Московский государственный технический университет им. Н.Э. Баумана
Кафедра «Системы обработки информации и управления»

Рубежный контроль №1

по дисциплине

«Методы машинного обучения»

Выполнил:
студент группы ИУ5-23М
Дин Но

Москва — 2021 г.

Дополнительные требования по группам:

Для студентов групп ИУ5-23М, ИУ5И-23М - для произвольной колонки данных построить график "Ящик с усами (boxplot)".

In [24]:
```
import numpy as np
import pandas as pd

data = pd.read_csv('Iris.csv')
data.head()
data.describe()
```

Out[24]:

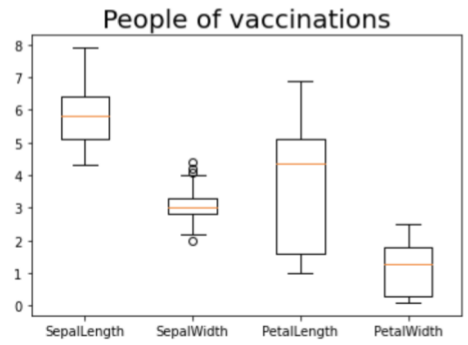
	Id	SepalLengthCm	SepalWidthCm	PetalLengthCm	PetalWidthCm
count	150.000000	150.000000	150.000000	150.000000	150.000000
mean	75.500000	5.843333	3.054000	3.758667	1.198667
std	43.445368	0.828066	0.433594	1.764420	0.763161
min	1.000000	4.300000	2.000000	1.000000	0.100000
25%	38.250000	5.100000	2.800000	1.600000	0.300000
50%	75.500000	5.800000	3.000000	4.350000	1.300000
75%	112.750000	6.400000	3.300000	5.100000	1.800000
max	150.000000	7.900000	4.400000	6.900000	2.500000

```
import matplotlib.pyplot as plt

box_1, box_2,box_3,box_4= data['SepalLengthCm'], data['SepalWidthCm'],data['PetalLengthCm'],data['PetalWidthCm']

plt.title('People of vaccinations',fontsize=20)
labels = 'SepalLength','SepalWidth','PetalLength','PetalWidth'

plt.boxplot([box_1, box_2,box_3,box_4], labels = labels, sym ="o")
plt.show()
```



Задача №17.

Для набора данных проведите нормализацию для одного (произвольного) числового признака с использованием преобразования Йео-Джонсона (Yeo-Johnson transformation).

```
from sklearn.preprocessing import PowerTransformer

featured_column=['SepalLengthCm', 'SepalWidthCm', 'PetalLengthCm', 'PetalWidthCm']
X=data[featured_column].values
Y=data['Species'].values

pt = PowerTransformer(method='yeo-johnson',standardize=False)
pt.fit(X)
transformed_data = pt.transform(X)
transformed_data
```

```
array([[1.37158622, 1.51724927, 1.46711008, 0.19710047],
       [1.35283713, 1.39747895, 1.46711008, 0.19710047],
       [1.33322896, 1.44707251, 1.35884371, 0.19710047],
       [1.32308066, 1.42257466, 1.57579675, 0.19710047],
       [1.3623149 , 1.5396171 , 1.46711008, 0.19710047],
       [1.39823725, 1.60394567, 1.79437097, 0.38912251],
       [1.32308066, 1.49438463, 1.46711008, 0.29368755],
       [1.3623149 , 1.49438463, 1.57579675, 0.19710047],
       [1.3020468 , 1.37175529, 1.46711008, 0.19710047],
       [1.35283713, 1.42257466, 1.57579675, 0.09924933],
       [1.39823725, 1.56150937, 1.57579675, 0.19710047],
       [1.34314472, 1.49438463, 1.68488843, 0.19710047],
       [1.34314472, 1.39747895, 1.46711008, 0.09924933],
       [1.2911399 , 1.39747895, 1.14363954, 0.09924933],
       [1.43130947, 1.62452594, 1.25101421, 0.19710047],
       [1.42328684, 1.7029694 , 1.57579675, 0.38912251],
       [1.39823725, 1.60394567, 1.35884371, 0.38912251],
       [1.37158622, 1.51724927, 1.46711008, 0.29368755],
       [1.42328684, 1.582946  , 1.79437097, 0.29368755],
```

Задача №37.

Для набора данных проведите процедуру отбора признаков (feature selection). Используйте класс SelectPercentile для 5% лучших признаков, и метод, основанный на взаимной информации.

```
from sklearn.feature_selection import SelectPercentile
from sklearn.feature_selection import mutual_info_classif

data_1=pd.read_csv('heart.csv')
data_1.head()
data_1.describe()
```

	age	sex	cp	trtbps	chol	fbs	restecg	thalachh	exng	oldpeak	slp	caa	
count	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.000000	303.0
mean	54.366337	0.683168	0.966997	131.623762	246.264026	0.148515	0.528053	149.646865	0.326733	1.039604	1.399340	0.729373	2.3
std	9.082101	0.466011	1.032052	17.538143	51.830751	0.356198	0.525860	22.905161	0.469794	1.161075	0.616226	1.022606	0.6
min	29.000000	0.000000	0.000000	94.000000	126.000000	0.000000	0.000000	71.000000	0.000000	0.000000	0.000000	0.000000	0.0
25%	47.500000	0.000000	0.000000	120.000000	211.000000	0.000000	0.000000	133.500000	0.000000	0.000000	1.000000	0.000000	2.0
50%	55.000000	1.000000	1.000000	130.000000	240.000000	0.000000	1.000000	153.000000	0.000000	0.800000	1.000000	0.000000	2.0
75%	61.000000	1.000000	2.000000	140.000000	274.500000	0.000000	1.000000	166.000000	1.000000	1.600000	2.000000	1.000000	3.0
max	77.000000	1.000000	3.000000	200.000000	564.000000	1.000000	2.000000	202.000000	1.000000	6.200000	2.000000	4.000000	3.0

```
X_1 = data_1.drop(['output'],axis = 1)
Y_1 = data_1.output

sp=SelectPercentile(mutual_info_classif, percentile=5).fit(X_1,Y_1)
X_new = sp.fit_transform(X,Y)
X_new.shape
```

```
(150, 1)
```

```
sp.get_support(indices=False)
```

```
array([False, False, False,  True])
```