EX. NO: 07

220801153

DATE : 12/04/2024

Linear regression model to predict the signal strength

AIM:

To develop a linear regression model to predict the signal strength based on the distance.

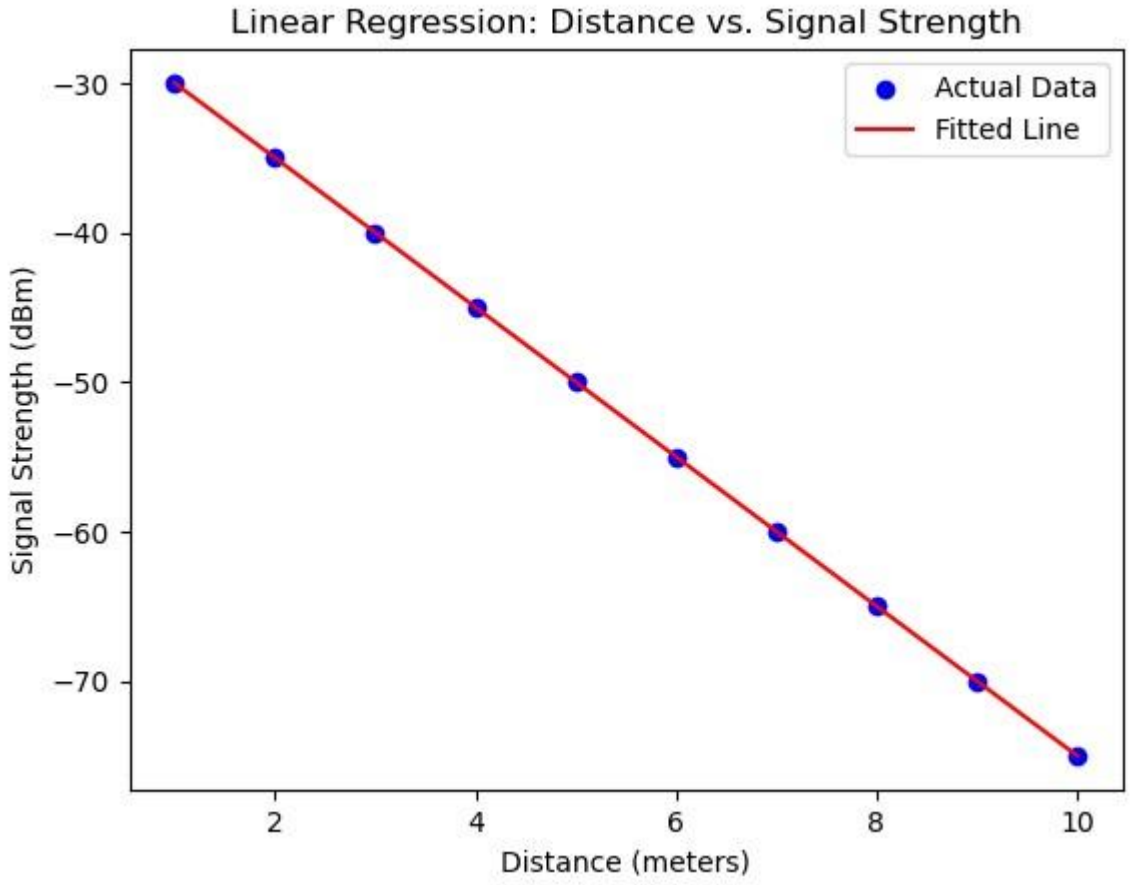
PROGRAM:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score

# Example dataset: Distance (meters) vs. Signal Strength (dBm)
data = {    'Distance': [1, 2, 3, 4, 5, 6, 7, 8, 9, 10],
    'Signal_Strength': [-30, -35, -40, -45, -50, -55, -60, -65, -70, -75]
}
# Convert the data into a DataFrame
df = pd.DataFrame(data)

# Separate features and target variable
X = df[['Distance']].values  # Feature: Distance
y = df['Signal_Strength'].values  # Target: Signal Strength

# Split the data 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 and train the linear regression model
model = LinearRegression()
model.fit(X_train, y_train)

# Make predictions
y_pred = model.predict(X_test)

# Evaluate the model

mse = mean_squared_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)

print(f'Mean Squared Error: {mse:.2f}')
print(f'R^2 Score: {r2:.2f}')

# Visualize the results
plt.scatter(X, y, color='blue', label='Actual Data')
plt.plot(X, model.predict(X), color='red', label='Fitted Line')
plt.xlabel('Distance (meters)')
plt.ylabel('Signal Strength (dBm)')
plt.title('Linear Regression: Distance vs. Signal Strength')
plt.legend()
plt.show()
```

OUTPUT:

Mean Squared Error: 0.00

R^2 Score: 1.00



RESULT :

The program was executed successfully.