Regression

- The term regression is used, when trying to find the relationship between variables.
- In ML, and in statistical modeling, that relationship is used to predict the outcome of future events.

Linear Regression

- LR uses the relationship between the data-points to draw a straight line through all them.
- This line can be used to predict future values.

$$y = m * x + b$$

$$m = \frac{n\sum xy - (\sum x)(\sum y)}{n(\sum x^2) - (\sum x)^2}$$

$$b = \frac{\sum y - m\sum x}{n}$$

$$sse = \sum(y_{org} - y_{pred})^2$$

In [125…

```
class LinearRegression:

    def __init__(self, df):
        self.df = df
        self.df_x_sum = sum(self.df['X'])
        self.df_y_sum = sum(self.df['Y'])
        self.n = len(df['X'])

    def m_value(self):
        m_nume = (self.n*(sum(self.df['X']*self.df['Y']))) - (self.df_x_sum*self.df_y_sum)
        m_deno = (self.n*(sum(self.df['X']**2))) - ((sum(self.df['X']))**2)
        return m_nume/m_deno

    def b_value(self):
        b_nume = (sum(self.df['Y'])) - (self.m_value()*(sum(self.df['X'])))
        return b_nume/self.n

    def y_pred(self):
        self.y_pred = [((self.m_value() * i) + self.b_value()) for i in (self.df['X'])]
        return self.y_pred

    def sse_value(self):
        sse_perf = [((y_org - y_pred)**2) for y_org , y_pred in zip(self.df['Y'], self.y_pred)]
        return (sum(sse_perf))

    def future_pred(self):
        for i in range(8,15):
            future_pred = (self.m_value() * i) + self.b_value()
        return future_pred
```

In [126…

```
df = {"X" : [i for i in range(1, 8)],
      "Y" : [1.5, 3.8, 6.7, 9.0, 11.2, 13.6, 16.0]}
import pandas as pd

df = pd.DataFrame(df)

lin_reg = LinearRegression(df)
```

In [127…

```
lin_reg.m_value()
```

Out[127…

```
2.4142857142857133
```

In [128…

```
lin_reg.b_value()
```

Out[128…

```
-0.8285714285714231
```

In [136…

```
df['Y_pred']= lin_reg.y_pred
df['Y_pred']
```

Out[136…

```
0     1.585714
1     4.000000
2     6.414286
3     8.828571
4    11.242857
5    13.657143
6    16.071429
Name: Y_pred, dtype: float64
```

In [130…

```
lin_reg.sse_value()
```

Out[130…

```
0.16857142857142873
```

In [131…

```
lin_reg.future_pred()
```

Out[131…

```
32.97142857142856
```

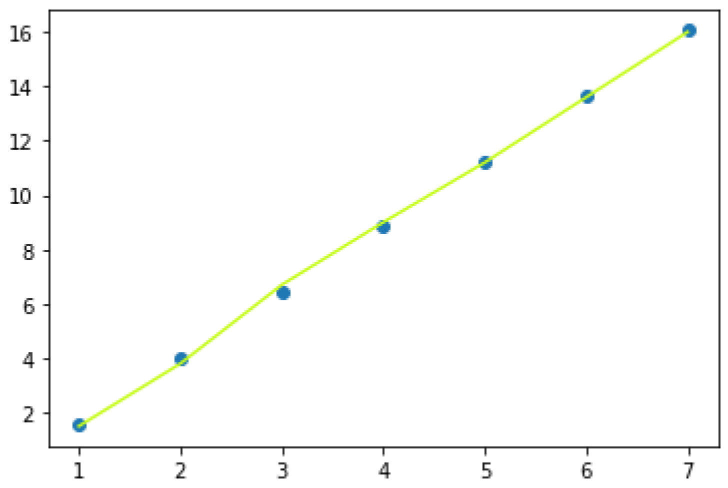
In [141…

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

plt.plot(df['X'], df['Y'], c='#c6ff1a')
#sns.lineplot(x=df['X'], y=df['Y'])
plt.scatter(df['X'], df['Y_pred'])
```

Out[141…

```
<matplotlib.collections.PathCollection at 0x1cc7a69ca30>
```



In [26]:

```
#without using pandas

df = {"X" : [i for i in range(1, 8)],
      "Y" : [1.5, 3.8, 6.7, 9.0, 11.2, 13.6, 16.0]}
d = [(i*j) for i,j in zip(df['X'], df['Y'])]
```

In [36]:

```
d
```

Out[36]:

```
[1.5, 7.6, 20.1, 36.0, 56.0, 81.6, 112.0]
```