Experiment No: 1

Aim: Collect , Clean , Integrate and Transform Healthcare Data based on specific dataset.

Theory:

1. Dataset

A dataset is a collection of data, often presented in tabular form, where each column represents a particular variable, and each row corresponds to a record. Datasets are fundamental units of data used for analysis, training machine learning models, and other data-driven tasks.

Key Components of a Dataset:

- Attributes/Features: The columns in the dataset, representing various characteristics or variables.
- Records/Instances: The rows in the dataset, each representing a single observation or data point.
- Data Types: The nature of the data in each column, such as numerical, categorical, date/time, etc.

2. Collection of Dataset

Data collection involves gathering information from various sources to create a dataset. This can be done through several methods, including:

- Surveys and Questionnaires: Collecting responses from individuals.
- Sensors and IoT Devices: Gathering real-time data from physical devices.
- Web Scraping: Extracting data from websites.
- APIs: Pulling data from other software applications.
- Databases: Extracting data stored in relational or non-relational databases.
- Manual Entry: Direct input of data by individuals.

Considerations for Data Collection:

- Data Quality: Ensuring the data is accurate, complete, and reliable.
- Ethics and Privacy: Collecting data ethically and respecting privacy concerns.
- Sampling: Choosing a representative sample to ensure the data collected is useful for analysis.

3. Cleaning the Dataset

Data cleaning is the process of detecting and correcting (or removing) errors and inconsistencies in the dataset to improve its quality. This is a critical step before any data analysis or modelling.

Steps in Data Cleaning:
- Handling Missing Values: Methods include removing records with missing values, imputing missing values, or using algorithms that handle missing data.
- Removing Duplicates: Identifying and removing duplicate records to avoid redundancy.
- Outlier Detection: Identifying and handling outliers that may skew the analysis.
- Data Type Correction: Ensuring that data types of columns are appropriate (e.g., converting strings to dates).
- Normalisation and Standardization: Scaling numerical data to a standard range or distribution.
- Error Correction: Fixing any inaccuracies or inconsistencies in the data entries.

4. Integration

Data integration is the process of combining data from different sources to provide a unified view. This is essential when data is scattered across multiple databases or formats.

Steps in Data Integration:
- Data Preprocessing: Cleaning and preparing data from each source.
- Schema Integration: Merging schemas from different data sources to create a coherent structure.
- Data Transformation: Standardizing data formats and structures.
- Data Loading: Consolidating data into a single repository or data warehouse.

Challenges in Data Integration:
- Heterogeneous Data: Handling different formats, structures, and data types.
- Data Redundancy: Managing duplicate data entries.
- Data Consistency: Ensuring data remains consistent across integrated sources.

5. Transformation

Data transformation involves converting data into a suitable format or structure for analysis. This step is crucial for ensuring that the data is compatible with the analysis tools and methods to be used.

Common Data Transformation Techniques:
- Normalization: Scaling data to a specific range, typically [0, 1].
- Standardization: Transforming data to have a mean of 0 and a standard deviation of 1.
- Aggregation: Summarizing data, such as computing averages or totals.
- Encoding Categorical Data: Converting categorical data into numerical formats (e.g., one-hot encoding, label encoding).
- Feature Engineering: Creating new features from existing data to improve model performance.
- Data Reduction: Reducing the volume but producing the same analytical results, such as through dimensionality reduction techniques.

Transformation Process:
- Identify the required transformations based on the analysis goals.

- Apply transformations systematically and verify their correctness.
- Ensure transformed data retains its integrity and usability for subsequent analysis.

Code:

```
import pandas as pd
import numpy as np

df = pd.read_csv('obesity_data.csv')

df.head()
```

	Age	Gender	Height	Weight	BMI	PhysicalActivityLevel	ObesityCategory
0	56	Male	173.575262	71.982051	23.891783	4	Normal weight
1	69	Male	164.127306	89.959256	33.395209	2	Obese
2	46	Female	168.072202	72.930629	25.817737	4	Overweight
3	32	Male	168.459633	84.886912	29.912247	3	Overweight
4	60	Male	183.568568	69.038945	20.487903	3	Normal weight

```
df.columns
```

```
Index(['Age', 'Gender', 'Height', 'Weight', 'BMI', 'PhysicalActivityLevel',
       'ObesityCategory'],
      dtype='object')
```

```
df.isnull().sum()
```

```
Age                     0
Gender                  0
Height                  0
Weight                  0
BMI                     0
PhysicalActivityLevel   0
ObesityCategory         0
dtype: int64
```

```
df['Gender'] = df['Gender'].map({'Male': 0, 'Female': 1})
df
```

	Age	Gender	Height	Weight	BMI	PhysicalActivityLevel	ObesityCategory
0	56	0	173.575262	71.982051	23.891783	4	Normal weight
1	69	0	164.127306	89.959256	33.395209	2	Obese
2	46	1	168.072202	72.930629	25.817737	4	Overweight
3	32	0	168.459633	84.886912	29.912247	3	Overweight
4	60	0	183.568568	69.038945	20.487903	3	Normal weight
...	...	...	...	...	...	...	...
995	18	0	155.588674	64.103182	26.480345	4	Overweight
996	35	1	165.076490	97.639771	35.830783	1	Obese
997	49	1	156.570956	78.804284	32.146036	1	Obese
998	64	0	164.192222	57.978115	21.505965	4	Normal weight
999	66	1	178.537130	74.962164	23.517168	1	Normal weight

1000 rows × 7 columns

```
from sklearn.preprocessing import StandardScaler, LabelEncoder
label_encoder = LabelEncoder()
df['ObesityCategory'] = label_encoder.fit_transform(df['ObesityCategory'])
df['PhysicalActivityLevel'] = label_encoder.fit_transform(df['PhysicalActivityLevel'])

df.head()
```

	Age	Gender	Height	Weight	BMI	PhysicalActivityLevel	ObesityCategory
0	0.339295	0	0.341864	0.050076	-0.160970	3	0
1	1.057320	0	-0.574985	1.209739	1.374115	1	1
2	-0.213033	1	-0.192164	0.111266	0.150129	3	2
3	-0.986291	0	-0.154567	0.882535	0.811514	2	2
4	0.560226	0	1.311635	-0.139776	-0.710797	2	0

```
scaler = StandardScaler()
numeric_columns = ['Age', 'Height', 'Weight', 'BMI']
df[numeric_columns] = scaler.fit_transform(df[numeric_columns])
```

	Age	Gender	Height	Weight	BMI	PhysicalActivityLevel	ObesityCategory
0	0.339295	0	0.341864	0.050076	-0.160970	3	0
1	1.057320	0	-0.574985	1.209739	1.374115	1	1
2	-0.213033	1	-0.192164	0.111266	0.150129	3	2
3	-0.986291	0	-0.154567	0.882535	0.811514	2	2
4	0.560226	0	1.311635	-0.139776	-0.710797	2	0
...	...	...	...	...	...	...	...
995	-1.759549	0	-1.403591	-0.458170	0.257160	3	2
996	-0.820593	1	-0.482874	1.705189	1.767533	0	1
997	-0.047334	1	-1.308268	0.490161	1.172337	0	1
998	0.781156	0	-0.568685	-0.853282	-0.546350	3	0
999	0.891622	1	0.823374	0.242315	-0.221481	0	0

1000 rows × 7 columns

Conclusion: The required dataset has been cleaned and the necessary transformations are applied on it.