Instacart Analysis

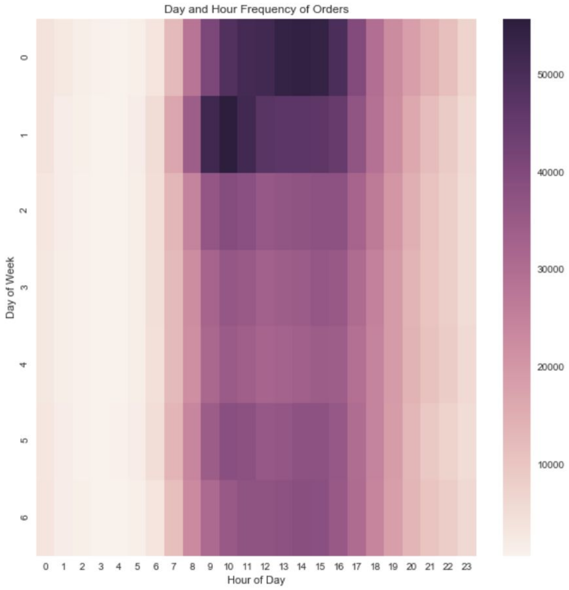
Using Python to predict customer’s reorder pattern

Dataset given

- Product to Aisle
- Product to Department
- Product Location with Aisle and Department
- Order ID with products ordered and reordered (prior) (history)
- Order ID with products ordered and reordered (trained) (current)

*order_products_train and order_products_prior comes with products in an order added in the cart and whether the product was reordered or not.

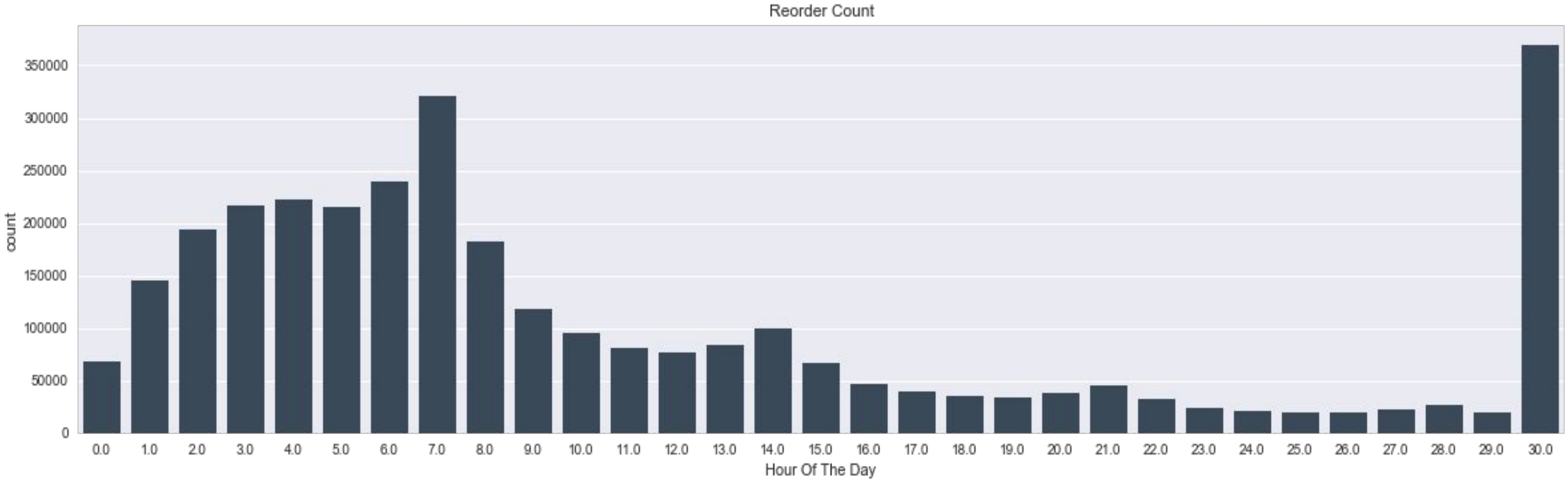
EDA for Orders Data



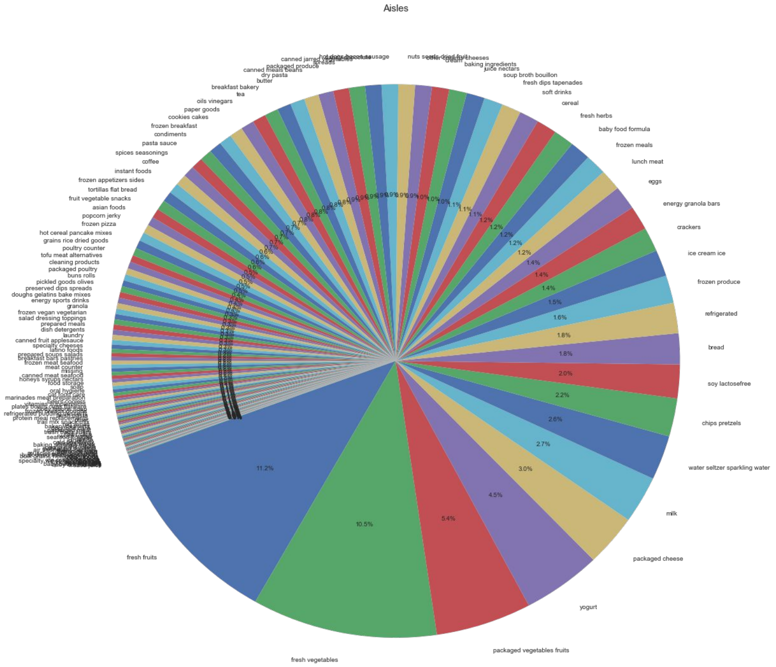
Days of the week value of 0 and 1, which corresponds to Sunday at 2 pm, generally most orders are done between 10 am to 5 pm
and Monday at 10 pm, generally crowded between 8 am to 5pm

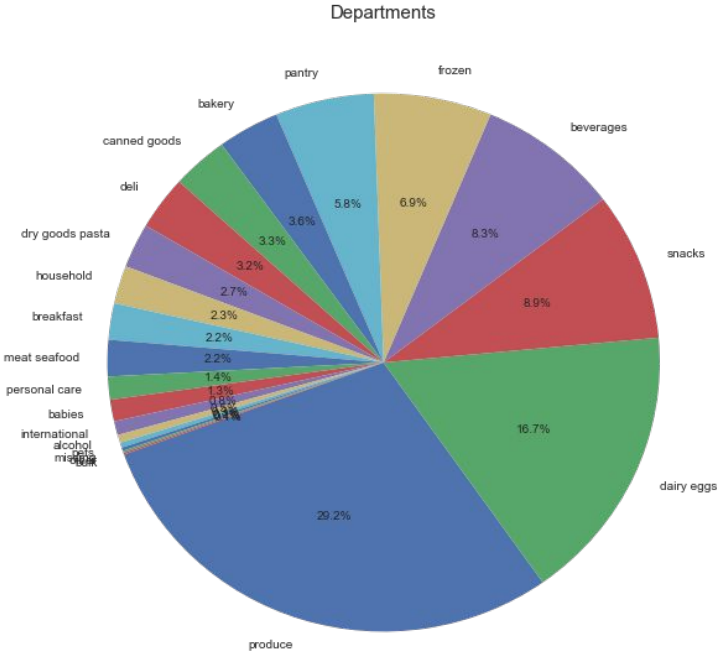
- Reorder rate happened mostly earlier in the week or later in the weekend
- Reorder are high during early morning than latter half of the day

Orders Trend shown is affected by membership trial

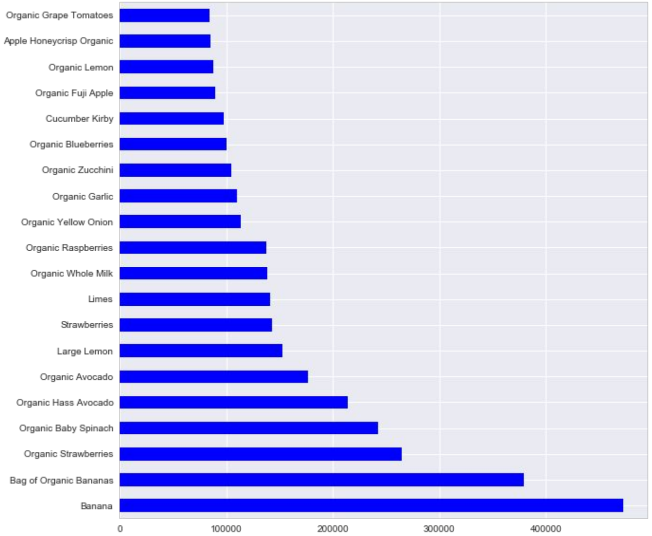


Products Trend shows Customers like to purchase healthy produce.





EDA for Products Data



Products accumulated and sorted by most ordered by customers. This visualization shows most customers has a trend of shopping healthy under Instacart platform for fruits and vegetables whether it be organic or non organic.

Feature Engineering

Product features:

- Purchase_count: How many people purchased this product
- Reordered_count: How many people re-ordered this product
- Product_reorder_rate: Reordered_count / Purchase_count

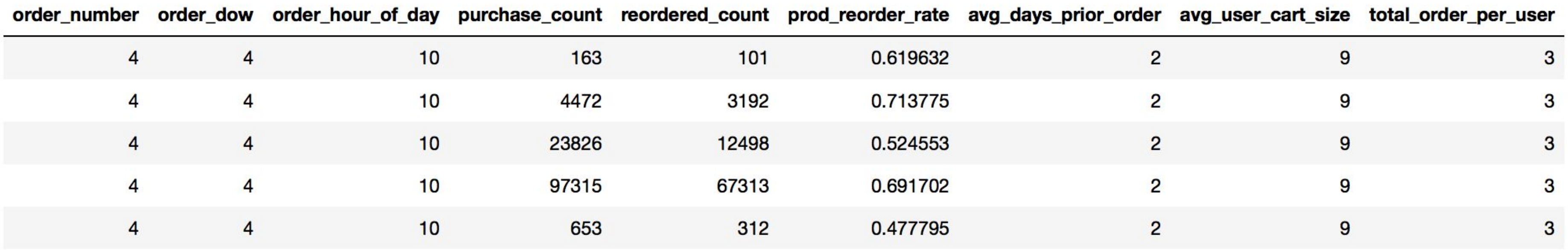
User features:

- Avg_reorder_days: average number of days the user comes back to shop at Instacart
- Avg_user_cart_size: average size of user cart per order
- Total_order_per_user: The total order made per user in the dataset by Kaggle

User-Product features:

- Purchase_count_spec: How many times this specific user buy this specific product
- Reorder_count_spec: How many times this specific user reorder this specific product
- Reorder rate: Reorder_count_spec / Purchase_count_spec

Preparing training dataset for prediction model



order_number	order_dow	order_hour_of_day	purchase_count	reordered_count	prod_reorder_rate	avg_days_prior_order	avg_user_cart_size	total_order_per_user
4	4	10	163	101	0.619632	2	9	3
4	4	10	4472	3192	0.713775	2	9	3
4	4	10	23826	12498	0.524553	2	9	3
4	4	10	97315	67313	0.691702	2	9	3
4	4	10	653	312	0.477795	2	9	3

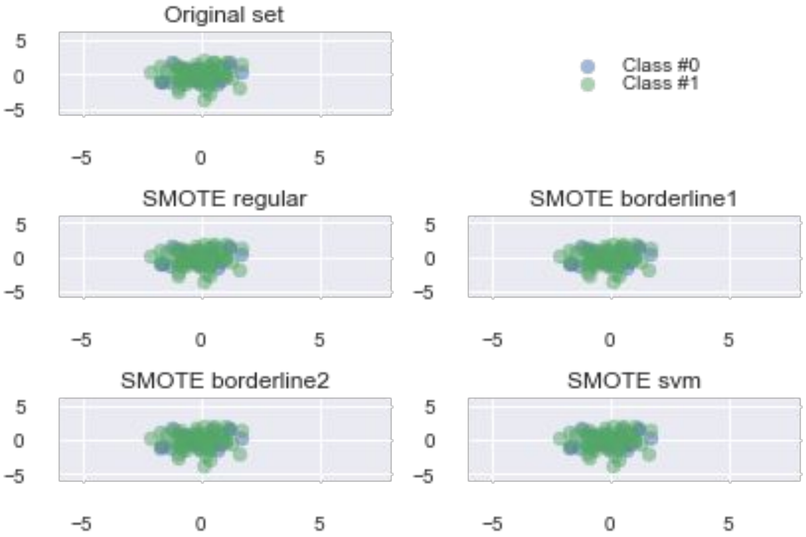
Results and observation

	Logistic Regressor	Decision Trees	Random Forest	LGBoost
Accuracy	0.648	0.686	0.664	0.698
Precision	0.650	0.680	0.670	0.690
Recall	0.650	0.690	0.660	0.700
f1-score	0.600	0.680	0.670	0.690
AUC Score	0.579	0.653	0.653	0.667

- This result gives an indication that the LGboost model is definitely a better predictive model to run to achieve higher accuracy than the other models. Evidently, the AUC-score of the Light gradient boosting yields better result due to a more distributed tasks among the nodes for computation.
- Highlist the best result and why better f1-score is desirable

No significant change in performance when SMOTE is applied.

Binary Class	0	1
Predicted	47143	160483
Resampled	65389	142237



Limitations

- Feature Engineering done differently:
  - using categorical purchase based on diet preferences such as vegetarian, pescetarian or just plainly a person who eats anything.
  - Products association with other products based on whether customer is single, couple or a family of three and larger.
  - Purchased involving social events or personal wellness
- Actual testing data is not represented with an equal amount of binary values. The oversampling of the data makes the result lean towards to the value 1, which brings down the accuracy of the prediction model.