KAGGLE 1C SALES PREDICTION COMPETITION

Kshitij Singla
Aug 11, 2021

Hello!

I Am Kshitij Singla

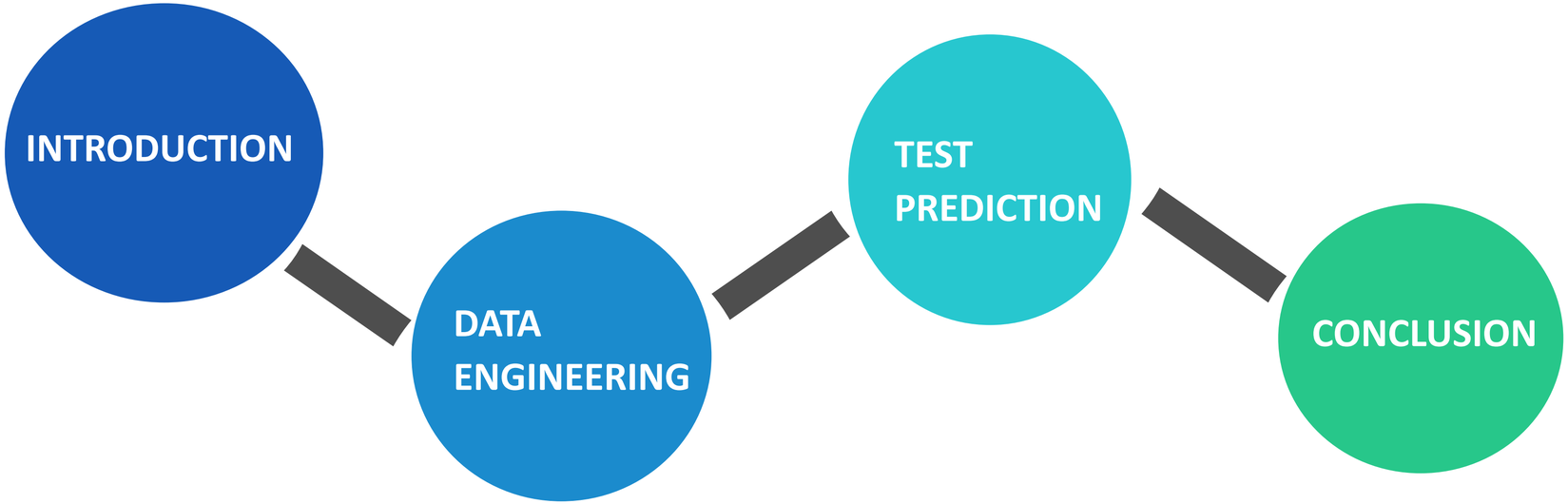
I am currently working in the Financial Forecasting team at Citibank – Have developed a model on Interest Rate Forecasting using PCA among others.

I graduated with a MS in Quantitative and Computational Finance from Georgia Tech and am enrolled to start studying MS in CS at GT this Fall.

I have developed a deep interest in the fields of Computer Science & Data Analytics and am thus looking for exciting opportunities in the same!

You can contact me at ksingla6@gatech.edu

FORECASTING PROCESS



Agenda

- ❑ INTRODUCTION
  - Intro to Competition
  - Data Challenges
- ❑ DATA ENGINEERING
- ❑ TEST PREDICTION
- ❑ CONCLUSION

Intro to Competition

Competition Setting: Predicting Future Sales for a Russian software retail company 1C; Given a timeseries starting 2013-2015 Oct for sales of 1C on various items in it's multiple shop outlets, the ask is to predict monthly sales for the various shop-item pairs in Nov 2015.

Data Timeline

Data Snippet

	date	date_block_num	shop_id	item_id	item_price	item_cnt_day
0	02.01.2013	0	59	22154	999.00	1.0
1	03.01.2013	0	25	2552	899.00	1.0
2	05.01.2013	0	25	2552	899.00	-1.0
3	06.01.2013	0	25	2554	1709.05	1.0
4	15.01.2013	0	25	2555	1099.00	1.0

Data Challenges

Russian Data **-** Nearly all of the data is in **Russian** text, needs to be translated

	shop_name	item_name	item_category_name
214195	Самара ТЦ "ПаркХаус"	СБ. Союз 55	Музыка - CD локального производства
214196	Самара ТЦ "ПаркХаус"	Настольная игра Нано Кёрлинг	Подарки - Настольные игры
214197	Самара ТЦ "ПаркХаус"	НОВИКОВ АЛЕКСАНДР Новая коллекция	Музыка - CD локального производства
214198	Самара ТЦ "ПаркХаус"	ТЕРЕМ - ТЕРЕМОК сб.м/ф (Регион)	Кино - DVD
214199	Самара ТЦ "ПаркХаус"	3 ДНЯ НА УБИЙСТВО (BD)	Кино - Blu-Ray

Large Data - The 'sales_train' dataframe comprises of **3 million** rows, we must trim the noise from the data as far as possible

Mix of shops and items Data **(with some Test data items unseen in Train)** - We are presented with transactions of various items and various shops in our train data set. Our model must be versatile enough to be able to predict sales for items for which we have no data in our train set too!

DATA ENGINEERING
-Data Transforms
-Data Cleaning
-Down Sampling Data
-Feature Creation

Agenda

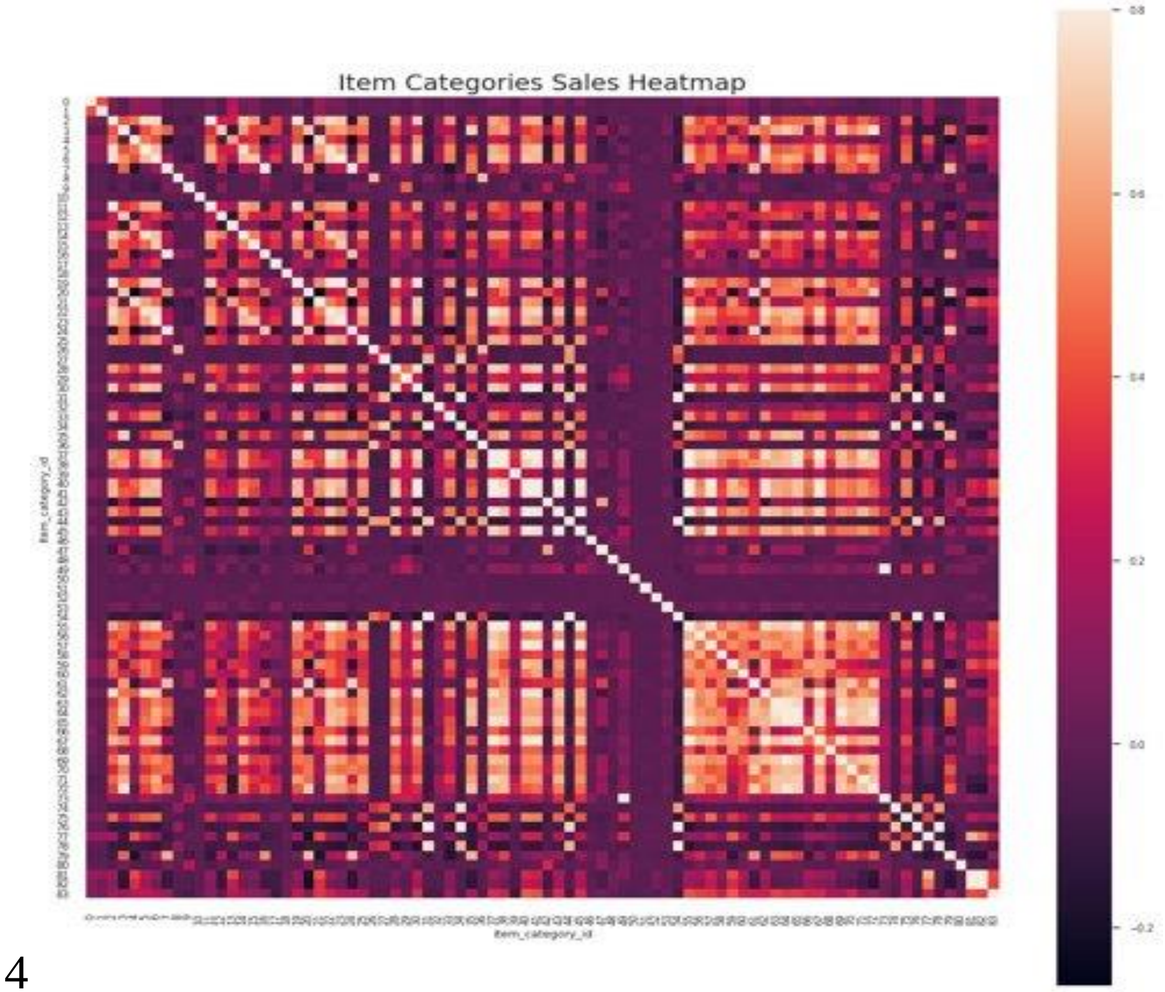
- ❑ INTRODUCTION
- ❑ DATA ENGINEERING
  - Data Transforms
  - Data Cleaning
  - Downsampling Data
  - Feature Creation
- ❑ TEST PREDICTION
- ❑ CONCLUSION

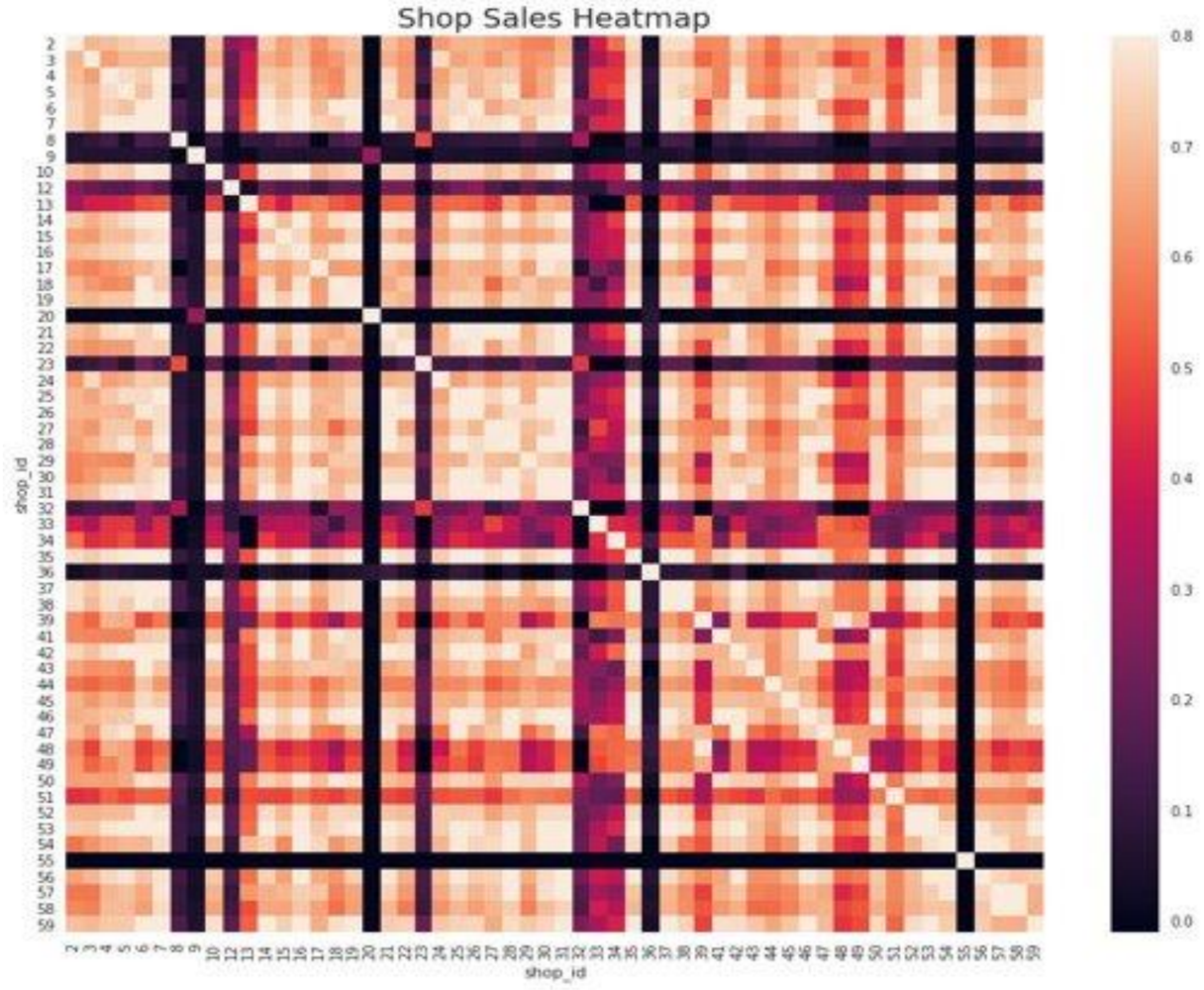
DATA ENGINEERING
-Data Transforms
-Data Cleaning
-Down Sampling Data
-Feature Creation

Data Transformations

Translating and Broadening name columns - By broadening shops for e.g. ‘Moscow shopping center’ became ‘Moscow’ as a ‘city’ feature, similarly for items for e.g. ‘

Removing item categories (16) and shops (4) absent in Test Set – The item categories' sales data were generally very uncorrelated and so was the case for some of the shops, so the training data for these could be eliminated without loss of much forecasting signal





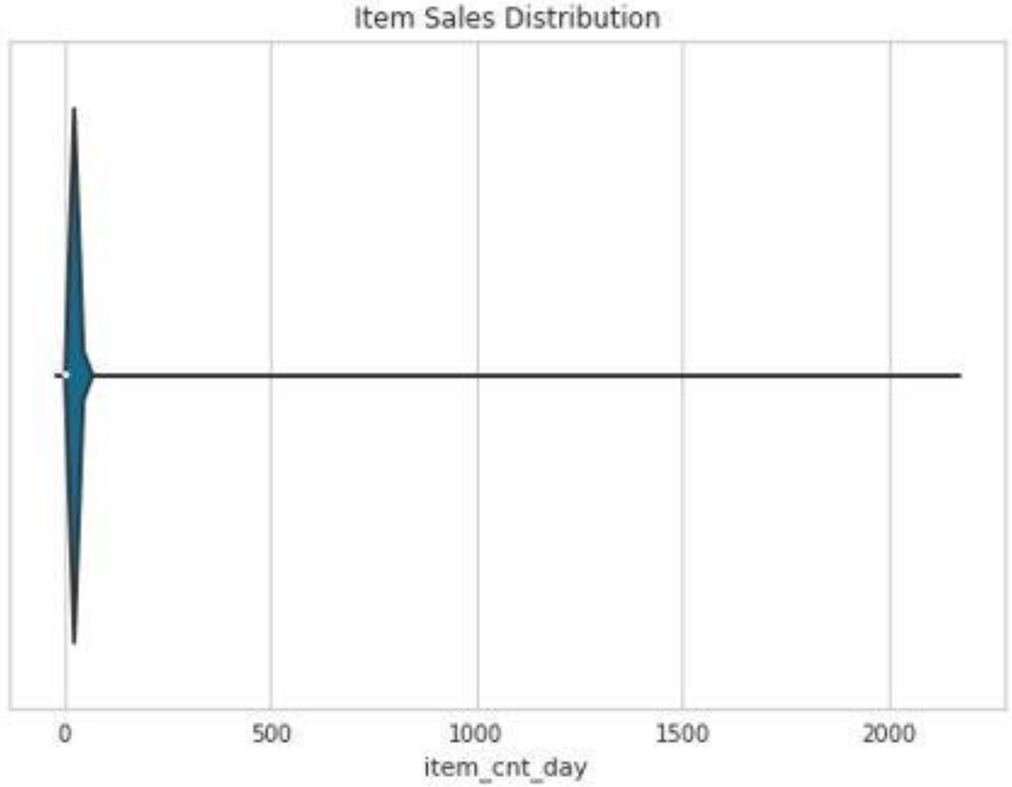
DATA CLEANING & OUTLIER DETECTION

- Sales of >700 in a single transaction of an item by a software retail company seems super unlikely. Let's get rid of them.
- No missing values in Data

```
Missing Values in Sales Data:
date              0
date_block_num    0
shop_id           0
item_id           0
item_price        0
item_cnt_day      0
Revenue           0
month             0
dtype: int64
```

```
sales['item_cnt_day'].value_counts(bins=3)
```

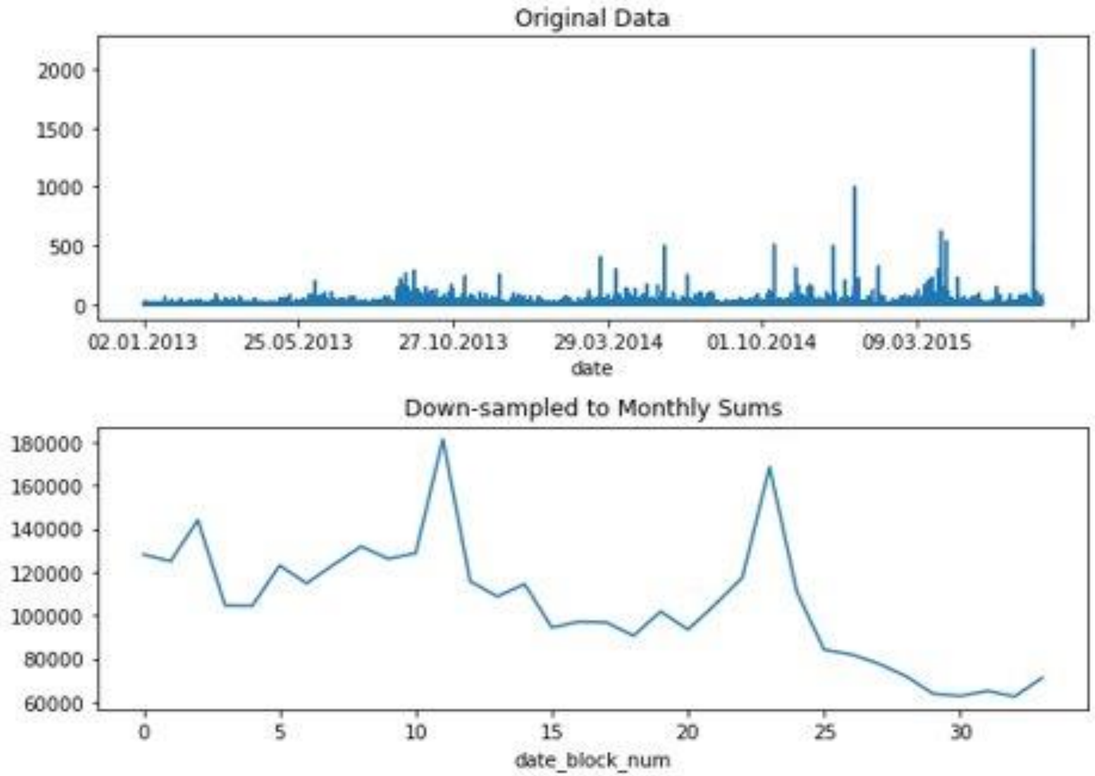
```
(-24.192, 708.333]     2935847
(708.333, 1438.667]          1
(1438.667, 2169.0]           1
Name: item_cnt_day, dtype: int64
```



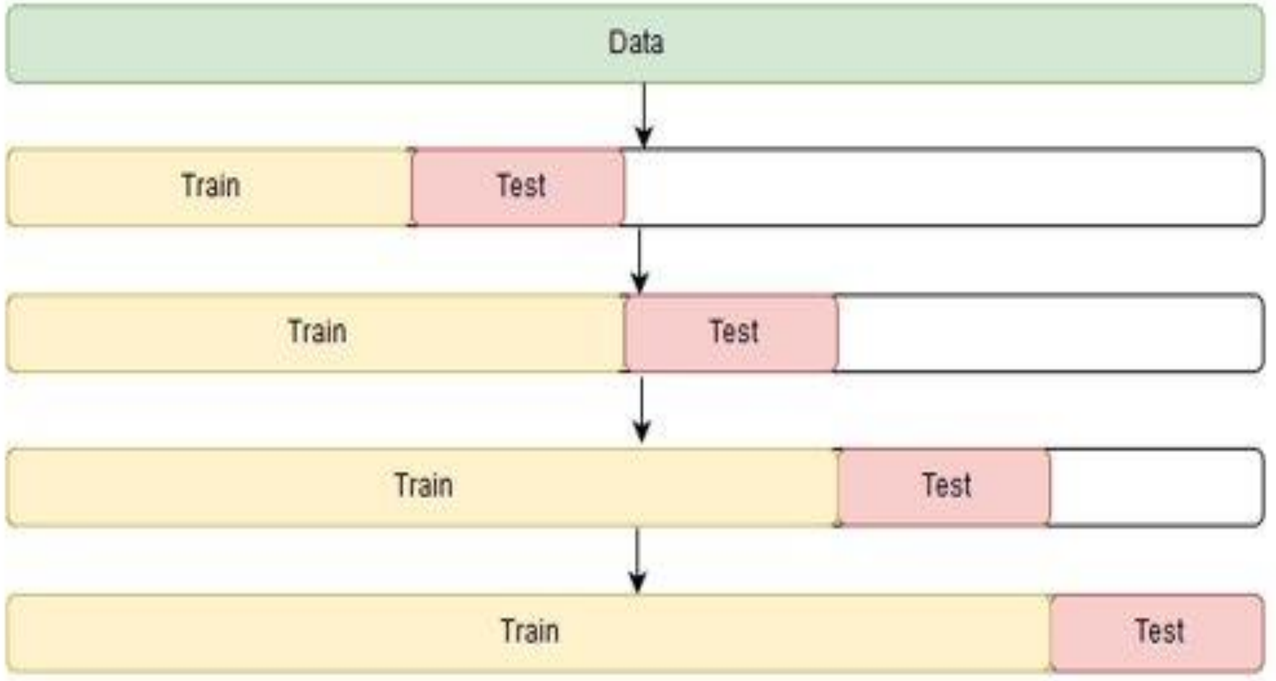
DATA ENGINEERING
-Data Transforms
-Data Cleaning
-Down Sampling Data
-Feature Creation

Down Sampling the Data

- Decreased frequency of data to ‘Monthly’ from ‘Daily’ to reflect the Test data structure
- But importantly, we are able to leverage the higher frequency data for features like mean, std, frequencies during the month
- Finally a rolling forward method was utilized for cross validation testing of the models



Roll Forward Validation



DATA ENGINEERING
-Data Transforms
-Data Cleaning
-Down Sampling Data
-Feature Creation

Automated Feature Creation

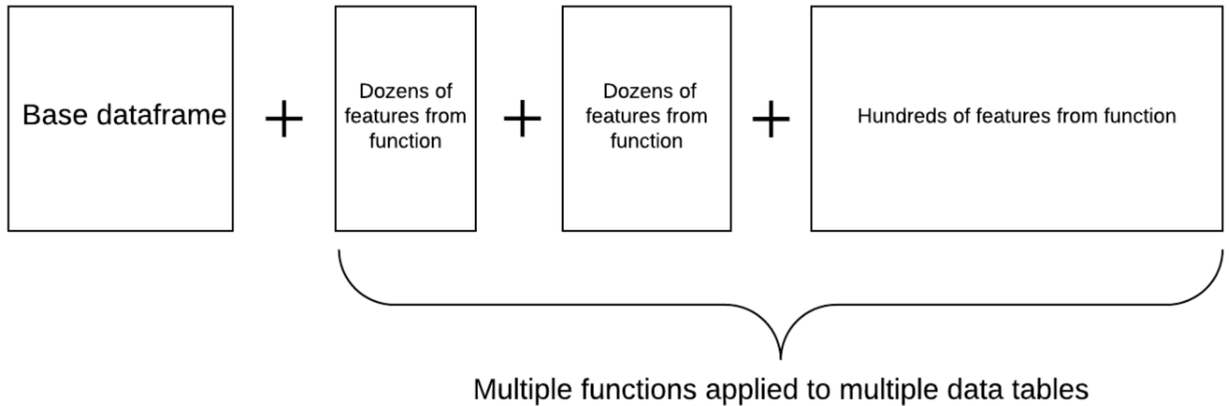
3 Steps Process

1. Generating Various Categorical Features -
A) Location features – city, city population, cross-location distances, shop type (shopping centre, online etc.)
B) Text features – First word, tf-idf vectorization, length of names
C) *Item related features* – Broad Category (Music/Game etc.), Platform (PS2,PS3, PC), Artist name
D) *Interaction features* – shop-item, shop-item category item sales, price*item sales
E) *Seasonality features* – Month number, number of days in month
F) *Uniqueness features* – No. of items, categories per month (**discovers data leakages about data generation process**)

2. Target encoding (mean, frequency counts and sum across the month for each category for item sales and price)
3. Lagging the features (only lagged features can be used while forecasting test data)

Extra features – Number of days since shop, item open (first sale of shop, item), difference/ratio of lagged features, binned price feature(lot of split points)

Automated Feature Discovery and Engineering



TEST PREDICTION
-Modelling
-Ensembling
-Post Processing

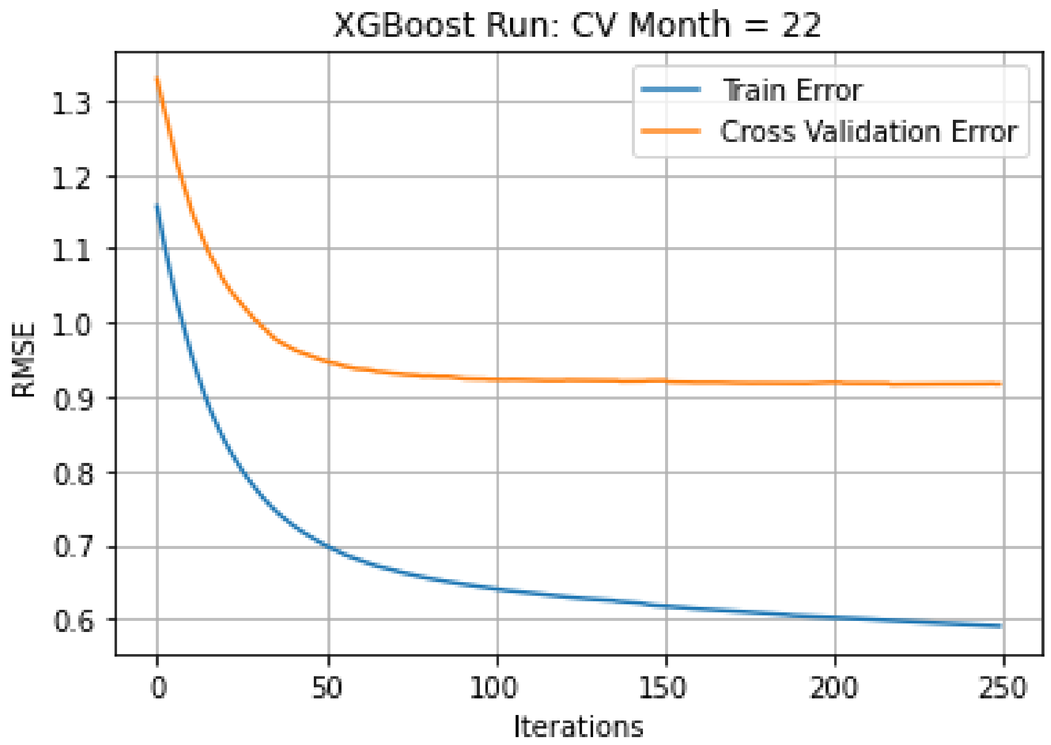
Agenda

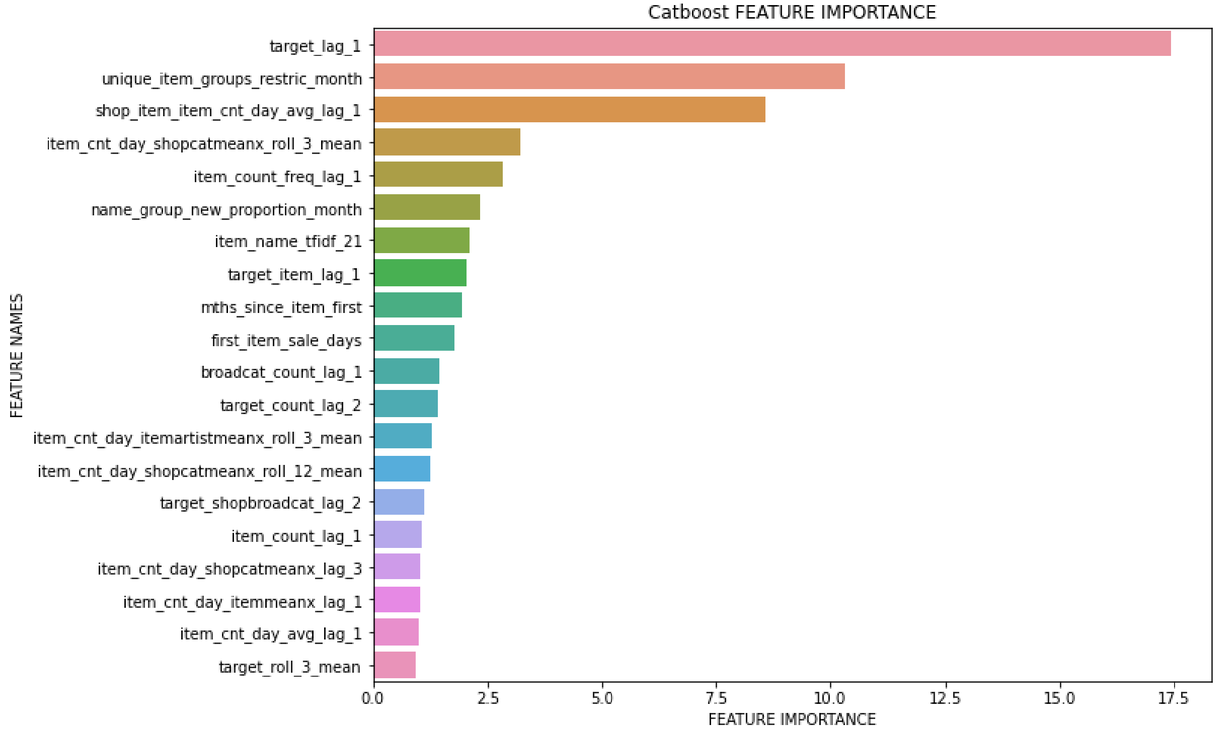
- ❑ INTRODUCTION
- ❑ DATA ENGINEERING
- ❑ TEST PREDICTION
  - Modelling
  - Ensembling
  - Post Processing
- ❑ CONCLUSION

TEST PREDICTION
-Modelling
-Ensembling
-Post Processing

Modelling

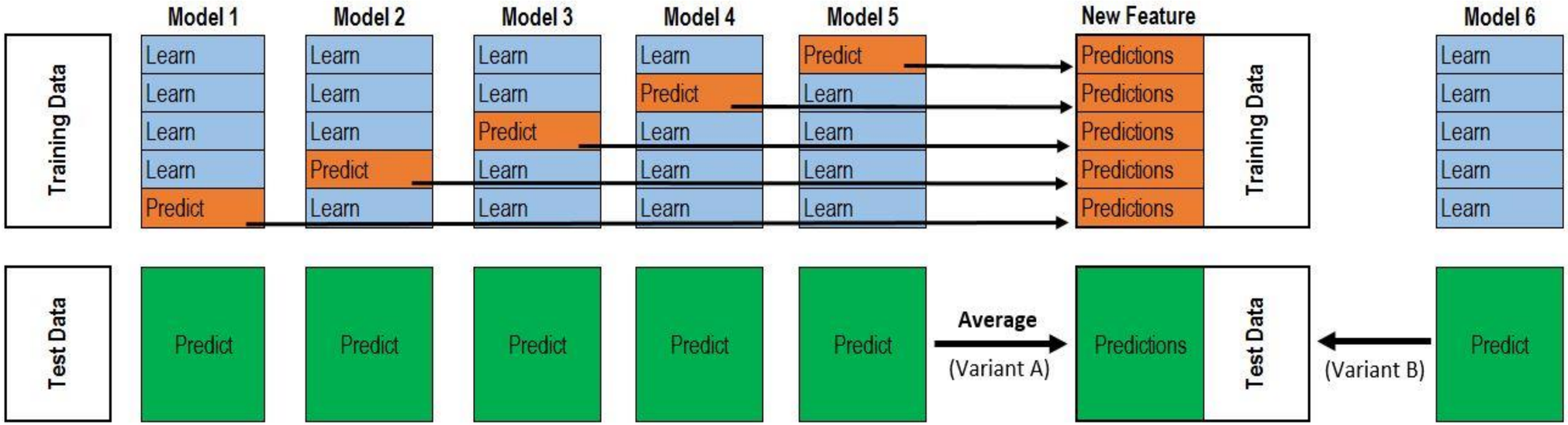
- Tree based models used primarily- LGBM, XGBoost, CatBoost
- Bayesian Hyperparameter tuning used
- Regularization parameters like the L2 weight, subsampling data & features extremely instrumental





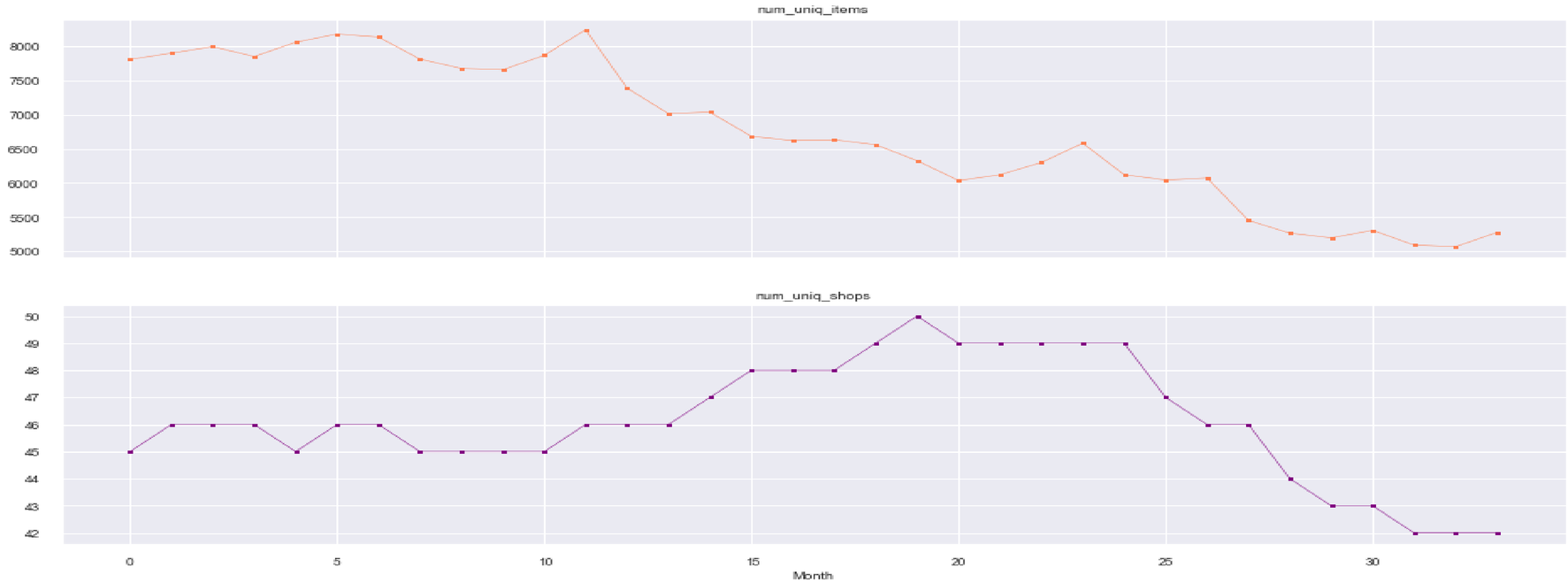
Ensembling

- Stacking, Bagging Ensembles used where meta-learner is also a tree based model
- Linear Regression and Tree Based models used
- Models trained on log target and target difference as well to account for seasonality in target data



TEST PREDICTION
-Modelling
-Ensembling
-Post Processing

Post Processing



- We see a clear decrease in both the number of item offerings and operating shops for 1C
- How many items are in test set that could be such that they are now outdated, i.e. have not been sold by any of the shops since past few months?

`Outdated items in test set: 6972`

- Predicting item sales for these items as 0.0 improved my Test RMSE score from ~0.85 to ~0.82
- Post processing based on intuition: Russian population is generally infatuated with gaming. I manually increased the sales values of the gaming items, which significantly improved model score as well. Tested differing growth bumps in validation data sets (found about 60% to be best)

CONCLUSION

Some Unexpected Learnings

- For tree based models, there are several advantages – **little scaling needed** (log, power transformations), XGBoost can handle NaNs too. However some surprising disadvantages – simple multiplication and division of features can vastly improve model. Case in point - 'Revenue' and feature ratios

- With mean and frequency encodings added for categorical variables, tree boosting methods could optimize for higher scores with **lower depth** in trees (easier to find those feature tgt relationships -> as seen by the **higher number of split points in tree**)

- Regularization (like smoothing) was especially a must while mean encoding because mean encoding generally leads to overfitting on train data, but poor performance during cross validation (especially because distributions of data tend to change over time). But a surprising discovery here was that with proper L2 reg and other params like low subsample data, **bias variance tradeoff was absent.** Increasing the model complexity to overfit train data directly improved the score on the test set as well (until it plateaued to my final score)

- Despite so much input data, external data in the form of city size and population was very handy and a lot of the given shops/item categories just served to add noise! Shows the importance of **Quality Data over Big Data**

- **Weak uncorrelated learners (better than baseline) scored much higher** than strong but correlated learners in terms of ensembling. For e.g. Linear Regression scored very low as an individual model, but when stacked with the other tree models' predictions, the ensemble scored a lot higher than just an ensemble of the tree models output

Thanks!

Any questions?

You can find me at: ksingla6@gatech.edu