 5.83, 5.87, 3.64]])

18

Passos

Aplicação do PCA;

- In[]: df_pca = pca.fit_transform(dados_arry)
- Out[]: array([[-1.29323673, -1.19958391, -0.43845083, -0.06969707],
  [1.22115701, -2.36326707, -0.22702296, -0.34350167],
  [3.1364505 , -3.17166614, -1.38397936, -0.20500736],
  ...,
  [0.87041784, -0.27171923, 1.35000229, 0.2579154],
  [0.37086401, 0.14897524, -0.47395727, -0.02282948],
  [-0.99014247, 0.49917005, -0.27269339, 0.17104691]])

19

Passos

Transformação dos dados das componentes principais em dataframe;

- In[]: dados_pca = pd.DataFrame(dados_pca,columns=['comp1', 'comp2','comp3', 'comp4'])
- Out[]:

	comp1	comp2	comp3	comp4
0	-1.293237	-1.199584	-0.438451	-0.069697
1	1.221157	-2.363267	-0.227023	-0.343502
2	3.136450	-3.171666	-1.383979	-0.205007
3	-0.162971	-1.283409	0.855324	0.100230
4	-0.233340	-1.049009	1.470914	0.139177

20

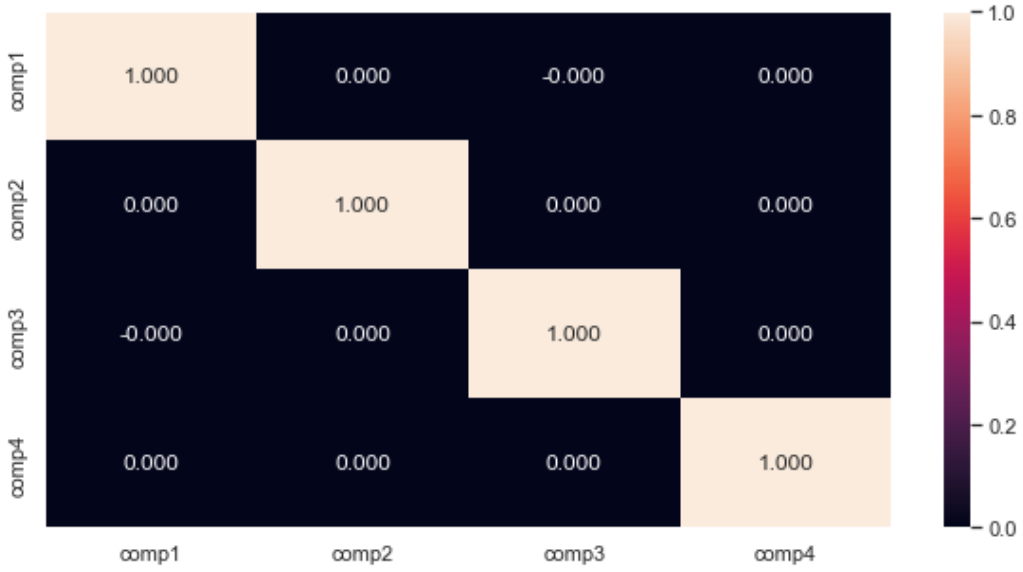
Passos

Visualização das importâncias das componentes principais no gráfico do cotovelo.

```
In [] :    Valor_PC = np.arange(pca.n_components_) +1
Valor_PC
plt.figure(figsize=(10,5))
plt.plot(Valor_PC, pca.explained_variance_ratio_, 'o-', linewidth=2,          color='blue')
plt.title('Gráfico da variância por componentes')
plt.xlabel('Componente Principal ')
plt.ylabel('Variância explicada (%)')
plt.show()
```

21

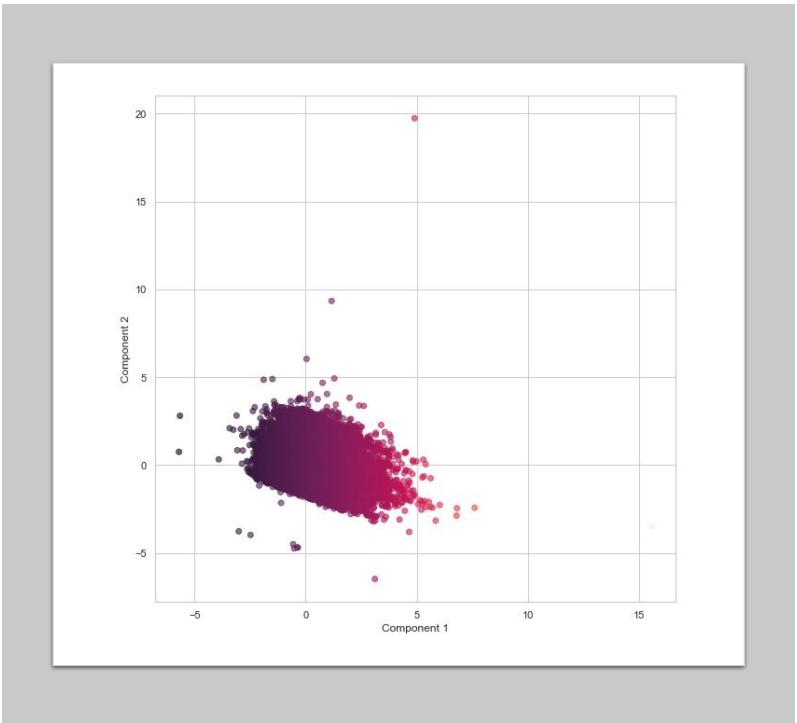
22



23

PCA

- Novo arranjo geométrico das variáveis do dataframe.



24

Exercício 21

- Utilizando o ficheiro 'data.csv' escreva um programa que faça as seguintes operações:
- Apresenta os gráficos de todas as variáveis numéricas;
- Mostra o histograma com a respetiva curva da normalidade;
- Exibe o gráfico boxplot de todas as variáveis;
- Apresenta o gráfico da correlação de todas as variáveis;
- Reduza o número de variáveis numéricas para 6;
- Apresenta a correlação entre as duas primeiras componentes.

25

Exercício 22

- Utilizando o ficheiro 'breast-cancer-Wisconsin escreva um programa que faça as seguintes operações:
- Renomeie as colunas com base no ficheiro 'breast-cancer-wisconsin.names'.
- Apresente os gráficos de todas as variáveis numéricas;
- Mostre o histograma com a respetiva curva da normalidade;
- Exibe o gráfico boxplot de todas as variáveis;
- Normalize os dados e apresente as diferenças graficamente;
- Apresente o gráfico da correlação de todas as variáveis;
- Reduza as variáveis numéricas para 3;
- Apresente o gráfico PC2 em função da PC1;
- Faça um gráfico 3D das PC1, PC2 , PC3;
- Apresente o gráfico de correlação entre a primeira e a segunda componente.

26