

|  |   |  |                           |
|--|---|--|---------------------------|
| SCHOOL OF COMPUTER SCIENCE AND ARTIFICIAL INTELLIGENCE                                 |   | DEPARTMENT OF COMPUTER SCIENCE ENGINEERING |                           |
| Program Name: M. Tech/MSc/MCA  |   | Assignment Type: Lab                       | Academic Year: 2025-2026  |
| Course Coordinator Name  |   | Venkataramana Veeramsetty                  |                           |
| Course Code  |   | Course Title                               | AI Assisted Coding        |
| Year/Sem   | I/I   | Regulation                                 | R25                       |
| Date and Day of Assignment   |   | Time(s)                                    |                           |
| Duration   | 2 Hours   | Applicable to Batches                      |                           |
| Assignment Number: 17.3 (Present assignment number) / 24 (Total number of assignments) |   |  |                           |
|  |   |  |                           |
| Q.No.  | Question  |  | Expected Time to complete |
| 1  | <p><b>Lab 17 – AI for Data Processing: Data Cleaning and Preprocessing Scripts</b></p> <p><b>Lab Objectives:</b></p> <ul style="list-style-type: none"> <li>Learn how to clean raw datasets using AI-assisted Python scripting.</li> <li>Apply preprocessing techniques such as handling missing values, encoding categorical data, and normalization.</li> <li>Automate repetitive data-cleaning tasks with AI-generated code.</li> <li>Understand how preprocessing impacts model performance.</li> </ul> <hr/> <p><b>Task 1 – Social Media Data Cleaning</b></p> <p><b>Task: Clean raw social media posts dataset.</b></p> <p><b>Instructions:</b></p> <ul style="list-style-type: none"> <li>- Remove stopwords, punctuation, and special symbols from post text.</li> <li>- Handle missing values in likes and shares columns.</li> <li>- Convert timestamp to datetime and extract features (hour, weekday).</li> <li>- Detect and remove spam/duplicate posts.</li> </ul> <p><b>Expected Output:</b> A cleaned dataset with structured features for sentiment/engagement analysis.</p> <p><b>Code:</b></p> |  | Week 9 - Monday           |

assignment17 > Task1.py > ...

```

1  import pandas as pd
2  import numpy as np
3  import nltk
4  import re
5  from bs4 import BeautifulSoup
6  from nltk.corpus import stopwords
7
8  # SHOW ALL COLUMNS + ROWS
9  pd.set_option('display.max_columns', None)
10 pd.set_option('display.expand_frame_repr', True)
11 pd.set_option('display.max_rows', None)
12
13 # DOWNLOAD STOPWORDS
14 nltk.download("stopwords")
15 stop_words = set(str)(stopwords.words("english"))
16
17 # LOAD CSV
18 df = pd.read_csv(r"C:\Users\moham_219zzho\OneDrive\Desktop\Aipp\assignment17\social_
19
20 print("=== BEFORE CLEANING ===")
21 print(df)
22
23 # HANDLE MISSING VALUES
24 df['likes'] = df['likes'].fillna(0).astype(int)
25 df['shares'] = df['shares'].fillna(0).astype(int)
26 df['post_text'] = df['post_text'].fillna("")
27
28 # CLEAN TEXT FUNCTION
29 def clean_text(text):
30     text = BeautifulSoup(text, "html.parser").get_text()

```

assignment17 > Task1.py > ...

```

27
28 # CLEAN TEXT FUNCTION
29 def clean_text(text):
30     text = BeautifulSoup(text, "html.parser").get_text()
31     text = re.sub(r"http\S+|www\S+", "", text)
32     text = re.sub(r"[^A-Za-z\s]", " ", text)
33     text = text.lower()
34     words = text.split()
35     words = [w for w in words if w not in stop_words]
36     return " ".join(words)
37
38 df["clean_post"] = df["post_text"].apply(clean_text)
39
40 # TIMESTAMP FEATURES
41 df["timestamp"] = pd.to_datetime(df["timestamp"], errors="coerce")
42 df["hour"] = df["timestamp"].dt.hour
43 df["weekday"] = df["timestamp"].dt.day_name()
44
45 # KEEP ALL ROWS (no deletion)
46 df["post_length"] = df["clean_post"].apply(lambda x: len(x.split()))
47 df = df[df["post_length"] >= 0]
48
49 # FINAL OUTPUT (ALL COLUMNS)
50 print("\n=== AFTER CLEANING ===")
51 print(df[[
52     "post_id", "user", "post_text",
53     "likes", "shares",
54     "timestamp", "clean_post",
55     "hour", "weekday"
56 ]])

```

```

46 df["post_length"] = df["clean_post"].apply(lambda x: len(x.split()))
47 df = df[df["post_length"] >= 0]
48
49 # FINAL OUTPUT (ALL COLUMNS)
50 print("\n=== AFTER CLEANING ===")
51 print(df[[
52     "post_id", "user", "post_text",
53     "likes", "shares",
54     "timestamp", "clean_post",
55     "hour", "weekday"
56 ]])
57
58 print("\nTask Completed ✓")
59
60

```

output:

```

PS C:\Users\moham_219zzho\OneDrive\Desktop\Aipp> ^C
PS C:\Users\moham_219zzho\OneDrive\Desktop\Aipp> & C:/Users/moham_219zzho/anaconda3/
.exe c:/Users/moham_219zzho/OneDrive/Desktop/Aipp/assignment17/Task1.py
[nltk_data] Downloading package stopwords to
[nltk_data] C:\Users\moham_219zzho\AppData\Roaming\nltk_data...
[nltk_data] Package stopwords is already up-to-date!
=== BEFORE CLEANING ===

```

|    | post_id | user    | post_text                     | likes | shares | \ |
|----|---------|---------|-------------------------------|-------|--------|---|
| 0  | 1       | user_1  | This is a sample POST!!! #fun | 20.0  | 1.0    |   |
| 1  | 2       | user_2  | <html>Great Day!</html>       | 20.0  | 3.0    |   |
| 2  | 3       | user_3  | This is a sample POST!!! #fun | 20.0  | 1.0    |   |
| 3  | 4       | user_4  | <html>Great Day!</html>       | 100.0 | NaN    |   |
| 4  | 5       | user_5  | This is a sample POST!!! #fun | 20.0  | 5.0    |   |
| 5  | 6       | user_6  | <html>Great Day!</html>       | 5.0   | 1.0    |   |
| 6  | 7       | user_7  | This is a sample POST!!! #fun | 20.0  | 5.0    |   |
| 7  | 8       | user_8  | <html>Great Day!</html>       | 10.0  | 5.0    |   |
| 8  | 9       | user_9  | This is a sample POST!!! #fun | 5.0   | 5.0    |   |
| 9  | 10      | user_10 | <html>Great Day!</html>       | 5.0   | 5.0    |   |
| 10 | 11      | user_11 | This is a sample POST!!! #fun | NaN   | 1.0    |   |
| 11 | 12      | user_12 | <html>Great Day!</html>       | 10.0  | 3.0    |   |
| 12 | 13      | user_13 | This is a sample POST!!! #fun | 5.0   | 5.0    |   |
| 13 | 14      | user_14 | <html>Great Day!</html>       | 5.0   | 1.0    |   |
| 14 | 15      | user_15 | This is a sample POST!!! #fun | NaN   | 5.0    |   |
| 15 | 16      | user_16 | <html>Great Day!</html>       | NaN   | 1.0    |   |
| 16 | 17      | user_17 | This is a sample POST!!! #fun | NaN   | 3.0    |   |
| 17 | 18      | user_18 | <html>Great Day!</html>       | 20.0  | 3.0    |   |
| 18 | 19      | user_19 | This is a sample POST!!! #fun | 10.0  | NaN    |   |
| 19 | 20      | user_20 | <html>Great Day!</html>       | NaN   | NaN    |   |

```

timestamp
x 2025-01-01 00:00:00
  2025-01-01 06:00:00
  2025-01-01 12:00:00

```

```

4 2025-01-02 00:00:00
5 2025-01-02 06:00:00
6 2025-01-02 12:00:00
7 2025-01-02 18:00:00
8 2025-01-03 00:00:00
9 2025-01-03 06:00:00
10 2025-01-03 12:00:00
11 2025-01-03 18:00:00
12 2025-01-04 00:00:00
13 2025-01-04 06:00:00
14 2025-01-04 12:00:00
15 2025-01-04 18:00:00
16 2025-01-05 00:00:00
17 2025-01-05 06:00:00
18 2025-01-05 12:00:00
19 2025-01-05 18:00:00

```

=== AFTER CLEANING ===

|    | post_id | user    | post_text                     | likes | shares | \ |
|----|---------|---------|-------------------------------|-------|--------|---|
| 0  | 1       | user_1  | This is a sample POST!!! #fun | 20    | 1      |   |
| 1  | 2       | user_2  | <html>Great Day!</html>       | 20    | 3      |   |
| 2  | 3       | user_3  | This is a sample POST!!! #fun | 20    | 1      |   |
| 3  | 4       | user_4  | <html>Great Day!</html>       | 100   | 0      |   |
| 4  | 5       | user_5  | This is a sample POST!!! #fun | 20    | 5      |   |
| 5  | 6       | user_6  | <html>Great Day!</html>       | 5     | 1      |   |
| 6  | 7       | user_7  | This is a sample POST!!! #fun | 20    | 5      |   |
| 7  | 8       | user_8  | <html>Great Day!</html>       | 10    | 5      |   |
| 8  | 9       | user_9  | This is a sample POST!!! #fun | 5     | 5      |   |
| 9  | 10      | user_10 | <html>Great Day!</html>       | 5     | 5      |   |
| 10 | 11      | user_11 | This is a sample POST!!! #fun | 0     | 1      |   |
| 11 | 12      | user_12 | <html>Great Day!</html>       | 10    | 3      |   |
| 12 | 13      | user_13 | This is a sample POST!!! #fun | 5     | 5      |   |
| 13 | 14      | user_14 | <html>Great Day!</html>       | 5     | 1      |   |
| 14 | 15      | user_15 | This is a sample POST!!! #fun | 0     | 5      |   |
| 15 | 16      | user_16 | <html>Great Day!</html>       | 0     | 1      |   |
| 16 | 17      | user_17 | This is a sample POST!!! #fun | 0     | 3      |   |
| 17 | 18      | user_18 | <html>Great Day!</html>       | 20    | 3      |   |
| 18 | 19      | user_19 | This is a sample POST!!! #fun | 10    | 0      |   |
| 19 | 20      | user_20 | <html>Great Day!</html>       | 0     | 0      |   |

|    | timestamp           | clean_post      | hour | weekday   |
|----|---------------------|-----------------|------|-----------|
| 0  | 2025-01-01 00:00:00 | sample post fun | 0    | Wednesday |
| 1  | 2025-01-01 06:00:00 | great day       | 6    | Wednesday |
| 2  | 2025-01-01 12:00:00 | sample post fun | 12   | Wednesday |
| 3  | 2025-01-01 18:00:00 | great day       | 18   | Wednesday |
| 4  | 2025-01-02 00:00:00 | sample post fun | 0    | Thursday  |
| 5  | 2025-01-02 06:00:00 | great day       | 6    | Thursday  |
| 6  | 2025-01-02 12:00:00 | sample post fun | 12   | Thursday  |
| 7  | 2025-01-02 18:00:00 | great day       | 18   | Thursday  |
| 8  | 2025-01-03 00:00:00 | sample post fun | 0    | Friday    |
| 9  | 2025-01-03 06:00:00 | great day       | 6    | Friday    |
| 10 | 2025-01-03 12:00:00 | sample post fun | 12   | Friday    |
| 11 | 2025-01-03 18:00:00 | great day       | 18   | Friday    |
| 12 | 2025-01-04 00:00:00 | sample post fun | 0    | Saturday  |
| 13 | 2025-01-04 06:00:00 | great day       | 6    | Saturday  |
| 14 | 2025-01-04 12:00:00 | sample post fun | 12   | Saturday  |
| 15 | 2025-01-04 18:00:00 | great day       | 18   | Saturday  |
| 16 | 2025-01-05 00:00:00 | sample post fun | 0    | Sunday    |
| 17 | 2025-01-05 06:00:00 | great day       | 6    | Sunday    |
| 18 | 2025-01-05 12:00:00 | sample post fun | 12   | Sunday    |
| 19 | 2025-01-05 18:00:00 | great day       | 18   | Sunday    |

## 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.

**Code:**

```
Welcome | social_media.csv | financial_data_preprocessed.csv | Task1.py | Task2.py X
assignment17 > Task2.py > ...
1  import pandas as pd
2  import numpy as np
3
4  # =====
5  # LOAD DATA
6  # =====
7  df = pd.read_csv(r"C:\Users\moham_219zho\Downloads\financial_data.csv")
8
9
10 # Convert date to datetime & sort (important for time-series)
11 df['date'] = pd.to_datetime(df['date'])
12 df = df.sort_values('date').reset_index(drop=True)
13
14 print("=== BEFORE CLEANING ===")
15 print(df)
16
17 # =====
18 # 1. HANDLE MISSING VALUES
19 # =====
20
21 # closing_price → forward fill → backward fill
22 df['closing_price'] = df['closing_price'].fillna(method='ffill').fillna(method='bfill')
23
24 # volume → replace missing with 0
25 df['volume'] = df['volume'].fillna(0)
26
27 # =====
28 # 2. CREATE LAG FEATURES
29 # =====
30 df['return_1d'] = df['closing_price'].pct_change(1) # 1-day return
31 df['return_7d'] = df['closing_price'].pct_change(7) # 7-day return
32
33 # =====
34
35 # =====
36 df['log_volume'] = np.log1p(df['volume']) # log(1 + volume)
37
38 # =====
39 # 4. DETECT OUTLIERS (IQR METHOD)
40 # =====
41 Q1 = df['closing_price'].quantile(0.25)
42 Q3 = df['closing_price'].quantile(0.75)
43 IQR = Q3 - Q1
44
45 lower = Q1 - 1.5 * IQR
46 upper = Q3 + 1.5 * IQR
47
48 df['is_outlier'] = (df['closing_price'] < lower) | (df['closing_price'] > upper)
49
50 # =====
51 # FINAL OUTPUT
52 # =====
53 print("\n=== AFTER CLEANING ===")
54 print(df)
55
56 # Save processed dataset
57 df.to_csv("financial_data_preprocessed.csv", index=False)
58
59 print("\nPreprocessing Complete ✓ Time-series dataset ready for forecasting models.")
60
```

**output:**

```
PS C:\Users\moham_219zzho\OneDrive\Desktop\Aipp> & C:/Users/moham_219zzho/anaconda3/python.exe c:/Users/moham_219zzho/OneDrive/Desktop/Aipp/assignment17/Task2.py
4 2025-01-05 165.06 5000.0 0.002308 NaN 8.517393 False
5 2025-01-06 137.99 1500.0 -0.164001 NaN 7.313887 False
6 2025-01-07 151.00 5000.0 0.094282 NaN 8.517393 False
7 2025-01-08 151.00 1500.0 0.000000 0.152320 7.313887 False
8 2025-01-09 190.09 1500.0 0.258874 0.450626 7.313887 False
9 2025-01-10 135.16 1000.0 -0.288968 -0.022422 6.908755 False
10 2025-01-11 146.99 5000.0 0.087526 -0.107420 8.517393 False
11 2025-01-12 120.55 2000.0 -0.179876 -0.269660 7.601402 False
12 2025-01-13 168.97 0.0 0.401659 0.224509 0.000000 False
13 2025-01-14 149.75 1500.0 -0.113748 -0.008278 7.313887 False
14 2025-01-15 149.75 1000.0 0.000000 -0.008278 6.908755 False
15 2025-01-16 180.33 5000.0 0.204207 -0.051344 8.517393 False
16 2025-01-17 162.83 1000.0 -0.097044 0.204720 6.908755 False
17 2025-01-18 147.76 1500.0 -0.092551 0.005238 7.313887 False
18 2025-01-19 164.34 5000.0 0.112209 0.363252 8.517393 False
19 2025-01-20 122.91 0.0 -0.252099 -0.272593 0.000000 False
20 2025-01-21 101.11 2000.0 -0.177366 -0.324808 7.601402 False
21 2025-01-22 101.11 0.0 0.000000 -0.324808 0.000000 False
22 2025-01-23 156.71 1000.0 0.549896 -0.130982 6.908755 False
23 2025-01-24 191.67 1000.0 0.223087 0.177117 6.908755 False
24 2025-01-25 167.01 2000.0 -0.128659 0.130279 7.601402 False
25 2025-01-26 193.82 1500.0 0.160529 0.179384 7.313887 False
21 2025-01-22 101.11 0.0 0.000000 -0.324808 0.000000 False
22 2025-01-23 156.71 1000.0 0.549896 -0.130982 6.908755 False
23 2025-01-24 191.67 1000.0 0.223087 0.177117 6.908755 False
24 2025-01-25 167.01 2000.0 -0.128659 0.130279 7.601402 False
25 2025-01-26 193.82 1500.0 0.160529 0.179384 7.313887 False
23 2025-01-24 191.67 1000.0 0.223087 0.177117 6.908755 False
24 2025-01-25 167.01 2000.0 -0.128659 0.130279 7.601402 False
25 2025-01-26 193.82 1500.0 0.160529 0.179384 7.313887 False
24 2025-01-25 167.01 2000.0 -0.128659 0.130279 7.601402 False
25 2025-01-26 193.82 1500.0 0.160529 0.179384 7.313887 False
25 2025-01-26 193.82 1500.0 0.160529 0.179384 7.313887 False
26 2025-01-27 116.65 2000.0 -0.398153 -0.050932 7.601402 False
27 2025-01-28 145.41 2000.0 0.246550 0.438137 7.601402 False
28 2025-01-29 145.41 2000.0 0.000000 0.438137 7.601402 False
26 2025-01-27 116.65 2000.0 -0.398153 -0.050932 7.601402 False
26 2025-01-27 116.65 2000.0 -0.398153 -0.050932 7.601402 False
27 2025-01-28 145.41 2000.0 0.246550 0.438137 7.601402 False
28 2025-01-29 145.41 2000.0 0.000000 0.438137 7.601402 False
29 2025-01-30 123.02 1500.0 -0.153978 -0.214983 7.313887 False
27 2025-01-28 145.41 2000.0 0.246550 0.438137 7.601402 False
28 2025-01-29 145.41 2000.0 0.000000 0.438137 7.601402 False
29 2025-01-30 123.02 1500.0 -0.153978 -0.214983 7.313887 False
29 2025-01-30 123.02 1500.0 -0.153978 -0.214983 7.313887 False
```

### 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.

**Code:**

assignment17 > task3.py > ...

```
1 import pandas as pd
2 from sklearn.preprocessing import StandardScaler, LabelEncoder
3 import os
4
5 print("Current working directory:", os.getcwd())
6
7 # -----
8 # 1. Load dataset
9 # -----
10 df = pd.read_csv(r"C:\Users\moham_219zzho\Downloads\iot_sensor.csv")
11
12 df['timestamp'] = pd.to_datetime(df['timestamp'])
13 df = df.sort_values('timestamp')
14
15 # -----
16 # 2. BEFORE CLEANING
17 # -----
18 before_cleaning = df.copy()
19 print("\n===== BEFORE CLEANING =====")
20 print(before_cleaning.head(), "\n")
21
22 # -----
23 # 3. CLEANING STEPS
24 # -----
25 df[['temperature', 'humidity']] = df[['temperature', 'humidity']].ffill()
```

| Problems | Output              | Debug Console | Terminal | Ports | Python: Task3 - assignment17 |
|----------|---------------------|---------------|----------|-------|------------------------------|
| 2        | 2025-02-01 02:00:00 | S1            | 24.0     | 50.0  |                              |
| 3        | 2025-02-01 03:00:00 | S2            | 24.0     | NaN   |                              |
| 4        | 2025-02-01 04:00:00 | S3            | 23.0     | 42.0  |                              |

===== AFTER CLEANING =====

assignment17 > task3.py > ...

```
23 # 3. CLEANING STEPS
24 # -----
25 df[['temperature', 'humidity']] = df[['temperature', 'humidity']].ffill()
26
27 df['temperature_clean'] = df['temperature'].rolling(window=10, min_periods=1)
28 df['humidity_clean'] = df['humidity'].rolling(window=10, min_periods=1).mean()
29
30 scaler = StandardScaler()
31 df[['temp_scaled', 'hum_scaled']] = scaler.fit_transform(
32     df[['temperature_clean', 'humidity_clean']]
33 )
34
35 encoder = LabelEncoder()
36 df['sensor_encoded'] = encoder.fit_transform(df['sensor_id'])
37
38 # -----
39 # 4. AFTER CLEANING
40 # -----
41 after_cleaning = df[['timestamp',
42     'sensor_id', 'sensor_encoded',
43     'temperature', 'temperature_clean', 'temp_scaled',
44     'humidity', 'humidity_clean', 'hum_scaled']]
45
46 print("===== AFTER CLEANING =====")
47 print(after_cleaning.head(), "\n")
```

```

after_cleaning = df[['timestamp',
                    'sensor_id', 'sensor_encoded',
                    'temperature', 'temperature_clean', 'temp_scaled',
                    'humidity', 'humidity_clean', 'hum_scaled']]

print("===== AFTER CLEANING =====")
print(after_cleaning.head(), "\n")

# -----
# 5. SAVE CLEANED FILE IN ASSIGNMENT FOLDER
# -----
save_local = "cleaned_iot_data.csv"
after_cleaning.to_csv(save_local, index=False)

print(f"✓ Cleaned file saved in current folder at: {os.path.abspath(save_local)}")

# -----
# 6. ALSO SAVE IN DOWNLOADS FOLDER
# -----
save_downloads = r"C:\Users\moham_219zzho\Downloads\cleaned_iot_data.csv"
after_cleaning.to_csv(save_downloads, index=False)

print(f"✓ Cleaned file also saved in Downloads")

```

output:

```

cleaned_iot_data.csv
PS C:\Users\moham_219zzho\OneDrive\Desktop\Aipp\assignment17> & C:/Users/moham_219zzho/OneDrive/Desktop/Aipp/assignment17/Task3.py
Current working directory: C:\Users\moham_219zzho\OneDrive\Desktop\Aipp\assignment17

===== BEFORE CLEANING =====
   timestamp sensor_id temperature humidity
0 2025-02-01 00:00:00      S2       24.0      40.0
1 2025-02-01 01:00:00      S3       30.0       NaN
2 2025-02-01 02:00:00      S1       24.0      50.0
3 2025-02-01 03:00:00      S2       24.0      NaN
4 2025-02-01 04:00:00      S3       23.0      42.0

===== AFTER CLEANING =====
   timestamp sensor_id sensor_encoded ... humidity humidity_clean hum_scaled
0 2025-02-01 00:00:00      S2          1 ...      40.0      40.000000 -2.576213
1 2025-02-01 01:00:00      S3          2 ...      40.0      40.000000 -2.576213
2 2025-02-01 02:00:00      S1          0 ...      50.0      43.333333 -0.520521
3 2025-02-01 03:00:00      S2          1 ...      50.0      45.000000  0.507325
0 2025-02-01 00:00:00      S2          1 ...      40.0      40.000000 -2.576213
1 2025-02-01 01:00:00      S3          2 ...      40.0      40.000000 -2.576213
2 2025-02-01 02:00:00      S1          0 ...      50.0      43.333333 -0.520521
3 2025-02-01 03:00:00      S2          1 ...      50.0      45.000000  0.507325
4 2025-02-01 04:00:00      S3          2 ...      42.0      44.400000  0.137301

```

## 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.

Code:

```
assignment17 > Task4.py > ...
1  import pandas as pd
2  import numpy as np
3  import os
4  from bs4 import BeautifulSoup
5  from sklearn.feature_extraction.text import TfidfVectorizer
6
7  # -----
8  # 1. Load Data
9  # -----
10 file_path = r"C:\Users\moham_219zzho\Downloads\movie_reviews-1.csv"
11 df = pd.read_csv(file_path)
12
13 # -----
14 # 2. BEFORE SUMMARY
15 # -----
16 print("\n===== BEFORE CLEANING =====\n")
17 print(df.describe(include='all'))
18
19 # -----
20 # 3. TEXT CLEANING
21 # -----
22 def clean_text(text):
23     if pd.isna(text):
24         return ""
25     text = text.lower()
26     text = BeautifulSoup(text, "html.parser").get_text()
27     return text
28
29 df['clean_review'] = df['review_text'].apply(clean_text)
30
31 # -----
32 # 4. TF-IDF ENCODING
```

```

30
31 # -----
32 # 4. TF-IDF ENCODING
33 # -----
34 vectorizer = TfidfVectorizer(stop_words='english', max_features=500)
35 tfidf_matrix = vectorizer.fit_transform(df['clean_review'])
36
37 print("\nTF-IDF Shape:", tfidf_matrix.shape)
38
39 # -----
40 # 5. FIX MISSING RATINGS
41 # -----
42 median_rating = df['rating'].median()
43 df['rating'] = df['rating'].fillna(median_rating)
44
45 # -----
46 # 6. NORMALIZE RATINGS
47 # -----
48 df['rating_normalized'] = df['rating'] / 10
49
50 # -----
51 # 7. AFTER SUMMARY
52 # -----
53 print("\n===== AFTER CLEANING =====\n")
54 print(df.describe(include='all'))
55
56 # -----
57 # 8. SAVE CLEANED OUTPUT
58 # -----
59 # Save inside same folder as Task4.py → assignment17/output/
60 current_directory = os.path.dirname(os.path.abspath(__file__))
61 output_folder = os.path.join(current_directory, "output")
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999

```

**Output:**

```
PS C:\Users\moham_219zzho\OneDrive\Desktop\Aipp\assignment17> & C:/Users/moham_219zzho/anaconda3/python.exe c:/
/moham_219zzho/OneDrive/Desktop/Aipp/assignment17/Task4.py
PS C:\Users\moham_219zzho\OneDrive\Desktop\Aipp\assignment17> & C:/Users/moham_219zzho/anaconda3/python.exe c:/
/moham_219zzho/OneDrive/Desktop/Aipp/assignment17/Task4.py

===== BEFORE CLEANING =====

      review_id      review_text      rating
count  15.000000           15  13.000000
unique    NaN           2    NaN
top      NaN  <p>Amazing movie!</p>    NaN
freq      NaN           8    NaN
mean     8.000000           NaN  6.461538
std      4.472136           NaN  2.933013
min      1.000000           NaN  2.000000
25%      4.500000           NaN  5.000000
50%      8.000000           NaN  8.000000
75%     11.500000           NaN  8.000000
max     15.000000           NaN 10.000000

TF-IDF Shape: (15, 5)

===== AFTER CLEANING =====

      review_id      review_text      rating      clean_review      rating_normalized
count  15.000000           15  15.000000           15  15.000000
unique    NaN           2    NaN           2    NaN
top      NaN  <p>Amazing movie!</p>    NaN  amazing movie!    NaN
freq      NaN           8    NaN           8    NaN
mean     8.000000           NaN  6.666667           NaN  0.666667
std      4.472136           NaN  2.768875           NaN  0.276887
min      1.000000           NaN  2.000000           NaN  0.200000
25%      4.500000           NaN  5.000000           NaN  0.500000
mean     8.000000           NaN  6.666667           NaN  0.666667
std      4.472136           NaN  2.768875           NaN  0.276887
min      1.000000           NaN  2.000000           NaN  0.200000
25%      4.500000           NaN  5.000000           NaN  0.500000
std      4.472136           NaN  2.768875           NaN  0.276887
min      1.000000           NaN  2.000000           NaN  0.200000
25%      4.500000           NaN  5.000000           NaN  0.500000
min      1.000000           NaN  2.000000           NaN  0.200000
25%      4.500000           NaN  5.000000           NaN  0.500000
50%      8.000000           NaN  8.000000           NaN  0.800000
75%     11.500000           NaN  8.000000           NaN  0.800000
max     15.000000           NaN 10.000000           NaN  1.000000

50%      8.000000           NaN  8.000000           NaN  0.800000
75%     11.500000           NaN  8.000000           NaN  0.800000
max     15.000000           NaN 10.000000           NaN  1.000000

max     15.000000           NaN 10.000000           NaN  1.000000

File saved successfully at:
```