

# Лабораторная работа №3

Импорт библиотек:

```
1 import pandas as pd
2 import numpy as np
3 import matplotlib.pyplot as plt
4 %matplotlib inline
```

Загружаем данные:

```
1 data = pd.read_csv("heart.csv")
2
3 x = data.drop('target', axis=1)
4 y = data['target']
5
6 # разделяем модель
7
8 from sklearn.model_selection import train_test_split
9 X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.20)
```

Обучаем ее:

```
1 from sklearn.svm import SVC
2 svcclassifier = SVC(kernel='linear')
3 svcclassifier.fit(X_train, y_train)
```

Предсказываем результаты

```
1 y_pred = svcclassifier.predict(X_test)
```

Оцениваем алгоритм:

```
1 from sklearn.metrics import classification_report, confusion_matrix
2 print(confusion_matrix(y_test, y_pred))
3 print(classification_report(y_test, y_pred))
```

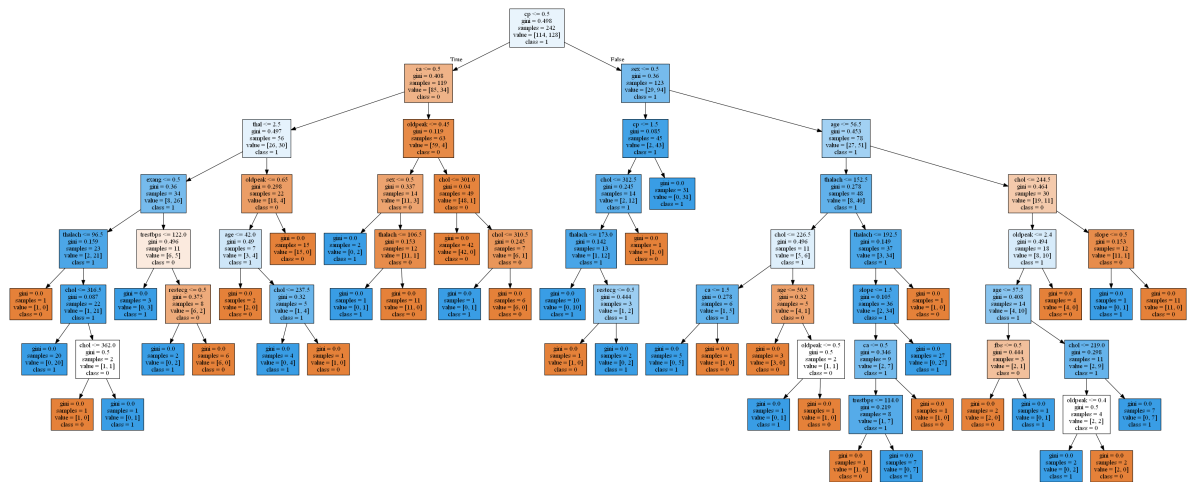
```
1 [[24  6]
2  [ 6 25]]
3
4           precision    recall  f1-score   support
5
6      0       0.80      0.80      0.80         30
7      1       0.81      0.81      0.81         31
8
9   accuracy                0.80         61
10  macro avg       0.80      0.80      0.80         61
11  weighted avg       0.80      0.80      0.80         61
```

# Дерево

```
1 from sklearn.tree import DecisionTreeClassifier
2 TreeClassifier = DecisionTreeClassifier()
3 TreeClassifier.fit(X_train, y_train)
4
5 yTree_pred = TreeClassifier.predict(X_test)
6
7 print(confusion_matrix(y_test, yTree_pred))
8 print(classification_report(y_test, yTree_pred))
```

1					
2	[[21 9]				
3	[ 8 23]]				
4		precision	recall	f1-score	support
5					
6	0	0.72	0.70	0.71	30
7	1	0.72	0.74	0.73	31
8					
9	accuracy			0.72	61
10	macro avg	0.72	0.72	0.72	61
11	weighted avg	0.72	0.72	0.72	61

Строим дерево и получаем:

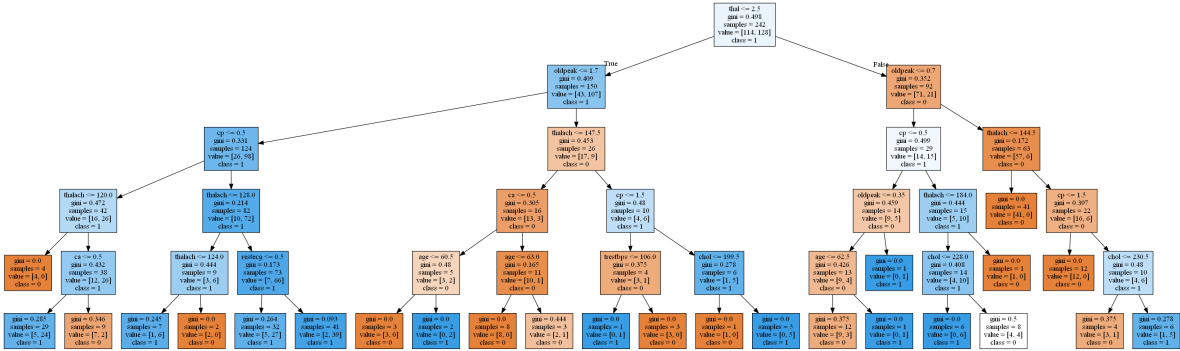


```
1 TreeClassifier2 = DecisionTreeClassifier(max_features=5, max_depth=5)
2 TreeClassifier2.fit(X_train, y_train)
3
4 yTree_pred = TreeClassifier2.predict(X_test)
```

Оценка:

```
1 print(confusion_matrix(y_test, yTree_pred))
2 print(classification_report(y_test, yTree_pred))
```

1	[[24 6]				
2	[10 21]]				
3		precision	recall	f1-score	support
4					
5	0	0.71	0.80	0.75	30
6	1	0.78	0.68	0.72	31
7					
8	accuracy			0.74	61
9	macro avg	0.74	0.74	0.74	61
10	weighted avg	0.74	0.74	0.74	61



# Вывод

В этой лабораторной работе нам предстояло классифицировать данные. Мы поняли, что это один из главных аспектов машинного обучения, так как используется повсеместно.

Мы разобрали два метода классификации опорных векторов и дерево решений. В наше случае, метод опорных векторов оказался точнее, так как данные были более хорошо расположены к этому методу.

# Authors



[Arthur Kupriyanov](#)



[Artyom Kolokolov](#)