Customer Segmentation Using Machine Learning Classification Algorithms

(EEN-366: Course Project)

Sarthak Gupta
sarthak_g@ee.iitr.ac.in
Enrollment: 20115126

Shreyas Pradeep Pakhare
shreyas_pp@ee.iitr.ac.in
Enrollment:20115138

Abstract

This project explores the use of machine learning classification algorithms for customer segmentation in the context of a retail business. The aim is to group customers into different segments based on their buying behavior and preferences to tailor marketing strategies and improve customer satisfaction. The project applies various classification algorithms, including K-means clustering, Decision Tree, Random Forest, SVM, Naive Bayes, and K-Nearest Neighbors to classify the customers into different segments based on their work experience, Spending Score, demographic information etc. In our work, Random Forest was found to give the best performance across various metrics. The findings of this project have practical implications for retailers seeking to improve their marketing strategies and customer engagement through personalized approaches.

1. Main Objectives

- Dividing a customer base into groups of individuals similar in specific ways relevant to marketing, such as age, gender, interests and spending habits.
- Comparing the results obtained by these models and deciding the best model for the dataset.

2. Introduction

Customer segmentation is a key strategy for businesses looking to improve their marketing efforts and increase customer satisfaction.

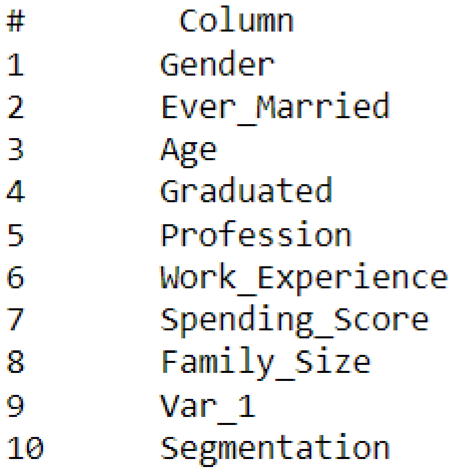
By dividing customers into specific groups based on shared characteristics, companies can tailor their marketing strategies and improve the effectiveness of their outreach.

In this project, we explore the use of machine learning classification algorithms for customer segmentation in the context of an automobile company seeking to expand into new markets.

We apply several machine learning classification algorithms, including K-means clustering, decision tree, random forest, SVM, and Naive Bayes, to classify the new customers into segments based on their behaviour and demographics.

We then compare the performance of these algorithms to determine which is most effective in predicting the right customer segment.

3. Dataset and Preprocessing



```
#        Column
1        Gender
2        Ever_Married
3        Age
4        Graduated
5        Profession
6        Work_Experience
7        Spending_Score
8        Family_Size
9        Var_1
10       Segmentation
```

Each entry in the dataset has 10 features with Gender, Ever_Married, Graduated, Spending_Score, Var_1 and Segmentation being categorical ones and the rest of them are numerical.

3.1 Categorical data

Categorical data is a type of data that represents discrete, qualitative variables that can take on a limited number of categories or levels.

Categorical data is different from numerical or continuous data, which represent quantitative variables that can take on a range of values within a specific range.

Categorical data is commonly used in fields such as market research, social sciences, and healthcare, where it is important to identify and analyze different categories or levels of a variable

3.2 Encoding technique

Machine learning algorithms typically require input data in numerical format, so encoding is necessary to transform categorical data into a format that can be used by these algorithms.

- Label encoding is a technique for converting categorical data into numerical data. In label encoding, each category or level of a categorical feature is assigned a unique integer value, such as 0, 1, 2, etc.
- One hot encoding is a technique for creating a binary feature for each unique category or level of a categorical feature. In one hot encoding, a new binary feature is created for each category, and the feature is assigned a value of 1 if the category is present and 0 if it is not.

In our case, Label encoding gave better results.

3.3 Standardization and Scaling

Scaling involves transforming numerical data so that it falls within a specific range. The goal of scaling is to standardize the range of values in a feature to make them more comparable.

Common scaling techniques include min-max scaling and z-score scaling.

Min-max scaling involves scaling the data to a range between 0 and 1 by subtracting the minimum value from each observation and dividing by the range of values.

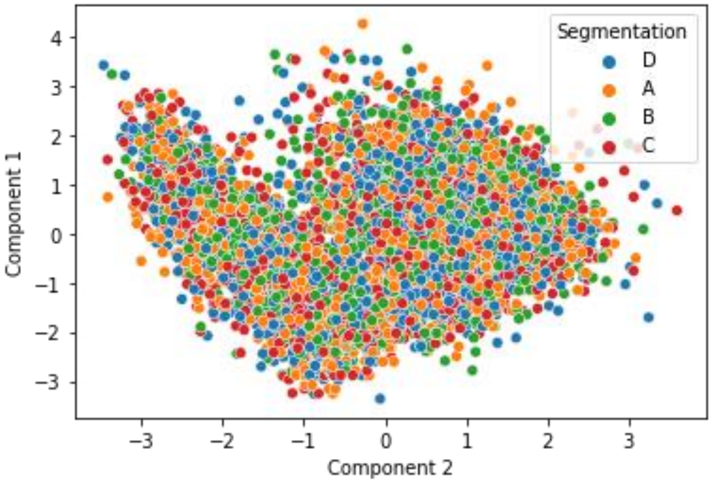
For the given dataset, there is no need for scaling random forests and decision trees.

In our case, we have applied min max scaler for SVM and KNN as they are distance-based algorithms so each feature should have equal weightage in feature space

By scaling the data, we make sure that different features are on the same scale and don't cause the algorithm to prioritise one feature over another

3.4 Visualizing the data in two dimensions

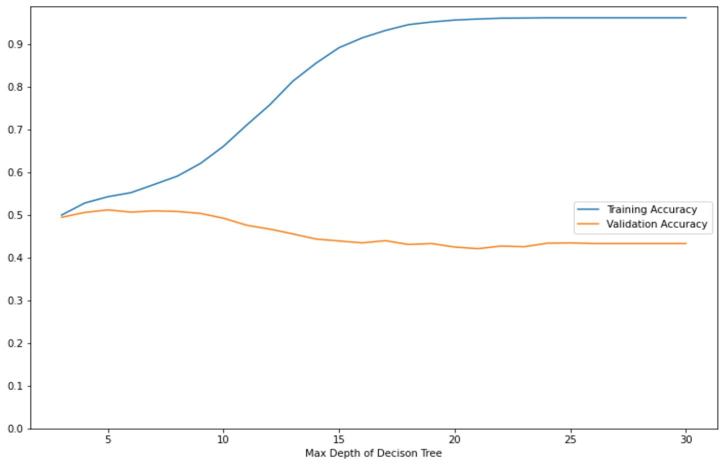
We have used PCA for visualization and reduced data to 2 dimensions from 8 dimensions. Explained variance between the two components is only 40%. Hence no tangible conclusion can be drawn from it. We need to use more components to see the effect.



4.1 Decision Tree

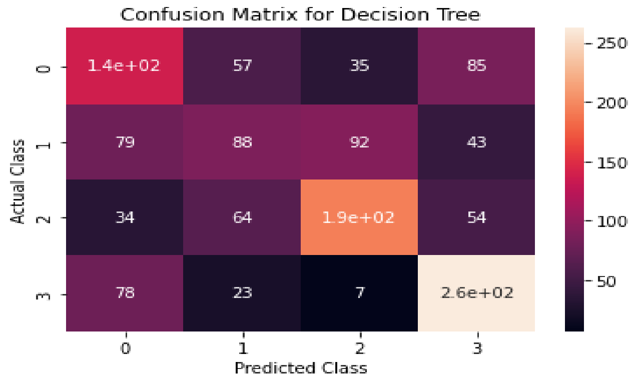
At each node in the tree, a decision is made based on a specific feature, and the data is split into subsets based on the outcome of that decision. Using GridSearchCV for hyperparameter tuning the best performance was obtained for criterion=entropy, splitting =best and max-depth=5.

In the following Figure, the training curve for this model is given. As the model trains, the training accuracy increases; however, validation accuracy decreases. This is a clear sign that our model is overfitting on the training set. We overcome this by limiting the maximum depth of the tree.



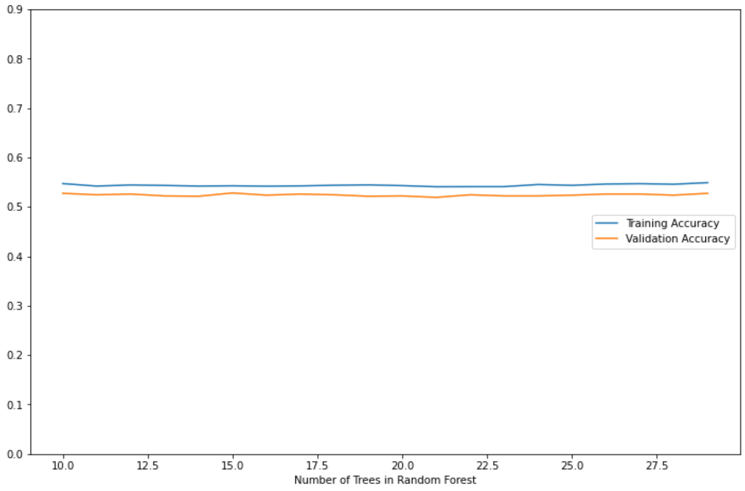
4.2 Decision Tree on Test Set

We have used the hyperparameters that gave us the best results on our cross-validation set on our test set and obtained the following metrics:
Precision = 0.5, Recall = 0.5, and Accuracy = 0.51.



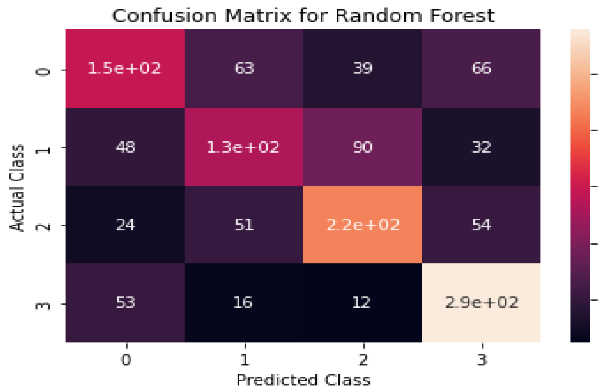
5.1 Random Forests

Random Forest uses bagging to randomly select samples and randomly select features to train several trees. This ensures that correlation amongst the trees is minimized. Using GridSearchCV as before, the best hyper-parameters were found to be: criterion = entropy, max_depth = 5, max_features = 4 and the number of trees = 23.
In the following Figure, the training curve for this model is given. As the model trains, the accuracy remains more or less constant indicating that increasing the number of trees after a certain threshold is not very useful.



5.2 Random Forest on Test Set

We have used the hyperparameters that gave us the best results on our cross-validation set on our test set and obtained the following metrics:
Precision = 0.58 Recall = 0.59 and Accuracy = 0.59

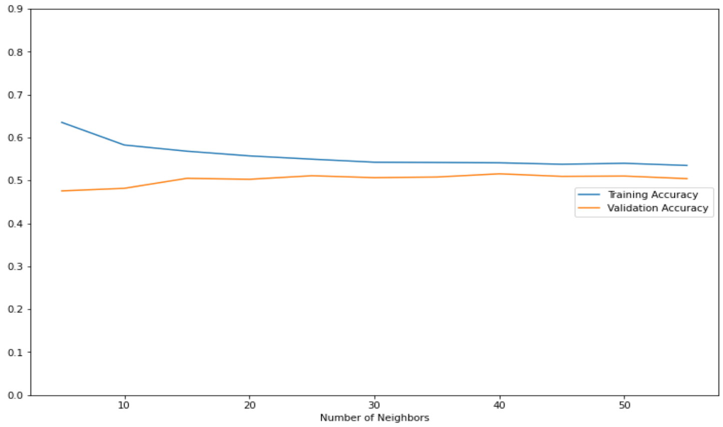


6.1 K-Nearest Neighbors (KNN)

The algorithm works by finding the k closest training samples to a new test sample, and then assigning the class label or output value of the test sample based on the majority vote or average value of its k nearest neighbours. We can either use majority voting or use distance-weighted KNN, which gives more weight to closer samples.

Using GridSearchCV, the best performance was found to be for uniform weight and n = 45.

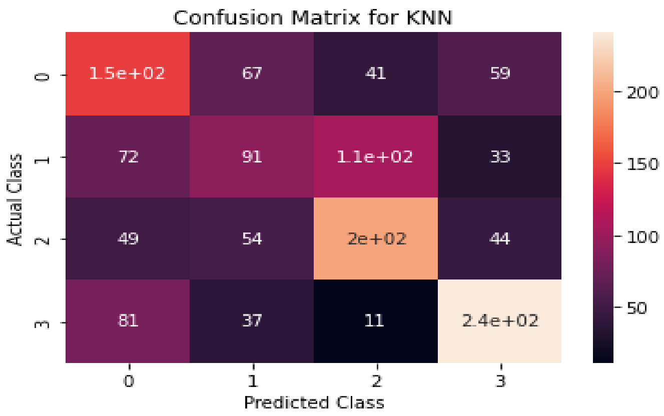
The figure below shows the variation in performance as we increase the value of n.



6.2 KNN on Test Set

Using the hyper-parameters found above, we run the model on the test set and obtain the following metrics:

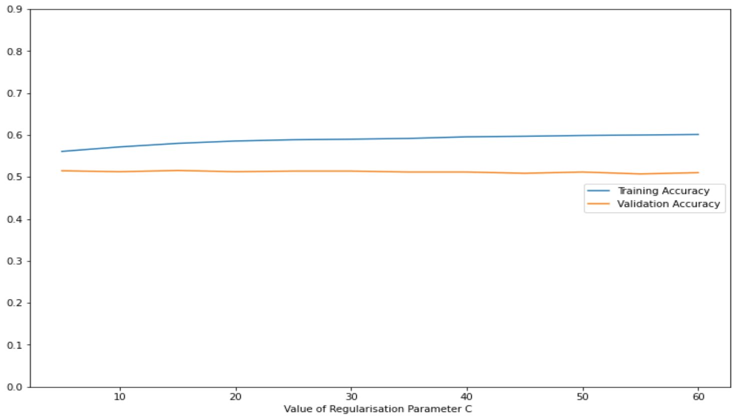
Precision = 0.50 Recall = 0.51 and Accuracy = 0.51



7.1 Support Vector Machines (SVM)

In the SVM model, we have varied the penalty term (C) value and calculated the validation accuracies. As the data is not linearly-separable we have taken the RBF Kernel.
C is the penalty applied for every misclassification. A high value of C results in hard-margin SVM while a low-value results in soft-margin SVM.

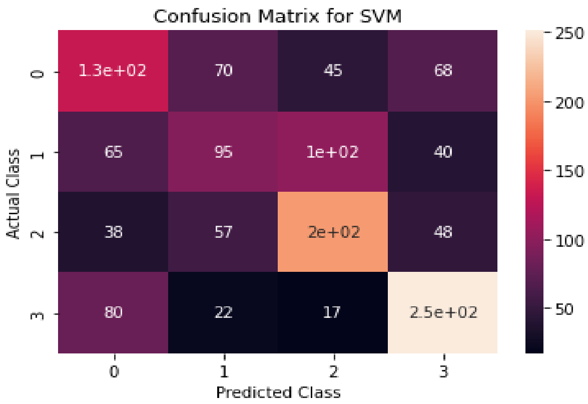
From the graph below, we can see that validation accuracy steadily decreases as the value of C increases. This may be because the margin of SVM shrinks as C increases. The best validation score was obtained for C = 41.



7.2 SVM on Test Set

Using the hyper-parameters found above, we run the model on the test set and obtained the following metrics:

Precision = 0.50 Recall = 0.51 and Accuracy = 0.51

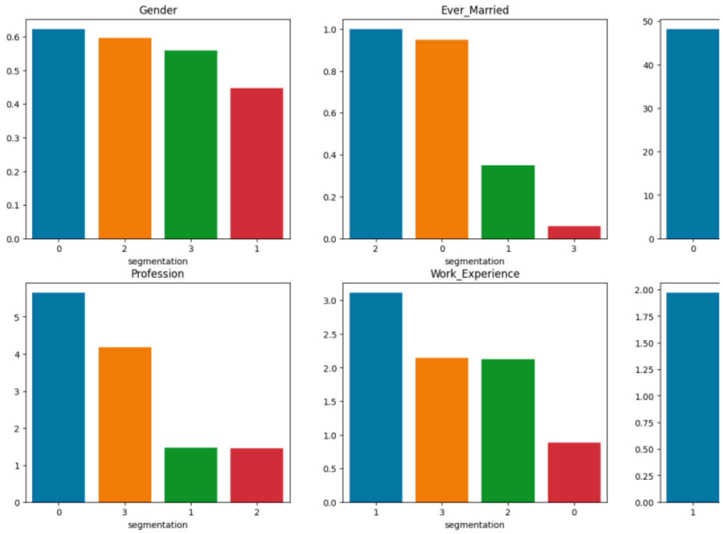


7. K Means-Clustering

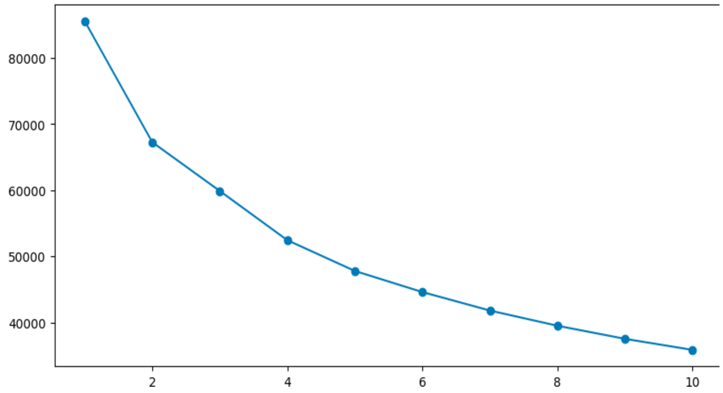
Unsupervised machine learning algorithm used for partitioning a set of data points into a specific number of clusters

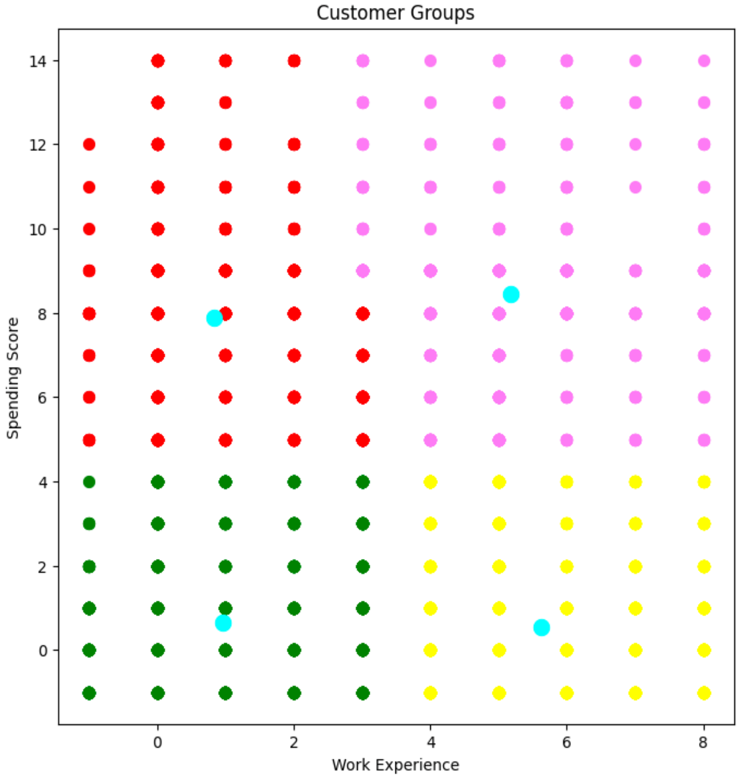
There is only one hyperparameter K(optimal number of clusters) which is decided using the elbow method.

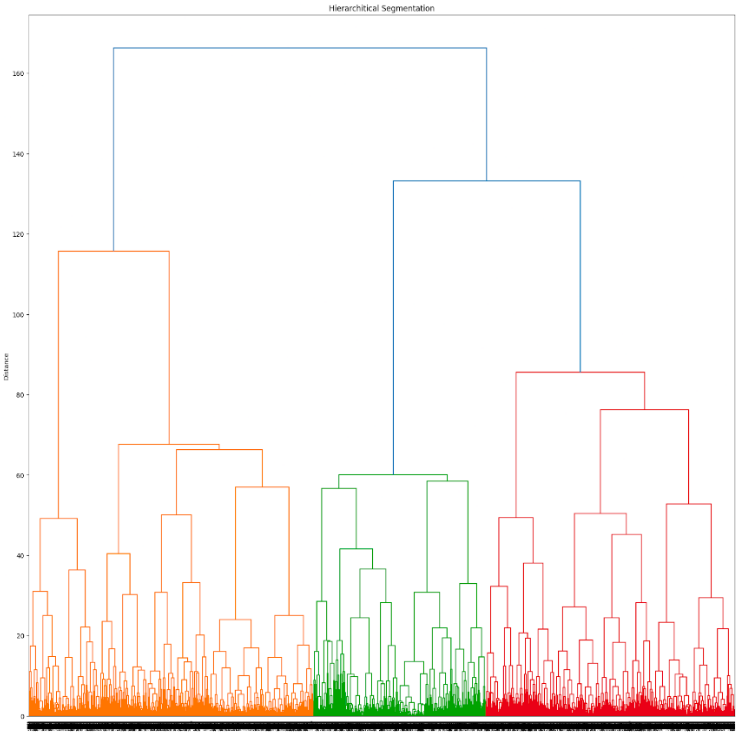
The elbow method involves plotting the within-cluster sum of squares (WCSS) against the number of clusters and looking for an "elbow" or point of inflection in the curve, which indicates the optimal number of clusters.



Each bar plot represents the mean value of a particular customer attribute (Gender, Ever_Married, Age, etc.) for each of the four segments identified by the K-means clustering Algorithm.







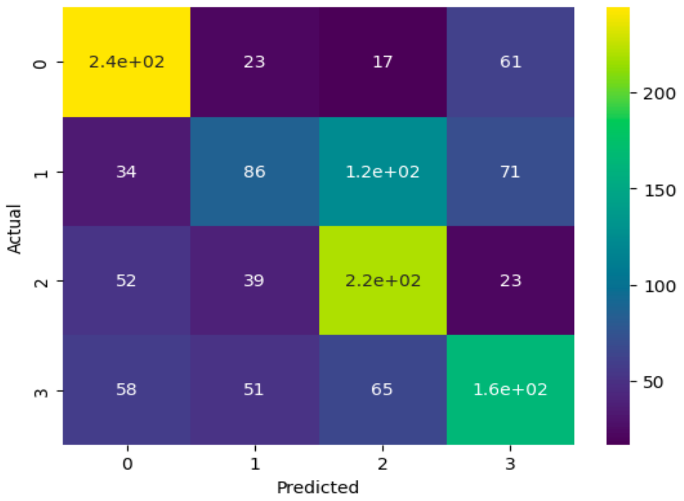
8.1 Naive Bayes

CategoricalNB is a variant of the Naive Bayes algorithm designed to handle categorical data, i.e., data that consists of discrete values or categories, as opposed to continuous numerical values. CategoricalNB is a simple but effective algorithm for handling categorical data, and it is often used as a baseline model in machine learning experiments.

However, it has some limitations, such as the assumption of independence between features (hence the "naive" in Naive Bayes) and the inability to handle continuous numerical data.

8.2 Naive Bayes on Test Set

The following results were obtained on test set using Naive Bayes :
Precision = 0.53 Recall = 0.53 Accuracy = 0.54



9. Results and Conclusions

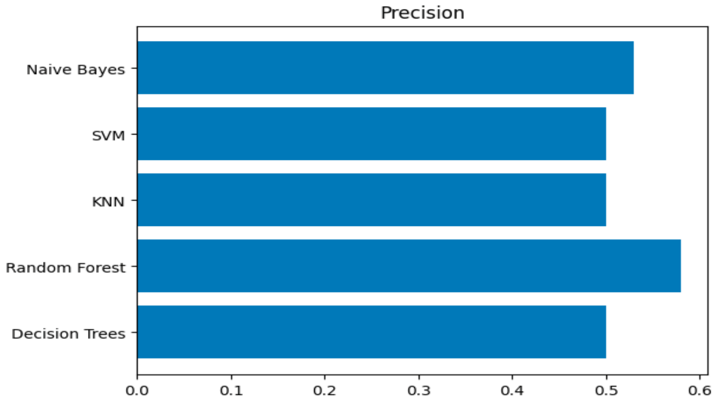
After simulations, Random Forest was able to achieve the best performance across all metrics.

A high precision score means that the model accurately identifies customers who belong to a specific segment.
A high recall score means that the model is good at identifying all the customers who belong to a particular segment.

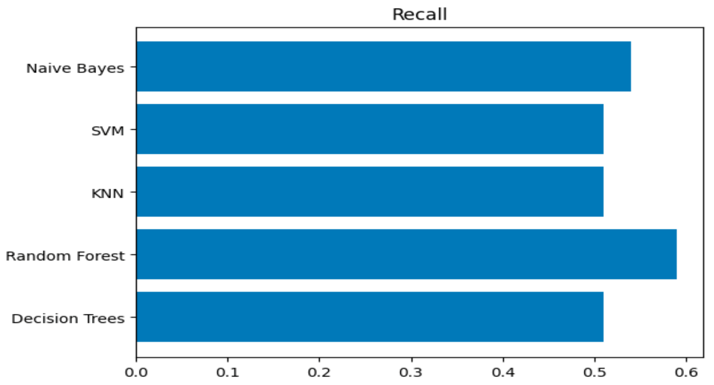
In the context of customer segmentation, F1-score provides a balance between precision and recall and gives an overall measure of how well the model performs.

All three metrics are important, but which one is the most important depends on the specific problem and business context.

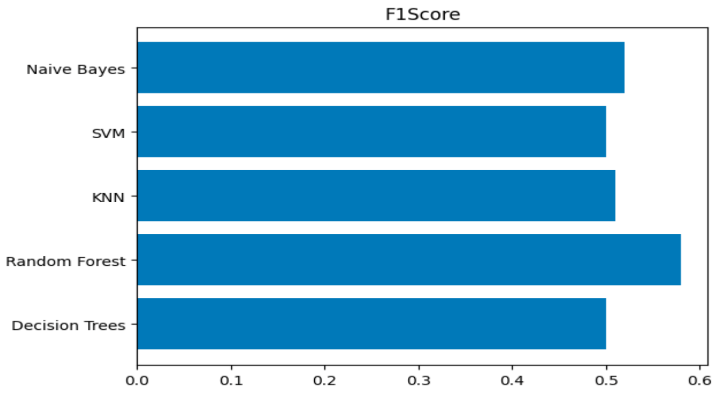
COMPARISON OF ML MODELS FOR VARIOUS METRICS



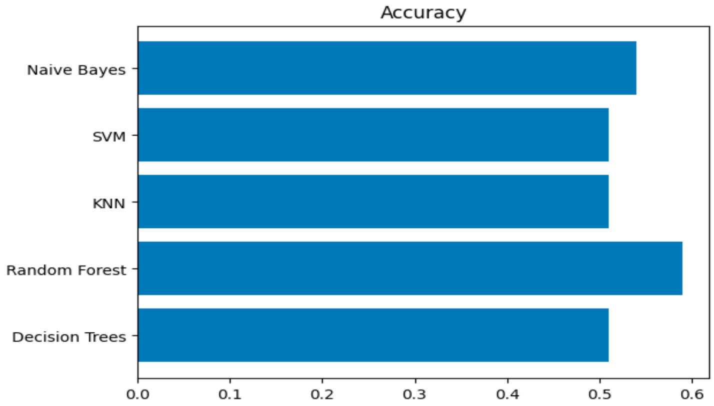
PRECISION FOR VARIOUS ML MODELS



RECALL FOR VARIOUS ML MODELS



F-1 SCORE FOR VARIOUS ML MODELS



ACCURACY FOR VARIOUS ML MODELS

10. References

1. *Kaggle Dataset for Customer Segmentation* *https://www.kaggle.com/datasets/abisheksudarshan/customer-segmentation?resource=download*
2. *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow, 2nd Edition. by Aurélien Géron.*
3. *Machine Learning: An Artificial Intelligence Approach- book by Tom Mitchelle*