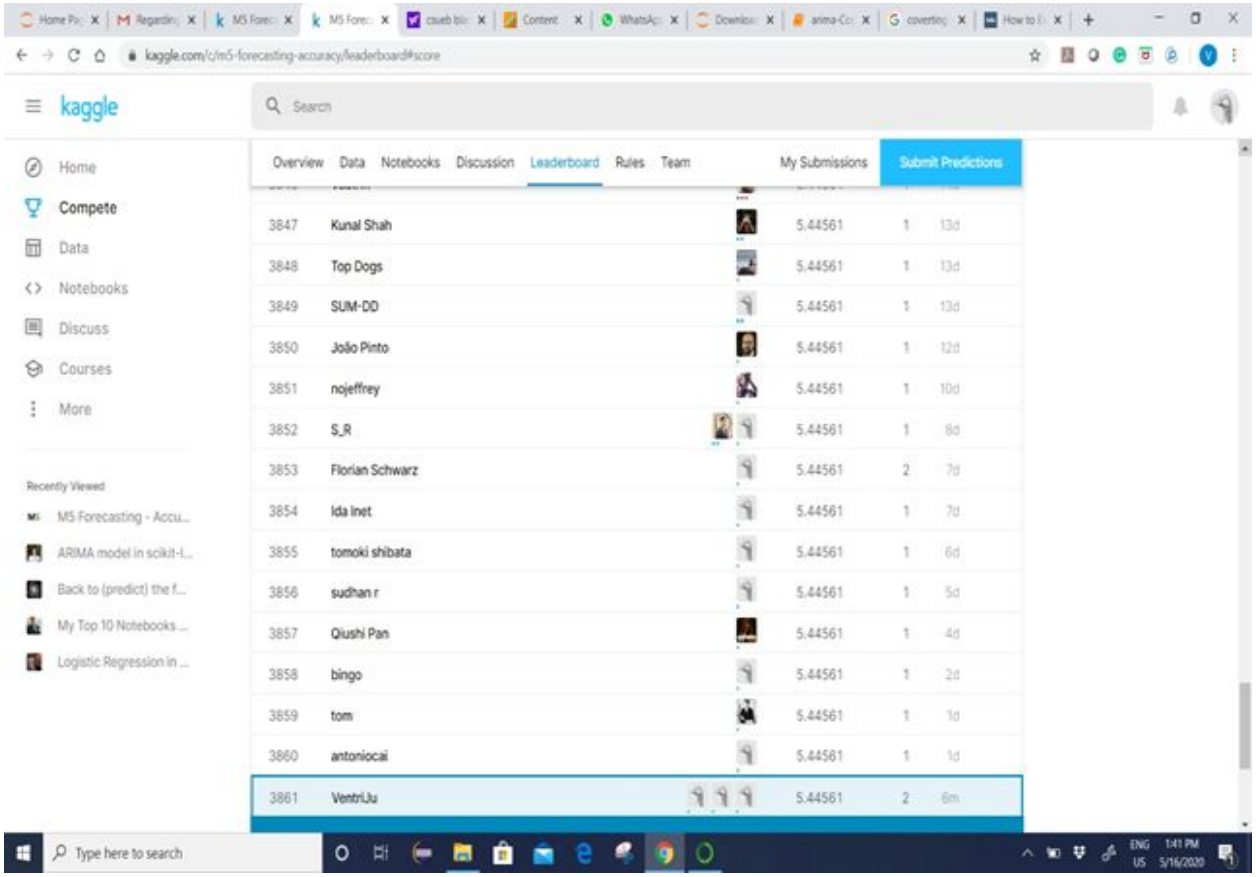
Kaggle Competition " M5 Forecasting - Accuracy"

Group Name: VentriJu

Members: Vennela Vysyaraju(vq8474), Tripti Prasad(py5697),Juhi Kushwah(tg5354)

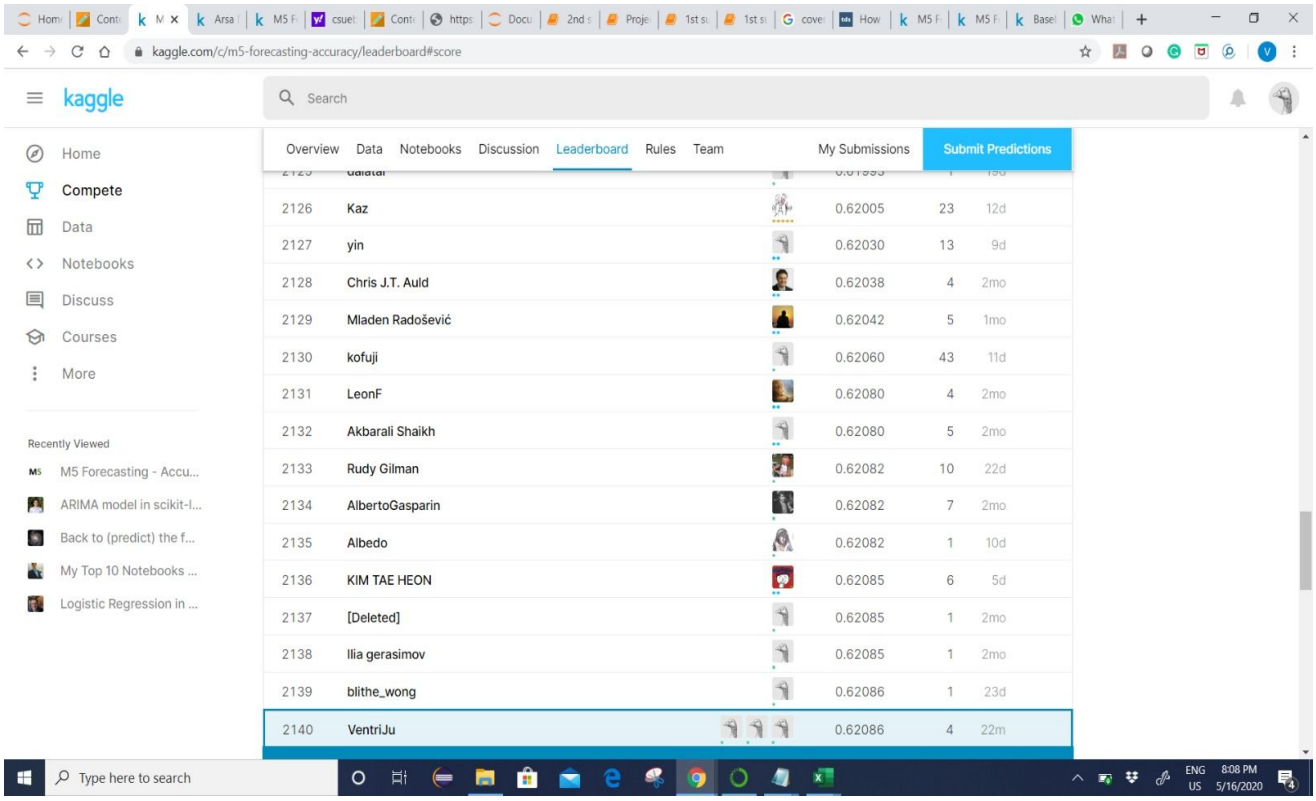
Submission1: Score - 5.44561

We used the dataset generated by the Professor for this submission.Have implemented the time series algorithm "Arima" for predicting the sales values for the next 28 days for 1 particular item id ID_HOUSEHOLD_1_001_TX_1. We submitted in Kaggle and received a score of 5.44561. Below is the screenshot of leader board position of our group:



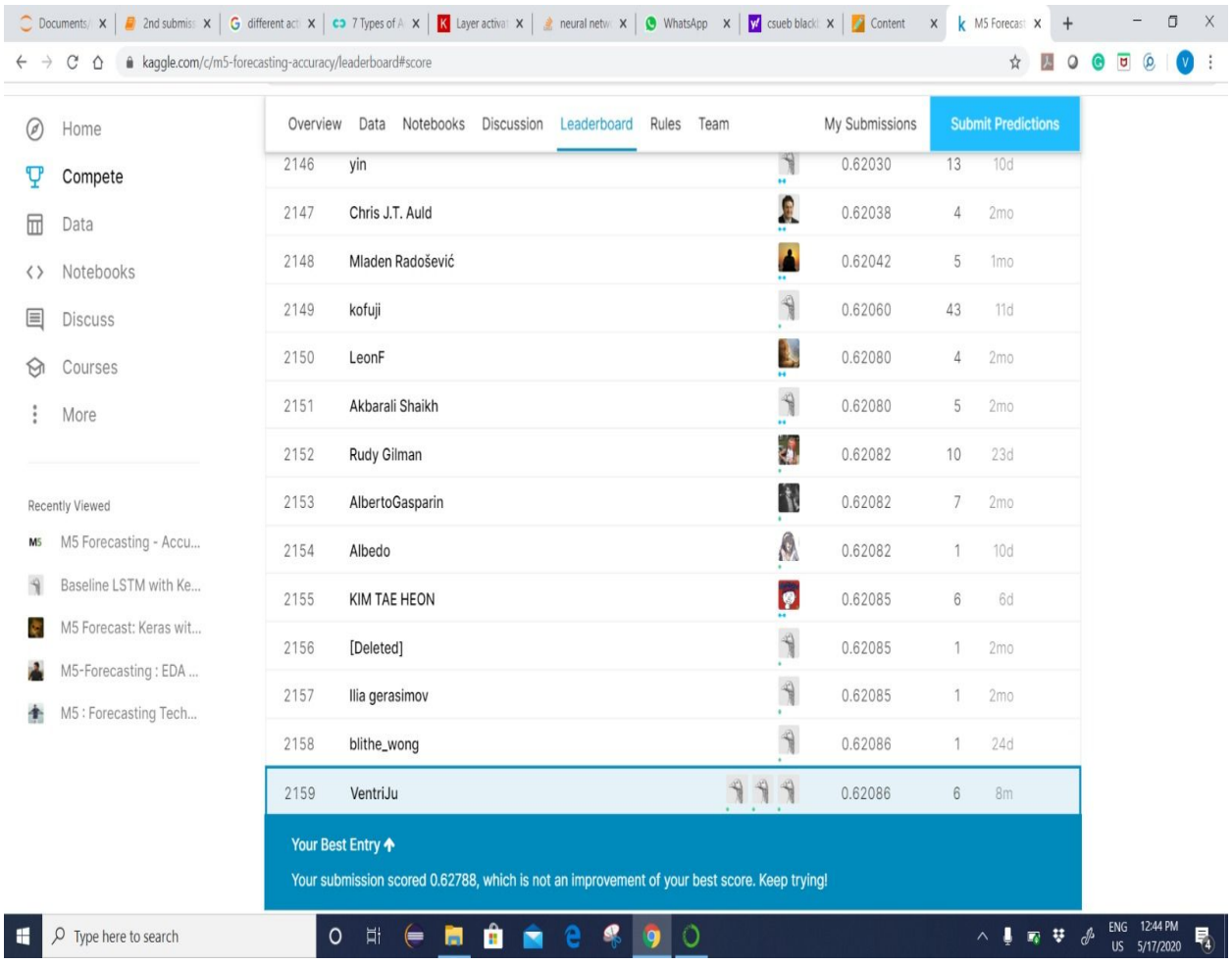
Submission2: Score- 0.62086

We used the original dataset of Walmart for this submission.Have forecasted values for all the items for the next 28 days using Keras categorical embedding V2 and imported tensorflow, regularizers libraries. Have used 3 hidden layers with activation function “tanh”. This method improved our score and got a score of 0.62086. Below is the leaderboard position of our group:



<u>Submission3</u>: Score-0.62788

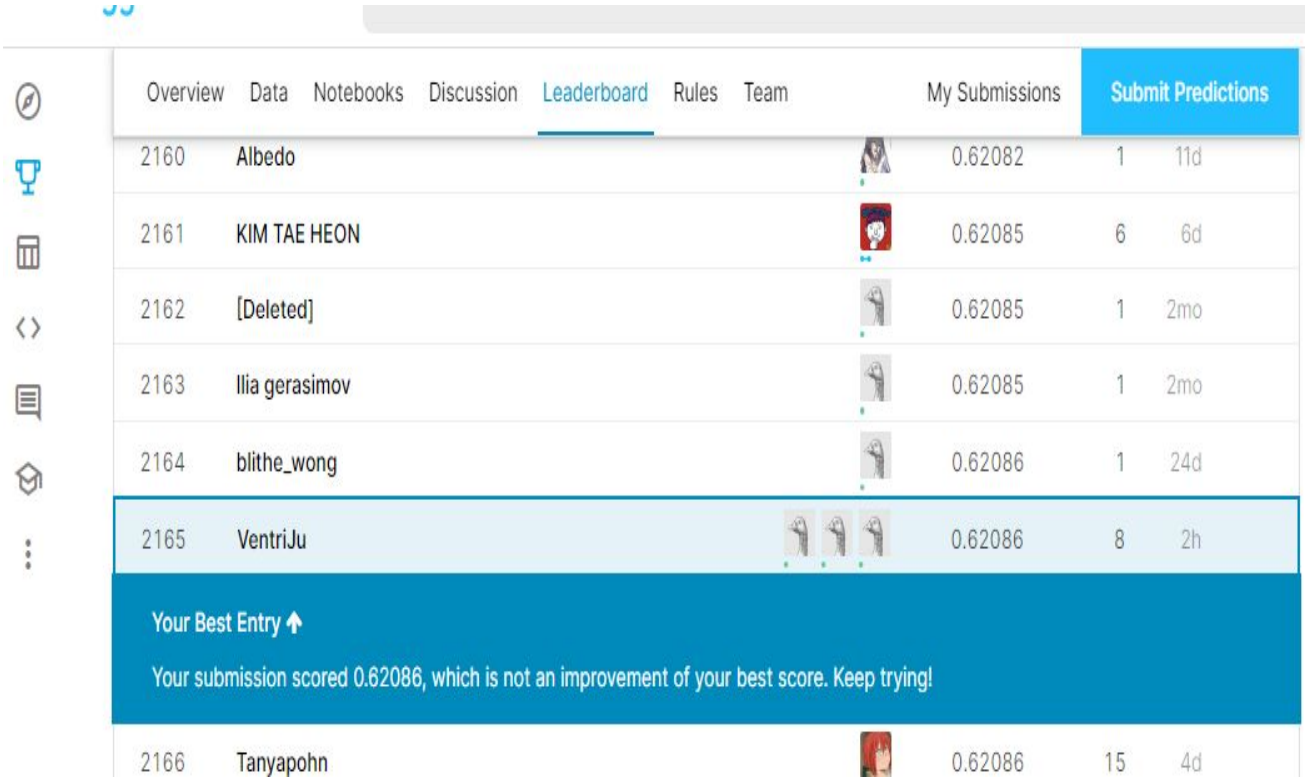
We used the original dataset of Walmart for this submission.Have implemented the same as submission2 which is using Keras categorical embedding V2 but with different activation functions like relu,selu and softplus for each and every different hidden layer and also given a different number of neurons for each hidden layer compared to submission2. This implementation gave us a score of 0.62788.This did not improve our leaderboard position and below is the screenshot of our score:



Submission4: Score- 0.62086

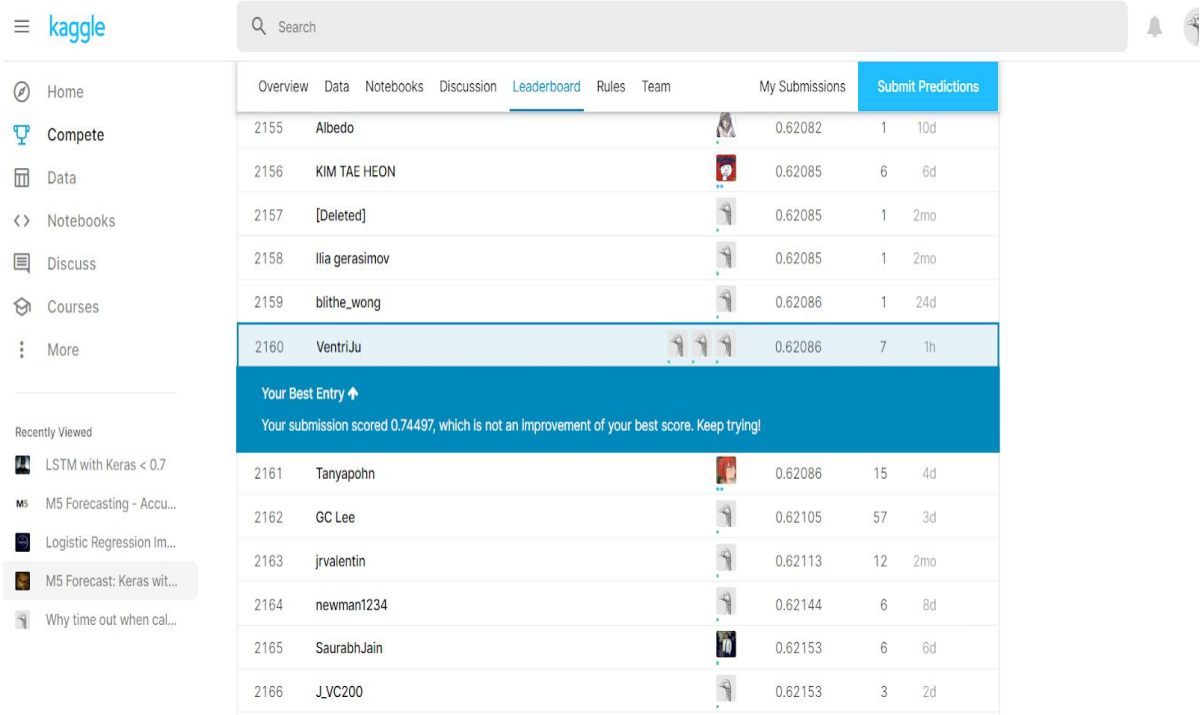
We used the Naive approach on the original dataset. One thing to observe here is that the score we got for Naive approach is somewhat equal to the score we got while using Keras model with tensorflow. My understanding from this observation is that memory size has a major role to play in analyzing the data in sales_train_validation.csv file. We had to create functions to manage the memory size for our dataset because without this function, the size created hindrance in training and storing values. To understand how this model is forecasting for individual values, we trained 10% of this data, however, instead of doing the cross-fold validation, we kept increasing the values to eventually forecast for 14, followed by 28 days, and while the average value remained unchanged, there was a significant difference in the prediction. This value ranged from [0.2-11.10]. The validation error comes out to be around 0.0883 or 8.83%.

Following is our best score on the leaderboard:-



Submission5: Score- 0.74497

In this approach we used the LSTM model to train past sales values for each 30490 item and a feature which represents if there is an event the following day. Sales of the past 14 days and event feature was used for predicting 15th day sales.Used sales_train_validation.csv file. We are using the last 14 days in order to predict day 1915 sales. In order to predict 1916th day, 13 days from our input data and 1 day from our prediction are used. After that we need to slide the window one by one.
12 days from input data + 2 days from our prediction to predict 1917th day
11 days from input data + 3 days from our prediction to predict 1918th day
....
14 days our prediction to predict the last 1941th day sales.
We submitted and got a score of 0.74497. We did not improve from previous submissions maybe because we used only the past days and one day before event to predict. There are a lot of zero values in the first year. Instead may be multi-step LSTM could have given better results.



Below is the screenshot of submissions of our group: