A

File for

Skill Based Mini Project

Data Science (2240522)

Submitted for the partial fulfillment of the degree of

Bachelor of Technology

In

Artificial Intelligence and Robotics

Submitted By

Chandan Singh - 0901AI221021

Submitted To

Dr. Abhishek Bhatt

Prof. Khemchand Shakywar

Assistant Professor

Centre for Artificial Intelligence

माधव प्रौद्योगिकी एवं विज्ञान संस्थान, ग्वालियर
MADHAV INSTITUTE OF TECHNOLOGY & SCIENCE, GWALIOR
Deemed University
(Declared under Distinct Category by Ministry of Education, Government of India)
NAAC ACCREDITED WITH A++ Grade
Gola Ka Mandir, Gwalior (M.P.)- 474005, INDIA
Ph.:+91-751-2409300, E-mail: vicechancellor@mitsgwalior.in, Website: www.mitsgwalior.in

July-Dec 2024

DECLARATION BY THE CANDIDATE

We hereby declare that the work entitled **"Skill Based Mini Project"** is our work, conducted under the guidance of **Dr. Abhishek Bhatt, Assistant Professor,** during the session July-Dec 2024. The report submitted by us is a record of Bonafide work carried out by us.

We further declare that the work reported in this report has not been submitted and will not be submitted, either in part or in full, for the award of any other degree or diploma in this institute or any other institute or university.

Chandan Singh - 0901AI221021
Chandraveer Singh Rajput - 0901AI221022
Deepak Patidar - 0901AI221023
Dev Sharma - 0901AI221024
Devyanshi Bhatane - 0901AI221025

Date-

Place-Gwalior

MADHAV INSTITUTE OF TECHNOLOGY AND SCIENCE, GWALIOR

Deemed University

(Declare under District by Ministry of Education, Government of India)

NAAC ACCREDITED WITH A++ Grade

CERTIFICATE

This is to certified that Chandan Singh - 0901AI221021, Chandraveer Singh Rajput - 0901AI221022, Deepak Patidar - 0901AI221023, Dev Sharma - 0901AI221024, Devyanshi Bhatane - 0901AI221025 have submitted the Skill Based Mini Project report entitled "**Consider any dataset from an online repository to design and implement aproblem using linear regression and logistic regression.**" under the guidance of Dr. Abhishek Bhatt, Assistant professor, in partial fulfillment of the requirement for the award of the degree of Bachelor of Technology in IT(Artificial Intelligence and Robotics) Centre for Artificial Intelligence from Madhav Institute of Technology and Science, Gwalior

Dr. Abhishek Bhatt
Assistant Professor
Centre for Artificial Intelligence

Prof. Khemchand Shakywar
Assistant Professor
Centre for Artificial Intelligence

Dr. Rajni Ranjan Singh
Professor & Head of Department
Centre for Artificial Intelligence

ACKNOWLEDGEMENT

We would like to express my greatest appreciation to all the individuals who have helped and supported me throughout this Skill Based Mini Project. We are thankfulto whole Centre for AI department for their ongoing support during the experiments,from initial advice and provision of contact in the first stages through ongoing adviceand encouragement, which led to the final report of this Skill Based Mini Project.

A special acknowledgment goes to my colleagues who helped me in completing the file by exchanging interesting ideas to deal with problems and sharing their experiences.

We wish to thank our professor Dr. Abhishek Bhatt as well for his undivided supportand interest which inspired us and encouraged us to go our own way without whomwe would have been unable to complete my project.

At the end, we want to thank my friends who displayed appreciation for our work andmotivated me to continue our work.

Table of Contents

MICROPROJECT

AIM: Write a program in python to join two data frames.

Theory:

DataFrames: In Pandas, a DataFrame is a two-dimensional labeled data structure with columns that can hold different data types. It's analogous to a spreadsheet or SQL table.

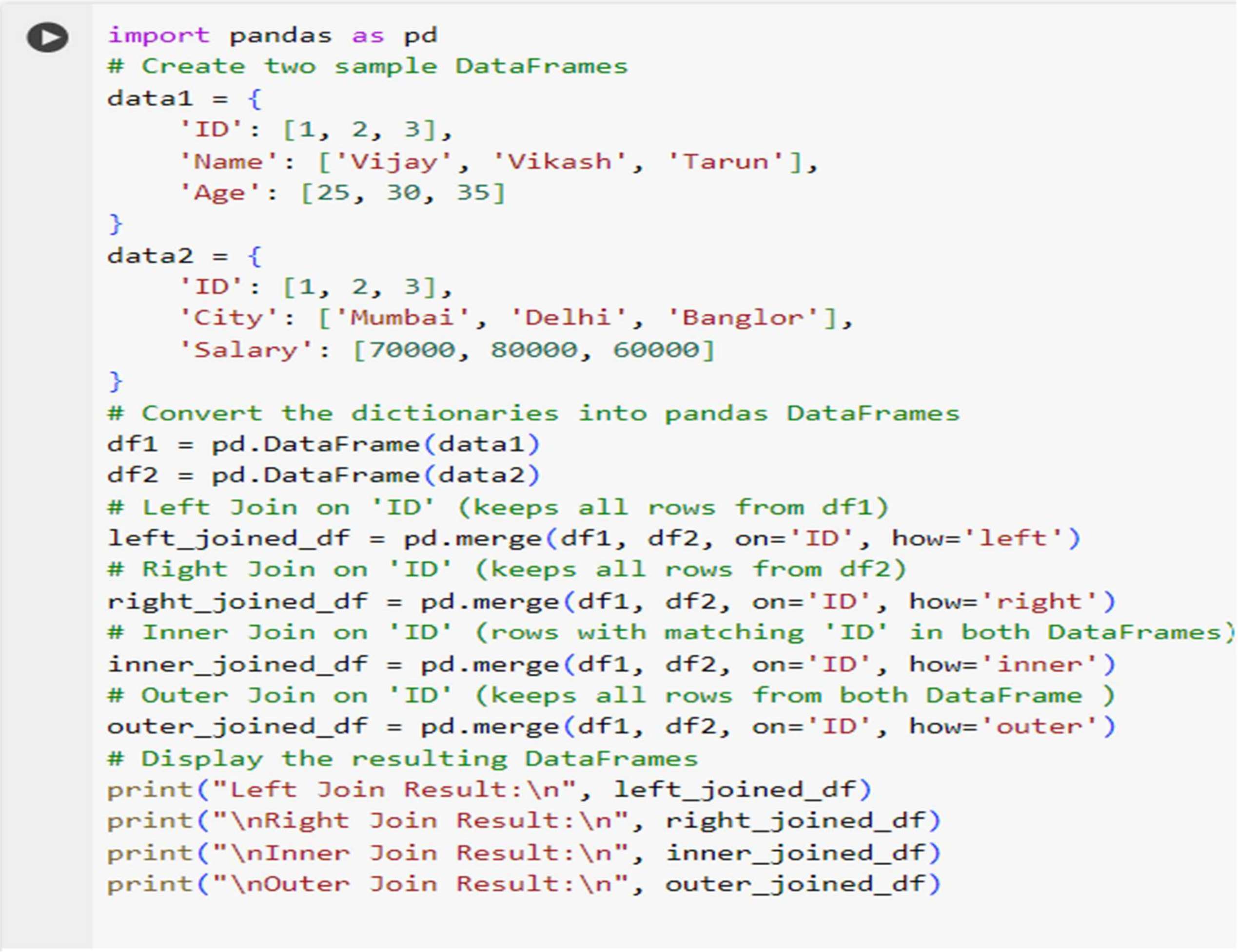
Joining DataFrames: Joining combines data from two or more DataFrames based on a common column or index. This is essential for relational operations in data analysis and machine learning.

Join Types:

- **InnerJoin:** Returns rows that have matching values in both DataFrames.
- **LeftJoin:** Returns all rows from the left DataFrame and matching rows from the right DataFrame.
- **RightJoin**: Returns all rows from the right DataFrame and matching rows from the left DataFrame.
- **OuterJoin**: Returns all rows from both DataFrames, including rows with missing values.

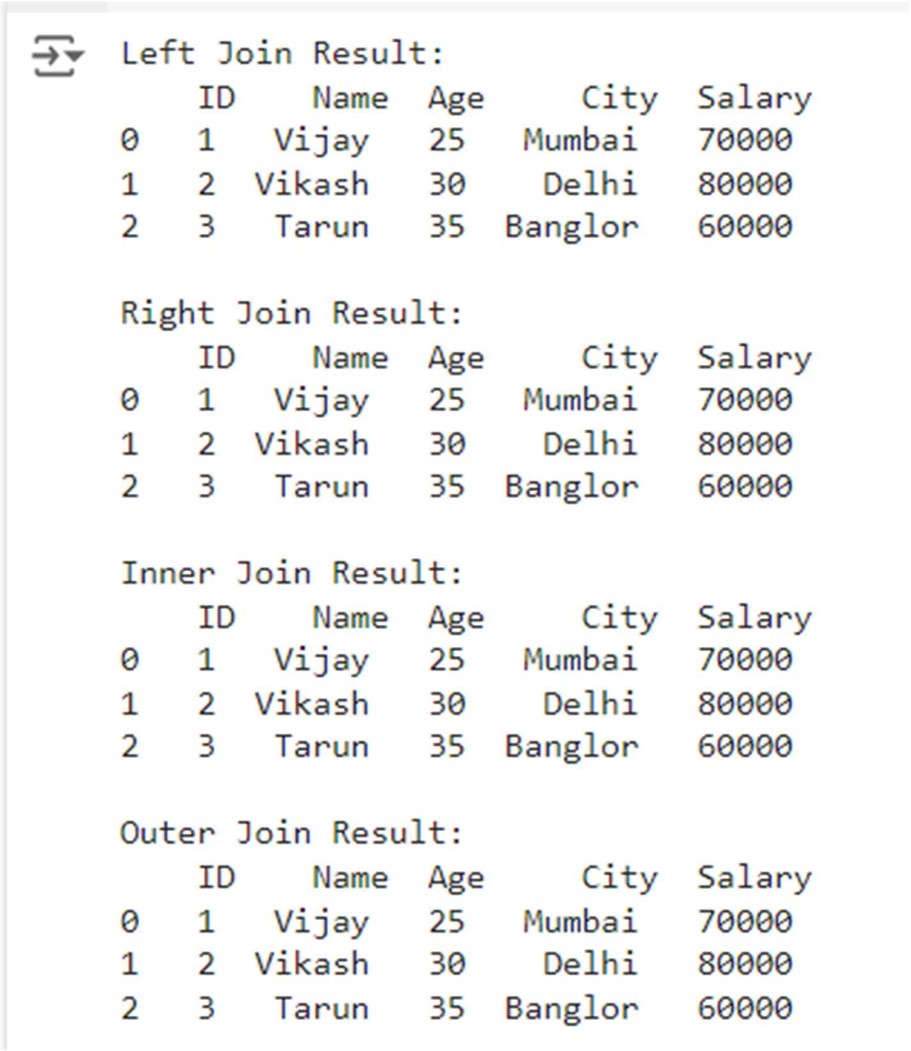
Join Keys: The common column(s) used for matching rows during the join operation. These can be specified using the on, left_on, right_on, or index parameters.

Code:



```
import pandas as pd
# Create two sample DataFrames
data1 = {
    'ID': [1, 2, 3],
    'Name': ['Vijay', 'Vikash', 'Tarun'],
    'Age': [25, 30, 35]
}
data2 = {
    'ID': [1, 2, 3],
    'City': ['Mumbai', 'Delhi', 'Banglor'],
    'Salary': [70000, 80000, 60000]
}
# Convert the dictionaries into pandas DataFrames
df1 = pd.DataFrame(data1)
df2 = pd.DataFrame(data2)
# Left Join on 'ID' (keeps all rows from df1)
left_joined_df = pd.merge(df1, df2, on='ID', how='left')
# Right Join on 'ID' (keeps all rows from df2)
right_joined_df = pd.merge(df1, df2, on='ID', how='right')
# Inner Join on 'ID' (rows with matching 'ID' in both DataFrames)
inner_joined_df = pd.merge(df1, df2, on='ID', how='inner')
# Outer Join on 'ID' (keeps all rows from both DataFrame )
outer_joined_df = pd.merge(df1, df2, on='ID', how='outer')
# Display the resulting DataFrames
print("Left Join Result:\n", left_joined_df)
print("\nRight Join Result:\n", right_joined_df)
print("\nInner Join Result:\n", inner_joined_df)
print("\nOuter Join Result:\n", outer_joined_df)
```

Output:



```
Left Join Result:
    ID     Name  Age     City  Salary
0   1   Vijay   25   Mumbai   70000
1   2  Vikash   30    Delhi   80000
2   3   Tarun   35  Banglor   60000

Right Join Result:
    ID     Name  Age     City  Salary
0   1   Vijay   25   Mumbai   70000
1   2  Vikash   30    Delhi   80000
2   3   Tarun   35  Banglor   60000

Inner Join Result:
    ID     Name  Age     City  Salary
0   1   Vijay   25   Mumbai   70000
1   2  Vikash   30    Delhi   80000
2   3   Tarun   35  Banglor   60000

Outer Join Result:
    ID     Name  Age     City  Salary
0   1   Vijay   25   Mumbai   70000
1   2  Vikash   30    Delhi   80000
2   3   Tarun   35  Banglor   60000
```

MACRO PROJECT

AIM: Write a python program to draw correlation matrix.

Theory:

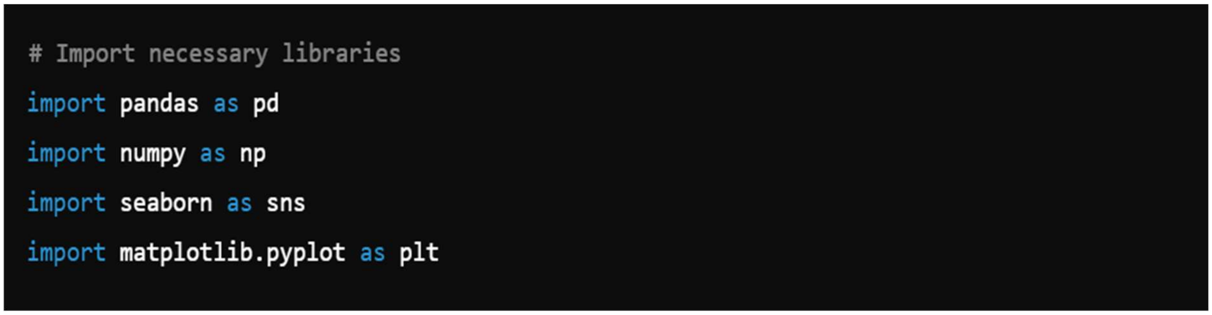
A correlation matrix is a table that displays the correlation coefficients between different variables in a dataset. Correlation coefficients quantify the strength and direction of the linear relationship between two variables. A correlation coefficient of 1 indicates a perfect positive correlation, -1 indicates a perfect negative correlation, and 0 indicates no correlation.
Correlation matrices are valuable tools in data analysis as they provide a concise and visual representation of the relationships between variables. They can help us identify:

- Which variables are strongly correlated.
- Which variables are weakly correlated.
- If there are any unexpected relationships between variables.

Understanding correlation helps us gain insights into the underlying structure of our data and make informed decisions about modeling and analysis.

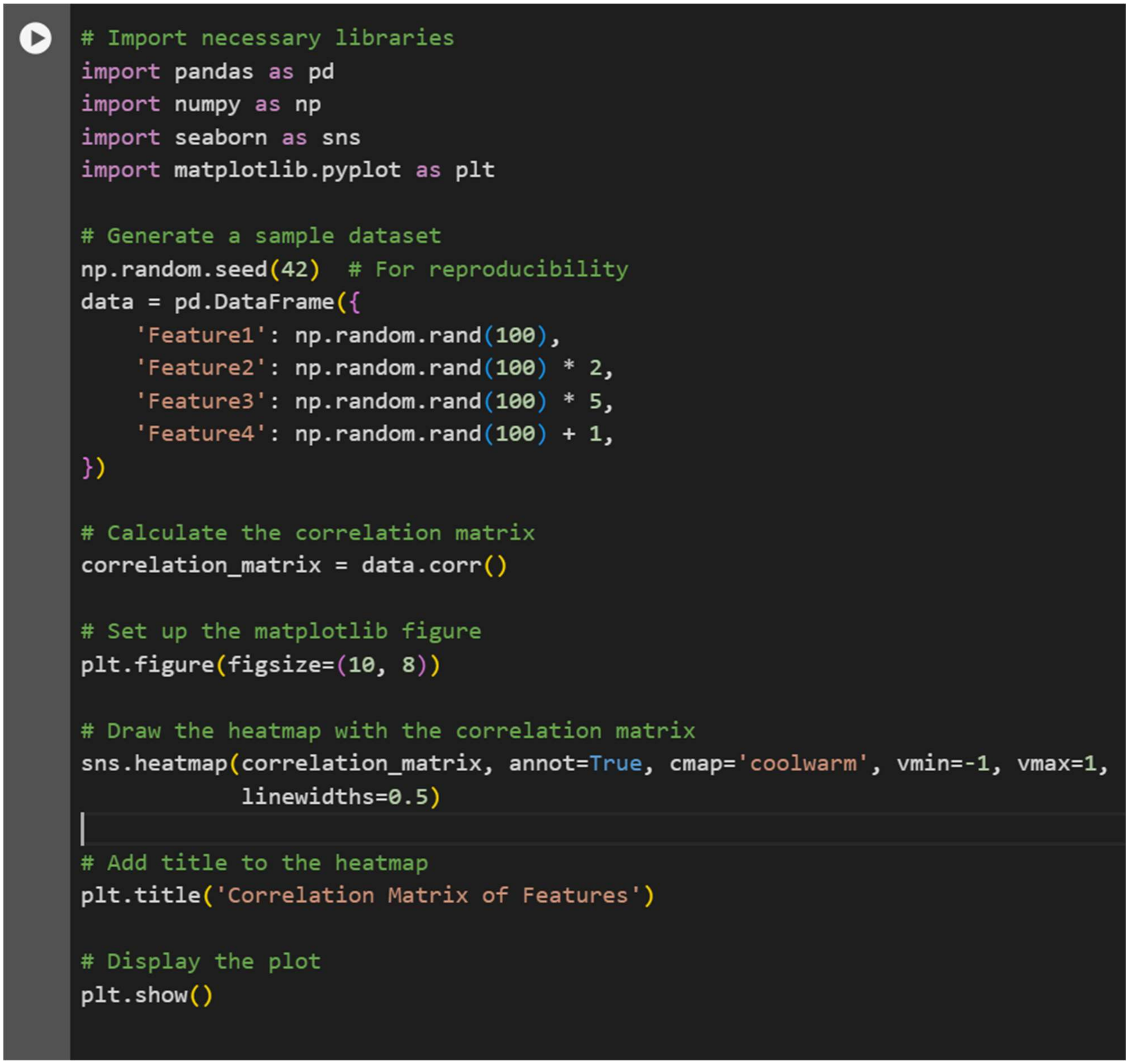
Libraries used :

Before we delve into the practical steps, we need to import the necessary libraries. The primary library for data manipulation and analysis is Pandas, which provides powerful data structures like Data Frames. **Pandas** is a powerful and widely used open-source data analysis and manipulation library for Python. It provides data structures and functions needed to work with structured data seamlessly.For visualization, we will use Matplotlib, a versatile plotting library in Python.**Matplotlib** is a comprehensive library for creating static, animated, and interactive visualizations in Python. It is widely used for data visualization and is particularly known for its ability to produce publication-quality plots.



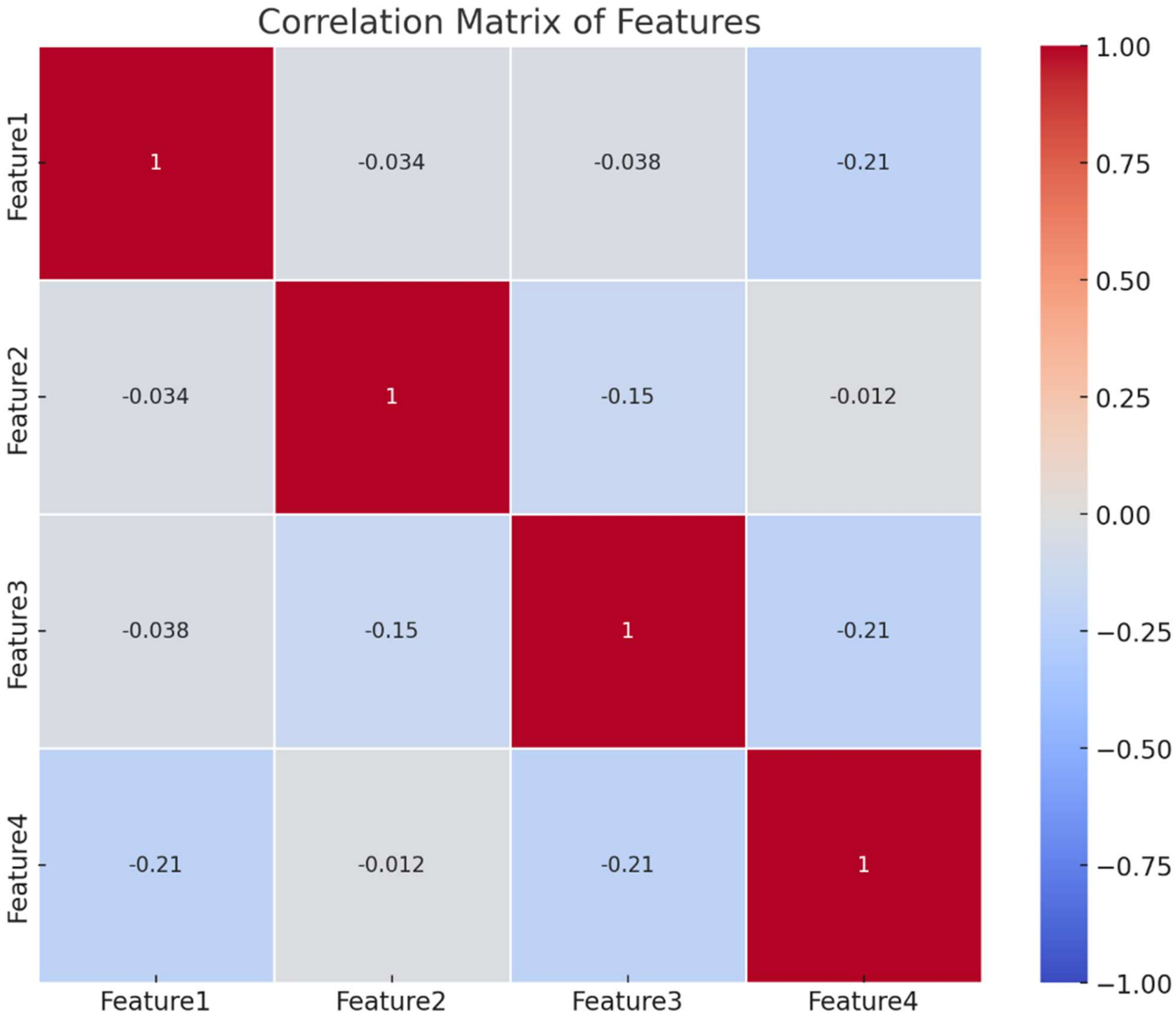
```
# Import necessary libraries
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
```

Code:



```python
# Import necessary libraries
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt

# Generate a sample dataset
np.random.seed(42)  # For reproducibility
data = pd.DataFrame({
    'Feature1': np.random.rand(100),
    'Feature2': np.random.rand(100) * 2,
    'Feature3': np.random.rand(100) * 5,
    'Feature4': np.random.rand(100) + 1,
})

# Calculate the correlation matrix
correlation_matrix = data.corr()

# Set up the matplotlib figure
plt.figure(figsize=(10, 8))

# Draw the heatmap with the correlation matrix
sns.heatmap(correlation_matrix, annot=True, cmap='coolwarm', vmin=-1, vmax=1,
            linewidths=0.5)

# Add title to the heatmap
plt.title('Correlation Matrix of Features')

# Display the plot
plt.show()
```

Output :



A correlation matrix heatmap showing the relationships between the features in the dataset. Each cell in the matrix displays a correlation coefficient, with color indicating the strength and direction of the relationship. Positive correlations are shown in shades of red, while negative correlations are in blue.

MINI PROJECT

AIM: Consider any dataset from an online repository to design and implement aproblem using linear regression and logistic regression.

Theory:

- **LINEAR REGRESSION**

Linear Regression is a fundamental statistical method used to model the relationship between a dependent variable and one or more independent variables. It is widely used for predictive analysis and understanding the relationship between variables.

- LOGISTIC REGRESSION

Logistic Regression is a statistical method used for binary classification problems, where the outcome variable is categorical and typically represents two classes (e.g., success/failure, yes/no, spam/not spam). It models the probability that a given input point belongs to a particular class.

A statistical technique for binary classification, logistic regression, is carried out by the given code. The likelihood that a given input belongs to a specific class is predicted via logistic regression. The model generates a probability score between 0 and 1 using a sigmoid function. The input can then be classified into one of two classes by thresholding this probability, which is typically set at 0.5.

- Libraries: The code imports essential libraries like pandas for data manipulation,

numpy for numerical operations, and sklearn (Scikit-Learn) for machine learning tasks.

- Load the Dataset

The code loads a dataset from a CSV file.

Here, X (the feature set) contains the independent variables, which include columns like ‘Total Cases', 'Discharged', 'Population'.

y (the target variable) is set to the Deaths column, which is the variable we are trying to predict.

- **Preprocess the Data**

Handle missing values and select relevant features.

- **Split the Data**

Split the dataset into training and testing sets.

- Train the Linear Regression Model and logistic regression
- Make Predictions with Linear Regression
- Evaluate the Linear Regression Model

Calculate evaluation metrics like Mean Squared Error (MSE) and R-squared (R^2).

- Train the Logistic Regression Model
- Make Predictions with Logistic Regression

- **Evaluate the Logistic Regression Model**

Calculate evaluation metrics like accuracy, confusion matrix, and classification report.

- **Visualize the Results**

Plot the confusion matrix for logistic regression

A confusion matrix shows the count of True Positives (TP), True Negatives (TN), False Positives (FP), and False Negatives (FN).

Sensitivity (Recall) and Specificity:

sensitivity = TP / (TP + FN) specificity = TN / (TN + FP)

Sensitivity measures the model’s ability to correctly identify positive cases. Specificity measures the model’s ability to correctly identify negative cases