

Slip1

Q1. Write a program to create a data frame and access rows and columns.

```
import pandas as pd
#create dataframe
df = pd.DataFrame({'Name': ['vighnesh', 'samad', 'faisal'], 'Age': [25, 30, 35]})
#accesses the first row of the data frame
print(df.loc[0])
#accesses the Name column of the data frame
print(df['Name'])
#accesses the second and third rows of the data frame
print(df.loc[[1, 2]])
```

Q2. Write a Python program to read "StudentsPerformance.csv" file. Solve Following: - To display the shape of the dataset. - To display the top rows of the dataset with their columns. Note: Download dataset from following link : ([https://www.kaggle.com/spscientist/students-performance-inexams?](https://www.kaggle.com/spscientist/students-performance-inexams?select=StudentsPerformance.csv)select=StudentsPerformance.csv)

```
import pandas as pd
# Read the StudentsPerformance.csv file into a Pandas DataFrame
df = pd.read_csv('StudentsPerformance.csv')
# Display the shape of the dataset
print(df.shape)
# Display the top rows of the dataset with their columns
print(df.head())
```

Slip2

Q1. Write a program to accept random data and draw the pivot table.

```
import pandas as pd
import numpy as np

df = pd.DataFrame(np.random.randn(10, 4), columns=['A', 'B', 'C', 'D'])
pivot_table = df.pivot_table(index=['A', 'B'])
print(pivot_table)
```

Q2. Write a program to accept random values and show training and testing data with the help of scatter plot.

```
import numpy as np
```

```

import matplotlib.pyplot as plt
# Get the number of training and testing data points from the user
num_training_points = int(input('Enter the number of training data
points: '))
num_testing_points = int(input('Enter the number of testing data
points: '))

# Generate random training and testing data
train_data = np.random.randn(num_training_points, 2)
test_data = np.random.randn(num_testing_points, 2)
# Create a scatter plot of the training data and testing data
plt.scatter(train_data[:, 0], train_data[:, 1], color='blue',
label='Training data')
plt.scatter(test_data[:, 0], test_data[:, 1], color='red',
label='Testing data')

# Set the title and labels of the plot
plt.title('Scatter plot of training and testing data')
plt.legend()
plt.show()

```

Slip3

Q1. Write a program accepting values in the dictionary and display them using pandas

```

import pandas as pd
my_dict = {
    "Name": ["vighnesh", "samad", "faisal"],
    "Age": [25, 30, 35],
    "Occupation": ["Software Engineer", "ServiceNow Developer",
"Lawyer"]
}
df = pd.DataFrame(my_dict)
print(df)

```

Q2. Consider the following observations/data. And apply simple linear regression and find out estimated coefficients b_1 and b_0 . Also analyze the performance of the model (Use sklearn package) $x = \text{np.array}([1, 2, 3, 4, 5, 6, 7, 8])$ $y = \text{np.array}([7, 14, 15, 18, 19, 21, 26, 23])$

```

import numpy as np
import matplotlib.pyplot as plt
from sklearn.linear_model import LinearRegression

x = np.array([1, 2, 3, 4, 5, 6, 7, 8]).reshape(-1, 1)

```

```

y = np.array([7, 14, 15, 18, 19, 21, 26, 23]).reshape(-1, 1)

model = LinearRegression()
model.fit(x, y)

b0 = model.intercept_[0]
b1 = model.coef_[0][0]
print("Estimated Coefficients:")
print(f"b0 (Intercept): {b0}")
print(f"b1 (Slope): {b1}")

y_pred = model.predict(x)
plt.scatter(x, y, label='original data')
plt.plot(x, y_pred, label='linear regression')

```

Slip4

Q1. Write a program accepting random values in the dictionary and make a pivot table.

```

import pandas as pd

my_dict = {
    "Name": ["vighnesh", "samad", "faisal"],
    "Age": [20, 40, 10]
}
df = pd.DataFrame(my_dict)
pivot_table = df.pivot_table(values='Age', index='Name')
print(pivot_table)

```

Q2. Read the Data set using the pandas library. find the following details

- a. Get the descriptive statistics
- b. what are the 7 most common job titles
- c. what is the email of employee with following phone no: 919096777860

```

import pandas as pd
myfile = pd.read_csv('ecommerce.csv')
#ans:a
print(myfile.describe())
#ans:b
jobtit = myfile['Job'].value_counts().head(7)
print('The 7 most common job titles are:\n', jobtit)
#ans:c

```

```
myfile[myfile['phone']== 9096777860]['Email']
```

Slip5

Q1. Write a Program Plot a pie chart for appropriate columns in dataset .

```
import pandas as pd
import matplotlib.pyplot as plt
myfile = pd.read_csv('ecommerce.csv')
#Plot a pie chart for the column
myfile['Purchase Price'].value_counts().plot.pie(figsize=(10, 10))
plt.show()
```

Q2. Read the Data set using the pandas library. find the following details a. Plot a heatmap to identify any correlation between the columns. b. Find the missing values in the dataset and if exist fill in missing data with appropriate values.

```
import pandas as pd
import seaborn as sns
df = pd.read_csv('StudentsPerformance.csv')
missing_values = df.isnull()
df = df.fillna('NaN')
sns.heatmap(df.corr(), annot=True)
```

Slip6

Q1. Write a Program Check which features are correlated to diabetes using different correlation coefficient.

```
import pandas as pd
import seaborn as sn

data = pd.read_csv("diabetes.csv")
sn.heatmap(data.corr(), annot=True)
print(data.corr())
```

Q2. Read the Data set using the pandas library. predict the the following a. What is the average age of people having diabetes? b. How many people in the dataset are less than the

average age and have diabetes. c. Check the relationship between Blood pressure and weight.

```
import pandas as pd
data = pd.read_csv("diabetes.csv")
average=data['Age'].mean()
#ans:a
print("Average :",average)
#ans:b
no_of_people= (data['Age'] < average) & data['Outcome']
print("no_of_people",no_of_people.sum())
#ans:c
correlation = data["BloodPressure"].corr(data["BMI"])
print("relationship :",correlation)
```

Slip7

Q1.Write a program to create a Data Frames which contain details of 5 Students and display the details.Students contain (Rollno,Studname,Address,Marks)

```
import pandas as pd
students = [
    [1, 'vighnesh', 'Pimpri', 90],
    [2, 'samad', 'kondwa', 85],
    [3, 'faisal', 'kondwa', 80],
    [4, 'aman', 'swargate', 75],
    [5, 'zoheb', 'camp', 70]
]

df = pd.DataFrame(students, columns=['Rollno', 'Studname',
    'Address', 'Marks'])
print(df)
```

Q2.Consider the following observations/data. And apply simple linear regression and find out estimated coefficients b1 and b1 Also analyze the performance of the model (Use sklearn package).

```
import numpy as np
import matplotlib.pyplot as plt
from sklearn.linear_model import LinearRegression

x = np.array([1, 2, 3, 4, 5, 6, 7, 8]).reshape(-1, 1)
```

```

y = np.array([7, 14, 15, 18, 19, 21, 26, 23]).reshape(-1, 1)

model = LinearRegression()
model.fit(x, y)

b0 = model.intercept_[0]
b1 = model.coef_[0][0]
print("Estimated Coefficients:")
print(f"b0 (Intercept): {b0}")
print(f"b1 (Slope): {b1}")

y_pred = model.predict(x)
plt.scatter(x, y, label='original data')
plt.plot(x, y_pred, label='linear regression')

```

Slip8

Q1. Write a Program Identify outliers in the dataset and draw Box plot.

```

import pandas as pd
import seaborn as sn

data = pd.read_csv("diabetes.csv")
sn.boxplot(data=data, orient="h")

```

Q2. Write a python program to implement Linear Regression models for a car dataset. Dataset can be downloaded from: [Kaggle.com](https://www.kaggle.com).

Error aara ye question me ,nahi hua mera !

Slip9

Q1. Write a program accepting values in the dictionary and make a pivot table.

```

import pandas as pd
dict = {'rollno': [1, 2, 3],
        'Name': ['Vighnesh', 'Faisal', 'Samad'],
        'Marks': [55, 59, 60],
        'Attempts': [1, 2, 1]}

df = pd.DataFrame(dict)
print(df)

```

Q2.Read the Data set using the pandas library. find the following details a. find missing value and replace with irrelevant. b. What are the 7 most common job titles? c. what is the email of employee with following cell no:919096777860

```
import pandas as pd

myfile = pd.read_csv('ecommerce.csv')
#ans:a
# Identify the missing values
missing_values = df.isnull()
df.fillna('NaN')
#ans:b
jobtit = myfile['Job'].value_counts().head(7)
print('The 7 most common job titles are:\n', jobtit)
#ans:c
myfile[myfile['phone']== 9096777860]['Email']
```

Slip10

Q1.Write a Pandas program to create and display a DataFrame from a specified dictionary data which has the index labels.

```
import pandas as pd
data = {'a': [1, 2, 3, 4, 5], 'b': [6, 7, 8, 9, 10], 'c': [11, 12, 13, 14, 15]}
df = pd.DataFrame(data, index=['A', 'B', 'C', 'D', 'E'])
print(df)
```

Q2.Draw confusion matrix with help of following data: data = {'y_actual': [1, 0, 0, 1, 0, 1, 0, 0, 1, 0, 1, 0], 'y_predicted': [1, 1, 0, 1, 0, 1, 1, 0, 1, 0, 0, 0]}

```
import numpy as np
import seaborn as sns
from sklearn.metrics import confusion_matrix
import matplotlib.pyplot as plt

x=np.array([1, 0, 0, 1, 0, 1, 0, 0, 1, 0, 1, 0])
y=np.array([1, 1, 0, 1, 0, 1, 1, 0, 1, 0, 0, 0])
cm=confusion_matrix(x,y)
sns.heatmap(cm,annot=True)
```

```
plt.title("Confusion matrix")
```

Slip11

Q1. Write a program to draw residual plots.

```
import matplotlib.pyplot as plt
import seaborn as sns
df = pd.read_csv('diabetes.csv')
sns.residplot(x='Glucose', y='BloodPressure', data=df)
plt.show()
```

Q2. Read the Data set using the pandas library. draw the following plots: a. Draw Distribution plot and its subtypes of plot.

```
import seaborn as sns
df = sns.load_dataset("diamonds")
df.head()
sns.distplot(df['carat'], kde = False)
sns.jointplot(x = 'carat', y = 'price', data = df, kind = 'kde')
sns.pairplot(df)
sns.rugplot(df['price'], height=0.5, axis='y', color='red')
```

Slip12

Q1. Write a program to accept csv file and draw the pivot table.

```
import pandas as pd
df = pd.read_csv("ecommerce.csv")
pivot_table = df.pivot_table(values='Purchase Price', index='phone')
print(pivot_table)
```

Q2. Write a program to accept random values and show training and testing data with the help of scatter plot.

```
import numpy as np
import matplotlib.pyplot as plt
import sklearn.model_selection as sk
```



```

x = np.random.randn(10)
y = np.random.randn(10)

X_train, X_test, y_train, y_test =sk.train_test_split(x, y,
test_size=0.25, random_state=42)
plt.scatter(X_train, y_train, label='Training Data')
plt.scatter(X_test, y_test, label='Testing Data')
plt.legend()
plt.show()

```

Slip13

1. Write a program to create a Data Frames which contain details of 5 Students and display the details.alter the “contact” column in student dataframe.

```

import pandas as pd
df = pd.DataFrame({'Name': ['vighnesh', 'samad', 'faisal', 'aman',
'zoheb'],
                  'Age': [25, 30, 35, 40, 45],
                  'Contact':
['78787878', '98989898', '12212121', '323232323', '656565655']})

print(df, "\n")
# Alter the "Contact" column
df['Contact'] = df['Contact'].str.replace('78787878', '45454545')
print(df)

```

Q2. Draw confusion matrix with help of following data: data = {'y_actual': [yes, No, No, yes, No, yes, No, No, yes, No, yes, No], 'y_predicted': [yes, yes, No, yes, No, yes, yes, No, yes, No, No, No] }

```

import numpy as np
import seaborn as sns
from sklearn.metrics import confusion_matrix
import matplotlib.pyplot as plt

x=np.array(['yes', 'No', 'No', 'yes', 'No', 'yes', 'No',
'No', 'yes', 'No', 'yes', 'No'])
y=np.array(['yes', 'yes', 'No', 'yes', 'No', 'yes', 'yes', 'No',
'yes', 'No', 'No', 'No'] )
cm=confusion_matrix(x,y)
sns.heatmap(cm,annot=True)

```

```
plt.title("Confusion matrix")
```

Slip14

Q1. Write a Program Plot a scatter plot for appropriate columns in dataset.

```
import pandas as pd
import matplotlib.pyplot as plt
df = pd.read_csv('diabetes.csv')

#Plot a scattered plot for the columns
plt.scatter(df['Glucose'], df['BloodPressure'])
plt.show()
```

Q2. Read the Data set using the pandas library. find the following details a. Describe the student details. b. Find the missing values in the dataset and if exist fill in missing data with appropriate values c. Drop irrelevant values.

```
import pandas as pd
df = pd.read_csv('diabetes.csv')
print(df.describe())
print(df.isnull().sum())
df.fillna("NaN")
df.dropna()
print(df)
```

Slip15

Q1. Write a Program alter rows and columns in a data set .

```
import pandas as pd
df = pd.DataFrame({'Name': ['vighnesh', 'samad', 'faisal', 'aman', 'zoheb'],
                   'Age': [25, 30, 35, 40, 45],
                   'Contact': ['pimpri', 'kondhwa', 'kondhwa', 'swargate', 'camp']})

df.loc[0, 'Age'] = 20
df.loc[2, 'Contact'] = 'Pune'
df = df.rename(columns={'Contact': 'Address'})
print(df)
```

Q2. Write a program to create series and data frame and apply following functions: a. apply() b. applymap() c. map()

```
import pandas as pd
series = pd.Series([1, 2, 3, 4, 5])
df = pd.DataFrame({'Name': ['vighu', 'sam', 'faisu', 'aman', 'zoheb'],
                  'Age': [25, 30, 35, 40, 45]})

def square(x):
    return x * x
apply_fun = series.apply(square)

def add_one(x):
    return x+1
applymap_fun = df.applymap(add_one)

def convert_to_string(x):
    return str(x)
map_fun = series.map(convert_to_string)

print(apply_fun)
print(applymap_fun)
print(map_fun)
```

Slip16

Q1. Write a Program Identify outliers in the dataset for each column shown in graph.

```
import pandas as pd
import matplotlib.pyplot as plt
df = pd.read_csv('diabetes.csv')

column_names = list(df)
for column in column_names:
    plt.boxplot(df[column])
    plt.title(column)
    plt.show()
```

Q2. Read or create seller datasets using pandas library perform following operations on it: a. How many different seller types are there. Show using appropriate graphs. b. Which seller

has minimum records in the dataset. c. Using appropriate graphs show the count of different layout types in the dataset.

Error aara ye question me ,nahi hua mera !

Slip 17

Q1.Read or create society's datasets using pandas library perform following operations on it:

a. How many houses in pune rental list do not have bedroom b. How many 2BHK apartments are there in kondhwa.

```
import pandas as pd
data = {
    'location': ['Pimpri', 'Pune', 'Pune', 'pune', 'Kondhwa'],
    'property': ['Rental', 'Rental', 'Rental', 'Apartment',
    'Apartment'],
    'bedroom': [3, 0, 0, 2, 2],
}

df = pd.DataFrame(data)
print(df, "\n")
#ans:a
print("houses in pune rental list do not have
bedroom:", len(df[(df['location'] == 'Pune') & (df['property'] ==
'Rental') & (df['bedroom'] == 0)]))
#ans:b
print("number of 2BHK apartments in Kondhwa:", len(df[(df['location']
== 'Kondhwa') & (df['property'] == 'Apartment') & (df['bedroom'] ==
2)]))
```

Q2.Read or create datasets using pandas library perform following operations on it: 1. Find the size of the dataset. 2. Give a count of all available airlines. 3. Which airline is the most common 4. Which airport has maximum departures

```
import pandas as pd
data = pd.read_csv('airlines.csv')
#ans:a
print("Size of the dataset:", data.shape)
#ans:b
print("available airlines:", data['Airport.Code'].value_counts())
#ans:c
print("airline is the most common:", airline_counts.idxmax())
#ans:4
```

```
print("airport has maximum  
departures", (data['Airport.Name'].value_counts()).idxmax())
```

Slip18

Q1. Write a program to display different distribution plots using your data set.

```
import seaborn as sns  
import pandas as pd  
df = pd.read_csv("diabetes.csv")  
df.head()  
sns.distplot(df['Glucose'], kde = False)  
sns.jointplot(x = 'Glucose', y = 'BloodPressure', data = df, kind  
= 'kde')  
sns.pairplot(df)  
sns.rugplot(df['BloodPressure'], height=0.5, axis='y', color='red')
```

Q2. Write a program to draw Residual plot with the help of a data set.

```
import matplotlib.pyplot as plt  
import seaborn as sns  
df = pd.read_csv('diabetes.csv')  
sns.residplot(x='Glucose', y='BloodPressure', data=df)  
plt.show()
```

Slip19

Q1. Write a Pandas program to find and replace the missing values in a given DataFrame which do not have any valuable information.

```
import pandas as pd  
myfile = pd.read_csv('marine_economy.csv')  
print(myfile)  
missing_values = myfile.isnull()  
  
myfile["year"] = myfile["year"].fillna(myfile["year"].mean())  
myfile["data_value"] =  
myfile["data_value"].fillna(myfile["data_value"].mean())  
  
missing_values = myfile.isnull()
```

```
print(missing_values)
```

Q2. Read or create datasets using pandas library perform following operations on it: a. select the rows the marks are between 30 and 35. b. select the rows where the number of attempts in the examination is less than 2 and marks greater than 30. c. calculate the sum of the examination attempts by the students.

```
import pandas as pd
dict = {'rollno': [1, 2, 3],
        'Name': ['Vighnesh', 'Faisal', 'Samad'],
        'Marks': [35, 39, 60],
        'Attempts': [1, 2, 1]}
df = pd.DataFrame(dict)
print(df, "\n")
#ans:a
mrange=df[df['Marks'].between(35, 40)]
print(mrange)
#ans:b
dis=df[(df['Attempts'] < 2) & (df['Marks'] > 30)]
print(dis)
#ans:c
s=df['Attempts'].sum()
print(s)
```

Slip20

Q1. Write a program accept csv file and draw the heatmap.

```
import pandas as pd
import seaborn as sn

data = pd.read_csv("diabetes.csv")
sn.heatmap(data.corr(), annot=True)
```

Q2. Draw confusion matrix with help of following data: data = {'actual': [1, 0, 0, 1, 0, 1, 0, 0, 1, 0, 1, 0], 'predicted': [1, 1, 0, 1, 0, 1, 1, 0, 1, 0, 0, 0]}

```
import numpy as np
import seaborn as sns
from sklearn.metrics import confusion_matrix
import matplotlib.pyplot as plt
```

```
x=np.array([1, 0, 0, 1, 0, 1, 0, 0, 1, 0, 1, 0])
y=np.array([1, 1, 0, 1, 0, 1, 1, 0, 1, 0, 0, 0])
cm=confusion_matrix(x,y)
sns.heatmap(cm,annot=True)
plt.title("Confusion matrix")
```