Birla Institute of Technology and Science Pilani, Hyderabad Campus

1st Semester 2024-25, BITS F464: Machine Learning

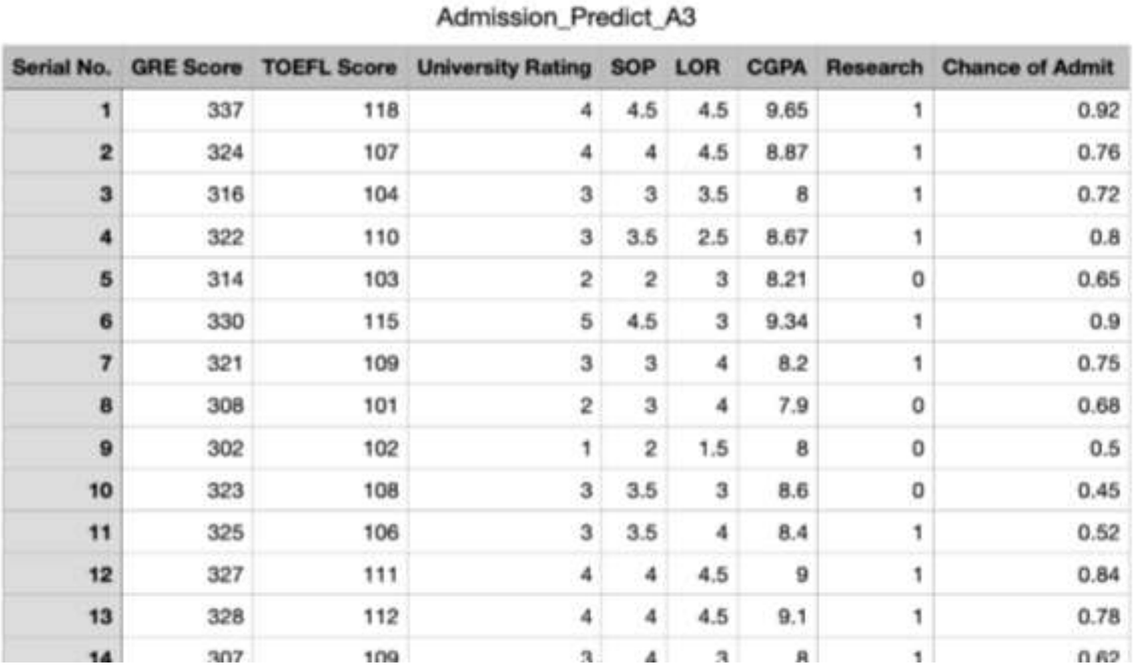
Assignment No: 3, Date Given:6th Oct 2024, Date of Sub: 20th Oct 2024

Regression (Linear and Logistic) and LDA using TensorFlow

Maximum Marks: 12

Assignment Overview:

In this assignment, you will implement both linear and logistic regression models using various machine learning libraries such as TensorFlow, NumPy, Pandas, and Scikit-Learn. Additionally, you will perform Linear Discriminant Analysis (LDA) as a dimensionality reduction technique before applying linear regression. The attached dataset (downloaded from kaggle) contains information about university applicants, including GRE scores, TOEFL scores, university ratings, statements of purpose (SOP), letters of recommendation (LOR), cumulative grade point averages (CGPA), and research experience. Your objective is to build predictive models that estimate the "Chance of Admit" for each applicant, starting with LDA for feature extraction. You can observe few records from the dataset (as shown below) where the last column is the target variable.



Admission_Predict_A3

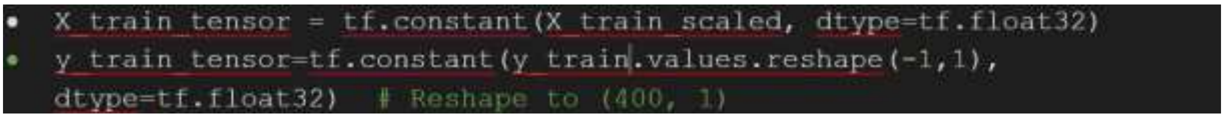
Serial No.	GRE Score	TOEFL Score	University Rating	SOP	LOR	CGPA	Research	Chance of Admit
1	337	118	4	4.5	4.5	9.65	1	0.92
2	324	107	4	4	4.5	8.87	1	0.76
3	316	104	3	3	3.5	8	1	0.72
4	322	110	3	3.5	2.5	8.67	1	0.8
5	314	103	2	2	3	8.21	0	0.65
6	330	115	5	4.5	3	9.34	1	0.9
7	321	109	3	3	4	8.2	1	0.75
8	308	101	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

Tasks to be Completed:

1. Data Loading and Preparation:
   - Import necessary libraries: OS, Pandas, NumPy, and TensorFlow.
   - Load the provided CSV file into a Pandas DataFrame.
   - Analyze the data and handle any missing values or outliers.
2. Data Scaling:
   - Select and apply an appropriate scaling technique (StandardScaler, MinMaxScaler, or RobustScaler) to normalize the features.
   - Analyze the distribution of the features and justify your choice of scaling method.
   - Discuss how scaling impacts model performance.
3. Dimensionality Reduction with LDA:
   - Discretize the "Chance of Admit": Convert the "Chance of Admit" variable into three categories (e.g., low, medium, high) using an appropriate binning technique.
   - Apply LDA: Use the categorized target variable to perform Linear Discriminant Analysis (LDA) on the input features. LDA will reduce the dimensionality of the feature space by projecting the data into a lower-dimensional space that maximizes class separability.
   - Justify the choice of number of components retained after LDA, and evaluate how this dimensionality reduction influences the dataset.
4. Linear Regression on LDA Transformed Data:
   - Use the transformed features from LDA as input for linear regression to predict the original continuous "Chance of Admit" values.
   - Compare the performance of the linear regression model using two different approaches:

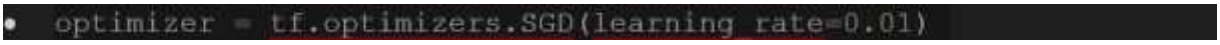
(1) TensorFlow:

- Convert the LDA-transformed data into TensorFlow tensors.



```
• 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 biases.
- Define the linear regression model and a loss function (Mean Squared Error).
- Use Stochastic Gradient Descent (SGD) optimizer with a learning rate of 0.01 and train for 1000 epochs.



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

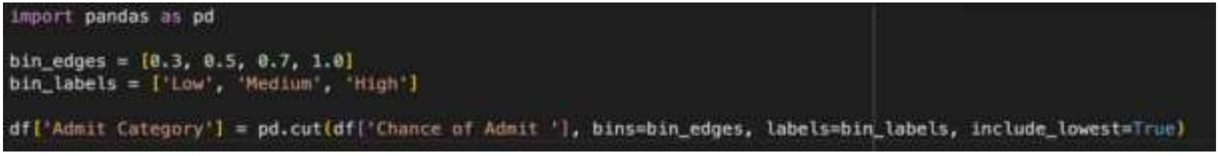
- Visualize the actual vs. predicted labels.

(2) Scikit-Learn:

- Use the LinearRegression class from Scikit-Learn to create and train the model.
- Visualize and compare the performance.

5. Logistic Regression:

- Convert the "Chance of Admit" into three categories (low, medium, and high) using an appropriate binning technique.



```
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)
```

- Implement logistic regression to predict the categorized admission chances using TensorFlow:
- Build the model using TensorFlow and use the Softmax activation function for multi-class classification.
- Tune the learning rate to optimize performance, evaluating the impact of different values.
- Compare the performance of logistic regression with other models like Random Forest (implemented in the previous Assignment), focusing on metrics such as accuracy, precision, recall, F1-score, and AUC-ROC.

6. Hyperparameter Tuning:

- Explore different learning rates for the models and assess their impact on convergence speed, accuracy, and loss.
- Use techniques like grid search or random search for tuning other hyperparameters (e.g., regularization strength).

7. Model Evaluation:

- Use k-fold cross-validation to evaluate the generalization of your models.
- Report model performance metrics including accuracy, precision, recall, F1-score, and AUC-ROC.
- Discuss areas of improvement and any potential biases in the models.

Additional Challenges (Optional):

- Assess how LDA's dimensionality reduction improves the interpretability and performance of linear regression.
- Experiment with techniques such as batch gradient descent and mini-batch gradient descent**,** and measure how they affect the training time and convergence speed compared to full-batch Stochastic Gradient Descent.

Notes:

- Make sure to install the Keras library before starting.
- Visualize key metrics like learning curves, model performance, and feature importance where relevant.

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 20th October 2024 midnight.

Note: Any clarification on this coding assignment may be emailed to I/C (C. R. Hota) or f20210564@hyderabad.bits-pilani.ac.in (Sai Charan).

References:

1. https://www.kaggle.com/code/arifali77/linear-and-logistic-regression
2. https://www.geeksforgeeks.org/ml-logistic-regression-using-tensorflow/
3. https://scikit-learn.org/dev/modules/generated/sklearn.discriminant_analysis.LinearDiscriminantAnalysis.html

----