# Starbucks - Capstone Project

## Introduction

This data set contains simulated data that mimics customer behavior on the Starbucks rewards mobile app. Once every few days, Starbucks sends out an offer to users of the mobile app. An offer can be merely an advertisement for a drink or an actual offer such as a discount or BOGO (buy one get one free). Some users might not receive any offer during certain weeks.

Not all users receive the same offer, and that is the challenge to solve with this data set.The task here is to combine transaction, demographic and offer data to determine which demographic groups respond best to which offer type. This data set is a simplified version of the real Starbucks app because the underlying simulator only has one product whereas Starbucks actually sells dozens of products.

## Data Sets

The data is contained in three files:

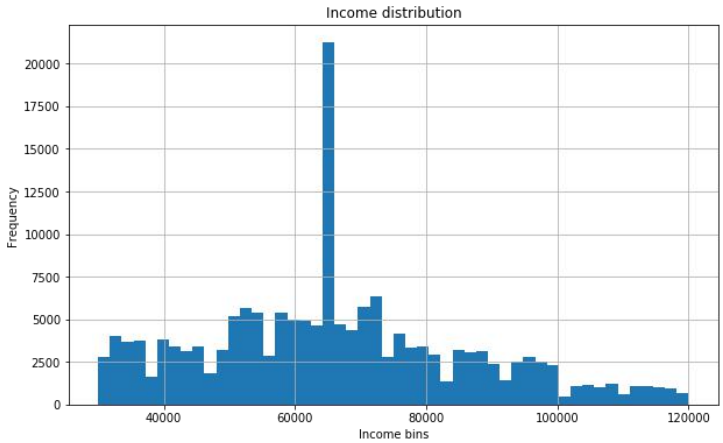
**portfolio.json** - containing offer ids and meta data about each offer (duration, type, etc.)
**profile.json** - demographic data for each customer
**transcript.json** - records for transactions, offers received, offers viewed, and offers completed

## Idea of Workflow

So the basic idea here is to preprocess each dataframe seperately in order to derive features from them and then combine all the 3 dataframes to form a singlr dataframe which can be fed to the model.This model will predict the respective target variable given to it.

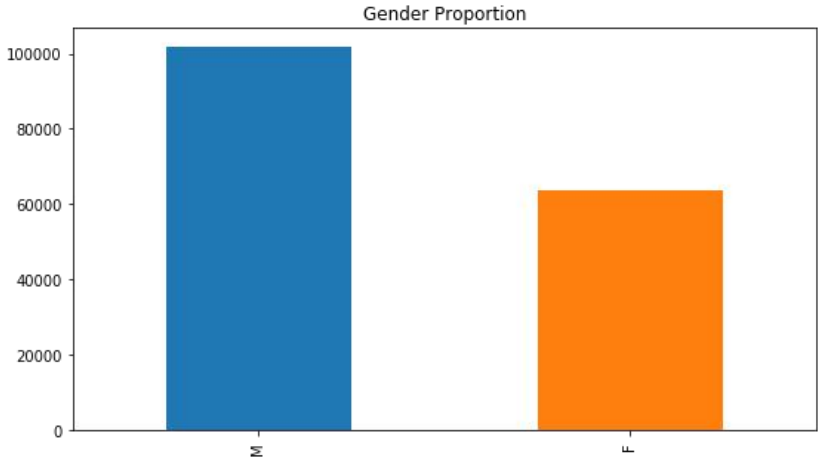
## Insights from the data

### a) Income distribution



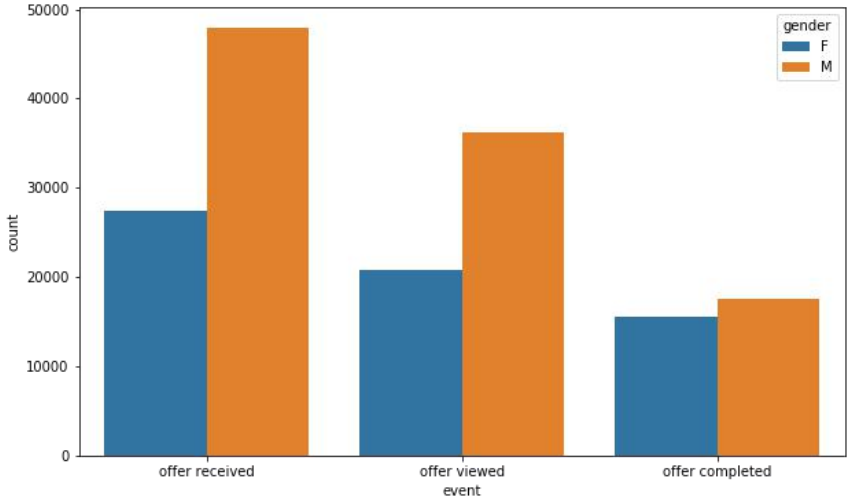

The income distribution looks like a skewed normal distribution

### b) Gender based distribution

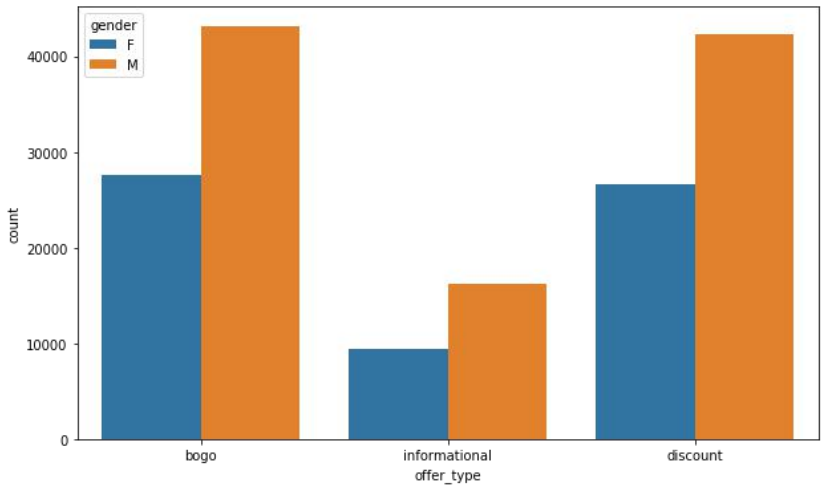



So we have higher male proportion than females

## (i) Event type based on gender

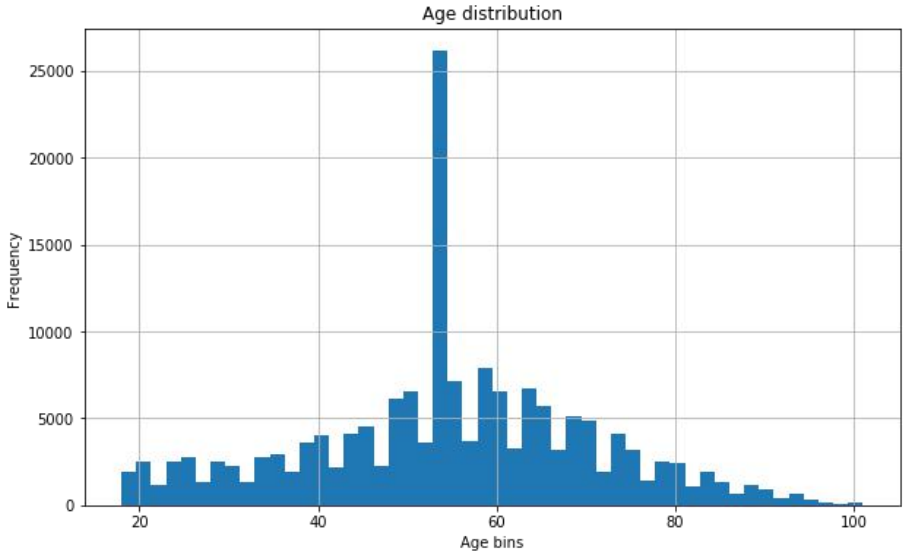



## (ii) Offer type based on gender




The count of Male dominated over female in every individual subsection as well.

## c) Age distribution




The age distribution is similar to that of a normal distribution curve

## Mapping strings with integers
## Reason :

The model cannot process features in string format.so we need to convert it into numbers so that they ccn be used in equations to calculate the cost function which should be optimized in order totrain the model for better predictions.This is done by map finction

## Modelling

The models that will be trained and benchmarked for two different cases are as follows,

1) **KNeighborsClassifier** - Classifier implementing the k-nearest neighbors vote.

2) **RandomForestClassifier** - A random forest is a meta estimator that fits a number of decision tree classifiers on various sub-samples of the dataset and uses averaging to improve the predictive accuracy and control over-fitting.

3) **AdaBoostClassifier** - An AdaBoost classifier is a meta-estimator that begins by fitting a classifier on the original dataset and then fits additional copies of the classifier on the same dataset but where the weights of incorrectly classified instances are adjusted such that subsequent classifiers focus more on difficult cases.

4) **ExtraTreesClassifier** - An extra-trees classifier is a meta estimator that fits a number of randomized decision trees (a.k.a. extra-trees) on various sub-samples of the dataset and uses averaging to improve the predictive accuracy and control over-fitting.

5) **BaggingClassifier** - A Bagging regressor is an ensemble meta-estimator that fits base regressors each on random subsets of the original dataset and then aggregate their individual predictions (either by voting or by averaging) to form a final prediction. Such a meta-estimator can typically be used as a way to reduce the variance of a black-box estimator (e.g., a decision tree), by introducing randomization into its construction procedure and then making an ensemble out of it.

## a) Preconditioning the model --- Min-Max Scaling

The data features can have different range of values and their absolute values have a big impact on performance of the model.So the significance of different features can be balanced by rescaling these range of values to a same interval.

## b) Principal Component Analysis (PCA)

This is done to identify the features which are highly connected to the target variable,so that we can reduce the number of features by maintaining the required variance and that make the model training even more efficient.

Number of Principle Components : 16
Variance Captured : 0.991165564251

## Inference

So with 16 features we were able to capture almost 99% of the variance out of the non linear data.In case of too many features we can remove the unnecessary features and only give the required one to the model.In this case as we only have a few more than 16,lets give it all to the model and see how it performs with that.

## c) Test-Train Splitting

Now the data should be splitted into test set and train set inorder to validate the predicting power of the ML model.Here we will be performing an 80%-20% split.Initially we need to choose a target variable which would be predicted by the model.

## d) Metrics and scoring -- quantifying the quality of predictions

Out of many metrics available in the sklearn framework,accuracy_score was selected as the evaluation metric as it is considered as the weighted average of precision and recall.The best performing model will have a accuracy value near 100 and the worsmodel will have accuracy_score closer to 0.

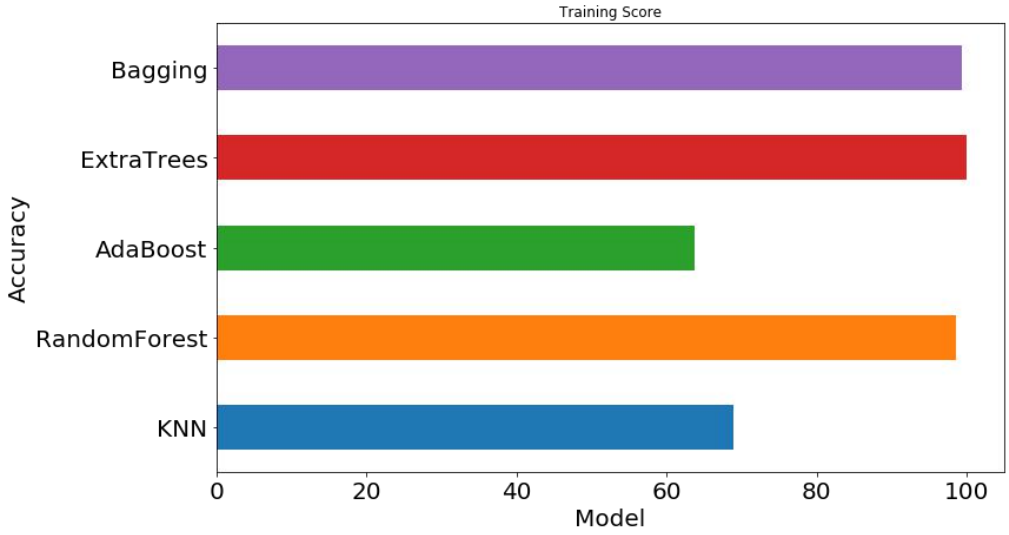
# Case 1

### Target Variable - Offer Type

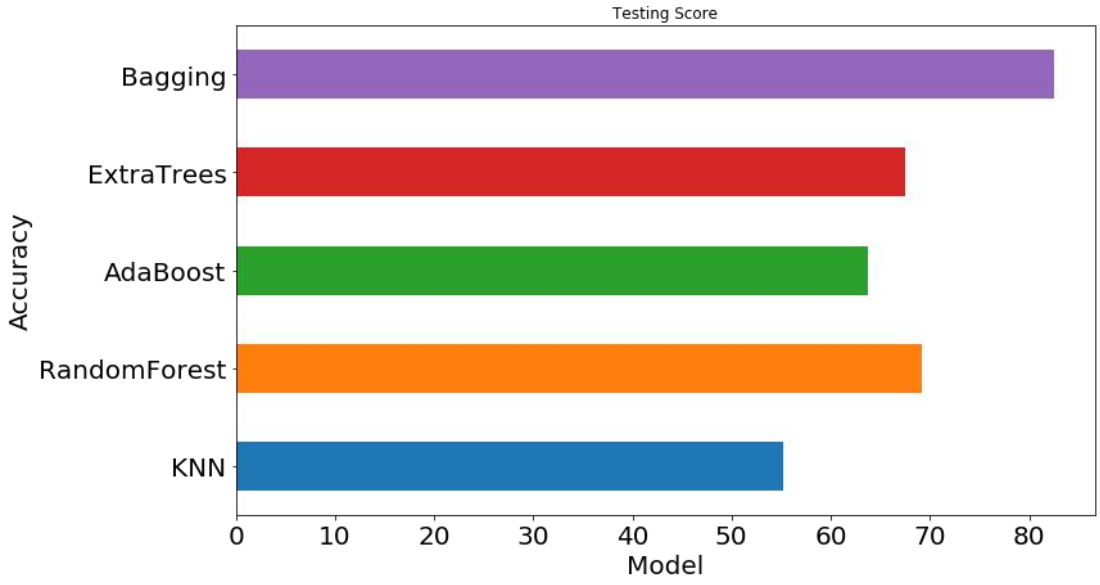
Here the model predicts the offer_type using all the other features.ie it classiffies the data into corresponding offer types,ie

a) bogo
b) discount
c) informational

## Training score comparison




## Testing score comparison




## Inference

This shows that apart from KNN model and AdaBoost model all the other three model performs really well to predict the type of offer.The Highest test_score is shown by Bagging Classifier which is the best to predict the type of offer.In predicitng the offer type,bagging classifier has the highest training as well as testing accuracy.So that wins the battle among all the other models in this particular case.

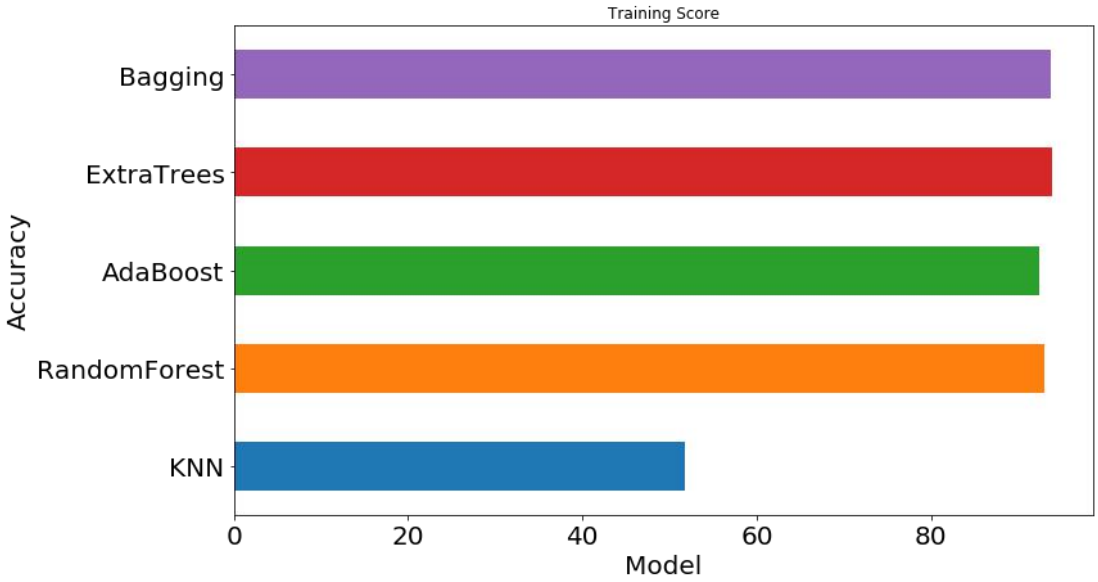
# Case 2

**Target Variable - Event type**

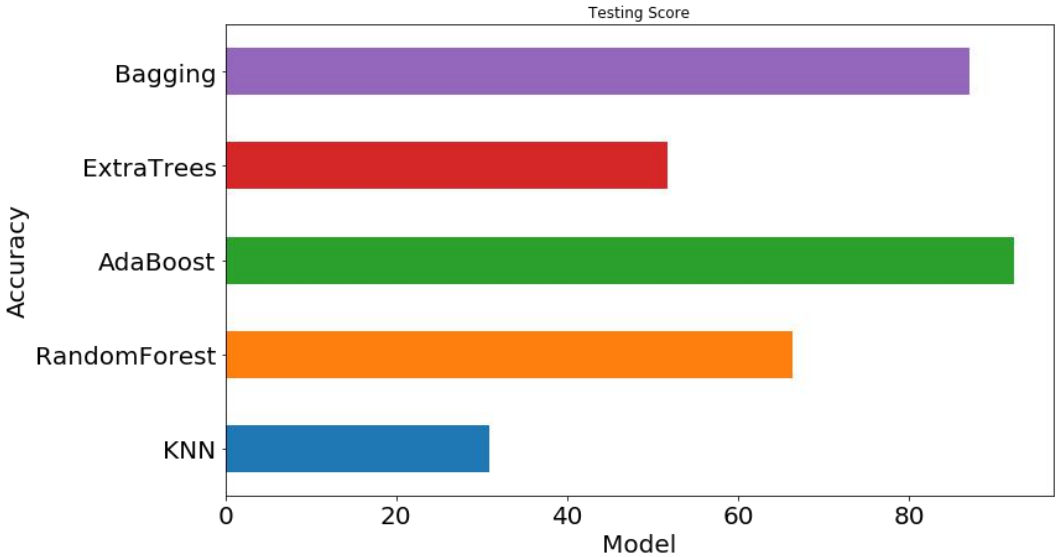
Here the model predicts the event_type using all the other features.ie it classiffies the data into corresponding event types,ie

a) offer received
b) offer viewed
c) offer completed

## Training score comparison




## Testing score comparison




## Inference

Here apart from KNN model every other model performs really well with accuracy above 90% in the case of training dataset.This may be due to the effective data preprocessing followed by the exploratory data analysis.But when it comes to validation of the model performance using the testing dataset,only the AdaBoost model performs well.So Adaboost classifier is the best one for this case.The performance of other models might have degraded due to the issue of overfitting.This can be rectified using addition of noise or with some regularization techniques.

# Conclusion

Initially the data in the form of 3 dataframes was processed separately using different techniques.Features were extraxcted from these individual dataframes and then cleaned to remove null values,infinite values etc.Now Feature engineering was performed in which diferent columns were extracted from some columns to make the data more readable to the model.One the individual dataframes were cleaned and modified,all of them were merged to form a single dataframe.Now this was fed to the models.So the model training and predictions were carried out for two cases by changing the target variable so that the models could predict multiple entities seperately.Initilly the model predicts the type of offer and then the type of event.5 different ML models were taken from Sklearn framework and were compared using the merged dataframe.The results of the model predicitions were also were also generated in order to find the model which performed well.regarding the scopes for improvement,The data set could have been made better by eliminating so many null values which I had to replace with appropriate counterparts.This would have reduced the quality of data.Other than that if we had more features to work with,we could have done PCA on a better data pool and obtained better results.

# References


1) https://stackoverflow.com/
2) https://pandas.pydata.org/docs/reference/index.html
3) https://pandas.pydata.org/docs/reference/api/pandas.get_dummies.html
4)Changing datetime format : https://pandas.pydata.org/docs/reference/api/pandas.to_datetime.html
5) Merging dataframes along column : https://pandas.pydata.org/pandas-docs/stable/user_guide/merging.html
6) One hot encoding : https://stackabuse.com/one-hot-encoding-in-python-with-pandas-and-scikit-learn/
7) Mapping in pandas : https://kanoki.org/2019/04/06/pandas-map-dictionary-values-with-dataframe-columns/
8) Min-Max scaling : https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.MinMaxScaler.html
9) Metrics : https://scikit-learn.org/stable/modules/classes.html#module-sklearn.metrics
10) Factor Analyzer : https://pypi.org/project/factor-analyzer/
11) Dropping infinite values : https://statisticsglobe.com/drop-inf-from-pandas-dataframe-in-python
12) SkLearn ensembles : https://scikit-learn.org/stable/modules/classes.html#module-sklearn.ensemble
13) https://stackoverflow.com/questions/12444716/how-do-i-set-the-figure-title-and-axes-labels-font-size-in-matplotlib
14) https://pandas.pydata.org/docs/reference/api/pandas.DataFrame.set_index.html
15) Visualization : https://pandas.pydata.org/pandas-docs/stable/user_guide/visualization.html#visualization-barplot
16) List of list to dataframe : https://datascience.stackexchange.com/questions/26333/convert-a-list-of-lists-into-a-pandas-dataframe