CAPSTONE PROJECT REPORT

Machine Learning Engineer Nanodegree
EDIDIONG ALIGBE
April 15, 2023.

Domain Background

Starbucks is an American chain of coffeehouse, located in over 80 countries of the world and has its headquarters in Seattle, Washington, United States of America. It is known for its high-quality coffee and espresso-based drinks like Caffe Latte, Americano, and Cappuccino. It also offers hot chocolate, cold beverages, teas, salads, pastries and sandwiches.

Starbucks offers promotions designed to attract new customers and encourage existing customers to make repeat purchases, building brand loyalty. These promotions could be in form of loyalty programs, buy one, get one free (BOGO), gift cards, etc. These offers come through emails, the web, mobile app, and social media. The company keeps a record of customers' interactions with these promotions and the data is useful to serve the customer better based on their activities and preferences

Machine Learning is an area of Artificial Intelligence that allows software applications to learn from data and predict outcomes without being explicitly programmed to do so. With more data, they learn to be more accurate and improve their performance.
The aim of the project is to use Machine Learning and the data from customers' interactions with promotions to help Starbucks to run their promotions successfully.

Dataset and inputs

The dataset for this project contains simulated data that mimics customer behaviour on the Starbucks mobile app. These data comes in three json files:

1. profile.json: Contains demographic data for each customer. It is made up of 17,000 customers with 5 fields -

- gender: (categorical) M, F, O, or null
- age: (numeric) missing value encoded as 118
- id: (string/hash)
- became_member_on: (date) format:YYYYMMDD
- income: (numeric)

2. portfolio.json: Contains data on offers sent during 30-day test period. It is made up of 10 offers with 6 fields -

- reward: (numeric) money awarded for the amount spent
- channels: (list) web, email, mobile, social

- difficulty: (numeric) money required to be spent to receive reward
- duration: (numeric) time for offer to be open, in days
- offer_type: (string) bogo, discount, informational
- id: (string/hash)

3. transcript.json: Contains records for customer transactions, offers received, offers viewed, and offers completed. It is made up of 306648 transactions with 4 fields -

- person: (string/hash)
- event: (string) offer received, offer
- value: (dictionary) different values depending on event type
- offer id: (string/hash) not associated with any "transaction"
- amount: (numeric) money spent in "transaction"
- reward: (numeric) money gained from "offer completed"
- time: (numeric) hours after start of test

Problem statement

The aim of sending promotions and offers to customers is to encourage customers to make purchases. It will not be an intelligent move to send the same promotion to all the customers at once, this may result in promotions that do not yield the expected results. The goal of this project is to make use of the data from Starbucks to determine what promotion to send to a demography of customers that will result in higher customer purchases.

Evaluation Metrics

This type of machine learning problem is a multi-class classification machine learning problem because you have to use a combination of different channels to send an offer. For example, for a Buy-one-get-one (BOGO) offer, Starbucks used a combination of email, web and social media, for another BOGO offer, Starbucks used email, web, mobile and social media to send these offers.

Accuracy: Accuracy is one of the most popular metrics in multi-class classification and it is directly computed from the confusion matrix. The formula of the Accuracy considers the sum of True Positive and True Negative elements at the numerator and the sum of all the entries of the confusion matrix at the denominator.

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

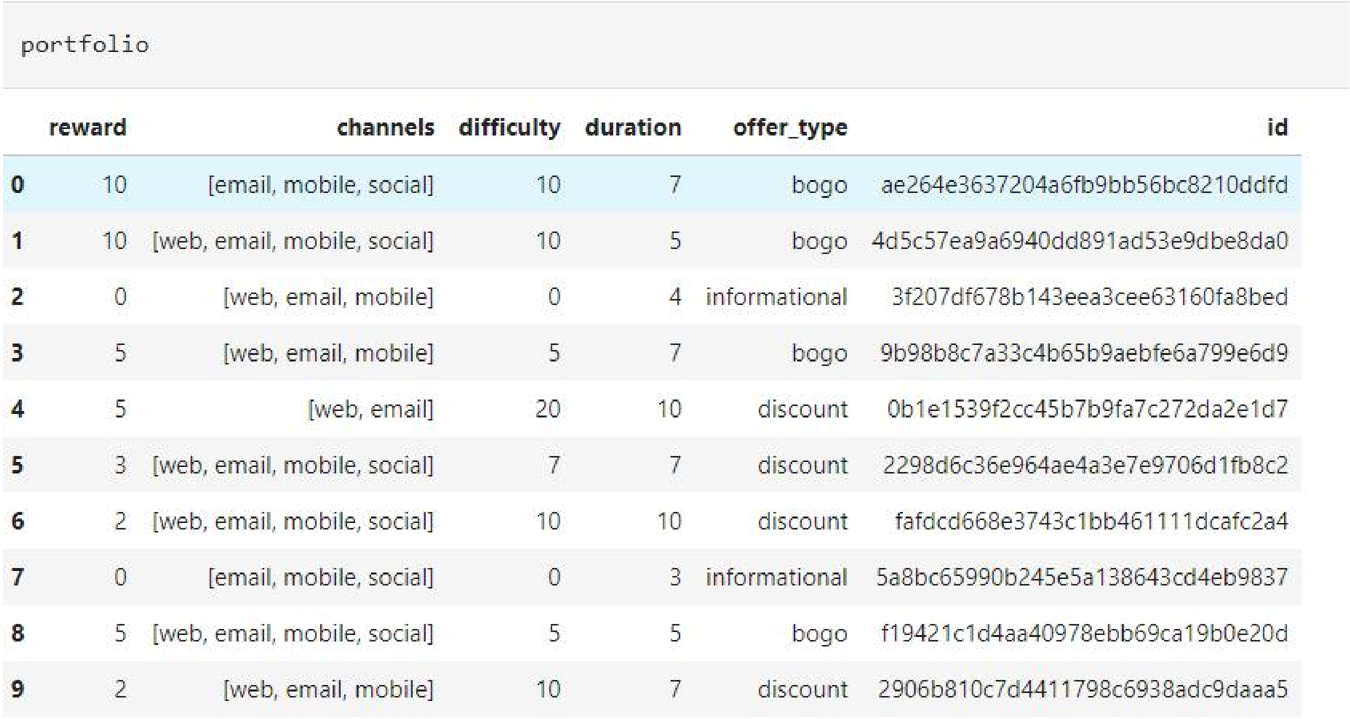
Benchmark

The best performing models from AutoGluon model training should have an accuracy between 0.95 and 0.78.

Data Analysis and Cleaning

1. **Portfolio data-frame**: The following steps below were used to clean the portfolio dataset.

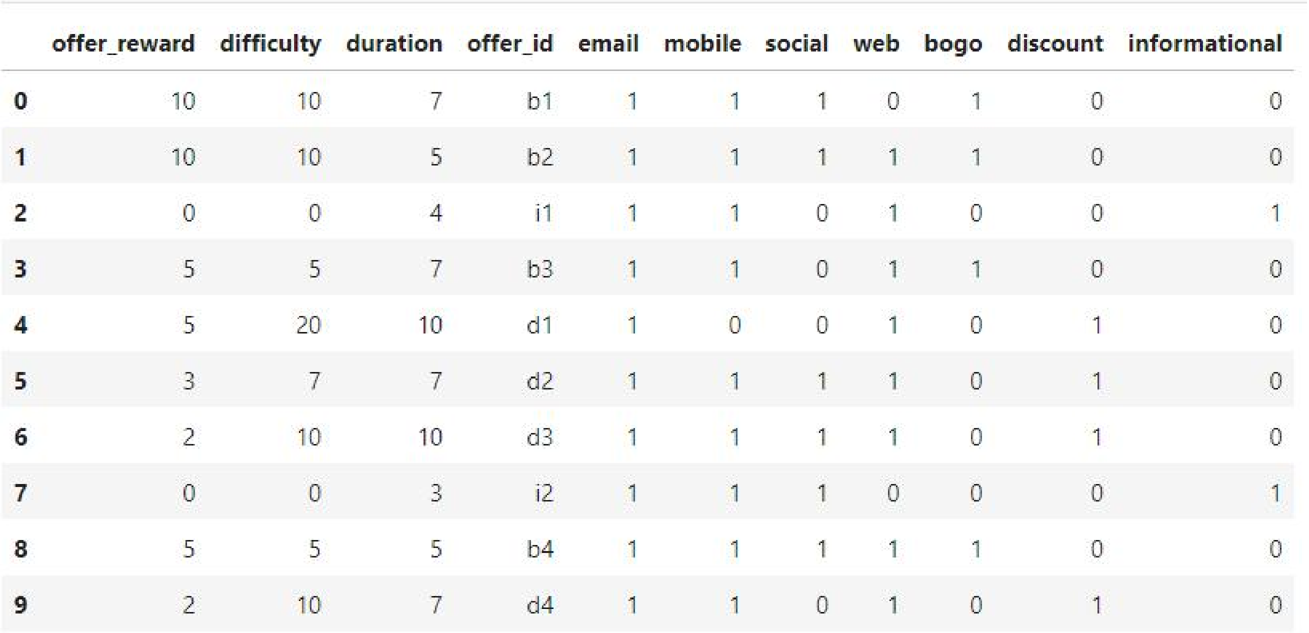
- Convert the categorical columns value to numerical values using one-hot encoding.
- Drop the original categorical columns.
- Change the name of the ‘id’ column to ‘offer_id’ and shorten the values.



```
portfolio
```

	reward	channels	difficulty	duration	offer_type	id
0	10	[email, mobile, social]	10	7	bogo	ae264e3637204a6fb9bb56bc8210ddfd
1	10	[web, email, mobile, social]	10	5	bogo	4d5c57ea9a6940dd891ad53e9dbe8da0
2	0	[web, email, mobile]	0	4	informational	3f207df678b143eea3cee63160fa8bed
3	5	[web, email, mobile]	5	7	bogo	9b98b8c7a33c4b65b9aebfe6a799e6d9
4	5	[web, email]	20	10	discount	0b1e1539f2cc45b7b9fa7c272da2e1d7
5	3	[web, email, mobile, social]	7	7	discount	2298d6c36e964ae4a3e7e9706d1fb8c2
6	2	[web, email, mobile, social]	10	10	discount	fafdcd668e3743c1bb461111dcafc2a4
7	0	[email, mobile, social]	0	3	informational	5a8bc65990b245e5a138643cd4eb9837
8	5	[web, email, mobile, social]	5	5	bogo	f19421c1d4aa40978ebb69ca19b0e20d
9	2	[web, email, mobile]	10	7	discount	2906b810c7d4411798c6938adc9daaa5

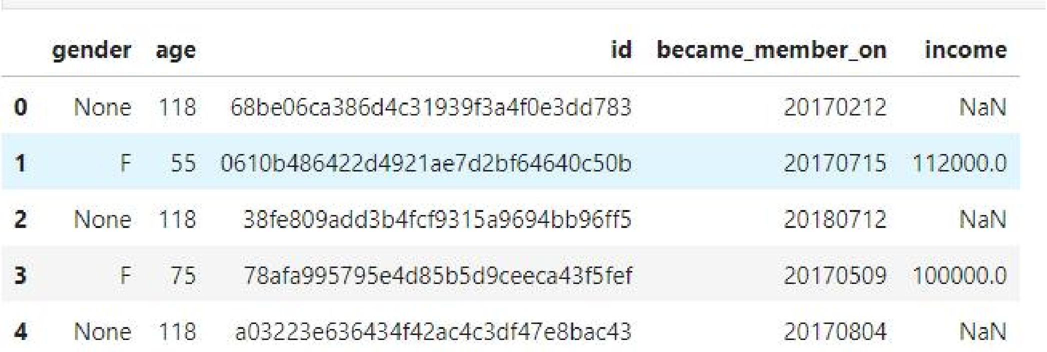
Portfolio data-frame before cleaning



	offer_reward	difficulty	duration	offer_id	email	mobile	social	web	bogo	discount	informational
0	10	10	7	b1	1	1	1	0	1	0	0
1	10	10	5	b2	1	1	1	1	1	0	0
2	0	0	4	i1	1	1	0	1	0	0	1
3	5	5	7	b3	1	1	0	1	1	0	0
4	5	20	10	d1	1	0	0	1	0	1	0
5	3	7	7	d2	1	1	1	1	0	1	0
6	2	10	10	d3	1	1	1	1	0	1	0
7	0	0	3	i2	1	1	1	0	0	0	1
8	5	5	5	b4	1	1	1	1	1	0	0
9	2	10	7	d4	1	1	0	1	0	1	0

Portfolio data-frame after cleaning

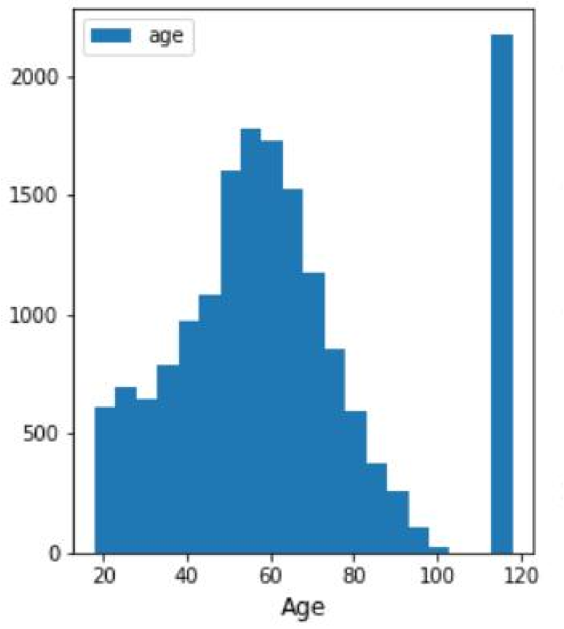
2. **Profile data-frame**: A sample of the data in profile data-frame looks like the figure below:



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

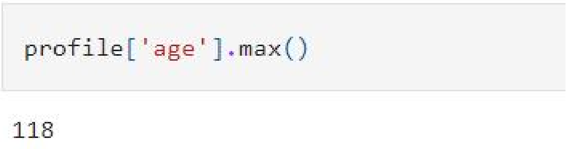
Profile data before cleaning

Before the data was cleaned, I was able to gather the information below from the data-frame.



Age graph in profile data-frame

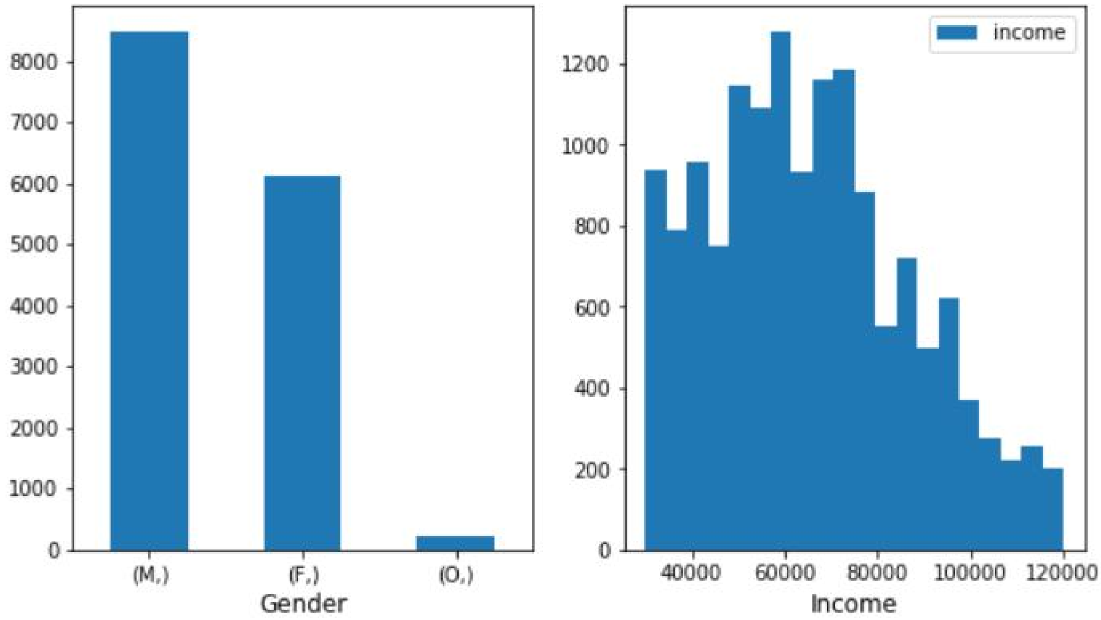
From the age graph, there is an outlier, close to 120. The highest age value is 118.



```
profile['age'].max()
```

```
118
```

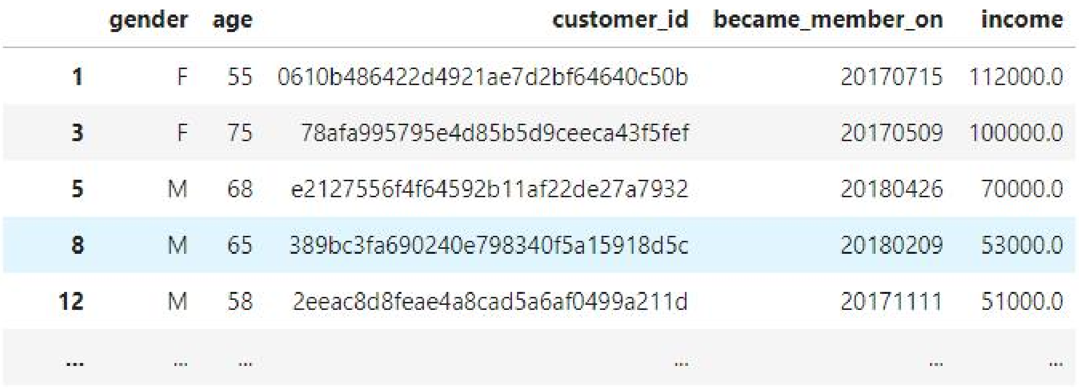
Highest age value



Gender and Income graph

The steps below were taken to clean the profile data-frame:

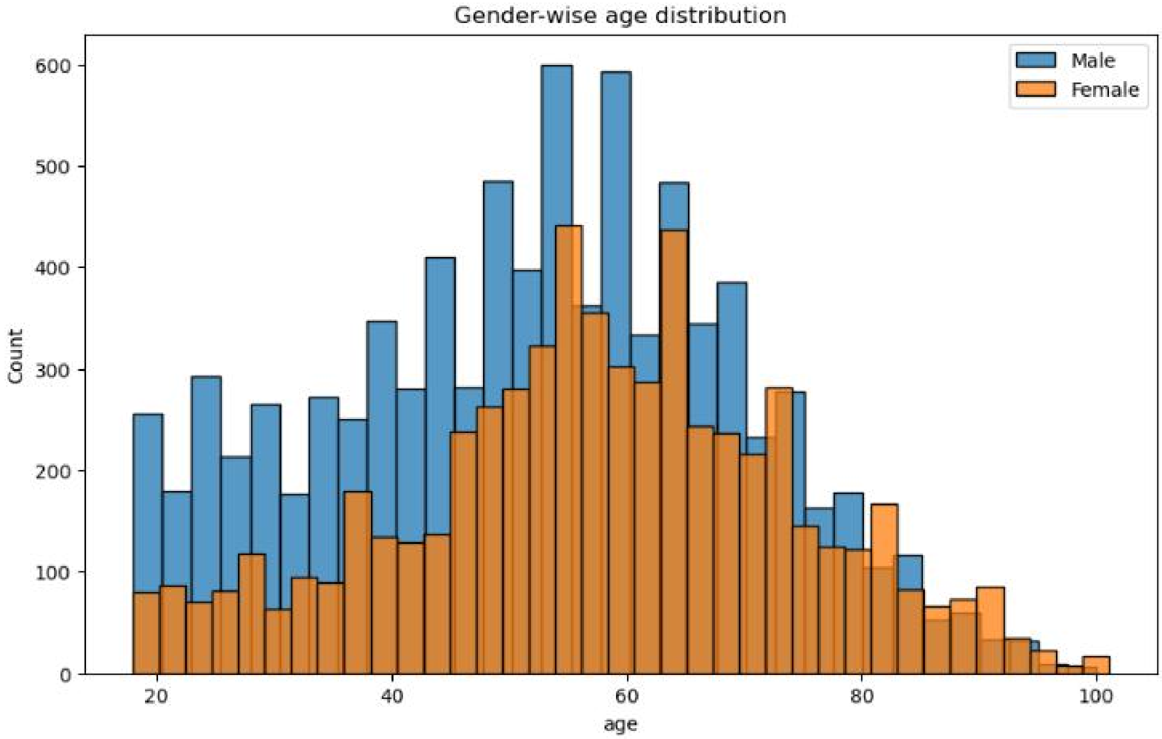
- I observed that 'gender' and 'income' columns both contain null where 'age' is 118. The rows where 'age' is 118 are dropped.
- Change the name of the 'id' column to 'customer_id'.



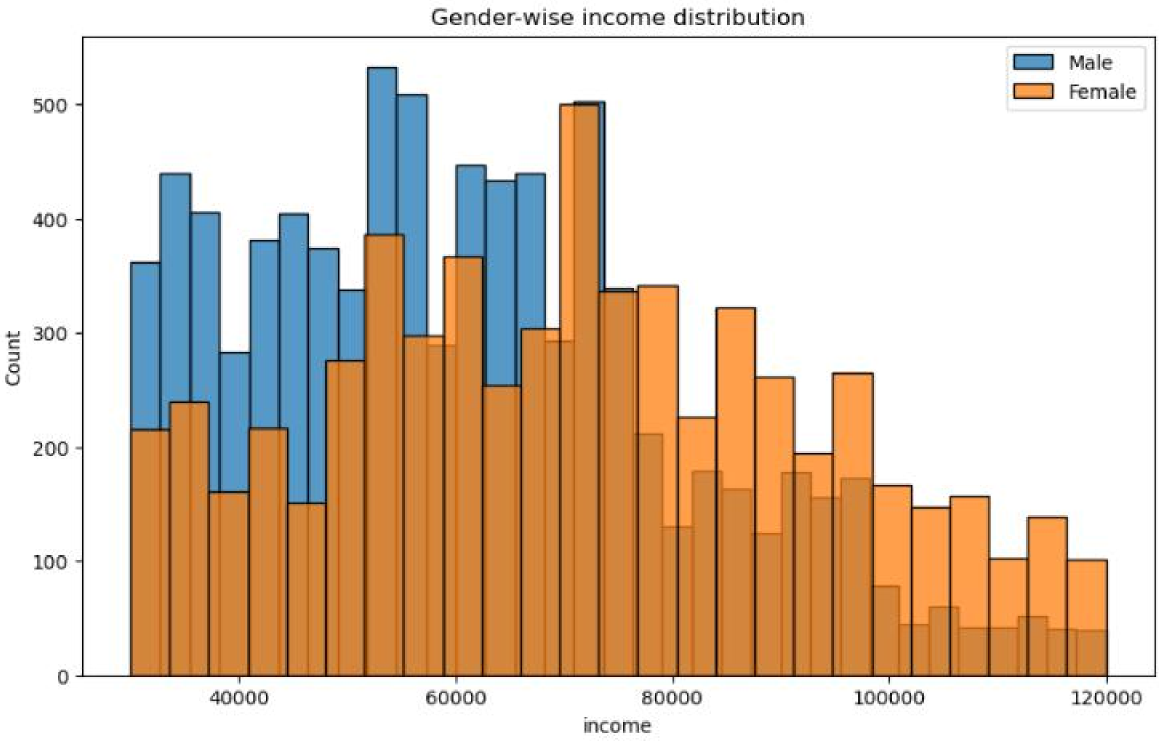
	gender	age	customer_id	became_member_on	income
1	F	55	0610b486422d4921ae7d2bf64640c50b	20170715	112000.0
3	F	75	78afa995795e4d85b5d9ceeca43f5fef	20170509	100000.0
5	M	68	e2127556f4f64592b11af22de27a7932	20180426	70000.0
8	M	65	389bc3fa690240e798340f5a15918d5c	20180209	53000.0
12	M	58	2eeac8d8feae4a8cad5a6af0499a211d	20171111	51000.0
...	...	...	...	...	...

Profile data after cleaning

After data cleaning, I pulled out the following information.

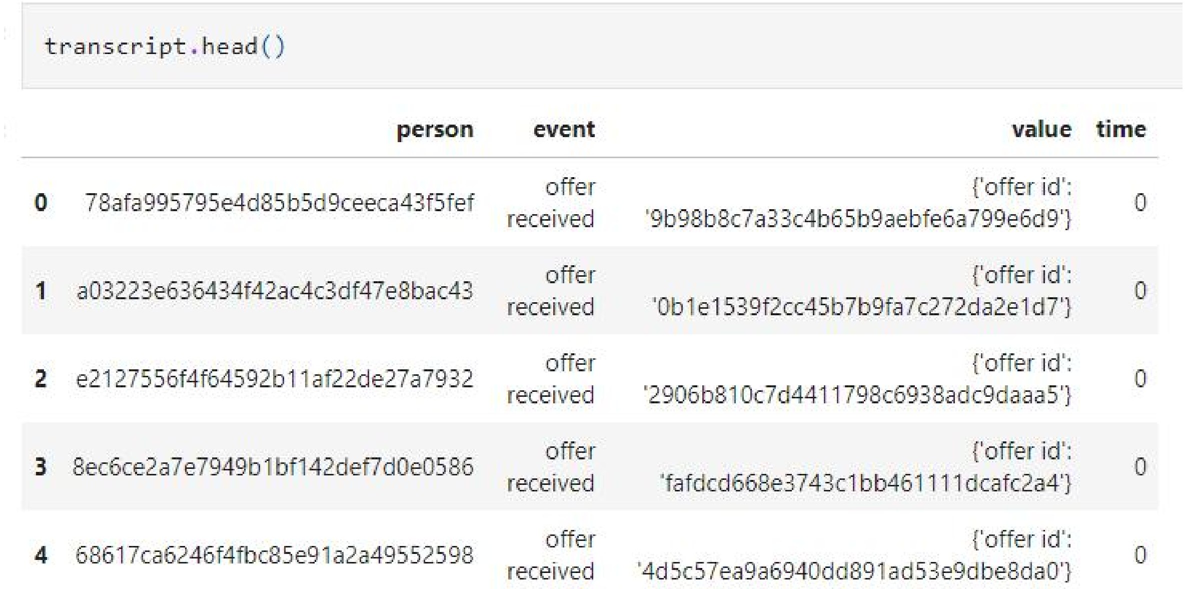


<u>Gender-wise age distribution</u>



Gender-wise income distribution

3. **Transcript data-frame:** A sample data from transcript data-frame.



```
transcript.head()
```

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

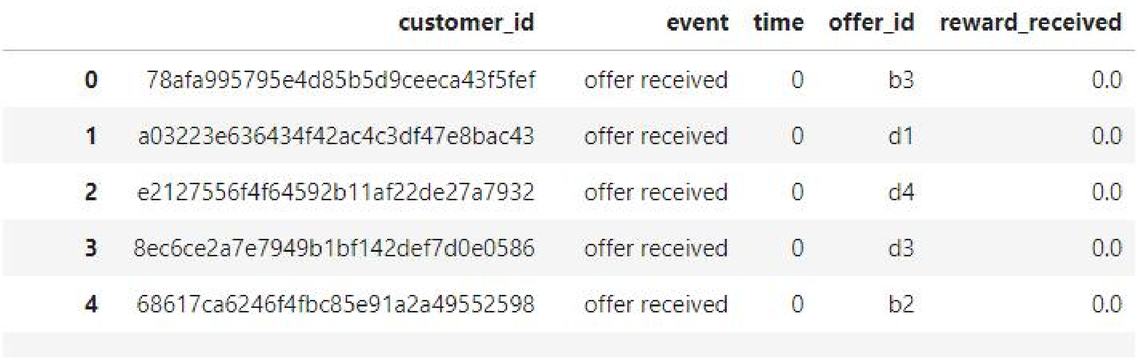
Transcript data before cleaning

There are no null values in the transcript data-frame. According to statistics 76,277 offers were received by customer. Among the offers sent 57,725 offers were viewed by customers and only 33,579 of these offers were completed by customers. The number of transactions carried out by customers during the time frame of these data are 139,953 transactions.

Transaction data does not give any information on offers, it covers both offer-related transactions and non-offers transactions. It does not specify whether the offer is based on an offer or not.

The steps below were taken to clean the data.

- Change the name of the 'person' column to 'customer_id'.
- Extract offer id, reward from 'value' column.
- Drop original 'value' column.
- Remove rows where 'event' column contain 'transaction'.
- Shorten offer_id column values.



	customer_id	event	time	offer_id	reward_received
0	78afa995795e4d85b5d9ceeca43f5fef	offer received	0	b3	0.0
1	a03223e636434f42ac4c3df47e8bac43	offer received	0	d1	0.0
2	e2127556f4f64592b11af22de27a7932	offer received	0	d4	0.0
3	8ec6ce2a7e7949b1bf142def7d0e0586	offer received	0	d3	0.0
4	68617ca6246f4fbc85e91a2a49552598	offer received	0	b2	0.0

Transcript data after cleaning

Data Preprocessing

The three datasets were merged together forming a new data-frame called 'df'. A new column was added called 'target' column. The values of these columns were computed from 'reward' and 'event' column using the algorithm below:

1. Create an empty list.
2. Loop through the rows of 'df' data-frame.
3. For each row
    - If 'event' is 'offer completed' and 'offer_id' is 'b1', add the value of 1 to the list.
    - If 'event' is 'offer completed' and 'offer_id' is 'b', add the value of 2 to the list.
    - If 'event' is 'offer completed' and 'offer_id' is 'b3', add the value of 3 to the list.
    - If 'event' is 'offer completed' and 'offer_id' is 'b4', add the value of 4 to the list.
    - If 'event' is 'offer completed' and 'offer_id' is 'd1', add the value of 5 to the list.
    - If 'event' is 'offer completed' and 'offer_id' is 'd2', add the value of 6 to the list.
    - If 'event' is 'offer completed' and 'offer_id' is 'd3', add the value of 7 to the list.
    - If 'event' is 'offer completed' and 'offer_id' is 'd4', add the value of 8 to the list.
    - Else add 0 to the list

The ‘customer_id’ column was dropped from the ‘df’ data-frame.

Model training

The model training is done using AutoGluon. AutoGluon is an open source AutoML library developed by Amazon Web Services (AWS) that automates machine learning (ML) and deep learning (DL) producing very efficient models. It automates other ML tasks such as pre-processing, feature engineering, model selection and hyperparameter.

1. The data was divided into training and testing data using Sklearn train_test_split function. AutoGluon inferred the problem to be multiclass classification problem. The number of columns use in training the model is 18, namely: gender, offer_id, income, age, time, offer_reward, difficulty, duration, mobile, social, web, bogo and discount, became_member_on, event and reward_received.

The multiclassification models selected by AutoGluon to train the models and their score is shown below:

SN	**Model**	**Score**
1	XGBoost_BAG_L1	
2	WeightedEnsemble_L3	1.000000
3	LightGBMXT_BAG_L2	1.000000
4	NeuralNetFastAI_BAG_L2	1.000000
5	LightGBMXT_BAG_L1	1.000000
6	WeightedEnsemble_L2	1.000000
7	NeuralNetFastAI_BAG_L1	1.000000
8	KNeighborsUnif_BAG_L1	0.778821
9	KNeighborsDist_BAG_L1	0.744170

The accuracy score of the models is not within the specified benchmark of 0.95 to 0.7. Evaluation on test data produced the accuracy value of 1.000000.

There seemed to be an issue of over-fitting.

Hyperparameter optimization

Since AutoGluon takes care of hyperparameter optimization perfectly, some columns will be dropped to help the accuracy of the models to fall within the specified range.

The following columns were dropped from the dataframe.

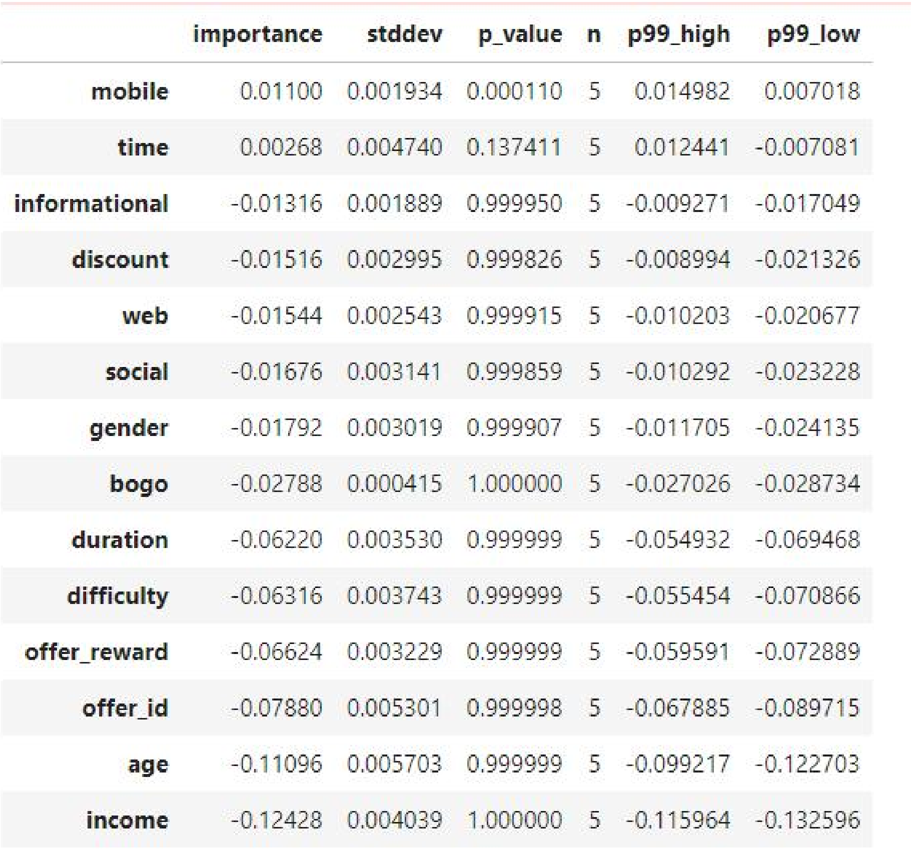
1. became_member_on
2. event
3. reward_received

Another model training with the dropped columns gave the scores below:

SN	Model	Score
1	WeightedEnsemble_L3	0.829794
2	LightGBMXT_BAG_L2	0.829794
3	NeuralNetFastAI_BAG_L2	0.826543
4	LightGBMXT_BAG_L1	0.814044
5	WeightedEnsemble_L2	0.814044
6	NeuralNetFastAI_BAG_L1	0.800721
7	KNeighborsUnif_BAG_L1	0.764087
8	KNeighborsDist_BAG_L1	0.738550

The models produced accuracy scores that falls within the benchmark range.

AutoGluon computed feature importance as follows:



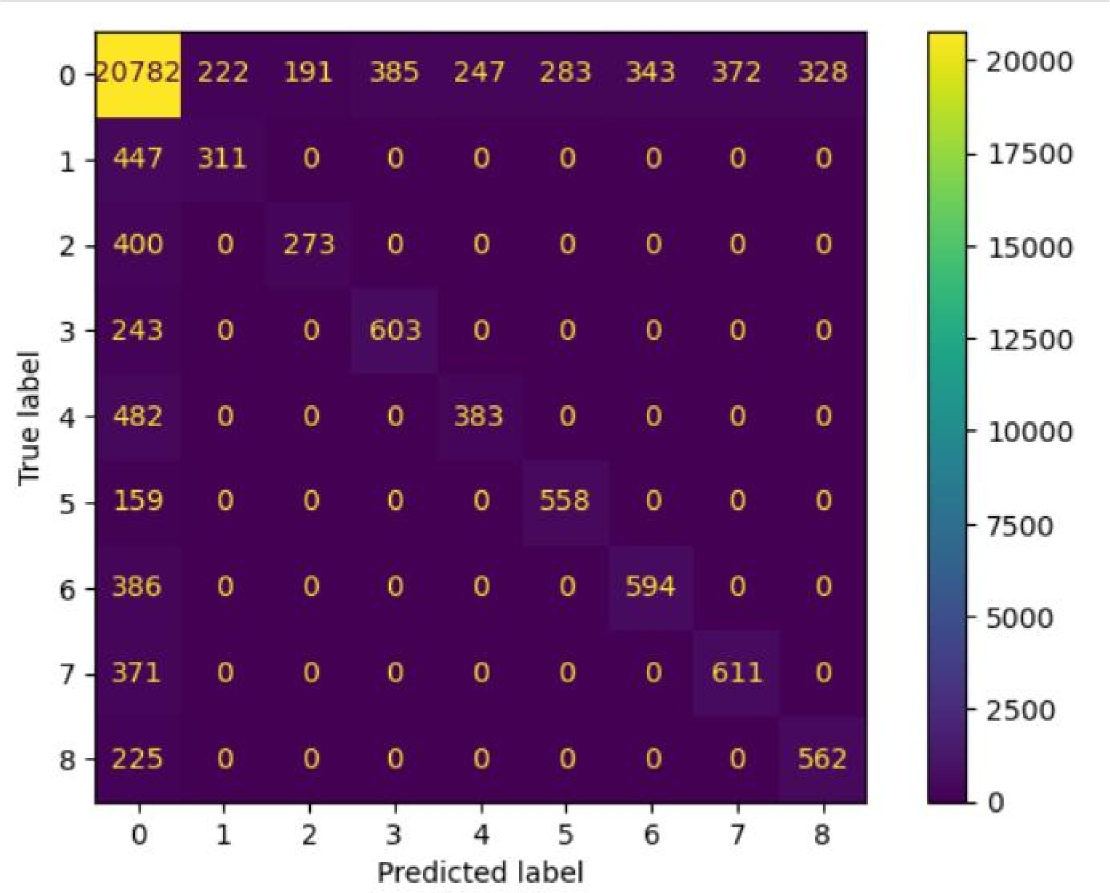
	importance	stddev	p_value	n	p99_high	p99_low
mobile	0.01100	0.001934	0.000110	5	0.014982	0.007018
time	0.00268	0.004740	0.137411	5	0.012441	-0.007081
informational	-0.01316	0.001889	0.999950	5	-0.009271	-0.017049
discount	-0.01516	0.002995	0.999826	5	-0.008994	-0.021326
web	-0.01544	0.002543	0.999915	5	-0.010203	-0.020677
social	-0.01676	0.003141	0.999859	5	-0.010292	-0.023228
gender	-0.01792	0.003019	0.999907	5	-0.011705	-0.024135
bogo	-0.02788	0.000415	1.000000	5	-0.027026	-0.028734
duration	-0.06220	0.003530	0.999999	5	-0.054932	-0.069468
difficulty	-0.06316	0.003743	0.999999	5	-0.055454	-0.070866
offer_reward	-0.06624	0.003229	0.999999	5	-0.059591	-0.072889
offer_id	-0.07880	0.005301	0.999998	5	-0.067885	-0.089715
age	-0.11096	0.005703	0.999999	5	-0.099217	-0.122703
income	-0.12428	0.004039	1.000000	5	-0.115964	-0.132596

Feature importance

Model Evaluation

Accuracy score: 0.829172406841168

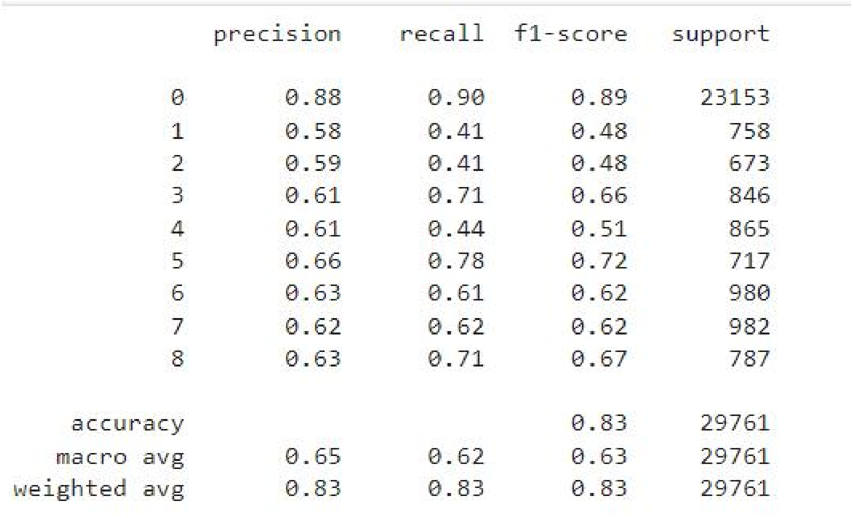
Confusion Matrix for the model training



	0	1	2	3	4	5	6	7	8
0	20782	222	191	385	247	283	343	372	328
1	447	311	0	0	0	0	0	0	0
2	400	0	273	0	0	0	0	0	0
3	243	0	0	603	0	0	0	0	0
4	482	0	0	0	383	0	0	0	0
5	159	0	0	0	0	558	0	0	0
6	386	0	0	0	0	0	594	0	0
7	371	0	0	0	0	0	0	611	0
8	225	0	0	0	0	0	0	0	562

Confusion Matrix

Metric scores calculated by evaluation:



```
              precision    recall  f1-score   support

           0       0.88      0.90      0.89     23153
           1       0.58      0.41      0.48       758
           2       0.59      0.41      0.48       673
           3       0.61      0.71      0.66       846
           4       0.61      0.44      0.51       865
           5       0.66      0.78      0.72       717
           6       0.63      0.61      0.62       980
           7       0.62      0.62      0.62       982
           8       0.63      0.71      0.67       787

    accuracy                           0.83     29761
   macro avg       0.65      0.62      0.63     29761
weighted avg       0.83      0.83      0.83     29761
```

Metric scores

Conclusion

The model precision score of 0.88 and F1-score of 0.89 shows that the model did not overfit. Deciding which features to train on was a challenge, would try with a combination of other features to see the performance of the models.