**Машинное обучение. Python. Задание для неполноценных.**

**Задача 1**

Создайте программу, которая принимает текст от пользователя и анализирует частоту

появления каждого слова в этом тексте. Программа должна выводить слова и их

количество в порядке убывания частоты.

- Программа должна разбивать текст на слова. Убедитесь, что слова очищены от знаков

препинания и приведены к нижнему регистру.

- Используйте словарь для хранения слов и их частоты. Ключом будет слово, а значением

— количество его появлений.

- Выведите слова и их частоту в порядке убывания частоты. Если несколько слов имеют

одинаковую частоту, выведите их в алфавитном порядке.

**Код программы:**

Изображение выглядит как текст, снимок экрана, программное обеспечение, Мультимедийное программное обеспечение

Автоматически созданное описание

**Пруф, что оно работает:**

Изображение выглядит как текст, снимок экрана, Шрифт

Автоматически созданное описание

**Описание кода:**

**from collections import Counter**

**import string**

**Импорт библиотек**:

* from collections import Counter: Импортируем Counter из модуля collections. Это специальный тип словаря, который позволяет легко подсчитывать количество элементов.
* import string: Импортируем модуль string для работы со строками. Нам понадобится набор знаков препинания для их удаления из текста.

**def analyze\_text\_frequency(text):**

**Определение функции**:

* Мы создаем функцию analyze\_text\_frequency, которая принимает один аргумент — текст, введённый пользователем. Эта функция будет анализировать частоту слов.

**text = text.lower().translate(str.maketrans('', '', string.punctuation))**

**Приведение текста к нижнему регистру и удаление знаков препинания**:

* text.lower(): Преобразуем весь текст в нижний регистр, чтобы слова вроде "Слово" и "слово" считались одинаковыми.
* str.maketrans('', '', string.punctuation): Создаём таблицу преобразования для удаления всех знаков препинания.
* translate(): Применяем созданную таблицу преобразования, чтобы удалить знаки препинания из текста.

**words = text.split()**

**Разбиение текста на отдельные слова**:

* text.split(): Разбиваем текст на отдельные слова, используя пробелы в качестве разделителей. Функция split() возвращает список слов.

**word\_count = Counter(words)**

**Подсчет частоты появления слов**:

* Counter(words): Используем Counter для подсчета частоты каждого слова в списке words. Counter создаёт словарь, где ключ — это слово, а значение — количество его появлений.

**sorted\_word\_count = sorted(word\_count.items(), key=lambda x: (-x[1], x[0]))**

**Сортировка слов по частоте и алфавиту**:

* word\_count.items(): Получаем пары (слово, частота) в виде списка кортежей.
* sorted(..., key=lambda x: (-x[1], x[0])): Сортируем список слов:
* -x[1]: Сортируем по частоте появления в убывающем порядке (большие значения идут первыми).
* x[0]: Если два слова имеют одинаковую частоту, сортируем их по алфавиту (в порядке возрастания).

**for word, count in sorted\_word\_count:**

**print(f"{word}: {count}")**

**Вывод результата**:

* Цикл for проходит по каждому слову и его частоте в отсортированном списке.
* print(f"{word}: {count}"): Выводит слово и количество его появлений.

**if \_\_name\_\_ == "\_\_main\_\_":**

**sample\_text = input("Введите текст для анализа: ")**

**analyze\_text\_frequency(sample\_text)**

**Основная программа**:

* if \_\_name\_\_ == "\_\_main\_\_": проверяет, была ли программа запущена напрямую, а не импортирована как модуль.
* input("Введите текст для анализа: "): Запрашиваем у пользователя ввод текста.
* analyze\_text\_frequency(sample\_text): Вызываем функцию для анализа частоты слов в тексте.

**Как программа работает:**

1. Пользователь вводит текст.
2. Программа очищает текст от знаков препинания и приводит его к нижнему регистру.
3. Программа разбивает текст на слова.
4. Частота появления каждого слова подсчитывается.
5. Слова сортируются по частоте и алфавиту (если частота одинаковая).
6. Программа выводит результат в виде списка слов и их количества.

**Задача 2**

Компания реализует оказание услуг в области ногтевого сервиса. Вас просят, как

налитика данных, рассчитать некоторую статистику по одной неделе.

Имеются 3 списка, сгенерированных в результате работы в эту неделю:

Первый список характеризует виды маникюра, которые делает данный салон:

Nail\_style=[“Шеллак”, “Френч”, “Обычный лак”, “Гель-лак”, “Акрил”]

Второй список характеризует цены на эти услуги:

Price = [2000, 1500, 1000, 3000, 3500, 3200]

Третий список показывает, сколько раз за неделю заказывали тот или иной вид

маникюра соответственно

Week = [4, 5, 4, 8, 6]

Хозяйка салона просит вас написать приложение, которое выводит в консоль:

Среднее значение посещений салона

Общее количество посещений салона

•

Выручку салона

•

Выручку по видам маникюра

•

Среднюю выручку в день по видам маникюра

**Код программы:**

Изображение выглядит как текст, снимок экрана, программное обеспечение, Шрифт

Автоматически созданное описание

**Пруф, что оно работает:**

Изображение выглядит как текст, снимок экрана, Шрифт, программное обеспечение

Автоматически созданное описание

**Описание кода:**

**Nail\_style = ["Шеллак", "Френч", "Обычный лак", "Гель-лак", "Акрил"]**

**Price = [2000, 1500, 1000, 3000, 3500]**

**Week = [4, 5, 4, 8, 6]**

**Инициализация данных:**

* Nail\_style: Список видов маникюра, которые предлагает салон.
* Price: Список цен на каждый вид маникюра (цены соответствуют порядку маникюра в списке Nail\_style).
* Week: Список, показывающий, сколько раз каждую услугу заказывали за неделю.

**average\_visits = sum(Week) / len(Week)**

**Среднее количество посещений за неделю**:

* sum(Week): Суммируем количество заказов за неделю по всем услугам.
* len(Week): Количество видов маникюра (всего 5).
* Делим общее количество посещений на количество видов маникюра, чтобы получить среднее число посещений.

**total\_visits = sum(Week)**

**Общее количество посещений за неделю**:

* + sum(Week): Суммируем все значения из списка Week, чтобы узнать, сколько всего было заказов за неделю.

**revenue\_by\_style = [Price[i] \* Week[i] for i in range(len(Nail\_style))]**

**Выручка по видам маникюра**:

* Используем цикл **list comprehension**, чтобы вычислить выручку по каждому виду маникюра.
* Price[i] \* Week[i]: Умножаем цену каждого маникюра на количество заказов этого маникюра за неделю.
* for i in range(len(Nail\_style)): Проходим по каждому элементу списка маникюров.

**total\_revenue = sum(revenue\_by\_style)**

**Расчет общей выручки салона**:

* sum(revenue\_by\_style): Суммируем все элементы списка revenue\_by\_style, чтобы получить общую выручку за неделю от всех видов маникюра.

**# Средняя дневная выручка по видам маникюра**

**average\_daily\_revenue\_by\_style = [revenue / 7 for revenue in revenue\_by\_style]**

**Расчет средней дневной выручки по каждому виду маникюра**:

* + Для каждого элемента списка revenue\_by\_style делим выручку на 7 (количество дней в неделе).
  + В результате получаем список average\_daily\_revenue\_by\_style, который содержит средний доход по каждому виду маникюра за день.

**# Вывод результатов**

**print(f"Среднее количество посещений салона за неделю: {average\_visits:.2f}")**

**print(f"Общее количество посещений салона за неделю: {total\_visits}")**

**print(f"Общая выручка салона за неделю: {total\_revenue} руб.")**

**Вывод статистики посещений и общей выручки**:

* Выводим среднее количество посещений, общее количество посещений и общую выручку салона за неделю. Значения округлены до двух знаков после запятой ({average\_visits:.2f}).

**print("\nВыручка по видам маникюра:")**

**for i in range(len(Nail\_style)):**

**print(f"{Nail\_style[i]}: {revenue\_by\_style[i]} руб.")**

**Вывод выручки по каждому виду маникюра**:

* Цикл for проходит по каждому виду маникюра.
* Выводим название маникюра (из списка Nail\_style) и соответствующую выручку (из списка revenue\_by\_style).

**print("\nСредняя дневная выручка по видам маникюра:")**

**for i in range(len(Nail\_style)):**

**print(f"{Nail\_style[i]}: {average\_daily\_revenue\_by\_style[i]:.2f} руб. в день")**

**Вывод средней дневной выручки по видам маникюра**:

* Цикл снова проходит по каждому виду маникюра.
* Выводим название маникюра (из списка Nail\_style) и среднюю дневную выручку (из списка average\_daily\_revenue\_by\_style), округленную до двух знаков после запятой.

**Как работает программа:**

1. **Данные**: Программа берет три списка — виды маникюра, цены и количество заказов за неделю.
2. **Расчеты**: Программа рассчитывает среднее и общее количество посещений, выручку по каждому виду маникюра, общую выручку и среднюю дневную выручку по каждому виду.
3. **Вывод**: Все результаты выводятся в консоль, включая выручку по каждому виду маникюра и средний дневной доход.