1. Abstract:

This project is a approach to solving the IEE-CIS-Fraud-detection binary classification problem in Kaggle competition using **deep neural network and Random forrest** identify a fraud or not**.** The link to the competition is given below: https://www.kaggle.com/c/ieee-fraud-detection/overview. The data is separated into train transaction and test transaction , train and test identity with a common merger on Transaction Id.

2. Methodology:

a) Feature Engineering1(for NN):

- First the train dataset is uploaded into local variable using the panda dataframe
- A left join is performed based on the Transaction id
- Some isFraud class has nan values that wont help so it those rows are dropped
- Columns that have more than 75% Nan values are being removed as they wont give details to the model for fitting and may overfit the model. Also it is used to constrict the size
- Some column names don't match the train so they are made same
- isFraud and the translation column is removed from the train. Is fraud is used as label and transaction id does not affect output.Same transactionid is removed from test dataset.
- Categorical column and numeric columns are being sorted form the final dataframe
- A pipeline is declared using the simpleImputer is used to fill constant value in numerical columns missing data and StandardScaler is used to normalize the data as Neural network deal with scaled values
- Column values are transformed using imputer of Na string to fill missing values and OneHotEncoder to transform the categorical columns
- There is an unbalance of 0s and 1s in the identification column so StratifiedKfold is used with 75% separation between test and valid. StratifiedKfold equally distributes classes between each fold.
- The preprocessor pipeline is fitted on the training and transforms in the validation and test

b) Feature Engineering2(for Random Forest):

- Steps 1-5 are same for both the models feature engineering
- Here new card and address columns that were assumed to be numerical taken as categorical columns
- Nan values in categorical are filled with string NA
- Using simple imputer all the columns in numerical are filled with mean

- New categorical columns are created such as amalgamation of all cards, M values, addr and finally amalgamation of productcd, allcard, allM
- Group statistic column based on categorical columns are being created. Such as average transactionAmt ,dist1 and sum amount and sum dist
- Frequency of the categorical columns are being created
- Correlation between numeric columns are checked and if both test and train has correlation >0.95 the column pair are selected and a new column with their sum are formed
- Based on randomforestclassifer feature selection those feature with importance > 0.002 are being selected

c) Deep NN model deployment:

- A DNN model is chosen with three Dense Layers 256,256,128 input sequentially
- Sandwiched between the layers are Dropout layer with 0.3 percentage dropout and batch normalization
- Batchnormalization normally distributed the input between 0-1 range that speeds up computation
- Dropout tries to generatize the data so that overfitting is prevented
- Adaptive adam optimizer is used that tunes learning gradient
- Epochs of 200 is set initially
- Callback of early stopping is used that waits 5 epochs until 0.01 decrease in loss in validation set is found to prevent overfitting
- The output is recorded in history variable
- Model is predicted using predict_proba in test dataset and transaction id and output is concatenated and submitted.

d) Deployment of Randomforrest/Classifier wt Hypttune

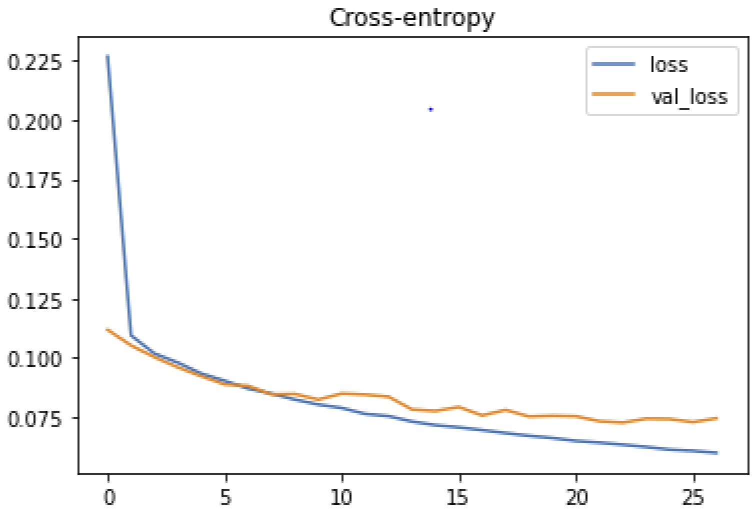
- A randomized search from sklearn is used to tune the hyperparameters of randomForrest classifier with cross validation of 3 and 4 iterations
- The reason for using randomized search over others is to mitigate the time complexity

e) Ensemble Random Forest and DNN:

- It is seen that the RF from one angle seems to perform better so a weighted ensemble of both of this model is done with ration 2:1.

3. Result Analysis:

- Aprivate score of 0.80 and public score of 0.859 is obtained for the deep neural network
- A private score of 0.881025 and 0.901407 public score is obtained on kaggle for the RF model
- The auc score from kaggle submission on the ensemble is public score:0.913273 and private score:0.887778
- It seem that the ensemble model seems to have performed better, both viewing the model from different angle



-
- As we can see from the above graph the train loss and val loss for the DNN go in tandem without val loss rising . This shows overfitting does not to occur