Основные понятия и примеры прикладных задач

Используемые пакеты

In [1]:
```
try:
    import google.colab
    IN_COLAB = True
except:
    IN_COLAB = False

if IN_COLAB:
    !wget -q -O requirements.txt
    !pip install -q -r requirements.txt
```

In [2]:
```
from sklearn.linear_model import LogisticRegression
import matplotlib.pyplot as plt
import seaborn as sns
import numpy as np
import pandas as pd
```

```
C:\Users\Kaffedra\Anaconda3\lib\site-packages\statsmodels\tools\_testing.py:19: FutureWarning: pandas.util.t
esting is deprecated. Use the functions in the public API at pandas.testing instead.
  import pandas.util.testing as tm
```

Ирисы Фишера

Выборка взята отсюда: https://archive.ics.uci.edu/ml/datasets/iris (https://archive.ics.uci.edu/ml/datasets/iris)

Загрузка выборки

In [3]:
```
dataset = pd.read_csv('https://raw.githubusercontent.com/andriygav/MachineLearningSeminars/master/sem1/data/i
                      header=None,
                      names=['длина чашелистика', 'ширина чашелистика',
                             'длина лепестка', 'ширина лепестка', 'класс'])
dataset.sample(5, random_state=0)
```

Out[3]:

	длина чашелистика	ширина чашелистика	длина лепестка	ширина лепестка	класс
114	5.8	2.8	5.1	2.4	Iris-virginica
62	6.0	2.2	4.0	1.0	Iris-versicolor
33	5.5	4.2	1.4	0.2	Iris-setosa
107	7.3	2.9	6.3	1.8	Iris-virginica
7	5.0	3.4	1.5	0.2	Iris-setosa

Начало работы с данными

1. Определить множество объектов:
   - Определить размер выборки
   - Определить признаки, которыми описываются объекты
2. Определить множество ответов
3. Определить тип задачи машинного обучения
4. ...

Множество объектов

В данной задачи множество объектов описывается $n = 4$ признаками:

1. Длина чашелистика
2. Ширина чашелистика
3. Длина лепестка

In [4]:
```
print('Размер выборки составляет l={} объектов.'.format(len(dataset)))
```

```
Размер выборки составляет l=150 объектов.
```

Все признаки являются вещественными признаками. Формально объекты $\mathbf{X}$ представляются в следующем виде:

$$\mathbf{X} \in \mathbb{R}^{l \times n},$$

где l число объектов, а n число признаков.

Получаем, что $\mathbf{X}$ это некоторая вещественная матрица размера $l \times n$.

Множество ответов

В данной задаче множество ответов состоит из трех элементов:

1. Iris-virginica
2. Iris-versicolor
3. Iris-setosa

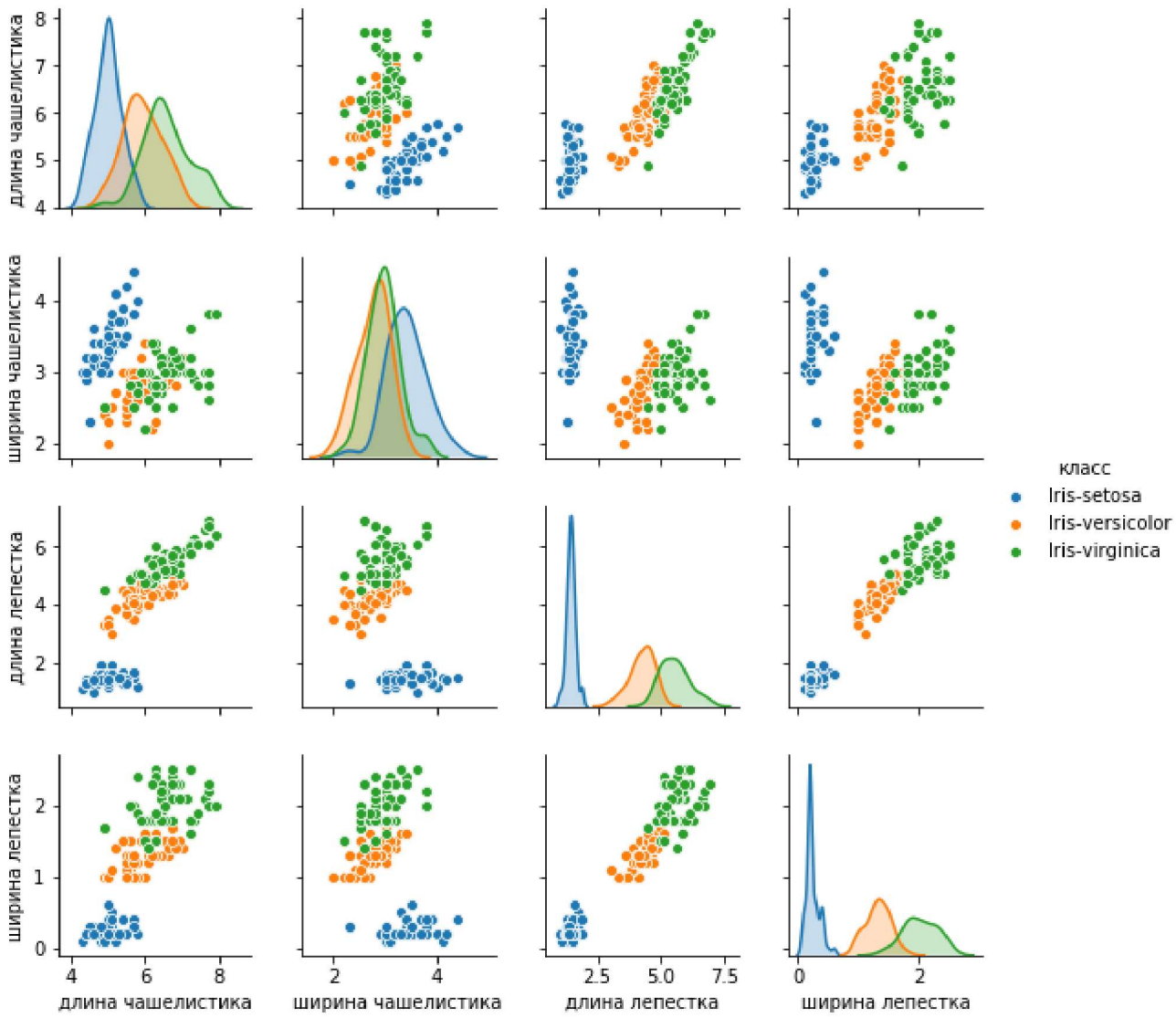
Задача машинного обучения

В нашем случае, так как мощность множества $|\mathbf{y}| = 3 \ll l = 150$ получаем задачу классификации на $M = 3$ класса.

Анализ данных

Сначала проэктируем все объекты на двумерные плоскости, для упрощения анализа

In [5]:
```
sns.pairplot(dataset, hue='класс', height=2)
plt.show()
```



Из рисунка видно, что класс синих точек (Iris-setosa) легко отделяется от двух других цветов. Оранджевые и зеленные точки отделяются не так просто в каждой из проэкций, но все равно можно провести прямую, которая отделит оранджевые точки от зеленых.

Построение модели

Преобразование данных

Как было сказано ранее нам требуется решить задачу классификации на 3 класса. Но для наглядноси рассмотрим бинарную классификацию (классификацию на несколько классов рассмотрим в следующей лекции).

Чтобы исходную задачу преобразовать в задачу бинарной классификации уберем из выборки все объекта класса Iris-setosa.

In [6]:
```
binary_dataset = dataset.drop(index=dataset.index[dataset['класс'] == 'Iris-setosa'])
```

Классы закодируем целыми числами -1 и 1.

In [7]:
```
binary_dataset.loc[dataset['класс'] == 'Iris-versicolor', dataset.columns == 'класс'] = -1
binary_dataset.loc[dataset['класс'] == 'Iris-virginica', dataset.columns == 'класс'] = 1
```

Получаем задачу бинарной классификации.

Модель алгоритмов

Модель алгоритмов $\mathfrak{F}$ в машинном обучении это некоторое множество функций, которые действуют из множества объектов в множество ответов, в нашем случае:

$$\mathfrak{F} = \{f | f : \mathbb{R}^n \to \{-1, 1\}, \text{еще какие-то ограничения}\},$$

обычно $\mathfrak{F}$ это некоторое параметрическое семество функций, тоесть разные функции f отличаются друг от друга только каким-то параметром. Простым примером параметрическим семейством функций для задачи бинарной классификации является семейство линейный классификатор:

$$\mathfrak{F}_{bcl} = \left\{ f\left(\theta, \mathbf{x}\right) = \text{sign}\left(\theta^{\mathsf{T}}\mathbf{x}\right) \middle| \theta \in \mathbb{R}^n \right\}.$$

Функция потерь

Машиное обучение это всегда выбор функции из множества $\mathfrak{F}$. Чтобы выбрать функцию, нужен некоторый критерий по которому она выбирается, то есть нужно упоррядочить все функции от худшей к лучшей. Для этого построем функционал $\mathcal{L}$, который каждой функции $f \in \mathfrak{F}$ ставит в соответствии число из $\mathbb{R}_+$. В машинном обучении обычно функционал качества

водиться как некоторая ошибка на выборке. В общем виде функционал качества можно представить в следующем виде:

$$\mathcal{L}\left(f, \mathbf{X}, \mathbf{y}\right) = \sum_{i=1}^{l} q\left(f, \mathbf{x}_i, y_i\right),$$

где q некоторая функция ошибки на некотором объекте $\mathbf{x}$. Функционал качества $\mathcal{L}$ называется эмперическим риском.

Оптимизационная задача

Далее нужно поставить задачу оптимизации для выбора $f \in \mathfrak{F}$. Здесь все просто, просто минимизируем эмперический риск:

$$\hat{f} = \arg\min_{f \in \mathfrak{F}} \mathcal{L}\left(f, \mathbf{X}, \mathbf{y}\right).$$

Важно! В результе функция $\hat{f}$ зависит от выборки $(\mathbf{X}, \mathbf{y})$, то есть для разных наборов данных оптимальная функция будет различная.

Вернемся к нашей задаче. В нашем случае функционал качества будет иметь следующий вид:

$$\mathcal{L}\left(\theta, \mathbf{X}, \mathbf{y}\right) = \sum_{i=1}^{l} \left[f\left(\theta, \mathbf{x}_i\right) \neq y_i\right],$$

и оптимизационная задача переписывается в виде:

$$\hat{\theta} = \arg\min_{\theta \in \mathbb{R}^n} \sum_{i=1}^{l} \left[f\left(\theta, \mathbf{x}_i\right) \neq y_i\right].$$

И на самом деле в дальнейшем будем решать иммено такие задачи, на поиск оптимального параметра. Само решение задачи линейной бинарной классификации будет на следующей лекции. Сейчас используемся библиотеками для решения данной задачи. Далее в примере будет найден параметр $\hat{\theta}$ не как решение непосредственно этой оптимизационной задачи, а немного измененной, но об этом позже в следующей лекции.

Поиск оптимального вектора параметров

Перейдем к двум матрицам:

1. Матрице объектов $\mathbf{X} \in \mathbb{R}^{l \times (n+1)}$
2. Вектору ответов $\mathbf{y} \in \{-1, 1\}^l$

Заметим, что объекты мы погрузили в пространство более большой размерности, добавив еще один признак, который у всех

In [8]:
```
X = binary_dataset.iloc[:, binary_dataset.columns != 'класс'].values
y = binary_dataset.iloc[:, binary_dataset.columns == 'класс'].values.reshape(-1)
X = np.array(np.hstack([X, np.ones([len(X), 1])]), dtype=np.float64)
y = np.array(y, dtype=np.int64)
```

In [9]:
```
model = LogisticRegression(random_state=0, max_iter=2000)
_ = model.fit(X, y)
```

Получаем вектор оптимальных параметров $\hat{\theta}$

In [10]:
```
model.coef_
```

Out[10]:
```
array([[-3.94426322e-01, -5.13378130e-01,  2.93108661e+00,
         2.41670685e+00, -5.18829563e-04]])
```

Загрузите файл winequality-red.csv (с помощью pd.read_csv).

Последний столбец отвечает за качество вина. Будем считать, что хорошее вино начинается с цифры 7 и выше. Остальное вино будем счиатать некачественным.

Сделайте классификацию вина по данному датасету. Два класса "Хорошее" и "Не качественное". Результатом будем считать уравнение гиперповерхности на гиперпространстве параметров, отделяющее один класс от другого.

In [ ]:
```
```