E-COMMERCE ANALYSIS

Created by Shweta Purushothaman

Batch no.: **WS103806**

Outline

- Introduction
- Data Overview
- Data Cleaning and Preprocessing
- Feature Engineering
- Exploratory Data Analysis (EDA)
- Modelling Approach
- Model Performance
- Insights and Findings
- Recommendations
- Challenges and Limitations
- Future Work
- Conclusion
- Appendix
- References

Introduction

- Project background and context

An international e-commerce company which sells electronic products wants to discover key insights from their customer database. They want to use some of the most advanced machine learning techniques to study their customers.

- Problems we want to find answers
  - What was Customer Rating? And was the product delivered on time?
  - Is Customer query is being answered if product importance is high, having highest customer rating or being delivered on time?

Data Overview

The dataset used for model building contained 10999 observations of 12 variables.

The data contains the following information:

- **ID**: ID Number of Customers.
- **Warehouse block**: The Company have big Warehouse which is divided in to block such as A,B,C,D,E.
- **Mode of shipment**: The Company Ships the products in multiple way such as Ship, Flight and Road.
- **Customer care calls**: The number of calls made from enquiry for enquiry of the shipment.
- **Customer rating**: The company has rated from every customer. 1 is the lowest (Worst), 5 is the highest (Best).
- **Cost of the product**: Cost of the Product in US Dollars.
- **Prior purchases**: The Number of Prior Purchase.
- **Product importance**: The company has categorized the product in the various parameter such as low, medium, high.
- **Gender**: Male and Female.
- **Discount offered**: Discount offered on that specific product.
- **Weight in grams**: It is the weight in grams.
- **Reached on time**: It is the target variable, where 1 Indicates that the product has NOT reached on time and 0 indicates it has reached on time.

Data Cleaning and Preprocessing

- Data Collection
  - Customer database of an electronic products selling e-commerce international company.
  - File source: .csv file
- Data Inspection
  - Shape of the dataset: 10999(rows), 12 (columns)
  - No. of duplicate records: 0
  - Data types: int64(continuous data) and object(categorical data)
    - Continuous data variables: ID, Customer_care_calls, Customer_rating, Cost_of_the_Product, Prior_purchases, Discount_offered, Weight_in_gms, Reached.on.Time_Y.N
    - Categorical data variables: Warehouse_block, Mode_of_Shipment, Product_importance, Gender
- Handling Missing Data
  - No. of missing values: 0

Data Cleaning and Preprocessing

Outlier Detection

Skewness	Distribution
0	Symmetrical / Normally Distributed
< -1 / +1 >	Highly negative/positive skewed
-1 to -0.5 / 0.5 to 1	Moderately negative/positive Skewed
-0.5 to 0.5	Approximately symmetric

- **Skewness**: it tells us the direction of potential outliers. Skewness is the spreadness of the Distribution.

Warehouse_block	Mode_of_Shipment	Customer_care_calls	Customer_rating	Cost_of_the_Product	Weight_in_gms	Reached.on.Time_Y.N	Prior_purchases	Discount_offered	Product_importance
-0.279666	-0.000299	0.391926	0.00436	-0.157117	-0.249747	-0.394257	1.681897	1.798929	0.585372
Approximately symmetric							Highly positive Skewed		Moderately positive Skewed

Kurtosis	Distribution
0	Mesokurtic distribution (normal distribution)
< 0	Platykurtic distribution (fewer outliers)
> 0	Leptokurtic distribution (more outliers and extreme values)

- **Kurtosis**: It helps identify whether the data has heavy tails (outliers) or is more concentrated around the mean (light tails) compared to a normal distribution. Kurtosis is a measure of tailedness of a distribution.

Warehouse_block	Customer_care_calls	Customer_rating	Cost_of_the_Product	Weight_in_gms	Reached.on.Time_Y.N	Product_importance	Mode_of_Shipment	Prior_purchases	Discount_offered
-1.364861693	-0.309399963	-1.295610522	-0.972263869	-1.44755807	-1.84460357	-0.627083465	0.109695854	4.003975843	1.999130776
Platykurtic Distribution							Leptokurtic Distribution		

- No. of outliers detected: 2 (Prior_purchases and Discount_offered)

Data Cleaning and Preprocessing

- Outlier Detection: we used Box Plot (it is the non-parametric method) technique

```
Prior_purchases Column
SUMMARY :
count    10999.00
mean         3.57
std          1.52
min          2.00
25%          3.00
50%          3.00
75%          4.00
max         10.00
Name: Prior_purchases,
BOX PLOT VALUES :
First Quartile : 3.0
Second Quartile : 4.0
IQR Range : 1.0
Lower Range : 1.5
Upper Range : 5.5

Total Outlier ABOVE UPPER RANGE : 1003
Total Outlier BELOW LOWER RANGE : 0
Total Outlier in 'Prior_purchases' Column in the Dataset : 1003
```

```
Discount_offered Column
SUMMARY :
count    10999.00
mean        13.37
std         16.21
min          1.00
25%          4.00
50%          7.00
75%         10.00
max         65.00
Name: Discount_offered,
BOX PLOT VALUES :
First Quartile : 4.0
Second Quartile : 10.0
IQR Range : 6.0
Lower Range : -5.0
Upper Range : 19.0

Total Outlier ABOVE UPPER RANGE : 2209
Total Outlier BELOW LOWER RANGE : 0
Total Outlier in 'Discount_offered' Column in the Dataset : 2209
```

- Outlier Treatment: Remove the outlier from the dataset based on Upper and Lower Range respectively

Feature Engineering

- Feature Selection

  Recursive Feature Elimination: removing the least important features based on model performance. Gender and ID columns which give us no relevant information for analysis. Hence we can drop them.

- Feature Extraction

  Categorical Data: Encoding categorical variables into numerical format

  (Warehouse_block, Mode_of_Shipment, Product_importance)

```
Warehouse_block: ['A' 'B' 'C' 'D' 'F']
Mode_of_Shipment: ['Flight' 'Ship' 'Road']
Product_importance: ['low' 'medium' 'high']
```

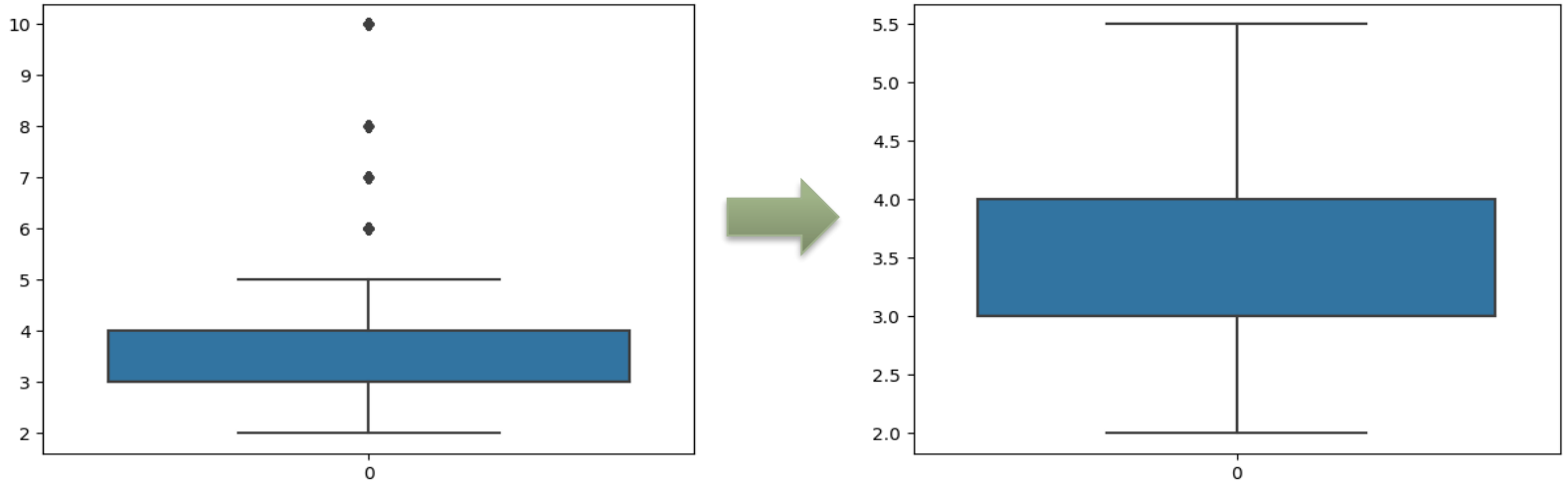


```
Warehouse_block: ['1' '2' '3' '4' '5']
Mode_of_Shipment: ['1' '2' '3']
Product_importance: ['1' '2' '3']
```

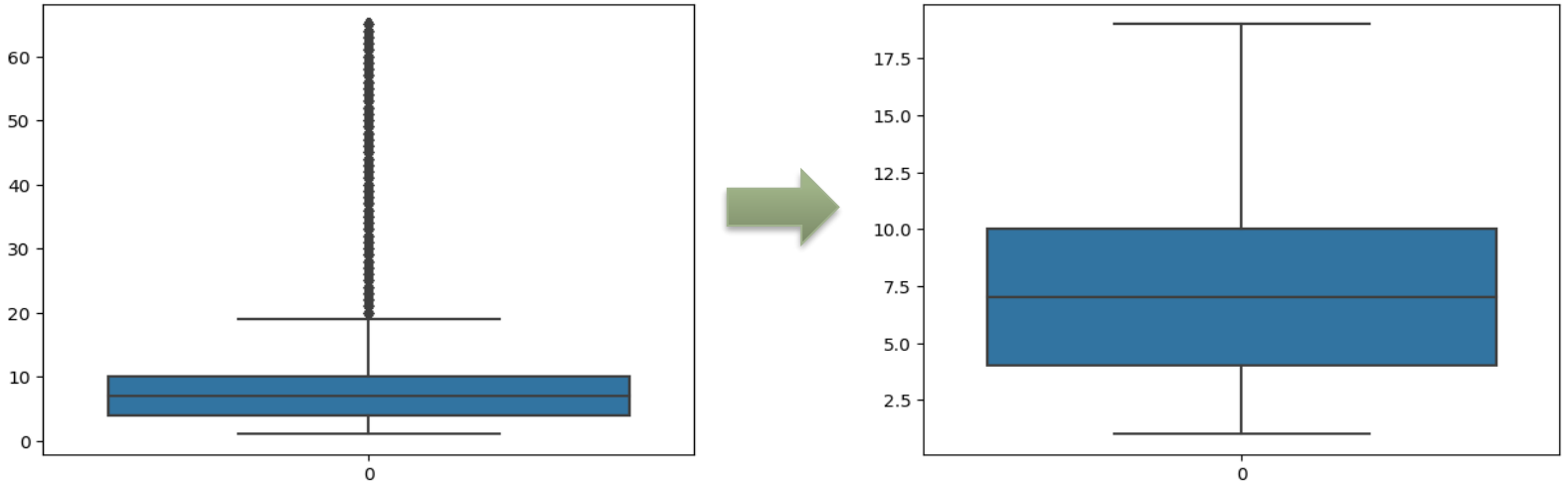
Exploratory Data Analysis (EDA)

Visualising the Outlier Treatment

Prior_purchases Column (Boxplot of before and after treatment)

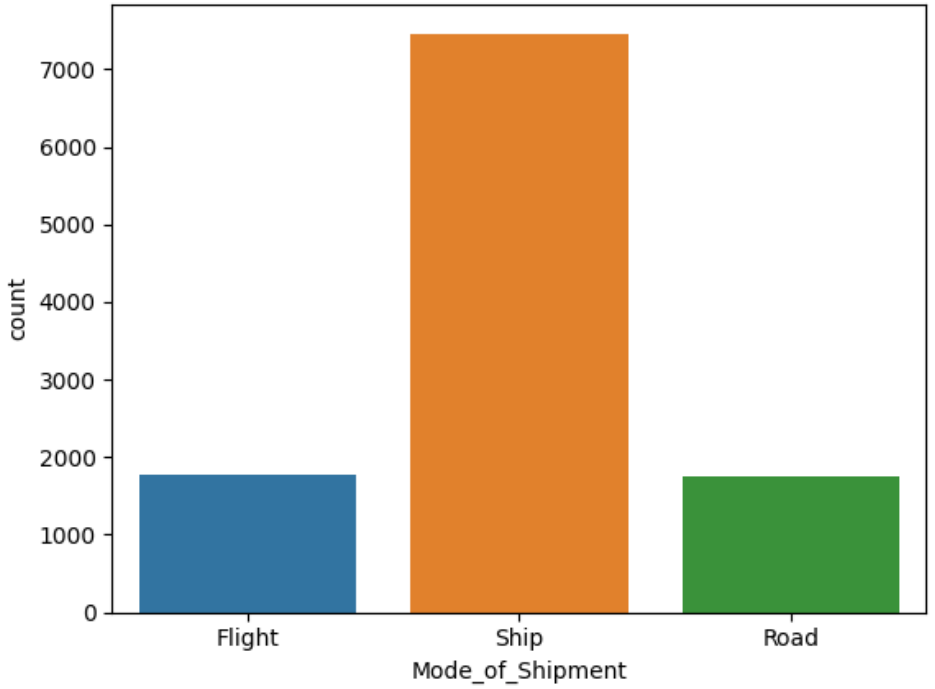


Discount_offered Column (Boxplot of before and after treatment)



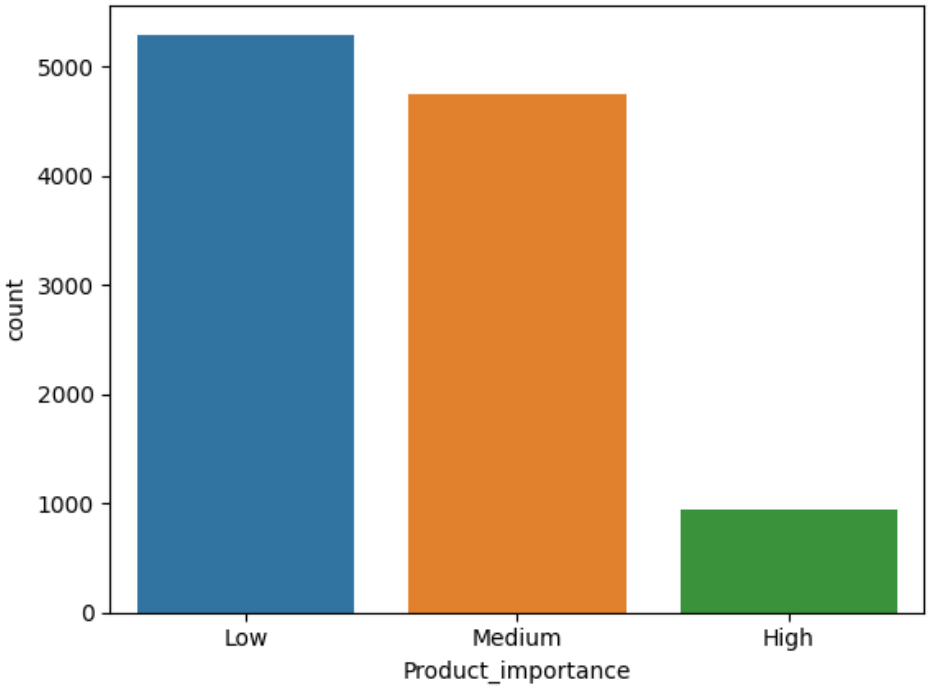
Exploratory Data Analysis (EDA)

Visualizing the mode of shipments



The maximum shipment was made using ships, secondly for flight and the minimum mode is for road

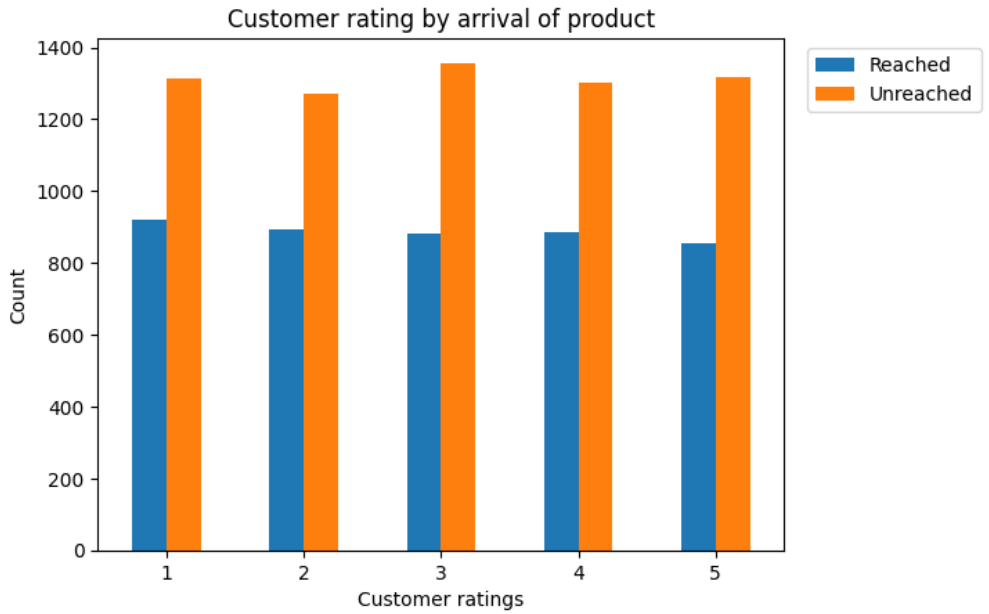
Visualizing the product importance



The maximum count is for low, secondly for medium and the minimum count is for high importance

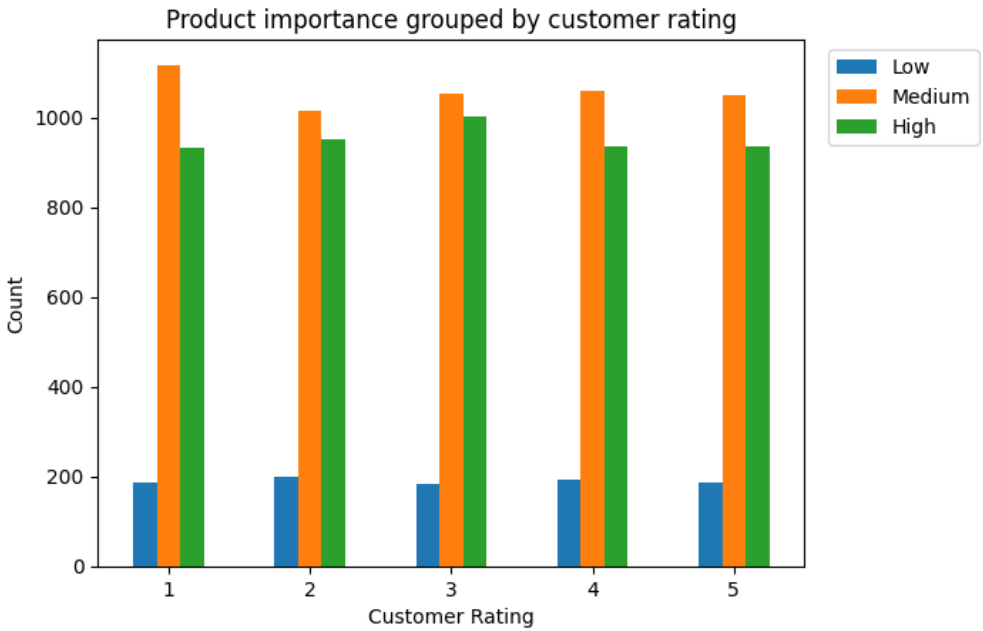
Exploratory Data Analysis (EDA)

Visualizing the Customer Rating against product reached on time



- For all the ratings the unreached counts is between 1200-1400 and the reached counts is between 800-1000
- The highest unreached items had the rating as 3
- The highest reached items had the rating as 1

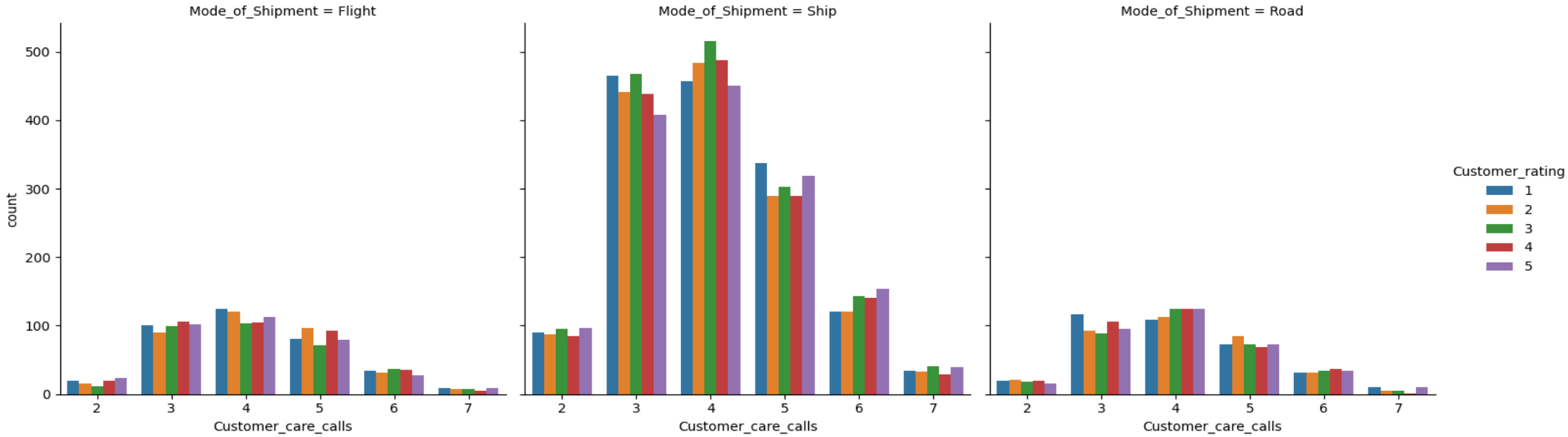
Visualizing different product importance grouped by customer ratings



- In a majority products with medium importance are more and the rating is 1
- Products with highest importance has maximum rating as 3 and with lowest importance has maximum rating as 1

Exploratory Data Analysis (EDA)

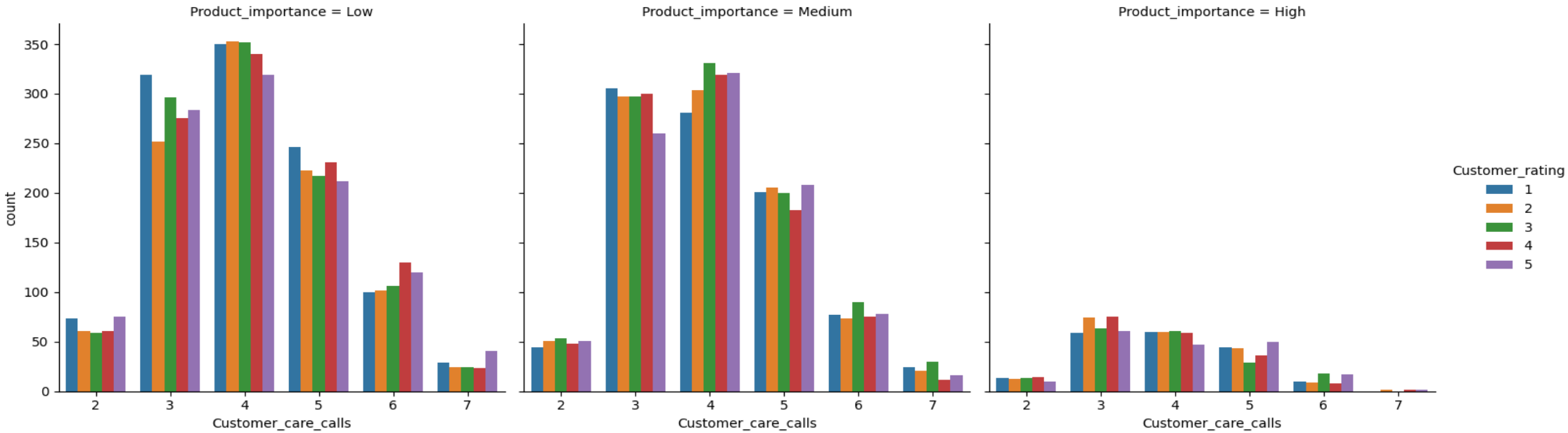
Visualizing each mode of shipment with number of calls made to the customer service and their customer ratings



- for all the modes of shipment the majority of number of calls made to the customer service was 3,4,5.
- for Flight shipment the frequency of high no. of call was 4 and the highest ratings given was 1.
- for Ship shipment the frequency of high no. of call was 4 and the highest ratings given was 3.
- for Road shipment the frequency of high no. of call was 4 and the highest ratings given was 3, 4, 5.

Exploratory Data Analysis (EDA)

Visualizing each product importance with their customer ratings and no. of calls made to the customer service



- for all the product importance entities the majority of number of calls made to the customer service was 3,4.
- for low importance items the frequency of high no. of call was 4 and the highest ratings given was 2.
- for medium importance items the frequency of high no. of call was 4 and the highest ratings given was 3.
- for high importance items the frequency of high no. of call was 3 and the highest ratings given was 4.

Exploratory Data Analysis (EDA)

Visualizing the correlation of different variables to check multi-collinearity

There is a high correlation between:

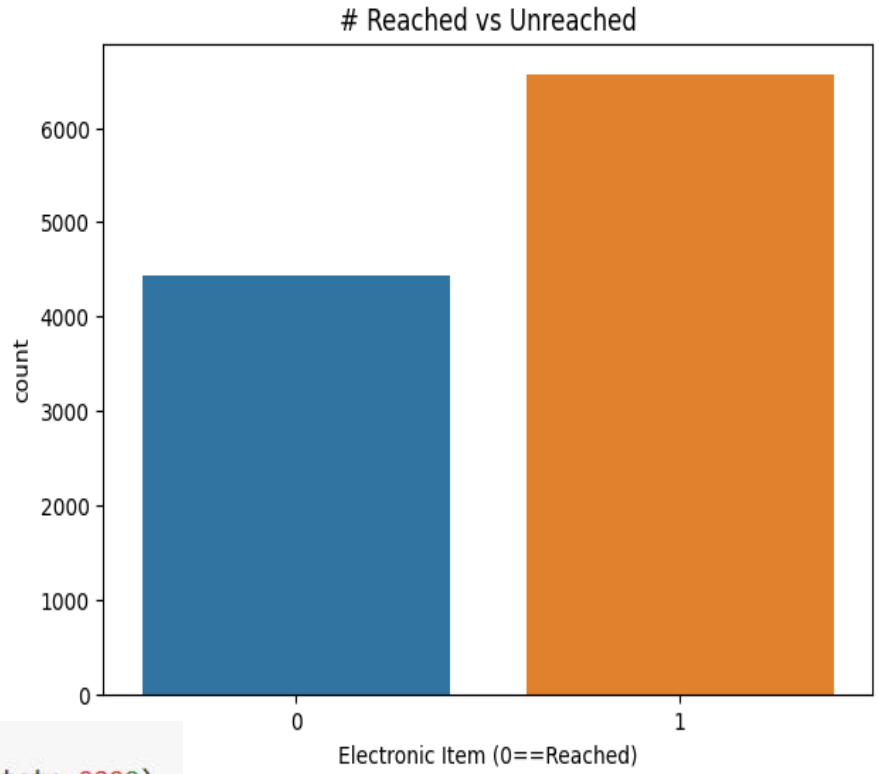
- Weight_in_gms & Discount_offered
- Weight_in_gms & Customer_care_calls
- Weight_in_gms & Reached.on.Time_Y.N
- Weight_in_gms & Prior_purchases

Modelling Approach

- For the dataset, dependent variable (y = output variable) is 'Reached.on.Time_Y.N' column and rest of the columns are independent variables (X = input variable)
- The target variable is dichotomous in nature (0/1). Hence, we will consider Binary classification for model building
- Binary classification is a supervised learning algorithm that categorizes new observations into one of two classes.
- There are several machine learning techniques which depends on factors such as the size and quality of your dataset, the complexity of the problem, and the computational resources.
- Few common techniques I have used to identify the best fit model for the dataset by comparing their results. Below are the techniques which I have used:
  - Naive Bayes Model (Bernoulli Naive Bayes)
  - Logistic Regression
  - K-Nearest Neighbors (KNN)
  - Support Vector Machines (SVM)
  - Decision Tree
  - Boosted Decision Tree
  - Bagging Decision Tree
  - Random Forest Algorithm

Modelling Approach

- Divide data into Input and Output
  - X as Input(Independent variables) and y as Output(Dependent variable).
  - ```
    X = data.drop('Reached.on.Time_Y.N', axis=1).values      #Input
    y = data['Reached.on.Time_Y.N'].values                   #Output
    ```
- Split data into two parts train and test
  - Divide the dataset in 2 parts : Training data(80%) and Testing data(20%)
  - X and y will have XTraining, XTesting and yTraining, yTesting
  - ```
    from sklearn.model_selection import train_test_split
    X_train, X_test, y_train, y_test = train_test_split(X, y, train_size=0.8, random_state=0000)
    ```



- Performance Testing
  - Training and test score
  - Receiver Operating Characteristic Area Under the Curve (ROC-AUC) score
  - The Root Mean Square Error (RMSE) value
  - Cross-validation
  - Confusion Matrix (Correct and Incorrect predictions)
  - Classification Report (Precision and recall – refer the Jupyter notebook)

Model Performance

$$p(x) = P[X = x] = \begin{cases} q = 1 - p & x = 0 \\ p & x = 1 \end{cases}$$

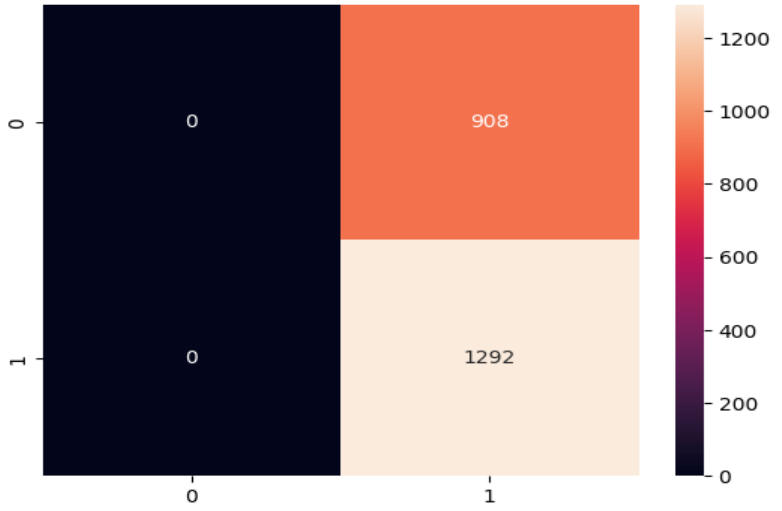
Naive Bayes Model (Bernoulli Naive Bayes)

- Bernoulli Naive Bayes is a part of the Naive Bayes family.
- Based on the Bernoulli Distribution and accepts only binary values, i.e., 0 or 1.
- Success: p and Failure: q
- q = 1 - (probability of Success) = 1 - p

$$X = \begin{cases} 1 & \text{Bernoulli trial} = \mathbf{S} \\ 0 & \text{Bernoulli trial} = \mathbf{F} \end{cases}$$

Result:

- Did not correctly identify any positive cases (TP = 0).
- Did not incorrectly classify any positive cases as negative (FN = 0).
- Model is not sensitive enough to the positive class
- Failing to recognize positive instances.



Evaluation metrics: Defined model name and metrics

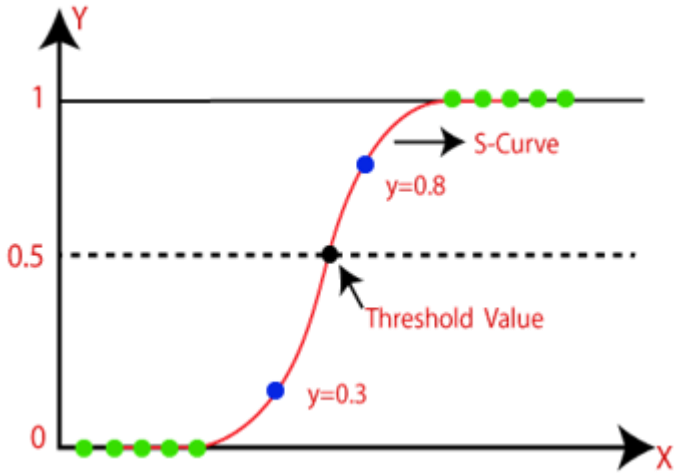
	Train Score	Test Score	ROC AUC Score	RMSE	Cross Validation Score	Correct Predictions	Incorrect Predictions
bnb	59.9045	58.7273	0.5	0.6424	59.669	1292	908

Model Performance

Logistic Regression

- It uses the concept of threshold value
- Values > threshold ~ 1 and values < threshold ~ 0
- The S-form curve is called the Sigmoid/logistic function used to map the predicted values to probabilities

$$p = \frac{1}{1+e^{-(\beta_0+\beta_1 x)}} \quad This\ is\ our\ sigmoid\ function$$



Evaluation metrics: Defined model name and metrics

	Train Score	Test Score	ROC AUC Score	RMSE	Cross Validation Score	Correct Predictions	Incorrect Predictions
lr	63.5981	63.0455	0.6192	0.6079	61.568	1387	813

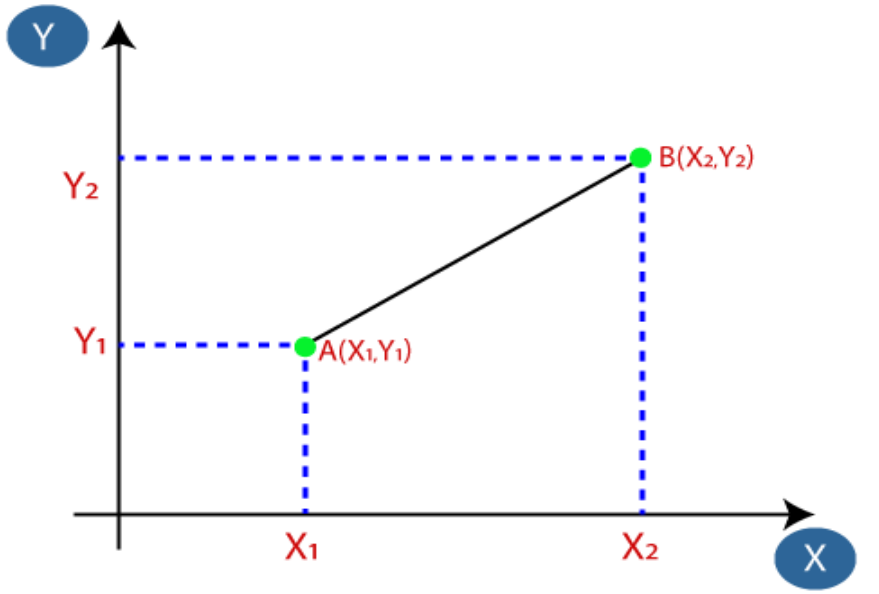
After Standardization (by using StandardScaler)

	Train Score	Test Score	ROC AUC Score	RMSE	Cross Validation Score	Correct Predictions	Incorrect Predictions
model1	63.5186	63.4091	0.6207	0.6049	61.8497	1395	805

Model Performance

K-Nearest Neighbors (KNN):

- Classifies data points based on the majority class among their k nearest neighbors in the feature space.
- When new data appears, it can be easily classified into a well suite category by calculating the distance between the data points for all the training samples.



Euclidean Distance between A_1 and $B_2 = \sqrt{(X_2-X_1)^2+(Y_2-Y_1)^2}$

Evaluation metrics: Defined model name and metrics

model (k=3)

Train Score	Test Score	ROC AUC Score	RMSE	Cross Validation Score	Correct Predictions	Incorrect Predictions
81.9752	64.3182	0.6323	0.5973	63.3225	1415	785

After selecting best k-value (by using GridSearchCV)

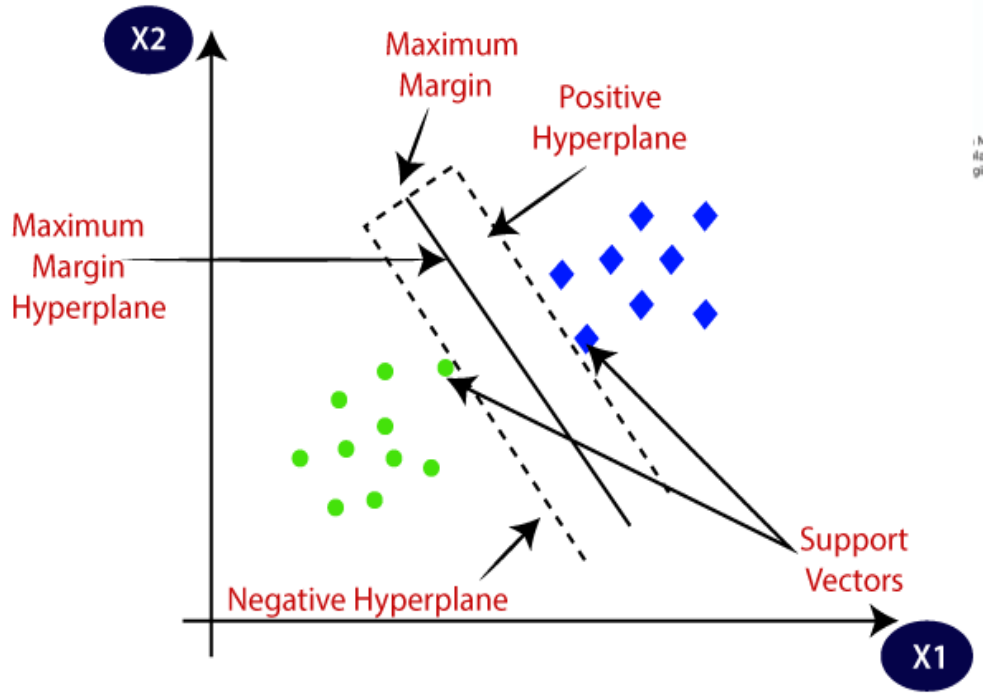
knn (k=48)

Train Score	Test Score	ROC AUC Score	RMSE	Cross Validation Score	Correct Predictions	Incorrect Predictions
69.7011	66.1818	0.6809	0.5815	64.3129	1456	744

Model Performance

Support Vector Machines (SVM):

- It creates the best line or decision boundary that can segregate n-dimensional space into classes in order to put the new data point in the correct category in the future.
- SVMs aim is to find the best decision boundary which is called a hyperplane.



Evaluation metrics: Defined model name and metrics

classifier

Train Score	Test Score	ROC AUC Score	RMSE	Cross Validation Score	Correct Predictions	Incorrect Predictions
66.2575	65.9545	0.6526	0.5834	64.8497	1451	749

After Standardization (by using StandardScaler)

SVMclassifier

Train Score	Test Score	ROC AUC Score	RMSE	Cross Validation Score	Correct Predictions	Incorrect Predictions
70.2125	66.6818	0.6944	0.5772	64.8497	1467	733

Model Performance

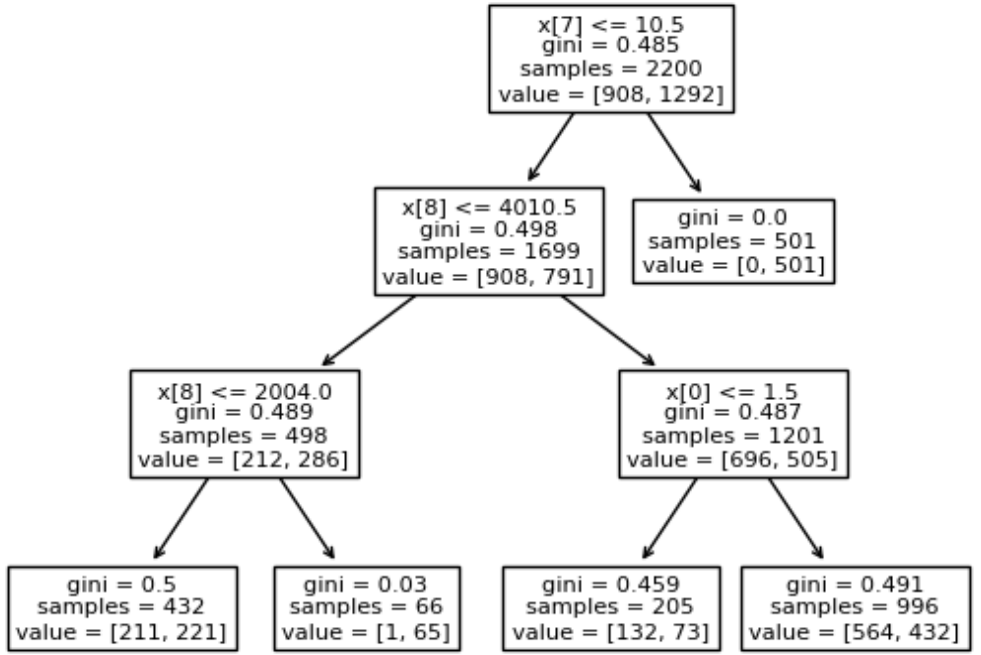
Decision Tree:

- It's is a flowchart-like tree structure where an internal node represents a feature, branch represents a decision rule, and each leaf node represents the outcome.
- Topmost node is the root node. It partitions the tree in a recursive manner called recursive partitioning.

$$Gini = 1 - \sum_{i=1}^{n} p^2(c_i)$$

$$Entropy = \sum_{i=1}^{n} -p(c_i)log_2(p(c_i))$$

where $p(c_i)$ is the probability/percentage of class c_i in a node.



clf(1)

Train Score	Test Score	ROC AUC Score	RMSE	Cross Validation Score	Correct Predictions	Incorrect_predictions
100	64.2273	0.6276	0.5981	63.4042	1413	787

After adding tuning parameters
(criterion="entropy/gini", max_depth=3)

clf(2)

Train Score	Test Score	ROC AUC Score	RMSE	Cross Validation Score	Correct Predictions	Incorrect_predictions
67.9623	67.4091	0.7138	0.5708	64.2491	1483	717

Model Performance

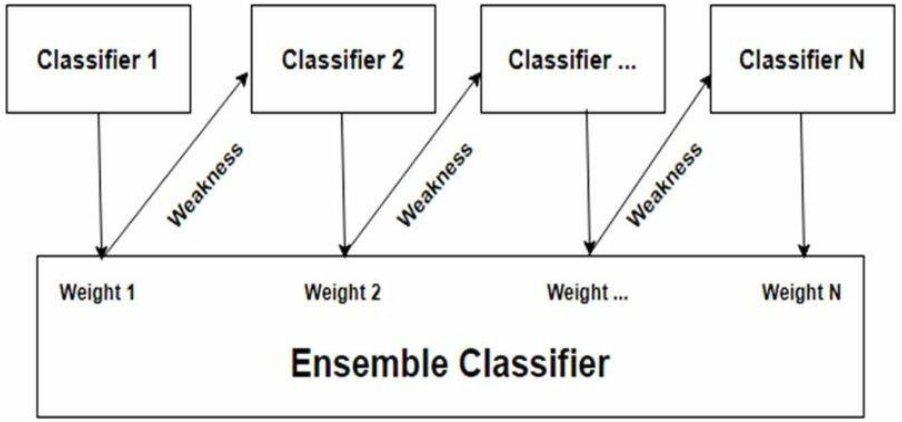
Ensemble algorithms in order to improve the accuracy with base model as Decision Tree

- It helps improve machine learning results by combining several models.
- This approach allows the production of better predictive performance compared to a single model.
- The most common Ensemble learning which we have used are as follow:
  - **Boosting (AdaBoost)**
    - AdaBoost is a good choice when you have a weak base learner and want to boost its performance.
    - It can work well in binary classification, particularly when you have a diverse set of weak learners.
  - **Bagging**
    - Bagging can be a good choice when you want to reduce variance and overfitting in your base model.
    - It's a simple yet effective ensemble technique.
  - **Random Forest**
    - Random Forest is a versatile and robust ensemble method that works well in a wide range of situations.
    - It can handle both numerical and categorical data, is less prone to overfitting, and typically doesn't require extensive hyperparameter tuning.

Model Performance

Adaptive boosting algorithm (AdaBoost)

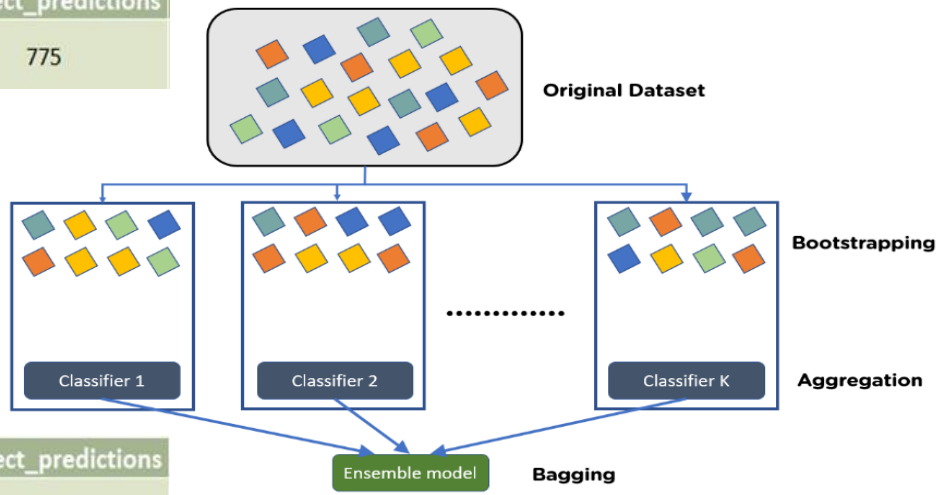
- It involves using very short (one-level) decision trees that are added sequentially to the ensemble which attempts to correct the predictions made by the model.
- This is achieved by weighing the training dataset, focusing on training examples on which prior models had prediction errors.
- After adding parameters still the model had under-fitting problem



boost_ model

Train Score	Test Score	ROC AUC Score	RMSE	Cross Validation Score	Correct Predictions	Incorrect_predictions
79.0431	64.7727	0.6423	0.5935	62.9132	1425	775

- **Bagging Algorithm (Bootstrap Aggregating)**
  - Bootstrapping is the method of randomly creating samples of data out of a population with replacement to estimate a population parameter.



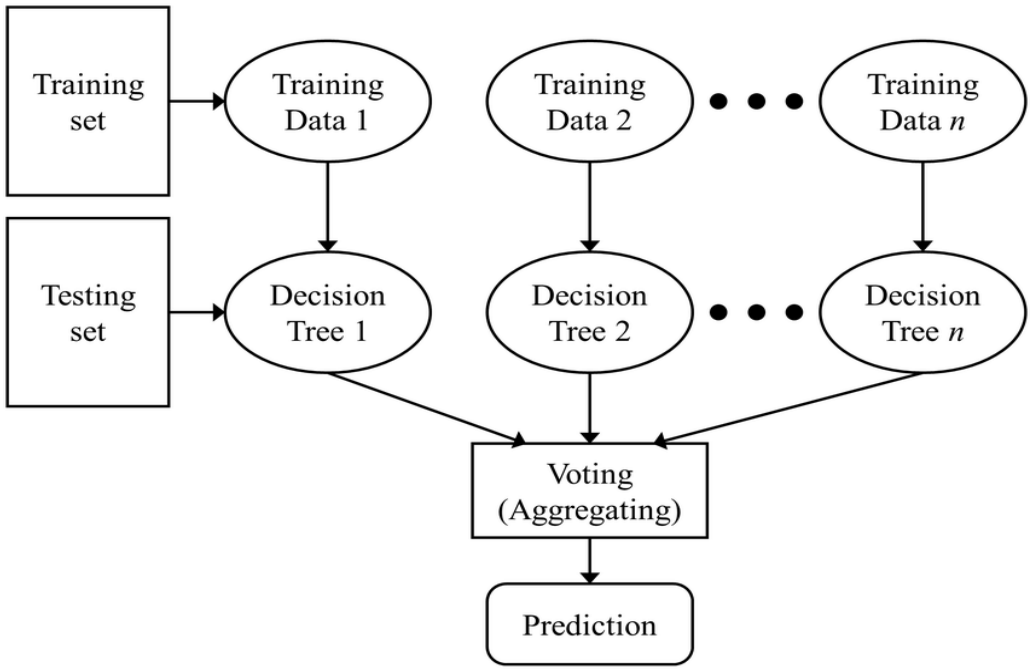
bag_ model

Train Score	Test Score	ROC AUC Score	RMSE	Cross Validation Score	Correct Predictions	Incorrect_predictions
67.8373	67.3636	0.7137	0.5712	65.4674	1482	718

Model Performance

Random Forest Algorithm:

- It combines the predictions of multiple decision trees to improve accuracy and generalization.
- The greater number of trees in the forest leads to higher accuracy and prevents the problem of overfitting.



Evaluation metrics: Defined model name and metrics

rfc

Train Score	Test Score	ROC AUC Score	RMSE	Cross Validation Score	Correct Predictions	Incorrect_predictions
100	64.4091	0.648	0.5965	63.1222	1417	783

After adding tuning parameters by using Gridsearch

Rfclassifier

Train Score	Test Score	ROC AUC Score	RMSE	Cross Validation Score	Correct Predictions	Incorrect_predictions
69.1556	68.2273	level of discriminat	0.5636	65.6581	1501	699

Insights and Findings

- Summarizing the key findings
  - Five classification machine learning models were implemented.
  - Out of all Decision Tree gave the best results.
  - Three ensemble learning algorithm was implemented with base model as Decision Tree.
  - Out of all ensemble algorithm, Random Forest algorithm gave the best result.
- Actionable insights derived from the data
  - In the first phase of classification models underwent issues like,
    - Overfitting problem
    - Underfitting problem
  - To overcome the issues we used Standardization, GridSearch for identifying the best tuning parameters, hyper-tuning

Recommendations

- In Output variable ‘0’ indicates order/item reached on time and ‘1’ indicates not reached on time.
- Count for ‘0’ is 4436 and for ‘1’ is 6563.
- Increase the count of items reached on time. It can be done by following ways:
  - **Capacity Planning**: determine the maximum number of items that can be processed within a given timeframe.
  - **Demand Forecasting**: anticipate future demand which can help to plan and allocate resources more effectively.
  - **Efficient Processes**: Optimize your production or service processes to maximize throughput.
  - **Inventory Management**: Maintain optimal level of inventory to meet demand without overstocking/understocking.
  - **Resource Allocation**: Ensure to have the right mix of resources, labor, equipment, materials, to meet service goals.
  - **Monitoring and Reporting**: Implement real-time monitoring systems to track service progress and identify issues.
  - **Supply Chain Management**: Collaborate closely with suppliers and logistic partners for smooth & timely supply chain.
  - **Scenario Planning**: Develop scenarios & contingency plans allowing to respond quickly to changing market conditions.
  - **Collaboration and Communication**: Foster open communication and collaboration among teams and departments to ensure everyone is aligned with production or service goals and can adapt to changing circumstances.

Challenges and Limitations

- Challenges
  - Selection of Machine learning models.
  - Finding the issues and rectifying the results.
  - Noisy data, skewness and kurtosis.
  - Changing categorical data into numerical data

- Limitations
  - Applied standardization at the start during EDA.
  - Weight_in_gms, Cost_of_the_Product and Discount_offered have greater values which can be standardized.
  - More columns could have been eliminated which doesn’t provide any insights for result.

Future Work

- Analysis can be extended by splitting the data in different ratio. And then compare the results.
- Implementing more machine learning techniques by including deep learning too.
- Visualizing every possible scenarios.
- Analysis can be improved by using GridSearch algorithm for choosing the best parameter for hyper-tuning at the start of each machine learning models.
- Using stacking ensembled learning algorithm.

Conclusion

- After analyzing all the models, we can conclude:
  - Out of all the models the Random Forest algorithm with base model as Decision tree gives the best result.
  - The accuracy is optimized.
  - Better and clearer predictions with highest precision, recall and f1-score.

```
Classification Report :
              precision    recall  f1-score   support

           0       0.57      0.92      0.71       908
           1       0.91      0.51      0.65      1292
```

  - Greater correct predictions and lesser incorrect predictions
  - Minimum RMSE value
  - High Cross Validation score

Appendix

- Ecommerce dataset

	ID	Warehouse_block	Mode_of_Shipment	Customer_care_calls	Customer_rating	Cost_of_the_Product	Prior_purchases	Product_importance	Gender	Discount_offered	Weight_in_gms	Reached.on.Time_Y.N
2	1	D	Flight	4	2	177	3	low	F	44	1233	1
3	2	F	Flight	4	5	216	2	low	M	59	3088	1
4	3	A	Flight	2	2	183	4	low	M	48	3374	1
5	4	B	Flight	3	3	176	4	medium	M	10	1177	1
6	5	C	Flight	2	2	184	3	medium	F	46	2484	1

- Statistical Data

	ID	Customer_care_calls	Customer_rating	Cost_of_the_Product	Prior_purchases	Discount_offered	Weight_in_gms	Reached.on.Time_Y.N
count	10999.00000	10999.000000	10999.000000	10999.000000	10999.000000	10999.000000	10999.000000	10999.000000
mean	5500.00000	4.054459	2.990545	210.196836	3.567597	13.373216	3634.016729	0.596691
std	3175.28214	1.141490	1.413603	48.063272	1.522860	16.205527	1635.377251	0.490584
min	1.00000	2.000000	1.000000	96.000000	2.000000	1.000000	1001.000000	0.000000
25%	2750.50000	3.000000	2.000000	169.000000	3.000000	4.000000	1839.500000	0.000000
50%	5500.00000	4.000000	3.000000	214.000000	3.000000	7.000000	4149.000000	1.000000
75%	8249.50000	5.000000	4.000000	251.000000	4.000000	10.000000	5050.000000	1.000000
max	10999.00000	7.000000	5.000000	310.000000	10.000000	65.000000	7846.000000	1.000000

References

- Machine learning models are implemented in Jupyter Notebooks
- The flowcharts are downloaded from
  - www.javatpoint.com
  - www.datacamp.com
  - www.kaggle.com
- The Dashboard is implemented in Tableau. Below is the link for accessing the responsive E-commerce Analysis Dashboard,
  - https://public.tableau.com/app/profile/shweta.purushothaman/viz/ShwetaPurushothaman_E-commerce_Capstone/Dashboard1?publish=yes

Thank you !!!