CSOC-IG-WEEK-1

NAVKAR JAIN

June 2025

Part-1 :- Data Processing And Cleaning For Neural Network

- The AppointmentDay and ScheduledDay can be used in many ways as good feature for No-Show like days between, booked day , time etc. But the most suited was daysbetween therefore i convert datatypes of both AppointmentDay and ScheduledDay to datetime datatype using pd.to_datetime function. Then i take difference between them and to remove time i convert that to day using .days (for converting in days between them in integer)

- After the above changes i drop all the unwanted features :-

  'AppointmentDay' and 'ScheduledDay' As using feature engineering we convert them into important feature named as DaysBetween therefore now the both features are of no use

  'Patient Id' and 'Appointment Id' As ID as no use in No-Show guess but one thing in noticed that many patient ID are repeated which can be useful but less number there so i did not use that but can be helpful

- Then i started string encoding in which i convert :-

  'Gender' F to 1 and M to 2

  'Neighbourhood' Their are 81 neighbourhood which can be converted in 81 different column which best way but increases the input by 81 so i used another way in which i convert the 81 neighbourhood in numbers from 1 to 81 but in particular way that more count of neighbourhood get highest number and in same way as values have i great importance.

  'No-Show' No to 0 and Yes to 1

- After that to remove risk i started converting all columns datatype to int64.

- Sometimes data is given in arrange way on basis of some feature due to which when we divide data in train , validation and test set it may be possible that some value of that feature is not present in train data so algorithm will not learn but present in validation or test so program will not able to predict therefore i shuffle the data.
- After this i convert the data in train , val and test set .

Part-2 :- Neural Network For No-Show Dataset Using Core Python , Pandas and Numpy

- Firstly i created a neural network which contain 3 layer with 10 inputs , 2 Hidden Layer with 64 and 32 neurons with Relu function, 1 output layer with sigmoid function.
- Then i initialize the weights and bases which i declare as global so not need to pass again and again using Hi initialization as Using Relu Function
- Then i construct the function for calculating the accuracy, F1-Score, PR-AUC and confusion matrix
- Then i construct binary cross entropy loss function using logits concept for numerical stability and sigmoid function which is also numerically stable
- Then i construct the feed forward and 3 function for back propogation for calculating gradients and at last train function
- Then i train the model which take roughly 3min 12 sec as 2000 epochs are there and at last calculate the accuracy, F1-Score , PR-AUC and confusion matrix
  - Accuracy:- Train=0.6740 Validation=0.6722 Test=0.6663
  - F1-Score:- Validation=0.38865 Test=0.412738
  - PR-AUC:- Validation=0.2661 Test=0.2900

Part-3 :- Pytorch Implementation of Neural Network

- Firstly i apply Z-score Normalization on the dataset
- Then convert the data in tensor
- Then create the tensor dataset for batch normalization , then split the data in train, validation and test set and at last convert in form batches
- Then create the class of model with initialization of weights using Hi and Xavier Initialization for 2 hidden layer and output layer respectively

- Use Adam optimizer for better results with learning rate of 0.0001
- Then we train the model which take 3min 12sec for 200 epochs
- Calculate the Accuracy , F1-Score , PR-AUC and Confusion Matrix
  - Accuracy:- Validation=0.6556 Test=0.6492
  - F1-Score:- Validation=0.4348 Test=0.4377
  - PR-AUC:- Validation=0.3506 Test=0.3606