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 практической работы № 3

по дисциплине

«Технологии аналитической обработки информации»

Выполнил:
студент группы КЭ-403
Гольденберг Д.И.

Проверил:
Преподаватель кафедры СП
Гоглачев А.И.

Челябинск, 2025 г.

ОГЛАВЛЕНИЕ

1. 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 (данные о результатах переписи населения, в т.ч. о годовом доходе -- ниже или выше $50000). В качестве обучающей выборки для построения дерева используйте 100% исходных данных.

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:

- построенные деревья решений для заданного набора данных;
- 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:

- формулировка задания;

- гиперссылка на каталог репозитория с исходными текстами, наборами данных и др. сопутствующими материалами;
- рисунки с результатами визуализации;
- пояснения, раскрывающие смысл полученных результатов.

2. РЕАЛИЗАЦИЯ АЛГОРИТМА

Код реализованной программы и всех проведенных экспериментов находится в репозитории по ссылке https://github.com/Goldria/analitycs/blob/main/3_decision_tree_classification/decision_tree_classification.ipynb

Набор данных взят из базы данных Бюро переписи населения. Данные представлены двумя выборками: обучающей (32 561 запись) и тестовой (16 281 запись). Общий объем данных составляет 48 842 записи.

Признаки в наборе данных: возраст (age), класс работы (workclass), финальный вес (fnlwgt), образование (education), количество лет образования (education-num), семейное положение (marital-status), род деятельности (occupation), отношения (relationship), раса (race), пол (sex), доход от капитала (capital-gain), потери капитала (capital-loss), количество рабочих часов в неделю (hours-per-week), страна проживания (native-country), доход (целевой признак: >50K или <=50K)

В наборе данных наблюдается дисбаланс классов:

- Доход >50K: 23.93%
- Доход <=50K: 76.07%

Перед обучением модели была проведена предобработка данных:

- Удаление строк с пропущенными значениями.
- Кодирование категориальных признаков с использованием Label Encoding.
- Балансировка классов методом RandomOverSampler, что увеличило размер выборки до 77 044 записей.

Код предобработки данных представлен в листинге 1.

Листинг 1 – Реализация предобработки данных

```
def balance_dataset(df):
    """Балансировка датасета"""
    X = df.drop(columns=['income'])
    y = df['income']
    y_numeric = (y == ' >50K').astype(int)

    ros = RandomOverSampler(random_state=42)
    X_resampled, y_resampled = ros.fit_resample(X, y_numeric)

    X_balanced = pd.DataFrame(X_resampled, columns=X.columns)
```

```
    y_balanced = pd.Series(y_resampled)
    balanced_dataset = pd.concat([X_balanced, y_balanced], axis=1)
    balanced_dataset['income'] = balanced_dataset['income'].map({0: '
<=50K', 1: ' >50K'})

    return balanced_dataset

def preprocess_data(df):
    """Функция предобработки данных."""
    df = balance_dataset(df)
    df = df.dropna()
    df = df.apply(lambda x: x.str.strip() if x.dtype == "object" else x)

    # Преобразование категориальных признаков в числовые
    label_encoders = {}
    for col in df.select_dtypes(include=["object"]).columns:
        le = LabelEncoder()
        df[col] = le.fit_transform(df[col])
        label_encoders[col] = le

    return df, label_encoders

df, label_encoders = preprocess_data(df)
```

Для обучения использовался алгоритм дерева решений (DecisionTreeClassifier) из библиотеки scikit-learn. Рассматривались три критерия разбиения:

1. Gini Index – мера неоднородности узла.
2. Entropy (Information Gain) – оценка уменьшения энтропии при разбиении.
3. Gain Ratio – нормализованное значение Information Gain.

Обучение проводилось на 100% данных, а также с разными соотношениями обучающей и тестовой выборок с соотношением обучающей и тестовой выборкой 8:2.

Реализация обучения с различными параметрами представлена в листинге 2.

Листинг 2 – Реализация обучения

```
X = df.drop(columns=["income"])
y = df["income"]

def train_and_evaluate(X, y, criterion, test_size=0.2):
    """Обучение и оценка модели дерева решений."""
    X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=test_size, random_state=42)
    clf = DecisionTreeClassifier(criterion=criterion, max_depth=15, ran-
dom_state=42)
    clf.fit(X_train, y_train)
```

```
    y_pred = clf.predict(X_test)

    metrics = {
        "Accuracy": accuracy_score(y_test, y_pred),
        "Precision": precision_score(y_test, y_pred),
        "Recall": recall_score(y_test, y_pred),
        "F1": f1_score(y_test, y_pred)
    }

    return clf, metrics

# Обучение модели
criterion = "gini"
test_size = 0.2
clf, metrics = train_and_evaluate(X, y, criterion=criterion,
test_size=test_size)
metrics
```

3. ЭКСПЕРИМЕНТЫ

3.1. Оценка качества модели

Для оценки качества классификации рассчитывались следующие метрики:

- Accuracy (Точность) – общая доля правильно классифицированных объектов.
- Precision (Прецизионность) – доля правильно предсказанных положительных примеров от всех предсказанных положительных.
- Recall (Полнота) – доля правильно найденных положительных примеров от всех истинно положительных.
- F1-score – гармоническое среднее Precision и Recall.

Результаты метрик для критерий gini и entropy представлены в таблице 1 и 2 соответственно.

Таблица 1 – Метрики критерия gini

Доля обучающей выборки	Accuracy	Precision	Recall	F1-score
0.6	0.8310	0.7820	0.9262	0.8480
0.7	0.8336	0.7824	0.9316	0.8505
0.8	0.8404	0.7841	0.9453	0.8572
0.9	0.8423	0.7843	0.9480	0.8584

Таблица 2 – Метрики критерия entropy

Доля обучающей выборки	Accuracy	Precision	Recall	F1-score
0.6	0.8251	0.7824	0.9094	0.8412
0.7	0.8259	0.7770	0.9222	0.8434
0.8	0.8313	0.7838	0.9214	0.8471
0.9	0.8372	0.7897	0.9231	0.8512

3.2. Визуализация результатов

Для визуализации построены деревья решений для разных значений train_size (0.6, 0.7, 0.8, 0.9) и разных критериев разбиения, деревья визуализированы с помощью graphviz.

Построены графики зависимости метрик (Accuracy, Precision, Recall, F1-score) от доли обучающей выборки на основе полученных оценок качеств метрики, графики представлены на рисунке 1.

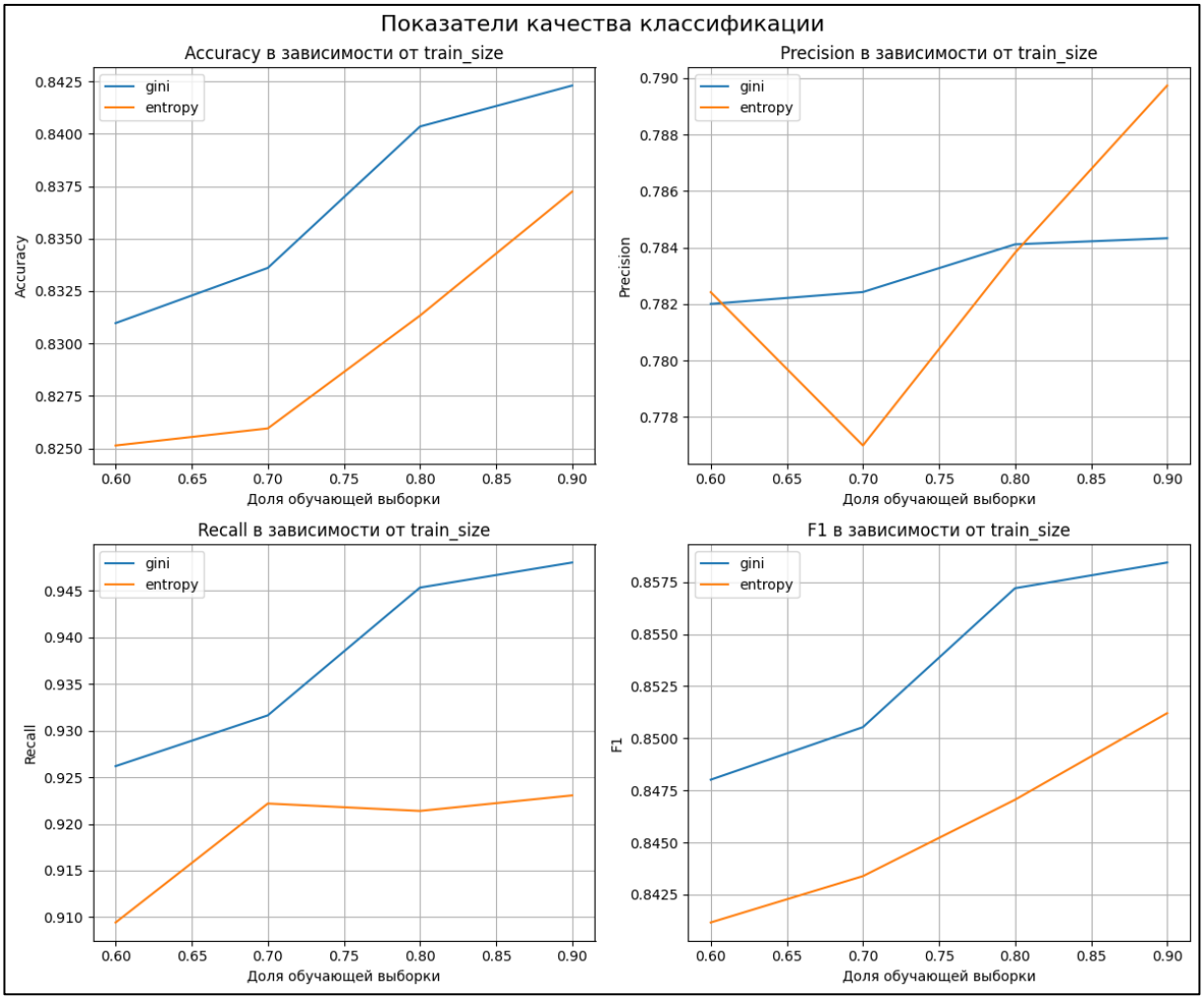

Рисунок 1 – Диаграмма зависимости показателей от доли обуч.выборки

На основе диаграммы можно сделать следующие выводы.

- Качество классификации возрастает с увеличением объема обучающей выборки.
- Gini Index и Entropy (Information Gain) показали схожие результаты, однако Gini давала немного более стабильные показатели на разных разбиениях.
- При увеличении train_size до 90% наблюдается эффект переобучения, что проявляется в снижении обобщающей способности модели на тестовой выборке.
- Балансировка классов улучшила качество классификации, увеличив Recall.

- Лучшее соотношение train-test в данном эксперименте – 80:20, обеспечивающее баланс между обобщающей способностью модели и точностью предсказаний.

Результаты экспериментов подтверждают, что дерево решений является мощным инструментом классификации, особенно при правильном выборе критерия разбиения и учете баланса классов. Таким образом, проведенные эксперименты подтвердили возможность успешного применения деревьев решений для классификации данных о доходах населения.