

MACHINE LEARNING ALGORITHMS

DAY-03

SIMPLE LINEAR REGRESSION:

STEPS:

To perform simple linear regression practically first we need to load the data into jupyter notebook then perform data analysis followed by data validation and build the machine learning model.

1. Data Analysis
2. Data Validation
3. Model Building
 - I. Build a Model
 - II. Train Model
 - III. Test the model
 - IV. Evaluate the model

1. Data Analysis:

- Load the data using pandas library into the jupyter notebook
Code:
`npd=pd.read_csv(r"C:\Users\INDUPRIYA\OneDrive\Attachments\Desktop\ML Models\NewspaperData.csv")`
- Check if the data has any unwanted columns, here my newspaper dataset has newspaper column it is categorical and contains all unique values so that is not necessary for the model building therefore drop the column.
Code : `del npd ["Newspaper"]`
- In the data daily will become feature and Sunday becomes the target.
- Perform the initial analysis on the dataset like checking the size, datatypes, null values.
`npd.shape`
`npd.dtypes`

```
npd.isna().sum()
```

```
npd.info()
```

```
npd.describe()
```

- Check if the two columns are colinear or not. For that we can simply plot a scatter plot and interpret from the visual. Matplotlib and seaborn libraries has to be used for visualization.

Code:

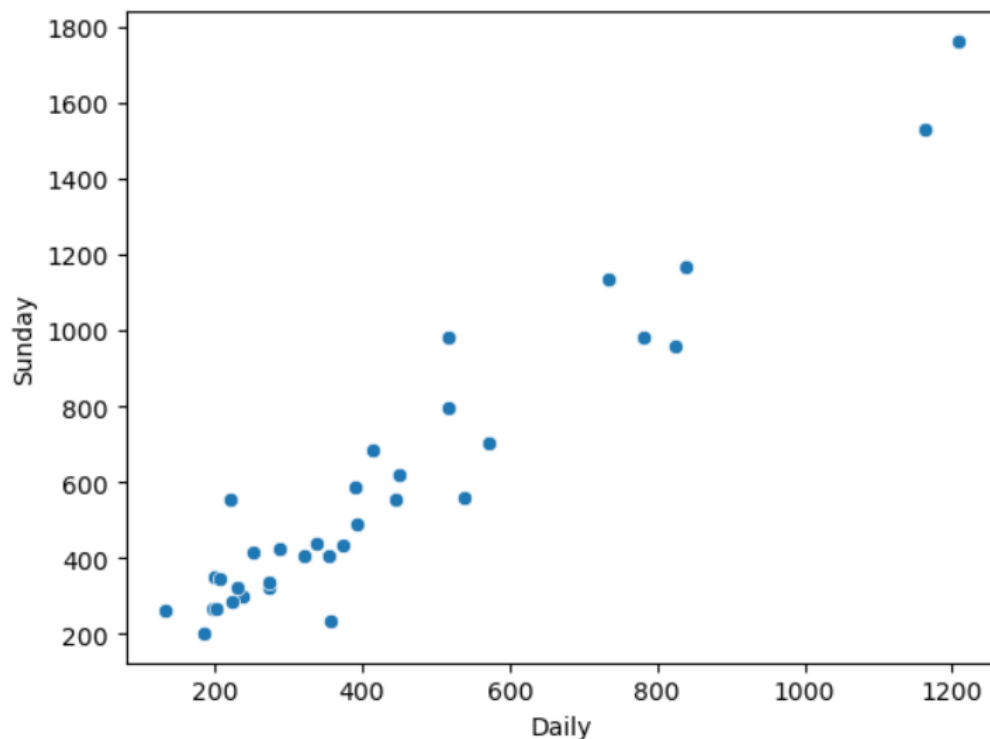
```
sns.scatterplot(x=npd["daily"],y=npd["sunday"])
```

```
plt.xlabel("Daily")
```

```
plt.ylabel("Sunday")
```

```
plt.show()
```

Result:



- From the visual we can conclude that the columns are having 100% positive collinearity.
2. Data Validation (will be covered in coming days)
 3. Model Building:
 - Before proceeding to model building store the feature and target to the variables for better accessibility.

```
x = npd[['daily']]
```

```
y = npd['sunday']
```

- Make sure you are storing the feature in dataframe.
 - Now import sklearn library to build the model.
- ```
from sklearn.linear_model import LinearRegression
```
- I. Model building: `slr = LinearRegression()`
  - II. Train the model: `slr.fit(x,y)`
  - III. Test the model: `y_pred = slr.predict(x).round(2)`
- Here the model generates the prediction line based on alpha and beta values
  - To see the values just type the code as below

```
slr.coef_
```

```
slr.intercept_
```

```
prediction line equation: $y_{pred} = 1.3397x + 13.8356$
```

IV. Model Evaluation:

```
from sklearn.metrics import r2_score
```

```
r2_score(y,y_pred)
```

```
output:
```

```
0.9180594095631381
```