3.4 Классификация

Наивная байесовская классификация

Набор моделей, которые предлагают быстрые и простые алгоритмы классификации.

Теорема Байеса:

$$P(A|B) = \frac{P(B|A) \cdot P(A)}{P(B)}$$

- $P(A|B)$ — **апостериорная вероятность**. Вероятность того, что гипотеза A верна, после того как мы получили данные B.
- $P(B|A)$ — **правдоподобие**. Вероятность получить данные B, если наша гипотеза A верна.
- $P(A)$ — **априорная вероятность**. Наша изначальная уверенность в гипотезе A до получения каких-либо данных.
- $P(B)$ — **полная вероятность** события.

$$P(B) = \sum_{i=1}^{n} P(B|A_i) \cdot P(A_i)$$

Практическая задача

Пусть есть два шестигранных кубика, и пусть $K_1 = (1, 2, 3, 4, 5, 6)$ и $K_2 = (1, 2, 3, 4, 5, 1)$.

Гипотеза (A): выбран K_1 или K_2

Событие (B): после броска выпала одна из цифр

$$P(K_1|6) = \frac{\frac{1}{6} \cdot \frac{1}{2}}{\frac{1}{12}} = 1 \qquad P(K_2|2) = \frac{\frac{1}{6} \cdot \frac{1}{2}}{\frac{1}{6}} = \frac{1}{2}$$

$$P(K_2|6) = \frac{0 \cdot \frac{1}{2}}{\frac{1}{12}} = 0 \qquad P(K_1|1) = \frac{\frac{1}{6} \cdot \frac{1}{2}}{\frac{1}{4}} = \frac{1}{3}$$

$$P(K_1|2) = \frac{\frac{1}{6} \cdot \frac{1}{2}}{\frac{1}{6}} = \frac{1}{2} \qquad P(K_2|1) = \frac{\frac{2}{6} \cdot \frac{1}{2}}{\frac{1}{4}} = \frac{2}{3}$$

Формула Байеса в машинном обучении

$$P(L|\text{признаки}) = \frac{P(\text{признаки}|L) \cdot P(L)}{P(\text{признаки})}$$

Бинарная классификация: L_1 и L_2

$$P(L_1|\text{признаки}) = \frac{P(\text{признаки}|L_1) \cdot P(L_1)}{P(\text{признаки})}$$

$$P(L_2|\text{признаки}) = \frac{P(\text{признаки}|L_2) \cdot P(L_2)}{P(\text{признаки})}$$

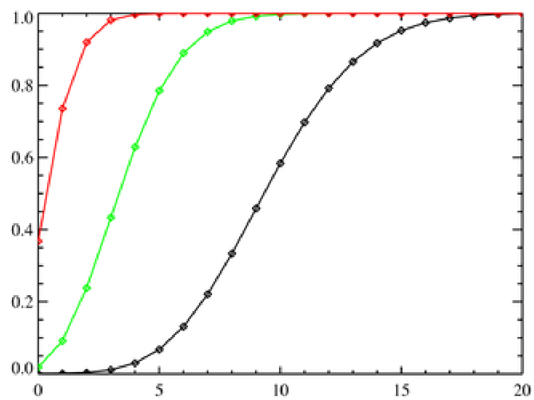
Такая модель называется генеративной моделью, та как она определяет некоторый гипотетический случайный случай который генерирует данные.

Наивные допущения относительно генеративной модели ⇒ можем отыскать грубые приближения для каждого класса.

Гауссовский наивный байесовский классификатор

Допущение состоит в том, что! данные всех категорий взяты из простого нормального распределения.

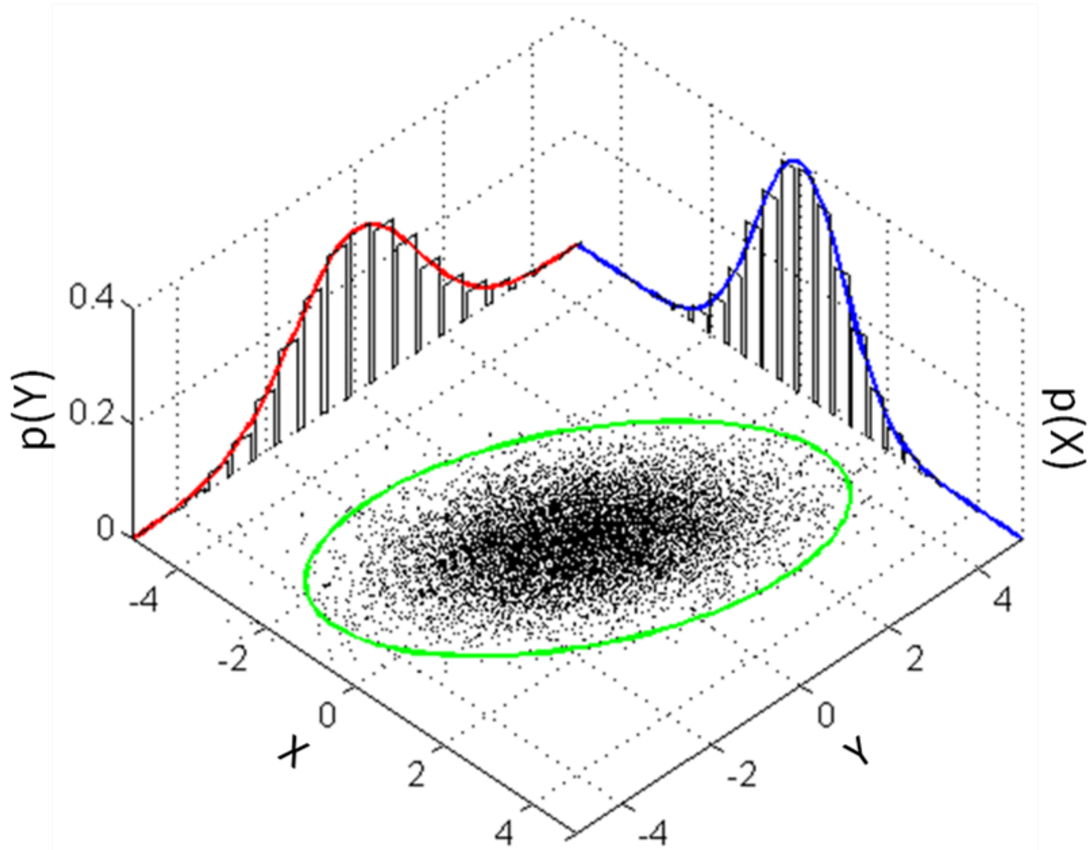
Пуассоновское распределение



$$P(X=k) = \frac{\lambda^k e^{-\lambda}}{k!}$$

- $P(X=k)$—вероятность того, что произойдет ровно **k** событий.
- λ—среднее количество событий, происходящих за данный интервал. Это единственный параметр распределения.
- k—количество событий, вероятность которого мы хотим найти (целое, неотрицательное число: 0, 1, 2, ...).
- e—число Эйлера, математическая константа (≈ 2.71828).
- $k!$—факториал числа k (например, 3!=1·2·3=6).

Двумерное нормальное распределение



$$f(x,y) = \frac{1}{2\pi\sigma_x\sigma_y} \exp\left[-\frac{1}{2}\left(\left(\frac{x-\mu_x}{\sigma_x}\right)^2 + \left(\frac{y-\mu_y}{\sigma_y}\right)^2\right)\right]$$

- $f(x,y)$ — совместная функция плотности вероятности для двух переменных x и y.
- μ_x, μ_y — математические ожидания (средние значения) для x и y соответственно. В ваших заметках они обозначены как $\bar{x}, \bar{y}$.
- σ_x, σ_y — стандартные отклонения для x и y.

Практическая реализация

```
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
import seaborn as sns
from sklearn.naive_bayes import GaussianNB



# ==============================================================
```

```
# ЧАСТЬ 1: КЛАССИФИКАЦИЯ "SETOSA" vs "VERSICOLOR"
# =====================================================================

print("--- Часть 1: Setosa vs Versicolor ---")

# --- Подготовка данных ---
iris = sns.load_dataset("iris")
data = iris[["sepal_length", "petal_length", "species"]]
data_df = data[
    (data["species"] == "setosa") | (data["species"] == "versicolor")
]
print(f"Размер данных: {data_df.shape}")

X = data_df[["sepal_length", "petal_length"]]
y = data_df["species"]

# --- Обучение модели ---
model = GaussianNB()
model.fit(X, y)

# --- Вывод параметров модели (средние и дисперсии) ---
print("\nПараметры для 'setosa':")
print(f"Средние (theta): {model.theta_[0]}")
print(f"Дисперсии (var): {model.var_[0]}")
print("\nПараметры для 'versicolor':")
print(f"Средние (theta): {model.theta_[1]}")
print(f"Дисперсии (var): {model.var_[1]}")

# --- Построение 2D графика ---
plt.figure(figsize=(12, 8))
plt.title('Setosa vs Versicolor: Данные, контуры и граница решений')
plt.xlabel('Длина чашелистика')
plt.ylabel('Длина лепестка')

# Отрисовка исходных точек
data_df_setosa = data_df[data_df["species"] == "setosa"]
data_df_versicolor = data_df[data_df["species"] == "versicolor"]
plt.scatter(
    data_df_setosa["sepal_length"],
    data_df_setosa["petal_length"],
    label='Setosa (реальные)',
    edgecolor='black'
)
plt.scatter(
    data_df_versicolor["sepal_length"],
```

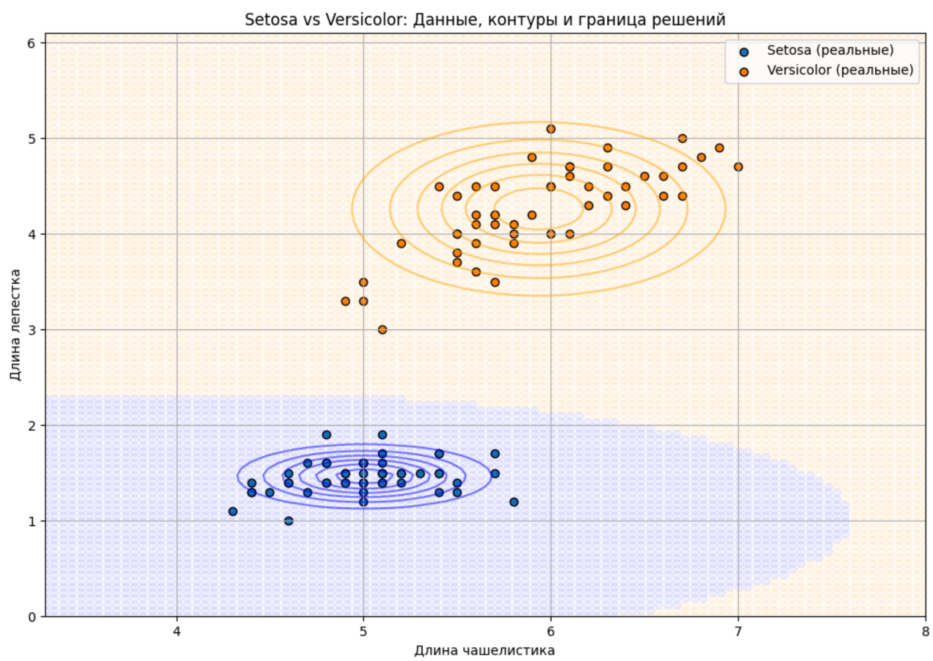
```
    data_df_versicolor["petal_length"],
    label='Versicolor (реальные)',
    edgecolor='black'
)

# Создание сетки для построения контуров и границы
x1_p = np.linspace(
    data_df["sepal_length"].min() - 1, data_df["sepal_length"].max() + 1, 100
)
x2_p = np.linspace(
    data_df["petal_length"].min() - 1, data_df["petal_length"].max() + 1, 100
)
x1_p, x2_p = np.meshgrid(x1_p, x2_p)
X_p = pd.DataFrame(
    np.c_[x1_p.ravel(), x2_p.ravel()],
    columns=["sepal_length", "petal_length"]
)

# Расчет плотности вероятности (PDF) для каждого класса
theta0 = model.theta_[0]
var0 = model.var_[0]

theta1 = model.theta_[1]
var1 = model.var_[1]

# Формула для 2D Гаусса, разбитая на несколько строк для читаемости
z1 = (
    1 / (2 * np.pi * np.sqrt(var0[0] * var0[1]))
    * np.exp(-0.5 * (
        ((x1_p - theta0[0])**2 / var0[0])
        + ((x2_p - theta0[1])**2 / var0[1])
    ))
)
z2 = (
    1 / (2 * np.pi * np.sqrt(var1[0] * var1[1]))
    * np.exp(-0.5 * (
        ((x1_p - theta1[0])**2 / var1[0])
        + ((x2_p - theta1[1])**2 / var1[1])
    ))
)

# Отрисовка контуров
plt.contour(x1_p, x2_p, z1, colors='blue', alpha=0.5)
plt.contour(x1_p, x2_p, z2, colors='orange', alpha=0.5)
```

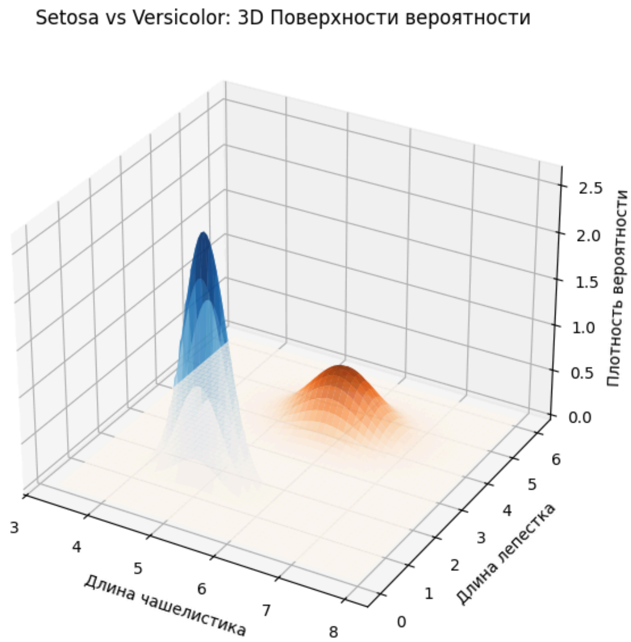
```
# Предсказание и отрисовка разделяющей поверхности
y_p = model.predict(X_p)
X_p["species"] = y_p
plt.scatter(
    X_p[X_p["species"] == "setosa"]["sepal_length"],
    X_p[X_p["species"] == "setosa"]["petal_length"],
    color='blue',
    alpha=0.05
)
plt.scatter(
    X_p[X_p["species"] == "versicolor"]["sepal_length"],
    X_p[X_p["species"] == "versicolor"]["petal_length"],
    color='orange',
    alpha=0.05
)
plt.legend()
plt.grid(True)
plt.show()

# --- Построение 3D графика ---
fig = plt.figure(figsize=(10, 7))
ax = plt.axes(projection='3d')
ax.set_title('Setosa vs Versicolor: 3D Поверхности вероятности')
ax.set_xlabel('Длина чашелистика')
ax.set_ylabel('Длина лепестка')
ax.set_zlabel('Плотность вероятности')

ax.plot_surface(x1_p, x2_p, z1, cmap='Blues', alpha=0.7)
ax.plot_surface(x1_p, x2_p, z2, cmap='Oranges', alpha=0.7)
plt.show()


# ==========================================================================
# ЧАСТЬ 2: КЛАССИФИКАЦИЯ "SETOSA" vs "VIRGINICA"
# ==========================================================================

print("--- Часть 2: Setosa vs Virginica ---")

# --- Подготовка данных ---
data_df = data[
    (data["species"] == "setosa") | (data["species"] == "virginica")
]
print(f"Размер данных: {data_df.shape}")

X = data_df[["sepal_length", "petal_length"]]
```

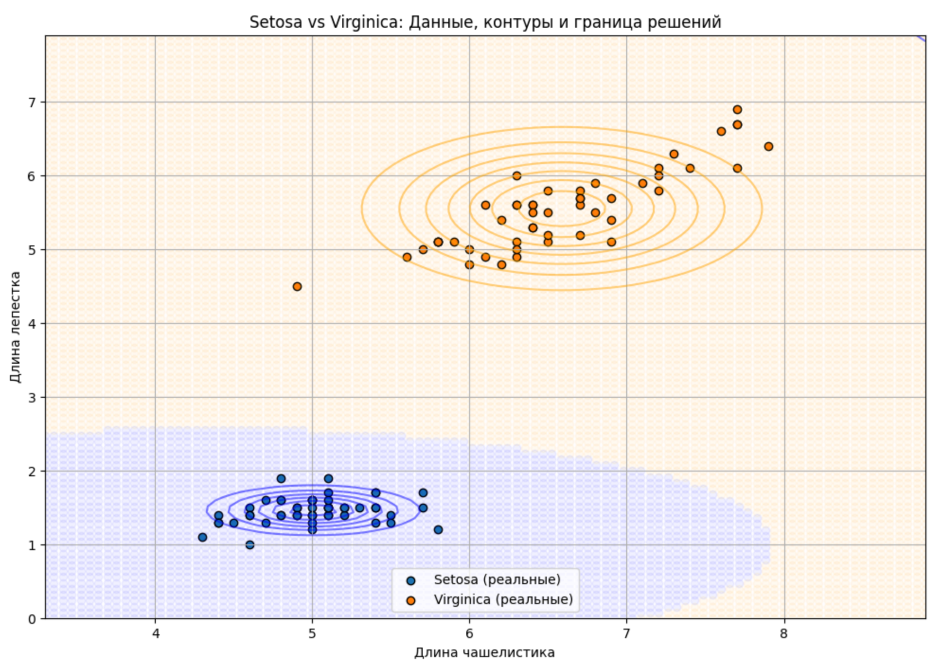
```
y = data_df["species"]

# --- Обучение модели ---
model = GaussianNB()
model.fit(X, y)

# --- Вывод параметров модели ---
print("\nПараметры для 'setosa':")
print(f"Средние (theta): {model.theta_[0]}")
print(f"Дисперсии (var): {model.var_[0]}")
print("\nПараметры для 'virginica':")
print(f"Средние (theta): {model.theta_[1]}")
print(f"Дисперсии (var): {model.var_[1]}")


# --- Построение 2D графика ---
plt.figure(figsize=(12, 8))
plt.title('Setosa vs Virginica: Данные, контуры и граница решений')
plt.xlabel('Длина чашелистика')
plt.ylabel('Длина лепестка')

# Отрисовка исходных точек
data_df_setosa = data_df[data_df["species"] == "setosa"]
data_df_virginica = data_df[data_df["species"] == "virginica"]
plt.scatter(
    data_df_setosa["sepal_length"],
    data_df_setosa["petal_length"],
    label='Setosa (реальные)',
    edgecolor='black'
)
plt.scatter(
    data_df_virginica["sepal_length"],
    data_df_virginica["petal_length"],
    label='Virginica (реальные)',
    edgecolor='black'
)

# Создание сетки
x1_p = np.linspace(
    data_df["sepal_length"].min() - 1, data_df["sepal_length"].max() + 1, 100
)
x2_p = np.linspace(
    data_df["petal_length"].min() - 1, data_df["petal_length"].max() + 1, 100
)
x1_p, x2_p = np.meshgrid(x1_p, x2_p)
```

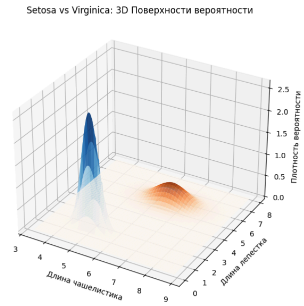
```
X_p = pd.DataFrame(
    np.c_[x1_p.ravel(), x2_p.ravel()],
    columns=["sepal_length", "petal_length"]
)

# Расчет плотности вероятности (PDF) для каждого класса
theta0 = model.theta_[0]
var0 = model.var_[0]

theta1 = model.theta_[1]
var1 = model.var_[1]

z1 = (
    1 / (2 * np.pi * np.sqrt(var0[0] * var0[1]))
    * np.exp(-0.5 * (
        ((x1_p - theta0[0])**2 / var0[0])
        + ((x2_p - theta0[1])**2 / var0[1])
    ))
)
z2 = (
    1 / (2 * np.pi * np.sqrt(var1[0] * var1[1]))
    * np.exp(-0.5 * (
        ((x1_p - theta1[0])**2 / var1[0])
        + ((x2_p - theta1[1])**2 / var1[1])
    ))
)

# Отрисовка контуров
plt.contour(x1_p, x2_p, z1, colors='blue', alpha=0.5)
plt.contour(x1_p, x2_p, z2, colors='orange', alpha=0.5)

# Предсказание и отрисовка разделяющей поверхности
y_p = model.predict(X_p)
X_p["species"] = y_p
plt.scatter(
    X_p[X_p["species"] == "setosa"]["sepal_length"],
    X_p[X_p["species"] == "setosa"]["petal_length"],
    color='blue',
    alpha=0.05
)
plt.scatter(
    X_p[X_p["species"] == "virginica"]["sepal_length"],
    X_p[X_p["species"] == "virginica"]["petal_length"],
    color='orange',
    alpha=0.05
```

```
)
plt.legend()
plt.grid(True)
plt.show()

# --- Построение 3D графика ---
fig = plt.figure(figsize=(10, 7))
ax = plt.axes(projection='3d')
ax.set_title('Setosa vs Virginica: 3D Поверхности вероятности')
ax.set_xlabel('Длина чашелистика')
ax.set_ylabel('Длина лепестка')
ax.set_zlabel('Плотность вероятности')
ax.plot_surface(x1_p, x2_p, z1, cmap='Blues', alpha=0.7)
ax.plot_surface(x1_p, x2_p, z2, cmap='Oranges', alpha=0.7)
plt.show()
```




Setosa vs Versicolor: 3D Поверхности вероятности
Плотность вероятности
Длина чашелистика
Длина лепестка
0.0
0.5
1.0
1.5
2.0
2.5
3
4
5
6
7
8
0
1
2
3
4
5
6


Setosa vs Virginica: Данные, контуры и граница решений
Длина лепестка
Длина чашелистика
Setosa (реальные)
Virginica (реальные)
0
1
2
3
4
5
6
7
4
5
6
7
8


Setosa vs Virginica: 3D Поверхности вероятности
Плотность вероятности
Длина чашелистика
Длина лепестка