Deep Learning – Assignment 2 - Part 3 - Research Summary Report

Submitted by:

Itay Bouganim, 305278384
Ido Rom, 312239858

Overview

The dataset we worked on during the assignment was Kaggle's Corporación Favorita Grocery Sales Forecasting data set which consists of items and the amount of sales for each item. We are provided with a training set consisting of items and the amount of unit salings for each item, a test set of items and another training set which consists of more relevant data. For example, Store data that contains more details about the store, such as city, state, etc. The goal is to create a classifier capable of receiving date, store_id and item_id and predict how many units of the item will be sold in the store at the relevant date.

3.a)

We downloaded the data from keras, and analyzed it.

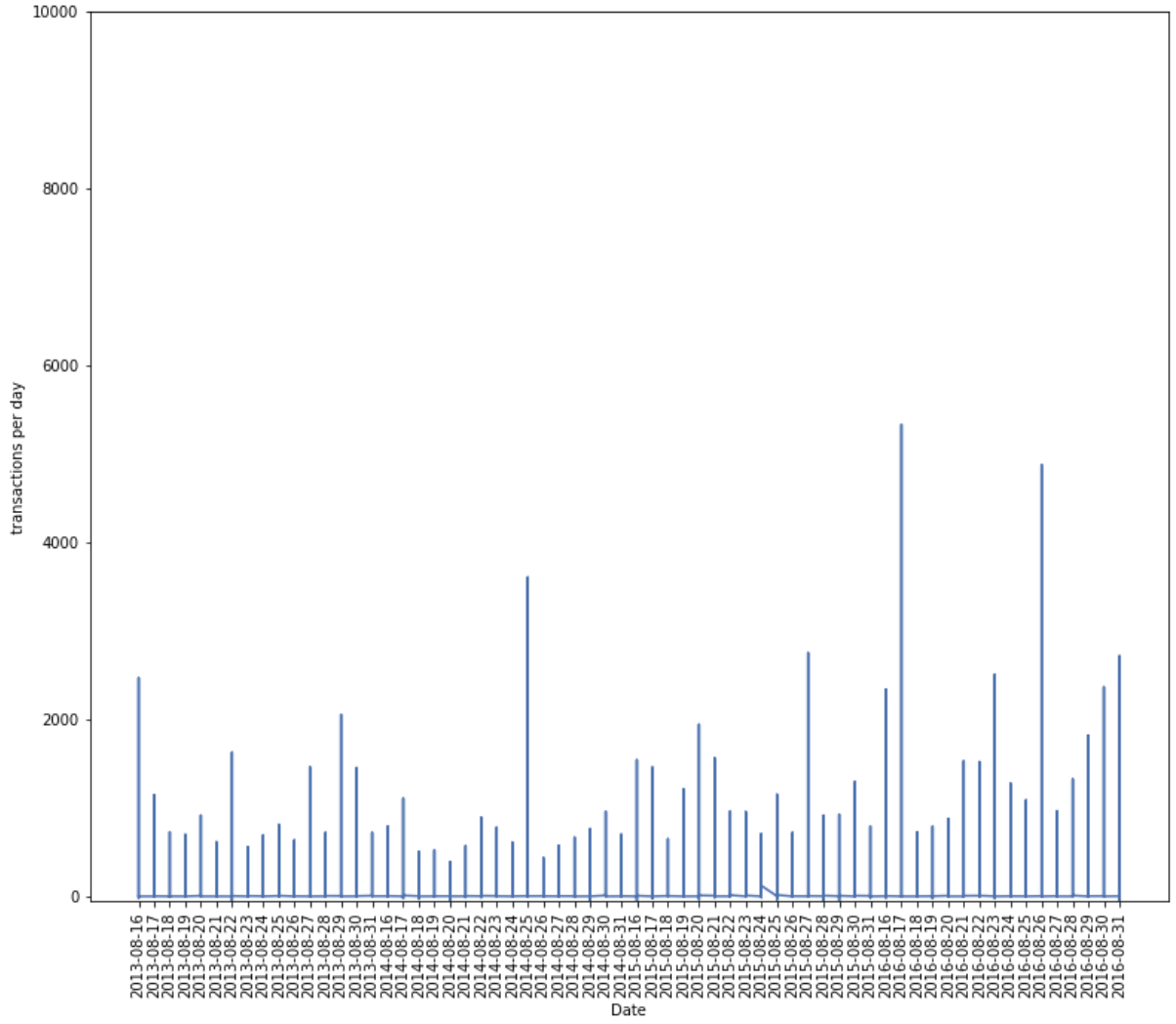
DATASET EXPLORATION & ANALYSIS

Train data and test data -

	id	date	store_nbr	item_nbr	unit_sales	onpromotion
0	0	2013-01-01	25	103665	7.0	None
1	1	2013-01-01	25	105574	1.0	None
2	2	2013-01-01	25	105575	2.0	None
3	3	2013-01-01	25	108079	1.0	None
4	4	2013-01-01	25	108701	1.0	None

Train data size - 125497040.
Test data size - 3370464.
Some points of train and data sets -

- We can see that the only data that is missing is onpromotion. So we wanted to replace the nan value by False or True in the probability they appear in the data, since we don't want null values.
- We realized in the test data analysis that all the test data is only between 08-16 - 08-31. For that reason, and because the train data is huge, we wanted to take only these dates from the train data.
- The train data size after the change - 4459012.

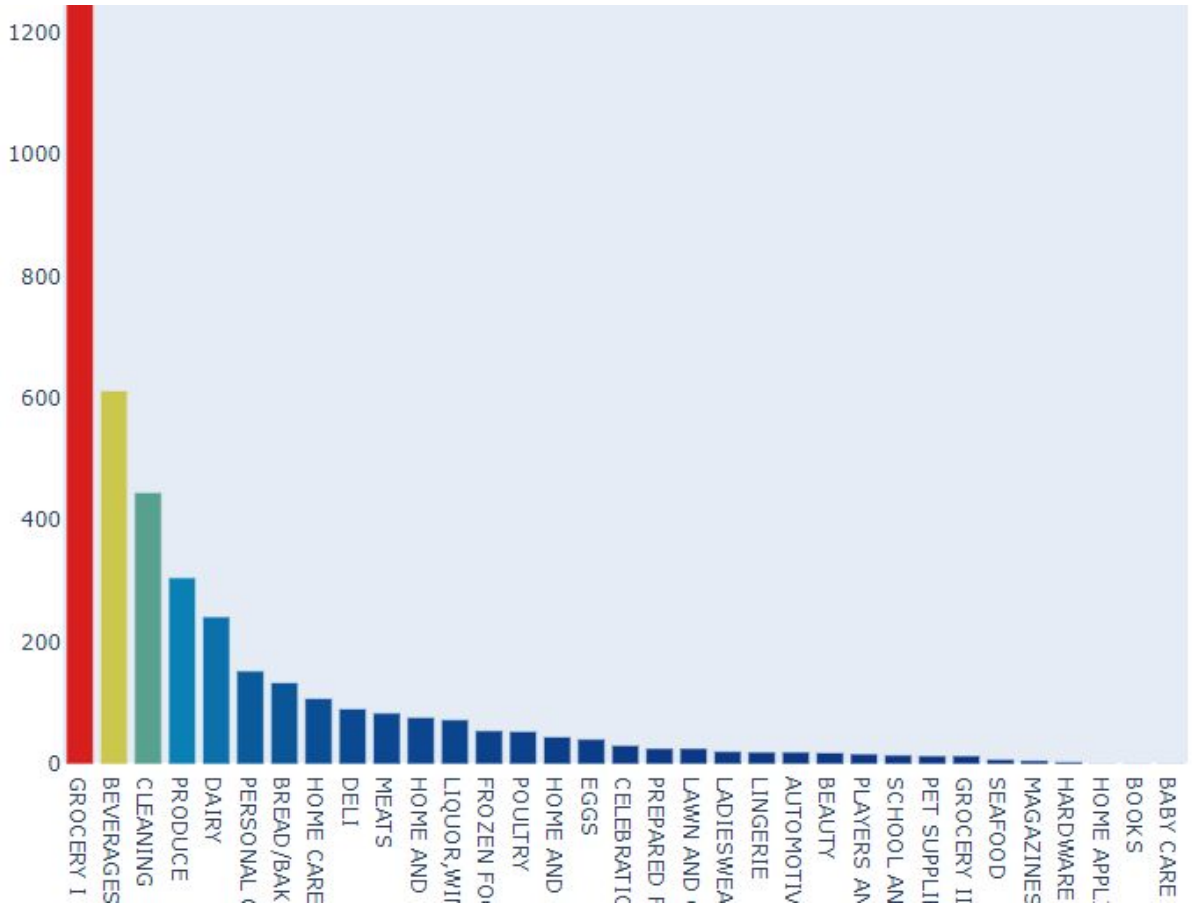


We can see that there are days that have extremely more transactions - maybe weekends or holidays.

Item data -

	item_nbr	family	class	perishable
0	96995	GROCERY I	1093	0
1	99197	GROCERY I	1067	0
2	103501	CLEANING	3008	0
3	103520	GROCERY I	1028	0
4	103665	BREAD/BAKERY	2712	1

Item data size - 4100.

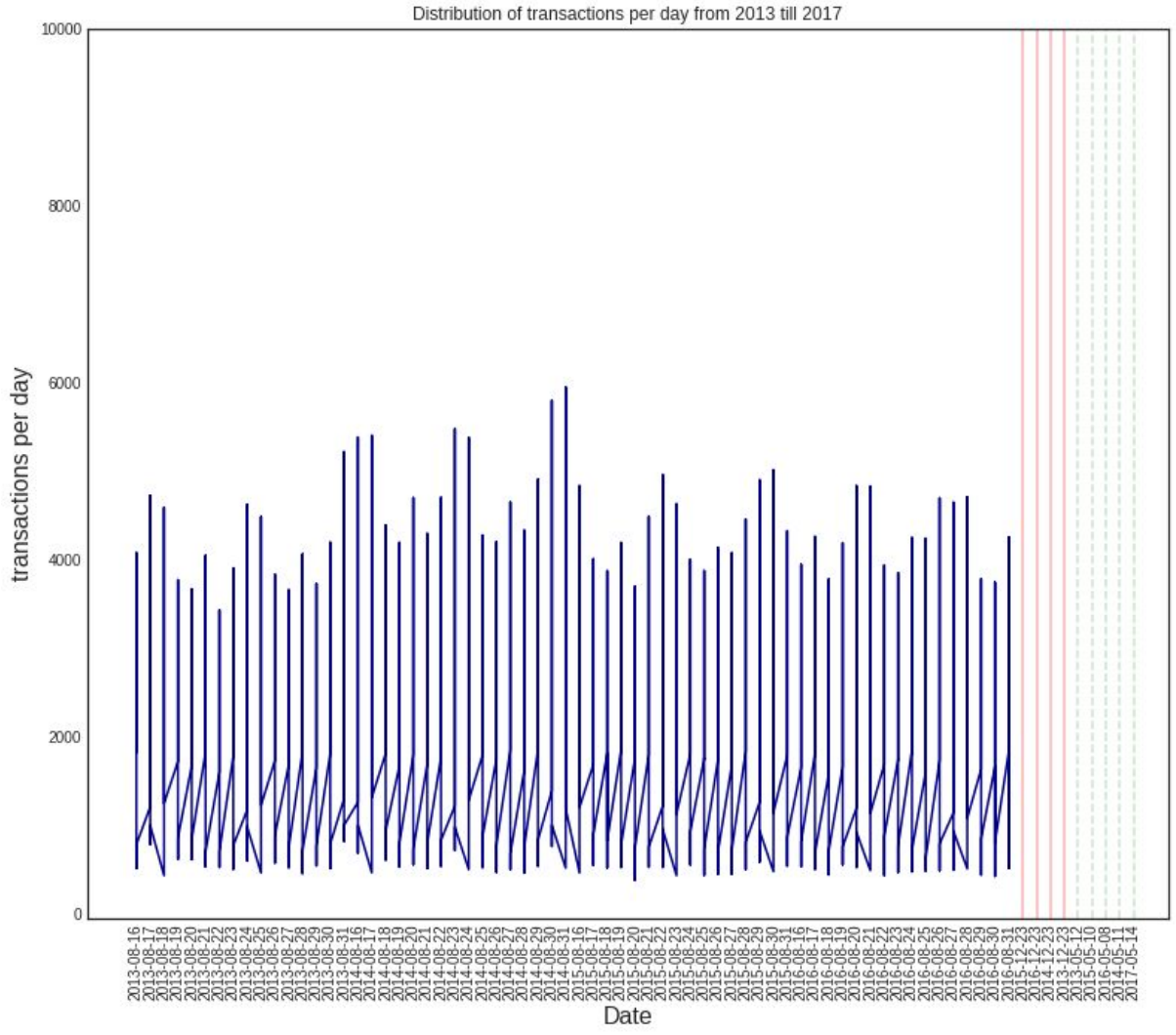


We can see that the items have different amounts for each family category.

Transactions data -

	date	store_nbr	transactions
0	2013-01-01	25	770
1	2013-01-02	1	2111
2	2013-01-02	2	2358
3	2013-01-02	3	3487
4	2013-01-02	4	1922

transactions data size - 83488.



- On that data we also took only transactions from 16-08.
- We can see that each day as different number of transactions. maybe weekends or holidays.

Holidays Events data -

We took only events between 16/08 - 31/08.

	date	type	locale	locale_name	description	transferred
16	2012-08-24	Holiday	Local	Ambato	Fundacion de Ambato	False
69	2013-08-24	Holiday	Local	Ambato	Fundacion de Ambato	False
132	2014-08-24	Holiday	Local	Ambato	Fundacion de Ambato	False
187	2015-08-24	Holiday	Local	Ambato	Fundacion de Ambato	False
271	2016-08-24	Holiday	Local	Ambato	Fundacion de Ambato	False
327	2017-08-24	Holiday	Local	Ambato	Fundacion de Ambato	False

Holidays events data size - 6.

We can see that there is only one holiday at that period of year.

Stores Events data -

	store_nbr	city	state	type	cluster
0	1	Quito	Pichincha	D	13
1	2	Quito	Pichincha	D	13
2	3	Quito	Pichincha	D	8
3	4	Quito	Pichincha	D	9
4	5	Santo Domingo	Santo Domingo de los Tsachilas	D	4

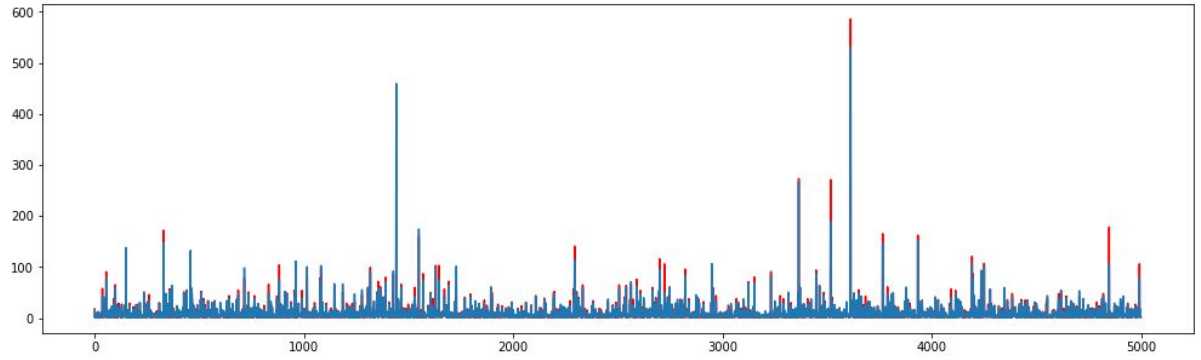
Stores data size - 54.

We decided not to use the oil data, since there is a lot of data and we thought it would be too much information for the model.

b)

In our data set, we need to predict the number of sales of a specific product. Since it is a regression problem, we choose to use RandomForestRegressor as the solid bench mark.

We ran the random forest algorithm with all the merged data, and it received 1.4987 mean absolute error on the validation data. In addition we submitted the predictions of the test data to kaggle and received a score of - 1.45468.



- We can see that random forest receives a good benchmark result.
- We can see from the graph that random forest success to generalize the data and to predict pretty well the extreme values.

c)

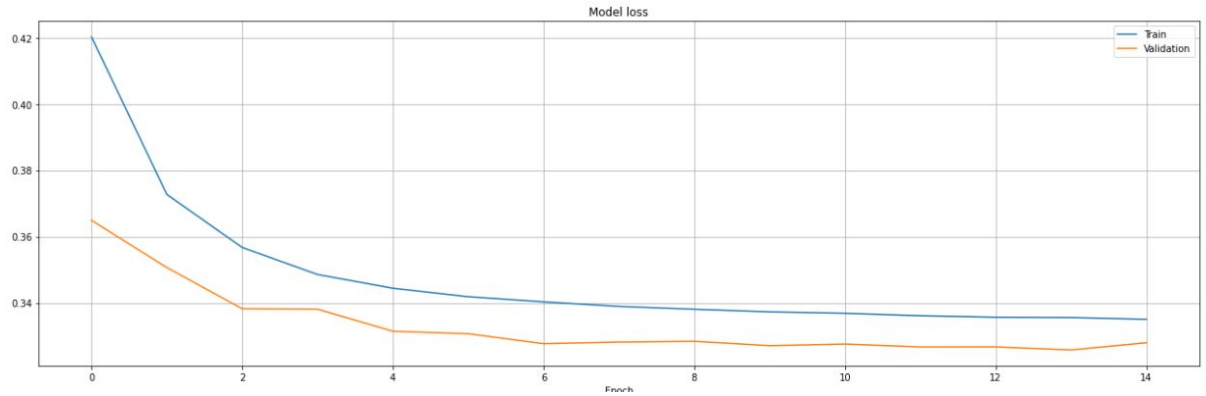
We merged the train data with all other data sets except the oil data. Then, we encoded that data in such a way we can use embeddings for it.

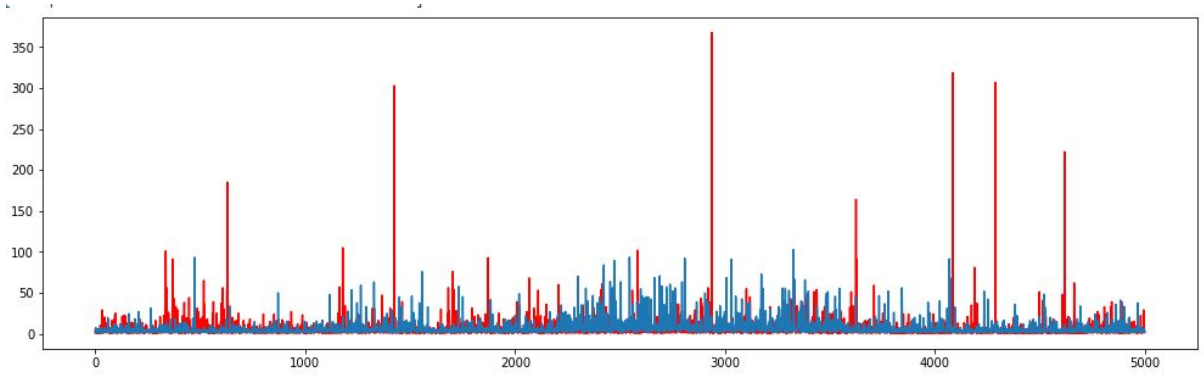
- We thought that all the other data is relevant, since it can add more information. For example if it is a holiday, there will be more purchases.
- We thought that use all the data for embeddings may cause too much information for the model, but we used all of it for the first models.

d)

At first step, we wanted a simple NN with embeddings containing only the two trivial features - store_nbr and item_nbr, to see how it works, and what the baseline results are.

```
Layer (type)                    Output Shape         Param #
Connected to
==================================================================================
==========================
input_33 (InputLayer)           [(None, 1)]          0
__________________________________________________________________________________
__________________________
input_34 (InputLayer)           [(None, 1)]          0
__________________________________________________________________________________
__________________________
embedding_32 (Embedding)        (None, 1, 5)         270
input_33[0][0]
__________________________________________________________________________________
__________________________
embedding_33 (Embedding)        (None, 1, 5)         20365
input_34[0][0]
__________________________________________________________________________________
__________________________
concatenate_3 (Concatenate)     (None, 1, 10)        0
embedding_32[0][0]

embedding_33[0][0]
__________________________________________________________________________________
__________________________
flatten_3 (Flatten)             (None, 10)           0
concatenate_3[0][0]
__________________________________________________________________________________
__________________________
batch_normalization_14 (BatchNo (None, 10)           40
flatten_3[0][0]
__________________________________________________________________________________
__________________________
dense_15 (Dense)                (None, 10)           110
batch_normalization_14[0][0]
__________________________________________________________________________________
__________________________
batch_normalization_15 (BatchNo (None, 10)           40
dense_15[0][0]
__________________________________________________________________________________
__________________________
dense_16 (Dense)                (None, 10)           110
batch_normalization_15[0][0]
__________________________________________________________________________________
__________________________
dense_17 (Dense)                (None, 1)            11
dense_16[0][0]
==================================================================================
==========================
Total params: 20,946
Trainable params: 20,906
Non-trainable params: 40
```





Loss on validation data - 0.32577

We can see that we receive much lower results than the benchmark.
we can try to improve it by:

- Add features from the other data sets with embeddings.
    - we can see that the algorithm fails to recognize the extreme values, that is why adding embeddings that contain holidays for example may help.
- Add another network to model the time series part of the model, such as LSTM.

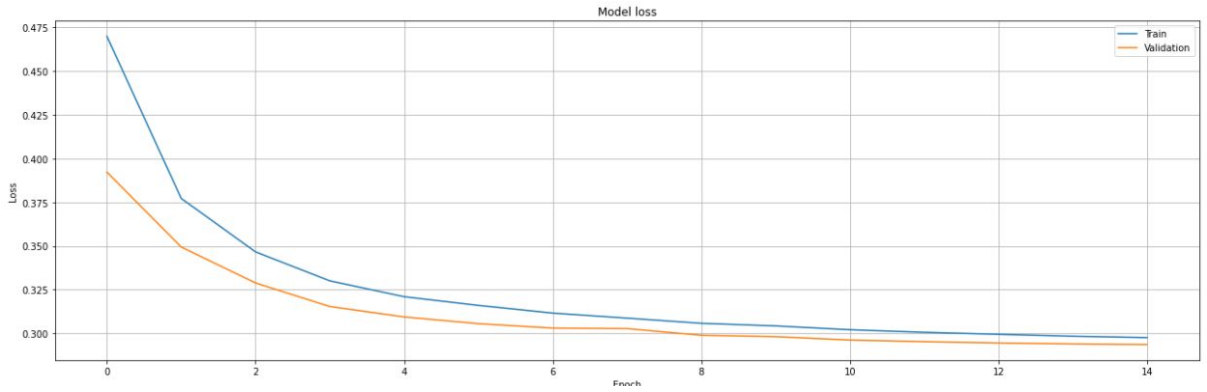
e)

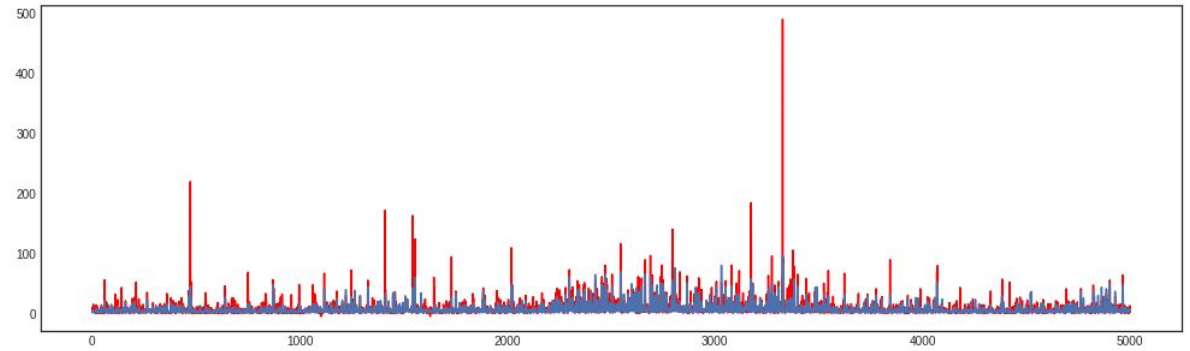
We tried to improve the model by adding more embeddings, for all the merged data we mentioned above.

```
Layer (type)                    Output Shape         Param #
Connected to
==================================================================================
==========================
input_227 (InputLayer)          [(None, 1)]          0
__________________________________________________________________________________
_________________________
input_228 (InputLayer)          [(None, 1)]          0
__________________________________________________________________________________
_________________________
input_229 (InputLayer)          [(None, 1)]          0
__________________________________________________________________________________
_________________________
input_230 (InputLayer)          [(None, 1)]          0
__________________________________________________________________________________
_________________________
input_231 (InputLayer)          [(None, 1)]          0
__________________________________________________________________________________
_________________________
input_232 (InputLayer)          [(None, 1)]          0
__________________________________________________________________________________
_________________________
input_233 (InputLayer)          [(None, 1)]          0
__________________________________________________________________________________
_________________________
input_234 (InputLayer)          [(None, 1)]          0
__________________________________________________________________________________
_________________________
input_235 (InputLayer)          [(None, 1)]          0
__________________________________________________________________________________
_________________________
input_236 (InputLayer)          [(None, 1)]          0
__________________________________________________________________________________
_________________________
input_237 (InputLayer)          [(None, 1)]          0
__________________________________________________________________________________
_________________________
input_238 (InputLayer)          [(None, 1)]          0
__________________________________________________________________________________
_________________________
embedding_202 (Embedding)       (None, 1, 5)         320
input_227[0][0]
__________________________________________________________________________________
_________________________
embedding_203 (Embedding)       (None, 1, 5)         265
input_228[0][0]
__________________________________________________________________________________
_________________________
embedding_204 (Embedding)       (None, 1, 5)         18320
input_229[0][0]
```

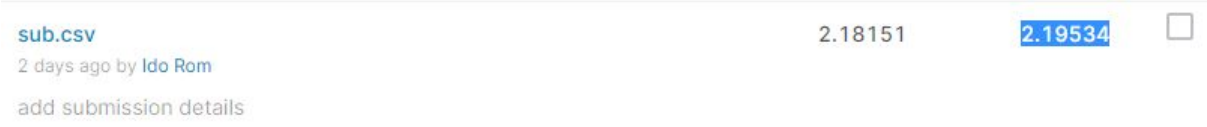
```
__________________________________________________________________________________________
_____________________________
embedding_205 (Embedding)       (None, 1, 5)          10
input_230[0][0]
__________________________________________________________________________________________
_____________________________
embedding_206 (Embedding)       (None, 1, 5)          160
input_231[0][0]
__________________________________________________________________________________________
_____________________________
embedding_207 (Embedding)       (None, 1, 5)          1590
input_232[0][0]
__________________________________________________________________________________________
_____________________________
embedding_208 (Embedding)       (None, 1, 5)          10
input_233[0][0]
__________________________________________________________________________________________
_____________________________
embedding_209 (Embedding)       (None, 1, 5)          9120
input_234[0][0]
__________________________________________________________________________________________
_____________________________
embedding_210 (Embedding)       (None, 1, 5)          110
input_235[0][0]
__________________________________________________________________________________________
_____________________________
embedding_211 (Embedding)       (None, 1, 5)          80
input_236[0][0]
__________________________________________________________________________________________
_____________________________
embedding_212 (Embedding)       (None, 1, 5)          25
input_237[0][0]
__________________________________________________________________________________________
_____________________________
embedding_213 (Embedding)       (None, 1, 5)          85
input_238[0][0]
__________________________________________________________________________________________
_____________________________
concatenate_13 (Concatenate)    (None, 1, 60)         0
embedding_202[0][0]

embedding_203[0][0]

embedding_204[0][0]

embedding_205[0][0]

embedding_206[0][0]

embedding_207[0][0]
```

```
embedding_208[0][0]

embedding_209[0][0]

embedding_210[0][0]

embedding_211[0][0]

embedding_212[0][0]

embedding_213[0][0]
__________________________________________________________________________
_______________________________
flatten_13 (Flatten)            (None, 60)           0
concatenate_13[0][0]
__________________________________________________________________________
_______________________________
batch_normalization_42 (BatchNo (None, 60)           240
flatten_13[0][0]
__________________________________________________________________________
_______________________________
dense_61 (Dense)                (None, 100)          6100
batch_normalization_42[0][0]
__________________________________________________________________________
_______________________________
dense_62 (Dense)                (None, 100)          10100
dense_61[0][0]
__________________________________________________________________________
_______________________________
batch_normalization_43 (BatchNo (None, 100)          400
dense_62[0][0]
__________________________________________________________________________
_______________________________
dense_63 (Dense)                (None, 100)          10100
batch_normalization_43[0][0]
__________________________________________________________________________
_______________________________
dense_64 (Dense)                (None, 100)          10100
dense_63[0][0]
__________________________________________________________________________
_______________________________
dense_65 (Dense)                (None, 1)            101
dense_64[0][0]
==========================================================================
============================
Total params: 67,236
Trainable params: 66,916
Non-trainable params: 320
```





Loss on validation data - 0.2936 after only 15 epochs.
We scored on kaggle - 2.19534



We can see an improvement in the loss, and the graph is more close to recognize the extreme features, but still not doing well in recognizing them.

Now, we tried to improve our model by two changes:

- We took off the ‘state’ and the ‘local’ since we thought they do not adding any logical issue for the data, and it may only confuse our model with too much information.
- We added more dense layers to increase the model.

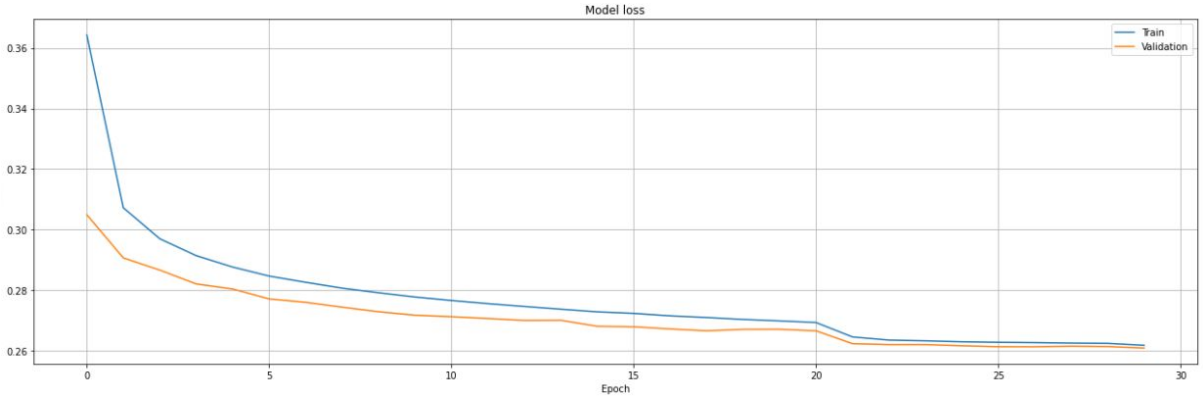
```
Layer (type)                 Output Shape              Param #
Connected to
==================================================================================
==========================
```

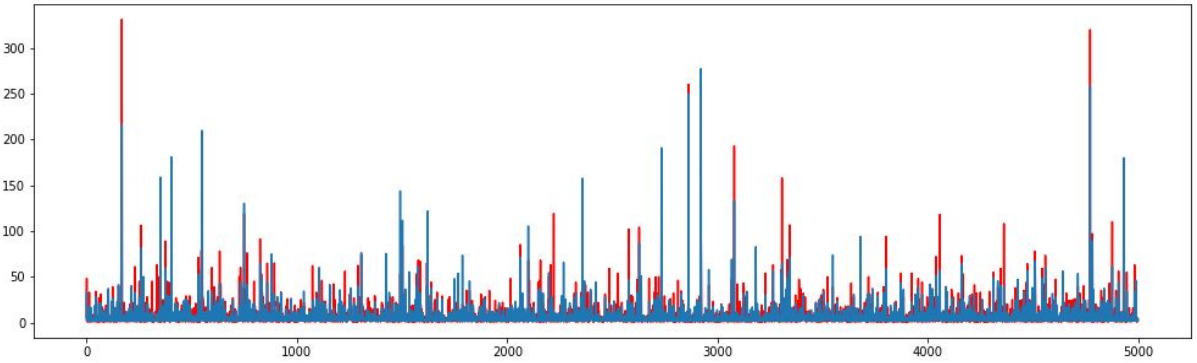
```
input_66 (InputLayer)           [(None, 1)]          0
__________________________________________________________________________________________
input_67 (InputLayer)           [(None, 1)]          0
__________________________________________________________________________________________
input_68 (InputLayer)           [(None, 1)]          0
__________________________________________________________________________________________
input_69 (InputLayer)           [(None, 1)]          0
__________________________________________________________________________________________
input_70 (InputLayer)           [(None, 1)]          0
__________________________________________________________________________________________
input_71 (InputLayer)           [(None, 1)]          0
__________________________________________________________________________________________
input_72 (InputLayer)           [(None, 1)]          0
__________________________________________________________________________________________
input_73 (InputLayer)           [(None, 1)]          0
__________________________________________________________________________________________
input_74 (InputLayer)           [(None, 1)]          0
__________________________________________________________________________________________
input_75 (InputLayer)           [(None, 1)]          0
__________________________________________________________________________________________
input_76 (InputLayer)           [(None, 1)]          0
__________________________________________________________________________________________
input_77 (InputLayer)           [(None, 1)]          0
__________________________________________________________________________________________
input_78 (InputLayer)           [(None, 1)]          0
__________________________________________________________________________________________
embedding_65 (Embedding)        (None, 1, 5)         270
input_66[0][0]
__________________________________________________________________________________________
embedding_66 (Embedding)        (None, 1, 5)         20365
input_67[0][0]
__________________________________________________________________________________________
embedding_67 (Embedding)        (None, 1, 5)         10
input_68[0][0]
__________________________________________________________________________________________
```

```
embedding_68 (Embedding)        (None, 1, 5)         165
input_69[0][0]
__________________________________________________________________________________
__________________________________
embedding_69 (Embedding)        (None, 1, 5)         1665
input_70[0][0]
__________________________________________________________________________________
__________________________________
embedding_70 (Embedding)        (None, 1, 5)         10
input_71[0][0]
__________________________________________________________________________________
__________________________________
embedding_71 (Embedding)        (None, 1, 5)         9120
input_72[0][0]
__________________________________________________________________________________
__________________________________
embedding_72 (Embedding)        (None, 1, 5)         110
input_73[0][0]
__________________________________________________________________________________
__________________________________
embedding_73 (Embedding)        (None, 1, 5)         25
input_74[0][0]
__________________________________________________________________________________
__________________________________
embedding_74 (Embedding)        (None, 1, 5)         85
input_75[0][0]
__________________________________________________________________________________
__________________________________
embedding_75 (Embedding)        (None, 1, 5)         10
input_76[0][0]
__________________________________________________________________________________
__________________________________
embedding_76 (Embedding)        (None, 1, 5)         10
input_77[0][0]
__________________________________________________________________________________
__________________________________
embedding_77 (Embedding)        (None, 1, 5)         10
input_78[0][0]
__________________________________________________________________________________
__________________________________
concatenate_5 (Concatenate)     (None, 1, 65)        0
embedding_65[0][0]

embedding_66[0][0]

embedding_67[0][0]

embedding_68[0][0]

embedding_69[0][0]
```

```
embedding_70[0][0]

embedding_71[0][0]

embedding_72[0][0]

embedding_73[0][0]

embedding_74[0][0]

embedding_75[0][0]

embedding_76[0][0]

embedding_77[0][0]
__________________________________________________________________________________
_________________________________
flatten_4 (Flatten)             (None, 65)           0
concatenate_5[0][0]
__________________________________________________________________________________
_________________________________
batch_normalization_22 (BatchNo (None, 65)           260
flatten_4[0][0]
__________________________________________________________________________________
_________________________________
dense_22 (Dense)                (None, 256)          16896
batch_normalization_22[0][0]
__________________________________________________________________________________
_________________________________
dense_23 (Dense)                (None, 128)          32896
dense_22[0][0]
__________________________________________________________________________________
_________________________________
batch_normalization_23 (BatchNo (None, 128)          512
dense_23[0][0]
__________________________________________________________________________________
_________________________________
dense_24 (Dense)                (None, 64)           8256
batch_normalization_23[0][0]
__________________________________________________________________________________
_________________________________
dense_25 (Dense)                (None, 32)           2080
dense_24[0][0]
__________________________________________________________________________________
_________________________________
dropout_22 (Dropout)            (None, 32)           0
dense_25[0][0]
__________________________________________________________________________________
_________________________________
```

```
dense_26 (Dense)                (None, 16)           528
dropout_22[0][0]
__________________________________________________________________________________
_______________________________
dense_27 (Dense)                (None, 1)            17
dense_26[0][0]
==================================================================================
============================
Total params: 93,300
Trainable params: 92,914
Non-trainable params: 386
```





The loss on the validation data: 0.26084

we received on kaggle submission 1.16948.

- We improved the random forest regressor benchmark!
- We can see that the algorithm recognizes the extreme features much better.
- We can try to add LSTM and the time series part of the problem may help to classify better the extreme values.

Now, we tried to concat an LSTM to our model.

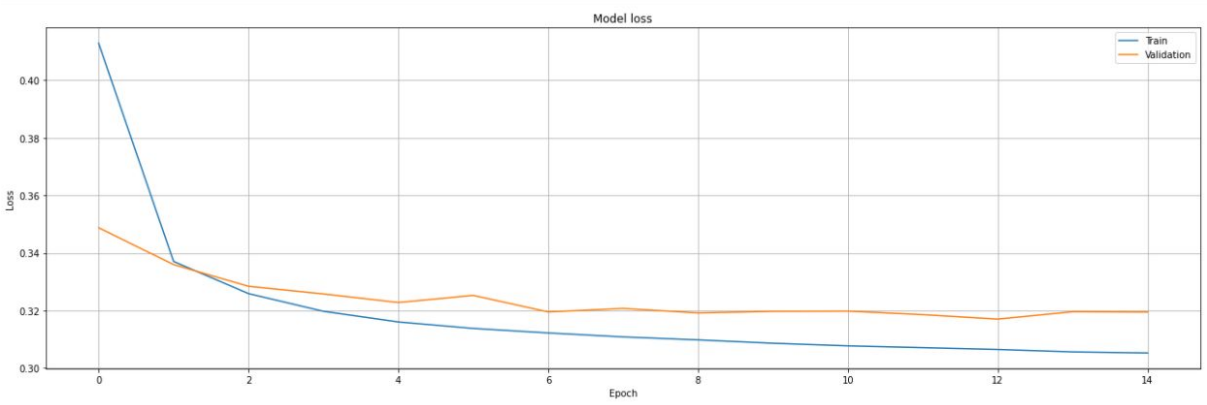
```
__________________________________________________________________________________________________
Layer (type)                    Output Shape         Param #     Connected to
==================================================================================================
input_16 (InputLayer)           [(None, 1)]          0
__________________________________________________________________________________________________
input_17 (InputLayer)           [(None, 1)]          0
__________________________________________________________________________________________________
input_18 (InputLayer)           [(None, 1)]          0
__________________________________________________________________________________________________
input_19 (InputLayer)           [(None, 1)]          0
__________________________________________________________________________________________________
input_20 (InputLayer)           [(None, 1)]          0
__________________________________________________________________________________________________
input_21 (InputLayer)           [(None, 1)]          0
__________________________________________________________________________________________________
input_22 (InputLayer)           [(None, 1)]          0
__________________________________________________________________________________________________
input_23 (InputLayer)           [(None, 1)]          0
__________________________________________________________________________________________________
input_24 (InputLayer)           [(None, 1)]          0
__________________________________________________________________________________________________
input_25 (InputLayer)           [(None, 1)]          0
__________________________________________________________________________________________________
input_26 (InputLayer)           [(None, 1)]          0
__________________________________________________________________________________________________
input_27 (InputLayer)           [(None, 1)]          0
__________________________________________________________________________________________________
input_28 (InputLayer)           [(None, 1)]          0
__________________________________________________________________________________________________
input_29 (InputLayer)           [(None, 1)]          0
```

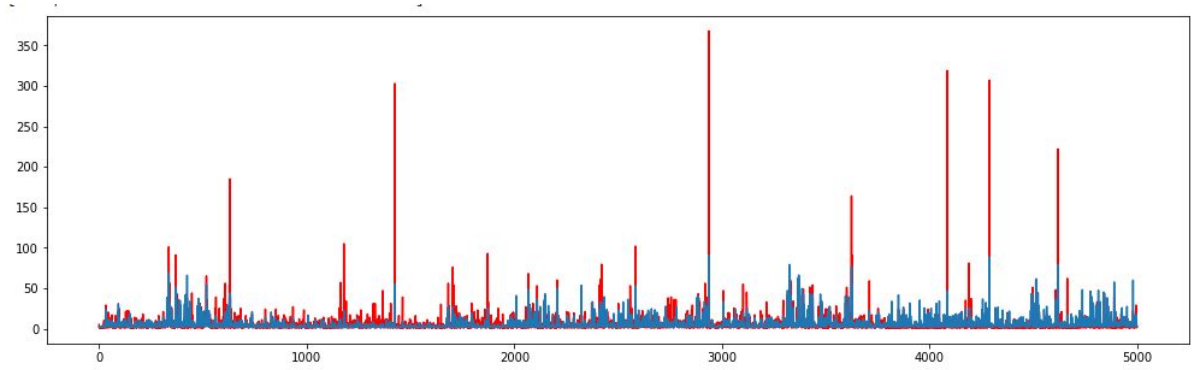
input_30 (InputLayer)	[(None, 1)]	0
embedding_15 (Embedding) input_16[0][0]	(None, 1, 5)	270
embedding_16 (Embedding) input_17[0][0]	(None, 1, 5)	20365
embedding_17 (Embedding) input_18[0][0]	(None, 1, 5)	10
embedding_18 (Embedding) input_19[0][0]	(None, 1, 5)	165
embedding_19 (Embedding) input_20[0][0]	(None, 1, 5)	1665
embedding_20 (Embedding) input_21[0][0]	(None, 1, 5)	10
embedding_21 (Embedding) input_22[0][0]	(None, 1, 5)	9120
embedding_22 (Embedding) input_23[0][0]	(None, 1, 5)	110
embedding_23 (Embedding) input_24[0][0]	(None, 1, 5)	80
embedding_24 (Embedding) input_25[0][0]	(None, 1, 5)	25
embedding_25 (Embedding) input_26[0][0]	(None, 1, 5)	85
embedding_26 (Embedding) input_27[0][0]	(None, 1, 5)	10

```
__________________________________________________________________________________________________
_________________________________
embedding_27 (Embedding)        (None, 1, 5)          10
input_28[0][0]
__________________________________________________________________________________________________
_________________________________
embedding_28 (Embedding)        (None, 1, 5)          10
input_29[0][0]
__________________________________________________________________________________________________
_________________________________
embedding_29 (Embedding)        (None, 1, 5)          10
input_30[0][0]
__________________________________________________________________________________________________
_________________________________
concatenate_1 (Concatenate)     (None, 1, 75)         0
embedding_15[0][0]

embedding_16[0][0]

embedding_17[0][0]

embedding_18[0][0]

embedding_19[0][0]

embedding_20[0][0]

embedding_21[0][0]

embedding_22[0][0]

embedding_23[0][0]

embedding_24[0][0]

embedding_25[0][0]

embedding_26[0][0]

embedding_27[0][0]

embedding_28[0][0]

embedding_29[0][0]
__________________________________________________________________________________________________
_________________________________
lstm_1 (LSTM)                   (None, 512)           1204224
concatenate_1[0][0]
__________________________________________________________________________________________________
_________________________________
```

```
flatten_1 (Flatten)             (None, 512)          0
lstm_1[0][0]
_____________________________________________________________________________
_____________________________
batch_normalization_6 (BatchNor (None, 512)          2048
flatten_1[0][0]
_____________________________________________________________________________
_____________________________
dropout_6 (Dropout)             (None, 512)          0
batch_normalization_6[0][0]
_____________________________________________________________________________
_____________________________
dense_6 (Dense)                 (None, 256)          131328
dropout_6[0][0]
_____________________________________________________________________________
_____________________________
batch_normalization_7 (BatchNor (None, 256)          1024
dense_6[0][0]
_____________________________________________________________________________
_____________________________
dropout_7 (Dropout)             (None, 256)          0
batch_normalization_7[0][0]
_____________________________________________________________________________
_____________________________
dense_7 (Dense)                 (None, 128)          32896
dropout_7[0][0]
_____________________________________________________________________________
_____________________________
batch_normalization_8 (BatchNor (None, 128)          512
dense_7[0][0]
_____________________________________________________________________________
_____________________________
dropout_8 (Dropout)             (None, 128)          0
batch_normalization_8[0][0]
_____________________________________________________________________________
_____________________________
dense_8 (Dense)                 (None, 64)           8256
dropout_8[0][0]
_____________________________________________________________________________
_____________________________
batch_normalization_9 (BatchNor (None, 64)           256
dense_8[0][0]
_____________________________________________________________________________
_____________________________
dropout_9 (Dropout)             (None, 64)           0
batch_normalization_9[0][0]
_____________________________________________________________________________
_____________________________
dense_9 (Dense)                 (None, 32)           2080
dropout_9[0][0]
```

```
_____
_____
batch_normalization_10 (BatchNo (None, 32)           128
dense_9[0][0]
_____
_____
dropout_10 (Dropout)            (None, 32)           0
batch_normalization_10[0][0]
_____
_____
dense_10 (Dense)                (None, 16)           528
dropout_10[0][0]
_____
_____
batch_normalization_11 (BatchNo (None, 16)           64
dense_10[0][0]
_____
_____
dropout_11 (Dropout)            (None, 16)           0
batch_normalization_11[0][0]
_____
_____
dense_11 (Dense)                (None, 1)            17
dropout_11[0][0]
==================================================================
============================
Total params: 1,415,306
Trainable params: 1,413,290
Non-trainable params: 2,016
```





Loss on validation data - 0.31701 after only 15 epochs.

We tried many different models using LSTM, and the one above gave the best results, unfortunately we didn't succeed in improving our score.

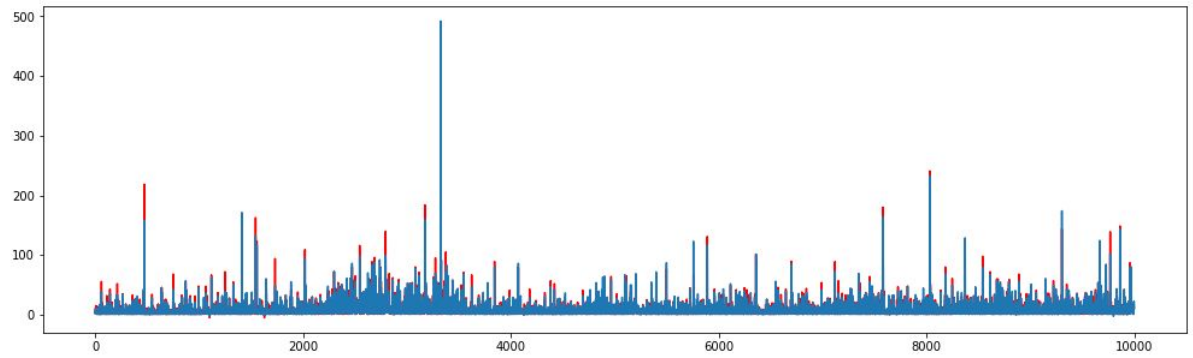
f)

Interesting insights we got from the embeddings of categorical features -

- Use of categorical embeddings may help with adapting the data analysis to the relevant features.
- We can see that adding embeddings receives better results than the simple algorithm.
- We need to think well about which feature we need to add to our concatenate data and which are less relevant.
- We still believe that adding LSTM in this type of problems may perform good results, even though we didn't succeed in doing it.

g)

We used the embeddings we received at our best model as a feature extractor for the random forest regression model.



valid data mean absolute error - 1.645, with only 5 estimators on the random forest model.

We can see that the results are much better than our result, the extreme values recognized very well!