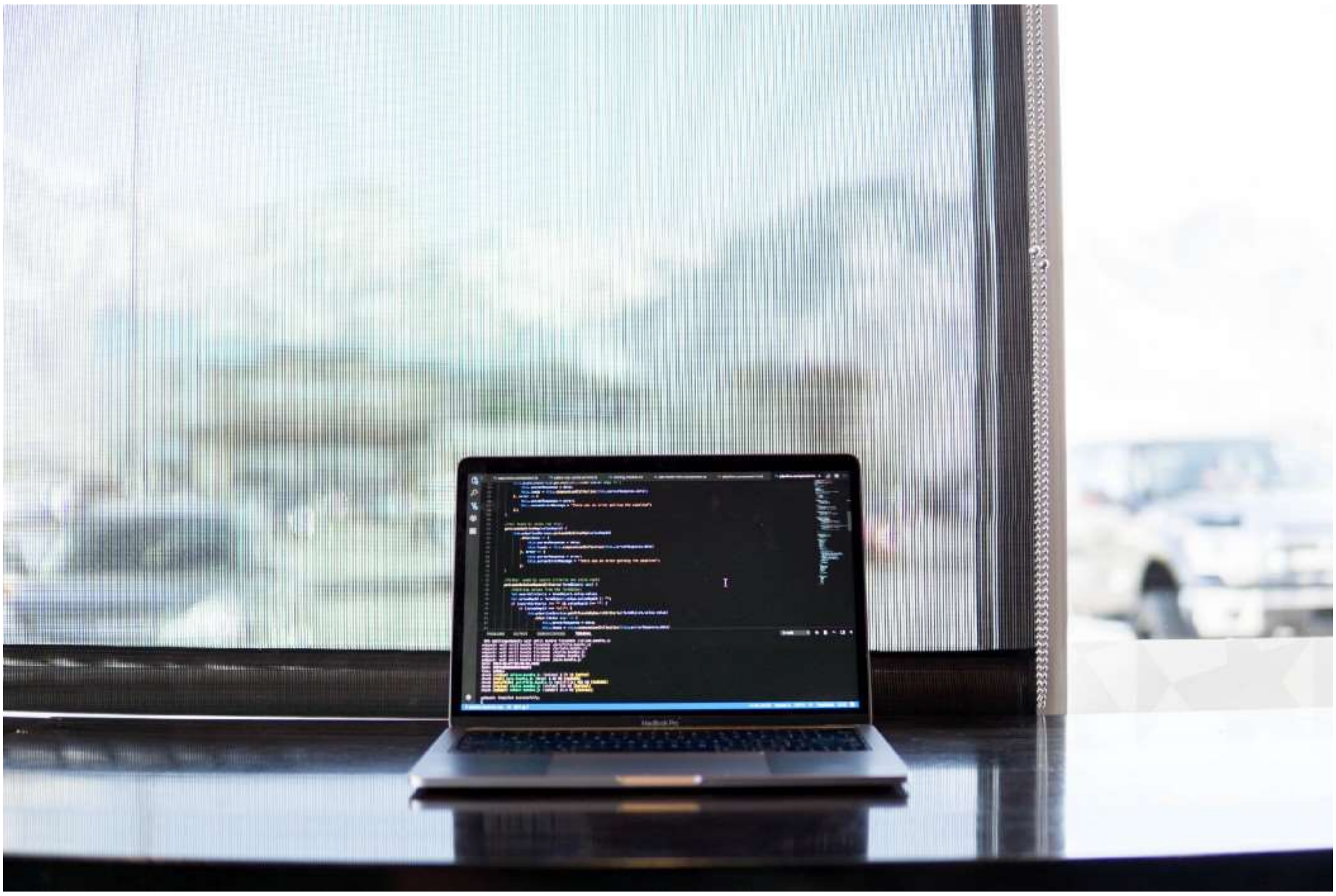
Published in Analytics Vidhya

Prabhat Pathak Follow
Jul 16, 2020 · 6 min read ★ · Listen

Implementing Linear Regression with Categorical variable Using Sklearn

Easy Steps for implementing Linear regression from Scratch



vidhya/a-beginners-guide-to-linear-regression-in-python-with-scikit-learn-6b0fe70b32d7

In this article, we will talk about

1. Imports
2. Dataset
3. Training and test dataset
4. predictions
5. Conclusion

Imports

We will be using pandas, numpy sklearn, seaborn and matplotlib

I've also imported ***warnings*** module so the Notebook remains clean:

```
import warnings
warnings.filterwarnings('ignore')

import pandas as pd
import numpy as np

import seaborn as sns
import matplotlib.pyplot as plt
```

now I am importing the dataset

```
df=df = pd.read_excel('Multiple_variable.xlsx',sheet_name='Sheet2')
df.head()
```

	Age	YearsExperience	Salary	Gender	Classification	Job
0	22	1.1	39343	Female	Low	Assistant
1	22	1.3	46205	Male	TOP	Professor
2	23	1.5	37731	Female	TOP	Administrative
3	24	2.0	43525	Female	Medium	Assistant
4	25	2.2	39891	Male	Medium	Professor

dataset

We have Age, Yearsof Experience, Salary, Gender, Classification and Job.

Data Pre-Processing

```
df.shape
df.describe()
```

	Age	YearsExperience	Salary
count	36.000000	36.000000	36.000000
mean	34.472222	6.008333	82228.277778
std	6.942565	3.031489	28784.838078
min	22.000000	1.100000	37731.000000
25%	29.000000	3.575000	57050.000000
50%	37.000000	5.600000	82225.500000
75%	40.250000	9.000000	110232.000000
max	49.000000	10.500000	122391.000000

From the above output, I will try to see the nature of the dataset, when I am saying nature that means is dataset is following normal distribution or not? or is the dataset is

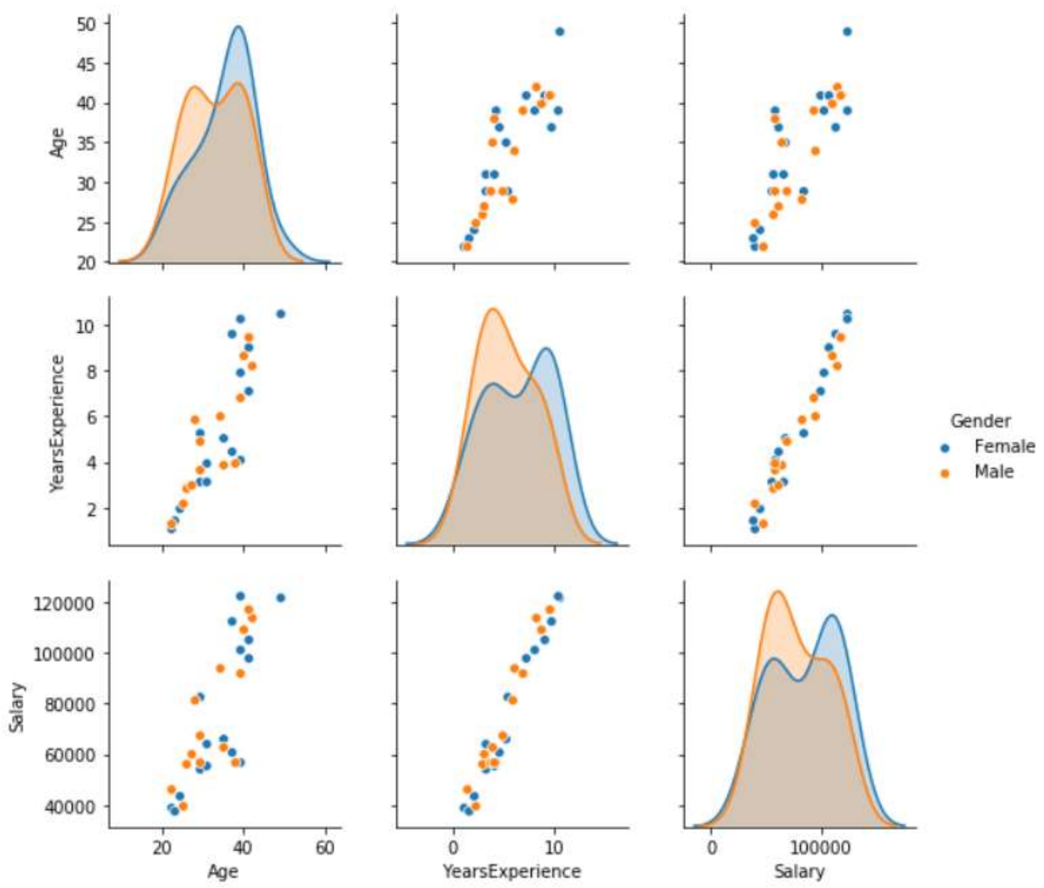
```
df.dtypes
```

```
Age                  int64
YearsExperience    float64
Salary               int64
Gender              object
Classification      object
Job                 object
dtype: object
```

EDA

```
sns.pairplot(df,hue='Gender')
```

```
<seaborn.axisgrid.PairGrid at 0x18972b35b48>
```



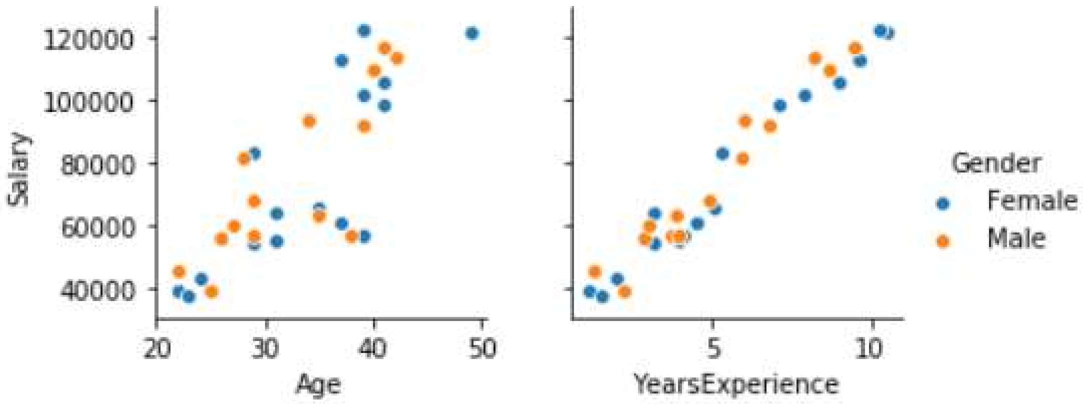
Pair plot

Open in app Get started

for more details, let's look at the below graph :

```
sns.pairplot(df,x_vars=['Age','YearsExperience'],y_vars=
['Salary'],hue='Gender')
```

<seaborn.axisgrid.PairGrid at 0x189742f1348>




```
df.corr()
```

	Age	YearsExperience	Salary
Age	1.000000	0.858866	0.825977
YearsExperience	0.858866	1.000000	0.982536
Salary	0.825977	0.982536	1.000000

Correlation and regression analysis are related in the sense that both deal with relationships among variables. The correlation coefficient is a measure of linear association between two variables. The values of the correlation coefficient are always between -1 and +1. A correlation coefficient of +1 indicates that two variables are perfectly related in a positive linear sense, a correlation coefficient of -1 indicates that two variables are perfectly related in a negative linear sense, and a correlation

```
sns.heatmap(df.corr(),annot=True,lw=1)
```

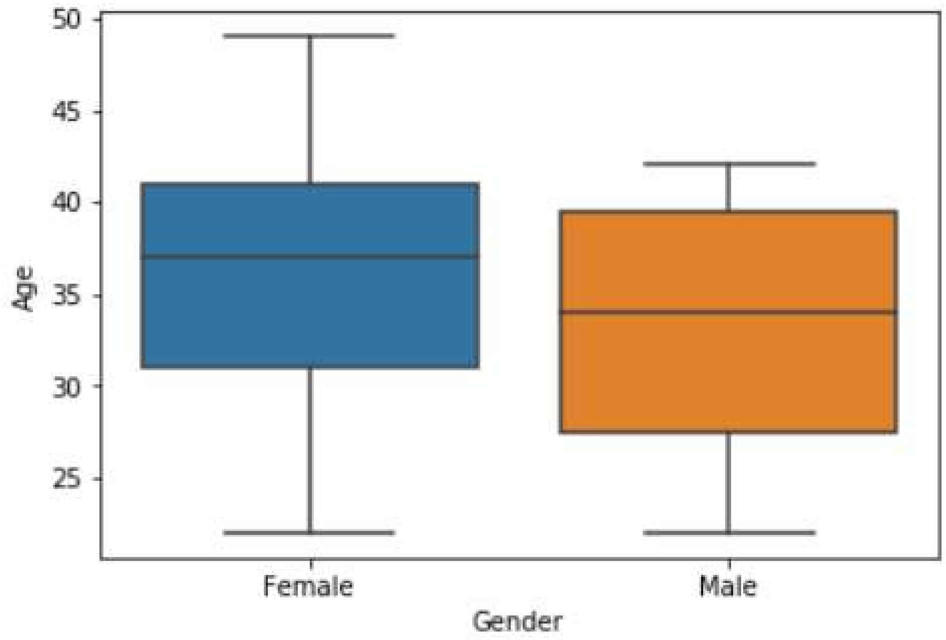
```
<matplotlib.axes._subplots.AxesSubplot at 0x189747a5688>
```

heatmap

If you want to see the trend between two variables you can see with the box plot.

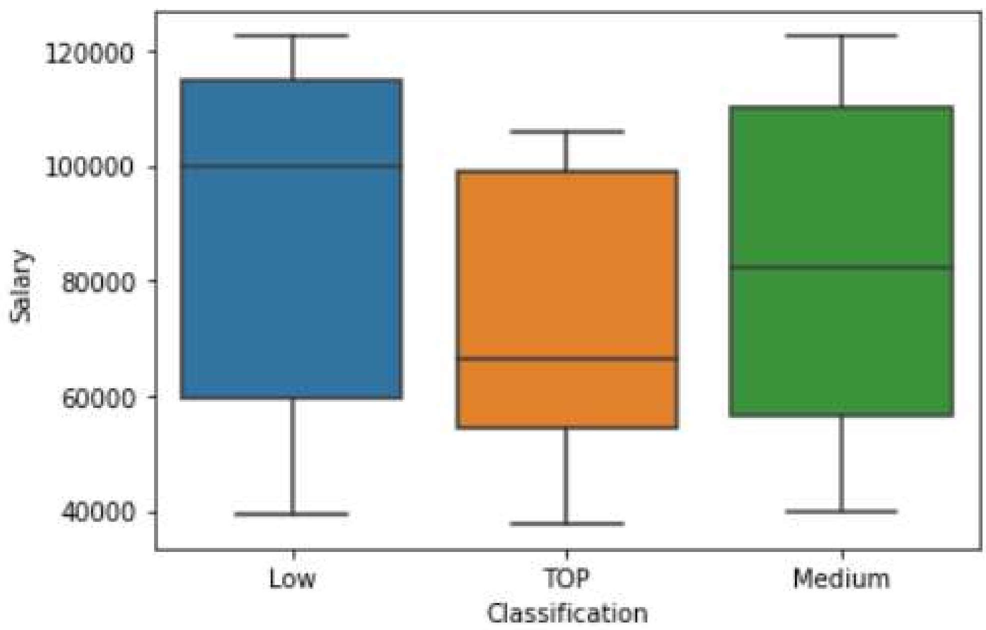
```
sns.boxplot(y='Age',x='Gender',data=df)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x18976f35d88>
```




```
sns.boxplot(y='Salary',x='Classification',data=df)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x189772104c8>
```



After doing all analysis we understand we have some categorical variables as well, so

A dummy variable (is, an indicator variable) is a numeric variable that represents categorical data, such as gender, race, etc.

What are the benefits of a Dummy Variable?

Regression results are easiest to interpret when dummy variables are limited to two specific values, 1 or 0. Typically, 1 represents the presence of a qualitative attribute, and 0 represents the absence.

so our independent variables would be

```
X = df[['Age', 'YearsExperience', 'Gender', 'Classification',
'Job']]
```

In the above data frame, we have Gender, Classification, and Job as a categorical variable, so we need to add dummy variables instead.

```
X = pd.get_dummies(data=X, drop_first=True)
X.head()
```

	Age	YearsExperience	Gender_Male	Classification_Medium	Classification_TOP	Job_Assistant	Job_Manager	Job_Professor	Job_Senior Manager
0	22	1.1	0	0	0	1	0	0	0
1	22	1.3	1	0	1	0	0	1	0
2	23	1.5	0	0	1	0	0	0	0
3	24	2.0	0	1	0	1	0	0	0
4	25	2.2	1	1	0	0	0	1	0

above code added the dummy variable in form of 0 and 1, which is easy to interpret for the regression model.

```
Y = df['Salary']
```

```
0      39343
1      46205
2      37731
3      43525
4      39891
5      56642
6      60150
7      54445
8      64445
9      57189
```

Creating a train and test dataset.


```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, Y,
test_size=0.4, random_state=101)

print(X_train.shape)
print(X_test.shape)
print(y_train.shape)
print(y_test.shape)
```

```
(21, 9)
(15, 9)
(21,)
(15,)
```

After splitting the dataset into a test and train we will be importing the Linear Regression model.

```
from sklearn.linear_model import LinearRegression

model = LinearRegression()

model.fit(X_train,y_train)
```

```
LinearRegression(copy_X=True, fit_intercept=True, n_jobs=None, normalize=False)
```

```
# print the intercept
print(model.intercept_)
```

```
40767.075290708104
```

The intercept (often labeled the constant) is the expected mean value of Y when all X=0. In a purely mathematical sense, this definition is correct. Unfortunately, it's frequently impossible to set all variables to zero because this combination can be an impossible or irrational arrangement.

```
coeff_parameter = pd.DataFrame(model.coef_,X.columns,columns=
['Coefficient'])
coeff_parameter
```

	Coefficient
Age	-548.323776
YearsExperience	10743.731522
Gender_Male	-655.537127
Classification_Medium	-6061.914786
Classification_TOP	-1234.672994
Job_Assistant	-1114.042048
Job_Manager	2291.025846
Job_Professor	964.080429
Job_Senior Manager	-2141.064227

The sign of each coefficient indicates the direction of the relationship between a predictor variable and the response variable.

- A positive sign indicates that as the predictor variable increases, the Target variable

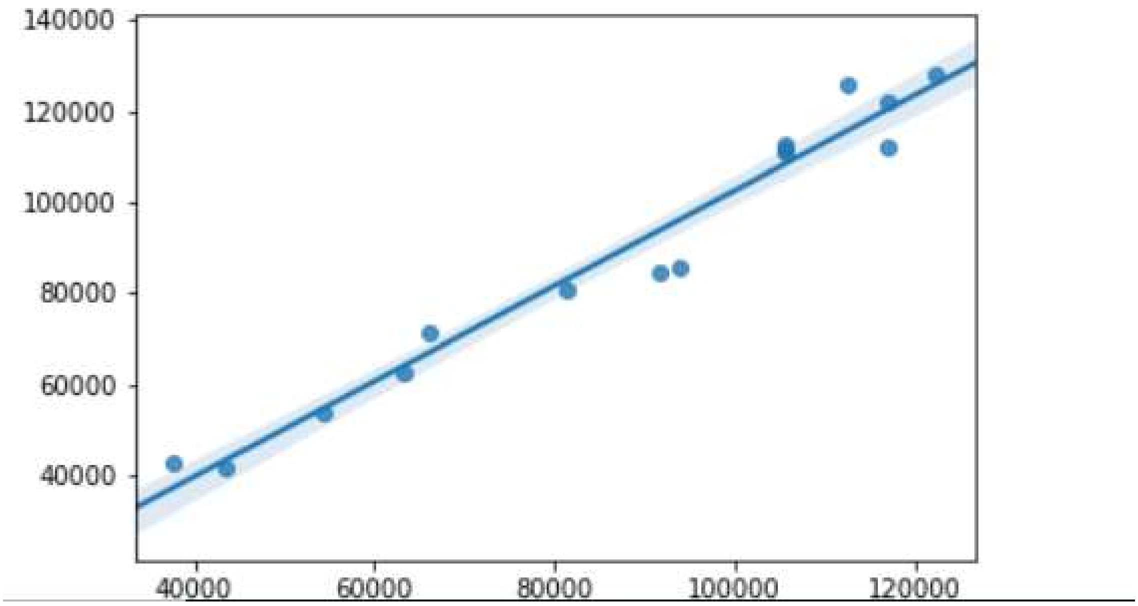
```
predictions = model.predict(X_test)
predictions
```

```
array([ 80970.53158392,  54147.79230197,  84608.3284182 , 112838.31994826,
       111603.646954  , 121986.73865586, 111208.49523541, 125909.94403861,
       112148.27092338,  43036.55273237, 127901.81847906,  85930.91891418,
        71270.93953831,  62820.7589416 ,  41918.81087788])
```

Yaay, here is your predicted variable.

```
sns.regplot(y_test,predictions)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x18978f9e848>
```



The above graph shows our model is predicting good results. lets see Rsquare value

```
import statsmodels.api as sm
X_train_Sm= sm.add_constant(X_train)

X_train_Sm= sm.add_constant(X_train)
ls=sm.OLS(y_train,X_train_Sm).fit()
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                 Salary   R-squared:                       0.971
Model:                            OLS   Adj. R-squared:                  0.952
Method:                 Least Squares   F-statistic:                     51.09
Date:                Wed, 15 Jul 2020   Prob (F-statistic):           4.20e-08
Time:                        00:54:58   Log-Likelihood:                -207.81
No. Observations:                  21   AIC:                             433.6
Df Residuals:                      12   BIC:                             443.0
Df Model:                           8
Covariance Type:            nonrobust
=========================================================================================
                            coef    std err          t      P>|t|      [0.025      0.975]
-----------------------------------------------------------------------------------------
const                  3.261e+04   9431.499      3.458      0.005    1.21e+04    5.32e+04
Age                    -548.3238    427.836     -1.282      0.224   -1480.499     383.852
YearsExperience        1.074e+04   1744.592      6.158      0.000    6942.592    1.45e+04
Gender_Male            -655.5371   3091.666     -0.212      0.836   -7391.698    6080.624
Classification_Medium -6061.9148   4374.305     -1.386      0.191   -1.56e+04    3468.878
Classification_TOP    -1234.6730   4574.620     -0.270      0.792   -1.12e+04    8732.568
Job_Assistant          7039.3730   3997.588      1.761      0.104   -1670.623    1.57e+04
Job_Manager            1.044e+04   5193.314      2.011      0.067    -870.818    2.18e+04
Job_Professor          9117.4955   4842.954      1.883      0.084   -1434.395    1.97e+04
Job_Senior Manager     6012.3508   6454.697      0.931      0.370   -8051.225    2.01e+04
==============================================================================
Omnibus:                        9.775   Durbin-Watson:                   2.131
Prob(Omnibus):                  0.008   Jarque-Bera (JB):                2.092
Skew:                           0.115   Prob(JB):                        0.351
Kurtosis:                       1.471   Cond. No.                     3.89e+17
==============================================================================
```

What Are the Adjusted R-squared?

We Use adjusted R-squared to compare the goodness-of-fit for regression models that contain different numbers of independent variables.

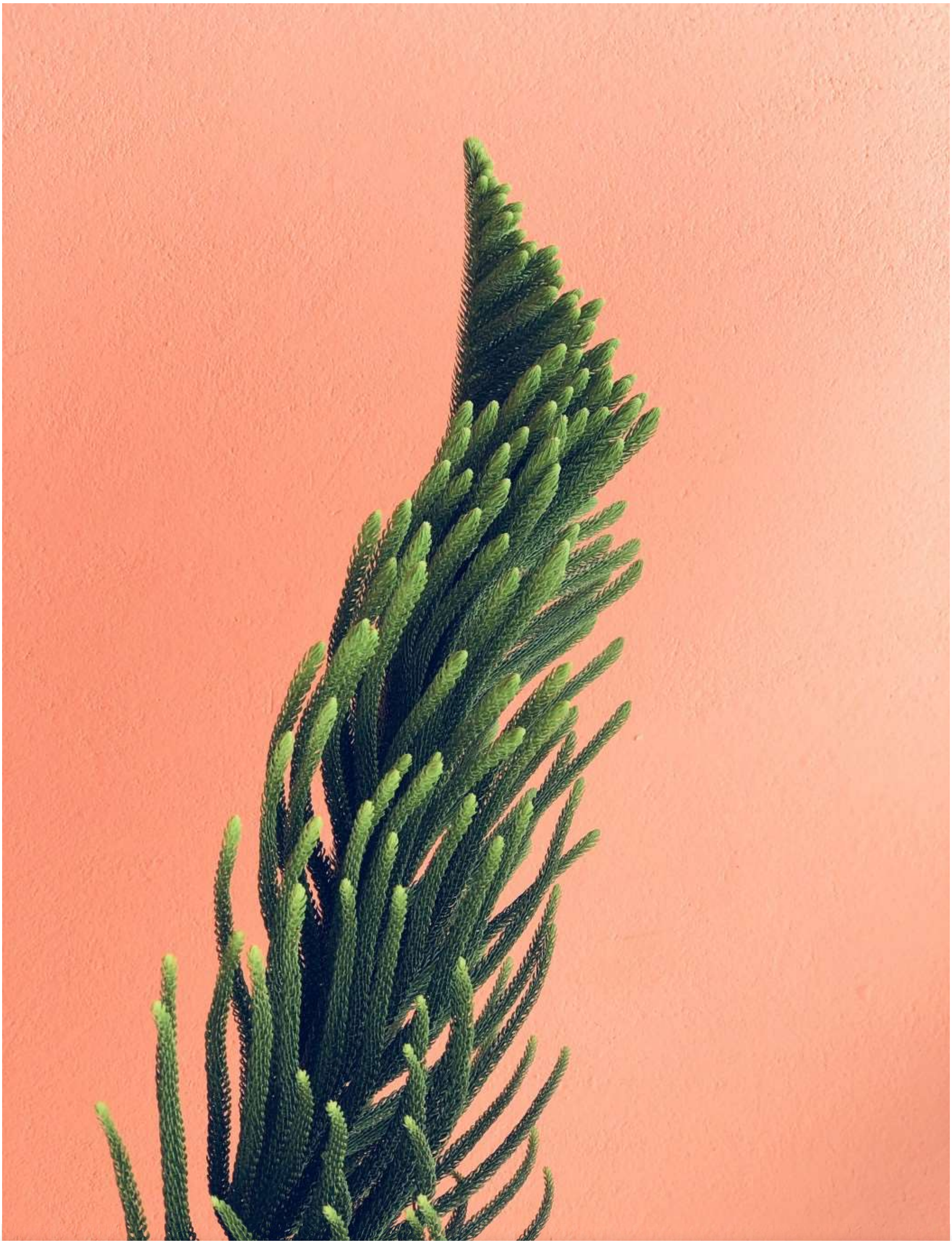
Let’s say you are comparing a model with five independent variables to a model with one variable and the five variable model has a higher R-squared. Is the model with five variables actually a better model, or does it just have more variables? To determine this, just compare the adjusted R-squared values!

I tried to touch base most of the concepts, you can try out to calculate some metrics like MSE and RMSE.

Conclusion :

As I said before Linear regression is really amazing algorithms and there are lots of things you can do with this. try out new things and let me know if you any suggestions.

Prabhat Pathak *(Linkedin profile) is an Associate Analyst.*



Sign up for Analytics Vidhya News Bytes

By Analytics Vidhya

Latest news from Analytics Vidhya on our Hackathons and some of our best articles! Take a look.

Your email

Get this newsletter