



ИНТЕЛЛЕКТУАЛЬНЫЙ
МЕГАПОЛИС

МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ



ИТ-класс

В МОСКОВСКОЙ ШКОЛЕ

НАПРАВЛЕНИЕ
БОЛЬШИЕ ДАННЫЕ И
ТЕХНОЛОГИИ ИСКУССТВЕННОГО
ИНТЕЛЛЕКТА

ПРАКТИЧЕСКИЙ ЭТАП

МОСКВА
2025





ИНТЕЛЛЕКТУАЛЬНЫЙ
МЕГАПОЛИС

МЕТОДИЧЕСКИЕ РЕКОМЕНДАЦИИ РАЗРАБОТАНЫ:

Евдокимова Екатерина Александровна, заместитель начальника
организационно-методического отдела УПРА РТУ МИРЭА

МОСКВА
2025

Методические рекомендации для учащихся
предпрофессиональных классов

для проведения практического этапа Московского конкурса межпредметных навыков и знаний «Интеллектуальный мегаполис. Потенциал» в номинации «ИТ-класс» по направлению

«Большие данные и технологии искусственного интеллекта»

Содержание

Рекомендации к выполнению конкурсных вариантов.....	3
Демонстрационный вариант конкурсных заданий практического этапа Конкурса	8
Список литературы для подготовки	15

Рекомендации к выполнению конкурсных вариантов

Материалы практического этапа Московского конкурса межпредметных навыков и знаний «Интеллектуальный мегаполис. Потенциал» (далее – Конкурс) предназначены для оценки уровня практической подготовки участников Конкурса.

Для успешного решения задач повышенного уровня участник должен обладать следующими навыками и знаниями:

1. Математические основы

Участник должен уверенно владеть базовыми математическими концепциями, широко применяемыми в анализе данных и технологиях искусственного интеллекта. В программу подготовки входят следующие разделы:

- *Комбинаторика*: Знание основных принципов комбинаторики, включая подсчёт числа вариантов, комбинаций и перестановок. Участник должен уметь решать задачи, связанные с выбором и упорядочиванием объектов.
- *Теория вероятностей*: Владение базовыми понятиями вероятности, знание дискретных и непрерывных распределений (например, биномиального и нормального). Участник должен уметь вычислять вероятности событий, использовать формулы полной вероятности и Байеса, а также применять вероятностные модели в практических ситуациях.

2. Визуализация данных

Участник должен знать основы визуализации данных и уметь строить графики для наглядного представления информации. Важно уметь **интерпретировать линейные графики, гистограммы и диаграммы**, а также выбирать подходящий тип визуализации для различных видов данных.

Участник должен уметь:

- Определять тип распределения по графику (гистограмме, плотности, функции вероятности).
 - Оценивать ключевые параметры распределения по внешнему виду.
 - Понимать отличия между разными типами распределений и их визуальными признаками.
- **Нормальное распределение (Gaussian distribution):**

Самое распространенное распределение, характеризуется колоколообразной симметричной кривой. Основные параметры – среднее значение (μ) и стандартное отклонение (σ). Примеры: распределение роста людей, оценок студентов в большом классе.

- **Биномиальное распределение (Binomial distribution):**

Описывает количество успехов в серии независимых испытаний с двумя возможными исходами (успех/неудача). Основные параметры – количество испытаний (n) и вероятность успеха (p). Примеры: число орлов при подбрасывании монеты n раз.

- **Пуассоновское распределение (Poisson distribution):**

Моделирует количество событий, происходящих за фиксированный интервал времени или в фиксированной области. Основной параметр – среднее количество событий (λ). Примеры: число звонков в службу поддержки за час.

- **Равномерное распределение (Uniform distribution):**

Все значения в определенном интервале имеют одинаковую вероятность. Примеры: случайное распределение чисел от 0 до 1, распределение выигрыша в простой лотерее.

- **Геометрическое распределение (Geometric distribution):**

Описывает количество испытаний до первого успеха в серии независимых испытаний с двумя возможными исходами. Основной параметр – вероятность успеха (p). Примеры: количество попыток при бросании монеты до первого орла.

3. Основы работы с SQL

Для успешного решения задач базового уровня, связанных с SQL, участник должен обладать следующими дополнительными навыками:

- Составлять SQL-запросы для выборки данных с учётом условий и ограничений.
- Выполнять группировку данных с использованием агрегатных функций.
- Применять HAVING для фильтрации по результатам агрегации (например, отбор групп с суммой выше заданного порога).
- Комбинировать WHERE и HAVING в одном запросе: первая фильтрует строки до агрегации, вторая — группы после неё.
- Формировать логически корректные запросы, содержащие сортировку, ограничение по количеству строк и использование конкретных колонок вместо SELECT *

Конструкция / Функция	Назначение
SELECT column1, column2	Извлечение данных из указанных столбцов
FROM table_name	Указание таблицы, из которой берутся данные
INNER JOIN table2 ON ...	Объединение таблиц по совпадающим значениям (только пересечение)
LEFT JOIN table2 ON ...	Сохранение всех строк из левой таблицы и совпадений из правой

RIGHT JOIN table2 ON ...	Сохранение всех строк из правой таблицы и совпадений из левой
WHERE condition	Фильтрация строк до агрегации по условиям
GROUP BY column	Группировка строк по указанным признакам
HAVING condition	Фильтрация после агрегации — отбор групп по агрегированным значениям
ORDER BY column [ASC, DESC]	Сортировка значений
LIMIT N	Ограничение количества возвращаемых строк
COUNT(*)	Подсчёт количества строк (всего или в группе)
SUM(column)	Сумма значений столбца
AVG(column)	Среднее значение
MIN(column) / MAX(column)	Минимум / максимум значений
AND, OR, NOT	Логические операторы в условиях

4. Работа с библиотеками Python

Для выполнения задач этого типа необходимо знать:

- Принципы хранения и обработки данных в структурах ndarray (NumPy), Series и DataFrame (Pandas).
- Различие между векторизованными и итеративными (циклическими) операциями и преимущества первых в вычислительной эффективности.
- Основные методы агрегации и группировки данных, включая среднее, сумму, подсчёт, медиану, а также работу с категориальными и временными признаками.
- Принципы импорта и экспорта данных из популярных форматов: CSV, Excel, JSON.
- Особенности представления и анализа временных данных (группировка по месяцам, извлечение даты и времени).
- Основы визуализации данных: типы графиков, их интерпретация и оформление в контексте аналитической задачи.

Уметь:

- Выполнять векторные операции с массивами и таблицами: арифметику, трансформации, фильтрацию.
- Загружать и сохранять данные с учётом кодировки, разделителей и структуры файлов.
- Преобразовывать данные времени в периоды, извлекать и использовать временные компоненты (месяц, год и др.).
- Группировать и агрегировать данные по нескольким признакам, формируя производные таблицы.

- Работать с пропущенными значениями: находить, удалять, заполнять, интерполировать.
- Строить графики различных типов и корректно их оформлять для последующей интерпретации результатов.
- Сохранять визуализации в графические файлы для отчётности.

Numpy	
Функция/Метод	Назначение
np.array()	Создание массива
array.shape	Получение размерности массива
array.reshape()	Изменение формы массива
np.mean(), np.sum()	Статистические функции
np.dot()	Матричное умножение
np.arange(), np.linspace()	Создание числовых последовательностей
Pandas	
Функция/Метод	Назначение
pd.read_csv()	Загрузка данных из CSV
df.to_csv()	Сохранение данных в CSV
df.head()	Просмотр первых строк
df['col']	Доступ к столбцу
df.loc[], df.iloc[]	Индексация по меткам и по позиции
df.groupby([...])	Группировка по признакам
df.agg({'col': 'mean'})	Агрегация значений
df.dropna(), fillna()	Работа с пропущенными значениями
pd.to_datetime()	Преобразование строк в даты
df['date'].dt.to_period('M')	Извлечение месяца/периода
df.sort_values()	Сортировка данных по столбцу
Mathplotlib	
Метод	Назначение
plt.hist()	Гистограмма
plt.plot()	Линейный график
plt.bar()	Столбчатая диаграмма
plt.scatter()	Диаграмма рассеяния
plt.boxplot()	Boxplot (ящик с усами)

<code>plt.xlabel()</code>	Подпись оси X
<code>plt.ylabel()</code>	Подпись оси Y
<code>plt.title()</code>	Заголовок графика
<code>plt.legend()</code>	Легенда
<code>plt.savefig()</code>	Сохранение графика в файл

Демонстрационный вариант конкурсных заданий практического этапа Конкурса

Задача 1.

Образовательная онлайн-платформа анализирует активность пользователей и популярность курсов для улучшения качества обучения и обновления контента. Вам, как аналитику, поручили подготовить выборки из базы PostgreSQL.

Структура базы данных:

- instructors (id, name, expertise) — преподаватели.
- courses (id, title, year, instructor_id, duration_hours) — курсы.
- students (id, name, email, registration_date) — студенты.
- course_enrollments (id, course_id, student_id, enrollment_date, completion_date) — записи о прохождении курсов.

Подготовьте SQL-запросы к базе данных для следующих целей:

1. Анализ активности преподавателей

Вывести имена преподавателей и общее количество часов всех их курсов, выпущенных после 2020 года. Отсортировать по убыванию суммарной продолжительности.

2. Идентификация вовлеченных студентов

Найти студентов, которые записывались на курсы более 3 раз за март 2023 года. Вывести имя, email и общее количество записей в этом месяце.

В качестве ответа необходимо предоставить два sql-запроса.

Запишите свое решение в файл: задача_1.txt.

Критерии оценивания:

Максимальная оценка за каждый запрос составляет 12 баллов. Общее количество баллов за задание – 24.

Корректный синтаксис		1
Запрос выполняется без ошибок в PostgreSQL	1	
Есть незначительные ошибки (например, пропущена запятая, неверный алиас), но запрос работает после исправления	0.5	
Синтаксические ошибки, из-за которых запрос не выполняется (например, неправильное использование ключевых слов, некорректные имена таблиц)	0	
Логика		7
Результат полностью соответствует условию (учтены фильтры, сортировка, группировка, исключены дубли)	7	
Частичные ошибки в логике (например, не учтена сортировка по убыванию, пропущен год в фильтре)	3	
Логика нарушена (например, вывод книг, изданных до 2015 года вместо после)	0	
Оптимальность и дополнительные требования		4
Использованы оптимальные методы (например, JOIN вместо подзапросов). Выполнены все дополнительные условия. Нет избыточных операций (например, SELECT * вместо конкретных колонок).	4	
Частично выполнено (например, NOT EXISTS заменен на LEFT JOIN, но результат верный)	2	
Грубые нарушения (например, запрос с CROSS JOIN, ведущим к декартову произведению)	0	
ИТОГО		12

Задача 2.

Вы — аналитик клиентского опыта в крупном интернет-магазине. Вам предоставили файл reviews_data.csv с историей отзывов клиентов за последние 12 месяцев.

Цель: оценить изменение клиентской удовлетворенности по продуктам и выявить группы с ухудшением восприятия.

Файл: reviews_data.csv, формат таблицы:

Колонка	Описание
product_id	Идентификатор продукта (целое число)

review_date	Дата отзыва (в формате YYYY-MM-DD)
rating	Оценка (целое число от 1 до 5)
sentiment	Категория тональности отзыва (positive/negative/neutral)
department	Отдел, к которому относится товар (строка)

Требуется:

1. **Загрузить данные из reviews_data.csv** — файл должен быть в том же проекте, что и программа.
2. **Преобразовать даты в формат месяца (YYYY-MM)** и сгруппировать данные по:
 - product_id
 - review_month (новый столбец, извлечённый из review_date)
 - Используйте pd.to_datetime(), .dt.to_period('M') или .dt.strftime('%Y-%m')
3. Для каждой группы рассчитать среднюю оценку (avg_rating) за месяц.
 - Используйте groupby(), agg() с mean
4. **Сформировать итоговую таблицу со столбцами (в указанном порядке):**
 - product_id
 - review_month
 - avg_rating
 - department
5. Отсортировать по review_month по возрастанию. Вывести первые 10 строк.
6. Построить **гистограмму распределения клиентских оценок** — rating
 - i. Ось X — rating
 - ii. Ось Y — количество отзывов, поставивших соответствующую оценку
 - Сохранить как rating_gist.png
 - Используйте matplotlib.pyplot (plot(), xticks(), legend(), hist() и т. д.)
 - Добавить подписи осей и заголовок графика.

Документацию по приведенным методам можно получить с помощью функции *help()*.

Критерии оценивания:

Максимальная количество баллов за задание – 24.

Загрузка и предобработка данных	4	
Данные загружены корректно Проведена проверка на пропуски (если требуется)	4	
Загрузка выполнена, но не учтены особенности данных (кодировка, разделители)	2	
Данные загружены с ошибками (например, неверный путь к файлу)	1	
Данные не загружены	0	
Расчет показателей	10	
Все метрики (динамика, средние, медианы и т.д.) рассчитаны верно Учтены краевые случаи (деление на ноль, отрицательные значения)	10	
Метрики есть, но есть ошибки в формулах или группировках	5	
Расчеты проведены, но не соответствуют условию задачи	1	
Показатели не рассчитаны	0	
Добавление новых столбцов	4	
Столбцы созданы через векторные операции (без циклов) Категоризация выполнена по условиям задачи	4	
Логика верна, но код не оптимален (например, использование apply)	2	
Столбцы добавлены, но с ошибками в условиях	1	
Новые столбцы отсутствуют	0	
Визуализация	5	
Построен график требуемого типа (heatmap, line, bar и т.д.). Наличие подписей осей, заголовка, легенды. График сохранен в файл	5	

График построен, но: Неверный тип визуализации, Отсутствуют подписи/легенда,	2	
Есть график, но не соответствует данным	1	
Визуализация отсутствует	0	
Оформление кода и вывод		1
Код читаем, есть комментарии (где необходимо). Результаты выведены в требуемом формате (например, print или to_markdown).	1	
Код не структурирован, вывод отсутствует	0	
ИТОГО		24

Задача 3.

Школьная столовая планирует меню на неделю (5 учебных дней). В распоряжении поваров — 8 разных горячих блюд. Каждый день подают **одно** горячее блюдо, и каждое блюдо можно использовать **не более одного раза в неделю**.

Но чтобы не было скучно, вводятся два условия:

- Блюдо понедельника всегда – овощное рагу.
- В течение недели **ровно одно** блюдо должно начинаться с буквы “**К**” (например, «Котлета»).

Вот список блюд, которые есть в распоряжении поваров:

1. Котлета по-домашнему
2. Каша гречневая с подливкой
3. Плов с курицей
4. Макароны по-флотски
5. Овощное рагу
6. Рыбные тефтели
7. Запеканка картофельная
8. Суп гороховый

Сколько различных вариантов меню можно составить на неделю?

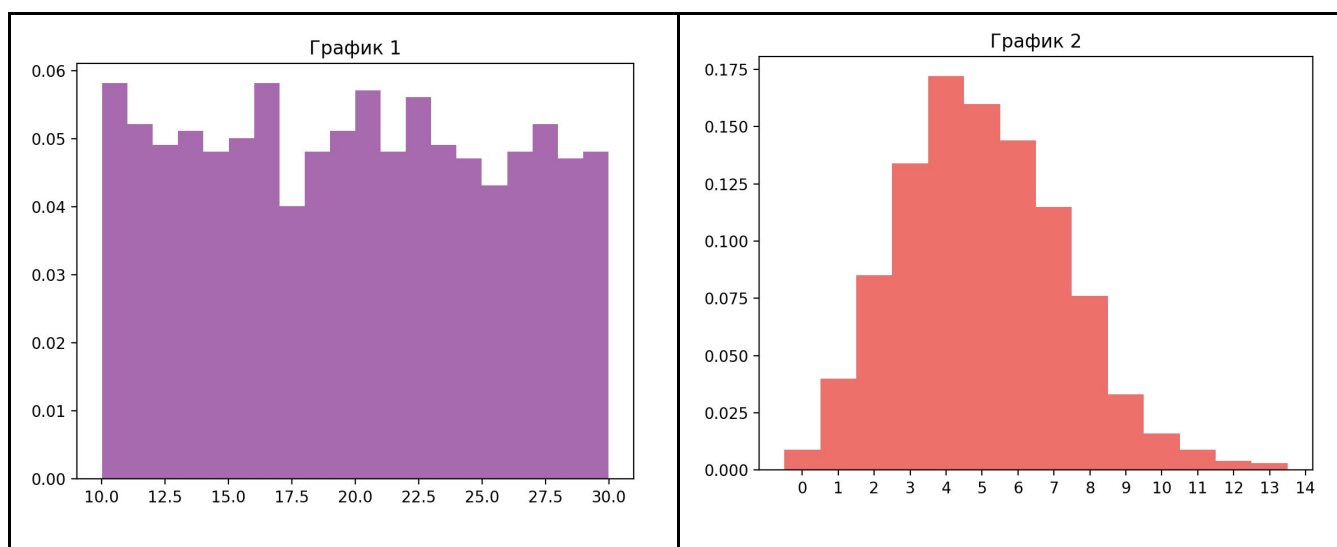
Запишите свое решение в файл: задача_2.txt. Ваше решение должно основываться на математических вычислениях - использование сторонних программ не допускается.

Критерии оценивания:

Максимальная количество баллов за задание – 6.

Логичность и отсутствие ошибок		3
Последовательность рассуждений, пояснение каждого шага	1	
Отсутствие арифметических, логических ошибок или противоречий	2	
Разделение на случаи (например, центральная цифра совпадает с первой/второй)		2
Выявление особенностей объектов, соответствующих условию (например, палиндромы, номера с повторяющимися цифрами)	1	
Правильное применение комбинаторных методов (правила умножения, сочетания, размещения)	1	
Предоставлен верный ответ		1
ИТОГО		6

Задача 4.



1. Определите тип распределения для каждого графика.
2. Оцените параметры распределений:
 - Для Графика 1: границы интервала (**a** и **b**).
 - Для Графика 2: параметр λ (лямбда).
3. Объясните, по каким визуальным признакам вы определили тип распределения.

Критерии оценивания:

Максимальная оценка за каждый график составляет 5 баллов. Общее количество баллов за задание – 10.

Определение типа распределения		1
Правильно указан тип распределения	1	
Тип распределения указан с неуверенностью или допущена несущественная ошибка (например, «похоже на нормальное» без точного названия)	0.5	
Тип распределения указан неверно или не указан вовсе	0	
Указание параметров распределения		2
Все параметры корректны и соответствуют графику	2	
Указан только один параметр правильно, или оба с частичной точностью	1	
Параметры не указаны или указаны с грубыми ошибками	0	
Обоснование выбора типа распределения		2
Обоснование подробное, логичное и основано на анализе формы графика (симметрия, пик, хвосты, размах и т.д.)	2	
Обоснование частично верное или не до конца раскрыто (общие фразы, без привязки к графику)	1	
Обоснование отсутствует или ошибочно	0	
ИТОГО		5

Список литературы для подготовки

1. Виленкин Н. Я., Виленкин А. Н., Виленкин П. А. Комбинаторика. – Наука, Главная редакция физико-математической литературы, 1969.
2. Райгородский А. Вероятность и алгебра в комбинаторике. – Litres, 2022.
3. Калмыков Р. К., Камалетдинова Э. В. Комбинаторные задачи в школьном курсе математики //Международный студенческий научный вестник. – 2014. – №. 4. – С. 9-9.
4. Тюрин Ю. Н. и др. Теория вероятностей и статистика //Экспериментальное учебное пособие для. – 2004. – Т. 10. – С. 2014-248.
5. Власов А. А. Статистические функции распределения. – 1966.
6. Савельев В. Статистика и котики. – Litres, 2018
7. Matplotlib: Visualization with Python // Matplotlib URL: https://matplotlib.org/stable/plot_types/index.html
8. "Поколение Python": курс для начинающих // Stepik URL: <https://stepik.org/course/58852/promo?search=5137972395>
9. NumPy // pythonworld URL: <https://pythonworld.ru/numpy>
10. Работаем с Pandas: основные понятия и реальные данные // URL: <https://skillbox.ru/media/code/rabotaem-s-pandas-osnovnye-ponyatiya-i-realnye-dannye/>
11. Интерактивный курс по SQL // URL: <https://sql-academy.org/ru/guide>
12. Практическое владение языком SQL // URL: <https://sql-ex.ru/?Lang=0>
13. Шпаргалка по SQL (postgres) // URL: <https://habr.com/ru/articles/745948/>