CS583A: Course Project

Vardaan Kishore

May 20, 2019

1 Summary

[Problem descriptions:] I have participated in a inactive competition with late submission, the project is to identify if or not a customer will make a transaction or no. [Implementation:] Here I have compared and contrasted various methods:

- Logistic Regression
- Naive Bayes
- Simple Feed Forward neural network
- CNN

[Evaluation metric:] Submissions are evaluated on area under the ROC curve between the predicted probability and the observed target. [Score and ranking:] Have attached screenshots of the code, there was no ranking as this was a late submission

2 Problem Description

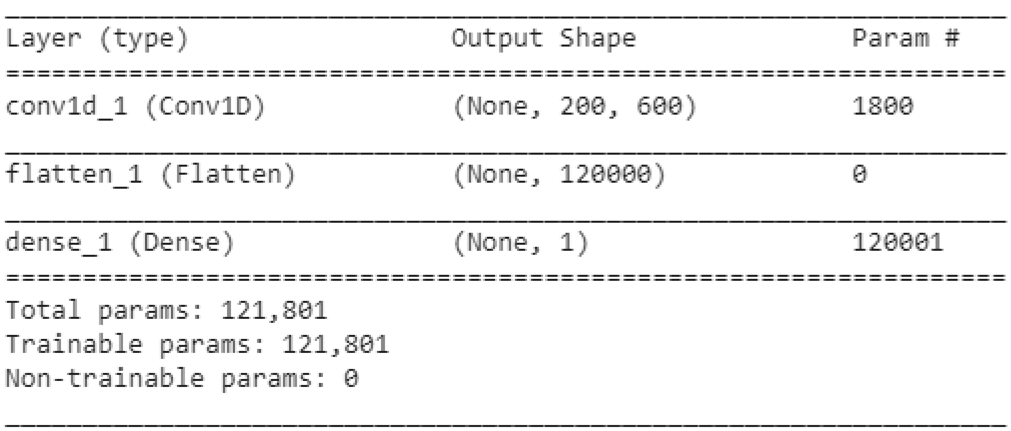
Problem. The problem is to classify if a customer will make a transaction or not based on 200 features and a binary targer,it is a binary classification problem. This is an important factor for banks to help them promote and make understand why customers do not make transactions The competition is at `https://www.kaggle.com/c/santander-customer-transaction-prediction/overview/evaluation`.

Data. The data are 200k x 202 training samples and the test is 200k x 201 . The number of classes is 2. The training set is not well-balanced: $n_1 = 10.049 percent$ and $n_0 = 89.951 percent$.

Challenges. There are quiet a few challengens in this dataset, first of all we need to identify the most important parameters and the correlations between these, there are quiet a few things that I understood that I will talk about later, the dataset is imbalanced.

3 Solution

Model. The model we finally choose is a CNN with quiet a few modifications



```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv1d_1 (Conv1D)            (None, 200, 600)          1800
_________________________________________________________________
flatten_1 (Flatten)          (None, 120000)            0
_________________________________________________________________
dense_1 (Dense)              (None, 1)                 120001
=================================================================
Total params: 121,801
Trainable params: 121,801
Non-trainable params: 0
_________________________________________________________________
```

Figure 1: CNN model

Implementation. We implement the CNN
We run the model using keras with tensorflow as backend, since I had tested 3 other models, I had to run this one on the cloud using the colab gpu by google. The other models were run locally on Razer blade 15 with rtx 2060 and i7 and 16 gb memory. The cnn took 1.5 hours to train.

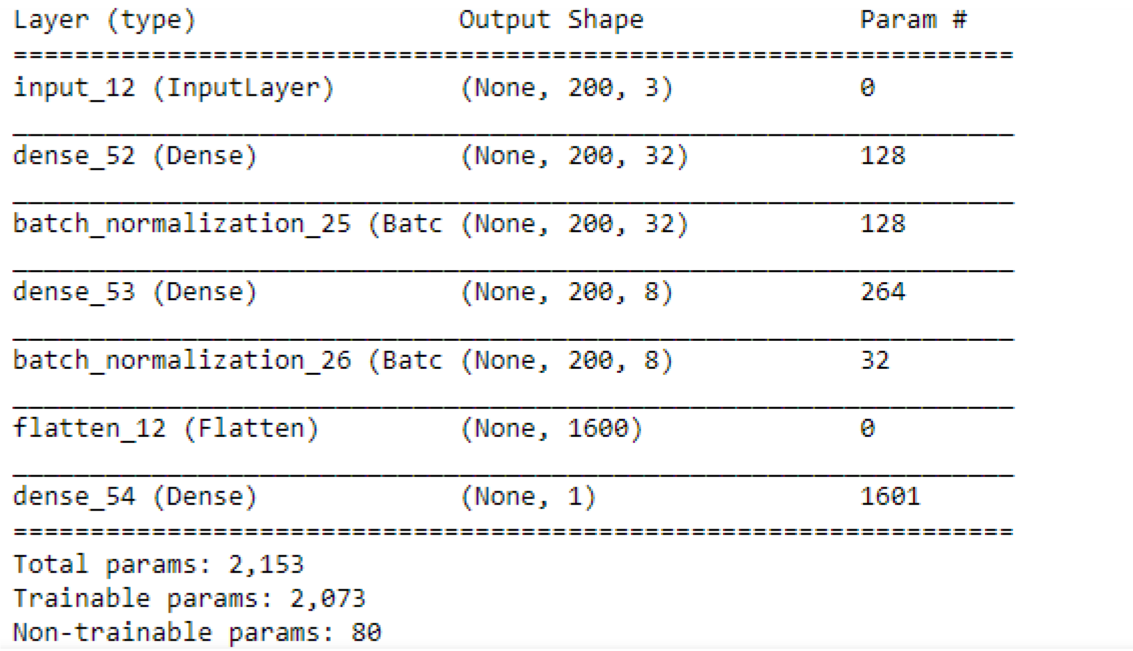
Settings. The loss function is categorical binary-cross entropy. The opitmizer is Adam. Here we use the CLR(Cyclic learning rate provided by the keras contrib package,it has a good range of lr's)
Kenel-size =2, strides=2, sigmoid activation
The kernel adn the stidres are two to help us in picking the new encoded feature and the original together. We also use600 filters Batch size=512
epochs =5
There can be a very good score given the resources to train, this was trained on very constrained conditions of memory and gpu.

Advanced tricks. Here based on a few discussion threads it was found that the test data had fake rows in it based on the frequency and similarity of the datapoints,so removed these, then based on another discussion thread, it was seen that all the parameters were independent of each other and if we added 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.

Cross-validation. The parameters were tuned using a 5 fold cross-validation

4 Compared Methods

Logistic Regression. I have used the oldest model in the book for classifcation, use 10 fold CV with with lbfgs with 1500 iterations, the best AUC score at the end of the cross validation was 0.85



```
Layer (type)                 Output Shape              Param #
=================================================================
input_12 (InputLayer)        (None, 200, 3)            0
_________________________________________________________________
dense_52 (Dense)             (None, 200, 32)           128
_________________________________________________________________
batch_normalization_25 (Batc (None, 200, 32)           128
_________________________________________________________________
dense_53 (Dense)             (None, 200, 8)            264
_________________________________________________________________
batch_normalization_26 (Batc (None, 200, 8)            32
_________________________________________________________________
flatten_12 (Flatten)         (None, 1600)              0
_________________________________________________________________
dense_54 (Dense)             (None, 1)                 1601
=================================================================
Total params: 2,153
Trainable params: 2,073
Non-trainable params: 80
```

Figure 2:

Naive Bayes. This is also a classic model in classification tasks, by using the GaussianNb from sklearn and fitting it to the data the AUC score was 0.89

Feed Forward neural network We use the keras for building and training this neural network. Below is the architecture of the data. Even here we remove the fake data and add extra features. The auc score from this was 0.90

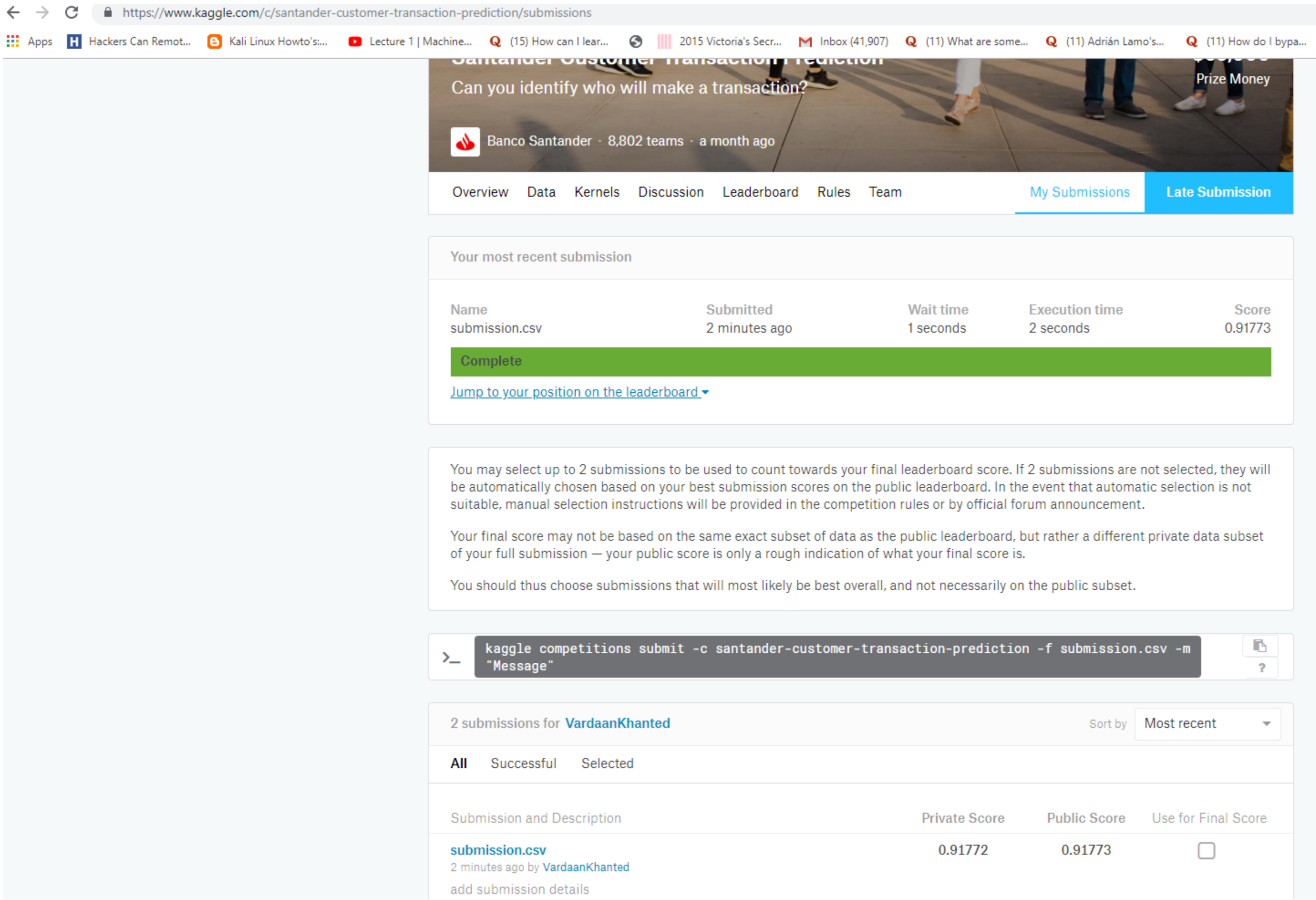
Advanced tricks. We finally adopted the model as shown in figure 3 We applied the following tricks.

- Data augmentation. This helps maintain the independence of features and forces the CNN to look at each feature independently
- Adding extra features based on discussion threads, we encode the frequency to the data this helps create additional 200 columns.

The final score was 0.91

5 Outcome

Participated in a inactive with late submission but going by the scores it can be seen that the ranking would be in the top 20 percent.Again given the constraints of the memory and the gpu we could do a lot better on the CNN n Figure 3. There html files uploaded have some manual stops given the memory issues and the gpu issues I had to restart them, the score submitted is the best of them.



Can you identify who will make a transaction?

Banco Santander · 8,802 teams · a month ago

Your most recent submission

Name	Submitted	Wait time	Execution time	Score
submission.csv	2 minutes ago	1 seconds	2 seconds	0.91773

Complete

Jump to your position on the leaderboard ▾

You may select up to 2 submissions to be used to count towards your final leaderboard score. If 2 submissions are not selected, they will be automatically chosen based on your best submission scores on the public leaderboard. In the event that automatic selection is not suitable, manual selection instructions will be provided in the competition rules or by official forum announcement.

Your final score may not be based on the same exact subset of data as the public leaderboard, but rather a different private data subset of your full submission — your public score is only a rough indication of what your final score is.

You should thus choose submissions that will most likely be best overall, and not necessarily on the public subset.

```
kaggle competitions submit -c santander-customer-transaction-prediction -f submission.csv -m "Message"
```

2 submissions for VardaanKhanted

Sort by Most recent

All Successful Selected

Submission and Description	Private Score	Public Score	Use for Final Score
submission.csv 2 minutes ago by VardaanKhanted add submission details	0.91772	0.91773	□

Figure 3: Caption