أكاديمية سدايا
SDAIA Academy

METIS

Predicting Instacart's Customers Behaviors

By : Manal AlQahtani, Nadia AlGhamdi

As Project 3 of SDAIA Data Science Bootcamp (T5)

OUTLINE

- Introduction
- Methodolgy
- Results
- Recommendations

أكاديمية سدايا
SDAIA Academy
METIS

INTRODUCTION

instacart

Instacart is a grocery ordering and delivery app that allows customers to select products on their app or website and a personal shopper handpicks those products by in-store shopping and delivers the order.

In this Kaggle Competition, Instacart made their anonymized data available for Machine Learning practitioners with an aim for best Machine Learning models to analyze customer reorder patterns and predict which products can a customer reorder based on their previous shopping data

أكاديمية سدايا
SDAIA Academy

METIS

Business Need

The business requirement here is to predict which previously purchased products will be in customer's next order

Methodology

Machine Learning classification Algorithm

TOOLS

- Pandas
- Matplotlib and Seaborn
- Sklearn and Imblearn

أكاديمية سدايا
SDAIA Academy

METIS

Data source

kaggle

Aisle
134 rows and two columns

Departments
21 rows and two columns

Products
49688 rows and 4 columns

Orders
3421083 rows and 7 columns

Order Products
33819106 rows and 4 columns

أكاديمية سدايا
SDAIA Academy

METIS

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

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

أكاديمية سدايا
SDAIA Academy
METIS

Dataset

33819106 rows

11 columns

Numerical features

8

Categorical features

3

	user_id	order_id	order_number	order_dow	order_hour_of_day	days_since_prior_order	product_name	add_to_cart_order	reordered	department	aisle
0	202279	2	3	5	9	8.0	Organic Egg Whites	1	1	dairy eggs	eggs
1	202279	2	3	5	9	8.0	Michigan Organic Kale	2	1	produce	fresh vegetables
2	202279	2	3	5	9	8.0	Garlic Powder	3	0	pantry	spices seasonings
3	202279	2	3	5	9	8.0	Coconut Butter	4	1	pantry	oils vinegars
4	202279	2	3	5	9	8.0	Natural Sweetener	5	0	pantry	baking ingredients

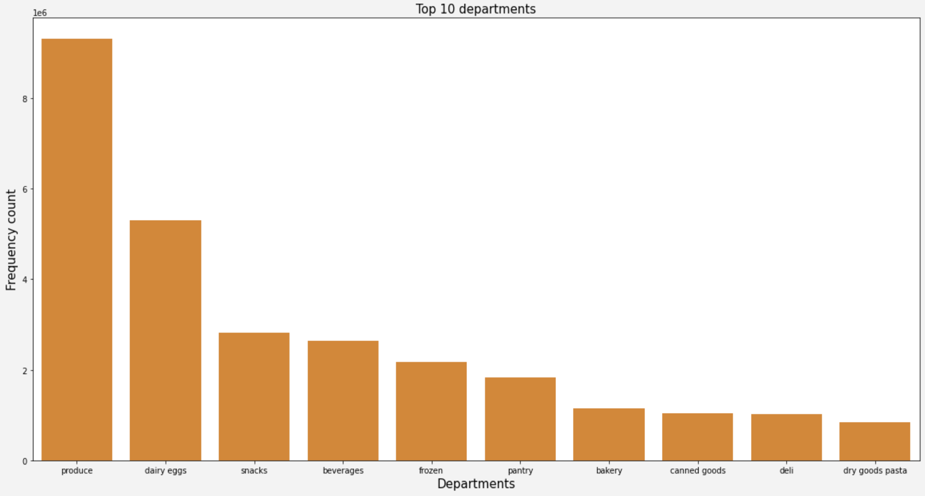
EDA

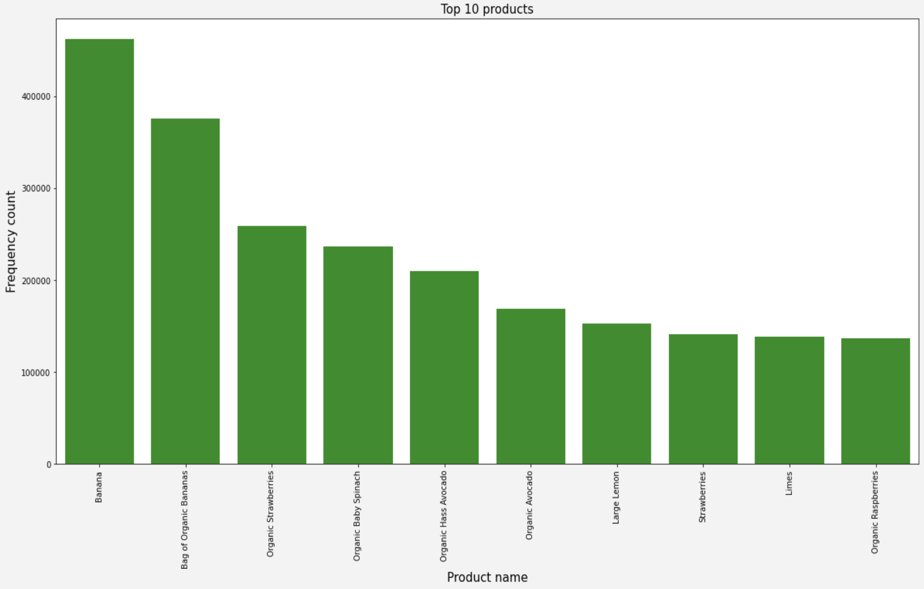
- Merged data files on similar columns (aisle_id, product_id, department_id, user_id) .
- Cleaning data : drop nulls, duplicated and reomove whitespacing .
- Applying some feature improvements :

  - Feature Seelction :
    - select columns that would achieve our goal, and drop unneceerary columns for us (e.g. eval_set) .
  - Feature Engineering :
    - Convert weekdays from numbers to labels (help in visualization) .
    - Get Number of order per hour out of order_id and order_hour_of_day .
    - Get the most sold products out of Product names .
    - Encoding categorical features (4 features) .
  - Feature Scaling :
    - standardize the data .

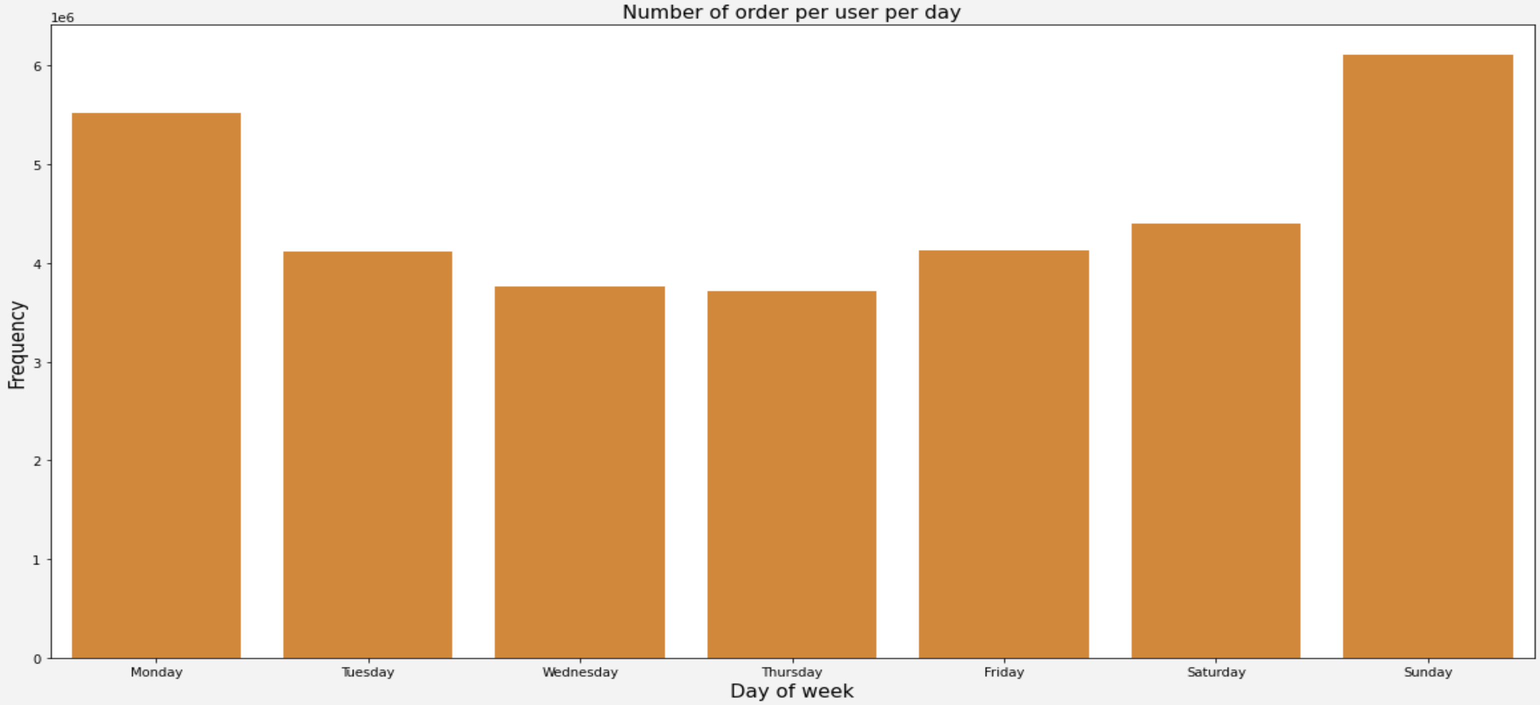
Which departments have higher reorder ?

What is the most reordered products ?





What is the most number of products were ordered and reordered and on what days of week ?



أكاديمية سدايا
SDAIA Academy

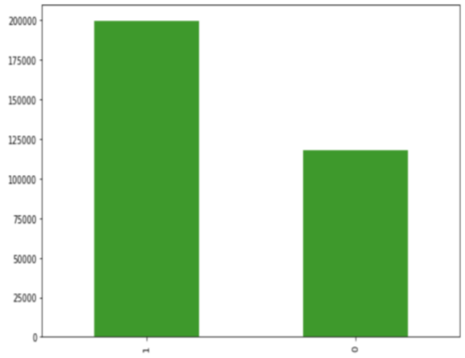
METIS

Due to time consumption, we've chose 1% of data to fit the model on (317410, 11)

Checking imbalancing data

```
percentage of class 1 is : 63 %
percentage of class 0 is : 37 %
```

```
plt.figure(figsize =(10,6))
data['reordered'].value_counts().plot(kind = 'bar',color='#3A9A22' );
```



Dealing with imbalancing data

SMOTE

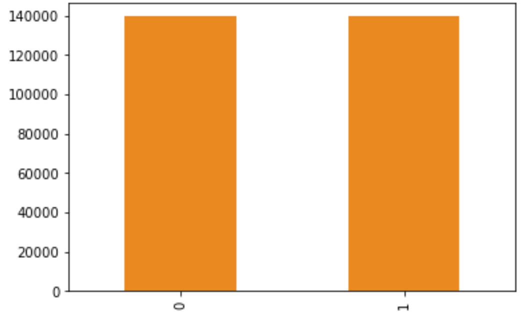
Random Oversampling

- How each method sampled data :

```
len(X_train_sm), len(y_train_sm)
```

```
(279752, 279752)
```

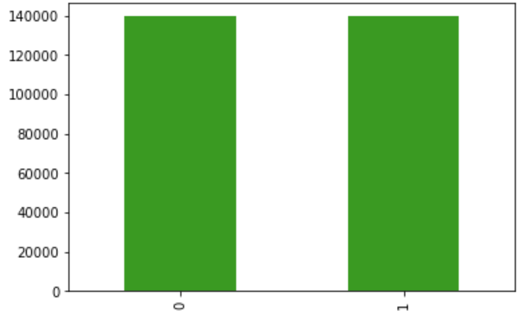
```
pd.Series(y_train_sm).value_counts().plot.bar();
```




```
len(X_train_os), len(y_train_os)
```

```
(279752, 279752)
```

```
pd.Series(y_train_os).value_counts().plot.bar();
```



Experiments

Models

1- KNN

2- Logistics Regression

3- Decision Tree

4- Random Forest

Sampling

1- SMOTE

2- Random Oversampling

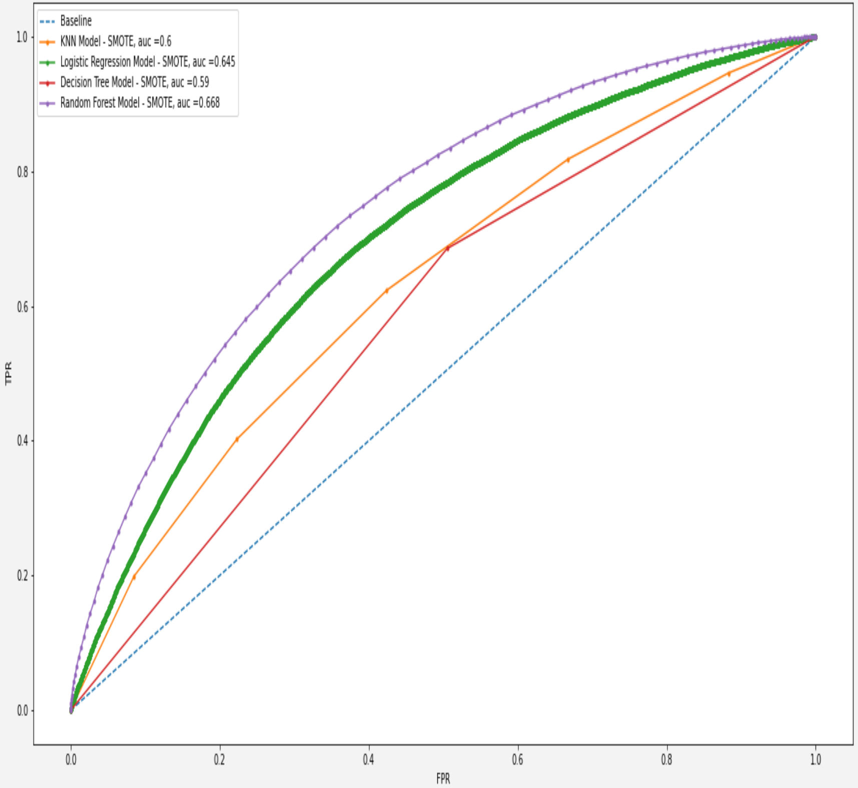
Results

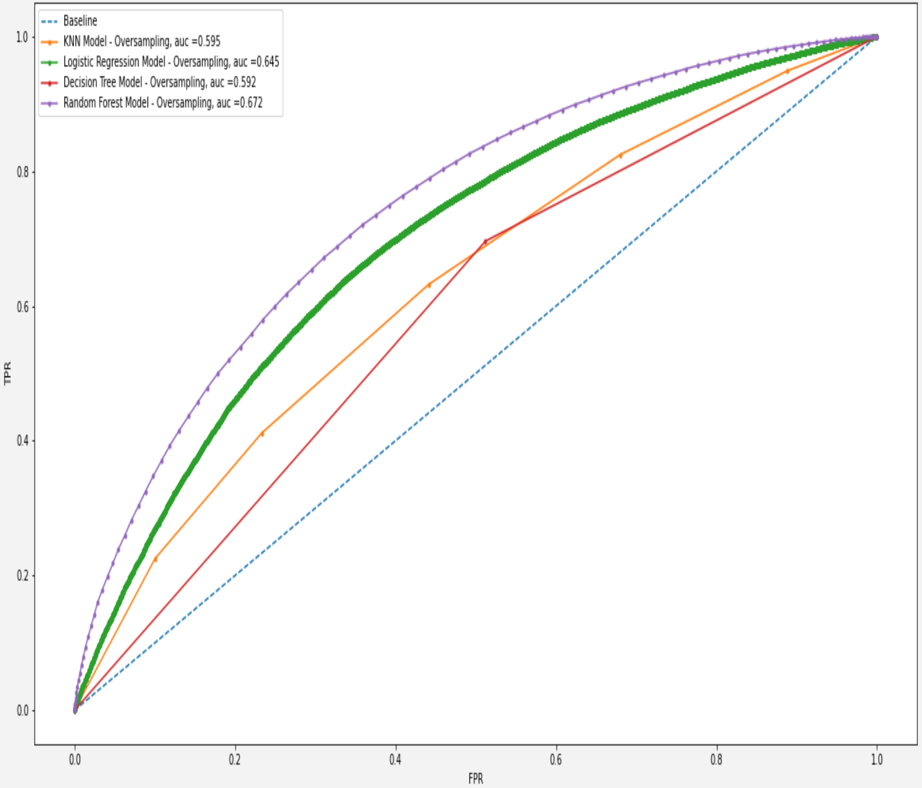
SMOTE

Model	Precision	Recall	F1	Accuracy
KNN	71.252	62.335	66.495	60.567
Logistic Regression	77.730	56.009	65.106	62.311
Descisionn Tree	69.580	68.543	69.057	61.441
Random Forest	74.166	81.315	77.576	70.490

Random Oversampling

Model	Precision	Recall	F1	Accuracy
KNN	70.719	63.203	66.750	60.472
Logistic Regression	77.637	56.329	65.288	62.399
Descisionn Tree	69.633	69.633	69.633	61.874
Random Forest	74.679	80.275	77.376	70.531


Baseline
KNN Model - SMOTE, auc =0.6
Logistic Regression Model - SMOTE, auc =0.645
Decision Tree Model - SMOTE, auc =0.59
Random Forest Model - SMOTE, auc =0.668
TPR
FPR
0.0
0.2
0.4
0.6
0.8
1.0

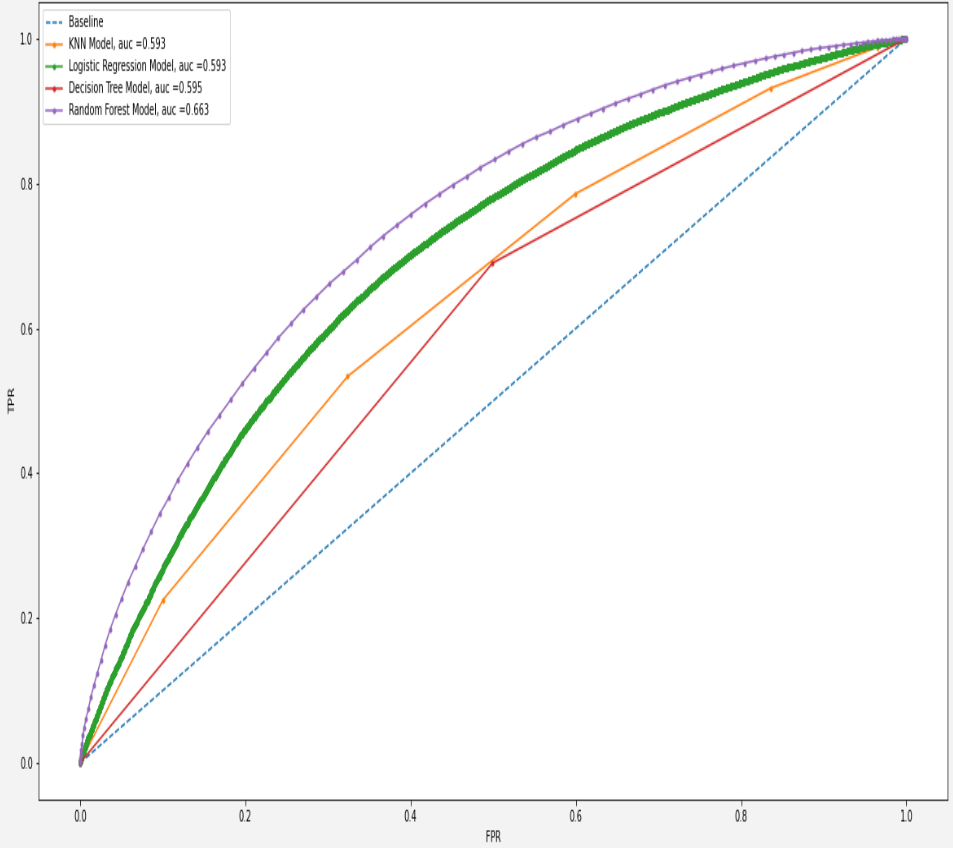

Baseline
KNN Model - Oversampling, auc =0.595
Logistic Regression Model - Oversampling, auc =0.645
Decision Tree Model - Oversampling, auc =0.592
Random Forest Model - Oversampling, auc =0.672
TPR
FPR
0.0
0.2
0.4
0.6
0.8
1.0

أكاديمية سدايا
SDAIA Academy

METIS

Without Sampling :

Model	Precision	Recall	F1	Accuracy
KNN	68.849	78.565	73.386	64.229
Logistic Regression	67.958	90.638	77.676	67.295
Descisionn Tree	69.982	69.011	69.493	61.964
Random Forest	73.305	84.413	78.468	70.918



Recommendations

Doing more experiments

on our data might give us

more accurate prediction !

Thank you

For your kind attention