Machine Learning Engineer ND

Capstone Project

I. Definition

1. Project Overview

   The Starbucks Capstone Challenge is an classic example of a sales problem where we want to make an offer to the customers that they can't refuse. We can solve the problem by looking at the past behavior of the customer and predict what items or offers the customer will be most interested in. By surfacing the right offers to the customer we can generate more sales. We have with us the information of past transactions of customers and related offers which were made to the customers.

   1. Research and Citation

      - https://www.sciencedirect.com/science/article/abs/pii/S0957417412006148
      - https://martech.org/machine-learning-for-next-best-offers/
      - https://towardsdatascience.com/implementing-a-profitable-promotional-strategy-for-starbucks-with-machine-learning-part-1-2f25ec9ae00c

   2. Dataset

      We had three datasets:

      portfolio.json
      : containing offer ids and meta data about each offer (duration, type, etc.)

      profile.json
      : demographic data for each customer

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

      1. Portfolio.json

         - id (string) - offer id
         - offer_type (string) - type of offer ie BOGO, discount, informational
         - difficulty (int) - minimum required spend to complete an offer
         - reward (int) - reward given for completing an offer
         - duration (int) - time for offer to be open, in days
         - channels (list of strings)

      2. Profile.json

         - age (int) - age of the customer
         - became_member_on (int) - date when customer created an app account
         - gender (str) - gender of the customer (note some entries contain 'O' for other rather than M or F)
         - id (str) - customer id
         - income (float) - customer's income

      3. transcript.json

         - event (str) - record description (ie transaction, offer received, offer viewed, etc.)
         - person (str) - customer id
         - time (int) - time in hours since start of test. The data begins at time t=0
         - value - (dict of strings) - either an offer id or transaction amount depending on the record

2. Problem Statement

   We needed to device a solution which will tell us which customer should get which type of offer (discount, bogo, informational) or should the customer be given any offer at all. The goals of the system are as follow

   ```
   ~Predict, based on customer past transcript, what offer will be most suitable for him
   ```

   To achieve this goal we performed following steps.

   1. Explore and Process the Data

      We explored the three dataset and tried to find the relationships between each of them. At the end of this steps we had achieved following outcomes

      - Processed all the customer data
      - Linked all the transaction data with the customer
      - Link all the offer data with the customers

   2. Clean the Data

      Since the dataset had missing feature values we had to find a solution for them before we could train the model on the data. Following feature had missing values

      - age
      - gender
      - income

   3. Train a Model

      - Finally we used the cleaned data to train a model and predict possible offer type for a particular customer

3. Metrics

   We used multiple metrics for the model we created at the different phases of the project

   - In order to create a model that predict the gender of the customer we used accuracy score and f{β} score = 0.5. The reason behind using these to score is to ensure we cover precision and accuracy.
   - For the models to predict Age and Income we used Root Mean Squared Score (RMSE) for bench-marking and then R2 Score for optimization. The reason behind using RMSE is to give more weight-age to bigger errors.
   - For Clustering the Customers data we will use the Sum Squared Error (SSE) to find the optimum cluster count.

II. Analysis

1. Data Exploration

   We received three different data set from starbucks in the form json files

   1. Customer Data (profile.json)

      The customer data dataset is the main focus of our project. The dataset has 6 features of which one is ID column so that leaves us with 5 features to run our model against. This is not enough but with the help of ID column and transcript dataset we will generate some more columns at a later stage.

      Table 1: profile.json columns

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

      1. Customer data had following statistics

         - Total Customers were 170000
         - Total missing data row 2175
         - Total 2175 columns have age = 118
         - Columns with missing Data
           - gender
           - income

   2. Events Data (transcript.json)

      Event data contained all the offer related event and transactions of the customers over a period of time. There are 4 features in this dataset, but the value column contains a json object we parsed it to add few more feature to the dataset

      Table 2: transcript original

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

      Table 3: transcript after exploding value column

person event	value	time	offer id	amount	offer_id
	78afa995795e4d85b5d9ceeca43f5fef	"offer received"	{'offer id': '9b98b8c7a33c4b65b9aebfe6a799e6d9'}	0	9b98b8c7a33c4b65b9aebfe6a799e6d
	a03223e636434f42ac4c3df47e8bac43	"offer received"	{'offer id': '0b1e1539f2cc45b7b9fa7c272da2e1d7'}	0	0b1e1539f2cc45b7b9fa7c272da2e1d
	e2127556f4f64592b11af22de27a7932	"offer received"	{'offer id': '2906b810c7d4411798c6938adc9daaa5'}	0	2906b810c7d4411798c6938adc9daaa
	8ec6ce2a7e7949b1bf142def7d0e0586	"offer received"	{'offer id': 'fafdcd668e3743c1bb461111dcafc2a4'}	0	fafdcd668e3743c1bb461111dcafc2a4
	68617ca6246f4fbc85e91a2a49552598	"offer received"	{'offer id': '4d5c57ea9a6940dd891ad53e9dbe8da0'}	0	4d5c57ea9a6940dd891ad53e9dbe8d

      1. Statistics

         - Total rows 306534
         - Total features 6

   3. Offer Data (portfolio.json)

reward	channels	difficulty	duration	offer_type	id
10	[email, mobile, social]	10	7	bogo	ae264e3637204a6fb9bb56bc8210ddfd
10	[web, email, mobile, social]	10	5	bogo	4d5c57ea9a6940dd891ad53e9dbe8da0
0	[web, email, mobile]	0	4	informational	3f207df678b143eea3cee63160fa8bed
5	[web, email, mobile]	5	7	bogo	9b98b8c7a33c4b65b9aebfe6a799e6d9
5	[web, email]	20	10	discount	0b1e1539f2cc45b7b9fa7c272da2e1d7
3	[web, email, mobile, social]	7	7	discount	2298d6c36e964ae4a3e7e9706d1fb8c2

2	[web, email, mobile, social]	10	10	discount	fafdcd668e3743c1bb461111dcafc2a4
0	[email, mobile, social]	0	3	informational	5a8bc65990b245e5a138643cd4eb9837
5	[web, email, mobile, social]	5	5	bogo	f19421c1d4aa40978ebb69ca19b0e20d
2	[web, email, mobile]	10	7	discount	2906b810c7d4411798c6938adc9daaa5

Offer Data set contains all the details of offer made to customers, it has 5 features column

2. Data Visualization
    1. Gender

        The gender distribution is skewed large number customer being male. We also had considerable portion of customer with missing gender class. In order to proceed with modeling we need to fill in the gap

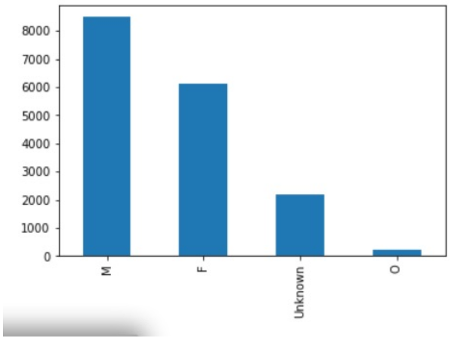


        Figure 1: Gender distribution

        1. Preprocessing
            - We applied Hot Encoding to gender table
    2. Age and Income

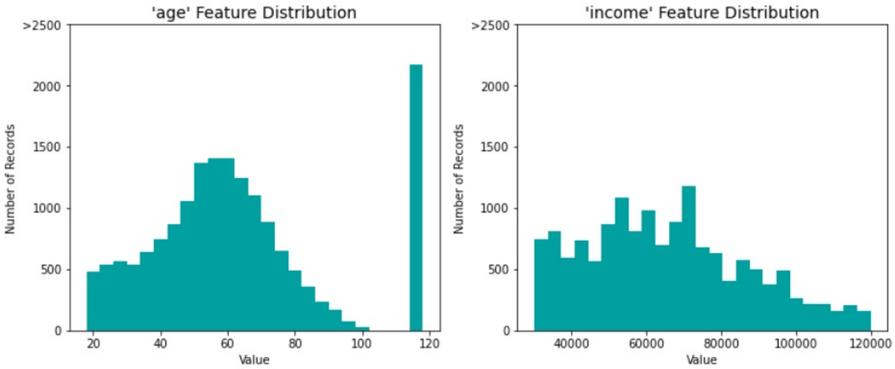


        Figure 2: Age and Income Distribution

        1. Age distribution

            Age distribution show a large number of customer aged at 118 which is highly unlikely, this probably means this age value is set by default for customers whose age is unknown

            1. Preprocessing
                - We replaced all the 118 value with NaN
        2. Income Distribution

            Income Distribution doesn't show the missing value as they are marked as NaN but we can see that that high earners are not so much fan of starbucks may be they have better place for coffee
3. Algorithm and Technique

    We used following technique to pre-process and transform the data

    1. Encoding
        - We encoded categorical data
        - We encoded large non value column like ids
        - Created dummy encoded column for list value columns
    2. Parsing

        Some column contained dictionary of value, so we parsed those column to generate corresponding feature column for modeling
    3. Joining

Since the data was divide into three datasets we used the ids to join the datasets together with the customer dataset

4. Pivoting and Aggregation

   we manipulated the combined data set using pivoting and group aggregation to generate some more useful features from customer offer history

5. Modeling the missing value

   We decided to use modeling to predict the missing values. Since 12% of the dataset had missing value and dropping them will impact the efficiency of our prediction in the end

6. Customer clustering

   With the cleaned dataset we will use clustering to identify possible customer group which can then be sent certain type of offers given on a set of heuristics. The reason we used clustering instead of classification is due to the fact that our data is not labeled by default and due to that we either had an option to label the data ourselves based on some heuristics or instead we could generate clusters using clustering and use some heuristics to label those clusters. We used the elbow method the find the optimum number of clusters

4. Benchmark

   1. Missing values

      We used multiple modeling algorithm to benchmark best fitting algorithms

      1. Gender

         1. Metrics Used

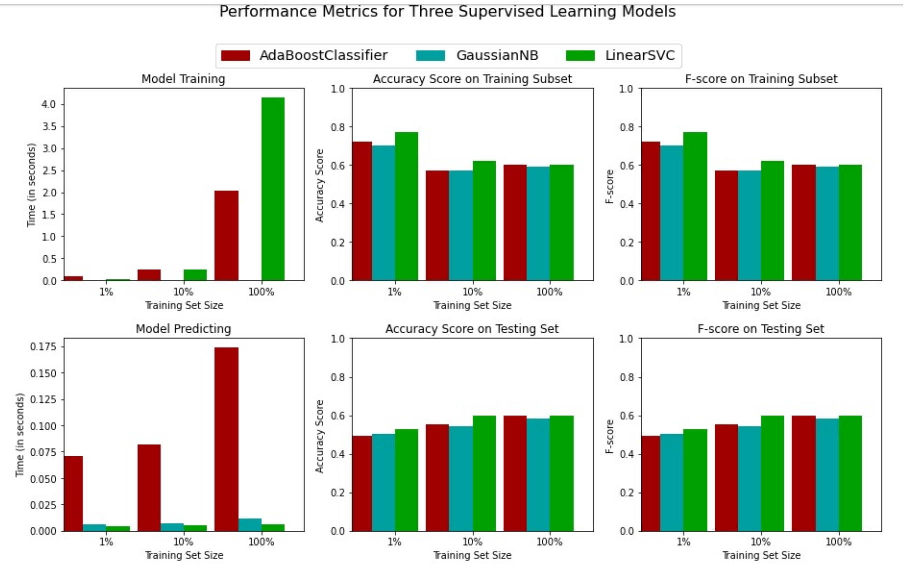
            We used the accuracy score, f{β} score where β is 0.5 and training time to compare the benchmarking models

         2. Benchmarking

            For Benchmarking the model we used following the models

            - AdaBoost Classifier
            - Guassion Naive Bayes
            - Linear SVC

            All three models were trained on 1%, 10% and 100% of dataset to see their benchmark values. The result of benchmarking are shown below



            - Gender Column was Encoded
            - profile_id Column was Encoded (To make it simpler)
            - offer_id column was Encoded (To make it simpler)
            - became_member_on column was broken down into year, month and day
            - channels column in portfolio dataset was broken down into for channel specific dummy columns
            - value column in transcript dataset was parsed to generate new columns of offer_id, amount and reward

      2. Income

         1. Metrics Used

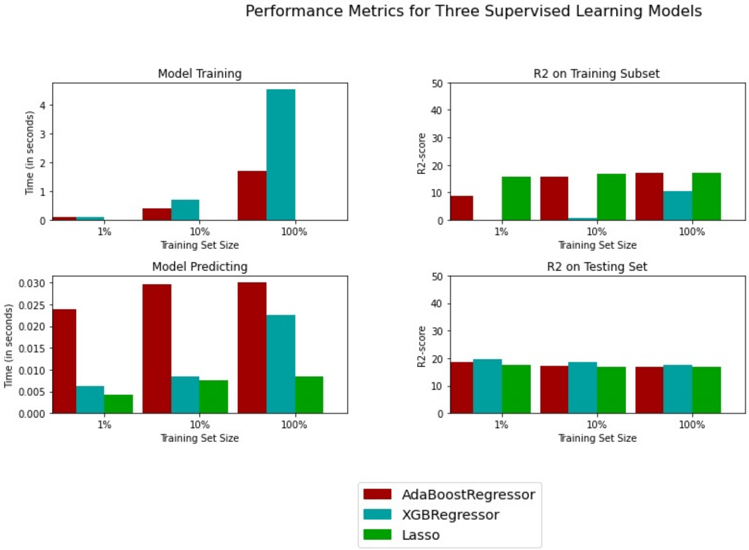
            Since these were regression problem we use following metrics

            - RMSE

         2. Benchmarking

            The algorithm used in benchmarking were

- AdaBoost Regressor
- XGB Regressor
- Lasso Bench Marking result is as followed



From the benchmark it turned out tha AdaBoost Regressor is the best option marginally.

3. Age

    1. Metrics Used

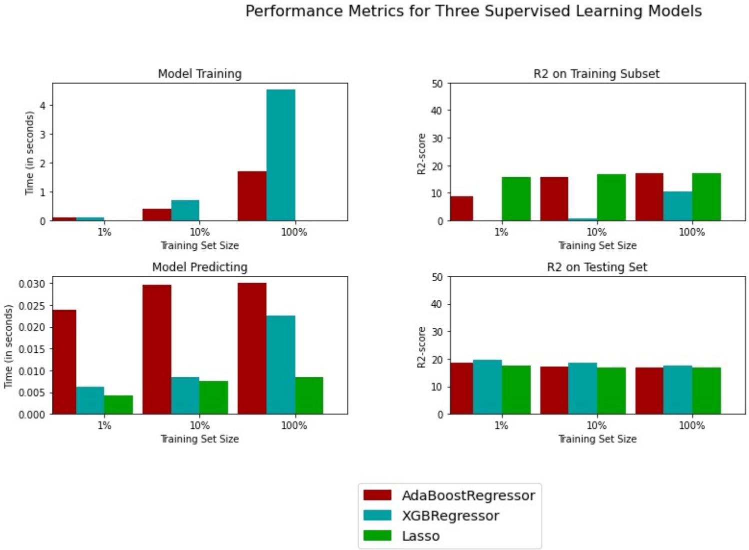
        Since these were regression problem we use following metrics

        - RMSE

    2. Benchmarking

        The algorithm used in benchmarking were

        - AdaBoost Regressor
        - XGB Regressor
        - Lasso Bench Marking result is as followed



From the benchmark it turned out tha AdaBoost Regressor is the best option marginally.

III. Methodology

1. Data Preprocessing

    1. Derived Features

        In order to increase the number features we generated from some features from transcript by performing following steps

        1. Get all the offer received, offers viewed and offer completed event in separate database and rename the time column to received_on,

viewed_on and completed_on

2. Join the three new data set on offer id
3. Use the following heuristics to flush out the wrong data produced due to join
    - offer viewed on should be greater than received on and less than completed_on or null
    - completed_on should be less than offer expiry time
4. We used LabelEncoder to encode both person_id and offer_id
5. Join the joined transcript dataset with customer data and portfolio data on p_id(customer id) and (o_id) portfolio id
6. Now pivot the combined table on p_id to get aggregated values

1. New Features

Following is the list of 33 features we derived:

total_offer_received
Total number of offer received of any type by the customer

total_offer_viewed
Total number of offer viewed by the customer

total_offer_completed
Total number of offer completed of any type completed by the customer

total_email_offers
Total number of offers sent through email

total_web_offers
Total number of offers sent through web

total_social_offers
Total number of offers sent through social platform

total_mobile_offers
Total number of offers set through mobile

total_reward_earned
Total number of reward earned by completing the offers

total_spent_on_offer
Total amount of money spent at starbucks

average_offer_difficulty
Average difficulty of received offers

average_offer_duratio
Average expiry period of received offers

total_offer_completed_bogo
Total number of offer received of bogo type by the customer

total_offer_completed_discount
Total number of offer received of discount type by the customer

total_offer_completed_informational
Total number of offer received of informational type by the customer

average_offer_difficulty_bogo
Average difficulty of received bogo offers

average_offer_difficulty_discount
Average difficulty of received discount offers

average_offer_difficulty_informational
Average difficulty of informational offers

average_offer_duration_bogo
Average expiry period of received bogo offers

average_offer_duration_discount
Average expiry period of received discount offers

average_offer_duration_informational
Average expiry period of received informational offers

total_email_offers_bogo
Total number of bogo offers sent through email

total_email_offers_discount
Total number of discount offers sent through email

total_email_offers_informational
Total number of informational offers sent through

total_mobile_offers_bogo
Total number of bogo sent through mobile

total_mobile_offers_discount
Total number of discount sent through mobile

total_mobile_offers_informational
Total number of informational sent through mobile

total_offer_received_bogo
Total number of bogo offers recieved by customer

total_offer_received_discount
Total number of discount offers recieved by customer

total_offer_received_informational
Total number of informational offers recieved by customer

total_reward_earned_bogo
Total reward earned on bogo offers

total_reward_earned_discount
Total reward earned on discount offers

total_reward_earned_informational
Total reward earned on informational offers

total_social_offers_bogo
Total number of bogo offers sent through social channels

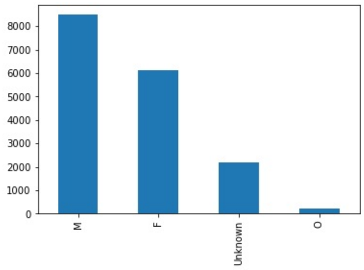
total_social_offers_discount
Total number of discount offers sent through social channels

total_social_offers_informational
    Total amount spent on informational offers sent through social channels
total_spent_on_offer_bogo
    Total amount spent on bogo offers
total_spent_on_offer_discount
    Total amount spent on discount offers
total_spent_on_offer_informational
    Total number of informational offers
total_offer_viewed_bogo
    Total number of bogo offers viewed by customer
total_offer_viewed_discount
    Total number of discount offers viewed by customer
total_offer_viewed_informational
    Total number of informational offers viewed by customer
total_web_offers_bogo
    Total number of bogo offers sent through web
total_web_offers_discount
    Total number of discount offers sent through web
total_web_offers_informational
    Total number of informational offers sent through web

2. Missing Gender Values

   With the extended list of features, we then decided to use machine learning to predict the genders of customer with missing gender data. The reason behind using a machine learning approach is to make sure that the data portrays the same trend as the rest of the data.

   We decided to benchmark multiple machine learning algorithm to see what will work best to predict these missing values. Since this was a multiclass classification problem we decided to use following algorithms to benchmark.



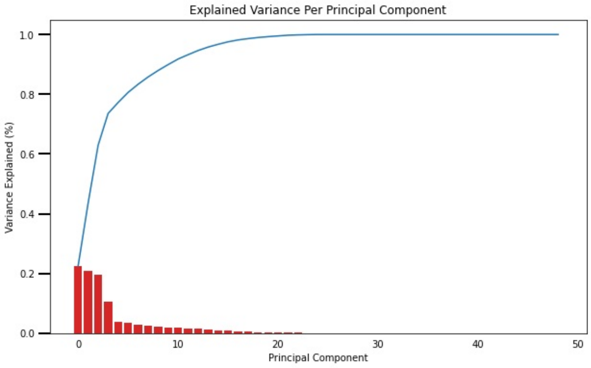
   The distribution of gender is shown below The unknown bar shows the count of missing gender data

   1. Feature Scaling

      Since the data was not originally scaled and it is important scale the data. We decided to do simple standardization

   2. PCA

      Then we did Principle Component Analysis to find out the number of principle component that cover most of the data variance.



      The explained variance graph showed that in order to cover the most of the explaned variance we needed atleast 25 principle component

      With this informational we retrained the PCA.

   3. Training and Benchmarking

      1. Metrics Used

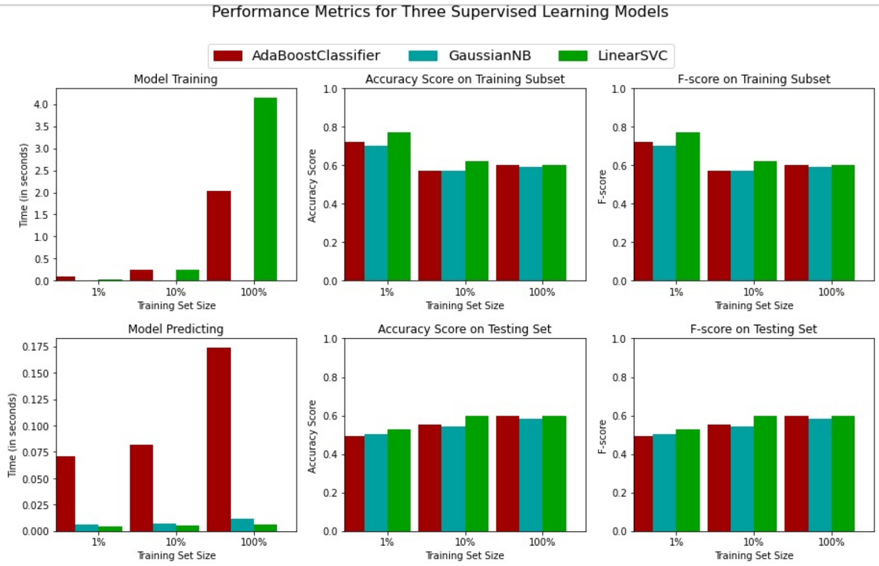
         We used the accuracy score, f{β} score where β is 0.5 and training time to compare the benchmarking models

      2. Benchmarking

         For Benchmarking the model we used following the models

         - AdaBoost Classifier
         - Guassion Naive Bayes
         - Linear SVC

All three models were trained on 1%, 10% and 100% of dataset to see their benchmark values. The result of benchmarking are shown below



3. Training

   Since AdaBoost Classifier was fractionally better than other two and had very small training duration we decided to use to model the data and fine tune it further

   1. Tuning Parameters

      n_estimators

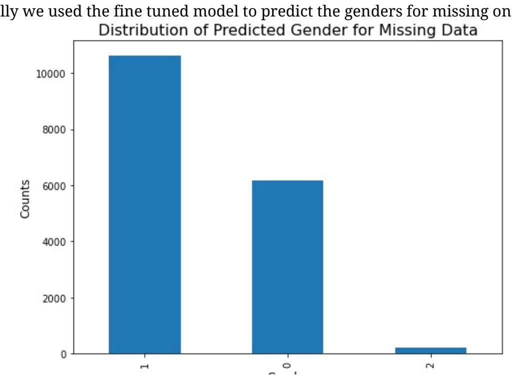
      we used a range of 80 - 102 estimators on GridSearch to achieve the fine tuned model

   2. Tuned Model Score

      - We achieve 60% accuracy and 0.604 f{β} score.
      - Since the data we used to predict gender had very little co-relation with gender value thus this score is good enough and will atleast give better value than doing a random guess

4. Predict Data

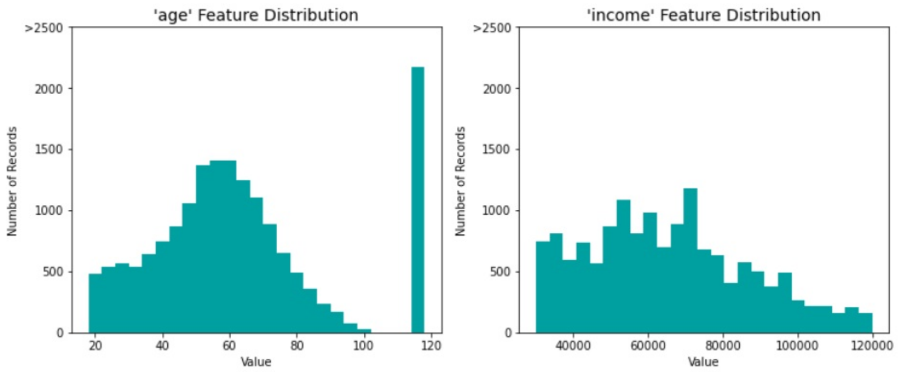
   Finally we used the fine tuned model to predict the genders for missing ones. The distribution after prediction was following



3. Missing Income and Age Data

   Similar to gender data we decided to predict the value of income and data column but unlike gender these values were Continuous and need regression algorithms to model them

   1. Data Distribution



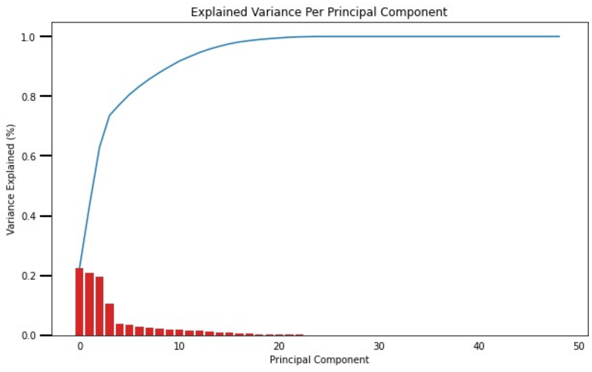
From the distribution of data we can see that large number of customer have age 118 which is probably default age set in case of missing data. So we first of all removed this value and replaced it with Nan, secondly, it looks like both the dataset are skewed to left side so the prediction should also be similar

2. Feature Scaling

   Since the data was not originally scaled and it is important scale the data. We decided to do simple standardization

3. PCA

   Then we did Principle Component Analysis to find out the number of principle component that cover most of the data variance.



The explained variance graph showed that in order to cover the most of the explaned variance we needed atleast 25 principle component

With this informational we retrained the PCA.

4. Training and Benchmarking

   1. Metrics Used

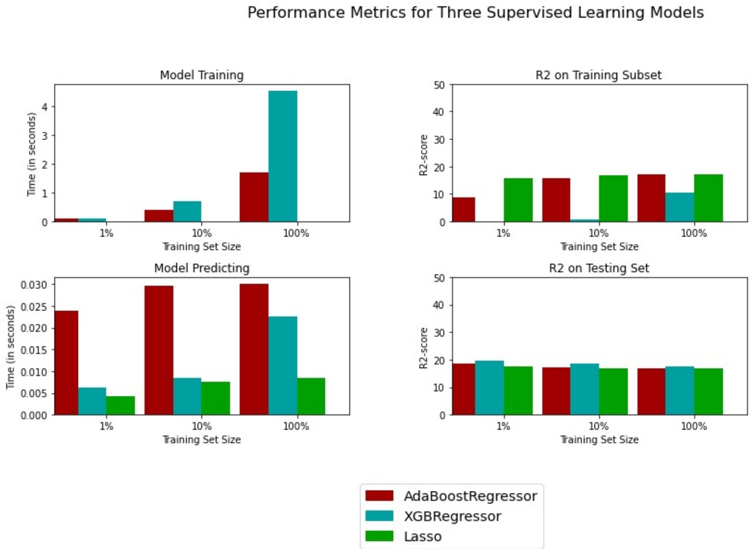
      Since these were regression problem we use following metrics

      - RMSE

   2. Benchmarking

      The algorithm used in benchmarking were

      - AdaBoost Regressor
      - XGB Regressor
      - Lasso Bench Marking result is as followed



From the benchmark it turned out tha AdaBoost Regressor is the best option marginally.

3. Training

   1. Age

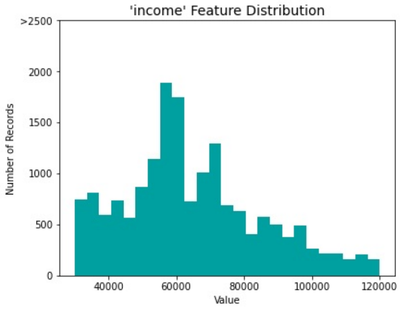
      - We achieved RMSE of ~16
      - That means the values predicted by our model had an error of around 16 years, but again since the data we used to predict is just the transactions data, it is highly unlike we could have achieved a better value with other models

   2. Income

      - We achieved RMSE of ~19468
      - That means the values predicted by our model had an error of around 19000 dollars, but again since the data we used to predict is just the transactions data, it is highly unlike we could have achieved a better value with other models

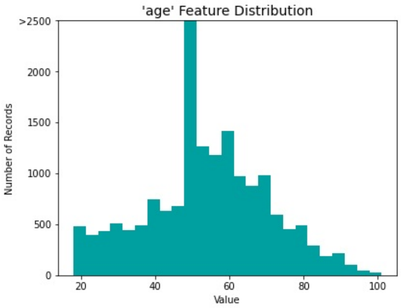
5. Predict Data

   1. Income



Income value distribution after prediction

   2. Age



Age value distribution after prediction

2. Implementation

Initially we had thought we will use classification model to do the prediction of offer type but in order to do that we need to manually label the customer data and based on some heuristics. So instead we changed the approach as we started the modeling customer data

1. Clustering instead of Classification

   The idea behind this new approach was that the users have difference in their transaction behavior and they can be grouped or cluster based on those behavior and finally once we have found those clusters we can easily apply heuristics on those clusters to see which offer will fit which cluster

2. Dataset

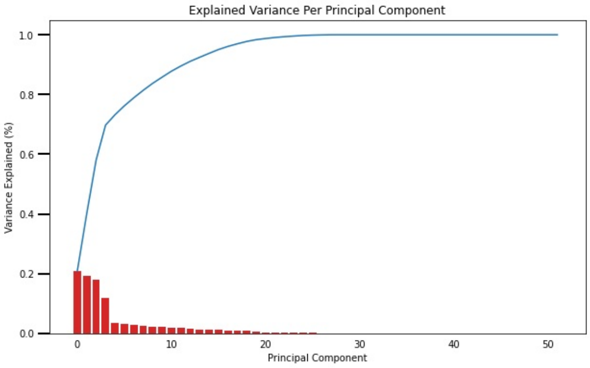
   We had 170000 customers with 53 feature each. The data is not a lot but was be enough perform the clustering operation

3. Feature scaling

   We standardized the features to make sure all of them will have the same importance while clustering

4. PCA

   Then we performed the PCA with all the data that we had created, the graph looked similar to before.

   1. Component Breakdown

      This time we also did a breakdown of top three component to see which features covered the most variance

      1. Principle Component 1

         - total_social_offers_discount
         - total_web_offers_discount
         - total_offer_received_discount
         - total_mobile_offers_discount
         - total_email_offers_discount

      2. Principle Component 2

         - total_email_offers
         - total_mobile_offers
         - total_offer_received
         - total_social_offers
         - total_web_offers

      3. Principle Component 3

         - total_social_offers_informational
         - total_web_offers_informational
         - total_offer_received_informational
         - total_mobile_offers_informational
         - total_email_offers_informational

      4. Principle Component Discussion

         - Looks like the most variance is captured by discount offers
         - Second thing that we can see from the above analysis is that the medium of delivering offer also captures a lot of variance
         - Informational offers also capture some variance

5. Modeling

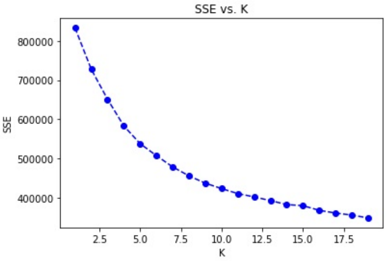
   1. Choosing the clustering algorithm

      We decided to use K-nearest neighbors algorithm for clustering. Due to following reasons

      - It works on all kind of data
      - It can do multi cluster grouping

   2. Finding the best n - number of clusters

      To find the optimum number of clusters and we ran the algorithm multiple time on the data and used the elbow method to



determine the optimum n

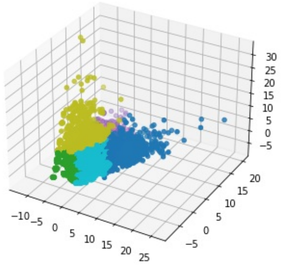
1. 6 is the optimum number of clusters

   Looking at the graph of Sum squared error vs K, we can see that 6 is the number of clusters which lies the elbow. This also falls inline with the fact that there are around the same numbers of offer categories

   - Discount
   - Bogo
   - Informational
   - No Offers
   - Sleeping Dogs

3. Predicting Cluster of customer

   Using the 6 as K we predicted clusters for each customer and assigned them. We also visualized the clusters to see how the show up



on multi dimensional space

4. Grouping data on clusters

   After prediction we grouped the data over clusters so that we can apply some heuristics on each cluster and find out which offer would be best for which customer

   1. Heuristics

      Sleeping Dogs
      : Group of customer whose spending is less during offers and more outside offers

      Discount
      : Group who has completed more than 50% Discount offer and has completed more discount offer than any other type

      Bogo
      : Group who has completed more than 50% Bogo offer and has completed more bogo offer than any other type

      Informational
      : Group who has spent more than 30% during Informational offer and has viewed more isformational offer than any other type

3. Refinement

   We used SSE to find the best number of clusters in the customers, in order to improve our cluster modeling we can add more customer specific demographics like their vicinity, their profession e.t.c. Also, it we be a good idea to retrain the model and generate new clusters every few days with new transactions data, that will help us incorporate change behavior of customers

IV. Results

1. Model Evaluation and Validation

   The fact the our model generated the cluster approximately equal to our expected customer group, gives us the confidence this model has been able to solve the problem we wanted to solve

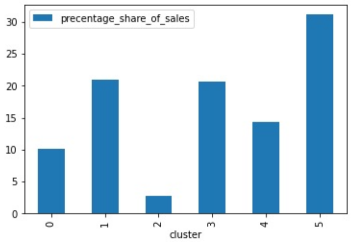
2. Justification

   Since this a clustering model our benchmarking is against other cluster numbers we used 6 as number clusters because according the elbow method that is the optimum cluster number. The SSE value is approximately 450000.

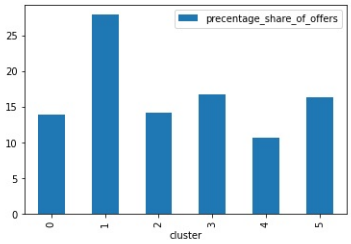
V. Conclusion

Now that we have a model that can predict the cluster of customer based on it past transaction, we can easily then decide which offer to send to them on quick heuristics.

1. Free-Form Visualization



Cluster 5 has the greatest share of sales, meaning that cluster 5 has more buying power and likes starbucks products



on the other hand, cluster 1 received the most offers over the period of time.

This shows that basically if we had used the basic rule that the customer who received more offer will pay us more is not likely.

2. Reflection

   We went from having 3 dataset to generating a clustering model that can identify the which offer will be good for a certain customer.

   The difficulty in this problem was basically looked like a classification problem but turned out to be a clustering problem as we didn't have properly labeled data.

   But the model that we have achieved has high probability of predicting the right offers for a group of customers

3. Improvement

   We can improve the model prediction by adding more customer specific demographics data and also retraining the model on more data over time.