Regresion_lineal

November 2, 2024

1 Codigo para el ejercicio de ventas ejemplo en clase

[1]:
```
import numpy as np
import pandas as pd
from sklearn import linear_model
```

[2]:
```
# Variable independiente
X_train = np.array([[1],[2],[3],[4],[5],[6]])
```

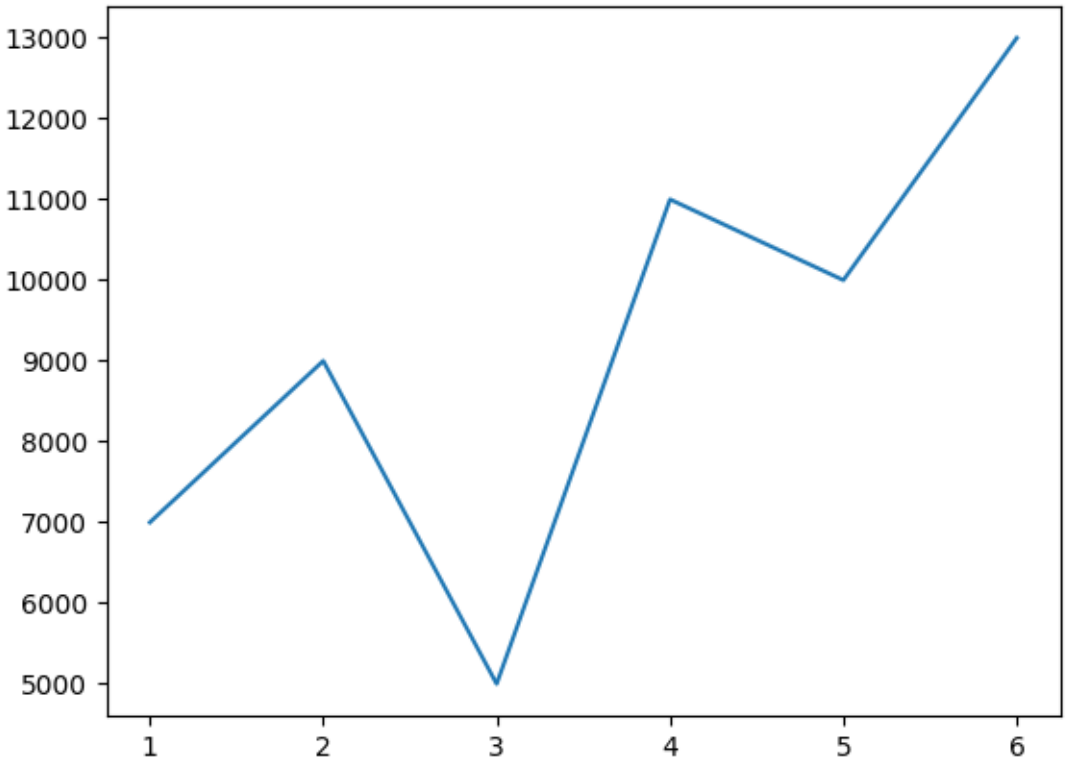
[3]:
```
# Variable dependiente
y_train = np.array([[7000],[9000],[5000],[11000],[10000],[13000]])
```

1.1 Graficando

[4]:
```
import matplotlib.pyplot as plt
```

[5]:
```
plt.plot(X_train, y_train)
```

[5]:
```
[<matplotlib.lines.Line2D at 0x1774be8b0>]
```



1.2 Crear modelo regresion lineal

[6]:
```
# Crear el modelo vacio
regre_ventas = linear_model.LinearRegression()
```

[8]:
```
regre_ventas.fit(X_train, y_train)
```

[8]:
```
LinearRegression()
```

[9]:
```
#Para revisar la pendiente
print(regre_ventas.coef_)
```

```
[[1114.28571429]]
```

[10]:
```
#Para revisar la intersecciòn
print(regre_ventas.intercept_)
```

```
[5266.66666667]
```

[11]:
```
# Vamos ahora a predecir con la funciòn predict
regre_ventas.predict([[7]])
```

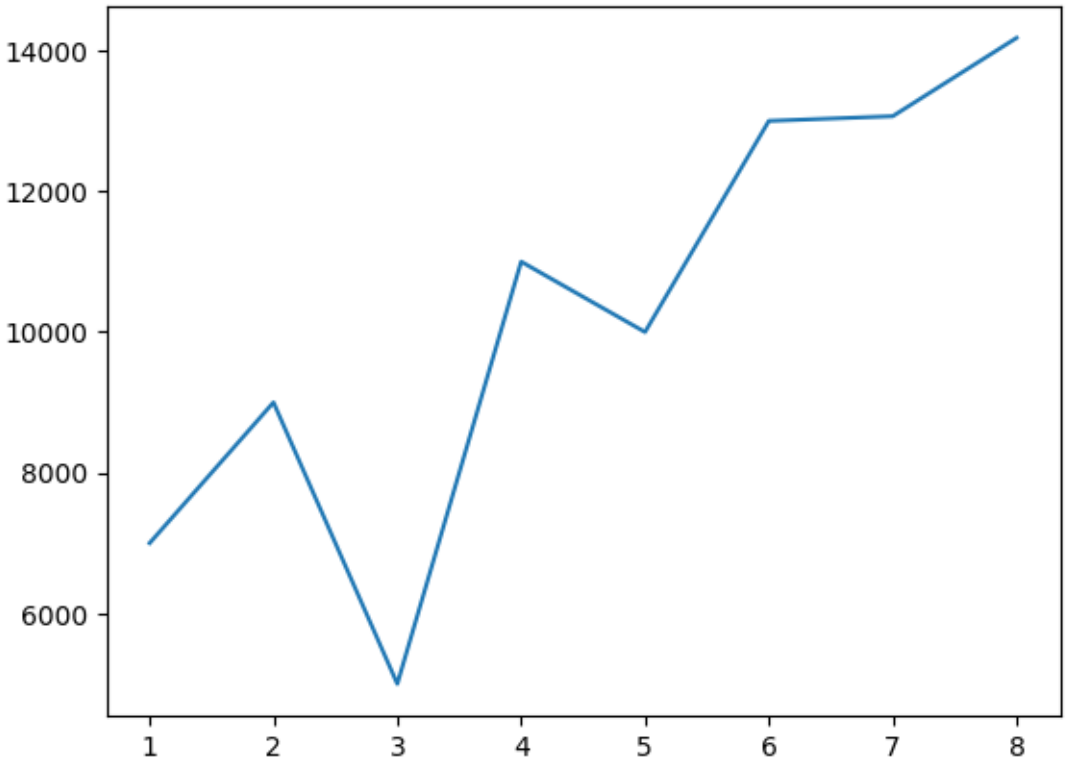
[11]:
```
array([[13066.66666667]])
```

[12]:
```
regre_ventas.predict([[8]])
```

[12]:
```
array([[14180.95238095]])
```

[13]:
```
X = np.array([[1],[2],[3],[4],[5],[6],[7],[8]])
y = np.array([[7000],[9000],[5000],[11000],[10000],[13000],[13067],[14180]])
plt.plot(X,y)
```

[13]:
```
[<matplotlib.lines.Line2D at 0x1775d0400>]
```



[ ]:
```

```