InstaMarketing

Georgetown Data Science
Cohort 15 – Summer 2019
Capstone Project

Project Team

Team Members:

- Charles Ping
- Joe DeRose
- Mohamed Osman
- Raymond Stanley
- Richard Colvin

GEORGETOWN UNIVERSITY
School of Continuing Studies
Professional Certificate

Data Science

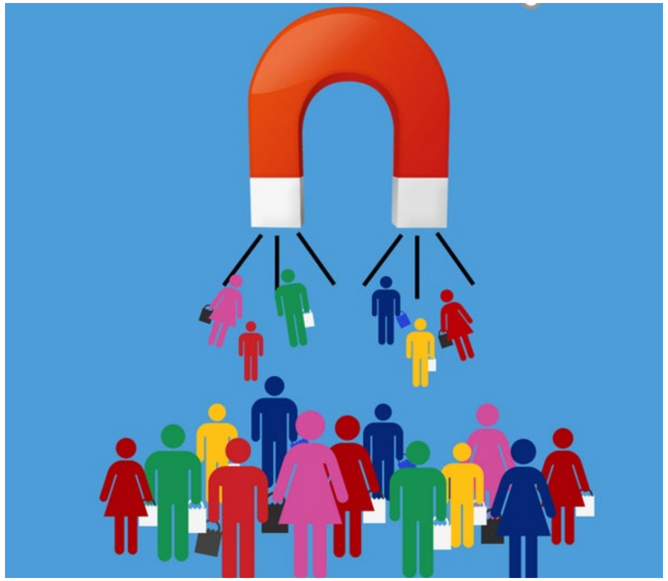
Overview

Presentation Agenda

- Project Background
- Data Source Summary
- Data Wrangling & Storage
- Feature Generation
- Exploratory Data Analysis
- Feature Ranking
- Data Modeling & Analysis
- Conclusion
- Data Product Demonstration

Project Background

- Utilizing data from a large online grocery delivery service, can we predict when an existing customer is likely to order again?



- Knowing when a customer is likely to order again can help optimize retention strategy
- Retention Statistics:
  - 2% increase in customer retention can lead to up to a 10% reduction in costs
  - Typical American business loses 15% of customers annually *(Smallbiztrends.com)*

Data Source Summary

- Dataset is sourced from Kaggle competition
- Utilizes Instacart online grocery delivery service
- Original goal of competition was to predict which products a customer will order next
- Dataset includes group of relational csv files containing over 3 million orders for 200,000 customers

Data Source Summary

Aisles.csv
aisles._id: integer (1:134)
aisle: string

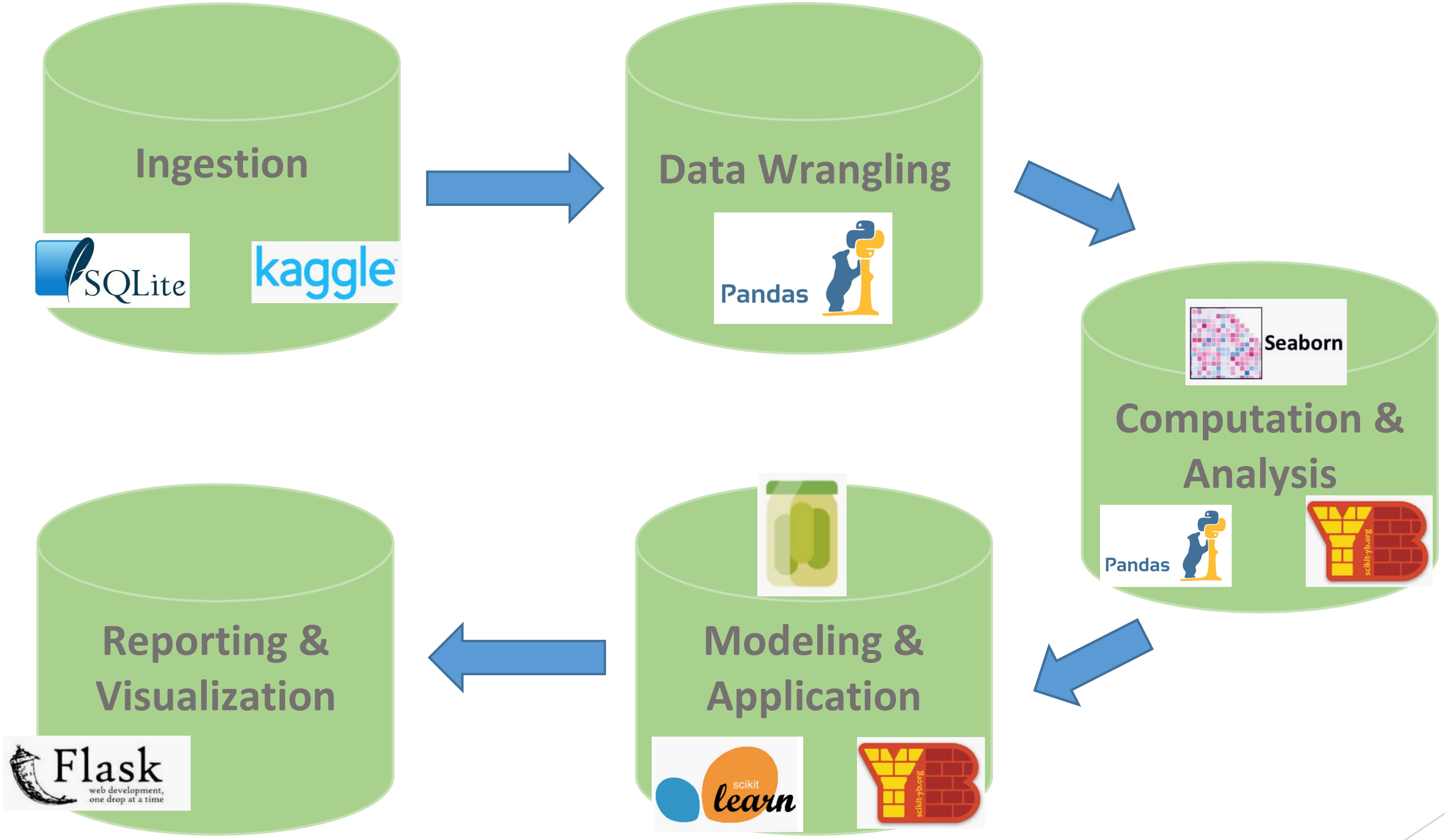
Departments.csv
department_id: integer (1:21)
department: string

Products.csv
product_id: integer (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

Orders.csv
order_id: integer
user_id: string
order_number: integer
order_dow: integer (1-7)
order_hour_of_day: integer (0-23)
day_since_prior_order: integer (0-30)

Project Pipeline



Data Wrangling & Storage

- Loaded csv files from Kaggle to tables in SQLlite database
- Five csv files with varying amounts of information loaded into five separate tables in SQLite
- Data verification: 206,209 users (instances) with range of 4 to 100 orders each
- Days since prior order range from 1-31 days, target group will be combination of this feature once EDA is performed
- Merged required features into one csv file and loaded to table in SQLite

Exploring the orders dataset

```
print('Total unique Count:')
orders_df.nunique()
```

```
Total unique Count:

[9]: order_id                  3421083
     user_id                    206209
     eval_set                        3
     order_number                  100
     order_dow                       7
     order_hour_of_day              24
     days_since_prior_order         31
     dtype: int64
```

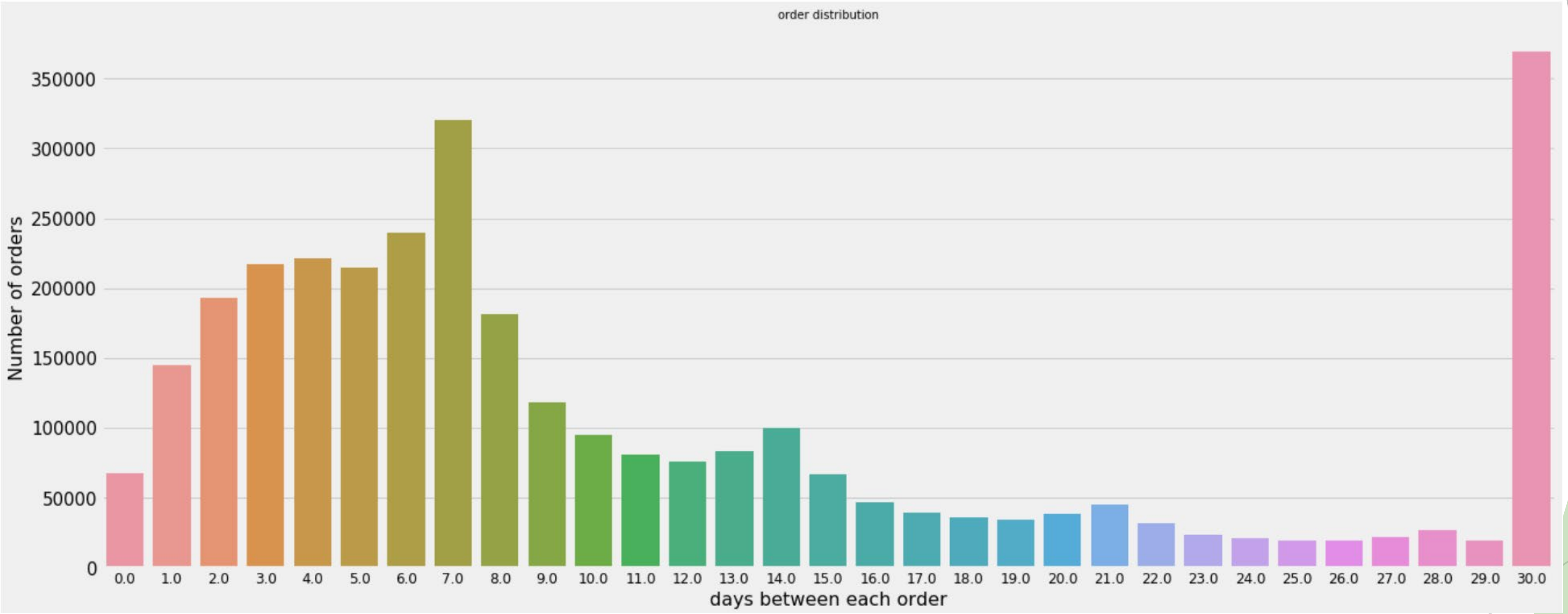
Table above referenced from Jupyter notebook uploaded to Github

Exploratory Data Analysis

- Several questions concerning our merged dataset to answer prior to generating target and feature
- What was the distribution of the days since prior order for a user?
- How many orders did the average user have?
- What days of the week and time of day were most popular for users to order?
- What is the most popular department type of products ordered?
- How many unique products did a user order during the time frame of our dataset?

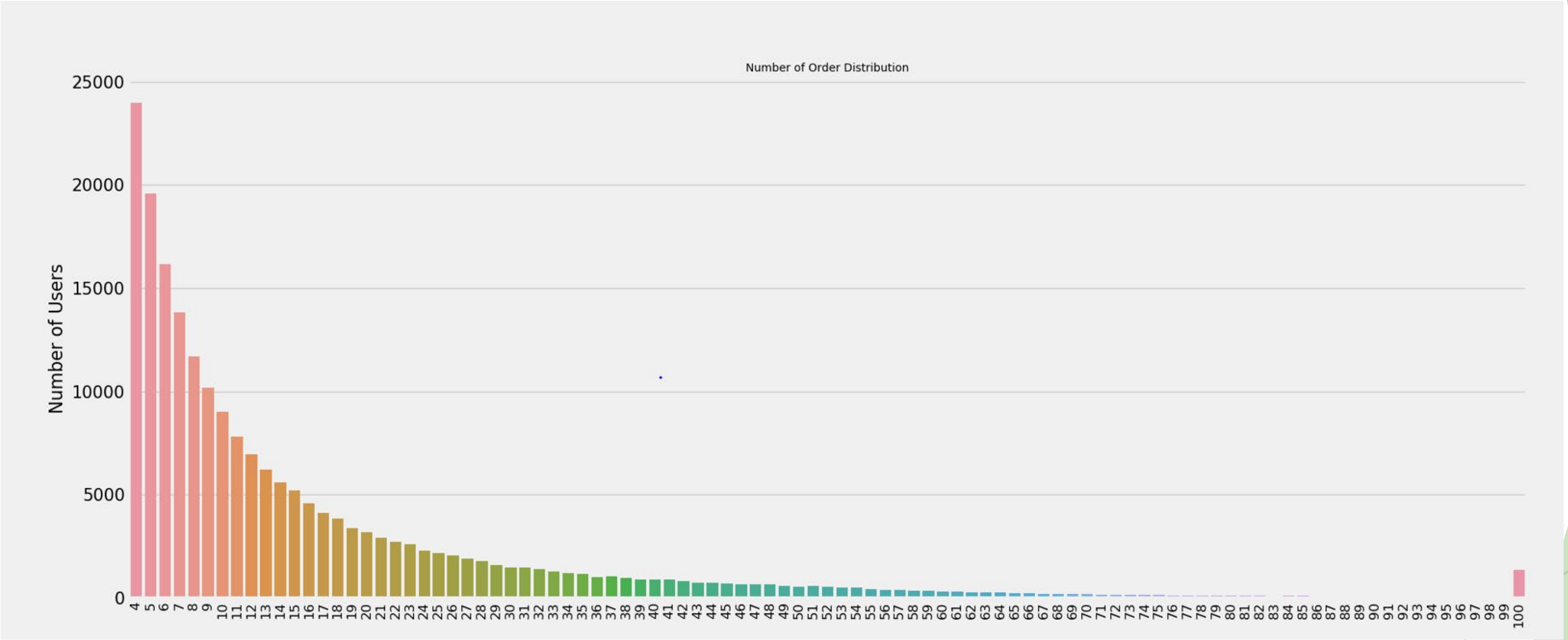
Exploratory Data Analysis

Distribution of Days Since Prior Order



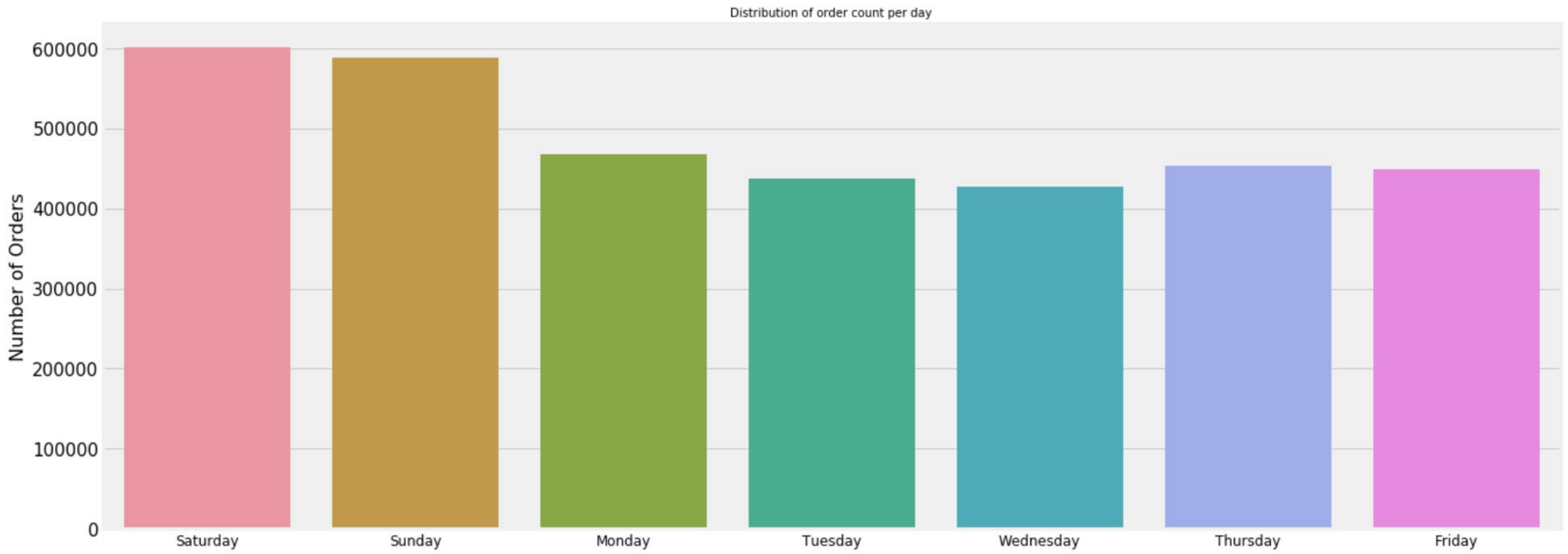
Exploratory Data Analysis

Number of Orders per User



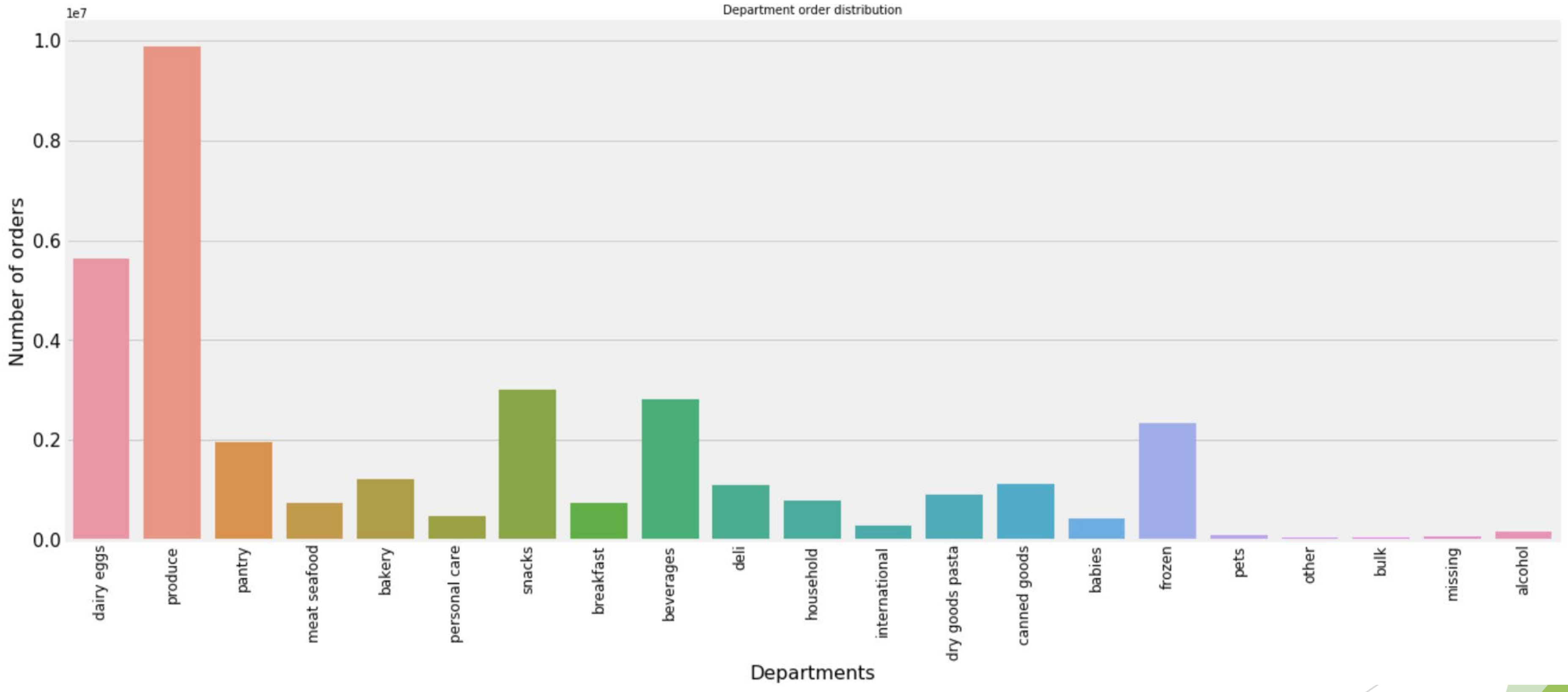
Exploratory Data Analysis

Order Count per Day of the Week



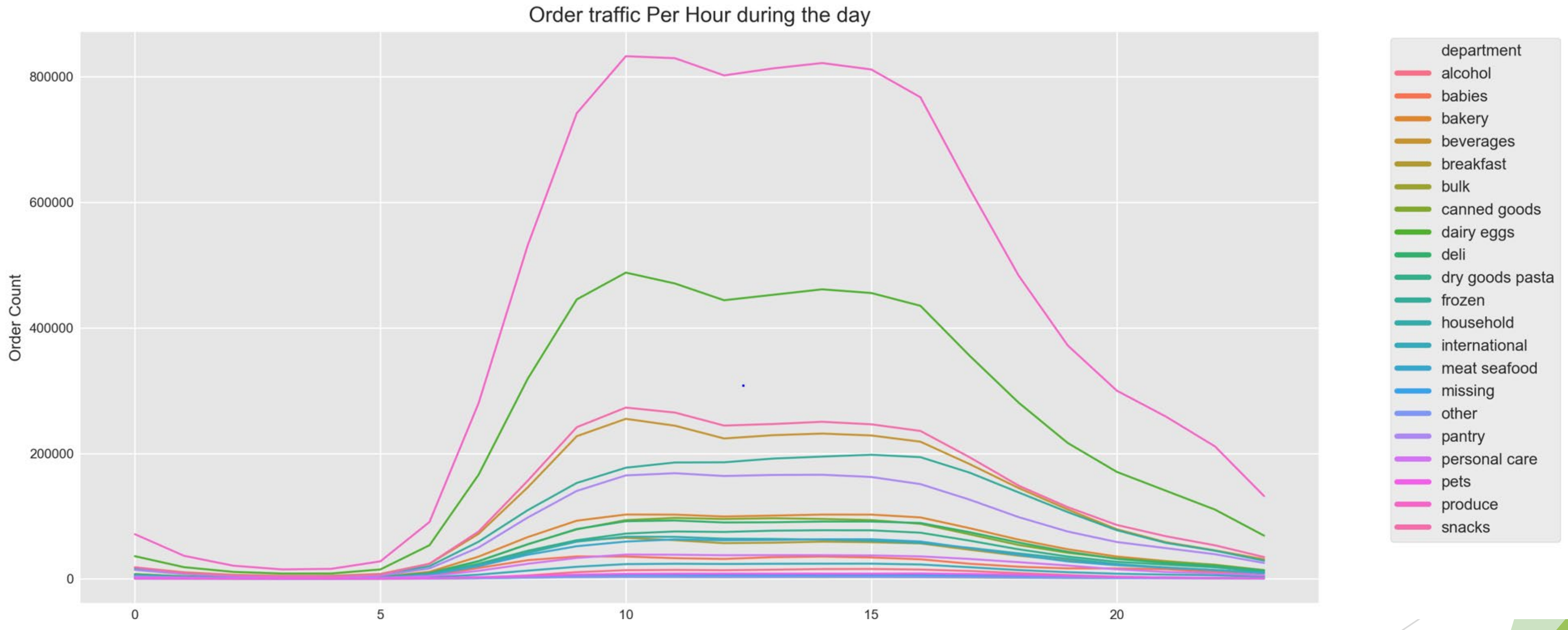
Exploratory Data Analysis

Order distribution by Department



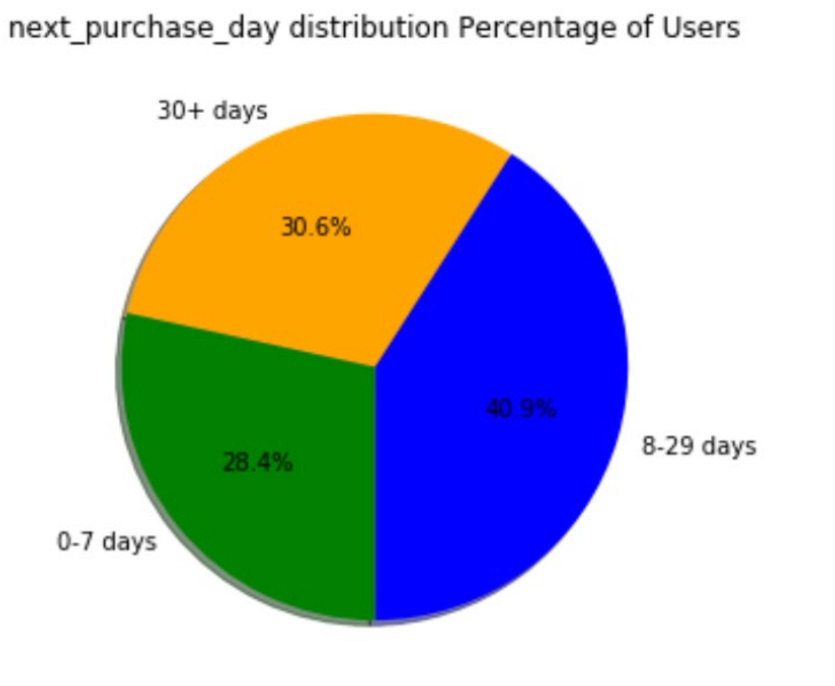
Exploratory Data Analysis

Order traffic per hour by Department type



Feature Generation

- Through preliminary EDA three distinct groups of next purchase day became evident
- Target defined as next purchase day for users grouped by 0-7 days, 8-29 days, and 30+days
- Multiclass Classification problem
- Assigned labels to each group; 0-7 days assigned “0”; 8-29 days assigned “1”; 30+ days assigned “2”
- Feature generation based upon intuitive predictive qualities from dataset

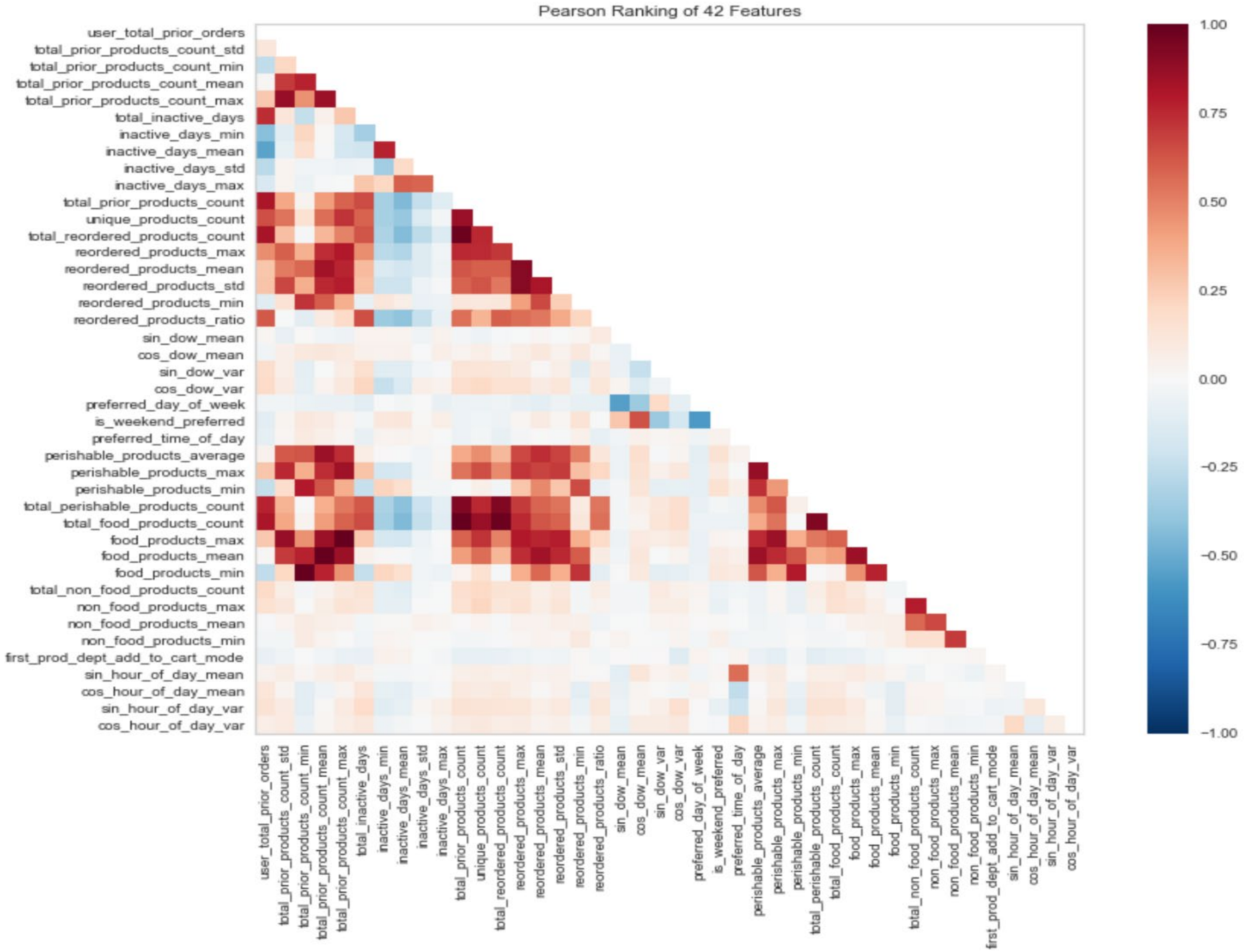


Feature Generation

- 42 total feature included in dataset
- Examples of features created include:
  - Number of inactive days per user
  - Number of unique products ordered by user
  - Number of reordered products by user
  - Preferred day of week for order by user
  - Preferred hour of day by user
    - Encoded (Morning=0; Afternoon=1; Night 2)
  - Number of perishable items by user
    - Bakery, Produce, Meat, Seafood, Dairy defined as perishable
  - Number of food vs nonfood items purchased by user
  - Mode of department type of first product added to cart by user

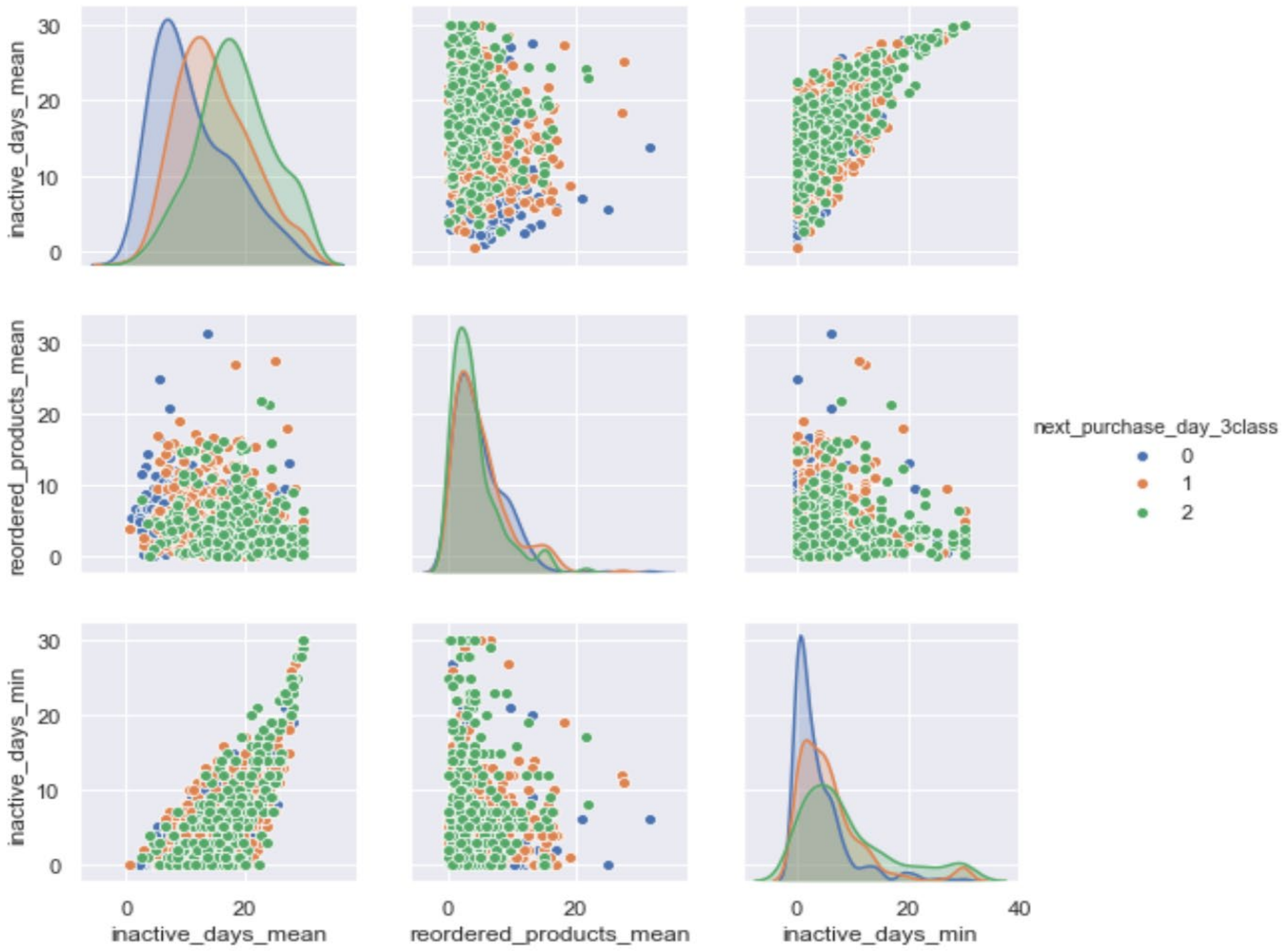
Feature Ranking

Yellowbrick 2D Rank – Feature Ranking



Feature Plot

Plot Top 3 Features Relative to Target



Data Modeling and Analysis

- Model selection process:
  - Used Cross Validation method to evaluate the performance of different models:
    - Tried different classification models from sklearn
    - Tried different scaler techniques for each model
  - Hyperparameter Tuning using GridSearchCV from sklearn

- Model Evaluation:
  - Feature Importance
  - Confusion Matrix
  - Classification Report

Data Modeling and Analysis

Multiclass Classification - Model Selection

- Cross Validation Scores:

model scaler	Bagging	CART	ExtraTrees	KNN	LDA	LR	LinearSVC	NB	RF	SVM	XGB
	0.476	0.432	0.504	0.442	**0.532**	0.525	0.531	0.481	0.522	0.375	0.518
MaxAbsScaler	0.483	0.432	**0.514**	0.445	0.532	**0.531**	0.532	0.481	0.522	**0.53**	0.518
MinMaxScaler	0.48	0.431	0.507	0.445	0.532	0.53	0.532	0.481	0.522	0.527	0.518
Normalizer	0.462	0.41	0.505	0.448	0.522	0.494	0.511	0.484	0.504	0.469	0.498
PowerTransformer-Yeo-Johnson	0.484	0.431	0.511	0.442	0.522	0.528	0.522	**0.495**	0.522	0.514	**0.522**
QuantileTransformer-Normal	0.466	**0.439**	0.503	0.432	0.512	0.51	0.511	0.478	**0.523**	0.485	0.519
QuantileTransformer-Uniform	**0.49**	0.438	0.511	0.441	0.52	0.522	0.512	0.49	0.523	0.525	0.519
RobustScaler	0.472	0.432	0.51	**0.455**	0.532	0.529	**0.533**	0.481	0.522	0.521	0.516
StandardScaler	0.482	0.432	0.512	0.44	0.532	0.527	0.532	0.481	0.522	0.523	0.517

Data Modeling and Analysis

Multiclass Classification - Model Selection

- GridSearchCV scores:

model	CV_Score_with_hyperparamter_tuning
Random Forest	0.533
Logistic Regression	0.532
SVM	0.531

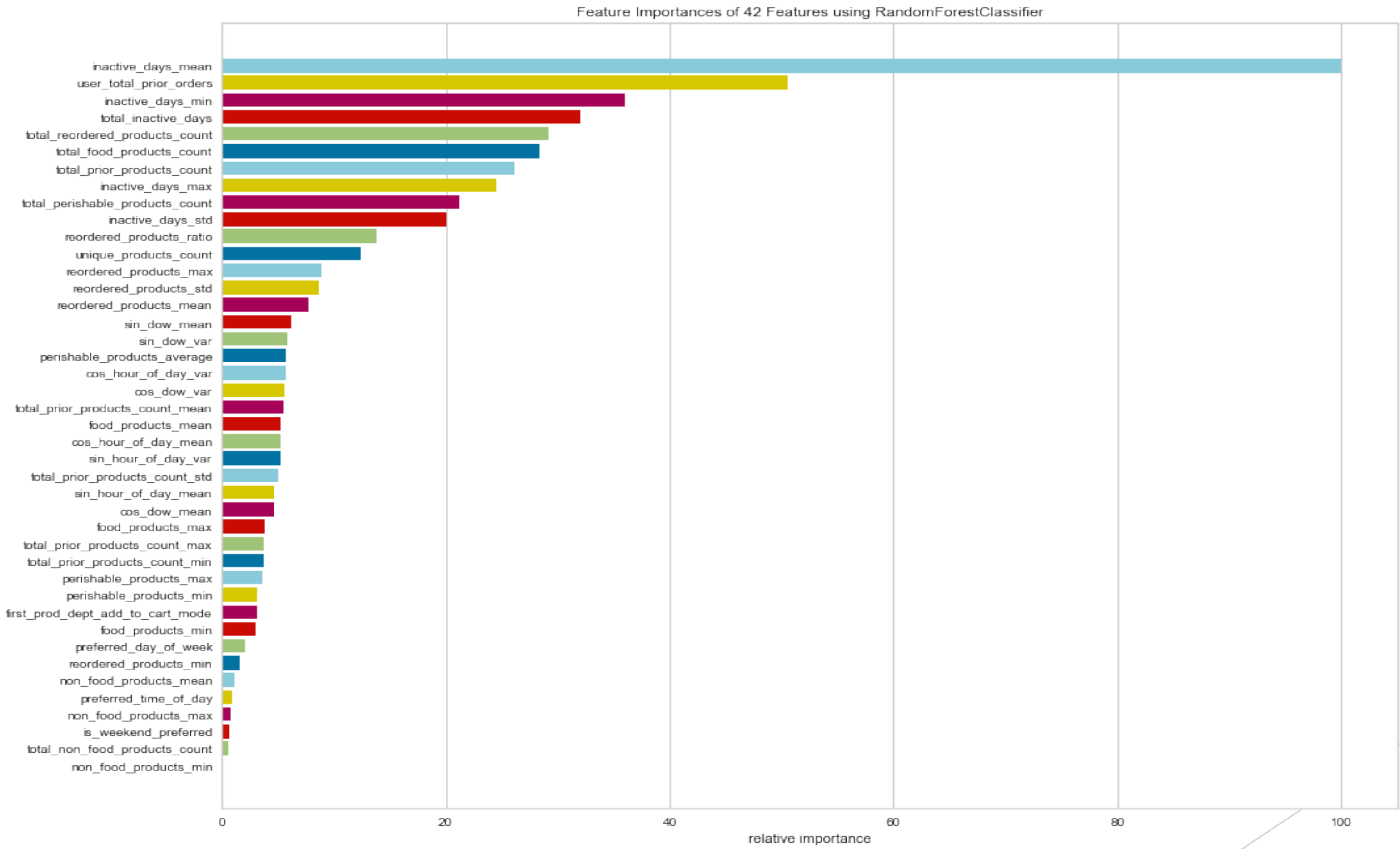
```
Best Estimator learned through GridSearch:

RandomForestClassifier(bootstrap=True, class_weight=None, criterion='gini',
            max_depth=5, max_features='auto', max_leaf_nodes=None,
            min_impurity_decrease=0.0, min_impurity_split=None,
            min_samples_leaf=1, min_samples_split=2,
            min_weight_fraction_leaf=0.0, n_estimators=500, n_jobs=None,
            oob_score=False, random_state=seed, verbose=0,
            warm_start=False)
```

Data Modeling and Analysis

Multiclass Classification - Model Evaluation

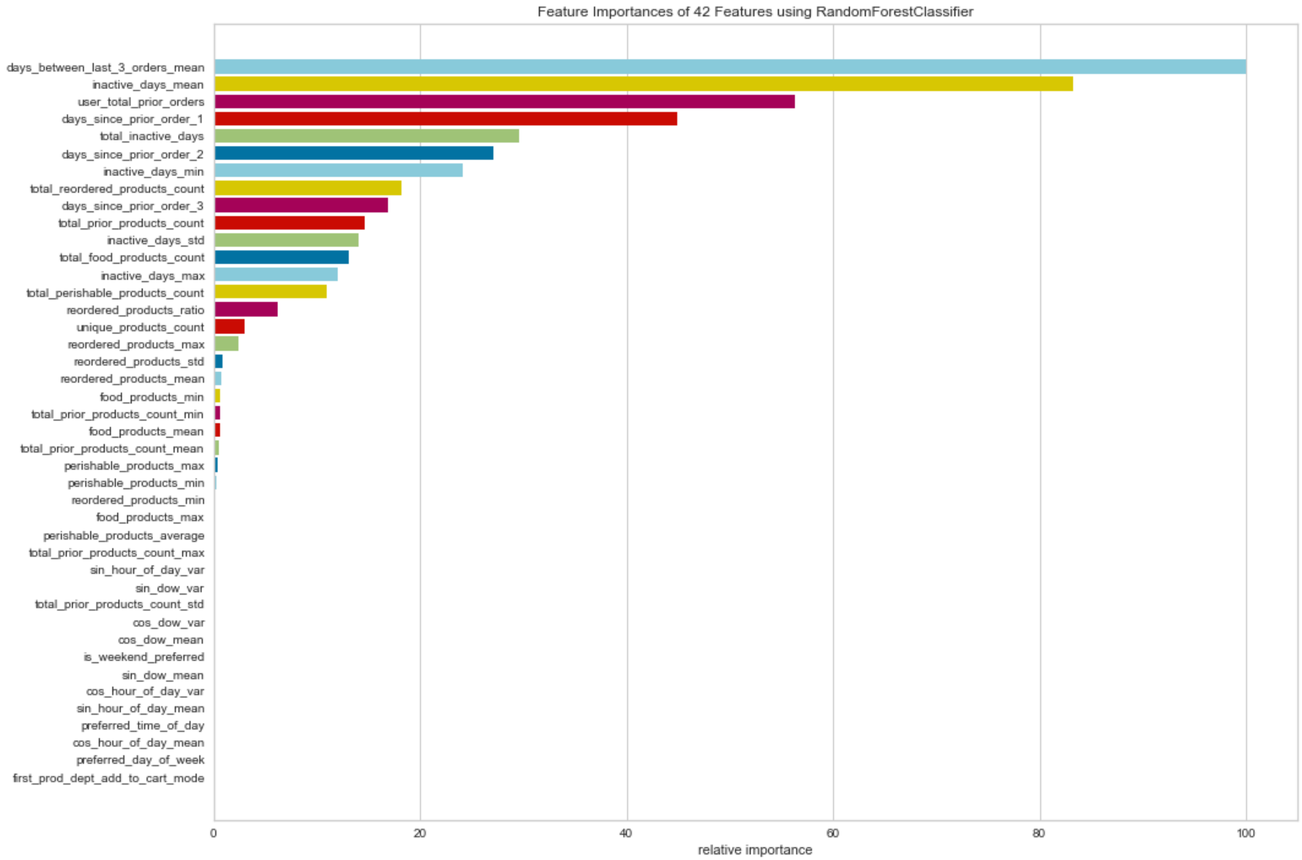
- Feature Importance using yellowbrick FeatureImportances visualizer:



Data Modeling and Analysis

Multiclass Classification - Model Evaluation

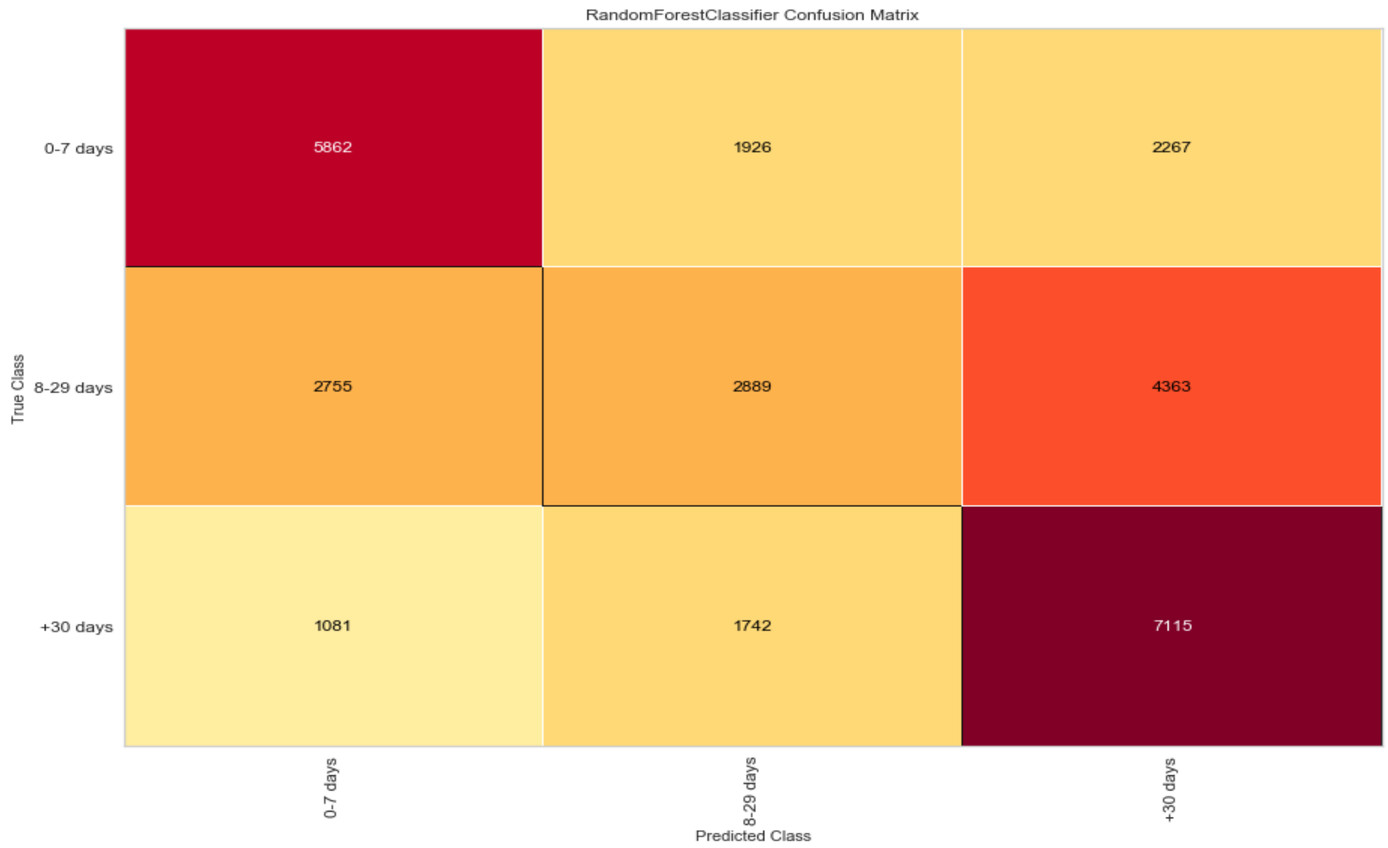
- Feature Importance after dropping features with low importance and adding new features:



Data Modeling and Analysis

Multiclass Classification - Model Evaluation

- Confusion Matrix using yellowbrick ConfusionMatrix visualizer:

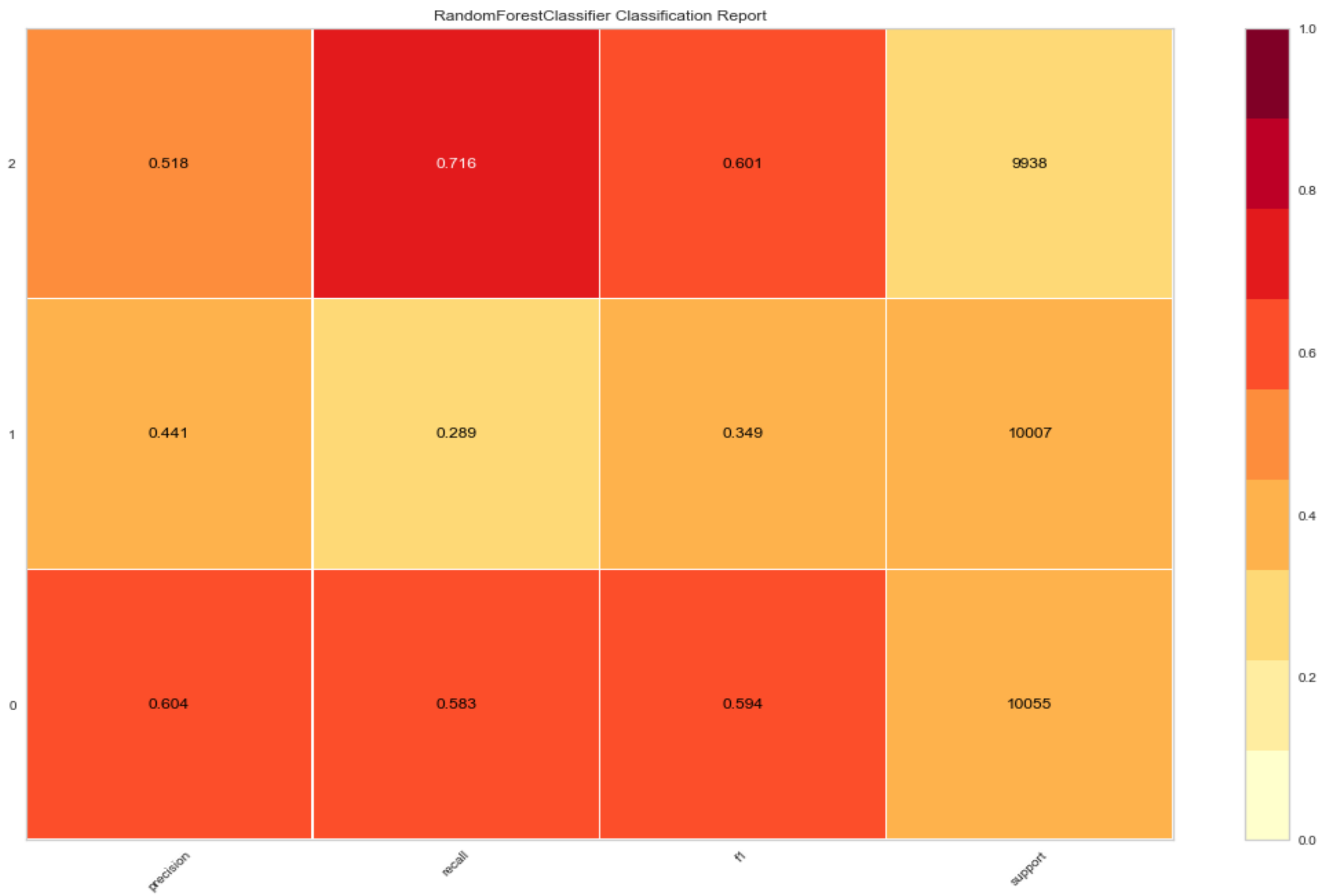


Accuracy0.5288666666666667

Data Modeling and Analysis

Multiclass Classification - Model Evaluation

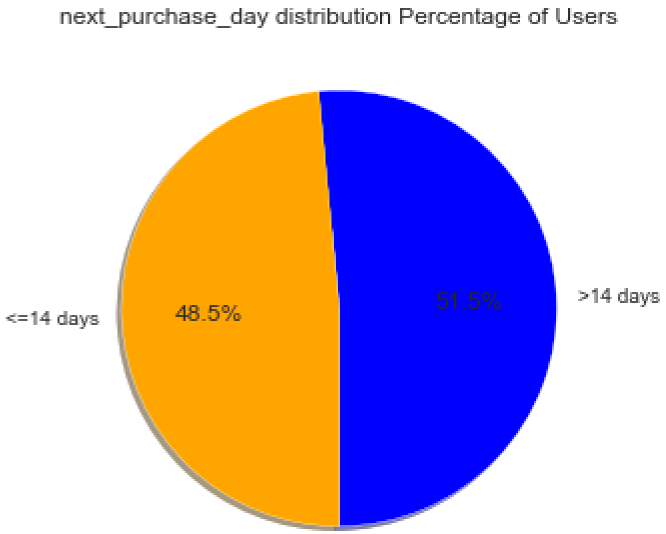
- Classification Report using yellowbrick ClassificationReport visualizer:



Data Modeling and Analysis

Another Approach

- Redefined the target (next purchase day) for users and regrouped them by <=14 days as 0 and >14 days as 1
- More balanced data set
- Binary Classification Problem
- Applied the same steps of model selection and evaluation as in the multiclass problem



Data Modeling and Analysis

Binary Classification - Model Selection

- Cross Validation Scores:

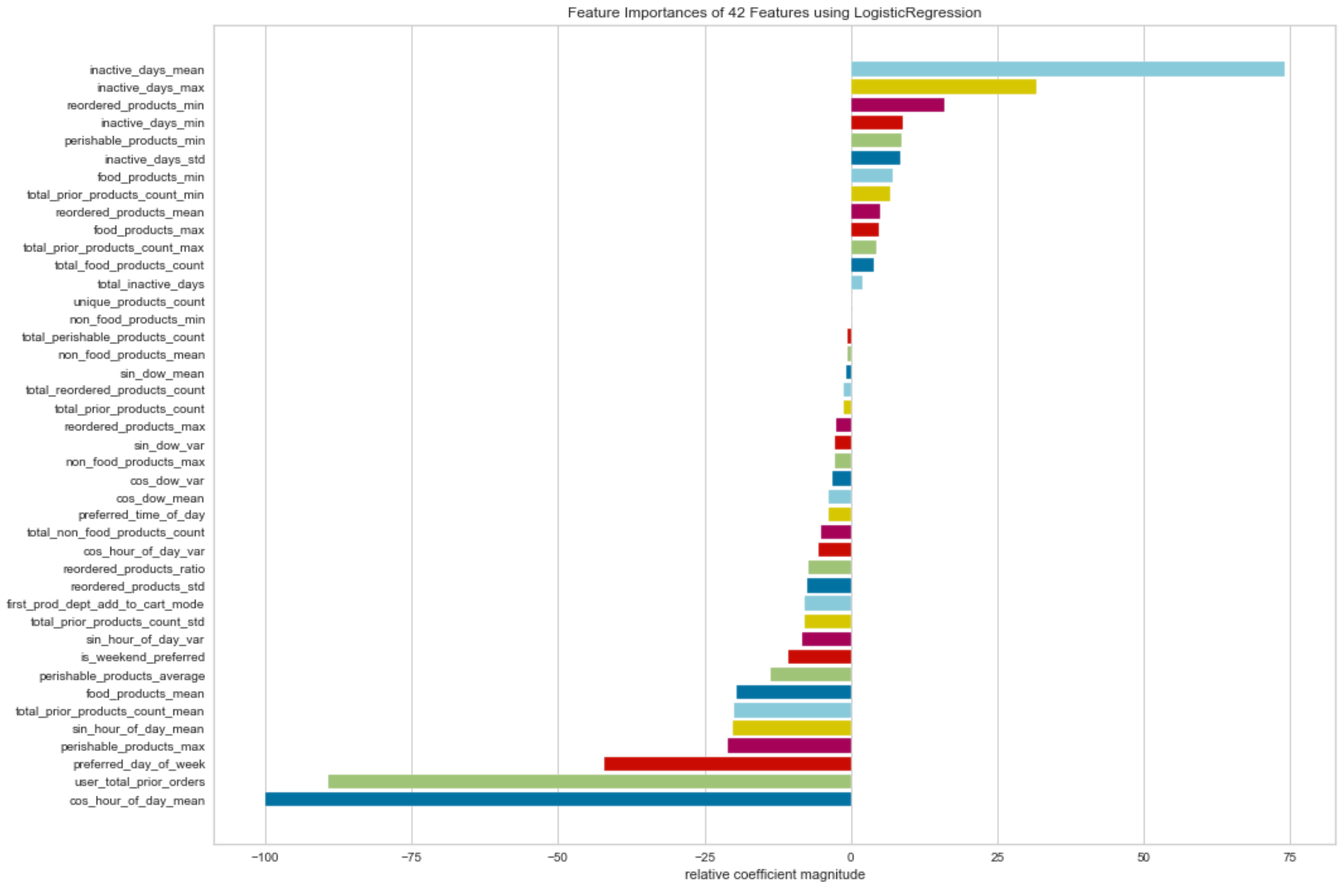
model	Bagging	CART	ExtraTrees	KNN	LDA	LR	LinearSVC	NB	RF	SVM	XGB
scaler											
	0.636	0.592	0.656	0.613	0.671	0.669	0.671	0.648	**0.676**	0.499	0.647
MaxAbsScaler	0.624	0.591	0.659	0.596	0.671	0.678	**0.676**	0.648	0.676	**0.679**	0.647
MinMaxScaler	0.632	0.592	0.658	0.612	0.671	0.677	0.676	0.648	0.676	0.676	0.645
Normalizer	0.632	0.586	0.663	0.619	0.674	0.651	0.673	0.628	0.673	0.641	**0.664**
PowerTransformer-Yeo-Johnson	0.626	0.592	0.666	0.62	**0.678**	0.676	0.675	**0.657**	0.676	0.67	0.638
QuantileTransformer-Normal	0.636	0.594	0.664	0.628	0.674	0.668	0.674	0.651	0.676	0.667	0.649
QuantileTransformer-Uniform	0.627	0.594	0.666	**0.634**	0.671	**0.682**	0.672	0.646	0.676	0.675	0.649
RobustScaler	**0.637**	0.592	**0.667**	0.632	0.671	0.68	0.674	0.648	0.676	0.673	0.646
StandardScaler	0.637	**0.595**	0.652	0.616	0.671	0.681	0.674	0.648	0.676	0.664	0.646

- Selected Logistic Regression model since it scored higher

Data Modeling and Analysis

Binary Classification - Model Evaluation

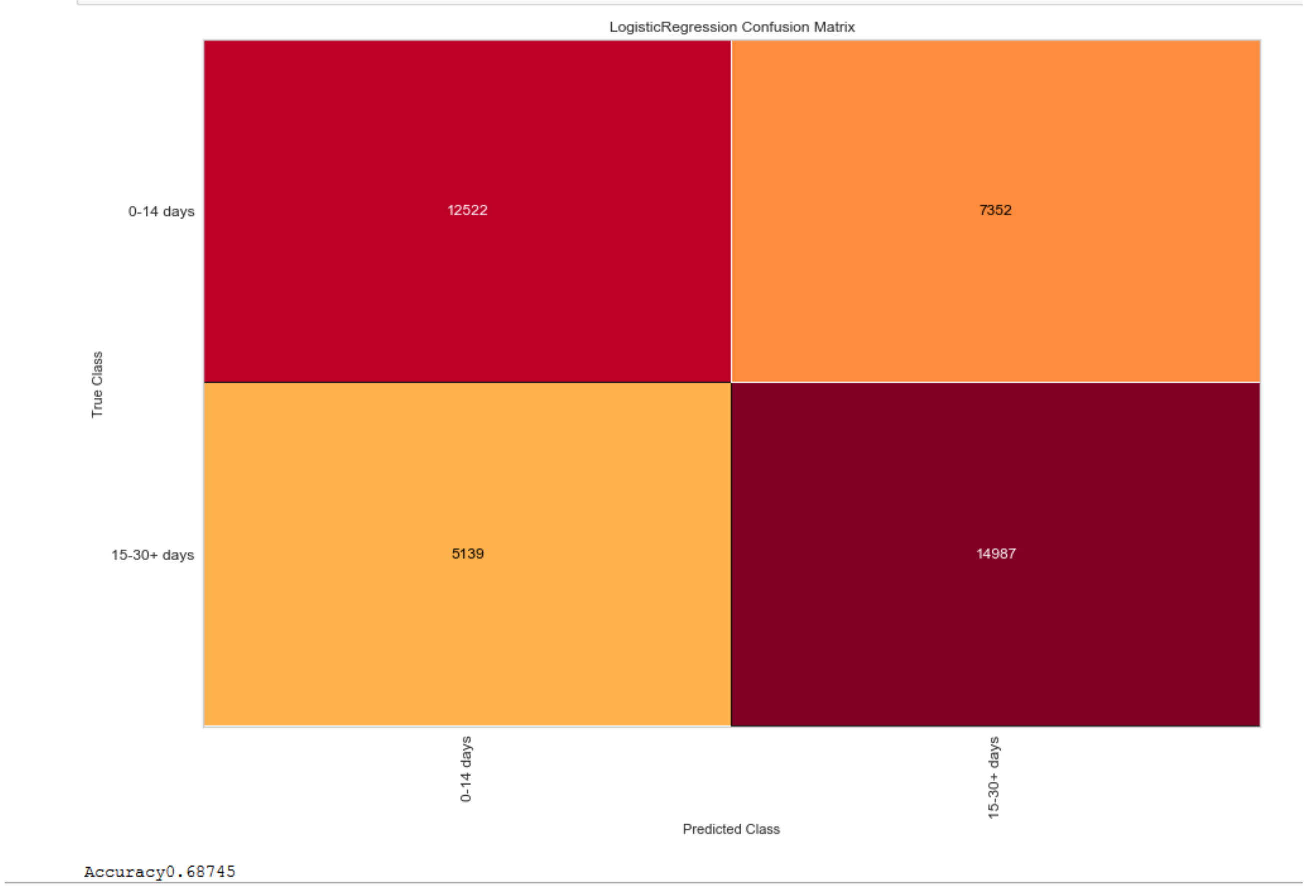
- Feature Importance using yellowbrick FeatureImportances visualizer:



Data Modeling and Analysis

Binary Classification - Model Evaluation

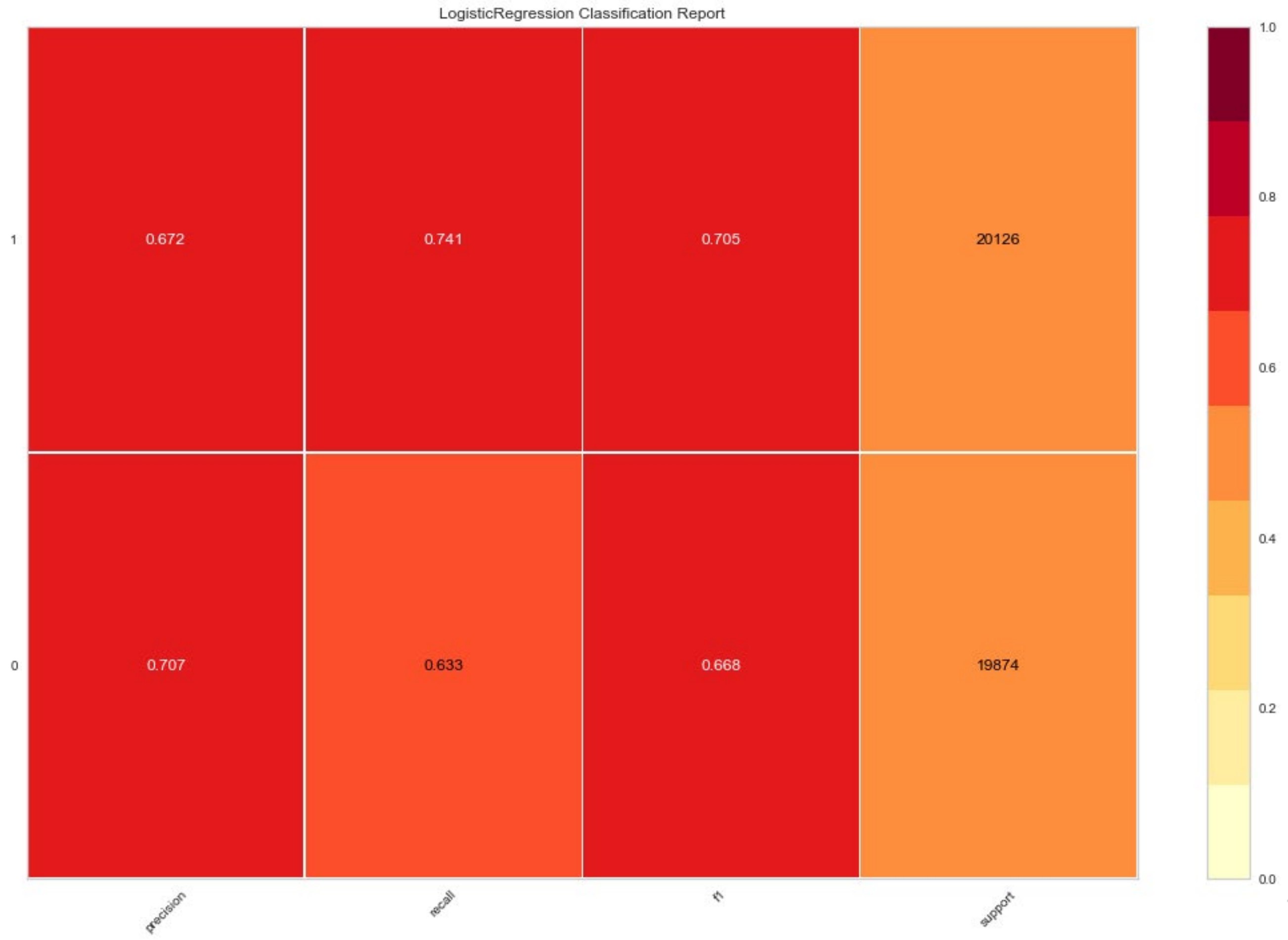
- Confusion Matrix using yellowbrick ConfusionMatrix visualizer:



Data Modeling and Analysis

Binary Classification - Model Evaluation

- Classification Report using yellowbrick ClassificationReport visualizer:



Resources

python™

Google Drive

Seaborn

kaggle

NumPy

ANACONDA

scikit-yb.org

pandas

$y_it = \beta' x_{it} + \mu_i + \epsilon_{it}$

SQLite

GitHub

instacart

jupyter

scikit learn

matplotlib

Flask

web development, one drop at a time

Conclusion

- Multiclass RF model with hypertuned score of 0.53 may not be high enough to deploy in real world setting; however grouping users into three groups is more useful for a business case
- If we had more time, finding ways to increase the Multiclass RF model score would be priority
- Ways to improve score:
  - Biggest flaw was most likely lack of strong features
  - Brining in data from other sources outside of the Instacart csv set could enhance outcome
  - Features such as user demographics, spending habits, and grocery store ordered from

Data Product

- Model demonstration – Flask App