Lab - 0

To Do :

(i) Initializing values directly into DataFrame

```
import pandas as pd
data = {
'USN' : ['001', '002', '003', '004', '005']
'Name' : ['Alice', 'Bob', 'charlie', 'David']
marks' : [25, 30, 35, 40, 45],
df = pd. DataFrame (data)
print ("Sample Data :")
print (df. head ())
```

(ii) Importing datasets from sklearn datasets

```
from sklearn. datasets import load diabetes
import pandas as pd
diabetes = load diabetes ()
df = pd. DataFrame (diabetes. data, columns:
diabetes. feature - names)
df [ 'target' ] = diabetes. target
print (df. head ())
```

(iii) Importing datasets from a specific CSV file

```
file - path = 'dataset of Diabetes. csv')
df = pd. read_csv (file - path)
print (df. head ())
```

(iv) Downloading datasets from existing dataset repositories like Kaggle, UCI

```
df = pd. read_csv ('/dataset of Diabetes. csv'
print ('sample data')
print (df. head ())
```

To-do 2

```python
import yfinance as yf
import pandas as pd
import matplotlib.pyplot as plt

tickers = ["HDFCBANK.NS", "ICICIBANK.NS",
           "KOTAKBANK.NS"]
start_date = "2024-01-01"
end_date = "2024-12-30"

data = yf.download(tickers, start=start_date,
                   end=end_date, group_by='tickers')
for ticker in tickers:
    stock_data = data[ticker]
    stock_data['Daily Returns'] = stock_data
    ['close'].pct_change()
    plt.figure(figsize=(12,6))
    plt.subplot(2,1,1)
    stock_data['close'].plot(title=-1"{ticker}
     - closing price")
    plt.ylabel("Price (INR)")
    plt.subplot(2,1,2)
    stock_data['Daily Returns'].plot(title=f
     "{ticker} - Daily Return", color='orange')
    plt.ylabel("Daily Return")
    plt.tight_layout()
    plt.show()
```

Code

(i) import pandas as pd
df = pd.read_csv ("training.csv")
print ("data loaded into data frame")

(ii) print ("Information of all columns")
print (df.info())

(iii) print ("statistical information")
print (df.describe())

(iv) print ("count of unique labels for each numeric column")
print (df ["..."].value_counts())

(v) print ("Columns with missing values")
missing_values = df.isnull().sum()
print (missing_values (missing_value > 0))
print (...)

Columns like Glucose, Blood Pressure and BMI that have missing values handled by imputing mean or median.

Columns like occupation and salary category that have missing values handled by mode or dropna()

2) Diabetes Dataset : The outcome column
categorical , encoded using label encod-

Adult Income Dataset : Column like
class , education were categorical encoded
using one - hot encoding

3) Min Max scaling : scales the data to a
fixed range (0 to 1) and is used when
data is to be bounded the model is sensitive
to the scale

Standardization scales the data to have a
mean of 0 and a standard deviation of
1 and is used when the data is normally
distributed or when the model assumes
a normal distribution

10/3/24