```
# %%

# ================================================================================
# PLOYNOMIAL REGRESSION
# ================================================================================


# %%

# importing the libraries

from sklearn.preprocessing import PolynomialFeatures
from sklearn.linear_model import LinearRegression
import matplotlib.pyplot as plt
from sklearn import datasets


# %%

# generating the dataset for 100 samples with linear relation from make_regression() fucntion of skearn

x, y = datasets.make_classification(n_samples=10,  # number of samples
                                    n_features=5)  # number of features


# %%

# visualizing each feature

fig, axs = plt.subplots(2, 3)
axs[0, 0].scatter(x[:, 0], y)
axs[0, 0].set_title('feature 1')
axs[0, 1].scatter(x[:, 1], y)
axs[0, 1].set_title('feature 2')
axs[0, 2].scatter(x[:, 2], y)
axs[0, 2].set_title('feature 3')
axs[1, 0].scatter(x[:, 3], y)
axs[1, 0].set_title('feature 4')
axs[1, 1].scatter(x[:, 4], y)
axs[1, 1].set_title('feature 5')

# Hide x labels and tick labels for top plots and y ticks for right plots.
for ax in axs.flat:
    ax.label_outer()


# %%

# tarining the linear model

lin_reg = LinearRegression()
lin_reg.fit(x, y)

# %%

# training the polynomial model

poly_reg2 = PolynomialFeatures(degree=3)
X_poly = poly_reg2.fit_transform(x)
lin_reg_2 = LinearRegression()
lin_reg_2.fit(X_poly, y)


# %%

# visualizing the result
```

```
67
68  plt.scatter(X_poly[:, 1], y, color='red')
69  plt.plot(X_poly[:, 1], lin_reg_2.predict(X_poly), color='blue')
70  plt.title('Sklearn regression')
71  plt.xlabel('x')
72  plt.ylabel('y')
73  plt.show()
74
75
76  # %%
77
78  # ============================================================================
79  # THE END
80  # ============================================================================
```