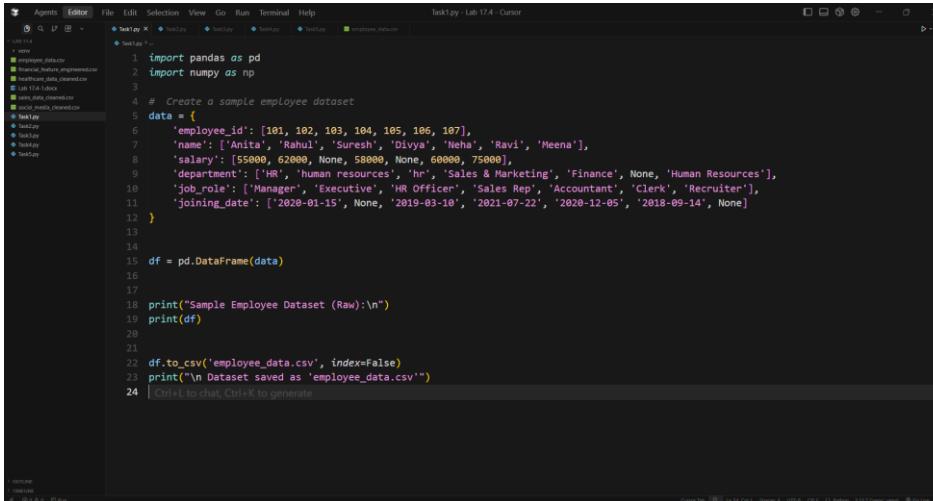


SCHOOL OF COMPUTER SCIENCE AND ARTIFICIAL INTELLIGENCE		DEPARTMENT OF COMPUTER SCIENCE ENGINEERING	
Program Name: B. Tech		Assignment Type: Lab	
Course Coordinator Name		Venkataramana Veeramsetty	
Instructor(s) Name		Dr. V. Venkataramana (Co-ordinator) Dr. T. Sampath Kumar Dr. Pramoda Patro Dr. Brij 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 - Thursday	Time(s)	
Duration	2 Hours	Applicable to Batches	
AssignmentNumber: 17.4(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: <ul style="list-style-type: none"> Learn how to clean raw datasets using AI-assisted Python scripting. 		Week9 - Thursday

	<ul style="list-style-type: none"> 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. 	
	<p>Task 1 – Employee Data Preprocessing</p> <p>Task: Use AI to generate a Python script for cleaning an employee dataset.</p> <p>Instructions:</p> <ul style="list-style-type: none"> Handle missing values in columns (salary, department, joining_date). Convert the "joining_date" column into proper datetime format. Standardize department names (e.g., "HR", "hr", "Human Resources" → "HR"). Encode categorical variables (department, job_role). <p>Expected Output:</p> <ul style="list-style-type: none"> A cleaned Pandas DataFrame with consistent departments, proper dates, and encoded features. <p>Code:</p>  <pre> 1 Agents Editor File Edit Selections View Go Run Terminal Help 2 Task1.ipynb Task1.ipynb Task1.ipynb Task1.ipynb Task1.ipynb employee_data.csv 3 4 # Create a sample employee dataset 5 data = [6 {'employee_id': [101, 102, 103, 104, 105, 106, 107], 7 'name': ['Anita', 'Rahul', 'Suresh', 'Divya', 'Neha', 'Ravi', 'Meena'], 8 'salary': [55000, 62000, None, 58000, None, 60000, 75000], 9 'department': ['HR', 'human resources', 'hr', 'Sales & Marketing', 'Finance', None, 'Human Resources'], 10 'job_role': ['Manager', 'Executive', 'HR Officer', 'Sales Rep', 'Accountant', 'Clerk', 'Recruiter'], 11 'joining_date': ['2020-01-15', None, '2019-03-10', '2021-07-22', '2020-12-05', '2018-09-14', None] 12 } 13 14 15 df = pd.DataFrame(data) 16 17 18 print("Sample Employee Dataset (Raw):\n") 19 print(df) 20 21 22 df.to_csv('employee_data.csv', index=False) 23 print("\nDataset saved as 'employee_data.csv'") 24 Ctrl+L to chat, Ctrl+K to generate </pre> <p>Output:</p>	

```

Agents Editor File Edit Selection View Go Run Terminal Help Task1py - Lab 17.4 - Cursor
PS C:\Users\srirah\Desktop\AIAC\Lab 17.4> & "c:/users/srirah/desktop/aiac/lab 17.4/env/scripts/activate.ps1"
PS C:\Users\srirah\Desktop\AIAC\Lab 17.4> & "c:/users/srirah/Desktop/aiac/lab 17.4/env/scripts/python.exe" "c:/users/srirah/Desktop/aiac/lab 17.4/Task1.py"
Sample Employee Dataset (Raw):
employee_id name salary department job_role joining_date
0 101 Anita 55000.0 HR Manager 2020-01-15
1 102 Rahul 60000.0 Human resources Executive None
2 103 Suresh 70000.0 HR Officer 2019-03-10
3 104 Divya 50000.0 Sales & Marketing Sales Rep 2021-07-22
4 105 Akash 40000.0 Finance Accountant 2018-12-05
5 106 Ravi 68000.0 None Clerk 2018-09-14
6 107 Meena 75000.0 Human Resources Recruiter None
Dataset saved as 'employee_data.csv'
PS C:\Users\srirah\Desktop\AIAC\Lab 17.4>

```

Task 2 – Sales Transaction Data Preprocessing

Task:

Use AI to generate a script for preprocessing a sales transaction dataset.

Instructions:

- Convert transaction dates to proper datetime format.
- Create a new column for “Month-Year” from the transaction date.
- Remove rows with negative or zero transaction amounts.
- Normalize the "transaction_amount" column using Min-Max scaling.

Expected Output:

- A preprocessed DataFrame with valid dates, normalized amounts, and no invalid records.

Code:

```

Agents Editor File Edit Selection View Go Run Terminal Help Task2py - Lab 17.4 - Cursor
File: Task2py
1 import pandas as pd
2 from sklearn.preprocessing import MinMaxScaler
3
4
5 data = {
6     'transaction_id': [1,2,3,4,5,6,7,8,9,10],
7     'customer_id': [101,102,103,104,105,106,107,108,109,110],
8     'transaction_date': ['2025-10-01','2025-10-03','2025-10-05','2025-11-01',
9                          '2025-11-02','2025-11-03','invalid_date','2025-11-05',
10                         '2025-11-06','2025-11-07'],
11     'transaction_amount': [250,0,-50,450,300,500,200,0,150,600],
12     'product': ['Notebook','Pen','Marker','Printer','Paper','Laptop','Mouse','Keyboard','Stapler','Desk']
13 }
14
15 df = pd.DataFrame(data)
16 print("Original Dataset:")
17 print(df)
18
19
20 df['transaction_date'] = pd.to_datetime(df['transaction_date'], errors='coerce')
21
22
23 df = df.dropna(subset=['transaction_date'])

```

Output:

```

Agents Editor File Edit Selection View Go Run Terminal Help Task2.py - Lab 17.4 - Cursor
PS C:\Users\urisha\Desktop\AI4C\Lab 17.4> & "C:\Users\urisha\Desktop\AI4C\Lab 17.4\venv\Scripts\activate.ps1"
(venv) PS C:\Users\urisha\Desktop\AI4C\Lab 17.4> & "C:\Users\urisha\Desktop\AI4C\Lab 17.4\venv\Scripts\python.exe" "C:/Users/urisha/Desktop/AI4C/Lab 17.4/Task2.py"
transaction_id customer_id transaction_date transaction_amount product
0 1 100 2025-10-01 250 Notebook
1 2 100 2025-10-01 0 Pen
2 3 100 2025-10-05 .50 Marker
3 4 100 2025-10-05 450 Projector
4 5 100 2025-11-01 300 Paper
5 6 100 2025-11-01 500 Laptop
6 7 100 2025-11-01 200 Headset
7 8 100 2025-11-05 0 Keyboard
8 9 100 2025-11-06 150 Stipend
9 10 100 2025-11-07 600 Desk

Cleaned Dataset
transaction_id customer_id transaction_date transaction_amount product month year transaction_amount_normalized
0 1 100 2025-10 250 Notebook 2025-10 0.222222
1 2 100 2025-10 0 Pen 2025-10 0.000000
2 3 100 2025-11 0.50 Marker 2025-11 0.333333
3 4 100 2025-11 300 Paper 2025-11 0.777778
4 5 100 2025-11 500 Laptop 2025-11 1.000000
5 6 100 2025-11 150 Stipend 2025-11 0.000000
6 7 100 2025-11 600 Desk 2025-11 1.000000

(venv) PS C:\Users\urisha\Desktop\AI4C\Lab 17.4>

```

Task 3 – Healthcare Patient Records Cleaning

Task:

Use AI to generate a script for cleaning healthcare patient records.

Instructions:

- Fill missing values in numeric columns (e.g., blood_pressure, heart_rate) with column mean.
- Standardize units (convert height from cm to meters).
- Correct inconsistent categorical labels (e.g., "M", "Male", "male" → "Male").
- Drop irrelevant columns such as patient_id after cleaning.

Expected Output:

- A cleaned healthcare dataset suitable for ML model training.

Code:

```

Agents Editor File Edit Selection View Go Run Terminal Help Task3.py - Lab 17.4 - Cursor
1 import pandas as pd
2 import numpy as np
3
4 data = [
5     {"patient_id": [1,2,3,4,5],
6      "name": ["Alice","Bob","Charlie","David","Eva"],
7      "gender": ["F", "M", "male", "Female", "M"],
8      "age": [30, 25, 35, 40, 30],
9      "height_cm": [160, 175, 180, np.nan, 165],
10     "weight_kg": [55, 70, 80, 90, np.nan],
11     "blood_pressure": [120, 130, np.nan, 140, 135],
12     "heart_rate": [80, np.nan, 70, 85, 90]
13 }
14
15 df = pd.DataFrame(data)
16 print("Original Dataset")
17 print(df)
18
19 numeric_cols = ['age', 'height_cm', 'weight_kg', 'blood_pressure', 'heart_rate']
20 for col in numeric_cols:
21     df[col].fillna(df[col].mean(), inplace=True)
22
23 df['height_m'] = df['height_cm'] / 100
24 df.drop(columns=['height_cm'], inplace=True) # drop old column
25
26 df['gender'] = df['gender'].replace({'M': 'Male', 'male': 'Male', 'F': 'Female', 'Female': 'Female'})
27
28 df.drop(columns=['patient_id'], inplace=True)
29
30 print("No cleaned Healthcare Dataset")
31 print(df)
32
33 df.to_csv('healthcare_data_cleaned.csv', index=False)
34
35
36

```

Output:

```
Agents Editor File Edit Selection View Go Run Terminal Help Task3.py - Lab 17.4 - Cursor
PS C:\Users\srinivas\Desktop\AIAC\Lab 17.4> & "c:/Users/srinivas/Desktop/AIAC/Lab 17.4/venv/Scripts/Activate.ps1"
PS C:\Users\srinivas\Desktop\AIAC\Lab 17.4> & "c:/Users/srinivas/Desktop/AIAC/Lab 17.4/venv/Scripts/python.exe" "c:/Users/srinivas/Desktop/AIAC/Lab 17.4/Task3.py"
[[{"patient_id": 1, "name": "Alice", "gender": "Female", "age": 25.0, "height_cm": 170.0, "weight_kg": 65.0, "blood_pressure": 120.0, "heart_rate": 1.0}, {"patient_id": 2, "name": "Bob", "gender": "Male", "age": 30.0, "height_cm": 175.0, "weight_kg": 70.0, "blood_pressure": 130.0, "heart_rate": 1.0}, {"patient_id": 3, "name": "Charlie", "gender": "Male", "age": 35.0, "height_cm": 180.0, "weight_kg": 80.0, "blood_pressure": 135.0, "heart_rate": 1.0}, {"patient_id": 4, "name": "Diana", "gender": "Female", "age": 30.0, "height_cm": 165.0, "weight_kg": 55.0, "blood_pressure": 110.0, "heart_rate": 1.0}, {"patient_id": 5, "name": "Eva", "gender": "Female", "age": 30.0, "height_cm": 165.0, "weight_kg": 55.0, "blood_pressure": 110.0, "heart_rate": 1.0}], [{"id": 1, "text": "I am loving this product! Check it out: https://example.com", "sentiment": 1}, {"id": 2, "text": "Great service overall! Product is", "sentiment": 1}, {"id": 3, "text": "This is a really amazing experience!!! Bravo", "sentiment": 1}, {"id": 4, "text": "Wow... it is one https://example.com", "sentiment": 1}, {"id": 5, "text": "I am not satisfied with this product.", "sentiment": 0}], df_clean_text = df['text'].apply(clean_text)
df['clean_text'] = df_clean_text
df['sentiment'] = df['sentiment'].apply(lambda x: 1 if x == 'positive' else 0)
df['neutral'] = df['sentiment'].apply(lambda x: 1 if x == 'neutral' else 0)
df['negative'] = df['sentiment'].apply(lambda x: 1 if x == 'negative' else 0)
df.to_csv('social_media_cleaned.csv', index=False)
Python Task1
Python Task2
Python Task3
```

The screenshot shows a Jupyter Notebook interface with a terminal tab open. The terminal displays Python code being run to clean a dataset and extract sentiment scores. The output shows the cleaned dataset and the resulting sentiment analysis for each tweet.

Task 4 – Social Media Sentiment Dataset Preparation

Task:

Use AI to write a script to preprocess a social media text dataset.

Instructions:

- Remove special characters, URLs, and emojis from text.
- Convert all text to lowercase.
- Tokenize and remove stopwords.
- Apply lemmatization for standardizing words.

Expected Output:

- A processed dataset with clean text, ready for NLP sentiment analysis.

Code:

```
Agents Editor File Edit Selection View Go Run Terminal Help Task4.py - Lab 17.4 - Cursor
[...]
def clean_text(text):
    text = re.sub(r'[^\w\s]', '', text)
    text = re.sub(r'[^a-zA-Z\s]', '', text)
    text = re.sub(r'\b[a-zA-Z]+\b', '', text)
    return text

df['clean_text'] = df['text'].apply(clean_text)

df['sentiment'] = df['text'].apply(lambda x: 1 if x.isupper() else 0)
df['neutral'] = df['text'].apply(lambda x: 1 if x.islower() else 0)
df['positive'] = df['text'].apply(lambda x: 1 if x.startswith('!') or x.startswith('?') else 0)
df['negative'] = df['text'].apply(lambda x: 1 if x.startswith('!') or x.startswith('?') else 0)

df['tokens'] = df['clean_text'].apply(tokenize)
df['lemmatized_tokens'] = df['tokens'].apply(lambda x: [lemmatizer.lemmatize(word) for word in x])
df['lemmatized_text'] = df['lemmatized_tokens'].apply(lambda x: ' '.join(x))
df['sentiment'] = df['text'].apply(lambda x: 1 if x.isupper() else 0)
df['neutral'] = df['text'].apply(lambda x: 1 if x.islower() else 0)
df['positive'] = df['text'].apply(lambda x: 1 if x.startswith('!') or x.startswith('?') else 0)
df['negative'] = df['text'].apply(lambda x: 1 if x.startswith('!') or x.startswith('?') else 0)

df.to_csv('social_media_cleaned.csv', index=False)
```

The screenshot shows a Jupyter Notebook interface with a terminal tab open. The terminal displays Python code being run to preprocess a social media text dataset. The code includes functions for cleaning text, extracting sentiment, tokenizing words, and applying lemmatization. The output is a CSV file named 'social_media_cleaned.csv'.

Output:

Task 5 – Financial Dataset Feature Engineering

Task:
Use AI to create a preprocessing script for a financial dataset.

Instructions:

- Handle missing values in stock price and volume.
- Create new features such as moving average (7-day, 30-day).
- Normalize continuous variables using StandardScaler.
- Encode categorical columns (sector, company_name).

Expected Output:

- A feature-engineered DataFrame with new indicators and normalized values for ML tasks.

Code:

```

import pandas as pd
from sklearn.preprocessing import StandardScaler, LabelEncoder

# Load Data
df = pd.read_csv('financial_features.csv')
print("Original Dataset:")
print(df)

# Handle missing values
df['stock_price'].fillna(df['stock_price'].mean(), inplace=True)
df['volume'].fillna(df['volume'].mean(), inplace=True)

# Create new features
df['MA_7'] = df['stock_price'].rolling(window=7, min_periods=1).mean()
df['MA_30'] = df['stock_price'].rolling(window=30, min_periods=1).mean() # 30-day reduced for small data

# Normalize continuous variables
scaler = StandardScaler()
df[['stock_price_scaled', 'volume_scaled', 'MA_7_scaled', 'MA_30_scaled']] = scaler.fit_transform(
    df[['stock_price', 'volume', 'MA_7', 'MA_30']])

# Encode categorical columns
le_sector = LabelEncoder()
le_company = LabelEncoder()

df['sector_encoded'] = le_sector.fit_transform(df['sector'])
df['company_encoded'] = le_company.fit_transform(df['company_name'])

print("Feature-Engineered Dataset:")
print(df)

# Save File
df.to_csv("financial_features_engineered.csv", index=False)
print("File saved as 'financial_features_engineered.csv'")

```

Output:

Agents Editor File Edit Selection View Go Run Terminal Help Task5.py - Lab 17.A - Cursor

Original Dataset:

	date	company_name	sector	stock_price	volume	M0_7	M0_30	stock_price_scaled	volume_scaled	M0_7_scaled	M0_30_scaled	sector_encoded	company_encdo
0	2025-01-01	AlphaCorp	Tech	120.5	980.0	130.5	120.5	109.489000	109.489000	-1.287061	-1.543851	-2.172938	-1.848888
1	2025-01-02	BetaLtd	Finance	98.3	850.0	98.3	98.3	98.3	98.3	-0.388770	-1.176238	-1.099829	1
2	2025-01-03	AlphaCorp	Tech	115.8	930.0	115.8	115.8	115.8	115.8	-0.889879	-0.889879	-1.176238	2
3	2025-01-04	BetaLtd	Finance	100.1	Nan	100.1	100.1	100.1	100.1	-0.189551	-0.189551	-0.213556	0.669985
4	2025-01-05	BetaLtd	Finance	100.2	920.0	100.2	100.2	100.2	100.2	-0.189551	-0.189551	-0.213556	0.669985
5	2025-01-06	AlphaCorp	Finance	120.8	910.0	120.8	120.8	120.8	120.8	-1.287061	-1.543851	-2.172938	-1.848888
6	2025-01-07	AlphaCorp	Energy	Nan	890.0	Nan	Nan	Nan	Nan	Nan	Nan	Nan	
7	2025-01-08	AlphaCorp	Tech	130.4	Nan	130.4	130.4	130.4	130.4	-1.287061	-1.543851	-2.172938	-1.848888
8	2025-01-09	AlphaCorp	Finance	100.8	880.0	100.8	100.8	100.8	100.8	-0.189551	-0.189551	-0.213556	0.669985
9	2025-01-10	GammaInc	Energy	99.9	Nan	99.9	Nan	99.9	Nan	-0.189551	-0.189551	-0.213556	0.669985

File saved as 'Financial_Feature_Engineered.csv'

(env) PS C:\Users\uriba\Desktop\MLC\Lab 17-A

File generated correctly

Task5.py - Lab 17.A - Cursor

Python Task1 Python Task2 Python Task3 Python Task4 Python Task5

Original Dataset:

	date	company_name	sector	stock_price	volume	M0_7	M0_30	stock_price_scaled	volume_scaled	M0_7_scaled	M0_30_scaled	sector_encoded	company_encdo
0	2025-01-01	AlphaCorp	Tech	120.5	980.0	130.5	120.5	109.489000	109.489000	-1.287061	-1.543851	-2.172938	-1.848888
1	2025-01-02	BetaLtd	Finance	98.3	850.0	98.3	98.3	98.3	98.3	-0.388770	-1.176238	-1.099829	1
2	2025-01-03	AlphaCorp	Tech	115.8	930.0	115.8	115.8	115.8	115.8	-0.889879	-0.889879	-1.176238	2
3	2025-01-04	BetaLtd	Finance	100.1	Nan	100.1	100.1	100.1	100.1	-0.189551	-0.189551	-0.213556	0.669985
4	2025-01-05	BetaLtd	Finance	100.2	920.0	100.2	100.2	100.2	100.2	-0.189551	-0.189551	-0.213556	0.669985
5	2025-01-06	AlphaCorp	Finance	120.8	910.0	120.8	120.8	120.8	120.8	-1.287061	-1.543851	-2.172938	-1.848888
6	2025-01-07	AlphaCorp	Energy	Nan	890.0	Nan	Nan	Nan	Nan	Nan	Nan	Nan	
7	2025-01-08	AlphaCorp	Tech	130.4	Nan	130.4	130.4	130.4	130.4	-1.287061	-1.543851	-2.172938	-1.848888
8	2025-01-09	AlphaCorp	Finance	100.8	880.0	100.8	100.8	100.8	100.8	-0.189551	-0.189551	-0.213556	0.669985
9	2025-01-10	GammaInc	Energy	99.9	Nan	99.9	Nan	99.9	Nan	-0.189551	-0.189551	-0.213556	0.669985

File saved as 'Financial_Feature_Engineered.csv'

(env) PS C:\Users\uriba\Desktop\MLC\Lab 17-A

File generated correctly

Task5.py - Lab 17.A - Cursor

Python Task1 Python Task2 Python Task3 Python Task4 Python Task5