Starbucks Capstone Project Report

Udacity - Machine Learning Engineer Nanodegree

Muhammad Umar Farrukh

June 2024

1 Definition

1.1 Project Overview

Starbucks is one of the word renowned coffee, they have huge variety of hot and cold beverages. They have their main head offices in Seattle, USA. They have IOS and Android mobile applications which allow their customers to order from anywhere in the world. These apps allow their customers to find the nearest Starbucks coffee shop and pick the beverage at their convenience. Their app have alot of promotional offers going on from time to time. They do alot of advertisements for different seasonal drinks and buy one get one free offers. This capstone project will help tailor the promotional offers to specific customers who would be likely to respond to that offer

Main three varieties of offers are covered in the portfolio section which are

1. BOGO referring to Buy One Get One in which user has to spend certain amount to qualify for the reward
2. Discount offer helps user gain a fractional amount of the total amount he has spend
3. Informational offer provides information to customer about certain offer being activated but doesnot has any reward or amount linked with it Refrence for these three types of offers was checked using valuecount function of data frame and a snippet is provided below to show the offers available in portfolio

Similarly communication of these offers is made through four channels which is through web, email, mobile and social media. These all are digital based channels through which communication is being made but time and value is spend on these communications so an ROI is expected once these promotional offers are being sent to the customers. That is where consumer preference based marketing campaigns implemented through Artificial Intelligence will come in action [2]. Similarly in this project I will try to build a machine learning model that will tailor the personalised offer for a customer and will help to reduce down the costs along with a increased revenue. This project is very important part for my learning as it will add huge benefit for me and my career, i have also read several other research papers such as Machine learning in marketing [11] defines the purpose and ambition of mahcine learning behind marketing campaigns, similarly another research paper [8] discusses the importance of self learning models in neuro marketing campaigns. These researches and the overall data quest led me to choose this project for my capstone project. Similarly antoher research paper [13] which discusses the campaigns launched by Starbucks on twitter or X and how the sentiments of the customers are studied using the machine learning models.

1.2 Problem Statement

Main Objective of this project is to tailor offers according to customers based on their responses to previous offers sent to them. All users wont be receiving the same offers, in this case a machine learning model implementation will help determine the best offer for a customer and their response to that offer. Data set consists of 30 days of promotional offers being given to the customers and how they have utilized those offers. Purchase offers sent to these customers was captured using feature name Event which had 4 main categories

1. Offer Received
2. Offer Viewed
3. Transaction
4. Offer Completed

This was available in transcript data frame which was then connected with portfolio and profile data frames.

1.3 Metrics

Metrics used for determining the model overall accuracy were Precision, Recall, F1 score. Since our target variable consisted of four classes so the average of overall classes for these metrics represented how well the model was performing. Precision is the proportion of True Positive - TP (predictions which are actually true) by True Positive adding False Positive [6]

$$Precision = \frac{TP}{TP + FP} \tag{1}$$

, Recall or sensitivity is the proportion of all true positive plus False Negative [6]

$$Recall = \frac{TP}{TP + FN} \tag{2}$$

. F1 score is the combination of the recall and precision attributing the harmonic mean of both. It is cored between 0 and 1 where 0 indicating worst and 1 indicating perfect precision.

$$F1 = \frac{2 * Precision * Recall}{Precision + Recall} \tag{3}$$

[1]. Ididnot account accuracy metric only in our model performance evaluation as that metric cannot justify the proportions among imbalanced data and also wont cover overfitting and granularity

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{4}$$

2 Analysis

Using Machine Learning techniques such as Exploratory Data Analysis, Hypothesis creation , Model Analysis and predictability will help me to achieve this task of finding most relevant offer according to customers need. Using the EDA technique i will visualize which offers were most accepted by the customers, how well they responded to it and what particular demographics are responding to most offers. Similarly hypothesis testing will help me to check different models and their outputs with the most important feature groups i have selected for that hypothesis. I ll start off with very basic models than advance to complex tree based models such as random forest or xgboost or catboost. For training purposes 80 percent of the data would be separated and trained with the basic models using aws sage maker notebook whereas 20 percent of the data would be sepertaed for testing purpose and that will be done also using standalong sage maker notebooks. In sage-maker notebooks all the cells will be executed and bench mark models will be saved from training practice to s3 bucket using pickle library and then would be reloaded and tested in the sagemaker notebook again.

2.1 Data Exploration

Data set is provided in three main json files which represents that most data is in form of unstructured data. The three main files provided are profile.json, portfolio.json and transcript.json.

Portfolio.json will help us to analyse what was offered to customer for how much duration, and what was the reward given to customer and through what channel communication which will help to analyse what is most relevant channel for communication suited for customer. Some analysis and shape of the portfolio data frame is shown below in the images. Using the value counts of offer type in portfolio helped us to know what three types of offers are available, similarly difficulty variable descriptive analysis helped us to know more about the variation of difficulty levels associated with the offers. portfolio.json

1. id (string) - offer id

2. offertype (string) - type of offer ie BOGO, discount, informational
3. difficulty (int) - minimum required spend to complete an offer
4. reward (int) - reward given for completing an offer
5. duration (int) - time for offer to be open, in days
6. channels (list of strings)

```
portfolio.shape
```

```
(10, 6)
```

```
portfolio['offer_type'].value_counts().to_frame().style.bar()
```

	offer_type
bogo	4
discount	4
informational	2

```
portfolio.head()
```

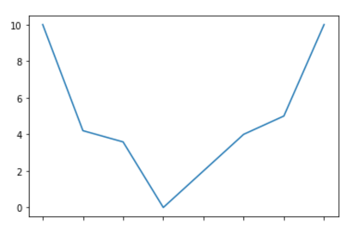
	channels	difficulty	duration	id	offer_type	reward
0	[email, mobile, social]	10	7	ae264e3637204a6fb9bb56bc8210ddfd	bogo	10
1	[web, email, mobile, social]	10	5	4d5c57ea9a6940dd891ad53e9dbe8da0	bogo	10
2	[web, email, mobile]	0	4	3f207df678b143eea3cee63160fa8bed	informational	0
3	[web, email, mobile]	5	7	9b98b8c7a33c4b65b9aebfe6a799e6d9	bogo	5
4	[web, email]	20	10	0b1e1539f2cc45b7b9fa7c272da2e1d7	discount	5

```
portfolio['difficulty'].describe()
```

```
count    10.000000
mean      7.700000
std       5.831905
min       0.000000
25%       5.000000
50%       8.500000
75%      10.000000
max      20.000000
Name: difficulty, dtype: float64
```

```
portfolio['reward'].describe().plot()
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7d63fc418128>
```



Profile data consists of demographics features of the customers fore.g what is their age, gender, income, id which will help to tailor specific event based offers for customers, schema of profile data frame is shown below in the bullet points. profile.json

1. age (int) - age of the customer

2. becamememberon (int) - date when customer created an app account
3. gender (str) - gender of the customer (note some entries contain 'O' for other
4. rather than M or F)
5. id (str) - customer id
6. income (float) - customer's income

Some data analysis of the profile data frame is shown below, in which age,gender,income and shape of data frame is checked:

```
profile.head()
```

	age	became_member_on	gender	id	income
0	118	20170212	None	68be06ca386d4c31939f3a4f0e3dd783	NaN
1	55	20170715	F	0610b486422d4921ae7d2bf64640c50b	112000.0
2	118	20180712	None	38fe809add3b4fcf9315a9694bb96ff5	NaN
3	75	20170509	F	78afa995795e4d85b5d9ceeca43f5fef	100000.0
4	118	20170804	None	a03223e636434f42ac4c3df47e8bac43	NaN

```
profile.shape
```

```
(17000, 5)
```

```
]: profile.shape
]: (17000, 5)
```

```
]: profile['age'].describe()
]: count    17000.000000
   mean        62.531412
   std         26.738580
   min         18.000000
   25%         45.000000
   50%         58.000000
   75%         73.000000
   max        118.000000
   Name: age, dtype: float64
```

```
]: profile['gender'].value_counts().to_frame().style.bar()
]:
```

	gender
M	8484
F	6129
O	212

```
]: profile['income'].describe()
]: count     14825.000000
   mean      65404.991568
   std       21598.299410
   min       30000.000000
   25%       49000.000000
   50%       64000.000000
   75%       80000.000000
   max      120000.000000
   Name: income, dtype: float64
```

Third is the transcript.json data which will help us to analze how the customer reacted on that offer and wether he went for that offer or not. Important key id is customer id which will help connect all these three data together and aggregate the values at a customer level. transcript.json

1. event (str) - record description (ie transaction, offer received, offer viewed, etc.)
2. person (str) - customer id
3. time (int) - time in hours since start of test. The data begins at time t=0
4. value - (dict of strings) - either an offer id or transaction amount depending on the record

```
transcript.head()
```

	event	person	time	value
0	offer received	78afa995795e4d85b5d9ceeca43f5fef	0	{'offer id': '9b98b8c7a33c4b65b9aebfe6a799e6d9'}
1	offer received	a03223e636434f42ac4c3df47e8bac43	0	{'offer id': '0b1e1539f2cc45b7b9fa7c272da2e1d7'}
2	offer received	e2127556f4f64592b11af22de27a7932	0	{'offer id': '2906b810c7d4411798c6938adc9daaa5'}
3	offer received	8ec6ce2a7e7949b1bf142def7d0e0586	0	{'offer id': 'fafdcd668e3743c1bb461111dcafc2a4'}
4	offer received	68617ca6246f4fbc85e91a2a49552598	0	{'offer id': '4d5c57ea9a6940dd891ad53e9dbe8da0'}

```
transcript['event'].value_counts()
```

```
transaction        138953
offer received      76277
offer viewed        57725
offer completed     33579
Name: event, dtype: int64
```

```
transcript['time'].describe()
```

```
count    306534.000000
mean        366.382940
std         200.326314
min           0.000000
25%         186.000000
50%         408.000000
75%         528.000000
```

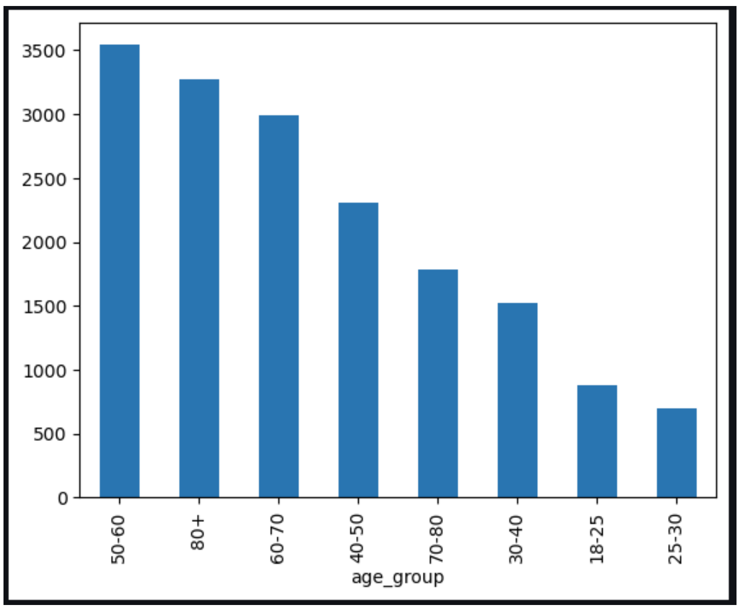
Third is the transcript.json data which will help us to analze how the customer reacted on that offer and wether he went for that offer or not. Important key id is customer id which will help connect all these three data together and aggregate the values at a customer level.

2.2 Data Cleaning

Once all the data from these three data frames was loaded, Each data frame was throughly checked and pre-processing steps were implemented. In each data frame, first data types were checked, second null values were checked, third imputation of those null values using mean and mode statistics techniques, conversion of features into new features using encoding techniques such as one hot encoding technique, or conversion of date variables into further year,day and month variable was done. In each data frame after all the cleaning these new features were formed :

1. In portfolio data frame, id was renamed to offer id, channels feature was sub categorised into email,mobile,social and web features using Multi Label binarizer technique (encoding technique) from sklearn library [7], similarly all the string offer id was converted to numerical offer ID such as Offer ID 1 so it can be checked what offers were completed in EDA part.

2. In profile table again id was renamed to customer id, age missing values were treated with mean value imputation of missing variables technique [5], became member feature was converted to datetime feature and then further converted into year,day and month feature. Age was converted to groups 8 groups which is shown below in figure. This age group feature was further converted to 8 features using one hot encoding technique from pandas library. Lastly gender was converted to 3 features.



```
[81]: profile['gender'].value_counts().to_frame().style.bar()
```

[81]:

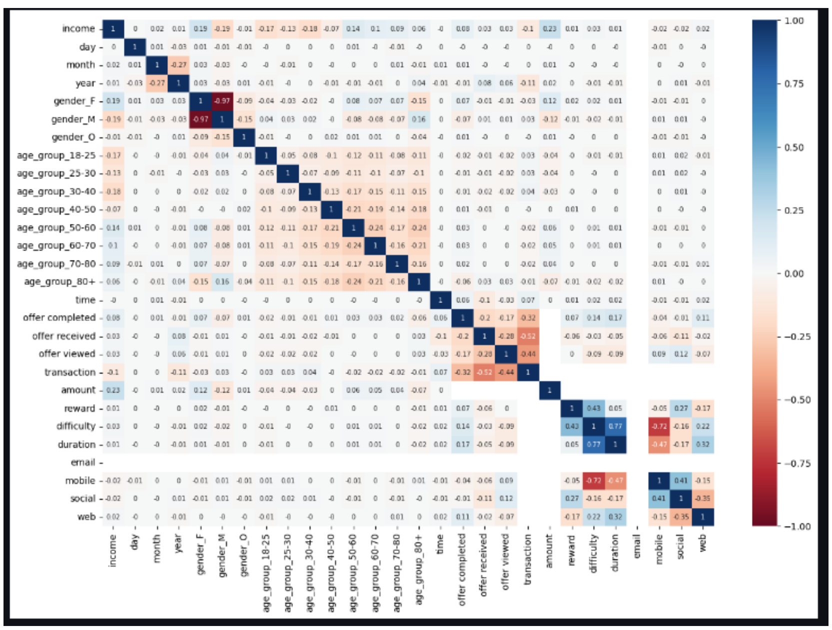
gender	count
M	10659
F	6129
O	212

3. Finally the third transcirpt table feature id was renamed to customer id, Value feature was converted to offer id and amount features using the extraction technique, also where no offer id was present Not Available was labelled and where amount was missing 0 value was used. Also the feature Event was converted to 4 categorical enocded features for Data analysis part only.

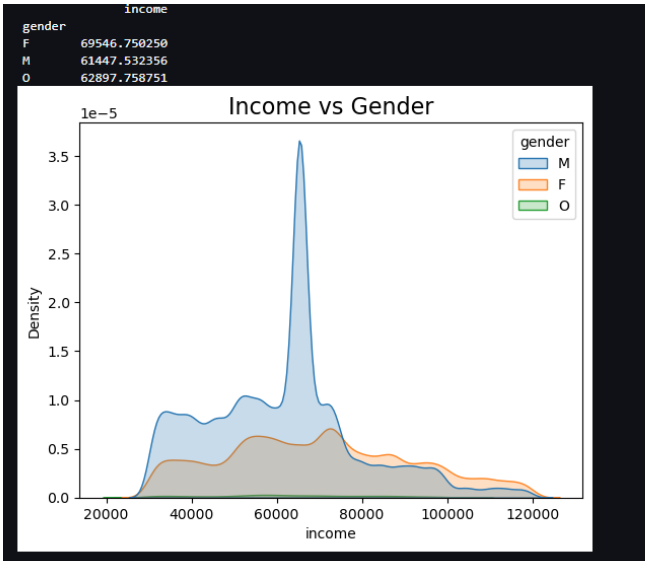
4. In end using the merging technique of pandas ,profile and transcript features were joined using the customer id key, and further joined with portfolio frame using offer id as key. Now this data frame was labelled as customer and was ready for Exploratory part.

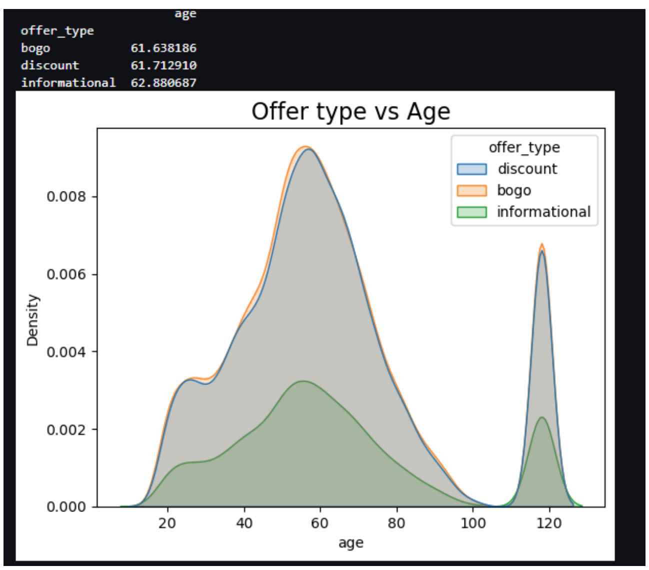
2.3 Exploratory Visualization

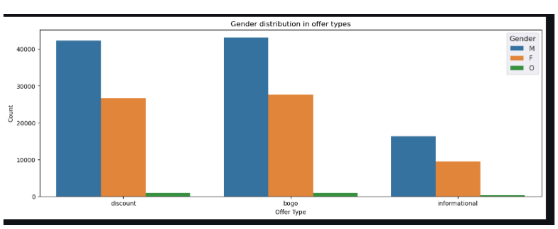
In exploratory analysis first feature correlation was checked so in case if some features are strongly correlated they can be checked and removed, also feature correlation helped me to note an important point here that reward, difficulty and reward were somehow strongly correlated so they will plan an important part for model learning. Visualization of correaltion plot using seaborn and pandas library is shown below. Second Exploration was done by comparing Income with

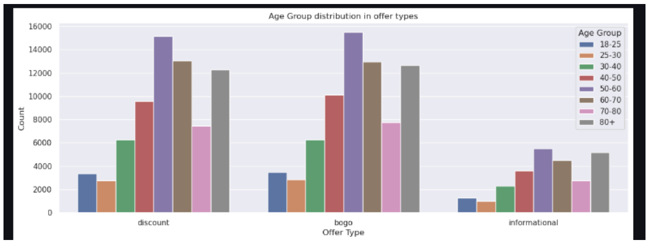


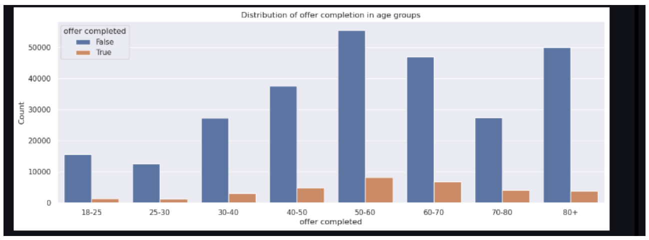
Gender by checking their mean values by grouping it according to gender which showed that on avearage in this population Female Gender population earned more than other two genders. A visual of income and Gender is shown. Third important exploration was done by checking what was average age for each offer sent which showed for BOGO and discount it was 62 and informational was 63. Fourth important exploration was done by checking what was offer distribution type according to gender which all three offers had men has highest feamale as second. Fifth exploration was done by checking which age group was highest for all three offers which showed 50-60 as highest and 60-70 as second highest Sixth for all age groups it was checked who completed most offers which showed 50-60 to complete most offers and 60-70 as second. Seventh exploration showed


income
gender
F 69546.750250
M 61447.532356
O 62897.758751
Income vs Gender
1e−5
gender
M
F
O
Density
income
3.5
3.0
2.5
2.0
1.5
1.0
0.5
0.0
20000
40000
60000
80000
100000
120000

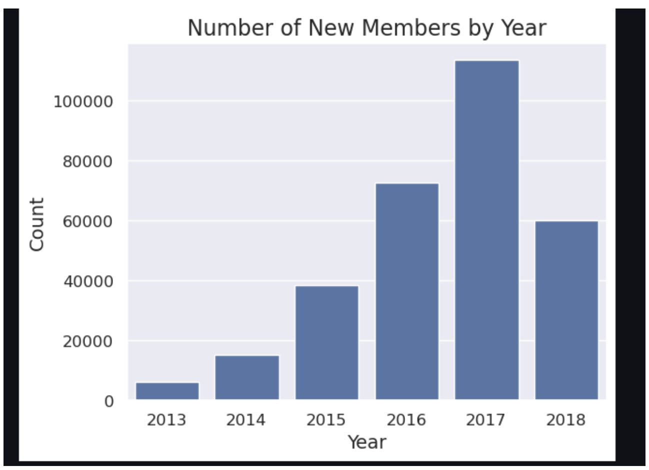

age
offer_type
bogo 61.638186
discount 61.712910
informational 62.880687
Offer type vs Age
offer_type
discount
bogo
informational
Density
age
0.008
0.006
0.004
0.002
0.000
20
40
60
80
100
120


Gender distribution in offer types
Gender
M
F
O
Count
Offer Type
discount
bogo
informational





what year we have the highest number of joiners which showed 2017 as highest, 2016 as second highest and 2018 as third highest.

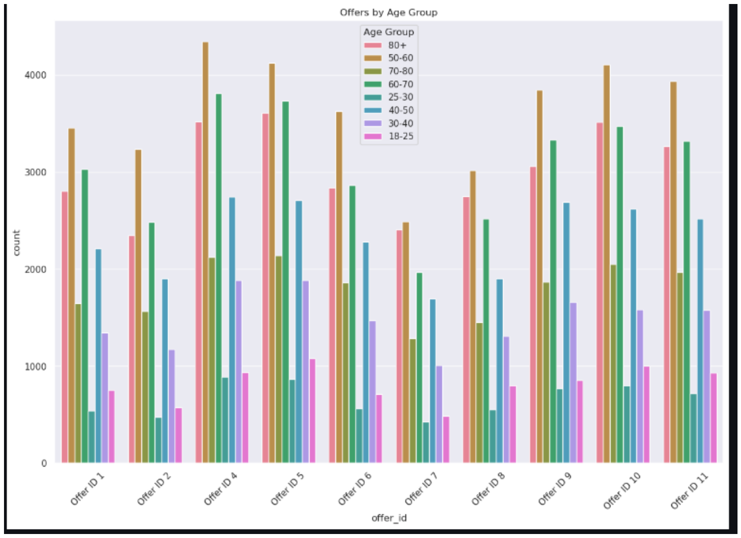


Eighth and final observation was done to check which offer did 50-60 age group completed the most which was offer id 4,5 and 10, this is shown in the visual as well.

2.4 Algorithms and Techniques

2.4.1 Decision trees

Decision tree classifiers are regarded to be a standout of the most well-known methods to data classification representation of classifiers. Different researchers from various fields and backgrounds have considered using it in machine study,



pattern recognition, and statistics. In various fields such as medical disease analysis, text classification, user smartphone classification, images, and many more the employment of Decision tree classifiers has been proposed in many ways. [4] It is also helpful as it can be termed as a benchmark model as well which is defined below

2.4.2 Benchmark Model

It is a fairly accurate model which can help assess implementation of the hypothesis, several different models will be implemented according to different hypothesis. I have choosen it as benchmark model because it has further extension to Random Forest Decision tree model, Gradient Boosting model, and ADA model. These all models are complex models of decision trees and can be for the up-scalling of overall accuracy of the model.

2.4.3 Random Forest

Random forests are a combination of tree predictors such that each tree depends on the values of a random vector sampled independently and with the same distribution for all trees in the forest. The generalization error for forests converges a.s. to a limit as the number of trees in the forest becomes large. The generalization error of a forest of tree classifiers depends on the strength of the individual trees in the forest and the correlation between them. Using a random selection of features to split each node yields error rates that compare favorably to Adaboost (Y. Freund and R. Schapire, Machine Learning: Proceedings of the Thirteenth International conference 148–156), but are more robust with respect to noise. Internal estimates monitor error, strength, and correlation and these are used to show the response to increasing the number of features used in the splitting. Internal estimates are also used to measure variable importance. [3]

2.4.4 XGboost

In XGBoost, individual trees are created using multiple cores and data is organized in order to minimize the lookup times. This decreases the training time of models which in turn increases the performance.[12]

2.4.5 ADA Boost

AdaBoost algorithm can be implemented to the multi-class case without reducing it to multiple two-class problems.= AdaBoost algorithm is equivalent to a forward stagewise additive modeling algorithm that minimizes a novel exponential loss for multi-class classification. [9]

2.4.6 Gradient Boost

Gradient boosting machines are a family of powerful machine-learning techniques that have shown considerable success in a wide range of practical applications. They are highly customizable to the particular needs of the application, like being learned with respect to different loss functions. [10]

3 Evaluation Metrics

I ll be using precision, recall, accuracy for the evaluation of my models. These metrics will help me to evaluate wether my model is overfitting or underfitting or how i need to re-assess or change the parameters or hyper parameters involved in my modelling approaches. All the metrics i have used are defined in the metrics section, F1 score is most important of them as it combines both precision and recall and helps us to fairly evaluate the model.

4 Methodology

These are the general steps i have taken for my project :

1. Set up of environment in which jupyter lab and s3 bucket along with the particular IAM policies will be set up
2. Implementing of pre-processing steps which will involve removing outliers, null or NA values, similarly changing and checking of the data types of the variables , then finally aggregating or creating new features in the data sets
3. Implementing Exploratory data analysis and forming hypothesis using the EDA visuals

4. Implementing several different modelling techniques to find the best model for the project
5. Analysis of the benchmark model and writing explanation of the model along with what evaluation metrics are helping to justify the picking of benchmarked model
6. Writing all the results in form of report

4.1 Data Preprocessing

All the data cleaning done in analysis is part of data pre-processing, whereas in addition to that some features were picked for creating training and testing data frames. List of modelling variables included the features shown below in the image. Event feature was converted into labelled feature where offer received

```
modelling_features = [ 'income', 'month',
        'year', 'gender_F', 'gender_M', 'gender_O', 'age_group_18-25',
        'age_group_25-30', 'age_group_30-40', 'age_group_40-50',
        'age_group_50-60', 'age_group_60-70', 'age_group_70-80',
        'age_group_80+', 'time','offer_type_id','email', 'mobile', 'social',
         'web','amount', 'reward', 'difficulty', 'duration']
X = customer[modelling_features]
```

was labelled as 0,offer viewed as 1,transaction as 2,offer completed as 3. Training and testing split was done using sklearn library where 80 percent of data was split as training and 20 percent as testing data. Shape of training and testing data frames is shown below:

```
from sklearn.model_selection import train_test_split
# Spliting the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, Y, test_size=0.2, random_state=0)

X_train.shape, X_test.shape

((245227, 24), (61307, 24))
```

4.2 Implementation

In Implementation part first Decision tree which is the benchamark model was implemented and feature importance was done second Random forest was implemented,third Xgboost was implemented Fourth ADA boost was implemented and finally gradient boost was implemented

4.3 Refinement

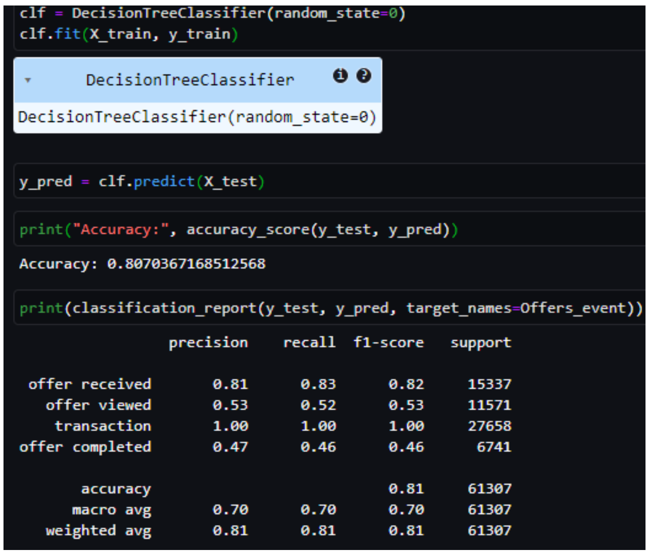
For refinement purpose grid search cv was implemented for ADA boost and Gradient Boost using sklearn library where number of estimators ranged from 200,500,1000,1500 and learning rate was set from 0.05,0.1,0.15,0.2. Best hyper parameters were selected for these two algorithms which helped us to improve the overall model performance.

5 Results

5.1 Model Evaluation and Validation

5.1.1 Decision Tree

Decision tree model which was benchmark model it has overall accuracy of 0.81. Overall Weighted precision of 0.81, macro average precision of 0.81,Recall micro average 0.70 and weighted average of 0.81. macro average f1 score of 0.70 and weighted average 0.81. Figure of results is shown in figure 1.



```
clf = DecisionTreeClassifier(random_state=0)
clf.fit(X_train, y_train)
```

```
DecisionTreeClassifier
DecisionTreeClassifier(random_state=0)
```

```
y_pred = clf.predict(X_test)
```

```
print("Accuracy:", accuracy_score(y_test, y_pred))
```

```
Accuracy: 0.8070367168512568
```

```
print(classification_report(y_test, y_pred, target_names=Offers_event))
```

```
                 precision    recall  f1-score   support

 offer received       0.81      0.83      0.82     15337
   offer viewed       0.53      0.52      0.53     11571
    transaction       1.00      1.00      1.00     27658
offer completed       0.47      0.46      0.46      6741

       accuracy                           0.81     61307
      macro avg       0.70      0.70      0.70     61307
   weighted avg       0.81      0.81      0.81     61307
```

Figure 1: Decision Tree Results

5.1.2 Random Forest

Random Forest has overall accuracy of 0.69. Overall Weighted precision of 0.53, macro average precision of 0.68,Recall micro average 0.53 and weighted average of 0.69. macro average f1 score of 0.53 and weighted average 0.68. Figure of results is shown in figure 2.

5.1.3 Xgboost

. XGboost had overall accuracy of 0.87. Overall Weighted precision of 0.80, macro average precision of 0.87,Recall micro average 0.79 and weighted average of 0.87. macro f1 score of 0.79 and weighted average 0.87. Figure of results is shown in figure 3.

```
clf_rf = RandomForestClassifier(random_state=0)

clf_rf.fit(X_train, y_train)
y_pred_rf = clf_rf.predict(X_test)
print(classification_report(y_test, y_pred_rf, target_names=Offers_event))
```

```
                 precision    recall  f1-score   support

 offer received       0.58      0.64      0.61     15337
   offer viewed       0.29      0.28      0.28     11571
    transaction       1.00      1.00      1.00     27658
offer completed       0.26      0.21      0.23      6741

       accuracy                           0.69     61307
      macro avg       0.53      0.53      0.53     61307
   weighted avg       0.68      0.69      0.68     61307
```

Figure 2: Random Forest Results

```
from xgboost import XGBClassifier
clf_xgb = XGBClassifier(random_state=0)

clf_xgb.fit(X_train, y_train)
y_pred_xgb = clf_xgb.predict(X_test)
print(classification_report(y_test, y_pred_xgb, target_names=Offers_event))
```

```
                 precision    recall  f1-score   support

 offer received       0.83      1.00      0.91     15337
   offer viewed       0.74      0.59      0.66     11571
    transaction       1.00      1.00      1.00     27658
offer completed       0.63      0.56      0.59      6741

       accuracy                           0.87     61307
      macro avg       0.80      0.79      0.79     61307
   weighted avg       0.87      0.87      0.87     61307
```

Figure 3: XGBoost Results

5.1.4 ADA Boost

Ada Boost was implemented with grid search cv to check for best paramters where learning rate of 0.2 and number of estimators as 1500 was shown in Gridsearch CV which were then used for modelling in ADA Boost. ADAboost had overall accuracy of 0.86. Overall Weighted precision of 0.76, macro average precision of 0.84,Recall micro average 0.73 and weighted average of 0.86. macro average f1 score of 0.72 and weighted average 0.84. Figure of results is shown in figure 4.

5.1.5 Gradient Boost

Gradient Boost was implemented with grid search cv to check for best paramters where learning rate of 0.2 and number of estimators as 1500 was shown in Gridsearch CV which were then used for modelling in Gradient Boost. It had overall accuracy of 0.87. Overall Weighted precision of 0.80, macro average precision of 0.87,Recall micro average 0.78 and weighted average of 0.87. macro average f1 score of 0.79 and weighted average 0.84. Figure of results is shown in figure 5.

```
{'learning_rate': 0.2, 'n_estimators': 1500}

ada = AdaBoostClassifier(learning_rate= 0.2, n_estimators= 1500)
ada.fit(X_train,y_train)

y_pred_ada = ada.predict(X_test)

print(classification_report(y_test, y_pred_ada, target_names=Offers_event))
                 precision    recall  f1-score   support

 offer received       0.83      1.00      0.91     15337
   offer viewed       0.63      0.69      0.66     11571
    transaction       1.00      1.00      1.00     27658
offer completed       0.57      0.21      0.31      6741

       accuracy                           0.86     61307
      macro avg       0.76      0.73      0.72     61307
   weighted avg       0.84      0.86      0.84     61307
```

Figure 4: ADABoost Results

```
gbc_clf_bst = GradientBoostingClassifier(learning_rate= 0.2, n_estimators= 1500)
gbc_clf_bst.fit(X_train,y_train)

GradientBoostingClassifier
GradientBoostingClassifier(learning_rate=0.2, n_estimators=1500)

y_pred_gbc = gbc_clf_bst.predict(X_test)

print(classification_report(y_test, y_pred_gbc, target_names=Offers_event))
                 precision    recall  f1-score   support

 offer received       0.83      1.00      0.91     15337
   offer viewed       0.74      0.58      0.65     11571
    transaction       1.00      1.00      1.00     27658
offer completed       0.61      0.56      0.58      6741

       accuracy                           0.87     61307
      macro avg       0.80      0.78      0.79     61307
   weighted avg       0.87      0.87      0.87     61307
```

Figure 5: Gradient Boost Results

Finally this Gradient boost was among the best performing model so it was pickled in S3 bucket which is shown in figure 6 and 7.

5.2 Justificatiion

Our benchmark model decision tree model results were quite helpful in setting up where the accuracy precision and recall will stand for our models, Random forest performed worse than decision tree due to complexity of algorthim whereas Xgboost performed better than the Random forest and Decision tree models. ADA and Gradient boost were implemented with GridSearch CV which showed best paramters so they performed much better than benchmark model, Random forest model. In final analysis i choosed Gradient as best performing model and pickled the model using pickle library. I then saved that pickled model in my

Figure 6: Saving Gradient boost to s3

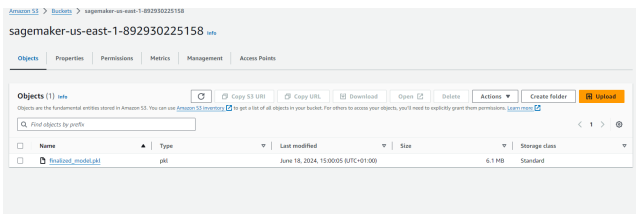


Figure 7: Pickled Model file in s3

s3 bucket. In refinement of my models if the hyper paramter tuning and cross validation can be done then we can know the best potential of XGboost, Decision and Random Forest as well, Similarly the saved model can be further used in an EC2 instance where it can be used to link inside the starbucks app which can help to predict wether a new customer will complete an offer or not. This project was challenging and fun for me as it involved alot of pre-processing of data, joining of the data and then implementing modelling techniques.This data which was used had alot of imbalancing in it as well if we considered only for class offer completed or not, but then in that process further implementation of Neural Networks would have been useful. I enjoyed this project as whole as it had a lot of challenges involved i hop i can take this project to future implementation of similar coffee shop business.

References

[1] Diego Antognini, Claudiu Musat, and Boi Faltings. “Multi-dimensional explanation of target variables from documents”. In: *Proceedings of the*

AAAI Conference on Artificial Intelligence. Vol. 35. 14. 2021, pp. 12507–12515.

[2] B Senthil Arasu, B Jonath Backia Seelan, and N Thamaraiselvan. "A machine learning-based approach to enhancing social media marketing". In: *Computers & Electrical Engineering* 86 (2020), p. 106723.

[3] Leo Breiman. "Random forests". In: *Machine learning* 45 (2001), pp. 5–32.

[4] Bahzad Charbuty and Adnan Abdulazeez. "Classification based on decision tree algorithm for machine learning". In: *Journal of Applied Science and Technology Trends* 2.01 (2021), pp. 20–28.

[5] A Rogier T Donders et al. "A gentle introduction to imputation of missing values". In: *Journal of clinical epidemiology* 59.10 (2006), pp. 1087–1091.

[6] Johannes Fürnkranz and Peter A Flach. "An analysis of rule evaluation metrics". In: *Proceedings of the 20th international conference on machine learning (ICML-03)*. 2003, pp. 202–209.

[7] Raul Garreta and Guillermo Moncecchi. *Learning scikit-learn: machine learning in python*. Vol. 2013. Packt Publishing Birmingham, 2013.

[8] Adam Hakim et al. "Machines learn neuromarketing: Improving preference prediction from self-reports using multiple EEG measures and machine learning". In: *International Journal of Research in Marketing* 38.3 (2021), pp. 770–791.

[9] Trevor Hastie et al. "Multi-class adaboost". In: *Statistics and its Interface* 2.3 (2009), pp. 349–360.

[10] Alexey Natekin and Alois Knoll. "Gradient boosting machines, a tutorial". In: *Frontiers in neurorobotics* 7 (2013), p. 21.

[11] Eric WT Ngai and Yuanyuan Wu. "Machine learning in marketing: A literature review, conceptual framework, and research agenda". In: *Journal of Business Research* 145 (2022), pp. 35–48.

[12] Santhanam Ramraj et al. "Experimenting XGBoost algorithm for prediction and classification of different datasets". In: *International Journal of Control Theory and Applications* 9.40 (2016), pp. 651–662.

[13] Hamid Shirdastian, Michel Laroche, and Marie-Odile Richard. "Using big data analytics to study brand authenticity sentiments: The case of Starbucks on Twitter". In: *International Journal of Information Management* 48 (2019), pp. 291–307.