Vidyavardhini's College of Engineering and Technology, Vasai
Department of Computer Science & Engineering (Data Science)

Academic Year: 2025-26

Subject: AI&ML in Healthcare

Name:	Bhaskar Mulik
Roll No & Branch:	29 - COMPS
Class/Sem:	BE/VII
Experiment No.:	01
Title:	Clean, Integrate and Transform Electronic Healthcare Records.
Date of Performance:	10-07-25
Date of Submission:	18-07-25
Marks:	
Sign of Faculty:	

Aim: To collect, clean, integrate and transform Electronic Healthcare Records.

Objective: To develop a robust and efficient data pipeline that collects, cleans, integrates, and transforms healthcare data from diverse sources, ensuring data accuracy, privacy compliance, and usability. This pipeline will facilitate comprehensive and reliable analyses, enabling informed decision-making and insights to drive improvements in healthcare delivery, patient outcomes, and research endeavors.

Theory: The Disease Symptoms and Patient Profile Dataset serves as a pivotal gateway to unraveling the intricate web of diseases. By meticulously intertwining symptomatology, patient demographics, and health metrics, this dataset presents an unprecedented opportunity to discern the concealed correlations within medical conditions. With symptoms such as fever, cough, fatigue, and difficulty breathing interwoven alongside crucial variables like age, gender, blood pressure, and cholesterol levels, this dataset holds the promise of unearthing latent patterns. A transformative resource for medical researchers, healthcare professionals, and data enthusiasts alike, its exploration promises to unveil distinctive symptom profiles and initiate an enthralling expedition into the realm of ailments. As users navigate this treasure trove, a profound revolution in healthcare comprehension beckons, destined to reshape our understanding of medical intricacies and pave the way for enhanced diagnostics and treatment strategies.
At the heart of the Disease Symptoms and Patient Profile Dataset lies a reservoir of invaluable insights waiting to reshape the landscape of healthcare knowledge. This meticulously curated compilation of symptoms, patient characteristics, and health indicators offers an unprecedented vantage point into the complex interplay of factors underlying various diseases. As medical researchers delve into the depths of this dataset, they embark on a journey of discovery, unveiling hidden relationships that have the potential to redefine diagnostic paradigms and treatment approaches. By deciphering the intricate tapestry woven from fever, cough, fatigue, and difficulty breathing, intricately interwoven with age, gender, blood pressure, and cholesterol levels, a new era of personalized and targeted healthcare strategies is on the horizon. This dataset not only promises to revolutionize medical research but also empowers healthcare professionals to make informed decisions that can lead to improved patient outcomes and a brighter future for the field of medicine.

Program and output

```
import pandas as pd

from sklearn.preprocessing import LabelEncoder, StandardScaler

from sklearn.impute import SimpleImputer

# 1. Data Collection (Simulated for demonstration)

# In a real-world scenario, you would load data from various sources (CSV, databases, APIs)

data = {

    'PatientID': [1, 2, 3, 4, 5, 6, 7, 8, 9, 10],

    'Age': [45, 62, 30, 55, 70, 28, 50, 68, 35, 42],

    'Gender': ['Male', 'Female', 'Female', 'Male', 'Female', 'Male', 'Female', 'Male', 'Female', 'Male'],

    'Fever': ['Yes', 'No', 'Yes', 'No', 'Yes', 'No', 'Yes', 'No', 'Yes', 'No'],

    'Cough': ['Yes', 'Yes', 'No', 'Yes', 'No', 'Yes', 'Yes', 'No', 'Yes', 'No'],

    'Fatigue': ['No', 'Yes', 'No', 'Yes', 'No', 'Yes', 'No', 'Yes', 'No', 'Yes'],

    'DifficultyBreathing': ['No', 'No', 'Yes', 'No', 'Yes', 'No', 'Yes', 'No', 'Yes', 'No'],

    'BloodPressure': [120, 140, 110, 130, 150, 115, 125, 145, 118, 122],

    'Cholesterol': [200, 240, 180, 220, 260, 190, 210, 250, 185, 205],

    'Diagnosis': ['Flu', 'Pneumonia', 'Asthma', 'Bronchitis', 'Pneumonia', 'Flu', 'Asthma', 'Bronchitis', 'Flu',
'Asthma'],

    'Hospital_Visit_Date': ['2023-01-15', '2023-01-20', '2023-01-22', '2023-02-01', '2023-02-05',
'2023-02-10', '2023-02-12', '2023-02-18', '2023-02-20', '2023-02-25']

}

df = pd.DataFrame(data)
```

```
# Simulate a second source with some missing data and different column names
data_source2 = {
    'ID': [11, 12, 13, 14, 15],
    'Age_Years': [38, None, 65, 52, 48],
    'Sex': ['Female', 'Male', 'Female', 'Male', 'Female'],
    'Has_Fever': [1, 0, 1, 0, 1],
    'BP_Systolic': [128, 135, None, 130, 120],
    'Cholesterol_Level': [210, 230, 200, 225, None],
    'Condition': ['Flu', 'Bronchitis', 'Pneumonia', 'Flu', 'Asthma']
}
df_source2 = pd.DataFrame(data_source2)
print("--- Original DataFrames ---")
print("DataFrame 1:")
print(df.head())
print("\nDataFrame 2:")
print(df_source2.head())
print("-" * 30)
# 2. Data Cleaning
# Handle missing values
imputer_age = SimpleImputer(strategy='mean')
```

```
df_source2['Age_Years'] = imputer_age.fit_transform(df_source2[['Age_Years']])

imputer_bp = SimpleImputer(strategy='mean')

df_source2['BP_Systolic'] = imputer_bp.fit_transform(df_source2[['BP_Systolic']])

imputer_cholesterol = SimpleImputer(strategy='mean')

df_source2['Cholesterol_Level'] = imputer_cholesterol.fit_transform(df_source2[['Cholesterol_Level']])

# Convert 'Yes'/'No' to numerical (0/1) for symptoms in df

for col in ['Fever', 'Cough', 'Fatigue', 'DifficultyBreathing']:

    df[col] = df[col].map({'Yes': 1, 'No': 0})

# Convert 'Has_Fever' (0/1) to 'Fever' (0/1) in df_source2

df_source2['Fever'] = df_source2['Has_Fever']

df_source2.drop(columns=['Has_Fever'], inplace=True)

print("\n--- After Cleaning Missing Values and Initial Conversions ---")

print("DataFrame 1:")

print(df.head())

print("\nDataFrame 2:")

print(df_source2.head())

print("-" * 30)

# 3. Data Integration

# Rename columns in df_source2 to match df for integration

df_source2 = df_source2.rename(columns={

    'ID': 'PatientID',
```

```
    'Age_Years': 'Age',
    'Sex': 'Gender',
    'BP_Systolic': 'BloodPressure',
    'Cholesterol_Level': 'Cholesterol',
    'Condition': 'Diagnosis'
})
# Add missing symptom columns to df_source2 and fill with 0 (assuming absence if not explicitly stated)
for col in ['Cough', 'Fatigue', 'DifficultyBreathing']:
    if col not in df_source2.columns:
        df_source2[col] = 0
# Select and reorder columns in df_source2 to match df
df_source2 = df_source2[df.columns.tolist()]
# Concatenate the dataframes
integrated_df = pd.concat([df, df_source2], ignore_index=True)
print("\n--- After Integration (Concatenation) ---")
print(integrated_df.head(12))
print("-" * 30)
# 4. Data Transformation
# Convert categorical features to numerical using Label Encoding
label_encoder_gender = LabelEncoder()
```

```
integrated_df['Gender_Encoded'] = label_encoder_gender.fit_transform(integrated_df['Gender'])

label_encoder_diagnosis = LabelEncoder()

integrated_df['Diagnosis_Encoded'] = label_encoder_diagnosis.fit_transform(integrated_df['Diagnosis'])

# Feature Scaling for numerical features

scaler = StandardScaler()

numerical_cols = ['Age', 'BloodPressure', 'Cholesterol']

integrated_df[numerical_cols] = scaler.fit_transform(integrated_df[numerical_cols])

# Convert 'Hospital_Visit_Date' to datetime objects and extract features

integrated_df['Hospital_Visit_Date'] = pd.to_datetime(integrated_df['Hospital_Visit_Date'])

integrated_df['Visit_Month'] = integrated_df['Hospital_Visit_Date'].dt.month

integrated_df['Visit_DayOfWeek'] = integrated_df['Hospital_Visit_Date'].dt.dayofweek

# Drop original categorical columns if encoded versions are preferred for modeling

transformed_df = integrated_df.drop(columns=['Gender', 'Diagnosis', 'Hospital_Visit_Date'])

print("\n--- After Transformation ---")

print(transformed_df.head(12))

print("\nDataFrame Info after Transformation:")

print(transformed_df.info())

print("\nUnique values for 'Gender_Encoded':", transformed_df['Gender_Encoded'].unique())

print("Unique values for 'Diagnosis_Encoded':", transformed_df['Diagnosis_Encoded'].unique())

print("-" * 30)

print("\n--- Final Transformed Dataset Sample ---")
```

```
print(transformed_df.sample(5))
```

Output:

```
--- Original DataFrames ---

DataFrame 1:

   PatientID  Age  Gender Fever Cough Fatigue DifficultyBreathing  BloodPressure  Cholesterol
Diagnosis Hospital_Visit_Date

0          1   45    Male   Yes   Yes      No                 No          120         200         Flu          2023-01-15

1          2   62  Female    No   Yes     Yes                 No          140         240    Pneumonia
2023-01-20

2          3   30  Female   Yes    No      No                Yes          110         180      Asthma
2023-01-22

3          4   55    Male    No   Yes     Yes                 No          130         220   Bronchitis          2023-02-01

4          5   70  Female   Yes    No      No                Yes          150         260    Pneumonia
2023-02-05

DataFrame 2:

   ID  Age_Years     Sex  Has_Fever  BP_Systolic  Cholesterol_Level   Condition

0  11       38.0  Female          1        128.0              210.0         Flu

1  12        NaN    Male          0        135.0              230.0  Bronchitis

2  13       65.0  Female          1          NaN              200.0   Pneumonia

3  14       52.0    Male          0        130.0              225.0         Flu

4  15       48.0  Female          1        120.0                NaN   Asthma

------------------------------
```

--- After Cleaning Missing Values and Initial Conversions ---

DataFrame 1:

PatientID Age Gender Fever Cough Fatigue DifficultyBreathing BloodPressure Cholesterol Diagnosis Hospital_Visit_Date

0 1 45 Male 1 1 0 0 120 200 Flu 2023-01-15

1 2 62 Female 0 1 1 0 140 240 Pneumonia 2023-01-20

2 3 30 Female 1 0 0 1 110 180 Asthma 2023-01-22

3 4 55 Male 0 1 1 0 130 220 Bronchitis 2023-02-01

4 5 70 Female 1 0 0 1 150 260 Pneumonia 2023-02-05

DataFrame 2:

ID Age_Years Sex BP_Systolic Cholesterol_Level Condition Fever

0 11 38.0 Female 128.0 210.0 Flu 1

1 12 49.4 Male 135.0 230.0 Bronchitis 0

2 13 65.0 Female 128.2 200.0 Pneumonia 1

3 14 52.0 Male 130.0 225.0 Flu 0

4 15 48.0 Female 120.0 216.2 Asthma 1

--- After Integration (Concatenation) ---

PatientID Age Gender Fever Cough Fatigue DifficultyBreathing BloodPressure Cholesterol Diagnosis Hospital_Visit_Date

0 1 45.0 Male 1 1 0 0 120.0 200.0 Flu 2023-01-15

1 2 62.0 Female 0 1 1 0 140.0 240.0 Pneumonia 2023-01-20

2 3 30.0 Female 1 0 0 1 110.0 180.0 Asthma 2023-01-22

3 4 55.0 Male 0 1 1 0 130.0 220.0 Bronchitis 2023-02-01

4 5 70.0 Female 1 0 0 1 150.0 260.0 Pneumonia 2023-02-05

5 6 28.0 Male 0 1 1 0 115.0 190.0 Flu 2023-02-10

6 7 50.0 Female 1 1 0 1 125.0 210.0 Asthma 2023-02-12

7 8 68.0 Male 0 0 1 0 145.0 250.0 Bronchitis 2023-02-18

8 9 35.0 Female 1 1 0 1 118.0 185.0 Flu 2023-02-20

9 10 42.0 Male 0 0 1 0 122.0 205.0 Asthma 2023-02-25

10 11 38.0 Female 1 0 0 0 128.0 210.0 Flu NaT

11 12 49.4 Male 0 0 0 0 135.0 230.0 Bronchitis NaT

--- After Transformation ---

PatientID Age Fever Cough Fatigue DifficultyBreathing BloodPressure Cholesterol Gender_Encoded Diagnosis_Encoded Visit_Month Visit_DayOfWeek

0 1 -0.347072 1 1 0 0 -0.671373 -0.575677 1 2 1.0 6.0

1	2	1.037466	0	1	1	0	0.999661	1.025345	0	3	1.0
4.0											
2	3	-1.543592	1	0	0	1	-1.428906	-1.376189	0	0	1.0
6.0											
3	4	0.395726	0	1	1	0	0.155127	0.224834	1	1	2.0
4.0											
4	5	1.678732	1	0	0	1	1.844195	1.825857	0	3	2.0
0.0											
5	6	-1.704258	0	1	1	0	-1.049440	-0.975933	1	2	2.0
5.0											
6	7	0.084400	1	1	0	1	-0.251106	-0.175422	0	0	2.0
0.0											
7	8	1.518065	0	0	1	0	1.424928	1.425599	1	1	2.0
5.0											
8	9	-1.142099	1	1	0	1	-0.809289	-1.176061	0	2	2.0
3.0											
9	10	-0.587905	0	0	1	0	-0.491263	-0.375550	1	0	2.0
5.0											
10	11	-0.892900	1	0	0	0	-0.080599	-0.175422	0	2	NaN
NaN											
11	12	-0.040004	0	0	0	0	0.490799	0.625090	1	1	NaN
NaN											

DataFrame Info after Transformation:

<class 'pandas.core.frame.DataFrame'>

RangeIndex: 15 entries, 0 to 14

Data columns (total 12 columns):

```
 #   Column              Non-Null Count  Dtype
---  ------              --------------  -----
 0   PatientID           15 non-null     int64
 1   Age                 15 non-null     float64
 2   Fever               15 non-null     int64
 3   Cough               15 non-null     int64
 4   Fatigue             15 non-null     int64
 5   DifficultyBreathing 15 non-null     int64
 6   BloodPressure       15 non-null     float64
 7   Cholesterol         15 non-null     float64
 8   Gender_Encoded      15 non-null     int32
 9   Diagnosis_Encoded   15 non-null     int32
 10  Visit_Month         10 non-null     float64
 11  Visit_DayOfWeek     10 non-null     float64
dtypes: float64(5), int32(2), int64(5)
memory usage: 1.4 KB
Unique values for 'Gender_Encoded': [1 0]
Unique values for 'Diagnosis_Encoded': [2 3 0 1]
------------------------------
--- Final Transformed Dataset Sample ---
```

PatientID	Age	Fever	Cough	Fatigue	DifficultyBreathing	BloodPressure	Cholesterol	Gender_Encoded	Diagnosis_Encoded	Visit_Month	Visit_DayOfWeek
2	3	-1.543592	1	0	0	1	-1.428906	-1.376189	0	0	1.0	6.0
12	13	1.357400	1	0	0	0	0.000000	-0.575677	0	3	NaN	NaN
6	7	0.084400	1	1	0	1	-0.251106	-0.175422	0	0	2.0	0.0
11	12	-0.040004	0	0	0	0	0.490799	0.625090	1	1	NaN	NaN
9	10	-0.587905	0	0	1	0	-0.491263	-0.375550	1	0	2.0	5.0

Conclusion:

The Disease Symptoms and Patient Profile Dataset experiment has unveiled intricate connections between symptoms, demographics, and health indicators. This dataset offers a powerful lens into disease dynamics, enabling researchers to uncover hidden patterns and correlations. The integration of fever, cough, fatigue, difficulty breathing, age, gender, blood pressure, and cholesterol levels has provided profound insights for personalized healthcare strategies. This endeavor promises to advance diagnostic accuracy, treatment efficacy, and patient outcomes, revolutionizing medical research and practice. As a result, this dataset stands as a transformative tool, paving the way for a more comprehensive understanding of diseases and ushering in a new era of informed healthcare decision-making.