10/3/25

ML - Lab - 2

PAGE :
DATE : / /

housing dataset :-

1)

```
import pandas as pd

# Load the csv file into a DataFrame

housing_df = pd.read_csv('/content/housing(1).csv')
print(housing_df.columns)

# display column information
print("Column Information:")
print(housing_df.info())

# Show statistical summary of numerical colums
print("\n Statistical Summary:")
print(housing_df.describe())

# Count unique values in the 'Ocean Proximity' colu
print("\n Unique values in 'Ocean proximity':")
print(housing_df['ocean_proximity'].value_columns())

# Identify columns with missing values
print("\n Columns with missing values:")
print(housing_df.isnull().sum())
```

o/p:- Column Information:

```
<class 'pandas.core.frame.DataFrame'>
Range Index: 20640 entries, 0 to 20639
Data columns (total 10 columns):
```

#	column	non-Null count	Dtype
0	longitude	20640 non-null	float 64
:			
9	Ocean Proximity	20640 non-null	object

PAGE :
DATE : / /

Statistical Summary:

	longitude	latitude	housing_median_age	median_house_v
count	20640.000000			20640.000
mean	. . .			. . .
std	. . .			
min	. . .			
25%	. . .			
50%	. .			
75%	. .			
Max	-114.310000			500001.00000

unique values in 'Ocean Proximity':

Ocean_Proximity

<1H OCEAN	9136
INLAND	6551
NEAR OCEAN	2658
NEAR BAY	2290
ISLAND	5

Name: count, dtype: int64

Columns with missing Values:

longitude	0
latitude	0
housing_median_age	0
total_rooms	0
total_bedrooms	207
population	0
households	0
median_income	0
median_house_value	0
Ocean_Proximity	0
dtype: int64	0

PAGE :
DATE : / /

Diabetes and Adult income Datasets :

Data Preprocessing

Write python code to implement the following data preprocessing techniques for diabetes & Adult income data sets

1. Data cleaning : Handling Missing values, Handling categorical data, Handling outliers

2. Data transformation : Min-Max scaler/ Normalization, standard scales

adult.csv & dataset of diabetes.csv

① Which columns in the dataset had missing values? How did you handle them?

- Adult Income Dataset (adult.csv)
  No columns had missing values

- Diabetes dataset (Dataset of Diabetes.csv)
  No columns had missing values

- Handling method : Since no missing values were found, no imputation was performed

2) Which categorical columns did you identify in the dataset? How did you encode them?

* Adult Income Dataset (adult.csv)

- categorical columns:
  - workclass
  - education
  - marital-status
  - occupation
  - relationship
  - race
  - gender
  - native-country
  - income

Encoding method: One-hot encoding was applied using pd.get_dummies(df, drop_first=True), which converted these categorical columns into numerical format

- Diabetes Dataset (Dataset of diabetes.csv)
  - Categories columns:
    - Gender
    - SS

- encoding method: one-hot encoding was applied pd.get_dummies(df, drop_first=True) to ~~or~~ transformation

PAGE:
DATE: / /

them into numeric format.

3) what is the difference b/w Min-Max scaling & standardization? when would you use one over the other.

feature	Min-Max Scaling	Standardization
Definition	Scales values b/w a fixed range (usually 0 & 1)	centers data around mean(0) with a SD of 1
formula	$x_{scaled} = x_{max} - x_{min}$	$x_{Standardized} = \ddot{o}x - \mu$
effect	Preserves original data distribution but scales it within a limited range	changes data to have zero variance.
when to use?	when preserving relationship & original range of data is important	when data follows a normal or distribution or contains outliers.

(4) When the one over the other

- Min Max scaling is useful when
  - you need data to be within a fixed range
  - The dataset does not contain extreme outliers.

PAGE :
DATE : / /

- Standardization (Z-score normalization) is useful when

- The dataset has varying units and a gaus (normal) distribution

- There are significant outliers, as standarization is less sensitive to them

p.03