МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное   
учреждение высшего образования

**"Южно-Уральский государственный университет**

**(национальный исследовательский университет)"**

**Высшая школа электроники и компьютерных наук**

**Кафедра системного программирования**

**ОТЧЕТ**о выполнении лабораторной работы № 3

по дисциплине  
«Технологии аналитической обработки информации»

|  |  |
| --- | --- |
|  | Выполнил: студент группы КЭ-403  Разуев Г.А.  Проверил:  Преподаватель кафедры СП  Гоглачев А.И. |

Челябинск-2025

ОГЛАВЛЕНИЕ

[1. ЗАДАНИЕ 2](#_Toc191729743)

[2. РЕАЛИЗАЦИЯ ПРОГРАММЫ 3](#_Toc191729744)

[3. ЭКСПЕРИМЕНТЫ 5](#_Toc191729745)

# ЗАДАНИЕ

1. Разработайте программу, которая выполняет классификацию заданного набора данных с помощью дерева решений. Параметрами программы являются набор данных, критерий выбора атрибута разбиения (Information gain, Gain ratio, Gini index).
2. Проведите эксперименты на наборе Census Income.
3. Выполните визуализацию построенных деревьев решений.
4. Доработайте программу, добавив в список ее параметров долю, которую занимает обучающая выборка от общего размера набора данных, и обеспечив вычисление и выдачу в качестве результатов следующих показателей качества классификации: аккуратность (accuracy), точность (precision), полнота (recall), F-мера.
5. Проведите эксперименты на наборе данных, фиксируя критерий выбора атрибута разбиения и варьируя соотношение мощностей обучающей и тестовой выборок от 60%:40% до 90%:10% с шагом 10%.
6. Выполните визуализацию полученных результатов в виде следующих диаграмм:
   1. построенные деревья решений для заданного набора данных;
   2. показатели качества классификации в зависимости от соотношения мощностей обучающей и тестовой выборок для заданного набора данных.
7. Подготовьте отчет о выполнении задания и загрузите отчет в формате PDF в систему. Отчет должен представлять собой связный и структурированный документ со следующими разделами:
   1. формулировка задания;
   2. гиперссылка на каталог репозитория с исходными текстами, наборами данных и др. сопутствующими материалами;
   3. рисунки с результатами визуализации;
   4. пояснения, раскрывающие смысл полученных результатов.

# РЕАЛИЗАЦИЯ ПРОГРАММЫ

Реализация программы представлена в классе «DecisionTreeModel», который инкапсулирует логику по обучению и использованию модели, а также по предобработке данных и их разбиении при обучении. Один из ключевых атрибутов класса – критерий выбора атрибута разбиения. Данный класс реализует стандартный интерфейс Scikit-Learn (имеет методы «fit» и «predict»), однако помимо данных метод «fit» имеет параметр размера тестовой выборки, что позволяет как просто обучать модель на всех данных, так и экспериментировать, сразу вычисляя метрики качества на тренировочных и тестовых данных. Для предобработки сырых данных реализован метод «\_preprocess\_data», который является приватным. Он создает объекты препроцессоров («StandardScaler» для числовых данных и «OneHotEncoder» для категориальных), а потом в зависимости от параметров обучения модели применяет преобразования к данным. Также данный класс реализует метод «visualize\_tree», который позволяет отобразить предварительно обученное дерево решений.

Код реализованной программы и всех проведенных экспериментов находится в репозитории по ссылке …

Программа была протестирована полном наборе данных «Census Income», который содержал как числовые, так и категориальные данные. Код применения реализованной программы представлен в листинге 1, результаты применения метода визуализации дерева представлены на рисунке 1.

Листинг 1 – Код применения реализованной программы

# Создаем экземпляр класса DecisionTreeModel с критерием

model = DecisionTreeModel(criterion='gain\_ratio', max\_depth=3)

# Обучение модели

model.fit(full\_df.drop(columns=['target']), y=full\_df.target, test\_size=0, verbose=True)

model.visualize\_tree()

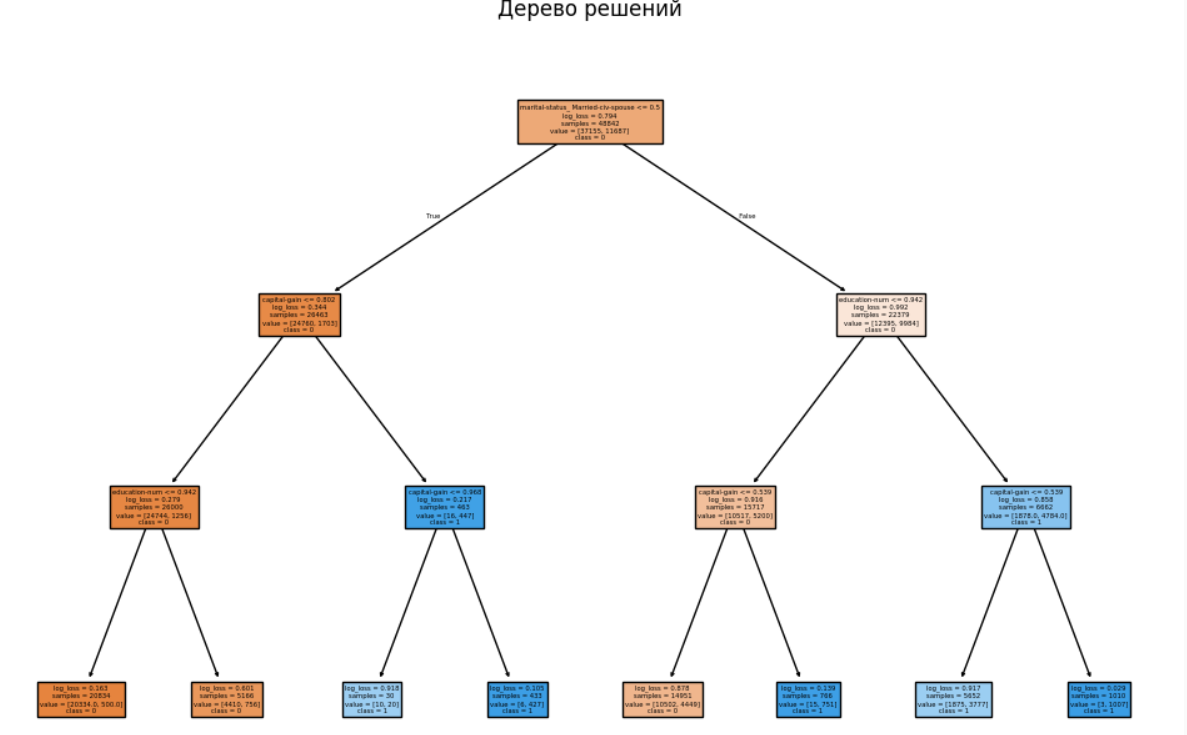


Рисунок 1 – Результат применения метода визуализации дерева

# ЭКСПЕРИМЕНТЫ

Были проведены эксперименты на наборе данных, с фиксацией критерия выбора атрибута разбиения и варьированием соотношение мощностей обучающей и тестовой выборок от 60%:40% до 90%:10% с шагом 10%. Код проведения эксперимента представлен в листинге 2.

Листинг 2 – Код применения реализованной программы

for criterion in criteria:

model = DecisionTreeModel(criterion)

for test\_size in test\_sizes:

\_ = model.fit(full\_df.drop(columns=['target']), full\_df['target'], test\_size=test\_size)

# Сохранение метрик

results[criterion]["accuracy"].append(model.accuracy)

results[criterion]["precision"].append(model.precision)

results[criterion]["recall"].append(model.recall)

results[criterion]["f1\_score"].append(model.f1\_score)

На основании проведенных экспериментов была выполнена визуализация результатов с использованием линейной диаграммы, на которой ось X – значение размера тестовой выборки, ось Y – метрика качества классификации. Для каждой метрики качества были построены свои сравнительные диаграммы, которые представлены на рисунках 2 – 6.

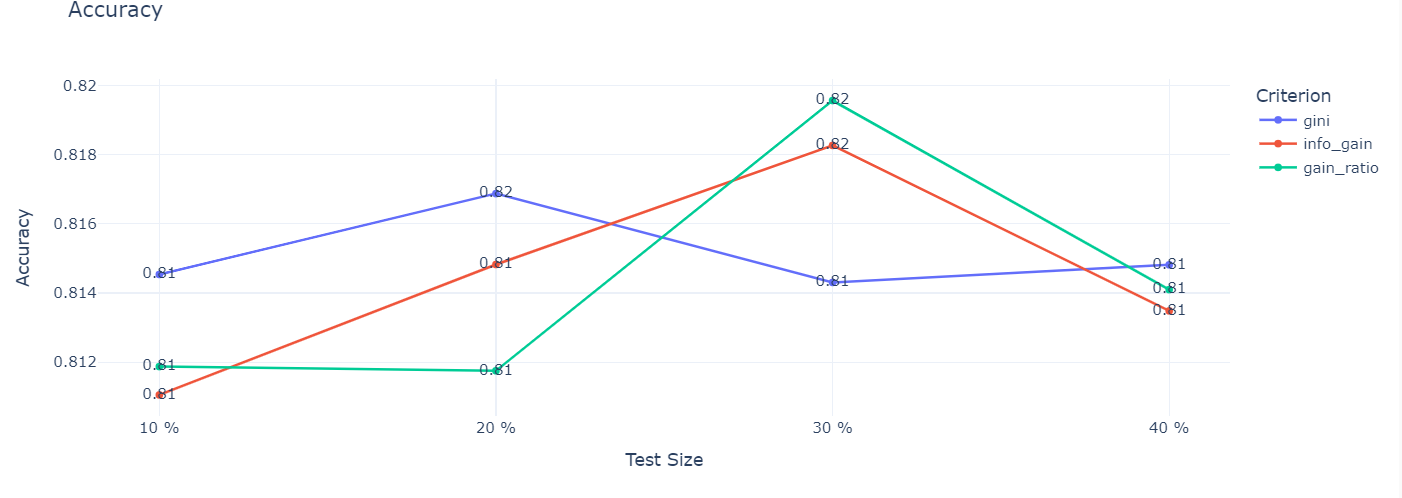


Рисунок 2 – Диаграмма сравнения Accuracy

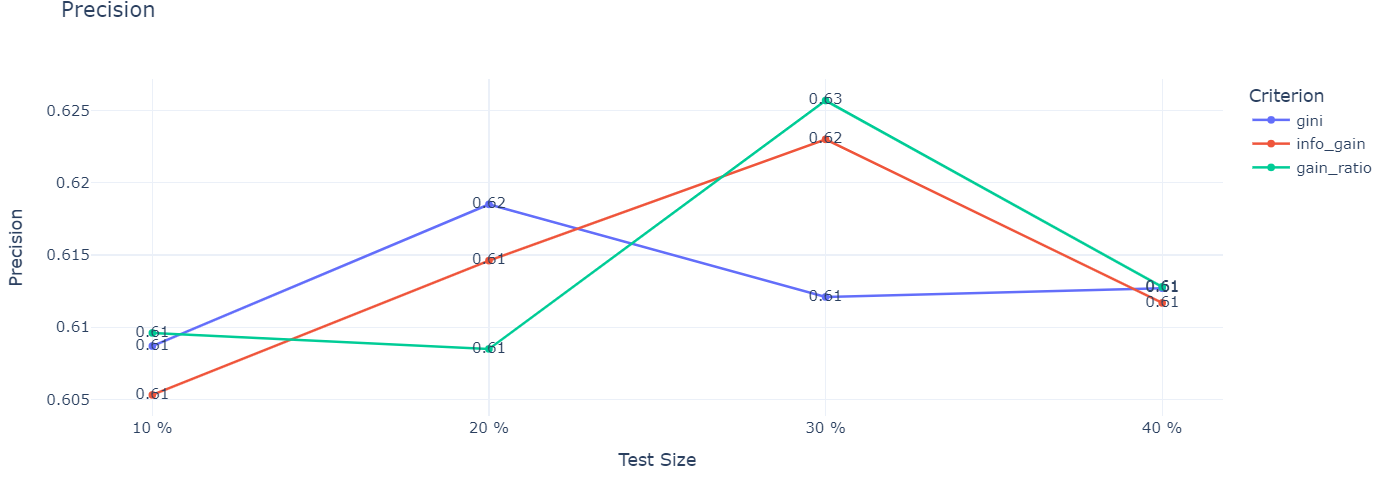


Рисунок 3 – Диаграмма сравнения Precision

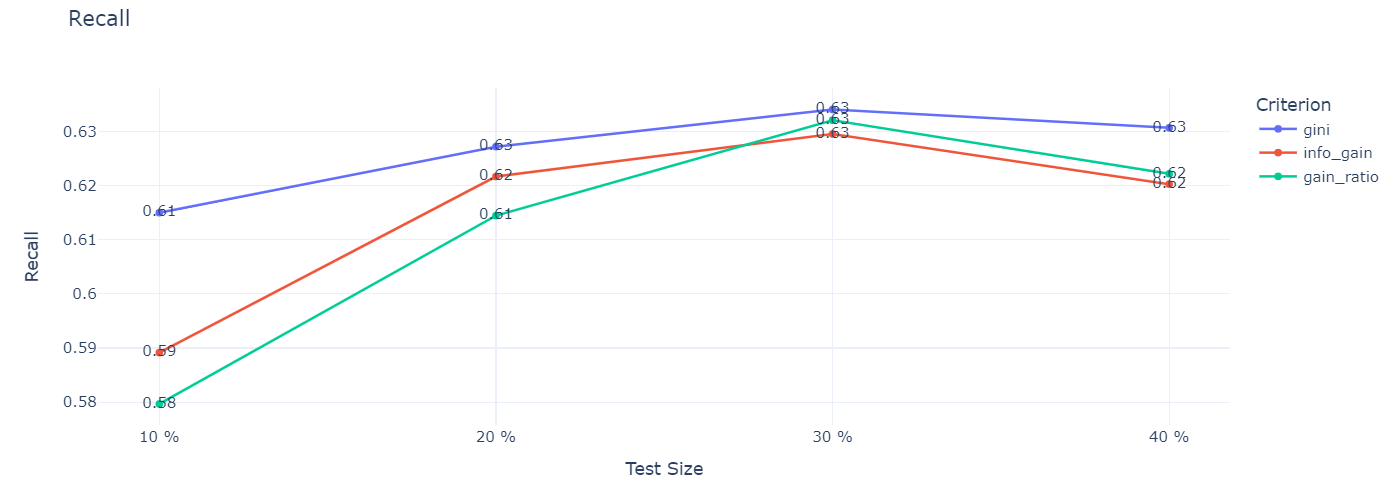


Рисунок 4 – Диаграмма сравнения Recall

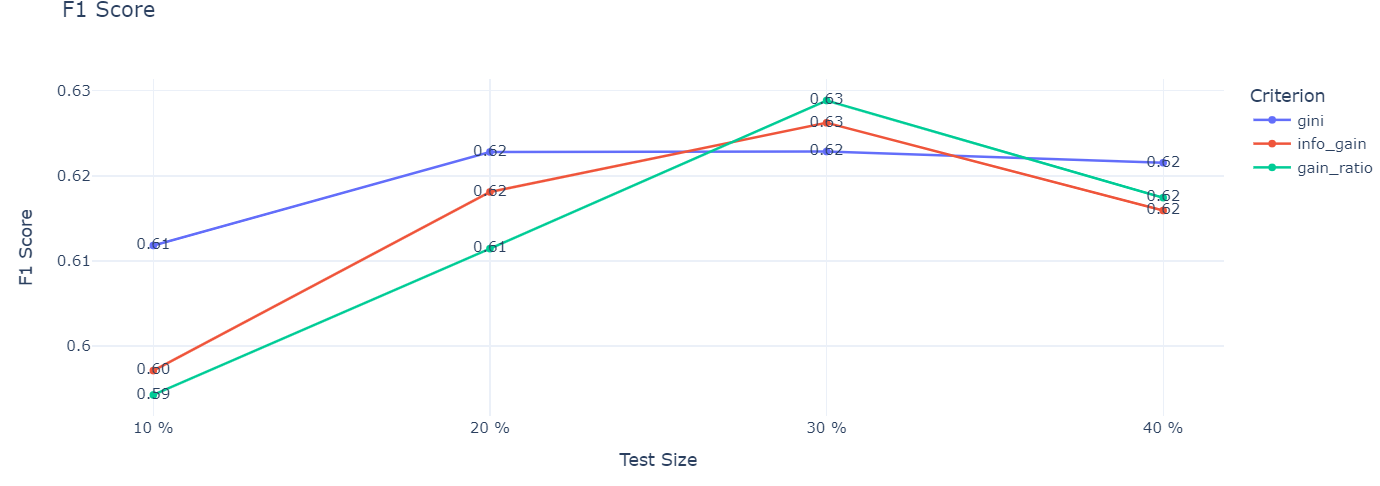


Рисунок 5 – Диаграмма сравнения F1 Score

Из анализа данных графиков можно сделать ряд выводов:

1. Лучшие результаты почти по всем метрикам достигаются при мощности тестовой выборки в 30 %, что говорит о том, что именно такой показатель надо использовать для обучения финальной модели;
2. Метрики качества у классификаторов на разных критериях отличаются на тысячные доли, что говорит об отсутствии принципиальной разницы между качеством предсказаний у этих критериев;
3. Однако уместнее всего сделать выбор в пользу критерия «gain\_ratio» т.к. он демонстрирует лучший результат в большинстве метрик при мощности тестовой выборки в 30 %.