**LINEAR REGRESSION:**

**CODE:**

import numpy as np

import pandas as pd

#"Importing the dataset

# divide the dataset into concepts and targets. Store the concepts into X and targets into y.

dataset = pd.read\_csv("C:\\Users\\prith\\Desktop\\MACHINE LEARNING\\breastcancer.csv")

X = dataset.iloc[:, :-1].values

y = dataset.iloc[:, -1].values

#Splitting the dataset into the Training set and Test

from sklearn.model\_selection import train\_test\_split

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size = 0.30, random\_state = 2)

#Feature Scaling

from sklearn.preprocessing import StandardScaler

sc = StandardScaler()

X\_train = sc.fit\_transform(X\_train)

X\_test = sc.transform(X\_test)

from sklearn.linear\_model import LogisticRegression

classifier = LogisticRegression(random\_state = 0)

classifier.fit(X\_train, y\_train)

#Logistic Regression (LR) classifier model

#Display the results (confusion matrix and accuracy)

from sklearn.metrics import confusion\_matrix, accuracy\_score

y\_pred = classifier.predict(X\_test)

cm = confusion\_matrix(y\_test, y\_pred)

print(cm)

print('Accuracy Score:confusion matrix')

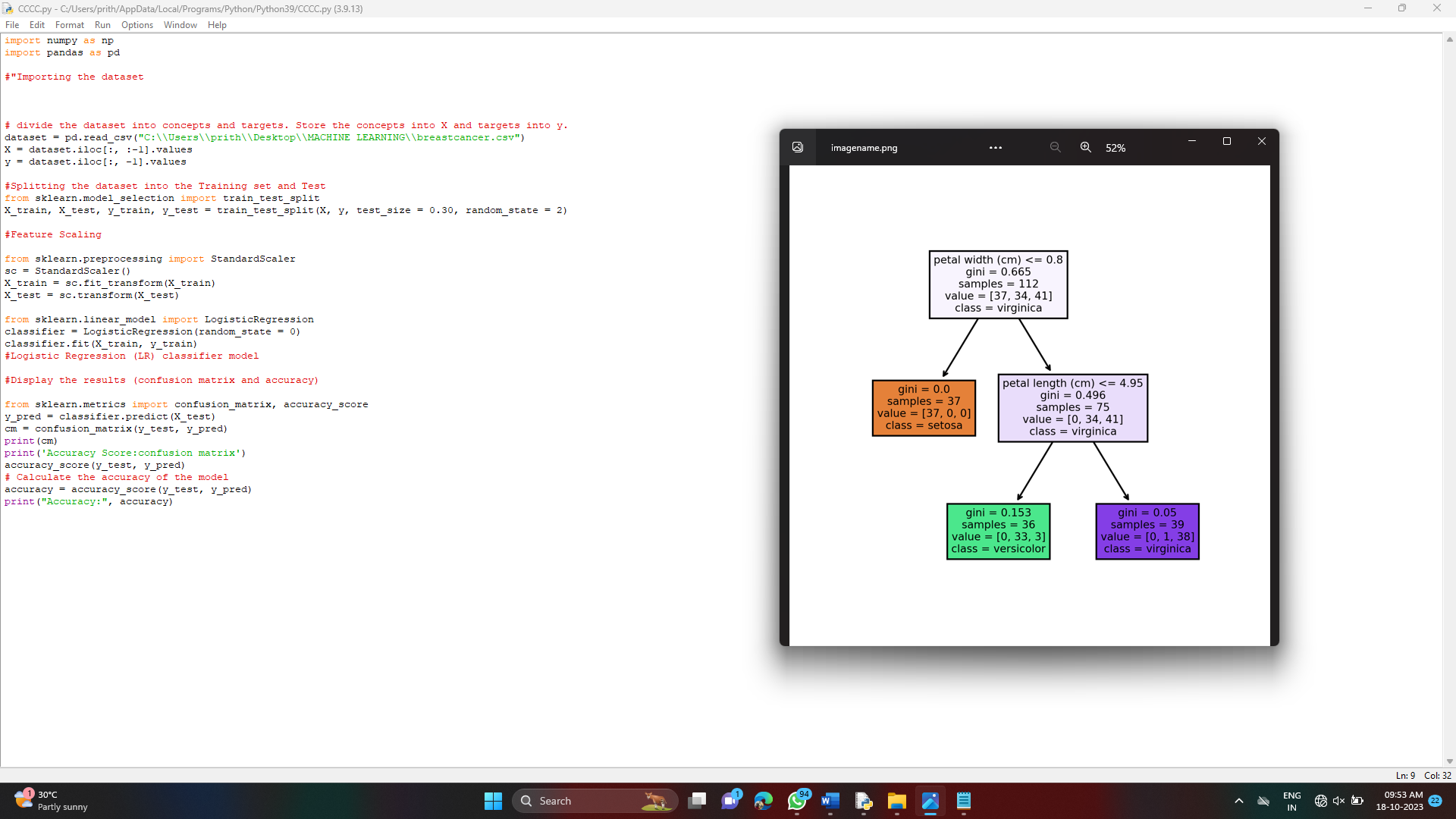
accuracy\_score(y\_test, y\_pred)

# Calculate the accuracy of the model

accuracy = accuracy\_score(y\_test, y\_pred)

print("Accuracy:", accuracy)

**OUTPUT:**



**CODE:**

import pandas as pd

from sklearn.model\_selection import train\_test\_split

from sklearn.linear\_model import LinearRegression

from sklearn.metrics import mean\_squared\_error

import matplotlib.pyplot as plt

# Load the dataset

dataset = pd.read\_csv('Salary\_Data.csv')

# Split the dataset into independent variables (X) and dependent variable (y)

X = dataset.iloc[:, :-1].values

y = dataset.iloc[:, -1].values

# Split the dataset into training and testing sets

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

# Create an instance of the Linear Regression model

model = LinearRegression()

# Fit the model to the training data

model.fit(X\_train, y\_train)

# Predict the salaries for the test data

y\_pred = model.predict(X\_test)

#model good or not

mse = mean\_squared\_error(y\_test, y\_pred)

print("Mean Squared Error:", mse)

#Visualising the Training set results Here scatter plot is used to visualize the results.

plt.scatter(X\_train, y\_train, color = 'red')

plt.plot(X\_train, model.predict(X\_train), color = 'blue')

plt.title('Salary vs Experience (Training set)')

plt.xlabel('Years of Experience')

plt.ylabel('Salary')

plt.show()

**OUTPUT:**

================================= RESTART: C:/Users/prith/Desktop/MACHINE LEARNING/LINEAR.py ================================

Mean Squared Error: 49830096.85590839

