Data science

"The best way to learn data science is to apply data science."

MOHAMMED VI POLYTECHNIC UNIVERSITY

Activity 4 31/03/2021

Bochra CHEMAM

Simple Linear Regression :

For this Activity session we will:

Learn how to build model which will predict using simple linear regression

1. Problem statement : For this activity, we will investigate the advertising dataset that you can find on kaggle .
2. We will use the dataset and analyse the relationship between TV Advertising and sales using a simple linear regression model
3. We will be able to build a linear regression model to predict sales using an appropriate predictor variable.
4. Get the Data :
    - Once your Dataset is downloaded , create you project in your Jupyter notebook or related.
    - Start by downloading the data to your workplace. Download your dataset using Pandas library as seen in the previous activities
5. **Now, let's make the Dataframe for the given data and check its head value.**
6. **Data Structure Understand your data :**
    - Let's take a look at the top five rows using the DataFrames head()
    - The info() method is useful to get a quick description of the data, in particular the total number of rows, and each attribute's type and number of non-null values in the dataset,

- Use The describe() method shows a summary of the numerical attributes The count, mean, min, and max rows are self-explanatory

- Checking Null values .isnull()

7. Let's see the correlation between different variables. Use .corr() to calculate correlation Matrix between different variables , you can use also seaborn , **sns.heatmap**

    - Interpret the results

8. let's go ahead and perform **simple linear regression** using TV as our feature variable.

9. **Model Building :**

    - **X= independent variable**
    - **Y=dependant varaible**
    - **So here in our example we will explain y{sales} with x(TV advertising)**

- **Train-Test Split**

10. Split Data (train – test):

- You now need to split our variable into training and testing sets.
- You'll perform this by importing train_test_split from the sklearn.model_selection library.
    - keep 80% of the data in your train dataset and the rest 20% in your test dataset

11. Building a linear Model

- You first need to import the **statsmodel.api** library using which you'll perform the linear regression.
- Add a constant to get an intercept (X_train_sm = sm.add_constant(X_train))
- Fit the resgression line using 'OLS' (lr = sm.OLS(y_train, X_train_sm).fit())

12. Lets Print the parameters Using (lr.params)

13. Performing a summary of all the different parameters of the regression line fitted Using print(lr.summary())

14. Let's visualize how well the model fit the data. Using scatter between x_train and y_train

15. let us plot the histogram of the error terms and see what it looks like.

    y_train_pred = lr.predict(X_train_sm)

    res = (y_train - y_train_pred)

    - What is the distribution of the residuals ? what can you conclude ?

16. **Predictions on the Test Set:**

    - Now that you have fitted a regression line on your train dataset, it's time to make some predictions on the test data.

- For this, you first need to add a constant to the **X_test data** like you did for **X_train** and then you can simply go on and predict the **y** values corresponding to **X_test** using the predict attribute of the fitted regression line.

```
X_test_sm = sm.add_constant(X_test)
y_pred = lr.predict(X_test_sm)
```

17. Lets evaluate the prediction on test set :

```
from sklearn.metrics import mean_squared_error

from sklearn.metrics import r2_score
```

- **Looking at the RMSE and R squared , Interpret !**