CS5542 BIG DATA APPS AND ANALYTICS

In Class Programming – 9

Regression

Kalyani Nikure
Kmn6bg@umkc.edu

Table of Contents

Description

Create a linear regression model in python using any dataset of your choice. For this model you can also create your own data. Find the best fit line in the data and calculate SSE (sum of square error) or MSE (Mean square error), Y intercept, and Slope for the relationship in data. Explain your findings and understanding of these terms in detail in the report.

Detailed Steps Explanation

1. Importing the libraries

```
[ ]  !pip install hvplot

[111] import pandas as pd
      import numpy as np
      import matplotlib.pyplot as plt
      import seaborn as sns
      import hvplot.pandas
      %matplotlib inline

      sns.set_style("whitegrid")
      plt.style.use("fivethirtyeight")
```

2. About dataset

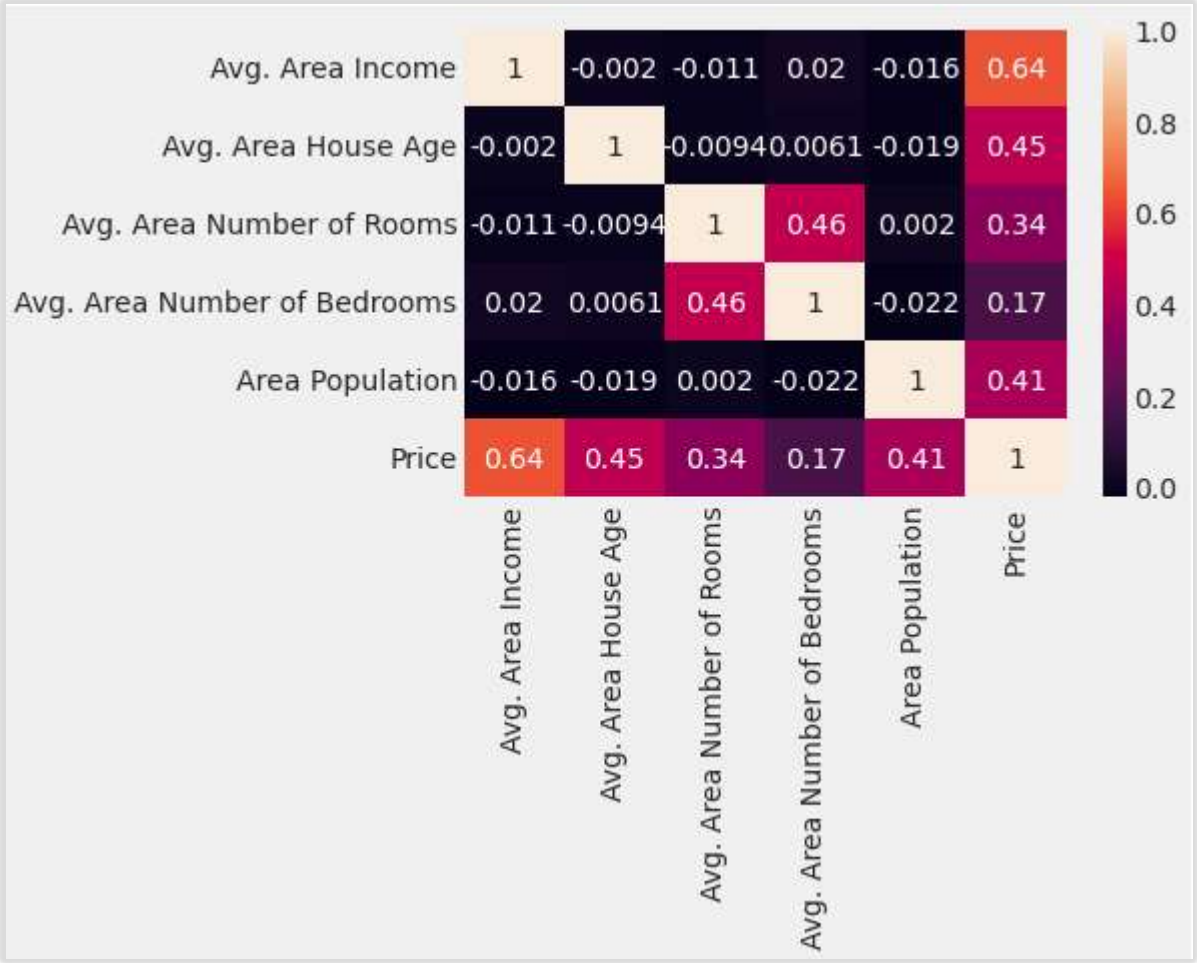
I have used a housing dataset for which contains USA Housing Prices with detailed information of other related columns like Avg. Area Income, Avg. Area House Age, Avg. Area Number of Rooms, Avg. Area Number of Bedrooms, Area Population, Price and Address of about 5000 houses. We will be creating a LR model to predict the house prices.

3. Loading the data

```
data = pd.read_csv('/content/USA_Housing.csv')
data.head()
```

	Avg. Area Income	Avg. Area House Age	Avg. Area Number of Rooms	Avg. Area Number of Bedrooms	Area Population	Price	Address
0	79545.458574	5.682861	7.009188	4.09	23086.800503	1.059034e+06	208 Michael Ferry Apt. 674\nLaurabury, NE 3701...
1	79248.642455	6.002900	6.730821	3.09	40173.072174	1.505891e+06	188 Johnson Views Suite 079\nLake Kathleen, CA...
2	61287.067179	5.865890	8.512727	5.13	36882.159400	1.058988e+06	9127 Elizabeth Stravenue\nDanieltown, WI 06482...
3	63345.240046	7.188236	5.586729	3.26	34310.242831	1.260617e+06	USS Barnett\nFPO AP 44820
4	59982.197226	5.040555	7.839388	4.23	26354.109472	6.309435e+05	USNS Raymond\nFPO AE 09386

4. Correlation between the columns



5. Train and test split

```
[118] X = data[['Avg. Area Income', 'Avg. Area House Age', 'Avg. Area Number of Rooms',
                'Avg. Area Number of Bedrooms', 'Area Population']]
      y = data['Price']
```

Train Test Split

```
[119] from sklearn.model_selection import train_test_split

      X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)
```

6. Scaling the data

```
from sklearn.preprocessing import StandardScaler
from sklearn.pipeline import Pipeline

pipeline = Pipeline([
    ('std_scalar', StandardScaler())
])

X_train = pipeline.fit_transform(X_train)
X_test = pipeline.transform(X_test)
```

7. Fit the Linear Regression model

```
from sklearn.linear_model import LinearRegression

lin_reg = LinearRegression(normalize=True)
lin_reg.fit(X_train,y_train)
```

8. Model evaluation

```
# print the Y intercept
print(lin_reg.intercept_)
```

```
1228219.1492415662
```

```
[103] coeff_df = pd.DataFrame(lin_reg.coef_, X.columns, columns=['Coefficient'])
      coeff_df
```

	Coefficient
Avg. Area Income	232679.724643
Avg. Area House Age	163841.046593
Avg. Area Number of Rooms	121110.555478
Avg. Area Number of Bedrooms	2892.815119
Area Population	151252.342377

Interpreatation of the coefficients:

- Holding all other features fixed, a 1 unit increase in **Avg. Area Income** is associated with an increase of **$21.52.**
- Holding all other features fixed, a 1 unit increase in **Avg. Area House Age** is associated with an increase of **$164883.28.**
- Holding all other features fixed, a 1 unit increase in **Avg. Area Number** of Rooms is associated with an increase of **$122368.67.**
- Holding all other features fixed, a 1 unit increase in **Avg. Area Number of Bedrooms** is associated with an increase of **$2233.80.**
- Holding all other features fixed, a 1 unit increase in **Area Population** is associated with an increase of **$15.15.**

9. Evaluation matrix for the model built

- **Mean Absolute Error (MAE)** is the mean of the absolute value of the errors.
- **Mean Squared Error (MSE)** is the mean of the squared errors.
- **Root Mean Squared Error (RMSE)** is the square root of the mean of the squared errors. The lower the RMSE better is model performance.
- **R-squared (R2)** is a statistical measure that represents the proportion of the variance for a dependent variable that's explained by an independent variable or variables in a regression model.

MAE is the easiest to understand, because it's the average error.

MSE is more popular than MAE, because MSE "punishes" larger errors, which tends to be useful in the real world.

RMSE is even more popular than MSE, because RMSE is interpretable in the "y" units.

All of these are loss functions, because we want to minimize them.

```
test_pred = lin_reg.predict(X_test)
train_pred = lin_reg.predict(X_train)

print('Model evaluation:\n_____________________________________')
print_evaluate(y_test, test_pred)
```

```
Model evaluation:
_____________________________________
MAE: 81135.56609336878
MSE: 10068422551.40088
RMSE: 100341.52954485436
R2 Square: 0.9146818498754016
_____________________________________
```

```
results_df = pd.DataFrame(data=[["Linear Regression", *evaluate(y_test, test_pred) , cross_val(LinearRegression())]],
                          columns=['Model', 'MAE', 'MSE', 'RMSE', 'R2 Square', "Cross Validation"])
results_df
```

	Model	MAE	MSE	RMSE	R2 Square	Cross Validation
0	Linear Regression	81135.566093	1.006842e+10	100341.529545	0.914682	0.917379

Final Observations:

After completing this ICP and training linear regression model, I can conclude that I have better understanding of multiple parameters related to a linear regression model. Few terms are discussed below:

- **SSE (Sum of Square error) or MSE (Mean square error)** is measure of how far off our model's predictions are from the observed values. A value of 0 indicates that all predications are spot on. A non-zero value indicates errors. A good model will always have lower value of MSE.
- **R Square** measures how much variance is captured by the model. The range for Ordinary Least Squares is [0,1]. It is possible to get negative values for R^2 but that would require a fitting procedure other than OLS or non-linear data.
- **Y-Intercept:** In a linear relation when X becomes zero and Y is still left with some remainder value which shows the constant value of Y even if all the dependent variables are zero.

Video Link

- https://youtu.be/fiKfrM_AQNQ

Conclusion

1. Lessons Learnt

- I learnt how to train a linear regression model.
- I understood what type of dataset is best for this model.

2. Challenges Faced

- The selection of right dataset was a challenge. I was looking for mostly the numerical dataset which mostly has one of them as predictor variable. The housing dataset is good fit as we are trying to predict the house prices depending on other house related parameters in the dataset.