Student: Katharina Mayr
Date: 2022-09-16
Email: katharina.mayr@adidas.com

Capstone Project Report– Starbucks Capstone Challenge

Problem Statement

Starbucks is interested in retaining customers and strengthening customer relationships by providing the best possible service and offers to customers. Starbucks wants to identify the best offer for each customer on an individual personalized offer that at the same time maximizes revenue and provides an optimal customer experience. Some customers might respond best to discount or bogo ("Buy one get one free") offers while offers prefer to receive purely informational offers or do not want to receive offers at all.

The challenge of the Starbucks Capstone project is to model or quantify the relationship between a person's demographics and his or her response to a specific offer type.

Three different data sources are utilized to solve the presented challenge. First, transactional data showing user purchases made on the Starbucks app including the timestamp of the purchase and the amount of money spent on the purchase is used. This data includes records of the user receiving, viewing, and completing offers as well as records for transactions that were made without an offer.

Second, information containing offer ids and meta data about each offer (duration, type, difficulty, reward, channels) is given. Lastly, demographic data is given for each customer (age, gender income, date of the creation of the app account).

To address the Starbucks Capstone Project challenge, a machine learning model that predicts how much someone will spend based on demographics and offer type will be built. More precisely, the XGBoost (eXtreme Gradient Boosting)[1] algorithm with regression objective will be utilized to predict the monetary spend of customers.

[1] Chen, T., & Guestrin, C. (2016). XGBoost: A Scalable Tree Boosting System. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 785–794). New York, NY, USA: ACM. https://doi.org/10.1145/2939672.2939785

The results of the XGBoost model will be compared to a simple linear regression model which takes the same input features.

To evaluate the performance of the proposed model the RSME (root-mean-square error) and R^2 score, two popular evaluation metrics for regression use cases, will be used.

Data Exploration and Analysis

Three different datasets are combined to solve the presented Starbucks challenge. These are described in more detail in this section:

First and most importantly, simulated transactional data which records received, viewed, and completed offers as well as all other transactions of customers on the Starbucks rewards mobile app is available. The dataset contains in total 306,534 rows and is made up of fours columns. It holds a customer id (person), a record description (i.e. transaction, offer received, offer viewed, etc., event), a time column indicating the time in hours since the start of the test (time) and a value column (value) which contains either an offer id or transaction amount depending on the record (see figure 1). In the case of an offer received or viewed in the event column, the value column contains an offer id while for a completed offer it contains the reward a customer received.

	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

Figure 1: Sample of transactional data

There are no missing values in the transactional data (see figure 2).

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 306534 entries, 0 to 306533
Data columns (total 4 columns):
 #   Column  Non-Null Count   Dtype
---  ------  --------------   -----
 0   person  306534 non-null  object
 1   event   306534 non-null  object
 2   value   306534 non-null  object
 3   time    306534 non-null  int64
dtypes: int64(1), object(3)
memory usage: 9.4+ MB
```

Figure 2: Missing values in transactional data

Second, portfolio data which includes meta data for each offer id is provided to solve the presented problem. It holds meta data on ten different offer ids. Next to the offer id (id), it contains a description of the type of offer (offer_type). Possible offer types are BOGO (“buy one get one free”, four offers present in the data), discount (four offers present in the data), and informational (two different offers present in the data). The difficulty column contains the minimum required spend to complete an offer, while the duration column holds information on the time an offer is open in days. Channels describes the channels through which customers could have potentially received the specific offer. Reward describes the monetary reward the customer receives when completing an offer. Reward, difficulty, and duration are of integer type, while channels and offer_type are of categorical / object type. Figure 3 shows a sample excerpt of the data.

	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

Figure 3: Sample of portfolio data

As can be seen in figure 4, the portfolio data does not contain any missing values.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10 entries, 0 to 9
Data columns (total 6 columns):
 #   Column      Non-Null Count  Dtype 
---  ------      --------------  ----- 
 0   reward      10 non-null     int64 
 1   channels    10 non-null     object
 2   difficulty  10 non-null     int64 
 3   duration    10 non-null     int64 
 4   offer_type  10 non-null     object
 5   id          10 non-null     object
dtypes: int64(3), object(3)
memory usage: 608.0+ bytes
```

Figure 4: Missing values in portfolio data

Third, demographic information about the customers will be used to predict the amount spent in the Starbucks mobile app. This data includes the age, income, and gender of a customer as well as the time since a customer is member in the Starbucks app. For about 13% of customers in this data set, information about the age, income and gender is missing. These customers are excluded from further analysis.

	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

Figure 5: Sample of profile data

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 17000 entries, 0 to 16999
Data columns (total 5 columns):
 #   Column            Non-Null Count  Dtype         
---  ------            --------------  -----         
 0   gender            14825 non-null  object        
 1   age               17000 non-null  int64         
 2   id                17000 non-null  object        
 3   became_member_on  17000 non-null  datetime64[ns]
 4   income            14825 non-null  float64       
dtypes: datetime64[ns](1), float64(1), int64(1), object(2)
memory usage: 664.2+ KB
```

Figure 6: Missing values in profile data

Methodology

The original format of the transactional data needs to be changed to make it fit as an input for a machine learning model. More precisely, most algorithms require one line per customer transaction, which contains the order amount as target feature as well as several other features that are used for the prediction of the named target feature. The desired format for the Starbucks capstone challenge is one line per transaction which, next to the amount spent as target feature, includes information in whether an offer was intentionally redeemed for this transaction as well as information about the demographics of the customer and the offer itself. To generate this format, as a first step, the json column is split into three columns, namely amount, which is filled in case of the event being a transaction, reward, which contains the reward value in case of the event column containing “offer completed” and an offer_id column for the events offer received, offer viewed and offer completed. For transaction events, in a second step, the offer_id column is filled with the offer id of the completed offer but only if the customer actually viewed and completed the offer to ensure an intentional use of the received offer. As subtracting the reward from the transactional amount resulted in negative amount values for some of the transaction, it is assumed that the subtraction had already been performed before.

After performing these changes to the format of the transactional data, it is joined with the portfolio data frame. After the join, the data showed some non-qualifying purchases, i.e. customers that received rewards although the difficulty level was not reached by their transaction amount. These 226 purchases (0.073%) are excluded from further analysis. The null values in offer id are filled with “no_special_offer”. As a next step, the demographic information contained in the profile data frame is joined to complete the data set used for modeling.

Several pre-processing steps are performed on the joined data set. First, the missing values in the data set are checked and observations with missing values excluded. Missing values are found in the columns age, income and gender which make up around 10.80% of observations. Second, the target feature amount is checked for outliers. Thereby, observations with an amount of greater 50 are excluded from further analysis (0.5% of observations). A histogram of the target feature can be seen below.

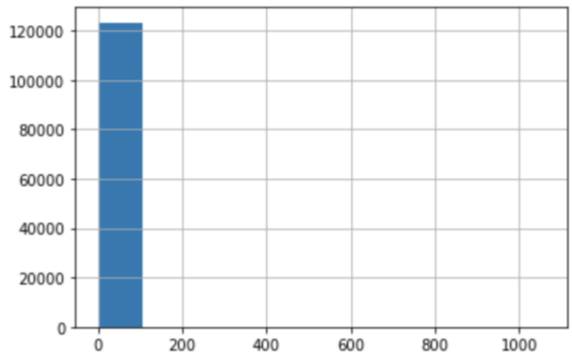

Figure 7: Histogram of monetary amount spent

Third, new features are generated based on the integer columns age and income by binning the values of these columns in four bins based on quantiles. A new feature, years_member which contains the number of years since a customer became a member is generated from the became_member_on column. Fourth, null values in the columns duration, difficulty and potential reward are filled with 0 and null values in channels and offer_type are filled with “no_channel” and “no_offer_type” respectively. Fifth, the data type of several columns is adjusted. Channels, gender and offer_type are cast as category type. For the integer attributes, histograms and a heatmap showing the correlation between the features are created (figure 8 & 9). The heatmap shows that the age and income of a customer are most strongly correlated with the amount spent. Figure 10 shows the relationship between the categorical attributes and the target feature.

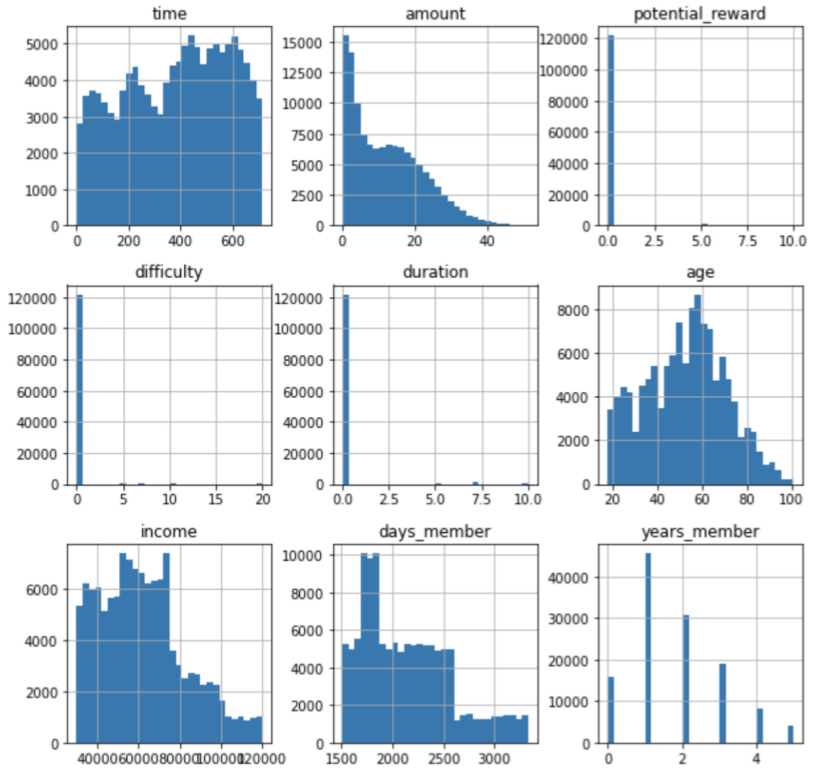

Figure 8: Histograms of integer attributes

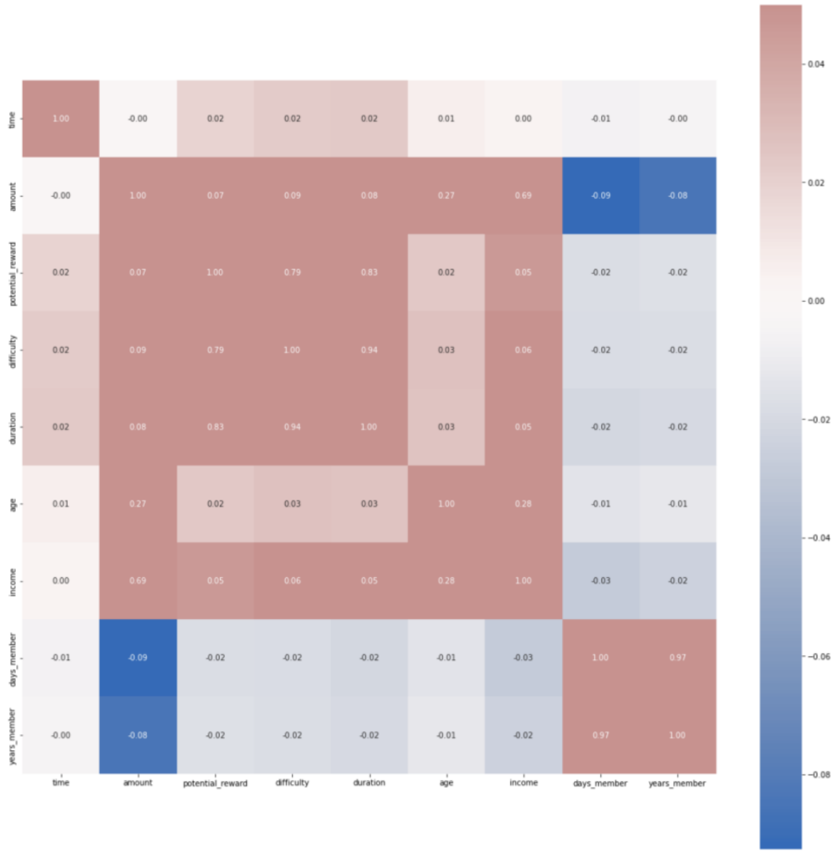

Figure 9: Heatmap showing the correlation between integer features

	channels	amount
0	all_channels	18.414729
1	no_channel	12.221574
2	no_social	17.343828
3	no_web	20.462022
4	no_web_social	24.924901

	offer_type	amount
0	bogo	17.569286
1	discount	20.944187
2	no_offer_type	12.221574

	gender	amount
0	F	15.501126
1	M	10.109713
2	O	13.648836

	age_bucket	amount
0	18-39	7.885347
1	40-54	13.008075
2	55-66	14.400632
3	67+	14.620870

	income_bucket	amount
0	0-46000	5.609383
1	46001-60000	8.948758
2	60001-74000	12.245870
3	74001+	23.539765

Figure 10: Relationship between categorical features and target feature amount

The final selection of features used for the prediction of the transaction amount can be seen in the table below.

Table 1: Final selection of features for prediction

Feature	Description
offer_id	Offer ID (category)
potential_reward	Monetary reward of offer (int64)

channels	Channels through which offer is received (category)
difficulty	Minimum required spend to complete an offer (float64)
duration	time an offer is open in days (float64)
offer_type	Type of offer: bogo, discount or informational (category)
gender	Gender of customer (category)
age_bucket	Age of customer in buckets (category)
income_bucket	Income of customer in buckets (category)
years_member	Years since customer became member (float64)

To make the features fit to be inputted into the XGBoost model, dummy features are created for each of the values of age_bucket, income_bucket, channel, gender, offer_type and offer_id. Finally, all features are standardized by removing the mean and scaling to unit variance.

Implementation & Refinement

For the implementation of a machine learning model, the data is split into a training and test set containing 80% and 20% of the records respectively. This results in a train data frame of 98,505 entries and a test data frame of 24,627 entries with 33 columns. These 33 columns include 32 features and the target feature amount.

As benchmark model, a simple regression model, which takes the same 32 features as the later XGBoost model is used to predict the target feature amount.

For the XGBoost model, the following specifications are used:

```
xgb = sagemaker.estimator.Estimator(
    image_uri=container,
    role=role,
    sagemaker_session =sess,
    instance_count =1,
    instance_type="ml.m4.xlarge",
```

```
    max_run=2000,
    max_wait = 2500,
    use_spot_instances=True
)
```

As this is a regression problem, 'reg:squarederror' is used as objective, which stands for regression with squared loss and "validation:rmse" is used as objective metric. To find the best hyperparameter configurations, hyperparameter tuning is performed with the following hyperparameter ranges:

```
hyperparameter_ranges = {
   "max_depth": IntegerParameter(10, 2000),
    "eta":ContinuousParameter(0,1),
    "gamma":IntegerParameter(4,20),
    "subsample":ContinuousParameter(0,1)
}
```

The following hyperparameter setting are found to produce the best performing model according to RSME:

```
{'_tuning_objective_metric': 'validation:rmse',
 'eta': '0.9454438838542568',
 'gamma': '15',
 'max_depth': '53',
 'num_round': '1500',
 'objective': 'reg:squarederror',
 'subsample': '0.926579500107056'}
```

Results

To evaluate the performance of the proposed model and to compare its performance to the benchmark model, the R2 score as well as the RSME (root-mean-square error), two popular evaluation metrics for regression use cases, are used.

Table 2: RSME and R2 score of both models

Metric	Simple linear regression	XGBoost
R2 of train data	0.508	0.537
R2 of test data	0.510	0.535
RSME of train data	0.492	0.463
RSME of test data	0.491	0.465

As can be seen from table 2, the RSME of the XGBoost model is considerably lower than the one of the linear regression model, while its R2 scores are higher. This indicates that the tuned XGBoost model outperforms the linear regression model in terms of both metrics. Figure 11 shows the features that are most important in the prediction of the monetary amount in the XGBoost model. For the predictions of the XGBoost model the features years_member seems to be particularly important.

Figure 11: Feature importance of XGBoost model of XGBoost model