

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.....	8
Критерии оценки заданий.....	14
Пример решения задач.....	16
Список литературы для подготовки	22

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 базового уровня участник должен обладать следующими навыками и знаниями:

1. Математические основы

Участник должен хорошо понимать основные математические концепции, используемые в анализе данных и технологиях искусственного интеллекта. Это включает в себя следующие разделы:

Комбинаторика: Понимание принципов комбинаторики, умение решать задачи, связанные с подсчетом вариантов, комбинаций и перестановок.

Теория вероятностей: Базовое знание вероятностных распределений, включая биномиальное и нормальное распределения. Участник должен уметь рассчитывать вероятности различных событий и понимать их применение в анализе данных.

2. Визуализация данных

Знание основ визуализации данных и умение строить графики для представления информации. Умение интерпретировать линейные графики, гистограммы и диаграммы. Понимание, какие типы графиков подходят для различных видов данных. Знание и понимание того, как распределения данных отображаются на графиках, например, понимание форм нормального распределения и его отличий от других типов распределений. Для успешного выполнения задач, связанных с анализом данных, участник должен понимать различные типы распределений, которые встречаются в статистике и применяются при обработке данных. Вот некоторые из наиболее распространенных типов распределений, знание которых важно:

- **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.

- **Экспоненциальное распределение (Exponential distribution):**

Описывает время между двумя последовательными событиями в процессе с постоянной интенсивностью (например, в процессе Пуассона). Основной параметр – интенсивность (λ). Примеры: время ожидания автобуса, время жизни радиоактивного атома.

- **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.

- **Распределение Стьюдента (Student's t-distribution)**

Это распределение, которое используется в статистике, особенно при работе с небольшими выборками. Оно похоже на нормальное распределение, но имеет более «тяжелые» хвосты, что делает его полезным для оценки среднего значения в случае, когда размер выборки мал и стандартное отклонение неизвестно.

3. Программирование на Python

Базовые навыки программирования на языке Python, которые необходимы для решения задач. Участник должен понимать различные типы данных в Python, таких как строки, числа, списки, словари, и уметь эффективно ими пользоваться. Знать синтаксис Python, включая операции с числами и строками, работу со списками и словарями, выполнение циклов и условных операторов. Уметь работать с основными библиотеками Python, такими как `math` для выполнения математических операций, `random` для генерации случайных чисел. Участник должен обладать навыками открытия, чтения и записи текстовых файлов, использование методов `.read()`, `.write()` и работа с контекстными менеджерами (например, `with open(...) as file`). Уметь считывать пользовательский ввод с помощью функции `input()` и выводить данные на экран с помощью `print()`.

4. Основы работы с SQL

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

- Умение формировать SELECT-запросы для извлечения данных из одной таблицы, используя фильтрацию с помощью WHERE.
- Умение использовать команду ORDER BY для сортировки данных в запросах по одному или нескольким столбцам, как по возрастанию (ASC), так и по убыванию (DESC).
- Навык использования команды LIMIT для ограничения числа возвращаемых строк. Это особенно полезно для получения первых нескольких записей из отсортированного результата.
- Умение применять функции MIN() и MAX() для получения минимального и максимального значения из набора данных.
- Знание функции AVG() для расчета среднего значения по столбцу.
- Применение функции SUM() для вычисления общей суммы по столбцу.
- Умение использовать функцию COUNT() для подсчета числа строк, соответствующих заданным условиям.
- Использование условий в запросах с помощью команды WHERE, чтобы извлекать только те данные, которые удовлетворяют определенным критериям. Также навык комбинирования условий с помощью операторов AND, OR, NOT.
- Знание команды GROUP BY для группировки строк, которые имеют одинаковые значения в определенных столбцах, а также умение использовать агрегатные функции в сочетании с группировкой.

Для выполнения задач повышенного уровня участникам требуется более глубокое понимание и навыки работы в различных областях. Ниже представлены ключевые области знаний и навыков:

1. Работа с библиотеками Python

NumPy:

Понимание основ работы с многомерными массивами (ndarray), выполнение операций над массивами (арифметические операции, матричные умножения).

Умение использовать функции NumPy для выполнения вычислений без явных циклов, что позволяет ускорить обработку данных.

Pandas:

Глубокое понимание структуры и методов работы с DataFrame и Series. Умение выполнять операции фильтрации, группировки и агрегации данных.

Знание методов заполнения пропущенных значений, их удаления или интерполяции.

Чтение и запись данных в форматах CSV, Excel, JSON. Умение эффективно работать с большими объемами данных.

Навыки создания разнообразных графиков (линейные, столбчатые, круговые диаграммы и д.р.) и настройка их внешнего вида.

Способность анализировать и интерпретировать визуализированные данные, выявлять тенденции и аномалии.

2. Машинное обучение

В данном типе задания от участника требуется:

Понимание разницы между задачами регрессии и классификации, а также методов обучения с учителем и без учителя.

Знание и умение применять основные алгоритмы машинного обучения:

- Линейная регрессия, полиномиальная регрессия.
- Логистическая регрессия, деревья решений, случайные леса, метод опорных векторов (SVM).

Оценка моделей с помощью метрик точности, полноты, F1-меры, ROC-кривой и AUC.

Умение выбирать подходящие методы для решения конкретной задачи и объяснять выбор на основе особенностей данных и цели анализа.

Определение и исправление ошибок в данных, правильное распределение данных на тренировочные и тестовые наборы.

Умение находить и исправлять ошибки в работе моделей, такие как переобучение, недообучение, неправильно выбранные гиперпараметры.

3. Нормализация данных и проектирование баз данных

При решении данной типа задач от участника требуется:

- Понимание первой, второй и третьей нормальных форм (1NF, 2NF, 3NF), их применение для предотвращения избыточности данных и обеспечения целостности базы данных.
- Умение нормализовать таблицы, выявлять и устранять избыточные данные, обеспечивать эффективное хранение и доступ к информации.

- Разработка схемы базы данных, определение таблиц, связей между ними и ключевых полей.
- Формирование сложных запросов для извлечения и обработки данных, использование JOIN, UNION, подзапросов и оконных функций для работы с несколькими таблицами.

Демонстрационный вариант конкурсных заданий практического этапа Конкурса

Ответы к задачам, записанные в файлы, необходимо сохранять в папку

Потенциал_ИТ_ФамилияИО

Задача 1.

В магазине 40% товаров - электроника, 30% - одежда, 30% - продукты.
5% электроники, 10% одежды и 1% продуктов имеют дефекты.
Какова вероятность выбрать дефектный товар?

В ответ запишите развернутое решение. Для обоснования воспользуйтесь символами: $P(A)$ для обозначения вероятности.

Запишите свое решение в файл: задача_1.txt

Задача 2.

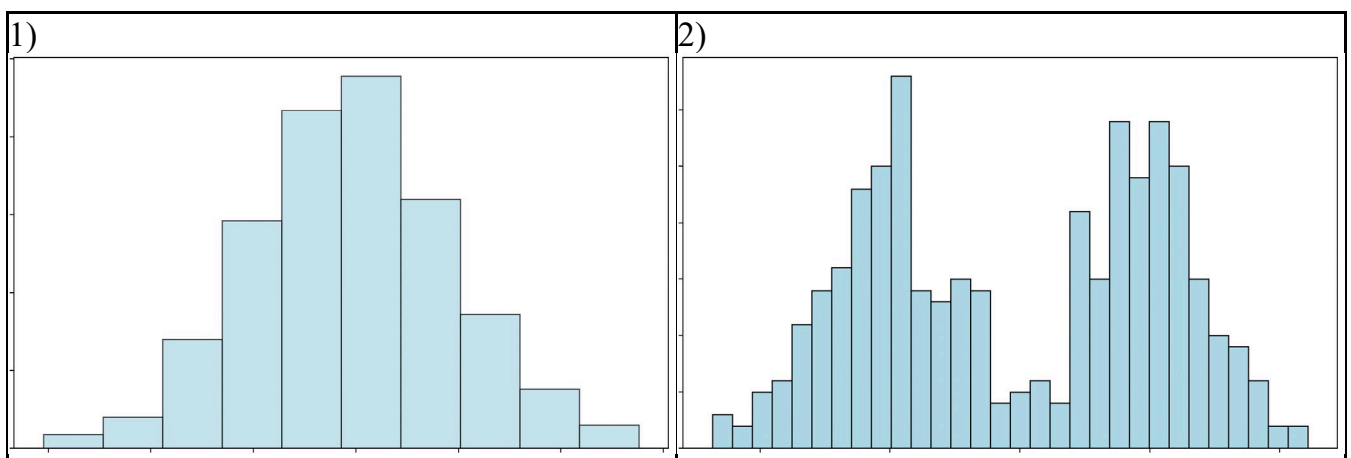
Перед вами четыре гистограммы, каждая из которых отображает распределение данных из одной из четырех выборок, описанных ниже. На гистограммах:

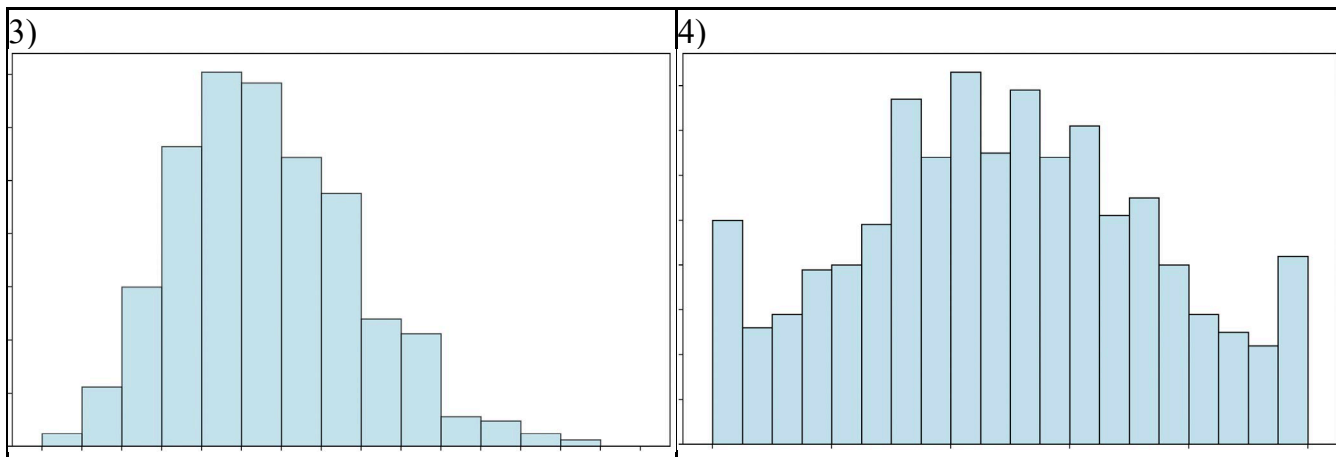
- **Ось абсцисс** показывает интервалы значений.
- **Ось ординат** отражает частоту.

Ваша задача — сравнить эти гистограммы с описаниями выборок и определить, какая из них соответствует каждому графику.

Свой ответ обоснуйте.

Запишите свое решение в файл: задача_2.txt





А) Выборка оценок студентов по 100-бальной системе, полученных на экзамене по физике в течение одного учебного года. Данная выборка включает в себя небольшое количество студентов.

Б) Выборка количества очков, набранных спортсменами в серии из десяти игр в школьной лиге. Максимальное количество очков, набранных одним спортсменом, составляет 15.

В) Выборка показателей веса учеников старших классов в общеобразовательной школе, измеренных в начале учебного года.

Г) Выборка результатов опроса о предпочтениях в еде среди учеников в школе. Участникам было предложено выбрать из 10 категорий продуктов.

Задача 3.

Ввод: стандартный ввод

Вывод: стандартный вывод

Напишите программу, которая считает количество вхождений первого символа в введенную строку. (Гарантируется, что в строке будет хоть один символ)

Формат вывода: В строке <введенная строка> найдено <количество (тип int)> символов.

Запишите свое решение в файл: *task_3.py*

Задача 4.

В базе данных есть таблица `students`, содержащая следующие поля: `id`, `name`, `age`, `grade`. Напишите SQL-запрос, который выберет имена всех студентов, которым 16 лет.

Запишите свое решение в файл: *задача_4.txt*

Задача 5.

Вам предоставлен набор данных (`library_books.csv`), содержащий информацию о книгах в библиотеке. Этот набор данных включает следующие столбцы:

- Название книги
- Автор
- Год издания
- Жанр
- Количество выданных за последний год
- Общее количество экземпляров

Используя библиотеку `pandas`, проведите анализ данных и предоставьте отчет по следующим критериям:

1. Определите топ-3 самых популярных книг (по количеству выданных за последний год).
2. Постройте один график, показывающий среднее количество выданных экземпляров для каждого жанра.
 - Убедитесь, что график имеет название и подписанные оси.

Формат вывода:

1)

Топ-3 популярных книг:

1. "Название книги 1" (Автор 1) - количество выданных экземпляров: X
2. "Название книги 2" (Автор 2) - количество выданных экземпляров: Y
3. ...

Решение задачи и полученные файлы сохраните в папке `задача_5`.

Содержимое папки оформить следующим образом:

Пункт задачи	Описание файла	Кодировка файла	Название файла
1	В файл необходимо поместить вывод программы в указанном формате	UTF-8	<code>задача_5_1.txt</code>
2	График, полученный в ходе решения задачи		<code>задача_5_2.png</code> (<code>задача_5_2.jpg</code>)
Общий файл с программой	Полный код программы с комментариями	UTF-8	<code>task_5.py</code>

Задача 6.

В этом коде используется алгоритм K-ближайших соседей для классификации данных из датасета Iris. Однако в процессе предсказания допущена ошибка: найдите и исправьте ее. В качестве ответа запишите исправленный код. Аргументируйте свой ответ.

```
from sklearn.datasets import load_iris
from sklearn.model_selection import train_test_split
from sklearn.neighbors import KNeighborsClassifier

# Загружаем данные
iris = load_iris()
X, y = iris.data, iris.target

# Разделяем данные
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2)

# Обучаем модель
knn = KNeighborsClassifier(n_neighbors=3)
knn.fit(X_train, y_train)

# Предсказания
y_pred = knn.predict(X_train)
print("Предсказания:", y_pred)
```

Код (решение) запишете в файл task_6.py или задача_6.txt.

Задача 7.

Вы работаете в компании, занимающейся продажей книг. Вам необходимо спроектировать базу данных для хранения информации о книгах, авторах и заказах. Ваша задача — создать схему базы данных, которая будет удобной для хранения и извлечения данных.

Условия задачи:

Книги:

- Каждая книга имеет уникальный идентификатор, название, год издания и цену.
- У каждой книги есть один автор.

Авторы:

- Каждый автор имеет уникальный идентификатор, имя и фамилию.
- Один автор может написать несколько книг.

Заказы:

- Каждый заказ имеет уникальный идентификатор, дату заказа и сумму.
- Один заказ может включать несколько книг, но каждая книга в заказе может быть только одна.

Задания:

1. Нормализация данных:

- Определите таблицы, которые необходимо создать для хранения информации о книгах, авторах и заказах (опишите текстом структуру БД).
- Опишите, как вы будете устранять избыточность данных.

2. Создание структуры базы данных:

- Напишите SQL-код для создания таблиц в PostgreSQL с использованием подходящих типов данных, первичных и внешних ключей.
- Обеспечьте поддержку целостности данных с помощью ограничений.

3. Запросы:

- Напишите SQL-запрос для получения списка всех книг с указанием их авторов.

Примеры возможного заполнения таблиц:

Таблица Авторы:

AuthorID	FirstName	LastName
1	George	Orwell
2	J.K.	Rowling
3	F. Scott	Fitzgerald

Таблица Книги:

BookID	Title	YearPublished	Price	AuthorID
1	1984	1949	9.99	1
2	Animal Farm	1945	7.99	1
3	Harry Potter and the Sorcerer's Stone	1997	14.99	2
4	Harry Potter and the Chamber of Secrets	1998	15.99	2
5	The Great Gatsby	1925	10.99	3

Таблица Заказы:

OrderID	OrderDate	TotalAmount
1	2024-08-01	25.98
2	2024-08-03	15.99
3	2024-08-05	7.99

Решение задачи и полученные файлы сохраните в папке задача_7
 Файлы в папке оформите следующим образом:

Пункт задачи	Что должно быть в файле	Кодировка файла	Название файла
Нормализация данных	В файле опишите структуру БД	UTF-8	задача_7_1.txt
Создание структуры БД	В файл добавьте SQL запрос для создания таблиц	UTF-8	задача_7_2.txt
Запросы	Добавьте SQL-запрос	UTF-8	задача_7_3.txt

Критерии оценки заданий

Задача 1.

Баллы	Критерии оценки
0	Дан неверный ответ, не предоставлено никакого объяснения.
2	Дан неправильный ответ, но представлены некоторые шаги решения.
4	Дан правильный ответ, но объяснение недостаточно ясно.
6	Дан правильный ответ и представлено полное и логичное решение.

Задача 2.

Баллы	Критерии оценки
0	Не сопоставлен ни один график с описанием
2	Правильно сопоставлен 1 график, но не связаны остальные
4	Правильно сопоставлены 2 графика
6	Правильно сопоставлено 3 графика
8	Правильно сопоставлены все графики

Задача 3.

Баллы	Критерии оценки
0	Код отсутствует или содержит критические ошибки, мешающие его выполнению.
	Не представлены правильные решения задачи.
	Отсутствие попытки анализа задачи.
3	Код работает, но содержит незначительные ошибки или недочеты.
	Решение задачи выполнено, но требует доработки.
	Основные операции в коде присутствуют, но неправильно использованы.
	Часть логики реализована, но не завершена.
6	Код работает без ошибок и оптимально решает задачу.
	Решение задачи логично и четко структурировано.
	Использованы правильные конструкции и подходы, код легко читаем.
	В случае необходимости, предоставлены комментарии для улучшения понимания.

Задача 4.

Баллы	Критерии оценки
0	Дан неверный ответ, не предоставлено работающего SQL-запроса.
3	Дан правильный SQL-запрос, но он может содержать незначительные ошибки или быть не полностью оптимизирован.
	Запрос выполняет задачу, но требует улучшений в плане производительности или правильности вывода.
6	Дан полностью правильный и оптимизированный SQL-запрос.
	Запрос выполняет задачу корректно и эффективно, без ошибок и с оптимальной производительностью.

Задача 5.

Критерий	Описание	Баллы
----------	----------	-------

Анализ топ-3 самых популярных книг	Определены и правильно указаны топ-3 книги по количеству выданных за последний год.	4
График	Построен график, показывающий среднее количество выданных экземпляров для каждого жанра. График должен быть четким и информативным.	3
Чистота и структурированность кода	Код написан аккуратно, с комментариями и правильной структурой, что облегчает понимание и повторное использование.	3
Всего баллов		10

Задача 6.

Баллы	Критерии оценки
0	Код отсутствует или содержит критические ошибки, мешающие его выполнению.
	Не представлены правильные решения задачи.
	Отсутствие попытки анализа ошибки или объяснений к исправлениям.
5	Код исправлен, но остаются незначительные ошибки или недочеты.
	Исправление кода представлено, но отсутствует обоснование выбора решения.
9	Код написан правильно и оптимально, решает задачу без ошибок.
	Ответы полные и содержат детальное объяснение всех исправлений, демонстрируя глубокое понимание алгоритма.

Задача 7.

Пункт задачи	Критерии оценки	Баллы
1. Нормализация данных		6
1.1 Определение таблиц и атрибутов	Определены все необходимые таблицы и атрибуты, соблюдены принципы 3NF	3
1.2 Устранение избыточности и зависимостей	Корректно устранены избыточности, правильно установлены связи и ключи	3
2. Создание структуры базы данных		7
2.1 Создание таблиц с правильными типами данных	Использованы корректные типы данных для всех атрибутов	3
2.2 Создание первичных и внешних ключей	Корректно определены первичные и внешние ключи	2
2.3 Обеспечение целостности данных	Применены ограничения (NOT NULL, UNIQUE, CHECK и т.д.)	2
3. Примеры запросов		2
3.1 Список книг с указанием авторов	Запрос корректно возвращает все книги с их авторами	2
Всего баллов за задачу		15

Приложение 4

Пример решения задач

Номер задачи	Решение	Ответ
1	$P(D) = 0.05 \cdot 0.4 + 0.1 \cdot 0.3 + 0.01 \cdot 0.3 = 0.02 + 0.03 + 0.003 = 0.053.$	Б
2	<p>А) Поскольку данная выборка содержит ограниченное количество данных, её поведение может быть охарактеризовано распределением Стьюдента, что предполагает наличие колебаний. В то же время, значения оценок будут стремиться к нормальному распределению. Поэтому лучше всего данную выборку описывает гистограмма 4.</p> <p>Б) Среднее количество очков, набранное спортсменами, подчиняется распределению Пуассона, что означает смещение в сторону среднего значения. Подходящими под это описание являются варианты 1 и 3. Однако, учитывая, что максимальное количество баллов составляет 15, вариант 1 не подходит, так как в нем используется 10 значений. Поэтому выбор падает на гистограмму 3.</p> <p>В) В общеобразовательной школе обучаются как девочки, так и мальчики, и их веса отличаются друг от друга. При равномерной выборке любой вес будет иметь нормальное распределение. Поскольку у нас имеются две выборки – для мальчиков и девочек – на графике мы должны увидеть сумму этих двух нормальных распределений, что проявится в форме двух отчётливых холмов. Таким образом, нам подходит гистограмма 2.</p> <p>Г) В данном случае остаточный принцип применим только к варианту 1. Кроме того, предпочтения в еде среди школьников схожи, что позволяет ожидать нормальное распределение на 10 позициях. Это именно то, что представлено на первой гистограмме.</p>	<p>А - 4</p> <p>Б - 3</p> <p>В - 2</p> <p>Г - 1</p>

3	<pre># Вводим строку input_string = input() # Первый символ строки first_char = input_string[0] # Считаем количество вхождений первого символа в строку count = input_string.count(first_char) # Выводим результат print(f"В строке {input_string} найдено {count} первых символов {first_char}.")</pre>	
4	SELECT name FROM students WHERE age = 16;	

Задача 5.

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

# Загрузка данных
df = pd.read_csv('library_books.csv')

# Предварительный просмотр данных
print(df.head())

top_3_books = df.nlargest(3, 'Количество выданных за последний год')

print("Топ-3 популярных книг:")
i = 1
for idx, row in top_3_books.iterrows():
    print(f"{i}. \"{row['Название книги']}\", {row['Автор']} - количество выданных экземпляров: {row['Количество выданных за последний год']}")
    i += 1

plt.figure(figsize=(12, 6))

# График: Среднее количество выданных экземпляров для каждого жанра
df.groupby('Жанр')['Количество выданных за последний год'].mean().plot(kind='bar')
plt.title('Среднее количество выданных экземпляров для каждого жанра')
plt.xlabel('Жанр')
plt.ylabel('Среднее количество выданных экземпляров')

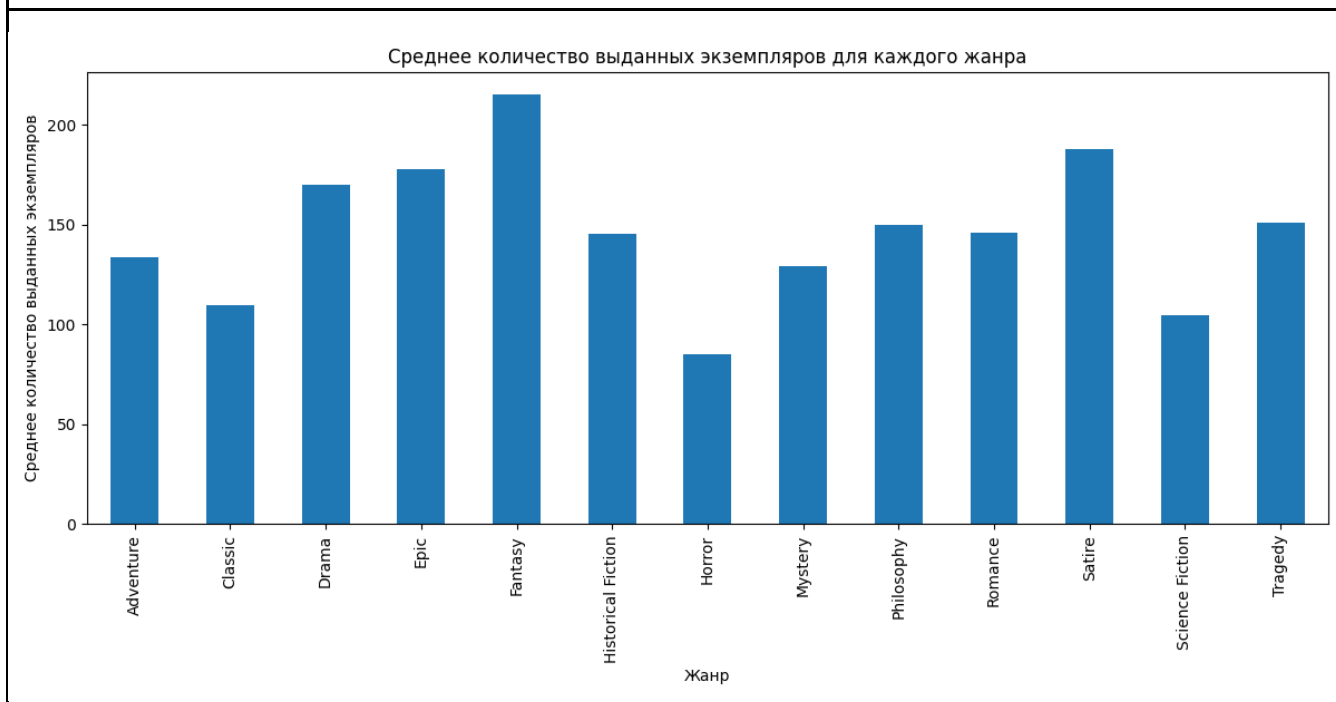
# Отображение графиков
plt.tight_layout()
plt.show()
```

Топ-3 популярных книг:

1. "Wuthering Heights" (George Orwell) - количество выданных экземпляров: 298

2. "Ulysses" (William Shakespeare) - количество выданных экземпляров: 295

3. "The Iliad" (J.R.R. Tolkien) - количество выданных экземпляров: 292



Задача 6.

В приведённом коде ошибка заключается в том, что предсказания выполняются на обучающем наборе данных `X_train`, а не на тестовом наборе `X_test`. Это может привести к неверной оценке качества модели, поскольку предсказания на обучающей выборке не показывают, как модель будет работать на новых (неизвестных) данных.

Чтобы исправить эту ошибку, нужно поменять `'X_train'` на `'X_test'` в методе `'predict'`. Вот исправленный код:

```
from sklearn.datasets import load_iris
from sklearn.model_selection import train_test_split
from sklearn.neighbors import KNeighborsClassifier

# Загружаем данные
iris = load_iris()
X, y = iris.data, iris.target

# Разделяем данные
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2)

# Обучаем модель
knn = KNeighborsClassifier(n_neighbors=3)
knn.fit(X_train, y_train)

# Предсказания
y_pred = knn.predict(X_test) # Исправлено с X_train на X_test
print("Предсказания:", y_pred)
```

Задача 7.

Для нормализации данных нужно создать несколько таблиц, чтобы устранить избыточность и обеспечить целостность данных.

Таблицы:

1. Authors (Авторы):

- `author_id` (PK): Уникальный идентификатор автора.
- `first_name`: Имя автора.
- `last_name`: Фамилия автора.

Один автор может написать несколько книг, поэтому связь между авторами и книгами будет один-ко-многим.

2. Books (Книги):

- `book_id` (PK): Уникальный идентификатор книги.
- `title`: Название книги.
- `year_published`: Год издания книги.
- `price`: Цена книги.
- `author_id` (FK): Ссылка на автора книги.

Каждая книга имеет одного автора, поэтому у каждой книги есть внешний ключ `author_id`, который ссылается на таблицу Authors.

3. Orders (Заказы):

- `order_id` (PK): Уникальный идентификатор заказа.
- `order_date`: Дата заказа.
- `total_amount`: Общая сумма заказа.

Один заказ может содержать несколько книг.

4. OrderDetails (Детали заказа):

- `order_detail_id` (PK): Уникальный идентификатор записи в таблице.
- `order_id` (FK): Ссылка на заказ.
- `book_id` (FK): Ссылка на книгу.
- `quantity`: Количество экземпляров книги в заказе.
- `price`: Цена книги на момент заказа.

Таблица OrderDetails используется для хранения информации о том, какие книги входят в заказ, и в каком количестве. В каждой записи — информация о конкретной книге в конкретном заказе.

```

-- Создаем таблицу для авторов
CREATE TABLE authors (
  author_id SERIAL PRIMARY KEY,
  first_name VARCHAR(100) NOT NULL,
  last_name VARCHAR(100) NOT NULL
);

-- Создаем таблицу для книг
CREATE TABLE books (
  book_id SERIAL PRIMARY KEY,
  title VARCHAR(255) NOT NULL,
  publication_year INT CHECK (publication_year > 0),
  price DECIMAL(10, 2) NOT NULL,
  author_id INT NOT NULL,
  FOREIGN KEY (author_id) REFERENCES authors(author_id) ON DELETE CASCADE
);

-- Создаем таблицу для заказов
CREATE TABLE orders (
  order_id SERIAL PRIMARY KEY,
  order_date DATE NOT NULL,
  total_amount DECIMAL(10, 2) NOT NULL
);

-- Создаем таблицу для связи заказов и книг
CREATE TABLE order_books (
  order_id INT NOT NULL,
  book_id INT NOT NULL,
  PRIMARY KEY (order_id, book_id),
  FOREIGN KEY (order_id) REFERENCES orders(order_id) ON DELETE CASCADE,
  FOREIGN KEY (book_id) REFERENCES books(book_id) ON DELETE CASCADE
);

```

3

```

SELECT
  Books.title AS book_title,
  CONCAT(Authors.first_name, ' ', Authors.last_name) AS author_name
FROM
  Books
JOIN
  Authors ON Books.author_id = Authors.author_id;

```

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. Введение в машинное обучение: как начать и что нужно знать // URL: https://serverspace.ru/about/blog/vvedenie-v-mashinnoe-obuchenie-kak-nachat-i-chto-nuzhno-znat/?utm_source=google.com&utm_medium=organic&utm_campaign=google.com&utm_referrer=google.com
14. Как привести данные в форму: что такое нормализация и зачем она нужна // URL: <https://practicum.yandex.ru/blog/chto-takoe-normalizaciya-dannyh/>
15. Шпаргалка по SQL (postgres) // URL: <https://habr.com/ru/articles/745948/>
16. Лучшие практики проектирования баз данных PostgreSQL // URL: <https://appmaster.io/ru/blog/luchshie-praktiki-proektirovaniia-baz-dannykh-postgresql>