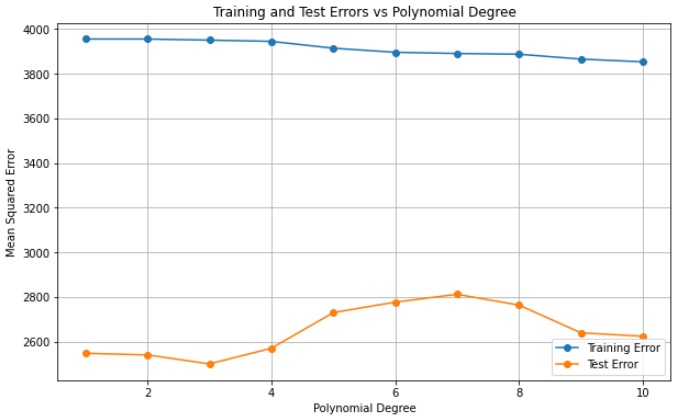
Shreeya Sampat

Lab 7 - Overfitting and Regularization

Step 1: Diabetes dataset and overfitting



We have two curves on this graph. Each represented by different colors.
We have Training Error curve which decreases as the Polynomial Degree increases. This makes sense because as the degree increases they have more flexibility to fit the training data more closely. Another
Then we have Test Error curve which decreases and then increases after the 4th Polynomial degree. This hints at overfitting.

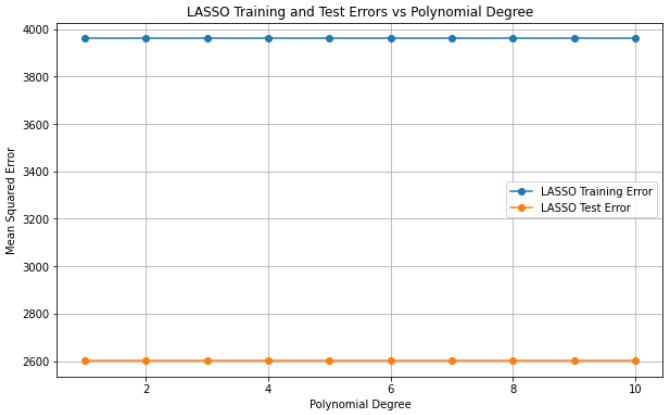
Python Code:

```
# Step 1
import numpy as np
import matplotlib.pyplot as plt
from sklearn.datasets import load_diabetes
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import PolynomialFeatures

# Load the diabetes dataset
diabetes = load_diabetes()
X = diabetes.data[:, 2].reshape(-1, 1)  # Third column
y = diabetes.target

# Split the data into training and test sets (last 20 points for testing)
trainX, testX = X[:-20], X[-20:]
trainY, testY = y[:-20], y[-20:]

# Lists to store errors
train_errors = []
test_errors = []

# Train models with polynomial degrees from 1 to 10
for degree in range(1, 11):
    # Generate polynomial features
```

Lab 7 - Overfitting and Regularization

```
    poly = PolynomialFeatures(degree)
    trainX_poly = poly.fit_transform(trainX)
    testX_poly = poly.transform(testX)

    # Train the linear regression model
    model = LinearRegression()
    model.fit(trainX_poly, trainY)

    # Predict on training and test sets
    train_pred = model.predict(trainX_poly)
    test_pred = model.predict(testX_poly)

    # Calculate mean squared errors
    train_error = mean_squared_error(trainY, train_pred)
    test_error = mean_squared_error(testY, test_pred)

    # Append errors to lists
    train_errors.append(train_error)
    test_errors.append(test_error)

# Plot the errors
plt.figure(figsize=(10, 6))
plt.plot(range(1, 11), train_errors, label='Training Error', marker='o')
plt.plot(range(1, 11), test_errors, label='Test Error', marker='o')
plt.xlabel('Polynomial Degree')
plt.ylabel('Mean Squared Error')
plt.title('Training and Test Errors vs Polynomial Degree')
plt.legend()
plt.grid(True)
plt.show()
```

Step 2: LASSO



As compared to the other graph from Step 1, this graph does not show as much dramatic change in the curves of both, Training Error, and Test Error. This indicates that the margin i\of error is lower with LASSO as compared to without it.

Shreeya Sampat

Lab 7 - Overfitting and Regularization

Python Code:

```
# Step 2
import numpy as np
import matplotlib.pyplot as plt
from sklearn.datasets import load_diabetes
from sklearn.linear_model import Lasso
from sklearn.metrics import mean_squared_error
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import PolynomialFeatures

# Load the diabetes dataset
diabetes = load_diabetes()
X = diabetes.data[:, 2].reshape(-1, 1)  # Third column
y = diabetes.target

# Split the data into training and test sets (last 20 points for testing)
trainX, testX = X[:-20], X[-20:]
trainY, testY = y[:-20], y[-20:]

# Lists to store errors
train_errors_lasso = []
test_errors_lasso = []

# LASSO regularization parameter
alpha = 0.1

# Train models with polynomial degrees from 1 to 10 using LASSO
for degree in range(1, 11):
    # Generate polynomial features
    poly = PolynomialFeatures(degree)
    trainX_poly = poly.fit_transform(trainX)
    testX_poly = poly.transform(testX)

    # Train the LASSO regression model
    lasso = Lasso(alpha=alpha, max_iter=10000)
    lasso.fit(trainX_poly, trainY)

    # Predict on training and test sets
    train_pred_lasso = lasso.predict(trainX_poly)
    test_pred_lasso = lasso.predict(testX_poly)

    # Calculate mean squared errors
    train_error_lasso = mean_squared_error(trainY, train_pred_lasso)
    test_error_lasso = mean_squared_error(testY, test_pred_lasso)

    # Append errors to lists
    train_errors_lasso.append(train_error_lasso)
```

Lab 7 - Overfitting and Regularization

```
    test_errors_lasso.append(test_error_lasso)

# Plot the errors for LASSO
plt.figure(figsize=(10, 6))
plt.plot(range(1, 11), train_errors_lasso, label='LASSO Training Error', marker='o')
plt.plot(range(1, 11), test_errors_lasso, label='LASSO Test Error', marker='o')
plt.xlabel('Polynomial Degree')
plt.ylabel('Mean Squared Error')
plt.title('LASSO Training and Test Errors vs Polynomial Degree')
plt.legend()
plt.grid(True)
plt.show()
```