UPES
UNIVERSITY WITH A PURPOSE

Pattern & Anomaly Detection Lab

Experiment 5

Polynomial Regression

Submitted By:

Dhruv Singhal

500075346

R177219074

AIML B3

Submitted To:

Dr. Gopal Phartiyal

Asst. Professor

SOCS

UPES

CODE:

```
import numpy as np
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.preprocessing import PolynomialFeatures
from sklearn.datasets import make_regression

#%%
# generate  dataset
X,y = make_regression(n_samples=100, n_features=5,noise=50)

#%%
#plot the graph
plt.scatter(X[:,2],y)

#%%
#Train test Split
X_train, X_test, y_train, y_test = train_test_split(X,y,test_size=0.3) #30%test and 70% train dataset.

#%%
#implement Polynomial Regression
poly = PolynomialFeatures(degree = 3)  # degree of polynomial=3
X_poly = poly.fit_transform(X_train)

poly.fit(X_poly, y_train)
model = LinearRegression()
model.fit(X_poly, y_train)
X_poly_test=poly.fit_transform(X_test)

#%%
#predicting through polynomial Regression(Linear Regression Model)
y_pred_=model.predict(X_poly_test)

#%%
#print rmse
from sklearn.metrics import mean_squared_error
rms = mean_squared_error(y_test, y_pred_)
print("RMS Score:")
print(rms)
#%%

#print log error

from sklearn.metrics import mean_squared_log_error
logerr=mean_squared_log_error(abs(y_test), abs(y_pred_))
print("RMS Log error")
print(logerr)


#%%
#print R^2 Coffecient of determination

from sklearn.metrics import r2_score
r2_sc=r2_score(y_test,y_pred_)
print("R^2")
print(r2_sc)
```

```
#%%
#implement through ridgeCV
from sklearn.linear_model import RidgeCV
model01=RidgeCV(alphas=[1e-3,1e-2,1e-1,1])
model01.fit(X_poly,y_train)

#%%
#prediction
y_pred_rid=model01.predict(X_poly_test)

#%%
#calculate rmse ,log error and R^2 score
rms_rid=mean_squared_error(y_test, y_pred_rid)
print("RMSE Error: ",rms_rid)
log_rid=mean_squared_log_error(abs(y_test), abs(y_pred_rid))
print("Log Error: ",log_rid)
r2_sc_ridge=r2_score(y_test,y_pred_rid)
print("R^2 Score: ",r2_sc_ridge)

#%%
#implement cross validation model
from sklearn.model_selection import KFold,cross_val_score
folds = KFold(n_splits = 5, shuffle = True, random_state = 100)
scores = cross_val_score(model, X_train, y_train, scoring='neg_mean_absolute_error', cv=folds)
print(np.average(scores) )
```

OUTPUT:

Console 2/A ×

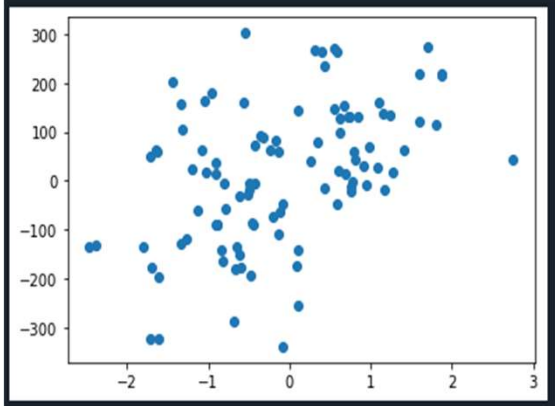
```
Python 3.7.6 (default, Jan  8 2020, 20:23:39) [MSC v.1916 64 bit (AMD64)]
Type "copyright", "credits" or "license" for more information.

IPython 7.26.0 -- An enhanced Interactive Python.

In [1]: runcell(0, 'B:/3rd year/5th sem/P&AD/implement polynomial regression.py')

In [2]: runcell(1, 'B:/3rd year/5th sem/P&AD/implement polynomial regression.py')

In [3]: runcell(2, 'B:/3rd year/5th sem/P&AD/implement polynomial regression.py')
```




```
In [4]: runcell(3, 'B:/3rd year/5th sem/P&AD/implement polynomial regression.py')

In [5]: runcell(4, 'B:/3rd year/5th sem/P&AD/implement polynomial regression.py')

In [6]: runcell(5, 'B:/3rd year/5th sem/P&AD/implement polynomial regression.py')

In [7]: runcell(6, 'B:/3rd year/5th sem/P&AD/implement polynomial regression.py')
RMS Score:
45549.752833300685

In [8]: runcell(7, 'B:/3rd year/5th sem/P&AD/implement polynomial regression.py')
RMS Log error
3.5901531754121336

In [9]: runcell(8, 'B:/3rd year/5th sem/P&AD/implement polynomial regression.py')
R^2
-1.8579184070667165
```

```
In [10]: runcell(9, 'B:/3rd year/5th sem/P&AD/implement polynomial regression.py')

In [11]: runcell(10, 'B:/3rd year/5th sem/P&AD/implement polynomial regression.py')

In [12]: runcell(11, 'B:/3rd year/5th sem/P&AD/implement polynomial regression.py')
RMSE Error:  11937.770611030275
Log Error:  0.8030949533472002
R^2 Score:  0.25099100990815915

In [13]: runcell(12, 'B:/3rd year/5th sem/P&AD/implement polynomial regression.py')
-41.14876980276445

In [14]:
```