CS 583 Deep Learning
Course Project
Shalin Barot

Santander Customer Transaction Prediction

- ## Problem Description

In this challenge, we have to help to identify which customers will make a specific transaction in the future, irrespective of the amount of money transacted. The data provided for this competition has the same structure as the real data we have available to solve this problem.

- ## Dataset Overview

In **train.csv** there are total 200k records and 202 columns out of which 1st column is ID_code which is ID of transaction and target is dependent variable which is either 0 or 1.
In **test.csv** there are 200k records and 201 columns where we don't have target variable and we have to predict for that value for that and submit in **submission.csv**. We don't have any missing value in any of the record of train and test dataset.

- ## Sample Distribution

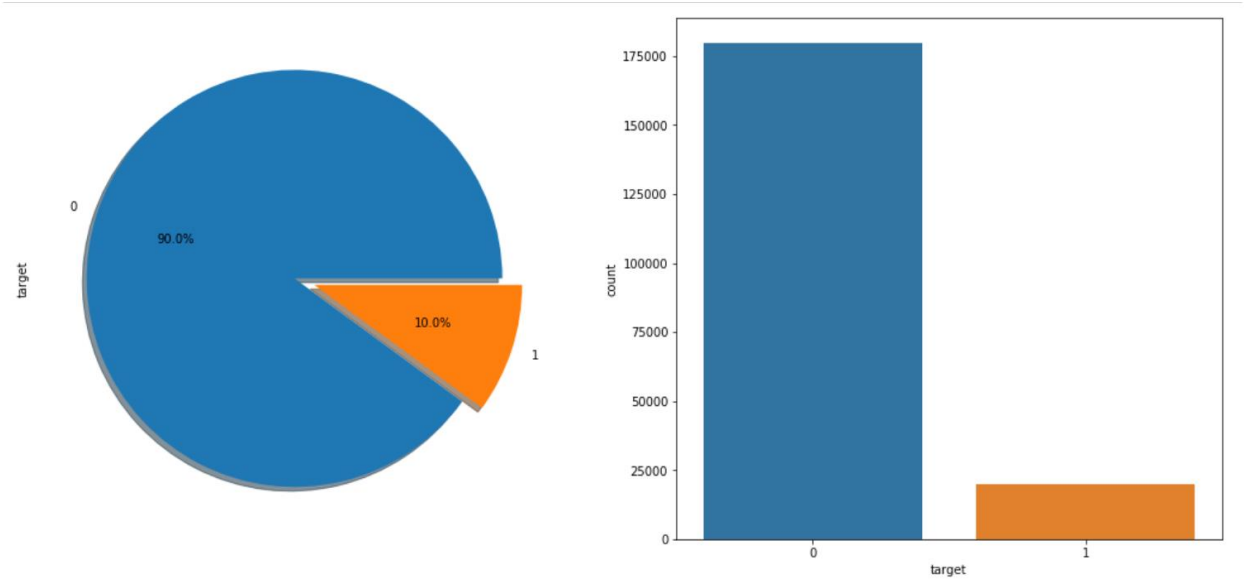

Figure 1

In the Figure 1 we can see that our data is highly imbalance, target 0 is 90% of the record and target 1 is 10% of the total record.

- # Models

I tried 3 different models for the given problem:

1) Logistic Regression

2) Neural Network

3) Convolution Neural Network

1) Logistic Regression

Even Though Logistic Regression is simple model it gave best CV score of **0.86** at the end which was pretty impressive. Here we used 10 fold CV and for logistic regression with max_iteration of 2000 though it was not able of converge for some fold but due to system limitation I kept at little low.

2) **Neural Network**

Next I tried Neural Network with 2 Dense Layer and 1 Flatten layer using keras, whose architecture is as shown in Figure 2. Which gave the best AUC score of **0.899**

```
nn_model().summary()
```

```
Model: "model_2"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_2 (InputLayer)         (None, 200, 1)            0
_________________________________________________________________
dense_3 (Dense)              (None, 200, 16)           32
_________________________________________________________________
flatten_2 (Flatten)          (None, 3200)              0
_________________________________________________________________
dense_4 (Dense)              (None, 1)                 3201
=================================================================
Total params: 3,233
Trainable params: 3,233
Non-trainable params: 0
_________________________________________________________________
```

Figure 2

3) Convolution Neural Network

Convolution neural network with one Conv layer, Dense Layer and Flatten Layer with output layer as Sigmoid performed the best. Architecture of the model can be seen in Figure 3. I used Adam optimizer. The loss function is categorical binary-cross entropy. Here we use the CLR with Kenel-size =2, strides=2, sigmoid activation. Then 600 lters with Batch size of 512 and epochs of 5. Where we got the best AUC score of **0.917**

```
get_model().summary()
```

```
Model: "sequential_2"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv1d_1 (Conv1D)            (None, 200, 600)          1800
_________________________________________________________________
flatten_9 (Flatten)          (None, 120000)            0
_________________________________________________________________
dense_17 (Dense)             (None, 1)                 120001
=================================================================
Total params: 121,801
Trainable params: 121,801
Non-trainable params: 0
_________________________________________________________________
```

Figure 3

Tricks used in CNN:

Based on a few discussion threads it was found that the test data had some fake rows in it based on the frequency and similarity of the datapoints, so we'll remove these. Then based on other thread, it was seen that all the parameters were independent of each other and if we add extra parameters by frequency encoding each of the 200 variables, this gave a large boost in the score. Also to ensure the CNN which is very good at finding correlations between variables, we try to augment the data and feed it to the network in such a way that it always sees a different subset of the data.

- **Result**

I participated in Inactive competition with late submission allowed. Where I got score of **0.92174** in private leaderboard and **0.92218** in public leaderboard which from the leaderboard comes at top **10%**. Which can be seen in Figure 4.

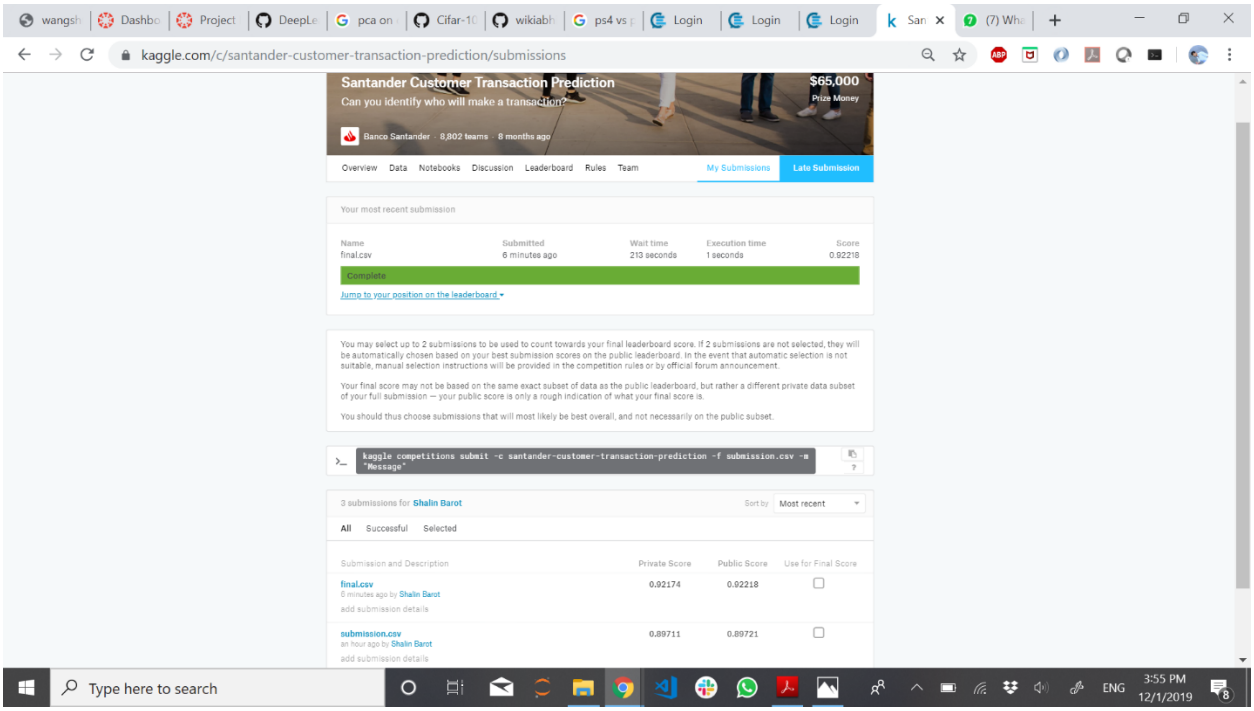

Figure 4