Market Basket Prediction

Predicting which products will Instacart customers buy again¶

Solomon Mengistu

Outline

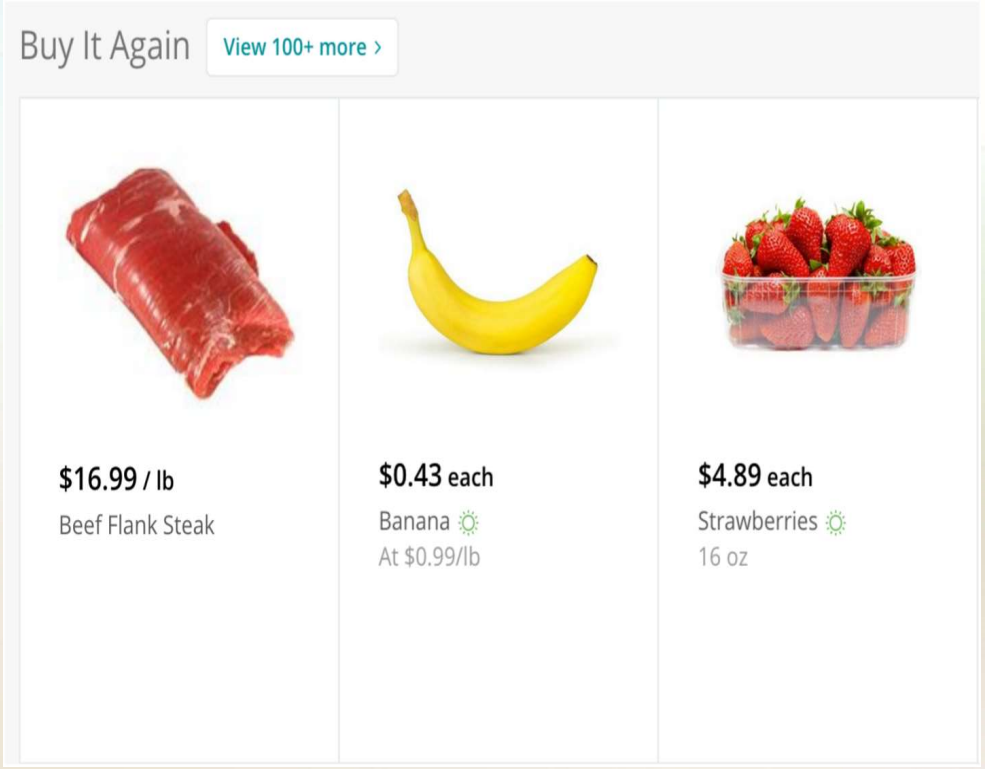
- Introduction
- Data overview
- Data exploration with visualization
- Feature engineering and data preparation
- Prediction models
- Dimensionality reduction
  - PCA
  - SelectKBest
- Neural Networks
- Evaluation
- Conclusion and recommendations

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
0	1	prepared soups salads
1	2	specialty cheeses
2	3	energy granola bars
3	4	instant foods
4	5	marinades meat prepar

Departments shape: (21, 2)

	department_id	department
0	1	frozen
1	2	other
2	3	bakery
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

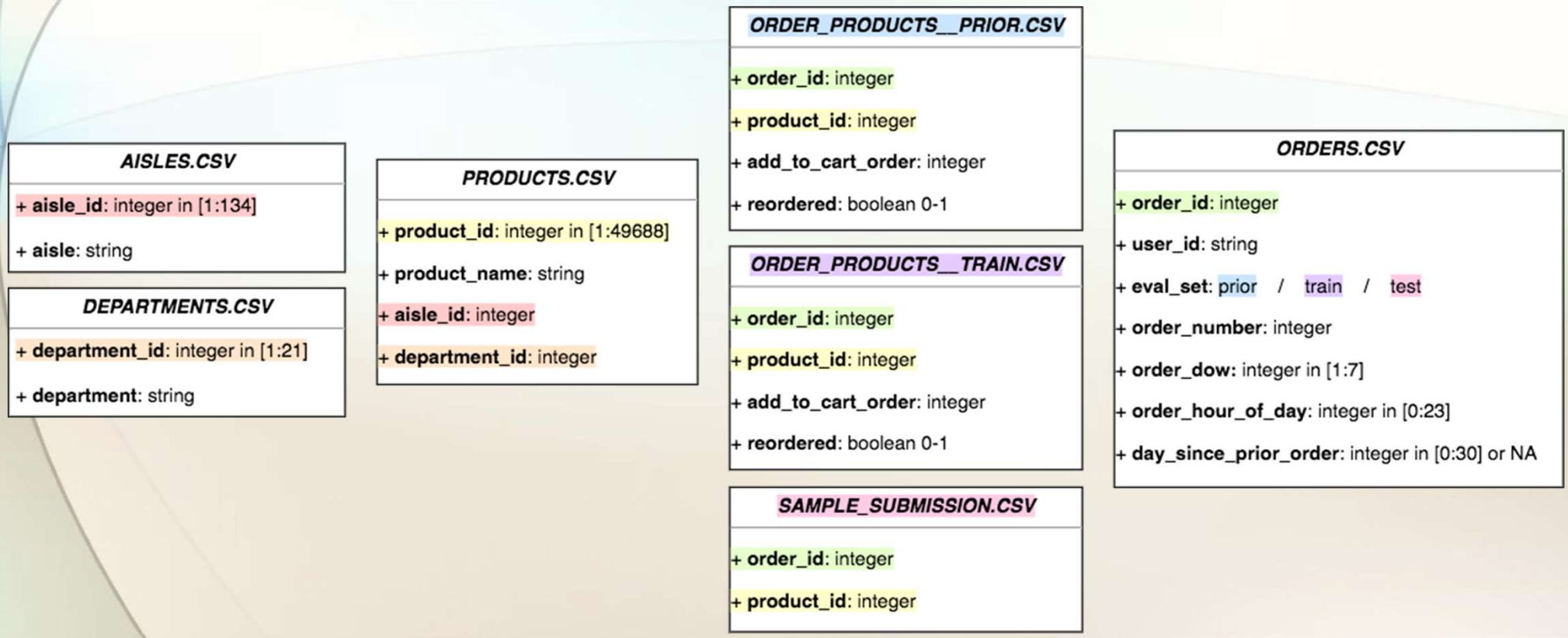
Order train shape: (1384617, 4)

	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)

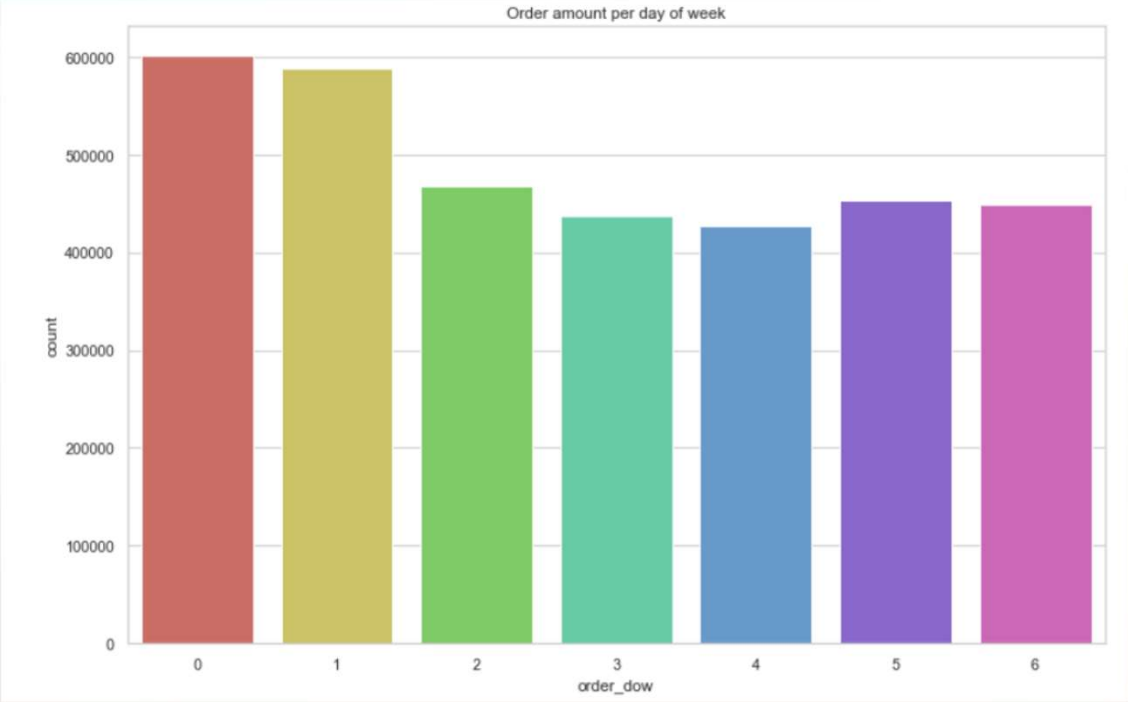
	order_id	user_id	eval_set	order_number	order_dow	order_hour_of_day	days_since_prior_order
0	2539329	1	prior	1	2	8	NaN
1	2398795	1	prior	2	3	7	15.0
2	473747	1	prior	3	3	12	21.0
3	2254736	1	prior	4	4	7	29.0
4	431534	1	prior	5	4	15	28.0

Data Overview



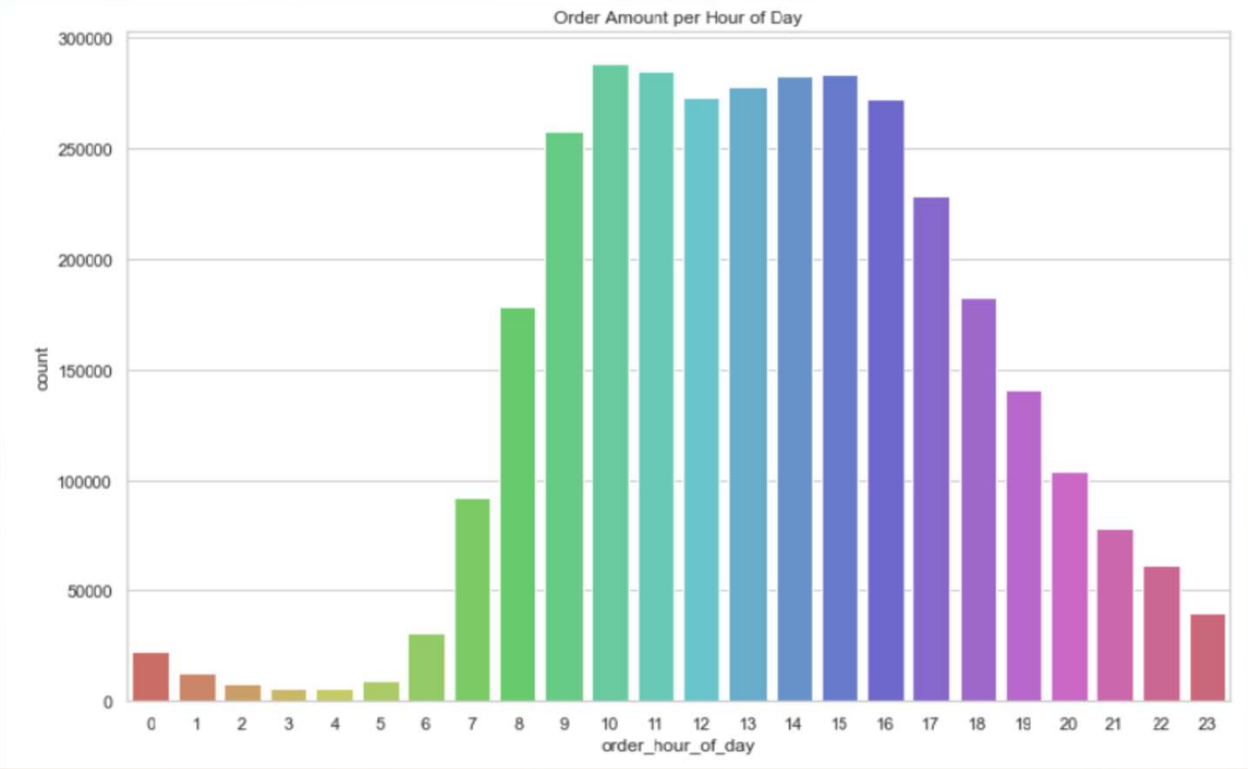
Data exploration

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



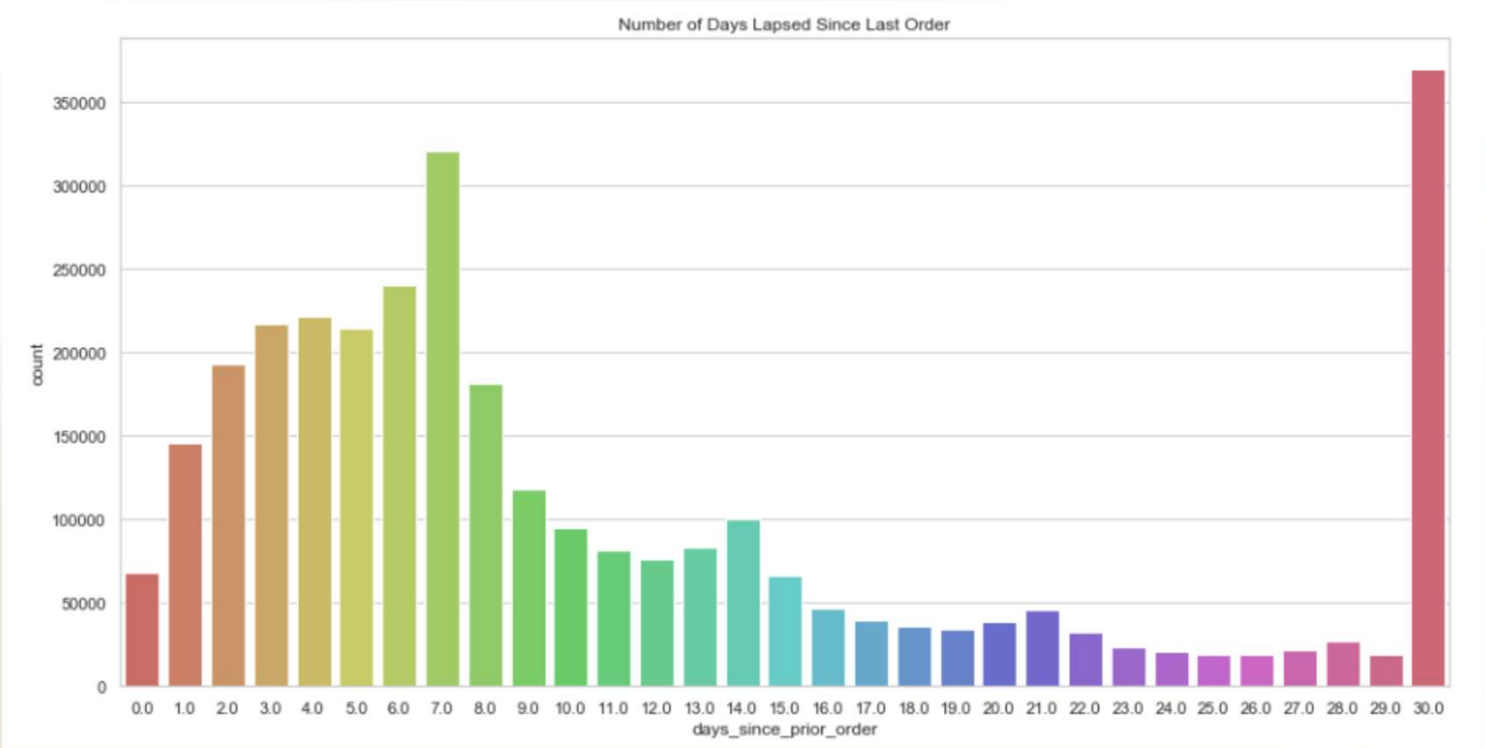
Data Exploration

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



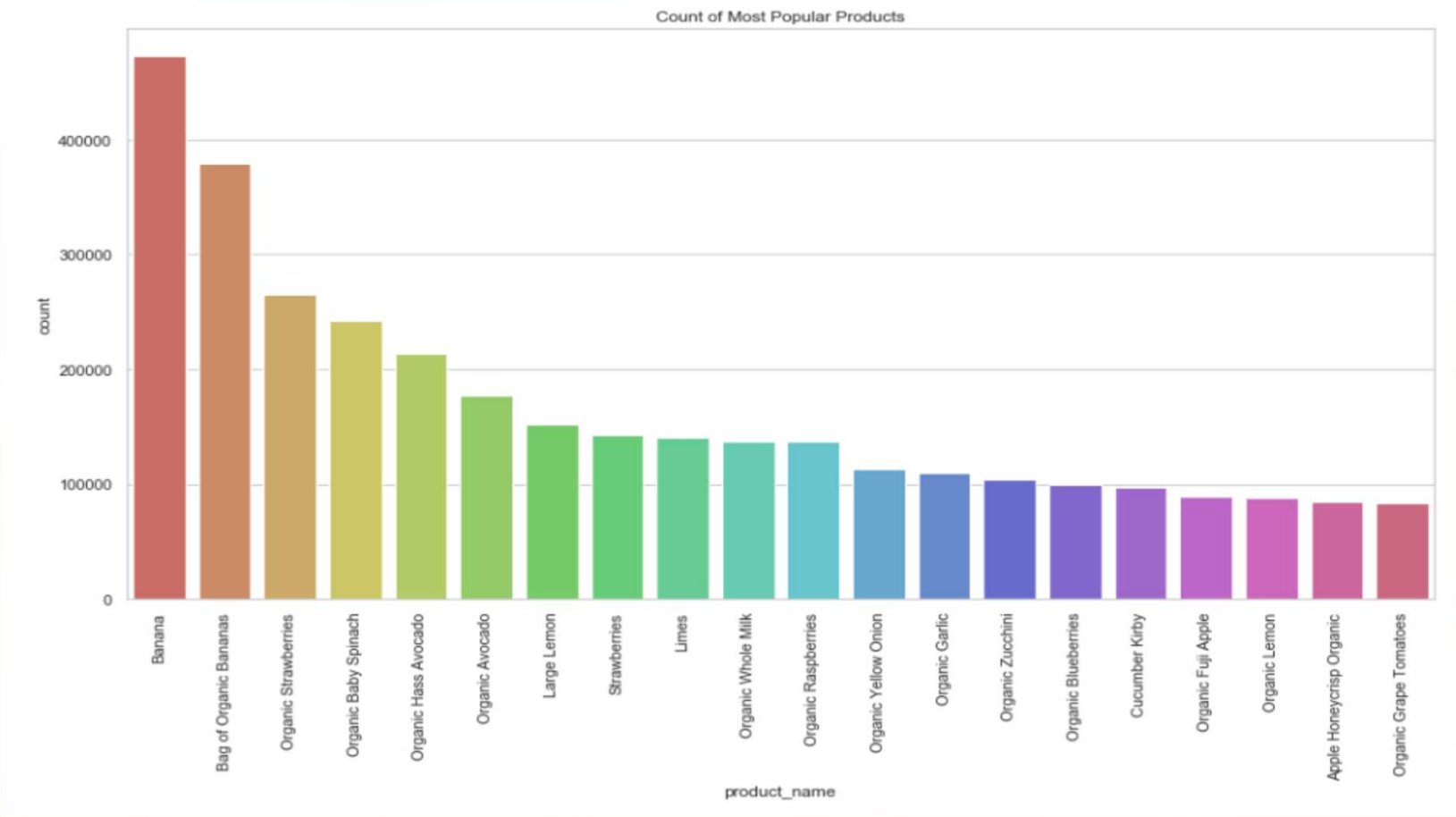
Data Exploration

- Days between orders
- Spikes every 7th day and end of month



Data Exploration

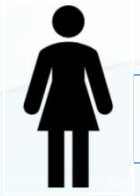
- Popular products
- Bananas and organics



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



User Id

product Id

0

User ID

product ID

1

User Id

Product Id

0

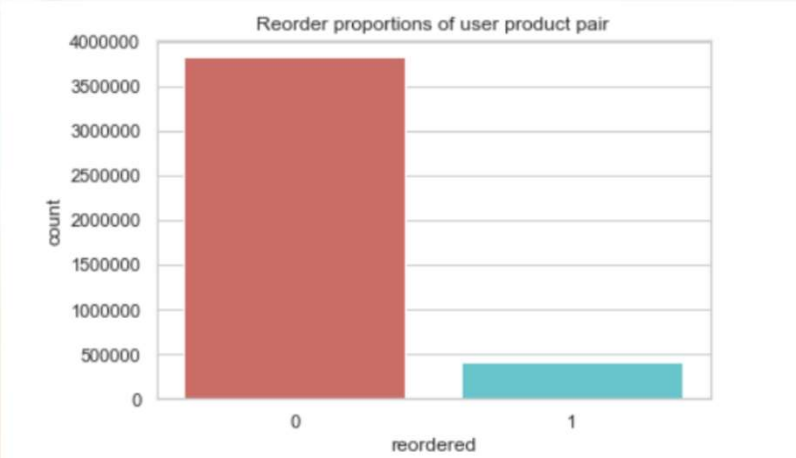
Feature Generation

- 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

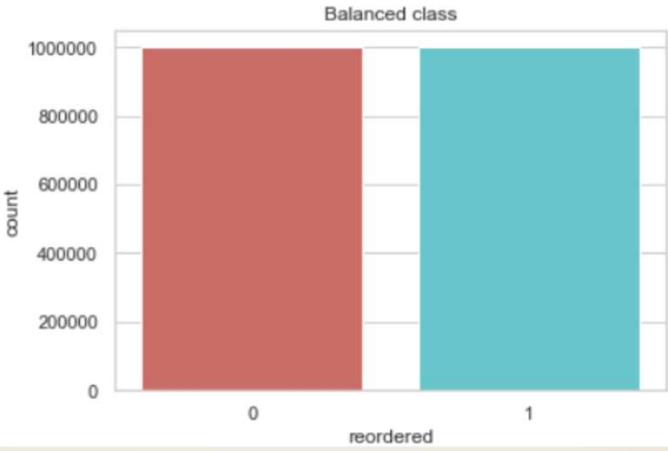
- Data set shape
  - 8 474 661 observations
  - 22 features
- We'll use half the data
- Sample data shape
  - 4 237 330 observations
  - 22 features

Class Imbalance

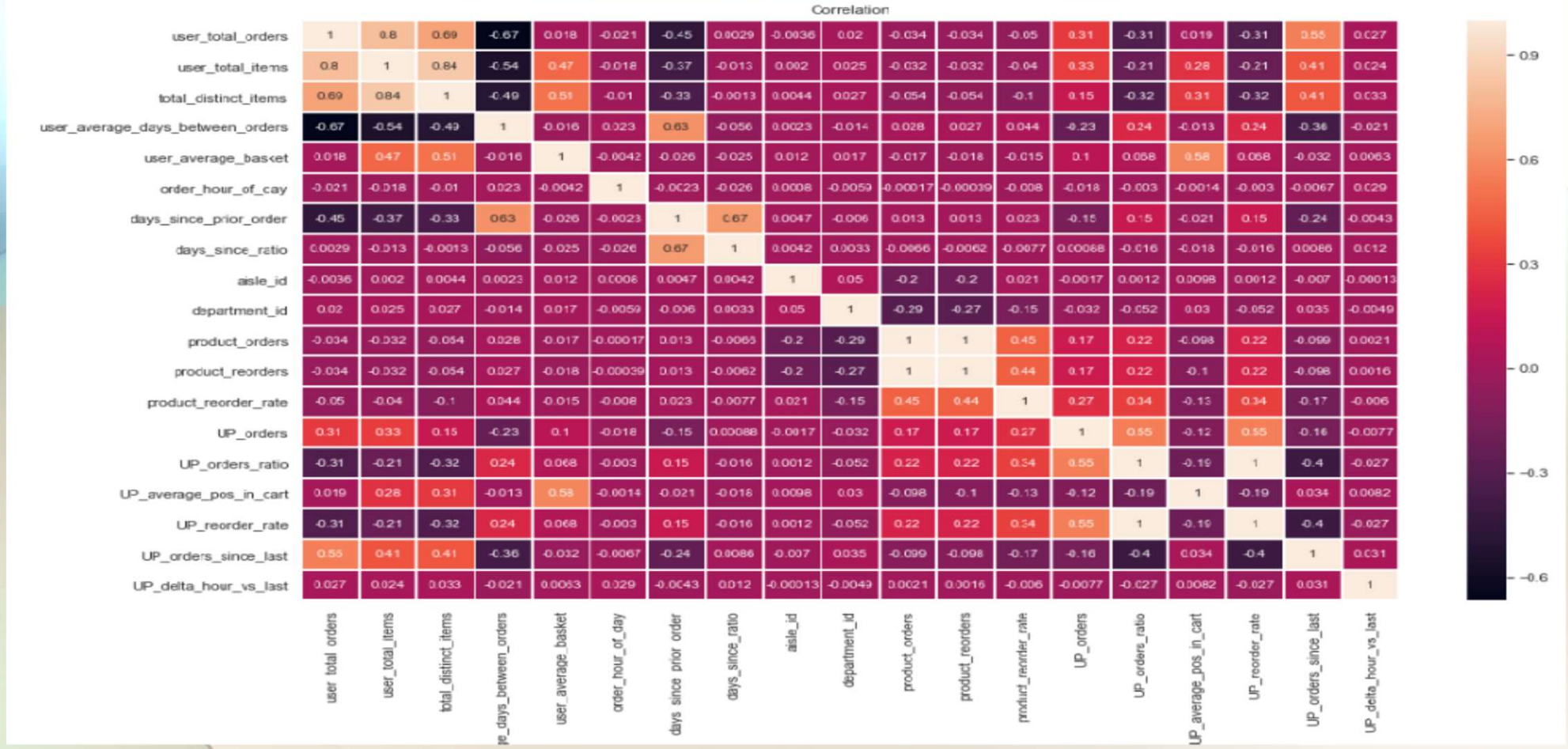



```
0    3822681
1     414649
```

```
1    1000000
0    1000000
Name: reordered, dtype: int64
Balanced train class
```



Heat map


Correlation
user_total_orders
user_total_items
total_distinct_items
user_average_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
UP_average_pos_in_cart
UP_reorder_rate
UP_orders_since_last
UP_delta_hour_vs_last
0.9
0.6
0.3
0.0
−0.3
−0.6

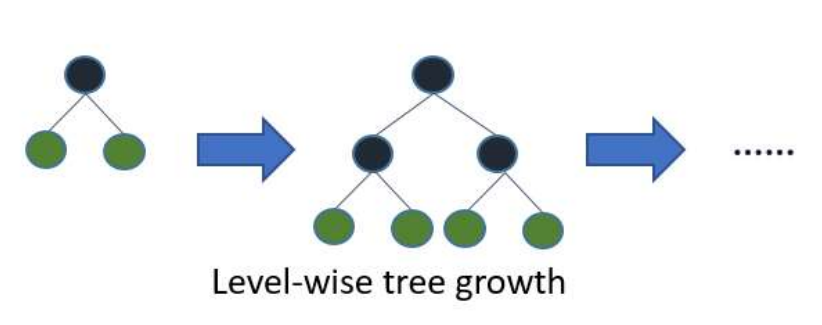
Prediction models

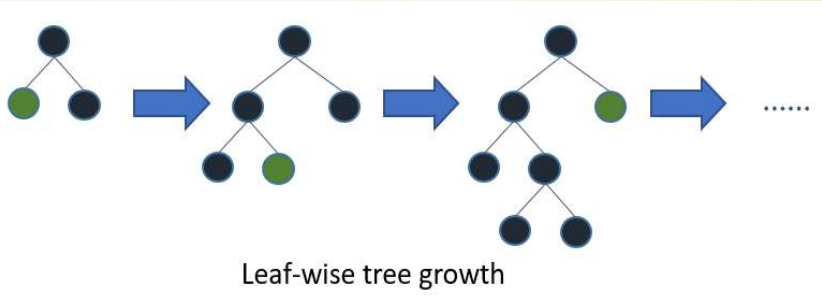
- **Gradient Boosting with Light GBM Model**
- **KNN Classifier**
- **Logistic Regression**
- **K-Means clustering**
- **Multi Layer Perceptrons (MLP)**
- **Convolutional Neural Networ (CNN)**

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

Light GBM

- Histogram based algorithm
- Grows tree vertically (leaf-wise)
- 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 matrix
[[148396 151903]
 [  5813 293888]]
              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
```

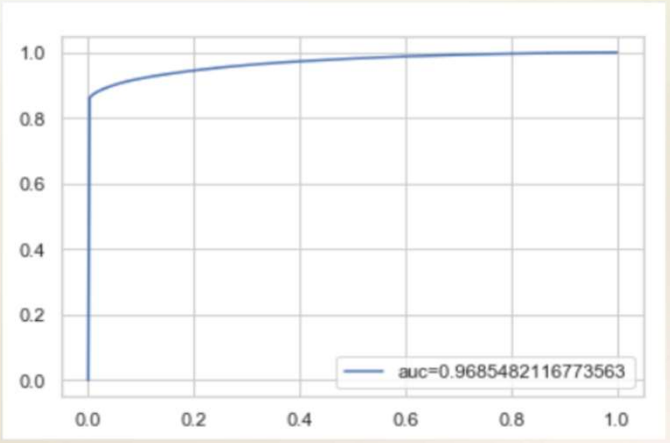
KNN Classifier

- 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 score: 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 matrix
[[143903  55977]
 [  8329 191791]]
```



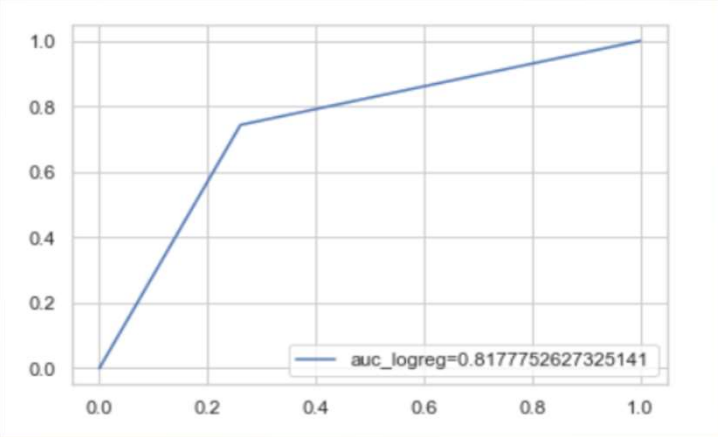
Logistic Regression

- 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

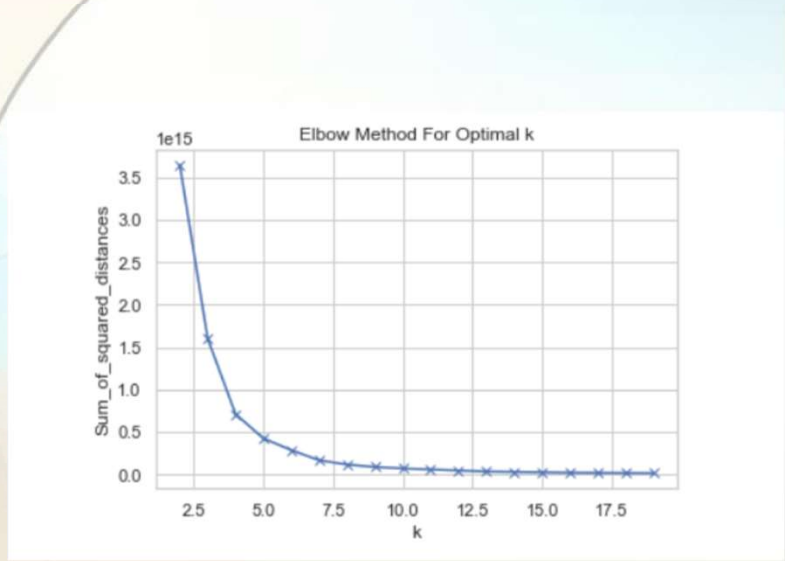
```
Confusion matrix
[[221975  78324]
 [ 76959 222742]]
              precision    recall  f1-score   support

           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
```

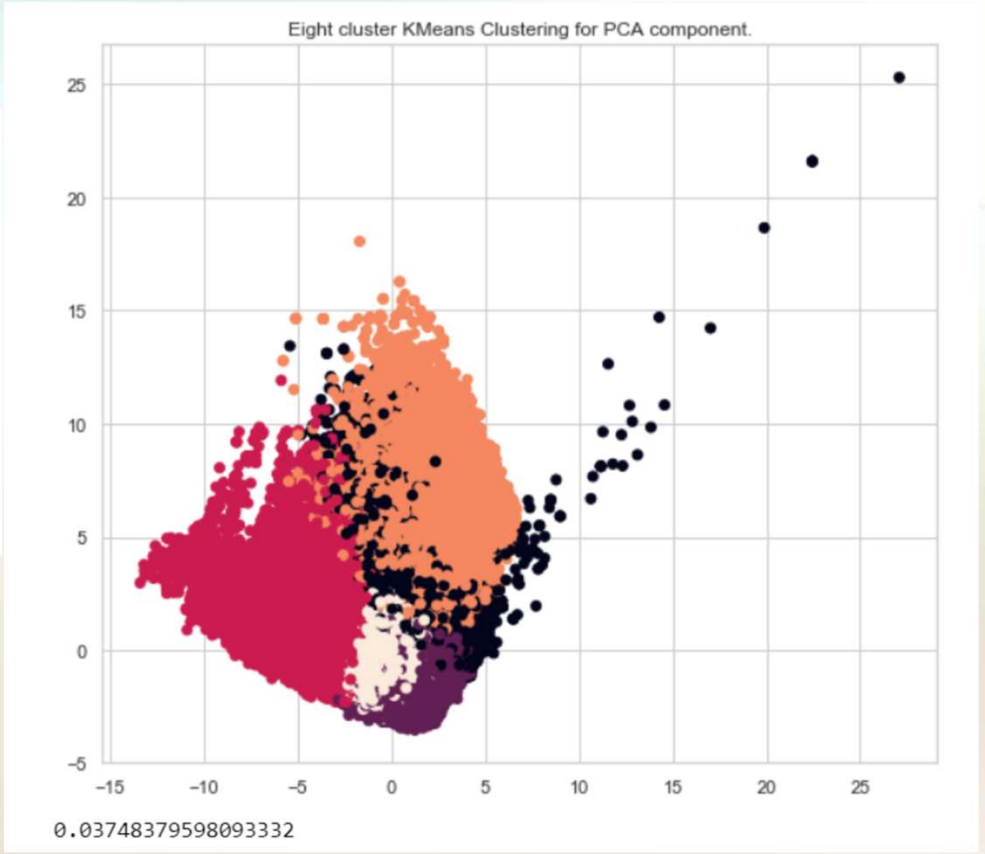


Predicting with K-means clusters

- Assumptions
  - Uniform varience
  - Equal number of observations
  - Cost function equally relevant
- Fails with
  - Non continuous features
  - Skewed distribution

Feature creation with K-means





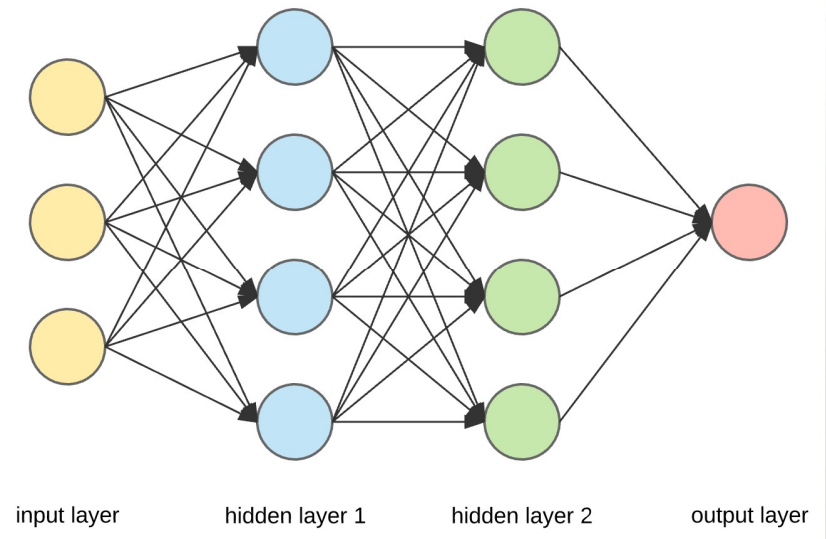
Feature creation with K-means

- 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

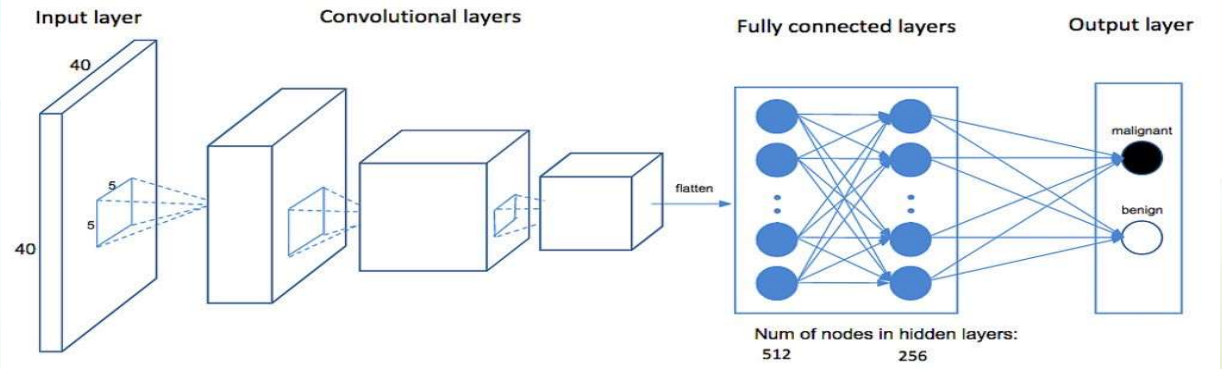
	precision	recall	f1-score	support
0	0.75	0.75	0.75	199880
1	0.75	0.76	0.75	200120
micro avg	0.75	0.75	0.75	400000
macro avg	0.75	0.75	0.75	400000
weighted avg	0.75	0.75	0.75	400000

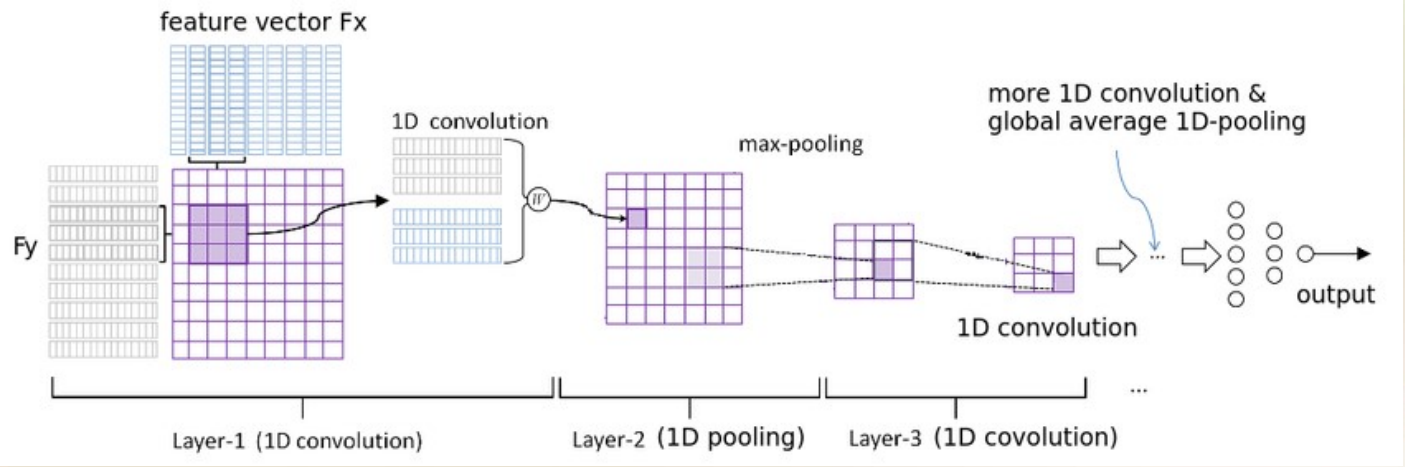
```
print(confusion_matrix(y_eval_nn, predictions))
```

```
[[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!

Questions?!