

Machine Learning Lab

Reg No: 23MCA1030

Name : Vinayak Kumar Singh

Exercise 2 : Multiple Linear Regression

House Price Prediction Dataset from LMS

Dataset url : https://lms.vit.ac.in/pluginfile.php/291531/mod_assign/introattachment/0/House%20Price%20prediction_Dataset.xlsx?forcedownload=1

Collab url : <https://colab.research.google.com/drive/15XIVRvtU5kiTnaDhQtUR1U1SDMZgRXIY?usp=sharing>

```
import pandas
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn import metrics

df = pandas.read_csv("/content/House_Price_prediction_Dataset.csv")

x = df[['Sqft', 'LotSize', 'Baths']]
y = df['Price']

regr = LinearRegression()
regr.fit(x, y)
#predict the Price of a house at 2600 Sqft with LotSize 24.1 and Baths as 3.5
predictedPrice = regr.predict([[2600, 24.1, 3.5]])

/usr/local/lib/python3.10/dist-packages/sklearn/base.py:439:
UserWarning: X does not have valid feature names, but LinearRegression
was fitted with feature names
  warnings.warn(

print(predictedPrice)

[633.20983301]

# Split the dataset into training and testing sets
# training set that contains 75% of the data and a testing set that
contains 25% of the data
x_train, x_test, y_train, y_test = train_test_split(x, y,
test_size=0.25, random_state=42)
```

Create a linear regression model

```

model = LinearRegression()
# Fit the model to the training data
model.fit(x_train, y_train)

LinearRegression()

# Make predictions on the test set
y_pred = model.predict(x_test)
y_pred = model.predict([[2600, 24.1, 3.5]])
print(y_pred)

[632.84779384]

/usr/local/lib/python3.10/dist-packages/sklearn/base.py:439:
UserWarning: X does not have valid feature names, but LinearRegression
was fitted with feature names
  warnings.warn(

# Evaluate the model
import numpy as np
from sklearn import metrics

y_test = np.array([[633.20983301]])
y_pred = np.array([[632.84779384]])

# Reshape the `y_test` array to the same shape as the `y_pred` array
y_test = y_test.reshape(-1, 1)

# Print the evaluation metrics
print('Mean Absolute Error:', metrics.mean_absolute_error(y_test,
y_pred))
print('Mean Squared Error:', metrics.mean_squared_error(y_test,
y_pred, squared=False))
print('Root Mean Squared Error:',
np.sqrt(metrics.mean_squared_error(y_test, y_pred, squared=False)))
print("R2-score: %.2f:" % metrics.r2_score(y_test, y_pred))

Mean Absolute Error: 0.362039170000000274
Mean Squared Error: 0.362039170000000274
Root Mean Squared Error: 0.6016969087505791
R2-score: nan:

/usr/local/lib/python3.10/dist-packages/sklearn/metrics/
_regression.py:918: UndefinedMetricWarning: R^2 score is not well-
defined with less than two samples.
  warnings.warn(msg, UndefinedMetricWarning)

# Print the coefficients and intercept
print('Coefficients:', model.coef_)
print('Intercept:', model.intercept_)

```

```
Coefficients: [ 0.16923473  5.98949191 18.30724368]  
Intercept: -15.584623005995923
```