Санкт-Петербургский Политехнический Университет имени Петра Великого
Институт Прикладной Математики и Механики
Кафедра "Прикладная Математика"

Отчет по лабораторной работе №2

по дисциплине

"Многомерный Статистический Анализ"

Выполнил студент:
Тырыкин Я. А.
группа 5030102/80401
Преподаватель:
Павлова Л. В.

Санкт-Петербург
2022

Содержание

1 Формулировка задачи

В данной работе требуется построить классификатор на основе дискриминантной функции и с его помощью провести классификацию объектов на 2 класса. Классификация проводится в два этапа: обучение и тестирование классификатора на модельных данных (распределенных по известному закону, с заранее известными параметрами) и обучение и тестирование классификатора на реальных данных, описание которых приводится ниже.

2 Описание данных для тестирования

Модельные данные представлены 3-мерными нормальными распределениями. Для исследования классификатора распределения берутся с такими параметрами, чтобы в одном случае данные хорошо разделялись на классы, а в другом - плохо.

Реальные данные представлены сведениями по кредитам в Германии. Данные содержат 1000 размеченных экземпляров данных с 25 признаками: 24 качественных и количественных атрибута, а последний - бинарный признак, обозначающий принадлежность экземпляра одному из двух классов. Данные располагаются по данной **ссылке**. Сам набор данных содержится в файле "german.data-numeric описание приведенных данных - "german.doc".

3 Построение и тестирование классификатора на модельных данных

Строим 2 набора данных,распределенному по 3-мерному нормальному закону. В первом случае используются нормальные распределения со следующими параметрами:

$$\mathbf{x} \sim N(\mu^{(1)}, \Sigma)$$

$$\mu^{(1)} = [1, 2, 3]$$

$$\mathbf{x} \sim N(\mu^{(2)}, \Sigma)$$

$$\mu^{(2)} = [3, 4, 5]$$

$$\Sigma = \begin{bmatrix} 1 & -0.1 & 0.2 \\ -0.1 & 3 & 0.3 \\ 0.2 & 0.3 & 4 \end{bmatrix}$$

где $\mu^{(1)}$ и $\mu^{(2)}$ - математические ожидания для нормальных случайных величин, использованных для генерации данных 2 классов, Σ - матрица ковариации для тех же 3-ехмерных нормальных распределений. В случае применения такой генерации данные хорошо разделяются на классы.

Полученные модельные данные для обучения и тестирования модели приведены ниже:

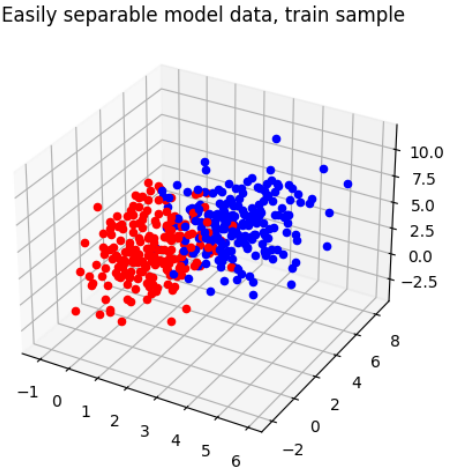

Рис. 1: Распределение обучающих выборок хорошо разделимых данных

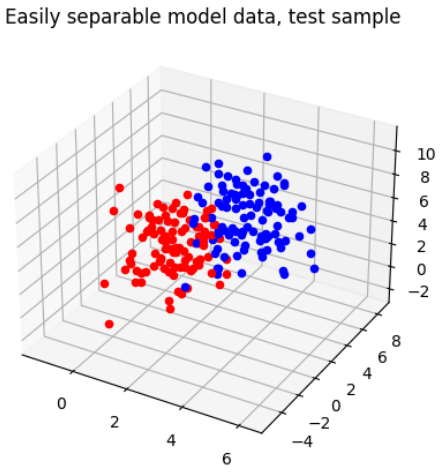

Рис. 2: Распределение тестовых выборок хорошо разделимых данных

Во втором случае используются нормальные распределения со следующими параметрами:

$$\mathbf{x} \sim N(\mu^{(1)}, \Sigma)$$

$$\mu^{(1)} = [1, 2, 3]$$

$$\mathbf{x} \sim N(\mu^{(2)}, \Sigma)$$

$$\mu^{(2)} = [1.2, 2.3, 3.4]$$

$$\Sigma = \begin{bmatrix} 1 & -0.1 & 0.2 \\ -0.1 & 3 & 0.3 \\ 0.2 & 0.3 & 4 \end{bmatrix}$$

Все величины определены аналогично предыдущему примеру. В случае использования таких параметров распределений данные плохо разделяются на классы.

Полученные модельные данные для обучения и тестирования модели приведены ниже:

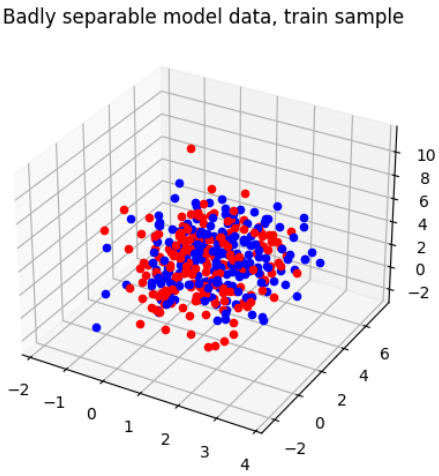

Рис. 3: Распределение обучающих выборок плохо резделимых данных

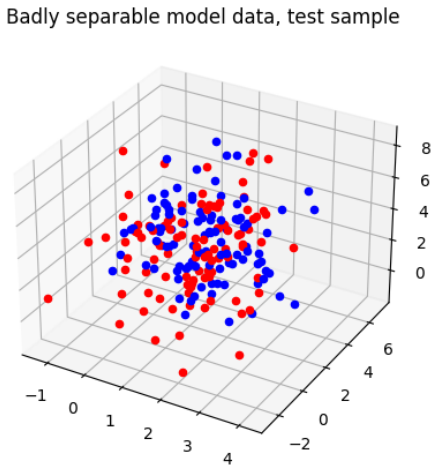

Рис. 4: Распределение тестовых выборок плохо резделимых данных

Для обоих наборов данных строятся 2 обучающих выборки размера $n_1, n_2 = 200$ (1 выборка строится одним 3-мерным нормальным распределением из описанных выше) и 1 тестовая выборка размера 100.

Классификатор строим на основе дискриминантной функции:

$$z(\mathbf{x}) = z = \alpha_1 x_1 + ... + \alpha_p x_p, p = 3 \tag{1}$$

Будем полагать, что если элемент пространства R^p - p-мерный вектор $\mathbf{x}$ - принадлежит классу D_1, то дискриминантная функция ищется как некоторая случайная

величина $z \sim N(\xi_1, \sigma_z^2)$ с параметрами ξ_1 и σ_z^2, определеямыми формулами:

$$\xi_1 = \sum_{j=1}^{p} \alpha_j \mu_j^{(1)} \tag{2}$$

$$\sigma_z^2 = \sum_{m=1}^{p} \sum_{j=1}^{p} \alpha_m \sigma_{mp} \alpha_j \tag{3}$$

Аналогичным образом на основе дискриминантной функции задается принадлежность экземпляра данных $\mathbf{x}$ второму классу D_2.

Параметры α_i же можно найти как решение следующей СЛАУ:

$$\begin{cases} \alpha_1\sigma_{11} + \alpha_2\sigma_{12} + ... + \alpha_p\sigma_{1p} = \mu_1^{(1)} - \mu_1^{(2)} \\ ... \\ \alpha_1\sigma_{p1} + \alpha_2\sigma_{p2} + ... + \alpha_p\sigma_{pp} = \mu_p^{(1)} - \mu_p^{(2)} \end{cases} \tag{4}$$

$$\alpha = \Sigma^{-1}(\mu^{(1)} - \mu^{(2)}) \tag{5}$$

3.1 Построение классификатора на основе хорошо разделенных данных

На основе тестовых выборок OB1 и OB2 размеров n_1 и $n_2 = 200$ получаем следующую дискриминантную функцию:

$$z(\mathbf{x}) = -2.09785x_1 - 0.79140053x_2 - 0.23316792x_3$$

Она уже, в свою очередь, используется для классификации набора данных на два класса по следующему правилу:

$$\begin{cases} z(\mathbf{x}) \leq c \rightarrow & \mathbf{x} \in D_1 \\ z(\mathbf{x}) > c \rightarrow & \mathbf{x} \in D_2 \end{cases} \tag{6}$$

$$c = \frac{\xi_1 + \xi_2}{2} \tag{7}$$

Характеристическая функция уже посчитана, найдем численные значения остальных параметров классификатора:

$$\xi_1 = \sum_{j=1}^{p} \alpha_j \mu_j^{(1)} = -4.257 \tag{8}$$

$$\xi_2 = \sum_{j=1}^{p} \alpha_j \mu_j^{(2)} = -10.81 \tag{9}$$

$$c = \frac{\xi_1 + \xi_2}{2} = -7.5339 \tag{10}$$

3.1.1 Применение классификатора на тестовой выборке

Результаты применения классификатора здесь и далее представим в виде четырехпольных таблиц сопряженности. В процессе классификации элементов тестовой выборки было получено следующее разделение на два класса:

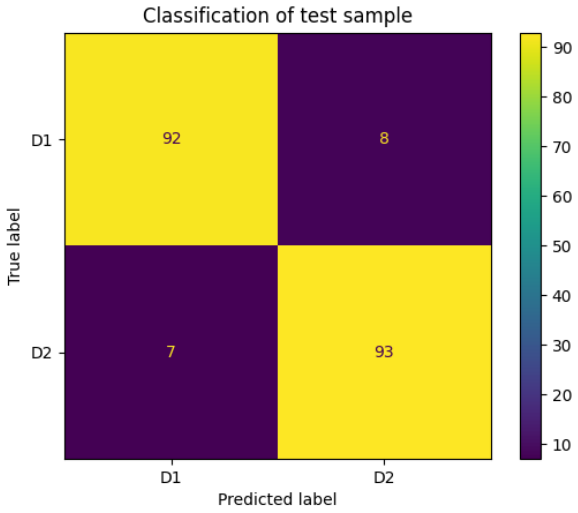

Рис. 5: Результаты классификации тестовой выборки

3.1.2 Применение классификатора на обучающей выборке

В процессе классификации элементов тренировочной выборки было получено следующее разделение на два класса:

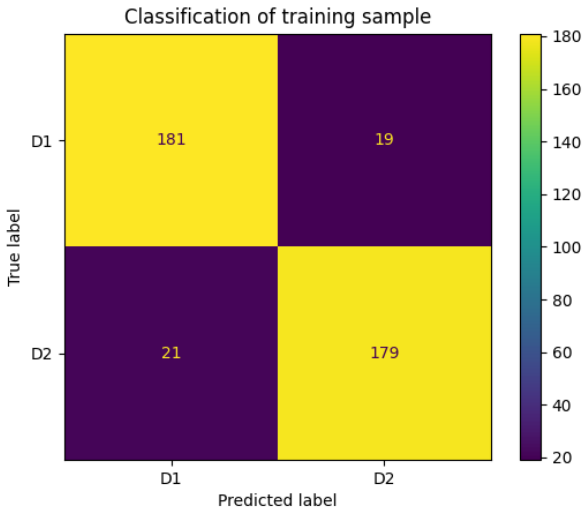

Рис. 6: Результаты классификации обучающих выборок

3.1.3 Оценки расстояния Махаланобиса и вероятностей ошибочной классификации

Оценки вероятностей ошибочной классификации будем строить 3 способами. Первый из них - с применением расстояния Махалонобиса:

$$\Delta^2 = \frac{(\xi_1 - \xi_2)^2}{\sigma_z^2} \tag{11}$$

$$D_H^2 = \frac{n_1 + n_2 - p - 3}{n_1 + n_2 - 2}\Delta^2 - p(\frac{1}{n_1} + \frac{1}{n_2}) \tag{12}$$

Формула оценки ошибочной классификации с применением данного расстояния и интеграла Лапласа:

$$P(2|1) = \Phi(\frac{K - \frac{\Delta^2}{2}}{\Delta}) \tag{13}$$

$$P(1|2) = \Phi(\frac{-K - \frac{\Delta^2}{2}}{\Delta}) \tag{14}$$

$$K = ln(\frac{q_2 c(1|2)}{q_1(2|1)}) \tag{15}$$

где q_1 и q_2 - вероятности принадлежности элемента $\mathbf{x}$ классу D_1 и D_2 соответственно, $c(1|2)$ и $c(2|1)$ - стоимости неправильной классификации элемента $\mathbf{x}$. В случае данной работы данные вероятности и стоимости полагаются равными друг другу: $q_1 = q_2, c(1|2) = c(2|1)$. В таком случае логарифм обращается в 0, а формулы $P(1|2)$

и $P(2|1)$ принимают вид:

$$P(2|1) = P(1|2) = \Phi(-\frac{\Delta}{2}) \tag{16}$$

$$\Phi(y) = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{y} e^{-\frac{t^2}{2}} dt \tag{17}$$

Второй метод оценки - при помощи относительного числа неправильно классифицированных объектов:

$$\hat{P}(2|1) = \frac{m_1}{n_1}$$

$$\hat{P}(1|2) = \frac{m_2}{n_2}$$

где n_1, n_2 - истинные количества объектов в первом и втором классе соответственно, а m_1, m_2 - числа неправильно классифицированных объектов и ошибочно отправленных в класс 1 и 2 соответственно.

Данные оценки являются смещенными, поэтому основное внимание будем обращать на оценку, полученную при помощи **процедуры скользящего экзамена**:

for x in OB1 **do**
  $x \leftarrow OB1.delete(x)$
  $\alpha, c \leftarrow count\_z\_parameters(OB1, OB2, p)$
  $prediction \leftarrow classification(z, x)$
  if then$z(x) > c$
    $prediction.append(1)$
  else
    $prediction.append(2)$
  end if
end for
for x in OB2 **do**
  $x \leftarrow OB2.delete(x)$
  $\alpha, c \leftarrow count\_z\_parameters(OB1, OB2, p)$
  if then$z(x) > c$
    $prediction.append(1)$
  else
    $prediction.append(2)$
  end if

end for

return *prediction*

Применяя все выше сказанное, получаем следующие оценки ошибочной классифицкации для хорошо разделимых данных в обучающих и тестовой выборках:

- Оценка на основании доли неверно классифицированных элементов тестовой выборки:

$$\hat{P}(1|2) = 0.07, \hat{P}(2|1) = 0.08$$

- Оценка на основании доли неверно классифицированных элементов обучающих выборок:

$$\hat{P}(1|2) = 0.105, \hat{P}(2|1) = 0.095$$

- Оценка на основании интеграла Лапласа:

$$D_H^2 = 6.4578$$

$$\hat{P}(1|2) = \hat{P}(2|1) = 0.1038$$

- Оценка на основании процедуры скользящего экзамена:

$$\hat{P}(1|2) = 0.095, \hat{P}(2|1) = 0.105$$

3.2 Построение классификатора на основе плохо разделенных данных

На основе тестовых выборок ОВ1 и ОВ2 размеров n_1 и $n_2 = 200$ получаем следующую дискриминантную функцию:

$$z(\mathbf{x}) = -0.21325x_1 - 0.0817x_2 - 0.04306x_3$$

Характеристическая функция уже посчитана, найдем численные значения остальных параметров классификатора:

$$\xi_1 = \sum_{j=1}^{p} \alpha_j \mu_j^{(1)} = -0.52419 \tag{18}$$

$$\xi_2 = \sum_{j=1}^{p} \alpha_j \mu_j^{(2)} = -0.59278 \tag{19}$$

$$c = \frac{\xi_1 + \xi_2}{2} = -0.55848 \tag{20}$$

3.2.1 Применение классификатора на тестовой выборке

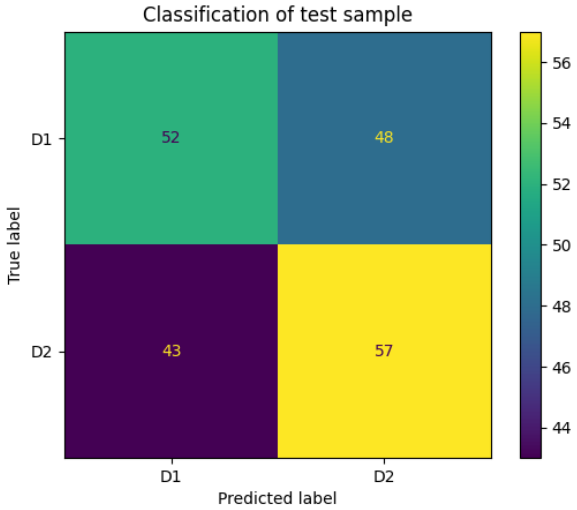

Рис. 7: Результаты классификации тестовой выборки

3.2.2 Применение классификатора на обучающей выборке

В процессе классификации элементов обучающих выборок было получено такое разделение на две группы:

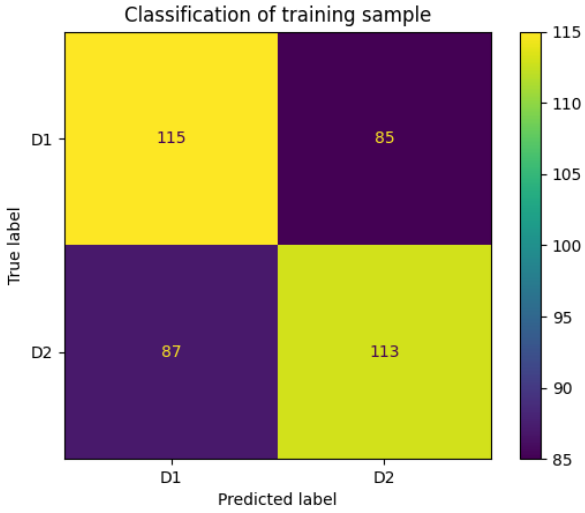

Рис. 8: Результаты классификации обучающих выборок

3.2.3 Оценки расстояния Махаланобиса и вероятностей ошибочной классификации

- Оценка на основании доли неверно классифицированных элементов тестовой выборки:

$$\hat{P}(1|2) = 0.48, \hat{P}(2|1) = 0.43$$

- Оценка на основании доли неверно классифицированных элементов обучающих выборок:

$$\hat{P}(1|2) = 0.425, \hat{P}(2|1) = 0.435$$

- Оценка на основании интеграла Лапласа:

$$D_H^2 = 0.037$$

$$\hat{P}(1|2) = \hat{P}(2|1) = 0.4617$$

- Оценка на основании процедуры скользящего экзамена:

$$\hat{P}(1|2) = 0.44, \hat{P}(2|1) = 0.455$$

4 Построение и тестирование классификатора на реальных данных

Изначально разобьем ввсе содержащиеся в фалйе данные на две обучающие выборки размеров $n_1 = 500, n_2 = 200$ элементов и две тестовые выборки $n_1 = 200, n_2 = 100$ элементов (так как изначально размеченные данные содержат 700 элементов, отсящихся к 1 классу, и 300 элементов, относящихся к 2 классу).

Так как изначально мы не знаем, какие параметры имеет распределение данных, взятых с репозитория, подсчитаем оценки математического ожидания и матрица ковариации признаков для каждой обучающей выборки. Для этого используем следующие формулы:

$$\mu^{(1)} \to \hat{\mu}^{(1)}, \mu^{(2)} \to \hat{\mu}^{(2)}$$

$$\hat{\mu}_j^{(k)} = \frac{1}{n_k}\sum_{i=1}^{n_k} x_{ij}^{(k)}, k = 1, 2$$

$$\Sigma \to S, S = \frac{1}{n_1 + n_2 - 2}[(n_1 - 1)S^{(1)} + (n_2 - 1)S^{(2)}]$$

$$S^{(k)} = (s_{ij}^k), l, j \in [1, p], k = 1, 2$$

$$s_{ij}^k = \frac{1}{n_k - 1}\sum_{i=1}^{n_k}(x_{il}^{(k)} - \hat{\mu}_l^{(k)})(x_{ij}^{(k)} - \hat{\mu}_j^{(k)}), k = 1, 2$$

Далее уже на основе данных параметров строим дискриминантную функцию по следующим формулам:

$$\text{OB1: } z_i^{(1)} = a_1 x_{i1}^{(1)} + ... + a_p x_{ip}^{(1)}, i \in [1, n_1]$$

$$\text{OB2: } z_i^{(2)} = a_1 x_{i1}^{(2)} + ... + a_p x_{ip}^{(2)}, i \in [1, n_2]$$

$$\xi_1 \to \overline{z}^{(1)} = \frac{1}{n_1}\sum_{i=1}^{n_1} z_i^{(1)}$$

$$\xi_2 \to \overline{z}^{(2)} = \frac{1}{n_2}\sum_{i=1}^{n_2} z_i^{(2)}$$

$$\sigma_z^2 \to s_z^2 = \sum_{l=1}^{p}\sum_{j=1}^{p} a_l s_{lj} a_j$$

Приводить параметры данных, значения дискриминантных функций не будем в силу того, что размерность реальных данных является достаточно высокой, и вписывать все промежуточные результаты в отчет нецелесообразно (с ними можно ознакомиться при исполнении программного кода, реализущего весь процесс исследований, описанных в лабораторной работе, ссылка на соответствующий репозиторий GitHub прикладывается ниже).

Классификация экземпляров данных производится на основании следующего правила:

$$\begin{cases} \text{D1} : \sum_{j=1}^{p} a_j x_j \geq \frac{\overline{z}^{(1)} + \overline{z}^{(2)}}{2} \\ \text{D2} : \sum_{j=1}^{p} a_j x_j < \frac{\overline{z}^{(1)} + \overline{z}^{(2)}}{2} \end{cases} \tag{21}$$

Интегральную добавку $ln\frac{q_2 c(1|2)}{q_1 c(2|1)}$ изначально полагаем нулевыми аналогично предложению в исследовании модельных данных ($q_1 = q_2, c(1|2) = c(2|1)$).

Расстояние Махалонобиса в оценках вероятности ошибочной классификации считается по аналогичному принципу, но вместо известных параметров $\mu^{(1)}$ и $\mu^{(2)}$ используются $\overline{z}^{(1)}$ и $\overline{z}^{(2)}$.

4.1 Классификация до понижения размерности задачи

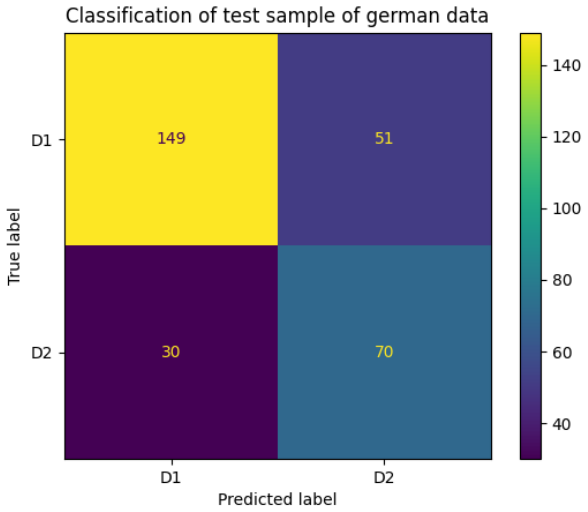

Рис. 9: Классификация тестовых выборок, взятых с репозитория

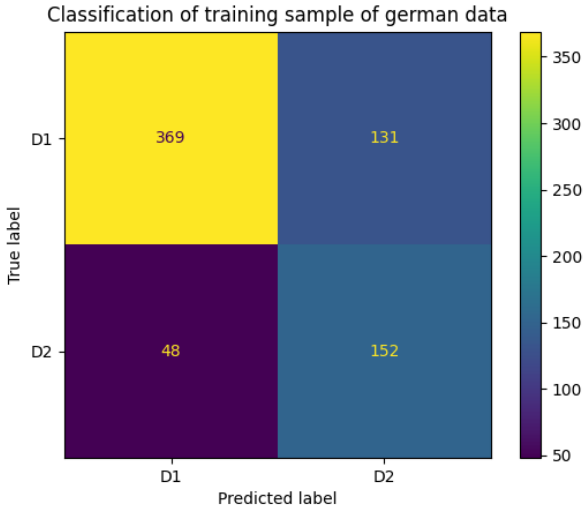

Рис. 10: Класификация обучающих выборок, взятых с репозитория

4.2 Оценки расстояния Махаланобиса и вероятностей ошибочной классификации

- Оценка на основании доли неверно классифицированных элементов тестовой выборки:

$$\hat{P}(1|2) = 0.255, \hat{P}(2|1) = 0.3$$

- Оценка на основании доли неверно классифицированных элементов обучающих выборок:

$$\hat{P}(1|2) = 0.262, \hat{P}(2|1) = 0.24$$

- Оценка на основании интеграла Лапласа:

$$D_H^2 = 1.279$$

$$\hat{P}(1|2) = \hat{P}(2|1) = 0.2858$$

- Оценка на основании процедуры скользящего экзамена:

$$\hat{P}(1|2) = 0.282, \hat{P}(2|1) = 0.29$$

4.3 Применение PCA для понижения числа размерностей

Классифицировать данные, содержащие большое количество признаков, может быть неудобно с точки зрения визуальной проверки и представления результата. Если отрисовать данные в пространстве вместе с результатами классификации, станет понятно, работает ли алгоритм классификации, и если да, то насколько хорошо. Для понижения размерности исходных данных может использоваться метод **PCA** (Principal Components Analysis).

Алгоритм работы данного метода приводится ниже:

$data \leftarrow data - \overline{data}$
$cov \leftarrow cov\_matrix(data)$
$eig\_values, eig\_vectors \leftarrow eigenvalues(cov)$
$p' \leftarrow 3$
$eig\_values \leftarrow sorted(eig\_values)$

$vectors\_subset \leftarrow sorted(eig\_vectors[:, :p'])$

$data\_reduced \leftarrow (vectors\_subset^T * data\_meaned^T)^T$

return $data\_reduced$

Применяя метод к данным *german*, получаем следующие промежуточные результаты:

```
Using PCA for german-data:
0 eigenvalue: 861.7518664079546
1 eigenvalue: 130.7901576529243
2 eigenvalue: 80.97022796550598
3 eigenvalue: 2.7236334714553116
4 eigenvalue: 1.5949036241027708
5 eigenvalue: 1.5182522132998977
6 eigenvalue: 1.083437449945327
7 eigenvalue: 1.0275076057018597
8 eigenvalue: 0.8769538655130998
9 eigenvalue: 0.516391445895743
10 eigenvalue: 0.46995522334148027
11 eigenvalue: 0.30624287962677343
12 eigenvalue: 0.25349043499752444
13 eigenvalue: 0.24150177169018938
14 eigenvalue: 0.2164611166373292
15 eigenvalue: 0.17810752021366769
16 eigenvalue: 0.12046931468810963
17 eigenvalue: 0.10209956986048
18 eigenvalue: 0.07599306982386643
19 eigenvalue: 0.04905517704697561
20 eigenvalue: 0.01569071088257549
21 eigenvalue: 0.017330779591223144
22 eigenvalue: 0.029482456266203935
23 eigenvalue: 0.03258306783020899
```

Рис. 11: Дисперсии главных компонент, полученные путем подсчета собственных чисел ковариационной матрицы

Видно, что основную долю дисперсии воспроизводят первые 3 признака с наибольшими значениями собственных чисел матрицы ковариации. Поэтому оставим только **три** обозначенных признака, так как остальные не так полезны в классификации и трехмерные данные удобно визуализировать.

4.4 Классификация после понижения числа признаков

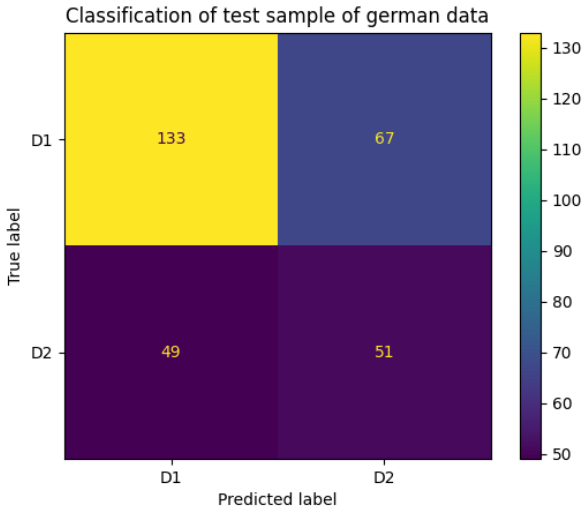

Рис. 12: Класификация тестовых данных, взятых с репозитория, после понижения числа признаков

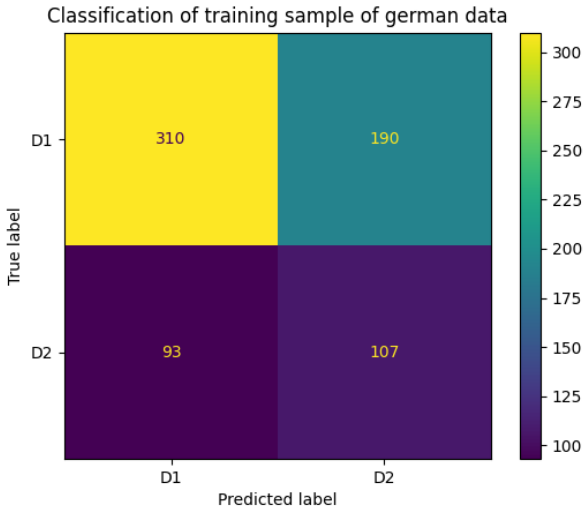

Рис. 13: Класификация обучающих выборок, взятых с репозитория, после понижения числа признаков

4.5 Оценки расстояния Махаланобиса и вероятностей ошибочной классификации после понижения размерности

- Оценка на основании доли неверно классифицированных элементов тестовой выборки:

$$\hat{P}(1|2) = 0.335, \hat{P}(2|1) = 0.49$$

- Оценка на основании доли неверно классифицированных элементов тренировочных выборок:

$$\hat{P}(1|2) = 0.38, \hat{P}(2|1) = 0.465$$

- Оценка на основании интеграла Лапласа:

$$D_H^2 = 0.1737$$

$$\hat{P}(1|2) = \hat{P}(2|1) = 0.4175$$

- Оценка на основании процедуры скользящего экзамена:

$$\hat{P}(1|2) = 0.382, \hat{P}(2|1) = 0.475$$

4.6 Сравнение классификации на других размерностях

Для проверки гипотезы о том, что 3 признаков мало для приемлемой классификации, была проведена классификация данных после понижения числа признаков до 4, 6 и 10 соотвественно. Были получены следующие результаты:

- Данные, содержащие 4 признака:

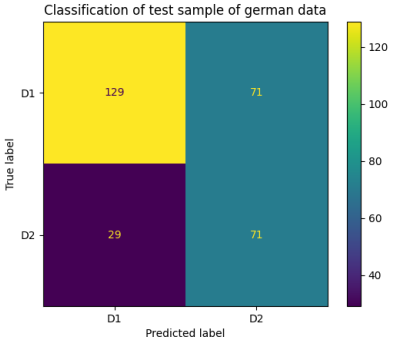

Рис. 14: Класификация тестовой выборки

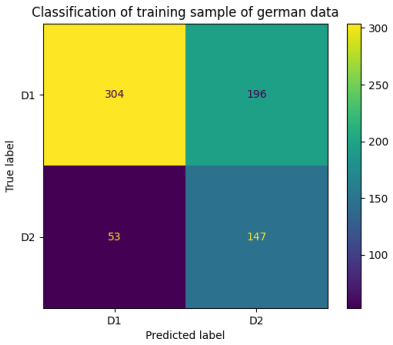

Рис. 15: Класификация обучающих выборок

- Данные, содержащие 6 признаков:

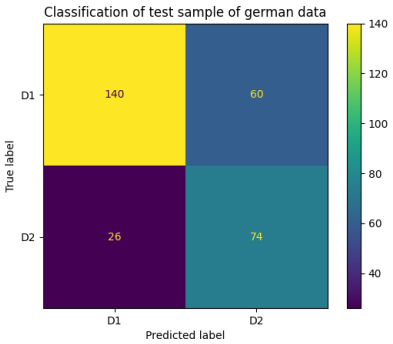

Рис. 16: Класификация тестовой выборки

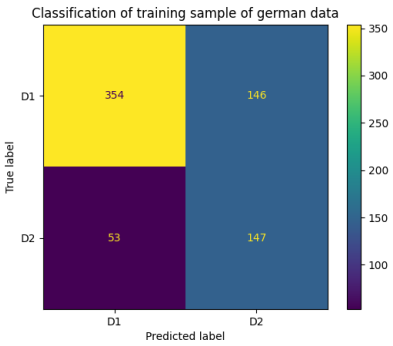

Рис. 17: Класификация обучающих выборок

- Данные, содержащие 10 признаков:

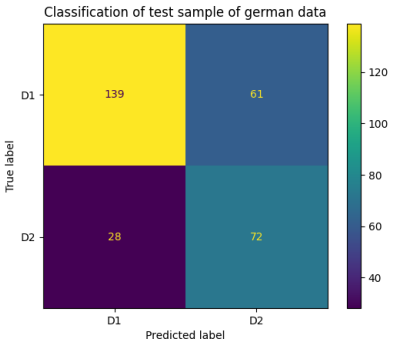

Рис. 18: Класификация тестовой выборки

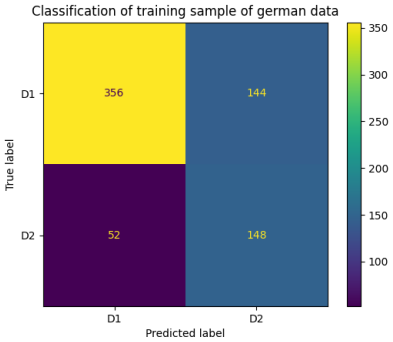

Рис. 19: Класификация обучающих выборок

4.7 Выводы по классификации данных с репозитория

Исходя из тестирования классификатора на разных размерностях, видно, что можно при помощи PCA понизить размерность данных до 4 без ощутимой потери качества, преобразования же до больших размерностей особого смысла не имеют. Кроме этого стоит отметить, что 4-ехмерные данные позволяют построить гораздо более качественный классификатор, нежели 3-ехмерные. Значит, в данной задаче лучше пренебречь визуализацией результата.

В целом, можно сказать, что PCA является достаточно полезным инструментом при анализе данных. Его стоит применять при слишком большом числе признаков у экземпляров исходных данных как минимум для снижения нагрузок на ЭВМ (работать с многомерными данными меньшей размерности менее затратно по временным и пространственным ресурсам), ведь качество классификации при этом не страдает, если параметр p' подобран разумно.

5 Модульная структура программы

Построение классификатора на основе дискриминантной функции, вычисление оценок вероятностей ошибочной классификации (как при помощи относительного числа ошибочно классифицированных объектов, так и при помощи процедуры скользящего экзамена), а также метод главных компонент и построение модельных данных производится в среде разработки PyCharm на языке программирования Python версии 3.10.1 с применением таких фреймворков, как **numpy**, **scikit-learn**, **matplotlib** .

Репозиторий с кодом данной лабораторной работы находится по данной **ссылке**.

6 Вывод по работе

Классификатор, основанный на дискриминантной функции, является достаточно эффективным инструментом бинарной классификации и в случае достаточно удачно распределенных данных может обеспечить с высокой вероятностью правдоподобное разделение данных на классы. К сожалению, задачи бинарной классификации в современном мире решаются все реже, и соответственно рассмотренный в работе метод чаще используется в виде более мощных модификаций.