TAMILNADU MARGINAL WORKERS ASSESSMENT

Data Analytics with cognos – Phase 3

DOCUMENTATION

Team Members:

1.Sri Ranjani.C(au613021205053)

2.Zainab Hira.J(au613021205063)

3.Gowthami.S(au613021205013)

4.Lavanya.G(au613021205029)

5.Jayasri.P(au613021205019)

Phase 3: Development Part 1

Problem Definition:

Start the data analysis by loading and preprocessing the dataset. Load the dataset using python and data manipulation libraries (e.g.,

pandas).

Dataset Link:

https://www.kaggle.com/datasets/osmi/mental-health-in-tech-survey

Overview of the process:

1.Import Libraries:

Begin by importing the necessary libraries, such as pandas for data manipulation.

2.Load the Dataset:

Use pd.read_csv() or other appropriate methods to load your dataset into a pandas DataFrame.

3.Explore the Dataset:

Display the initial rows, check for missing values, and explore basic statistics to understand the structure and content of the data.

4.Handle Missing Values:

Decide on an appropriate strategy for dealing with missing values, such as dropping rows or filling values based on a specific strategy.

5.Additional Preprocessing Steps:

Depending on the nature of your data, consider additional preprocessing steps such as feature scaling, handling outliers, processing date-time features, dealing with text data, feature engineering, or discretization.

6.Save Preprocessed Dataset (Optional):

Save the preprocessed dataset to a new file if significant changes have been made.

Loading the dataset:

1.Importing libraries

Here, for preprocessing the dataset and manipulate the data, pandas is the library used to frame the data.

Code:

```
Import pandas as pd
```

2.Loading the dataset

In this step, we are framing the data into the table using DataFrame in pandas, and display the head or 5 rows of the dataset.

Code:

```
# Replace with the actual filename

file_path='C:/Users/IT/Downloads/survey.csv'

df = pd.read_csv(file_path)
```

Preprocessing the dataset

3.Explore the dataset:

After framing data, the first few or five rows of the data in displayed using the head() function.

Code:

```
print(df.head())
```

Output:

```
  Timestamp  Age  Gender         Country state self_employed  \
0  2014-08-27 11:29:31   37  Female   United States    IL           NaN
1  2014-08-27 11:29:37   44       M   United States    IN           NaN
2  2014-08-27 11:29:44   32    Male          Canada   NaN           NaN
3  2014-08-27 11:29:46   31    Male  United Kingdom   NaN           NaN
4  2014-08-27 11:30:22   31    Male   United States    TX           NaN

  family_history treatment work_interfere    no_employees  ...  \
0             No       Yes         Often           6-25  ...
1             No        No        Rarely  More than 1000  ...
2             No        No        Rarely           6-25  ...
3            Yes       Yes         Often         26-100  ...
4             No        No         Never         100-500  ...

                leave mental_health_consequence phys_health_consequence  \
0       Somewhat easy                       No                      No
1          Don't know                    Maybe                      No
2  Somewhat difficult                       No                      No
3  Somewhat difficult                      Yes                     Yes
4          Don't know                       No                      No

      coworkers supervisor mental_health_interview phys_health_interview  \
0  Some of them        Yes                      No                 Maybe
1            No         No                      No                    No
2           Yes        Yes                     Yes                   Yes
3  Some of them         No                   Maybe                 Maybe
4  Some of them        Yes                     Yes                   Yes

  mental_vs_physical obs_consequence comments
0                Yes              No      NaN
```

1	Don't know		No	NaN
2	No		No	NaN
3	No		Yes	NaN
4	Don't know		No	NaN

[5 rows x 27 columns]

4.Check for missing values:

In this step, the missing values or null values, if it present in the data are separated and number of null values are shown through this code.

Code:

```
print("Missing values:\n", df.isnull().sum())
```

Output:

```
Missing values:
 Timestamp                  0
Age                         0
Gender                      0
Country                     0
state                     515
self_employed              18
family_history              0
treatment                   0
work_interfere            264
no_employees                0
remote_work                 0
tech_company                0
benefits                    0
care_options                0
wellness_program            0
```

seek_help 0

anonymity 0

leave 0

mental_health_consequence 0

phys_health_consequence 0

coworkers 0

supervisor 0

mental_health_interview 0

phys_health_interview 0

mental_vs_physical 0

obs_consequence 0

comments 1095
dtype: int64

5.Check datatype:

In this step, the data type of the columns are discussed Code:

```
print("Data Types:\n", df.dtypes)
```

Output:

Data Types:

Timestamp	object
Age	int64
Gender	object
Country	object

state	object
self_employed	object
family_history	object
treatment	object
work_interfere	object
no_employees	object
remote_work	object
tech_company	object
benefits	object
care_options	object
wellness_program	object
seek_help	object
anonymity	object
leave	object
mental_health_consequence	object
phys_health_consequence	object
coworkers	object

supervisor object

mental_health_interview object

phys_health_interview object

mental_vs_physical object

obs_consequence object

comments object
dtype: object

6.Check basic statistics:

the statistics of the columns such as count, mean, std, min, max, 25%, 50%, 75% are shown through the describe() function command.

Code:

```
print("Summary Statistics:\n", df.describe())
```

Output:

Summary Statistics:

	Age
count	1.259000e+03
mean	7.942815e+07
std	2.818299e+09
min	-1.726000e+03

25% 2.700000e+01

50% 3.100000e+01

75% 3.600000e+01
max 1.000000e+11

7.Additional Preprocessing steps:

Perform any other preprocessing steps that are specific to your dataset and analysis goals. This may include scaling numeric features, handling outliers, or creating new features.

8.Saving Preprocessed dataset:

In this step, if we made substantial changes to the dataset and want to save the preprocessed version, you can use the following Code.

Code:

```
# Save the preprocessed dataset to a new CSV file
df.to_csv('preprocessed_dataset.csv', index=False)
```

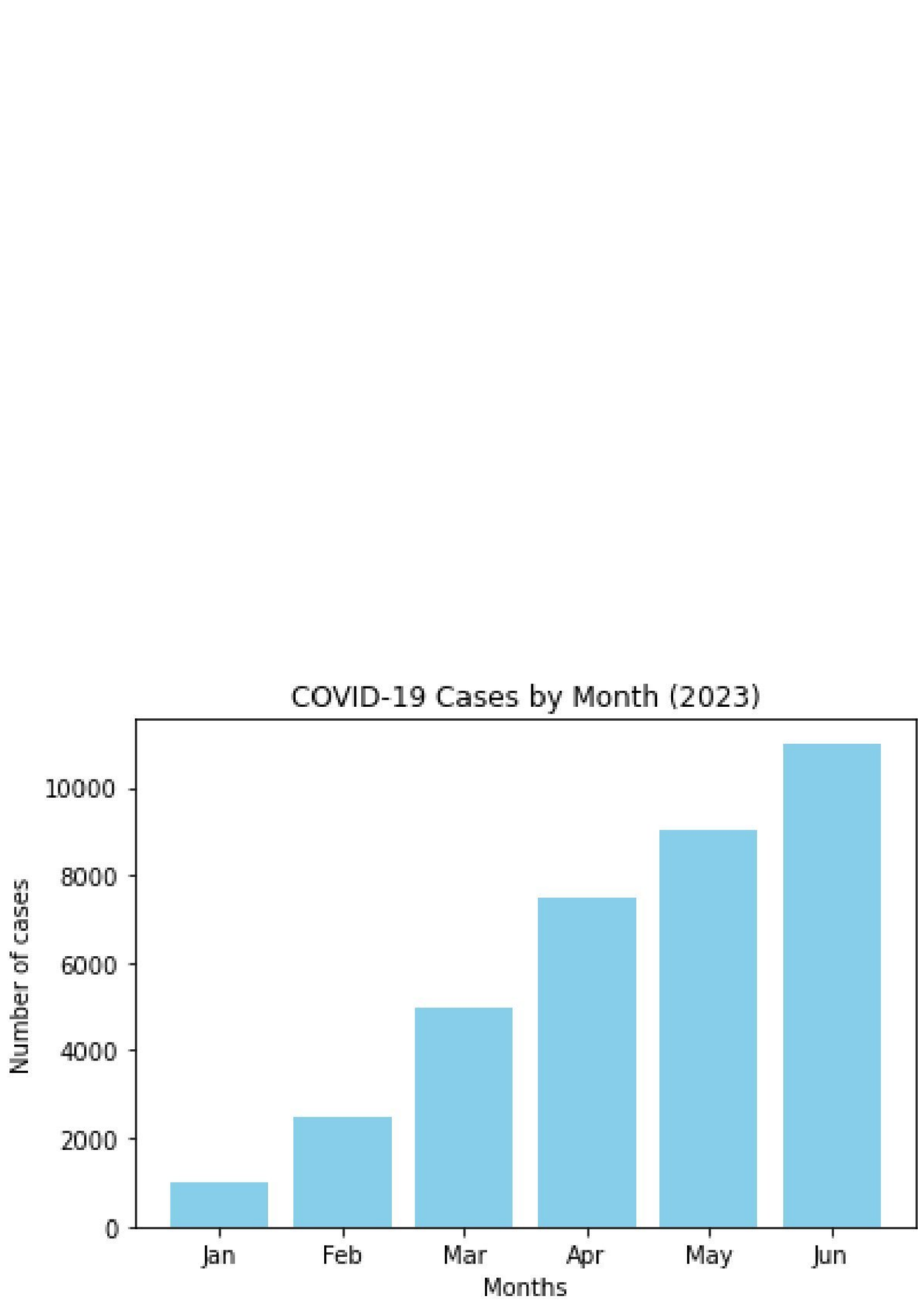
VISUALIZATION SOURCE CODE:

```
Import matplotlib.pyplot as plt

months=[‘Jan’,’Feb’,’Mar’,’Apr’,’May’,’Jun’]

Cases=[1000,2500,5000,7500,9000,11000]

plt.bar(months ,cases,color=’skyblue’)

plt.Xlabel(‘Months’)

plt.Ylabel(‘Number of cases’)
```

plt.title(‘COVID-19 cases by month(2023)’)

plt.show()

OUTPUT:



CONCLUSION:

In conclusion ,the outlined data loading and preprocessing steps provide a foundational framework for preparing a dataset for analysis in python using the pandas library

In conclusion,