Contents

[**Executive summary** 2](#_Toc62158127)

[**Introduction** 2](#_Toc62158128)

[**Project Management** 2](#_Toc62158129)

[**Project goals** 2](#_Toc62158130)

[**Deliverables** 2](#_Toc62158131)

[**Program stack** 2](#_Toc62158132)

[**Timeline** 3](#_Toc62158133)

[**Member contribution** 3](#_Toc62158134)

[**Methodology** 4](#_Toc62158135)

[**Data understanding** 4](#_Toc62158136)

[**Data Cleaning** 6](#_Toc62158137)

[**Exploratory Data Analysis** 9](#_Toc62158138)

[Spending trend(age) by offered type of coupon & channel. 9](#_Toc62158139)

[Participant rate of event 9](#_Toc62158140)

[Finding out which year and month have joined to become a member. 10](#_Toc62158141)

[**Feature Engineering** 11](#_Toc62158142)

[**Data Modelling** 12](#_Toc62158143)

[**Result** 16](#_Toc62158144)

[**API** 17](#_Toc62158145)

[**Visualization Dashboard** 17](#_Toc62158146)

[**Discussion** 18](#_Toc62158147)

[**Conclusion** 18](#_Toc62158148)

[**References** 19](#_Toc62158149)

# **Executive summary**

The purpose of this project is to explore the relationship between the user profile and offer types in order to build a model which can suggest which offer is effective for a customer. The process of data science is conducted by the team from data preprocessing, data analysis, feature extraction and modelling to have a profound understanding about the data set and develop an efficient model that can evaluate the success of an offer. Overall, the result illustrated that the team has successfully developed a high-performance model which is LightBGMClassifier with a dashboard and APIs are deployed. However, more work can be done to improve the project such as developing models to predict the best offer type for customers or performing clustering to represent the customer demographic.

# **Introduction**

Starbucks is being well known as the largest coffeehouse company in the world at the moment. However, it is also being famous as a world-leading data-driven company that utilizes the use of data to elevate their business. One of the most famous business data-driven business approaches of Starbucks is personalization promotion. Starbuck uses the data gathered from the Starbucks reward mobile app and performs analysis to send the most suitable offer to a customer. In the final assignment of the course Practical Data Science, a group of four students will try to simulate the analysis process to build a model that can predict which type of offer is effective for the customer.

# **Project Management**

## **Project goals**

As stated above, this notebook will try to answer the question of whatis **the most effective offer that customer is most likely to use?** The process of data science is carried out in this notebook to have a deeper understanding of customer demographic and their behavior attributes from the given data set. Then, we will try to predict the most suitable type of promotion for each individual.

## **Deliverables**

At the end of this project several models are trained and evaluated, the model with the highest efficiency will be selected. The chosen model will be exported and deployed using Flask.  In addition, a dashboard for visualization which have interactive graphs and model evaluation is deployed using dash framework.

## **Program stack**

For the purpose of “extract, transform, and load” (ETL) data, we use **iPython** and **Jupyter Notebook**. Jupyter is a free, open-source interactive web-based computational notebook. Computational notebooks have been around for several years; however, Jupyter has exploded in popularity over the past couple of years. This capable tool supports multi-language programming and therefore became the *de facto* choice for data scientists for practicing and sharing various codes, quick prototyping, and exploratory analysis.

**Dash** is an open-source Python framework for building web applications. It is built on top of Flask, Plotly.js, React and React Js. With Dash, we can easily perform data analysis, data exploration, visualization, modelling, instrument control, and reporting. It enables you to build dashboards using pure Python. Dash is open source, and its apps run on the web browser. Dash app code is declarative and reactive, which makes it easy to build complex apps that may contain many interactive factors.

**Scikit-learn** is the most useful and free-of-charge library for machine learning in Python. It features various algorithms like support vector machines, random forests, and k-neighbors, and it also supports Python numerical and scientific libraries like NumPy and SciPy.

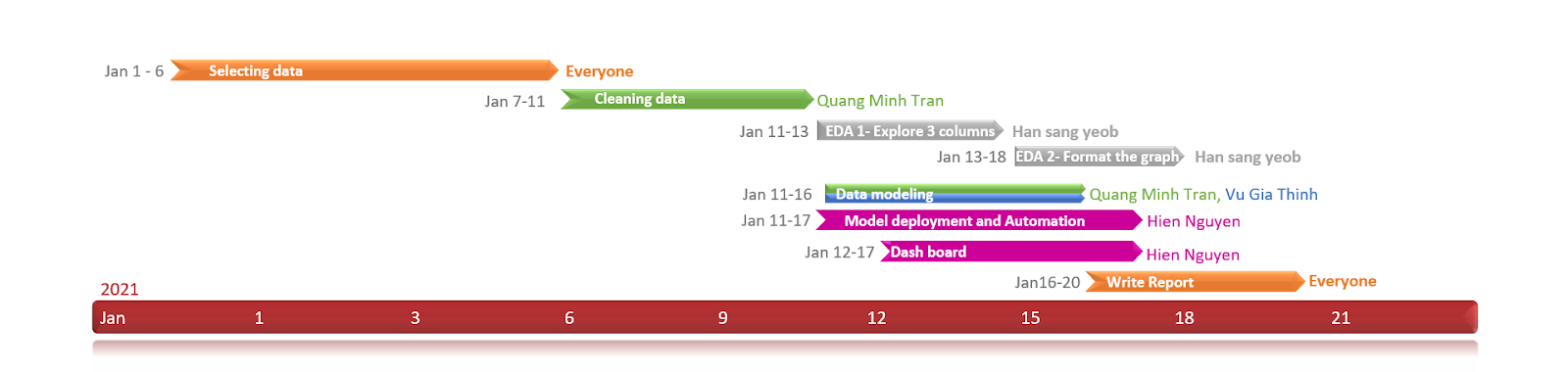
**LightGBM** is a gradient boosting framework that uses tree-based learning algorithms. It is designed to be distributed and efficient with the following advantages:

* Faster training speed and higher efficiency.
* Lower memory usage.
* Better accuracy.
* Support of parallel and GPU learning.
* Capable of handling large-scale data.

**Flask** is a lightweight Web Server Gateway Interface WSGI web application framework that was created to make getting started easy and making it easy for new beginners. With the tendency to scale up to complex applications. Flask gives the developer varieties of choice when developing web applications, it provides you with tools, libraries, and mechanics that allow you to build a web application, but it will not enforce any dependencies or tell you how the project should look like. The web application can be a blog, commercial website, or some web pages, it still allows the developers the opportunity to use some extensions provided by the community that allows you to add more functionality to the web application.

## **Timeline**

This project timeline in the last 4 weeks of the semester. Below is the Gantt chart to overview the whole project timeline in 21 days of Jan.



## **Member contribution**

|  |  |  |
| --- | --- | --- |
| **No** | **Name** | **Contribution** |
| 1 | Tran Minh Quang (s3757281) | Data cleaning, Data modelling, Report, Slide |
| 2 | Han sang yeob  (s3821179) | Data cleaning, EDA, Report, Slide |
| 3 | Hien Nguyen Sy  (s3536650) | Model deployment and automation, Visualization dashboard, Report |
| 4 | Vu Gia Thinh  (s3820373) | Data modelling, Report |

# **Methodology**

## **Data understanding**

In the beginning of the understanding stage, data scientists must understand the row dataset. For this project, we have used data in three json file,

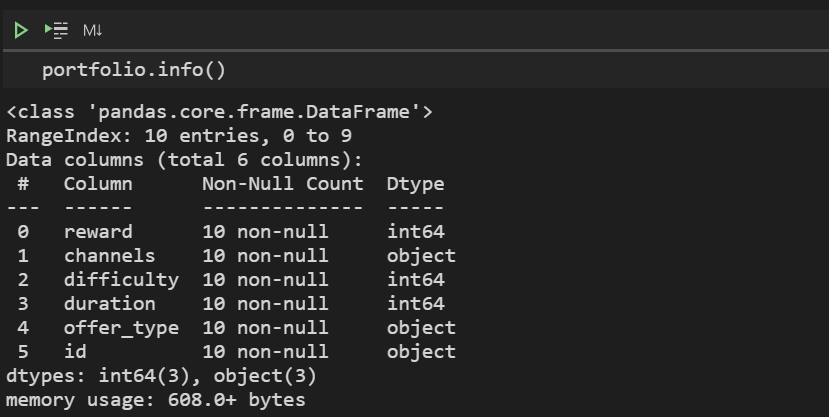
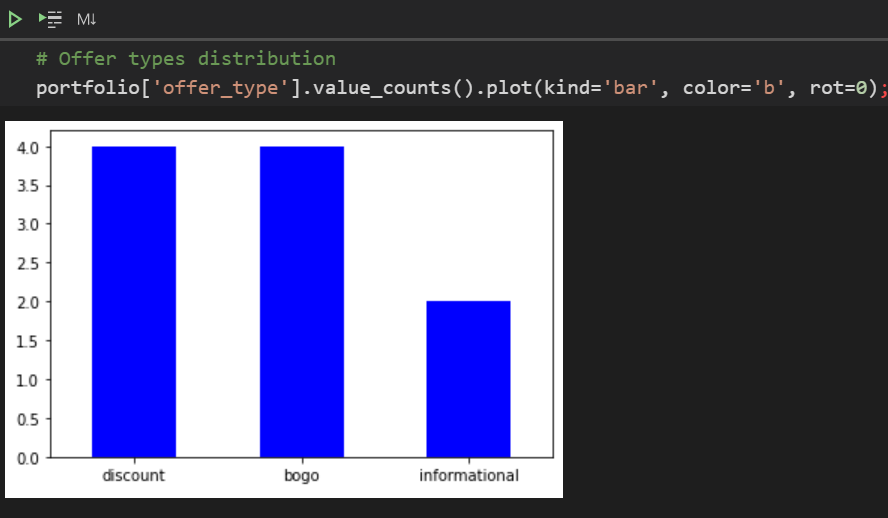
* portfolio.json: offer id and its relevant data
* profile.json: customer demographic data
* transcript.json: record for transactions, offers received, offers viewed, and offers completed

 To fully understand the data, we will scheme through all the columns in each data set.

* portfolio.json
  + id: offer id
  + channels: means of customer receiving offer.
  + difficulty: dollar amount needs to spend in order to complete offer.
  + duration: time length that offer valid.
  + offer\_type: types of offer.
  + reward: dollar amount needs to complete offer.
* profile.json
  + age: age of customer
  + gender: gender of customer
  + id: customer id
  + income: customer's annual income
  + became\_member\_on: day became member.
* transcript.json
  + person: customer id
  + time: time of transaction in hour
  + value: dictionary type, can be offer id or transaction.
  + event: transaction type

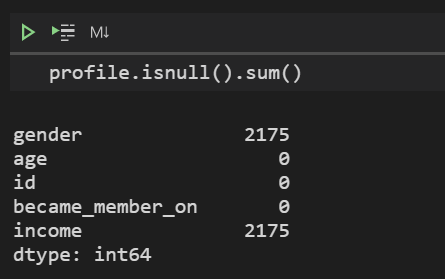
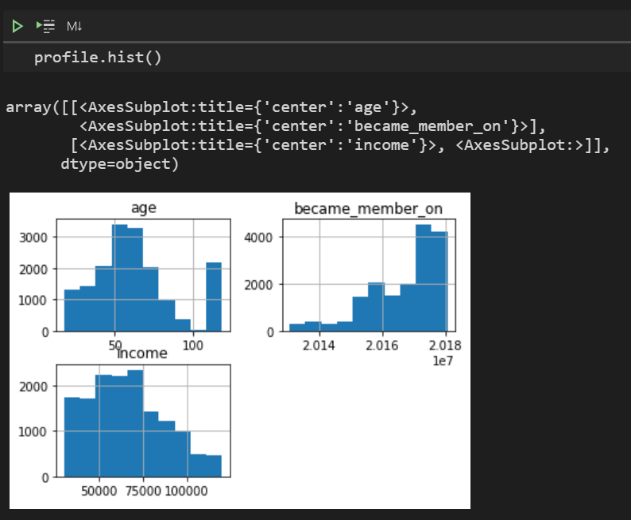
After understanding all the meaning of the columns, we need to describe the data sets. Reading the json files to the jupyter notebook, we can use pd.read\_json.

To describe the each data set, for portfolio.json,

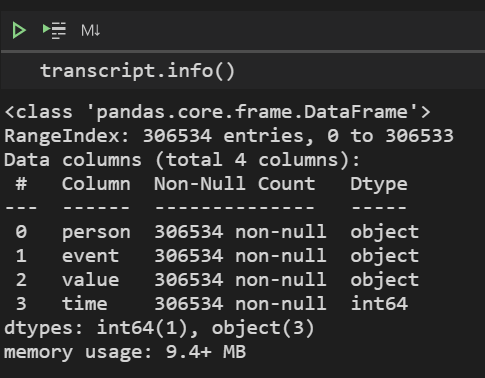
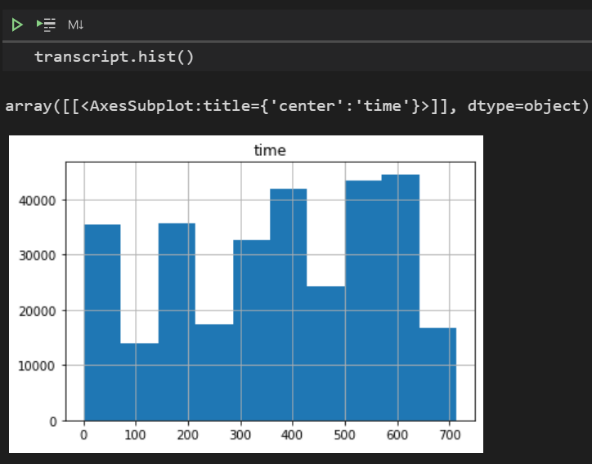
As we can see that the data in the portfolio does not have any null value, the categorical features of the data are correct and there do not exist any type error. The only job we need to fix in order to make the data clean is to One-hot encode the categorical feature of the data (channels, offer\_type)

For profile.json,

There exist null values in gender and income features, therefore we must handle the missing value in the data cleaning process. Also, the max age number is 118 which does not make sense, thus, all the customers who are above 100 should be changed to Nan value. We also need to transform the date time format of became\_member\_on feature and encoded the gender columns.

For transcript.json,

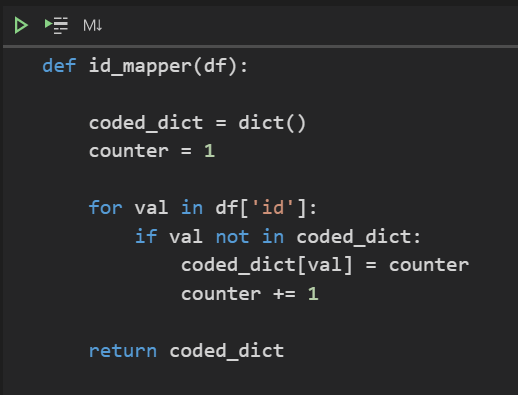
         

Regarding the data from transcript.json, we have to split the value col to offer\_id and amount. Also, splitting event col based on offer and transaction. Finally, we have to drop the customer id value that doesn't exist in the profile.json.

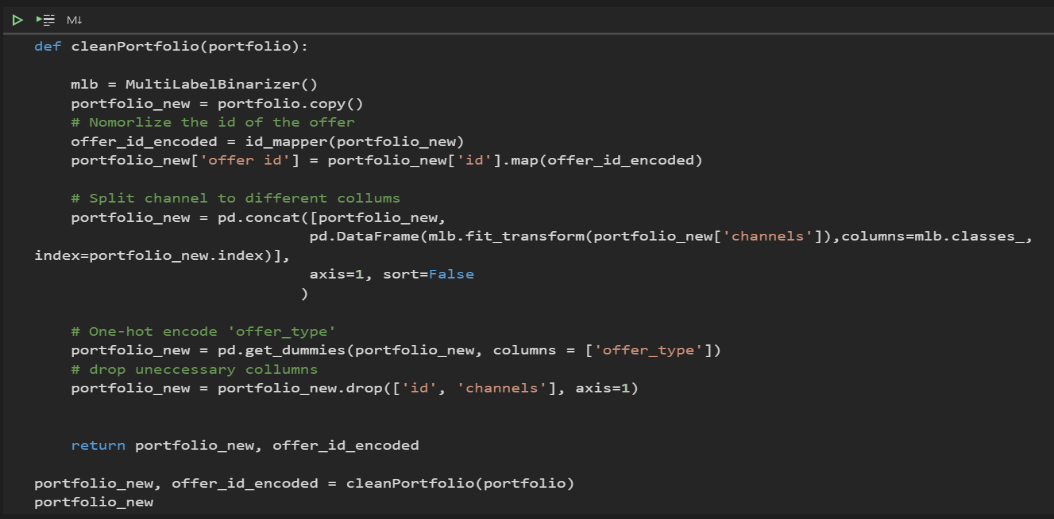
## **Data Cleaning**

After understanding the three data sets, we need to start cleaning the data sets.

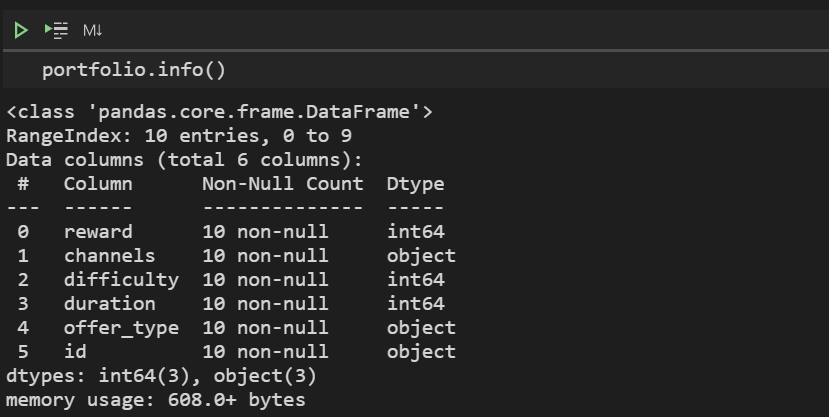
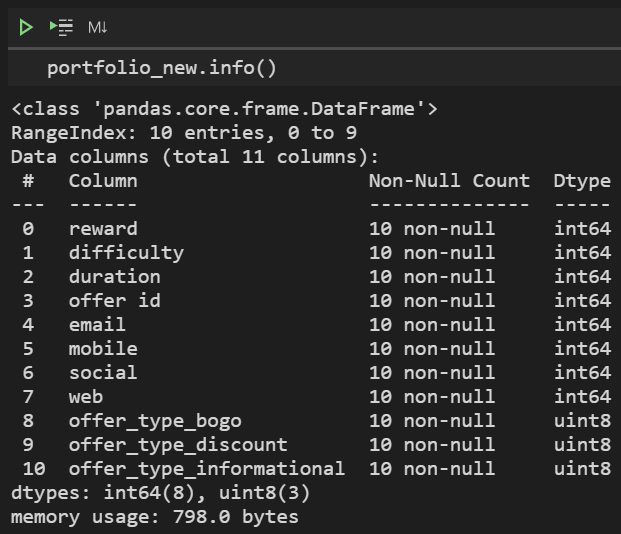
For portfolio.json,



Mapper function is to map unique integer ids to customers. Furthermore, the data frame will hold id variables in code\_dict which is a dictionary with original and encoded id values.



This is a code for cleaning the portfolio.json. Firstly, we will use id\_mapper to normalize id. of the offer. Then we will split the channel into different columns. To compare the information of the portfolio.json into portfolio\_new.json

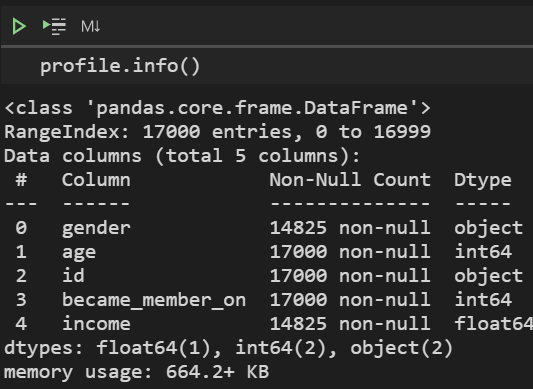
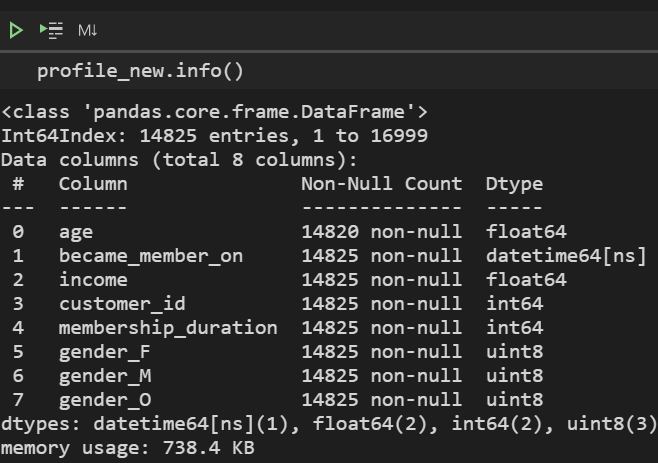
*Before cleaning   After cleaning*

As we can see, the data is represented as we expected, offer id is normalized to integer. The categorical columns channel and offer\_type is one hot encoded.

For profile.json,



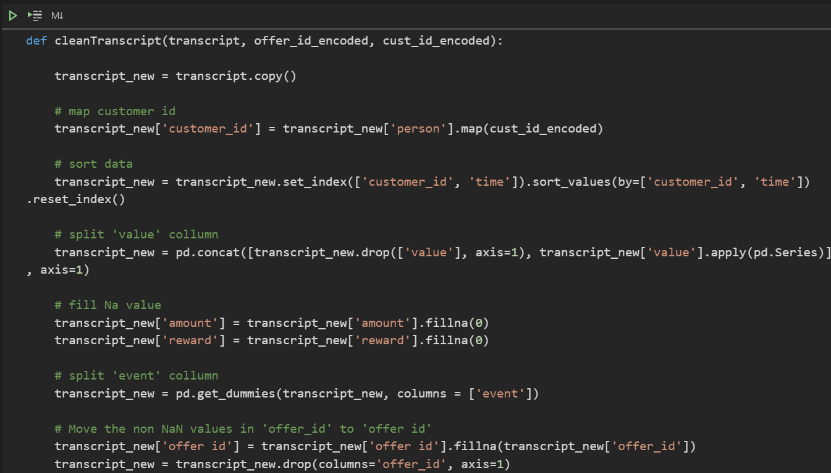
First of all, we will drop all the columns that have missing values of gender and age. Then we will change became\_member\_on into date value using lambda function. After that we will change the customer's age which is over 100 into NaN value. membership duration is a newly added column to calculate the duration. After using the class above, the data info will look like,

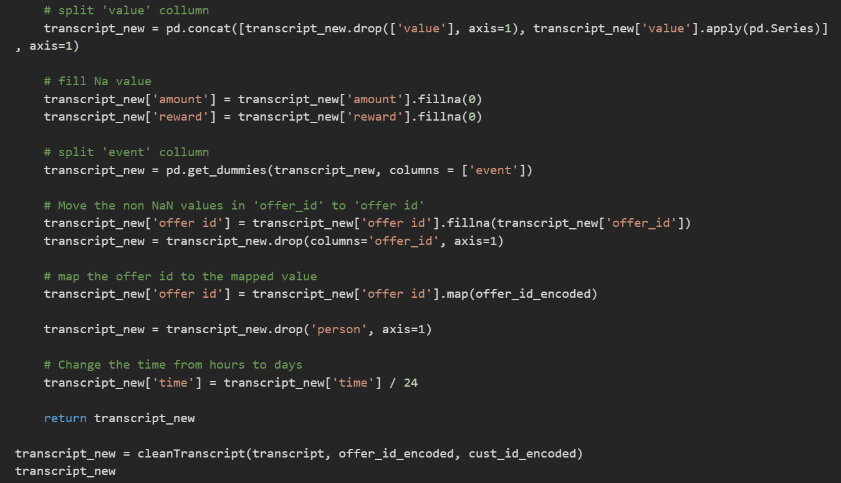
    

*Before cleaning                   After cleaning*

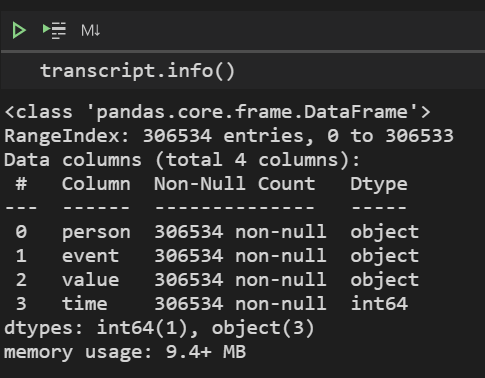
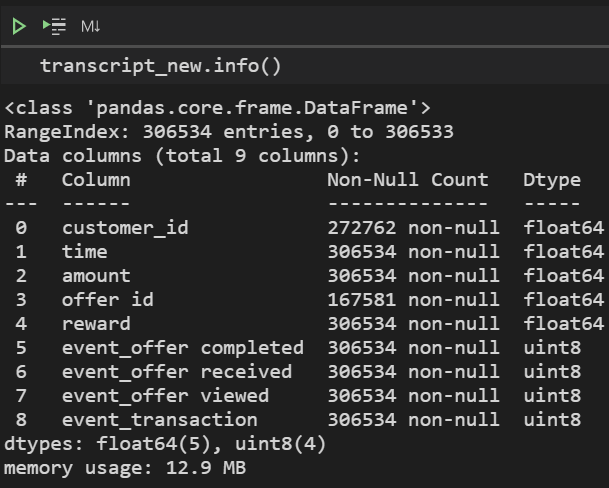
As we can see, gender columns have been modified to three different columns as well as the id. After dropping the null values, the total number of rows have been reduced from 17,000 to 14825.

For transcript.json,





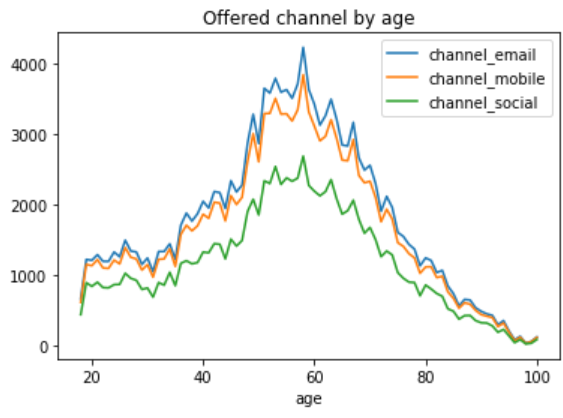
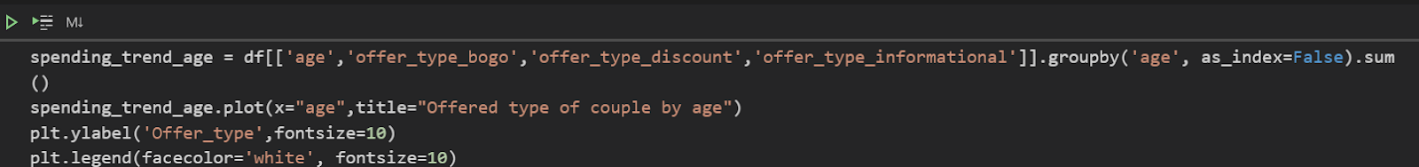
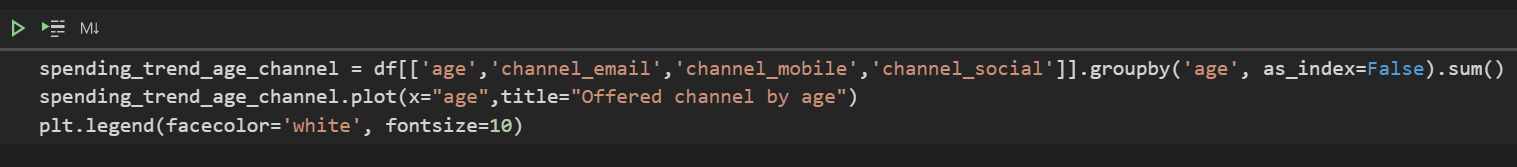
This is the function to clean to transcript.json, input will be the ‘transcriptmember’ which is a row dataframe read from the original data set. Second input will be cust\_id\_encoded and offer\_id\_encoded which is an encoded map of the custer\_id and offer\_id. We will fill out all the Na values in amount and reward to 0 and map the offer\_id to mapped value. Finally change the time from hours to days since the value is too huge. After all these processes, the info will look like

*Before cleaning   After cleaning*

## **Exploratory Data Analysis**

### Spending trend(age) by offered type of coupon & channel.

****

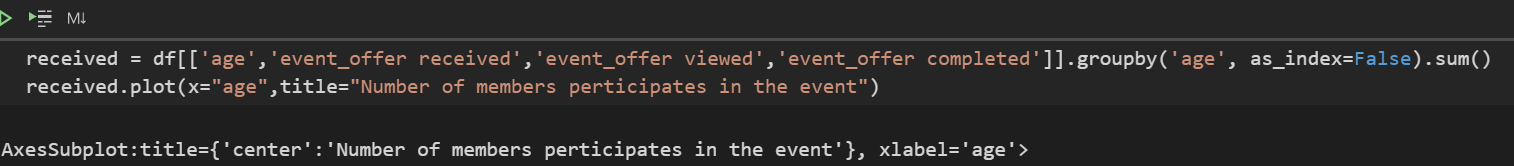
By using. plot function,

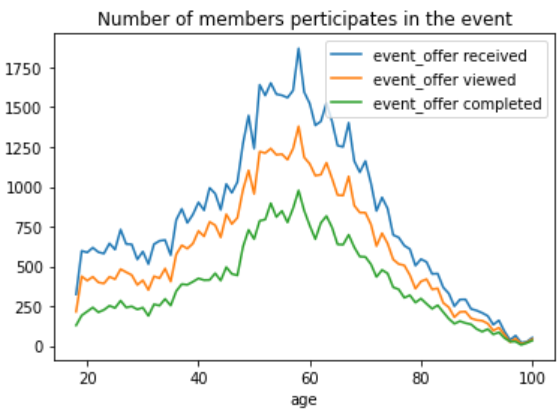
1. We can say that most of the members get their coupons through the email and mobile. However, the number of the members who got notice via email is slightly higher then the members who got notification through the mobile.

2. The type of the coupons are usually bogo or discount.  Members in their early 20s, early 30 get buy one get one coupon.

3. Starbucks is not promoting enough coupons by informational offer and channel through sns.

### Participant rate of event

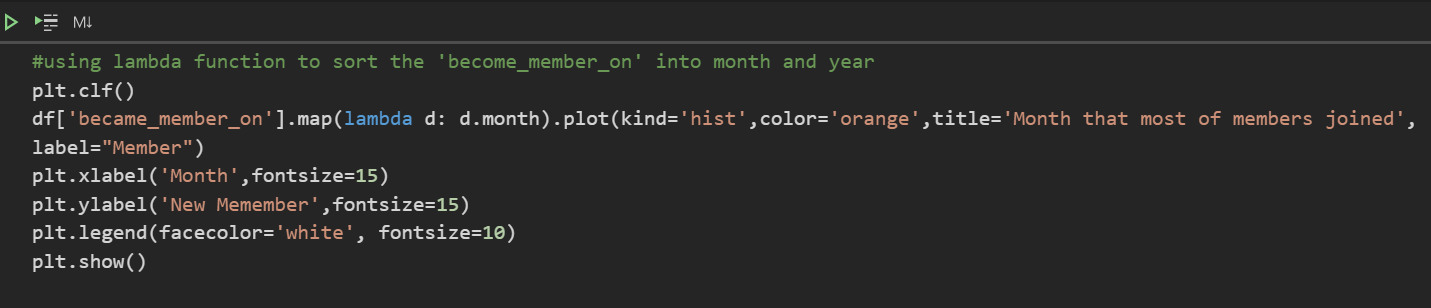
****

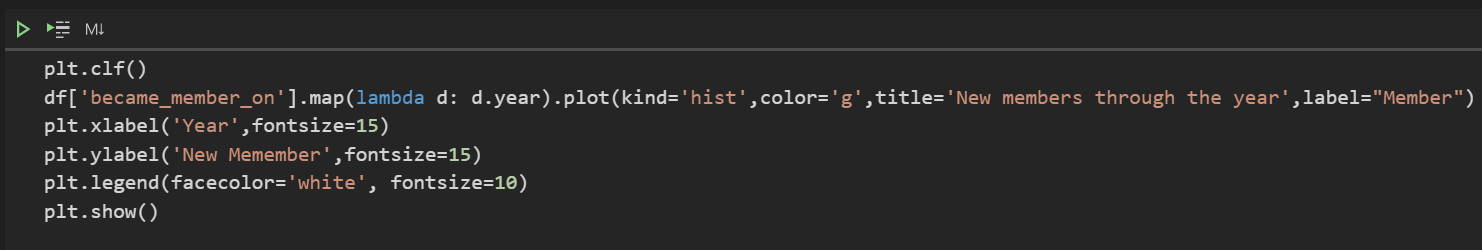
****

*Figure*

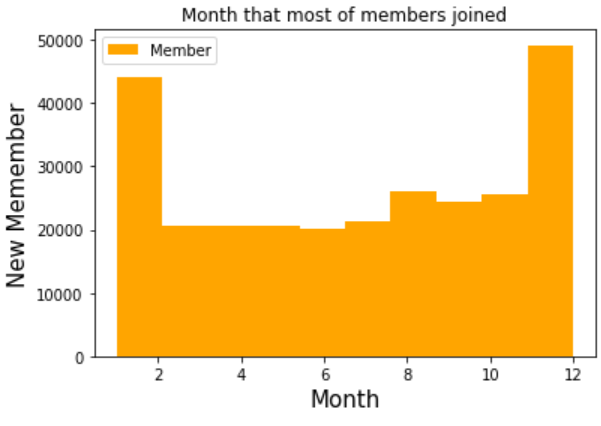
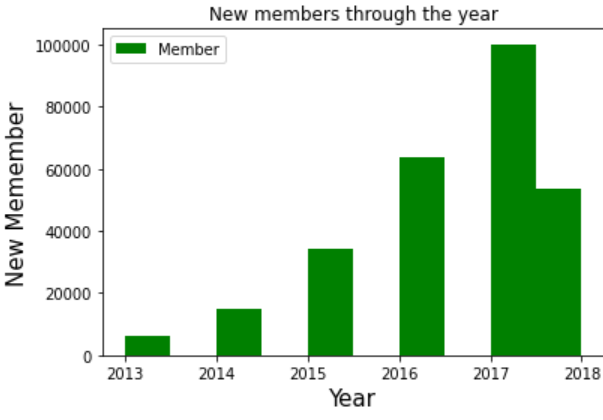
Even though Starbucks offers various promotions, only half of the users actually use the offer to get the drinks.

### Finding out which year and month have joined to become a member.

****



First, we used lambda to get month from became\_member\_on and displayed in the plot by month and the number of new members. Same method was used when it comes to the year graph. The result is,

*Figure*

1. We can say that December and January gained almost double the number of members from 2013 - 2018.

2. Starbucks is getting more and more members through the year.

3. Regarding this dataset being realized in the middle of 2018, we can assume that Starbucsk will get more new members at the end of 2018.

4. Nevertheless, the number of new members started to decrease from 2016. It seems like starbucks needs to improve their promotions to get more new members.

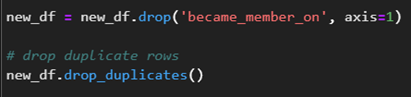
## **Feature Engineering**

1) After merging three datasets into one, there appears to have existed many NaN values in the data set. The first thing we have to do is to handle all NaN values that exist in the data set. We can see that all of the attributes related to the customer and offer are null. Since we only care about the transcripts with offer and customer information, we can drop all the rows without offer id and age values using dropna() function.

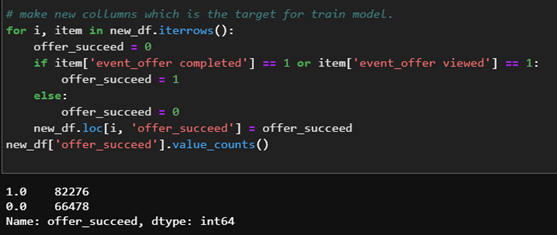


*Before dropping NaN value After dropping NaN values.*

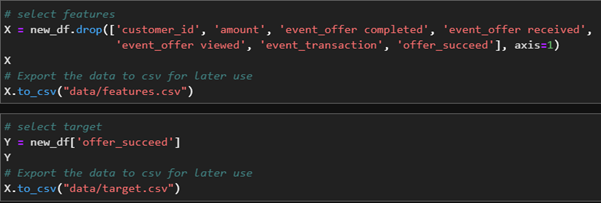
2) Next, using drop\_duplicates(), we drop duplicate rows that exist in the data.



3) From the target feature, as stated at the beginning of the assignment: **All of the offers which are viewed and completed are considered as success offers**. So we have to create a binary column offer\_succeed from event\_offers column in order to retrieve the results whether the customer.

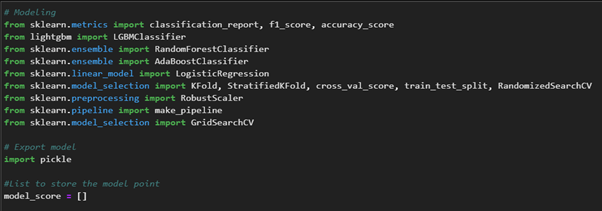


4) After modifying the data, we can see that there are 25 columns with the same amount of rows and fully converted to number type attributes. The final step we do is to select the list of features needed to train the model.



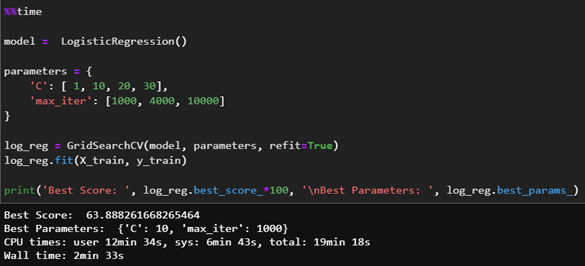
## **Data Modelling**

As stated from the beginning, we have to build a binary classification model to predict whether the offer is effective or not. Our team has selected four models for our project. The first two models are quite basic one which are LogisticRegression and RandomForest. The two remainings, AdaBoostClassifier and LGBMClassifier, are more advanced ones which are considered more effective. Before diving in the modelling and training process, first we import all the libraries needed.



1) LogisticRegression

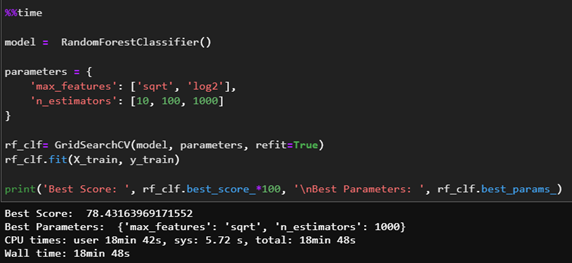
The LogisticRegression will construct a linear decision boundary to decide effective and non-effective offers. When having more than one explanatory variable, LogisticRegression is often used to calculate odds ratio. Excluding the response variable is binomial, the procedure is quite like multiple LinearRegression calculations. The result is the impact of each variable on the odds ratio of the observed event of interest. The main advantage is to avoid confounding effects by analyzing the association of all variables together. In this part we will use LogisticRegression method and perform parameter tuning with GridSearchCV to fit the training dataset to the model



The C value is Inverse of regularization strength, given how Scikit cites it as beings: C = 1 / λ. The relationship would be the smaller value denotes stronger regularization, and the C value that is best fit for this model is 1. Also, the maximum iteration that the model needs to fit the train data is 1000.

2) RandomForestClassifier

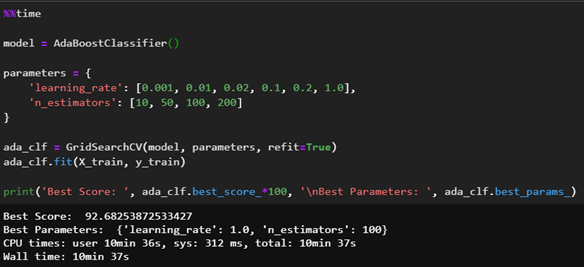
Random forest is a supervised learning algorithm. The "forest" it builds, is an ensemble of decision trees, commonly trained with the “bagging” method - a combination of learning models increases the overall result. The RandomForestClassifier uses several decision trees and trains each tree individually with a random fraction of the data. One big advantage of random forest is that it can be used for both classification and regression problems, which form the majority of current machine learning systems. We will also perform parameter tuning in max\_feature an n\_estimator parameter and fit the model with train data.



The max\_features parameter representing the maximum number of features Random Forest is allowed to try in an individual tree. The *sqrt* option will take square root of the total number of features in the individual run. For example, if the total number of variables is 100, we can only take 10 of them in an individual tree. *log2* is another similar type of option for max\_features. The best fit for this model is *sqrt*. n\_estimator is the number of trees you want to build before taking the maximum voting or averages of predictions. Higher number of trees give you better performance but makes your code slower, and 1000 is best fitted for this model.

3) AdaBoostClassifier

The AdaBoostClassifier is a combination of multiple classifiers in order to achieve higher accuracy score. The methodology of AdaboostClassifier is to set the weight of the classifier and train it iteratively. The most important parameter of AdaBoostClassifier is n\_estimators - number of weak learner and learning\_rate - weight of weak learner. We will use GridSearchCV to perform parameter tuning on these parameters and evaluate the model performance.



The parameter learning\_rate is the contribution of each model to the weights and defaults to 1. Reducing the learning rate will mean the weights will be increased or decreased to a small degree, forcing the model train slower (but sometimes resulting in better performance scores). n\_estimators is the number of models to iteratively train. The best result recorded in this model consists of learning\_rate 1.0 and n\_estimators 100.

4) LightGBMClassifier

LightGBMClassifier is a gradient boosting framework that uses tree-based learning algorithms. It is designed to be distributed and efficient as compared to other boosting algorithms. We will train the model to fit with our training data.

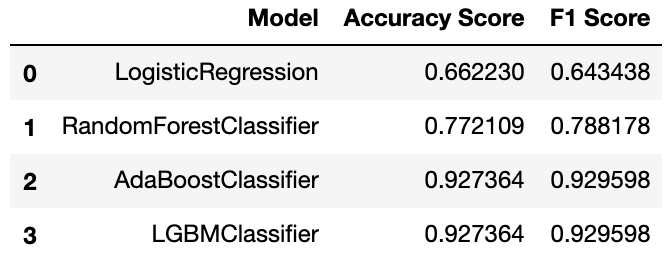


*Figure*

num\_leaves is one of the most important parameters that controls the **complexity** of the model. With it, you set the maximum number of leaves each weak learner has. Large num\_leaves increase accuracy on the training set and the chance of getting hurt by overfitting. max\_depth parameter controls max depth of each trained tree and will have an impact on the best value for the num\_leaves parameter, model performance, and training time. Pay attention if you use a large value of max\_depth, your model will likely be **overfit** to the train set. boosting\_type parameter depends on the processing unit type, the number of objects in the training dataset and the selected learning mode. min\_data\_in\_leaf is a very important parameter to prevent over-fitting in a leaf-wise tree. Its optimal value depends on the number of training samples and num\_leaves. Setting it to a large value can avoid growing too deep a tree but may cause under-fitting. The best parameter sets for this model is boosting\_type: gbdt, max\_depth: 5, min\_data\_in\_leaf: 20, num\_leaves: 6.

# **Result**

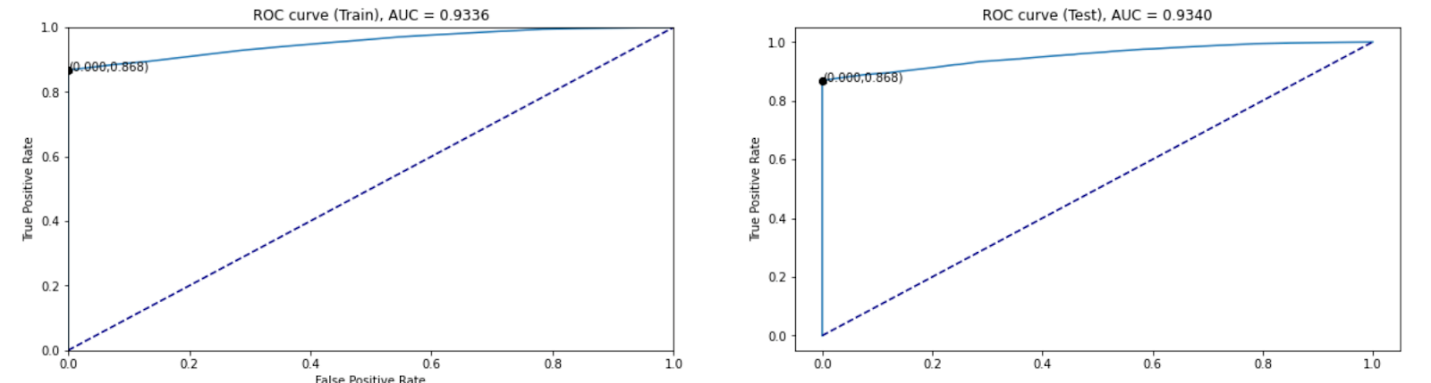
After we got information from EDA (Exploratory data analysis), we conducted to learn and evaluate many types of methods to fit the solving solution problem. We decided to use 4 types of model. Finally, the Model gives the following results:



From the table, we can see that AdaBoostClassifier and LGBMClassifier have approximately the same accuracy score and F1 score. However, the LGBMClassifier has a faster training time, so we can conclude that the best model is the LGBMClassifier

**LightGBM Model AUC - ROC Curve**

AUC - ROC is a method of calculating the performance of a classification model according to different classification thresholds. Assuming that with a binary classification problem (2 classes) using logistic regression, choosing classification threshold [0..1] will affect the classification ability of the model and it is necessary to calculate the impact level of the threshold. AUC stands for Area Under The Curve and ROC stands for Receiver Operating Characteristic. The ROC is a curve representing probability and the AUC represents the classification of the model. AUC-ROC is also known as AUROC (Area Under The Receiver Operating Characteristics). The meaning of AUROC can be interpreted as follows: It is the probability that a randomly sampled positive sample will be ranked higher than a randomly drawn negative sample. The higher the AUC, the more accurate the model is in classifying the classes.



As noted above, the closer the AUC number is to 1, the more accurate the model will classify. The closer the AUC is to 0.5, the worse classification performance will be. If it is closed to 0, the model will reverse classify the result (classify positive as negative and vice versa). From the graph, the AUC number is approximately 0.93 of both Train data and Test data, which means our model has a high performance.

# **API**

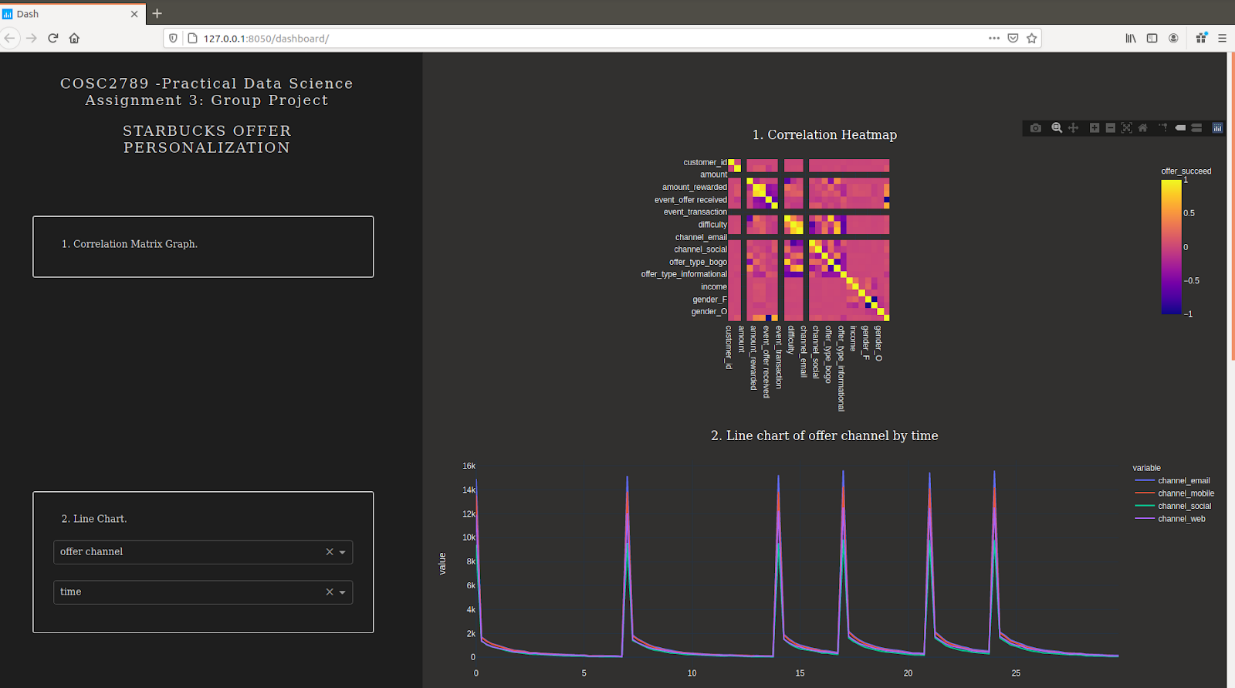
There are 2 type of API:

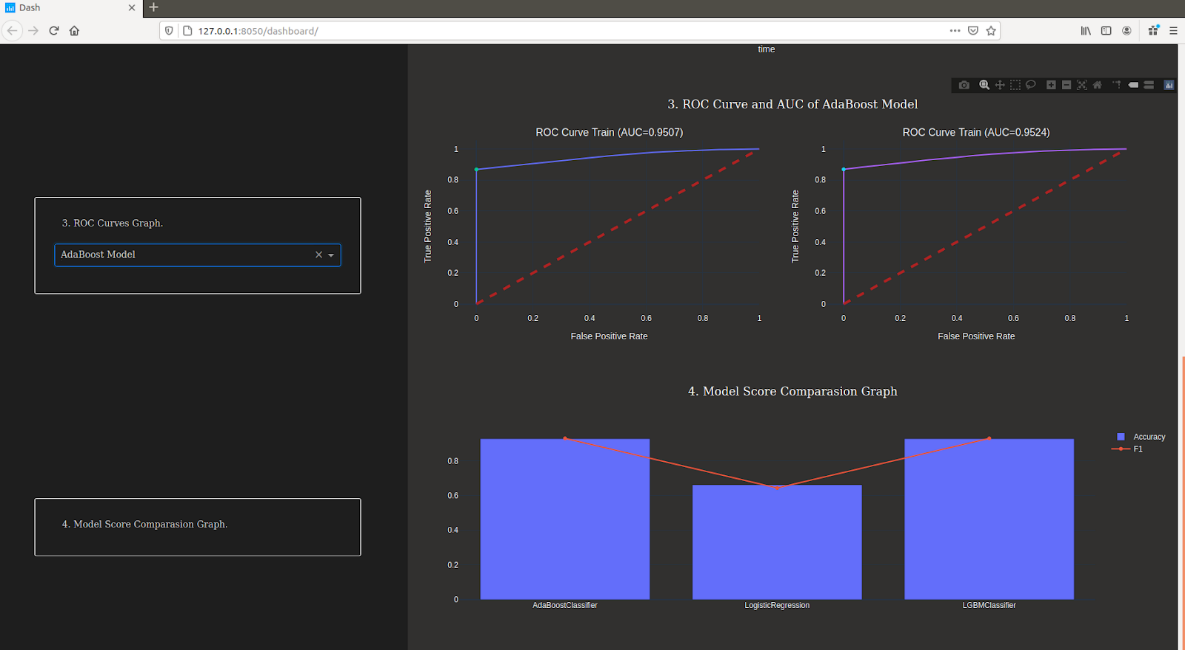
|  |  |  |
| --- | --- | --- |
| HTTP methods | Route | Description |
| GET | /api/evaluate/<model\_name> | Return a model score |
| GET | /api/predict/<model\_name> | Return a list of result predicted by model |
| POST | /api/predict\_offer\_effective | Return the result predict whether offer effective or not against specific customer |

# **Visualization Dashboard**

We use Dash library to build our dashboard with 4 plots which are:

* Correlation Matrix
* Using line chart to statistically the columns for each different offer group
* AUC and ROC curve
* Model score comparison graph

****

*****.*

# **Discussion**

In general, we have built a model with high accuracy and this project has illustrated how Starbucks use their data science to improve their business. However, several think we can improve our model. Firstly, in the feature selection process, we only use features from the data set. We can make some features from the given dataset to increase the accuracy of the model. Secondly, the model we use is a binary classification which is not a complicated one. We can divide the offer by its offer types and make predictions about the impact of the offer type to each customer which uses a multi classification model or clustering to have a deeper understanding about the customer demographic and gain more insight from the data.

# **Conclusion**

In conclusion, from the data of customer behavior of Starbucks application, we have applied the process of data science to draw some insights and build a predictive model to evaluate the effectiveness of an offer based on the customer profile. The first part of this process is data cleaning which is one of the most challenging parts in this project, because the raw data is in JSON type with dictionary structure. Thus, a tremendous amount of work has to be done when dealing with data such as one hot encoding categorical features, drop null value, merge the data sets. After finished cleaning, in the EDA part we have analyzed the data to acquire some insights about the customer spending trend with different offer types and customer membership duration over time. Finally, we have performed parameter tuning with 4 different binary classification models and found out that the LightGBMClassifier is the most effective model with the accuracy and f1 score over 0.9. In the future, some improvement can be done with this project. From the information, we can make more features to increase the accuracy of the model, other high-performance models such as XGBoost or CatBoost could be considered and the project can be taken to a further step which is to predict the best offer type for a specific customer.

# **References**

1. Albon, C. (2017, December 20). *Adaboost Classifier*. Chris Albon. <https://chrisalbon.com/machine_learning/trees_and_forests/adaboost_classifier/>
2. Bahmani, M. J. (2020, December 14). *Understanding LightGBM Parameters (and How to Tune Them)*. Neptune.Ai. <https://neptune.ai/blog/lightgbm-parameters-guide>
3. Donges, N. (2020, September 3). *A complete guide to the random forest algorithm*. Built In. <https://builtin.com/data-science/random-forest-algorithm>
4. Nature Editorial. (2018, October 30). *Why Jupyter is data scientists’ computational notebook of choice*. Nature. <https://www.nature.com/articles/d41586-018-07196-1>
5. Pal, S. (2020, August 14). *Scikit-learn Tutorial: Machine Learning in Python*. Dataquest. <https://www.dataquest.io/blog/sci-kit-learn-tutorial/>
6. Plotly. (2019, August 7). *Introducing Dash - Plotly*. Medium. <https://medium.com/plotly/introducing-dash-5ecf7191b503>
7. Sperandei, S. (2014). Understanding logistic regression analysis. *Biochemia Medica*, 12–18. <https://doi.org/10.11613/bm.2014.003>
8. Srivastava, T. (2020, June 26). *Tuning the parameters of your Random Forest model*. Analytics Vidhya. <https://www.analyticsvidhya.com/blog/2015/06/tuning-random-forest-model/>
9. *Welcome to LightGBM’s documentation! — LightGBM 3.1.1.99 documentation*. (n.d.). LightGBM. <https://lightgbm.readthedocs.io/en/latest/>
10. Business Insider. 2021. *Starbucks Is Rolling Out A New System To Convince You To Buy Exactly Want It Wants You To Buy*.  <https://www.businessinsider.com/starbucks-develops-personalization-system-2016-12>
11. Google Developers. 2021*. Classification: ROC Curve And AUC  |  Machine Learning Crash Course.* <https://developers.google.com/machine-learning/crash-course/classification/roc-and-auc>
12. Acutecaretesting.org. 2021*. ROC Curves – What Are They And How Are They Used?.*
13. <https://acutecaretesting.org/en/articles/roc-curves-what-are-they-and-how-are-they-used>
14. Medium. 2021*. Understanding AUC - ROC Curve.* [*https://towardsdatascience.com/understanding-auc-roc-curve-68b2303cc9c5*](https://towardsdatascience.com/understanding-auc-roc-curve-68b2303cc9c5)