**Practical session: Regression model**

1. **Linear Regression**

The goal is now to build a model able to linearly explain the target variable **Y** using the other available explanatory variables: X1 , X2 , …., Xm. The linear regression model is given by:

Where is the hypothesis function, using the model parameters . In this case using:

Recall that training a model means setting its parameters so that the model best fits the training set. the most common performance measure of a regression model is the Root Mean Square Error (RMSE):

In practice, it is simpler to minimize the Mean Square Error (MSE) than the RMSE, and it leads to the same result (because the value that minimizes a function also minimizes its square root)

**Part 1: Simple Linear Regression**

**Q1**: Build a model with hypothesis  to predict the land price from an input land area (see data\_1.1.csv). Using Gradient Descent is to minimize a cost function (MSE) denoted by:

=

Update the parameter values: *Gradient Descent step*

1. Calculate , and loss function. Display graphic for loss function.
2. Testing learning rate and plot your model following the learning rate.

**Q2:** Now build a model with hypothesis: for dataset (data\_1.1.csv)

Using some optimizations.

**Part 2: Multi Linear Regression: Boston Housing Dataset**

Apply machine learning techniques using Linear Regression to be predict the values: Median value of owner-occupied homes **(MEDV).** Before build model you following many tasks below:

* **Load Data and Check Data**
* Load the dataset
* Use appropriate methods to check the structure and contents of the dataset.
* **Preprocess the Data**
* Check for duplicate entries and remove them.
* Identify and handle missing values (either by removing or imputing).
* Clean or encode categorical variables if necessary.
* **Check Correlation**
* Calculate the correlation matrix.
* Identify which features are strongly correlated with the target variable.
* **Visualization (Scatter Plot)**
* Create scatter plots to visualize the relationship between key features and the target variable.
* **Split Data (80% Train, 20% Test)**
* Define the features and target variables.
* Split the dataset into training and testing sets, using an 80/20 ratio

**Part 3: Multi Linear Regression**

Description: By using house price dataset. Please Convert house price into polynomial form. Each feature will quadratic 1, 2, 3, 4 respectively.

1. Finding gredient descent and it’s cost for normal value with quadratic value.

1. **Logistic Regression**

**Part 1:** Create and defines the LogisticRegression class, which implements logistic regression from scratch using gradient descent. The following steps explain how the class works:

* \_\_init\_\_ Method:
  + Initializes the logistic regression model with two parameters:
    - learning\_rate: Defines the step size for gradient descent updates.
    - n\_iterations: Specifies the number of iterations for the optimization process.
* fit(X, y) Method:
  + Fits the model to the training data (X is the feature matrix, y is the target labels).
  + Initializes the weights and bias to zero.
  + Iteratively updates the weights and bias using gradient descent based on the sigmoid function and the error between predicted values and actual target values.
* predict(X) Method:
  + Uses the learned weights and bias to predict the binary class (0 or 1) for the input data X.
  + Applies the sigmoid function to convert the linear combination of inputs into probabilities, and then thresholds the probabilities to make binary predictions.
* \_sigmoid(z) Method:
  + Implements the sigmoid function, which maps the linear model output to a probability between 0 and 1.

Demonstrates how to use the LogisticRegression class defined in the previous file to train and test the model using the breast cancer dataset from sklearn.

**Part 2:** **Logistic Regression: Breast Cancer Dataset**

Apply machine learning techniques using Logistic Regression. We are going to implement a Logistic Regression classification on breast cancer dataset using PyTorch. We will also look into data pre-processing like, deleting null values, and dropping columns etc.

[Breast Cancer Dataset (kaggle.com)](https://www.kaggle.com/datasets/uciml/breast-cancer-wisconsin-data)

* **Load Data**
* Load the dataset
* Use appropriate methods to check the structure and contents of the dataset.
* **Data Preprocessing**
* Handle any null values by dropping or imputing them.
* Drop any unnecessary columns
* Clean or encode categorical variables if necessary.
* **Split Data into Features**
* Separate the features (X) and the target (y)
* **Normalize the Data**
* Normalize the features for better performance (Scaler, etc.)
* **Split Data into Train and Test**
* Split the dataset into training and testing sets, using an 80/20 ratio.
* **Convert Data to PyTorch Tensors**
* Convert the training and test sets into PyTorch tensors
* **Apply Logistic Regression Model**
* Create a logistic regression model using PyTorch
* **Define Loss Function and Optimizer**
* Use binary cross-entropy as the loss function and SGD as the optimizer.
* **Train Model**
* Train the model over a set number of epochs.
* **Evaluate**
* Plot confusion metrics