Следующее домашнее задание состоит из двух вариантов: базовый и продвинутый, студенту необходимо выбрать один вариант на выбор. Оба варианта оцениваются в формате «зачет/незачет». Выполненная работа должна быть представлена в формате .ipynb или ссылкой на ноутбук в Google Colaboratory. В начале ноутбука должен быть указан выбранный вариант. В ноутбуке под ячейкой должен быть результат выполнения, если ячейка должна что-то выводить.

Оформление:

1. В ноутбуке должны содержаться тематические разделы (заголовки первого или второго уровня) — например, по пунктам задания.
2. Код должен сопровождаться либо комментариями внутри ячеек, либо кратким текстом перед ячейкой. Пожалуйста, помните, что документировать код — это очень важно.

В качестве данных для работы используются результаты предыдущего домашнего задания (тема «Анализ данных»). **Проверьте, чтобы номера заказов были уникальными и не дублировались в разных строках.**

Требуемые колонки: store\_id, order\_price\*, profit, delivery\_distance, planned\_prep\_time, region\_id, items\_count, prepared\_on\_time. Если Выбрали продвинутый вариант, то добавить колонки day\_of\_the\_week, закодированные hour\_start, minutes\_start.

\***order\_price** – это сумма цен товаров, находящихся в каждом заказе.

Базовый вариант:

1. Загрузить предобработанные данные из самостоятельной работы №2.
2. Разделить данные на признаки и целевую переменную. Целевая переменная — «prepared\_on\_time».
3. Нормализовать данные (с помощью минимаксной нормализации или стандартизации).
4. Разделить данные на обучающую и тестовую выборки в отношении 80% на 20%.
5. Используйте метод GridSearchCV для поиска оптимальных гиперпараметров алгоритма KNN для классификации (KNeighborsClassifier): количество соседей k, метрика расстояния (metric), алгоритм вычисления ближайших соседей (algorithm). Для оценки качества используйте метрику accuracy.
6. Получите наилучшие гиперпараметры и заново обучите KNN с использованием этих параметров. Оцените качество классификации на тестовой выборке. Вывести метрики: accuracy, precision, recall, f1-score, classification\_report.
7. Сделать выводы по метрикам.
8. Визуализируйте зависимости точности модели KNN в зависимости от количества соседей: обучите несколько KNN-классификаторов с количеством соседей в диапазоне, который содержит оптимальное значение из подобранных ранее гиперпараметров (например, [k-5;k+5]) и получите оценки на кросс-валидации. Составьте график: по горизонтали - количество соседей, по вертикали - значение accuracy на кросс-валидации.

Продвинутый вариант:

1. Загрузить предобработанные данные из предыдущего домашнего задания.
2. Разделить данные на признаки и целевую переменную. Целевая переменная — «prepared\_on\_time».
3. Разделить данные на обучающую и тестовую выборки в отношении 70% на 30%.
4. Используйте метод GridSearchCV для поиска оптимальных гиперпараметров DecisionTreeClassifier (глубина дерева (max\_depth), минимальное количество объектов в листе (min\_samples\_leaf), критерий разделения (criterion)). Поскольку данные несбалансированны, используйте параметр class\_weight, чтобы задать веса классам (например, указать значение “balanced”).
5. Визуализируйте дерево и напишите, какие признаки были наиболее значимыми на основе параметра feature\_importances\_.
6. Получите наилучшие гиперпараметры и заново обучите DecisionTreeClassifier с использованием этих параметров. Оцените качество классификации на тестовой выборке. Вывести метрики: accuracy, precision, recall, f1-score, classification\_report.
7. Сделать выводы по метрикам.