Problem 1 (OLS, linear regression)

Given a data set $\{(x_i, y_i)\}_{i=1}^{N}$, where $x_i, y_i \in \mathcal{R}$, we want to find a "least squares line" $\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x$ to fit the data set with minimum squared residuals $\sum_{i=1}^{N}(y_i - \hat{y}_i)^2$.

a. Show that:

$$\hat{\beta}_1 = \frac{\sum_{i=1}^{n} x_i y_i - \frac{1}{n}(\sum_{i=1}^{n} x_i)(\sum_{i=1}^{n} y_i)}{\sum_{i=1}^{n} x_i^2 - \frac{1}{n}(\sum_{i=1}^{n} x_i)^2} = \frac{\sum_{i=1}^{n} x_i y_i - n\bar{x}\bar{y}}{\sum_{i=1}^{n} x_i^2 - n\bar{x}^2}$$

$$\hat{\beta}_0 = \frac{1}{n}\sum_{i=1}^{n} y_i - \hat{\beta}_1 \frac{1}{n}\sum_{i=1}^{n} x_i = \bar{y} - \hat{\beta}_1 \bar{x}$$

Where,

$$\bar{x} = \frac{1}{n}\sum_{i=1}^{n} x_i \,; \quad \bar{y} = \frac{1}{n}\sum_{i=1}^{n} y_i$$

b. Use **Python** to generate and visualize a synthetic data set $\{x_i, y_i\}, i = 1,2, \dots, n$, that satisfy

$$y_i = -5x_i + 15 + \epsilon_i, i = 1,2, \dots, n$$

Where, x_is are values evenly spaced between $1\ and\ 10$, i.e., $1 \le x_i \le 10$; ϵ_i is a noise follows a normal distribution with zero mean and standard deviation of 1.

c. Use **Python (do not use sklearn package)** to find the "least squares line" $\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x$ for the data set generated in 1.b and create a figure showing the data points in the data set and the calculated "*least squares line*". Calculate the sum of the squares of the residuals associated with the "*least squares line*".

d. **Use sk-learn package** to verify your result in 1.c

Problem 2 (Polynomial Regression)

Use Python to carry out the following:

a. Generate a synthetic data set with 100 data points (x_i, y_i), where the input $-5 \leq x_i \leq 5$ are evenly spaced and the output $y_i = 12\sin(x_i) + 0.5x_i^2 + 2x_i + 5$ are polluted by Gaussian noise with zero mean and standard deviation of 2. Create a figure showing the scattering of the generated noisy data points and the true pattern of $y = f(x)$.

b. Create a sklearn pipeline model for **polynomial regression** by chaining PolynomialFeatures feature mapping with degree of 5 and a LinearRegression model.

c. Split the data set into training set and test set with test size = 20%. Train the **polynomial regression** model on the training set and make prediction on the test set. Calculate the mean squared test error.

d. Create a figure to show the prediction curve of the trained model on top of the training data points and the true pattern as in a.

Problem 3 (Polynomial Regression, hyperparameter tuning using GridSearchCV)

Use Python to carry out the following:

a. Create a sklearn pipeline model for **polynomial regression** by using PolynomialFeatures feature mapping and a LinearRegression model.

b. Split the data set in Q2.a into training set and test set with test size = 20%. To find the best polynomial model on the given dataset, cross validation method is used to tune the degree of the polynomial feature transformation. Please use GridSearchCV method in sklearn to find the best degree and mean squared test error of the best model.

c. Create a figure to show the prediction curve of the trained model on top of the training data points and the true pattern.

Problem 4 (Underfitting/Overfitting, Polynomial Regression)

Use Python to carry out the following:

a. Generate a synthetic data set with 100 data points (x_i, y_i), where the input $-5 \leq x_i \leq 5$ are evenly spaced and the output $y_i = 12\sin(x_i) + 0.5x_i^2 + 2x_i + 5$ are polluted by Gaussian noise with zero mean and standard deviation of 2. Create a figure showing the scattering of the generated noisy data points and the true pattern of $y = f(x)$.

b. Create a sklearn pipeline model for **polynomial regression** by using PolynomialFeatures feature mapping and LinearRegression model.

c. Split the data set into training set and test set with test size = 20%.

d. Train the **polynomial regression** model on the training set and then use the trained model to make prediction on the test set. Show prediction curves with different polynomial degrees on the test data scatter plot.

e. Repeat step d for the degree of polynomial taking value in [1, 2, 5, 8, 12, 14, 16, 18, 20].

f. Observe the plots, which polynomial degree gives the model the best fitting to the training data? Is that degree optimal for the model?

g. Plot training errors and test errors curves (mean square error) with regard to polynomial degrees. Based on the plots, verify your observation in c. Describe what is underfitting and what is overfitting. Which degree of polynomial feature mapping gives the best generalization performance?