Learn How to Build a Linear Regression Model in PyTorch

Overview

Linear Regression assumes linear relationships between the predictor and the target variable. To be more specific, the output variable can be calculated by the linear combination of the input variables. Linear regression generally uses the ordinary least squares method for calculating the coefficient of the input variable. The linear regression model can be used for solving many business problems like predicting the number of sales, predicting blood pressure, predicting house prices, and many more.

In this project, a linear regression model for predicting the number of days subscribed has been implemented from scratch in Pytorch. If you haven't already visited, here is the previous project of the series Learn How to Build PyTorch Neural Networks From Scratch.

Aim

- To understand the linear regression and loss function
- To build a linear regression model from scratch in PyTorch for predicting the number of days subscribed

Data Description

The dataset used in this project has information about the number of days subscribed based on the various features.

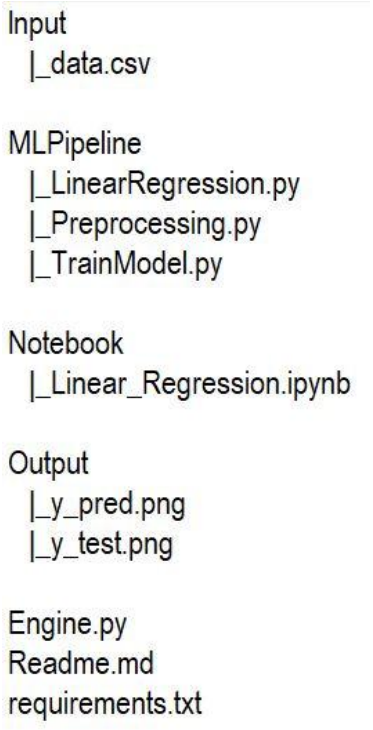
Tech Stack

- ➔ Language: Python
- ➔ Libraries: pandas, pytorch, matplotlib, sci-kit learn, numpy, torchvision, seaborn

Approach

- Data Cleaning
- Data Preprocessing
- Building Linear Regression model
- Model Training

Modular Code Overview:



```
Input
  |_data.csv

MLPipeline
  |_LinearRegression.py
  |_Preprocessing.py
  |_TrainModel.py

Notebook
  |_Linear_Regression.ipynb

Output
  |_y_pred.png
  |_y_test.png

Engine.py
Readme.md
requirements.txt
```

Once you unzip the pytorch_linear_regression.zip file, you can find the following folders within it.

1. Input
2. ML_Pipeline
3. Notebook
4. Output
5. Engine.py
6. Readme.md
7. requirements.txt

1. The Input folder contains the data that we have for analysis. In our case, it contains data.csv

2. The Notebook folder contains the jupyter notebook file of the project

3. The Output folder contains the plot of the predicted and actual target variable

4. The ML_pipeline is a folder that contains all the functions put into different python files, which are appropriately named. These python functions are then called

inside the Engine.py file.

5. The requirements.txt file has all the required libraries with respective versions. Kindly install the file by using the command **pip install -r requirements.txt**

6. **All the instructions for running the code are present in Readme.md file**

Takeaways

1. What are Tensors?
2. Difference between PyTorch Tensors and Numpy array
3. What is data preprocessing?
4. What are categorical and numerical features?
5. Why do we need data preprocessing?
6. How to perform label encoding categorical features?
7. How to scale numerical features?
8. What is Linear Regression?
9. Types of linear regression
10. What is the Loss function?
11. What is the mean squared error, and why do we use it?
12. What is R-squared?
13. What is model optimization?
14. What is stochastic gradient descent?
15. Implementation of linear regression in PyTorch