|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **SCHOOL OF COMPUTER SCIENCE AND ARTIFICIAL INTELLIGENCE** | | | | | **DEPARTMENT OF COMPUTER SCIENCE ENGINEERING** | | | | |
| **Program Name:** B. Tech | | | | **Assignment Type: Lab** | | | **Academic Year:**2025-2026 | | |
| **Course Coordinator Name** | | | | Venkataramana Veeramsetty | | | | | |
| **Instructor(s) Name** | | | | |  | | --- | | Dr. V. Venkataramana (Co-ordinator) | | Dr. T. Sampath Kumar | | Dr. Pramoda Patro | | Dr. Brij Kishor Tiwari | | Dr.J.Ravichander | | Dr. Mohammand Ali Shaik | | Dr. Anirodh Kumar | | Mr. S.Naresh Kumar | | Dr. RAJESH VELPULA | | Mr. Kundhan Kumar | | Ms. Ch.Rajitha | | Mr. M Prakash | | Mr. B.Raju | | Intern 1 (Dharma teja) | | Intern 2 (Sai Prasad) | | Intern 3 (Sowmya) | | NS\_2 ( Mounika) | | | | | | |
| **Course Code** | | | 24CS002PC215 | **Course Title** | | AI Assisted Coding | | | |
| **Year/Sem** | | | II/I | **Regulation** | | R24 | | | |
| **Date and Day**  **of Assignment** | | | Week9 - WednesDay | **Time(s)** | |  | | | |
| **Duration** | | | 2 Hours | **Applicable to**  **Batches** | |  | | | |
| **AssignmentNumber:17.3**(Present assignment number)/**24**(Total number of assignments) | | | | | | | | | |
|  | | | | | | | | | |
|  | **Q.No.** | **Question** | | | | | | ***Expected Time***  ***to complete*** |  |
|  | 1 | **Lab 17 – AI for Data Processing: Data Cleaning and Preprocessing Scripts**  **Lab Objectives:**   * Learn how to clean raw datasets using AI-assisted Python scripting. * Apply preprocessing techniques such as handling missing values, encoding categorical data, and normalization. * Automate repetitive data-cleaning tasks with AI-generated code. * Understand how preprocessing impacts model performance.   **Task 1 – Social Media Data Cleaning**  **Task: Clean raw social media posts dataset.**  **Instructions:**  - Remove stopwords, punctuation, and special symbols from post text.  - Handle missing values in likes and shares columns.  - Convert timestamp to datetime and extract features (hour, weekday).  - Detect and remove spam/duplicate posts.  **Expected Output:** A cleaned dataset with structured features for sentiment/engagement analysis.  **Task 2 – Financial Data Preprocessing**  **Task: Preprocess a stock market dataset.**  **Instructions:**  - Handle missing values in closing\_price and volume.  - Create lag features (1-day, 7-day returns).  - Normalize volume column using log-scaling.  - Detect outliers in closing\_price using IQR method.  **Expected Output:** A time-series dataset ready for forecasting models.  **Task 3 – IoT Sensor Data Preparation**  **Task: Clean and preprocess IoT temperature and humidity logs.**  **Instructions:**  - Handle missing values using forward fill.  - Remove sensor drift (apply rolling mean).  - Normalize readings using standard scaling.  - Encode categorical sensor IDs.  **Expected Output:** A structured dataset optimized for anomaly detection.  **Task 4 – Real-Time Application: Movie Reviews Data Cleaning**  **Task: A streaming platform wants to analyze customer reviews.**  **Instructions:**  - Standardize text (lowercase, remove HTML tags).  - Tokenize and encode reviews using AI-assisted methods (TF-IDF or embeddings).  - Handle missing ratings (fill with median).  - Normalize ratings (0–10 → 0–1 scale).  - Generate a before vs after summary report.  **Expected Output:** A cleaned dataset ready for sentiment classification.  ✅ Deliverables (For All Tasks)   1. AI-generated prompts for code and test case generation. 2. At least 3 assert test cases for each task. 3. AI-generated initial code and execution screenshots. 4. Analysis of whether code passes all tests. 5. Improved final version with inline comments and explanations. 6. Compiled report (Word/PDF) with prompts, test cases, assertions, code, and output.Top of Form | | | | | | Week9 - Monday |  |

**Task 1 – Social Media Data Cleaning**

**Task: Clean raw social media posts dataset.**

**Instructions:**

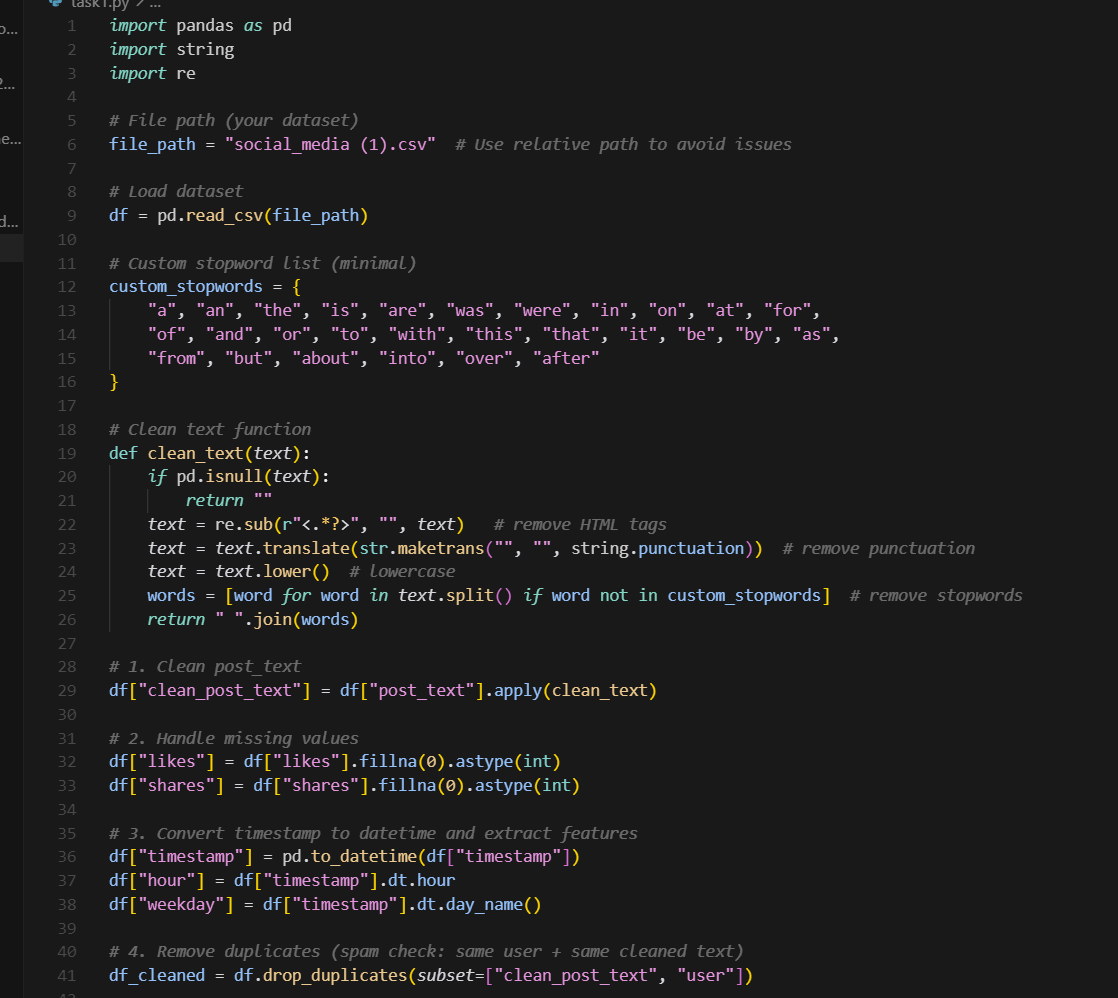
- Remove stopwords, punctuation, and special symbols from post text.

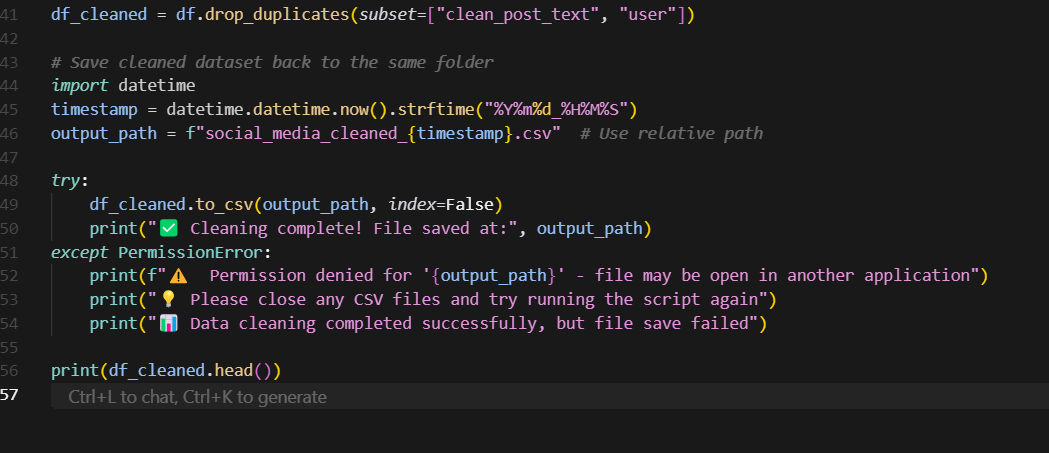
- Handle missing values in likes and shares columns.

- Convert timestamp to datetime and extract features (hour, weekday).

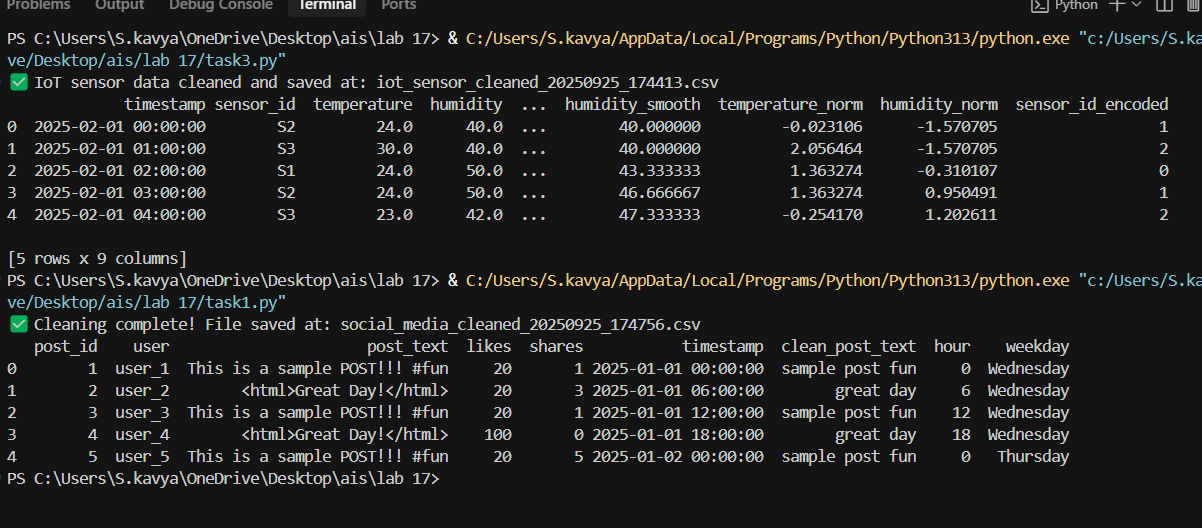
- Detect and remove spam/duplicate posts.

**Expected Output:** A cleaned dataset with structured features for sentiment/engagement analysis





Output:



**Task 2 – Financial Data Preprocessing**

**Task: Preprocess a stock market dataset.**

**Instructions:**

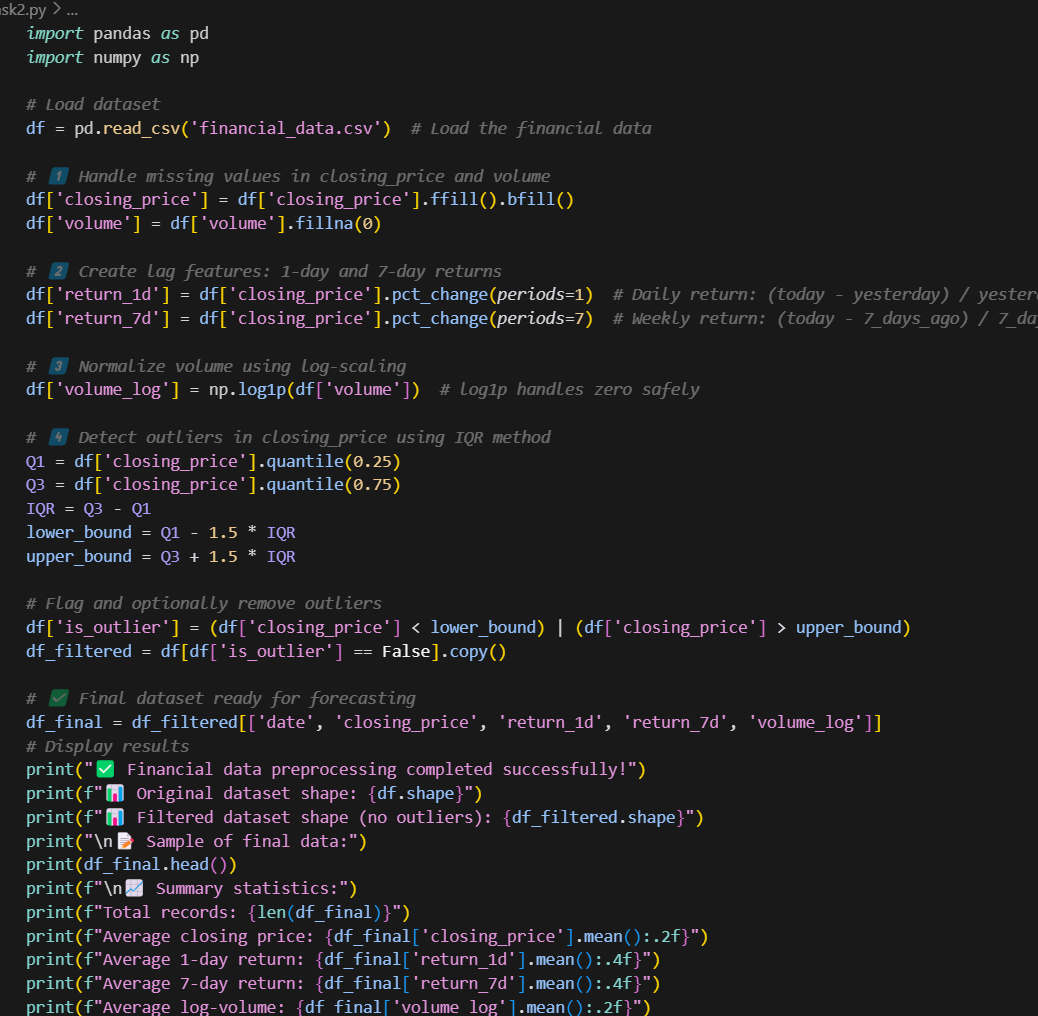
- Handle missing values in closing\_price and volume.

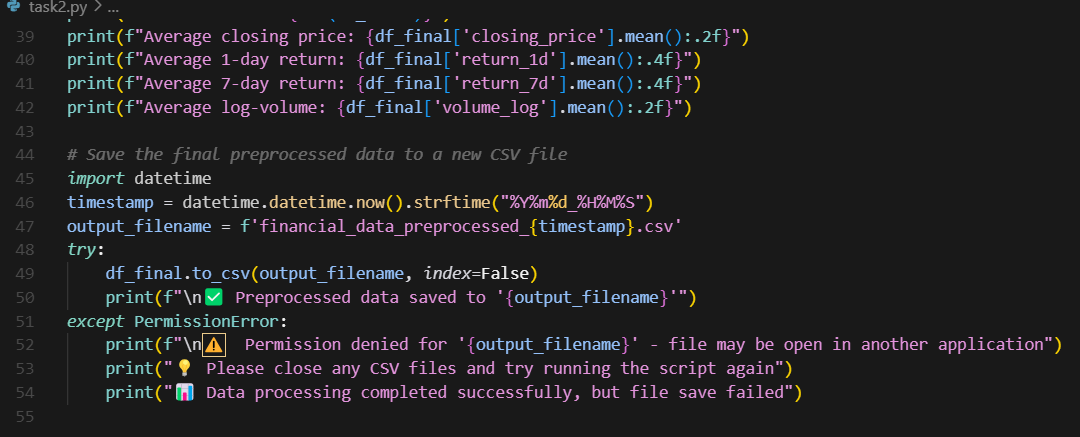
- Create lag features (1-day, 7-day returns).

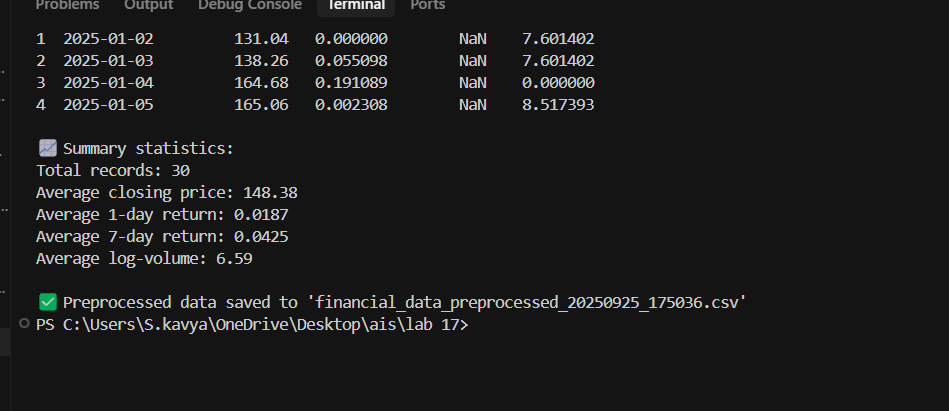
- Normalize volume column using log-scaling.

- Detect outliers in closing\_price using IQR method.

**Expected Output:** A time-series dataset ready for forecasting models







**Task 3 – IoT Sensor Data Preparation**

**Task: Clean and preprocess IoT temperature and humidity logs.**

**Instructions:**

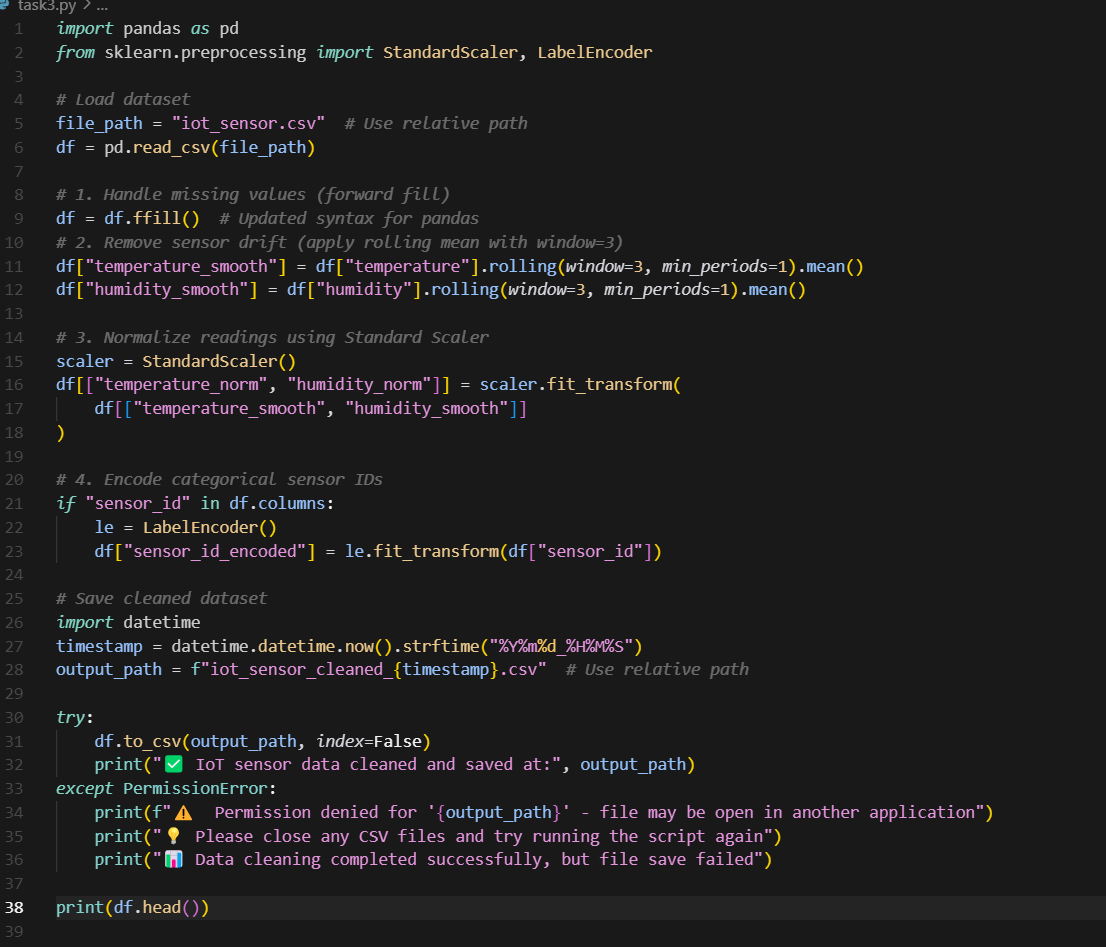
- Handle missing values using forward fill.

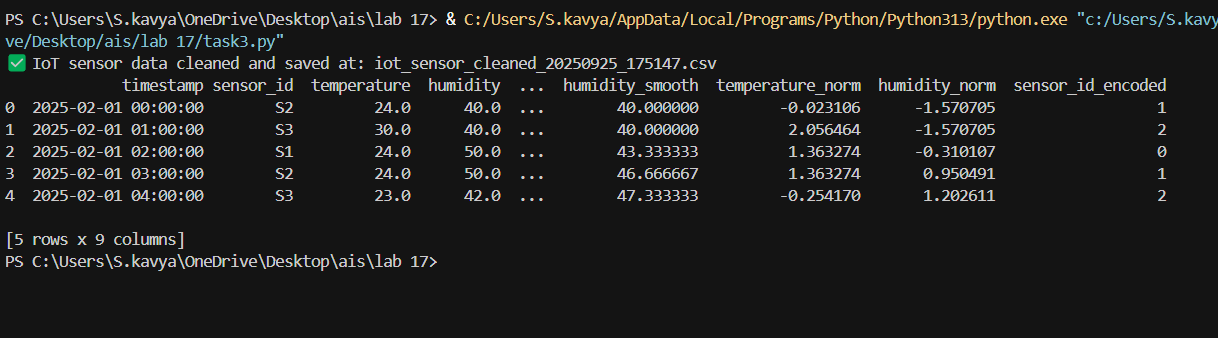
- Remove sensor drift (apply rolling mean).

- Normalize readings using standard scaling.

- Encode categorical sensor IDs.

**Expected Output:** A structured dataset optimized for anomaly detection.





**Task 4 – Real-Time Application: Movie Reviews Data Cleaning**

**Task: A streaming platform wants to analyze customer reviews.**

**Instructions:**

- Standardize text (lowercase, remove HTML tags).

- Tokenize and encode reviews using AI-assisted methods (TF-IDF or embeddings).

- Handle missing ratings (fill with median).

- Normalize ratings (0–10 → 0–1 scale).

- Generate a before vs after summary report.

**Expected Output:** A cleaned dataset ready for sentiment classification.

✅ Deliverables (For All Tasks)

1. AI-generated prompts for code and test case generation.
2. At least 3 assert test cases for each task.
3. AI-generated initial code and execution screenshots.
4. Analysis of whether code passes all tests.
5. Improved final version with inline comments and explanations.

Compiled report (Word/PDF) with prompts, test cases, assertions, code, and output.

