Lab2. Описание работы

В данной лабораторной работе вам снова предстоит провести цикл анализа данных о прокате городских велосипедов в Сеуле. Напоминаю, что набор данных, который вы будете использовать, включает в себя записи о количестве аренды велосипедов (почасовые данные), а также информацию о погоде и праздниках.

Работа ориентирована на выполнение в среде Jupyter Notebook (Google Colab)

Цели:

1. **Глубокий анализ временных рядов**: агрегирование, поиск сезонных закономерностей, выявление выбросов.
2. **Обогащение и соединение данных**: интеграция дополнительных источников (праздники, погода), работа с пропусками.
3. **Аналитика и пользовательские метрики**: сегментация клиентов (или станций), анализ влияния погоды, простое прогнозирование.
4. **Автоматизация**: написание функций для формирования отчётов и сводных таблиц, визуализация зависимостей.
5. **Финальная часть**: составление отчёта с ключевыми выводами и создание простого интерактивного дашборда.

Цель работы

Отработать полный цикл анализа данных с использованием библиотеки Pandas, начиная от предварительной обработки и агрегирования временных рядов и заканчивая продвинутыми методами аналитики и визуализации, включая интерактивный дашборд.

Ниже перечислены этапы, которые необходимо выполнить в рамках этой лабораторной работы. Каждый этап включает в себя несколько пунктов. Вы можете оформлять решение в одном или нескольких ноутбуках, соблюдая логику выполнения задания.

1. Глубокий анализ временных рядов

1.1 Агрегация по периодам времени

- Сгруппируйте данные по разным периодам: **по дням**, **по неделям** и **по месяцам**.
- Рассчитайте суммарное и среднее число прокатов за каждый день, неделю и месяц.
- Проанализируйте полученные агрегированные ряды: как меняется прокат велосипедов от дня к дню, от недели к неделе, от месяца к месяцу?
- Постройте визуализации:
    - Линейный график для помесячного тренда (суммарные прокаты).
    - Столбчатая диаграмма для средних значений по дням недели и т.д.

1.2 Сезонные закономерности

- Исследуйте паттерны использования велосипедов в разные дни и время суток.
- Сравните будни и выходные:
    - Насколько сильно различается **среднее число прокатов** в будние дни по сравнению с выходными?
- Проанализируйте **суточный цикл** прокатов:
    - Когда наблюдаются пики (утром, в обед, вечером)?
    - Постройте график распределения прокатов в течение суток (по часам) отдельно для будней и для выходных.
- Оформите выводы:
    - Укажите, есть ли утренние и вечерние пики в будни (поездки на работу/с работы),
    - Как выглядит использование велосипедов по выходным (более равномерное, сдвиг пиков и т.п.).

1.3 Выбросы и аномалии

- Постройте временной график прокатов (например, **количество прокатов vs дата/время**).
- Выявите аномальные точки — дни или часы с необычно высоким или низким числом прокатов.
- Определите даты/часы таких выбросов, предложите гипотезы о причинах (праздники, городские мероприятия, погодные условия).
- Отразите результаты в отчёте, опишите, почему эти точки выглядят аномально.

---

2. Обогащение и соединение данных

2.1 Интеграция нескольких источников

- Объедините данные проката с дополнительной информацией. Например:
    - **Признак праздничного дня** (отдельная таблица с официальными праздниками в Сеуле/Южной Корее - https://english.visitkorea.or.kr/svc/contents/contentsView.do?vcontsId=14003)
    - **Погодные данные** (температура, осадки, влажность и т.д.).
- Используйте функции `merge` / `join` библиотеки Pandas. Объединение по дате и часу (или по дате, если нужен только дневной срез).
- Убедитесь, что в результирующем DataFrame присутствуют все необходимые столбцы для дальнейшего анализа.

2.2 Работа с пропущенными данными

- Проверьте датасет на наличие пропусков (`NaN`) после объединения.
- При обнаружении пропусков:
    - Выберите метод обработки: заполнение (средним, медианой, интерполяция) или удаление строк.
    - Обоснуйте свой выбор (например, если пропусков мало, их можно удалить; если много — целесообразнее заполнить).
- Убедитесь, что после обработки пропусков датасет не содержит `NaN` и готов к анализу.

---

3. Нестандартная аналитика и пользовательские метрики

3.1 Анализ поведения клиентов (сегментация)

- Если есть данные о пользователях, выполните сегментацию:
    - **Частые арендаторы**, **периодические**, **редкие** (критерий по количеству аренд в месяц или иной).
    - Для каждой группы изучите поведение (в какие часы суток/дни недели аренда чаще).
- Если данные о конкретных пользователях недоступны, можно провести анализ на уровне станций или районов:
    - Сегментируйте станции по интенсивности использования,
    - Исследуйте, какие станции используются больше утром, вечером, в выходные, и т.д.

3.2 Влияние внешних факторов

- Изучите, как погодные условия влияют на спрос на велосипеды.
- Рассмотрите переменные: температура, влажность, осадки.
- Проведите корреляционный анализ между этими показателями и количеством поездок.
- Постройте диаграмму рассеяния: «температура vs количество прокатов».
- Разбейте данные на категории (дни с дождем vs без дождя) и сравните среднее число прокатов.
- Посмотрите на диапазоны температуры (холодно/умеренно/жарко) и оцените, как меняется спрос.
- Сформулируйте выводы о том, какие факторы наиболее сильно влияют на число прокатов. Есть ли пороговые значения (например, при < 0°C спрос резко падает)?

3.3 Пользовательские метрики и прогнозирование

- Предложите собственные метрики, помогающие оценивать и прогнозировать спрос:
    - Например, **индекс комфортности** (учёт температуры и отсутствия осадков).
    - Постройте зависимость между этим индексом и количеством прокатов.
- Попробуйте сделать простую модель прогнозирования:
    - Линейная регрессия (температура, осадки, праздничный день и т.п. в качестве признаков).
    - Или простая формула вида «если температура > 20°C и без дождя, ожидается X прокатов».
- Интерпретируйте результаты (насколько метрики и прогнозы полезны, точны).

---

4. Автоматизация работы с данными

4.1 Функции для формирования отчётов

- Напишите несколько вспомогательных функций на Python (используя Pandas/Matplotlib и т.д.).
- Пример: `generate_report(data, period)`, где `period` может быть `'day'`, `'week'`, `'month'`. Функция должна:
    - Формировать сводку (общее число прокатов, среднее, максимум/минимум).
    - Строить график тренда.
- Другой пример: функция, принимающая условия (диапазон температур, факт дождя) и возвращающая среднее число прокатов.
- Продемонстрируйте работу функций на нескольких примерах (например, отчет по летним месяцам vs зимним).

4.2 Сводные таблицы и визуализация зависимостей

- Используя `pivot_table`, сформируйте сводные таблицы. Примеры:
    - Строки — **месяц**, столбцы — **день недели**, значения — **среднее число прокатов**.
    - Строки — **категории погоды** (ясно, дождь, снег), столбцы — **время дня** (утро/день/вечер), значения — **среднее число аренды**.
- Постройте на основе сводных таблиц подходящие визуализации (тепловые карты, столбчатые диаграммы и т.д.).
- Сконцентрируйтесь на 2-3 ключевых графиках, наиболее наглядно иллюстрирующих результаты анализа.

---

5. Финальная часть задания

5.1 Отчёт с выводами

Подготовьте краткий отчёт (1-2 страницы в формате Markdown внутри Jupyter Notebook или отдельный документ), в котором отразите:

- **Описание данных** (что в датасете, какие поля, откуда взяты).
- **Основные тенденции** (итоги агрегирования, общий уровень спроса, рост/падение).
- **Сезонность** (дни недели, месяцы, время суток).
- **Выбросы** (какие точки аномальны, возможные причины).
- **Влияние погоды** (температура, осадки, пороговые эффекты).
- **Сегментация** (если применима, по клиентам/станциям/частоте использования).
- **Выводы** (короткий обзор всех важных находок: кто/когда/почему чаще всего пользуется прокатом, какие факторы влияют сильнее всего).

6.2 Интерактивный дашборд

- Создайте интерактивный дашборд для наглядной презентации результатов. Минимальный функционал:
    - Один интерактивный график временного ряда (с возможностью выделять диапазон дат, наводить курсор для детализации).
    - Один интерактивный график, показывающий зависимость от фактора (например, влияние температуры или факта осадков).
- Средства реализации:
    - **Plotly Express** (графики интерактивны по умолчанию),
    - **Plotly Graph Objects** / **Dash**,
    - Либо виджеты `ipywidgets` в Jupyter Notebook (для простых фильтров, слайдеров).
- Коротко опишите, как пользоваться дашбордом (какие есть элементы управления, фильтры).
- Добавьте скриншот и/или продемонстрируйте интерактивность непосредственно в Jupyter Notebook.

Результат выполнения

1. **Jupyter Notebook** (или несколько ноутбуков) с пошаговым решением:
    - Код (с комментариями),
    - Графики и таблицы,
    - Выводы по каждому этапу.
2. **Итоговый отчёт** (можно в виде Markdown-ячеек в том же Notebook) на 1-2 страницы с обобщающими результатами.
3. **Интерактивный дашборд**, демонстрирующий ключевые метрики и дающий возможность самостоятельного исследования данных.

Дополнения и рекомендации

- Старайтесь писать структурированный код: делите тетрадь на смысловые блоки, используйте заголовки Markdown.
- Поясняйте каждый шаг анализа: зачем он нужен, что показывает, как интерпретировать результат.
- Используйте функции и avoid дублирования кода (DRY — *Don’t Repeat Yourself*).
- Регулярно сохраняйте результаты промежуточных вычислений (например, в файл CSV), чтобы не терять их при закрытии ноутбука.
- Если время позволяет, экспериментируйте с дополнительными метриками и видами визуализаций (boxplot, violin plot и т.д.).

Формат сдачи

- Загрузите (или представьте ссылку на) ваш Jupyter Notebook с проделанной работой.
- Убедитесь, что все ячейки выполнены, графики и выводы видны.
- Если используете дополнительные файлы (HTML-отчет, скриншоты, CSS-стили и т.д.), приложите их в репозиторий вместе с ноутбуком.