Flipkart Products Analysis

A Thesis submitted to

Great Lakes Institute of Management

In partial fulfilment of requirements for

the award of

Post Graduate Programme in Data Science and Engineering

By

Surya Deepthi Dupati

Saikumar Menta

Sadasivuni Datta Sai Teja Mitra

Bala Manikanta Sai Srinivas Dhommeti

Under guidance of

Jayveer Nanda

Table of Contents

Contents

Problem Statement **3**

Data set and Data Description **4**

Project Methodology **5**

Pre Processing Data Analysis **5**

Exploratory Data Analysis and Business Insights **8**

Feature Engineering(Statistical Testing) **12**

SMOTE Analysis **15**

Model Building **15**

Peformance Metrics **21**

Business Justification **22**

Hyper Parameters **22**

Hyper Parameter Tuning **23**

Boosting Algorithms **24**

Conclusion: **24**

References: **25**

PROBLEM STATEMENT

These days e-commerce domain is a rapid and emerging domain. Similarly, there are lot many other competitors emerging with better strategies. To give a strong competition Flipkart comes up with the strategy of Flipkart advantage, which means it is a service provided by Flipkart for their sellers. The service is warehouse space, logistics, packaging. The major advantage here is that Flipkart can provide customers with faster delivery and quality assurance. This is given by the tag Flipkart Advantage. There are filters on the search page that will help customers to select from these products. Flipkart Assured enables customers to enjoy a higher standard of shopping and faster, hassle-free shipping.

The problem here faced by Flipkart is the storage of the products sent by sellers. Not all the products in the warehouse sell better. Not every seller is a good seller. Not every product is a good product. So, an algorithm needs to be built that predict is the product is eligible for the Flipkart advantage tag or not so that the product can be stored in the warehouse. Analyzing the dataset that contains the information like sales, pricing, brand and product specifications and predicting the products which fall under Flipkart Advantage Tag.

PROJECT OUTCOME

The outcome of our project is to build a robust machine learning algorithm that helps the business client incorrectly classifying a given product as Flipkart Advantage or not, that computes an array of data inputs including sales, pricing, brand value, product specifications etc.

INDUSTRY REVIEW

More than 1 billion people have shopped using Flipkart, making the e-commerce giant one of the most popular and trending e-retailers. Flipkart uses an undifferentiated targeting strategy, since people of all demography purchase items online which is available to everyone where the delivery is possible. National & Multinational E-commerce companies are giving neck-to-neck competition to each other, due to which their positioning is very important. Flipkart has positioned itself as a trustworthy and customer-friendly E-commerce brand.

The online retail industry market is of a size of around 60 billion USD. It is expected to reach 200 billion by the year 2026. The Indian and global e-commerce industry is on the verge of exponential growth, and the introduction of high-speed internet has fueled the process across the nation.

DATA SET

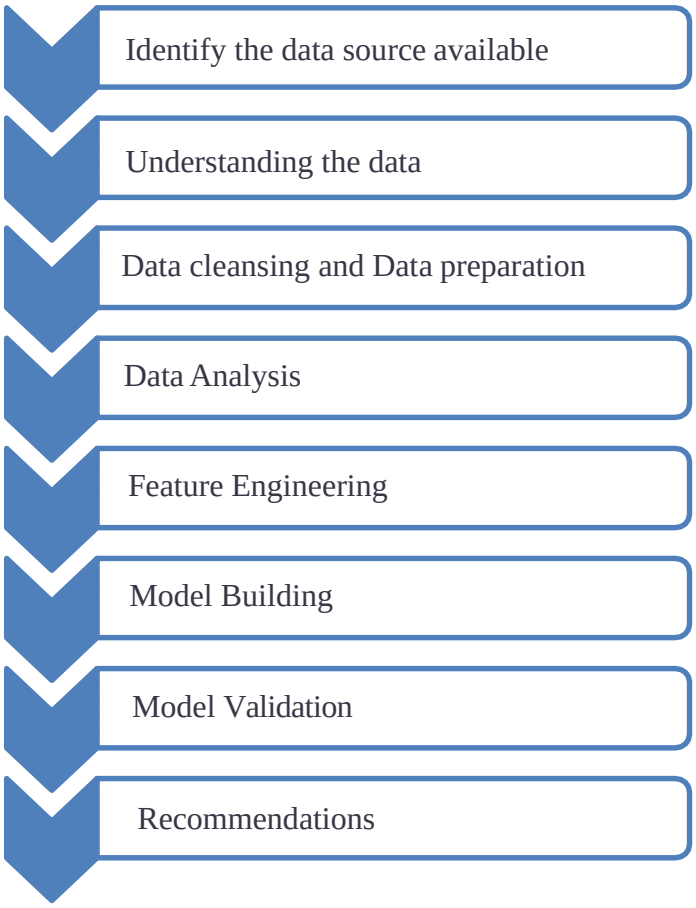
- A dataset is a collection of data, and it can be structured or unstructured.
- A structured data is represented in a tabular format, where every column of the table represents a particular variable, and each row corresponds to a given record of the dataset.
- Unsupervised/unstructured data is not represented in a tabular form, data that we fetch from Facebook, Twitter, and Netflix, etc.

DATA DESCRIPTION

This is a pre-crawled dataset, taken as subset of a bigger dataset (more than 5.8 million products) that was created by extracting data from Flipkart.com, a leading Indian eCommerce store. The dataset has 20000 rows with 15 features. Refer to the below-detailed structure of the dataset.

Variable Name	Variable Description
product URL	The data consists of the web address to the Products
Unique ID	A unique ID will be given by the website
ProductName	The data consists of Names of the Products
crawl timestamp	The time of the crawl
Pid	Product ID
Product category tree	The data will show the category of which the Product belongs to
retail price	The data shows the Retail Prices of the Products
discounted price	The data shows the Discounted Prices of the Products
image	The data shows the images of the Products
isFKAdvantage_product	The data helps to represent whether the product is an advantage to Flipkart
description	The data describes the product details
product rating	The data talks about the rating of the product
overall rating	The data talks about the overall rating of the product
brand	The data represents the Brand to which the Product belongs to
product specifications	The data represents the product specifications

PROJECT METHODOLOGY



PRE-PROCESSING DATA ANALYSIS

Data Preparation:

Data preprocessing is a crucial step that helps enhance the quality of data to promote the extraction of meaningful insights from the data. Data preprocessing in Machine Learning refers to the technique of preparing (cleaning and organizing) the raw data to make it suitable for building and training Machine Learning models.

Data Preparation is the process of collecting, cleaning, and consolidating data into one file or data table, primarily for use in the analysis.

Acquire the dataset:

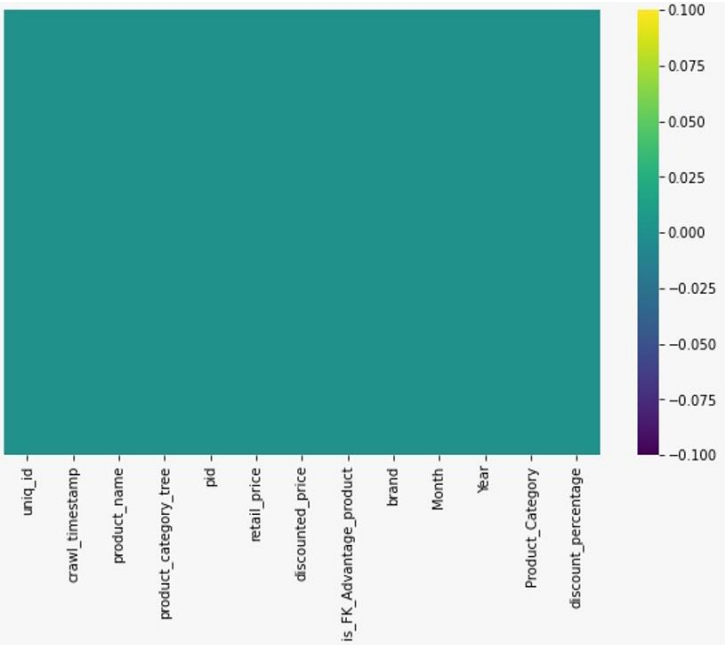
https://www.kaggle.com/PromptCloudHQ/flipkart-products

Missing/Null Values:

Impute or drop features with missing values based on the percentage of missing values and relevance for model building.

In the dataset retail price, discounted price, description features have null values less than 10%. So, we have dropped the rows. The null values of brand features are imputed by product name. Since the brand feature has null values of around 30%.

The product rating and overall rating features have null values of more than 90%. So, we have dropped both features.



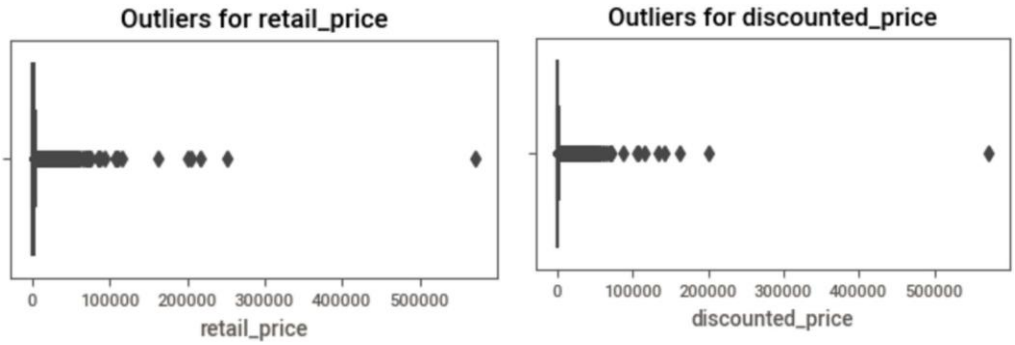

```
# checking for null values
df.isnull().sum()/len(df)*100
```

uniq_id	0.0
crawl_timestamp	0.0
product_url	0.0
product_name	0.0
product_category_tree	0.0
pid	0.0
retail_price	0.0
discounted_price	0.0
image	0.0
is_FK_Advantage_product	0.0
description	0.0
brand	0.0
product_specifications	0.0
Month	0.0
Year	0.0
Product_Category	0.0

dtype: float64

OUTLIERS:

Outliers are the extreme that deviates from other observations on data, they may indicate variability in measurement, experimental errors or a novelty.



The outliers are present in the retail price and discounted price column. As the prices are significantly unique to each product category, which ranges from very low value to very high value so, we are not excluding the outliers instead we would implement a transformation technique considering the outliers thereby reducing the skewness.

REDUNDANT COLUMNS

The below are the redundant features that have all the unique values and are dropped from the dataset.

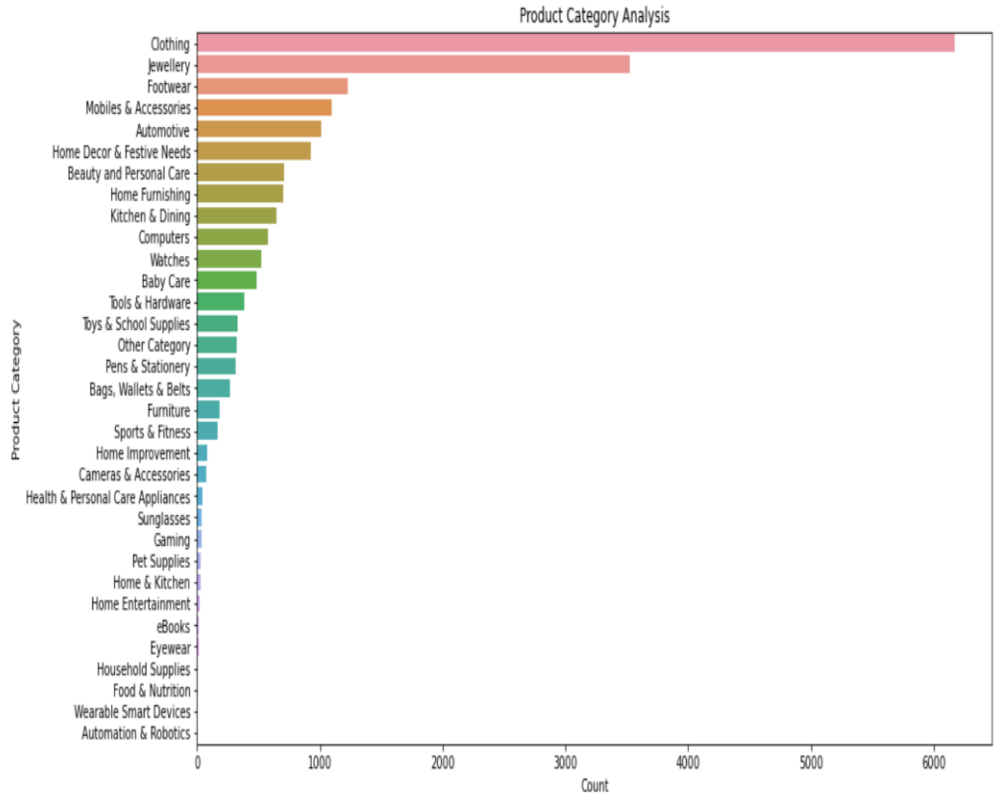
- pid
- product url
- image
- description
- product specification

EXPLORATORY DATA ANALYSIS & BUSINESS INSIGHTS

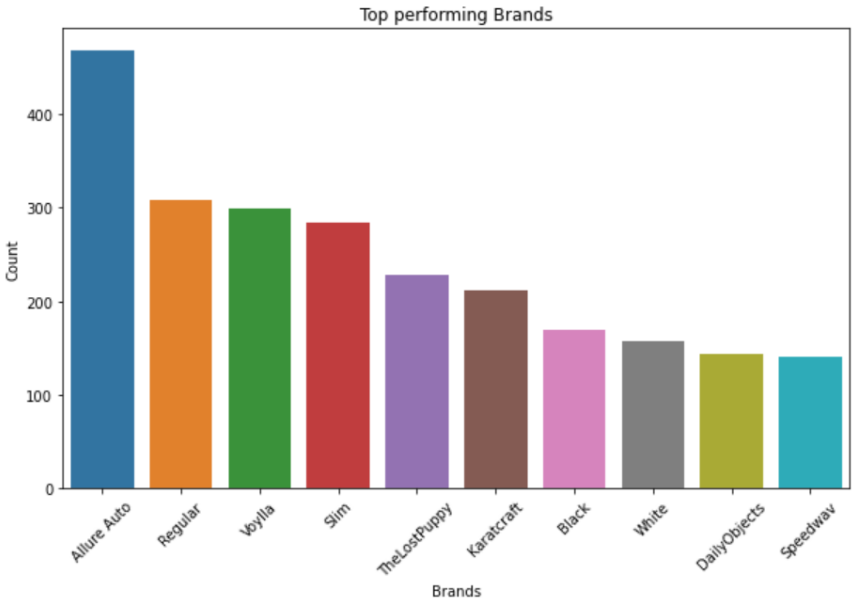
Univariate analysis:

```
# Top ten high performing products based on sales count

df['product_name'].value_counts().head(10)
```

```
TheLostPuppy Back Cover for Apple iPad Air                          134
TheLostPuppy Back Cover for Apple iPad Air 2                         95
S4S Stylish Women's Push-up Bra                                      94
Voylla Metal, Alloy Necklace                                         66
WallDesign Small Vinyl Sticker                                       65
HomeeHub Polyester Multicolor Self Design Eyelet Door Curtain        58
DailyObjects Back Cover for Apple iPad 2/3/4                         52
Nimya Solid Men's Polo Neck T-Shirt                                  50
S4S Comfortable Women's Full Coverage Bra                            45
Grafion by Grafion - Comfort Feel Women's Full Coverage Bra          44
Name: product_name, dtype: int64
```

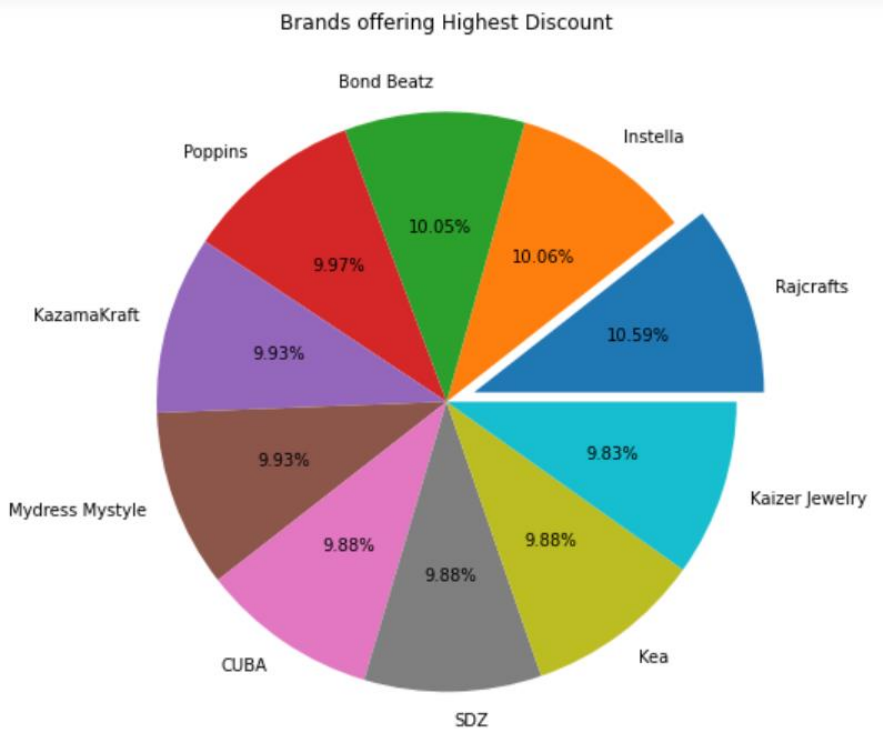


As per the above analysis we infer that most of the products fall under clothing



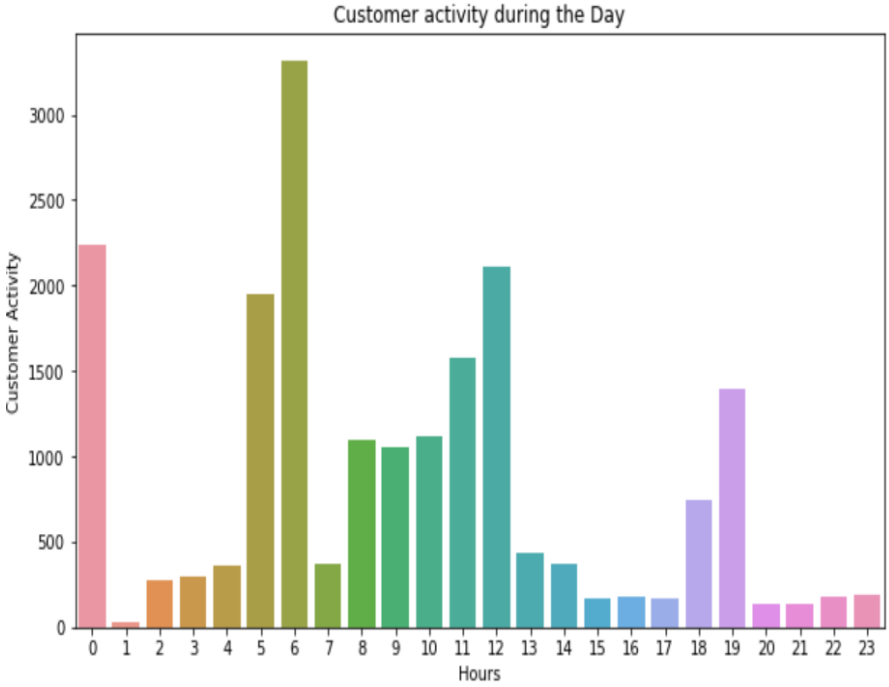
Allure Auto is the highest-selling brand compared to other brands.

Bivariate analysis:

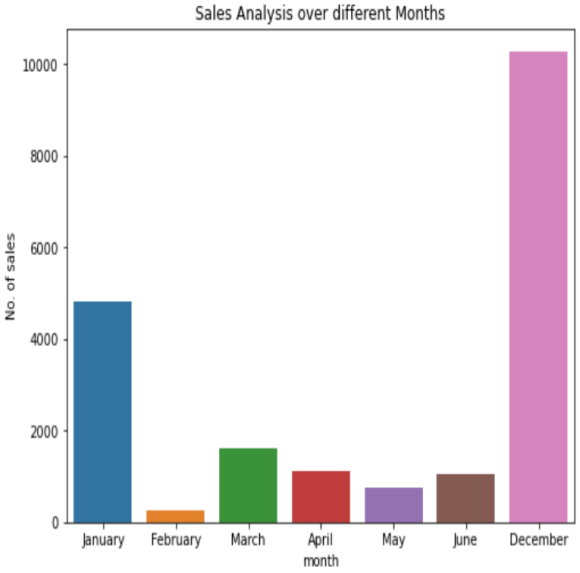


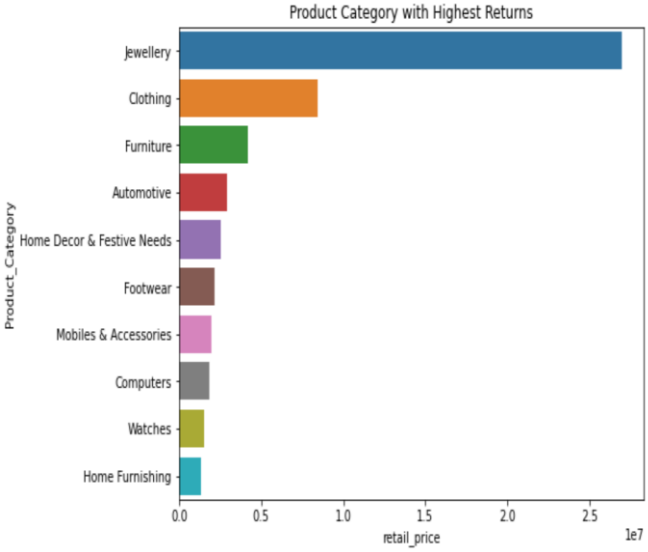
Raj crafts brand is providing the highest discounts compared to other brands.

The retail and discount prices are very high in the month of January and decline in Febru ary and May.



Most of the customers are actively purchasing at day time when compare to nig ht times.

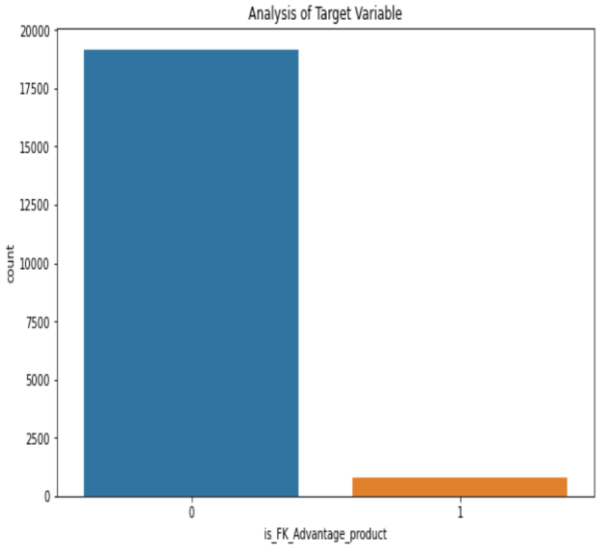




December month having drastically highest number of orders compared to other months

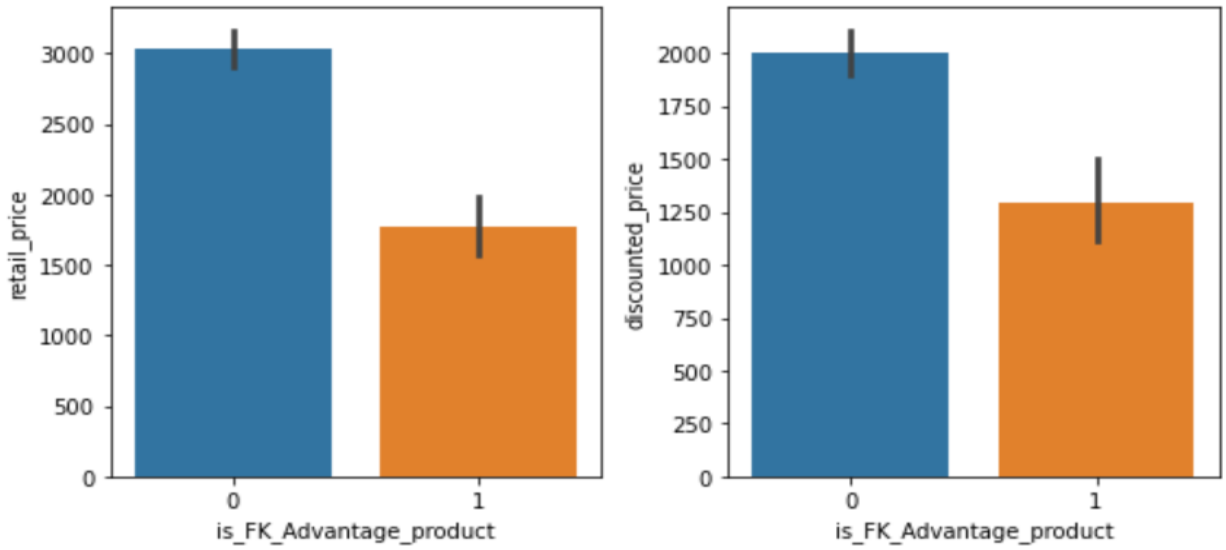
Jewellery category having highest returns when compared to other categories

Univariate analysis of target feature:



By analyzing the Target feature the data is an imbalance

Bi-variate analysis with target feature:



By the above analysis, we infer that variance exists with the target variable. Hence the retail price and discounted price features are contributing significantly in predicting the target variable.

FEATURE ENGINEERING (STATISTICAL TESTING):

Two sample independent t-test:

- The two-sample t-test is used to compare the equality of means of two populations for unpaired data.
- The hypothesis to test whether the population means are equal

$H_{0:}\ \mu_1 = \mu_2$ against $H_a: \mu_1 \neq \mu_1$

It implies

H_0: The two-population means are equal (i.e., $\mu_1 = \mu_2$) against

H_a: The two-population means are not equal μ_0 (i.e., $\mu_1 \neq \mu_2$)

Two sample independent t-test on month and year features:

```
# Two sample independent T test
ttest_ind(features['Month'],features['is_FK_Advantage_product'])
```

```
Ttest_indResult(statistic=213.89756936942086, pvalue=0.0)
```

```
# Two sample independent T test
ttest_ind(features['Year'],features['is_FK_Advantage_product'])
```

```
Ttest_indResult(statistic=530906.0245021164, pvalue=0.0)
```

- It is observed that the pvalue is 0.0 which is less than 0.05 significance level, hence Ho is rejected and Ha is selected.
- There is a significant effect of month and year columns on the target variable, hence considering both features for analysis.

Two sample independent t-test on retail and discounted price features:

```
# Two sample independent T test
ttest_ind(features['retail_price'],features['is_FK_Advantage_product'])
```

```
Ttest_indResult(statistic=46.646199763257286, pvalue=0.0)
```

```
# Two sample independent T test
ttest_ind(features['discounted_price'],features['is_FK_Advantage_product'])
```

```
Ttest_indResult(statistic=37.962744465365965, pvalue=8.2338412501065e-310)
```

- It is observed that the pvalue is 0.0 which is less than 0.05 significance level, hence Ho is rejected and Ha is selected.
- There is a significant effect of the retail price and discounted price features on the target variable, hence considering both features for analysis.

Chi-Square test of Independence:

- The Chi-Square test of independence is used to determine if there is a significant relationship between two nominal (categorical) variables.
- The hypothesis to test the independence of attributes

H_0: The attributes are independent against

H_a: The attributes are dependent

Chi-square contingency test on Brand and Product category features:

```
# cross tabulation
obs=pd.crosstab(features['is_FK_Advantage_product'],features['brand'])
stats.chi2_contingency(obs)
```

```
(10857.321825838919,
 0.0.
```

```
# cross tabulation
obs=pd.crosstab(features['is_FK_Advantage_product'],features['Product_Category'])
stats.chi2_contingency(obs)
```

```
(1755.935857445103,
 0.0.
```

- It is observed that the pvalue is 0.0 which is less than 0.05 significance level, hence Ho is rejected and Ha is selected.
- There is a significant effect of Brand and Product Category columns on the target variable, hence considering both features for analysis.

Splitting the data into 70:30 training and testing proportions:

```
X_train, X_test, y_train, y_test=train_test_split(X,y,test_size=0.3,random_state=10)
```

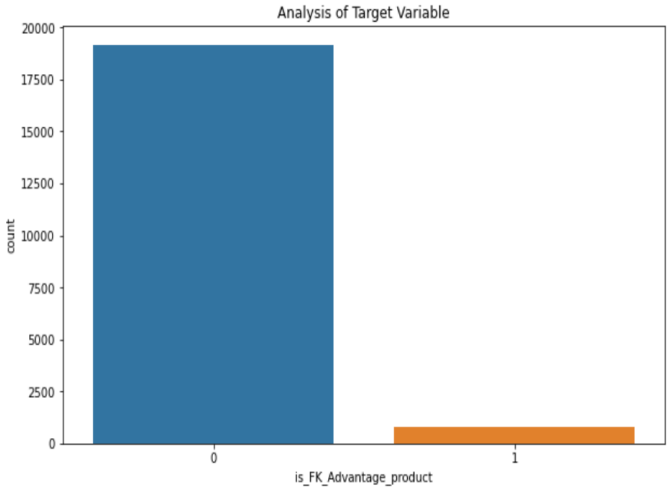
Normalization of training and testing data:

- Feature scaling is a method used to normalize the range of independent variables or features of data. In data processing, it is also known as data normalization.

```
sc=StandardScaler()
X_train_sc=pd.DataFrame(sc.fit_transform(X_train),columns=X_train.columns)
X_test_sc=pd.DataFrame(sc.transform(X_test),columns=X_test.columns)
```

SMOTE ANALYSIS:

- SMOTE (synthetic minority oversampling technique) is one of the most commonly used oversampling methods to solve the imbalance problem.
  It aims to balance class distribution by randomly increasing minority class examples by replicating them.

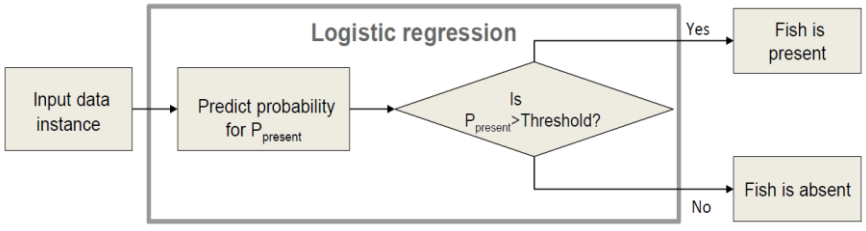


The data is an imbalance

MODEL BUILDING:

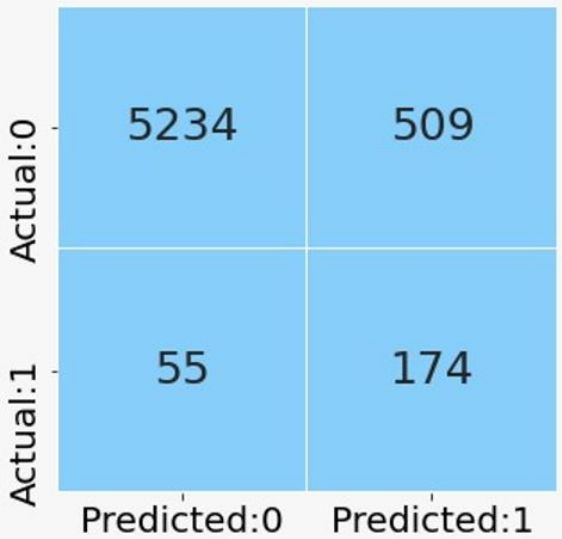
Logistic regression:

- Logistic Regression is a binary classification algorithm. It predicts the probability of occurrence of a label class.
- Consider that logistic regression is used to identify whether the product falls under the advantage category or not.

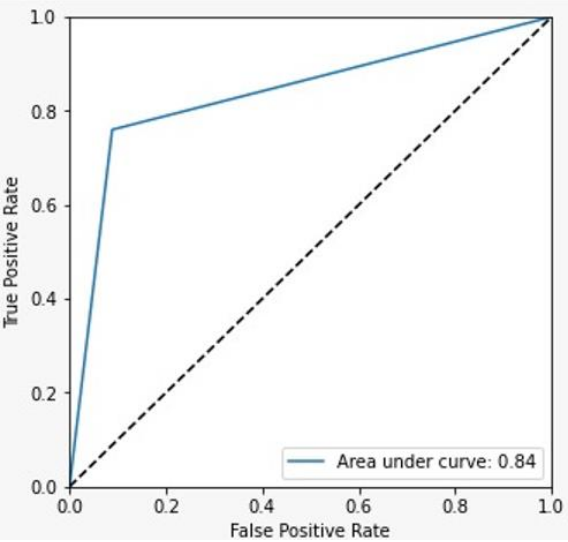



```
LR=LogisticRegression(random_state=10)
model_evaluation('Logistic',LR,X_train_smt,y_train_smt,X_test_sc,y_test)
```

Confusion Matrix:



ROC Curve:



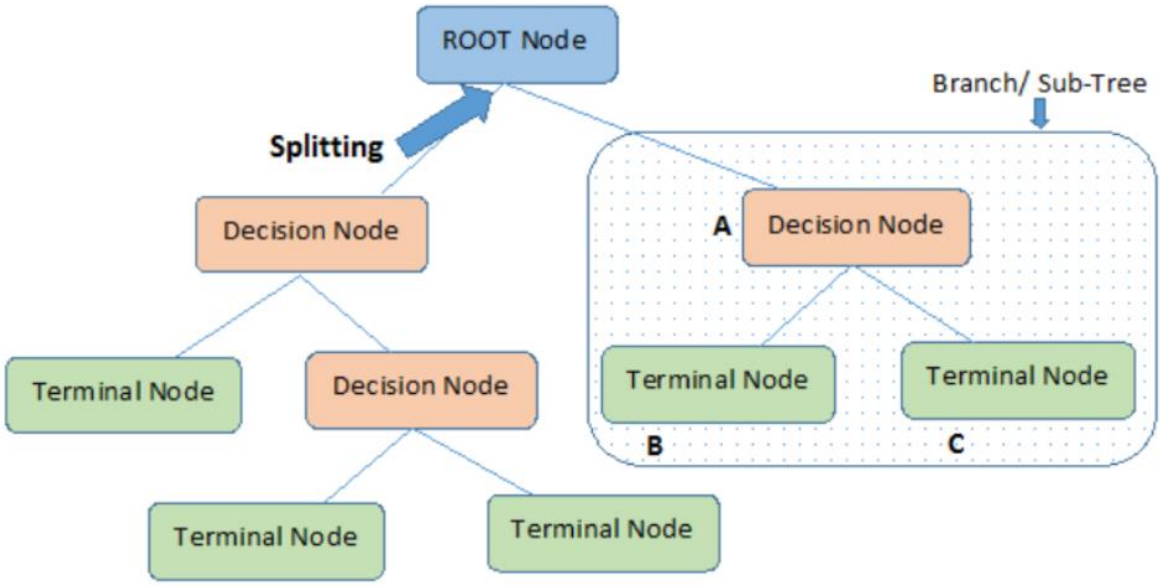
Classification Report:

Classification report:

	precision	recall	f1-score	support
0	0.99	0.91	0.95	5743
1	0.25	0.76	0.38	229
accuracy			0.91	5972
macro avg	0.62	0.84	0.67	5972
weighted avg	0.96	0.91	0.93	5972

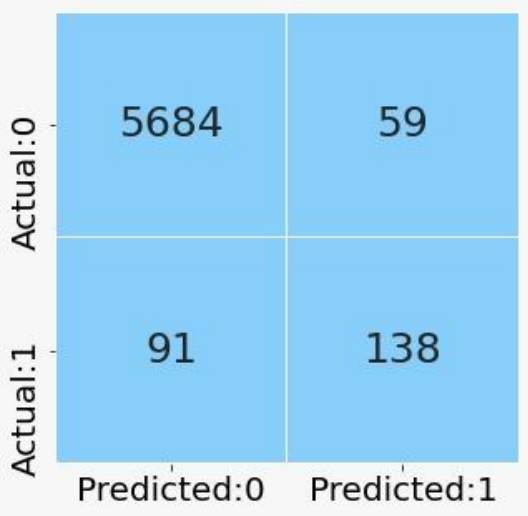
Decision Tree Algorithm:

- Decision trees can be used for classification as well as regression problems.
- The name itself suggests that it uses a flowchart like a tree structure to show the predictions that result from a series of feature-based splits.
- It starts with a root node and ends with a decision made by leaves.

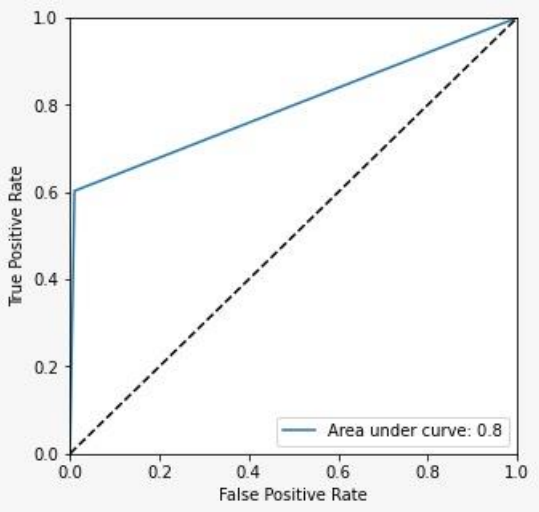



```
DT=DecisionTreeClassifier(random_state=10)
model_evaluation('DecisionTree',DT,X_train_smt,y_train_smt,X_test_sc,y_test)
```

Confusion Matrix:



ROC Curve:



Classification Report:

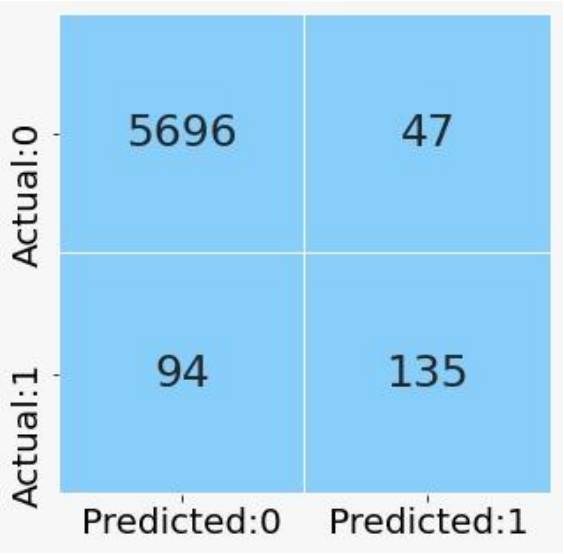
```
Classification report:

              precision    recall  f1-score   support

           0       0.98      0.99      0.99      5743
           1       0.70      0.60      0.65       229

    accuracy                           0.97      5972
   macro avg       0.84      0.80      0.82      5972
weighted avg       0.97      0.97      0.97      5972
```

Random Forest Algorithm:

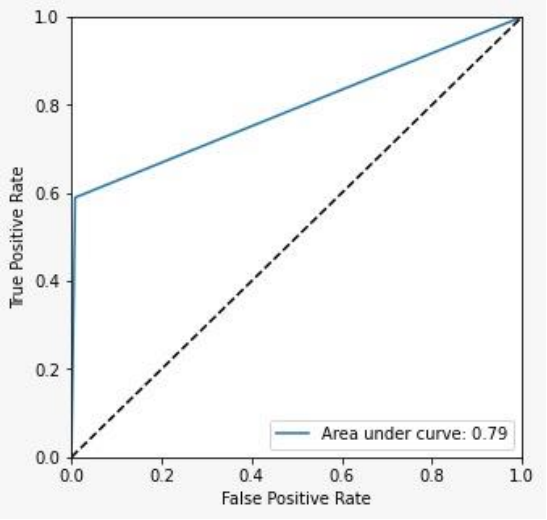
- Random Forest consists of several independent decision trees that operate as an ensemble.
- It is an ensemble learning algorithm based on bagging.

```
RF=RandomForestClassifier(random_state=10)
model_evaluation('RandomForest',RF,X_train_smt,y_train_smt,X_test_sc,y_test)
```

Confusion Matrix:



ROC Curve:



Classification Report:

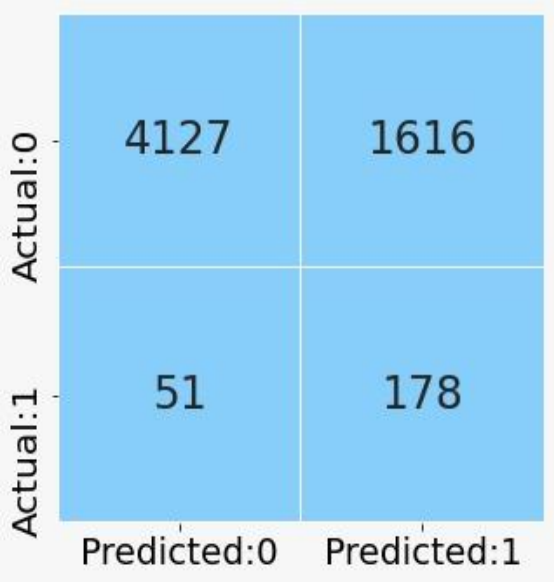
```
Classification report:

              precision    recall  f1-score   support

           0       0.98      0.99      0.99      5743
           1       0.74      0.59      0.66       229

    accuracy                           0.98      5972
   macro avg       0.86      0.79      0.82      5972
weighted avg       0.97      0.98      0.98      5972
```

KNN Algorithm:

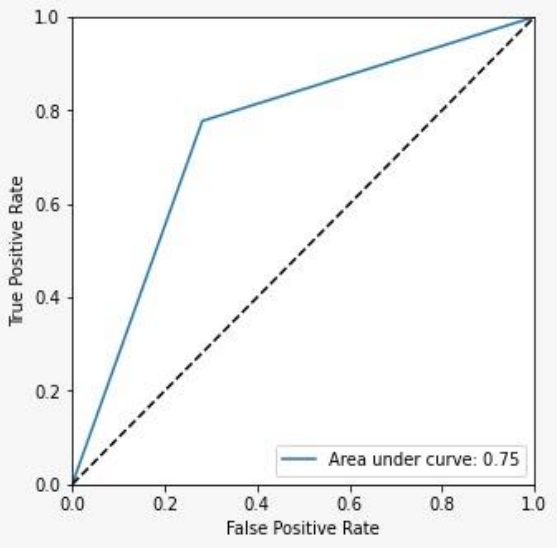
- The K -Nearest Neighbour algorithm classifies the data based on the similarity measure.
- K specifies the number of nearest neighbours to be considered.

```
knn=KNeighborsClassifier()
model_evaluation('knn',knn,X_train_smt,y_train_smt,X_test_sc,y_test)
```

Confusion Matrix:



ROC Curve:



Classification report:

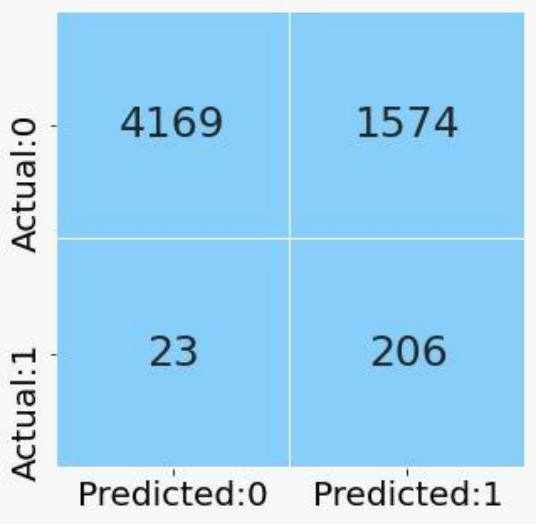
```
Classification report:

              precision    recall  f1-score   support

           0       0.99      0.72      0.83      5743
           1       0.10      0.78      0.18       229

    accuracy                           0.72      5972
   macro avg       0.54      0.75      0.50      5972
weighted avg       0.95      0.72      0.81      5972
```

Naïve Bayes Algorithm:

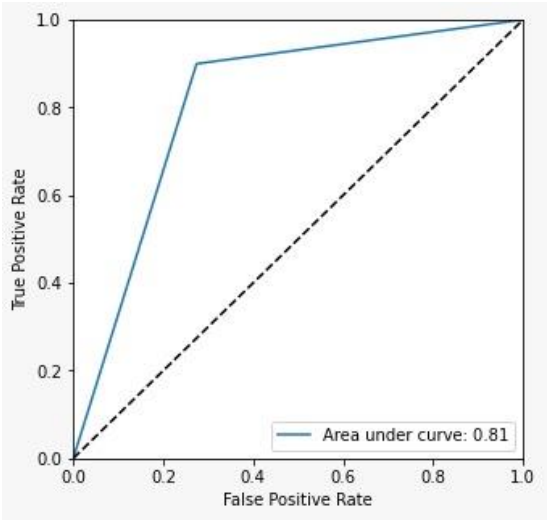
- A naïve bayes classifier uses the Bayes theorem for classification.
- It is an eager learning algorithm. Since it does not wait for test data to learn, it can classify the new instance faster.

```
GNB=GaussianNB()
model_evaluation('GNB',GNB,X_train_smt,y_train_smt,X_test_sc,y_test)
```

Confusion Matrix:



ROC Curve:



Classification Report:

```
Classification report:

              precision    recall  f1-score   support

           0       0.99      0.73      0.84      5743
           1       0.12      0.90      0.21       229

    accuracy                           0.73      5972
   macro avg       0.56      0.81      0.52      5972
weighted avg       0.96      0.73      0.81      5972
```

Models Comparison:

	Model	tn	fp	fn	tp	Precision	Recall	F1_score	AUC	Train_Accuracy	Test_Accuracy	Kappa_score
0	Logistic	5234	509	55	174	0.254758	0.759825	0.381579	0.835598	0.949007	0.905559	0.343896
1	DecisionTree	5684	59	91	138	0.700508	0.602620	0.647887	0.796173	0.998731	0.974883	0.634940
2	RandomForest	5696	47	94	135	0.741758	0.589520	0.656934	0.790668	0.998731	0.976390	0.644874
3	knn	4127	1616	51	178	0.099220	0.777293	0.175976	0.747953	0.907720	0.720864	0.115845
4	GNB	4169	1574	23	206	0.115730	0.899563	0.205077	0.812745	0.935382	0.732585	0.147125

PERFORMANCE METRICS:

- **Confusion Matrix:**

  It is the performance measure for the classification problem. It is a table used to compare predicted and actual values of the target variable.

- **ROC:**

  ROC curve is the plot of TPR against the FPR values obtained at all possible threshold values.

PERFORMANCE EVALUATION METRICS:

- **Accuracy:**

  Accuracy is the fraction of predictions that our model got correct. Higher the accuracy of the model better is the model.

- **Precision:**

  Precision is the proportion of positive cases that were correctly predicted.

- **Recall:**

  A recall is the proportion of actual positive cases that were correctly predicted.

- **F1 score:**

  F1score is the harmonic mean of precision and recall values for a classification model.

- **Cohen Kappa score.**

  Kappa statistic is a measure of inter-rater reliability or degree of agreement.

BUSINESS JUSTIFICATION:

- The main motive is to improve the performance of the model. As per the business scenario, the model is predicting the product is an advantage but in reality, it is not an advantage. In this scenario, the customer is betrayed and will lose the customer. This is a type 2 error.
- In the second scenario if the model is predicted the product is not an advantage but in reality, the product is an advantage. In this scenario, a customer might not choose the product which is also a loss to the company. This is a type 1 error.
- However, both the errors are costly but type 2 error is costlier than type 1. Here we need to reduce the false positives as minimum as possible. So, we choose precision as a performance metric.
- From the above model comparison table, the Random Forest model has good precision compared to other models. So, we are considering the Random Forest as a base model.

HYPER-PARAMETERS:

Pre-pruning can be done by specifying the following hyperparameters:

max_depth:

- It is the maximum length of the decision allowed to grow.
- Once the max_depth value is reached the tree will not grow further.

min_samples_split:

- The minimum samples are required to split an internal node.

min_samples_leaf:

- The minimum samples are required to be at the leaf node.
- A node will split further only if its child nodes will have the min_sample_leaf.
- May give the effect of smoothing.

n-estimators:

- This is the number of trees you want to build before taking the maximum voting or averages of predictions.

HYPER-PARAMETER TUNING:

- The Random Forest model is overfitted since there is a significant difference in the training and testing accuracies. Hence hyper tuning the parameters is needed to reduce the overfitting of the model.
- The hyperparameters can be tuned using the Grid Search method. It considers all the combinations of the hyperparameters and returns the optimal hyperparameter values.
- The tuned parameters resulted from the Grid search method as follows:
    - Criterion: Gini
    - Max_depth:16
    - Min_sample_split:6
    - n_estimators:120
    - min_sample_leaf:1

	Model	tn	fp	fn	tp	Precision	Recall	F1_score	AUC	Train_Accuracy	Test_Accuracy	Kappa_score
6	RF_tunedp2	5334	409	104	125	0.234082	0.545852	0.327654	0.737317	0.941952	0.914099	0.28952

Inference:

By the best parameters from the grid search method, we have got the above scores. However, the FP increased to 409 from 47 which is not acceptable as per the business problem. The model didn't perform well by considering the hyperparameters obtain from the Grid Search method. Hence boosting algorithms are considered to improve the performance of the model.

Boosting Algorithms:

- To improve the performance of the model, boosting classifiers such as Adaboost, Gradient boosting and XGboost classifiers are considered.
- The below are the results obtained from the boosting algorithms.

	Model	tn	fp	fn	tp	Precision	Recall	F1_score	AUC	Train_Accuracy	Test_Accuracy	Kappa_score
8	AdaBoost	5290	453	98	131	0.224315	0.572052	0.322263	0.746587	0.902979	0.907736	0.282750
9	GB	5421	322	111	118	0.268182	0.515284	0.352765	0.729608	0.920785	0.927495	0.318385
10	XGB	5633	110	95	134	0.549180	0.585153	0.566596	0.783000	0.984657	0.965673	0.548744

- Out of all the model's Gradient boosting model reducing the over fitting of the model but increased the FP values. This is not acceptable to a business problem.
- So, Hyper tuning the Gradient boosting model to reduce the False Positive values.

	Model	tn	fp	fn	tp	Precision	Recall	F1_score	AUC	Train_Accuracy	Test_Accuracy	Kappa_score
0	RandomForest	5696	47	94	135	0.741758	0.589520	0.656934	0.790668	0.998731	0.976390	0.644874
1	GB	5421	322	111	118	0.268182	0.515284	0.352765	0.729608	0.920785	0.927495	0.318385
5	GB_5_8	5667	76	89	140	0.648148	0.611354	0.629213	0.799060	0.996454	0.972371	0.614877

Inference:

After hyper tuning the Gradient boosting model, we infer that there is a reduction in FP values and by comparing the training and testing accuracies the model is overfitted again.

Conclusion:

Hyper tuning the parameters of the Random Forest model, the model has not given better results compared to the base model. Since the main requirement is to maximize the precision so consider the random forest base model as final model.

The Random Forest model provides better results (precision=0.74) with an accuracy of 0.97 as per our business scenario.

References:

- https://towardsdatascience.com/predicting-e-commerce-sales-with-a-random-forest-regression-3f3c8783e49b
- https://blog.mlreview.com/gradient-boosting-from-scratch-1e317ae4587d
- https://towardsdatascience.com/evaluating-machine-learning-classification-problems-in-python-5-1-metrics-that-matter-792c6faddf5
- https://www.salesforce.com/eu/learning-centre/customer-service/customer-retention/
- https://www.knowledgeisle.com/wp-content/uploads/2019/12/2-Aur%C3%A9lien-G%C3%A9ron-Hands-On-Machine-Learning-with-Scikit-Learn-Keras-and-Tensorflow_-Concepts-Tools-and-Techniques-to-Build-Intelligent-Systems-O%E2%80%99Reilly-Media-2019.pdf