Birla Institute of Technology and Science Pilani, Hyderabad Campus

2nd Semester 2023-24, BITS F464: Machine Learning

Assignment No: 3, Date Given:16th Feb 2024, Date of Sub: 1st March 2024

Linear and Logistic Regression using TensorFlow

Max. Marks: 05

In this assignment you will perform linear regression and logistic regression using tensorflow. In the context of linear regression, the objective is to discern a linear association between input features and the target variable. This entails the formulation of a computational graph, the delineation of a loss function, and the refinement of model parameters in TensorFlow. Analogously, logistic regression, predominantly employed for binary classification, necessitates the construction of a computational graph, definition of a loss function, and enhancement of model parameters. In instances where binary classification may not be suitable for a specific problem, the alternative is to undertake multiclass classification, employing the Softmax activation function in lieu of sigmoid which was covered in the class.

You are given with the below dataset (downloaded from Kaggle):

The dataset consists of essential information regarding university applicants, featuring distinct attributes such as GRE scores, TOEFL scores, university ratings, statements of purpose (SOP), letters of recommendation (LOR), cumulative grade point averages (CGPA), and research experience. Each row represents an individual applicant, uniquely identified by the "Serial No." column. The primary goal is to utilize this dataset for predictive modelling, specifically to determine the likelihood of admission ("Chance of Admit") for each student. Leveraging machine learning techniques, the features provided in the dataset, spanning academic achievements, recommendations, and research experience, will be employed to develop a model capable of predicting the probability of an applicant's admission to the university.

Admission_Predict_A3

Serial No.	GRE Score	TOEFL Score	University Rating	SOP	LOR	CGPA	Research	Chance of Admit
1	337	118	4	4.5	4.5	9.65	1	0.92
2	324	107	4	4	4.5	8.87	1	0.76
3	316	104	3	3	3.5	8	1	0.72
4	322	110	3	3.5	2.5	8.67	1	0.8
5	314	103	2	2	3	8.21	0	0.65
6	330	115	5	4.5	3	9.34	1	0.9
7	321	109	3	3	4	8.2	1	0.75
8	308	101	2	3	4	7.9	0	0.68
9	302	102	1	2	1.5	8	0	0.5
10	323	108	3	3.5	3	8.6	0	0.45
11	325	106	3	3.5	4	8.4	1	0.52
12	327	111	4	4	4.5	9	1	0.84
13	328	112	4	4	4.5	9.1	1	0.78
14	307	109	3	4	3	8	1	0.62

(Please Turn Over)

Note: Before starting with the assignment, you need to import tensorflow and install Keras library. Following are the tasks to be performed:

Tasks

- Import OS environment variables and other supporting libraries like Pandas, NumPy, TensorFlow.
- Load the provided .csv file into the code converting it into a Pandas dataframe.
- As we can see different columns have different ranges of values, you have to perform scaling using standard scaler. While performing standard scaler from scikitlearn, it automatically converts pandas dataset to numpy arrays, therefore you do not need to explicitly convert it, as tf accepts numpy.
- Next convert the data into Tenserflow tensors as follows:

```
X_train_tensor = tf.constant(X_train_scaled, dtype=tf.float32)
y_train_tensor=tf.constant(y_train.values.reshape(-1,1),
dtype=tf.float32)  # Reshape to (400, 1)
```

- Initialize weights and bias as TensorFlow variables, define the linear regression model and the loss function with the help of TenserFlow 'matmul()', 'square()' and 'reduce_mean()' methods.(We use these functions since we are operating on TensorFlow variables)
- Chose Stochastic Gradient Descent (SGD) optimizer with a learning rate of 0.01. It determines the size of the step taken during optimization when updating the model parameters (weights and biases) in the direction that minimizes the loss function. Choosing an appropriate learning rate can ensure the stability of the training process. A well-chosen learning rate prevents the model from oscillating around the minimum or diverging.

```
optimizer = tf.optimizers.SGD(learning_rate=0.01)
```

- Train the model for 1000 epochs, in each of which predictions are made using the current model parameters, loss is calculated using the mean squared error, gradients of the loss with respect to weights and biases are calculated and the optimizer updates the weights and biases using the gradients.
- Evaluate the trained model using the testing data and print the MSE as the evaluation parameter. Actual vs predicted probabilities are to be plotted using matplotlib to visualize the performance of the model.
- Implement logistic regression, display the loss for each of the 50 epochs, and conclude by presenting the accuracy. Figure out if Logistic regression with categorical output is meaningful in such a scenario. One may use the Binning process which is a Pre-processing step needed to map numerical values to categorical for converting a regression into classification problem in Logistic regression as described below:

(Please Turn Over)

- In transitioning to logistic regression for the given dataset, certain modifications are imperative. Given that the "Chance of Admit" column exhibits a continuous nature, while logistic regression is traditionally employed for classification, a crucial step involves implementing a "binning" process. This entails categorizing the admission chances into three distinct levels: low, medium, and high.

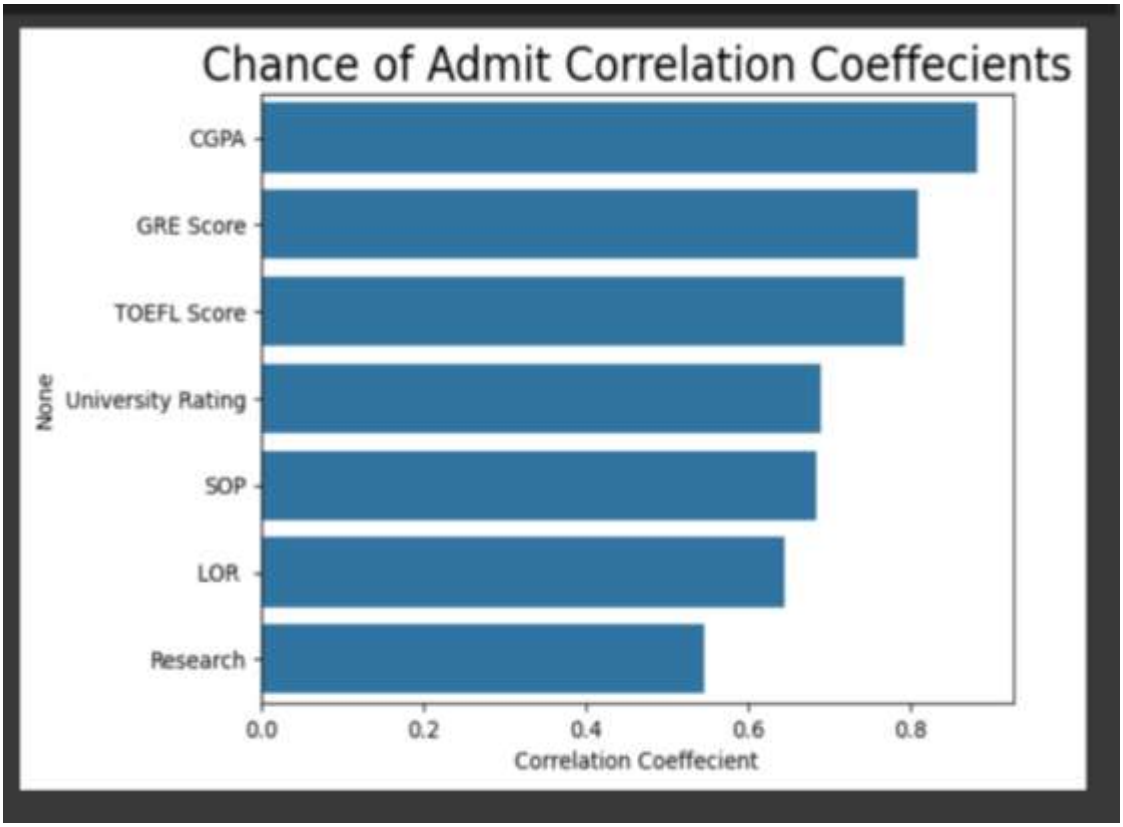
```
import pandas as pd

bin_edges = [0.3, 0.5, 0.7, 1.0]
bin_labels = ['Low', 'Medium', 'High']

df['Admit Category'] = pd.cut(df['Chance of Admit '], bins=bin_edges, labels=bin_labels, include_lowest=True)
```

- It is essential to acknowledge that logistic regression primarily facilitates binary classification; however, with the introduction of three classes in our scenario, the appropriate activation function becomes softmax. Softmax, as a multiclass activation function, is adept at converting the logistic regression model into a robust tool for handling multiple classes. Therefore, write in the discussions how softmax would help us in multiclass classification.

Additional Tasks:

- Throughout the tasks mentioned earlier, you likely encountered the concept of the learning rate. Now, it's important to determine the optimal learning rate. Analyse the consequences of setting it too high or too low by observing the impact on model performance. Illustrate these effects by showcasing changes in the model's performance.
- Determine the correlation coefficients between all variables and the target, then create a bar plot (as shown in Fig.1) visualizing the significant contributions of each feature to the target. Conduct feature selection by choosing the top features, and subsequently assess the model's performance. Determine if there is an improvement, if it remains consistent, or if there is a decline in performance.
- Assess the effectiveness of linear regression and logistic regression by conducting a comparative analysis. Utilize k-fold cross-validation to split the dataset, considering the usage of multiple models in the evaluation process.
- Perform Logistic regression on the same dataset with TensorFlow, using Sigmoid function as the activation, convert the target variable to binary value with a threshold of 0.5, and compare this model's accuracy with your linear regression model to find out which one is a better model, write down the observations in your notebook (code file). Vary the hyper-parameters and the threshold value and note the observations.

(Please Turn Over)



(Fig.1)

Submission Instructions:

Maintain the same grouping as that of the first assignment. No new groups are allowed at this stage. Clean your Notebook code (ipynb) and name your submission file as IDNo.ipynb. Write a readme.text with your group members name and ID numbers. Compress these two files into a single Zip, and name your Zip file using your idno (in lowercase). Make only one submission on behalf of your group in Google Class Assignment Submission page. Deadline for submitting your work is 12th February 2024 midnight.

Note: Any clarification on this coding assignment may be emailed to I/C or f20202001@hyderabad.bits-pilani.ac.in, or f20210982@hyderabad.bits-pilani.ac.in.

References:

1. https://www.javatpoint.com/linear-regression-in-tensorflow
2. https://www.geeksforgeeks.org/understanding-logistic-regression/
3. https://www.altoros.com/blog/using-linear-regression-in-tensorflow/
4. https://www.altoros.com/blog/using-logistic-and-softmax-regression-with-tensorflow/
