Exercises: Part 1/2 Machine Learning

1. Explain whether each scenario is a classification or regression problem, and indicate whether we are most interested in inference or prediction. Finally, provide n and p.
    a. We collect a set of data on the top 500 firms in the US. For each firm we record profit, number of employees, industry and the CEO salary. We are interested in understanding which factors affect CEO salary.

    b. We are considering launching a new product and wish to know whether it will be a success or a failure. We collect data on 20 similar products that were previously launched. For each product we have recorded whether it was a success or failure, price charged for the product, marketing budget, competition price, and ten other variables.

    c. We are interested in predicting the % change in the USD/Euro exchange rate in relation to the weekly changes in the world stock markets. Hence, we collect weekly data for all of 2012. For each week we record the % change in the USD/Euro, the % change in the US market, the % change in the British market, and the % change in the German market.

2. Describe three real-life applications in which classification might be useful. Describe the response, as well as the predictors. Is the goal of each application inference or prediction? Explain your answer.

3. Describe three real-life applications in which regression might be useful. Describe the response, as well as the predictors. Is the goal of each application inference or prediction? Explain your answer.

4. Describe three reasons why the error term ϵ in the general form $Y = f(x) + \epsilon$ might be non-zero?

5. Use the following Linear Regression code to train (model.fit()) a linear regression model where the input variable is news sentiment score and the response is stock price:

```
from sklearn.linear_model import LinearRegression
import numpy as np
news_sentiment = np.array([0.2, 0.5, 0.3, -0.1, 0.4, 0.6, 0.1, -0.2, 0.3, 0.0]).reshape(-1,1)
stock_price = [50, 55, 48, 45, 52, 58, 53, 47, 51, 50]
model = LinearRegression()
model.fit(news_sentiment,stock_price)
print(f"Intercept: {model.intercept_}, Coefficients: {model.coef_}")
```

    a. The intercept represents the parameter β_0 and the coefficients are the parameters $\beta_1, \dots, \beta_p$ for the number of p predictors, of the Linear Regression Model (Part 2 Slide 10). Given the parameters produced by the model in the code, what is the stock price for a news sentiment score 0.55? What do the parameters tell us about the relationship between news sentiment score and stock price?

b. Use the method model.predict() on the training input.
```
training_preds = model.predict(news_sentiment)
print("Training Predictions: ", training_preds)
```
Use these predictions and the true stock price values for the training points to calculate the training MSE.

c. What does this tell us about our prediction model?

d. Now use the following code to plot the linear regression fit through the training points.
Does the training MSE seem visually correct? Verify your answer for a.

```
import matplotlib.pyplot as plt
plt.rc('font', size=18); plt.rcParams['figure.constrained_layout.use'] = True
plt.scatter(news_sentiment, stock_price, color='red', marker='+')
plt.plot(news_sentiment,training_preds)
plt.xlabel('input x'); plt.ylabel('output y'); plt.legend(['Training Points','Linear Regression'])
plt.show()
```