Project Phase 1 Report

CSE 587 - Data Intensive Computing

Marketing Campaign Value Analysis

Team Members

Name	Email	UBIT	UBN	Contribution
Gupta Suchit	suchitni@buffalo.edu	suchitni	50518842	33.33%
Gupta Nikhil	ngupta22@buffalo.edu	ngupta22	50534276	33.33%
Pandey Ashmita	pandey7@buffalo.edu	pandey7	50485164	33.33%

Evaluation Criteria	Gupta Suchit	Gupta Nikhil	Pandey Ashmita
How effectively did your group mate work with you?	5	5	5
Contribution in writing the report	5	5	5
Demonstrates a cooperative and supportive attitude	5	5	5
Contributes significantly to the success of the project .	5	5	5
Total	20	20	20

Problem Statement

Background

Digital Marketing is one of the top buzzwords of this decade, which does not come as a surprise as the marketing industry is 2nd largest in value of roughly around $460 billion. Marketing takes around 14% of a typical large scale company's budget, generates exponential revenue returns across various verticals. [1]

With the advent of smart technologies and tremendous amounts of user data, we have moved from mass marketing to targeted marketing, where it is extremely important to know the background of the audience before reaching them. In order to save on resources as well as the big fear of forming a bad perception of the consumer, it is important to comprehensively analyze customer behavior and background.

Problem Statement

We identified the potential of this market and are working on a project which will answer the following questions:

- How effective will the campaign be on a new set of users?
- We also try to recommend the best marketing campaign to run for any new user.
- Based on user's background we can predict user's spending patterns

Potential

Given the scale of this problem statement in terms of monetary value as well the population reach, we can make a significant impact to aid an organization's marketing team and we plan to automate this system of campaign strategy design by requiring only minimal manual intervention.

Dataset Overview

The dataset chosen contains information of customer interactions with a company's marketing campaigns. By analyzing this data, we can derive actionable insights and strategies for future marketing initiatives, thereby increasing the value of these campaigns for both end users as well as businesses.

This dataset comprises of:

- Rows/Records: 2240
- Columns/Features: 29

This structured dataset provides information about customers, encompassing demographic details, personal and professional background, spending habits, and responses to previous marketing campaigns.
Source: Kaggle [2]

Phase 1 Overview

In the previous phase we successfully understood the dataset, performed cleaning of inconsistencies and errors, standardized the data and performed exploratory data analysis and derive insights and correlations among the data points.

We plotted multiple graphs and used visualizations techniques to get a general idea on which parameters are more important or relate closely to consumer behavior and set a basic set of output expectations from our models in the next phase.

Phase 2 Objective

In this phase we aim to create models that help us understand this data and consumer behavior mathematically and perform predictions for unknown or missing or future data points. This will help us understand which bucket of consumers the current marketing techniques work the best upon and which category should we target our campaigns on.

Models

1. Logistic Regression

It is extremely crucial to understand how consumers interact with marketing campaigns and have early predictions whether a consumer with certain spending patterns and interactions will give a positive response or not.

This strategy can help us generate targeted campaigns and get a better response turn out ratio, thus resulting in saving operations labor as well as costs.

To achieve this, we split our dataset into training and testing sets in a 80:20 ratio respectively. Our target variable is consumer “Response”. By using the correlation matrix and visualizations from the previous phase, we choose the following predictor variables:

1. **Consumer attributes**: Income, Age, Customer_From
2. **Consumer spending pattern**: Wines, Fruits, Meat, Fish, Sweet, Gold
3. **Consumer responses to previous campaigns**: AcceptedCmp3, AcceptedCmp4, AcceptedCmp5, AcceptedCmp1, AcceptedCmp2
4. **Consumer past behavior**: Deal_Purchases, Web_Purchases, Catalog_Purchases, Store_Purchases, Web_Visits

```python
logistic_regression_data = data[['Income', 'Wines', 'Fruits', 'Meat', 'Fish', 'Sweet', 'Gold', 'Children', 'Age', 'Customer_From' ,
logistic_regression_X = logistic_regression_data.drop(columns=['Response'])
logistic_regression_y = logistic_regression_data[['Response']]

lgr_X_train, lgr_X_test, lgr_y_train, lgr_y_test = train_test_split(logistic_regression_X, logistic_regression_y, test_size=0.2, rar
```

Python

We train our Logistic Regression Model on training data.

```
lgr = LogisticRegression()
lgr.fit(lgr_X_train, lgr_y_train)
```

We subsequently test our model on the testing data.

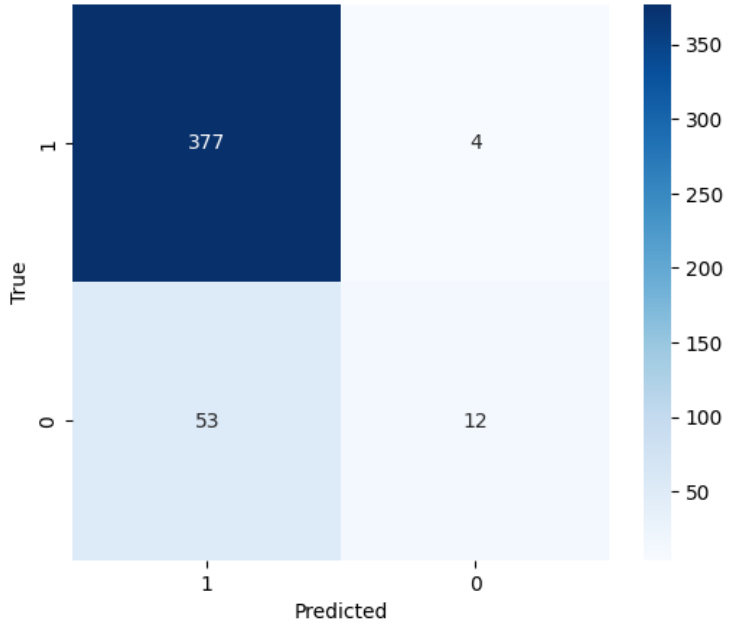
```
lgr_y_pred = lgr.predict(lgr_X_test)

accuracy = accuracy_score(lgr_y_test, lgr_y_pred)
print('Accuracy Score for logistic regression on Response is ', accuracy*100)
```

✓ 0.1s

```
Accuracy Score for logistic regression on Response is  87.21973094170403
```

We get a testing accuracy of 87.2% on our testing data

Since, accuracy is not the best parameter to judge the results, we plot a confusion matrix to find Precision and Recall as well



Now, for a new set of consumers, we are ready to understand if our campaign will work on them or not, or for the existing set of consumers, we try targeting them with different variety of campaigns and observe the difference in response behavior.

2. Linear Regression

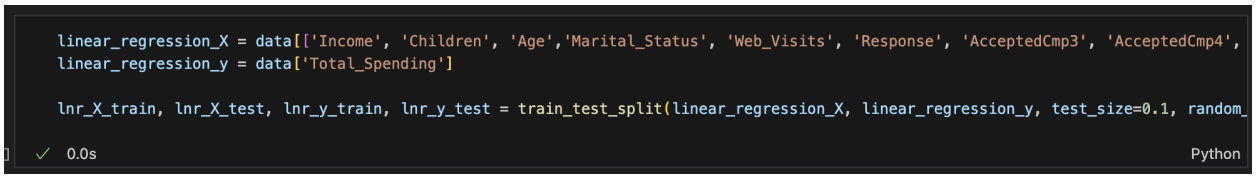
Revenue Prediction is really crucial for any organization for planning the future prospects as well as general market obligations. Generally, revenue is calculated with current consumers as base. But using Machine Learning models, based on past user interactions and characteristics, we can predict what will be the revenue from prospective consumers and thus more accurately predict the financial standing of the organization.

For predicting the total spending of any consumer, we use linear regression as our predictor model and Total Spending as our target variable.

We only use general consumer characteristics and interaction with the previous campaigns and deliberately leave out any consumer spending variables to remove biases from our prediction. Hence, we choose the following predictor variables

1. **Consumer Attributes**: Income, Children, Age,Marital_Status
2. **Consumer responses to previous campaigns**: AcceptedCmp3, AcceptedCmp4, AcceptedCmp5, AcceptedCmp1, AcceptedCmp2, Web_Visits, Response
3. **Past purchases**: Deal_Purchases, Web_Purchases, Catalog_Purchases, Store_Purchases

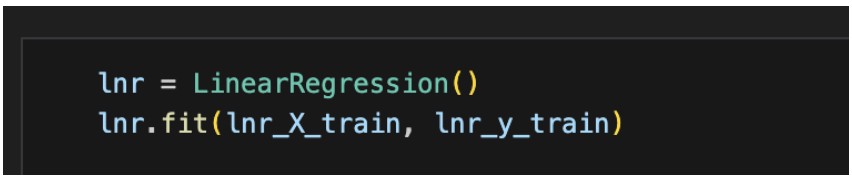
We split our training and testing dataset into 90:10 ratio respectively



```
linear_regression_X = data[['Income', 'Children', 'Age','Marital_Status', 'Web_Visits', 'Response', 'AcceptedCmp3', 'AcceptedCmp4',
linear_regression_y = data['Total_Spending']

lnr_X_train, lnr_X_test, lnr_y_train, lnr_y_test = train_test_split(linear_regression_X, linear_regression_y, test_size=0.1, random_
```

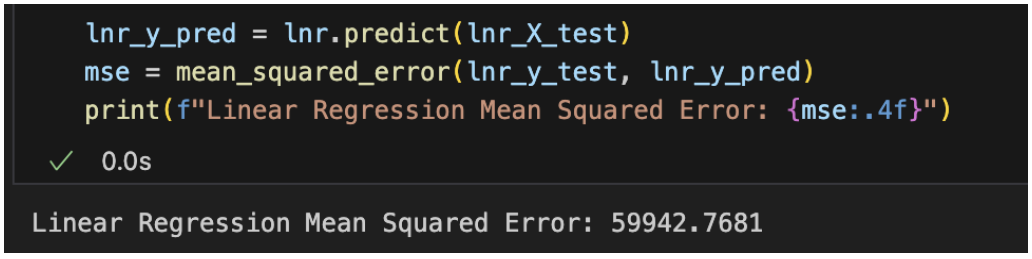
✓ 0.0s Python

We then train our data on the training dataset



```
lnr = LinearRegression()
lnr.fit(lnr_X_train, lnr_y_train)
```

Subsequently, we test our data on the testing dataset and calculate the mean squared error

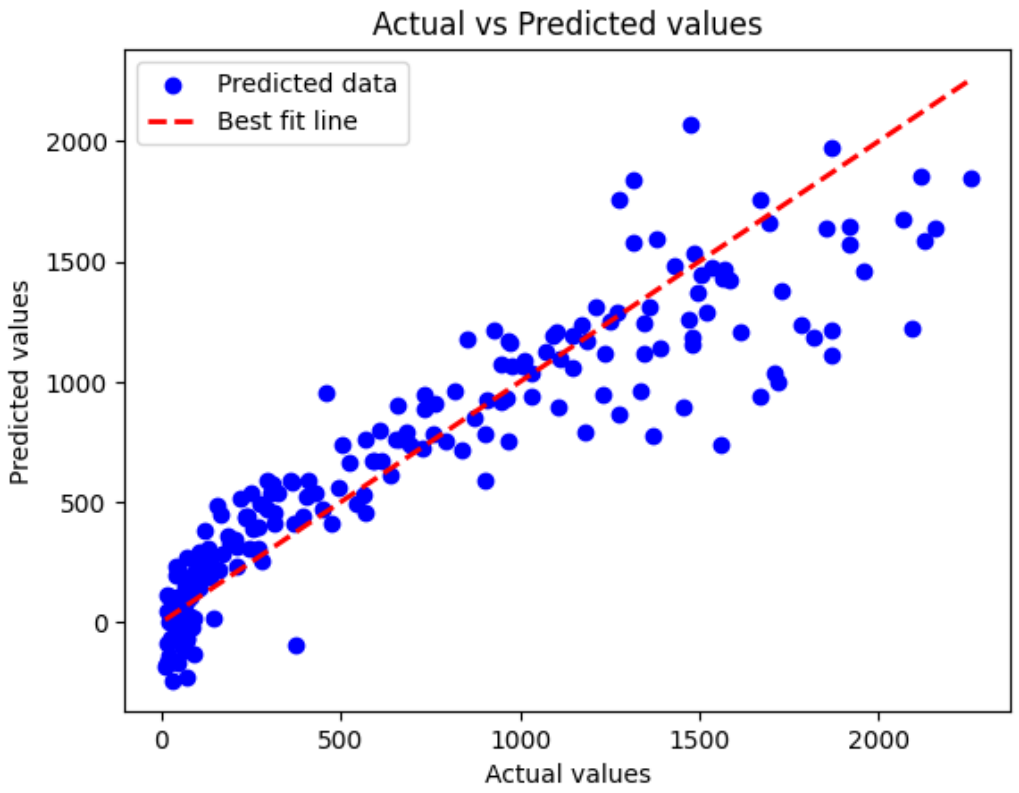


```
lnr_y_pred = lnr.predict(lnr_X_test)
mse = mean_squared_error(lnr_y_test, lnr_y_pred)
print(f"Linear Regression Mean Squared Error: {mse:.4f}")
```

✓ 0.0s

```
Linear Regression Mean Squared Error: 59942.7681
```

To get an idea of the extent of loss, we visualize our predictor line and the actual values



This can help us predict the revenue from prospective consumers and thus help to make better future decisions for the organization.

3. K Means

Consumer loyalty is highly regarded for any business to be sustainable and such customers are often given preferential treatment in order to maintain healthy face value and long term connection.

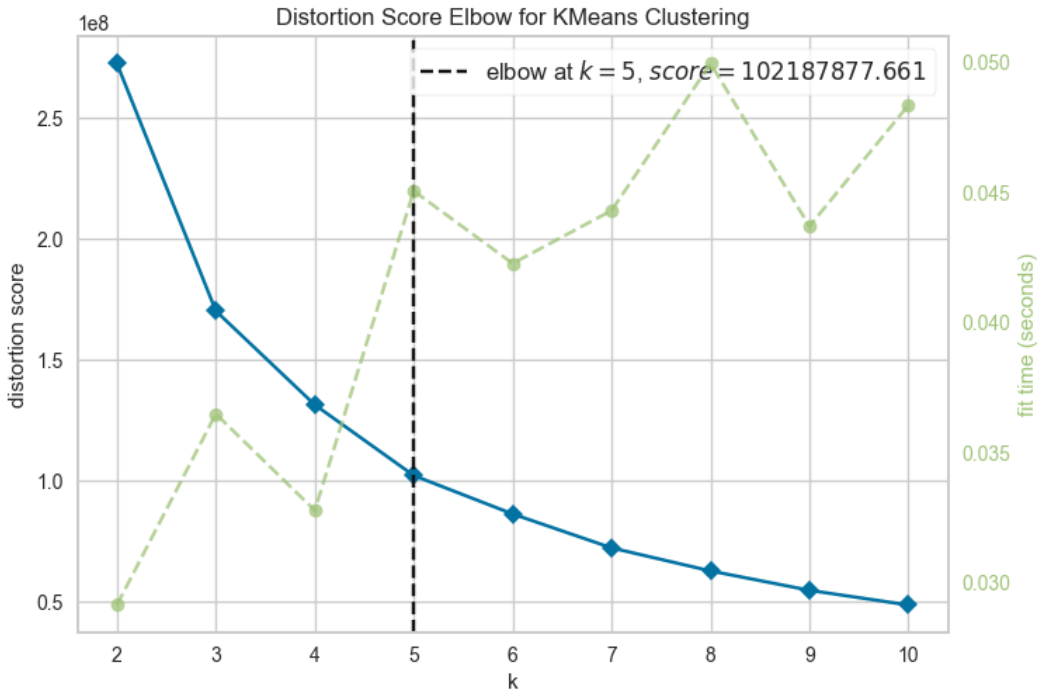
We are using the K Means method to find out the most loyal customers based on the time since when the consumers signed up and how much the consumer has spent.

We first find out the best K using elbow method, for this we use an existing python package called yellowbrick, to visualize the best k

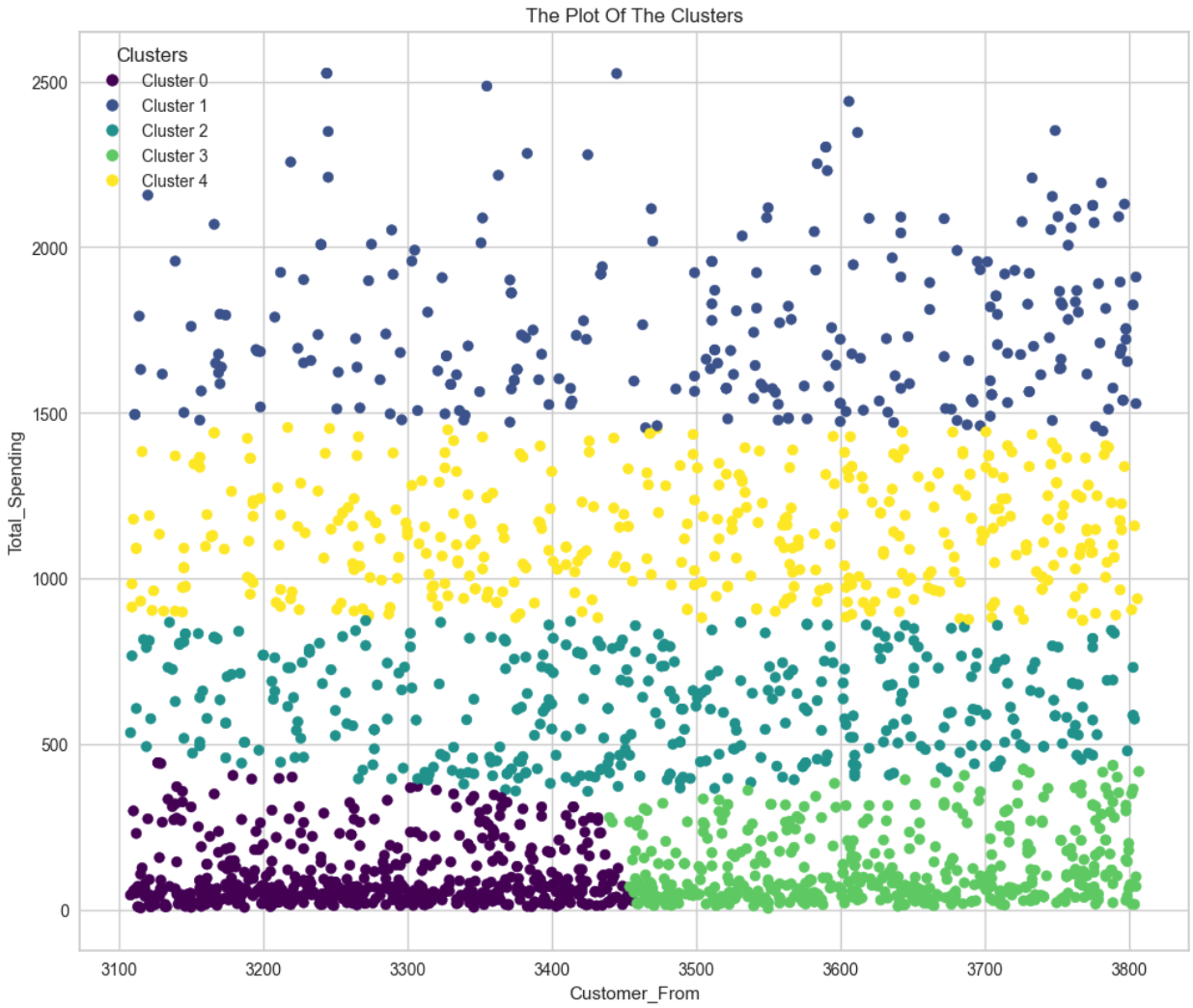
```
kmeans = KMeans()
k_means_data = data[['Customer_From', 'Total_Spending']]
optimal_k_finder = KElbowVisualizer(kmeans, k=10)
optimal_k_finder.fit(k_means_data)
optimal_k_finder.show()

plt.show()
```

✓ 0.5s

We get the following elbow graph for optimal K



We then follow it by clustering our consumers into 5 clusters and get the following clusters



Here, we can see that consumers in Cluster 1 are the most loyal or important users as they have spent more in a wide range of time.
However, users in Cluster 1 have not been very active.

It is crucial to identify that consumers in Cluster 4 and Cluster 2 are essential as these consumers are at the edge and business can easily convert them to long term customers.

4. KNN

It is important that campaigns address the actual need of the consumer rather than blindly pushing products that are irrelevant. Hence, it is important to understand what category the consumer belongs to, will the consumer be able to afford the product etc.

To achieve this, we classify our users into luxury spenders and try to predict if the user qualifies as a luxury spender by referring to basic attributes and other purchase patterns of the user.

Using KNN we can identify which category a consumer belongs to based on past consumer patterns. Using this information we can make targeted campaigns for luxury users and aim high margin and value products to this category to improve business.

We start by splitting our data into a training and testing set in 80:20 ratio

```
knn_X = data[['Income','Catalog_Purchases','Store_Purchases','Web_Purchases','Age']]
knn_y = data[['Luxury_Consumer']]

knn_X_train, knn_X_test, knn_y_train, knn_y_test = train_test_split(knn_X, knn_y, test_size=0.2, random_state=42)
```

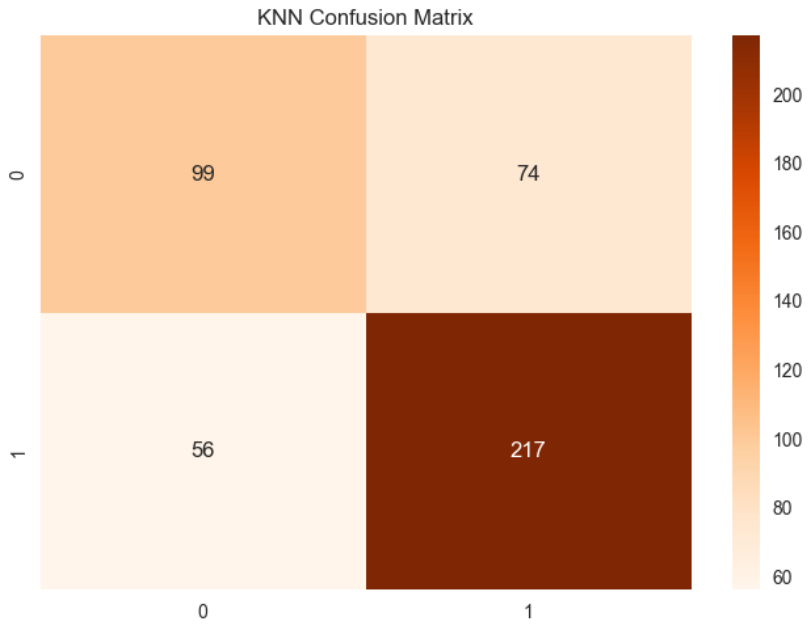
We scale the dataset and train the model on training data

```
# Standardize the features
scaler = StandardScaler()
knn_X_train_scaled = scaler.fit_transform(knn_X_train)
knn_X_test_scaled = scaler.transform(knn_X_test)

knnClassifier = KNeighborsClassifier(n_neighbors=5)
knnClassifier.fit(knn_X_train_scaled, knn_y_train)
```

Subsequently we test our model to find how well does it perform to classify unknown users

```
knn_y_pred = knnClassifier.predict(knn_X_test_scaled)

accuracy = accuracy_score(knn_y_test, knn_y_pred)
confusion_mat = confusion_matrix(knn_y_test, knn_y_pred)
print(f"KNN Accuracy: {accuracy:.4f}")
sns.heatmap(confusion_matrix(knn_y_test, knn_y_pred), annot=True, fmt='d', cmap='Oranges')
plt.title('KNN Confusion Matrix')
plt.show()
```

We get 70% accuracy on our model and we draw the confusion matrix to draw relevant insights using precision and recall.



We can see that our model correctly identifies the majority of True positive and True negative scenarios. This accuracy still works even if we identify users as non luxury users and push expensive campaigns on them, we still have a chance of upselling and thus generating good profits.

5. Decision Tree

Inactive consumers is a big problem for any business and it is important to identify why the consumers went inactive in order to stay in touch with the consumer and the competition.
To begin with this we use basic consumer info to predict if the consumer is worth paying attention or not.
We use “Need Attention” as the target variable, which represents that a consumer is inactive for more than 30 days.
We use Income, Total_Purchase and Response as the predictor variable.

We divide our training and testing data in 80:20 ratio and use Standard Scalar to scale the same to get consistent results.

```
dt_X = data[['Income', 'Total_Spending', 'Customer_From', 'Response']]
dt_y = data[['Need_Attention']]

dt_X_train, dt_X_test, dt_y_train, dt_y_test = train_test_split(dt_X, dt_y, test_size=0.2, random_state=42)

# Standardize the features
scaler = StandardScaler()
dt_X_train_scaled = scaler.fit_transform(dt_X_train)
dt_X_test_scaled = scaler.transform(dt_X_test)
```

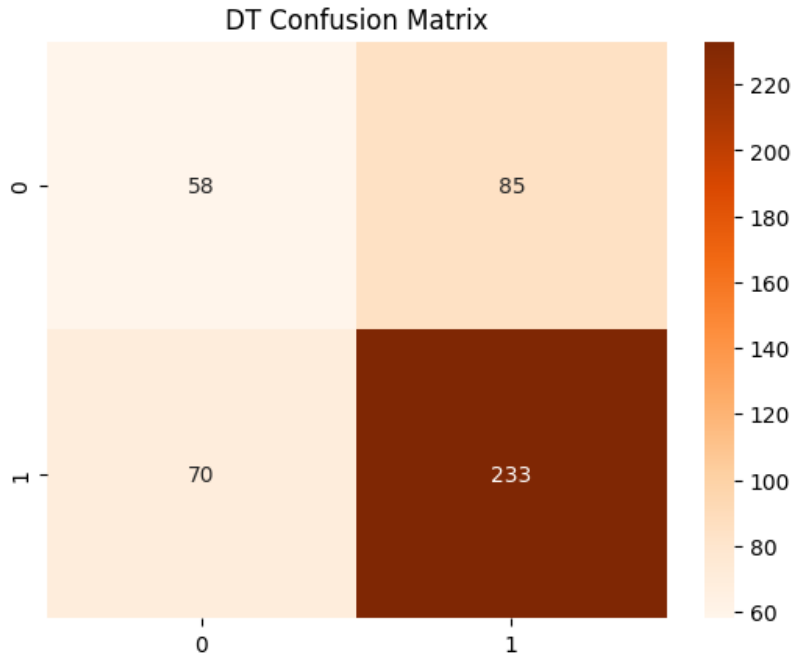
✓ 0.0s Python

We first fit our model on training data and then test it.

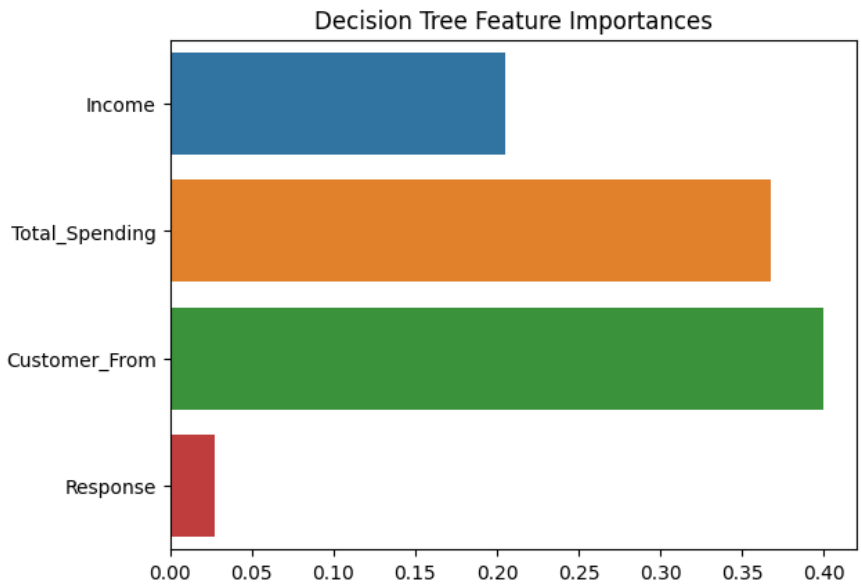
```
# Decision Trees
decisionTree = DecisionTreeClassifier(random_state=42)
decisionTree.fit(dt_X_train_scaled, dt_y_train)
dt_y_pred = decisionTree.predict(dt_X_test_scaled)
```

We get an accuracy of 65% which is based on the most basic input variables and determines highly unpredictable consumer behavior.

We plot the confusion matrix for the same to get more insights on false positive and true negative ratios



We also plot the impact that each input variable causes



We can see that customer_from and total_spending has highest impact which makes sense as old customers who spend more have highest retention value

This model helps us find the retention value of consumers and hence help to increase customer reorder value or repeat business which is one of the most important parameters for sustainable businesses.

6. Gradient Boosting

In our dataset we can observe that the organization ran 6 campaigns in total. It is crucial for any business to target the consumers that responded the earliest as such kind of users are the easiest to scale the business with. Additionally it is important to identify such consumers and design campaigns that regularly gather consumers in the first campaign to save costs on a large scale

Using gradient boosting we aim to target the consumers that responded back in the first campaign itself. Here the target variable is “AcceptedCmp1”
We consider Income, Customer_From and Total_Spending as our predictor variables

We split our data into 80:20 ratio for training and testing respectively

```
GB_X = data[['Income','Customer_From','Total_Spending']]
GB_y = data[['AcceptedCmp1']]

GB_X_train, GB_X_test, GB_y_train, GB_y_test = train_test_split(GB_X, GB_y, test_size=0.2, random_state=42)
```

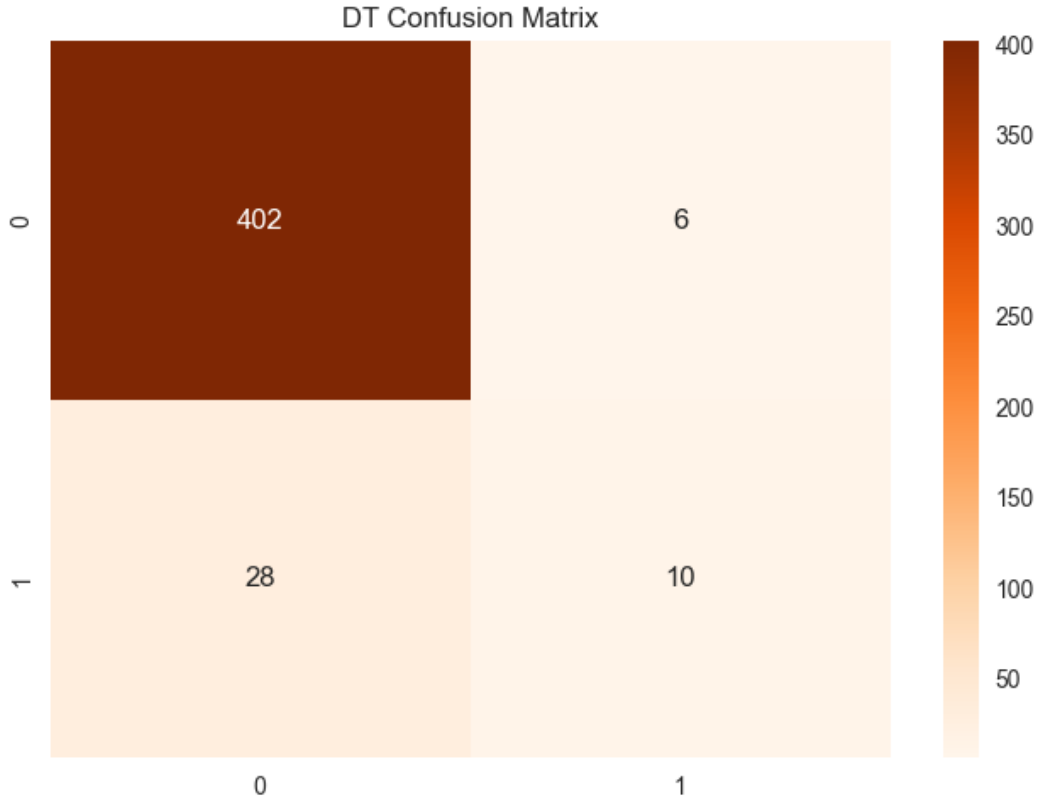
We scale our predictor variables and train our gradient boosting model

```
# Standardize the features
scaler = StandardScaler()
GB_X_train_scaled = scaler.fit_transform(GB_X_train)
GB_X_test_scaled = scaler.transform(GB_X_test)

gradientBoosting = GradientBoostingClassifier(random_state=42)
gradientBoosting.fit(GB_X_train_scaled, GB_y_train)
```

Finally, we test test our model on testing data

```
GB_y_pred = gradientBoosting.predict(GB_X_test_scaled)

accuracyGB = accuracy_score(GB_y_test, GB_y_pred)
print(f"GB Accuracy: {accuracyGB:.4f}")
sns.heatmap(confusion_matrix(GB_y_test, GB_y_pred), annot=True, fmt='d', cmap='Oranges')
plt.title('DT Confusion Matrix')
plt.show()
```

We get an accuracy of 92% and draw confusion matrix to analyze the results in depth



Here we can see that we correctly identify the majority of the true positive cases which is crucial to our campaign as the point of this model is to identify the users who are highly likely to respond in the first

campaign itself. Thus, the model serve to be very important in order to target consumers with right campaign

Citations:

[1]: Marketing statistics: https://www.wordstream.com/blog/ws/2022/04/19/digital-marketing-statistics

[2]: Dataset source: https://www.kaggle.com/datasets/imakash3011/customer-personality-analysis/data

References:

1. Matplotlib: https://matplotlib.org/stable/
2. Pandas: https://pandas.pydata.org/docs/user_guide/index.html
3. Seaborn: https://seaborn.pydata.org/tutorial/introduction.html
4. Scikit Learn: https://scikit-learn.org/stable/user_guide.html
5. CSE 587 - Data Intensive Computing: Lecture Slides