Credit Card Debit Card Customer Behavior Mapping

Problem Statement: Most of the banks run promotional campaign only for Credit Card customers but almost none for Debit card customer. In reality there are many Debit card customer showing the true behavior of Credit Card customers. So now bank wants to run personalized promotional campaign for Debit Card customers who all are showing true Credit card behavior.

Objective:

1. Find the Debit Card customers who all are showing true Credit Card behavior and use Maya.ai (product personalization recommendation system) for creating campaign for all these customers. Score each and every customer and sort them.

2. Use threshold on the above data and use it for Credit Card Acquisition.

3. As we have clean data, i.e. people who only have Credit card and people who only have Debit card and people who have both Debit Card and Credit Card, we understand the behavior of people who have both cards and build model to predict whether they have credit card or not. The main aim to build this model is to find that the customer who owns only debit card may have Credit Card of different bank. So bank this information for calibrating its customer Life Time Value (LTV.)

HYPOTHESIS & TEST

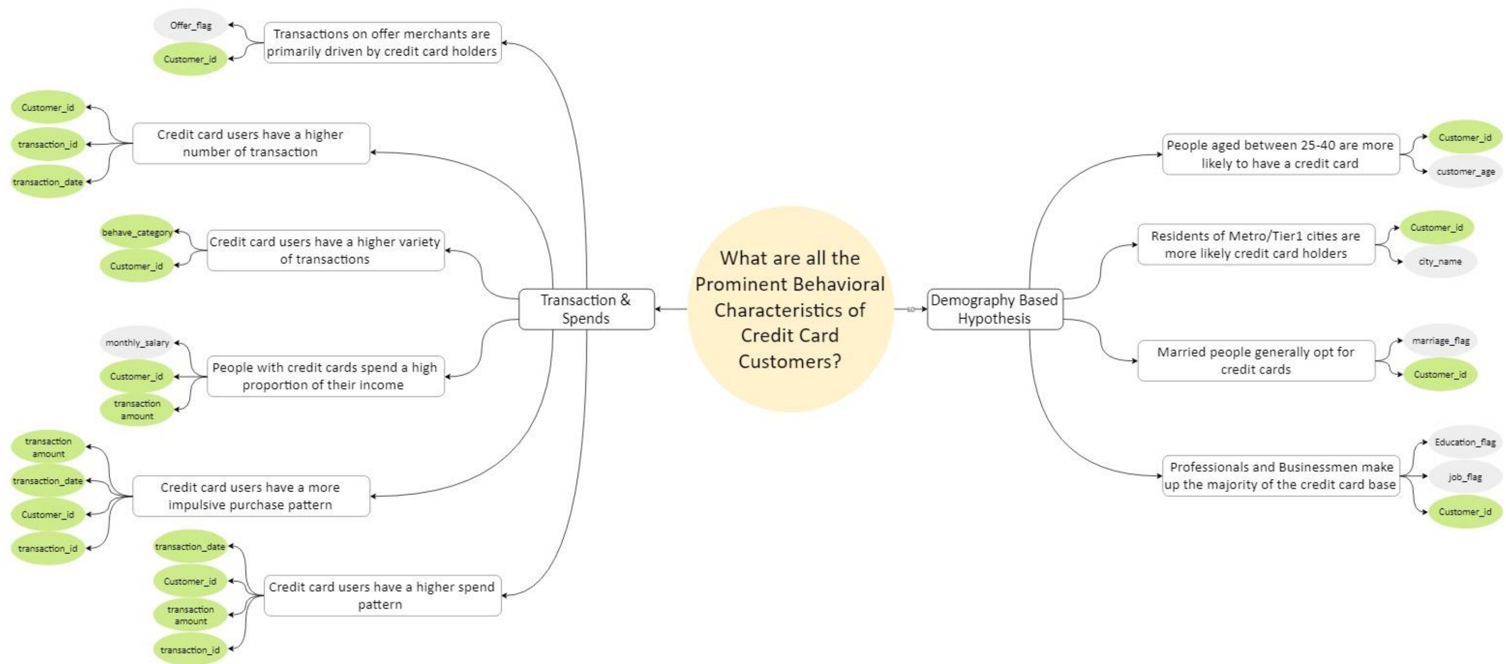
How do we help banks identify debit card customers to offer a credit card to?

Why do they need this?

Credit cards make money, credit card acquisition costs money. If acquisitions are done without intelligence, the ROI is poor (sometime negative), rendering the entire exercise futile

How does this work

Use our ensemble classification (ML based) model that learns the latent behavioral and spend patterns from the bank's existing credit card base and uses this to identify debit card customers for acquisition

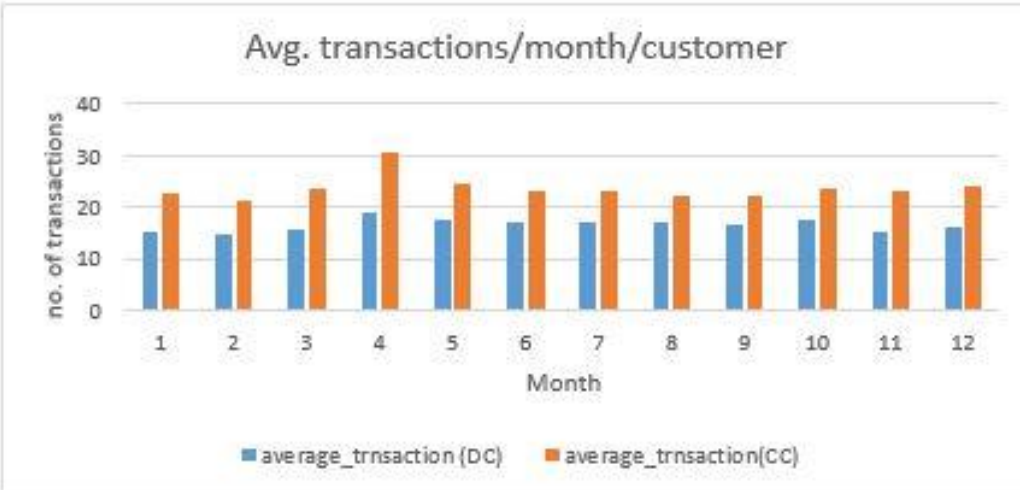


Sl. No.	test	predictor variables	test type aplha=0.05	Result t	p	Accepted / Rejected
1	cc cus	total_tran_12	2 sample t test	80.038	0	A
2	cc cus	total_spends_12	2 sample t test	13.676	0	A
3	cc cus	total_behav_category	2 sample t test	117.45	0	A
4	for cc customers no. of atm withdrawals are less than dc		2 sample t test			
5	for cc customers spend/income is higher than dc		2 sample t test			
6	freq of point redemption higher for cc as compared to dc		2 sample t test			
7	cc cus	change	2 sample t test	5.3907	0	A
8	cc customers avail offers more than dc		2 sample t test			
9	age gı	total_tran_amount	2 sample t test	2.099467	0.0357	A
10	age gı	total_tran_12	2 sample t test	24.3539	0	A
11	cc hol	lifestyle	2 sample t test	30.129	0	A
12	wome	lifestyle	2 sample t test	2.760348	0.005779	R
13	age gı	lifestyle	2 sample t test	-2.618	0.0088	R
14	cc hol	ticket_size	2 sample t test	46.7214	0	A
15	wome	ticket_size	2 sample t test	-1.5822	0.1136	R
16	wome	ticket_size	2 sample t test	-6.64	0	R
17	25-38	ticket_size	2 sample t test	-4.122	0.000038	R
18	more no. of customers having a steady income are cc as compared to dc					
19	marrie	total_tran_amount	2 sample t test	0.00535	0.995737	R
20	marrie	total_tran_amount	2 sample t test	-0.028627	0.977177	R
21	age gı	total_tran_amount	2 sample t test	7.73	0	A
22	age gı	total_tran_12	2 sample t test	3.22	0.001	A

VISIUAL DATA REPRESENTATAION

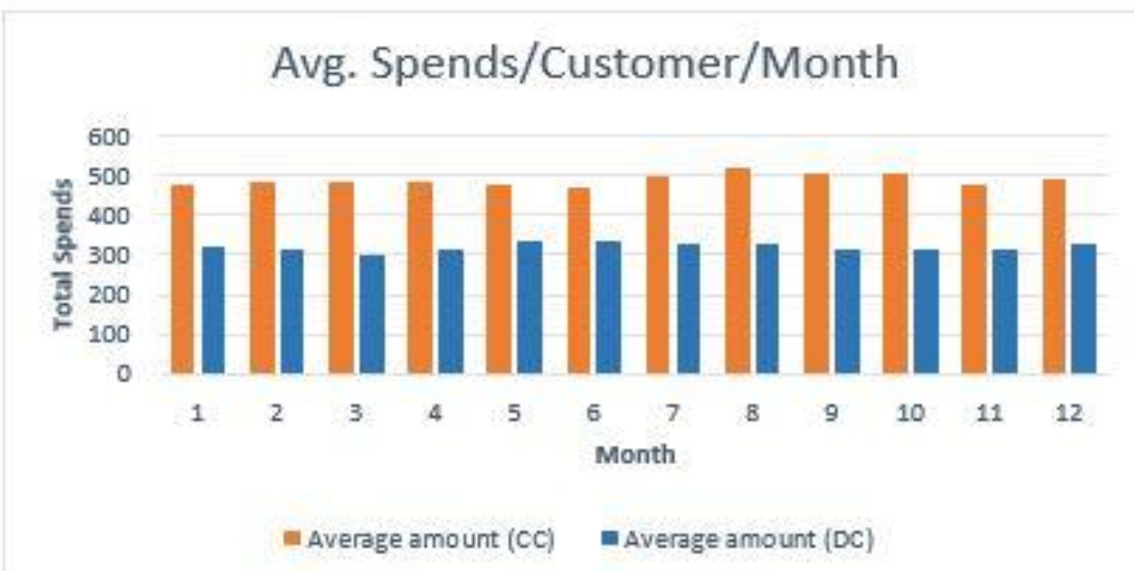
1 **Credit card customers transact more than debit card customers**

Conclusion TRUE



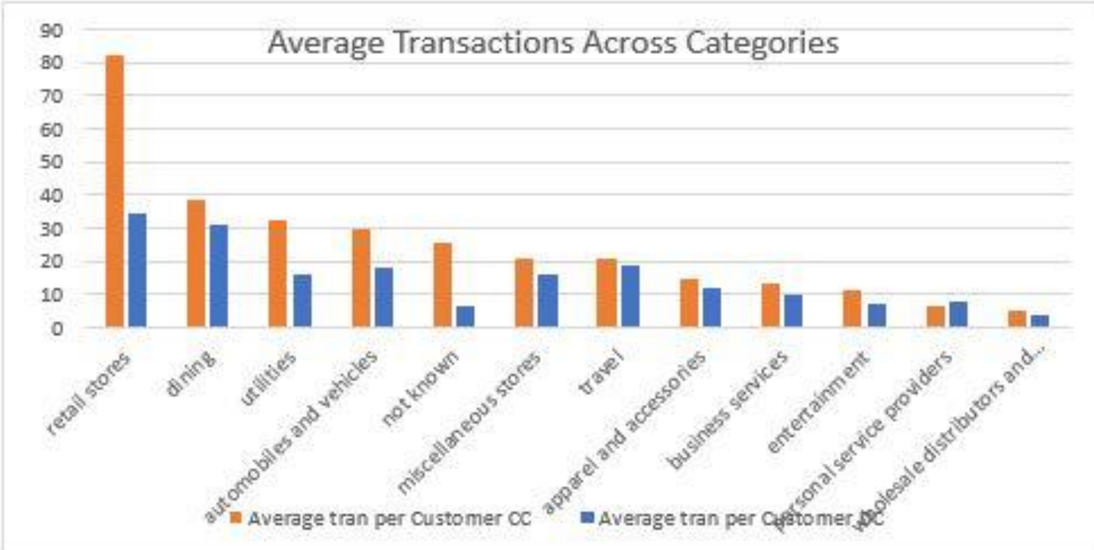
2 **Average monthly spends of CC customers is more than the DC customers**

Conclusion TRUE



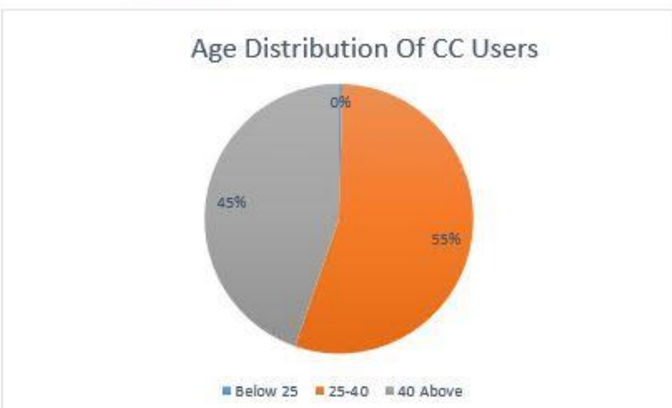
3 **Average transaction across different categories of credit card users are more as compare to debit card users**

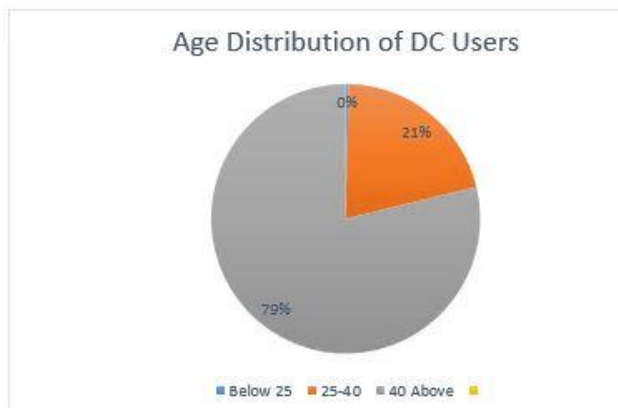
Conclusion TRUE



4 **Customer in age range 25-40 are more likely to be the credit card customer**

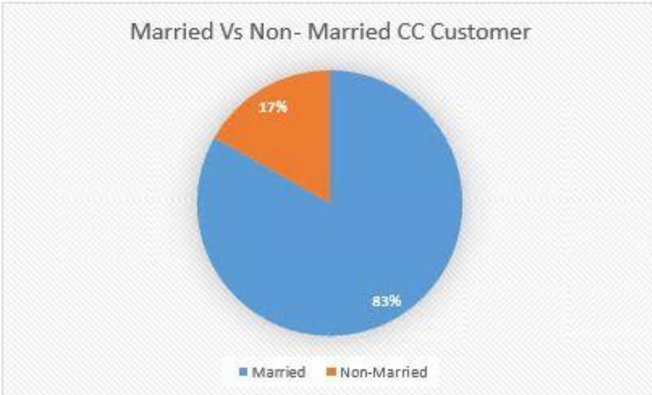
Conclusion TRUE

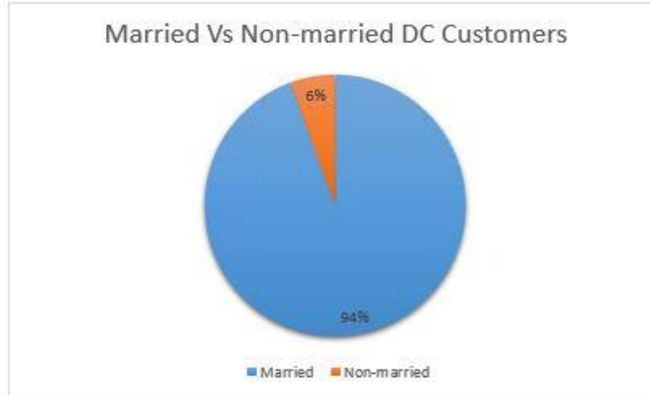




5 Married people are more likely to take credit card

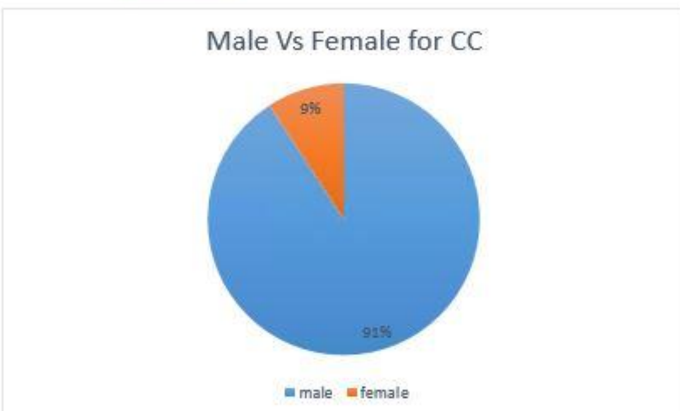
Conclusion TRUE

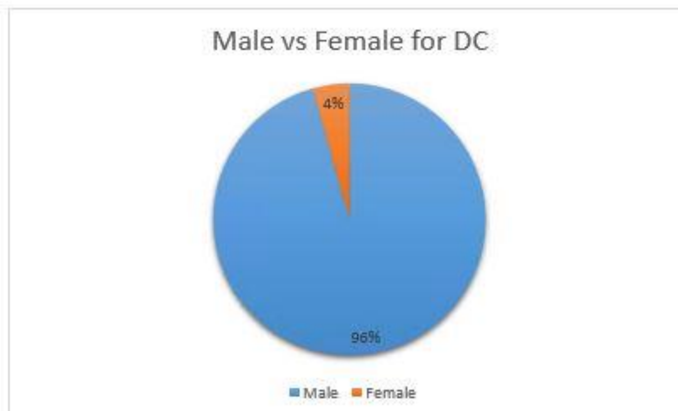




6 Majority of the CC holders are Male

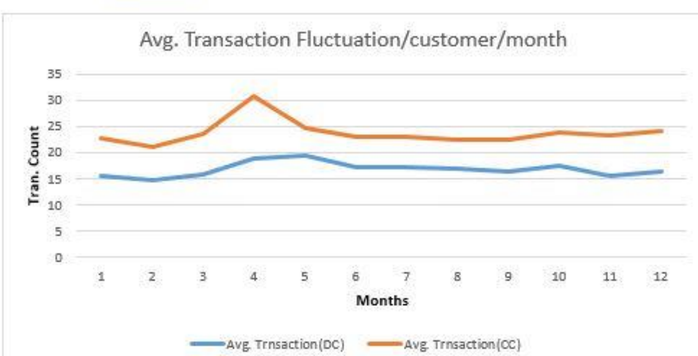
Conclusion TRUE

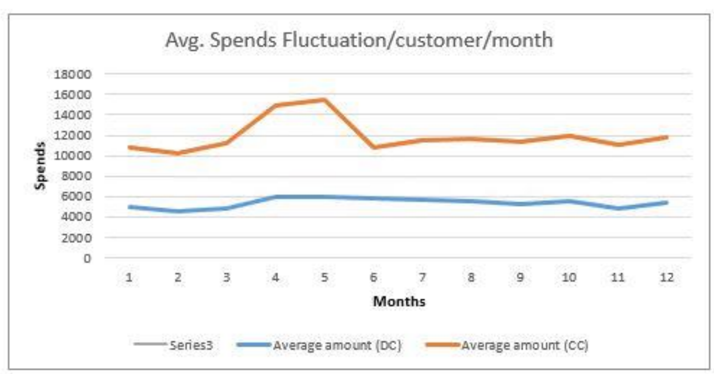




7 CC customer have high fluctuation in monthly spends and transaction count

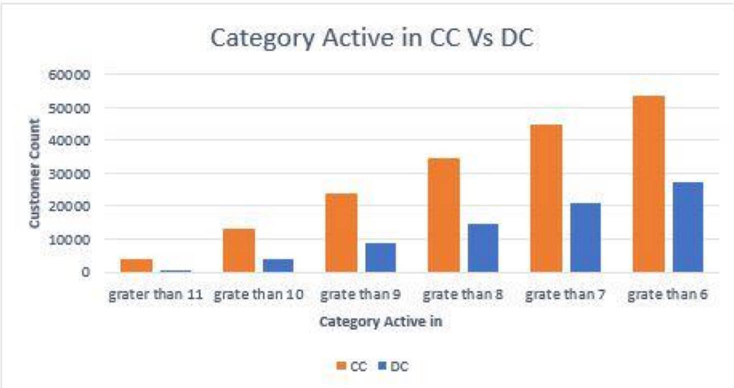
Conclusion TRUE





8 Credit card customers are active in more categories as compare to Debit card customers

Conclusion TRUE



	Behave Category
1	retail stores
2	dining
3	utilities
4	automobiles and vehicles
5	not known
6	miscellaneous stores
7	travel
8	apparel and accessories
9	business services
10	entertainment
11	personal service providers
12	wholesale distributors and manufacturers

MODELING:

- We are going to use only positive data for training the model that is Credit Card customer behaviors data and use this data to score Debit Card customer on the basis of similar behavior match.
- As we are using only one class for training the model so we have to go for one class ML model approach . We are going to use One-Class-SVM and PU Bagging.

<u>For More Information :</u>

- One-Class-SVM : http://rvlasveld.github.io/blog/2013/07/12/introduction-to-one-class-support-vector-machines/

- PU Bagging: https://roywright.me/2017/11/16/positive-unlabeled-learning/

One-Class-SVM

jupyter One Class SVM Last Checkpoint: 07/02/2019 (autosaved)

Logout

File Edit View Insert Cell Kernel Widgets Help

Trusted | Python 3

In [106]:
```
# importing the libraries
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
```

In [83]:
```
#importing the data

# CC database with all features(Normalised)
cc = pd.read_csv("C:/Users/manish/Desktop/working/Temp/LOG_CC_TRAIN.csv")

# DC database with all features(Normalised)
dc = pd.read_csv("C:/Users/manish/Desktop/working/Temp/data_final_dc - LOG.csv")

# DC database with all features(non-normalised)
dc_all = pd.read_csv("C:/Users/manish/Desktop/working/TRAIN-TEST DATA/data_final_cc 86k.csv")
```

In [84]:
```
#Setting the threshold
#cc = cc.loc[(cc['total_spends_12'] >12) | (cc['total_tran_12'] >1000 )]
```

In [85]:
```
#Min_Max_Scaling for CC and DC data
from sklearn import preprocessing
min_max_scaler = preprocessing.MinMaxScaler()
dc = min_max_scaler.fit_transform(dc)
cc = min_max_scaler.fit_transform(cc)
```

In [86]:
```
#Fitting the model
from sklearn import svm
clf = svm.OneClassSVM(kernel = "rbf", nu = 0.1, gamma = 0.1)
clf.fit(cc)
```

Out[86]:
```
OneClassSVM(cache_size=200, coef0=0.0, degree=3, gamma=1, kernel='rbf',
            max_iter=-1, nu=0.1, random_state=None, shrinking=True, tol=0.001,
            verbose=False)
```

In [87]:
```
#Testing-Predicting
y_score = clf.score_samples(dc)
```

In [88]:
```
# Min_Max_Scaling for Result
from sklearn import preprocessing
y_score = y_score.reshape(-1,1)
min_max_scaler = preprocessing.MinMaxScaler()
y_score = min_max_scaler.fit_transform(y_score)
```

In [89]:
```
#Converting to DataFrame
y_score = pd.DataFrame(y_score)
y_score.rename(columns = {0:'Score'}, inplace = True)
```

In [90]:
```
#Concatig the Data
df = pd.concat([y_score,dc_all], axis = 1)
```

In [91]:
```
#Exporting the data to disk EXCEL fromate
df.to_excel("C:/Users/manish/Desktop/working/ITERATION DATA/Mashraque_result1.xlsx", index = False)
```

In [92]:
```
#Exporting the data to disk CSV formate
df.to_csv("C:/Users/manish/Desktop/working/ITERATION DATA/Mashraque_result15.csv", index = False)
```

In [93]:
```
#t =y_score.loc[y_score['Score']>0.5]
#t.count()
```

PU BAGGING

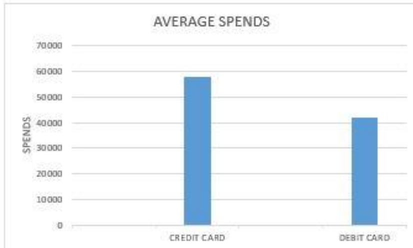
```
import pandas as pd
import numpy as np

#we will build 100 decision trees
from sklearn.tree import DecisionTreeClassifier
n_estimators=100
s=20000
estimator=DecisionTreeClassifier(max_depth=15)

#load data
df1=pd.read_csv("D:/amrita/PU bagging/data_final_cc.csv")

df2=pd.read_csv("D:/amrita/PU bagging/data_final_dc.csv")

#join data for cc and dc customers
df=pd.concat([df1,df2])

df=df.reset_index()
df=df.drop(['index'],axis=1)

#feature selection
X=df.loc[:,['apparel_and_accessories',
            'automobiles_and_vehicles',
            'business_services',
            'dining','miscellaneous_stores',
            'retail_stores',
            'total_behav_category',
            'total_spends_12',
            'total_tran_12',
            'travel',
            'utilities',
            'change',
            'ticket_size']]

#predictor variable
y=df.loc[:,['cardtype']]

#keep copy
y_orig=y.copy()
```

```
#for each data point record how many times it has been out of bag and the sum of  OOB scores
num_out_bag = pd.DataFrame(np.zeros(shape=y.shape),index=y.index)

sum_out_bag = pd.DataFrame(np.zeros(shape=y.shape),index=y.index)

#keep track of indices of positive and unlabelled data points
index_positive =(y.loc[y.cardtype ==1,:]).index
index_unlabeled =(y.loc[y.cardtype ==0,:]).index

for _ in range (n_estimators):
    #get a bootstrap sample of unlabelled data points for this round
    in_bag_index=np.random.choice(index_unlabeled,replace=True,size=s)

    #find out of bag data points for this round
    out_bag_index=list(set(index_unlabeled)-set(in_bag_index))

    #get training data(all positives and bootstrap sample of unlabelled points) and build the tree

    X_in_bag=(df.loc[df.cardtype==1,[
        'apparel_and_accessories',
        'automobiles_and_vehicles','business_services',
        'dining',
        'miscellaneous_stores',
        'retail_stores',
        'total_behav_category',
        'total_spends_12',
        'total_tran_12',
        'travel',
        'utilities',
        'change','ticket_size']]).append(X.loc[in_bag_index,:])

    y_in_bag=(df.loc[df.cardtype==1,['cardtype']]).append(y.loc[in_bag_index,:])

    estimator.fit(X_in_bag, y_in_bag)

    #record the OOB scores from this round
    num_out_bag.loc[out_bag_index,0]+=1
    sum_out_bag.loc[out_bag_index,0]+=estimator.predict_proba(X.loc[out_bag_index])[:,1]

```

```
#store the scores for each customer
results=sum_out_bag/num_out_bag

#assign manually probability 1 for cc data points
#results=results.fillna(1)

#map the results back to the customer_id and other features
pu_result=pd.concat([df,results],axis=1)

#filtering the dc user
pu_result = pu_result.loc[pu_result['cardtype'] == 0]

#pu_result.to_csv("D:/amrita/PU bagging/Final/pubaggingresult.csv")
```

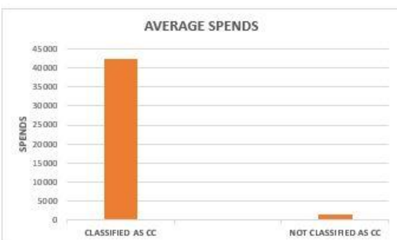
RESULTS

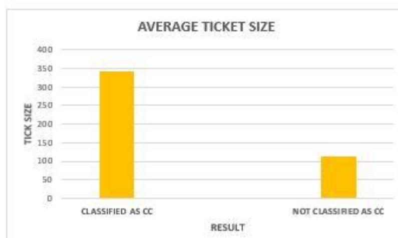
CREDIT CARD VS DEBIT CATRD

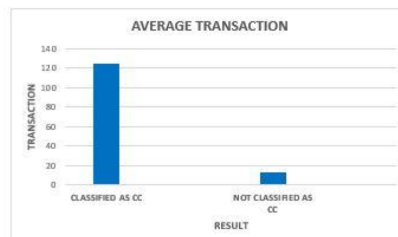


MODEL RESULT

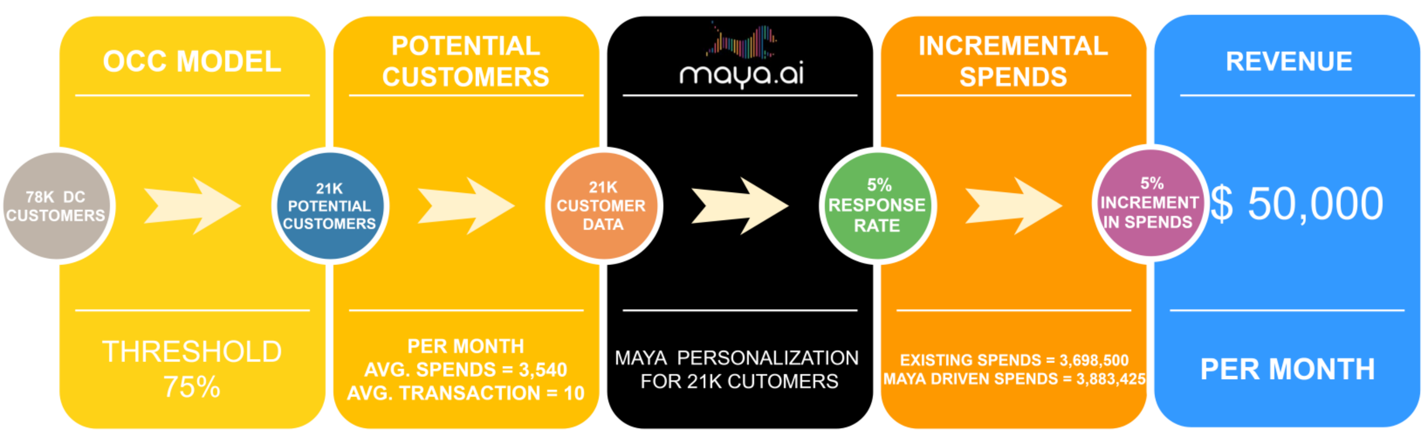
DC CLASSIFIED VS NOT CLASSIFIED



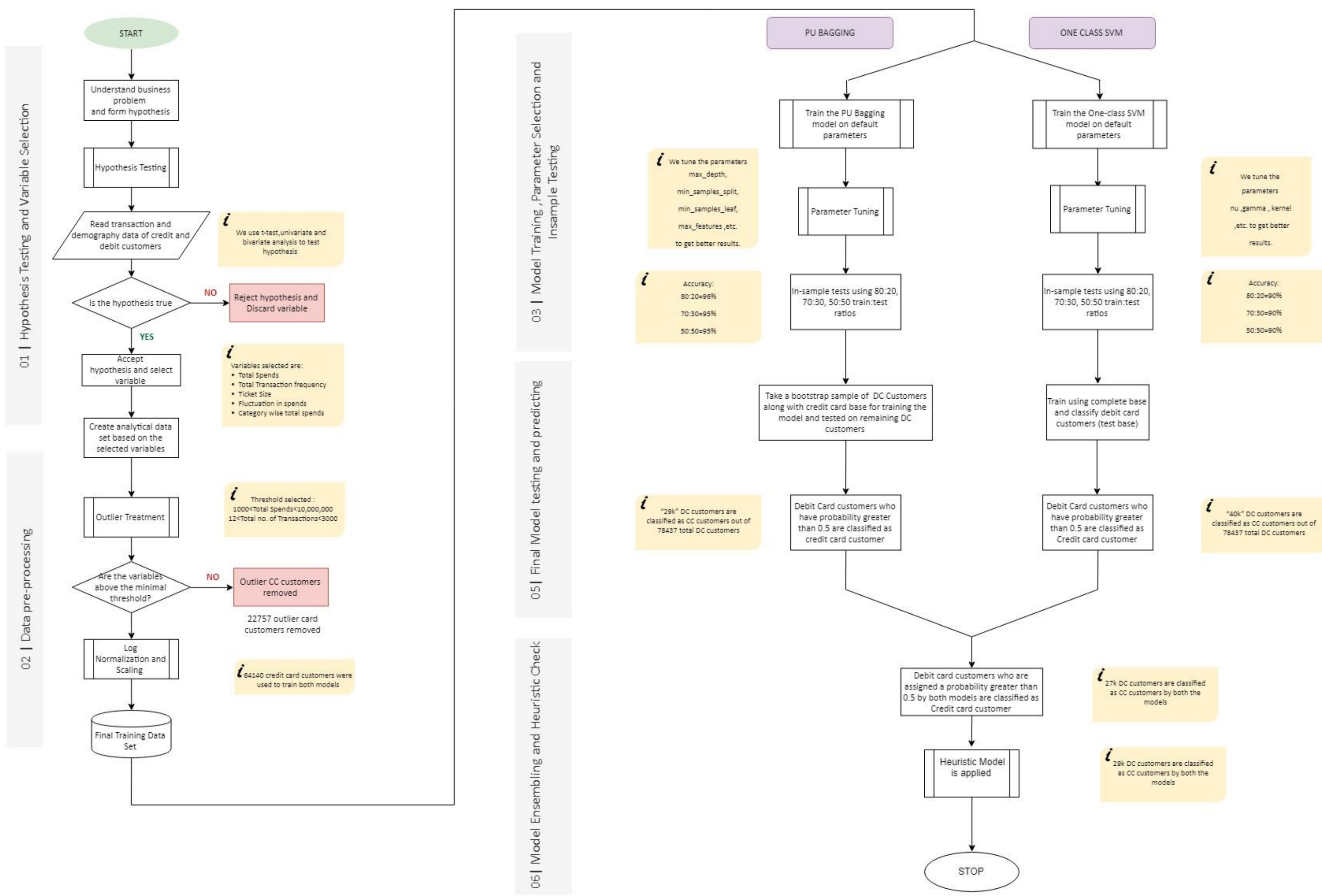




REVENUE MODEL



<u>WORK FLOW</u>



01 | Hypothesis Testing and Variable Selection
START
Understand business problem and form hypothesis
Hypothesis Testing
Read transaction and demography data of credit and debit customers
We use t-test,univariate and bivariate analysis to test hypothesis
Is the hypothesis true
NO
Reject hypothesis and Discard variable
YES
Accept hypothesis and select variable
Variables selected are:
• Total Spends
• Total Transaction frequency
• Ticket Size
• Fluctuation in spends
• Category wise total spends
Create analytical data set based on the selected variables
02 | Data pre-processing
Outlier Treatment
Threshold selected : 1000<Total Spends<10,000,000 12<Total no. of Transactions<3000
Are the variables above the minimal threshold?
NO
Outlier CC customers removed
22757 outlier card customers removed
Log Normalization and Scaling
64140 credit card customers were used to train both models
Final Training Data Set
03 | Model Training , Parameter Selection and Insample Testing
PU BAGGING
ONE CLASS SVM
Train the PU Bagging model on default parameters
Train the One-class SVM model on default parameters
We tune the parameters max_depth, min_samples_split, min_samples_leaf, max_features ,etc. to get better results.
Parameter Tuning
Parameter Tuning
We tune the parameters nu ,gamma , kernel ,etc. to get better results.
Accuracy: 80:20=96% 70:30=95% 50:50=95%
In-sample tests using 80:20, 70:30, 50:50 train:test ratios
In-sample tests using 80:20, 70:30, 50:50 train:test ratios
Accuracy: 80:20=90% 70:30=90% 50:50=90%
05| Final Model testing and predicting
Take a bootstrap sample of DC Customers along with credit card base for training the model and tested on remaining DC customers
Train using complete base and classify debit card customers (test base)
"29k" DC customers are classified as CC customers out of 78437 total DC customers
Debit Card customers who have probability greater than 0.5 are classified as credit card customer
Debit Card customers who have probability greater than 0.5 are classified as Credit card customer
"40k" DC customers are classified as CC customers out of 78437 total DC customers
06| Model Ensembling and Heuristic Check
Debit card customers who are assigned a probability greater than 0.5 by both models are classified as Credit card customer
27k DC customers are classified as CC customers by both the models
Heuristic Model is applied
29k DC customers are classified as CC customers by both the models
STOP