Министерство науки и высшего образования Российской Федерации Федеральное государственное бюджетное образовательное учреждение высшего образования

«Ульяновский государственный технический университет»

Кафедра «Вычислительная техника»

Дисциплина «Системы искусственного интеллекта»

Лабораторная работа №2

«Pandas и основы Python»

Выполнил:

студент группы ИВТАПбд-41

Сокольский Р.С

Проверил:

преподаватель кафедры «ВТ»

Святов К.В.

Ульяновск, 2025

Оглавление

[1. Постановка задачи 3](#_Toc210251350)

[2. Реализация 4](#_Toc210251351)

[3. Тестирование 6](#_Toc210251352)

[4. Выводы 8](#_Toc210251353)

# **1. Постановка задачи**

1. Сгенерировать с использованием библиотеки pandas csv-файл, содержащий 3 столбца: x1, x2, y (> 400 строк, генерация с использованием функции linspace). Вид функции определяется вариантом (диапазон выбирается студентом).
2. Открыть файл с использованием библиотеки pandas и построить графики: y(x1) (x2 - константа), y(x2) (x1 - константа). Графики необходимо построить с использованием matplotlib (для функции y отобразить точки на графике).
3. Вывести для каждого столбца (x1, x2, y): среднее, минимальное и максимальное значения.
4. Сохранить в новый csv-файл те строки, для которых выполняется условие: x1 меньше среднее x1 или x2 меньше среднее x2
5. С использованием mplot3D построить 3D график функции y(x1, x2) в отдельном окне

Вариант 22:

# **2****. Реализация**

В начале происходит генерация данных:

**Листинг 1.** Генерация данных

|  |
| --- |
| **n\_points = 500**  **x1 = np.linspace(-2\*np.pi, 2\*np.pi, n\_points)**  **x2 = np.linspace(-np.pi, np.pi, n\_points)**  **y = np.cos(x1 + x2)**  **data = pd.DataFrame({**  **'x\_1': x1,**  **'x\_2': x2,**  **'y': y**  **})**  **data.to\_csv('lab2\_data.csv', index=False)**  **print("Файл 'lab2\_data.csv' успешно создан!")**  **print(f"Размер данных: {data.shape}")**  **print()** |

В результате создаётся csv-файл с 500 точками. Далее идёт отображение 2D-графиков:

**Листинг 2.** Код отображения 2D-графиков

|  |  |
| --- | --- |
| |  | | --- | | **data = pd.read\_csv('lab2\_data.csv')**  **plt.figure(figsize=(15, 5))**  **plt.subplot(1, 2, 1)**  **x2\_const = data['x\_2'].mean()**  **x1\_for\_plot = np.linspace(data['x\_1'].min(), data['x\_1'].max(), n\_points)**  **y\_for\_plot\_x1 = np.cos(x1\_for\_plot + x2\_const)**  **plt.scatter(x1\_for\_plot, y\_for\_plot\_x1, alpha=0.7, color='blue', s=1)**  **plt.xlabel('x\_1')**  **plt.ylabel('y')**  **plt.title(f'y(x\_1) при x\_2 = {x2\_const:.2f} (константа)')**  **plt.grid(True, alpha=0.3)**  **plt.subplot(1, 2, 2)**  **x1\_const = data['x\_1'].mean()**  **x2\_for\_plot = np.linspace(data['x\_2'].min(), data['x\_2'].max(), n\_points)**  **y\_for\_plot\_x2 = np.cos(x1\_const + x2\_for\_plot)**  **plt.scatter(x2\_for\_plot, y\_for\_plot\_x2, alpha=0.7, color='red', s=1)**  **plt.xlabel('x\_2')**  **plt.ylabel('y')**  **plt.title(f'y(x\_2) при x\_1 = {x1\_const:.2f} (константа)')**  **plt.grid(True, alpha=0.3)**  **plt.tight\_layout()**  **plt.show()** |   После этого идёт отображение статистики: |

**Листинг 3.** Код отображения статистики

|  |  |
| --- | --- |
| |  | | --- | | **for column in data.columns:**  **print(f"Столбец {column}:")**  **print(f"- Среднее: {data[column].mean():.4f}")**  **print(f"- Минимальное: {data[column].min():.4f}")**  **print(f"- Максимальное: {data[column].max():.4f}\n")** | |

Потом идёт сохранение отфильтрованных данных:

**Листинг 4.** Код сохранения отфильтрованных данных

|  |  |
| --- | --- |
| |  | | --- | | **mean\_x1 = data['x\_1'].mean()**  **mean\_x2 = data['x\_2'].mean()**  **print(f"- Среднее x\_1: {mean\_x1:.4f}")**  **print(f"- Среднее x\_2: {mean\_x2:.4f}\n")**  **filtered\_data = data[(data['x\_1'] < mean\_x1) | (data['x\_2'] < mean\_x2)]**  **filtered\_data.to\_csv('lab2\_filtered\_data.csv', index=False)**  **print(f"- Отфильтрованный файл 'lab2\_filtered\_data.csv' создан!")**  **print(f"- Исходных строк: {len(data)}")**  **print(f"- Отфильтрованных строк: {len(filtered\_data)}\n")** | |

И наконец, отображение 3D-графика:

**Листинг 5.** Код отображения 3D-графика

|  |  |
| --- | --- |
| |  | | --- | | **fig = plt.figure(figsize=(12, 8))**  **ax = plt.axes(projection='3d')**  **grid\_resolution = 25**  **X1, X2 = np.meshgrid(np.linspace(-2\*np.pi, 2\*np.pi, grid\_resolution), np.linspace(-np.pi, np.pi, grid\_resolution))**  **Y = np.cos(X1 + X2)**  **surf = ax.plot\_surface(X1, X2, Y, cmap='viridis', alpha=0.8, linewidth=0, edgecolor='none', antialiased=True)**  **ax.scatter(X1.flatten(), X2.flatten(), Y.flatten(), color='red', s=1, alpha=1.0, edgecolors='darkred', linewidth=0.5)**  **fig.colorbar(surf, ax=ax, shrink=0.5, aspect=5, label='y')**  **ax.set\_xlabel('x\_1')**  **ax.set\_ylabel('x\_2')**  **ax.set\_zlabel('y')**  **ax.set\_title('3D-график: y = cos(x\_1 + x\_2)')**  **plt.tight\_layout()**  **plt.show()** | |

# **3. Тестирование**

1. Генерация CSV-файла.



2. Построение графиков.

Изображение выглядит как График, линия, текст, число

Содержимое, созданное искусственным интеллектом, может быть неверным.

3. Статистика по столбцам.

Изображение выглядит как текст, снимок экрана, Шрифт, дизайн

Содержимое, созданное искусственным интеллектом, может быть неверным.

4. Сохранение отфильтрованных данных.

Изображение выглядит как текст, снимок экрана, Шрифт

Содержимое, созданное искусственным интеллектом, может быть неверным.

5. Построение 3D-графика.

Изображение выглядит как текст, диаграмма, снимок экрана, График

Содержимое, созданное искусственным интеллектом, может быть неверным.

# **4.** **Выводы**

В данной лабораторной работе были сгенерированы данные для двухмерной квадратичной функции. Построены 2D- и 3D-графики, демонстрирующие зависимость. Проанализированы статистические свойства данных.