Санкт-Петербургский национальный исследовательский университет ИТМО

**https://lh6.googleusercontent.com/74sUYbhMTcYflKKxD3NS6NRK3vGq4LirVuF-1T8ARD_bAjUbbgvNzJmFVDKIOb-Zp9KNFG39cBqYh4pOXnZBS7epMxAFBxbFW4GZ-i7wKLX-QzOiZCJmmVBKGW54ox96VA3I5NrxvNb5J0i27zgtKyo**

Факультет программной инженерии и компьютерной техники

Направление подготовки 09.03.04 Программная инженерия

Дисциплина «Система искусственного интеллекта»

**Лабораторная работа №6**

**Деревья решений**

Студент

*Бобрусь Александр Владимирович., P33091*

Преподаватель

*Авдюшина Анна Евгеньевна*

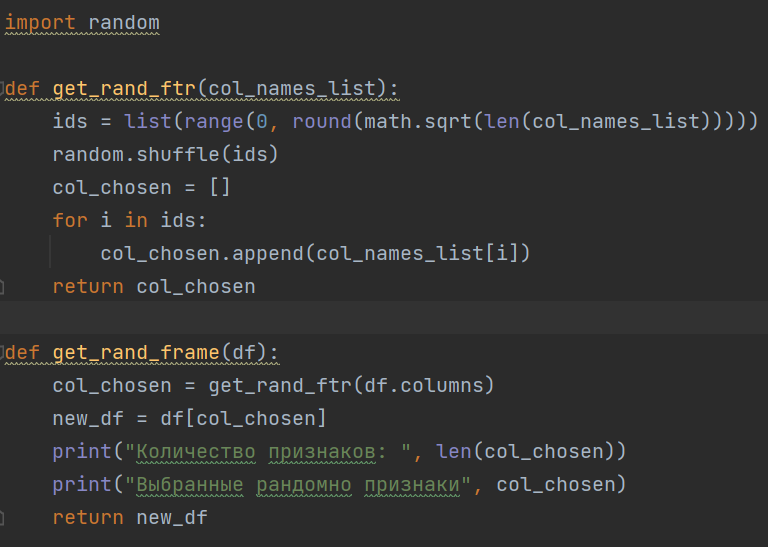
Санкт-Петербург, 2023 г.

**Задание**

1. Д[атасет с данными про оценки студентов инженерного и педагогического факультетов](https://archive.ics.uci.edu/dataset/856/higher+education+students+performance+evaluation) (для данного датасета нужно ввести метрику: студент успешный/неуспешный на основании грейда)
2. Отобрать случайным образом sqrt(n) признаков
3. Реализовать без использования сторонних библиотек построение дерева решений (numpy и pandas использовать можно, использовать списки для реализации дерева - нельзя)
4. Провести оценку реализованного алгоритма с использованием Accuracy, precision и recall
5. Построить AUC-ROC и AUC-PR (в пунктах 4 и 5 использовать библиотеки нельзя)

Лабораторная работа была реализована на языке Python.

**Случайный отбор sqrt(n) признаков**



**Реализация построения дерева решений**

Дерево решений (Decision Tree) в машинном обучении - это графическая модель, представляющая собой древовидную структуру, в которой каждый узел представляет собой решение по какому-либо признаку или атрибуту, а каждое ребро между узлами представляет собой возможный результат этого решения. В листьях дерева обычно находится прогноз или классификация.

Деревья решений используются в машинном обучении для решения задач классификации и регрессии. Они могут быть применены в различных областях, включая бизнес, медицину, финансы, науку о данных и многое другое. Вот некоторые из основных назначений деревьев решений:

1. Классификация (Classification): Деревья решений могут использоваться для классификации объектов на различные категории или классы. Например, классификация электронных писем на спам и не-спам, классификация изображений на категории или диагностика болезней на основе медицинских данных.

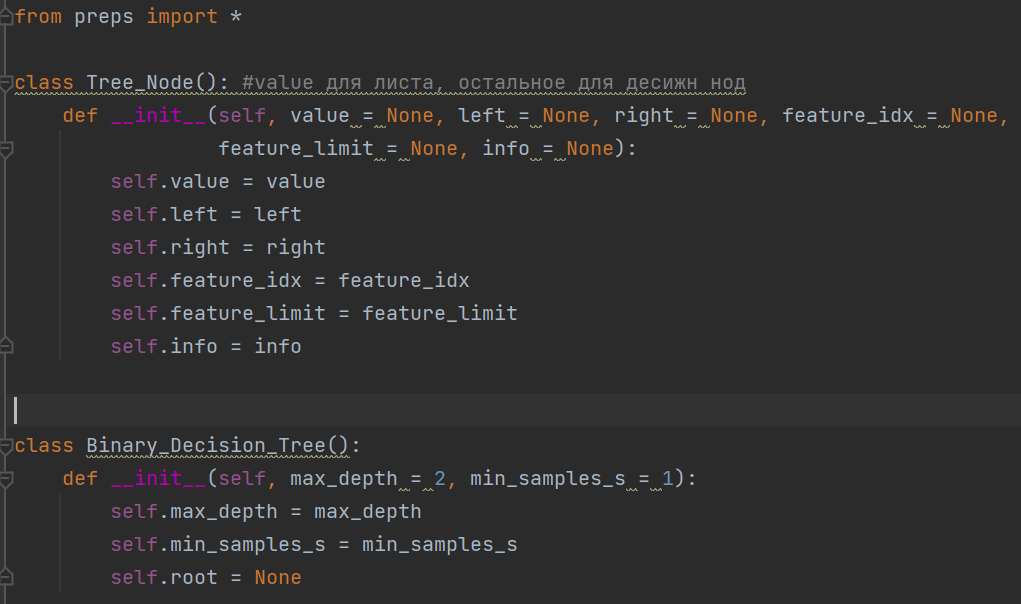
2. Регрессия (Regression): Деревья решений также могут быть применены для задач регрессии, где целью является предсказание числовых значений. Например, предсказание цены недвижимости на основе характеристик дома.

3. Принятие решений (Decision Making): Деревья решений могут быть использованы для поддержки принятия решений в бизнесе и других областях. Они могут помочь определить оптимальные стратегии или действия на основе входных данных.

4. Интерпретация и объяснение (Interpretability and Explanation): Деревья решений обладают высокой интерпретируемостью, что позволяет аналитикам и специалистам в области понимать, как модель принимает решения и какие признаки важны для прогнозирования.

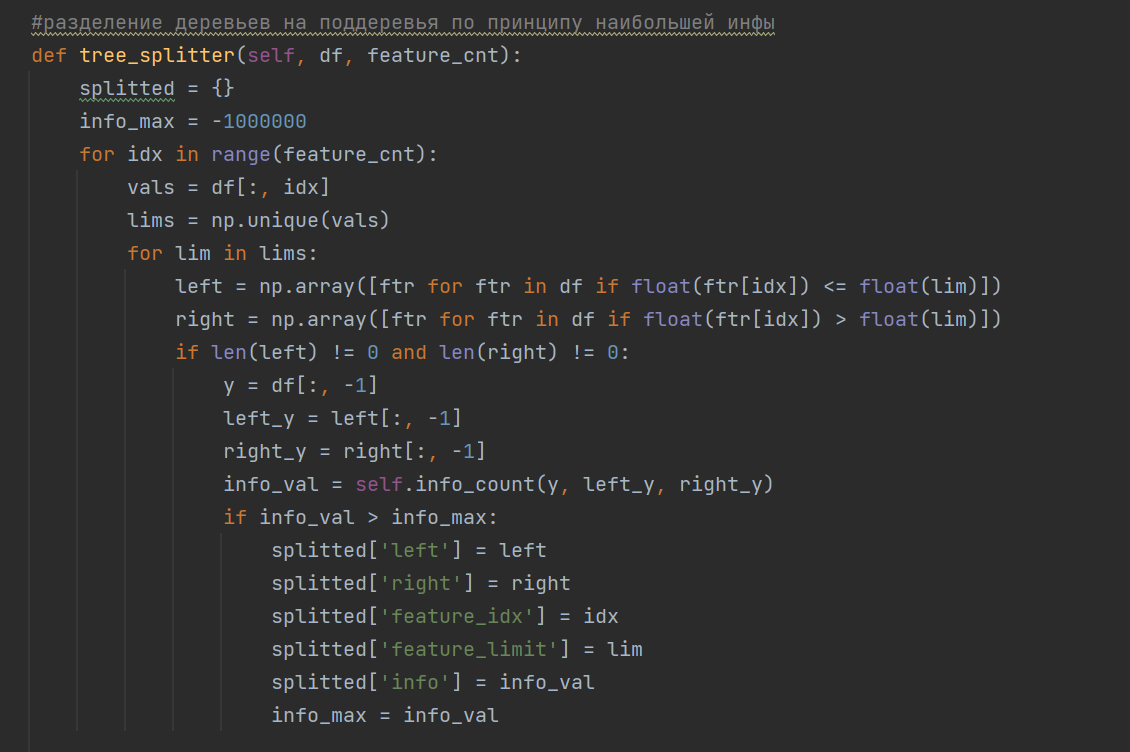
5. Обработка категориальных и числовых данных (Categorical and Numerical Data Handling): Деревья решений могут легко работать с категориальными и числовыми данными, что делает их удобными для анализа разнообразных типов информации.

Одним из основных преимуществ деревьев решений является их способность справляться с нелинейными и сложными зависимостями в данных. Кроме того, они могут быть применены как для задач классификации, так и для задач регрессии. Важно подбирать параметры и настраивать деревья решений, чтобы избежать переобучения и получить наилучшую производительность.

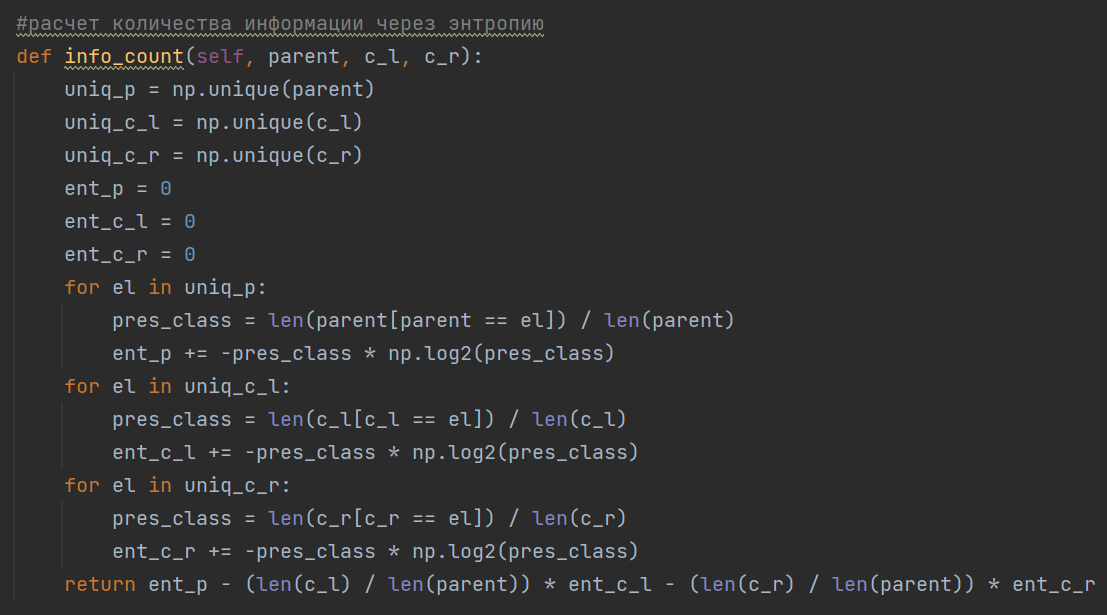


Tree\_Node: Это класс, представляющий узел дерева. Узел может быть листовым (в этом случае у него есть `value`) или узлом принятия решения (в этом случае есть `left` и `right` дочерние узлы, индекс признака `feature\_idx` и пороговое значение признака `feature\_limit`).

Binary\_Decision\_Tree: Это класс, представляющий само бинарное решающее дерево. Он имеет следующие методы:



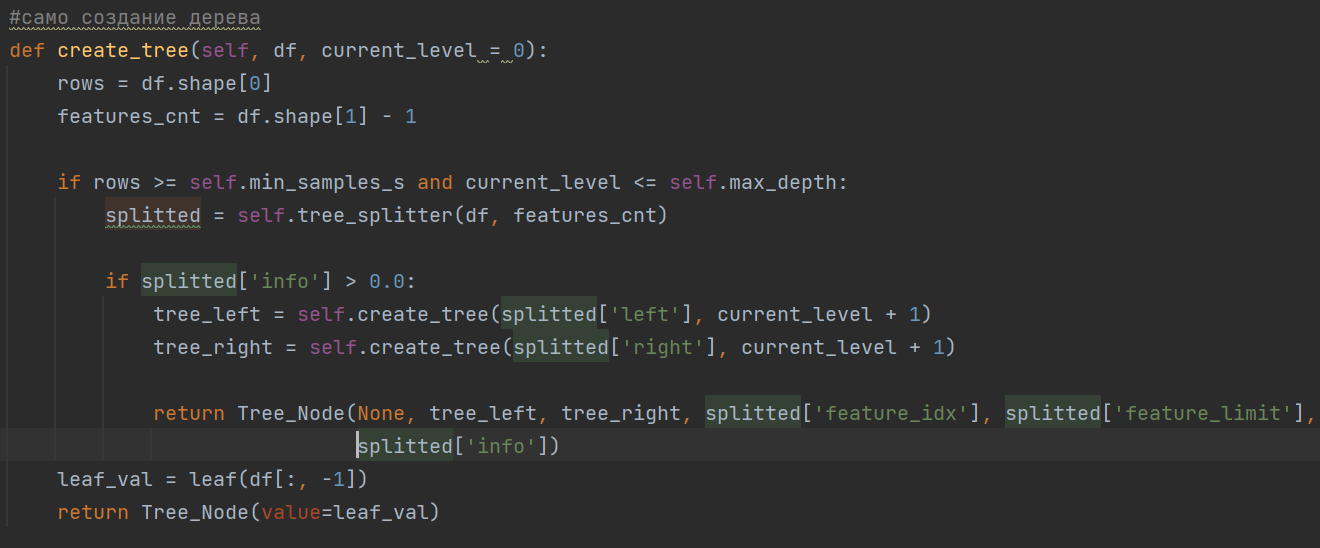
tree\_splitter: Метод, который выполняет разделение дерева на поддеревья, выбирая наилучшее разбиение с точки зрения информационного выигрыша.



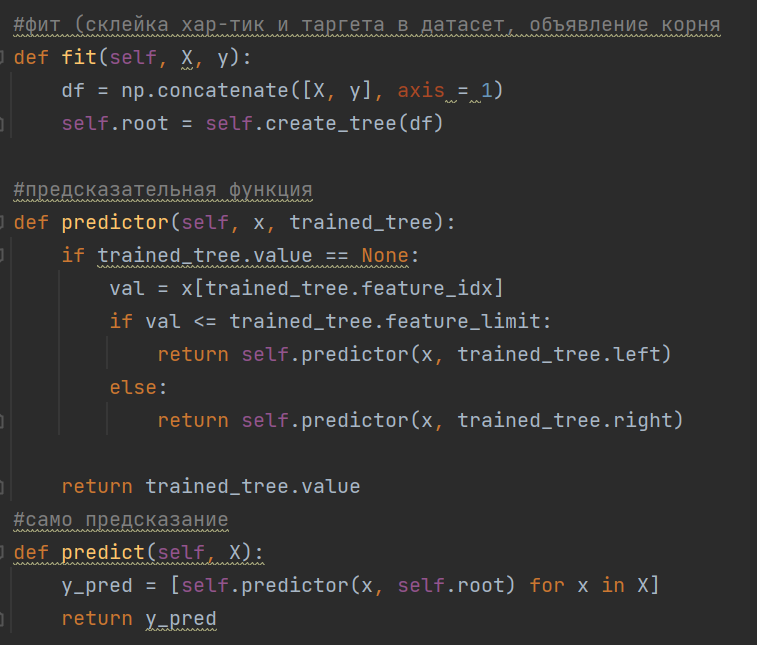
info\_count: Метод, который рассчитывает информационный выигрыш через энтропию.

Энтропия в машинном обучении - это мера неопределенности или разнообразия в данных. Она часто используется в контексте алгоритмов классификации, особенно в деревьях решений. Энтропия помогает определить, насколько хорошо определен класс данных в наборе обучающих примеров.

Идея заключается в том, что чем ниже энтропия, тем более чисто (однородно) разделены данные, и, следовательно, дерево решений может более уверенно классифицировать объекты.



create\_tree: Метод для создания дерева. Он рекурсивно разбивает данные, пока не достигнута максимальная глубина или не удовлетворено условие остановки. В результате создается дерево решений.

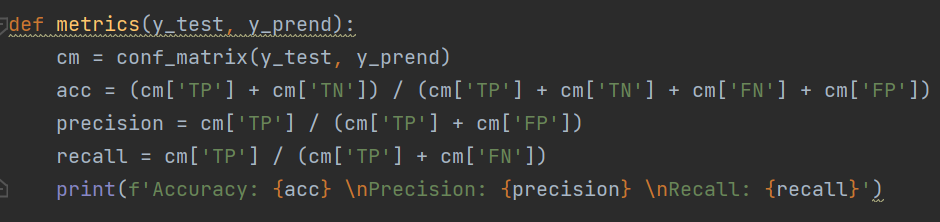
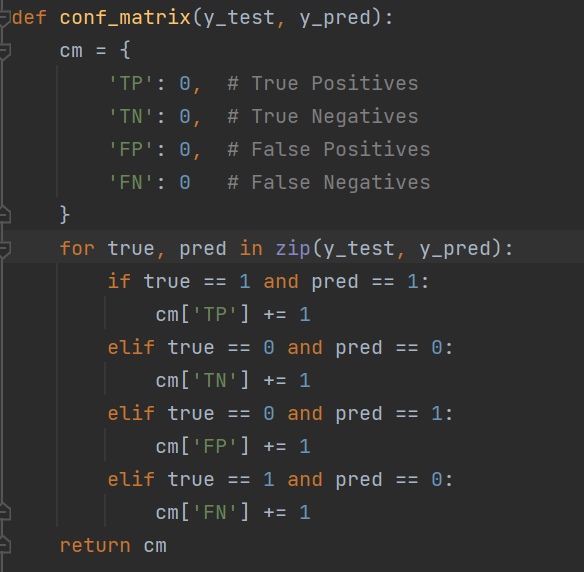


fit: Метод для обучения дерева на данных. Он объединяет признаки и целевую переменную в один датасет и вызывает `create\_tree` для построения дерева.

predictor: Вспомогательный метод для рекурсивного предсказания значения на основе обученного дерева.

predict: Метод для предсказания значений для входных данных `X` с использованием обученного дерева.

**Оценка реализованного алгоритма с использованием Accuracy, precision и recall**



Этот код содержит две функции для оценки производительности модели классификации с использованием матрицы ошибок (confusion matrix) и вычисления нескольких метрик:

1. conf\_matrix(y\_test, y\_pred): Эта функция вычисляет матрицу ошибок на основе истинных меток `y\_test` и предсказанных меток `y\_pred`. Матрица ошибок содержит четыре элемента: TP (True Positives), TN (True Negatives), FP (False Positives) и FN (False Negatives). Функция возвращает словарь с этими значениями.

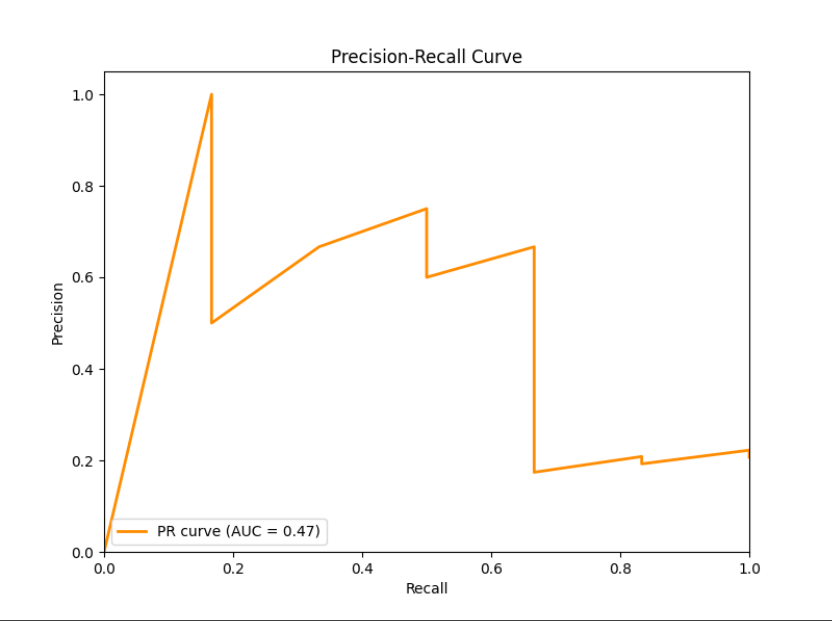
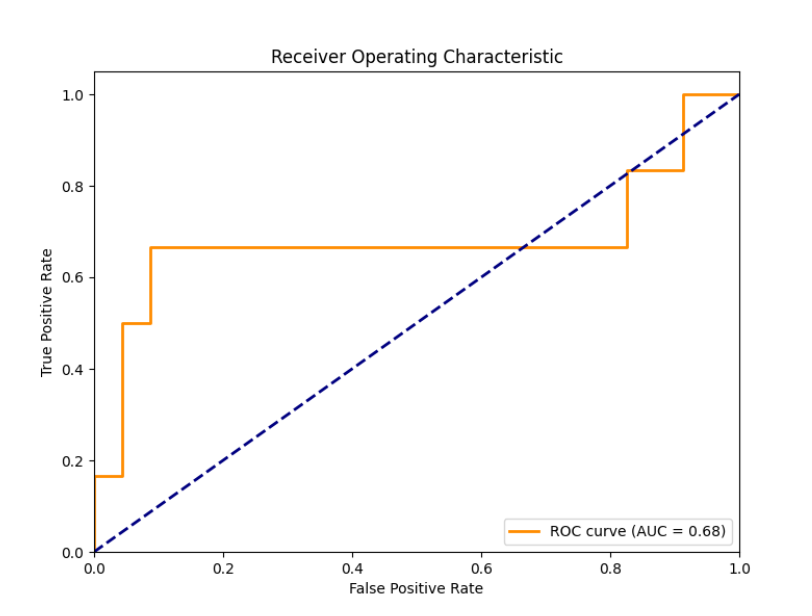
Матрица ошибок - это инструмент для оценки качества работы классификатора в задачах машинного обучения, особенно в задачах классификации бинарного типа (где есть два класса, например, "положительный" и "отрицательный").

Матрица ошибок часто используется для вычисления различных метрик качества классификации, таких как accuracy, precision, recall.

**Построение AUC-ROC и AUC-PR**

Графики кривых ROC и PR полезны для визуальной оценки производительности модели классификации, особенно при наличии дисбаланса классов. Кривая ROC помогает понять, как модель различает положительные и отрицательные классы, а кривая PR позволяет оценить точность и полноту модели при разных порогах. Вызов функции `draw\_plots` визуализирует обе кривые и выводит соответствующие значения AUC**.**

**Графики**



AUC (Area Under the Curve) - это метрика, используемая для оценки производительности моделей машинного обучения, особенно в задачах классификации. AUC измеряет площадь под кривой, которая обычно относится к кривой ROC (Receiver Operating Characteristic) или к кривой PR (Precision-Recall). Важно заметить, что AUC применяется к бинарным классификационным задачам, где у нас есть два класса: положительный (обычно обозначается как "1") и отрицательный (обычно обозначается как "0").

В контексте кривой ROC (ROC-кривой):

AUC-ROC и AUC-PR - это две метрики, используемые для оценки производительности моделей классификации, особенно в задачах бинарной классификации. Давайте разберемся, что это такое:

1. AUC-ROC (Area Under the Receiver Operating Characteristic Curve):

- ROC (Receiver Operating Characteristic) - это график, который представляет зависимость True Positive Rate (TPR) от False Positive Rate (FPR) при различных порогах классификации. TPR - это доля правильно классифицированных положительных примеров, а FPR - доля неправильно классифицированных отрицательных примеров.

***Кривая ROC помогает понять, как модель различает положительные и отрицательные классы***

- AUC-ROC - это площадь под ROC-кривой. Она измеряет общую способность модели различать между классами (положительными и отрицательными). Чем больше AUC-ROC, тем лучше модель способна делать правильные предсказания.

AUC-ROC равен 1, если модель всегда правильно классифицирует объекты, и 0.5, если модель классифицирует объекты случайным образом.

В контексте кривой ROC (Receiver Operating Characteristic), линия (с темно-синим цветом) является основной кривой ROC, которая отображает отношение True Positive Rate (TPR) к False Positive Rate (FPR) при различных порогах классификации. Она показывает, как изменяется TPR и FPR при изменении порога вероятности.

В контексте кривой PR (Precision-Recall):

- Precision-Recall (PR) Curve - это график, который представляет зависимость Precision от Recall при различных порогах классификации.

Precision - это доля правильно классифицированных положительных примеров среди всех положительных предсказаний, а Recall - доля правильно классифицированных положительных примеров среди всех реальных положительных примеров.

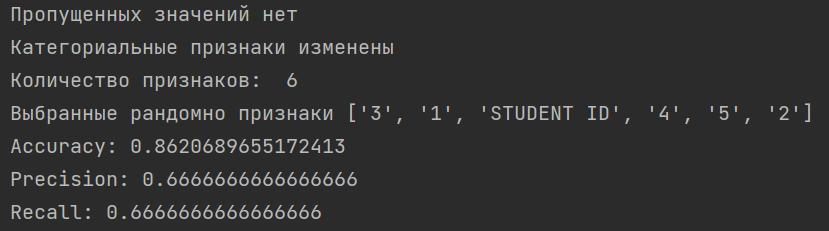
- AUC-PR - это площадь под PR-кривой. Она измеряет способность модели делать правильные положительные предсказания, учитывая точность и полноту.

AUC-PR может быть полезным, когда классы несбалансированы, и оценка ложных положительных предсказаний более важна.

- AUC-PR измеряет площадь под кривой PR и также служит метрикой производительности. В отличие от AUC-ROC, в случае кривой PR, большое значение AUC-PR указывает на лучшую производительность модели, если существует дисбаланс в классах.

Общий вывод: *AUC - это числовая метрика, которая помогает оценить, насколько хорошо модель различает классы в задачах бинарной классификации.* Высокое значение AUC свидетельствует о хорошей способности модели разделять положительные и отрицательные случаи при разных порогах вероятности.

**Вывод**



Исходя из предоставленных результатов, модель имеет следующие показатели:

- Accuracy: 0.8620689655172413

- Precision: 0.6666666666666666

- Recall: 0.6666666666666666

Анализ этих метрик позволяет сделать следующие выводы:

1. Accuracy в районе 0.86 означает, что ваша модель правильно классифицировала примерно 86% объектов. Значит, что модель правильно классифицировала всех успешных и неуспешных студентов. Это хороший результат, но следует помнить, что Accuracy может быть обманчивой метрикой в случае дисбаланса классов.

2. Precision в районе 0.67 указывает на то, что из объектов, которые ваша модель классифицировала как положительные, чуть больше половины действительно являются положительными. Precision = 0.67 означает, что из всех студентов, которых модель классифицировала как принадлежащих к "положительному классу" (например, студентам, которые успешно сдали экзамен или достигли определенного уровня успеха), только 67% действительно принадлежат к этому классу.

3. Recall в районе 0.67 показывает, какая доля фактических успешных студентов была правильна классифицирована моделью. То есть, если, например, есть 200 успешных студентов, и модель правильно классифицировала 134 из них как успешных, то Recall составляет 67%.