# CMPE 407: Machine Learning

İstanbul
Bilgi University

Final Project

Bank Marketing study


| Full Name | Student ID |
|---|---|
| Mohammed Obidou | 119200016 |

Lab Instructor: Prof. Özgür Özdemir



**Department of Computer Engineering**
**Istanbul Bilgi University**
**Istanbul, Turkiye**

**May 25, 2023**

# Contents

**1 Introduction** **1**

**2 Dataset** **1**
2.1 Attributes . . . . . 1
2.2 Visualization . . . . . 2

**3 Reprocessing** **5**

**4 Experiments** **6**
4.1 Normal Classification Models . . . . . 6
4.2 Hyper-parameters and cross-validation . . . . . 7
4.3 Ensemmble Models . . . . . 8
4.4 Optimization of Ensemble . . . . . 9
4.5 Average accuracy using different methods . . . . . 9
4.6 Accuracy of all the models . . . . . 10

**5 Results** **10**

**6 Conclusion** **10**

# 1 | Introduction

This study aims at predicting whether the customer will subscribe and deposit to the bank ultimately deciding whether or not the marketing has worked, and so I have used the famous bank marketing data-set, a lot of studies and researches were done on this data-set so it's perfect for comparing my work with other papers. Also the data have 17 features and have a lot of potential to demonstrate visualization, reprocessing, models and model's tuning techniques. I have used classification since we're predicting a binary target. y-target = deposit:yes deposit:no

# 2 | Dataset

The data set contains 17 columns, 11162 rows.

## 2.1 | Attributes

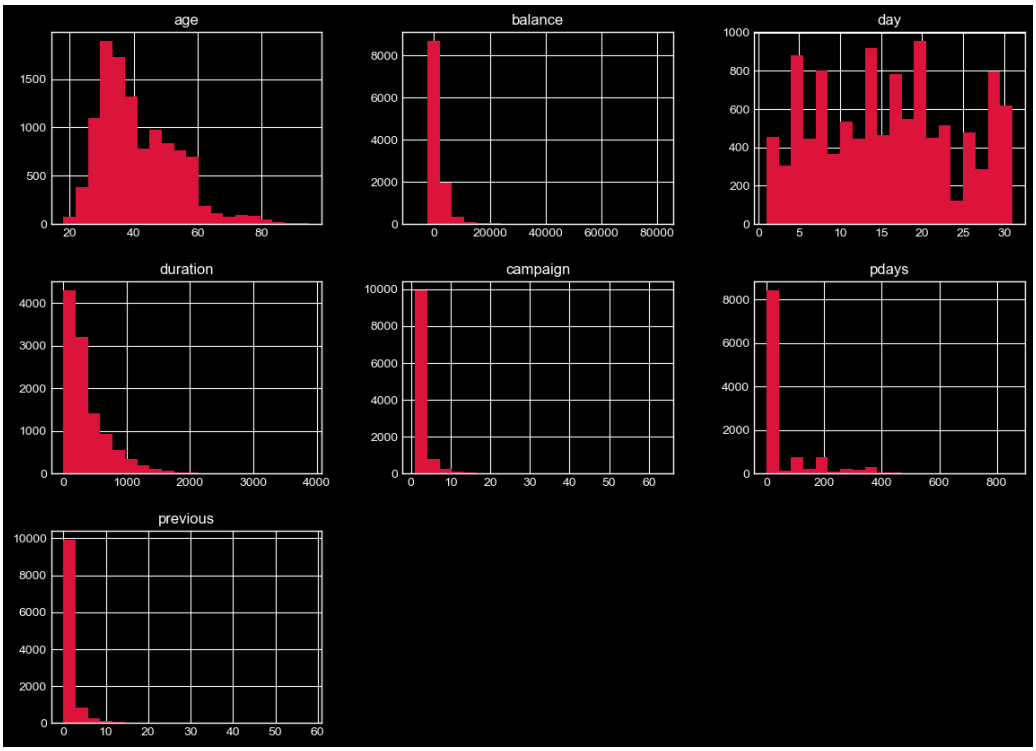
- Age (numeric)
- ob : type of job (categorical: 'admin.', 'blue-collar', 'entrepreneur', 'housemaid', 'management', 'retired', 'self-employed', 'services', 'student', 'technician', 'unemployed', 'unknown')
- Marital : marital status (categorical: 'divorced', 'married', 'single', 'unknown' ; note: 'divorced' means divorced or widowed)
- Education (categorical: 'basic.4y', 'basic.6y', 'basic.9y', 'high.school', 'illiterate', 'professional.course', 'university.degree', 'unknown')
- Default: has credit in default? (categorical: 'no', 'yes', 'unknown')
- Housing: has housing loan? (categorical: 'no', 'yes', 'unknown')
- Loan: has personal loan? (categorical: 'no', 'yes', 'unknown')
- contact: contact communication type (categorical: 'cellular','telephone')
- Month: last contact month of year (categorical: 'jan', 'feb', 'mar', ..., 'nov', 'dec')
- Day of week: last contact day of the week (categorical:'mon','tue','wed','thu','fri')
- Duration: last contact duration, in seconds (numeric).
- Campaign: number of contacts performed during this campaign and for this client (numeric, includes last contact)
- Pdays: number of days that passed by after the client was last contacted from a previous campaign (numeric; 999 means client was not previously contacted)
- Previous: number of contacts performed before this campaign and for this client (numeric)
- Poutcome: outcome of the previous marketing campaign (categorical: 'failure','nonexistent','success')
- deposite: did customer deposit: yes or no

## 2.2 | Visualization

**Important:** the data contains more visualization but due to the number of the plots i opted not to include all of them, please look at the notebook for more visualization plots.

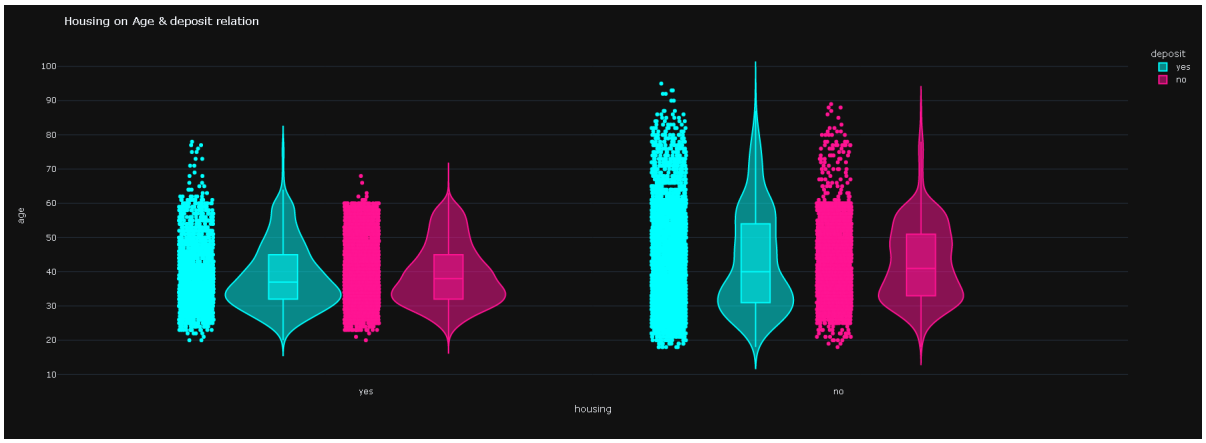
**Information on Subscription based on marital status**
we can see that the majority of the bank customers are married. 52.62% of people with loans did not open an account to deposit. while 47.38% did open an account divorced people had the least open accounts in the bank after single marital


**Numerical features data distribution using histogram**
Days is evenly distributed, Majority of people has balance below 20,000$, Majority of age is between 30-40 years old Duration majority is below 1000 campaign majority is between 0 and 5



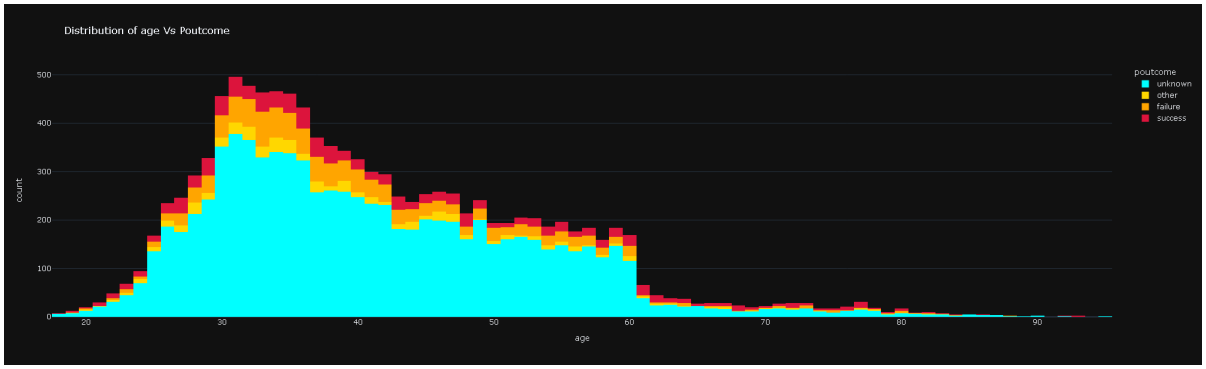
**Age relation to housing and deposit**

More people above the age of 60 has no housing older people with no housing has deposited more
The majority of people has no housing between the age 20-60 people with housing has less deposit compared to older people with housing



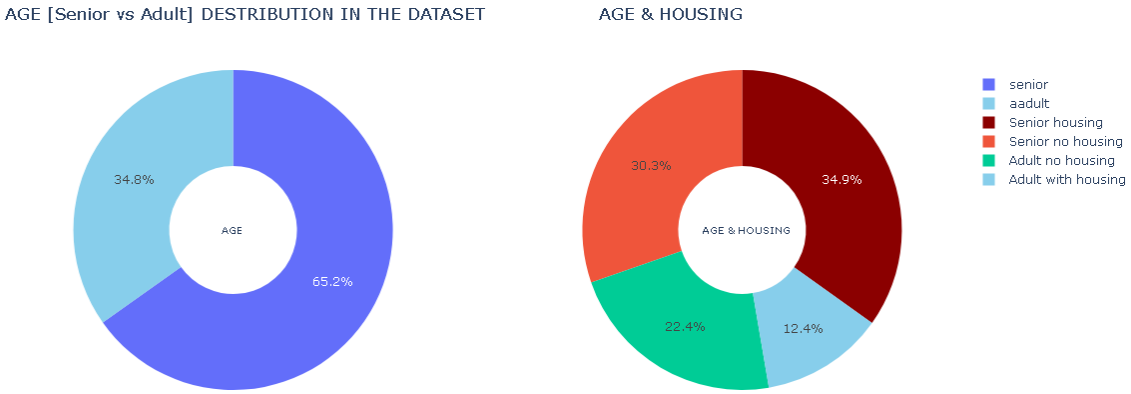

**Ring plot to show relation between age and housing**

Data distribution has the majority count between age of 30-40 majority of successful campaign are at age of 31 majority of failure campaign are customers of age 34 majority of undetermined campaign outcome are of age 31



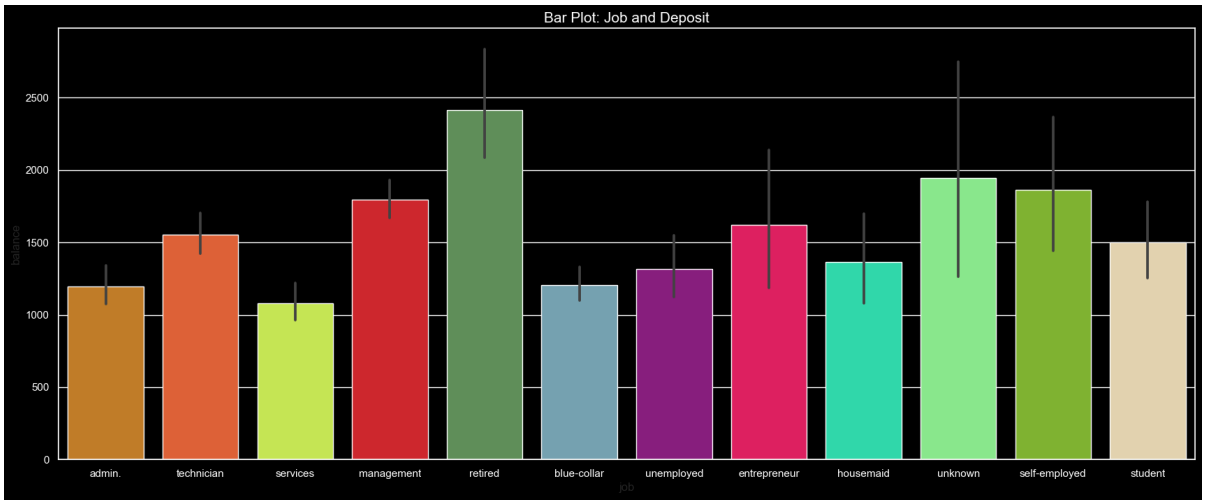

**Age relation to outcome of previous marketing campaign**

majority of people in our data are above the age of 45 65.2% to 34.8% who are under the age of 45
30% are senior with no housing 34.9% are seniors with housing
22.4% are adults with no housing to 12.4% adults with housing



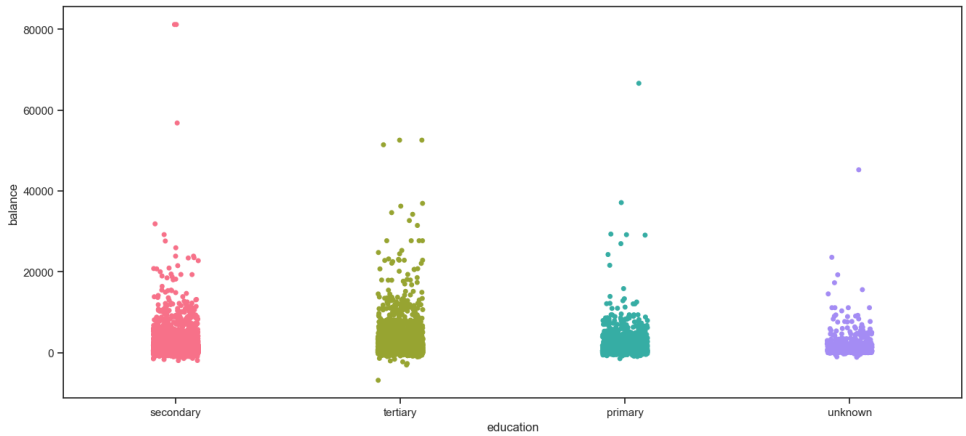

**Job to deposit**

majority of deposits in the bank are retired people

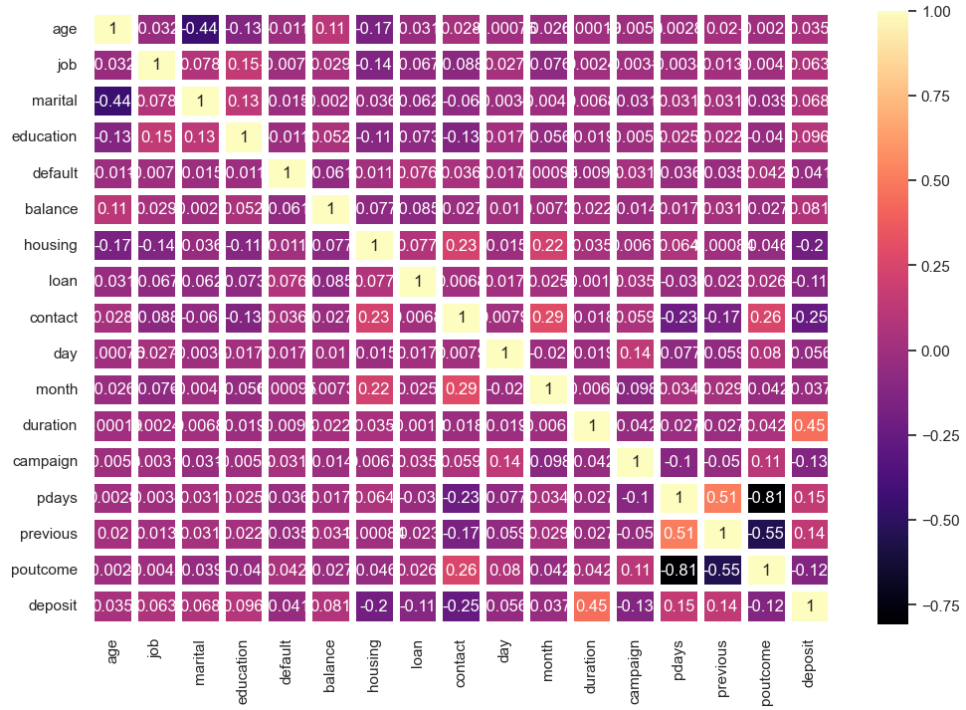



**Education to balance**

the highest balance above 80,000 has secondary school education the majority are tertiary

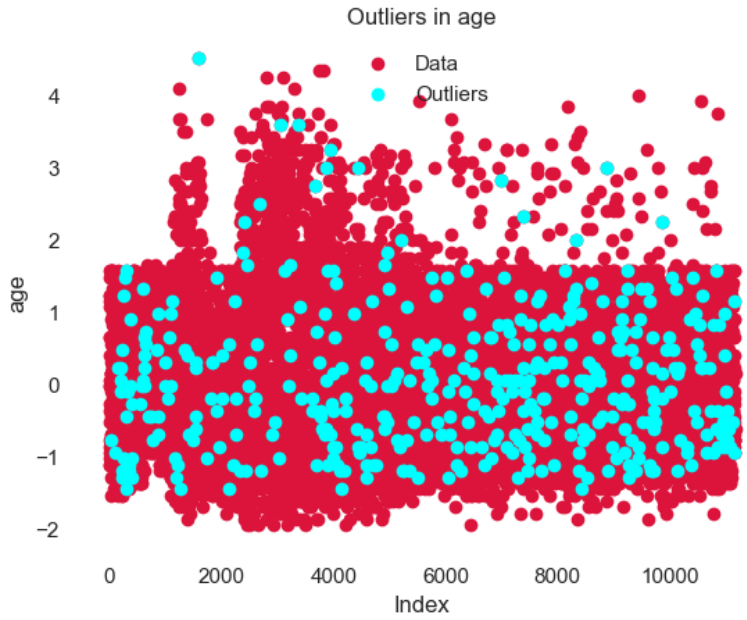


**Heatmap correlation between attributes**
Deposit has highest correlation with duration at 0.45
Previous campaign outcome has the highest correlation with campaign at 0.11
Pdays has highest correlation to previous at 0.51




## 3 | Reprocessing

- **Null Values:** the dataset contains no null values
- **Duplicates:** the dataset contains no duplicates
- **Categorical data are:** 'job', 'marital', 'education', 'default', 'housing', 'loan', 'contact', 'month', 'poutcome', 'deposit' so I used is used to convert categorical labels into numeric representations. It assigns a unique integer value to each category in the variable. For example, if you have a categorical variable "Color" with categories "Red", "Blue", and "Green", LabelEncoder will assign them numerical labels like 0, 1, and 2, respectively. The labels are encoded in a single column.
- **Normalization** is a data prepossessing technique used to scale numeric features to a specific range. It aims to bring all the features to a similar scale to avoid any bias or dominance of a particular feature due to its magnitude. I have used StandardScaler and made a copy of the dataset because we want normalized values for search based algorithms only and not Tree based algorithms
- **Outliers:** used z-score method with threshold of 5 to determine and drop outliers




- **Feature Selection:** used Extra tree, correlation and RFE methods to determine the features then I created a method to extract the common features and after inspecting the features I decided on 10 features

# 4 | Experiments

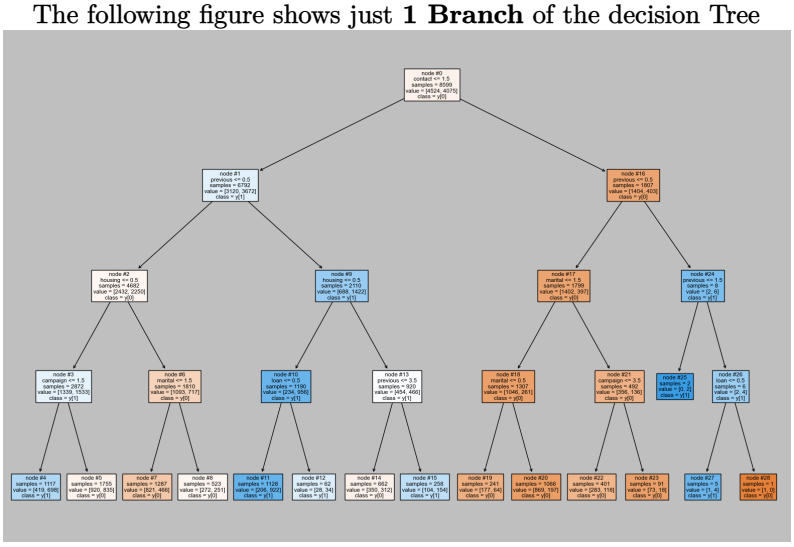
**Multiple models and methods were used in this experiments to evaluate the performance and ultimately decide the best model**

## 4.1 | Normal Classification Models

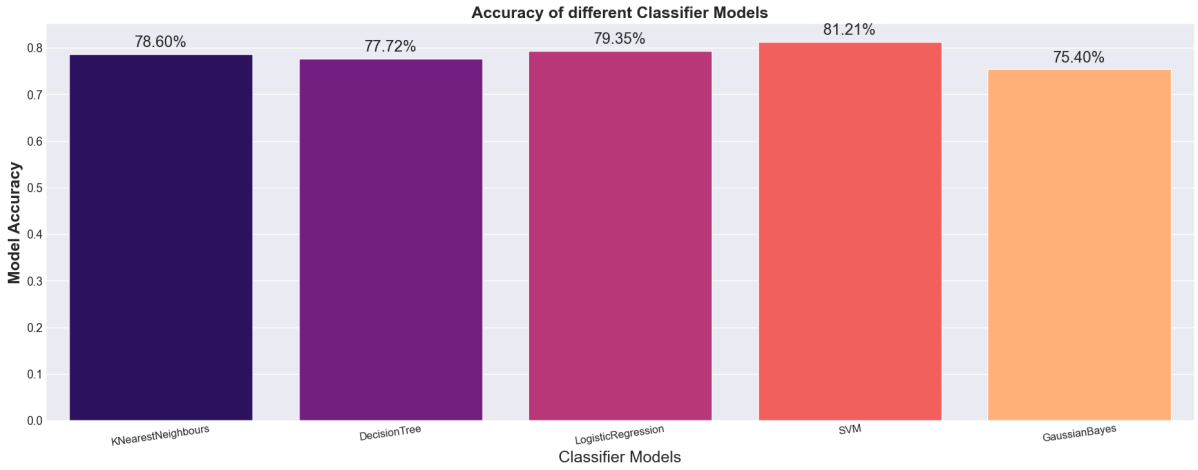
- **K-nearest neighbors:** is a simple yet powerful supervised machine learning algorithm used for both classification and regression tasks. It works by finding the K nearest data points in the training set to a given test data point and making predictions based on the majority class. distance metric (e.g., Euclidean distance) is used to determine the similarity between data points. The algorithm assumes that similar data points are likely to belong to the same class or have similar output values. K is a hyperparameter that determines the number of neighbors to consider.
- **Decision Tree:** are machine learning models that use a tree-like structure to make decisions or predictions based on a series of features or conditions. They partition the data based on these conditions, enabling them to classify or regress new instances by traversing the tree from the root to a leaf node that represents the final prediction.
- **Logistic Regression:** Logistic regression is a statistical model used for predicting binary outcomes, where the target variable is categorical with two classes. It is based on the concept of the logistic function (also called sigmoid function), which maps any real-valued number to a value between 0 and 1, representing the probability of belonging to the positive class.
- **Support vector:** is a search-based algorithm. It falls under the category of supervised learning algorithms and is commonly used for classification and regression tasks. SVM aims to find the optimal hyperplane that separates the data points of different classes with the largest margin
- **Gaussian Naive Bayes:** is a probabilistic classifier based on the Bayes' theorem that assumes the features to be continuous and follows a Gaussian (normal) distribution. It calculates the posterior

probability of a class given the features using the assumption of feature independence, making it computationally efficient but with the drawback of oversimplified assumptions.

The following figure shows just **1 Branch** of the decision Tree

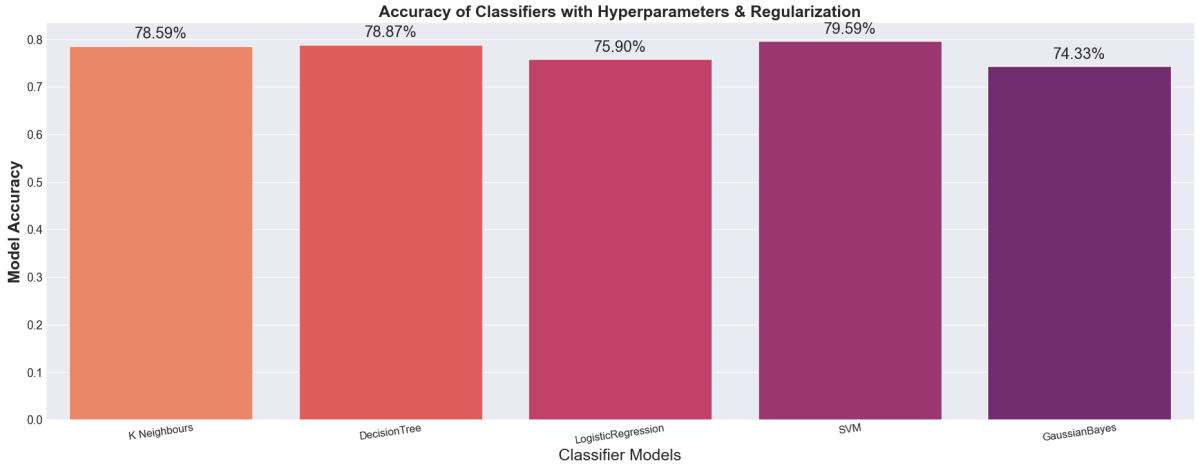


**The accuracy of the models:**



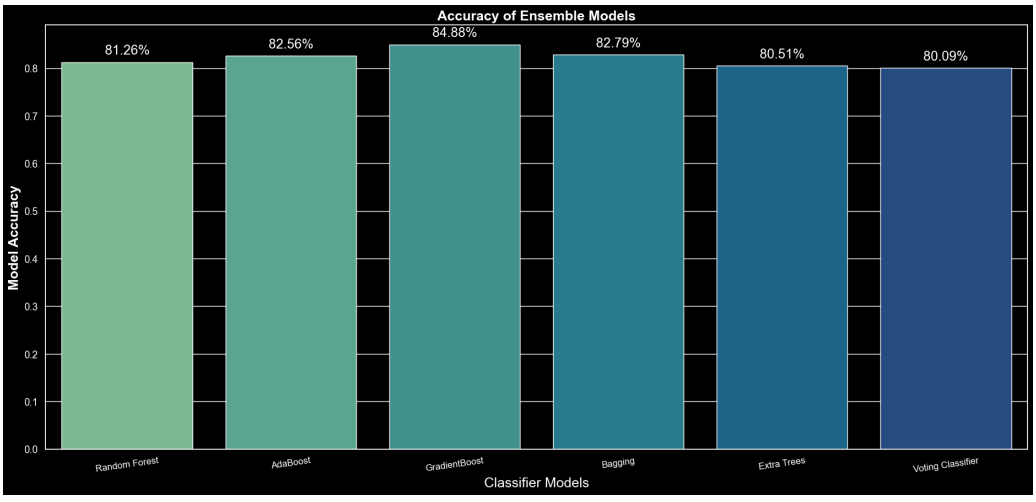

## 4.2 | Hyper-parameters and cross-validation

- **Hyperparameters** are parameters that are not learned from the data by the model itself, but rather set by the user prior to training. GridSearchCV is a technique used to systematically search for the best combination of hyperparameters by evaluating the model's performance on a grid of possible hyperparameter values. It exhaustively tries all possible combinations, using cross-validation, to find the optimal set of hyperparameters that maximize the model's performance.
- **Cross-validation** is a technique used to evaluate the performance of a machine learning model by partitioning the available data into multiple subsets or "folds." The model is trained on a subset of the data called the training set and then evaluated on the remaining subset called the validation set. This process is repeated multiple times, with different subsets used for training and validation, allowing for a more robust estimation of the model's performance and generalization ability.
- **Regularization:** is a technique used to prevent over fitting by adding a penalty added to the loss function during training to control the complexity of the model and prevent overfitting.



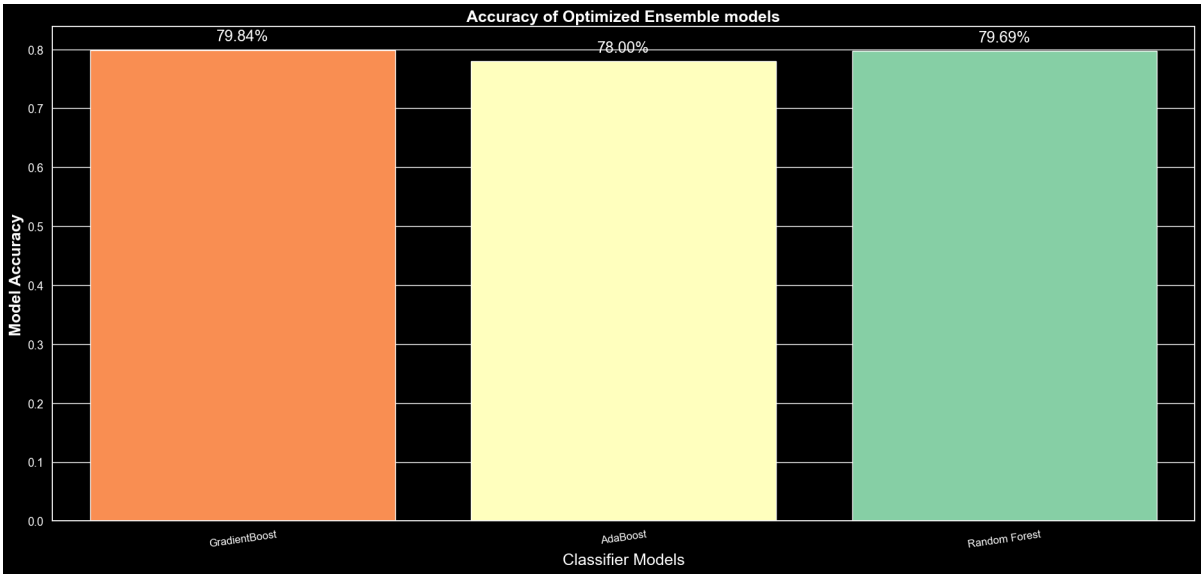

## 4.3 | Ensemmble Models

- **Random forest:** Ensemble of decision trees that combines their predictions to make final predictions. Each tree is trained on a random subset of features and data samples to reduce overfitting
- **ADAboost:** Ensemble method that combines weak classifiers to create a strong classifier Each classifier is trained sequentially, with more emphasis on misclassified samples in the previous rounds.
- **Gradient descent:** Ensemble method that builds models sequentially, where each model corrects the mistakes of the previous model. Each model is trained to minimize the loss function by gradient descent.
- **Bagging:** Ensemble method that creates multiple subsets of the original dataset through bootstrap sampling. Each subset is used to train a separate model, and predictions are aggregated by majority voting or averaging
- **Extra Tress:** Ensemble method similar to Random Forest, where each tree is trained on a random subset of features and data samples. The splitting of nodes in the trees is done randomly, which further increases the randomness and reduces overfitting
- **Voting Classifier:** Ensemble method that combines predictions from multiple individual classifiers to make final predictions. It uses a majority voting scheme (hard voting) or weighted averaging (soft voting) to determine the final prediction.



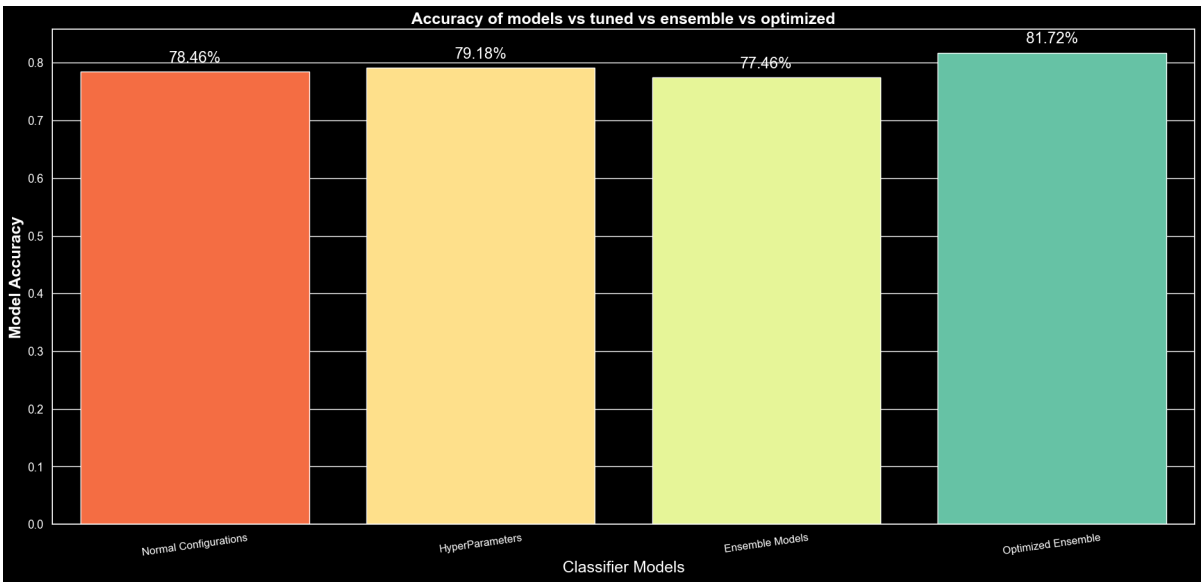
## 4.4 | Optimization of Ensemble

some optimizers like adam optimization algorithm commonly used for training deep learning models. It combines the benefits of both stochastic gradient descent (SGD) and adaptive learning rate methods. our models that we used have their own optimization methods and wehen applied we gain this accuracies




## 4.5 | Average accuracy using different methods

- now we calculate the average accuracy of models implemented without tuning
- calculate the average accuracy of models implemented hyperparameters, regularization and cross-validation
- calculate the average accuracy of ensemble models without optimization
- calculate the average accuracy of ensemble models with optimization



## 4.6 | Accuracy of all the models

- The model with the highest accuracy is Gradient Boost.
- Following Gradient Boost, the accuracy ranks are ADA, Bagging, Random Forest, and SVM.
- The best hyperparameters for each model are:
  - Gradient Boosting Best Parameters: learning_rate = 0.1, n_estimators = 100
  - AdaBoost Best Parameters: base_estimator = DecisionTreeClassifier(max_depth=1), n_estimators = 200
  - Random Forest Best Parameters: max_depth = 10, n_estimators = 200
  - Gaussian Naive Bayes Best Hyperparameters: var_smoothing = 1e-08
  - SVM Best Hyperparameters: C = 10, gamma = 0.1
  - Logistic Regression Best Hyperparameters: C = 10.0, penalty = l2
  - Decision Tree Best Hyperparameters: max_depth = 7, min_samples_leaf = 3, min_samples_split = 5
  - K-nn Best Hyperparameters: n_neighbors = 7, weights = uniform
- **The measuring metric used is accuracy, which is the default and most commonly used metric. However, precision, f-score, and recall metrics can also be used.**

# 5 | Results

- Gradient Boosting is the highest performing model it is an ensemble method that combines multiple weak learners (usually decision trees) to create a strong predictive model. By iteratively improving the model based on the errors of previous iterations, Gradient Boosting can effectively capture complex patterns and interactions in the data. basically it's a super decision Tree that is trained with hyperparameters, preventing over-fitting, and with cross-validation, it also handles imbalanced data and has feature importance feature, it thrives on complicated and dependent data
- GaussianBayes is the worst performing model because it assumes that all the data is independent which is not the case here since Pdays Poutcome, campaign and deposit are dependent to each other, also it's a simple linear model that best work for linear less complex problems

# 6 | Conclusion

In conclusion, this study is based on the bank marketing dataset which includes 17 features and the target is to predict whether the customer would subscribe/deposit to the bank or not.
I have applied data preprocessing techniques such as cleaning null values, duplicates, outliers, label encoder, normalization etc...
then we experimented using multiple different classifiers and compared their accuracy's
we tried using hyperparameters (grid-search) to find the optimal parameters of each model and applied regularization to prevent over-fitting also I have used cross-validation to ensure the reliability and generalize my models
then we tried the ensemble methods then we used optimization to compare all the different models on all the different configuration
and due to the complexity of gradient-boosting algorithm which is a super decision tree in a way that thrives in feature importance we were able to gain an accuracy of 84.88%. We can try to gather more data, try other preprocessing techniques especially the unkown campaign columns to deal with them and try other hyperparameter techniques to improve our model's accuracy

**Table 5.1:** Model Accuracy

| Model | Accuracy |
|---|---|
| GradientBoost | 0.8488 |
| Bagging | 0.8298 |
| AdaBoost | 0.8256 |
| SVM | 0.8121 |
| Random Forest | 0.8051 |
| Voting Classifier | 0.8009 |
| GradientBoost_tuned | 0.7984 |
| Random Forest_tuned | 0.7969 |
| SVM_tuned | 0.7959 |
| LogisticRegression | 0.7935 |
| Extra Trees | 0.7930 |
| DecisionTree_tuned | 0.7887 |
| KNearestNeighbours | 0.7860 |
| K Neighbours_tuned | 0.7859 |
| AdaBoost_tuned | 0.7800 |
| DecisionTree | 0.7772 |
| LogisticRegression_tuned | 0.7590 |
| GaussianBayes | 0.7540 |
| GaussianBayes_tuned | 0.7433 |