Министерство образования Республики Беларусь

Учреждение образования

“Белорусский государственный университет

информатики и радиоэлектроники”

Факультет информационных технологий и управления

Кафедра интеллектуальных информационных технологий

**Лабораторная работа №4**

по дисциплине «Статистические основы индуктивного вывода»

Выполнил: Поплавский Э. Э.

Группа: 021703

Проверил: Name N. N.

Минск 2023

1. **Цели**

Приобрести навыки построения дерева решений. Интерпретировать работу дерева решений.

1. **Задание**
   1. Выбрать с сайта kaggle.com набор данных в формате .csv, пригодный для построения дерева решений, загрузить и подготовить его к дальнейшей обработке. Если набор уже имеет метки классов – удалить метки.
   2. Построить модель дерева решений
   3. Оценить точность модели построив матрицу сходства
   4. Получить интерпретацию построенной модели
   5. Указать какие знания можно получить из набора
   6. Сохранить IPython Notebook
2. **Выполнение**

Для выполнения лабораторной работы был выбран набор данных для распознавания вин из Калифорнийского университета в Ирвине.

Этот набор данных является результатом химического анализа вин, выращенных в одном регионе Италии, но полученных из трех разных сортов. В результате анализа было определено количество 13 компонентов, содержащихся в каждом из трех видов вин.

В лабораторной работе реализовано построение дерева решений для классификации данных по колонке "name" - номер, обозначающий определенный сорт вина. Остальные содержащиеся атрибуты данных в наборе: алкоголь, яблочная кислота, пепел, щелочность золы, магний, общее количество фенолов, флаваноиды, нефлаваноидные фенолы, проантоцианы, интенсивность цвета, оттенок, od280/od315 разбавленных вин, пролин. (В оригинале: alcohol, malic acid, ash, alcalinity of ash, magnesium, total phenols, flavanoids, nonflavanoid phenols, proanthocyanins, color intensity, hue, od280/od315 of diluted wines, proline)

Набор данных содержит 178 записей. Пример записей в файле с набором данных представлен на Рисунке 1.

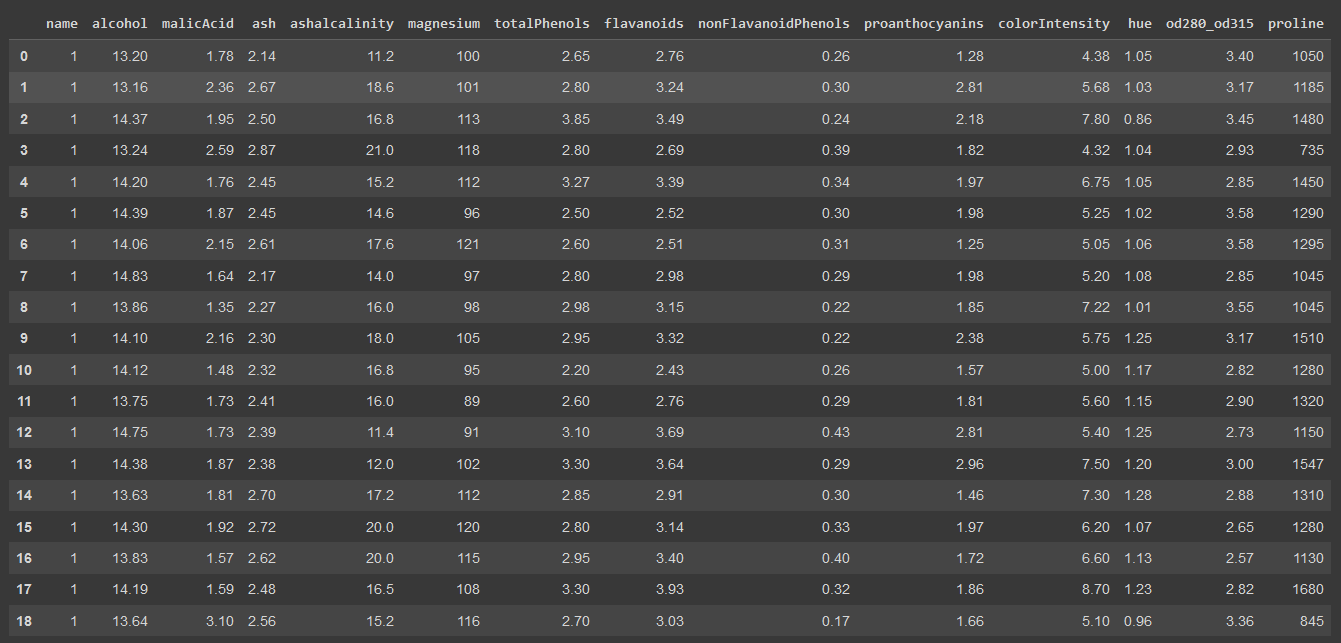


Рисунок 1. Пример данных из выбранного набора.

Далее будет приведен код, использованный для построения дерево решений, а так же сопутствующие вычисления и графики.

Импорт необходимых библиотек

import sklearn

import pandas as pd

import seaborn as sns

import matplotlib.pyplot as plt

from sklearn import tree

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

import os

Проверка загружен ли csv файл

print(os.path.exists('wine.csv'))

Чтение датасета

PATH = "wine.csv"

dataset = pd.read\_csv(PATH)

dataset.columns = ['name', 'alcohol', 'malicAcid', 'ash', 'ashalcalinity',

                   'magnesium', 'totalPhenols', 'flavanoids',

                   'nonFlavanoidPhenols', 'proanthocyanins', 'colorIntensity',

                   'hue', 'od280\_od315', 'proline']

Визуальная проверка данных на мультиколлинеарность при помощи  тепловой карты, представленной на Рисунке 2.

sns.heatmap(round(abs(dataset.corr()), 1,), annot=True)

plt.show()

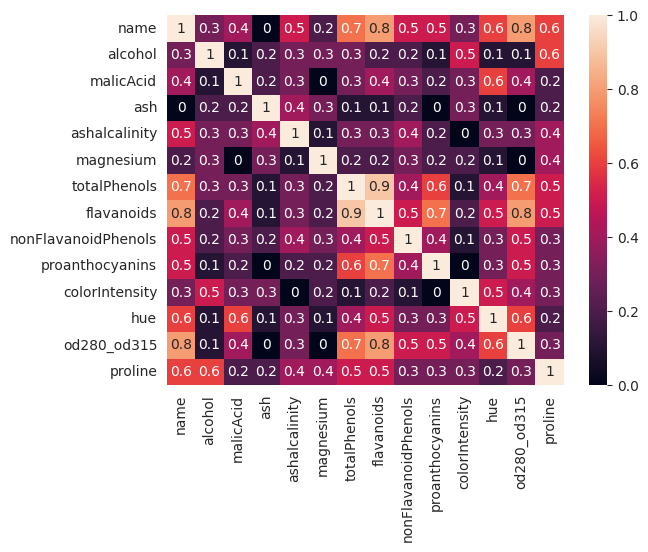


Рисунок 2. Тепловая карта связей столбцов данных в наборе.

Из набора будут удалены столбцы totalPhenols, flavanoids, od280\_od315, proline, hue так как они мультиколлинеарны. Факторизация столбцов не требуется. Пример данных из набора после удаления столбцов представлен на Рисунке 3.

dataset.drop(

[

    "totalPhenols",

    "flavanoids",

    "od280\_od315",

],

axis=1,

inplace=True,)

dataset.head(20)

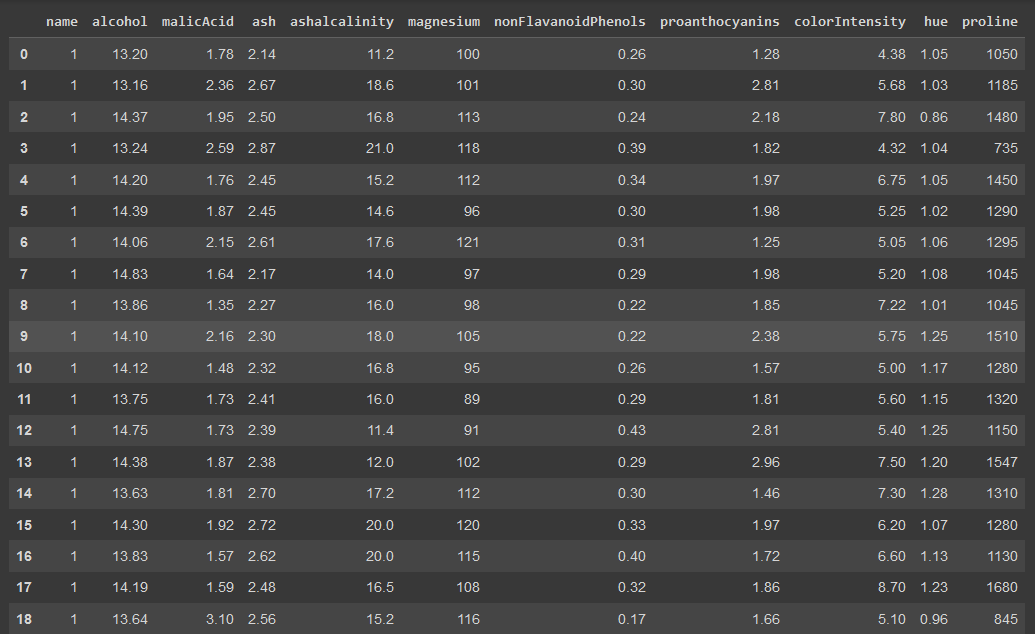


Рисунок 3. Пример данных из выбранного набора, после удаления столбцов.

Тепловой таблица после удаления столбцов представлена на Рисунке 4.

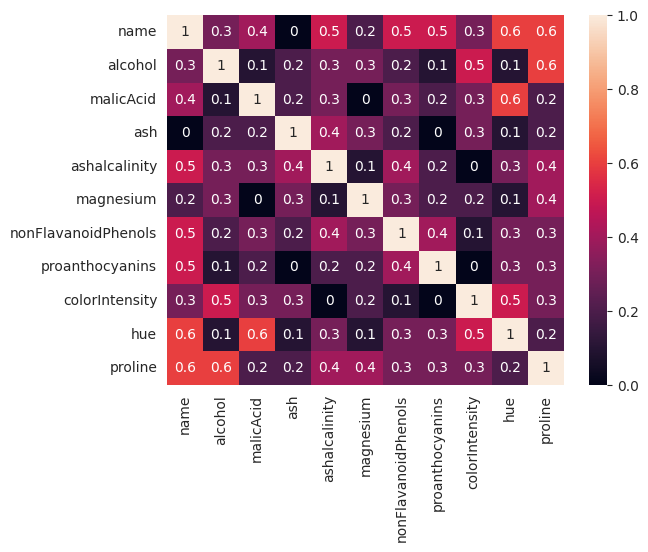


Рисунок 4. Тепловая таблица данных, после удаления столбцов.

Перед построением модели данные будут поделены на тренировочные и тестовые

train\_input, test\_input, train\_output, test\_output = train\_test\_split(

    dataset.drop("name", axis=1),

    dataset["name"],

    test\_size=0.2

)

Код ниже строит модель дерева решения, а также выводит на экран матрицу сходства предсказанных значений тестового набора и эталонных значений (Рисунок 5).

# logic

model = tree.DecisionTreeClassifier()

model.fit(train\_input, train\_output)

predictions = model.predict(test\_input)

confusion\_matrix = sklearn.metrics.confusion\_matrix(predictions, test\_output)

# output

sns.heatmap(confusion\_matrix, annot=True)

plt.title('Матрица сходства')

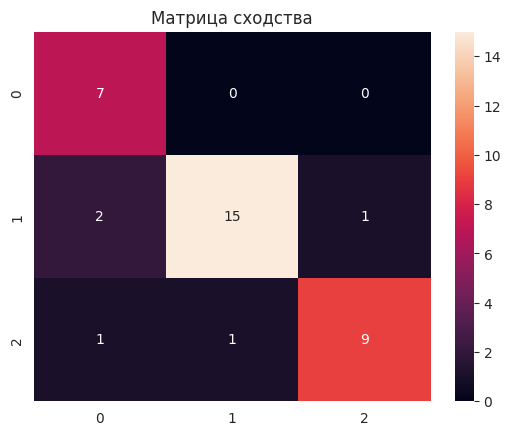


Рисунок 5. Матрица сходства предсказанных значений и эталонных значений.

Получения самого дерева решения (Рисунок 6) используется следующий код

DPI = 450

plt.figure(dpi=DPI)

tree.plot\_tree(model)

plt.title("Дерево решений")



Рисунок 6. Дерево решений.

Полученное дерево показывает алгоритм, по которому идет отношение определенных данных к какому-либо классу. Его точность отображается матрицей соотношений, как видно из нее лучше всего тестовые данные были распределены для 2 класса вин, классы 1 и 3 были определены примерно одинаково. Чем больше массив обучающих данных, тем качественнее идет классификация при использование DI3

1. **Вывод**

Во время выполнения лабораторной работы были приобретены навыки построения дерева решений. В ходе выполнения был выбран с сайта kaggle.com набор данных в формате .csv, пригодный для построения дерева. После он был загружен и подготовлен к дальнейшей обработке. Под подготовкой подразумевается удаления столбца данных, размечающих данные (меток классов), а так же удаления мультиколлинеарных столбцов. Была построено дерево решений при помощи языка Python и библиотеки sklearm. Было определено какие знания можно получить из набора данных, мы соотнесли исходные размеченные данные и полученные значения, получив интерпретацию прохождения данных через дерево.