



MARKET BASKET ANALYSIS

Project Report
BUAN 6340: Programming for Data Science

ABSTRACT

Market Basket Analysis is a modeling technique based upon the theory that if you buy a certain group of items, you are more (or less) likely to buy another group of items. In this analysis, a forecasting model is developed using machine learning algorithms to improve the accurate forecasts of product sales.

PREPARED BY:

Group 1

ATHARVA TIPRE
AVINASH PANIGRAHI
SIDHANTH KAPOOR
VEDANTA TANEJA

Table of Contents

Goal	2
Project Introduction	2
Dataset Description	2
Step by Step Analysis:.....	3
Data Cleaning and Wrangling	3
Exploratory Data Analysis.....	3
Merging and Creating Final Dataset	7
Modelling Fitting and Prediction.....	8
Xgboost.....	8
Logistic Regression:	8
Random Forest	8
Light Gbm.....	8
Recommendations	8
Future Work.....	9

Goal

This project aims to predict the products which have been purchased previously and the user is most likely to buy the same products in their next order.

Project Introduction

Instacart is a grocery ordering and delivery application, by serving the customers to order groceries from participating retailers like Sam's Club, Kroger, Aldi, etc. with the shopping being done by a personal shopper. For this project, we are using the anonymized data of customer orders on Instacart, which they have open-sourced on Kaggle.

Here we are interested to predict the following –

- Which previously purchased products will be in a user's next order?
- Which product the user will try for the first time?
- What are the patterns in user consumption?
- What factors influence the purchasing decision (time of the day, day of the week)

Instacart essentially falls in the e-commerce industry, hence creating an efficient and smooth experience for a user would be a priority given the fact that every company tries their best for the same. If we try to find the solution to the questions mentioned above, it will help Instacart develop a systematic user experience and could be their USP (Unique Selling Proposition)

Dataset Description

The dataset is anonymized and contains a sample of over 3 million grocery orders from more than 200,000 Instacart users. For each user, the orders are provided between 4 and 100, with the sequence of products purchased in each order. It also contains the week and hour of the day the order was placed and a relative measure of time between orders.

Link for the data. (<https://www.kaggle.com/c/instacart-market-basket-analysis/data>)

There are six data sets in this project:

- 1) **Aisles.csv**: has 2 columns aisle_id and aisles
- 2) **Department.csv**: has 2 columns again department_id and department
- 3) **Order_products_prior.csv**: has 4 columns, order_id, product_id, add_to_cart_order and reordered.
- 4) **Order_products_train.csv**: has 4 columns same as order_products_prior
- 5) **Orders.csv**: has 8 columns, order_id, user_id, eval_set, order_number, order_dow, order_hour_of_day, days_since_prior_order.
- 6) **Products.csv**: has 4 columns, product_id, product_name, aisle_id, department_id.

Step by Step Analysis:

Data Cleaning and Wrangling

The first step to any Machine Learning problem lies in the efficient process of preprocessing the data. We analyzed the data set to identify features with missing values. We used `isnull()` function to check for null values and fetched the percentage of missing values in each data set.

	Total	Percentage
order_id	0	0.000000
user_id	0	0.000000
eval_set	0	0.000000
order_number	0	0.000000
order_dow	0	0.000000
order_hour_of_day	0	0.000000
days since prior order	206209	0.060276

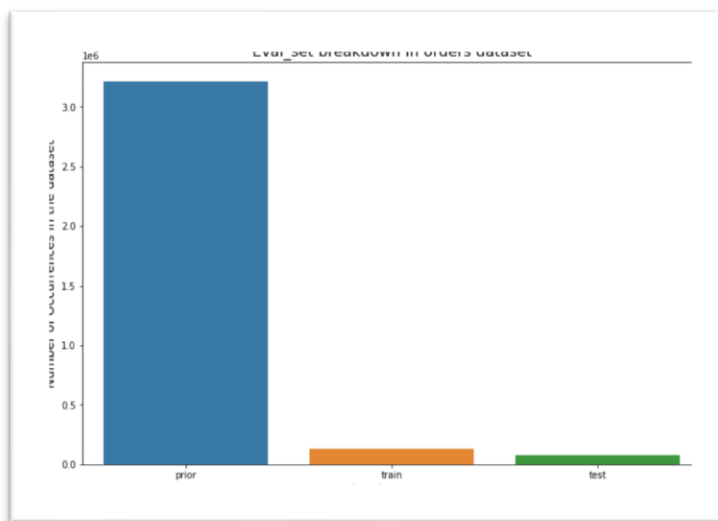
fig(a) This is table of containing percentage of missing values of the **product** table.

To find the missing values we took the **total** for each data set using `.isnull()` and `.sum()`. We calculated the percentage of the null values to have a better picture of whether we should eliminate the feature or impute with any statistical metrics.

We found null values only in the **orders** dataset in the feature `days_since_prior_order`. And since the percentage of the null values is less than 6% we went ahead and dropped the rows.

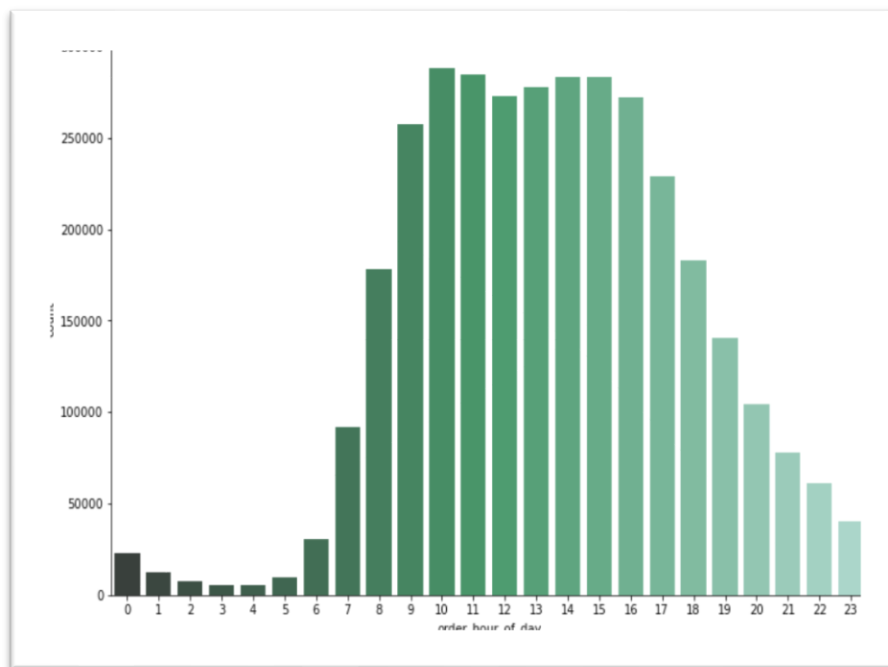
Exploratory Data Analysis

We found out the breakdown of the `eval_set` in prior, train, test, and then plotted them to get a perspective about the distribution.

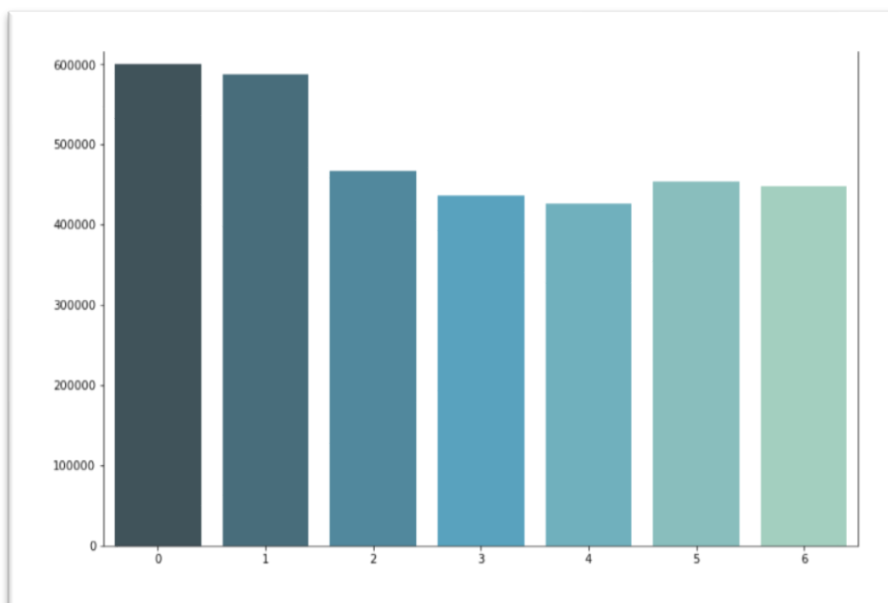


Fig(b) Breakdown of the `eval_set` in all three data sets

Post that we went ahead to find the distribution of products against the hour of the day and then against the day of the week. We concluded that the number of orders is **higher on Sunday and Monday** which makes sense since people want to shop for groceries either at the start of the week or on the weekend. On the other hand, **it's the least in the mid-week which is Thursday followed by Wednesday**. Now to know more, let's look at the orders for the hours on a given day of the week.

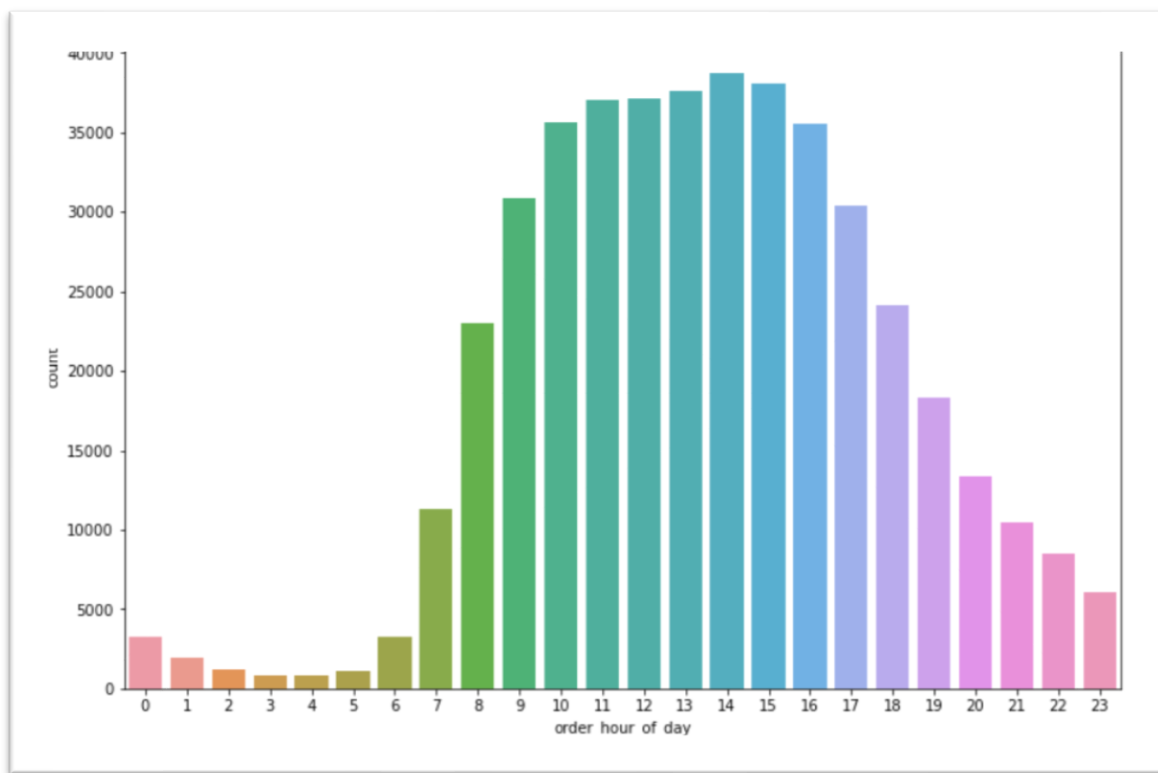


Fig(c) Distribution of products against the hour of the day



Fig(d) Distribution of products against the days of the week.

We performed visualization on the distribution of the number of products against each hour of the day for a week. Below is the plot of *Saturday*.



Fig(e) Distribution of products against all the hours of Saturday

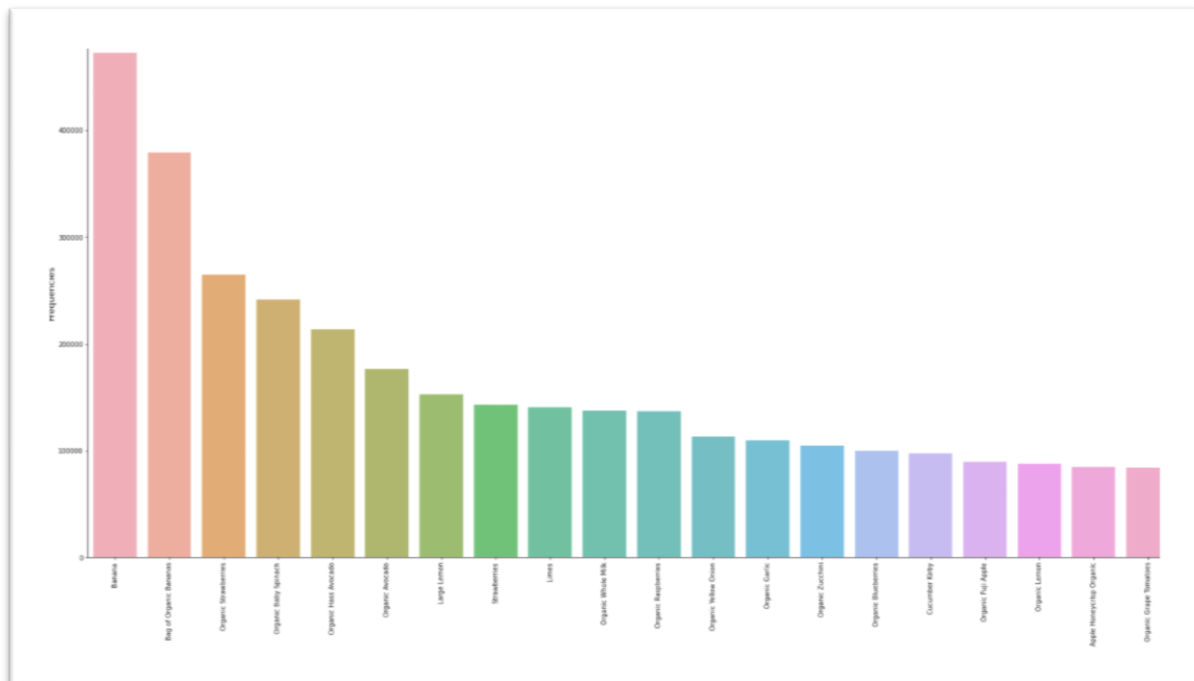
The inference we got from all the plots of all days is that the distribution follows the same pattern for all the weekdays. There is a difference in the peak hours of weekdays and weekends. For Saturday from the graph above we can see that peak hour is around the afternoon whereas for weekdays we found that to be around 10-11 in the morning. This deduction was conclusive because it gave us an idea about the traffic density under respective days and the possible average count of orders.

We generated a heat map to give us a better visualization of the orders in terms of hours of the day and day of the week in a single dataset. We found that peak orders are in the afternoon on Sunday and Monday, from 9 AM-4 PM.

We wanted to see if there is a definite pattern for the number of orders in a month, so we plotted an aggregated count of orders on each day of a month. we saw that the 7th day is where we have a spike, and then a relatively small peak at days **14, 21, and 28** which indicates that the frequency of order is every **7** days. And then again there's a huge peak at the end of the month indicating that there's a monthly peak.

Our next focus was to understand which products were ordered the most and have a department wise frequency. The visualization showed us that fruits like bananas, strawberries and organic

products. The fresh food and fresh vegetables aisles are the most frequently visited. Department wise frequency is most for produce and dairy eggs.



Fig(f) Top 20 Ordered Products

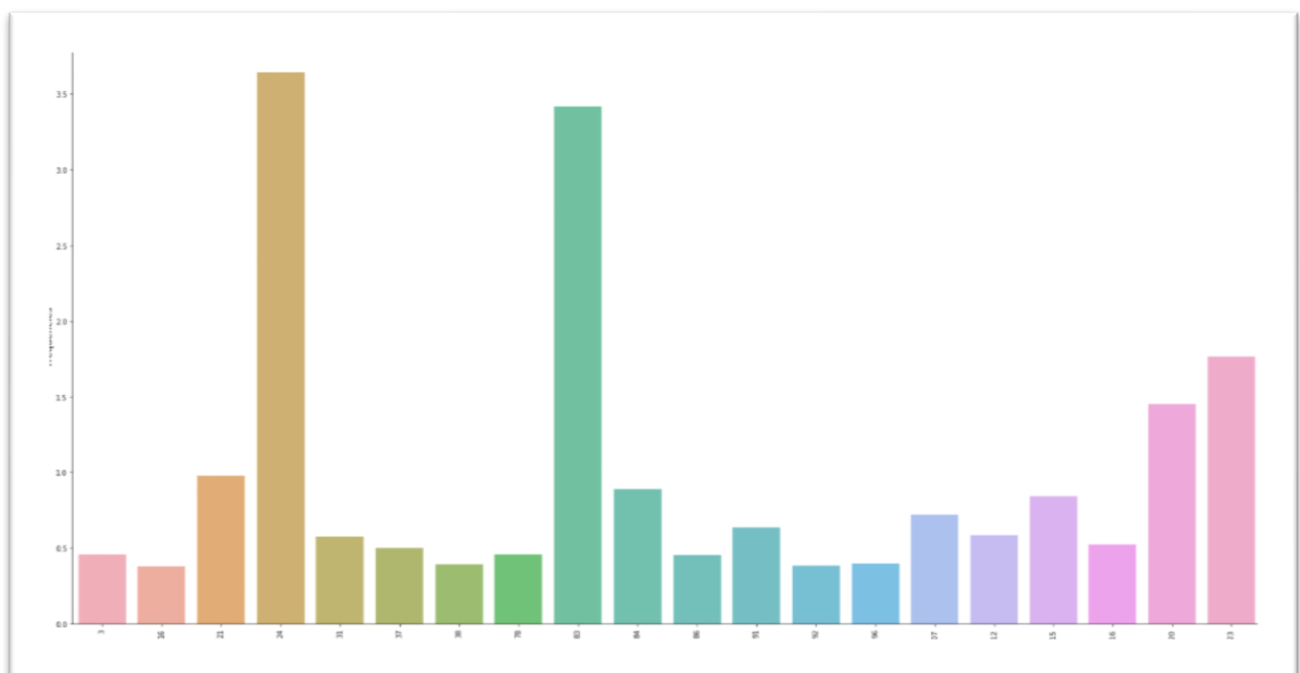
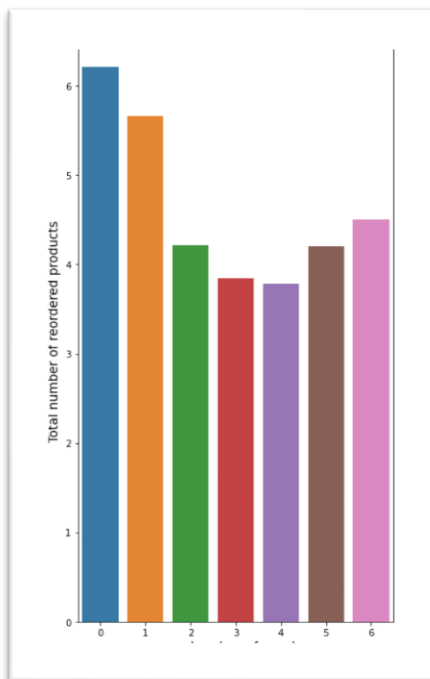


Fig (g) Top 20 ordered Aisles

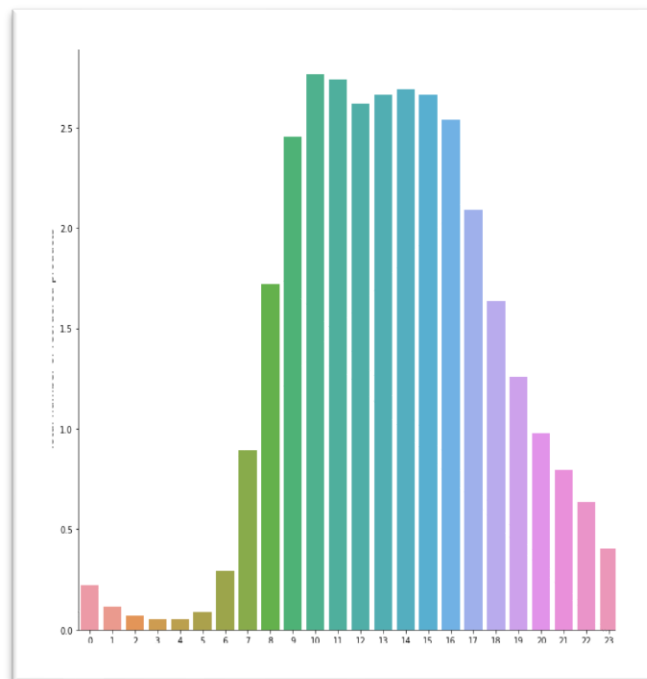
Furthermore, we combined the data sources *order_products_prior*, *orders* to find out the aggregated count of total number of reorders that has occurred in a week. From the graph (Fig(h)) we could see that most products are reordered on Sunday followed by Monday and Saturday and that follows the same trend as orders placed over the week.

We tried to perform the same comparison by adding a level of granularity to the reorders by the hour of the day. It showed that most products are reordered from 10-11AM followed by 1-3pm. This aligns with the number of products ordered during the week and the weekends.



Fig(h)

Total number of reorders in a week



Fig(i)

Total reorders in each hour of the day

Merging and Creating Final Dataset

We combined the datasets required to create the final training and test datasets. We did that by first combining all the products purchased by the customer. Next, we create another dataset that has the customer's reorders and reorder status. Aggregating based on the customer and product we get the count of how many times a user has ordered a certain item. Succeeding step was to merge the training data frame, test data frame and the newly created data frame that has the aggregated count of each product for each user.

Additionally, we performed a sanity check to see if there are any nulls or duplicate features. Once we are done with the sanity check we split the dataset into train, test and went ahead to perform modelling

Modelling Fitting and Prediction

Xgboost

We ran XGBoost with a *max_depth* of 2 and *learning_rate* of 0. The accuracy we got for this model on the test data was around **90.4%**. The accuracy was pretty decent as we hoped XGBoost to perform well.

Logistic Regression:

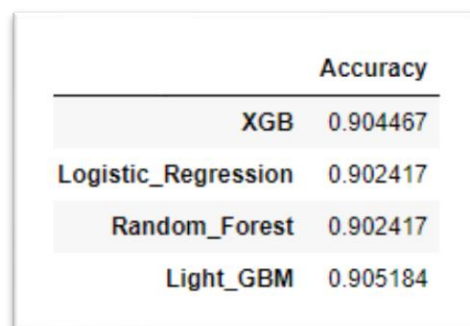
In an ideal scenario where the target variable has a binary outcome, logistic regression is always considered as a baseline model upon which we can improve. We ran a logistic regression with a **C = 0.02**. The accuracy we got was 90.24% which was near to XGBoost's.

Random Forest

Implemented Random Forest with a **max_depth = 11** and **n_estimators = 24**. It gave us the same accuracy as the of the logistic. No improvement to the model.

Light Gbm

At last we used the most advanced classification ensemble model in the opensource community Light GBM. We got the highest accuracy of all the models ie 90.51%.



	Accuracy
XGB	0.904467
Logistic_Regression	0.902417
Random_Forest	0.902417
Light_GBM	0.905184

Fig(j) Model Comparison

Recommendations

- This analysis can be used to run marketing campaigns targeting specific customers in order to increase the sales.
- Knowing the department wise sales and orders Instacart and further implement design changes to their inventory.
- Since this project gives us a pre-determined idea about a user's choice, Instacart can enhance customer experience by giving the user an option to add a list of items which he is more likely going to add.
- With the given knowledge of peak hours, sales force can be increased by hiring more employees to serve the peak hour traffic.
- Personalized customer marketing.

Future Work

We can use better algorithms which perform better at multi category classification. There is open ground to include Collaborative filtering and Association Rules to increase sales and give user recommendations. Since all the models that we have applied are linear in nature it opens up the possibilities to apply nonlinear models to obtain better accuracy.