Министерство науки и высшего образования Российской Федерации
Федеральное государственное бюджетное образовательное учреждение высшего образования
«Московский государственный технический университет имени Н.Э. Баумана
(национальный исследовательский университет)»
(МГТУ им. Н.Э. Баумана)

ФАКУЛЬТЕТ «Информатика и системы управления»

КАФЕДРА «Программное обеспечение ЭВМ и информационные технологии»

Лабораторная работа № 6 по дисциплине «Методы машинного обучения»

Тема Классификатор «Ирисов Фишера» с использованием байесовского подхода

Студент Сапожков А.М,

Группа ИУ7-23М

Преподаватель Солодовников В.И.

Москва, 2025

Содержание

ВВЕДЕНИЕ

В теории машинного обучения байесовский подход к классификации представляет собой фундаментальную методологию, основанную на строгом математическом аппарате теории вероятностей. Данный подход позволяет не только осуществлять классификацию объектов, но и получать вероятностные оценки принадлежности объекта к каждому из классов.

Целью данной лабораторной работы является изучение линейного дискриминанта Фишера на примере построения классификатора «Ирисов Фишера» с использованием байесовского подхода.

Задачи данной лабораторной работы:

1) осуществить исследование и подготовку исходных данных;
2) построить гистограммы распределения значений для каждого признака и для каждого класса;
3) произвести визуализацию проекций классов на плоскости, где по осям отложены различные комбинации пар признаков;
4) построить матрицы корреляций между различными признаками, как для всей выборки в целом, так и для отдельных классов;
5) построить классификатор с использованием байесовского подхода;
6) оценить точность, полноту, F-меру; построить матрицу ошибок.

1 Аналитическая часть

1.1 Классификация

Классификация (classification) — это задача присвоения меток класса (class label) наблюдениям (Observation) объектам из предметной области. Множество допустимых меток класса конечно. В свою очередь класс — это множество всех объектов с данным значением метки. Требуется построить алгоритм, способный классифицировать (присвоить метку) произвольный объект из исходного множества. Классификация, как правило, на этапе настройки использует обучение с учителем.

1.2 Линейный дискриминант Фишера

Линейный дискриминантный анализ (ЛДА), а также связанный с ним линейный дискриминант Фишера — методы статистики и машинного обучения, применяемые для нахождения линейных комбинаций признаков, наилучшим образом разделяющих два или более класса объектов или событий. Полученная комбинация может быть использована в качестве линейного классификатора или для сокращения размерности пространства признаков перед последующей классификацией.

Линейный дискриминант Фишера в первоначальном значении - метод, определяющий расстояние между распределениями двух разных классов объектов или событий. Он может использоваться в задачах машинного обучения при статистическом (байесовском) подходе к решению задач классификации.

Предположим, что обучающая выборка удовлетворяет помимо базовых гипотез байесовского классификатора также следующим гипотезам:

— классы распределены по нормальному закону;

— матрицы ковариаций классов равны.

Тогда статистический подход приводит к линейному дискриминанту, и именно этот алгоритм классификации в настоящее время часто понимается под термином *линейный дискриминант Фишера*.

2 Технологическая часть

2.1 Средства реализации

В качестве языка программирования для реализации алгоритмов был выбран язык программирования Python ввиду наличия библиотек для обучения регрессионных моделей, таких как sklearn и numpy.

2.2 Реализация алгоритмов

На листинге 2.1 представлена реализация алгоритма фильтрации спама с использованием наивного байесовского классификатора.

Листинг 2.1 — Классификация «Ирисов Фишера» с использованием байесовского подхода

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn import datasets
from sklearn.model_selection import train_test_split
from sklearn.metrics import confusion_matrix, classification_report,
   matthews_corrcoef
from sklearn.discriminant_analysis import LinearDiscriminantAnalysis
from sklearn.preprocessing import LabelEncoder

from google.colab import drive
drive.mount('/content/drive')

df = pd.read_excel('/content/drive/MyDrive/Colab Notebooks/ml_lab_06/ЛР6_Ирисы
   Фишера.xlsx')
X = df[['Длина чашелистика', 'Ширина чашелистика', 'Длина лепестка', 'Ширина
   лепестка']]
y = df['Вид ириса']

label_encoder = LabelEncoder()
y_encoded = label_encoder.fit_transform(y)

plt.figure(figsize=(12, 8))
for i, feature in enumerate(X.columns):
  plt.subplot(2, 2, i+1)
  for species in df['Вид ириса'].unique():
    subset = df[df['Вид ириса'] == species]
    plt.hist(subset[feature], label=species, alpha=0.5)
```

```
  plt.xlabel(feature)
  plt.ylabel('Count')
  plt.legend(loc='upper right')
plt.tight_layout()
plt.show()

pairs = [(i, j) for i in range(4) for j in range(i+1, 4)]
plt.figure(figsize=(15, 9))
for idx, (i, j) in enumerate(pairs):
  plt.subplot(2, 3, idx+1)
  for species in df['Вид ириса'].unique():
    species_mask = df['Вид ириса'] == species
    plt.scatter(X.loc[species_mask, X.columns[i]], X.loc[species_mask
       , X.columns[j]], label=species)
  plt.xlabel(X.columns[i])
  plt.ylabel(X.columns[j])
  plt.legend()
plt.tight_layout()
plt.show()

# Commented out IPython magic to ensure Python compatibility.
from mpl_toolkits import mplot3d
# %matplotlib inline
import numpy as np
import matplotlib.pyplot as plt
plt.figure(figsize=(7, 7))
ax = plt.axes(projection='3d')
ax.set_xlabel('Длина лепестка')
ax.set_ylabel('Ширина лепестка')
ax.set_zlabel('Ширина чашелистика')

for species in df['Вид ириса'].unique():
  species_mask = df['Вид ириса'] == species
  dims = [X[species_mask][label] for label in ['Длина лепестка', 'Ширина
     лепестка', 'Ширина чашелистика']]
  ax.scatter(dims, label=species)

plt.figure(figsize=(5, 4))
sns.heatmap(X.corr(), annot=True, cmap='coolwarm')
plt.title('Overall Correlation Matrix')
plt.show()
```

```
for species in df['Вид ириса'].unique():
  species_data = X[df['Вид ириса'] == species]
  plt.figure(figsize=(5, 4))
  sns.heatmap(species_data.corr(), annot=True, cmap='coolwarm')
  plt.title(f'Correlation Matrix for {species}')
  plt.show()

X_train, X_test, y_train, y_test = train_test_split(X, y_encoded,
   test_size=0.3, random_state=7)
lda = LinearDiscriminantAnalysis()
lda.fit(X_train, y_train)
y_pred = lda.predict(X_test)

print("\nClassification Report:")
print(classification_report(y_test, y_pred, target_names=
   label_encoder.classes_))
print("\nAdditional Metrics:")
mcc = matthews_corrcoef(y_test, y_pred)
print(f"MCC: {mcc:.4f}")

plt.figure(figsize=(8, 6))
cm = confusion_matrix(y_test, y_pred)
sns.heatmap(cm, annot=True, fmt='d', cmap='Blues',
            xticklabels=label_encoder.classes_,
            yticklabels=label_encoder.classes_)
plt.title('Confusion Matrix')
plt.xlabel('Predicted')
plt.ylabel('True')
plt.show()

probabilities = lda.predict_proba(X_test)

plt.figure(figsize=(10, 6))
for i, species in enumerate(df['Вид ириса'].unique()):
  plt.hist(probabilities[:, i], label=species, alpha=0.5)
plt.title('Posterior Probability Distributions')
plt.xlabel('Probability')
plt.ylabel('Count')
plt.legend()
plt.show()
```

3 Исследовательская часть

3.1 Среда для тестирования

Для тестирования разработанного алгоритма применялась облачная платформа Google Colab, не требующая установки ПО на локальный компьютер.

3.2 Исследование признаков

3.3 Результат классификации

Листинг 3.1 — Отчёт по результатам классификации

```
Classification Report:
precision    recall  f1-score   support

setosa              1.00      1.00      1.00        12
versicolor          0.94      0.94      0.94        16
virginica           0.94      0.94      0.94        17

accuracy                              0.96        45
macro avg           0.96      0.96      0.96        45
weighted avg        0.96      0.96      0.96        45

Additional Metrics:
MCC: 0.9326
```

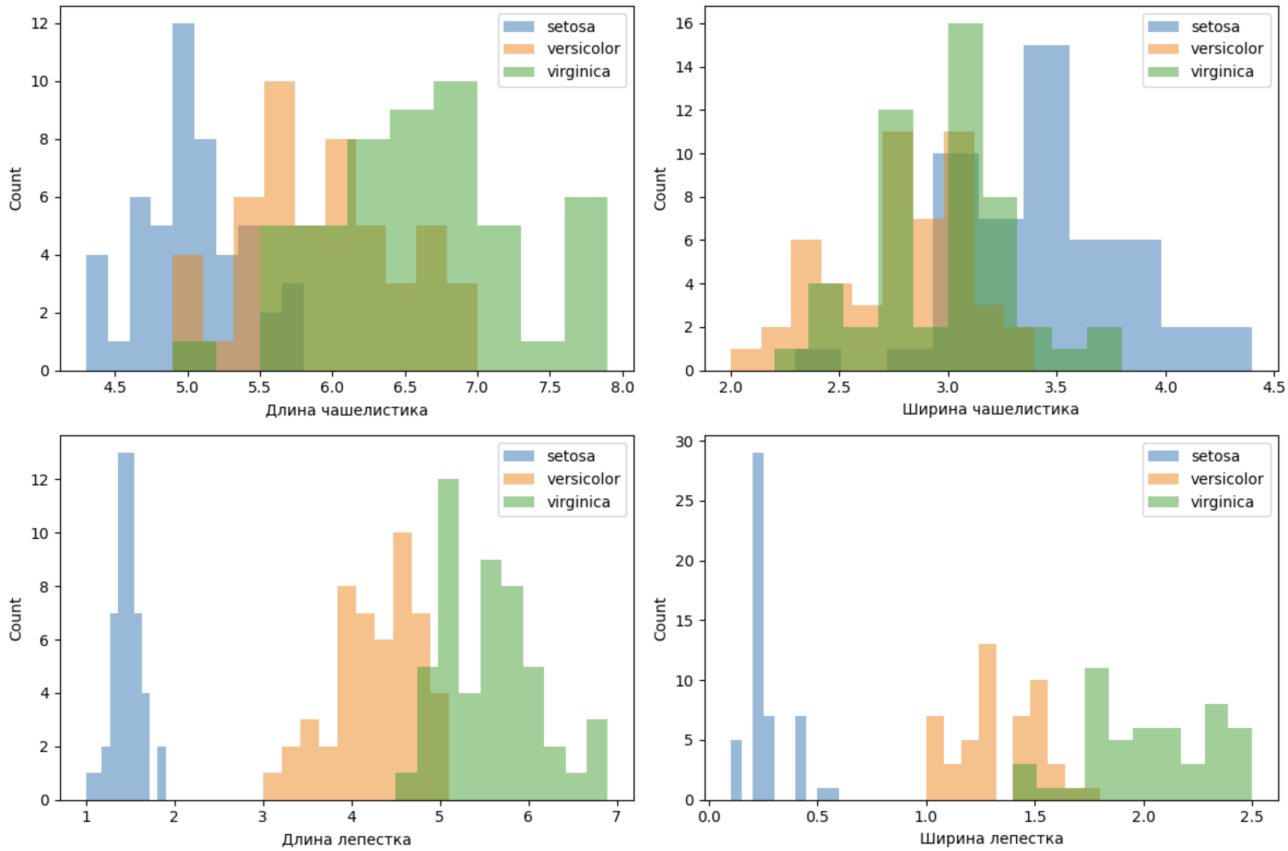

Рисунок 3.1 — Гистограммы распределения значений для каждого признака и для каждого класса

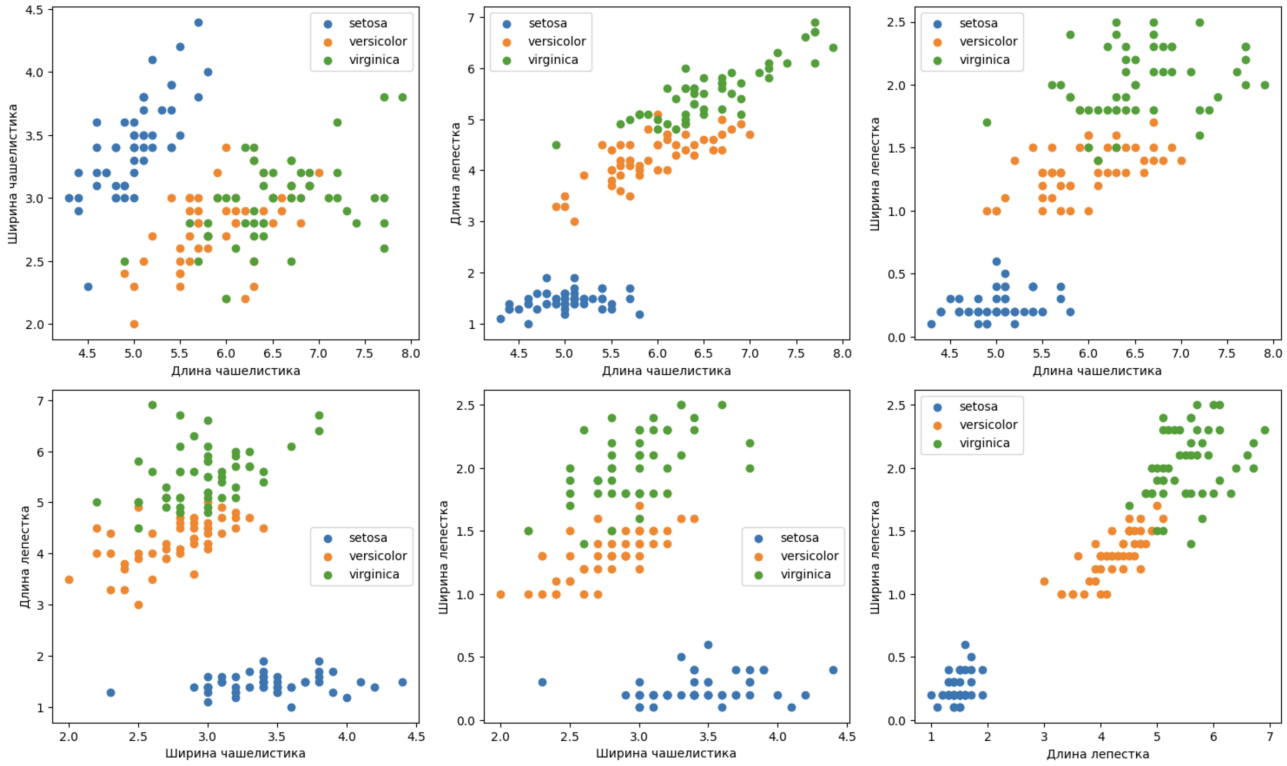

Рисунок 3.2 — Визуализация проекций классов на все возможные пары признаков

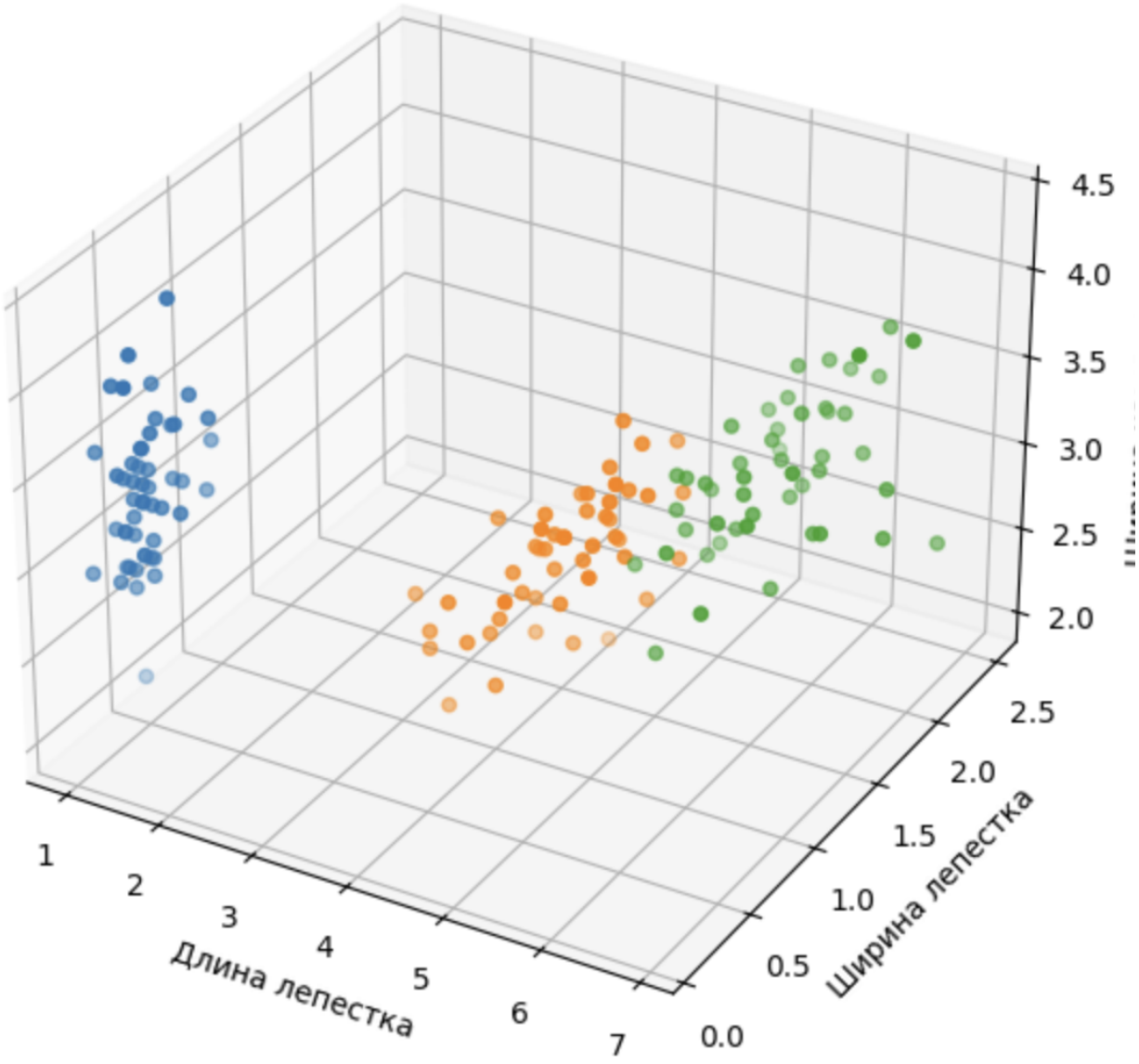

Рисунок 3.3 — Визуализация проекций классов на пространство признаков «Длина лепестка», «Ширина лепестка», «Ширина чашелистника»

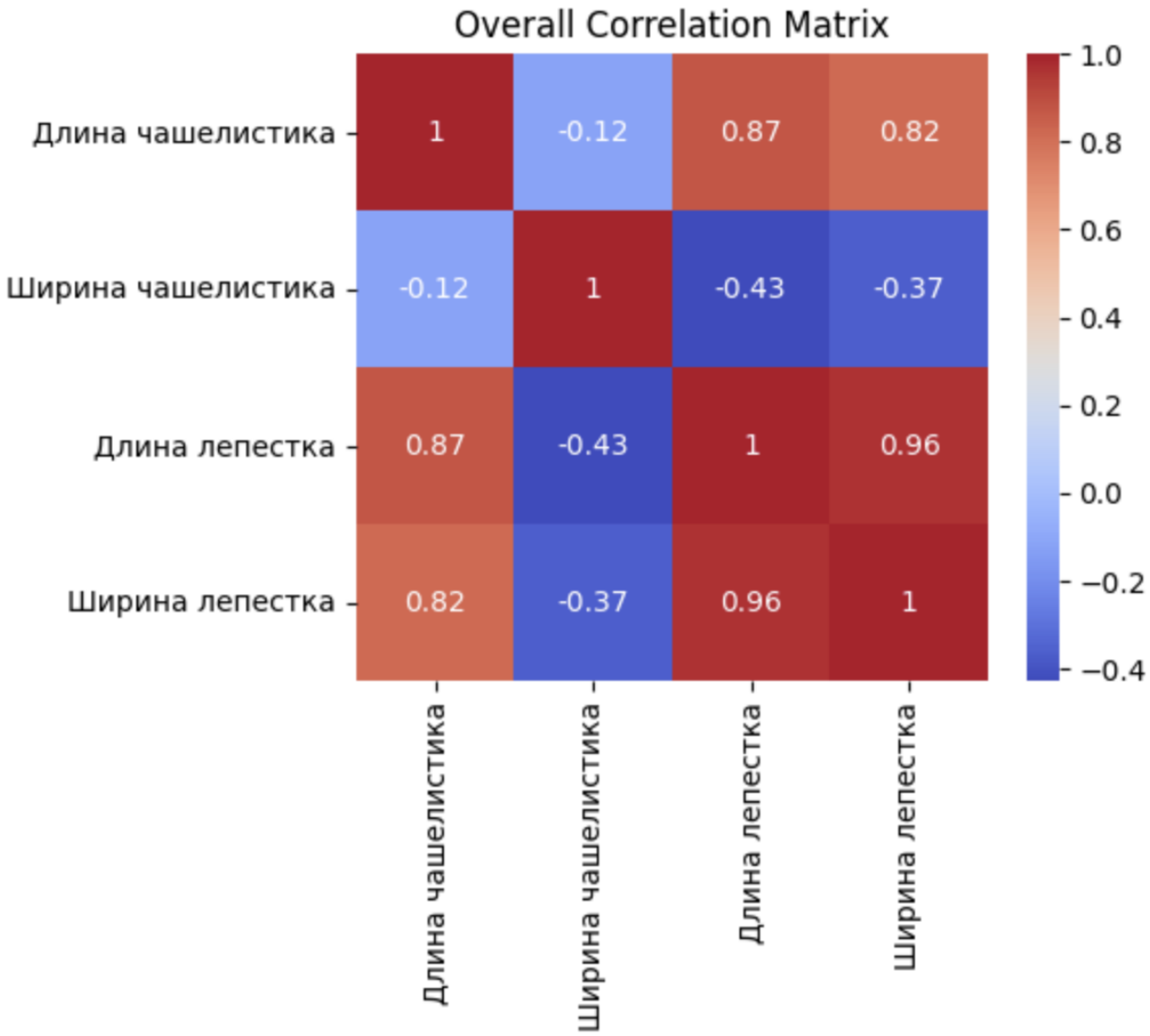

Рисунок 3.4 — Матрица корреляции признаков по всем классам

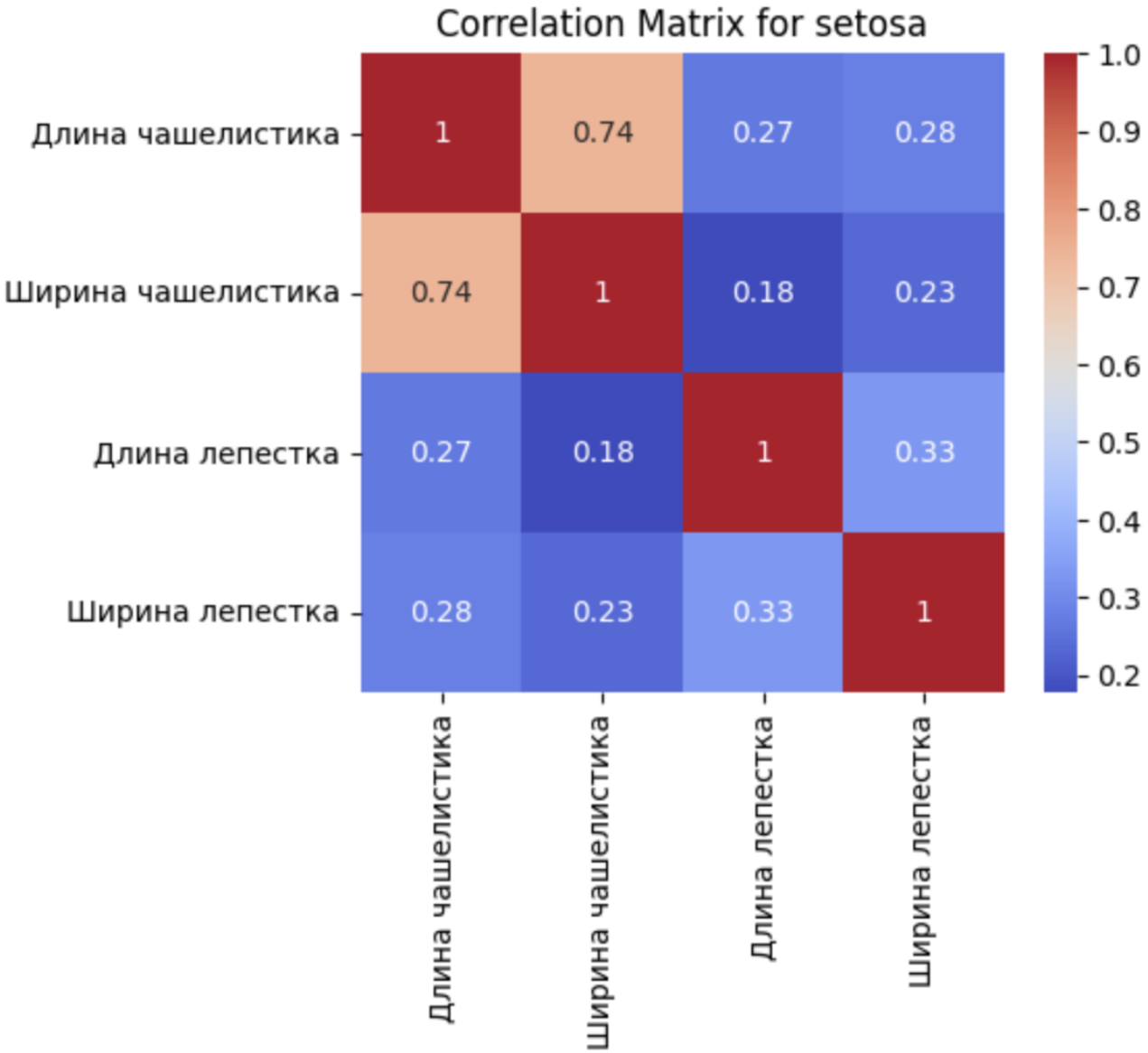

Рисунок 3.5 — Матрица корреляции признаков для класса «setosa»

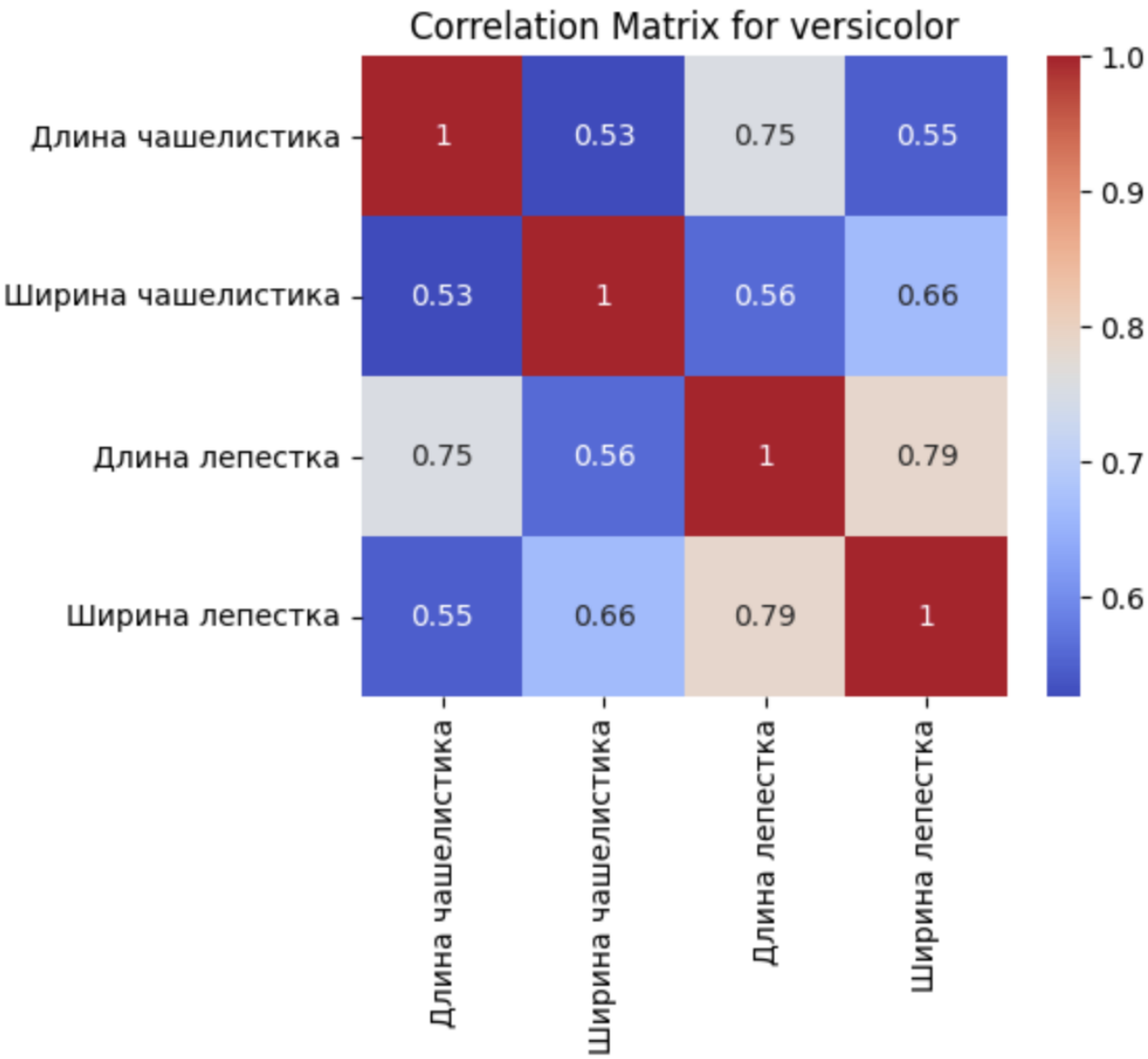

Рисунок 3.6 — Матрица корреляции признаков для класса «versicolor»

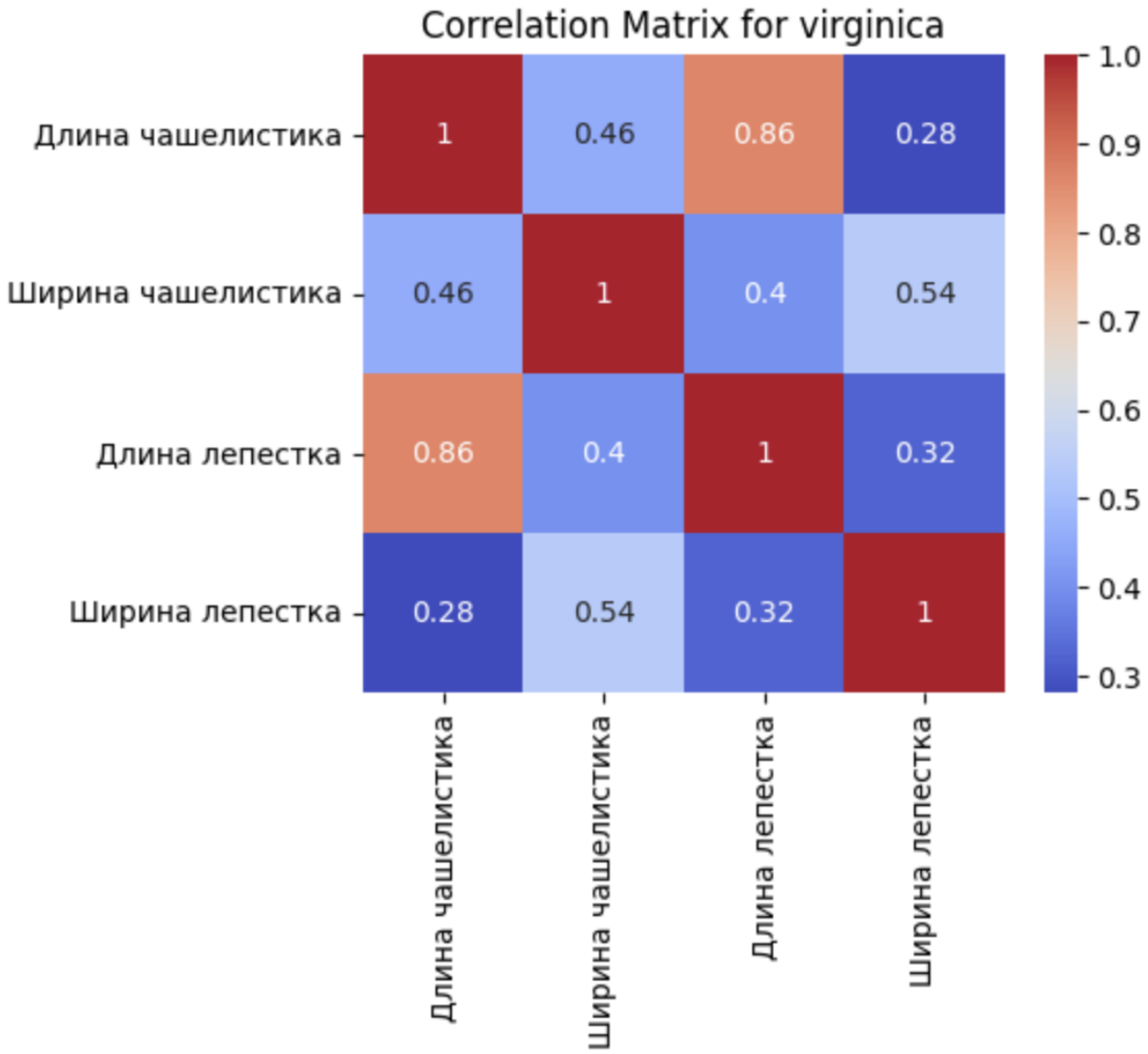

Рисунок 3.7 — Матрица корреляции признаков для класса «virginica»

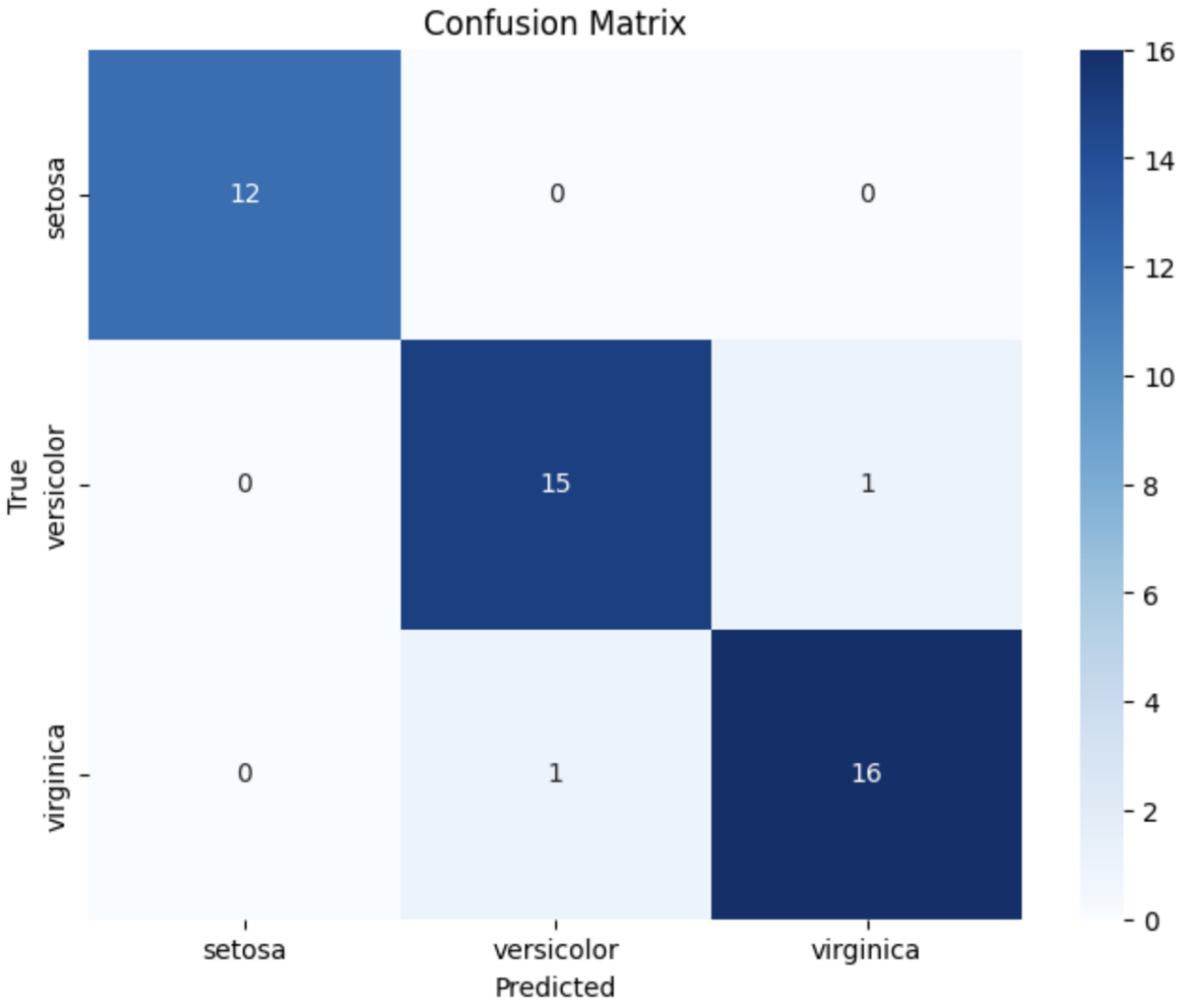

Рисунок 3.8 — Матрица ошибок

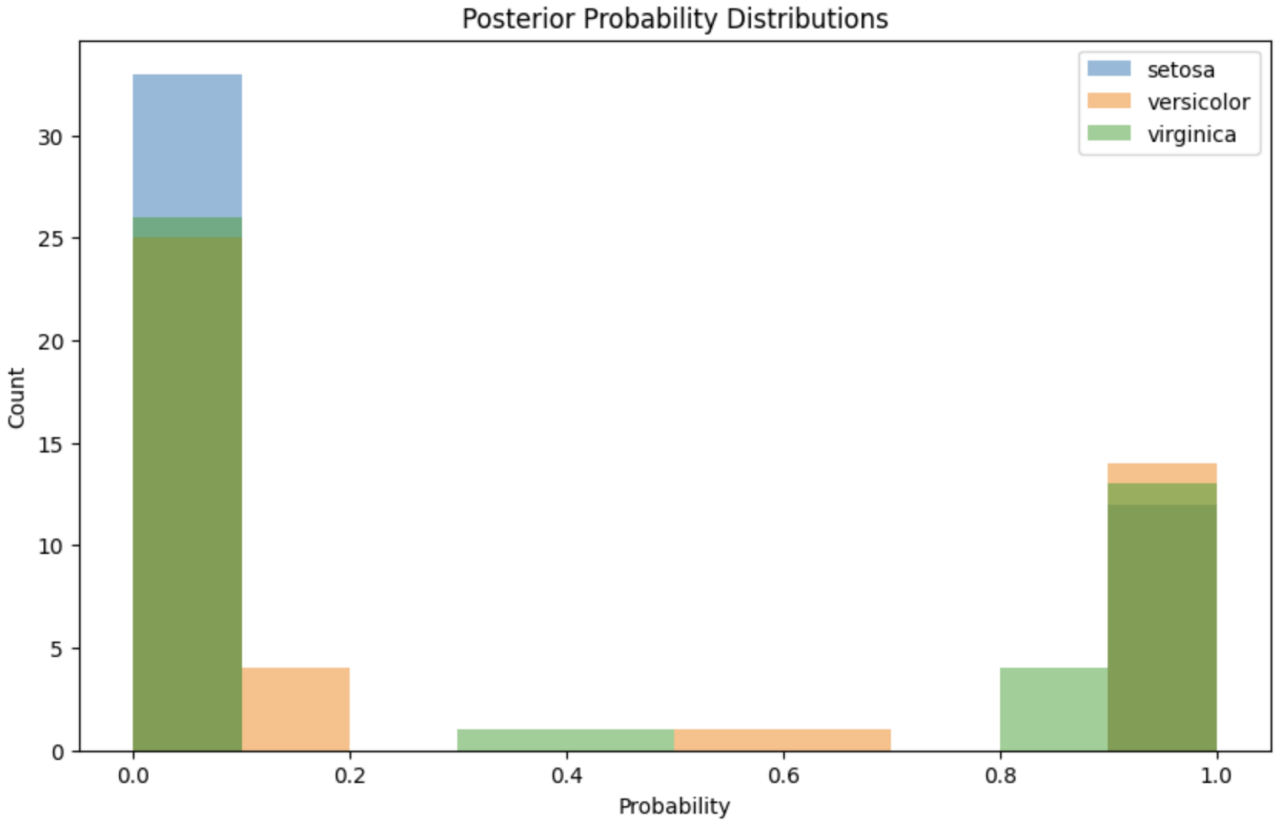

Рисунок 3.9 — Апостериорное распределение вероятностей для каждого класса

ЗАКЛЮЧЕНИЕ

В рамках лабораторной работы было проведено изучение линейного дискриминанта Фишера на примере построения классификатора «Ирисов Фишера» с использованием байесовского подхода.

1. Осуществлено исследование и подготовку исходных данных.
2. Построены гистограммы распределения значений для каждого признака и для каждого класса.
3. Произведены визуализацию проекций классов на плоскости, где по осям отложены различные комбинации пар признаков.
4. Построены матрицы корреляций между различными признаками, как для всей выборки в целом, так и для отдельных классов.
5. Построен классификатор с использованием байесовского подхода.
6. Оценена точность, полнота, F-мера. Построена матрица ошибок.

Для построенного классификатора метрика MCC составила 0.84.