**Starbucks Capstone Project Report**

**Sukruta Pardeshi | 21 March 2023**

**Udacity – AWS Machine Learning Engineer Nanodegree**

**Problem Definition:**

* **Project Overview**
* **Domain Background:**

Starbucks is an enthusiastic retailer of coffee and other beverages with its corporate headquarters in Seattle, Washington. Registered users of their mobile application can order coffee for pickup while on the go, pay in-store using the app, and get rewards points. Also, this app provides these users with promos for extra points that could simply be a drink marketing or it could be a real deal, like a discount or a BOGO (buy one, get one free) deal. The goal of this project is to identify the customers who are most likely to respond to an offer by personalizing promotional offers for them based on their answers to prior offers.

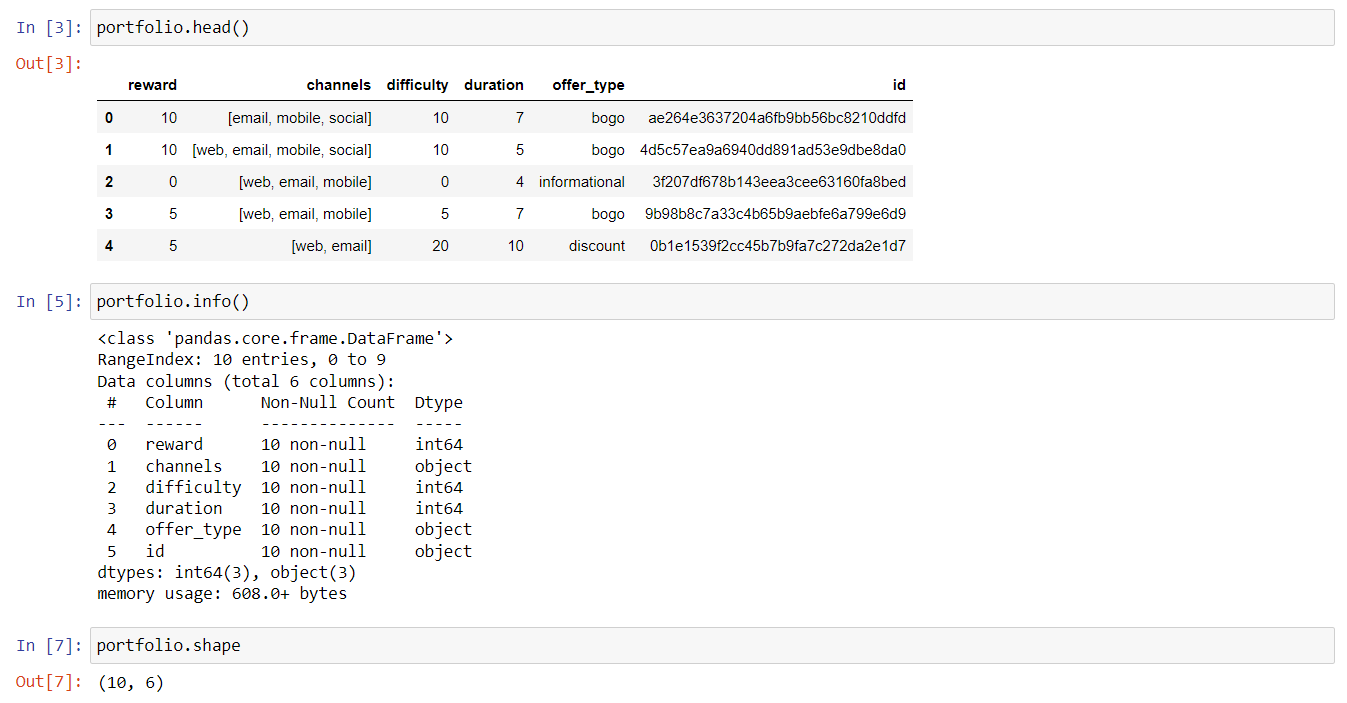
* **Datasets and Inputs:**

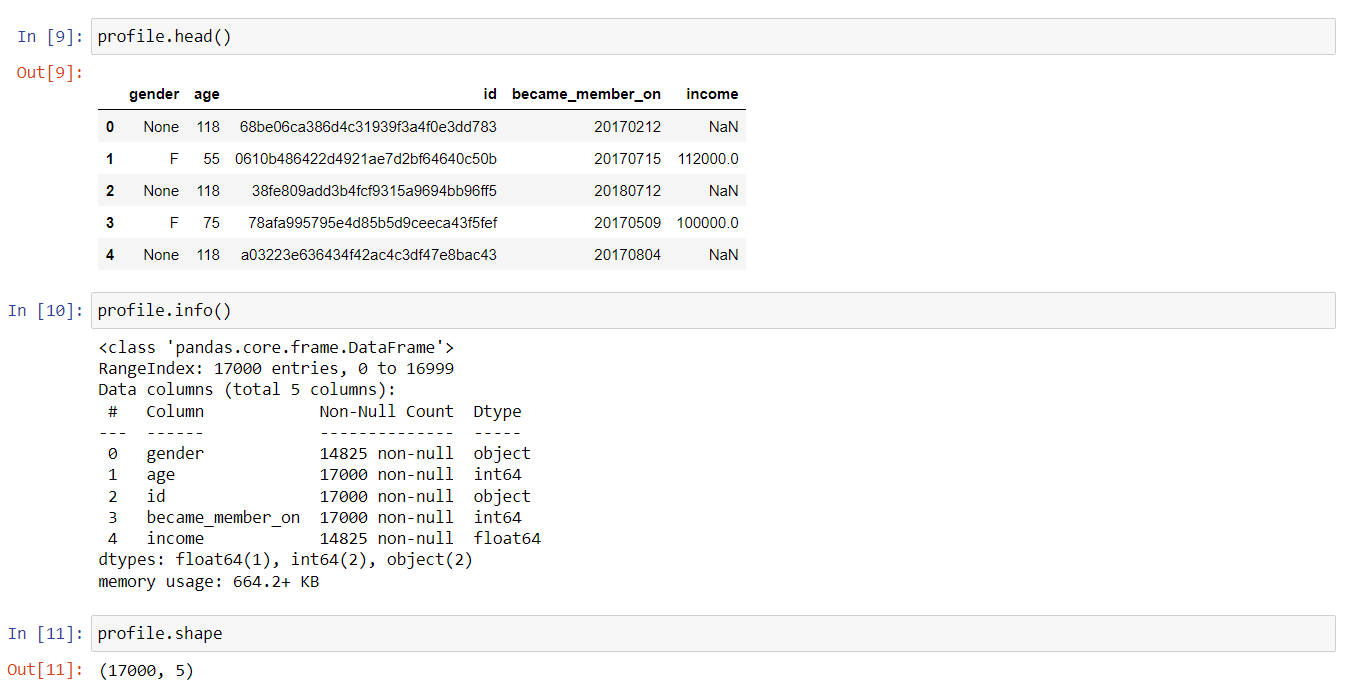
The simulated data in this data set closely resembles consumer activity on the Starbucks Rewards mobile app. Starbucks delivers offers to customers who use its mobile app every few days. The data set is provided in form of three JSON files:

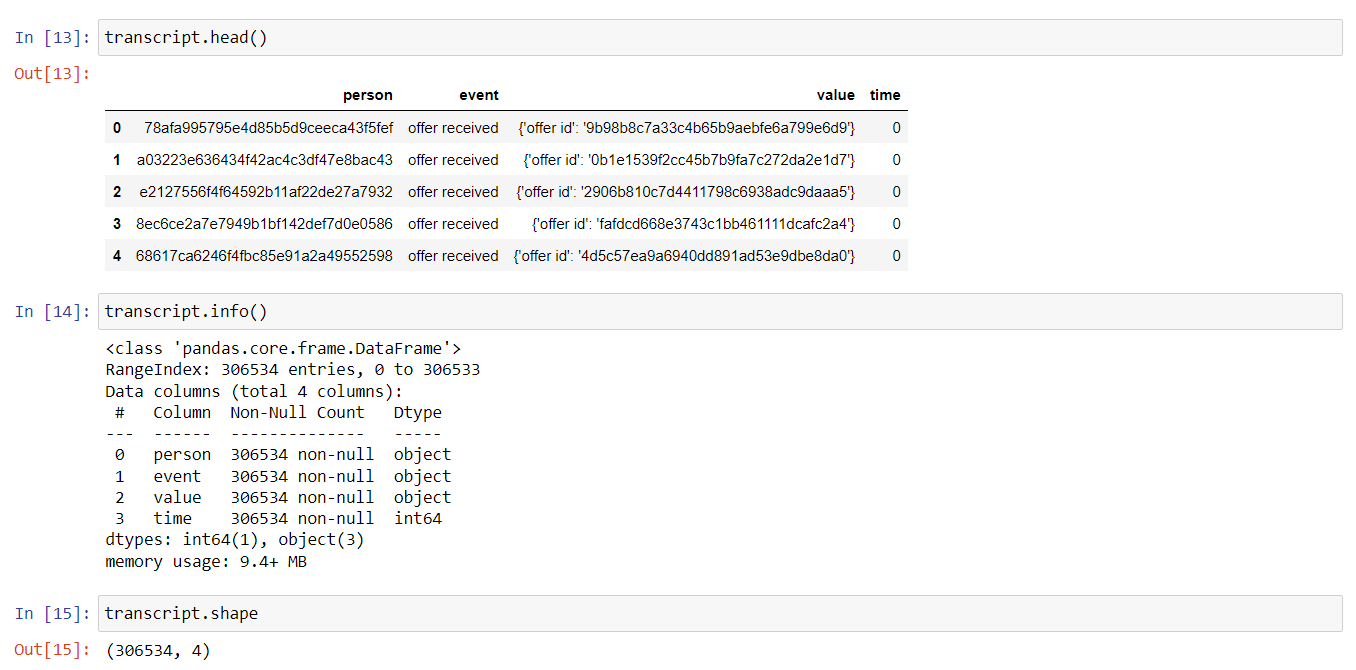
- portfolio.json - containing offer ids and metadata offer (duration, type, etc.)

- profile.json - demographic data for each customer.

- transcript.json - records for transactions, offers received, offers viewed, and offers completed.







* **Problem Statement**

My objective is to evaluate which kind of offer to deliver to each user based on their response to the offers that have already been made to them. The objective is to utilize the data set provided by Starbucks, which was collected over 30 days, to address the fact that not all users receive the same offer. Also, I'll create a machine-learning model that predicts how a customer will react to an offer.

* **Evaluation Metrics**

To evaluate the effectiveness of the strategy and identify which model produces the best outcomes, I will use the F1 score as the model metric. It can be understood as the weighted average of recall and precision. The traditional or balanced F-score (F1 score), which has a greatest value of 1 and a worst value of 0, is the harmonic mean of precision and recall.

**Problem Analysis:**

* **Data Exploration**

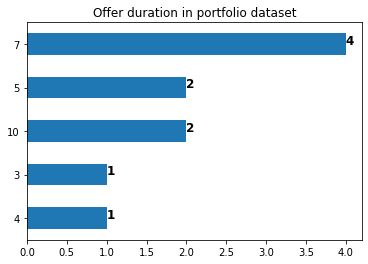
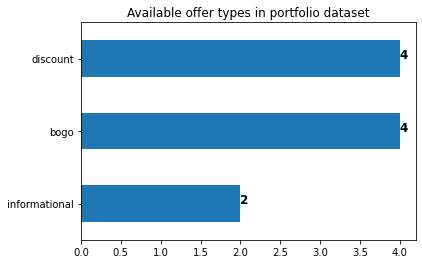
Here three dataset parts were explored individually to gain the major information. These parts were – portfolio.json, profile.json, and transcript.json.

Let’s see what were the important insights for me to explore from the data and preferably build the ML models to solve the problem.

**Portfolio.json:**

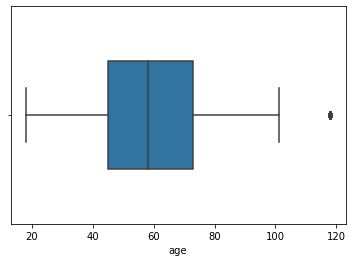
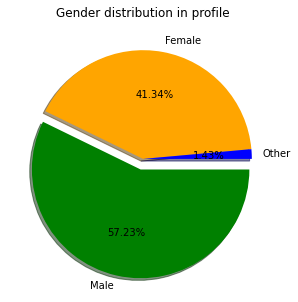
As it’s just the exploration part, there was no cleaning done here. Just the data was explored to find out the offer types and duration. This dataset contains 10 rows and 6 columns, and there were 3 integer attributes and 3 object attributes.

You can see the types of offers present in the dataset through the below bar graph representation.



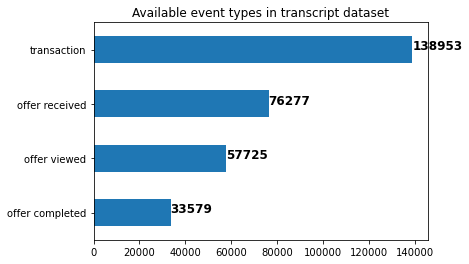
**Profile.json:**

In this dataset, there were 17000 rows and 5 columns having datatypes as float64(1), int64(2), and object(2). According to the profile data frame and checking null values, it was explored that 2175 values of gender & income both were missing where age is 118. Therefore, all the missing age values were encoded as 118. Also, an outlier was explored where people with an age greater than 80 don't use the app much or they may not drink many beverages. Gender distribution in the profile dataset was also explored.

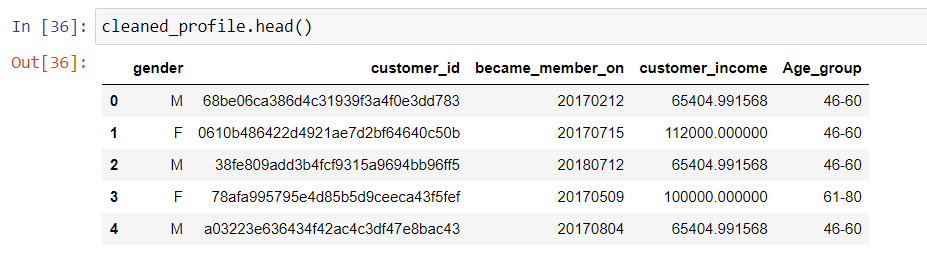
**Transcript.json:**

In this dataset, there were 306534 rows and 4 columns having datatypes as int64(1), and object(3). There was not much enough to explore in this part. Only the types of events that were present were explored which were measured to train and test the ML models.



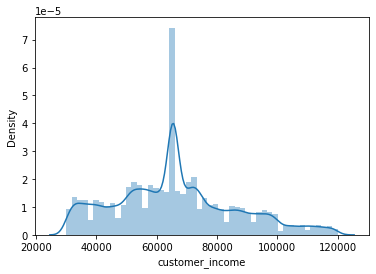
* **Data Cleaning**

Here all the unnecessary data, renaming of columns for ease of training, or encoding of the categorical values was implemented. In the portfolio data frame, only the column names were renamed as there was nothing much to clean. In the profile dataset, the following tasks were implemented to clean the data – renaming some column names, imputing missing age & income values with mean and missing gender values with mode, removing outliers from the dataset i.e. people with an age above 80, and classifying ages for EDA into categories: Under 20, 20-45, 46-60, 61-80. In the transcript dataset, the following tasks were implemented– renaming some column names, and expanding the keys of the ‘value’ column into new columns. Finally, all these three cleaned dataset parts were combined into one whole dataset for performing the EDA and training and testing the ML models. Here’s a glimpse of the cleaned profile data –

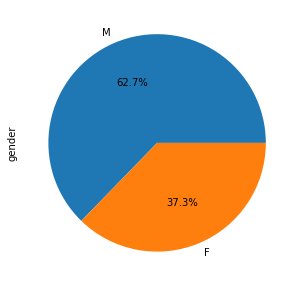


* **Performing EDA**

The average customer income that used Starbucks App lies between the range of 30000-120000. Here, the most used offer by the customers was explored which were the BOGO and discount offers that had nearly the same distributions. Customers within the age group of 46-60 used the Starbucks application frequently while those between the age group of 61-80 ranked second in using the App frequently.

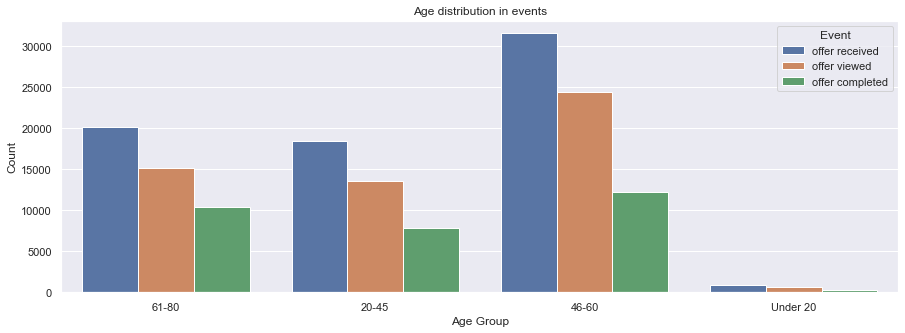


Actions to what these groups of customers do with the received offer were explored. Most of the customers don't pay attention to the offer and don't even have a look at it. Also, more several customers just view & ignore the offer than the ones who complete the offer. For better insights, the gender analysis was done to find out which group of customers most likely received the offers. It was observed that 63% of the male customers were engaging in these offers more compared to 37% of the female customers.

There were more male customers than female customers, and in general, people bought the discount offer then followed by the BOGO offer, and then the informational. Also, it was revealed that the age group of 41-60 was receiving more offers compared to the other groups.





**Project Implementation:**

* **Building and Evaluating the ML models and the benchmark model**

Before training the data some more cleaning on the data was performed to ease the model fitting process. The following tasks were implemented –

1. Encoding categorical data such as gender, offer type, channel, and age groups.

2. Encoding the 'event' data to numerical values: offer received -> 1, offer viewed -> 2, offer completed -> 3.

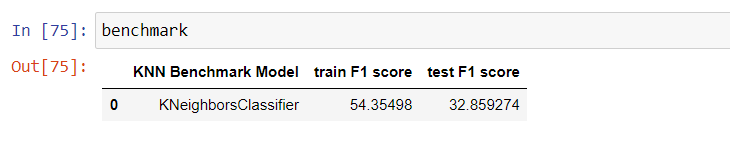
3. Encoding offer\_id and customer\_id because it was irrelevant in the training process.

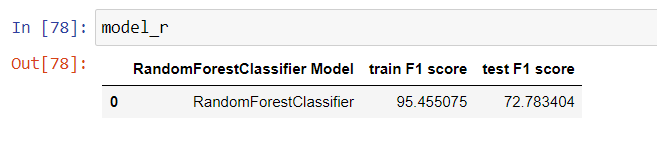
4. Dropping the column 'became\_member\_on' and adding separate columns for month and year.

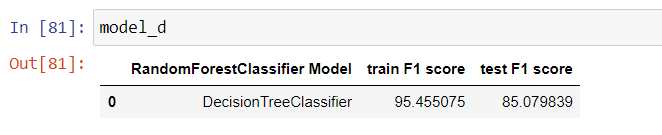
5. Scaling and normalizing the numerical data.

Final data is ready after tasks 1-5. Now the data was split (both features and their labels) into training and test sets, taking 60% of the data for training and 40% for testing. It contained – Training set: of 94501 rows, and a Testing set: of 63002 rows. Here, the F1 score was considered as the model metric to assess the quality of the approach and determine which model gives the best results. It can be interpreted as the weighted average of precision and recall. The traditional or balanced F-score (F1 score) is the harmonic mean of precision and recall, where an F1 score reaches its best value at 100 and worst at 0.

The three ML models – KNeighborsClassifier (also benchmark model), RandomForestClassifier, and DecisionTreeClassifier were trained and tested on the training and testing data respectively. No special parameters were provided to the models except for the ‘random\_state’ parameter and the benchmark model: ‘n\_neighbors=5’ parameter. A function was built to evaluate the train and test F1 scores of the models. After evaluating the models on training and testing data the following results were obtained –

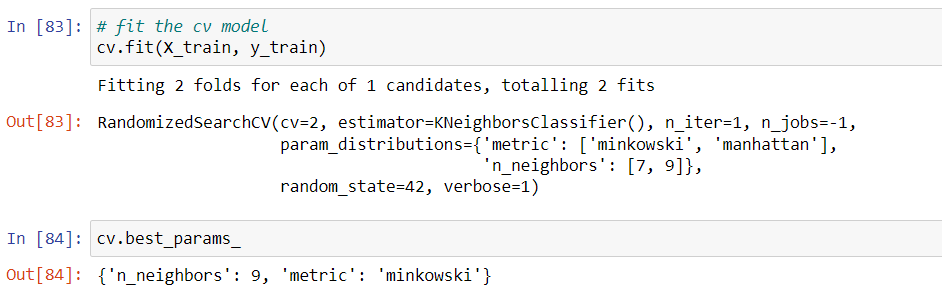




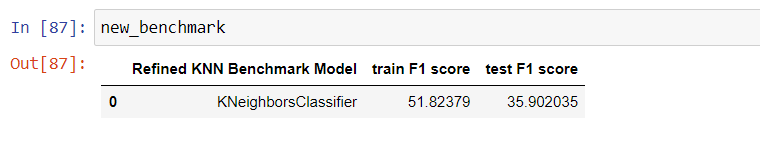


* **Model Refinement and Validation**

As we can see the models RandomForestClassifier, and DecisionTreeClassifier gave a sufficiently better result compared to the benchmark model therefore I performed hyperparameter tuning on the benchmark model – KNeighborsClassifier to refine the model. To validate the model and tune the hyperparameters, I used the RandomizedSearchCV validation method. The reason behind this was the GridSearchCV was consuming more computational power and time and wasn’t providing the results. I refined the benchmark model on the parameter – ‘n\_neighbors’ list and ‘metric’ list. I also used fewer parameters to avoid the greater usage of CPU/GPU of my laptop as it was not providing results. Also, I lessened the number of iterations and the folds in the RandomizedSearchCV validation process for the same reason mentioned previously.



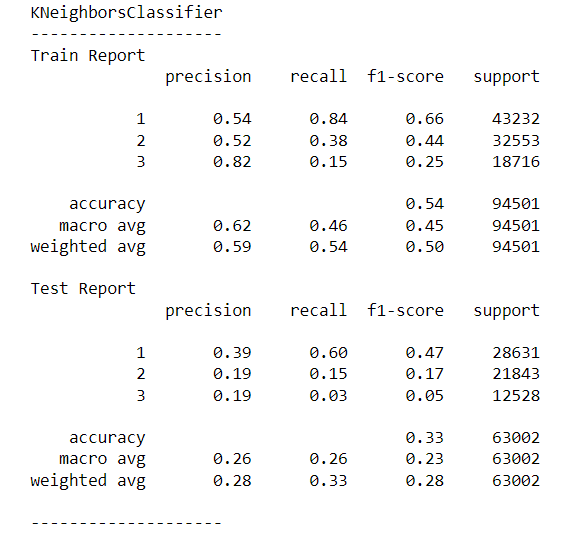
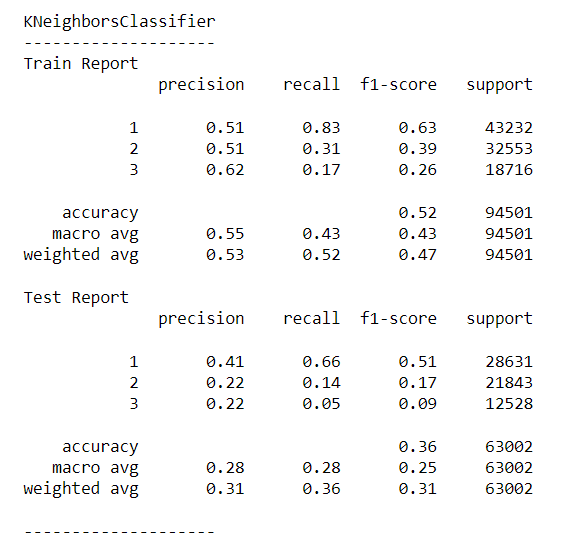
After validation and tuning, the best parameters were obtained which were n\_neighbors=9 and metrics= ‘minkowski’. Later, I again trained the KNeighborsClassifier benchmark model on these parameters to improve the model F1 score. There wasn’t much difference obtained, the training score was fair and similar to the previous model but the testing score improved from 33% to 36%. The refined benchmark model was evaluated on the f1 score metrics giving the results as –

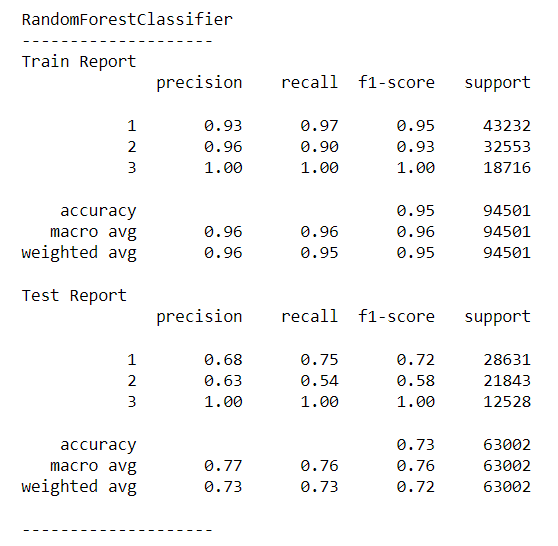
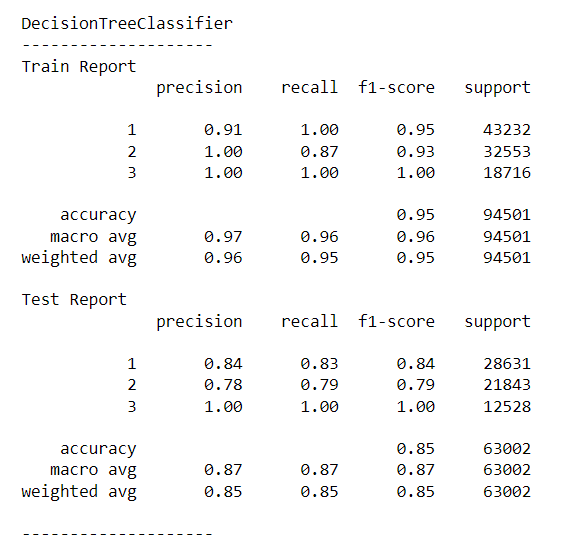


**Project Results:**

* **Model Evaluation Report**

The ML models as well as the refined model were evaluated on the F1 score metrics. The training and testing F1 scores of these models are already mentioned in the Model Refinement and Validation section of the report. Let us see a detailed classification report and the accuracy distribution of all these models.

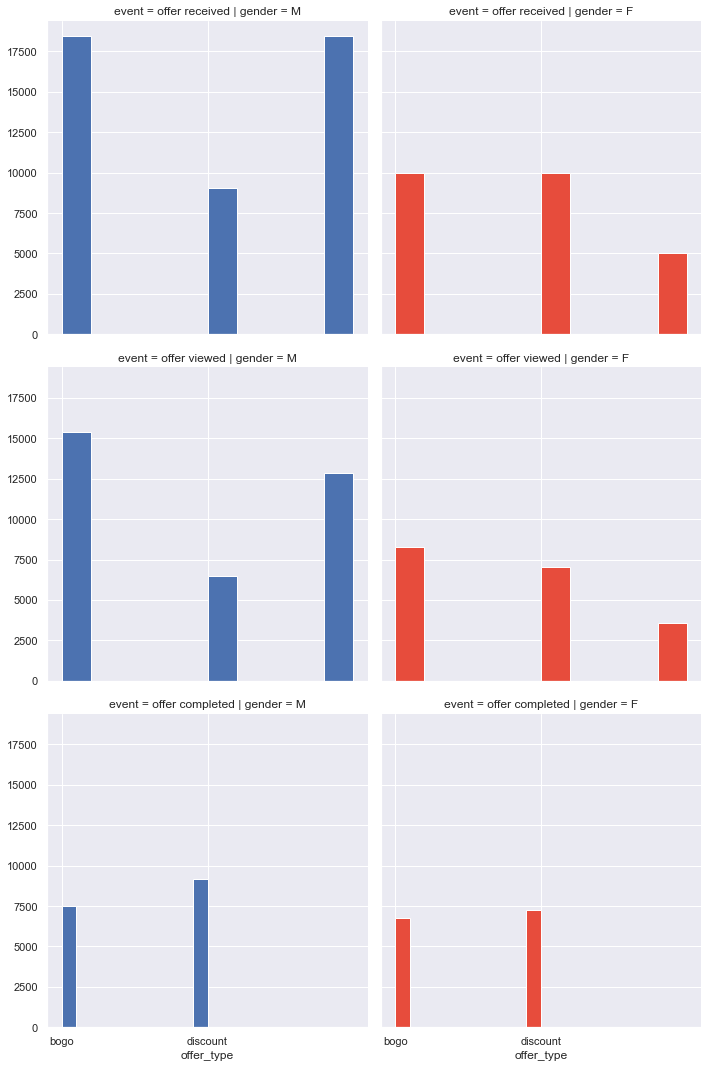
 

**Conclusions:**

* **Conclusion and Justification**

The males represent 62.7% of the data and use the Starbucks app more than the females. Specifically, both males & females in the age group 46-60 use the App the most. Discount offers are preferred by the customers. Also, there is less number of customers who complete the offer as compared to the ones who just view & ignore it. We can look more at the figures & information in the Exploratory Data Analysis section to best determine which kind of offers to send to the customers.



The validation set (test data set) is used to evaluate the model. Both models are better than the benchmark. The best score is created by the DecisionTreeClassifier model, as its validation F1 score is 85.07, which is much higher than the benchmark. The RandomForestClassifier model scores well as well compared to the benchmark, with a test F1 score of 72.78. The Refined KNeighborsClassifier after hyperparameter tuning didn't give much improvement but its validation was 3% higher than the benchmark model. Our problem to solve is not that sensitive which requires a very high F1 score, so the scores are good & sufficient and can be used for the classification purpose to predict whether a customer will respond to an offer. The comparison of all the models is given below –

