11주차 Overffitng 실습

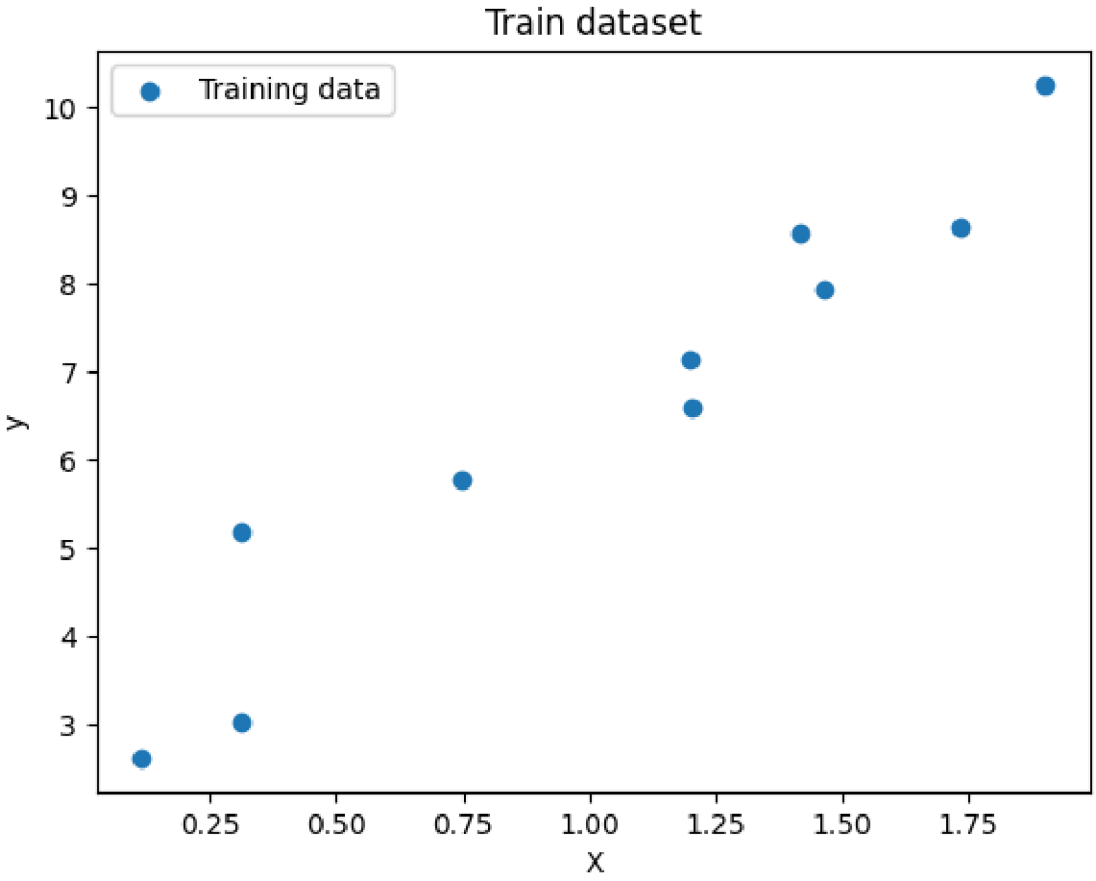
- 다항회귀 모델을 선언하고, 차수에 따른 모델의 경향을 확인한다.

In [ ]:

```
import numpy as np
import matplotlib.pyplot as plt
from sklearn.linear_model import LinearRegression, Lasso, Ridge
from sklearn.preprocessing import PolynomialFeatures
```

In [ ]:

```
# 데이터 생성
np.random.seed(42)
X_train = 2 * np.random.rand(10, 1)
y_train = 4 + 3 * X_train + np.random.randn(10, 1)

# 데이터 분포 확인
plt.scatter(X_train, y_train, label='Training data')
plt.title('Train dataset')
plt.xlabel('X')
plt.ylabel('y')
plt.legend()
plt.show()
```

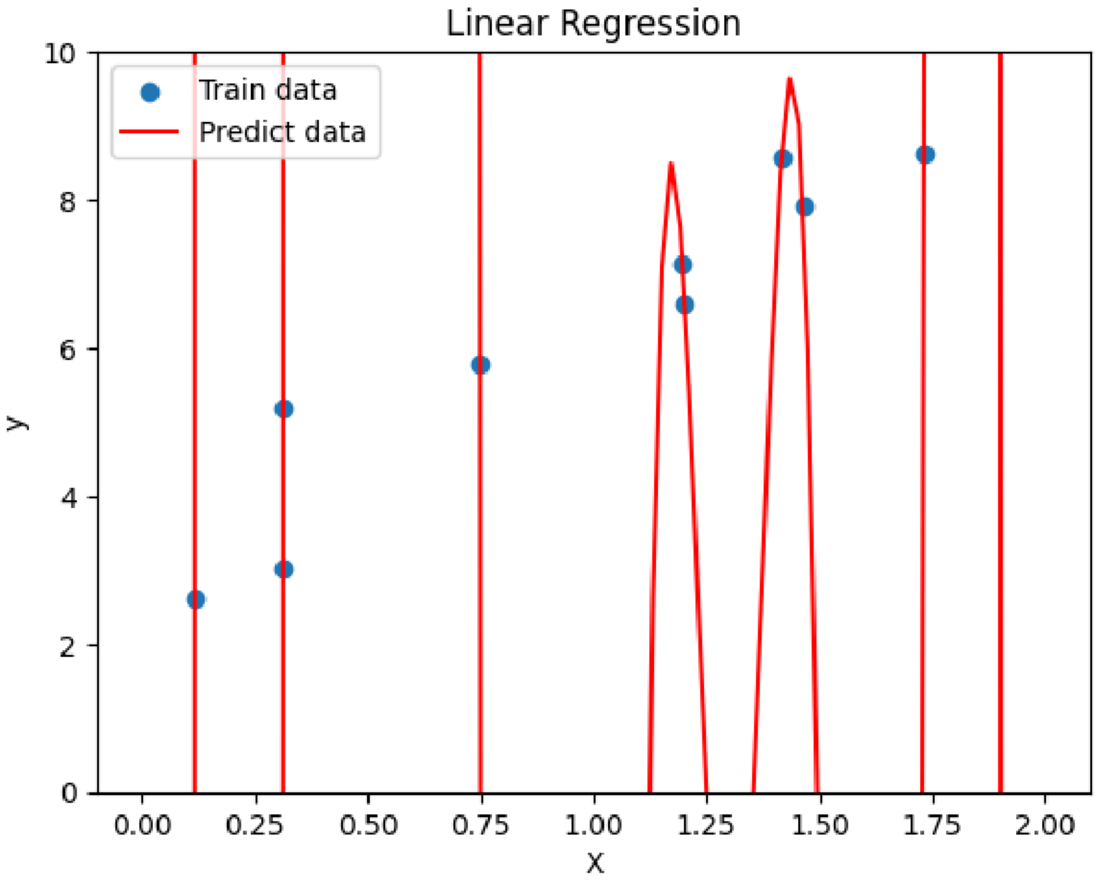


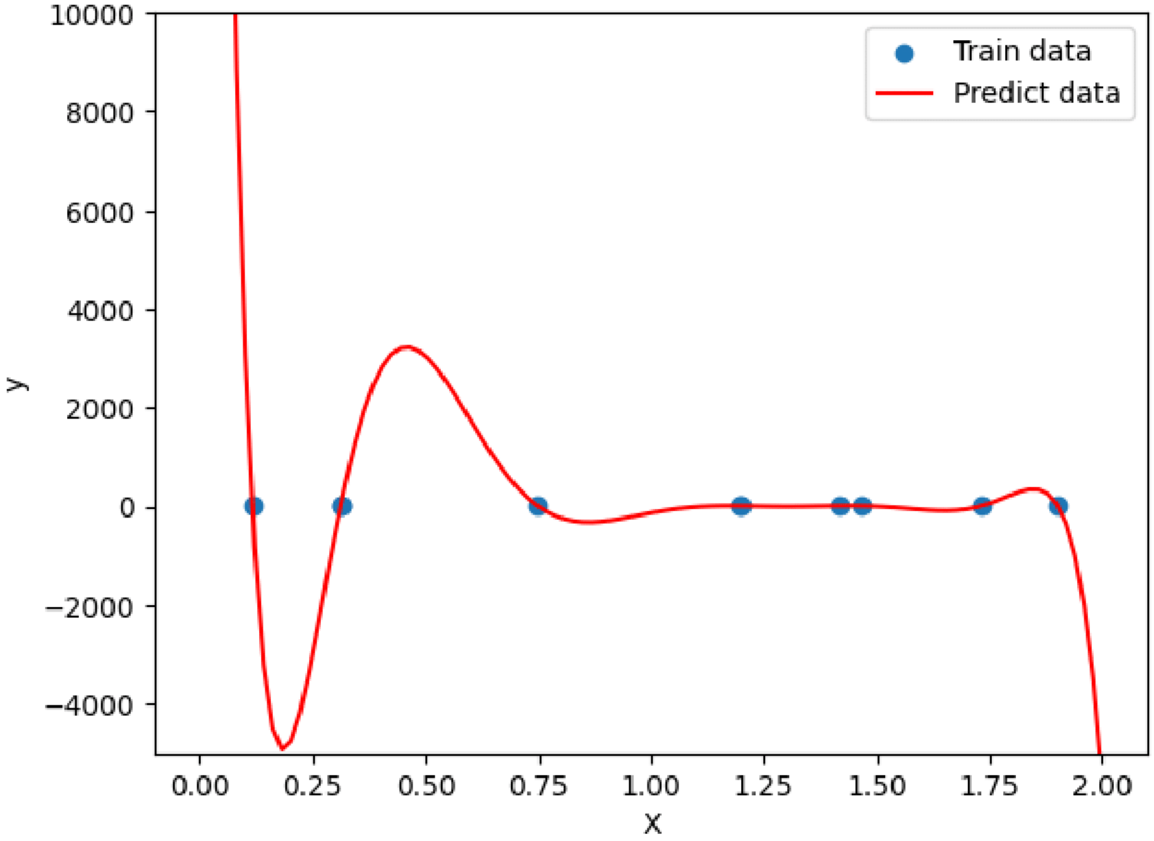
In [ ]:

```
# 도표 출력을 위한 함수 선언
def plot_predict_result(model, max = 10, min = 0):

    X_test = np.linspace(0, 2, 100)[:, np.newaxis]
    y_test = 4 + 3 * X_test + np.random.randn(100, 1)

    X_poly_test = poly_features.transform(X_test)

    y_predict = model.predict(X_poly_test)

    plt.scatter(X_train, y_train, label='Train data')
    plt.plot(X_test, y_predict, label='Predict data', color='r')
    plt.xlabel('X')
    plt.ylabel('y')
    plt.ylim([min, max])
    plt.legend()
    plt.show()
```

In [ ]:

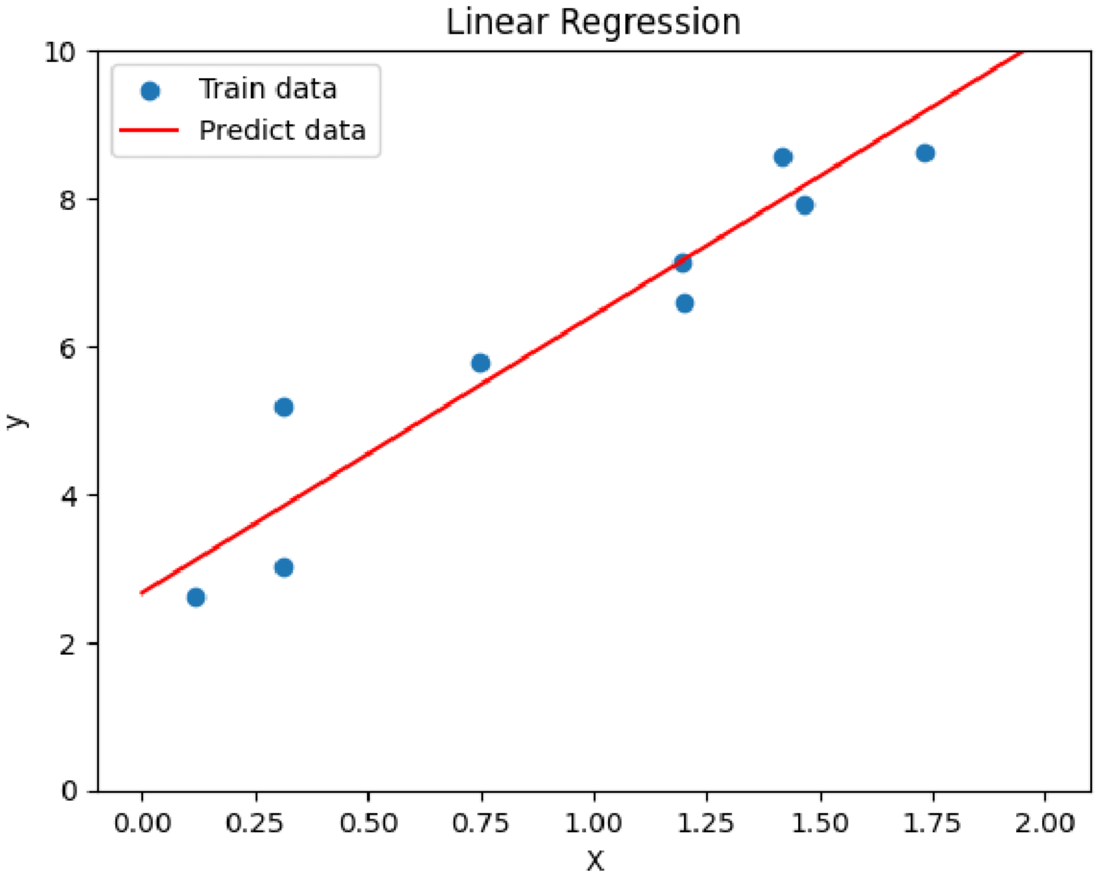
```
# 9차 회귀 모델 -> 너무 높은 차수 (overfitting)
poly_features = PolynomialFeatures(degree=9, include_bias=False)
X_poly_train = poly_features.fit_transform(X_train)

# Linear Regression 모델 학습
lin_reg = LinearRegression()
lin_reg.fit(X_poly_train, y_train)

plt.title("Linear Regression")
plot_predict_result(lin_reg)
plot_predict_result(lin_reg, 10000, -5000)
```





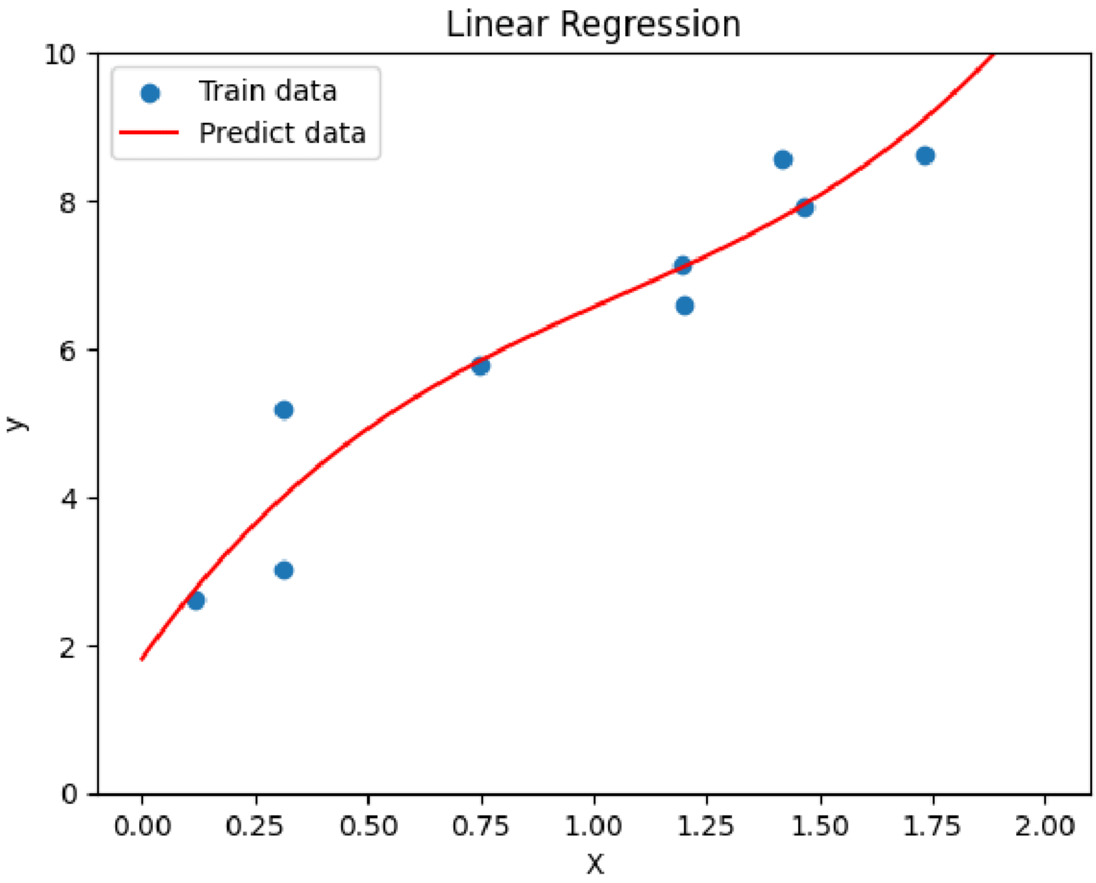
In [ ]:

```
# 1차 회귀 모델 -> 단항 회귀 모델 (linear)
poly_features = PolynomialFeatures(degree=1, include_bias=False)
X_poly_train = poly_features.fit_transform(X_train)

# Linear Regression 모델 학습
lin_reg = LinearRegression()
lin_reg.fit(X_poly_train, y_train)

plt.title("Linear Regression")
plot_predict_result(lin_reg)
```



In [ ]:

```
# 3차 회귀 모델
poly_features = PolynomialFeatures(degree=3, include_bias=False)
X_poly_train = poly_features.fit_transform(X_train)

# Linear Regression 모델 학습
lin_reg = LinearRegression()
lin_reg.fit(X_poly_train, y_train)

plt.title("Linear Regression")
plot_predict_result(lin_reg)
```

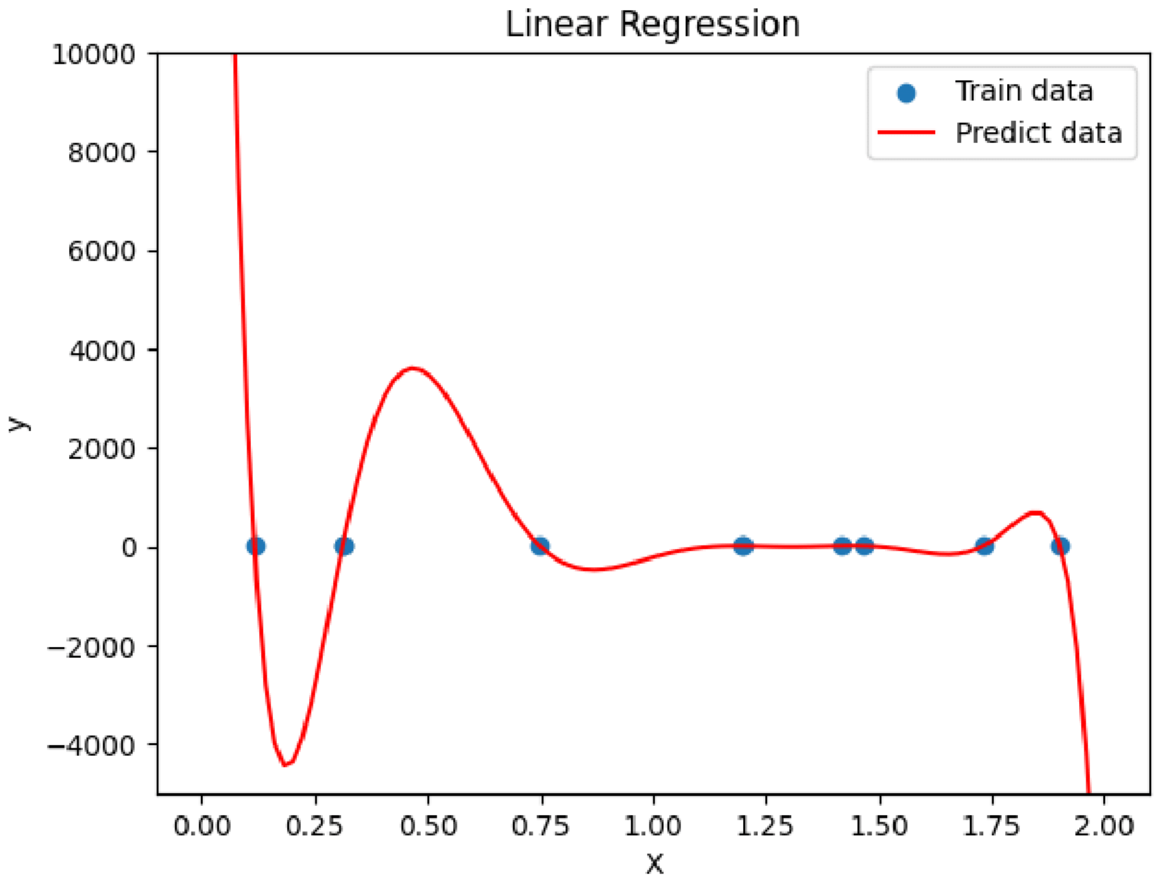


Regularization by Weight Penalty

- Overfitting된 다항회귀 모델에 Regularization을 적용하여 Overfittng을 완화한다.
  - Lasso (L1 Norm)
  - Ridge (L2 Norm)

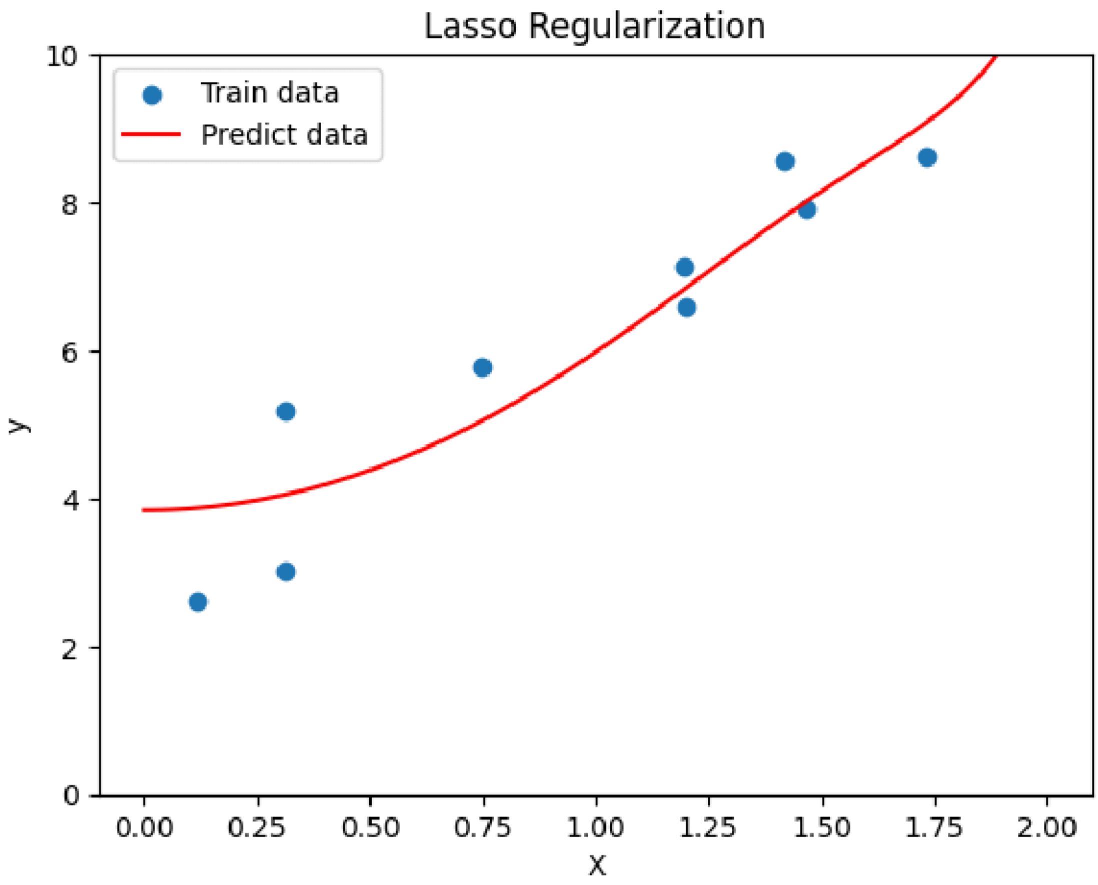
In [ ]:

```
poly_features = PolynomialFeatures(degree=10, include_bias=False)
X_poly_train = poly_features.fit_transform(X_train)

# Linear Regression 모델 학습
lin_reg = LinearRegression()
lin_reg.fit(X_poly_train, y_train)
plt.title("Linear Regression")
plot_predict_result(lin_reg, 10000, -5000)

plt.show()
```



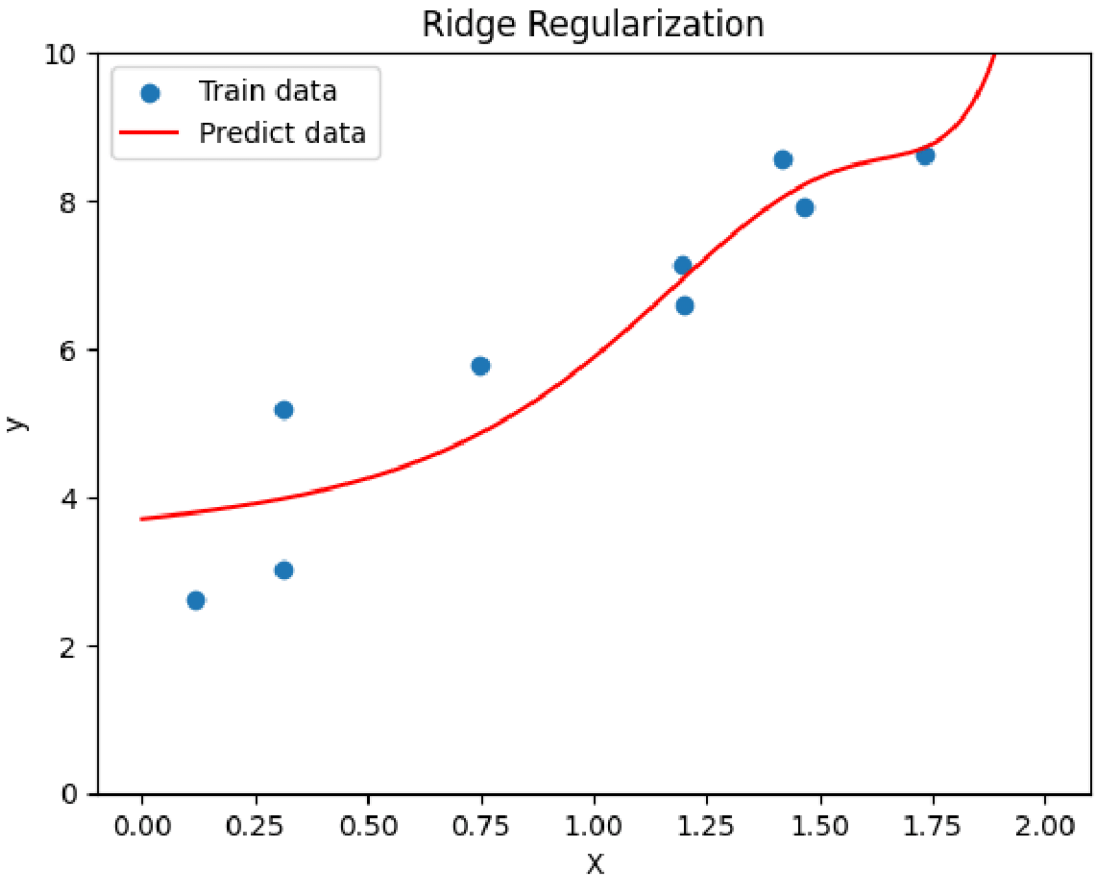
In [ ]:

```
# Lasso(L1) Norm을 적용한 모델 학습

lasso_reg = Lasso(alpha=0.1, max_iter=100000)
lasso_reg.fit(X_poly_train, y_train)

# Lasso 모델 예측 결과
plt.title("Lasso Regularization")
plot_predict_result(lasso_reg, 10, 0)

plt.show()
```



In [ ]:

```
# Ridge(L2) Norm을 적용한 모델 학습

ridge_reg = Ridge(alpha=1, max_iter=10000)
ridge_reg.fit(X_poly_train, y_train)

# Ridge 모델 예측 결과
plt.title("Ridge Regularization")
plot_predict_result(ridge_reg, 10 , 0)

plt.show()
```


Ridge Regularization
Train data
Predict data
y
X