Problem 1. *Linear Regression by Hand*

Suppose we have the following dataset:

X	Y
1	3
2	5
3	7
4	9
5	11

Table 1: Dataset

We want to perform Linear Regression to fit a line of the form $Y = \theta_0 + \theta_1 \cdot X$ to this data.

Steps:

1. *Calculate the Mean of X and Y*
2. *Calculate the Deviations*
3. *Calculate the Cross-Deviation and Squared Deviations*
4. *Calculate the Coefficients θ_1 and θ_0 Using the formulas for θ_1 and θ_0 :*

$$\theta_1 = \frac{\sum((X - \bar{X}) \cdot (Y - \bar{Y}))}{\sum(X - \bar{X})^2}$$
$$\theta_0 = \bar{Y} - \theta_1 \cdot \bar{X}$$

5. *Final Linear Regression Equation*

Problem 2. *Linear Regression Closed Form*

Given $X = \begin{bmatrix} 1 & 0 \\ 1 & 1 \\ 1 & 2 \\ 1 & 3 \end{bmatrix}$ *and* $y = \begin{bmatrix} 2 \\ 1 \\ 4 \\ 4 \end{bmatrix}$.

The coefficients θ_0, θ_1 for the best fit line $f(x) = \theta_0 + \theta_1 x$ are given by $\begin{bmatrix} \theta_0 \\ \theta_1 \end{bmatrix} = \left(X^T X\right)^{-1} X^T y$.

Find the best-fit line.

Problem 3. *You have been given a data set containing gas mileage, horsepower, and other information for 398 makes and models of vehicles. For each vehicle, we have the following information (you don't need the car name):*

#	Column	Non-Null Count	Dtype
0	acceleration	398	float64
1	car name	398	object
2	cylinders	398	int64
3	displacement	398	float64
4	horsepower	398	int64
5	model year	398	int64
6	mpg	398	float64
7	origin	398	int64
8	weight	398	int64

Table 2: Data columns information

To download the data use the Setify library as follows:

```
! pip install setify
from setify import datasets
df = datasets.data('miles_per_gallon')
```

Listing 1: Load Miles per Gallon Dataset

You will need to develop a model to predict vehicle mileage (mpg) from other vehicle characteristics. Your goal is to determine which polynomial transformation yields the best predictive performance.

Steps:

1. *Fit a linear regression model to a polynomial feature transformation of the provided training set at each of these possible degrees: [1, 2, 3, 4, 5, 6, 7]. For each* ***hyperparameter*** *setting, record the training set error and the validation set error.*

2. *Select the model* ***hyperparameters*** *that minimize your fixed validation set error. Using your already-trained* ***LinearRegression*** *model with these best* ***hyperparameters****, compute the error on the test set.*

3. ***Figure****: Make a line plot of a mean-squared error on the y-axis vs. polynomial degree on the x-axis. Show two lines, one for error on the training set (use style 'b:', a dotted blue line) and another line for error on the validation set (use style 'rs-', a solid red line with square markers). Set the y-axis limits between [0, 70].*

4. *If your goal is to select a model that will generalize well to new data from the same distribution, which polynomial degree do you recommend based on this assessment? Are there other degrees that seem to give nearly the same performance?*

Problem 4. *Logistic Regression by hand*
Use the following dataset to train a logistic regression model, only one update of the weights is required, you can assume initial weights, also, use cross-entropy as a loss function (show all the steps)

x_1	x_2	y
1	2	0
5	8	1

Table 3: Dataset

Problem 5. *Logistic Regression coding question In this machine learning exercise, we will work with the Iris dataset available in Setify iris. We will focus on only the first two features of the dataset. Our goal is to build a binary classifier, where we assign 0 for the class 'setosa' and 1 for the combined classes 'versicolor' and 'virginica'.*

To begin, we will split the data into training and testing sets. Then, we will proceed with training a classifier using the chosen features. Once the model is trained, we will evaluate its accuracy on the testing data.

The accuracy of our model will give us a measure of how well it performs in predicting the target classes. We will then go on to visualize and explain a confusion matrix, which provides valuable insights into the classifier's performance.

For this task, we can leverage popular libraries such as scikit-learn to facilitate the training process and assess the accuracy of our model.

```
import matplotlib.pyplot as plt
from setify import datasets
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score, confusion_matrix
```

Listing 2: Load Iris Dataset

Problem 6. *Consider the following toy dataset with two features (x1 and x2) and their corresponding class labels (0 or 1):*

Data Point	x_1	x_2	Class
1	1	2	0
2	2	3	0
3	3	5	0
4	4	7	1
5	5	8	1

Table 4: Dataset

Let's assume we have a new data point with features ($x_1 = 3$, $x_2 = 6$) that we want to classify. Use the KNN algorithm to classify the data point.

Steps:

1. *Choose K*
2. *Calculate distance*
3. *Select K Nearest Neighbors*
4. *Vote*
5. *Predict*