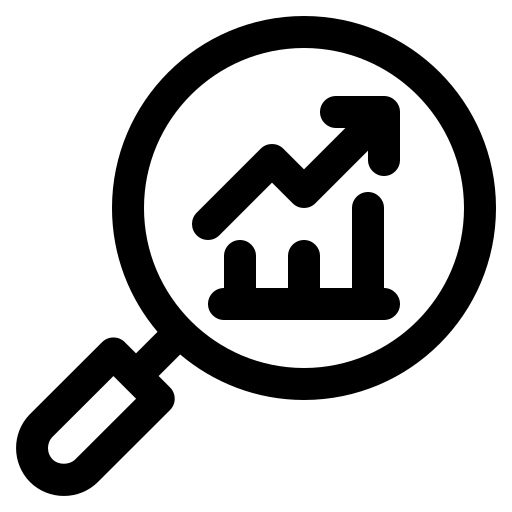
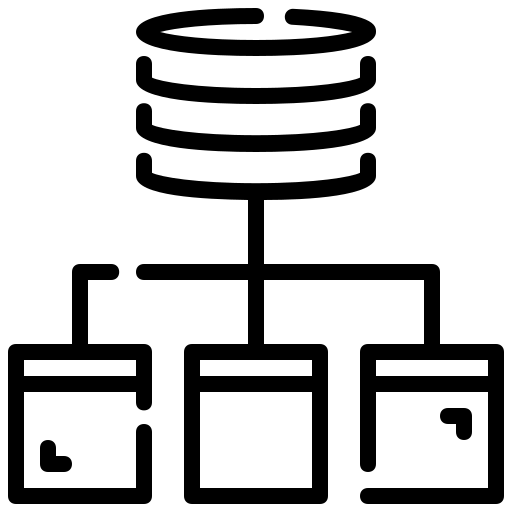
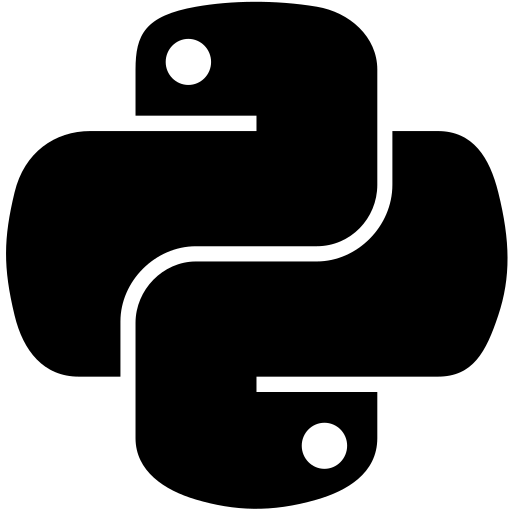
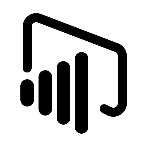
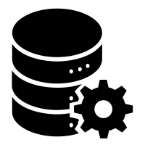


**Bank Marketing Model Performance**



Portfolio

**HOZANA GUIMARAES REIS**

**Table of Contents**

[**1.** **Objective** 4](#_Toc161052196)

[**2.** **Libraries** 5](#_Toc161052197)

[**2.1** **Importing Necessary Libraries.** 5](#_Toc161052198)

[**3.** **Exploratory Data Analysis (EDA)** 5](#_Toc161052199)

[**3.1**  **Load the Data** 5](#_Toc161052200)

[**3.2** **Data Description and Missing Values Check** 6](#_Toc161052201)

[**3.3** **Data Information** 7](#_Toc161052202)

[**3.4** **Univariable Check** 8](#_Toc161052203)

[**3.4.1 Numerical Columns** 8](#_Toc161052204)

[**3.4.2 Categorical Columns** 9](#_Toc161052205)

[**3.5** **Bivariable Check** 10](#_Toc161052206)

[**3.5.1 Categorical Columns** 10](#_Toc161052207)

[**3.5.2 Numerical Columns** 12](#_Toc161052208)

[**3.6** **Correlation** 13](#_Toc161052209)

[**4.** **Data processing** 14](#_Toc161052210)

[**4.1** **Categorical Treatment** 16](#_Toc161052211)

[**4.2** **Numerical Treatment** 19](#_Toc161052212)

[**5.** **Machine Learning Model** 19](#_Toc161052213)

[**5.1** **Splitting the Data into Train and Test** 19](#_Toc161052214)

[**5.2** **Classifiers** 20](#_Toc161052215)

[**5.2.1** **Logistic Regression** 20](#_Toc161052216)

[**5.2.2** **Decision Tree** 21](#_Toc161052217)

[**5.2.3** **Random Forest** 22](#_Toc161052218)

[**5.2.4** **Support Vector Machines (SVM)** 23](#_Toc161052219)

[**5.2.5** **Gradient Boosting** 24](#_Toc161052220)

[**5.2.6** **K- Nearest Neighbors (KNN)** 25](#_Toc161052221)

[**5.2.7** **Naïve Bayes** 26](#_Toc161052222)

[**5.3** **1st Model Evaluation** 27](#_Toc161052223)

[**5.4** **ROC Curve from Gradient Boosting and Random Forest** 28](#_Toc161052224)

[**5.5** **Feature Importance** 29](#_Toc161052225)

[**5.5.1** **Gradient Boosting Feature Importance** 30](#_Toc161052226)

[**5.5.2** **Random Forest Feature Importance** 31](#_Toc161052227)

[**5.5.3** **Feature Importance Conclusion (Gradient Boosting)** 31](#_Toc161052228)

[**5.5.4** **Feature Importance Conclusion (Random Forest)** 32](#_Toc161052229)

[**5.5.5** **Feature Importance Conclusion** 32](#_Toc161052230)

[**6.** **Model Performance Dashboard** 33](#_Toc161052231)

[**7.** **Conclusion** 34](#_Toc161052232)

# **Objective**

The objective is to use all the information to predict whether someone will end up saving money with the bank. This helps the bank decide who to focus on when they're trying to get people to save money with them. The Analysis will come across 7 classifiers and each performance for better decision-making. Additionally, the feature importance of various factors will be examined to gain insights into the key drivers influencing the predicted outcomes.

**Technologies Used:**

**Programming Language:** Python

**Data Analysis:** Pandas, NumPy

**Machine Learning:** Scikit-learn.

**Visualization:** Matplotlib, Seaborn

**Business Intelligence:** Power BI

**Classifiers:** Logistic Regression, Decision Tree, Gradient Boosting, Random Forest, Support Vector Machine (SVM), K-Nearest Neighbors (KNN), Naive Bayes

# **Libraries**

## A screen shot of a computer Description automatically generated**Importing Necessary Libraries.**

Note: More Libraries will be used, they are placed along with the code

# **Exploratory Data Analysis (EDA)**

## **A screenshot of a computer Description automatically generated Load the Data**

## **A screenshot of a document Description automatically generatedA white background with green text Description automatically generatedData Description and Missing Values Check**

## **Data Information**

**A screenshot of a computer

Description automatically generated**

## **Univariable Check**

### **A screenshot of a computer code Description automatically generated3.4.1 Numerical Columns**

**A group of blue and green bars

Description automatically generated with medium confidence**

### **A group of different colored bars Description automatically generatedA computer screen shot of a computer code Description automatically generated3.4.2 Categorical Columns**

**A screenshot of a computer

Description automatically generated**

## **Bivariable Check**

### **3.5.1 Categorical Columns**

**A computer code on a white background

Description automatically generated**

**A screenshot of a graph

Description automatically generated**

### **3.5.2 Numerical Columns**

**A computer code on a white background

Description automatically generated**

**A collage of different colored boxes

Description automatically generated**

## **Correlation**

**A computer code with text

Description automatically generated**

**A table of statistical data

Description automatically generated with medium confidence**

# **Data processing**

I decided to remove the columns with high correlation (More than 0.95)

**A screenshot of a computer code

Description automatically generated**

**A computer code with text

Description automatically generated with medium confidence**

**A diagram of a number of variables

Description automatically generated with medium confidence**

I decided to keep the columns I think is relevant to the model.

**A screenshot of a computer program

Description automatically generated**

## **A screenshot of a computer program Description automatically generatedCategorical Treatment**

**A screenshot of a computer

Description automatically generated**

**A close-up of a white background

Description automatically generatedA screenshot of a computer code

Description automatically generatedA screenshot of a computer program

Description automatically generated**

**A screenshot of a computer

Description automatically generatedA screenshot of a computer code

Description automatically generatedA screenshot of a computer

Description automatically generated**

## **A screenshot of a computer Description automatically generatedNumerical Treatment**

# **Machine Learning Model**

## **Splitting the Data into Train and Test**

**A computer code with text

Description automatically generated with medium confidence**

## **Classifiers**

### **Logistic Regression**

**A screenshot of a computer program

Description automatically generatedA blue and white graph

Description automatically generated**

**Logistic Regression achieved an accuracy of 90.0% and demonstrated good performance in classifying the target variable. It correctly classified 6908 instances of the negative class (no) and 314 instances of the positive class (yes).**

### **Decision Tree**

**A screenshot of a computer program

Description automatically generated**

**A blue rectangular box with numbers and a black text

Description automatically generated with medium confidence**

**Decision Tree achieved an accuracy of 88.3%. It correctly classified 6604 instances of the negative class and 505 instances of the positive class.**

### **Random Forest**

**A screenshot of a computer program

Description automatically generated**

**A blue and white graph with numbers and labels

Description automatically generated**

**Random Forest achieved an accuracy of 90.2% and demonstrated robust performance. It correctly classified 6807 instances of the negative class and 417 instances of the positive class.**

### **Support Vector Machines (SVM)**

**A screenshot of a computer program

Description automatically generated**

**A blue and white graph

Description automatically generated**

**SVM achieved an accuracy of 90.0%. It correctly classified 6984 instances of the negative class and 234 instances of the positive class.**

### **Gradient Boosting**

**A screenshot of a computer program

Description automatically generatedA blue and white graph

Description automatically generated**

**Gradient Boosting achieved an accuracy of 90.9% and demonstrated excellent performance. It correctly classified 6839 instances of the negative class and 452 instances of the positive class.**

### **K- Nearest Neighbors (KNN)**

A screenshot of a computer program

Description automatically generatedA blue and white graph

Description automatically generated

**KNN achieved an accuracy of 89.2%. It correctly classified 6823 instances of the negative class and 336 instances of the positive class.**

### **Naïve Bayes**

A computer screen shot of a program code

Description automatically generatedA blue and white graph

Description automatically generated

**Naive Bayes achieved an accuracy of 88.9%. It correctly classified 6748 instances of the negative class and 389 instances of the positive class.**

## **1st Model Evaluation**

**Summarizing this part: Gradient Boosting achieved the highest accuracy among the classifiers tested, followed closely by Random Forest. These models demonstrated robust performance in predicting the target variable, Let's check it further.**

**Double-check if the Gradient Boosting and Random Forest are the best among other classifiers comparing through Precision, Recall, and F1 score.**

A screenshot of a computer program

Description automatically generated

A screenshot of a computer

Description automatically generated

## **A screenshot of a computer Description automatically generatedA screenshot of a computer screen Description automatically generatedROC Curve from Gradient Boosting and Random Forest**

**A screenshot of a computer program

Description automatically generated**

**A graph of a positive rate

Description automatically generated with medium confidence**

## **Feature Importance**

A white background with black text

Description automatically generated

### **Gradient Boosting Feature Importance**

A screenshot of a computer program

Description automatically generated

### **Random Forest Feature Importance**

A screenshot of a computer

Description automatically generated

### **Feature Importance Conclusion (Gradient Boosting)**

**Duration:** This feature has the highest importance, indicating that the duration of the call has a significant impact on the outcome.

**Pdays:** The number of days that passed after the client was last contacted from a previous campaign is also a crucial factor.

**Cons.conf.idx and Cons.price.idx:** These are economic indicators, suggesting that the overall economic context plays a role.

**Age Group and Previous Contacts:** These features have relatively lower importance but still contribute to the model.

### **Feature Importance Conclusion (Random Forest)**

**Duration:** Similarly, the duration of the call is the most critical predictor in the Random Forest model.

**Cons.conf.idx and Cons.price.idx:** Economic indicators remain significant in this model as well.

**Job and Campaign:** Job type and number of contacts during this campaign also have notable importance.

**Education and Age Group:** These features also contribute significantly to the model's predictions.

### **Feature Importance Conclusion**

Both models highlight the importance of the call duration and economic indicators (cons.conf.idx and cons.price.idx). Other factors such as job type, education level, and age group also play essential roles in predicting the outcome of the marketing campaign. Overall, these insights can guide marketing strategies to focus on specific customer demographics and tailor communication strategies based on economic conditions and call duration.

# **A screenshot of a graph Description automatically generatedModel Performance Dashboard**

# **Conclusion**

After integrating all classifiers into the Power BI dashboard and comparing their performance, it's evident that Gradient Boosting and Random Forest consistently outperform other classifiers. This conclusion is drawn from various performance metrics such as Accuracy, Precision, Recall, F1 Score, and AUC. Moreover, analyzing the feature importance reveals that the duration of the call holds significant importance for both classifiers. This insight suggests that investing time in client calls could be an effective strategy for the bank.