

**Московский государственный технический  
университет им. Н.Э. Баумана**

Факультет «Информатика и системы управления»  
Кафедра ИУ5 «Системы обработки информации и управления»

Курс «Парадигмы и конструкции языков программирования»  
Лабораторная работа №5  
«Реализация алгоритма гауссовского байесовского классификатора на  
Python»

Выполнил:

Студент ИУ5-34Б  
ИУ5

Бурдуковский И.О.

Подпись и дата:

Проверил:

Преподаватель каф.

Гапанюк Ю. Е.

Подпись и дата:

Москва, 2023 г.

## Задание

Реализация алгоритма гауссовского байесовского классификатора на Python.

## Код программы

```
import numpy as np
from scipy.stats import multivariate_normal

class GaussianBayesianClassifier:
    def fit(self, X, y):
        self.classes = np.unique(y)
        self.class_priors = {}
        self.mean_vectors = {}
        self.cov_matrices = {}
        for c in self.classes:
            X_c = X[y == c]
            self.class_priors[c] = len(X_c) / len(X)
            self.mean_vectors[c] = np.mean(X_c, axis=0)
            self.cov_matrices[c] = np.cov(X_c, rowvar=False)

    def predict(self, X):
        predictions = []
        for x in X:
            posteriors = []
            for c in self.classes:
                prior = self.class_priors[c]
                mean = self.mean_vectors[c]
                cov = self.cov_matrices[c]
                likelihood = multivariate_normal(mean=mean, cov=cov).pdf(x)
                posterior = prior * likelihood
                posteriors.append(posterior)
            predictions.append(np.argmax(posteriors))
        return np.array(predictions)

X_train = np.array([[1, 2], [2, 3], [3, 4], [4, 5], [1, 3], [2, 4]])
y_train = np.array([0, 0, 1, 1, 0, 1])

classifier = GaussianBayesianClassifier()
classifier.fit(X_train, y_train)

X_test = np.array([[1.5, 2.5], [3.5, 4.5]])
predictions = classifier.predict(X_test)
print(predictions)
```

## Результаты

```
[0 1]
```