МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
федеральное государственное бюджетное образовательное учреждение
высшего образования
«Тольяттинский государственный университет»

Институт Математики, физики и информационных технологий
(наименование института полностью)

Кафедра «Прикладная математика и информатика»
(наименование)

01.03.02 Прикладная математика и информатика
(код и наименование направления подготовки, специальности)

Компьютерные технологии и математическое моделирование
(направленность (профиль)/специализация)

ВЫПУСКНАЯ КВАЛИФИКАЦИОННАЯ РАБОТА (БАКАЛАВРСКАЯ РАБОТА)

на тему «Исследование методов дискриминантного анализа базы данных»

Студент Д.В. Зорин _____
(И.О. Фамилия) (личная подпись)

Руководитель М.А. Тренина
(ученая степень, звание, И.О. Фамилия)

Консультант М.В. Дайнеко
(ученая степень, звание, И.О. Фамилия)

Тольятти 2021

Аннотация

Тема выпускной квалификационной работы: «Исследование методов дискриминантного анализа базы данных».

Работа была выполнена студентом Тольяттинского Государственного Университета, института математики, физики и информационных технологий, группы ПМИп-1702а, Зориным Данилом Вадимовичем.

Выпускная квалификационная работа посвящена исследованию и реализации методов дискриминантного анализа и применению методов для анализа базы данных.

Цель работы – решение задачи классификации с применением методов дискриминантного анализа.

Объект исследования – методы дискриминантного анализа.

Предмет исследования – данные о сердечных заболеваниях.

Задачи работы:

- изучить теоретический материал о дискриминантном анализе;
- разработать алгоритм для решения задачи классификации;
- выполнить программную реализацию разработанного в предыдущей задаче алгоритма;
- провести исследование эффективности алгоритма, реализованного в предыдущей задаче.

Актуальность работы обусловлена анализом раннее не исследованных данных с применением методов дискриминантного анализа.

Результатом работы является программа, реализующая методы дискриминантного анализа и исследование базы данных о сердечных заболеваниях сердца с построением графиков точности реализуемых методов дискриминантного анализа. Бакалаврская работа содержит пояснительную записку объемом 44 страниц, включая 38 формул, 2 таблицы, список используемой литературы из 21 наименований, включая 9 зарубежных.

Abstract

The title of the graduation work is «Research of methods of discriminant analysis of the database».

This work is dedicate to the realization of research and implementation of discriminant analysis methods and application of methods for database analysis.

The aim of this work is to solving the classification problem using discriminant analysis methods.

The object of the study is methods of discriminant analysis.

The subject of the study is a dataset on heart diseases.

Relevance of this work lies in due to the analysis of previously unexplored data using discriminant analysis methods.

The graduation work is divide by several logically connected sections.

In first section to discusses the general formulation of the classification problem by discriminant analysis, and theoretical information about working with large amounts of data and from analysis. In a second section to describes the input data for the analysis and develops the software for solving the problem. In third section to analyzes the results of the algorithm. In conclusion, the results and conclusions of the work done.

The result of the graduation work is a program that implements the methods of discriminant analysis and the research of the database of heart diseases of the heart with the creation of graphs of the accuracy of the implemented methods of discriminant analysis. The graduation work consists of an explanatory note on 44 pages, including 38 formulas, 2 tables, a list of references from 21 titles, including 9 foreign sources.

Содержание

Введение

Дискриминантный анализ – это метод статистического анализа, который направлен на решение проблемы классификации объекта по заранее известным группам. Впервые решение задачи классификации этим методом продемонстрировал Рональд Фишер, который на примере данный по более чем ста ирисам трёх видов построил правила (функции) отнесения нового ириса к одной из известных групп.

Начиная с этого момента и далее дискриминантный анализ стал применяться в разный направлениях, в первую очередь в биологии, медицине, экономике, социологии и не только. Благодаря тому, что в результате анализа можно не только получить значение различий групп, но и определить те признаки, которые вносят больший вклад в определение принадлежности к группе сегодня методы дискриминантного анализа применяются даже в распознавании образов и машинном обучении, а самих методов стало больше, поэтому применение и реализация данного метода статистического анализа представляет научно–практический интерес.

В данной выпускной бакалаврской работе объектом исследования является данные о сердечных заболеваниях. Таким образом, актуальность бакалаврской работы обусловлена анализам раннее не исследованных данных с применением методов дискриминантного анализа.

Предметом исследования являются методы дискриминантного анализа.

Целью выпускной квалификационной работы является решение задачи классификации с применением методов программирования по работе с большими объемами данных.

Для достижения поставленной цели необходимо решить следующие задачи:

- изучить теоретический материал о дискриминантном анализе;
- разработать алгоритм для решения задачи классификации;

- выполнить программную реализацию разработанного в предыдущей задаче алгоритма;
- провести исследование эффективности алгоритма, реализованного в предыдущей задаче.

Выпускная квалификационная работа состоит из введения, трех разделов, заключения, списка используемых источников.

В разделе 1 рассматривается общая постановка задачи классификации методом дискриминантного анализа, и теоретическая информация о работе с большими объемами данных и из анализом. В разделе 2 описываются входные данные для анализа и разрабатывается программное обеспечение для решения задачи. В разделе 3 проводится анализ полученных результатов работы алгоритма. В заключении представлены результаты и выводы о проделанной работе.

1 Теория дискриминантного анализа и статистика

1.1 Постановка задачи

Сформулируем математически задачу классификации. Положим, что имеется объект наблюдения x, определяющийся вектором данных из p признаков:

$$x = (x_1, x_2, \ldots, x_p) \in X, \quad (1)$$

где X – некоторое множество значений.

Пусть множество X разбито на некоторое количество не пересекающихся между собой k классов W_j:

$$Y = \{1, \ldots, k\}, \quad (2)$$

$$\cup_{j=1}^{k} W_j = X, W_j \cap W_i = \emptyset \ (i, j \in Y, i \neq j). \quad (3)$$

Тогда требуется создать классификатор, способный устанавливать принадлежность $x \in X$ к одному из доступных классов.

В данной работе рассматриваются открытые базы данных в которых принадлежность x явно установлено. Поэтому построение и проверка классификатора будет происходит по следующему принципу. Исходные данные разделяются на два множества – обучающую и тестовую выборку. По первому множеству строится классифицирующая функция, на определяются классы для всех объектов во втором множестве. На основе полученных результатов происходит сравнение исходных данных с результатами классификации. На основании подобия выносится степень точности построенного классификатора.

1.2 Дискриминантный анализ

1.2.1 Основа дискриминантного анализа

Положим, что исходные данные из некоторого множества W_j нормально распределены:

$$x \in W_j \Leftrightarrow x \sim N(\mu_j, \Sigma_j), j = \overline{1,k} \quad (4)$$

Определяется класс через минимизацию величины вероятности ошибочной классификации:

$$\sum_{j=1}^{k} p_j \left(\sum_{\substack{i=1, \\ j \neq i}}^{k} P(i|x) \right), \quad (5)$$

$$P(j|x) = \frac{p_j f_j(x)}{\sum_{j=1}^{k} p_j f_i(x)} \quad (6)$$

где p_j – вероятность принадлежности объекта к множеству W_j;

$P(i|x)$ – вероятность ошибки определения элемента.

В результате, x будет относиться к классу, который имеет наибольшую апостериорную вероятность.

1.2.2 Линейный дискриминантный анализ

В основе линейного дискриминантного анализа лежит функция различия, которая строится на основе параметров – признаков, со своими коэффициентами значимости.

$$F = a_1 x_1 + a_2 x_2 + \cdots + a_i x_i, \quad (7)$$

где F – значение дискриминантной функции;

a – коэффициент вклада признака x_i.

Данная функция, представленная в канонической форме, строится для каждой группы. Основными её параметрами являются коэффициенты вклада признака. Поэтому чем больше различие средних значений соответствующего коэффициента в группах, тем более точной будет функция различия. Также, при построении других функций, их значения должны быть некоррелированными со всеми предыдущими. Количество самих функций различия зависит от числа групп и признаков. Если количество признаков меньше числа групп, то числом функций будет являться количество признаков, в противном случае, когда количество признаков класса больше числа групп, функций различия следует построить по числу групп без единицы.

Рассмотрим, как получают коэффициенты a_i функции различия. Для этого на основе статического метода измерим степень разности между объектами. То есть, нужно найти центры для каждой группы. Это делается путём матричных преобразований сумм квадратов и попарных произведений.

Если данную матрицу разделить на число признаков, то получится ковариационная матрица внутри группы.

$$T = \left(\frac{1}{n_k}\sum_{k=1}^{n}(X_{ik} - X_i)(X_{jk} - X_j)\right), \quad (8)$$

где T – ковариационная матрица;

n – количество признаков;

X_{ik} – среднее значение признака в k-том классе;

X_i – среднее значение признака во всём классе.

Для нахождения общей ковариационной матрицы, например, для двух групп, требуется рассчитать следующую матрицу $\hat{T}$, а для нахождения всех коэффициентов a_i функции распределения требуется рассчитать вектор столбец A по следующей формуле:

$$\hat{T} = \frac{(n_1 * T^1 + n_2 * T^2)}{n_1 + n_2 - 2}, A = \hat{T}^{-1}(X_1 - X_2) \quad (9)$$

Теперь для построения функции различия известно всё необходимое. Остаётся только решить задачу классификации. Это делается путём вычисления значения дискриминантных функций относительно своих центров класса и сравнения этих значения со значением функции по неизвестному объекту.

К примеру, существует два класса A и B, если среднее значение функции по классу A больше B, то неизвестный объект N принадлежит классу A при условии, что F_N больше среднего значения F по всем объектам, иначе новый объект принадлежит классу B. При условии $F_B > F_A$, N принадлежит B при F_N больше среднего значения F, в противном случае к A.

Свойства линейного дискриминантного анализа:

- Линейный дискриминантный анализ предполагает, что данные являются гауссовыми. Точнее, она предполагает, что все классы имеют одну и ту же ковариационную матрицу;
- Линейный дискриминантный анализ находит линейные границы решений в размерном подпространстве *K-1*. Как таковая, она не подходит, если существуют взаимодействия более высокого порядка между независимыми переменными;
- Линейный дискриминантный анализ хорошо подходит для много классовых задач, но ее следует использовать с осторожностью, когда распределение классов не сбалансировано, так как приоры оцениваются по наблюдаемым счетам. Таким образом, наблюдения редко будут классифицироваться на нечастые классы.

1.2.3 Квадратичный дискриминантный анализ

Квадратичный дискриминантный анализ тесно связана с линейным дискриминантным анализом, где предполагается, что измерения нормально

распределены, а ковариационная матрица оценивается отдельно для каждого класса так как, в отличие от линейного дискриминантного анализа, здесь нет предположения, что ковариация каждого из классов идентична.

Недостатком квадратичного дискриминантного анализа является то, что его нельзя использовать в качестве метода уменьшения размерности. Для оценки параметров, требуемых при квадратичной дискриминации, требуется больше вычислений и данных, чем в случае линейной дискриминации. Если нет большой разницы в групповых ковариационных матрицах, то последние будут выполнять так же хорошо, как и квадратичная дискриминация. Квадратичная дискриминация является общей формой Байесовской дискриминации.

$$\widehat{\sum}_k(\lambda) = (1-\lambda)\widehat{\sum}_k + \lambda\,\widehat{\sum}_k \qquad (10)$$

В квадратичном дискриминантном анализе для каждого k-го класса, где их количество в диапазоне от 1 до K необходимо оценить $\widehat{\sum}_k$, а не предполагать, что $\widehat{\sum}_k = \sum$. Дискриминантная функция является квадратичной в x:

$$\delta_k(x) = -\frac{1}{2}\log|\Sigma_\mathrm{k}| - \frac{1}{2}(x-\mu_k)^T\Sigma_\mathrm{k}^{-1}(x-\mu_k) + \log\pi_k \qquad (11)$$

Поскольку данный метод оценивает ковариационную матрицу для каждого класса, она имеет большее количество эффективных параметров, чем в линейном анализе.

Из ковариационной матрицы Σ_k нужно рассматривать только диагональ и верхний правый треугольник или же левый нижний. Эта область ковариационной матрицы имеет $\frac{p(p+1)}{2}$ элемента. Так как необходимо

оценивать K таких матриц, то к ковариационным матрицам относятся параметры $K \cdot \frac{p(p+1)}{2}$.

Таким образом, эффективное число параметров в квадратичном дискриминантном анализ равно:

$$q = \text{К} - 1 + \text{Кр} + \text{Кр}\frac{(\text{р}+1)}{2} \quad (12)$$

Поскольку количество параметров квадратично p, то следует использовать с осторожностью, когда пространство признаков большое объектов класса.

Иногда свойства линейного дискриминантного анализа являются ограничением для классификации объекта. Поэтому линейный дискриминантный анализ следует изменить под задачу. Рассмотрим его вариации.

1.2.4 Смешанный дискриминантный анализ

Этот вид анализа использует алгоритм максимизации ожиданий для расчета оценок максимальной вероятности. Алгоритм работает в четыре этапа.

На первом этапе производится первоначальное определение параметров (инициализация). Для текущих целей инициализация осуществляется с помощью кластера k-средних, где k равно числу подгрупп. Кластер k-средних выполняется для каждой группы на основе нескольких случайных стартов для идентификации кластеров подгрупп. Параметры для первоначальной инициализации берутся из анализа кластеров k-средних каждой подгруппы.

Следующим шагом смешанный дискриминантный анализ вычисляет вес от каждой подгруппы. Далее пересчитываются средние значения и ковариации для определения оценок максимального правдоподобия для каждой из подгрупп. И наконец смешанный дискриминантный анализ

итеративно чередует эти шаги до тех пор, пока каждый кластер подгрупп не максимизирует правильное назначение сходимости.

Иначе говоря, кластеры подгрупп итеративно пересчитываются для максимизации точного группового назначения, а апостериорные вероятности вычисляются из смеси конечных апостериорных вероятностей подгрупп. Учитывая использование нескольких центров из предполагаемых подгрупп, можно рассчитать нелинейные границы принятия решений, что позволит моделировать ненормальные многомерные распределения.

1.2.5 Гибкий дискриминантный анализ

Гибкий дискриминантный анализ – метод, использующий линейный дискриминант. анализ, разработанный Фишером (1936) в качестве основы.

В линейном дискриминантном анализе можно выбрать несколько оптимальных переменных из всего их количества, чтобы определить и разделить группы, доступные в переменной ответа. Линейный дискриминантный анализ выполняет эту задачу путем множественных линейных регрессий; но, если нелинейных уравнений больше, то линейный дискриминантный анализ столкнется с проблемой. Поэтому Хасти и другие в 1994 году использовали гибкий ряд уравнений, а не линейные уравнения, чтобы соответствовать значениям.

Используя эту гибкость, линейные уравнения являются не единственными уравнениями, которые подгоняются и тестируются на данных, а также используются другие типы уравнений (квадратичные, кубические, логарифмические и т.д.), и, наконец, выбирается наилучшее уравнение.

1.2.6 Регуляризированный дискриминантный анализ

Регуляризированный дискриминантный анализ использует ту же общую схему, что и линейный и квадратичный дискриминантный анализ, но оценивает ковариацию новым способом, который объединяет ковариацию

квадратичного дискриминантного анализа $\widehat{\sum}_k$ с ковариацией в линейном дискриминантном анализе $\widehat{\sum}$ с помощью параметра настройки λ.

$$\widehat{\sum}_k(\lambda) = (1-\lambda)\widehat{\sum}_k + \lambda\,\widehat{\sum}_k \qquad (13)$$

Для внесения дополнительной настройки γ в модификацию ковариационной матрицы эту функцию можно переписать к виду:

$$\widehat{\sum}_k(\lambda,\gamma) = (1-\gamma)\widehat{\sum}_k(\lambda) + \gamma\frac{1}{p}\mathrm{tr}\left(\widehat{\sum}_k\right), \qquad (14)$$

где γ и λ – параметры смешивания, принимающие значения от 0 до 1.

Для четырех крайних значений γ и λ, ковариационная структура сводится к частным случаям:

- ($\gamma = 0$, $\lambda = 0$): Квадратичный дискриминантный анализ – индивидуальная ковариация для каждой группы;
- ($\gamma = 0$, $\lambda = 1$): Линейный дискриминантный анализ – общая ковариационная матрица;
- ($\gamma = 1$, $\lambda = 0$): Условные независимые переменные – похожи на наивный байесовский метод, но вариации переменных внутри группы (главные диагональные элементы) равны;
- ($\gamma = 1$, $\lambda = 1$): Классификация с использованием евклидова расстояния, но дисперсия одинакова для всех групп. Объекты попадают в группу с ближайшим средним значением.

1.3 Проверка нормальности распределения выполнения условия Критерий Шапиро–Уилкса

Пусть имеется одномерная выборка данных, тогда, пользуясь критерием Шапиро–Уилкса, эти данные можно проверить и определить

получены ли они из нормальной генеральной совокупности или нет. Изначально выносится гипотеза о нормальности распределения генеральной совокупности и рассматривается статистика W:

$$W = \frac{\gamma^2}{s_2}, \quad (15)$$

$$\gamma = \sum_{i=1}^{n} a_{n-i+1} * (x_{n-i+1} - x_i), \quad (16)$$

$$s^2 = \sum_{i=1}^{n} (x_i - \bar{x})^2, \bar{x} = \frac{1}{\mathrm{n}} \sum_{i=1}^{n} x_i \quad (17)$$

Пользуясь таблицей, расположенной в [8] можно определить значение всех коэффициентов a_{n-i+1}. После этого вычисляется W_α – пороговое критическое значение. Если $W < W_\alpha$ тогда условие о нормальном распределении гипотезы не выполняется, а значит данные распределены не нормально[8].

1.4 Отбор значащих признаков

1.4.1 Важность отбора

В сложных задачах данные, подлежащие анализу, находятся в необработанном виде. Возможны случаи, когда имеются пропущенные значения в данных для некоторых объектов или рассматривается очень большое количество признаков, что очень сильно влияет на результат исследования.

Если количество признаков равно или даже превышает количество наблюдений, то при оценке ковариационной матрицы оцениваемых параметров будет больше, чем наблюдений, а значит, ковариационная матрица будет плохо обусловлена.

Помимо такого случая, возможен вариант, когда наоборот число наблюдений больше числа признаков. Ковариационная матрица также может быть плохо обусловленной, что чревато неадекватным правилом

классификации, так как обратная ковариационная матрица используется в формулах дискриминантных функций.

Недостаточная обусловленность может быть вызвана не только числом параметров измерения, но и тем, что некоторые параметры зависимы между собой и достаточно близки.

По выше изложенным причинам до начала этапа анализа данных следует все параметры вносящие небольшую точность измерения удалить, которые являются избыточными для дискриминантного анализа. Для этого отбираются те параметры, чей вклад в оценку определения класса объекта довольно высок. Если все оставшиеся параметры одинаково значимы, то необходимо произвести дополнительное измерение влияния на составление функции классификации, а все прочие (наименее значимые) отбросить.

Фактически, такое действие приводит к увеличению точности определения класса. Помимо этого, увеличивается устойчивость к отклонениям за счёт уменьшения размерности пространства признаков.

Количество методов выбора значимых переменных велико. В данной работе будет использоваться простая фильтрация всех параметров исходных данных по результатам вычисления лямбды Уилкса для каждого параметра базы данных.

1.4.2 Лямбда Уилкса

Лямбда Уилкса – это тестовая статистика, используемая в MANOVA для проверки наличия различий между средними значениями идентифицированных групп по комбинации зависимых переменных. Лямбда Уилкса является прямой мерой доли дисперсии в комбинации зависимых переменных, которая не учитывается независимой переменной. Если большая часть дисперсии учитывается независимой переменной, то это говорит о том, что существует эффект от группирующей переменной.

Статистику лямбда Уилкса можно математически скорректировать в статистику, которая имеет приблизительно F распределение. Это облегчает вычислить P значение. Часто авторы представляют F значение и степени

свободы, как в приведенной выше работе, вместо того, чтобы дать фактическое значение статистики лямбды Уилкса.

Положим, данные описываются некоторым вектором набора критериев (параметров), $t = (t_1, t_2, \ldots, t_r)$. Тогда разобьём все данные по соответствующим классам:

$$(t_{ij});\ i = 1, \ldots, n_j\ ;\ j = 1, \ldots, k;\ \sum_{j=1}^{k} nj = n, \tag{18}$$

Лямбда Уилкса применяется, когда необходимо проверить гипотезу о равенстве средних $H_0 : \mu_1 = \mu_2 = \cdots = \mu_k$, что определяется следующей формулой:

$$\Lambda(t) = \frac{|W_{tt}|}{|W_{tt} + B_{tt}|}, \tag{19}$$

$$B = \sum_{j=1}^{k} n_j \left(\overline{t_J} - \bar{t}\right)^T \left(\overline{t_J} - \bar{t}\right), \tag{20}$$

$$W = \sum_{j=1}^{k} \sum_{j=1}^{n_j} \left(\overline{t_J} - \bar{t}\right)^T \left(\overline{t_J} - \bar{t}\right), \tag{21}$$

$$\overline{t_J} = \frac{1}{n_j} \sum_{j=1}^{n_j} t_{ij}, \bar{t} = \frac{1}{n} \sum_{i=1}^{n} \sum_{j=1}^{k} t_{ij} \tag{22}$$

где B и W – групповые матрицы.

Данная гипотеза выполняется, если полученное значение Λ меньше какого-то установленного порога. Далее происходит проверка на добавочную информацию. Суть данной проверки заключается в измерении величины значимости вклада дополнительных переменных в измерении гипотезы:

$$H_0 : \mu_1 = \mu_2 = \cdots = \mu_k \tag{23}$$

Эти добавочные переменные могут повлиять на значение Λ и, если это значение изменилось не больше установленного ограничения, то добавочную переменную можно удалить из рассмотрения, так как её вклад будет незначительным, а значит она не имеет большого влияния на разграничение

классов. Пусть имеется k групп, которые определяются набором наблюдений, соответствующего класса:

$$\left(t_{ij}, x_{ij}\right);\ i\ =\ 1,\ldots,n_j\ ;\ j\ =\ 1,\ldots,k;\ \sum_{j=1}^{k} nj\ =\ n \tag{24}$$

Исходя из этого, составим матрицы W и B, как показано далее.

$$W\ = \begin{pmatrix} W_{tt} & W_{tx} \\ W_{xt} & W_{xx} \end{pmatrix}, B = \begin{pmatrix} B_{tt} & B_{tx} \\ B_{xt} & B_{xx} \end{pmatrix} \tag{25}$$

Следующим шагом вычисляются статистики Уилкса до и после добавления набора признаков x и рассматривается статистика:

$$\Lambda(t,x) = \frac{|W|}{|W+B|}, \Lambda(t) = \frac{|Wtt|}{|Wtt+Btt|} \tag{26}$$

$$\Lambda(x|t) = \frac{\Lambda(t,x)}{\Lambda(t)} \tag{27}$$

Если полученное значение меньше определённого порога, то выдвинутая гипотеза H_0 опровергается и получается, что, чем меньше $\Lambda(t,x)$ по отношению к $\Lambda(t)$, то тем будет меньше $\Lambda(x|t)$, и в итоге это приводит к тому, что x содержит в себе большее количество дополнительной информации.

1.4.3 Пошаговый выбор

Пусть x – произвольный вектор наблюдение размером p тогда алгоритм пошагового выбора выглядит следующим образом:

Обнуляется число признаков, устанавливается пороговое значение Λ^0. Вычисляется $\Lambda(x_j)$ для каждого x_j, где $j \in [1,p]$, после чего в модель включается та переменная, чьё значение статистики минимальное. Далее тот признак добавленный в модель, обозначается за t_1. Среди остальных p - 1 переменных ищется признак с наименьшей лямбдой Уилкса:

$$\Lambda(x_j \mid t_1) = \frac{\Lambda(t_{(1)}, x_{(j)})}{t_j}, j = \overline{1, p-1} \quad (28)$$

$$\Lambda \leq \Lambda^0 \quad (29)$$

Дальнейшие действия повторяются для всех последующих переменных, пока все эти переменные не будет иметь свою статистику.

В результате проделанной операции получается набор из переменных, демонстрирующий лучший результат при анализе, чем изначальный набор признаков.

1.5 Методы определения величины ошибки

1.5.1 Ошибка обученной модели

На практике абсолютно точной модели классификатора достичь не получится. Поэтому важнейшим этапом в исследовании является измерение погрешности в оценке классификатора, созданного на основе обучающей выборки. Чтобы как-то измерять погрешность вводится такое понятие, как величина ошибки, которое определяется, как математическое ожидание неправильной классификации случайного объекта x:

$$Err = E\left(I\left(y \neq m(x)\right)\right), \quad (30)$$

где y – номер класса;

$m(x)$ – построенная модель;

I – логическая переменная, принимающая значения 1 и 0.

Оценка полученной величины позволяет определить степень точности классификатора. Помимо этого, можно проанализировать и сравнить несколько таких моделей. В некоторых ситуациях число записей в исходной

выборке довольно большое и этим можно воспользоваться. Исходную выборку можно поделить на три части, она из которых будет самой большой и выполнять роль обучения классификатора. Вторая и третья часть будут выступать в роли контрольной и тестовой выборки.

Такое разделение исходных данных позволяет производить измерение точности ошибки полученной модели более точно, так как при анализе контрольной выборки выбирается наилучшая модель и на финальную стадию попадает более точный классификатор, который в свою очередь на выходе получается ещё более точным за счёт окончательной оценки величины ошибки выбранного классификатора происходящем на последнем этапе.

На ограниченных выборках применяются другие методы оценки величины ошибки, которые используют в своих методах величину ошибки на обучении:

$$err = \frac{1}{n}\sum_{i=1}^{n} I\left(y_i \neq m(xi)\right) \quad (31)$$

Это необходимо применять из-за недостаточной точности вычисления ошибки при отсутствии достаточного количества исследуемых данных.

1.5.2 Bootstrap 0.632

Данный метод оценки ошибки основан на том факте, что вероятность конкретного наблюдения из исследуемой выборки равняется $\frac{1}{n}$ при условии, однократного выбора и $1 - \frac{1}{n}$ в противном случае.

Следовательно, наблюдаемое событие, не принадлежащее bootstrap выборки, имеет следующую вероятность:

$$(1 - 1n)n \approx e - 1 = 0.368 \quad (32)$$

Оценка величины ошибки выглядит следующим образом:

$$Err\ (0.632) = \ 0.368 * err + 0.632 * Err(LOOB), \quad (33)$$

$$Err(LOOB) = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{|C-i|}\sum_{b\in C^{-i}} I\left(y_i \neq \ m_b(x_i)\right), \quad (34)$$

где C^{-i}– набор индексов, определяющие bootstrap выборки,

$|C^{-i}|$ – количество таких выборок,

m_b – классификатор по выборке b.

Исходя из этого, в bootstrap выборке будет находиться $0.632 \cdot n$ наблюдений.

1.5.3 Bootstrap 0.632+

Данная модификация bootstrap, учитывает величину относительной частоты переобучения:

$$R \ = \frac{Err(LOOB) - err}{\gamma - err}, \quad (35)$$

$$\gamma \ = \frac{1}{n^2}\sum_{i=1}^{n}\sum_{j=1}^{n} I\left(y_i \ \neq \ m\left(x_j\right)\right), \quad (36)$$

где γ – оценка полученная классификацией.

Таким образом, оценка bootstrap 0.632+ определяется, как:

$$Err\ (0.632 +) = \ (1 \ - \ \alpha) * \ err \ + \ \alpha * Err(LOOB), \quad (37)$$

$$\alpha \ = \frac{0.632}{1 - 0.368R} \quad (38)$$

Как видно, относительная частота обучения лежит в интервале [0,1], где левое значение говорит о том, что эффекта переобучения нет.

1.6 Вывод по первому разделу

В данном разделе были сформулирована задача дискриминантного анализа и рассмотрены несколько методов дискриминантного анализа.

Дополнительно к этому были рассмотрены некоторые уникальные модификации, которые используются в узконаправленных задачах.

За довольно небольшой период времени линейный дискриминантный анализ многократно видоизменялся, порождая другие методы классификации. Это обусловлено тем, что сам линейный метод дискриминантного анализа имеет ограничения как по анализируемым данным, так и по точности, но тем не менее, основа всех методов остаётся одинаковой.

Помимо этого, существуют дополнительные методы, направленные на предварительный анализ и оптимизацию исходных данных. В данном разделе был рассмотрен один из них, а именно – пошаговый отбор – стратегия, позволяющая отделить значащие признаки от менее значимых.

Так же была рассмотрено пара методов определения величины ошибки обучения. Это – две версии bootstrap (0.632, 0.632+) имеющие как низкую дисперсию, так и низкое определение величины ошибки. Но, даже беря во внимание это, bootstrap методы до сих пор не теряют своей актуальности, и используются на маленьких выборок достаточно широко.

2 Программная реализация алгоритмов дискриминантного анализа

2.1 Обзор исходных данных анализа

2.1.1 Ирисы Фишера

Данный набор данных о цветках ириса – это определенный набор информации, собранный биологом Рональдом Фишером в 1930-х годах. В собранных данных описываются конкретные биологические характеристики различных типов цветков ириса, а именно – длина и ширина как педалей, так и чашелистиков, которые являются важнейшей частью половой системы цветка. Полученный набор был собран Фишером с целью количественного определения морфологического измерения трёх родственных наборов ирисов.

Набор данных содержит 150 записей по трём класса по 50 экземпляров в каждом, где каждый класс относится к одному из трёх видов ириса. Один класс линейно отделим от двух других, а оставшиеся не являются линейно отделимыми друг от друга.

База данных содержит следующие атрибуты:

- длина чашелистика,
- ширина чашелистика,
- длина лепестка,
- ширина лепестка,
- класс ириса: сетоза, версиколор, виргинский.

2.1.2 Набор данных по сердечным заболеваниям

Данный каталог содержит 4 базы данных, касающиеся диагностики сердечных заболеваний, где все атрибуты имеют числовое значение. Данные были собраны из четырех следующих мест:

- Фонда Кливлендской клиники;
- Венгерского институт кардиологии, Будапешт
- Медицинского центр В.А., Лонг-Бич, Калифорния;

– Университетской больницы, Цюрих, Швейцария.

Каждая база данных имеет одинаковый формат экземпляра и имеют 76 необработанных атрибутов. Ниже приведён перечень этих атрибутов.

Таблица 1 – Атрибут и его краткое описание

Название	Описание	Название	Описание
1 id	patient identification number	39 xhypo	Yes/ no
2 ccf	social security number	40 oldpeak	ST depression induced by exercise relative to rest
3 age	age in years	41 slope	the slope of the peak exercise ST segment
4 sex	Male/ female	42 rldv5	height at rest
5 painloc	chest pain location	43 rldv5e	height at peak exercise
6 painexer	provoked by exertion or otherwise	44 ca	number of major vessels (0–3) colored by flourosopy
7 relrest	(1 = relieved after rest; 0 = otherwise)	45 restckm	irrelevant
8 pncaden	sum of 5, 6, and 7)	46 exerckm	irrelevant
9 cp	chest pain type	47 restef	rest raidonuclid ejection fraction
10 trestbps	resting blood pressure	48 restwm	rest wall motion abnormality
11 htn	–	49 exeref	exercise radinalid ejection fraction
12 chol	serum cholestoral in mg/dl	50 exerwm	exercise wall motion
13 smoke	Yes/no	51 thal	Normal/fixed defect/reversable defect
14 cigs	cigarettes per day	52 thalsev	–
15 years	number of years as a smoker	53 thalpul	–
16 fbs	fasting blood sugar > 120 mg/dl true/false	54 earlobe	–
17 dm	history of diabetes or no such history	55 cmo	month of cardiac cath
18 famhist	family history of coronary artery disease (yes/no)	56 cday	day of cardiac cath
19 restecg	resting electrocardiographic results	57 cyr	year of cardiac cath
20 ekgmo	month of exercise ECG reading	58 num	diagnosis of heart disease (angiographic disease status)
21 ekgday	day of exercise ECG reading	59 lmt	–
22 ekgyr	year of exercise ECG reading	60 ladprox	–
23 dig	digitalis used furing exercise ECG yes/ no	61 laddist	–
24 prop	(Beta blocker used during exercise ECG	62 diag	–
25 nitr	nitrates used during exercise ECG	63 cxmain	–

Продолжение таблицы 1

Название	Описание	Название	Описание
26 pro	calcium channel blocker used during exercise ECG	64 ramus	–
27 diuretic	diuretic used used during exercise ECG	65 om1	–
28 proto	exercise protocol	66 om2	–
29 thaldur	duration of exercise test in minutes	67 rcaprox	–
30 thaltime	time when ST measure depression was noted	68 rcadist	–
31 met	mets achieved	69 lvx1	–
32 thalach	maximum heart rate achieved	70 lvx2	–
33 thalrest	resting heart rate	71 lvx3	–
34 tpeakbps	first peak exercise blood pressure	72 lvx4	–
35 tpeakbpd	second peak exercise blood pressure	73 lvf	–
36 dummy	–	74 cathef	–
37 trestbpd	resting blood pressure	75 junk	–
38 exang	exercise induced angina	76 name	last name of patient

Но на практике используются лишь 14 атрибутов под номерами: 3, 4, 9, 10, 12, 16, 19, 32, 38, 40, 41, 44, 51, 58.

Помимо этого, авторы баз данных попросили, чтобы в любых публикациях, полученных в результате использования данных, были указаны имена главных исследователей, ответственных за сбор данных в каждом учреждении. К ним относятся:

1. Андраш Яноши, доктор медицинских наук, Венгерский институт кардиологии. Будапешт.
2. Уильям Штайнбрунн, доктор медицинских наук, Университетская клиника, Цюрих, Швейцария.
3. Маттиас Пфистерер, доктор медицинских наук, Университетская клиника, Базель, Швейцария.
4. Роберт Детрано, доктор медицины, доктор философии, Медицинский центр В.А., Лонг-Бич и Фонд Кливленд Клиник.

2.2 Разработка приложения

2.2.1 Выбор технологий

Перейдём к программной реализации одного из методов дискриминантного анализа. Программную часть реализуем на языке Python 3, так как этот язык чаще чем любой другой язык используют для обработки больших данных, сложных математических вычислений, создания нейронных сетей и не только. Так же его отличает понятный синтаксис и большое количество готовых библиотек.

Так, например, язык Python имеет библиотеку NumPy, которую можно рассматривать как альтернативу пакету MATLAB, способную выполнять математические операции над матрицами и массивами с быстрой скоростью. Помимо этого, Python имеет библиотека Pandas, которая используется для обработки и анализа больших данных. Также существует библиотека sklearn, в которой помимо достаточно много реализаций нейронных сетей и функций для предварительной обработки данных.

В итоге линейный и квадратичный методы дискриминантного анализа базы данных будут реализовываться на языке Python 3 с применением стандартных библиотеки обработки и визуализации информации NumPy, Pandas, Sklearn. Сама разработка будет производится в online среде Google Colab – размещенной на хосте службе Jupyter для ноутбуков, которая не требует настройки для использования с предоставлением бесплатного доступа к вычислительным ресурсам, включая графические процессоры.

2.2.2 Разработка

Все библиотеки загружаются один раз и далее используются во всей программе. Эта процедура изображена на рисунке ниже.

```
import numpy as np
import pandas as pd
from matplotlib import pyplot as plt
import matplotlib.colors as colors
from mpl_toolkits.mplot3d import Axes3D
from mpl_toolkits import mplot3d
from sklearn import linear_model, datasets
import seaborn as sns
import itertools
from pandas import DataFrame
from sklearn.model_selection import train_test_split
```

Рисунок 1 – Импортирование библиотек

Исходя из определений, линейного и квадратичного дискриминантного анализа, создадим функции классификации и алгоритмы определения класса, показанные на рисунках 2 и 3:

```
def LDA_score(X,MU_k,SIGMA,pi_k):
    return (np.log(pi_k) - 1/2 * (MU_k).T @
            np.linalg.inv(SIGMA)@(MU_k) +
            X.T @ np.linalg.inv(SIGMA)@ (MU_k)).flatten()[0]

def predict_LDA_class(X,MU_list,SIGMA,pi_list):
    scores_list = []
    classes = len(MU_list)

    for p in range(classes):
        score = LDA_score(X.reshape(-1,1),MU_list[p]
                          .reshape(-1,1),SIGMA,pi_list[0])
        scores_list.append(score)

    return np.argmax(scores_list)
```

Рисунок 2 – Функция классификации линейным методом

```
def QDA_score(X,MU_k,SIGMA,pi_k):
    SIGMA_inv = np.linalg.inv(SIGMA)
    return (np.log(pi_k) - 1/2 *
            np.log(np.linalg.det(SIGMA_inv)) - 1/2 *
            (X - MU_k).T @ SIGMA_inv @ (X - MU_k)).flatten()[0]

def predict_QDA_class(X,MU_list,SIGMA_list,pi_list):
    scores_list = []
    classes = len(MU_list)

    for p in range(classes):
        score = QDA_score(X.reshape(-1,1),MU_list[p]
                          .reshape(-1,1),SIGMA_list[p],pi_list[p])
        scores_list.append(score)

    return np.argmax(scores_list)
```

Рисунок 3 – Функция классификации квадратичным методом

Лямбда Уилкса вычисляется для каждого параметра выборки следующим образом, как показано на рисунке 4.

```
# shapiro wilk
def lambdaSW(data, size_l, size_r):
  s = []
  for i in data.iloc[:,size_l:size_r]:
    s.append(round(shapiro(data[i])[0],5))
  return s
```

Рисунок 4 – Функция вычисления лямбды Уилкса

Определение точности алгоритмом bootstrap происходит в готовых библиотечных функциях. По проверяемым данным вычисляется точность по двум версиям алгоритма. Выходные данные представлены в виде массива.

```
def Accuracy(X, y):
  tree = DecisionTreeClassifier(random_state=0)
  scores_1 = bootstrap_point632_score(tree, X, y, method='.632')
  scores_2 = bootstrap_point632_score(tree, X, y, method='.632+')
  acc1 = np.mean(scores_1)
  acc2 = np.mean(scores_2)
  return [round(acc1, 4), round(acc2, 4)]
```

Рисунок 5 – Функция вычисления точности методами bootstrap

Основа для анализа данных готова. Следующие шаги по инициализации, предобработки, анализа и вывода полученного результата будут завесить от различных факторов. Так структура и метод хранения данных будут влиять на первый этап анализа – инициализация, приведения входящих данных к нужному формату – на второй этап. При анализе больших данных появляются дополнительные условия, которые должны быть отражены в обработке данных, поэтому третий шаг – анализ данных, будет тем больше и сложнее, чем сложнее решаемая задача, что зависит как от точности входных данных, так и от числа параметров исследования. Вывод полученного результата происходит в соответствии с предыдущем шагом и на последнем этапе, должны быть отражены все действия работы программы.

Так как программа зависит от решаемой задачи, покажем реализацию логики описанных выше четырёх этапов (инициализация, предобработка, анализ, вывод полученного результата) на примере анализа данных трёх видов ириса.

Ирисы Фишера – набор базовых данных для анализа. Поэтому инициализация таких данных не вызывает сложностей, так как такие «базовые» данные встроены в функции многих библиотек. На рисунке 6 показано чтение данных «Ирисы Фишера»

```
iris = sns.load_dataset("iris")
iris = iris.rename(columns={'species': 'category_name'})
iris['category_int'] = iris.category_name.astype('category').cat.codes
```

Рисунок 6 – Чтение исходных данных

Потребуется определить дополнительные переменные и разделить входные данные на таблицу признаков и данные о принадлежности к классу, разделение данных на тестовую и обучающую выборку, что показано на рисунке 7.

```
iris = sns.load_dataset("iris")
iris = iris.rename(columns={'species': 'category_name'})
iris['category_int'] = iris.category_name.astype('category').cat.codes
iris_train, iris_test = train_test_split(iris, test_size=0.2, random_state=42)
class_count = iris_train['category_name'].nunique()
class_unique_count = iris_train['category_name'].value_counts().values
class_all_count = iris_train['category_name'].count()
class_name = iris_train['category_name'].unique()
X_data = iris_test.iloc[:,0:-2]
y_labels = iris_test['category_int'].copy()
y_pred = list()

colors=["#000099","#009900", "#449999", "#994444"]
```

Рисунок 7 – Определение вспомогательных переменных

Для построения линейного и квадратичного классификатора потребуется определить пару переменных в соответствии с установленными входными параметрами и на основе этого произвести классификацию, как на рисунке 8 и 9.

```
species_g = iris_train.iloc[:,0:-1].groupby('category_name')
  .mean().values
split_s_g = [(i+1) for i in range(class_count-1)]
mu_list = np.split(species_g, split_s_g)
sigma = iris_train.iloc[:,0:-2].cov().values
pi_list = class_unique_count / class_all_count
```

```
#Classify and compute accuracy accuracy
y_pred.append(np.array( [predict_LDA_class( np.array(row)
  .reshape(-1,1), mu_list, sigma, pi_list)
  for row in X_data.to_numpy() ] ))
```

Рисунок 8 – Классификация линейным дискриминантным анализом

```
sigma_g = iris_train.iloc[:,0:-1].groupby('category_name').cov()
sigma_list = np.split(sigma_g.values,[4,8], axis = 0)
```

```
y_pred.append(np.array(  [predict_QDA_class( row.reshape(-1,1),
                                             mu_list, sigma_list, pi_list)
  for row in X_data.to_numpy() ] ))
```

Рисунок 9 – Классификация квадратичным дискриминантным анализом

На рисунках 10 происходит операция генерации графиков и вычисление результата точности проверки алгоритмами bootstrap. На рисунках 11 и 12 показан консольный вывод после выполнения этого кода.

```
classes = np.append(class_name,["error"], axis=0)
fig = plt.figure(figsize=(9, 18))
current_axis = fig.add_subplot(3, 1, 1)
current_axis.set_title('LDA')
X_show = np.array(X_data.values)
X_show = np.append(X_show, np.array(y_pred[0]).reshape(-1, 1), axis = 1)
show_df = DataFrame(X_show)
show_df.columns = ['0','1','2','3', 'class']
show_df['class'] = show_df['class'].apply(lambda x: classes[int(x)])
pd.plotting.andrews_curves(show_df, 'class', ax=current_axis, color=colors)
al = Accuracy(X_show[:,0:-1], X_show[:,-1])
current_axis = fig.add_subplot(3, 1, 2)
current_axis.set_title('QDA')
X_show = np.array(X_data.values)
X_show = np.append(X_show, np.array(y_pred[1]).reshape(-1, 1), axis = 1)
show_df = DataFrame(X_show)
show_df.columns = ['0','1','2','3', 'class']
show_df['class'] = show_df['class'].apply(lambda x: classes[int(x)])
pd.plotting.andrews_curves(show_df, 'class', ax=current_axis, color=colors)
aq = Accuracy(X_show[:,0:-1], X_show[:,-1])
current_axis = fig.add_subplot(3, 1, 3)
current_axis.set_title('Target')
pd.plotting.andrews_curves(iris_test.iloc[:,0:-1], 'category_name',
                           ax=current_axis, color=colors)
plt.show()
print(al, aq)
```

Рисунок 10 – Построение графиков и вычисление ошибки

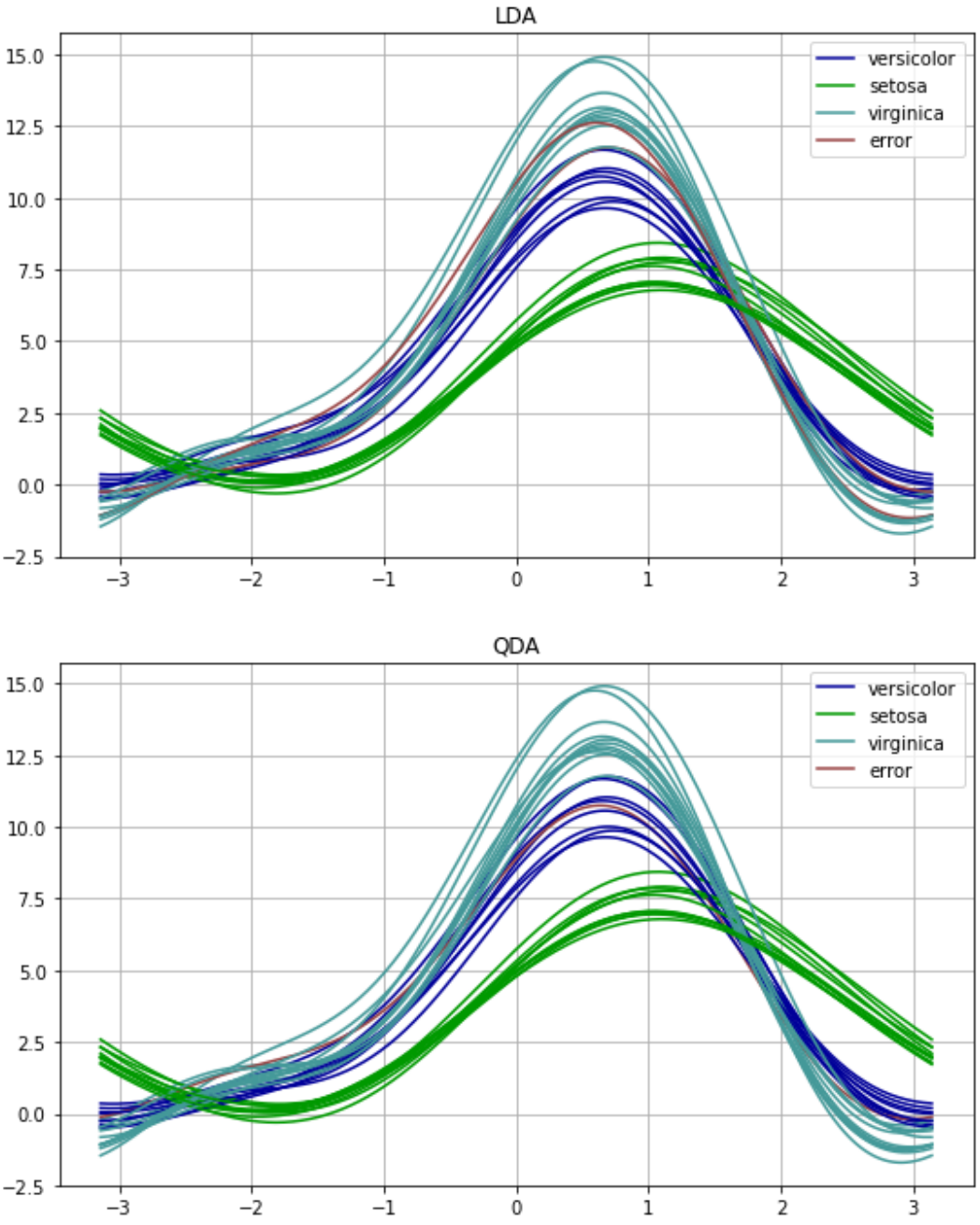
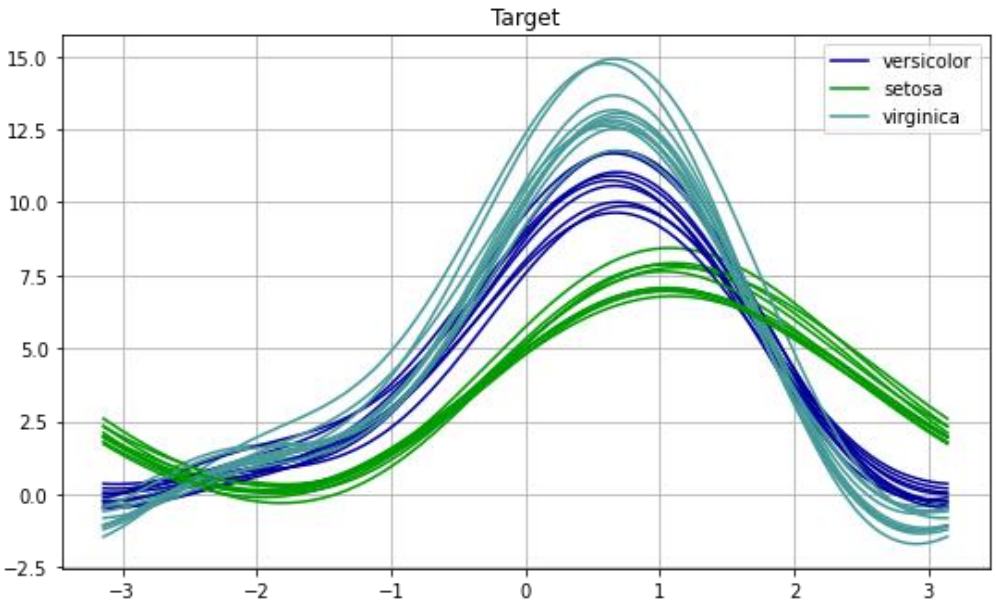


Рисунок 11 – Вывод графиков классификации по двум методам




```
Bootstrap632 =  [0.8982, 0.8676]

Bootstrap632+ =  [0.9291, 0.9031]
```

Рисунок 12 – График ожидания и точность

Как можно заметить, некоторые объекты имеют характерные признаки как одного класса, так и другого и именно такие объекты, располагающиеся на границе нескольких классов классифицировались неверно.

2.3 Вывод по второму разделу

В данном разделе были определены функции линейного и квадратичного дискриминантного анализа, функция вычисления лямбды Уилкса и функции по определению точности построенной модели. На примере простого анализа «обучающих» данных о трёх видах ирисов была продемонстрирована работа созданных функций, показан «базовый» процесс классификации, модификации которого способны анализировать любые другие данные и на чём будет основана работа анализа более сложных данных по сердечным заболеваниям, чей результат будет описан далее.

3 Тестирование

3.1 Описание ситуации

В этом разделе будет проводится анализ базы данных о сердечных заболеваниях о которой рассказывалось ранее. Всего имеется более трёхсот записей и свыше семидесяти числовых признаков. Такие данные позволяют продемонстрировать анализ «больших» данных, какие встречаются в современных задачах. Для решения этой задачи будет создана программа в соответствии с реализацией показанной в предыдущем разделе. Ознакомиться с ней можно в приложении на диске.

Перед началом процесса обучения будет происходить ряд операций, а именно:

- Предварительная обработка данных,
- Обучение классификатора и классификация,
- Визуализация полученных результатов и анализ.

Процесс обучения на исходной выборке будет происходить описанными ранее методами линейного и квадратичного дискриминантного анализа. В настоящих задачах зачастую, ковариационные матрицы могут быть различными, поэтому в зависимости от исследования и данных следует выбираться подходящий классификатор. Но в данный работе данные анализируются при помощи двух методов сразу, поэтому на этапе анализа дынных будет происходить два почти одинаковых действия, отличающихся лишь функцией классификации.

На последнем этапе будут рассматриваются современные методы, такие как bootstrap 0.632 и 0.632+, благодаря которым можно наиболее точно определить величину ошибочной классификации, по сравнению со стандартным определением точности, а именно – вычислении ошибки на обучении.

3.2 Анализ входных данных

В исходных данных имеется 76 признаков и некоторые из них имеют меньший вес в функции классификации. Конечно, при учёте всех признаков должна быть получена самая точная модельная, но такое часто не выполняется так как сами признаки могут быть неверно посчитаны или вообще пропущены в некоторых записях. Так в исследуемых данных находится десять целых десять признаков, которые не имеют своего описания, поэтому анализировать их нет смысла. Если анализировать данные дальше, то выяснится, что лишь только группа из четырнадцати признаков имеют важный приоритет по сравнению с другими и это подтверждается другими исследованиями проводимыми на основе этих данных.

Итак, имеется набор из четырнадцати данных по следующим пунктам:

- Возраст,
- Пол,
- Тип боли наблюдаемый в груди,
- Артериальное давление в состоянии покоя,
- Холестерин в сыворотке крови,
- Уровень сахара в крови,
- Результаты электрокардиографии,
- Максимальная частота сердечных сокращений,
- Стенокардия,
- Затухание в сегменте после нагрузки,
- Наклон пика в сегменте ST в ЭКГ,
- Количество крупных сосудов,
- Дефект сердца,
- Ангиографический статус заболевания, класс.

Для отбора важных признаков посчитаем для каждого из них свою лямбду Уилкса. Результаты анализа представлены таблице ниже.

Таблица 2 – Атрибут и его значимость вклада

№ признака	Название признака	Λ	№ признака	Название признака	Λ
1	age	0.98637	8	thalach	0.97632
2	sex	0.58573	9	exang	0.59126
3	cp	0.79016	10	oldpeak	0.84418
4	trestbps	0.96592	11	slope	0.74465
5	chol	0.94688	12	ca	0.72812
6	fbs	0.42399	13	thal	0.75058
7	restecg	0.67932	14	num	–

Как можно видеть наибольший вес имеют такие признаки, как возраст, давление, холестерин, и максимальная частота сердечных сокращений.

3.3 Полученный результат

Если отобрать только самые значащие признаки и установить пороговое значение в $a \in [0{,}85; 0{,}9]$, то результаты классификации методами линейного и квадратичного дискриминантного анализа покажет следующий результат. На рисунке 13, 14, 15 показано три графика Эндрю с результатами классификациями тестовой выборки методами линейного и квадратичного дискриминантного анализа и с заранее установленным значениями классов, где группа здоровых людей обозначена единицей, а все люди имеющие проблемы с сердцем обозначены нулём.

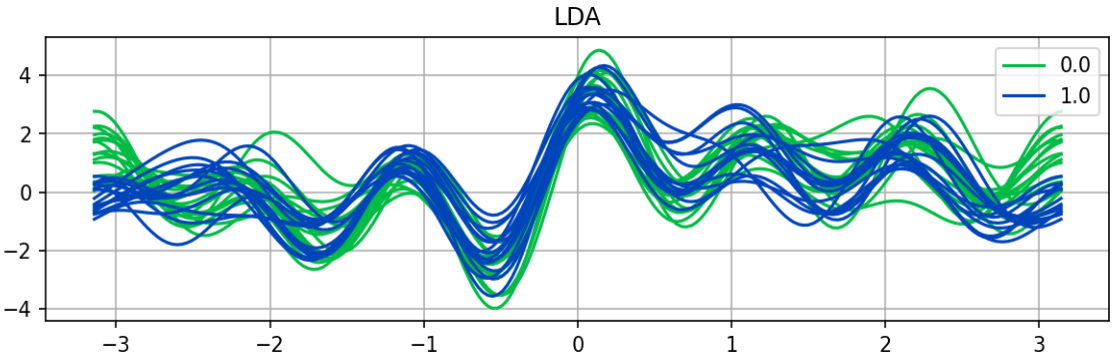


Рисунок 13 – Линейный анализ по четырём параметрам

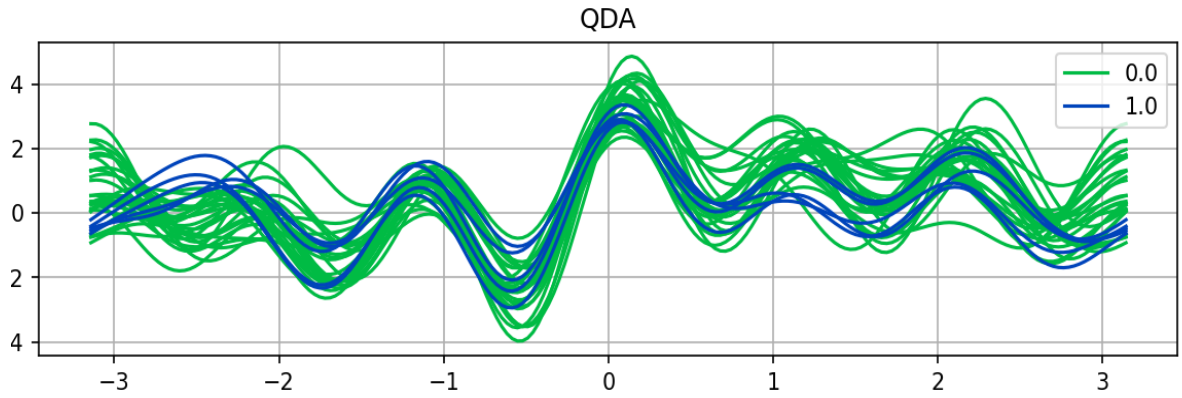


Рисунок 14 – Квадратичный анализ по четырём параметрам

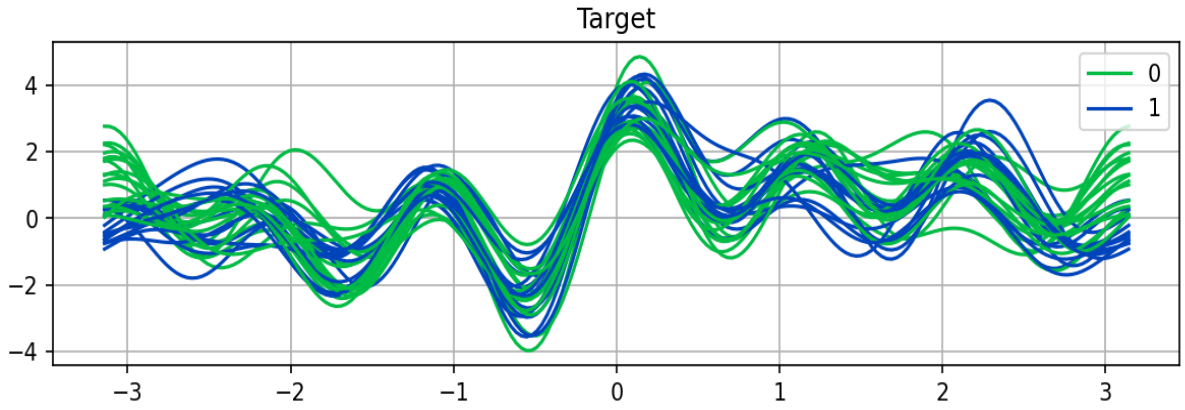


Рисунок 15 – Исходные значения установленных классов

Теперь сравним результаты и определим точность моделей.

3.3 Сравнение методов, определения точности модели

Посчитаем точность полученной модели. На рисунке 16 и 17 изображено два графика с результатами определения точности двумя версиями метода bootstrap. Как можно видеть, наибольшая точность двух методов достигается при установке порогового значения лямбды Уилкса более чем 0,8. Так же, квадратичный метод имеет большую точность (85%)

по сравнению с линейным (50%) при значении порога лямбды Уилкса равным 0,85. В этом можно наглядно убедиться, если построить график по признаку, вносящего наибольший вклад в функцию классификации на основе результатов из таблицы 2. На рисунке 18, 19 и 20 показаны результаты классификации для двух методов и исходных значений по признаку «возраст».

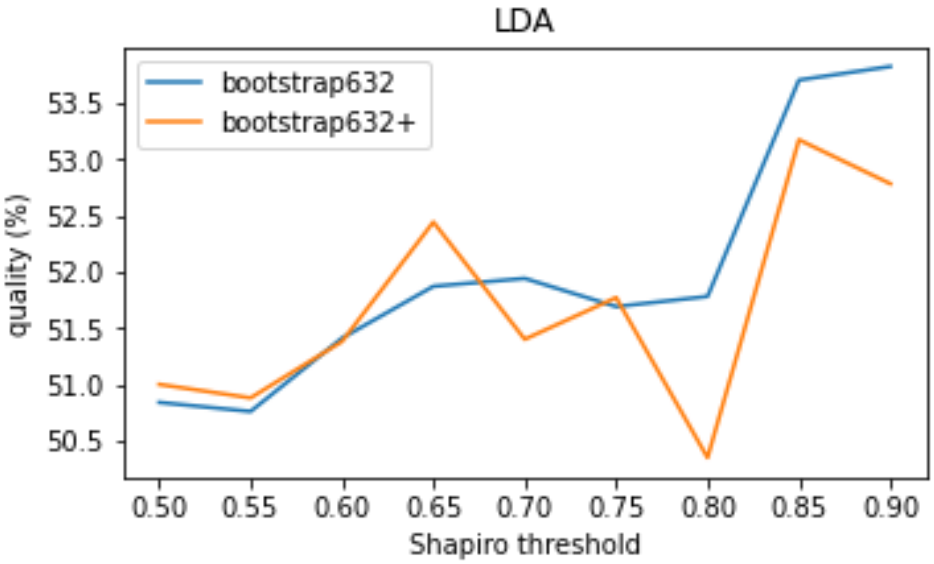


Рисунок 16 – Точность классификации линейного метода

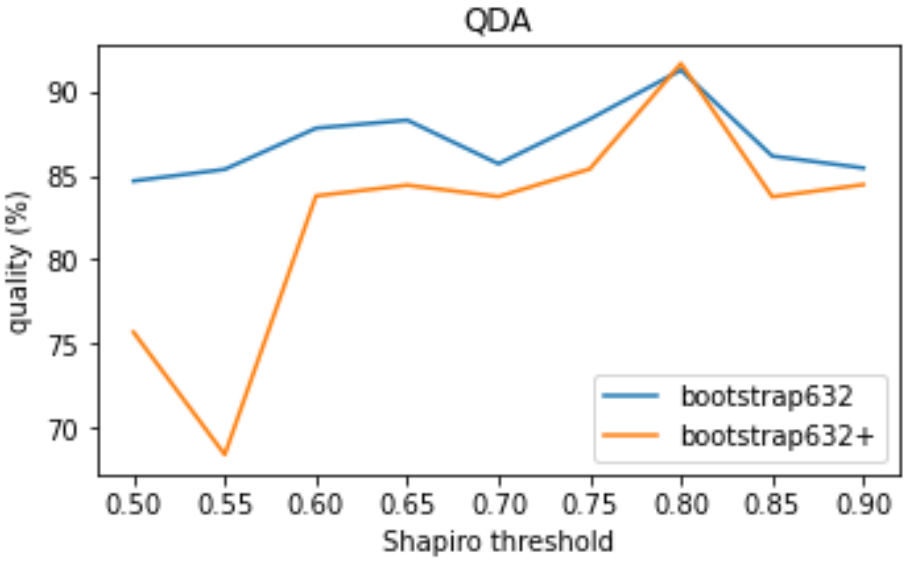


Рисунок 17 – Точность классификации квадратичного метода

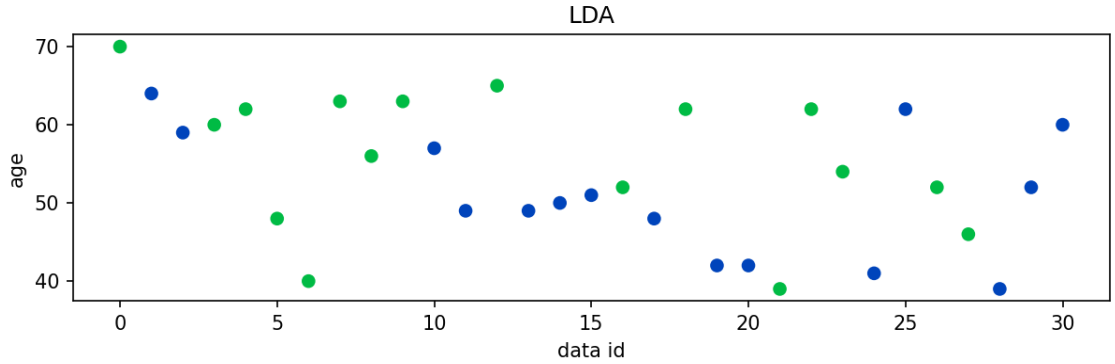


Рисунок 18 – Результат полученный линейным методом

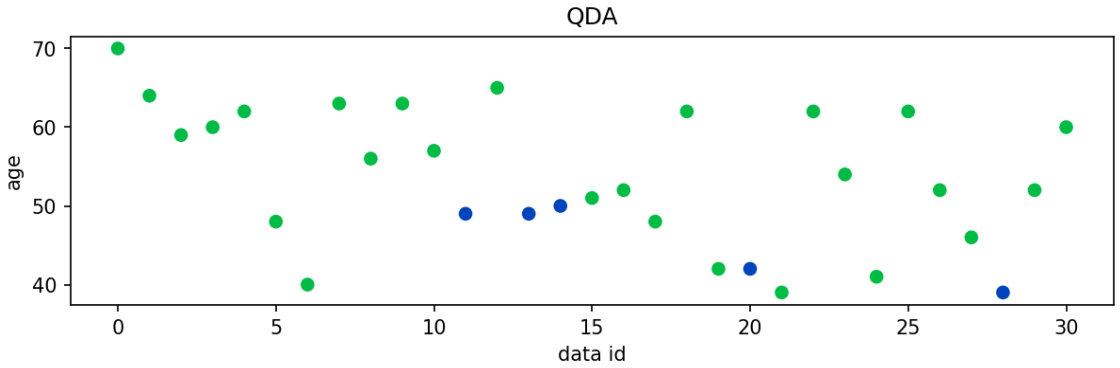


Рисунок 19 – Результат полученный квадратичным методом

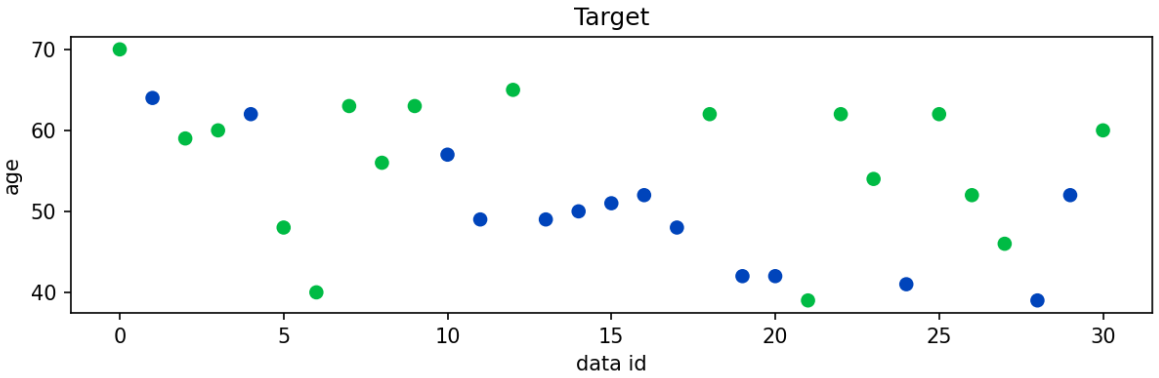


Рисунок 20 – Изначально определённые данные

Заключение

Тема бакалаврской работы была посвящена исследованию методов дискриминантного анализа, а также проблеме классификации при работе с многомерными данными.

В ходе выполнения был изучен теоретический материал, а также изучены несколько методов дискриминантного анализа:

- Линейный дискриминантный анализ;
- Квадратичный дискриминантный анализ;
- Смешанный дискриминантный анализ;
- Гибкий дискриминантный анализ;
- Регуляризированный дискриминантный анализ.

Для проведения исследования потребовалось рассмотреть методы определения ошибки bootstrap, а также метод отбора значащих признаков из общей совокупности признаков.

В ходе дальнейшего выполнения работы было разработано программное обеспечение на базе языка программирования Python с использованием библиотеки по работе с табличными данными, визуализацией полученных результатов и статистики.

В результате был рассмотрен ряд методов дискриминантного анализа, а также провелось исследование базы данных о сердечных заболеваниях с применением двух классификаторов линейного и квадратичного дискриминантного анализа. Предварительно исходные данные подверглись процедуре на выявление значащих признаков по значению лямбды Уилкса, а оценка точности классификации измерялась полученными результатами методами bootstrap 0.632 и bootstrap 0.632+. В итоге, была получена точность классификации (52–55%) для линейного дискриминантного анализа и (80–90%) для квадратичного дискриминантного анализа.

Список используемой литературы

1. Буре В. М., Щербакова А. А. Применение дискриминантного анализа и метода деревьев принятия решений для диагностики офтальмологических заболеваний // Вестник Санкт-Петербургского университета. Серия 10: Прикладная математика. Информатика. Процессы управления. 2013. № 1. С. 70–76.

2. Дж О. Ким Факторный, дискриминантный и кластерный анализ; Книга по Требованию - Москва, 2012. - 216 с.

3. Драницына М. А., Захарова Т. В. Дискриминантный анализ для классификации и прогнозирования результатов лечения // Системы и средства информатики, 2013. Т. 23. №2. С. 89-95.

4. Ефимов, В. М. Многомерный анализ биологических данных: учеб. пособие / В. М. Ефимов, В. Ю. Ковалева. -- С.П.: Институт систематики и экологии животных СОРАН, 2008. -- 87 с.

5. Кобзарь А. И. Прикладная математическая статистика. Для инженеров и научных работников. М.: ФИЗМАТЛИТ, 2006. 816 с.

6. Кондрашова, Н. В. Решение задачи медицинской диагностики линейным дискриминантным анализом и МГУА [Текст] / Н. В. Кондрашова, В. А. Павлов, А. В. Павлов // УСиМ. - 2013. - № 2. - С. 79-88.

7. Лутц, Марк. Программирование на Python: - СПб.: Символ-Плюс, 2015

8. Марк, Лутц Программирование на Python. Том 1 / Лутц Марк. - М.: Символ-плюс, 2013. - 822 с.

9. Митрофанов А.А., Кичук И.В., Соловьева Н.В., Кувшинова Я.В., Чаусова С.В., Вильянов В.Б., Русалова М.Н., Олимпиева С.П. Использование дискриминантного анализа электроэнцефалограммы в диагностике шизофрении. Журнал неврологии и психиатрии им. С.С. Корсакова. 2019;119(1):44-50

10. Саммерфилд, Марк. Программирование на Python 3. Подробное руководство. - СПб.: Символ-Плюс, 2017

11. Светлана Амирова, Юрий Мильчесвский. Дискриминантный анализ и структура белка. – М.: LAP Lambert Academic Publishing, 2014. – 104 с.

12. Тюрин, В. В. Дискриминантный анализ в биологии: монография / В. В. Тюрин, С. Н. Щеглов. – Краснодар: Кубанский гос. ун-т, 2015. – 123 с.

13. B.B. Welch, Practical programming in Tcl/Tk, 4th edn. (Prentice Hall/PTR, Upper Saddle River, NJ, 2003)

14. Dudoit S., Fridlyand J., Speed T. P. Comparison of discrimination methods for the classification of tumors using gene expression data // Journal of the American Statistical Association. 2002. Vol. 97 (457). P. 77–87.

15. Fisher R. A. The use of multiple measurements in taxonomic problems //Annals of Eugenics. 1936. №7. P. 179–188.

16. Hand D. J., Henley W. E. Statistical Classification Methods in Consumer Credit Scoring: A Review // Journal of the Royal Statistical Society. Series A (Statistics in Society). 1997. Vol. 160 (3). P. 523–541.

17. Haroon Barakat,El- Sayed Nigm and Osama Khaled. Evaluation of Air Pollutants Using Bootstrapping Extremes Models. – М.: LAP Lambert Academic Publishing, 2014. – 132 с.

18. Hastie T, Tibshirani R, Friedman J. The Elements of Statistical Learning. Data Mining, Inference, and Prediction. 2-nd Edition. Springer. 2009

19. Mark Lutz. Programming Python 4e. – М.: , 2011. – 1632 с.

20. Mohammad Samsul Alam and Syed Shahadat Hossain. Design Sensitivity of Bootstrap Methods in Variance Estimation. – М.: LAP Lambert Academic Publishing, 2013. – 72 с.

21. Rencher A. C. Methods of Multivariate Analysis. 2nd Ed. New York: John Wiley & Sons, Inc., 2002. 738 p.