National University of Computer and Emerging Sciences



Lab Exercise 11

AL2002-Artificial Intelligence Lab

|  |  |
| --- | --- |
| Course Instructor | Dr. Hajra Waheed |
| Lab Instructor (s) | Muhammad Ahmed  Hamza Ayub |
| Sections | B1 - B2 |
| Semester | Spring 2023 |

Department of Computer Science

FAST-NU, Lahore, Pakistan

Table of Contents

[1](#_30j0zll) Objectives 3

[2](#_1fob9te) Linear Regression 3

[3](#_3znysh7) Exercise (50 Marks) 3

[3.1](#_3dy6vkm) Steps to perform 3

[4](#_tyjcwt) Submission Instructions 4

# Objectives

After performing this lab, students shall be able to understand the following:

* Linear Regression
* Multiple Linear Regression

# Linear Regression:

Linear regression is a statistical method that aims to find a linear relationship between a dependent variable (also known as the response variable) and one or more independent variables (also known as predictor variables).

How to Implement Linear Regression from Scratch:

In this implementation, we will focus on simple linear regression, where we have only one independent variable.

The general equation of a straight line is given by:

**y = mx + c**

Where:

* y is the dependent variable
* x is the independent variable
* m is the slope of the line
* c is the y-intercept (the point where the line intersects the y-axis)

The goal of linear regression is to find the values of m and c that best fit the data. In other words, we want to find the line that has the smallest distance between the predicted values and the actual values.

Here are the steps involved in implementing linear regression from scratch:

1. **Data Preparation:** Collect and clean the data. Ensure that the data is in a format that is suitable for linear regression. This includes removing any missing or erroneous values, and ensuring that the data is numerical.
2. **Define the Problem:** Determine the target variable and the independent variables that will be used in the model. Also, determine the type of linear regression that will be used, i.e., simple or multiple linear regression.
3. **Define the Model:** Define the mathematical equation that will be used to predict the target variable. For simple linear regression, the equation is:

**y = mx + c**

For multiple linear regression, the equation is:

**y = b0 + b1x1 + b2x2 + ... + bnxn**

where y is the target variable, xi are the independent variables, and bi are the coefficients.

1. **Define the Cost Function:** Define a cost function that measures the difference between the predicted values and the actual values. The most commonly used cost function for linear regression is the Mean Squared Error (MSE) function, which is defined as:

**MSE = 1/n \* sum((y - y\_hat)^2)**

where n is the number of observations, y is the actual value of the target variable, and y\_hat is the predicted value of the target variable.

1. **Gradient Descent:** Use gradient descent to optimize the coefficients of the model to minimize the cost function. Gradient descent is an iterative optimization algorithm that updates the coefficients in the direction of the steepest descent of the cost function. The formula for updating the coefficients is:

**b\_i = b\_i - alpha \* (1/n) \* sum((y\_hat - y) \* x\_i)**

where alpha is the learning rate, n is the number of observations, x\_i is the ith feature, y is the actual value of the target variable, and y\_hat is the predicted value of the target variable.

1. **Train the Model:** Use the prepared data to train the model. This involves applying gradient descent to update the coefficients of the model iteratively until the cost function is minimized.

1. **Make Predictions:** Use the trained model to make predictions on new data. This involves applying the learned coefficients to the new data using the mathematical equation defined in step 3.
2. **Evaluate the Model:** Evaluate the performance of the model using appropriate metrics such as Mean Absolute Error (MAE) or R-squared.

**Implementation**

**Step 1: Import the required libraries:**

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

**Step 2: Load the data:**

data = pd.read\_csv('data.csv')

**Step 3: Explore the data:**

data.head() #Explore the complete dataset , remove missing values and outliers if any

data.describe()

**Step 4: Create the X and y variables:**

X = data['independent\_variable'].values

y = data['dependent\_variable'].values

**Step 5: Calculate the mean of X and y:**

mean\_x = np.mean(X)

mean\_y = np.mean(y)

**Step 6: Calculate the slope (m) and y-intercept (c):**

n = len(X)

numerator = 0

denominator = 0

for i in range(n):

numerator += (X[i] - mean\_x) \* (y[i] - mean\_y)

denominator += (X[i] - mean\_x) \*\* 2

m = numerator / denominator

c = mean\_y - (m \* mean\_x)

**Step 7: Create the predicted values:**

y\_pred = m \* X + c

**Step 8: Calculate the mean squared error (MSE):**

mse = np.mean((y - y\_pred) \*\* 2)

**Step 9: Visualize the data and the regression line:**

plt.scatter(X, y)

plt.plot(X, y\_pred, color='red')

plt.show()

This is a simple implementation of linear regression. There are many variations of linear regression, such as multiple linear regression, polynomial regression, and logistic regression. Each of these variations involves different techniques for finding the line that best fits the data.

# Exercise (50 Marks)

Implement multiple linear regression from scratch in Python that includes,

* Load the dataset
* Preprocess the data
* Split the data
* Implement the Algorithm
* Train the model
* Evaluate the mode on test data
* Interpret the results

You are required to write your own the fit(), predict(), MAPE() , MAE() ,r2\_score() method along with the implementation.

Use the following dataset: **titanic\_train.csv, titanic\_test.csv**

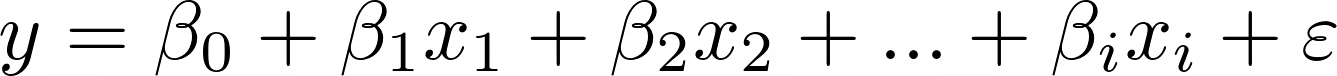
**Goal:** Predict whether or not the passenger survived the sinking of the Titanic.

Remove unnecessary features if any.

## Data Dictionary

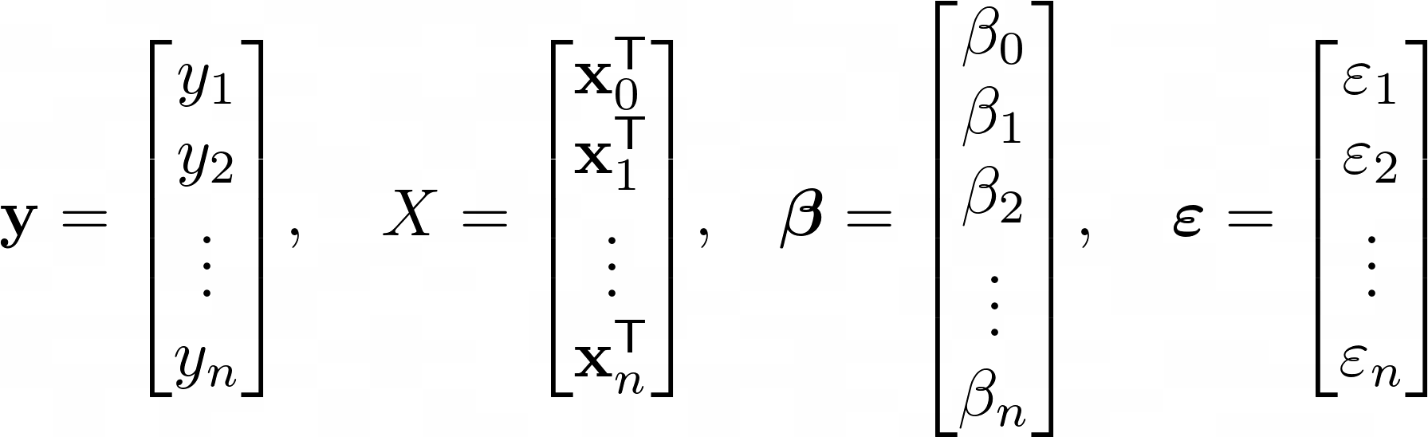
|  |  |  |
| --- | --- | --- |
| Variable | Definition | Key |
| survival | Survival | 0 = No, 1 = Yes |
| pclass | Ticket class | 1 = 1st, 2 = 2nd, 3 = 3rd |
| sex | Sex |  |
| Age | Age in years |  |
| sibsp | # of siblings / spouses aboard the Titanic |  |
| parch | # of parents / children aboard the Titanic |  |
| ticket | Ticket number |  |
| fare | Passenger fare |  |
| cabin | Cabin number |  |
| embarked | Port of Embarkation | C = Cherbourg, Q = Queenstown, S = Southampton |

The general formula for multiple linear regression outcome is as follows:

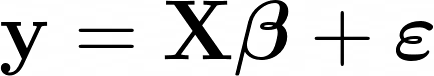


* β0 is known as the intercept.
* β0 to βi are known as coefficients.
* x1 to xi are the features of the data set.
* ε are the residual terms.

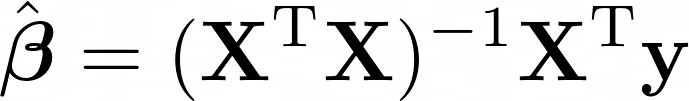
We can also represent the formula in vector notation.



The output equation would look like:



To find the coefficient values, we use the ordinary least square method which minimizes the sum of the square of the residuals. The formula for resulting coefficients in vector notation is:

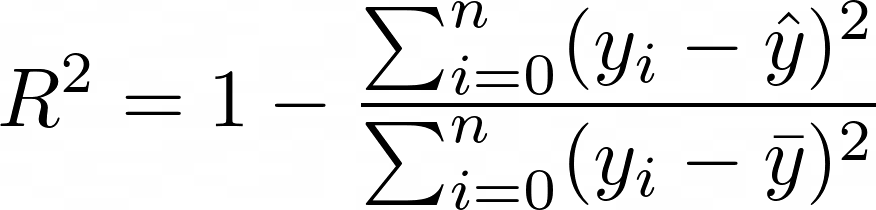


The derivation of this formula is out of the scope of this lab. If you want to learn, read more about it here: <https://web.stanford.edu/~mrosenfe/soc_meth_proj3/matrix_OLS_NYU_notes.pdf>

The outcome of the algorithm, beta β with a hat (^) on top, is a vector containing all of the coefficients that can be used to make predictions by using the formula presented in the beginning for multiple linear regression.

Assuming that you have a number of predictions for some observed data, how can you measure how well the model predicted the ground truth?

You can measure the accuracy of how well the multiple linear regression model performs by using a metric called r squared.



* yi is one value of y at index i.
* ŷ is pronounced as y hat and is the predicted values of y.
* y̅ is pronounced as y bar and is the average of y.

The metric measures the relationship between the residual sum of squares (RSS) and the total sum of squares (TSS). The RSS is computed as the ground truth minus the predicted ground truth, while the TSS is computed as the ground truth minus the average of the ground truth.

**Good Luck! ☺**

# Submission Instructions

Always read the submission instructions carefully.

* Rename your Jupyter notebook to your roll number and download the notebook as **.ipynb** extension.
* To download the required file, go to **File->Download .ipynb**
* Only submit the **.ipynb** file. DO NOT **zip** or **rar** your submission file.
* Submit this file on Google Classroom under the relevant assignment.
* Late submissions will not be accepted.