Министерство образования Республики Беларусь

Учреждение Образования

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Кафедра интеллектуальных информационных технологий

Лабораторная работа №2

Статистические основы индуктивного вывода

Проверил:

Выполнили: Гаевой Александр(*121731*)

Минск

2023 г.

Содержание:

1. Цель ……………………………………………………………………………………3
2. Код программы………………………………………………………………………..4
3. Объяснение кода………………………………………………………………………5
4. Вывод программы……………………………………………………………………..7
5. Тезисы………………………………………………………………………………….8
6. Список источников……………………………………………………………………9

## Цель работы: Создать наивный Баесовский классификатор1, в последствии натренировать его на датасете “ Heart Failure Prediction ”, в котором содержится информация о пациентах и их здоровье,

## Задачей является предсказание смертности от сердечной недостаточности по значениям других признаков, таких как возраст, анемия, креатинин и диабет.

Код программы:

# Импортируем необходимые библиотеки

import pandas as pd

import numpy as np

from sklearn.naive\_bayes import GaussianNB

from sklearn.model\_selection import train\_test\_split

from sklearn.metrics import accuracy\_score

# Загружаем датасет из ссылки

data = pd.read\_csv("dts.csv")

# Выбираем признаки и целевую переменную

X = data.drop("DEATH\_EVENT", axis=1)

y = data["DEATH\_EVENT"]

# Разбиваем данные на обучающую и тестовую выборки

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

# Создаем и обучаем байесовский классификатор

clf = GaussianNB()

clf.fit(X\_train, y\_train)

# Предсказываем смертность от сердечной недостаточности для тестовой выборки

y\_pred = clf.predict(X\_test)

# Выводим предсказанные и реальные значения смертности для тестовой выборки

print(pd.DataFrame({"Predicted": y\_pred, "Actual": y\_test}))

# Вычисляем точность классификации

acc = accuracy\_score(y\_test, y\_pred)

print(f"Accuracy: {acc:.2f}")

Объяснение работы кода:

* Сначала я импортирую необходимые библиотеки для работы с данными и машинным обучением. Это pandas, numpy, sklearn.naive\_bayes, sklearn.model\_selection и sklearn.metrics. Они позволяют мне загружать, обрабатывать, моделировать и оценивать данные.
* Затем я загружаю датасет из ссылки с помощью функции pd.read\_csv. Эта функция читает файл csv из интернета или локального диска и преобразует его в объект pandas.DataFrame, который представляет собой таблицу с данными. Я сохраняю этот объект в переменную data.
* Далее я выбираю признаки и целевую переменную для задачи классификации. Признаки - это характеристики пациентов, такие как возраст, анаемия, креатинин и диабет, которые могут влиять на исход сердечной недостаточности. Целевая переменная - это факт смерти от сердечной недостаточности, который я хочу предсказать. Я использую метод drop для удаления столбца DEATH\_EVENT из data и сохраняю оставшиеся столбцы в переменную X. Я сохраняю столбец DEATH\_EVENT в переменную y.
* Затем я разбиваю данные на обучающую и тестовую выборки с помощью функции train\_test\_split. Эта функция случайным образом разделяет данные на две части: одну для обучения модели, а другую для проверки ее качества. Я задаю размер тестовой выборки равным 0.2, то есть 20% от всех данных. Я также задаю параметр random\_state равным 42, чтобы воспроизводить результаты при повторном запуске кода. Я сохраняю четыре части данных в переменные X\_train, X\_test, y\_train и y\_test.
* Далее я создаю и обучаю байесовский классификатор с помощью класса *GaussianNB2* из библиотеки sklearn.naive\_bayes. Это модель машинного обучения, которая использует теорему Байеса для вычисления вероятности принадлежности к каждому классу на основе признаков. Я создаю объект clf и вызываю метод fit для обучения модели на данных X\_train и y\_train.
* Затем я предсказываю смертность от сердечной недостаточности для тестовой выборки с помощью метода predict. Этот метод принимает данные X\_test и возвращает массив y\_pred с предсказанными значениями 0 или 1 для каждого пациента.
* Наконец, я вычисляю точность классификации с помощью функции accuracy\_score из библиотеки sklearn.metrics. Эта функция принимает реальные значения y\_test и предсказанные значения y\_pred и возвращает долю правильных предсказаний. Я сохраняю эту долю в переменную acc и выводу ее на экран с помощью функции print.

Вывод программы:

id Predicted Actual

281 0 0

265 0 0

164 0 1

9 1 1

77 0 0

278 0 0

93 0 1

…

25 1 1

209 0 0

59 1 1

Accuracy: 0.73

PS C:\Users\lenovo\Desktop\STATOIV Lab\_1>

Это таблица, в которой каждая строка соответствует одному пациенту, а каждый столбец - предсказанному или реальному значению смертности. Значение 1 означает, что пациент умер от сердечной недостаточности, а значение 0 - что нет. Вы можете сравнить эти значения и посмотреть, где модель сделала правильное или неправильное предсказание. Например, если в первой строке таблицы Predicted = 0 и Actual = 1, это означает, что модель ошибочно предсказала, что пациент не умер от сердечной недостаточности, хотя на самом деле он умер.

Тезисы:

[Наивный байесовский классификатор ]1— простой вероятностный классификатор, основанный на применении теоремы Байеса со строгими (наивными) предположениями о независимости.

В зависимости от точной природы вероятностной модели, наивные байесовские классификаторы могут обучаться очень эффективно. Во многих практических приложениях для оценки параметров для наивных байесовых моделей используют метод максимального правдоподобия ; другими словами, можно работать с наивной байесовской моделью, не веря в баесовскую вероятность и не используя байесовские методы.

Несмотря на наивный вид и, несомненно, очень упрощенные условия, наивные байесовские классификаторы часто работают намного лучше нейронных сетей во многих сложных жизненных ситуациях.

[GaussianNB]2 - это модель машинного обучения, которая использует теорему Байеса для классификации объектов на основе их признаков. Теорема Байеса позволяет вычислить вероятность принадлежности объекта к определенному классу, используя априорную информацию о распределении классов и признаков. GaussianNB предполагает, что признаки имеют нормальное (гауссовское) распределение внутри каждого класса. Это означает, что для каждого признака и класса можно оценить среднее значение и дисперсию. Затем можно использовать формулу нормального распределения для вычисления вероятности того, что объект имеет определенное значение признака при условии, что он принадлежит определенному классу. GaussianNB учитывает все признаки и классы и выбирает тот класс, для которого вероятность наибольшая.

Список источников:

1. [Наивный байесовский классификатор — Википедия (wikipedia.org)](https://ru.wikipedia.org/wiki/%D0%9D%D0%B0%D0%B8%D0%B2%D0%BD%D1%8B%D0%B9_%D0%B1%D0%B0%D0%B9%D0%B5%D1%81%D0%BE%D0%B2%D1%81%D0%BA%D0%B8%D0%B9_%D0%BA%D0%BB%D0%B0%D1%81%D1%81%D0%B8%D1%84%D0%B8%D0%BA%D0%B0%D1%82%D0%BE%D1%80)
2. [sklearn.naive\_bayes.GaussianNB — scikit-learn 1.3.2 documentation](https://scikit-learn.org/stable/modules/generated/sklearn.naive_bayes.GaussianNB.html)
3. [How Naive Bayes Algorithm Works? (with example and full code) | ML+ (machinelearningplus.com)](https://www.machinelearningplus.com/predictive-modeling/how-naive-bayes-algorithm-works-with-example-and-full-code/)
4. [python - Do the score() and predict() methods on GaussianNB also fit the model? - Stack Overflow](https://stackoverflow.com/questions/28035406/do-the-score-and-predict-methods-on-gaussiannb-also-fit-the-model)