1. **Introduction**
   1. **Task Description:**

In this project, we will be developing a random forest algorithm to analyze customer data from a company that sells medical devices. Our goal is to provide the business owner with insights into the similarities between customers and the relationships between his customer data using data mining and business intelligence skills.

* 1. **Data Description:**

A picture containing text, screenshot, font

Description automatically generated

The data consists of customer data from a medical device sales company. The dataset includes the customer ID, the year of operation of the business, the number of sales calls made to that customer, the number of targeted emails sent to that customer, the number of sales to that customer, and the customer satisfaction index. The data has no demographic data about the customers.

The grain of the data is the column CustomerSatisfaction to define it as a unique in the project to complete the task. the column CustomerSatisfaction contains numerical values:

* -1 to present as not satisfied (unhappy).
* 0 to present as neutral.
* 1 to present as satisfied (happy).

1. Data Preparation
   1. Data Exploration

A screen shot of a computer

Description automatically generated with medium confidence

1- One of the first things is the data has the same datatype.

2- The data does not have missing values.

3- The data has specific rage which there are no outliers.

4- Because the data has all numerical values there is no need to convert to dummy variables. However, in the data, the column, **year** contain distances for the data, so, the data must be normalized.

We explored the data to gain a better understanding of its features and characteristics. We used pandas methods like describe() and info() to see basic statistics and information about the data. We also used visualizations like histograms, scatter plots, and box plots to see the distributions and relationships between the different features.

* 1. Data Visualization

Principle Component Analysis (PCA):

dataset is small, it's still important to consider scaling and normalization when using PCA.

PCA plot has overlapping clusters, it may be more difficult for a random forest model to accurately classify your samples, since the clusters may not be well-separated.

A picture containing screenshot, colorfulness

Description automatically generated

Bar Chart:

to compare the values of different categories.

A picture containing text, screenshot, rectangle, colorfulness

Description automatically generated A picture containing text, screenshot, rectangle, diagram

Description automatically generated

A screenshot of a graph

Description automatically generated with medium confidence

Pie Chart:

to show the relative proportions of different categories.

A screen shot of a graph

Description automatically generated with low confidence A screen shot of a graph

Description automatically generated with low confidence

A screen shot of a pie chart

Description automatically generated with medium confidenceA screen shot of a computer

Description automatically generated with low confidence

Histogram:

to show the distribution of a single variable.

A screenshot of a graph

Description automatically generated with medium confidence A screen shot of a graph

Description automatically generated with medium confidenceA screenshot of a graph

Description automatically generated with medium confidence A screenshot of a graph

Description automatically generated with medium confidence

Cluster Method (K-Means):

Since we don't have a good elbow graph for your k-means cluster analysis, it can mean that it is difficult to determine the optimal number of clusters for your data. The elbow graph is a plot of the number of clusters on the x-axis and the within-cluster sum of squares (WCSS) on the y-axis. The WCSS measures the sum of the squared distances between each data point and its assigned cluster centroid.

The elbow graph is used to identify the optimal number of clusters by looking for the "elbow" point, which is the point of diminishing returns in terms of reducing the WCSS as the number of clusters increases. If you don't have a clear elbow point, it can be difficult to determine the optimal number of clusters.

A picture containing text, line, screenshot, plot

Description automatically generated

Cluster for other columns:

If there is no relationship between two columns using k-means clustering, it means that the algorithm did not find any meaningful patterns or structure in the data that would suggest a relationship between those two columns. This can happen if the data is too noisy or if there is no underlying structure to the data that can be captured by k-means clustering.

A picture containing text, screenshot, line, number

Description automatically generated A picture containing text, screenshot, font, display

Description automatically generated

A picture containing text, screenshot, line, number

Description automatically generated

Correlation:

The correlation between two variables in visualization is 0.1, it means that there is a weak positive relationship between the variables. A correlation coefficient of 0.1 indicates that there is a positive association between the variables, but the strength of the relationship is weak.

For more information, that correlation does not necessarily imply causation. A correlation coefficient of 0.1 suggests that there is a weak positive relationship between the variables, but it does not necessarily mean that changes in one variable cause changes in the other variable.

A screenshot of a graph

Description automatically generated with low confidence

Area Under the Curve (AUC):

As a classification, I got an AUC of 0.88 for the highest class in the model, it means that the model is performing well at distinguishing between the positive cases in that class and the negative cases in the other classes.

The AUC measures the model's ability to distinguish between positive and negative cases at different threshold settings. An AUC of 0.88 indicates that the model can correctly identify a high proportion of positive cases in the highest class while minimizing the number of false positives in the other classes.

It's important to keep in mind that the AUC is just one metric for evaluating model performance, and it should be used in conjunction with other metrics and considerations. Additionally, the specific research question and the nature of the data should be considered when interpreting the AUC and other model performance metrics.

However, these are not got enough information because we are using only classification to analyze the model.

A picture containing text, screenshot, line, plot

Description automatically generated

Plotting histogram and normal probability to fine the relationship between the customers admiration and the customers. Grouping columns such as years with customers to show percentages of a whole and represents percentages at a set point in time. Plotting chat Pie to see the connection between of the columns. Plotting the correlation graph to assess a possible linear association between two continuous variables. Area Under the Curve (AUC) measures how well parameters tuning, for Class 1,2, and 3, as a separate AUC graph line in the same plot

* 1. Preprocessing

A screen shot of a computer

Description automatically generated with low confidence

Since the model does not have many features and is not too complicated, I choose Random Forest as my classification model.

In this task, Random forests or random decision forests is an algorithm model is applying to answer all the consideration to the business owner. Random forests are an ensemble learning method for classification, regression and other tasks that operates by constructing a multitude of decision trees at training time. For classification tasks, the output of the random forest is the class selected by most trees.

**Pipeline** is a simple way to keep your data preprocessing and modeling code organized and make the passing the data from one stage to another much smoother. Specifically, a pipeline bundles preprocessing and modeling steps so you can use the whole bundle as if it were a single step.

Many data scientists hack together models without pipelines, but pipelines have some important benefits. Those include:

1. **Cleaner Code:** Accounting for data at each step of preprocessing can get messy. With a pipeline, you won't need to manually keep track of your training and validation data at each step.
2. **Fewer Bugs:** There are fewer opportunities to misapply a step or forget a preprocessing step.
3. **Easier to Productionize:** It can be surprisingly hard to transition a model from a prototype to something deployable at scale. We won't go into the many related concerns here, but pipelines can help.
4. **More Options for Model Validation:** You will see an example in the next tutorial, which covers cross-validation.

receiving a score of 67% on your cross-validation, this means that model correctly classified 67% of the test cases. While this may seem low, it's important to keep in mind that the accuracy of a model can vary depending on many factors such as the quality of the data, the complexity of the model, and the size of the training and testing sets.

It's also worth noting that the score you received on your cross-validation may not be the final accuracy of your model. You may need to adjust the model parameters or try different algorithms to improve its performance.

1. Feature generation and transformation

We generated and transformed features as needed. We used techniques like scaling, normalization to improve the performance of our model. We normalize the year column, two decimals, to keep the data in the same range. We also did the normalization on the whole data since we used the PCA.

A screenshot of a computer

Description automatically generated with medium confidence

1. Model development



We developed a random forest model to analyze the data. We used scikit-learn's RandomForestClassifier or RandomForestRegressor to create a model with the desired number of trees and other hyperparameters.

random forest models are often robust to noise and outliers, so it's possible that the model could still identify important patterns or relationships in your data.

To improve the performance of your random forest model, you might consider using feature selection techniques to identify the most informative variables, or using other machine learning algorithms that are better suited to your specific data and research question. It's also important to carefully evaluate the performance of your model using appropriate metrics and cross-validation techniques, to ensure that you're not overfitting the data or making unrealistic assumptions about the underlying structure of your data.

1. Results and conclusion

As a Result:

A screenshot of a computer

Description automatically generated with medium confidence

**F1-score**

* customer satisfaction survey, a F1 score (class 1) of 67% is a relatively not good score, as it suggests that the model is accurately identifying a significant proportion of happy customers. This information can be used to identify areas of strength and weakness in the business, and to make data-driven decisions to improve customer satisfaction.
* customer satisfaction survey, a F1 score (class 0) of 70% is a relatively not good score, as it suggests that the model is accurately identifying a significant proportion of happy customers. This information can be used to identify areas of strength and weakness in the business, and to make data-driven decisions to improve customer satisfaction.
* customer satisfaction survey, a F1 score (class -1) of 65% is a relatively not good score, as it suggests that the model is accurately identifying a significant proportion of unhappy customers. This information can be used to identify areas of strength and weakness in the business, and to make data-driven decisions to improve customer satisfaction.

Recall

* The recall score is like a grade that tells us how well the model is doing in identifying all the happy customers. A score (class 1) of 59% means that the model is identifying some happy customers, but it is missing a significant number of them.
* The recall score is like a grade that tells us how well the model is doing in identifying all the happy customers. A score (class 0) of 70% means that the model is identifying some neutral customers, but it is missing a significant number of them.
* The recall score is like a grade that tells us how well the model is doing in identifying all the happy customers. A score (class -1) of 70% means that the model is identifying some dissatisfied customers, but it is missing a significant number of them.

Precision

* score is a grade that tells us how well the model is doing in accurately identifying happy customers. A score (class 1) of 77% means that the model is doing a decent job, but there is still room for improvement.
* score is a grade that tells us how well the model is doing in accurately identifying neutral customers. A score (class 0) of 69% means that the model is doing a suitable job, but there is still room for improvement.
* score is a grade that tells us how well the model is doing in accurately identifying unhappy customers. A score (class -1) of 61% means that the model is doing a suitable job, but there is still room for improvement.

As a Conclusion:

The classification algorithm that we developed is designed to help us make important business decisions based on data that we collect from our customers. Specifically, we're using the algorithm to classify our customers into one of three groups based on their behavior and preferences.

Unfortunately, the accuracy of the algorithm is not as high as we had hoped. This means that the algorithm is not doing as good of a job as we would like at correctly classifying our customers. However, it's important to keep in mind that accuracy is just one of many metrics that we can use to evaluate the performance of the algorithm.

There are several factors that may be contributing to the poor performance of the algorithm, including the size and quality of the dataset, the balance of the classes, and the specific features that we're using to classify our customers. We're currently working to identify ways to improve the accuracy of the algorithm, and we're exploring other metrics such as precision, recall, and F1 score to get a more complete picture of the algorithm's performance.

Overall, we believe that the algorithm is still a useful tool for helping us make business decisions, but we also recognize that there is room for improvement. We're committed to working with you to identify ways to improve the accuracy of the algorithm and to ensure that we're making the best possible decisions based on the data that we collect from our customers.

GitHub link:

[Click Here](https://github.com/dia20/sells_medical_devices.git) to access the repository (README file and SalesOfMedicalDevices.ipynb code)