Udacity Machine Learning Engineer

Kristie Hunter

March 17, 2020

1 Problem Definition

1.1 Project Overview

This Capstone project is for the Starbucks data set. The domain is in the retail industry and is based on the eternal desire of companies to predict consumer behaviour to increase revenue and the customer-base by increasing loyalty. The retail industry is known for needing to stay on top of, if not inventing, trends[1] and that is dependent on human behaviour.

Machine Learning has been used to analyze and predict this behaviour. An example of this use case is in a research paper written by Stubseid and Arandjelović[2] that used 72 features to classify data into a binary decision of not making a purchase (0) and making a purchase (1), which included data such as demographic and price. They used Naive Bayes and Random Forest that predicted the class with 66% and 72% accuracy respectively, which will be explored more deeply in the benchmark section of this report.

The goal of this project is to optimize rewards to get the best customer value.

1.2 Problem Statement

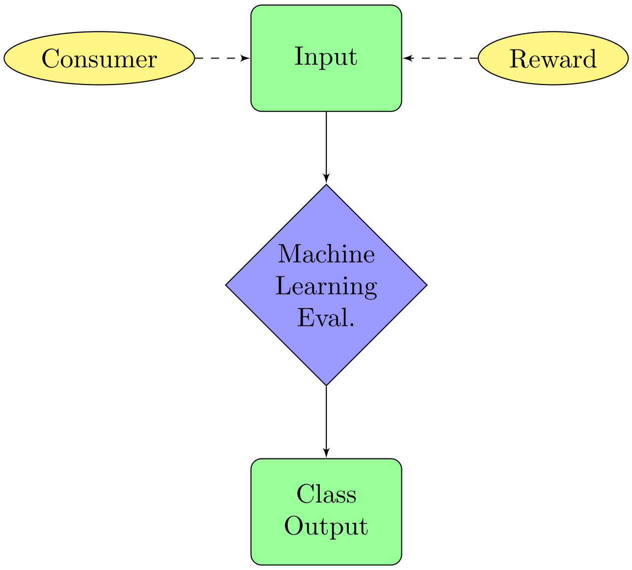
This particular problem statement is to find the most gratifying reward system for Starbucks customers that positively affected their sales as well as which demographics make purchases with no reward incentive. There also may be some rewards that work in some areas and not in others, as well as the fact that some rewards may be unnecessary if the consumer would be making purchases without a reward.

The goal for the machine learning is to use classification to determine what is the most cost effective way to ensure consumer transactions (which may or may not include a reward offer) based on the input of a consumer with a specific set of features covering both the user's personal attributes as well as those of the rewards program they are interacting with.

The following diagram illustrates how a very high level view of the classification model works:

[1] "Retailing Industry." Dictionary of American History, Encyclopedia.com, `https://www.encyclopedia.com/history/dictionaries-thesauruses-pictures-and-press-releases/retailing-industry`.

[2] Stubseid, Saavi, AND Arandjelovic, Ognjen. "Machine Learning Based Prediction of Consumer Purchasing Decisions: The Evidence and Its Significance" AAAI Workshops (2018)



A more in-depth explanation of the input data and class output will follow in another section, but the classes detail whether or not a reward (or lack of a reward) was effective for a particular consumer-reward pairing.

1.3 Metrics

Some of the key evaluators for this project is based on the following pieces of information sourced by visioncritical[3]:

- Customer retention of 5% increases revenue
- The engagement rate of the average rewards program is 50%
- Participating consumers make purchases 90% more often and the average transaction is 60% more

This can be used as the benchmark because a successful rewards program would surpass these values. For programs marked as successful, the participation rate should be above 50%, the customers who participated in that reward were retained as customers and the frequency/amount of their transaction is above the benchmark noted.

The goal of rewards programs is to maximize the Customer Lifetime Value[4], which means that the brand loyalty pays off for the cost of the offer in the long run, which is defined as:

$$\text{CLV} = \text{averageValuePurchase} \times \text{numberPurchasesPerYear} \times \text{lengthRelationship}$$

In addition to the CLV, smile.io defines a number of metrics[5] that would provide data used in judging the values needed to calculate this success:

- Engagement Rate $=$ numberEngaged $\div$ numberTotal
- Redemption Rate $=$ numberRedeemed $\div$ numberIssued

We can also calculate the average transaction value as:

$$sum(\text{transaction}_x) \div \text{numberTransactions}$$

and the frequency as:

$$\text{numberTransaction} \div \text{programLength}$$

[3] "13 Stunning Customer Loyalty Stats, Including the Surprising Ineffectiveness of Loyalty Programs." Vision Critical, `https://www.visioncritical.com/blog/customer-loyalty-stats`.

[4] "Customer Lifetime Value (CLV) Definition - What Is Customer Lifetime Value (CLV)." Shopify, `https://www.shopify.ca/encyclopedia/customer-lifetime-value-clv`.

[5] "The Metrics You Need to Measure Customer Loyalty Online." smile.io, `https://blog.smile.io/measure-customer-loyalty-online`.

Successful programs, as determined by the machine algorithm, should score higher in these values than the non-successful ones.

For the full evaluation, we want to use the highest "quality" algorithm as determined by the Validation Data run through the different algorithms and calculating the F1-Score and the Matthews Correlation Coefficient. Whichever algorithm has the best value will be used to predict the test set.

The F1-Score gives us the balance between how sure the algorithm is that data points belong to the assigned classes and how well the algorithm can classify the class. This means for the lesser represented class, it will weight the accuracy knowing that it can't be classified as well with less data points.

The Matthews Correlation Coefficient gives us a sense of how well the classes are being predicted. If one model gives a better calibre of classification, that model should be used against the testing data, regardless of the standard accuracy metrics.

Based on the predictions in the testing data, we will calculate the Engagement Rate and Redemption Rate that are predominantly classified as successful. From those rewards, we can calculate the CLV for all consumers using the average transaction value and the number of transactions made throughout the length of the program.

Using the benchmark model, we want to determine the following data in order to see if a rewards program is truly successful by averages in the retail industry:

- If the Engagement Rate for successful programs is higher than 50%, the program is better than average
- If consumers who redeemed the successful progams spent above 60% more in the average transaction value than those who didn't, the program is better than average
- If consumers had an average transaction over the length of the program that was at least 5% greater than the total average, the consumer can be qualified as retained

2 Analysis

2.1 Data Exploration

The three types of data that were included in the problem were Profile (demographic information of the consumers), Portfolio (reward program information) and Transcript (complete transaction list). Samples of the data are as follows:

Profile:

Gender	Age	ID	Became Member On	Income
F	55	0610b486422d4921ae7d2bf64640c50b	20170715	112000.0
F	75	78afa995795e4d85b5d9ceeca43f5fef	20170509	100000.0
M	68	e2127556f4f64592b11af22de27a7932	20180426	70000.0
M	65	389bc3fa690240e798340f5a15918d5c	20180209	53000.0
M	58	2eeac8d8feae4a8cad5a6af0499a211d	20171111	51000.0

Portfolio:

Reward	Channels	Difficulty	Duration	Offer Type	ID
10	[email, mobile, social]	10	7.0	bogo	ae264e...
10	[web, email, mobile, social]	10	5.0	bogo	4d5c57...
0	[web, email, mobile]	0	4.0	informational	3f207d...
5	[web, email, mobile]	5	7.0	bogo	9b98b8...
5	[web, email]	20	10.0	discount	0b1e15...

Difficulty is defined as the minimum amount required to spend in order to complete the reward.

Transcript:

Person	Event	Value	Time
78afa9...	offer received	'offer id': '9b98b8c7a33c4b65b9aebfe6a799e6d9'	0
a03223...	offer received	'offer id': '0b1e1539f2cc45b7b9fa7c272da2e1d7'	0
e21275...	offer received	'offer id': '2906b810c7d4411798c6938adc9daaa5'	0
8ec6ce...	offer received	'offer id': 'fafdcd668e3743c1bb461111dcafc2a4'	0
68617c...	offer received	'offer id': '4d5c57ea9a6940dd891ad53e9dbe8da0'	0

Time is defined as the time since the start of the program.
Value contains the monetary transactions of purchases if any exist.

The classes were initially defined based on three pieces of information which included whether or not the reward was offered, whether or not the reward was completed and the value of transactions made in the window that the reward was valid for. This led to 7 distinct classes (which can be seen in the table below). However, through preprocessing and working through the data, it became apparent that there were additional determining factors of a successful offer and subsequent analytics.

For example, some offers were received by consumers more than once, which could influence whether or not the transaction was completed. In addition, if an offer was never received, the success or failure of the reward is unknown and does not provide useful data in predicting class outcomes.

One metric that helps distinguish the remaining data of offers that were received lies in the value of transactions in a given window which is calculated with the following:

$$\text{margin} = \text{reward} - \text{difficulty}$$
$$\text{profit} = \text{transactions} > \text{margin}$$

If the transactions were less than or equal to the margin, that becomes an indicator for whether or not the reward was "successful enough" (as determined by the analytics detailed later in this report). The margin for the BOGO rewards is 0 because the amount required to spend is equal to the reward. The margin on the discount rewards is the difference between the total required to redeem and the reward. The margin on information rewards is any value greater than 0 (or 0 if no required amount is specified). The goal is to have the consumer spend more than they are being asked to in order to generate a successful consumer relationship or, conversely, shows that the consumer may need a larger reward in order to spend the required amount.

One important piece of data in this exploration is that the window for a transaction start when the offer is received and can either end when the offer expires or when the offer is completed. If the offer is completed and transactions are made after, they are not taken into consideration about the transaction value of the offer itself, but that data will be calculated for the Customer Lifetime Value.

After going through and processing the data, there are two class values:

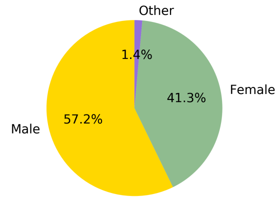
	Completed	Successful
Class 0	No	No
Class 1	Yes	Yes

2.2 Exploratory Visualization

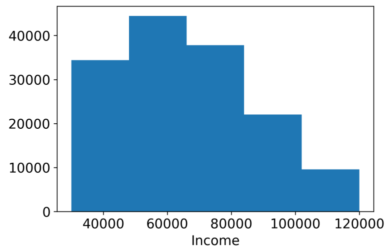
2.2.1 Profiles

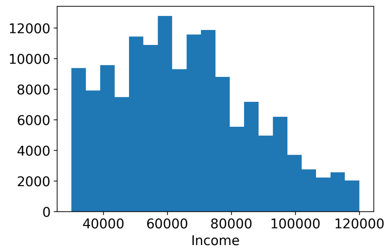
Looking at the first group of data, which is our consumer profiles, we can extract the following visualizations.

Gender is predominantly Male, with approximately 16% less being Female and a very small representation of Non-Binary gender.

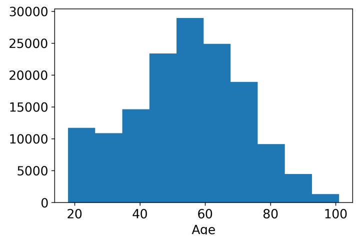


Income looked at in a histogram with 5 buckets and 20 buckets places a large portion in the demographic in the middle class income range.





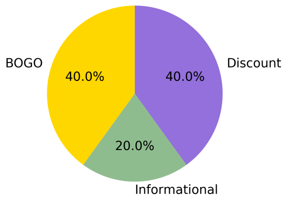
Age looked at in a histogram with 10 buckets centers the demographic largely around middle-aged consumers.



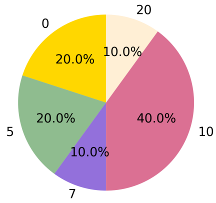
2.2.2 Portfolio

For the rewards themselves, we can see the following from the visualizations:

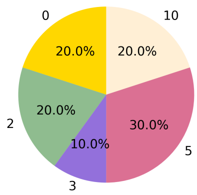
Offer Types were mostly "discount" and "bogo".



Offer Difficulty had a range of 0 to 20; the most common difficulty was 10 but 50% were below 10.



Offer Reward had a range of 0 to 10 and were mostly evenly distributed.



2.2.3 Transactions

2.3 Algorithms and Techniques

The algorithms that were tested in order to find the best predictive model are detailed in the following sections:

2.3.1 Logistic Regression

Logistic Regression can perform multi-class classification by using a one-versus-rest approach where data either belongs to a specific class that the algorithm is looking at (one of n classifiers) or it belongs to something else[6]. By using this approach, it sets a good baseline for collecting data into the classes; since the algorithm has to give a "yes" or "no" answer to the function about whether or not it belongs to a certain class, it can demonstrate if there is class overlap if there are fuzzy predictions.

The parameters that will be modified in the algorithm to find the best model using the sklearn package[7] are:

- Penalty
- Stopping Criteria Tolerance
- Solver Type
- Maximum Iterations

[6] Murphy, Kevin P. Machine Learning: a Probabilistic Perspective. MIT Press, 2013. p. 509

[7] "Sklearn.linear_model.LogisticRegression." Scikit, `scikit-learn.org/stable/modules/generated/sklearn.linear_model.LogisticRegression.html`.

2.3.2 Random Forest

Decision Trees are an excellent way of separating data by common features. Since a lot of the data is numerical, and a number of features are boolean, the splitting points in the decision tree are easier. The downside to this algorithm is that there are a high number of permutations if the data is similar. Each node separates the target values based on the "split criterion as well as the threshold parameter"[8]. If the distinction between classes is ambiguous at any given split, the classification can be skewed.

Random Forest is also one of the classifiers used in the benchmark for this problem, and the comparison will be valuable.

The parameters that will be modified in the algorithm to find the best model using the sklearn package[9] are:

- Number of Trees in Forest
- Maximum Depth
- Minimum Number of Samples for Split
- Minimum Number of Samples for Leaf
- Number of Features
- Bootstrap

2.3.3 Multi-Layer Perceptron

Multi-layer neural networks are designed to handle multi-class classification because it is "m separate learning problems for an m-output problem"[10]. This means that we can get weighted guesses on each class' probability given the hidden layers in the neural network. The downside to this algorithm is that it is computationally expensive and not usually scalable for large datasets.

The parameters that can be modified in the algorithm to find the best model using the sklearn package[11]:

- Activation Function
- Solver Algorithm for Weight Optimization
- Penalty Parameter
- Learning Rate for Updates

2.3.4 Gaussian Naive Bayes

Gaussian Naive Bayes for classification allows an algorithm to make an assumption that the features are independent of one another and there is a probability correlation between a feature and the class directly[12]. For this particular dataset, having features such as human demographic means that the features likely are dependent to some extent, but because the end result of the prediction is a product of those probabilities, this algorithm is worth testing. Naive Bayes is the second classifier used in the benchmark for this problem, allowing us to do a direct comparison.

There were no variable parameters tested for this algorithm; the default values were used.

2.3.5 K-Nearest Neighbours

K-Nearest Neighbours separates the data based on similarity to "nearby" data points that works best with lower dimension data[13]. The maximum number of the features in this dataset would be 13 and

[8] Bishop, Christopher M. Pattern Recognition and Machine Learning. Springer New York, 2016. p. 664

[9] "Sklearn.ensemble.RandomForestClassifier." Scikit, `https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html`.

[10] Russell, Stuart Jonathan., and Peter Norvig. Artificial Intelligence: a Modern Approach; Prentice Hall, 2003. p. 746

[11] "Sklearn.neural_network.MLPClassifier." Scikit, `https://scikit-learn.org/stable/modules/generated/sklearn.neural_network.MLPClassifier.html`.

[12] Bishop, Christopher M. Pattern Recognition and Machine Learning. Springer New York, 2016. p. 380

[13] Murphy, Kevin P. Machine Learning: a Probabilistic Perspective. MIT Press, 2013. p. 18

presumably would have substantial correlation between demographic information and offer portfolio information that would lead to the class assignment.

The parameters that can be modified in the algorithm to find the best model using the sklearn package[14] are:

- Number of Neighbours
- Weight Function
- Distance Algorithm
- Power

2.4 Benchmark

The benchmark for this program, as mentioned in the Project Overview is the work done by Stubseid and Arandjelović[15] where they used Naive Bayes and Random Forest to predict whether or not a consumer made a successful purchase with 66% (Naive Bayes) and 72% (Random Forest) accuracy.

In addition to Accuracy, there are some other values that would be more beneficial to the success of something like a rewards program.

The following pieces of information sourced by visioncritical[16] give us a sense of what a long-term successful rewards program would look like:

- Customer retention of 5% increases revenue
- The engagement rate of the average rewards program is 50%
- Participating consumers make purchases 90% more often and the average transaction is 60% more

For program(s) marked as successful, the participations rate should be above 50%, the customers who participated in that reward were retained as customers, and the frequency and amount of their transaction is above the benchmark noted.

The other metric evaluated is the Matthews Correlation Coefficient, which gives us a sense of how well the classes are being predicted, which is not mentioned in the aforementioned benchmark studies.

3 Methodology

3.1 Data Preprocessing

The Data Preprocessing took the following 5 steps:

1. Remove outlier data from the Profile data
2. Generate base dataframes for the consumers
3. Get all transactions made by each consumer
4. For offering of a reward for the individual consumers, track the relevant transactions
5. Encode all column values
6. Assign class value

The Profile data had data points with consistently *null* values for age, income and gender, so those data points were removed to prevent skewing the demographic features in the model. This took the Profile data from 17,000 items to 12,650.

For every transaction item, a dictionary was generated for each consumer ID correlated to the

[14] "Sklearn.cluster.KNeighborsClassifier." Scikit, `https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KNeighborsClassifier.html`.

[15] Stubseid, Saavi, AND Arandjelovic, Ognjen. "Machine Learning Based Prediction of Consumer Purchasing Decisions: The Evidence and Its Significance" AAAI Workshops (2018)

[16] "13 Stunning Customer Loyalty Stats, Including the Surprising Ineffectiveness of Loyalty Programs." Vision Critical, `https://www.visioncritical.com/blog/customer-loyalty-stats`.

transaction–with the offer IDs as the keys and a catch-all for non-offer related events. For every instance that an offer was offered, a data point was added to and updated in the data frame.

As noted earlier, some offers were made more than once to the same consumer, so the total number of data points is every instance of any offer ever issued to any consumer, which is 66,501.

For training purposes, any non-numerical values (Boolean or String) were encoded in the data frame for easier processing in the algorithm models.

The classes were assigned based on the above mentioned criteria in the Data Exploration section, which created a distribution of the following:

	Class 1	Class 2	Total
Total	34501	32000	66501
Training	20700	19200	39900
Validation	6900	6400	13300
Testing	6901	6400	13301

3.2 Implementation

The implementation followed these steps, which will be covered in more detail in the following sections:

1. Loading the pre-processed data into data frame
2. Separate Training, Validation and Testing subsets
3. Separate Target from Features
4. Use GridSearchCV to train the models on training data with variable parameters
5. Compare the best models against the validation data and the benchmark accuracy
6. Test the best models on the validation set

After the last step in that list, some refinement took place, which will be discussed in the next section.

The splitting of the data, as noted in the table above, consisted of using randomized shuffle splitting to select 60% of the total data for Training, 20% for Validation and 20% for Testing. The target for each was set as the column tracking whether or not the offer was completed.

Using the five algorithms mentioned in the Algorithms and Techniques section, the program ran the different combinations of parameters and generated metrics based on the values/equations mentioned in the Benchmark section using 10-fold Cross-Validation.

Training Time:

Model	Time (mins)
Logistic Regression	24.1
Decision Tree	14.7
Neural Network	312.9
Naive Bayes	0.01
K-Nearest Neighbours	2.4

3.3 Refinement

The best models determined by GridSearchCV against the training data using all features were initially as follows:

Model	Accuracy	Matthews
Logistic Regression	68.6%	0.370
Decision Tree	77.2%	0.558
Neural Network	53.3%	0.0
Naive Bayes	72.6%	0.452
K-Nearest Neighbours	69.7%	0.395

From the above results we can see that the Decision Tree has the highest accuracy as well as the best quality of predictions using the Matthews Correlation Coefficient.

To refine the algorithms to use for validation, we want to take into account the Accuracy on the training data, the Matthews Correlation Coefficient on the training data and the time it took to train the model (for scalability in future use). The total "score" is the sum of the rankings, where 3 would be the best score (having a ranking of 1 in each category) and 15 would be the worst.

Model	Accuracy	Matthews	Time	Score	Overall Rank
Decision Tree	1	1	3	5	1
Naive Bayes	2	2	1	5	1
K-Nearest Neighbours	3	3	2	8	3
Logistic Regression	4	4	4	12	4
Neural Network	5	5	5	15	5

From this assessment we can see that the Neural Network has caveats across all areas that makes it ineffective to run against the validation set. The Neural Network had the lowest accuracy, was the slowest and had the lowest Matthews Correlation Coefficient (at 0, which is random guessing).

For the remaining algorithms, the best parameters selected using GridSearchCV were as follows:

Decision Tree:

- Criterion = entropy
- Splitter = best
- Max Depth = 8
- Min Samples Leaf = 2
- Min Samples Split = 2
- Max Features = None

K-NN:

- Num Neighbours = 8
- Weights = distance
- Algorithm = kd_tree
- P (power) = 1

Logistic Regression:

- Penalty = l1
- C = 0.1
- Fit Intercept = False
- Solver = liblinear
- Max Iterations = 500

Naive Bayes had no modified parameters.

After adding some values to each of the parameters to test if they boosted the training scores, none of the parameters changed. Running the top algorithms against the validation set with the given models, we get the following results:

Model	Accuracy	Matthews
Logistic Regression	68.7%	0.375
Decision Tree	76.9%	0.539
Naive Bayes	72.6%	0.454
K-Nearest Neighbours	70.3%	0.406

These values are in line with the predictions from the training data, which means there are no overfitting issues and the model can be used for the test data.

4 Results

4.1 Model Evaluation and Validation

Using the best models after the refinement, testing against the test data resulted in the following accuracy results:

Model	Accuracy
Decision Tree	77.1%
Naive Bayes	72.2%
K-Nearest Neighbours	70.1%
Logistic Regression	68.4%

Compared to the Benchmark of 66% Accuracy using Naive Bayes and 72% Accuracy using Random Forest, the Decision Tree performs 5% better than its counterpart and the Naive Bayes performs 6% better than its counterpart. The Decision Tree was the best performing model overall.

Using the best scored predictions from the Decision Tree model, we use the predicted classes saved from the evaluation to determine the remaining analytics.

4.2 Justification & Analytics

We have three Benchmark metrics that we want to calculate in addition to the Accuracy to ensure that these programs are effective, not just interacted with. We also want to evaluate the Customer Lifetime Value.

4.2.1 Premise 1: Engagement Rate of 50%

Given the predictions, we want to confirm that any reward marked as successful has an Engagement Rate of at least 50%. The Engagement Rate is calculated as follows:

$$\text{Engagement Rate} = \frac{\text{numViewed}}{\text{numTotal}} \tag{1}$$

For each reward, the Engagement Rates are as follows:

ID	OfferID	Number of Views	Number of Offers	Engagement Rate
1	ae26....ddfd	650	760	85.5%
2	4d5c....e8da0	673	693	97.1%
3	3f20....8bed	211	363	58.1%
4	9b98....e6d9	391	687	56.9%
5	0b1e....e1d7	276	719	38.4%
6	2298....b8c2	700	728	96.1%
7	fafd....c2a4	745	771	96.6%
8	5a8bc....9837	325	380	85.5%
9	f194....e20d	705	731	96.4%
10	2906....aaa5	453	760	59.6%

This means that 6 of the 10 rewards had a high engagement rate over 85% and 9 out of 10 had an engagement rate above the benchmark of 50%.

4.2.2 Premise 2: Participants Increase Spending

Given the predictions, we want to confirm that any reward marked as successful has transaction totals inside the offer window are 60% more than their unsuccessful counterparts.

For the total value, we can calculate:

ID	OfferID	Successful	Total Txn	Duration	Avg Value	Difference
1	ae26...ddfd	Yes	12,965.00	62,708	0.21	+91%
		No	7,763.16	73,488	0.11	
2	4d5c...e8da0	Yes	11,605.62	44,184	0.26	+136%
		No	5,911.24	51,384	0.11	
3	3f20...8bed	Yes	8,556.75	34,848	0.25	+67%
		No	11,819.22	80,424	0.15	
4	9b98...e6d9	Yes	9,604.63	55,248	0.17	+112%
		No	5,077.50	66,264	0.08	
5	0b1e...e1d7	Yes	18,647.10	98,436	0.19	+137%
		No	9,198.02	109,548	0.08	
6	2298...b8c2	Yes	13,338.90	47,538	0.28	+180%
		No	5,461.03	56,874	0.1	
7	fafd...c2a4	Yes	15,579.31	59,256	0.26	+225%
		No	5533.93	71,418	0.08	
8	5a8bc...9837	Yes	6,532.44	27,360	0.24	+43%
		No	11,292.20	59,904	0.19	
9	f194...e20d	Yes	14,040.38	43,038	0.33	+154%
		No	4,789.49	35,928	0.13	
10	2906...aaa5	Yes	13,983.86	66,456	0.21	+162%
		No	4,844.02	58,170	0.08	

This means that with one exception (Offer 8), the successful rewards incited at least 60% more spending than the unsuccessful counterparts, which was the benchmark for a truly successful program. Something to keep in mind with this data is that this is the predicted values on only the test data, which represents 20% of the overall data.

4.2.3 Premise 3: Customer Retention of 5%

Given the predictions, we want to confirm that consumers who had predominantly successful rewards (of the predicted values on the test data) made transactions at least 5% higher than the average transaction value for the duration of the rewards program. For comparison, the average transaction value (for the entire length of the program) of all the data is $120.06.

Success Bracket	Count	Avg Txn Value	Difference
0 - 25%	3902	67.28	-78%
26 - 50%	1126	123.99	+3%
51 - 75%	201	161.83	+35%
76 - 100%	3746	171.61	+43%

This means that successful rewards have a sway on the amount being spent by the average consumer. With a total average value of $120, the 0 - 25% bracket falling below that threshold is unsurprising. With a benchmark of 5% to be considered successful, rewards that are 50%+ successful lead to a substantial jump in transaction value (from $124 to $162). Also interesting to note is that the difference between the 3rd and 4th success brackets is not as substantial as the 2nd to 3rd success bracket.

4.2.4 Customer Lifetime Value

To determine the Customer Lifetime Value, we can use the data collected in the previous premise calculations with the following equations:

$$\text{CLV} = \text{avgValuePurchase} \times \text{numPurchasesPerYear} \times \text{lengthRelationship} \tag{2}$$
$$\text{AVP} = sum(\text{transaction}_x) \div \text{numTransactions} \tag{3}$$
$$\text{PPY} = \text{numTransaction} \div \text{programLength} \tag{4}$$
$$\text{LR} = \text{newestDate} - \text{memberSince} \tag{5}$$

Note that since we only have the transactions over the length of the program and not over an entire year, the data is slightly skewed.
The total Average Customer Lifetime Value of all the data is $2,344.34. Using the same brackets as the Customer Retention, we get the following values:

Success Bracket	Average CLV	Difference
0 - 25%	1,247.00	-88%%
26 - 50%	2,376.82	+1%
51 - 75%	3,257,.12	+39%
76 - 100%	3,428.66	+46%

This demonstrates that using successful rewards programs improves the Customer Lifetime Value, and most notably that the more successful the rewards, the greater the increase in the CLV.

5 Conclusion

Using the Decision Tree model, there is 77% Accuracy that any given rewards program can be predicted to be successful or not for any given customer demographic. With this data we can expect to see an increase in consumer spending, increased customer retention and increased Customer Lifetime Value for the consumers that are presented with successful rewards.