LinearRegression

May 16, 2025

0.1 Importing Necessary Libraries

[6]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import sklearn
from sklearn.linear_model import LinearRegression
```

0.2 Creating Custom Dataset

[8]:

```
np.array([1, 2, 3, 4, 5, 6, 7, 8, 9, 10])
```

[8]:

```
array([ 1,  2,  3,  4,  5,  6,  7,  8,  9, 10])
```

[ ]:

```
X = np.array([1, 2, 3, 4, 5, 6, 7, 8, 9, 10]).reshape(-1, 1)
y = np.array([2.3, 4.5, 6.1, 8.9, 11.0, 12.9, 15.2, 18.5, 20.0, 21.9])

print("X: \n", X)
print("Y: \n", y)
```

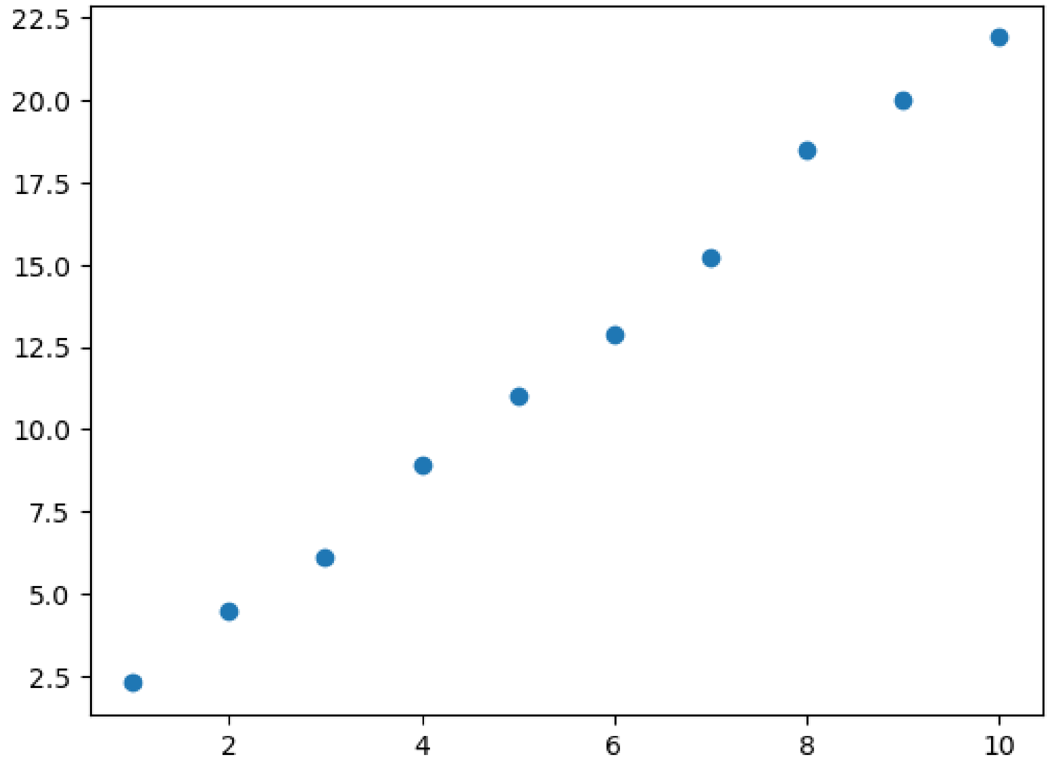
```
X:
 [[ 1]
 [ 2]
 [ 3]
 [ 4]
 [ 5]
 [ 6]
 [ 7]
 [ 8]
 [ 9]
 [10]]
Y:
 [ 2.3  4.5  6.1  8.9 11.  12.9 15.2 18.5 20.  21.9]
```

0.3 Visualizing the created Data Points

[14]:

```
plt.scatter(X,y)
```

[14]:
```
<matplotlib.collections.PathCollection at 0x7acd28975690>
```



0.4 Using Machine Learning Approach (Scikit Learn Library) for Linear Regression

[12]:
```
# Defining the Model
model = LinearRegression()
# Fitting the Model
model.fit(X, y)
```

[12]:
```
LinearRegression()
```

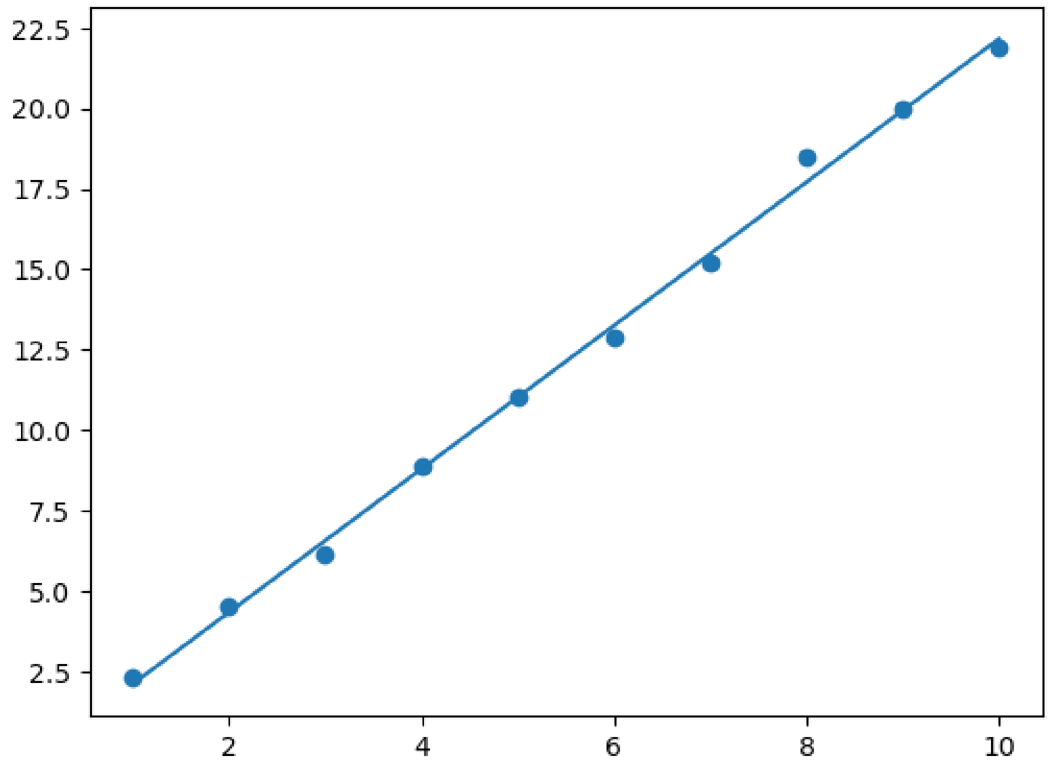
[13]:
```
# Generating Model Predictions
y_pred = model.predict(X)
y_pred
```

[13]:
```
array([ 2.10181818,  4.33030303,  6.55878788,  8.78727273, 11.01575758,
       13.24424242, 15.47272727, 17.70121212, 19.92969697, 22.15818182])
```

0.4.1 Visualize the Linear Regression Line

[14]:
```
plt.scatter(X,y)
plt.plot(X,y_pred)
```

[14]:
```
[<matplotlib.lines.Line2D at 0x7fedf50c9b10>]
```



[16]:
```
model.intercept_, model.coef_
```

[16]:
```
(np.float64(-0.12666666666666693), array([2.22848485]))
```

0.5 Using Statsmodels Library for Regression (This is preferred in this course)

[18]:
```
import statsmodels.api as sm
X_const = sm.add_constant(X)
model = sm.OLS(y, X_const).fit()
print(model.summary())
```

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                      y   R-squared:                       0.997
Model:                            OLS   Adj. R-squared:                  0.997
Method:                 Least Squares   F-statistic:                     2745.
```

```
Date:                Wed, 14 May 2025   Prob (F-statistic):          1.95e-11
Time:                        08:00:03   Log-Likelihood:               -3.5633
No. Observations:                  10   AIC:                            11.13
Df Residuals:                       8   BIC:                            11.73
Df Model:                           1
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const         -0.1267      0.264     -0.480      0.644      -0.735       0.482
x1             2.2285      0.043     52.392      0.000       2.130       2.327
==============================================================================
Omnibus:                        3.667   Durbin-Watson:                   2.210
Prob(Omnibus):                  0.160   Jarque-Bera (JB):                1.246
Skew:                           0.846   Prob(JB):                        0.536
Kurtosis:                       3.353   Cond. No.                         13.7
==============================================================================

Notes:
[1] Standard Errors assume that the covariance matrix of the errors is correctly
specified.
```

```
/home/fm-pc-lt-276/micromamba/envs/otg-fraud/lib/python3.11/site-
packages/scipy/stats/_axis_nan_policy.py:430: UserWarning: `kurtosistest`
p-value may be inaccurate with fewer than 20 observations; only n=10
observations were given.
  return hypotest_fun_in(*args, **kwds)
```

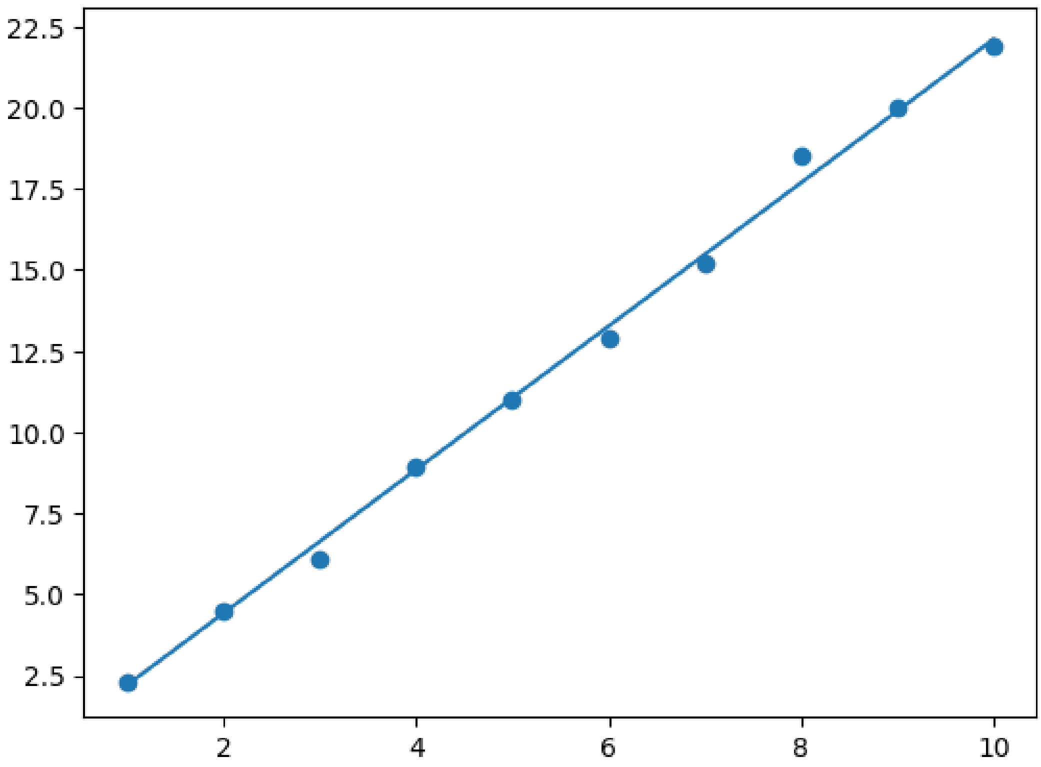
0.5.1 Generating predictions and Visualizing

[7]:

```
y_pred = model.predict(X)
plt.scatter(X,y)
plt.plot(X,y_pred)
```

[7]:

```
[<matplotlib.lines.Line2D at 0x7dbc09526790>]
```


22.5
20.0
17.5
15.0
12.5
10.0
7.5
5.0
2.5
2
4
6
8
10