crescer-ai

January 31, 2024

[7]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

[3]:
```
df=pd.read_csv('scr-dataset.csv')
```

[4]:
```
df.head()
```

[4]:

	x	y
0	1.0	1.750768
1	1.1	1.699704
2	1.2	1.607502
3	1.3	1.479060
4	1.4	1.320438

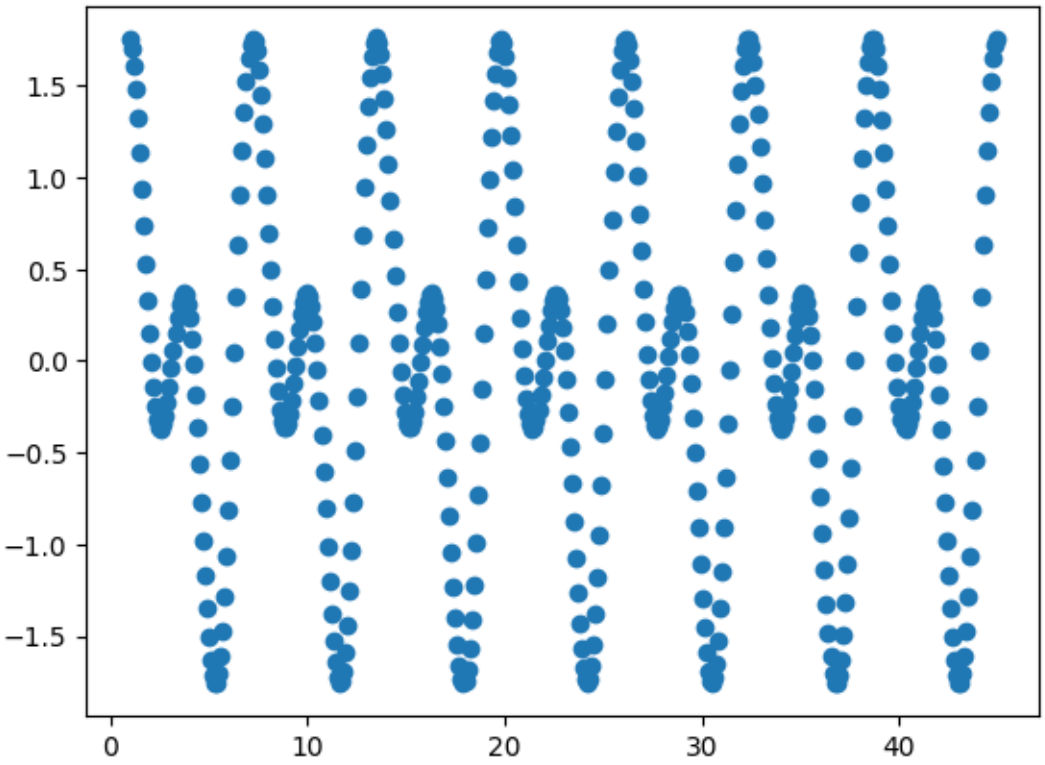
Checking Null Values

[6]:
```
df.isna().sum()
```

[6]:
```
x    0
y    0
dtype: int64
```

[19]:
```
x=df[['x']]
y=df['y']
```

[20]:
```
plt.scatter(x,y,label='Given Data')
plt.show()
```



As It seems the relation between the x and y are not linear we can fit the line using polynomial Regresion

Splitting The Dataset into Train and Test Data

[21]:
```
from sklearn.model_selection import train_test_split

xtrain,xtest,ytrain,ytest=train_test_split(x,y,test_size=0.3,random_state=0)
```

Fitting a Model

[22]:
```
from sklearn.linear_model import LinearRegression
from sklearn.preprocessing import PolynomialFeatures
from sklearn.pipeline import make_pipeline
```

[23]:
```
model = make_pipeline(PolynomialFeatures(2), LinearRegression())
model.fit(xtrain, ytrain)
```

[23]:
```
Pipeline(steps=[('polynomialfeatures', PolynomialFeatures()),
                ('linearregression', LinearRegression())])
```

THe Degree for the polynomial is kept 2 because it helps the model fitting the data more closely and even keeping more degree can cause overfitting

Testing

[27]:
```
pred=model.predict([[50]])
```

```
C:\Users\Harshal\AppData\Local\Packages\PythonSoftwareFoundation.Python.3.11_qbz
5n2kfra8p0\LocalCache\local-packages\Python311\site-
packages\sklearn\base.py:464: UserWarning: X does not have valid feature names,
but PolynomialFeatures was fitted with feature names
  warnings.warn(
```

[30]:
```
print("The Value OF y For x=50 is : ",pred[0])
```

```
The Value OF y For x=50 is :  -0.27107355309693804
```

The Value OF y For x=50 is : -0.27107355309693804

[ ]:
```

```