

---

---

# InstaCart Challenge











— W207 Section 4 Final Project —  
Aditi, Yang, Josh

---

---

# InstaCart Challenge

Quintessential Question:

	Item 1	Item 2	Item 3	Item 4
Order 1				
Order 2				
Order 3	???	???	???	???

What will buyers put in their basket next time?

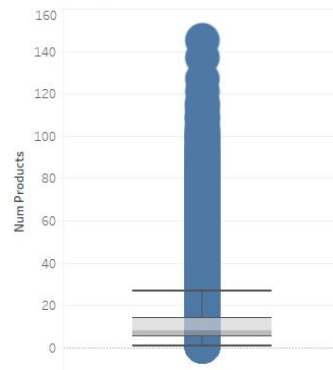


# Initial Exploration of the Data

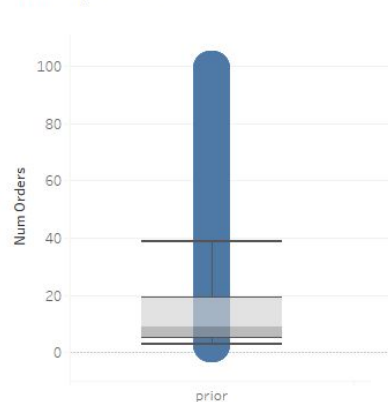
- Instacart provided sample data for the competition in a number of csv files
- 3.4 million Orders
- 200 thousand unique Users
- 500 thousand unique Products
- 13 million records, combination of orders and products
- EC2 instance to host our notebook

Orders	
Prior	3.2 million
Train	131 thousand
Test	75 thousand

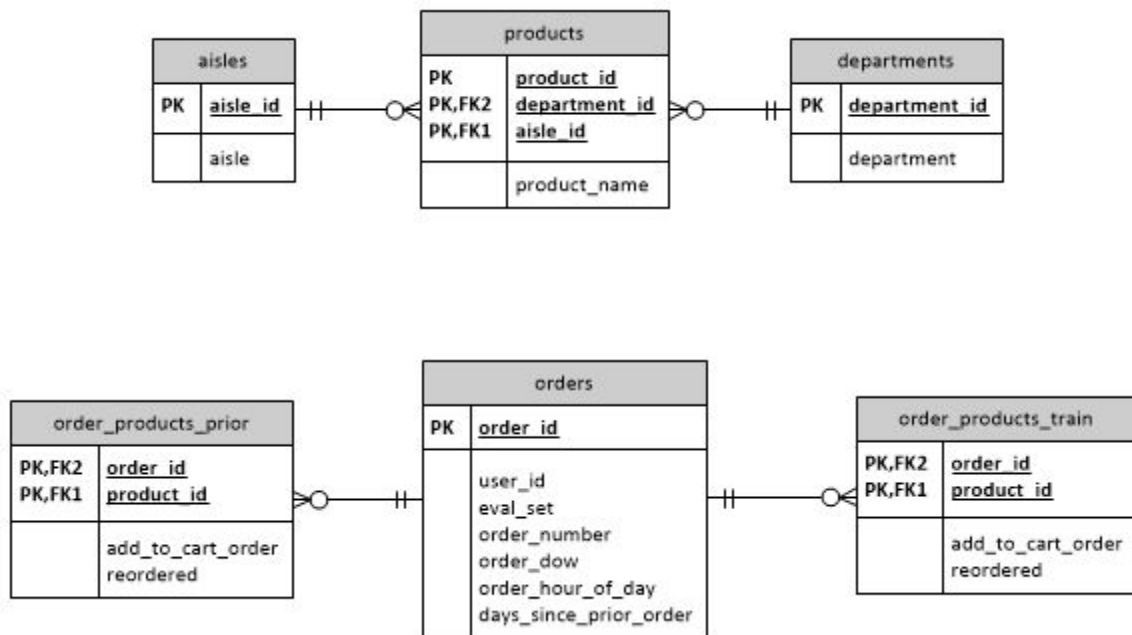
Products per Order



Orders per User





# Structure of Data



# Feature Engineering

## Baseline Submission

- The next order will be the same as the previous order: if it  **yesterday**, then it will  **today**.

## First Intuition

- **Considered it a NLP problem**
- **Use Count Vectorizer on Product Names -**
- Chose to **ignore the independence** assumption
  - ◆ Sparse matrix (9K words)
  - ◆ Ignoring the temporal aspect of orders
  - ◆ **No Infra in place - wasn't feasible**

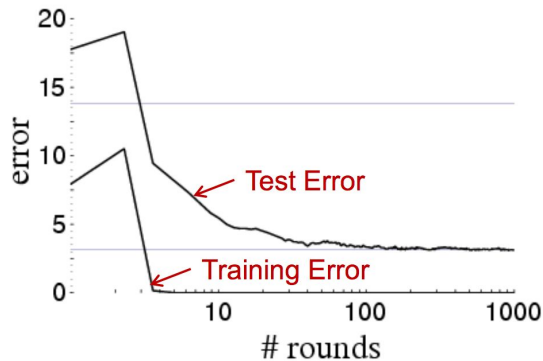
## Final Model

- Moved to EC2 instance
- Instacart had successfully used **XGBoost** on the dataset
- We chose to explore Boosting for this problem at hand - **Gradient Boosted Machine**



# Why Gradient Boosted Machines?

- Instead of learning single good classifier, learn many weak classifiers that are good in different parts of the input space



- Robust to overfitting
- Simple to implement and effective classifier
- Can be used for continuous and categorical data



# Test and Train -> Feature Extraction

## Train Data

order_id	user_id	eval_set	order_number	order_dow	order_hour_of_day	days_since_prior_order
1187899	1	train	11	4	8	14.0
1492625	2	train	15	1	11	30.0
2196797	5	train	5	0	11	6.0
525192	7	train	21	2	11	6.0
880375	8	train	4	1	14	10.0

## Test Data

order_id	user_id	eval_set	order_number	order_dow	order_hour_of_day	days_since_prior_order
2774568	3	test	13	5	15	11.0
329954	4	test	6	3	12	30.0
1528013	6	test	4	3	16	22.0
1376945	11	test	8	6	11	8.0
1356845	12	test	6	1	20	30.0



	0	1
order_id	1.187899e+06	1.187899e+06
product_id	1.712200e+04	1.960000e+02
user_total_orders	1.100000e+01	1.100000e+01
user_total_items	5.900000e+01	5.900000e+01
total_distinct_items	1.800000e+01	1.800000e+01
user_average_days_between_orders	1.900000e+01	1.900000e+01
user_average_basket	5.363636e+00	5.363636e+00
order_hour_of_day	8.000000e+00	8.000000e+00
days_since_prior_order	1.400000e+01	1.400000e+01
days_since_ratio	7.368421e-01	7.368421e-01
aisle_id	2.400000e+01	7.700000e+01
department_id	4.000000e+00	7.000000e+00
product_orders	1.388000e+04	3.579100e+04
product_reorders	9.377000e+03	2.779100e+04
product_reorder_rate	6.755764e-01	7.764801e-01
UP_orders	1.000000e+00	1.000000e+01
UP_orders_ratio	9.090909e-02	9.090909e-01
UP_average_pos_in_cart	6.000000e+00	1.400000e+00
UP_reorder_rate	9.090909e-02	9.090909e-01
UP_orders_since_last	6.000000e+00	1.000000e+00
UP_delta_hour_vs_last	7.000000e+00	0.000000e+00

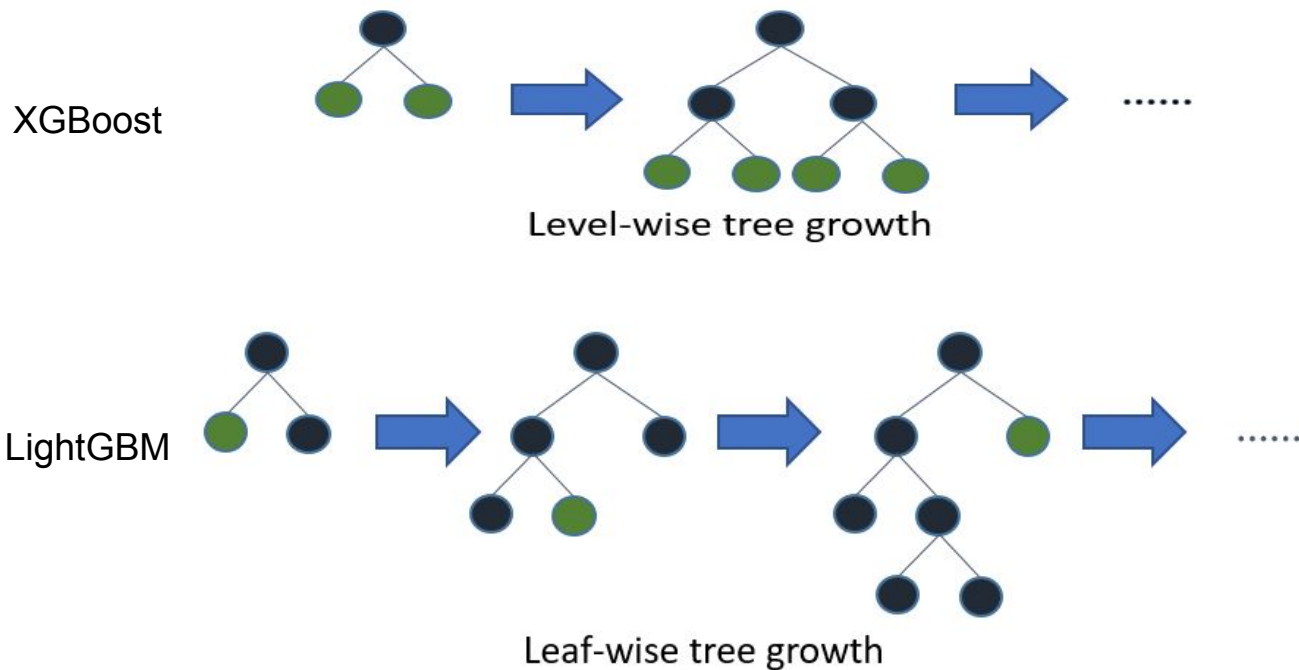
User  
Features

Order  
Features

Product  
Features

User  
Product  
Features

# Why LightGBM?





# Model Selection

- There are many different parameters involved in lightGBM:
  - Rounds
  - Tree depth
  - Weighting
  - Subsampling
- Integration with SKLearn
- Reliance on Microsoft and their documentation
- We measured success of the model using the F1 score, per Kaggle instructions



# Future Work

- Explore different models, RNN, and Word2Vec
- Obtain more data and engineer more features: geographic, demographic
- Add to the model, way to weigh overall product popularity vs personal preferences
- Make the model perform on a robust architecture



# References

Kaggle Instacart Challenge

<https://www.kaggle.com/c/instacart-market-basket-analysis>

A Kaggle Master Explains Gradient Boosting

<http://blog.kaggle.com/2017/01/23/a-kaggle-master-explains-gradient-boosting/>

XGBoost: A Scalable Tree Boosting System

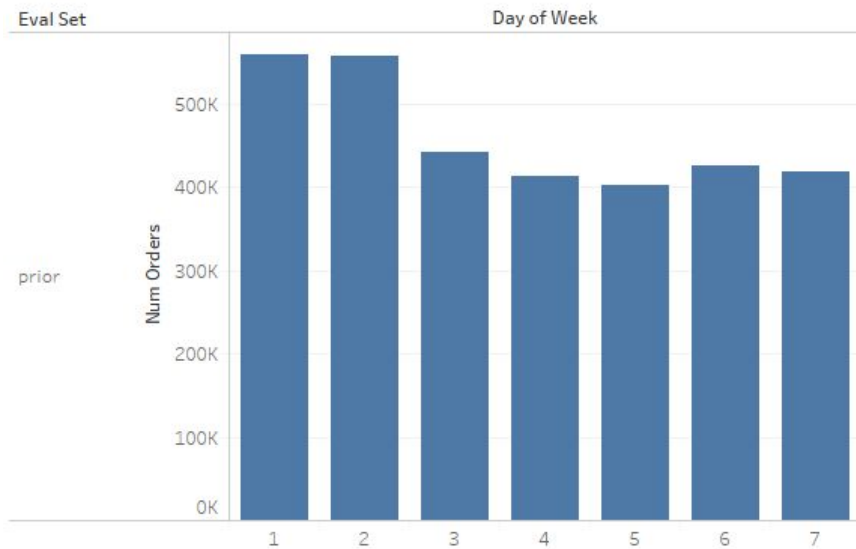
<https://arxiv.org/abs/1603.02754>

VP of Data Science @ InstaCart discussing how they approach the problem

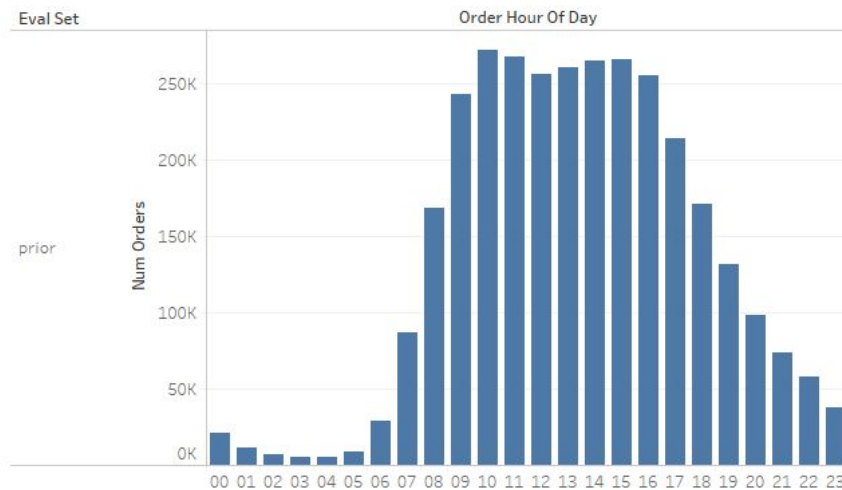
<https://tech.instacart.com/3-million-instacart-orders-open-sourced-d40d29ead6f2>

# Appendix: Orders and Time

Day of Order



Order Hour of Day



# Appendix: Orders and Time continued

Order Day and Hour

Eval Set	Order Hour ..	Day of Week						
		1	2	3	4	5	6	7
prior	00	3,692	3,475	2,906	2,767	2,476	2,989	3,067
	01	2,235	1,735	1,485	1,407	1,414	1,539	1,781
	02	1,299	1,063	892	889	837	953	1,137
	03	888	711	679	605	640	796	801
	04	750	761	700	666	694	857	747
	05	1,076	1,523	1,330	1,265	1,251	1,466	1,061
	06	3,138	5,101	4,524	4,314	4,135	4,573	3,007
	07	11,530	15,792	12,550	11,739	11,823	12,590	10,632
	08	26,223	32,563	23,488	21,305	20,597	22,623	21,522
	09	37,990	49,533	34,499	30,653	29,645	32,258	28,918
	10	44,999	52,999	37,258	34,016	33,056	36,205	33,352
	11	47,357	49,008	36,186	33,275	31,849	35,695	34,636
	12	47,729	44,550	33,752	31,580	30,337	33,501	34,757
	13	49,997	44,235	34,646	32,194	30,778	34,117	35,207
	14	50,484	44,220	35,102	32,863	31,665	35,131	36,091
	15	50,020	43,913	35,408	33,909	32,095	35,218	35,569
	16	45,930	42,200	35,482	33,284	32,175	33,700	33,178
	17	36,874	34,610	30,216	28,619	27,577	28,086	28,098
	18	27,347	27,293	24,886	23,536	22,872	22,700	22,364
	19	20,972	20,926	18,922	18,105	18,194	17,513	16,988
	20	16,984	15,422	14,223	13,064	13,351	12,557	12,508
	21	13,425	11,292	10,071	9,718	10,169	8,956	9,805
	22	10,440	8,458	7,671	7,732	8,274	7,006	7,959
	23	6,393	5,322	5,079	4,895	5,308	4,953	5,663