A

# Data mining Lab

**PCC-DS-403**

Submitted during 4 th semester in partial fulfilment of the requirements for the award of degree of

**Bachelor of Technology**

in

**Computer Engineering (Spec. In Data Science)**

By

**Mansi (23001016038)**

Under supervision of

**Dr. Disha Garg**

****

**Department of Computer Engineering**

**J.C. Bose University of Science & Technology,**

**YMCA Faridabad – 121006**

**Introduction to Python**

**What is Python?**

Python is a high-level, interpreted programming language known for its simplicity, readability, and versatility. Created by Guido van Rossum and first released in 1991, Python was designed to emphasize code readability and ease of use. Its clean syntax, which closely resembles the English language, makes it especially appealing to beginners, while its powerful features make it equally popular among professional developers.

**Key Features of Python**

One of Python’s most significant advantages is its straightforward syntax. Unlike many other programming languages, Python uses indentation to define code blocks, which not only reduces the need for extra symbols but also makes the code more readable and maintainable.

Python is also dynamically typed, meaning you don’t have to declare variable types explicitly. It supports multiple programming paradigms, including procedural, object-oriented, and functional programming. These features give developers the flexibility to use different approaches based on the nature of their project.

Another strength of Python is its portability. Python code can run on various operating systems without modification, making it a cross-platform language. It is also open-source, meaning it is freely available for anyone to use, modify, and distribute.

**Applications of Python**

Python is a general-purpose language, used in a wide array of fields. It is particularly strong in:

* **Web Development**: Frameworks like Django and Flask help developers build robust and scalable web applications.
* **Data Science and Analytics**: Libraries like pandas, NumPy, and Matplotlib have made Python a standard tool for data manipulation and visualization.
* **Machine Learning and Artificial Intelligence**: Tools such as TensorFlow and scikit-learn enable developers to build intelligent systems and predictive models.
* **Automation and Scripting**: Python is widely used to automate repetitive tasks, manage system operations, and handle file processing.
* **Cybersecurity**: Python is favored for writing security tools and penetration testing scripts due to its efficiency and large collection of libraries.

**Python in Education**

Python is often the first programming language taught in schools and universities. Its easy-to-understand syntax allows students to focus on learning programming concepts rather than struggling with complex syntax rules. Additionally, it is widely used in non-computer science fields such as biology, finance, and social sciences, where researchers use Python for data analysis and visualization.

**Community and Support**

Python boasts one of the largest and most active programming communities in the world. This ensures a wealth of learning resources, forums, tutorials, and third-party tools. The Python Software Foundation (PSF) oversees the development of the language and supports various community initiatives such as conferences, workshops, and educational programs.

**Conclusion**

Python’s simplicity, readability, and versatility make it a powerful and accessible tool for a wide range of applications. From beginners learning their first lines of code to professionals building complex systems, Python caters to all. Its continued growth, strong community, and wide industry adoption position Python as one of the most valuable programming languages in the world today.

**Libraries in python**

Python has become a cornerstone language in data science, machine learning, and scientific computing due to its extensive ecosystem of libraries. Among the most essential are **NumPy**, **Pandas**, **Matplotlib**, **Seaborn**, **SciPy**, and **Scikit-learn (sklearn)**. Each library plays a critical role in data processing, analysis, visualization, and modeling. This document provides an in-depth overview of these libraries, their uses, and simple example programs.

**1. NumPy (Numerical Python)**

**Introduction:**

NumPy is the foundational library for numerical computing in Python. It introduces a powerful N-dimensional array object, along with a collection of mathematical functions to operate on these arrays.

**Key Features:**

* Provides support for arrays and matrices.
* Performs element-wise operations efficiently.
* Offers broadcasting capabilities.
* Contains linear algebra, Fourier transform, and random number capabilities.

**Simple Example:**

import numpy as np

arr = np.array([1, 2, 3, 4, 5])

print("Mean:", np.mean(arr))

**Uses in Python:**

* Serves as the base for other libraries like Pandas, SciPy, and Scikit-learn.
* Enables efficient storage and manipulation of numerical data.
* Facilitates numerical operations in machine learning and scientific applications.
* Often used in image and signal processing.

**2. Pandas (Panel Data)**

**Introduction:**

Pandas is the go-to library for data manipulation and analysis. It introduces two primary data structures: Series (1D) and DataFrame (2D), which make it easy to handle structured data.

**Key Features:**

* Offers data alignment and handling of missing data.
* Facilitates data filtering, merging, reshaping, and aggregation.
* Provides built-in functions for time series data.
* Easily reads and writes data to and from multiple file formats (CSV, Excel, SQL, etc.).

**Simple Example:**

import pandas as pd

data = {'Name': ['Alice', 'Bob'], 'Age': [25, 30]}

df = pd.DataFrame(data)

print(df.describe())

**Uses in Python:**

* Used for data cleaning, transformation, and exploration.
* Commonly used in data science workflows before visualization or machine learning.
* Supports time series analysis.
* Integrates smoothly with NumPy, Matplotlib, and Scikit-learn.

**3. Matplotlib**

**Introduction:**

Matplotlib is a comprehensive library for creating static, animated, and interactive visualizations in Python. It is highly customizable and serves as the foundation for other visualization libraries.

**Key Features:**

* Offers various plot types (line, bar, scatter, pie, etc.).
* Provides fine control over plot appearance.
* Supports subplots and complex layouts.
* Can output in multiple formats (PNG, PDF, SVG).

**Simple Example:**

import matplotlib.pyplot as plt

x = [1, 2, 3, 4]

y = [10, 20, 25, 30]

plt.plot(x, y)

plt.title("Simple Line Plot")

plt.show()

**Uses in Python:**

* Used for exploratory data analysis.
* Helps in visualizing patterns, distributions, and trends.
* Ideal for generating publication-quality figures.
* Compatible with Jupyter Notebooks for inline plotting.

**4. Seaborn**

**Introduction:**

Seaborn is a statistical data visualization library built on top of Matplotlib. It simplifies complex visualizations and integrates closely with Pandas data structures.

**Key Features:**

* Automatically manages aesthetics and themes.
* Supports advanced plots like violin plots, heatmaps, and pair plots.
* Works directly with Pandas DataFrames.
* Offers built-in functions for statistical aggregation and analysis.

**Simple Example:**

import seaborn as sns

import pandas as pd

df = pd.DataFrame({'x': [1, 2, 3, 4], 'y': [10, 15, 20, 25]})

sns.scatterplot(x='x', y='y', data=df)

**Uses in Python:**

* Enhances data storytelling with minimal code.
* Ideal for statistical exploration of datasets.
* Often used in conjunction with Matplotlib and Pandas for rich insights.
* Helps in visualizing relationships and distributions.

**5. SciPy (Scientific Python)**

**Introduction:**

SciPy builds on NumPy and provides additional modules for scientific and technical computing. It is widely used in mathematics, engineering, and physical sciences.

**Key Features:**

* Contains modules for optimization, integration, interpolation, signal and image processing, and linear algebra.
* Provides high-level commands for manipulating data.
* Highly optimized for performance.

**Simple Example:**

from scipy import stats

data = [2, 4, 4, 4, 5, 5, 7, 9]

print("Mode:", stats.mode(data).mode[0])

**Uses in Python:**

* Used in solving differential equations, optimizations, and statistical tests.
* Integral in computational biology, astronomy, and physics.
* Helpful in building simulations and models in scientific research.
* Commonly used in academic and research-based Python projects.

**6. Scikit-learn (sklearn)**

**Introduction:**

Scikit-learn is the most widely used machine learning library in Python. It provides simple and efficient tools for data mining and data analysis, built on NumPy, SciPy, and Matplotlib.

**Key Features:**

* Offers algorithms for classification, regression, clustering, and dimensionality reduction.
* Supports model selection and evaluation through cross-validation and metrics.
* Easy to use with a consistent API.
* Integrates seamlessly with Pandas and NumPy.

**Simple Example:**

from sklearn.linear\_model import LinearRegression

import numpy as np

X = np.array([[1], [2], [3], [4]])

y = np.array([2, 4, 6, 8])

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

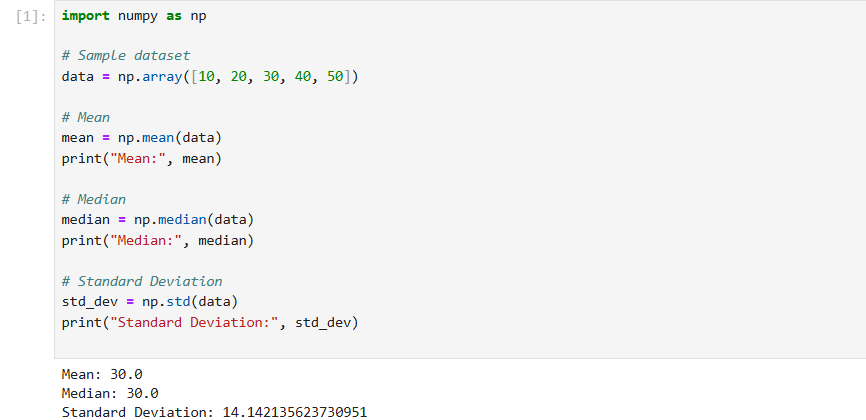
print("Predicted:", model.predict([[5]]))

**Uses in Python:**

* Used in building predictive models from data.
* Ideal for developing prototypes and running quick experiments.
* Supports pipelines for streamlined workflows.
* Used extensively in industry for practical machine learning applications.

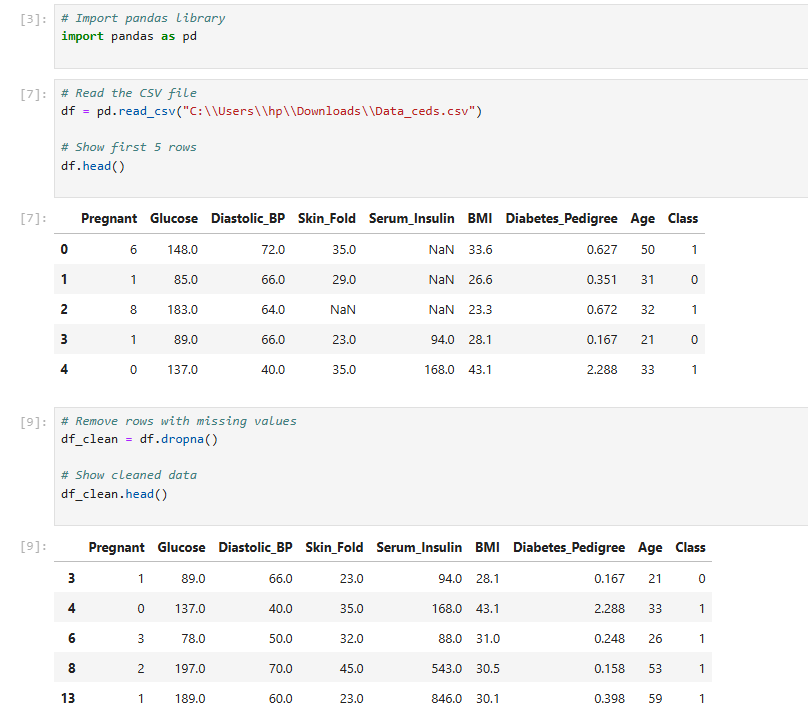
**Program 1**

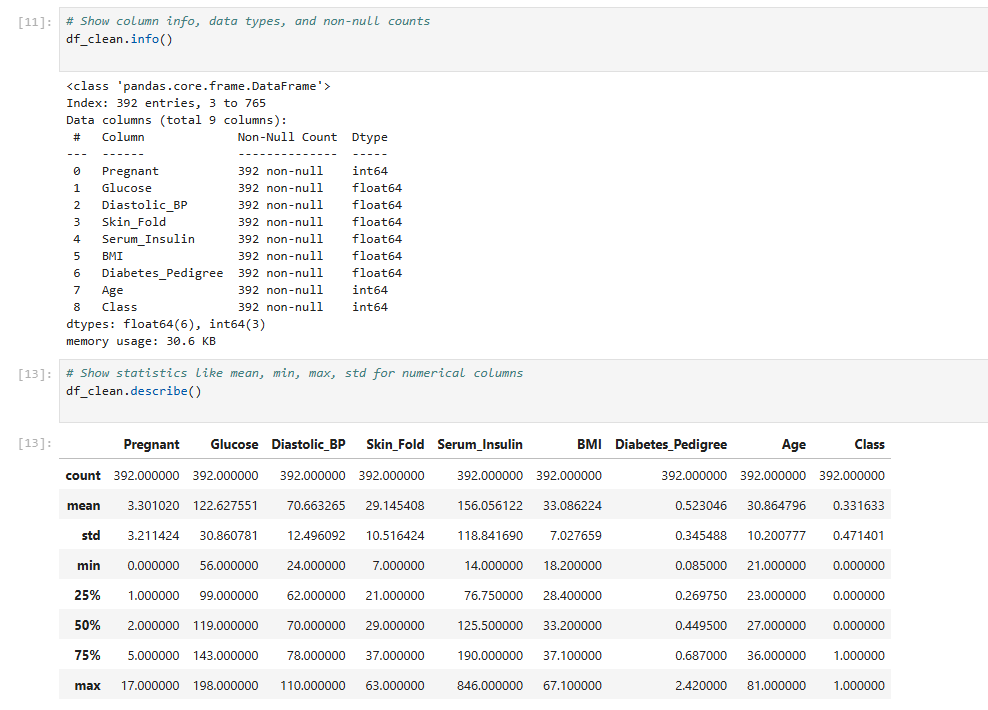
**Aim:**Create a python program to calculate mean ,median and standard deviation of a dataset using numpy array.

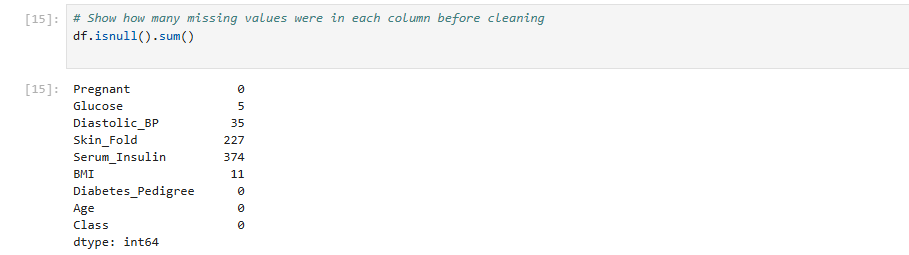


**Program 2**

**Aim:** Create a program to read a CSV file to clean the data and perform some basic analysis using pandas.

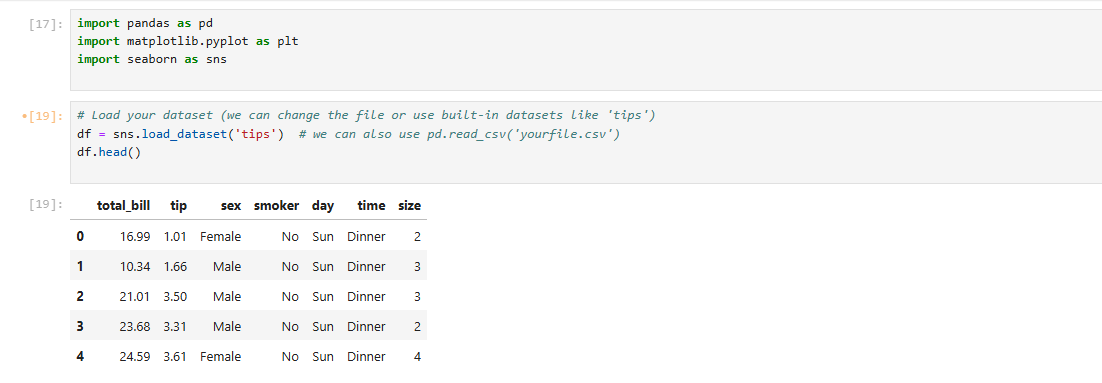


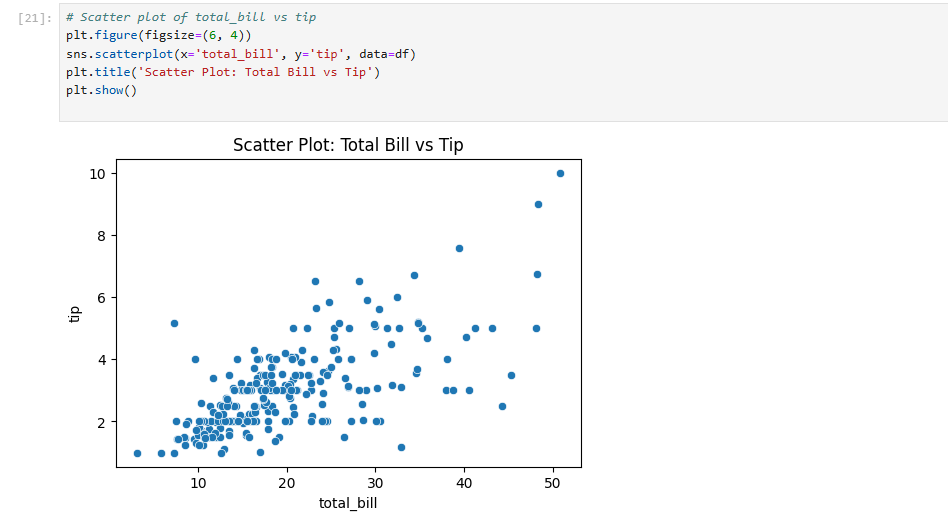
****

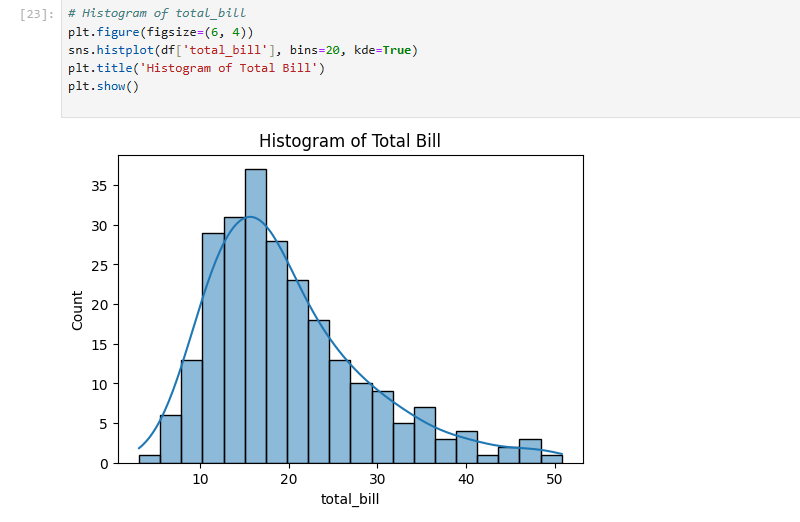
****

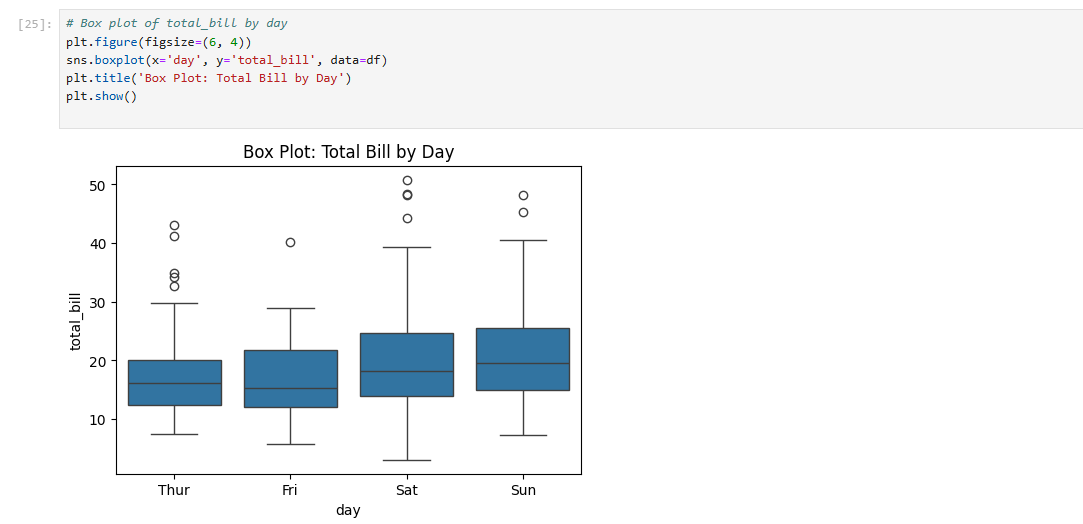
**Program 3**

**Aim:**Create a program to plot a scatter plot a histogram box plot using matplotlib and seaborn.

****

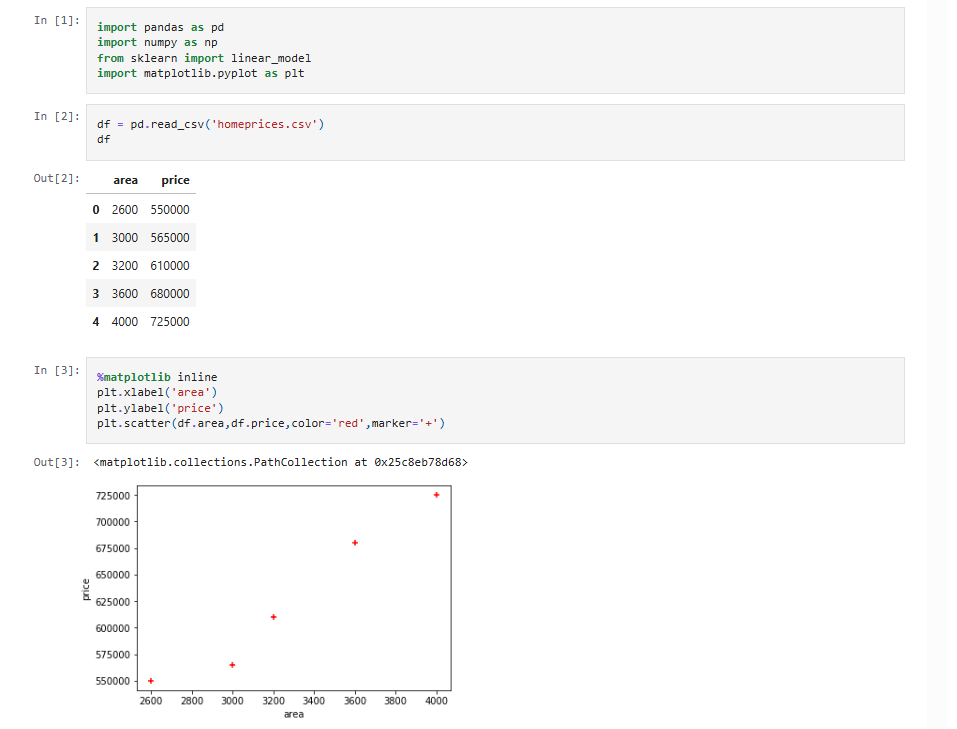
****

****

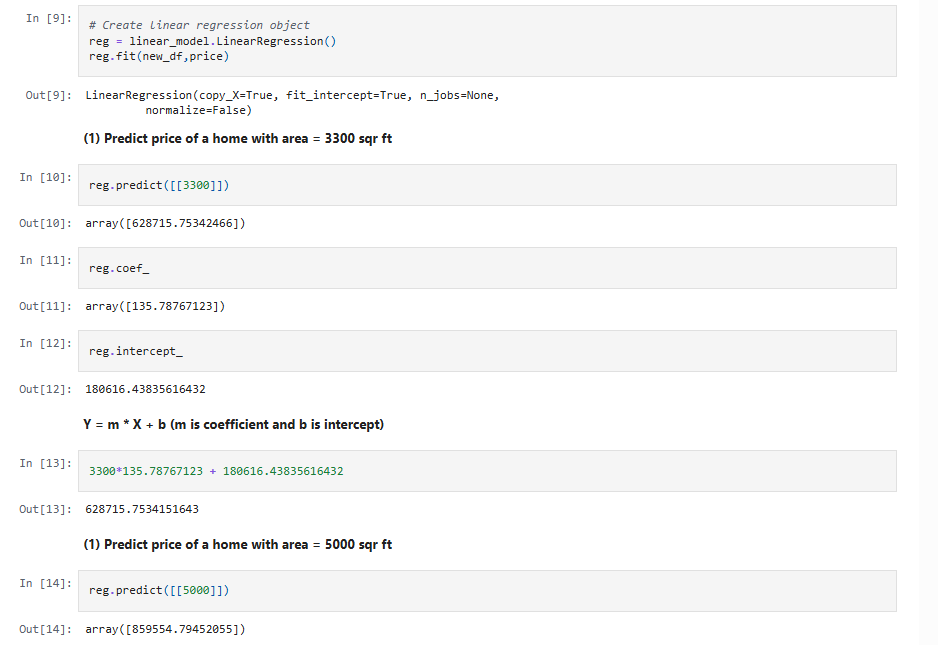
****

**Program 4**

**Aim:**Write a python program to implement linear regression.

****

****

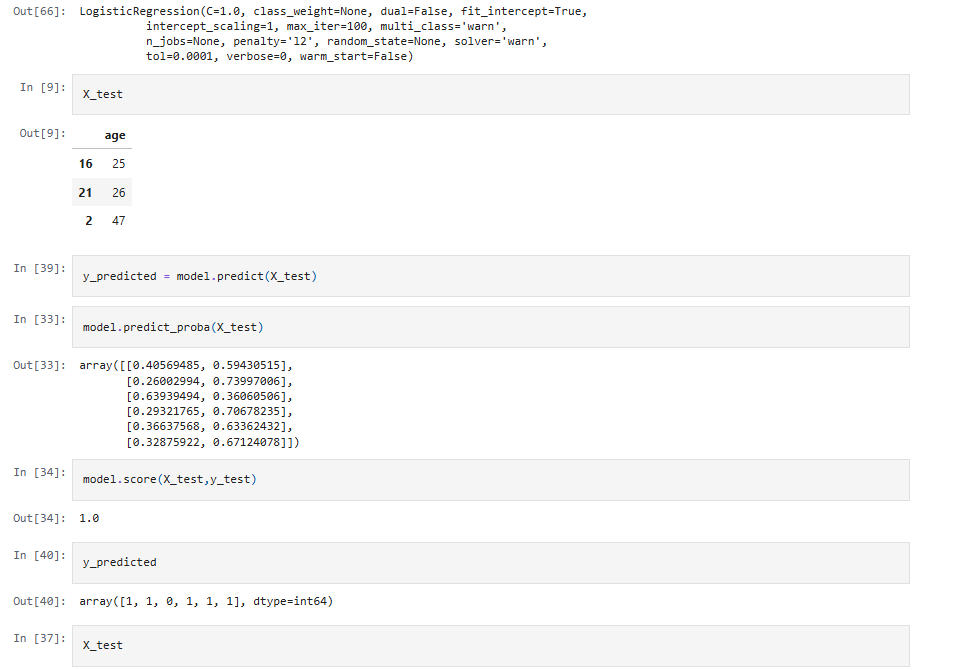
****

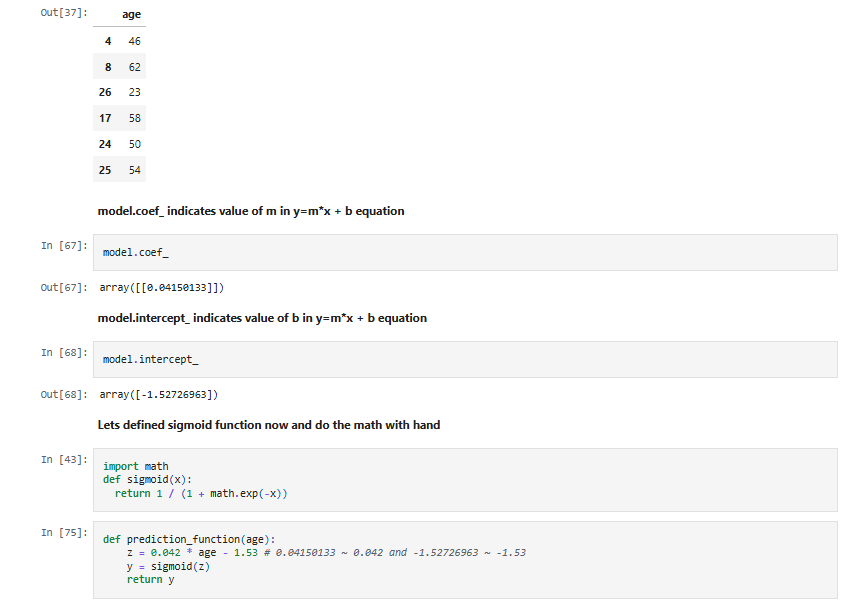
**Program 5**

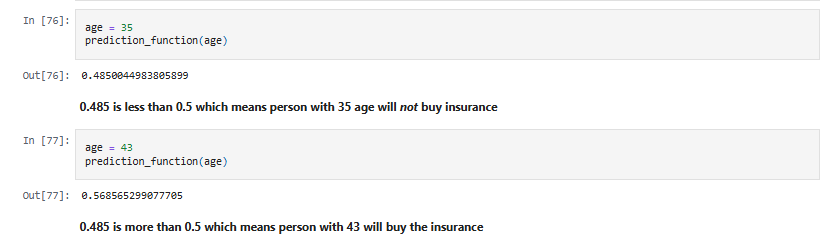
**Aim:**Write a python program to implement logistic regression.

****

****

****

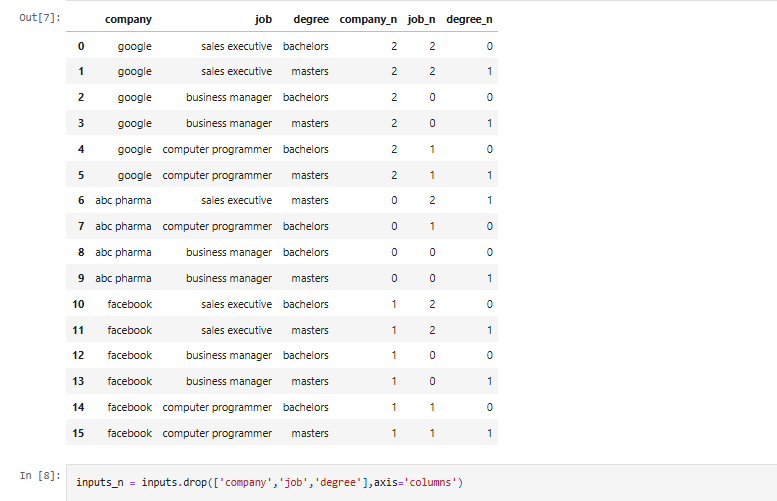
****

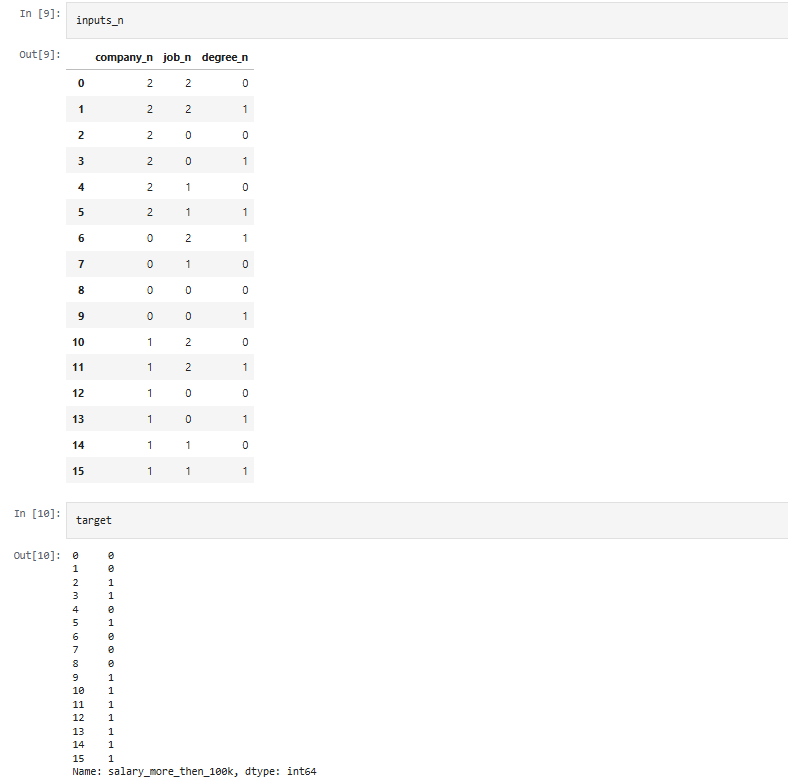
****

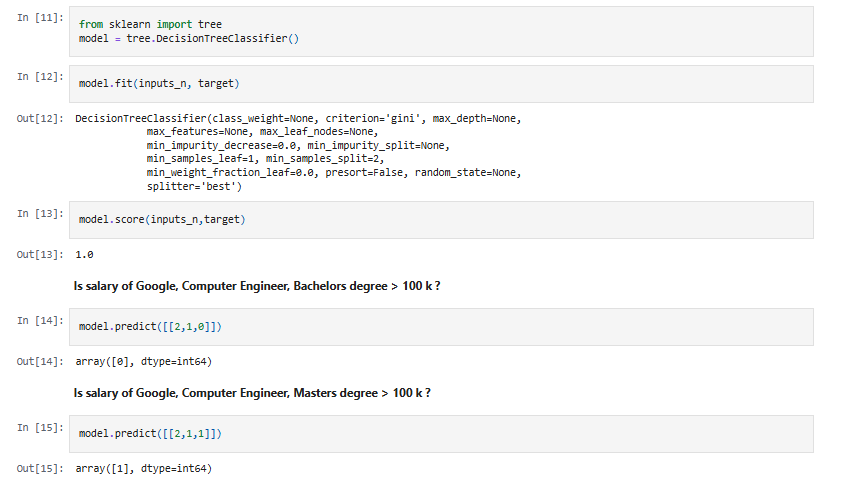
**Program 6**

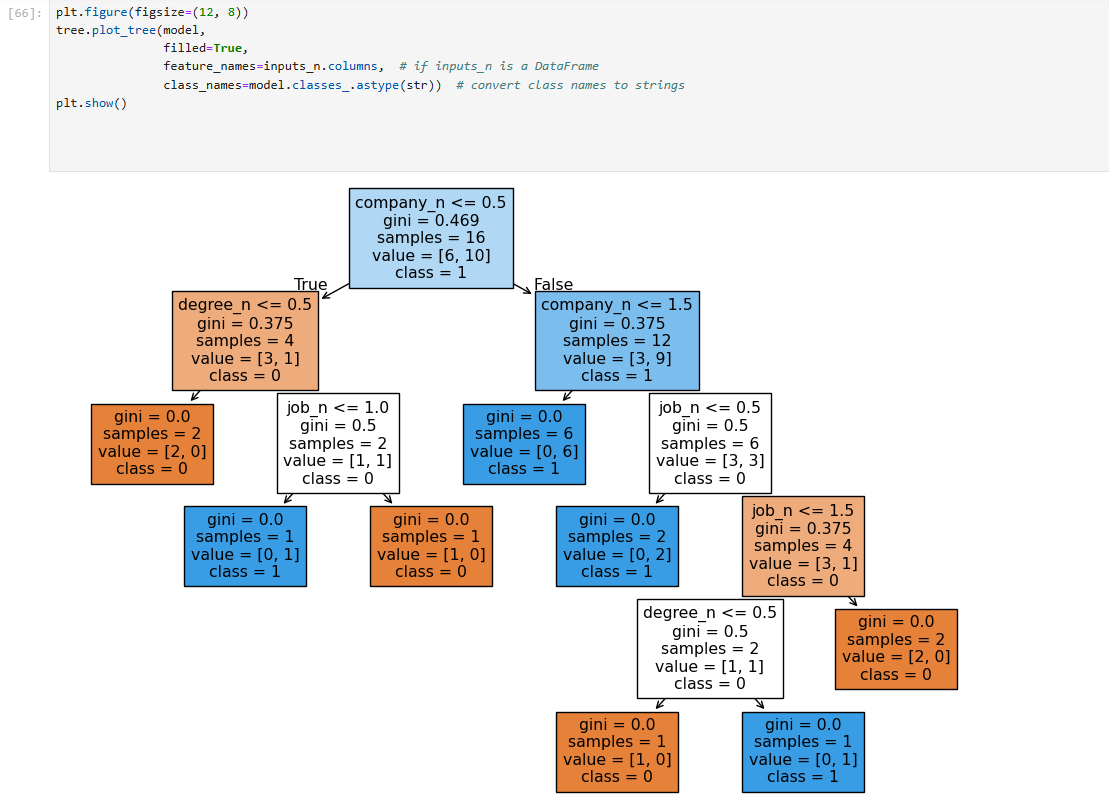
**Aim:**Write a program to implement decision tree.

****

****

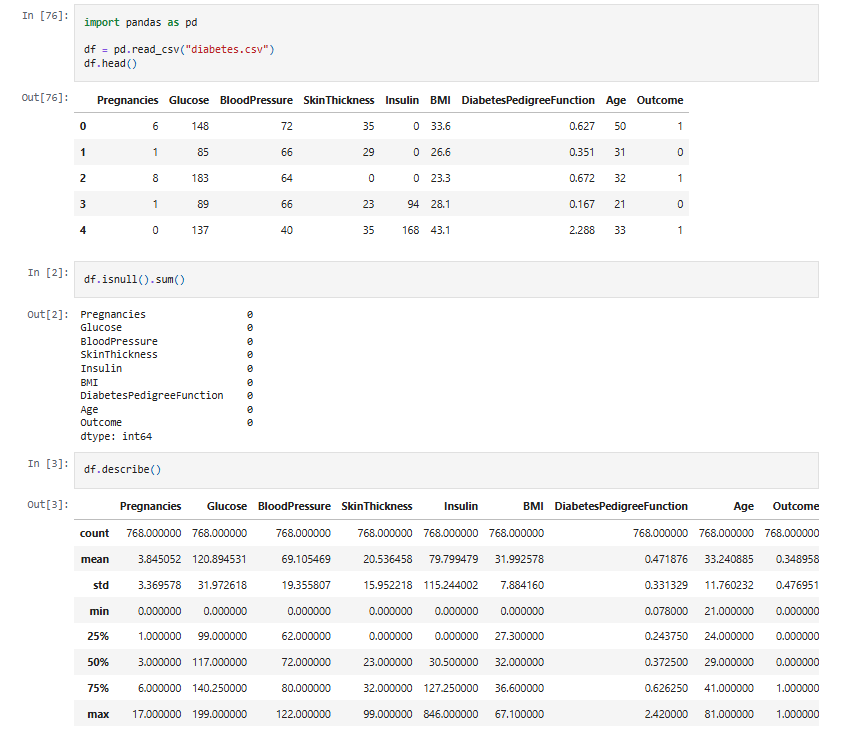
****

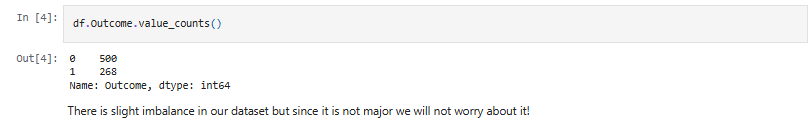
****

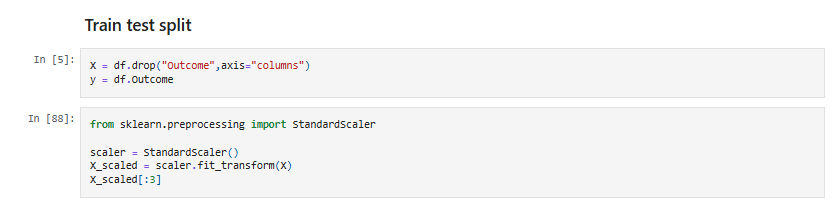
****

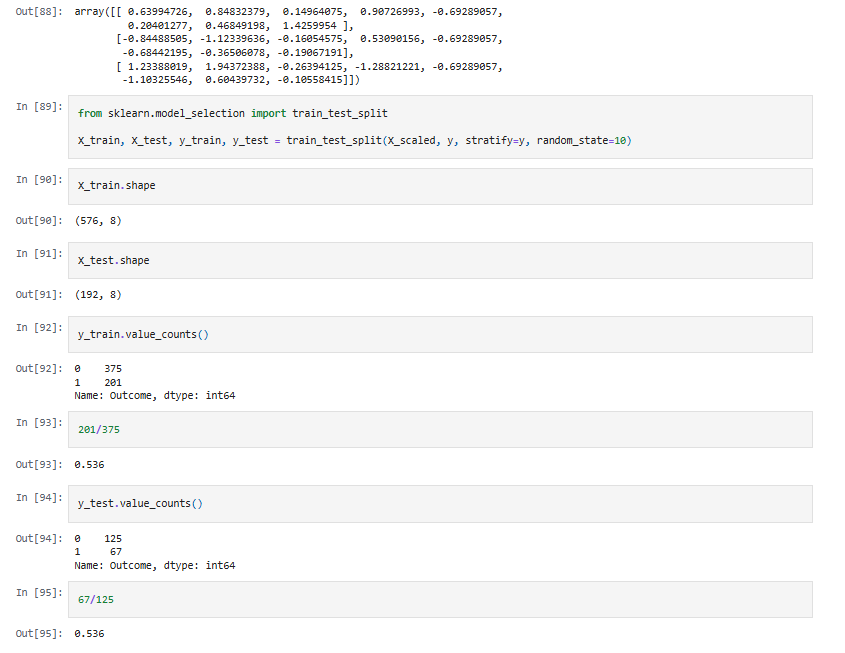
**Program 7**

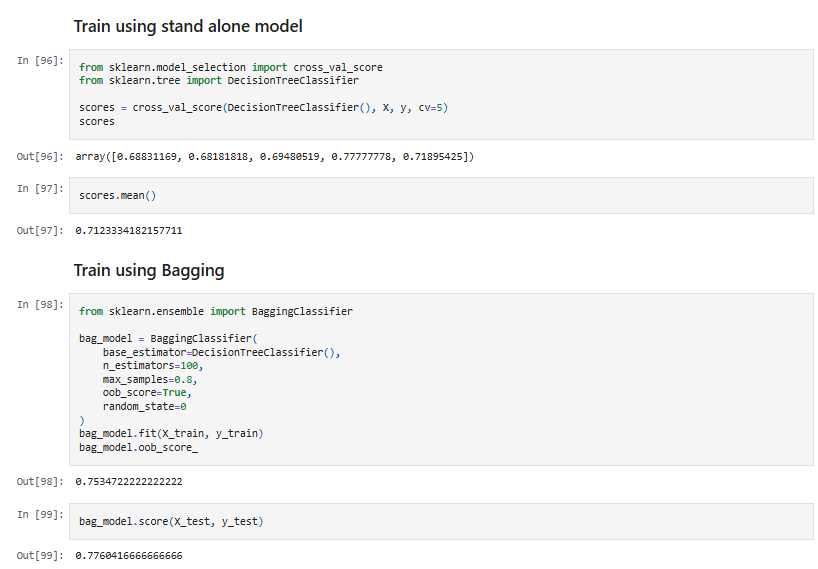
**Aim:**Write a python program to implement ensemble techniques.

****

****

****

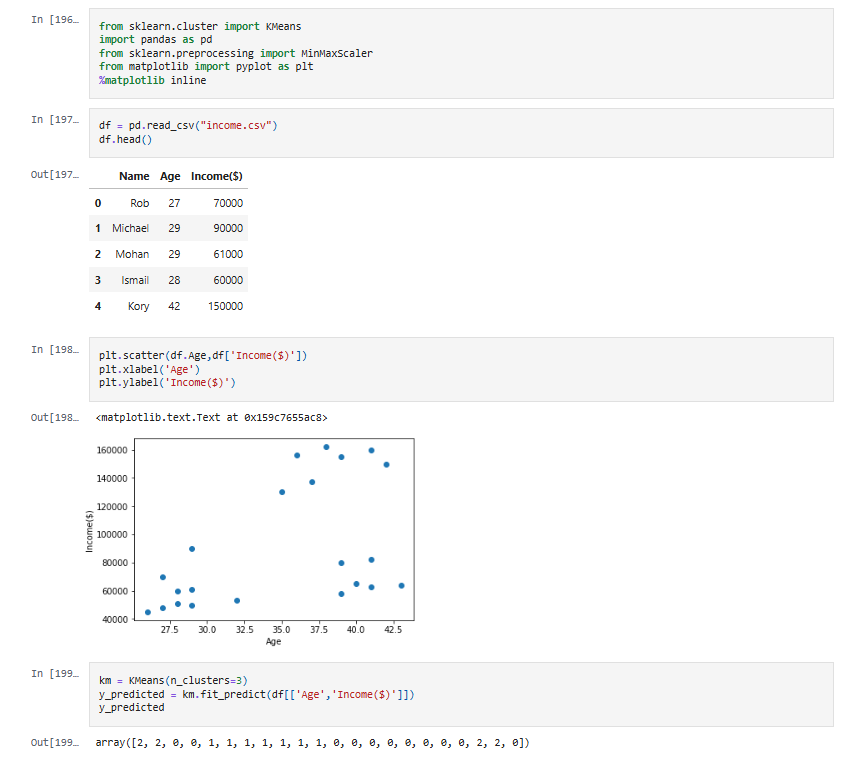
****

****

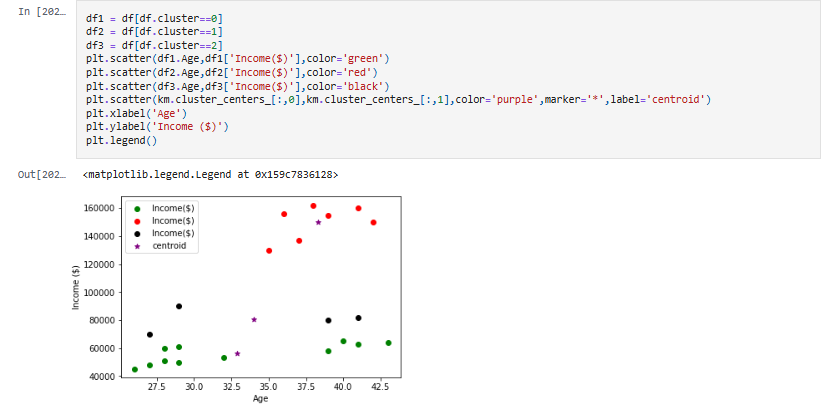
****

**Program 8**

**Aim:** Write a python program to implement clustering.

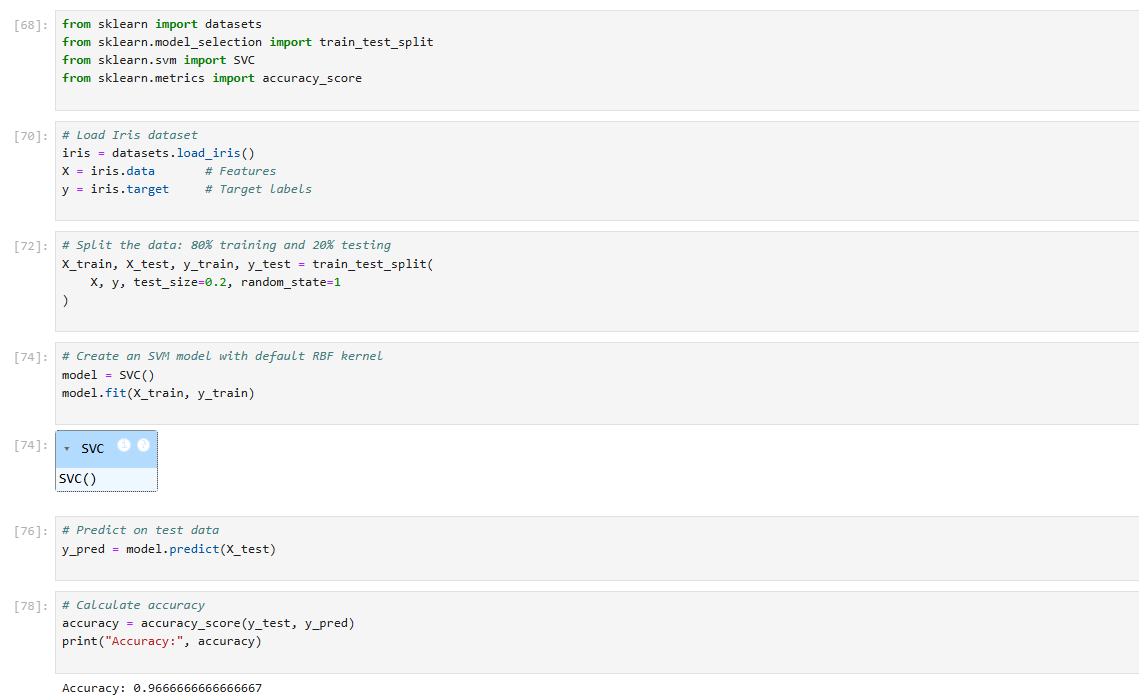
****

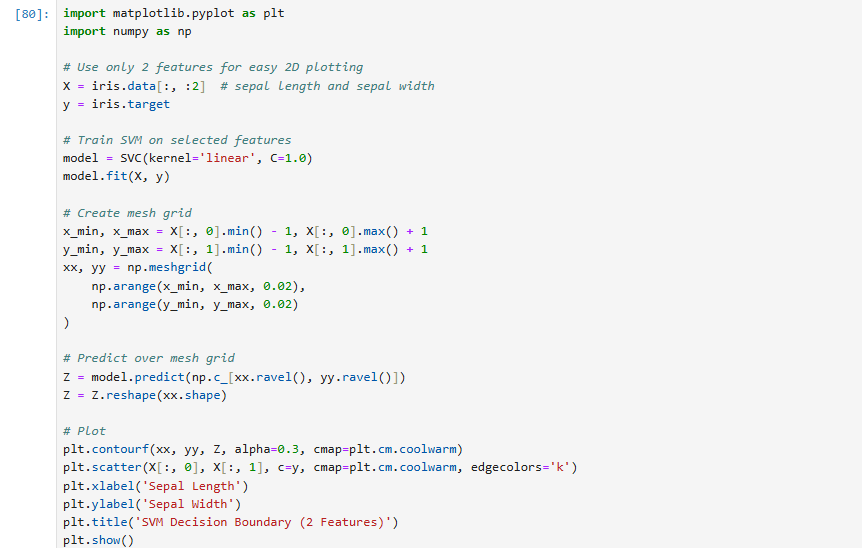
****

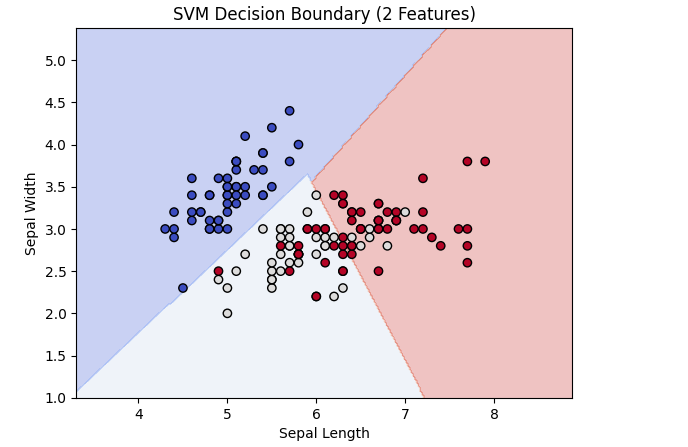
****

**Program 9**

**Aim:** Write a python program to implement SVM (Support Vector Machine).

****

****

****

**Program 10**

**Aim:** Write a python program to implement KNN classifier .

****

**INDEX**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Sr.no | Name of program | Date | Page no. | Signature |
| **1** | Introduction to Python |  |  |  |
| **2** | Create a python program to calculate mean ,median and standard deviation of a dataset using numpy array. |  |  |  |
| **3** | create a program to read a CSV file to clean the data and perform some basic analysis using pandas. |  |  |  |
| **4** | Create a program to plot a scatter plot a histogram box plot using matplotlib and seaborn. |  |  |  |
| **5** | Write a python program to implement Linear Regression. |  |  |  |
| **6** | Write a python program to implement Logistic Regression. |  |  |  |
| **7** | Write a python program to implement Decision Tree. |  |  |  |
| **8** | Write a python program to implement Ensemble Techniques. |  |  |  |
| **9** | Write a python program to implement Clustering. |  |  |  |
| **10** | Write a python program to implement SVM (Support Vector Machine). |  |  |  |