

Евклидово расстояние:

между розовой и голубой точками 8.602325267042627
между розовой и зеленой точками 5.0990195135927845
между розовой и желтой точкой 19.313207915827967
между голубой и зеленой точкой 8.366600265340756
между зеленой и желтой точкой 15.588457268119896
между голубой и желтой точкой 18.57417562100671

Квадрат Евклидова расстояния:

между розовой и голубой точками 74.0
между розовой и зеленой точками 25.999999999999996
между розовой и желтой точками 373.00000000000006
между голубой и зеленой точкой 70.0
между зеленой и желтой точкой 243.0
между голубой и желтой точкой 345.0

Расстояние Чебышева:

между розовой и голубой точками 7.0
между розовой и зеленой точками 4.0
между розовой и желтой точкой 15.0
между голубой и зеленой точкой 6.0
между зеленой и желтой точкой 11.0
между голубой и желтой точкой 16.0

Расстояние Чебышева:

между розовой и голубой точками 14.0
между розовой и зеленой точками 8.0
между розовой и желтой точкой 29.0
между голубой и зеленой точкой 14.0
между зеленой и желтой точкой 23.0
между голубой и желтой точкой 29.0

1.3.2 Задание

Создать 5x5 матрицу со значениями в строках от 0 до 4. Для создания необходимо использовать функцию `arange`

```
In [17]: import numpy as np
         Z=np.zeros((5,5))
         Z+=np.arange(5)
         print(Z)
```

```
[[0. 1. 2. 3. 4.]
 [0. 1. 2. 3. 4.]
 [0. 1. 2. 3. 4.]
 [0. 1. 2. 3. 4.]
 [0. 1. 2. 3. 4.]]
```

2.3.1 Задание

Для предыдущего примера поэкспериментируйте с параметрами классификатора:

1. Установите другое количество ближайших соседей ($k=1, 5, 10$)
2. Установите размер тестовой выборки 15% от всего датасета.
3. Постройте графики и оцените качество моделей, проанализируйте результаты

```
In [44]: from sklearn.model_selection import train_test_split
         from sklearn.neighbors import KNeighborsClassifier
         import seaborn as sns
         import matplotlib.pyplot as plt
         import numpy as np
         k=[1,5,10]
         iris=sns.load_dataset('iris')
         X_train,X_test,y_train,y_test=train_test_split(iris.iloc[:, :-1], iris.iloc[:, -1], test_size=0.15)
         X_train.shape, X_test.shape, y_train.shape, y_test.shape
         X_train.head()
         y_train.head()

fig=plt.figure()
model=KNeighborsClassifier(n_neighbors=k[0]) #1 сосед
model.fit(X_train,y_train)
y_pred=model.predict(X_test)
ax1=fig.add_subplot(3,2,1) #добавили график к фигуре
sns.scatterplot(x='petal_width', y='petal_length', data=iris, hue='species', s=50) #обозначили точки
plt.xlabel('Длина лепестка, см') #дали имена осям
plt.ylabel('Ширина лепестка, см')
plt.legend(loc=2) #сделали легенду графика
plt.grid() #сделали график в сетку

for i in range(len(y_test)):
    if np.array(y_test)[i]!=y_pred[i]:
        plt.scatter(X_test.iloc[i,3], X_test.iloc[i,2], color='red', s=100)
from sklearn.metrics import accuracy_score
number=str(round(accuracy_score(y_test,y_pred),3)) #вычисляем оценку
title_in_string="Оценка для моделей лепестка (1 сосед): "+number
ax1.title.set_text(title_in_string)

model=KNeighborsClassifier(n_neighbors=k[0])
model.fit(X_train,y_train)
y_pred=model.predict(X_test)
ax2=fig.add_subplot(3,2,2)
sns.scatterplot(x='sepal_width', y='sepal_length', data=iris, hue='species', s=50)
plt.xlabel('Длина чашелистника, см')
plt.ylabel('Ширина чашелистника, см')
plt.legend(loc=2)
plt.grid()

for i in range(len(y_test)):
    if np.array(y_test)[i]!=y_pred[i]:
        plt.scatter(X_test.iloc[i,1], X_test.iloc[i,0], color='red', s=100)
from sklearn.metrics import accuracy_score
number=str(round(accuracy_score(y_test,y_pred),3))
title_in_string="Оценка для моделей чашелистника (1 сосед): "+number
ax2.title.set_text(title_in_string)

model=KNeighborsClassifier(n_neighbors=k[1]) #5 сосед
model.fit(X_train,y_train)
y_pred=model.predict(X_test)
ax3=fig.add_subplot(3,2,3)
sns.scatterplot(x='petal_width', y='petal_length', data=iris, hue='species', s=50)
plt.xlabel('Длина лепестка, см')
plt.ylabel('Ширина лепестка, см')
```

```

plt.legend(loc=2)
plt.grid()

for i in range(len(y_test)):
    if np.array(y_test)[i]!=y_pred[i]:
        plt.scatter(X_test.iloc[i,3],X_test.iloc[i,2],color='red',s=100)
from sklearn.metrics import accuracy_score
number=str(round(accuracy_score(y_test,y_pred),3))
title_in_string="Оценка для моделей лепестка (5 соседей): "+number
ax3.title.set_text(title_in_string)

model=KNeighborsClassifier(n_neighbors=k[1])
model.fit(X_train,y_train)
y_pred=model.predict(X_test)
ax4=fig.add_subplot(3,2,4)
sns.scatterplot(x='sepal_width',y='sepal_length',data=iris,hue='species',s=50)
plt.xlabel('Длина чашелистника, см')
plt.ylabel('Ширина чашелистника, см')
plt.legend(loc=2)
plt.grid()

for i in range(len(y_test)):
    if np.array(y_test)[i]!=y_pred[i]:
        plt.scatter(X_test.iloc[i,1],X_test.iloc[i,0],color='red',s=100)
from sklearn.metrics import accuracy_score
number=str(round(accuracy_score(y_test,y_pred),3))
title_in_string="Оценка для моделей чашелистника (5 соседей): "+number
ax4.title.set_text(title_in_string)

model=KNeighborsClassifier(n_neighbors=k[2])#10 соседей
model.fit(X_train,y_train)
y_pred=model.predict(X_test)
ax5=fig.add_subplot(3,2,5)
sns.scatterplot(x='petal_width',y='petal_length',data=iris,hue='species',s=50)
plt.xlabel('Длина лепестка, см')
plt.ylabel('Ширина лепестка, см')
plt.legend(loc=2)
plt.grid()

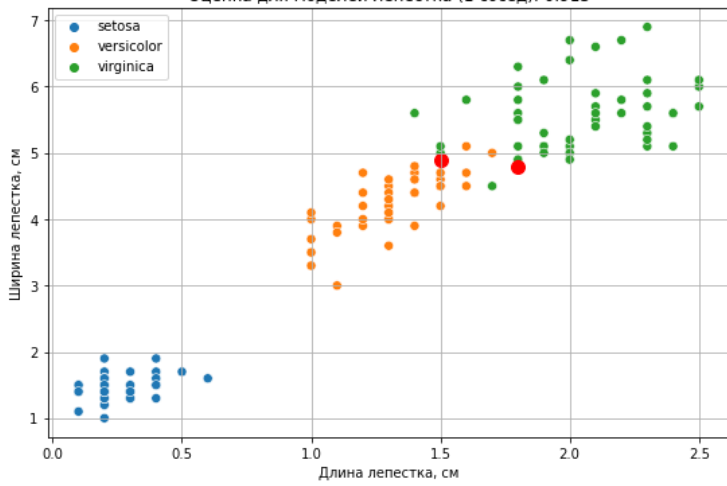
for i in range(len(y_test)):
    if np.array(y_test)[i]!=y_pred[i]:
        plt.scatter(X_test.iloc[i,3],X_test.iloc[i,2],color='red',s=100)
from sklearn.metrics import accuracy_score
number=str(round(accuracy_score(y_test,y_pred),3))
title_in_string="Оценка для моделей лепестка (10 соседей): "+number
ax5.title.set_text(title_in_string)

model=KNeighborsClassifier(n_neighbors=k[2])
model.fit(X_train,y_train)
y_pred=model.predict(X_test)
ax6=fig.add_subplot(3,2,6)
sns.scatterplot(x='sepal_width',y='sepal_length',data=iris,hue='species',s=50)
plt.xlabel('Длина чашелистника, см')
plt.ylabel('Ширина чашелистника, см')
plt.legend(loc=2)
plt.grid()

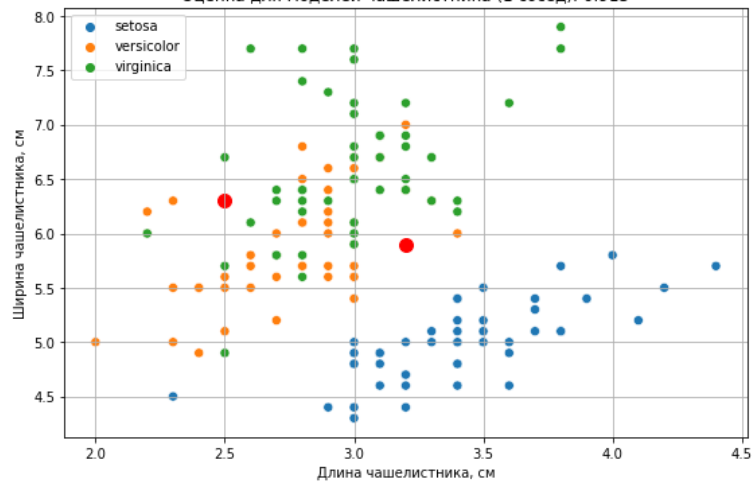
for i in range(len(y_test)):
    if np.array(y_test)[i]!=y_pred[i]:
        plt.scatter(X_test.iloc[i,1],X_test.iloc[i,0],color='red',s=100)
from sklearn.metrics import accuracy_score
number=str(round(accuracy_score(y_test,y_pred),3))
title_in_string="Оценка для моделей чашелистника (10 соседей): "+number
ax6.title.set_text(title_in_string)

```

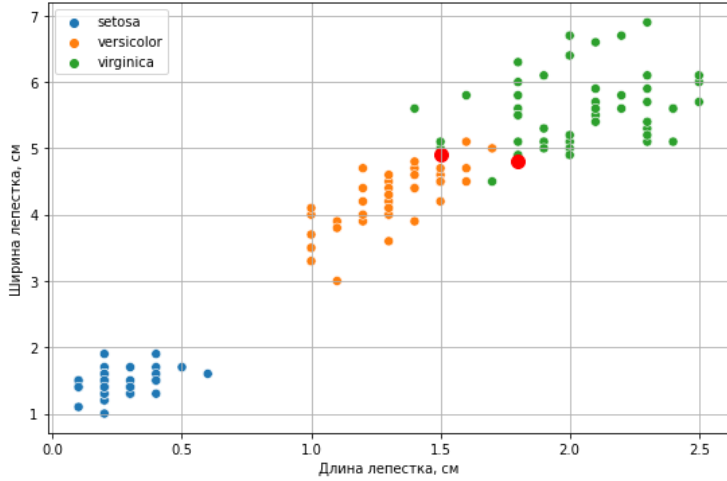
Оценка для лепестка (1 сосед): 0.913



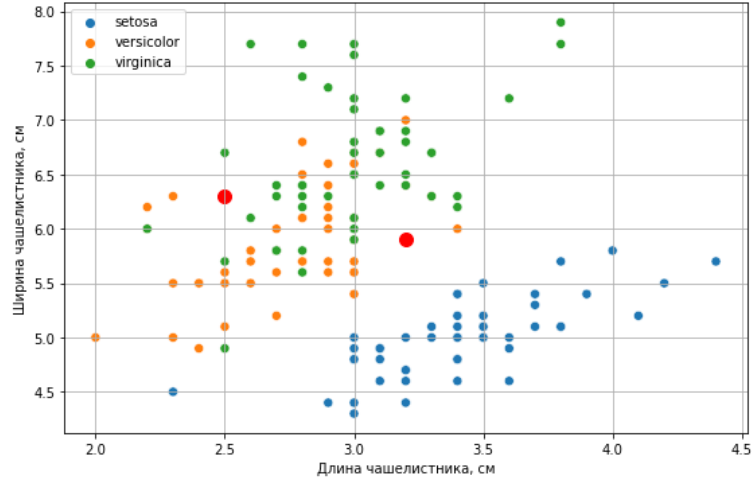
Оценка для модели чашелистика (1 сосед): 0.913



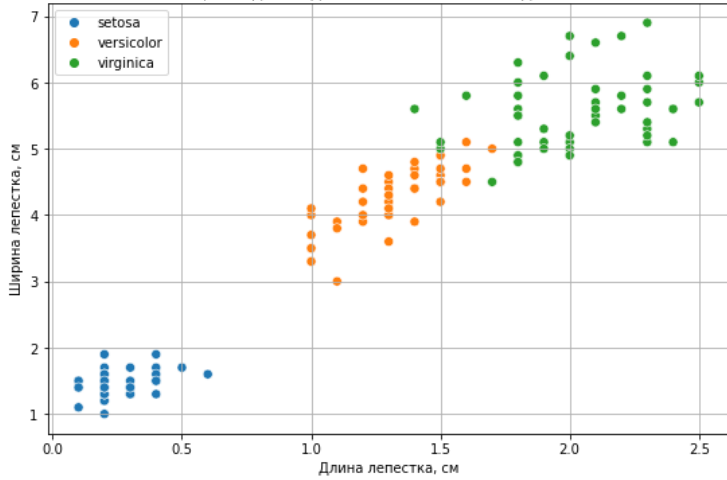
Оценка для моделей лепестка (5 соседей): 0.913



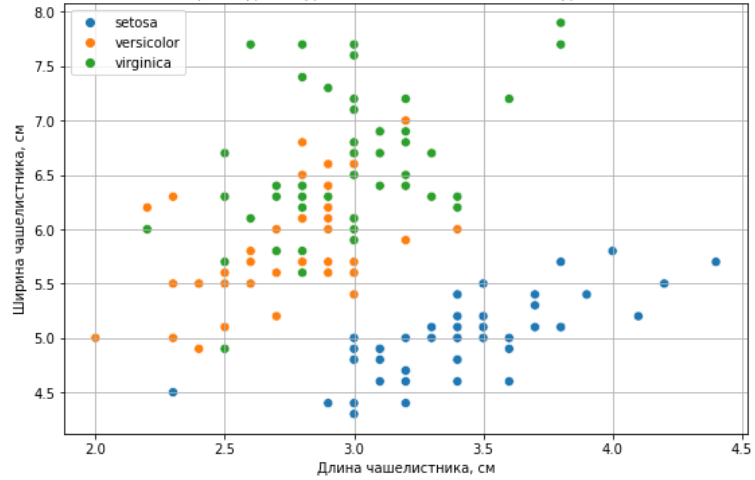
Оценка для моделей чашелистика (5 соседей): 0.913



Оценка для моделей лепестка (10 соседей): 1.0



Оценка для моделей чашелистика (10 соседей): 1.0



3.3.2 Задание

Опредлите набор признаков человека, по аналогии рабочей тетради 1, -например, цвет глаз и конвертируйте его в матрицу признаков

```
In [52]: from sklearn.feature_extraction import DictVectorizer
data_dict=[{"зеленый": 2}, {"голубой": 1, "карий": 1}, {"карий": 2}, {"голубой": 2}]
dictvectorizer=DictVectorizer(sparse=False)
features=dictvectorizer.fit_transform(data_dict)
features
```

```
Out[52]: array([[0., 0., 2., 0.],
 [1., 0., 0., 1.],
 [0., 0., 0., 2.],
 [0., 2., 0., 0.]])
```