ANALYZING THE CONVERSION RATE OF A BANK'S MARKETING CAMPAIGN

Author : Quang Đăng Nguyễn

Hanoi, January 14th, 2024

TABLE OF CONTENTS

LIST OF TABLES, GRAPHS, FIGURES

LIST OF TABLES

LIST OF GRAPHS, FIGURES

ABTRACT

Banking institutions can better leverage funds deposited if they are confident the funds will remain in the account longer than traditional deposit accounts. This study seeks to isolate the best performing model to predict whether a client will subscribe to a bank term deposit as a result of a telemarketing campaign. The insights from this study can also lead to an increase in success in future campaigns by indicating the most important predictors. For this study, ANN setting (4), (2), Random Forest, and SVM with RBF kernel have the highest overall performance when predicting whether a client will subscribe to a bank term deposit. To evaluate the success of our model, we considered various alternative metrics rather than accuracy alone, due to the unbalanced nature of the data. The alternative metrics considered included kappa score, precision, and balanced accuracy. When comparing our models that employed balancing techniques, we saw an increase in performance for balanced accuracy, kappa scores, and specificity. The most significant predictors were previous outcome, month, age, and contact. These predictors may indicate that the characteristics of the telemarketing campaign are important to consider and may explain a large portion of the overall success of a campaign.

Keyword: *Machine Learning, Technical analysis, Banking institutions, Telemarketing campaign.*

1. Introduction: Describe the dataset and the problem statement, what is the nature of this case?

Marketing campaigns based on phone calls are a common strategy employed by financial institutions to promote term deposits among their customers. However, accurately identifying customers who are likely to register for such deposits is crucial for optimizing campaign effectiveness. Therefore, this study aims to develop a classification and prediction model using the bank-addition-full.csv data set. This research study focuses on the classification and prediction of customer registration for term deposits in marketing campaigns. The study utilizes 41,188 examples with 20 inputs derived from the data source Bank Data. The primary objective of the study is to develop a model that can accurately classify customers as either registering (yes) or not registering (no) for term deposits based on various features. The data set covers a period from May 2008 to November 2010 and is sorted chronologically. The structure and data types of the features and targets are presented in the table below. This data set bears similarities to the data set analyzed in a previous study by Moro, Cortez, and Rita (2014).

	Name	**Description**	**Type**
Features (Independent Variables)			
Demographic data/ Bank client data	age		Integer
	job	type of job (categorical: 'admin.', 'blue-collar', 'entrepreneur',' housemaid', 'management', 'retired', 'self-employed', 'services', 'student', 'technician', 'unemployed', 'unknown')	Categorical
	marital	marital status (categorical: 'divorced', 'married', 'single', 'unknown'; note: 'divorced' means divorced or widowed)	Categorical
	education	(categorical: 'basic.4y',' basic.6y',' basic.9y',' high. school', 'illiterate',' professional. course',' university. degree', 'unknown')	Categorical
	default	Is credit in default?	Binary
	housing	has a housing loan?	Binary
	loan	has a personal loan?	Binary
Related to the last contact of the current campaign	contact	contact communication type (categorical: 'cellular', 'telephone')	Categorical
	month	last contact month of the year (categorical: 'Jan', 'Feb', 'mar', ..., 'Nov', 'Dec')	Categorical
	day_of_week	last contact day of the week	Categorical
	duration	last contact duration, in seconds (numeric). Important note: this attribute highly affects the output target (e.g., if duration=0 then y='no'). Yet, the duration is not known before a call is performed. Also, after the end of the call, y is known. Thus, this input should only be included for benchmark purposes and should be discarded if the intention is to have a realistic predictive model.	Integer
Other indicators	campaign	number of contacts performed during this campaign and for this client (numeric, includes the last contact)	Integer
	pdays	number of days that passed by after the client was last contacted from a previous campaign (numeric; -1 means the client was not previously contacted)	Integer

	previous	number of contacts performed before this campaign and for this client	Integer
	poutcome	outcome of the previous marketing campaign (categorical: 'failure',' nonexistent', 'success')	Categorical
Social and economic indicators	emp.var.rate	employment variation rate, with a quarterly frequency	Float
	cons.price.idx	monthly average consumer price index	Float
	cons.conf.idx	monthly average consumer confidence index	Float
	euribor3m	daily three month Euribor rate	Float
	nr.employed	quarterly average of the total number of employed citizens	Float
Target (Dependent Variable)			
	y	has the client subscribed a term deposit?	Binary

Table 1: Variables Information

This research report examines a dataset obtained from a telemarketing campaign conducted by a Portuguese bank. The campaign aimed to persuade potential customers to subscribe to a term deposit product. The dataset contains a comprehensive range of information, including customers' personal and financial attributes, their interactions with the bank, socio-economic factors, and the campaign's outcome (whether customers subscribed or not). The objective of this study is to develop a predictive model that can effectively identify customers who are more likely to subscribe to the term deposit product. Additionally, the study seeks to gain insights into the factors influencing customers' subscription decisions. This task entails binary classification, where the target variable is the subscription status (yes or no). By accurately predicting customer behavior, the bank can optimize its marketing resources and strategies, targeting individuals with a higher likelihood of subscribing, thus minimizing wastage of time and resources on less probable prospects.

The significance of this classification task stems from several factors. Firstly, it enables the bank to optimize its marketing efforts by focusing on customers who are more likely to subscribe. This targeted approach helps allocate resources efficiently and reduces costs associated with reaching out to less interested individuals. Secondly, by offering a product that aligns with customers' needs and preferences, the bank can enhance customer satisfaction and loyalty while avoiding unnecessary annoyance from unwanted calls. Ultimately, this leads to increased subscriptions, improved revenue, and enhanced profitability for the bank. To accomplish this task, various machine learning methods can be employed, such as decision trees, random forests, and naive Bayes, among others. These methods will be discussed in detail in Part 3 of this report. Notably, the emphasis will be placed on minimizing false negatives (recall/specificity) since missing a customer who would have subscribed incurs higher costs. Conversely, a limited number of false positives (sensitivity) is considered less impactful, as the associated loss is relatively minimal if the marketing campaign fails to attract those customers.

This perspective aligns with the trade-off between precision and recall in binary classification problems. In this case, the focus is on optimizing for recall (sensitivity) to ensure capturing as many positive instances (subscribers) as possible, even at the expense of potentially higher false positives. This strategy acknowledges that the consequences of missing positive instances outweigh dealing with some false positives. A suitable measure for model comparison is the recall of the 'yes' category in the target variable column, reflecting the subscription status, even if it may result in a slight sacrifice in accuracy.

2. Does data have any defects or issues? State the solution if any!

These are some of the common data defects or issues that can be found in the dataset:

First, fortunately, there is no missing value within the numerical variables. However, there are variables of clients' information that contain an 'Unknown' value which is missing values.

Second, machine learning models often require numerical input, and dealing with categorical variables might be necessary. Thus, encode categorical variables using techniques like one-hot encoding or label encoding.

Third, the target variable (y) is highly imbalanced, as only 11.27% of the examples belong to the positive class (yes), while 88.73% belong to the negative class (no). This means that the models may be biased toward the majority class and fail to capture the characteristics of the minority class. A possible solution is to apply some resampling techniques, such as oversampling, undersampling, or synthetic minority oversampling technique (SMOTE).

Forth, some of the numerical attributes, such as age, duration, campaign, pdays, previous, and cons.price.idx, may have outliers, which are extreme values that deviate significantly from the rest of the data. Outliers can distort the distribution and statistics of the data, and affect the accuracy and robustness of the models. A possible solution is to detect and remove the outliers using some methods, such as box plots, z-scores, or interquartile range (IQR).

Fifth, some machine learning algorithms are sensitive to the scale of input features. Thus, standardize or normalize numerical features to ensure they have similar scales.

Sixth, highly correlated features may provide redundant information, which doesn't contribute significantly to the model's predictive power. Correlated features can also make it difficult to interpret the model and understand the true impact of each feature on the predictions. Thus, remove redundant features to simplify the model without sacrificing performance and prioritize feature selection or engineering to create a more interpretable and understandable model.

Moreover, there may be other issues that are specific to the domain or the problem, such as feature relevance, feature correlation, feature scaling, etc. Therefore, it is important to perform a thorough exploratory data analysis (EDA) and data preprocessing before building any predictive models. However, there are also some other limitations regarding this dataset but they depend on the bank data collection team to fix.

One of the limitations of the dataset was the absence of customer balance data, which is an important factor in predicting deposit subscriptions. Another dataset from the UCI website had this information, but it was not possible to combine the two datasets without a common key or index. The dataset also did not include any information about bank interest rates, the clients’ religious beliefs, measurements of relationships, year, or geographic location. The size and network of the branches may have influenced the deposit mobilization as well. Other aspects that could affect the campaign outcome were the bank’s reputation and other marketing channels, such as TV, radio, and online.

Moreover, the bank’s other financial products, such as call accounts and savings accounts, could also play a role. The data on the bank staff who conducted the campaign was also inadequate. The staff’s gender, education level, work experience, and communication skills could have an impact on the results. Furthermore, there was no information about the competitive situation in the banking sector, or other financial institutions such as insurance companies, investment management firms, pension funds, and similar deposit institutions. The research objective was to examine the bank’s existing customers. Although there was inflation data that could indicate the cost of living, the overall macroeconomic situation could also influence the deposit subscription.

3. What kind of model could be used in this case? Explain

There are various kinds of models that could be used for this case, depending on the data characteristics, the performance criteria, and the interpretability requirements. Some of the possible models are:

First, decision tree: These are simple and intuitive models that use a tree-like structure to split the data into homogeneous groups based on some criteria. They can handle both numerical and categorical features and can provide a clear explanation of how the predictions are made. However, they may suffer from overfitting, instability, and bias problems.

Second, random forests: These are ensemble models that combine multiple decision trees and use a majority vote or an average to make the final prediction. They can improve the accuracy, robustness, and generalization of decision trees, and can also measure the feature importance. However, they may lose some interpretability, require more computational resources, and may not perform well on imbalanced data.

Third, naive Bayes: These are probabilistic models that use Bayes’ theorem and assume conditional independence among the features. They can handle both numerical and categorical features and can provide a probabilistic output. They are also fast, simple, and scalable. However, they may not capture the feature interactions and may perform poorly if the independence assumption is violated

Third, support vector machines: These are linear models that use a kernel function to map the data into a higher-dimensional space, and find the optimal hyperplane that separates the classes with the maximum margin. They can handle both linear and non-linear problems and can achieve high accuracy and generalization. However, they may be sensitive to outliers, require parameter tuning, and may not be very interpretable

Fourth, deep learning: These are neural network models that use multiple layers of non-linear transformations to learn complex and abstract features from the data. They can handle both numerical and categorical features and can achieve state-of-the-art performance on various tasks. However, they may require a large amount of data, computational power, and time, and may not be very transparent or explainable

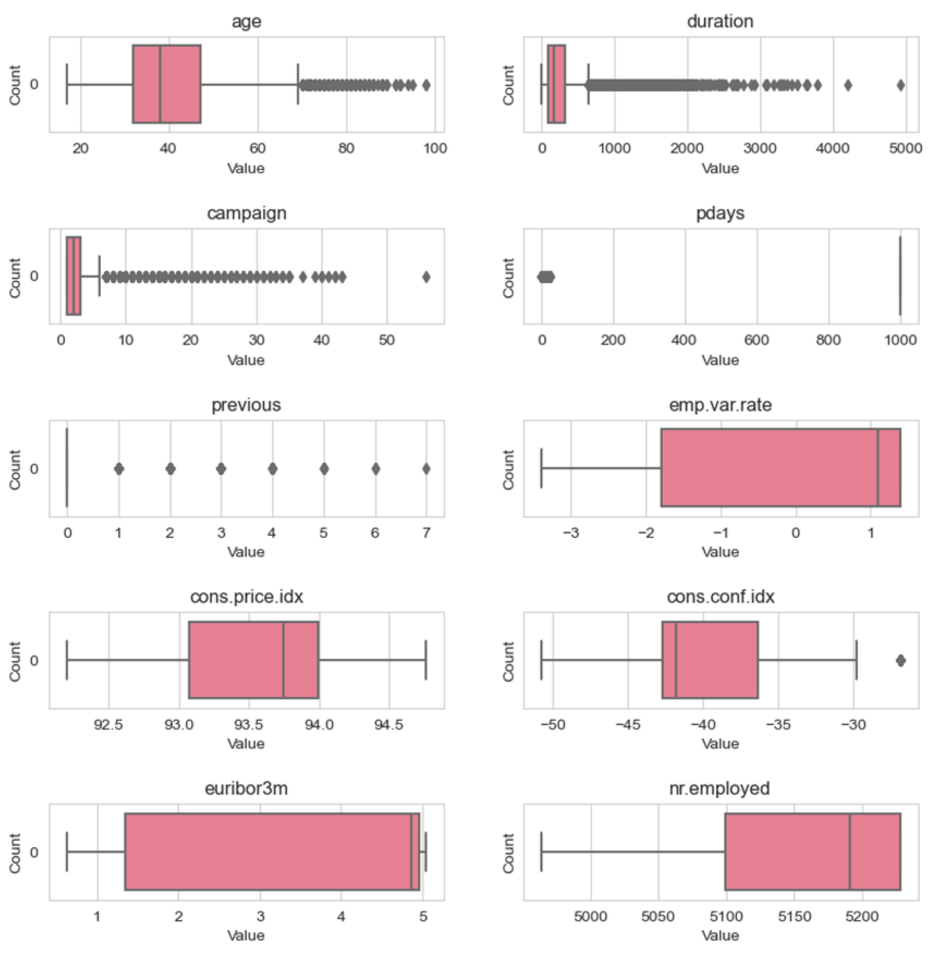
Fifth, logistic regression: This is a linear model that uses a logistic function to map the data into a probability between 0 and 1, and then classify the data into two classes based on a threshold. It can handle both numerical and categorical features and can provide a probabilistic output. It is also fast, simple, and interpretable. However, it may not capture the non-linear relationships, the feature interactions, or the class imbalance in the data.

Sixth, k-nearest neighbors: This is a non-parametric model that uses the distance or similarity between the data points to assign the class label of a new point based on the majority vote of its k closest neighbors. It can handle both numerical and categorical features and can adapt to the local structure of the data. It is also intuitive and easy to implement. However, it may be sensitive to noise, outliers, and irrelevant features, and may require a lot of computational resources and time to find the nearest neighbors

These are some of the common models that could be used for this case, but there may be other models that are also suitable, such as gradient boosting, etc. The choice of the model depends on the specific problem, the data availability and quality, and the desired outcome. Therefore, it is important to compare and evaluate different models and select the best one for the task.

4. Perform data exploratory analysis (you could use descriptive analysis or charts)!

4.1 Quantitative Variables



Graph 1: Boxplots of Numerical Variables

The boxplots above represent various parameters from the bank dataset. Each plot illustrates the distribution of a specific variable, including age, duration, campaign, pdays, previous, emp.var.rate, cons.price.idx, cons.conf.idx, euribor3m and nr.employed.

Age: The box plot shows that the age distribution is skewed to the right, meaning that most clients are younger than the median age of 39. The outliers indicate that there are some very old clients, up to 98 years old.

Duration: The box plot shows that the duration distribution is also skewed to the right, meaning that most contacts lasted less than the median duration of 180 seconds. The outliers suggest that there are some very long contacts, up to 4918 seconds.

Campaign: The box plot shows that the campaign distribution is skewed to the right as well, meaning that most campaigns contacted clients fewer times than the median number of 2. The outliers imply that there are some campaigns that contacted clients very frequently, up to 56 times.

Pdays: The box plot shows that the pdays distribution is highly concentrated at one value, 999, which means that most clients were not previously contacted by the bank. The few values below 999 indicate that some clients were contacted in the past, up to 27 days before the current campaign.

Previous: The box plot shows that the previous distribution is also highly concentrated at one value, 0, which means that most clients have not been contacted before this campaign. The few values above 0 indicate that some clients have been contacted before, up to 7 times.

Emp.var.rate: The box plot shows that the emp.var.rate distribution has some distinct peaks and valleys, which reflect the changes in the employment variation rate over time. The values range from -3.4 to 1.4, with a median of 1.1.

Cons.price.idx: The box plot shows that the cons.price.idx distribution is slightly skewed to the left, meaning that most values are higher than the median of 93.75. The values range from 92.2 to 94.77, with some outliers below 92.2.

Cons.conf.idx: The box plot shows that the cons.conf.idx distribution is skewed to the right, meaning that most values are lower than the median of -40.5. The values range from -50.8 to -26.9, with an outlier above -26.9.

Euribor3m: The box plot shows that the euribor3m distribution has some distinct peaks and valleys as well, which reflect the changes in the Euribor three-month rate over time. The values range from 0.63 to 5.04, with a median of 4.86.

Nr.employed: The box plot shows that the nr.employed distribution has two distinct clusters, one around 5099 and one around 5228, which correspond to the number of employees in the bank at different periods. The median is 5191, and there are no outliers. In short, some variables such as age, duration, campaign, pdays, previous, cons.conf.idx should remove outliers.

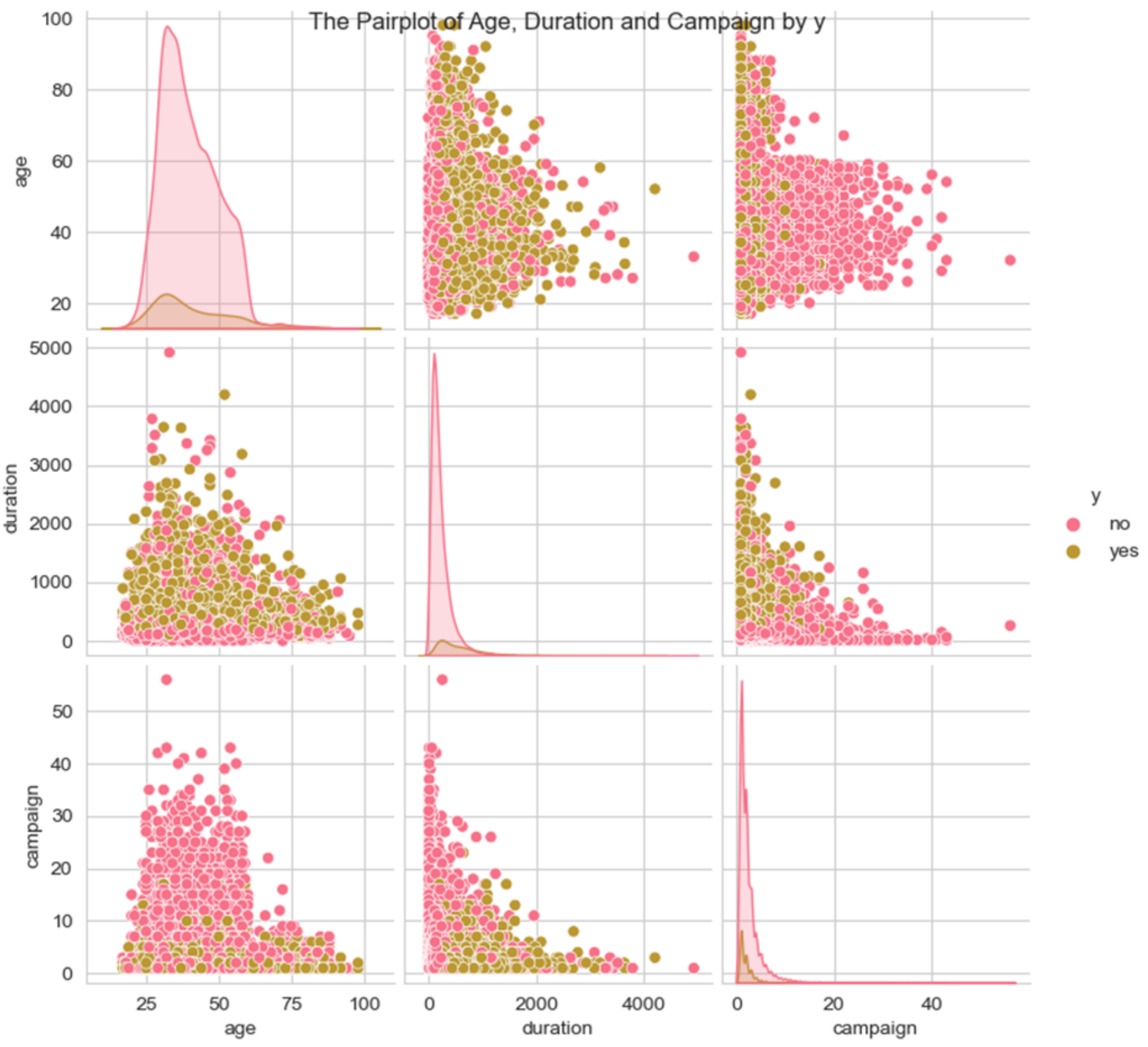
The table below shows more details about max, min, mean, median of these numerical variables:

	age	**duration**	**campaign**	**pdays**	**previous**	**emp.var. rate**
count	41188	41188	41188	41188	41188	41188
mean	40.02406	258.285	2.567593	962.4755	0.172963	0.081886
std	10.42125	259.2792	2.770014	186.9109	0.494901	1.57096
min	17	0	1	0	0	-3.4
25%	32	102	1	999	0	-1.8

50%	38	180	2	999	0	1.1
75%	47	319	3	999	0	1.4
max	98	4918	56	999	7	1.4

	cons.price.idx	**cons.conf.idx**	**euribor3m**	**nr.employed**
count	41188	41188	41188	41188
mean	93.57566	-40.5026	3.621291	5167.036
std	0.57884	4.628198	1.734447	72.25153
min	92.201	-50.8	0.634	4963.6
25%	93.075	-42.7	1.344	5099.1
50%	93.749	-41.8	4.857	5191
75%	93.994	-36.4	4.961	5228.1
max	94.767	-26.9	5.045	5228.1

Table 2: Descriptive statistics of numerical variables

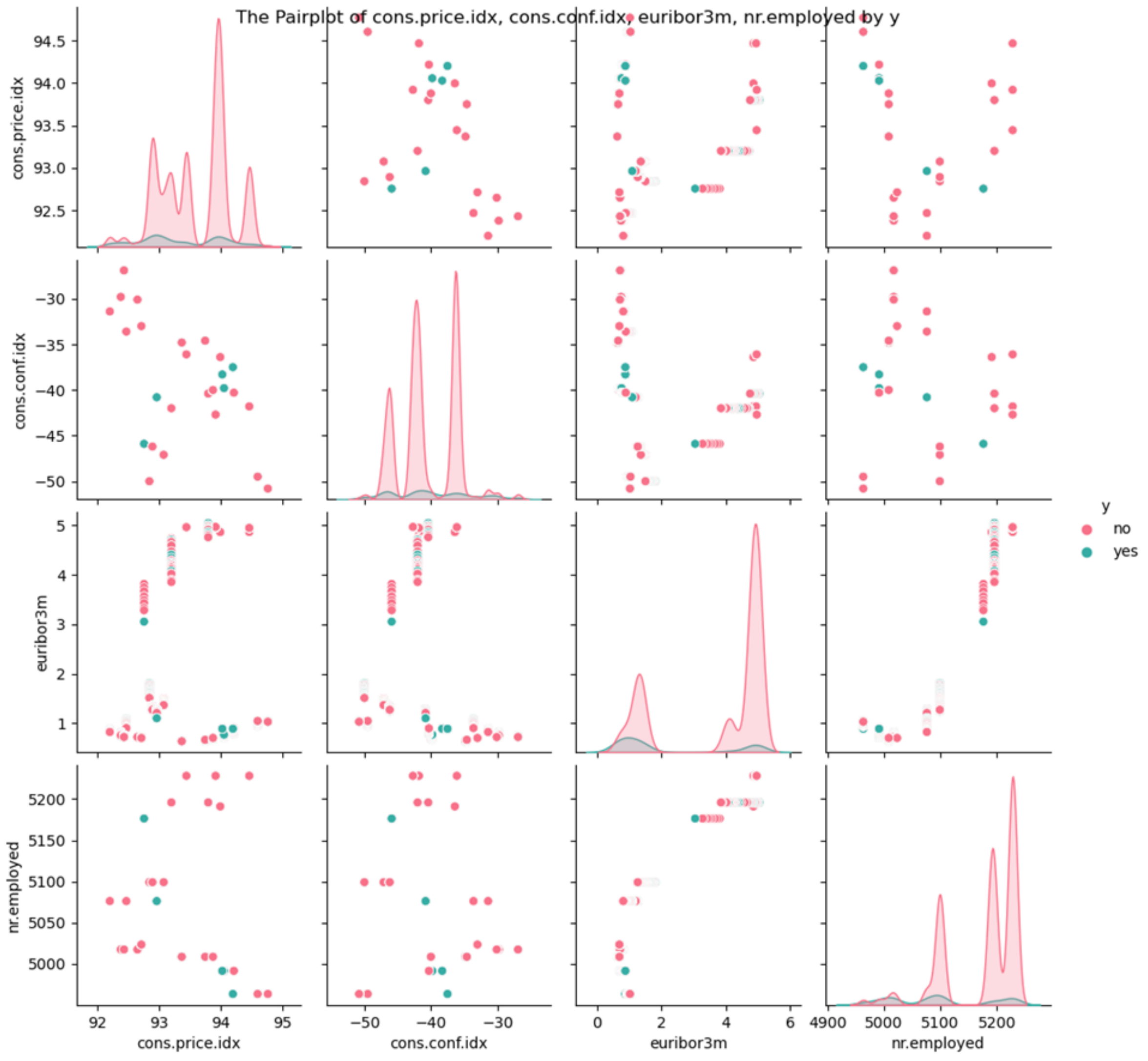


Graph 2: Density Plots of Subscription by Customer Information

The plot shows the relationship between three variables: age, duration, and campaign. The plot consists of a matrix of scatter plots and density plots, colored by the subscription of deposit (y). Here are some observations from the plot:

The density plot shows that most of the clients are between 30 and 40 years old. The scatter plots show that there is no clear correlation between age and duration or campaign. However, there seems to be a slight tendency for older clients to subscribe more than younger ones.

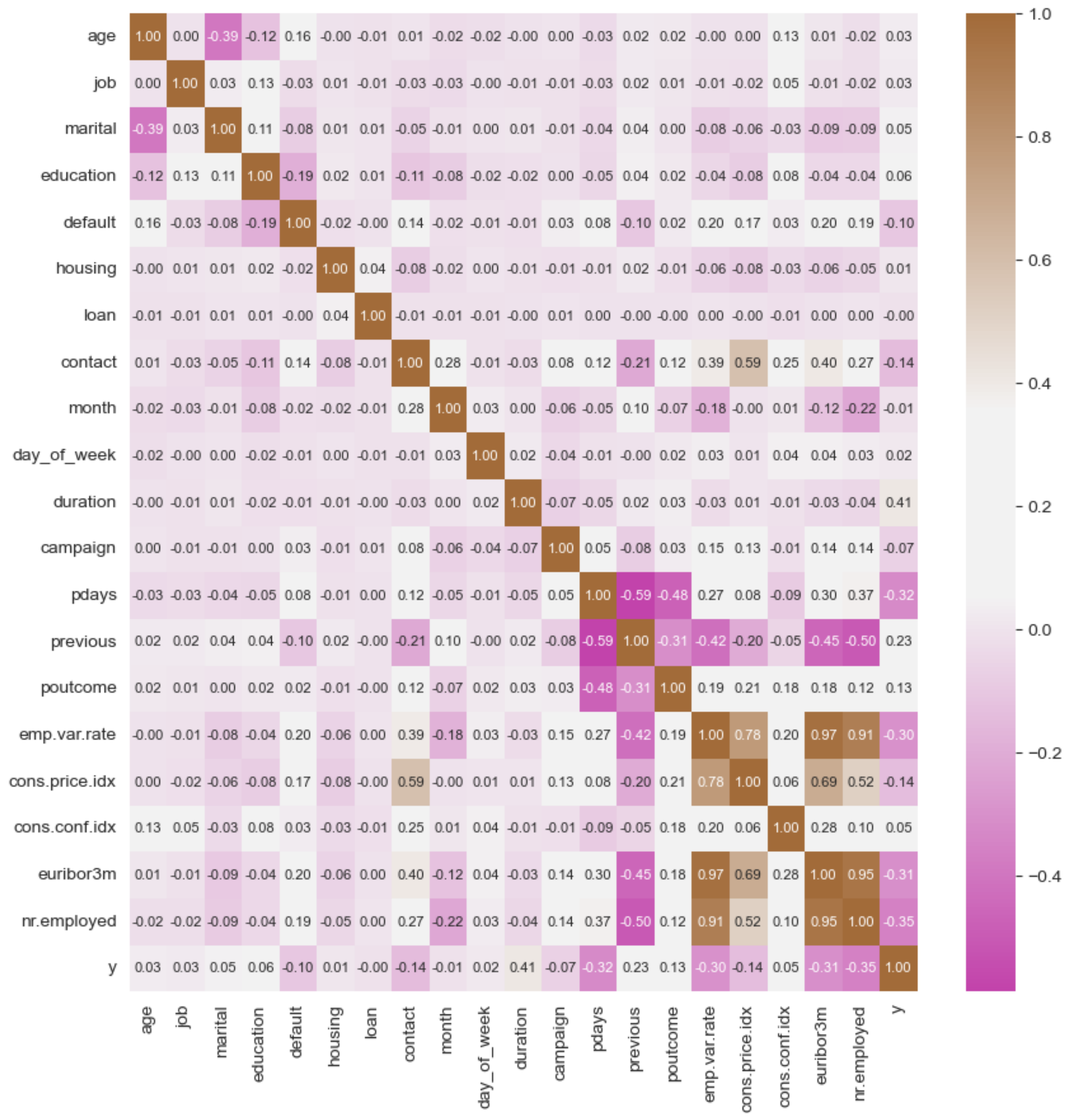
Also, the density plot shows that most of the calls have a duration of less than 500 seconds. The scatter plots show that there is a positive correlation between duration and subscription, meaning that longer calls are more likely to result in a subscription. However, there are also some outliers with very long calls that did not result in a subscription. The density plot shows that most of the campaigns involved less than 10 contacts. The scatter plots show that there is a negative correlation between campaign and subscription, meaning that more contacts are less likely to result in a subscription. However, there are also some outliers with very high number of contacts that resulted in a subscription.



Graph 3: Density Plots of Subscription by consumer price index, consumer confidence index, euribor 3m, and number of employed people

The plot shows the relationship between various economic indicators (consumer price index, consumer confidence index, daily three month Euribor rate, and number of employed people) and subscription (yes/no). The data is visualized as scatter plots and density plots. It appears that there isn't a clear pattern or correlation visible between the economic indicators and subscription status in the scatter plots. The density plots show the distribution of each economic indicator. On the diagonal line from the top left to bottom right are histograms showing the distribution of each variable. The distribution of these variables as discussed with boxplots, “nr.employed” is skewed to the left, and “cons.conf.idx” is skewed to the right.

The graph consists of multiple scatter plots and histograms arranged in a grid format. Each plot is labeled with economic indicators such as “cons.price.idx”, “cons.conf.idx”, “euribor3m”, and “employed”. There are two distinct colors used to represent data points: red for ‘no’ subscription, and green for ‘yes’. However, there doesn’t appear to be any clear trend or pattern in most scatterplots; data points are dispersed randomly.



Graph 4: Correlation Matrix between all variables

This is a correlation matrix that shows the correlation coefficients between different variables. The color scale on the right side of the image indicates the strength of the correlation, ranging from -0.4 to 1.0. The diagonal line from the top left to bottom right consists of 1.0s, indicating a perfect positive correlation between identical variables, as expected. Most coefficients are close to zero, suggesting weak correlations between different variables.

Based on the correlation matrix, there are a few pairs of variables that have high correlation coefficients. For example, the following pairs of variables have a correlation coefficient greater than 0.7:

emp.var.rate and cons.price.idx have a correlation coefficient of 0.78

emp.var.rate and euribor3m have a correlation coefficient of 0.97.

nr.employed and euribor3m have a correlation coefficient of 0.94.

emp.var.rate and nr.employed have a correlation coefficient of 0.91

	feature	VIF
0	age	1.02
1	duration	1.01
2	campaign	1.04
3	pdays	1.61
4	previous	1.80
5	emp.var.rate	33.07
6	cons.price.idx	6.34
7	cons.conf.idx	2.65
8	euribor3m	64.35
9	nr.employed	31.68

Table 3: VIF indicators of numerical variables

The table provided displays the Variance Inflation Factor (VIF) for each variable in a dataset. VIF is a measure of how much the variance of an estimated regression coefficient increases if your predictors are correlated. High VIF values indicate potential multicollinearity, which can lead to unstable and less reliable regression coefficients.

In summary, several variables exhibit high VIF values, suggesting strong multicollinearity. It's essential to address multicollinearity issues as it can affect the stability and interpretability of your regression model. Consider exploring relationships between highly correlated variables, removing redundant predictors, or using regularization techniques to mitigate multicollinearity. A common rule is to remove variable with VIF greater than 5 to avoid multicollinarity.

emp.var.rate: VIF = 33.07 - A high VIF, suggesting potential multicollinearity issues with other variables. Consider exploring relationships and addressing any correlation concerns.

cons.price.idx: VIF = 6.34 - Elevated VIF, indicating some multicollinearity. It may be worth investigating relationships with other predictors.

euribor3m: VIF = 64.35 - A high VIF, indicating potential multicollinearity. Investigate relationships with other predictors and consider addressing this issue.

nr.employed: VIF = 31.68 - A high VIF, suggesting potential multicollinearity. Examine relationships with other variables to ensure model stability.

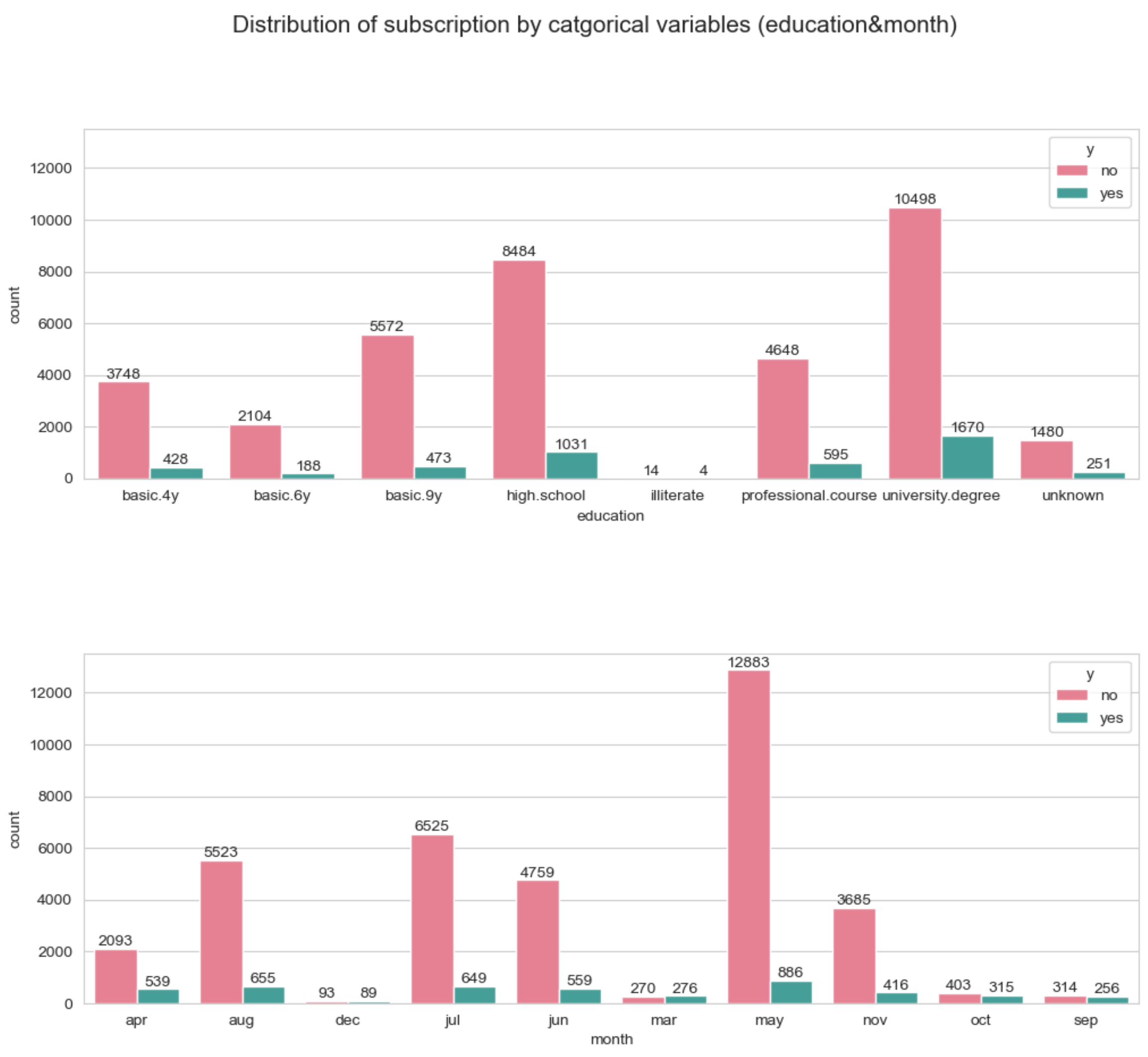
4.2 Categorical Variables

Records of Number of ‘Unknown’ values in 6 customers’ categorical variables

	TOTAL RECORDS	UNKNOWN	% UNKNOWN
JOB	41188	330	1.0%
MARITAL	41188	80	0.0%
EDUCATION	41188	1731	4.0%
DEFAULT	41188	8597	21.0%
HOUSING	41188	990	2.0%
LOAN	41188	990	2.0%

The table shows the number and percentage of unknown records for six categorical variables in the bank-additional-full.csv dataset. The variables with the highest number and percentage of unknown records are default (21%) and education (4%). The variables with the lowest number and percentage of unknown records are marital (0%) and job (1%). The variables housing and loan have the same number and percentage of unknown records (2%). The issue of unknown records can affect the quality and reliability of data analysis and modeling. Unknown data can introduce bias, reduce statistical power, and limit the generalizability of results. Since it is not quite easy to acquire full information of customers and it is usually expensive to do that so we should treat unknown value as a category in this problem. As a result, this research will use ‘Unknown’ value as a missingness indicator or dummy variables to account for the presence of unknown values in the analysis because of avalability of the data. The missing values are already recorded as “Unknown”.

Checking for distribution of categorical variables within the dataset

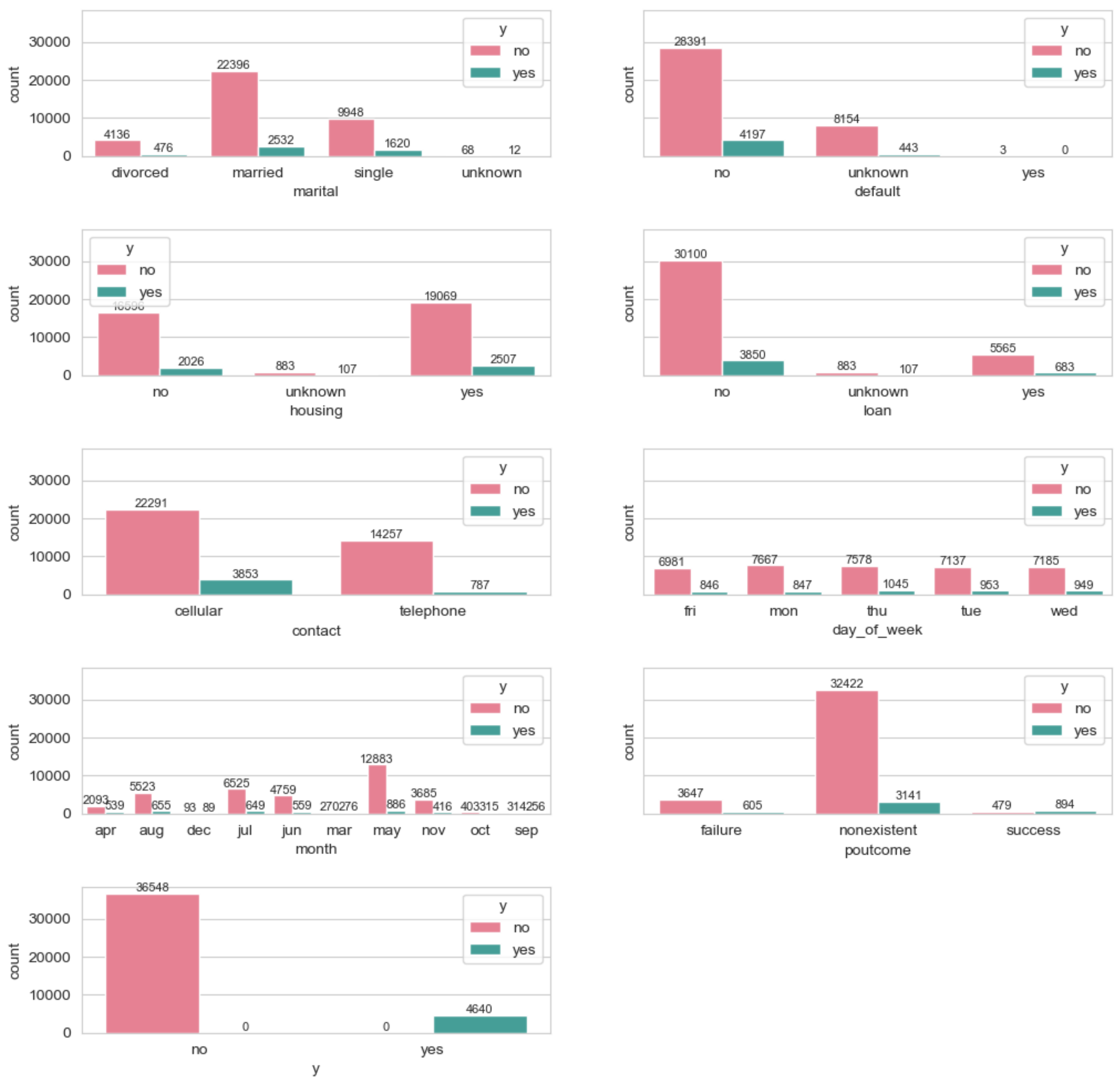


Graph 5: Countplot of subscription by Job&Education

The countplot of y by some categorical variables shows the distribution of subscription to a deposit (y) by different job types. The plot has two bars for each job type, one for ‘yes’ and one for ‘no’, indicating whether the clients subscribed or not. The bars are colored green and red, respectively. The plot also shows the count values on each bar.

Job: The countplot shows that the majority of the clients have an admin job (10,472), followed by those who have a blue-collar (9,250) or a technician (6,756) job. The unknown job category has the lowest count (330). The proportion of subscribers is highest for the student (31.4%) and retired (25.2%) categories, followed by the unemployed (14.2%) and admin (13.5%) categories. The blue-collar (6.9%) and services (8.1%) categories have the lowest proportions of subscribers. This suggests that job type may have an influence on the decision to subscribe to the deposit.

Education: The countplot shows that the majority of the clients have a university degree (12,172), followed by those who have a high school education (9,524). The illiterate category has the lowest count (18). The proportion of subscribers is highest for the illiterate category (22.2%), followed by the university degree category (13.7%). The basic.4y, basic.6y, and basic.9y categories have the lowest proportions of subscribers (10.4%, 8.8%, and 7.8%, respectively). This suggests that education level may have an influence on the decision to subscribe to the deposit.



Graph 6: Barplots of y (subscription) by categorical variables

Here are some detailed comments on the countplots of y (deposit subscription) by categorical variables ('marital','default', 'housing', 'loan','contact', 'day_of_week','month','poutcome','y'):

The first countplot shows that the majority of the clients are married (24,924), followed by single (11,520) and divorced (4,614). The unknown category has the lowest count (80). The proportion of subscribers is higher for the single category (14%), followed by the divorced (10.3%) and married (10.2%) categories. The unknown category has the lowest proportion of subscribers (15%). This suggests that marital status may have an influence on the decision to subscribe to the deposit.

The second countplot shows that the majority of the clients have a housing loan (21,526), followed by those who do not have a housing loan (18,664). The unknown category has the lowest count (990). The proportion of subscribers is slightly higher for those who do not have a housing loan (10.8%), followed by those who have a housing loan (10.5%). The unknown category has the lowest proportion of subscribers (10.8%). This suggests that having a housing loan may have a negative impact on the decision to subscribe to the deposit.

The third countplot shows that the majority of the clients do not have a personal loan (33,984), followed by those who have a personal loan (6,206). The unknown category has the lowest count (990). The proportion of subscribers is higher for those who do not have a personal loan (10.7%), followed by those who have a personal loan (9.8%). The unknown category has the lowest proportion of subscribers (10.8%). This suggests that having a personal loan may have a negative impact on the decision to subscribe to the deposit.

The forth countplot shows that the majority of the clients were contacted by cellular phone (26,185), followed by those who were contacted by telephone (14,995). The proportion of subscribers is higher for those who were contacted by cellular phone (14.7%), followed by those who were contacted by telephone (5.2%). This suggests that the type of contact may have an influence on the decision to subscribe to the deposit.

The fifth countplot shows that the majority of the clients were contacted in May, followed by July and August. The December, March, October, and September months have the lowest counts. The proportion of subscribers is highest for the December, March, October, and September months, followed by April and November. The May month has the lowest proportion of subscribers. This suggests that the month of the last contact may have a significant influence on the decision to subscribe to the deposit.

The sixth countplot shows that the clients were contacted almost evenly across the different days of the week, with Thursday having the highest count and Friday having the lowest count. The proportion of subscribers is slightly higher for Thursday and Tuesday, followed by Monday and Wednesday. The Friday day has the lowest proportion of subscribers. This suggests that the day of the week of the last contact may have a minor influence on the decision to subscribe to the deposit.

The seventh countplot shows that the majority of the clients have a nonexistent outcome of the previous marketing campaign, followed by those who have a failure outcome. The success outcome has the lowest count. The proportion of subscribers is highest for the success outcome, followed by the nonexistent and failure outcomes. This suggests that the outcome of the previous marketing campaign may have a strong influence on the decision to subscribe to the deposit.

Finally, the countplot of y shows the distribution of subscription to a deposit (y) in a dataset. The plot has two bars, one for ‘yes’ and one for ‘no’, indicating whether the clients subscribed or not. The bars are colored green and red, respectively. The plot also shows the count values on each bar.

The plot reveals that the majority of the clients did not subscribe to the deposit, as the ‘no’ bar is much higher than the ‘yes’ bar. The count of ‘no’ responses is about 36,000, while the count of ‘yes’ responses is about 4,600. This means that only about 11% of the clients subscribed to the deposit. This suggests that the deposit product was not very popular or attractive among the clients.

This means that the models may be biased towards the majority class and fail to capture the characteristics of the minority class. A possible solution is to apply some resampling techniques, such as oversampling, undersampling, or synthetic minority oversampling technique (SMOTE).

5. Is there any special point or potential issue that the analyst must pay attention to? (Handling the data)

5.1 Missing values

As previously mentioned, there are some significant issues that occur in the dataset like imbalance output or outliers. Fortunately, there is no missing value within the **numerical** variables. However, there are variables of clients’ information that contain ‘Unknown’ value which is literally missing values. The issue of missing values can affect the quality and reliability of data analysis and modeling. Unknown data can introduce bias, reduce statistical power, and limit the generalizability of results. Therefore, it is important to handle these data appropriately, depending on the type and amount of missingness, the nature of the variables, and the research question. Some possible recommendations for handling unknown records are:

- Impute unknown values using statistical methods such as mean, median, mode, regression, or multiple imputation.

- Remove unknown values from the dataset if they are few and randomly distributed, or if they are not relevant to the research question.

- Use missingness indicators or dummy variables to account for the presence of unknown values in the analysis.

- Collect more data or information to reduce the amount of missingness in the dataset.

Since it is not quite easy to acquire full information of customers and it is usually expensive to do that so we should treat unknown value as a category in this problem. As a result, this research will use ‘Unknown’ value as a missingness indicator or dummy variables to account for the presence of unknown values in the analysis because of avalability of the data. The missing values are already recorded as “Unknown”.

5.2 Outliers

Some of the numerical attributes, such as age, duration, campaign, pdays, previous, and cons.price.idx, may have outliers, which are extreme values that deviate significantly from the rest of the data. Outliers can distort the distribution and statistics of the data, and affect the accuracy and robustness of the models. A possible solution is to detect and remove the outliers using some methods, such as box plots, z-scores, or interquartile range (IQR).

Since the age, duration, and campaign, pdays, previous, and cons contain a large number of outliers as described and by looking at the difference between means and max values from table 2 and boxplot 1, the research will remove outliers from these variables by using the 1.5 * IQR rule. Here is the example code to remove outliers:

```
def remove_outliers(df, col_name):
  Q1 = df[col_name].quantile(0.25)
  Q3 = df[col_name].quantile(0.75)
  IQR = Q3 - Q1
  df_filtered = df[(df[col_name] >= Q1 - 1.5 * IQR) & (df[col_name] <= Q3 + 1.5 * IQR)]
  return df_filtered


col_names = ['age', 'duration', 'campaign', 'pdays', 'previous', 'cons.conf.idx']


for col_name in col_names:
  bank_df = remove_outliers(bank_df, col_name)
```

The 1.5 * IQR rule is a commonly used method for identifying outliers in a dataset. The IQR (interquartile range) is the range between the first quartile (Q1) and the third quartile (Q3) of the dataset. The rule states that any data point that falls below Q1 - 1.5 * IQR or above Q3 + 1.5 * IQR is considered an outlier. The reason for using 1.5 * IQR is to balance the sensitivity of the range and the decision rule. A larger value would make the outlier(s) to be considered as data point(s) while a smaller one would make some of the data point(s) to be perceived as outlier(s).

After removing the outliers, here is the result of the dataset:

	age	**duration**	**campaign**	**pdays**	**previous**	**emp.var.rate**
count	30360	30360	30360	30360	30360	30360
mean	39.67108	204.9326	2.081555	999	0	0.388261
std	9.480333	140.2143	1.3027	0	0	1.418314
min	18	0	1	999	0	-3.4
25%	32	100	1	999	0	-0.1
50%	38	167	2	999	0	1.1
75%	47	277	3	999	0	1.4
max	69	644	6	999	0	1.4

	cons.price.idx	**cons.conf.idx**	**euribor3m**	**nr.employed**
count	30360	30360	30360	30360
mean	93.64646	-40.4101	3.970239	5181.932
std	0.538624	4.164977	1.556323	60.6395
min	92.201	-50.8	0.634	4963.6
25%	93.2	-42.7	4.021	5191
50%	93.918	-41.8	4.859	5195.8
75%	93.994	-36.4	4.962	5228.1
max	94.767	-29.8	5.045	5228.1

5.3 Multicolinarity

Highly correlated features may provide redundant information, which doesn't contribute significantly to the model's predictive power. Correlated features can also make it difficult to interpret the model and understand the true impact of each feature on the predictions. Thus, remove redundant features to simplify the model without sacrificing performance and prioritize feature selection or engineering to create a more interpretable and understandable model.

Based on the VIF table (Table 3), this research removes all variables that have VIF > 5. These variables include:

emp.var.rate: VIF = 33.07 - A high VIF, suggesting potential multicollinearity issues with other variables. Consider exploring relationships and addressing any correlation concerns.

cons.price.idx: VIF = 6.34 - Elevated VIF, indicating some multicollinearity. It may be worth investigating relationships with other predictors.

euribor3m: VIF = 64.35 - A high VIF, indicating potential multicollinearity. Investigate relationships with other predictors and consider addressing this issue.

nr.employed: VIF = 31.68 - A high VIF, suggesting potential multicollinearity. Examine relationships with other variables to ensure model stability.

```
data_no_multicolinarity = bank_df.drop(['cons.price.idx','emp.var.rate','euribor3m','nr.employed'], axis=1)
```

5.4 Encoding categorical variables:

Machine learning models often require numerical input, and dealing with categorical variables might be necessary. Thus, encode categorical variables using techniques like one-hot encoding or label encoding. As previously discussed, all categorical variables including features that are job, marital, education, default, housing, loan, contact, month, day_of_week, poutcome, and the target are y (subscription) encoded by the one-hot encoding technique.

```
# Select the columns to encode

cols_to_encode = ['job','marital','education','default','housing', 'loan',

            'contact','month','day_of_week','poutcome']

# To include the categorical data in the regression, let's create dummies

# There is a very convenient method called: 'get_dummies' which does that seemlessly

# It is extremely important that we drop one of the dummies, alternatively we will introduce multicollinearity

df_with_dummies = pd.get_dummies(data_no_multicolinarity, columns=cols_to_encode, drop_first=True, dtype=int)

df_with_dummies.columns
```

5.5 Overfitting

Overfitting is a concept in data science, which occurs when a statistical model fits exactly against its training data. When this happens, the algorithm unfortunately cannot perform accurately against unseen data, defeating its purpose. Generalization of a model to new data is ultimately what allows us to use machine learning algorithms every day to make predictions and classify data.

For this problem, using training, validation, and testing is one of the solutions to prevent overfitting for training data. First, the model will be trained on the training set, and then tested on the validation test to find a better threshold for classification. Finally, the model will be evaluated by the test set which is totally new data for the model.

```
from sklearn.model_selection import train_test_split
# Split the data into train, validation, and test sets
X_train, X_temp, y_train, y_temp = train_test_split(X, y, test_size=0.2, random_state=0)
X_valid, X_test, y_valid, y_test = train_test_split(X_temp, y_temp, test_size=0.5, random_state=0)
```

5.6 Imbalanced output classes

The target variable (y) is highly imbalanced, as only 11.27% of the examples belong to the positive class (yes), while 88.73% belong to the negative class (no). This means that the models may be biased towards the majority class and fail to capture the characteristics of the minority class. A possible solution is to apply some resampling techniques, such as oversampling, undersampling, or synthetic minority oversampling technique (SMOTE). This research will use synthetic minority oversampling technique (SMOTE) because of some reasons. By providing additional synthetic examples for the minority class, enabling more accurate learning of minority class patterns, SMOTE expands the feature space, creating more flexible decision boundaries that better capture the complexities of the minority class, contributing to improved classification accuracy. Not only does SMOTE help prevent overfitting on the majority class by introducing diversity in the dataset, making the model less likely to memorize specific instances and ensuring better generalization, but it also is adaptable to a variety of machine learning algorithms, making it a versatile tool for addressing imbalanced datasets across different modeling approaches.

Furthermore, the implementation of SMOTE is straightforward, with readily available libraries in popular programming languages, making it accessible for researchers and practitioners dealing with imbalanced classification problems. SMOTE's focus on generating synthetic samples based on the feature space enhances its robustness to noisy data, making it beneficial in real-world scenarios where datasets may contain outliers or noise. This SMOTE will be done for the training set after splitting the data into 3 parts training, validation, and testing dataset. Because it is unnecessary to apply SMOTE on validation or test set since the test set must be totally new to the model so that the metrics show authentic and reliable results.

```
from imblearn.over_sampling import SMOTE
# Apply SMOTE to oversample the minority class
smote = SMOTE(random_state=42)
X_train, y_train = smote.fit_resample(X_train, y_train)
```

5.7 Scaling input features:

Some machine learning algorithms are sensitive to the scale of input features. Thus, standardize or normalize numerical features to ensure they have similar scales. These are some of the common data defects or issues that can be found in the dataset. However, there may be other issues that are specific to the domain or the problem, such as feature relevance, feature correlation, feature scaling, etc. This research will standardize numerical inputs after splitting the data into 3 parts training, validation, and testing dataset. Because it is unnecessary to apply standardization on validation or test set since the test set must be totally new to the model so that the metrics show authentic and reliable results.

Putting all numerical variables on the left side of the dataframe:

```
from sklearn.preprocessing import StandardScaler
columns_to_scale = X.columns[:4]
sc = StandardScaler()
X_train[columns_to_scale] = sc.fit_transform(X_train[columns_to_scale])
X_test[columns_to_scale] = sc.transform(X_test[columns_to_scale])
```

5.8 Threshold Tuning for Machine Learning Model:

ROC and the Youden Index will be used to locate the optimal threshold. The Receiver Operating Characteristic (ROC) curve is a graphical representation of the performance of a binary classifier system as its discrimination threshold is varied. It plots the true positive rate (TPR) against the false positive rate (FPR) at various threshold settings. The area under the ROC curve (AUC) is a measure of the classifier’s ability to distinguish between positive and negative classes. A perfect classifier has an AUC of 1, while a random classifier has an AUC of 0.5 . The Youden Index is a statistic that summarizes the ROC curve by providing a single number that represents the maximum potential effectiveness of a diagnostic test. It is defined as the maximum vertical distance between the ROC curve and the diagonal line connecting the points (0,0) and (1,1) . The Youden Index is used to determine the optimal threshold for a binary classifier by maximizing the difference between the true positive rate and the false positive rate . By using the ROC curve and the Youden Index, we can identify the optimal threshold for a binary classifier that maximizes the classifier’s ability to distinguish between positive and negative classes. This threshold is the point on the ROC curve that maximizes the Youden Index . Once we have identified the optimal threshold, we can use it to classify new data points as positive or negative based on their predicted probabilities .

Once we train the model on the training set. The ROC curve and the Youden Index will be used to identify the optimal threshold by evaluating the validation set. When the optimal threshold is found on the validation set, the model will apply this threshold to classify new data points on the test set based on the predicted probabilities. The reason we should apply this threshold finding method on the validation set only is that if we apply this method on the training set, the model will be overfitted on the training set, the model is already trained to best fit the training set and now is set the threshold based on that information leads to an overfitting situation on the training set. On the other hand, applying this threshold finding method to the test set will wreck the rudimentary rule of the test set where it was built to be new to the model so that we can evaluate the model performance objectively. If you optimize the threshold on the test set, you will still end up with the best-performing model, but your estimate of the performance on unseen data may be overly optimistic. Therefore, it is better to use a validation set to tune the threshold and then use the test set to estimate the performance of the model on unseen data.

6. Model Performance, Evaluation, and Recommendation.

For each model, the data preprocessed will be first separated into training, validation, and testing as described in the previous section to prevent overfitting. Then, the training data will be applied the synthetic minority oversampling technique (SMOTE) to address imbalanced output data. Then, the numerical features (age, duration, campaign, and cons.conf.idx) are standardized. Now that the training set is ready for our machine learning model to train on. After training the model on the training set, I will use ROC and Youden Index on the validation set to tune the threshold. Finally, I will evaluate the performance of the model on the test set both when tuned and not tuned threshold.

6.1 Logistic Regression

Logistic regression is a statistical model that estimates the probability of an event occurring based on a given dataset of independent variables. It can be used for classification and predictive analytics, and it produces a probability between 0 and 1 . The logistic regression model is a type of regression analysis that models the log-odds of an event as a linear combination of one or more independent variables. In binary logistic regression, there is a single binary dependent variable, coded by an indicator variable, where the two values are labeled “0” and “1”, while the independent variables can each be a binary variable (two classes, coded by an indicator variable) or a continuous variable (any real value). The corresponding probability of the value labeled “1” can vary between 0 (certainly the value “0”) and 1 (certainly the value “1”), hence the labeling. The logistic regression model itself simply models the probability of output in terms of input and does not perform statistical classification (it is not a classifier), though it can be used to make a classifier, for instance by choosing a cutoff value and classifying inputs with probability greater than the cutoff as one class, below the cutoff as the other.

Logistic regression is a statistical model that estimates the probability of an event occurring based on a given dataset of independent variables. It can be used for classification and predictive analytics, and it produces a probability between 0 and 1.

Advantages of logistic regression include: Logistic Regression is easier to implement, and interpret, and very efficient to train. It makes no assumptions about distributions of classes in feature space. It can easily extend to multiple classes (multinomial regression) and a natural probabilistic view of class predictions. It can interpret model coefficients as indicators of feature importance.

Disadvantages of logistic regression include: Logistic Regression assumes a linear relationship between the dependent variable and the independent variables. It can only be used to predict discrete functions. It is less powerful than other algorithms for complex relationships. It requires average or no multicollinearity between independent variables.

Here is the evaluation result of Logistic Regression on test set with both non-adjusted threshold and adjusted-threshold:

Metric	LR	LR (with adjusted threshold 0.9 by using YoudenI method)
Accuracy	0.9368	0.947
Recall	0.4038	0.0962
Precision	0.3889	0.4286
F1-Score	0.3962	0.1571
AUC	0.8762	0.8762

Table 4.1: Performance Evaluation Metrics of Logistic Regression (LR)

Predicted values			Predicted values	
0	1		0	1

Actual values	0	2781	99
	1	93	63

Actual values	0	2860	20
	1	141	15

Table 4.2: Confusion Matrix of Logistic Regression(without threshold tuning) and Logistic Regression (with threhold tuning)

Both models have high accuracy, with the LR model with an adjusted threshold showing a slight improvement from 93.68% to 94.7%. However, accuracy alone may not provide a complete picture, especially in imbalanced datasets. The adjusted threshold significantly decreases the recall from 40% to 9.62%. This suggests that the LR model with the adjusted threshold is less effective at capturing positive instances. It may be allowing more false negatives. The adjusted threshold leads to a slight increase in precision from 38.89% to 42.86%. It indicates that with the adjusted threshold, the LR model is making fewer false positive predictions, but at the cost of a decrease in recall. The adjusted threshold results in a significant decrease in the F1-Score from 39.62% to 15.71%, reflecting a notable trade-off between precision and recall. The model with the adjusted threshold is less balanced in handling false positives and false negatives.

In summary, adjusting the threshold in the Logistic Regression model has a substantial impact on recall, precision, and the F1-Score. The trade-off between precision and recall should be carefully considered, and the choice of threshold depends on the specific goals and costs associated with false positives and false negatives in the given application.

6.2 Support Vector Machine (SVM)

Support Vector Machine (SVM) is a powerful machine learning algorithm that can be used for linear or nonlinear classification, regression, and even outlier detection tasks. SVMs can be used for a variety of tasks, such as text classification, image classification, spam detection, handwriting identification, gene expression analysis, face detection, and more.

SVM is a supervised learning algorithm that works by finding the best boundary (or hyperplane) that separates the data into different classes. In the case of classification, an SVM algorithm finds the best boundary that separates the data into different classes. The boundary is chosen in such a way that it maximizes the margin, which is the distance between the boundary and the closest data points from each class. These closest data points are called support vectors. SVMs can also be used for non-linear classification by using a technique called the kernel trick. The kernel trick maps the input data into a higher-dimensional space where the data becomes linearly separable. Common kernels include the radial basis function (RBF) and the polynomial kernel.

Advantages of SVM include: SVMs are effective in handling high-dimensional data, which is common in many applications such as image and text classification. SVMs can model non-linear decision boundaries, which can be very useful in many applications. SVMs can perform well with small datasets. SVMs can handle high-dimensional data, which is common in many applications such as image and text classification.

Disadvantages of SVM include: SVMs can be sensitive to the choice of kernel. SVMs can be computationally expensive when the dataset is large.

In binary classification, SVMs can be used to classify data into two classes. The goal is to find the best boundary that separates the data into two classes. The boundary is chosen in such a way that it maximizes the margin, which is the distance between the boundary and the closest data points from each class. These closest data points are called support vectors.

The RBF kernel is a popular choice for SVMs because it works well in practice and is relatively easy to calibrate . Here are some reasons to choose the RBF kernel: The RBF kernel can model complex decision boundaries that are not linearly separable. The RBF kernel can handle high-dimensional data, which is common in many applications such as image and text classification. The RBF kernel can handle non-linearly separable data by mapping the input data into a higher-dimensional space where the data becomes linearly separable. The RBF kernel is a good choice when the number of features is small compared to the number of instances.

The polynomial kernel is a kernel function commonly used with support vector machines (SVMs) and other kernelized models in machine learning. It represents the similarity of vectors (training samples) in a feature space over polynomials of the original variables, allowing learning of non-linear models. The polynomial kernel looks not only at the given features of input samples to determine their similarity, but also combinations of these. In the context of regression analysis, such combinations are known as interaction features. The degree of the polynomial kernel is a hyperparameter that determines the degree of the polynomial used in the kernel function

Metric	SVM (rbf kernel)	SVM (rbf) With threshold 0.0361	SVM (poly kernel)	SVM (poly) With threshold 0.1 found

		found best from roc of validation set		best from roc of validation set
Accuracy	0.9470	0.8070	0.9424	0.8877
Recall	0.4038	0.8910	0.4679	0.7500
Precision	0.4809	0.1963	0.4424	0.2792
F1-Score	0.4390	0.3218	0.4548	0.4070
AUC	0.9131	0.9131	0.9038	0. 9038

Table 5.1: Performance Evaluation Metrics of Support Vector Machine (SVM)

		Predicted values	
		0	1
Actual values	0	2812	68
	1	93	63

		Predicted values	
		0	1
Actual values	0	2788	92
	1	83	73

Table 5.2: Confusion Matrix of Suppport Vector Machine (RBF kernel) and Decision Tree (Poly kernel) – without threhold tuning

The choice between SVM with RBF kernel and SVM with poly kernel may depend on the specific requirements of the application. Adjusting the threshold impacts the trade-off between precision and recall, and the optimal threshold depends on the desired balance for a given application. AUC remains consistent, suggesting that discriminative ability is preserved despite threshold adjustments.

Overall, the SVM model with the adjusted threshold shows improvements in recall but at the cost of precision and the F1-Score. The trade-off between precision and recall should be carefully considered, and the choice of threshold depends on the specific goals and costs associated with false positives and false negatives in the given application. If a balanced trade-off between precision and recall is important, the SVM with rbf kernel or poly kernel without adjusted thresholds might be suitable. But SVM with rbf kernel performs a bit better accuracy and AUC compared with SVM with polynomial kernel. If capturing positive instances is crucial and the increased sensitivity is acceptable, consider the SVM models with adjusted thresholds.

6.3 Naive Bayes

It is a supervised learning algorithm that works by modeling the probability distribution of the features for each class and then using Bayes' theorem to calculate the posterior probability of each class given the input features. For this dataset, we will apply Bernoulli Naive Bayes and Gaussian Naive Bayes model: Bernoulli Naive Bayes is a variant of the Naive Bayes algorithm that is used for binary classification tasks . It is a supervised learning algorithm that works by modeling the probability distribution of the features for each class and then using Bayes' theorem to calculate the posterior probability of each class given the input features. Bernoulli Naive Bayes is commonly used for text classification tasks, such as spam detection and sentiment analysis. If the inputs are binary categorical, then Bernoulli Naive Bayes is a good choice. Our data, after one-hot encoding categorical variables, contains more than 30 binary categorical variables.

If having both categorical and continuous variables, then Gaussian Naive Bayes is also a good choice. Gaussian Naive Bayes is a variant of the Naive Bayes algorithm that is used for binary classification tasks.

Advantages of Naive Bayes include: Naive Bayes is a simple algorithm that is computationally efficient and can be trained quickly. Naive Bayes can handle high-dimnsional data, which is common in many applications such as text classification Naive Bayes can perform well with small. Naive Bayes can handle irrelevant features well, which can be useful in many applications

Disadvantages of Naive Bayes include: Naive Bayes assumes that the features are independent of each other, which may not be true in many real-world applications. But in our problem, we used VIF to get rid of variables that appeared to be dependent on each other (VIF >5). Bernoulli Naive Bayes is not suitable for continuous data, which requires a different variant of the Naive Bayes algorithm called Gaussian Naive Bayes. Gaussian Naive Bayes is not suitable for high-dimensional data, which requires a different variant of the Naive Bayes algorithm called Multinomial Naive Bayes.

Metric	BernoulliNB	BernoulliNB (with adjusted threshold 0.0496 by using YoudenI method)	GaussianNB	GaussianNB (with adjusted threshold 0.7 by using YoudenI method)
Accuracy	0.9153	0.7246	0.7625	0.7767
Recall	0.2949	0.7244	0.4615	0.4487
Precision	0.2383	0.1247	0.1016	0.1057
F1-Score	0.2636	0.2128	0.1665	0.1711
AUC	0.7994	0.7994	0.6508	0.6508

Table 6.1: Performance Evaluation Metrics of Naive Bayes (NB)

		Predicted values	
		0	1
Actual values	0	2733	147
	1	110	46

		Predicted values	
		0	1
Actual values	0	2243	637
	1	84	72

Table 6.2: Confusion Matrix of Bernoulli Naive Bayes (without tuning) and Gaussian Naive Bayes (without tuning).

The Bernoulli Naive Bayes model with the adjusted threshold using the Youden Index shows a significant improvement in recall but at the cost of precision and the F1-Score. This indicates that the model is better at capturing positive instances but makes more false positive predictions. The Gaussian Naive Bayes model, without threshold adjustment, performs better in terms of accuracy compared to Bernoulli Naive Bayes but still has limitations in capturing positive instances (low recall). Both Bernoulli Naive Bayes and Gaussian Naive Bayes models, when adjusted with the Youden Index, exhibit low precision and F1-Score, suggesting a trade-off between capturing positive instances and minimizing false positives. In summary, adjusting the threshold using the Youden Index in Naive Bayes models can lead to improvements in recall but may result in a trade-off with precision and the F1-Score.

Recommendations: The choice of the best model depends on the specific priorities of the application. Researchers and practitioners should carefully consider the implications of false positives and false negatives in their context. If capturing positive instances is critical, the adjusted threshold in Bernoulli Naive Bayes, optimized for high recall, is recommended. For applications prioritizing overall accuracy, the standard Bernoulli Naive Bayes model without threshold adjustment may be suitable. In scenarios where precision is a primary concern, the standard Bernoulli Naive Bayes model demonstrated superior performance. The medium AUC values for both Bernoulli Naive Bayes and Gaussian Naive Bayes leave a room for futher improve for the efficacy in discrimination tasks.

6.4 K-nearest neighbors

K-Nearest Neighbors (KNN) is a supervised learning algorithm that can be used for binary classification tasks. It works by finding the K closest data points in the feature space to a given test data point and using the majority class among them to classify the test data point. KNN is a non-parametric algorithm, meaning that it does not make any assumptions about the underlying distribution of the data.

Advantages of KNN include: KNN is a simple algorithm that is easy to understand, making it a popular choice for beginners in the field of machine learning. KNN is a non-parametric algorithm, meaning that it does not make any assumptions about the underlying distribution of the data. This makes it a flexible algorithm that can be used in a wide range of applications. KNN does not require any training process, which means that it can be used in real-time applications where data is continuously being generated. KNN can handle large datasets without suffering from the curse of dimensionality, which is a common problem in other machine learning algorithms.

Disadvantages of KNN include: KNN can be sensitive to outliers in the data, which can significantly affect its performance. Outliers are data points that are significantly different from the rest of the data, and they can have a disproportionate impact on the KNN algorithm’s classification results. KNN can be computationally expensive, particularly for large datasets. This is because the algorithm needs to compute the distance between each test data point and every training data point, which can be time-consuming. KNN requires a good choice of the K parameter, which determines the number of nearest neighbors used for classification. If K is too small, the algorithm may be too sensitive to noise in the data, while if K is too large, the algorithm may miss important patterns in the data.

In binary classification, KNN can be used to classify data into two classes. The goal is to find the K closest data points in the feature space to a given test data point and use the majority class among them to classify the test data point.

Metric	**KNN**	**KNN (with adjusted threshold 0.9125 by using YoudenI method)**
Accuracy	0.8399	0.9391
Recall	0.8675	0.253
Precision	0.2368	0.4078
F1-Score	0.3721	0.3123
AUC	0.9129	0.9129

Table 7.1: Performance Evaluation Metrics of K-nearest neighbors (KNN)

Predicted values			**Predicted values**	
0	1		0	1

Actual values	0	2406	464
	1	22	144

Actual values	0	2809	61
	1	124	42

Table 7.2: Confusion Matrix of K-nearest neighbors (without tuning) and K-nearest neighbors (with threshold tuning)

The KNN model with the adjusted threshold shows a substantial improvement in accuracy, indicating better overall performance. The adjusted threshold significantly decreases the recall. This suggests that the KNN model with the adjusted threshold is less effective at capturing positive instances, potentially resulting in more false negatives. The adjusted threshold leads to an increase in precision. This indicates that with the adjusted threshold, the KNN model is making fewer false positive predictions. The adjusted threshold results in a decrease in the F1-Score, indicating a trade-off between precision and recall. The model with the adjusted threshold is less balanced in handling false positives and false negatives.

The KNN model with the adjusted threshold demonstrates a notable improvement in accuracy, primarily driven by a reduction in false positives. In summary, the KNN model with the adjusted threshold is more conservative, reducing false positives but at the cost of lower recall and a less balanced F1-Score.

6.5 Decision Tree

Decision Tree is a supervised learning algorithm that can be used for binary classification tasks. It works by recursively partitioning the feature space into smaller regions, with each partition corresponding to a decision node in the tree . The goal is to find the best partitioning of the feature space that separates the data into different classes. When building a decision tree, the entropy criterion is used to measure the impurity of a node in the tree. The entropy of a node is defined as the sum of the negative logarithms of the probabilities of each class in the node, weighted by the proportion of instances in the node that belong to each class. The entropy criterion is used to determine which feature to split on at each node in the tree The Gini criterion is a measure of impurity used in decision trees to determine the quality of a split. It measures the probability of misclassifying a randomly chosen element if it were randomly labeled according to the distribution of classes in the subset. The Gini criterion is one of the most commonly used criteria for splitting nodes in decision trees. In the context of decision trees, the Gini criterion is used to evaluate the quality of a split by measuring the degree of impurity of the resulting subsets.

Advantages of Decision Tree include: Decision Trees are easy to understand and interpret, making them a popular choice for beginners in the field of machine learning. It can handle both categorical and continuous data, which is useful in many applications. Additionally, decision trees can model non-linear decision boundaries, which can be very useful in many applications. And it can handle missing values in the data, which is common in many real-world applications.

Disadvantages of Decision Tree include: Decision Trees can be prone to overfitting the training data, which can lead to poor generalization performance on new data. Decision Trees can be sensitive to small perturbations in the data, which can lead to different tree structures and classification results. Decision Trees can be biased towards features with many levels, which can lead to over-representation of these features in the tree. Also, Decision Trees can be computationally expensive when the dataset is large or the tree is deep.

In binary classification, Decision Trees can be used to classify data into two classes. The goal is to find the best partitioning of the feature space that separates the data into different classes.

Metric	Decision Tree (entropy criterion)	Decision Tree (gini criterion)
Accuracy	0.9361	0.9321
Recall	0.4103	0.4103
Precision	0.3855	0.3596
F1-Score	0.3975	0.3832
AUC	0.6874	0.6853

Table 8.1: Performance Evaluation Metrics of Decision Tree

		Predicted values	
		0	1
Actual values	0	2778	102
	1	92	64

		Predicted values	
		0	1
Actual values	0	2766	114
	1	92	64

Table 8.2: Confusion Matrix of Decision Tree (entropy criterion) and Decision Tree (gini criterion)

Both models achieve high accuracy, indicating a good overall classification performance on the dataset. However, it's important to consider other metrics, especially in imbalanced datasets. The recall, also known as sensitivity or true positive rate, is relatively low. This suggests that a substantial proportion of actual positive instances are being classified as negatives. Improving recall might be important, especially in scenarios where missing positive instances is costly. The F1-score balances precision and recall. The values suggest a moderate trade-off between precision and recall. A higher F1-score indicates a better balance between false positives and false negatives. AUC measures the area under the receiver operating characteristic curve. The values suggest a moderate discriminative ability of the models, but there is room for improvement. In summary, while both decision tree models exhibit high accuracy, there is room for improvement in terms of recall, precision, and the overall trade-off between these metrics (F1-score). Additionally, the AUC values indicate a moderate discriminative ability, suggesting that the models may benefit from further tuning or potentially trying different algorithms. Consideration of the specific requirements and costs associated with false positives and false negatives should guide the model evaluation and potential adjustments.

6.6 Random Forest

Random Forest is a popular machine learning algorithm that was first introduced by Leo Breiman and Adele Cutler in 2001. It is a powerful algorithm that can provide accurate predictions for both classification and regression problems. Random Forest is based on decision tree algorithms, which are simple, easy-to-understand models that work well for many problems, but they can also be unstable and prone to overfitting. Random Forest overcomes these limitations by using an ensemble of decision trees, where each tree is trained on a random subset of the data and a random subset of the features.

Random Forest is a robust algorithm that can handle noisy data and outliers. It is less likely to overfit the data, which means it can generalize well to new data. Random Forest is one of the most accurate machine learning algorithms and can handle both classification and regression problems. However, it has some limitations in terms of interpretability, overfitting, training time, and memory usage. In binary classification, Random Forest can be used to predict the class of a new sample based on the values of its features [1]. The algorithm works by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees.

In summary, Random Forest is a powerful algorithm that can provide accurate predictions for both classification and regression problems. It is robust, accurate, and can handle noisy data and outliers. However, it has some limitations in terms of interpretability, overfitting, training time, and memory usage. In binary classification, Random Forest can be used to predict the class of a new sample based on the values of its features

Metric	**Random Forest**	**Random Forest (with adjusted threshold 0.3533 by using YoudenI method)**
Accuracy	0.9509	0.9407
Recall	0.3654	0.5128
Precision	0.5327	0.4348
F1-Score	0.4335	0.4706
AUC	0.9369	0.9369

Table 9.1: Performance Evaluation Metrics of Random Forest

		Predicted values	
		0	1
Actual values	0	2778	102
	1	92	64

		Predicted values	
		0	1
Actual values	0	2766	114
	1	92	64

Table 9.2: Confusion Matrix of Random Forest (without threshold tuning) and Random Forest (with threshold tuning)

Both models exhibit high accuracy, with the Random Forest slightly outperforming the adjusted threshold version. However, accuracy alone may not provide a complete picture, especially in imbalanced datasets.

The adjusted threshold significantly improves the recall, indicating that the Random Forest model with the adjusted threshold is better at capturing positive instances. This is important, especially if false negatives are costly. The adjusted threshold leads to a decrease in precision. This suggests that more instances are predicted as positive, including some false positives. The trade-off between precision and recall is evident. The adjusted threshold improves the F1-Score, indicating a better balance between precision and recall. It reflects an overall improvement in the model's ability to correctly classify positive instances while controlling false positives.

In summary, adjusting the threshold in the Random Forest model has a noticeable impact on recall, precision, and the F1-Score. The trade-off between precision and recall is essential, and the choice of threshold depends on the specific requirements and costs associated with false positives and false negatives in the given application.

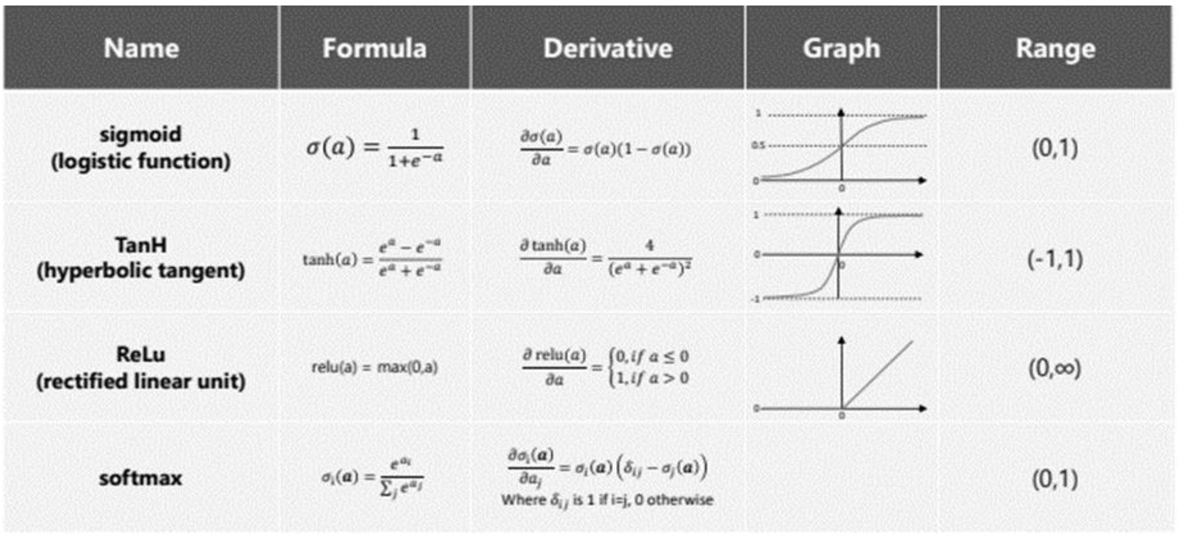
6.7 Artificial Neural Network (ANN)

Artificial Neural Networks (ANN) are a type of machine learning algorithm that are modeled after the structure of the human brain. They are used to recognize patterns in data and can be applied to a wide range of tasks, including binary classification. In binary classification, the output layer of an ANN has one node and a sigmoid activation function is used. The sigmoid function maps any input value to a value between 0 and 1, which is useful for binary classification because it can be interpreted as the probability of the input belonging to one of the two classes. While using softmax activation function, the output has two nodes, one for the probability that any input value maps into 0 and one for the probability that any input value maps into. And basically when we assess the result we only take the node assigned with the probability that any input value maps into.

Hyperparameters are parameters that are not learned by the ANN during training, but are set before training begins. They can have a significant impact on the performance of the ANN. Some of the hyperparameters that can be tuned include the number of neurons in each layer, the activation function used in each layer, the optimizer used to train the network, the learning rate, the batch size, and the number of epochs. In the context of neural networks, an epoch refers to one complete pass through the entire training dataset. During each epoch, the neural network is trained on all the training examples exactly once. An epoch is made up of one or more batches, where a batch is a subset of the training data that is used to update the weights of the neural network.

The number of epochs is a hyperparameter that can be tuned to improve the performance of the neural network. A larger number of epochs can lead to better performance on the training data, but it can also lead to overfitting, where the model becomes too specialized to the training data and performs poorly on new data. The batch size is another hyperparameter that can be tuned to improve the performance of the neural network. The batch size determines the number of samples used in each iteration of the training process. A larger batch size can lead to faster training times, but it requires more memory and may result in suboptimal performance.

The activation function is another important hyperparameter that can be tuned to improve the performance of the neural network. The choice of activation function in the hidden layer is critical to the performance of the ANN.



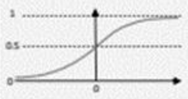
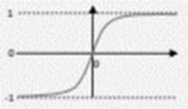
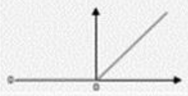
Name	Formula	Derivative	Graph	Range
sigmoid (logistic function)	$\sigma(a) = \frac{1}{1+e^{-a}}$	$\frac{\partial\sigma(a)}{\partial a} = \sigma(a)(1-\sigma(a))$		(0,1)
TanH (hyperbolic tangent)	$\tanh(a) = \frac{e^{a}-e^{-a}}{e^{a}+e^{-a}}$	$\frac{\partial \tanh(a)}{\partial a} = \frac{4}{(e^{a}+e^{-a})^2}$		(-1,1)
ReLu (rectified linear unit)	relu(a) = max(0,a)	$\frac{\partial\, \mathrm{relu}(a)}{\partial a} = \begin{cases} 0, if\ a \leq 0 \\ 1, if\ a > 0 \end{cases}$		(0,∞)
softmax	$\sigma_i(\boldsymbol{a}) = \frac{e^{a_i}}{\sum_j e^{a_j}}$	$\frac{\partial\sigma_i(\boldsymbol{a})}{\partial a_j} = \sigma_i(\boldsymbol{a})\left(\delta_{ij} - \sigma_j(\boldsymbol{a})\right)$ Where δ_{ij} is 1 if i=j, 0 otherwise		(0,1)

Table 10.1: Activation Functions used for ANN

The sigmoid activation function, commonly employed in artificial neural networks, is a non-linear activation function that transforms input values into a range between 0 and 1. The sigmoid function is particularly useful in binary classification tasks, where it models the probability of an input belonging to a certain class. One of its key advantages lies in its smooth gradient, enabling efficient optimization during the backpropagation process in training neural networks. However, the sigmoid function is susceptible to the vanishing gradient problem, which can hinder the training of deep networks.

The Rectified Linear Unit (ReLU) is a popular activation function used in deep learning neural networks. It is a piecewise linear function that outputs the input directly if it is positive, otherwise, it outputs zero. ReLU has become the default activation function for many types of neural networks because it is easier to train and often achieves better performance. ReLU overcomes the vanishing gradient problem, allowing models to learn faster and perform better. It is also computationally efficient, making it a good choice for large datasets. ReLU is one of the most widely used activation functions in deep learning.

The Softmax activation function is a mathematical function that takes a vector of real numbers as input and transforms it into a probability distribution. It assigns a probability to each element in the input vector, ensuring that the sum of these probabilities equals. Softmax is commonly used in machine learning models as an activation function to normalize the model's output into a probability distribution over a discrete set of classes. It is especially useful for classification tasks like sentiment analysis.

The hyperbolic tangent function, or Tanh, is a mathematical function that maps real numbers to the range of [-1, 1]. It is defined as the ratio of the hyperbolic sine and hyperbolic cosine functions. The Tanh function is commonly used as an activation function in artificial neural networks. It is similar to the sigmoid function, but its output is zero-centered, which makes it easier to train neural networks. The Tanh function is also used in statistics and signal processing.

Early stopping is a technique used to prevent overfitting in neural networks. It involves monitoring the performance of the model on a validation set during training and stopping the training process when the performance on the validation set stops improving. This can help prevent the model from memorizing the training data and improve its generalization performance. The primary advantages of ANNs include their ability to learn and generalize from data, their flexibility in addressing problems with non-linear shapes, and their capacity to approximate unknown functions. ANNs can also be used to solve problems for which linear regression methods fail.

However, ANNs have some downsides as well. One of the main disadvantages of ANNs is that they require a large amount of data to train effectively. Additionally, ANNs can be computationally expensive and require significant computational resources to train and run. Finally, ANNs can be difficult to interpret and explain, which can make it challenging to understand how they are making predictions or decisions.

Overall, ANNs are a powerful tool for machine learning and can be used to solve a wide range of problems. However, they are not without their limitations and require careful consideration when used in practice.

Metric	**ANN setting (1)**	**ANN setting (2)**	**ANN setting (3)**	**ANN setting (4)**	**ANN setting (5)**
Accuracy	0.945	0.9526	0.9117	0.9575	0.9229
Recall	0.4744	0.4615	0.8205	0.4167	0.7628
Precision	0.4654	0.5455	0.3478	0.6311	0.3766
F1-Score	0.4698	0.5	0.4885	0.5019	0.5042
AUC	0.926	0.9447	0.9447	0.9575	0.9462

Table 10.2: Performance Evaluation Metrics of Artificial Neural Network (ANN)

		Predicted values	
		0	1
Actual values	0	2795	85
	1	82	74

		Predicted values	
		0	1
Actual values	0	2820	60
	1	84	72

Table 10.3: Confusion Matrix of ANN setting (1) and ANN setting (2)

		Predicted values	
		0	1
Actual values	0	2640	240
	1	28	128

		Predicted values	
		0	1
Actual values	0	2842	38
	1	91	65

Table 10.4: Confusion Matrix of ANN setting (3) and ANN setting (4)

ANN setting hyperparameters (1) is as followed: The model has 4 layers, 2 hidden layers, 1 input layer and 1 output layer, 100 epoches, the batch size of 150. Each hidden layers consists 150 nodes and uses Relu activation function. The output layer uses sigmoid activation function. Early stopping after 7 times which means monitoring the performance of the model on the validation set and stopping the training process when performance on the validation set stops improving for 7 times.

ANN setting hyperparameters (2) is as followed: The model has 5 layers, 3 hidden layers, 1 input layer and 1 output layer, 100 epoches, the batch size of 100. Each hidden layers consists 200 nodes and uses Tanh activation function. The output layer uses softmax activation function. Early stopping after 13 times which means monitoring the performance of the model on the validation set and stopping the training process when performance on the validation set stops improving for 13 times.

ANN setting hyperparameters (3) is the ANN model in setting (2) but the threshold was tuned by ROC and Youden Index from the validation set: The model has 5 layers, 3 hidden layers, 1 input layer and 1 output layer, 100 epoches, the batch size of 100. Each hidden layers consists 200 nodes and uses Tanh activation function. The output layer uses softmax activation function. Early stopping after 13 times which means monitoring the performance of the model on the validation set and stopping the training process when performance on the validation set stops improving for 13 times. Threshold was tuned so it is 0.02.

ANN setting hyperparameters (4) is as followed: The model has 5 layers, 3 hidden layers, 1 input layer and 1 output layer, 100 epoches, the batch size of 100. Each hidden layers consists 225 nodes and uses Tanh activation function. The output layer uses softmax activation function. Early stopping after 13 times which means monitoring the performance of the model on the validation set and stopping the training process when performance on the validation set stops improving for 13 times.

ANN setting hyperparameters (5) is the ANN model in setting (4) but the threshold was tuned by ROC and Youden Index from the validation set: The model has 5 layers, 3 hidden layers, 1 input layer and 1 output layer, 100 epoches, the batch size of 100. Each hidden layers consists 225 nodes and uses Tanh activation function. The output layer uses softmax activation function. Early stopping after 13 times which means monitoring the performance of the model on the validation set and stopping the training process when performance on the validation set stops improving for 13 times. Threshold was tuned so it is 0.02.

Some comments on the performance metric table:

Setting (4) stands out with the highest accuracy, precision, and AUC. It achieves a good balance between sensitivity (recall) and precision.

Setting (3) has a high recall (82.05%), but at the cost of lower precision. This setting might be suitable for applications where capturing positive instances is a priority.

Setting (2) has a high accuracy and precision but a lower recall. This might be a good choice when minimizing false positives is crucial.

Setting (5) shows a balanced performance with decent accuracy, recall, and precision. Setting (1) and Setting (2) also perform well, with a balance between precision and recall.

In summary, each setting of the ANN demonstrates varying trade-offs between accuracy, recall, precision, and AUC. The optimal setting depends on the specific requirements and priorities of your application.

6.8 Summarization and model comparison:

Metric	Accuracy	Recall	Precision	F1-Score	AUC
ANN setting (4)	0.9575	0.4167	0.6311	0.5019	0.9462
ANN setting (2)	0.9526	0.4615	0.5455	0.5	0.9447
Random Forest	0.9509	0.3654	0.5327	0.4335	0.9369
SVM (rbf)	0.947	0.4038	0.4809	0.439	0.9131
LR (with adjusted threshold 0.9 by using validation set)	0.947	0.0962	0.4286	0.1571	0.8762
ANN setting (1)	0.945	0.4744	0.4654	0.4698	0.926
SVM (poly kernel)	0.9424	0.4679	0.4424	0.4548	0.9038
Random Forest (with adjusted threshold 0.3533 by using validation set)	0.9407	0.5128	0.4348	0.4706	0.9369
KNN (with adjusted threshold 0.9125 by using validation set)	0.9391	0.253	0.4078	0.3123	0.9129

LR	0.9368	0.4038	0.3889	0.3962	0.8762
Decision Tree (entropy criterion)	0.9361	0.4103	0.3855	0.3975	0.6874
Decision Tree (gini criterion)	0.9321	0.4103	0.3596	0.3832	0.6853
ANN setting (5)	0.9229	0.7628	0.3766	0.5042	0.9462
BernoulliNB	0.9153	0.2949	0.2383	0.2636	0.7994
ANN setting (3)	**0.9117**	**0.8205**	**0.3478**	**0.4885**	**0.9447**
SVM (poly) With threshold 0.0361 found best from roc of validation set	0.8877	0.75	0.2792	0.407	0. 9038
KNN	**0.8399**	**0.8675**	**0.2368**	**0.3721**	**0.9129**
SVM (rbf) With threshold 0.0361 found best from roc of validation set	0.807	0.891	0.1963	0.3218	0.9131
GaussianNB (with adjusted threshold by using validation set)	0.7767	0.4487	0.1057	0.1711	0.6508
GaussianNB	0.7625	0.4615	0.1016	0.1665	0.6508
BernoulliNB (with adjusted threshold 0.0496 by using validation set)	0.7246	0.7244	0.1247	0.2128	0.7994

Table 11: Performance Evaluation Metrics of All Machine Learning Models used

In the pursuit of identifying the optimal model for a binary classification task, a comprehensive research analysis was conducted, encompassing a diverse set of machine learning algorithms. The table includes several models, such as Artificial Neural Networks (ANN) with different settings, Random Forest, Support Vector Machines (SVM) with different kernels, Logistic Regression (LR), Decision Tree, K-Nearest Neighbors (KNN), Bernoulli Naive Bayes, and Gaussian Naive Bayes. Accuracy is a measure of overall correctness. The models achieve accuracies ranging from 72.46% to 95.75%.

It's important to consider the specific requirements of the application but this application did not state that we should prioritize on false negative or false positive or the accuracy or merely minimize both. Thus, the assessment should be able to assess the best models in each scenario. For that reason, the research will analyze based on each of the metrics in the table.

Highest Accuracy and Overall Performance:

Some models, like ANN setting (4), ANN setting (2), and Random Forest, show relatively high accuracy at 95.75%, 95.26%, 95.09% and AUC at 94.62%, 94.47%, 93.69%. Support Vector Machine with RBF kernel and Logistic Regression with adjusted threshold also exhibit a balanced performance, but with lower accuracy than the top 3 models. However, Logisitic Regression model demonstrates a much lower recall which is just 9.6%

which indicates the trade-off between accuracy and recall in this model is higher than the other top models.

Consideration for Positive Instances (Recall):

The model SVM (RBF kernel) with adjusted threshold and KNN gave quite high recall of 89.1% and 86.7%. However, considering the tradeoff between recall and precision and the trade-off between recall and accuracy, the model ANN with setting (5) and (3) appear to be superior.

Balance between Recall and Precision: Based on the table, if prioritizing false negatives, then the best model would be ANN setting (3) with a recall of 82.05%. Recall is the fraction of true positives over the sum of true positives and false negatives. A high recall value indicates that the model is able to correctly identify most of the positive cases, which is what you want to prioritize if you want to minimize false negatives. It is important to note that this model has a precision of 34.78%. Therefore, if you also want to minimize false positives, you may want to consider other models with higher precision values such as ANN setting (5) with a precision of 37.66%.

If false positive is prioritized, precision is the utmost import then the best model is ANN setting (4), (2), random forest and SVM (RBF kernel) with 59.14%, 54.55%, 53.27%, and 48.09% respectively. However considering the balance between precision and recall, ANN setting (2) and SVM (RBF kernel) win.

Prioritizing a balanced performance across multiple metrics

If a balanced performance across multiple metrics is desired, Random Forest (without threshold adjustment) or ANN Setting (2), (5) could be strong contenders since they got F1-score of 0.5 and 0.47. The F1 score is the harmonic mean of the precision and recall. It thus symmetrically represents both precision and recall in one metric.

Prioritizing the overall discriminatory power

The top four models, prioritized for their balanced performance and discriminatory power, are as follows:

ANN setting (4): This model exhibits a notable AUC of 0.9462, signifying its superior ability to distinguish between positive and negative instances. Additionally, its high accuracy (95.75%), recall (35.26%), precision (59.14%), and F1-score (44.18%) contribute to its standing as a top-performing candidate.

ANN setting (2): Boasting an AUC of 0.9447, this ANN setting (2) model showcases high accuracy (95.26%), recall (46.15%), precision (54.55%), and F1-score (50.0%). Its performance across multiple metrics positions it as a strong contender, reflecting its adaptability to various evaluation criteria.

Random Forest (with adjusted threshold 0.3533 by using validation set): With an AUC of 0.9369, this Random Forest model demonstrates robust discriminatory capabilities. It achieves a commendable balance between accuracy (94.07%), recall (51.28%), and F1-score (47.06%). The utilization of an adjusted threshold from the validation set underscores its commitment to fine-tuning for optimal operational characteristics.

Recommendations:

Overall, artificial neural network is an extremely powerful machine learning model, if the objective is to find a model that beats every other models across multiple metrics then ANN setting (4), (2), Random Forest, and SVM with RBF kernel have the highest overall performance. And in case of preferring safety for true positive, especially with normal advertising campaign, we would like to maximize the revenue from term deposits, then we want to minimize false negative. Consequently, we should select the model with a high recall and high discriminatory power which is ANN setting (5) or (3). Otherwise, we should select the neural network model setting (4) or (2).

7. References

Alfandash (2020). *RPubs - Bank Telemarketing: Classification Supervised Machine Learning*. [online] rpubs.com. Available at: https://rpubs.com/alfandash/lbb-classification-2.

Analytics Vidhya. (2021). *Softmax | What is Softmax Activation Function | Introduction to Softmax*. [online] Available at:

https://www.analyticsvidhya.com/blog/2021/04/introduction-to-softmax-for-neural-network/.

Aqeel, M. (2022). *Advantages & Disadvantages of KNN in Machine Learning*. [online] Universel news. Available at: https://www.universelnews.com/2022/05/06/advantages-disadvantages-of-knn-in-machine-learning/.

baeldung (2020a). *Advantages and Disadvantages of Neural Networks | Baeldung on Computer Science*. [online] www.baeldung.com. Available at: https://www.baeldung.com/cs/neural-net-advantages-disadvantages.

baeldung (2020b). *Epoch in Neural Networks | Baeldung on Computer Science*. [online] www.baeldung.com. Available at: https://www.baeldung.com/cs/epoch-neural-networks.

Berezovsky, O. (2020). *Supervised machine learning - Binary logistic regression overview*. [online] Logic2020.com. Available at: https://logic2020.com/insight/decision-tree-classifier-overview/ [Accessed 13 Jan. 2024].

Bouckaert, R.R. (2004). Naive Bayes Classifiers That Perform Well with Continuous Variables. *Springer eBooks*, pp.1089–1094. doi:https://doi.org/10.1007/978-3-540-30549-1_106.

Brownlee, J. (2019). *A Gentle Introduction to the Rectified Linear Unit (ReLU) for Deep Learning Neural Networks*. [online] Machine Learning Mastery. Available at: https://machinelearningmastery.com/rectified-linear-activation-function-for-deep-learning-neural-networks/.

Brownlee, J. (2021). *How to Choose an Activation Function for Deep Learning*. [online] Machine Learning Mastery. Available at: https://machinelearningmastery.com/choose-an-activation-function-for-deep-learning/.

Desai, R. and Khairnar, V.D. (2021). Hybrid Prediction Model for the Success of Bank Telemarketing. *Lecture notes in networks and systems*, pp.693–710. doi:https://doi.org/10.1007/978-981-16-2422-3_54.

Editorial (2022). *Pros and cons of Support Vector Machine (SVM)*. [online] RoboticsBiz. Available at: https://roboticsbiz.com/pros-and-cons-of-support-vector-machine-svm/.

Fleiss, A. (2023). *What are the advantages and disadvantages of random forest?* [online] Rebellion Research. Available at: https://www.rebellionresearch.com/what-are-the-advantages-and-disadvantages-of-random-forest.

GeeksforGeeks. (2020a). *Major Kernel Functions in Support Vector Machine (SVM)*. [online] Available at: https://www.geeksforgeeks.org/major-kernel-functions-in-support-vector-machine-svm/.

GeeksforGeeks. (2020b). *Support vector machine in Machine Learning*. [online] Available at: https://www.geeksforgeeks.org/support-vector-machine-in-machine-learning/.

Krishnamurthy, B. (2022). *ReLU Activation Function Explained | Built In*. [online] builtin.com. Available at: https://builtin.com/machine-learning/relu-activation-function.

Mayo, J. (2018). *AI Building Blocks: How TanH Works*. [online] Medium. Available at: https://joemayo.medium.com/ai-building-blocks-how-tanh-works-51dc5a4e0ce0 [Accessed 13 Jan. 2024].

Moro, S., Cortez, P. and Rita, P. (2014). A data-driven approach to predict the success of bank telemarketing. *Decision Support Systems*, 62, pp.22–31.

Mustapha, A. (2017). Customer Profiling using Classification Approach for Bank Telemarketing. *JOIV : International Journal on Informatics Visualization*, [online] 1. Available at: https://www.academia.edu/71283241/Customer_Profiling_using_Classification_Approach_for_Bank_Telemarketing [Accessed 13 Jan. 2024].

OpenGenus IQ: Computing Expertise & Legacy. (2023). *9 Advantages and 10 disadvantages of Naive Bayes Algorithm*. [online] Available at: https://iq.opengenus.org/advantages-and-disadvantages-of-naive-bayes-algorithm/.

R, S. (2021). *Tuning Hyperparameters of An Artificial Neural Network Leveraging Keras Tuner*. [online] Analytics Vidhya. Available at:

https://www.analyticsvidhya.com/blog/2021/06/tuning-hyperparameters-of-an-artificial-neural-network-leveraging-keras-tuner/.

Rout, A.R. (2020). *Advantages and Disadvantages of Logistic Regression*. [online] GeeksforGeeks. Available at: https://www.geeksforgeeks.org/advantages-and-disadvantages-of-logistic-regression/.

scikit learn (2018). *1.4. Support Vector Machines — scikit-learn 0.20.3 documentation*. [online] Scikit-learn.org. Available at: https://scikit-learn.org/stable/modules/svm.html.

scikit learn (2019). *sklearn.tree.DecisionTreeClassifier — scikit-learn 0.22.1 documentation*. [online] Scikit-learn.org. Available at: https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html.

Smitha Shekar B (2021). This work is licensed under a Creative Commons Attribution 4.0 International License Bank Marketing Data Classification Using Machine Learning. *International Advanced Research Journal in Science, Engineering and Technology*, [online] 8(2). doi:https://doi.org/10.17148/IARJSET.2021.8226.

Stack Overflow. (n.d.). *How to choose the right kernel functions*. [online] Available at: https://stackoverflow.com/questions/16964975/how-to-choose-the-right-kernel-functions [Accessed 13 Jan. 2024].

Team, T.A. (2020). *Why Choose Random Forest and Not Decision Trees – Towards AI — The Best of Tech, Science, and Engineering*. [online] Available at: https://towardsai.net/p/machine-learning/why-choose-random-forest-and-not-decision-trees.

Tékouabou, S.C.K., Gherghina, Ş.C., Toulni, H., Neves Mata, P., Mata, M.N. and Martins, J.M. (2022). A Machine Learning Framework towards Bank Telemarketing Prediction. *Journal of Risk and Financial Management*, 15(5), p.269. doi:https://doi.org/10.3390/jrfm15060269.

TowardsAnalytic (2023). *Understanding Softmax Activation Function in Neural Networks*. [online] TowardsAnalytics. Available at: https://www.towardsanalytic.com/softmax-activation-function/ [Accessed 13 Jan. 2024].

www.oreilly.com. (n.d.). *Advantages and disadvantages of decision trees - Mastering Machine Learning with scikit-learn - Second Edition [Book]*. [online] Available at: https://www.oreilly.com/library/view/mastering-machine-learning/9781788299879/195817d2-bd70-484b-8642-c40f98e85591.xhtml [Accessed 13 Jan. 2024].

www.programiz.com. (n.d.). *NumPy tanh() (With Examples)*. [online] Available at: https://www.programiz.com/python-programming/numpy/methods/tanh [Accessed 13 Jan. 2024].