Market Basket Prediction

Predicting which products will Instacart customers buy again

¶

Solomon Mengistu

Outline

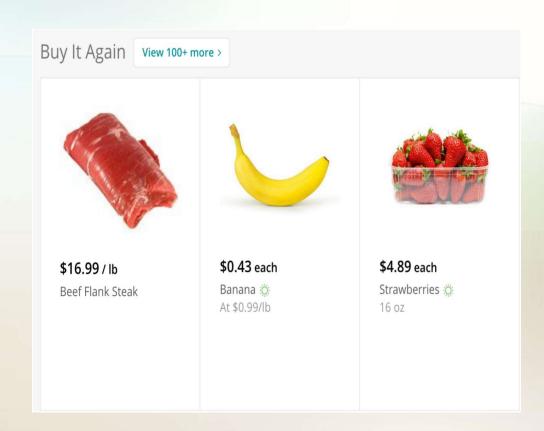
- Introduction
- Data overview
- Data exploration with visualization
- Feature engineering and data preparation
- Prediction models
- Dimensionality reduction
 - PCA
 - SelectKBest
- Neural Networks
- Evaluation
- Conclusion and recommendations

Introduction

- About Instacart
 - Online grocery shopping service
- Objective of the project
 - Predict if a customer will buy an item again

Introduction

- How the result can be used
 - Recommendation
 - Shopping experience
 - Customer retention



Data Overview

- Six relational tables
 - 3M+ orders
 - 200 000+ users
 - 4 100 orders of each user

Data Overview

Products shape: (49688, 4)

	product_id	product_name	aisle_id	department_id
0	1	Chocolate Sandwich Cookies	61	19
1	2	All-Seasons Salt	104	13
2	3	Robust Golden Unsweetened Oolong Tea	94	7
3	4	Smart Ones Classic Favorites Mini Rigatoni Wit	38	1
4	5	Green Chile Anytime Sauce	5	13

Aisles shape: (134, 2)

	aisle_id	aisle	Departments shape: (21, 2)							
0	1	prepared soups salads		department_id	department					
1	2	specialty cheeses	0	1	frozen					
2	3	energy granola bars	1	2	other					
3	4	instant foods	2	3	bakery					
4	5	marinades meat prepar	3	4	produce					
*			4	5	alcohol					

prior shape (32434489, 4)

	order_id	product_id	add_to_cart_order	reordered
0	2	33120	1	1
1	2	28985	2	1
2	2	9327	3	0
_	_	45040	4	4

Order train shape: (1384617, 4)

	order_id	product_id	add_to_cart_order	reordered
0	1	49302	1	1
1	1	11109	2	1
2	1	10246	3	0
3	1	49683	4	0
4	1	43633	5	1

Orders shape: (3421083, 7)

	order_id	user_id	eval_set	order_number	order_dow	order_hour_of_day	days_since_prior_order
0	2539329	1	prior	1	2	8	NaN
1	2398795	1	prior	2	3	7	15.0
2	473747	1	prior	3	3	12	21.0
3	2254736	1	prior	4	4	7	29.0
4	431534	1	prior	5	4	15	28.0

Data Overview

AISLES.CSV

- + aisle_id: integer in [1:134]
- + aisle: string

DEPARTMENTS.CSV

- + department_id: integer in [1:21]
- + department: string

PRODUCTS.CSV

- + product_id: integer in [1:49688]
- + product_name: string
- + aisle_id: integer
- + department_id: integer

ORDER_PRODUCTS__PRIOR.CSV

- + order_id: integer
- + product_id: integer
- + add_to_cart_order: integer
- + reordered: boolean 0-1

ORDER_PRODUCTS__TRAIN.CSV

- + order_id: integer
- + product_id: integer
- + add_to_cart_order: integer
- + reordered: boolean 0-1

SAMPLE SUBMISSION.CSV

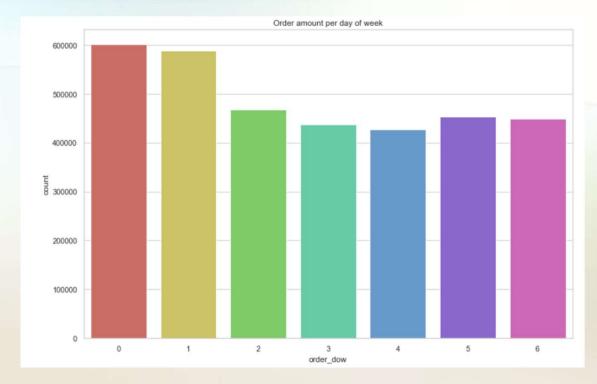
- + order_id: integer
- + product_id: integer

ORDERS.CSV

- + order_id: integer
- + user_id: string
- + eval_set: prior / train / test
- + order_number: integer
- + order_dow: integer in [1:7]
- + order_hour_of_day: integer in [0:23]
- + day_since_prior_order: integer in [0:30] or NA

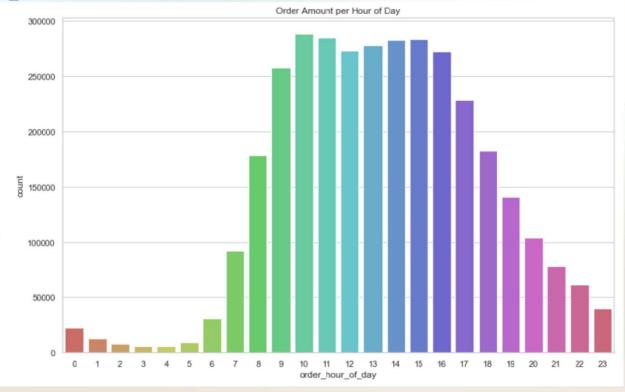
Data exploration

- Orders per day of week
- Similar shopping pattern except for 0 and 1



Data Exploration

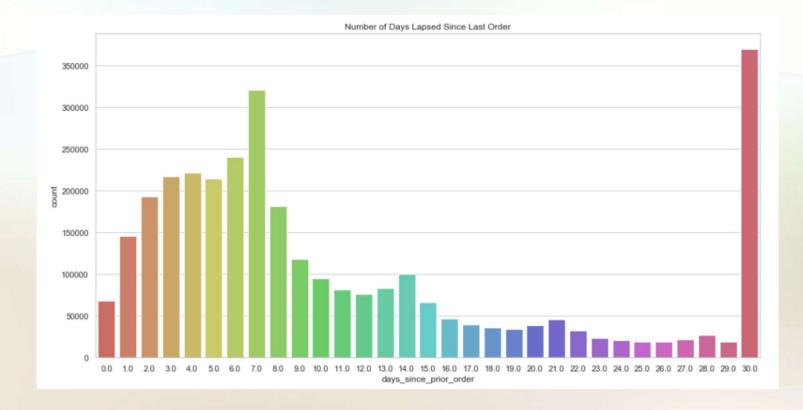
- Orders by hour of day
- Most shopping during the day
 - Significant activity in the evening



Data Exploration

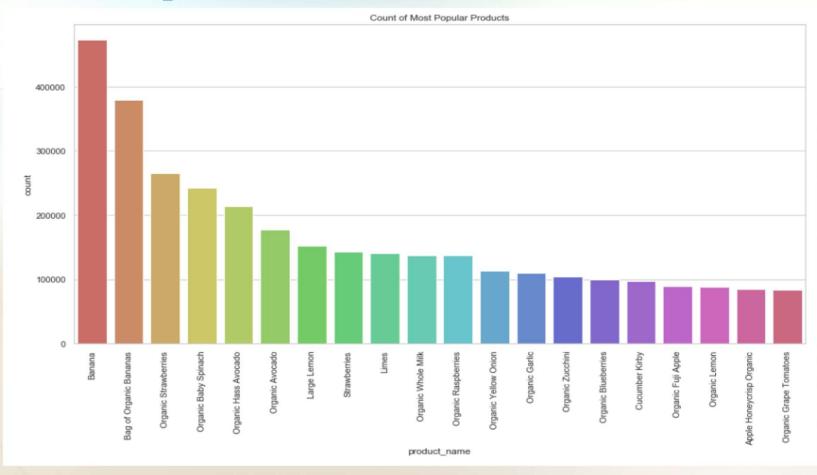
Days between orders

Spikes every 7th day and end of month



Data Exploration

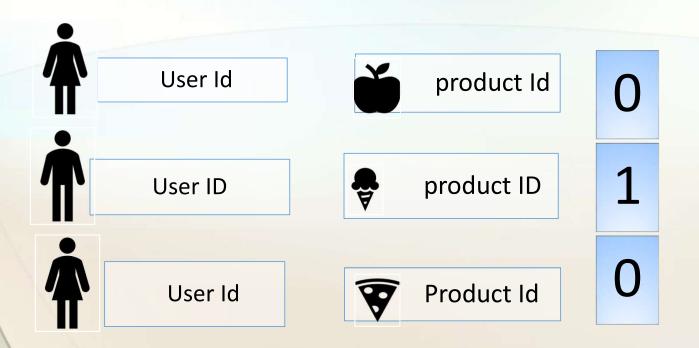
Popular products
Bananas and organics



Problem Approach

- Make a list of all unique products that a user has ordered previously
- The list will be a unique user product pair.
- Use labels to train models
- Train test split for evaluation

Problem Approach



Feature Generation

- Map features from different tables using a function
- Domain knowledge helpful in creating new features
 - Lifestyle
 - Human psychology

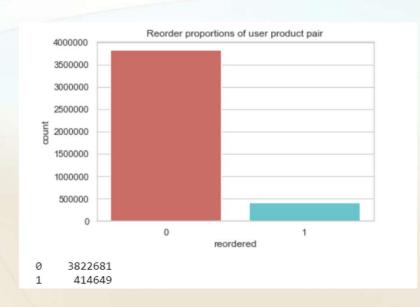
- Self explanatory names
- Some Features
 - Total orders
 - Total items
 - Total distinct items
 - Average days between orders
 - Average basket size

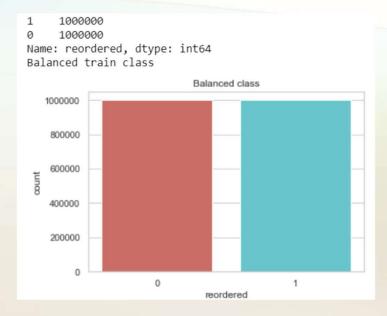
Sampling

- Data set shape
 - 8 474 661 observations
 - 22 features

- We'll use half the data
- Sample data shape
 - 4 237 330 observations
 - 22 features

Class Imbalance





Heat map

									C	orrelatio	ип										
user_total_orders	1	0.8	0.69	-0.67	0.018	-0.021	-0.45	0.0029	-0.0036	0.02	-0.034	-0.034	-0.05	0.31	-0.31	CO19	-0.31	0.55	0.027		
user_total_items	0.8	1	0.84	-0.54	0.47	-0.018	-0.37	-0.013	0.002	0.025	-0.032	-0.032	-0.04	0.33	-0.21	0.28	-0.21	0.41	0.024	-	0.9
total_distinct_items	0.69	084	1	-0.49	0.51	-0.01	-0.33	-0.0013	0.0044	0.027	-0.054	-0.054	-0.1	0.15	-0.32	0.31	-0.32	0.41	0.033		
user_average_days_between_orders	-0.67	-0.54	-0.49	1	0.016	0.023	0.63	-0 056	0.0023	-0.014	0.028	0027	0.044	-0.23	0.24	-0.013	0.24	-0.36	-0.021		
user_average_basket	0.018	047	0.51	-0.016	1	-0.0042	0.026	-0 025	0.012	0.017	-0.017	-0.018	-0.015	0.1	0.058	0.58	0.068	-0.032	0.0063	-	0.6
order_hour_of_cay	-0.021	-0.018	-0.01	0.023	-0.0042	1	-0.0C23	-0 026	0.0008	-0.0059	0.00017	-0.00039	-0.008	-0.018	-0.003	-0.0014	-0.003	-0.0067	0.029		
days_since_prior_order	-0.45	-0.37	-0.33	063	0.026	-0.0023	1	C67	0.0047	-0.006	0.013	0013	0.023	-0.15	0.15	-0.021	0.15	0.24	0.0043		
days_since_ratio	0.0029	-0.013	-0.0013	-0.056	0.025	-0.026	0.67	1	0.0042	0.0033	-0.0066	-0.0062	-0.0077	0.00088	-0.016	-0.018	-0.016	0.0086	0.012		
aisle_id	-0.0036	0.002	0.0044	0.0023	0.012	0.0008	0.0047	0.0042	1	0.05	-0.2	0.2	0.021	-0.0017	0.0012	0.0098	0.0012	-0.007	0.00013		0.3
department_id	0.02	0.025	0.027	-0.014	0.017	-0.0059	0.006	0.0033	0.05	1	-0.29	-0.27	-0.15	-0.032	-0.052	0.03	-0.052	0.035	-0.0049		
product_orders	-0.034	-0.032	0.054	0.028	-0.017	-0.00017	0.013	-0.0065	-0.2	-0.29	-1	1	0.45	0.17	0.22	-0.098	0.22	-0.099	0.0021		
product_reorders	-0.034	-0.032	0.054	0.027	-0.018	-0.00039	D.013	-0.0062	-0.2	-0.27	1	1	0.44	0.17	0.22	-0.1	0.22	-0.098	0.0016	-	0.0
product_reorder_rate	-0.05	-0.04	-0.1	0.044	0.015	800.0-	0.023	-0.0077	0.021	-0.15	0.45	0.44	1	0.27	0.34	-0.13	0.34	-0.17	-0.006		
UP_orders	0.31	033	0.15	-0.23	0.1	-0.018	-0.15	0.00088	-0.0017	-0.032	0.17	0.17	0.27	1	0.55	-0.12	0.55	-0.16	-0.0077		
UP_orders_ratio	-0.31	-0.21	-0.32	024	0.068	-0.003	0.15	-0.016	0.0012	-0.052	0.22	0.22	0.34	0.55	1	-0.19	1	0.4	-0.027		-0.3
UP_average_pos_in_cart	0.019	028	0.31	-0.013	0.53	-0.0014	0.021	-0018	0.0098	0.03	-0.098	-0.1	-0.13	-0.12	-0.19	1	-0.19	0.034	0.0082		0.0
UP_reorder_rate	-0.31	-0.21	-0.32	024	0.068	-0.003	0.15	-0.016	0.0012	-0.052	0.22	0.22	0.34	0.55	1	-0.19	1	0.4	-0.027		
UP_orders_since_last	0.55	041	0.41	-0.36	0.032	-0.0067	-0.24	0.0086	0.007	0.035	-0.099	-0.098	-0.17	-0.16	-04	0.034	-0.4	1	0.031		
UP_delta_hour_vs_last	0.027	0.024	0.033	-0.021	0.0063	0.029	-0.0C43	0.012	-0.00013	-0.0049	0.0021	0.0016	-0.006	-0.0077	-0.027	0.0082	-0.027	0.031	1	-	-0.6
	user total orders	user_total_items	total_distinct_items	days_between_orders	user average basket	order_hour_of_day	days since prior order	days_since_ratio	aisle_id	department_id	product_orders	product_reorders	product_reorder_rate	UP_orders	UP_orders_ratio	P_average_pos_in_cart	UP_reorder_rate	UP_orders_since_last	UP_delta_hour_vs_last		
				26.												_					

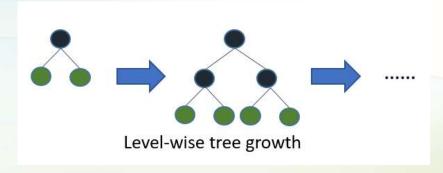
Prediction models

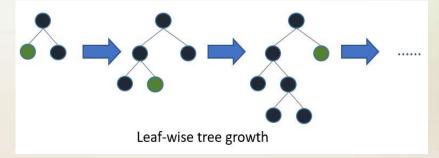
- Gradient Boosting with Light
 GBM Model
- KNN Classifier
- Logistic Regression
- K-Means clustering
- Multi Layer Perceptrons (MLP)
- Convolutional Neural Networ (CNN)

- Dimensionality reduction
 - PCA
 - SelectKBest
- Parameter tuning
 - GridSearchCV
 - Manual Tuning

Light GBM

- Histogram based algorithm
- Grows tree vertically (leafwise)
- Other algorithm grows trees horizontally (level-wise)





Dimensionality reduction

- PCA
 - 12 components
 - 90% variance retained
 - Reduces overfitting

- SelectKBest
 - 13 features
 - Low correlation b/n variables means PCA and SelectKBest less likely to improve the results

Light GBM

- Best score
- Accuracy: 0.73714

Confusion [[148396 [5813	1519	03]			
		precision	recall	f1-score	support
	0	0.96	0.49	0.65	300299
	1	0.66	0.98	0.79	299701
micro	avg	0.74	0.74	0.74	600000
macro	avg	0.81	0.74	0.72	600000
weighted	avg	0.81	0.74	0.72	600000

KNN Classifier

- Distance between dtata points
- Prone to overfitting
- Works best with
 - Linear relationship b/n variables
 - Uniform measurement types

KNN Classifier

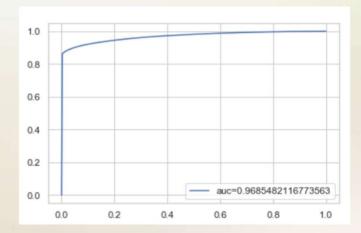
Best score (with PCA)

Training set score: 1.0

• Test set score: 0.839235

Runtime: 9353.14 seconds

Training set	score: 1.0			
Test set scor	e: 0.839235 precision	recall	f1-score	support
0	0.95	0.72	0.82	199880
1	0.77	0.96	0.86	200120
micro avg	0.84	0.84	0.84	400000
macro avg	0.86	0.84	0.84	400000
weighted avg	0.86	0.84	0.84	400000
Confusion mat [[143903 559 [8329 1917	77]			



Logistic Regression

Assumes independence b/n features

With GridSearchCV

Regularization parameter

Logistic Regression

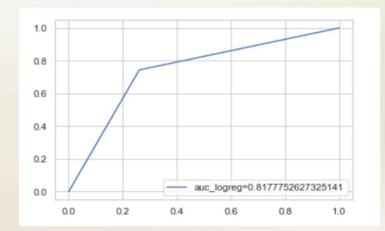
Best score

Training set score: 0.74064875

Test set score: 0.741195

Runtime: 3759.31 seconds

Confusion [[221975 [76959	783	24]			
2		precision	recall	f1-score	support
	0	0.74	0.74	0.74	300299
	1	0.74	0.74	0.74	299701
micro	avg	0.74	0.74	0.74	600000
macro	avg	0.74	0.74	0.74	600000
weighted	avg	0.74	0.74	0.74	600000

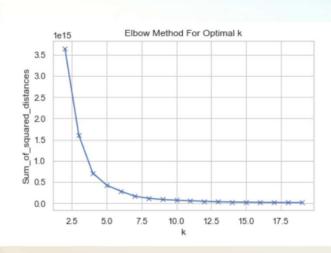


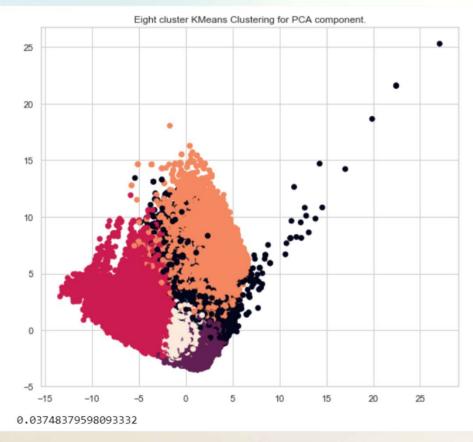
Predicting with K-means clusters

- Assumptions
 - Uniform varience
 - Equal number of observations
 - Cost function equally relevant

- Fails with
 - Non continuous features
 - Skewed distribution

Feature creation with K-means





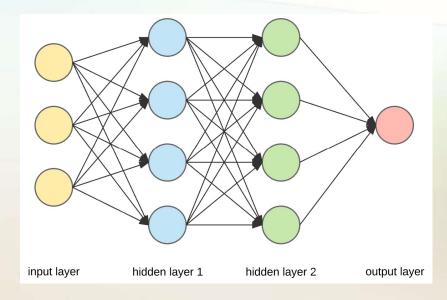
Feature creation with K-means

 Clusters as supervised features

	8	9	10	11	cluster
73	0.263030	0.907111	-1.260193	0.190990	4
753	-0.732012	0.263739	-0.434459	-0.255671	4
75	-0.986117	-1.125017	0.206761	-0.749581	4
301	1.101414	1.788032	-0.329314	0.535244	1
722	-1.522040	0.310858	-1.071489	0.410851	1

Multi Layer Perceptron (MLP)

- Using Sci-Kit learn
- Five layers
- Number of neurons in each
 - 64, 64, 32, 16, 2
- Non-linear input transformation
- ...into a space where it becomes linearly separable.



Multi Layer Perceptron (MLP)

Score

- Train set score 0.753816875
- Eval set score 0.7531925
- Runtime: 627.09 seconds

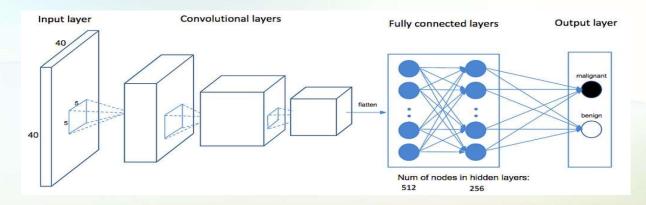
		precision	recall	f1-score	support
	0	0.75	0.75	0.75	199880
	1	0.75	0.76	0.75	200120
micro	avg	0.75	0.75	0.75	400000
macro		0.75	0.75	0.75	400000
weighted		0.75	0.75	0.75	400000

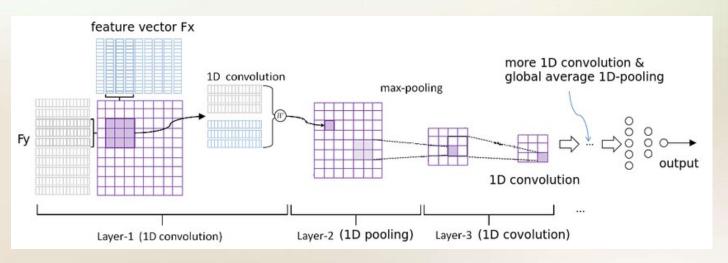
```
print(confusion_matrix(y_eval_nn, predictions))

[[149774 50106]
  [ 48617 151503]]
```

Convolutional Neural Networks (CNN)

- Works well with
 - Spatial data
 - Sequential data





Convolutional Neural Networks (CNN)

Score

• Test loss: 0.5522323341655732

Test accuracy: 0.74641

Runtime: 62472.04 seconds

Evaluation summary

Model	Train score	Eval score	Precision	Recall	F1- score	Auc	Run time (sec)
Light GBM		73	81	74	72		277
Light GBM (PCA)		70	79	71	68		174
KNN	100	79	84	79	79	97	517
KNN (PCA)	100	83	86	84	84	96	9353
Logistic Regression	74	74	74	74	74	81	3759
Logistic Regression (PCA)	72	72	73	73	73	81	53
Logistic Regression (SelectKBest)	74	74	74	74	74	81	1208
Logistic Regression (Kmeans)	72	73	73	73	73	81	119
MLP	75	75	75	75	75		627
CNN		74					62472

Conclusion and Recommendation

- Good result using CNN for tabular data
- Most models performed well
- MLP is the best model to choose
 - Learns better with non linear transformation
 - No overfitting

- More indicators needed
 - User level data
 - Location
 - Store name

Conclusion and Recommendation

- Models can be improved with
 - More data
 - More tuning
 - Better features

- Device used
 - Microsoft Windows 10
 - Intel Core i7 8th generation 1.90GHz, 1.8GHz
 - Python 3.6

Thank you to my mentor Hoa Tran

Thank you!

