Mini Project 1

Abo Akademi University
Department of Informatics Technology

Machine Learning
Year 2023

Student Lamin Jatta

Contents

0.1 Introduction

In this assignment, we will be working on a real-world problem of direct marketing campaigns conducted by a banking institution. The campaigns were carried out through phone calls, and our task is to predict whether or not clients would subscribe to the bank's term deposit product. The provided dataset includes multiple contacts with the same client, which will be used to train a machine-learning model that can make these predictions.
Objectives:

- Understand the provided data, perform any necessary cleaning or preprocessing, and familiarize ourselves with the problem domain.
- Develop a machine learning model that can accurately predict whether or not a client will subscribe to the bank's term deposit product.
- Evaluate the performance of the model and suggest ways to improve its accuracy.

Methodology:

- Data Exploration: The first step is to understand the data provided. This includes checking for missing values, outliers, and any other issues that need to be addressed before building the model.
- Data Preprocessing: The next step is to perform any necessary cleaning or preprocessing on the data. This may include filling in missing values, removing outliers, and encoding categorical variables.
- Model Building: After the data has been cleaned and preprocessed, we will build a machine-learning model to make predictions about whether or not clients will subscribe to the bank's term deposit product. Different algorithms will be considered, and the best one will be selected based on its performance.
- Model Evaluation: The final step is to evaluate the performance of the model using metrics such as accuracy, precision, recall, and F1 score. Based on the evaluation results, suggestions for improving the model's accuracy will be provided.

0.2 Data Exploration

0.2.1 Descriptive statistics

Obtaining basic information about the dataset such as the number of rows, columns, and data types of each feature.
In this datasets we have 21 columns and 41188 rows of data in the recorded fields below:

- **age**: Age of person in numerical value
- **Job**: Type of job in categorical level

- **Marital**: Marital status in categorical level
- **Education**: Education levels in categorical level
- **Default**: has credit in default? in categorical level
- **Housing**: has housing loan? in categorical level
- **Loan**: has personal loan? in categorical level
- **contact**: Contact communication type in categorical level
- **Month**: Last contact month of the year
- **Day of Week**: Last contact day of the week
- **Duration**: Last contact duration, in seconds
- **campaign**: Number of contacts performed during this campaign and for this client
- **P_Day**: Number of days that passed by after the client was last contacted from a previous campaign
- **previous** Number of contacts performed before this campaign and for this client
- **poutcome**: Outcome of the previous marketing campaign
- **emp.var.rate**: Employment variation rate
- **cons.price.idx**: Consumer price index
- **cons.conf.idx**: Consumer confidence index
- **euribor3m**: Euribor 3 month rate
- **nr.employed**: number of employees
- **y**: has the client subscribed to a term deposit?

From the statistical distribution of the numerical data from the dataset, we can see the distribution of the age; as the mean age of the contact person is 40 years old the minimum age of the contact person is 17 years and the oldest is 98 years old.

	age	duration	campaign	pdays	previous	emp.var.rate	cons.price.idx	cons.conf.idx	euribor3m	nr.employed
count	41188.00000	41188.000000	41188.000000	41188.000000	41188.000000	41188.000000	41188.000000	41188.000000	41188.000000	41188.000000
mean	40.02406	258.285010	2.567593	962.475454	0.172963	0.081886	93.575664	-40.502600	3.621291	5167.035911
std	10.42125	259.279249	2.770014	186.910907	0.494901	1.570960	0.578840	4.628198	1.734447	72.251528
min	17.00000	0.000000	1.000000	0.000000	0.000000	-3.400000	92.201000	-50.800000	0.634000	4963.600000
25%	32.00000	102.000000	1.000000	999.000000	0.000000	-1.800000	93.075000	-42.700000	1.344000	5099.100000
50%	38.00000	180.000000	2.000000	999.000000	0.000000	1.100000	93.749000	-41.800000	4.857000	5191.000000
75%	47.00000	319.000000	3.000000	999.000000	0.000000	1.400000	93.994000	-36.400000	4.961000	5228.100000
max	98.00000	4918.000000	56.000000	999.000000	7.000000	1.400000	94.767000	-26.900000	5.045000	5228.100000

Figure 1: Statistical distribution of numerical data.

Some of the data are categorical variables this will be normalised using the **Label Encoder** in the sklearn library. The variable is:

- Jobs: ['housemaid' 'services' 'admin.' 'blue-collar' 'technician' 'retired' 'management' 'unemployed' 'self-employed' 'unknown' 'entrepreneur' 'student']

- Marital: ['married' 'single' 'divorced' 'unknown']

- Education: ['basic.4y', 'high school', 'basic.6y', 'basic.9y', 'professional course' 'unknown' 'university degree' 'illiterate']

In addition to this are some other variables that are also categorical and as well will be encoded to the numerical variables for preprocessing including the output or target variable **y**.

0.2.2 Missing value analysis

Checking for missing values in the dataset and deciding how to handle them. This specific dataset is missing some values, but instead of being marked as "NaN" (Not a Number), they are marked as "unknown". This means that the data is missing, but the missing values have been filled with a special string value ("unknown") instead of the typical NaN value. This is a common practice to handle missing values in datasets. In this case, the "unknown" value is used to indicate that there is no information about the particular data point.
It's important to be aware of this and take it into account when performing data analysis since missing values can have a significant impact on the results and conclusions drawn from the data. For example, in this case of the dataset, you should be careful when calculating the statistics of the 'unknown' values and when building the model, it is important to impute the 'unknown' values with appropriate values or drop the unknown rows.
Keeping the "unknown" values in the dataset can affect the performance of the machine-learning models in several ways:

- **Data Quality**: The presence of "unknown" values can indicate a lack of data quality. These values can skew the statistics and make it difficult to draw meaningful conclusions from the data.

- **Model performance**: Models may not be able to handle "unknown" values, and they may produce unexpected or inaccurate results if they are included in the training dataset.

- **Data Imbalance:** The presence of "unknown" values can cause data imbalance in the dataset, which can cause the model to perform poorly.

To mitigate these effects, it is a good practice to handle "unknown" values by either replacing them with appropriate values or by removing the rows that contain them. The appropriate approach will depend on the specific dataset and the problem you are trying to solve.
It's also important to note that just removing the rows containing "unknown" values might lead to the loss of important information and might affect the model performance. Therefore you have to be careful in choosing the appropriate approach based on the data and the problem.

0.2.3 Univariate, Bivariate analysis and Data visualization

Analyze each feature individually to understand the distribution of the data, identify any outliers, and check for any inconsistencies or errors.
Examining the relationship between each feature and the target variable, in order to identify which features may be important for the problem.
Using graphical methods such as histograms, bar plots, and scatter plots to better understand the data and identify patterns or trends.

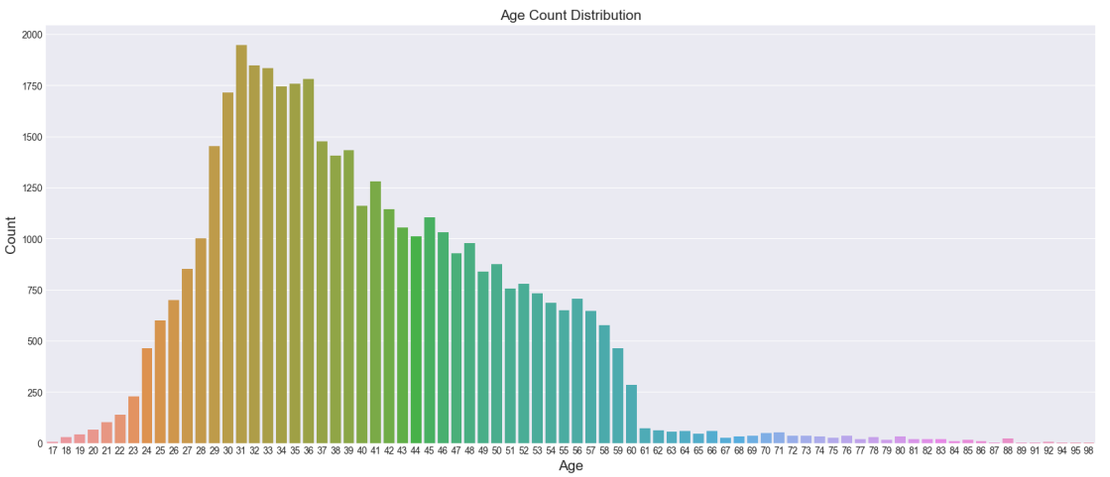

Figure 2: Age distribution of the data.

Figure 3 shows that on average, the customers who have subscribed to the term deposit are older than the customers who have not subscribed. This means that the average age of customers who bought the term deposit is greater than the average age of customers who did not.

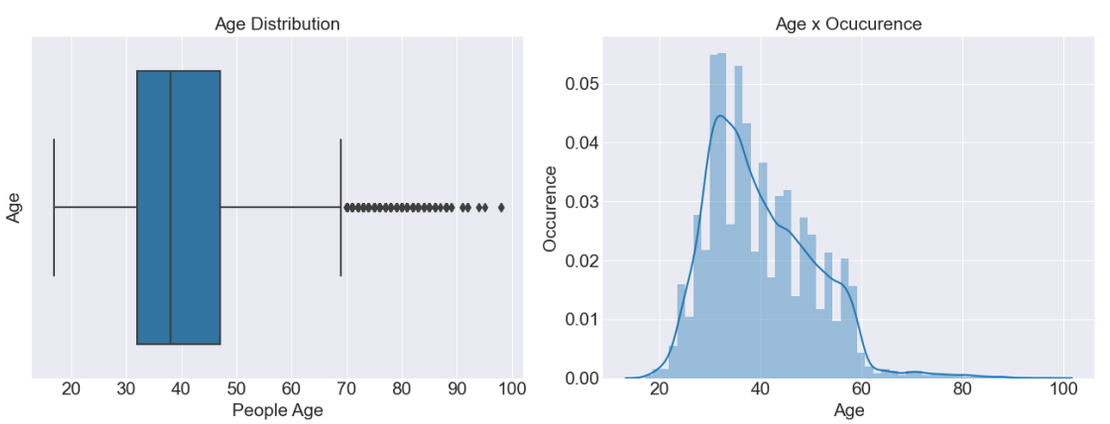

Figure 3: Age distribution of the data.

the data provided shows that:

- The minimum age is 32 years old.
- The median age is 38 years old.
- The 75th percentile of the ages is 47 years old.

- The maximum age is 98 years old.

The value of 69.5 as an "outlier" is not a statistical value, it's just a value that separates values above it from the rest of the data. Outliers are defined as values that are significantly different from the rest of the data. In this case, any age above 69.5 is considered an outlier. If we use that as a threshold then we from the age for the

- Numerber of outliers: 469
- Number of clients: 41188
- Outliers are: 1.14 %

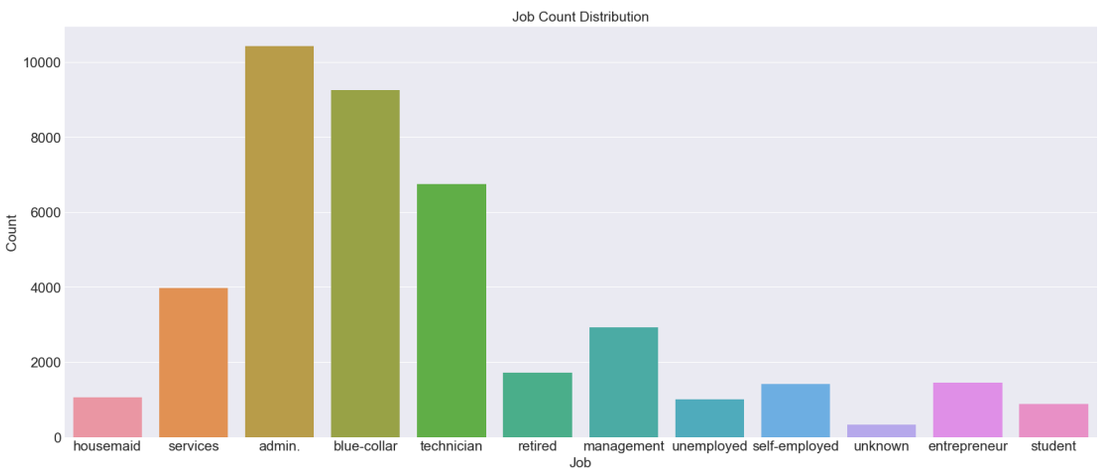

Figure 4: Job distribution of the data.

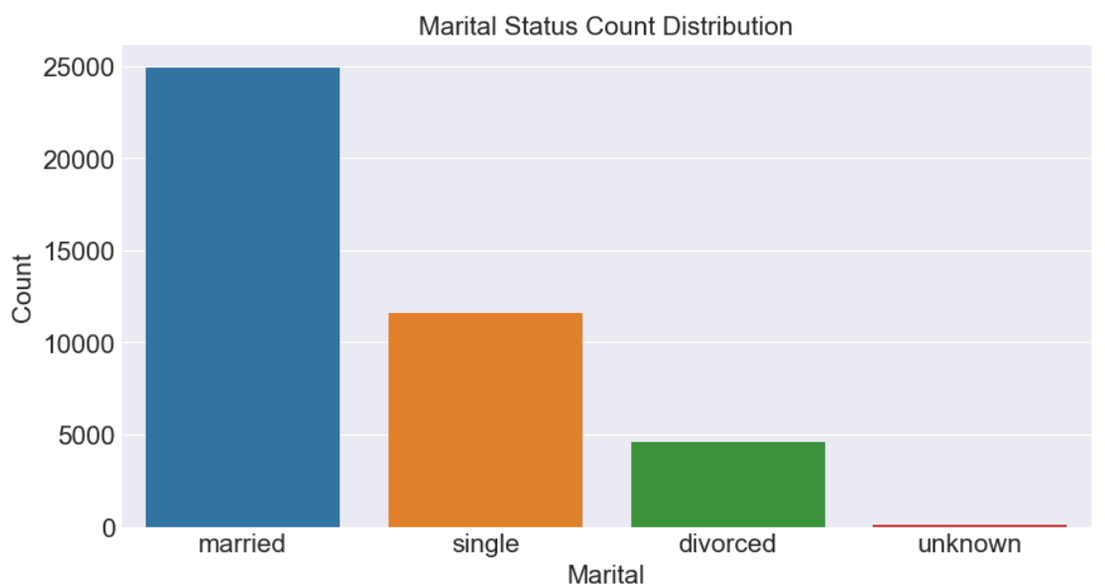

Figure 5: Marital status distribution of the data.

Figure 5 describes the marital status of a group of people, and it is presented as a categorical variable with four possible values: "divorced", "married", "single", and

"unknown". The two most common categories in this dataset are "married" and "single", which suggests that most of the people in the dataset fall into one of those two categories. The categories "divorced" and "unknown" are less common, and may represent a smaller percentage of the total group. The note in the data also explains that "divorced" includes individuals who are both divorced or widowed.

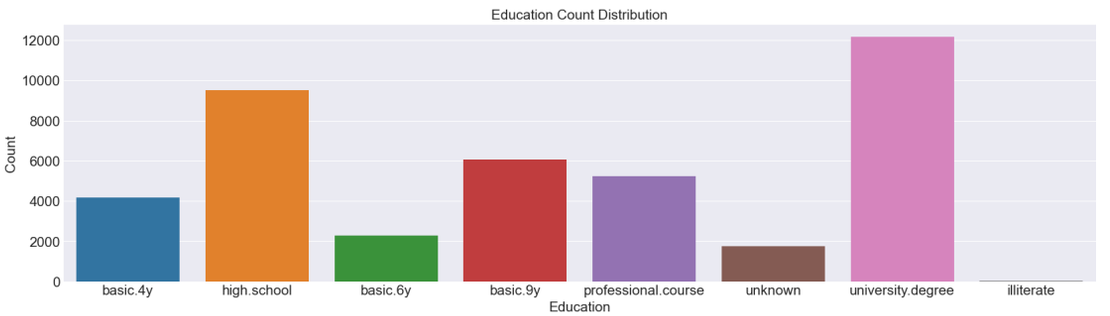

Figure 6: Educational distribution of the data.

The data provided is a categorical variable representing different types of jobs. In Figure 6 the data is describing the distribution of different types of jobs within a certain population, such as a group of individuals who have been targeted for a marketing campaign. The biggest three groups are represented by administrative jobs, blue-collar jobs, and technicians, which make up approximately more than 50% of the whole group. The smallest three groups are represented by the unemployed, students, and an unknown group. The marketing team assumed that these groups might not have savings to deposit. The campaign was not targeting self-employed entrepreneurs. Around 30% of the dataset have higher education. Around 23% of the dataset has only a high school diploma. The rest of the dataset has only 4 to 9 years of basic education or professional courses.
This could mean that the department was targeting mainly individuals and not legal entities; also, it can mean that deposit offers would be of interest only to individuals and not self-employed or entrepreneurs. Interestingly, groups "management," "retired," and "services" were not actively approached despite the fact that these groups might have savings to invest in deposits.

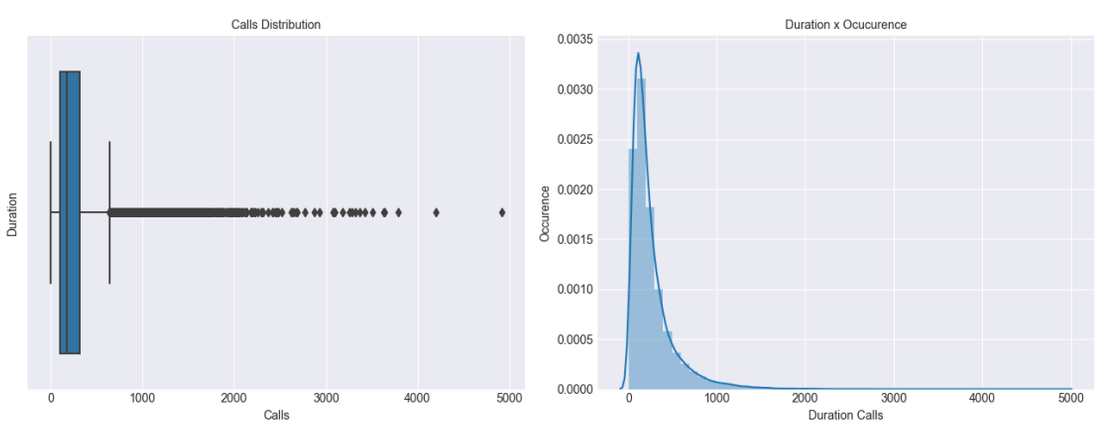

Figure 7: Call duration distribution of the data.

In Figure 7, This statistical result describes the distribution of call durations in

minutes. The maximum duration of a call is 82 minutes, the minimum duration is 0 minutes, and the average (mean) duration is 4.3 minutes. The standard deviation (STD) of the duration of calls is also 4.3 minutes.
The maximum and minimum values give an idea of the range of the data, while the mean and standard deviation provide information about the central tendency and spread of the data, respectively. A high standard deviation indicates that the data is more spread out, while a low standard deviation indicates that the data is more concentrated around the mean. Note that if the call duration is equal to 0, then is obvious that this person didn't subscribe.

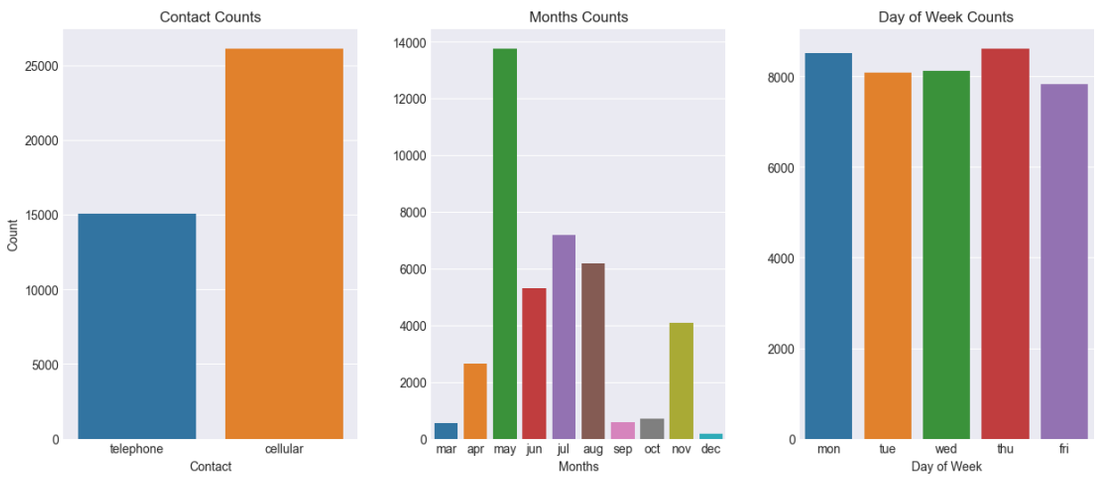

Figure 8: Contact, month, and Day of week distribution of the data.

0.3 Modelling

Before model selection and evaluation, First, we import the train_test_split function from the sklearn.model_selection module. This function is used to split the dataset into training and testing sets.
In this case, the target variable 'y' is separated from the rest of the data and assigned to the variable 'y'. The rest of the data is being assigned to the variable 'bank_final'. The train_test_split function is then used to divide this data into training and testing sets, with 20% of the data being used for testing and 80% for training. Then we use the random_state parameter to be set to 101, which is used to ensure the reproducibility of the results.
A StandardScaler which is a preprocessing technique used in machine learning and statistics to standardize a dataset. It is used to transform a dataset so that it has a mean of 0 and a standard deviation of 1. This is done by subtracting the mean from each data point and then dividing it by the standard deviation.
This is useful for many machine learning algorithms because many algorithms assume that the data is normally distributed and that the scale of the features is similar. By standardizing the data, the algorithm can converge faster and potentially produce better results. Additionally, standardizing the data helps to prevent one feature from dominating the others, which can be especially important in algorithms like linear regression.
confusion_matrix, accuracy_score are used to evaluate the performance of the model. The confusion matrix is a table that is used to define the performance

of a classification algorithm, where predicted values are compared with the actual values. The accuracy score is used to evaluate the overall performance of a classification algorithm, it is defined as the ratio of correct predictions to total predictions.

0.3.1 Logistic Regression

Logistic regression is a widely used statistical method that is particularly useful for analyzing data with binary outcomes (i.e. outcomes with two possible values, such as success or failure, yes or no, etc.). It is a type of generalized linear model that uses a logistic function to model the probability of a certain binary outcome. In this case, the data being analyzed has a binary outcome (e.g. whether a customer will sign up for a deposit or not) and logistic regression is a suitable method for modelling this outcome. The model is fit on the training data, then the accuracy is computed on test data and also cross-validation is done to check the robustness of the model
Additionally, logistic regression is relatively easy to implement and interpret, making it a popular choice for many types of data analysis. It's also good for small datasets and when there are not too many features.
Then the fit method is called on the log model with training data (X_train) and the corresponding target values (y_train) to train the model.

0.3.2 Procedure

First, the LogisticRegression() function is imported from sklearn.linear_model library, and then an instance of the logistic regression model is created and assigned to the variable 'logmodel'.
Then the fit method is called on logmodel with training data (X_train) and the corresponding target values (y_train) to train the model.
The log-pred variable stores the predictions made by the logistic regression model on the test data (X_test).
Then, the confusion_matrix function is used to create a confusion matrix and the accuracy_score function is used to calculate the accuracy of the model. The accuracy score is rounded and multiplied by 100 to get the percentage.
Finally, the cross_val_score function is used to perform k-fold cross-validation on the trained model using k_fold defined previously. It returns an array of accuracy scores for each fold, and the mean of these scores is assigned to the variable LOGCV.

	T	F
T	7106	173
F	618	341

The confusion matrix defines the performance of a classification algorithm. It is mainly used in supervised learning, and it allows us to evaluate the accuracy of a model by comparing the predicted output and the actual output. Each row of the matrix represents the instances in a predicted class while each column represents the instances in an actual class (or vice versa).
In this specific confusion matrix:

- 7106: Number of correct predictions of the class "not term deposit"

- 173: Number of incorrect predictions of the class "term deposit" as "not term deposit"

- 618: Number of incorrect predictions of the class "not term deposit" as "term deposit"

- 341: Number of correct predictions of the class "term deposit"

The overall accuracy of the logistic regression model is calculated by dividing the number of correct predictions by the total number of predictions. In this case, $(7106 + 341)/(7106 + 173 + 618 + 341) = 90\%$. This means the model correctly predicted the class 90% of the time.

0.3.3 K-Nearest Neighbour

K-Nearest Neighbors (KNN) is a supervised learning algorithm that can be used for classification and regression tasks. It is often used for datasets where the relationship between the features and the target variable is not clear and linear. In this case, it is used to model the data and predict whether a customer will subscribe to a term deposit or not, based on the other features in the dataset. KNN works by finding the k nearest data points to a given test point and then classifying the test point based on the majority class of the k nearest points. The number of nearest neighbours, k, is a hyperparameter that can be tuned to optimize the performance of the model.
It's also a good choice when the data set is small, because it does not require much training data, and it can be used to classify new examples that are similar to the ones in the training set.

0.3.4 Procedure

using the k-nearest neighbours (KNN) algorithm to train a classifier on the given training data (X_train, y_train) and then using it to make predictions on the test data (X_test).
The specific implementation is using the KNeighborsClassifier class from the scikit-learn library, with the number of neighbours to consider (n_neighbors) set to 22.
The fit function is used to train the model on the training data, and the predict function is used to make predictions on the test data.
The confusion matrix function and accuracy_score function are used to evaluate the performance of the model on the test data. The confusion matrix shows the number of true positives, true negatives, false positives, and false negatives. The accuracy score shows the proportion of correct predictions made by the model.
The cross_val_score function is used to estimate the performance of the model using k-fold cross-validation. The number of folds is specified by the cv parameter, which is set to k_fold. This will split the data into 10 folds and train and evaluate the model on each fold. The n_jobs parameter is used to specify the number of CPU cores to use for parallel computation. In this case, it is set to 1, so only one core is used. Finally, the scoring parameter is set to 'accuracy' which means it will return the accuracy of the model, and the mean of all the accuracy of each fold is returned.

	T	F
T	7163	116
F	706	253

In this case, the confusion matrix shows that the KNN model has made 7163 correct predictions for the negative class and 253 correct predictions for the positive class. It has also made 116 incorrect predictions for the negative class and 706 incorrect predictions for the positive class. The accuracy score for this model is around 90%.

0.3.5 Support Vector Machine

SVMs are particularly well-suited for classification problems with complex, non-linear decision boundaries. This is because, unlike other algorithms such as logistic regression, SVMs can effectively capture non-linear decision boundaries by transforming the input data into a higher-dimensional space where a linear decision boundary can be applied.
In this data modeling, SVM is used as a choice because it is a powerful algorithm that can handle complex, non-linear decision boundaries, that can be useful when the data is not linearly separable. Additionally, SVM is less prone to overfitting than other algorithms such as decision trees, which can be an issue when the dataset is not very large.

0.3.6 Procedure

the SVM algorithm is being trained using the X_train data and y_train labels, and then being used to make predictions on the X_test data. The confusion matrix and accuracy score are then printed to evaluate the performance of the model. It's also using the cross-validation method (KFold) to evaluate the performance.

	T	F
T	6719	560
F	605	354

In this specific confusion matrix, the model predicted 6719 times "no" and 560 times "yes" when the actual value was "no". Therefore, 6719 are the true negatives. 560 times the model predicted "yes" when the actual value was "no", these are the false positives.
The model also predicted 605 times "no" and 354 times "yes" when the actual value was "yes". Therefore, 354 are the true positives. 605 times the model predicted "no" when the actual value was "yes", these are the false negatives.
The accuracy of 86% is the ratio of correctly predicted observations to the total observations. $(TP + TN)/total$.

0.4 Other models that I use for educational purposes

0.4.1 Decision tree

A Decision Tree is an ideal choice for bank marketing data because it can handle both categorical and numerical data, and it can help to identify important features

for the prediction of the target variable. It can also help to visualize the relationships between different variables, and it's a straightforward and easy-to-interpret method. Additionally, Decision Trees can handle non-linear relationships and can handle large datasets. These capabilities make Decision Trees a popular choice for classification problems, like in the bank marketing data, where we want to predict the target variable (e.g. whether a customer will subscribe to a term deposit or not) based on other variables in the data.

0.4.2 Random forest

Random forest is a type of ensemble machine learning algorithm that is used for classification and regression problems. It is ideal for Bank marketing data because of its several advantages:

- Handling missing values: Random forests can handle missing values in the data, which is a common issue in real-world datasets.
- Non-linearity: The algorithm can handle non-linear relationships between features and target variables, making it suitable for complex datasets.
- Outlier detection: Random forest can detect outliers in the data and reduce their influence on the model.
- Feature importance: Random forest algorithms can determine the importance of each feature, helping to identify the most important predictors for the target variable.
- Improved accuracy: The algorithm uses multiple decision trees and aggregates their predictions, which helps to reduce overfitting and improve the overall accuracy of the model.

These advantages make the random forest a suitable algorithm for the Bank marketing data.

0.4.3 Gaussian Naive Baysian

Gaussian Naive Bayesian (GNB) is a simple probabilistic classifier based on Bayes' theorem, with strong independence assumptions between features. Gaussian Naive Bayesian algorithm can be ideal for Bank marketing data if the data is well structured and the features are independent of each other. GNB is particularly useful when there are many features and they are mostly independent, making the computational cost of the algorithm low. GNB is also very fast to train and is easy to understand, making it a useful starting point for many data problems.
However, GNB assumes that the features are normally distributed and independent of each other, which might not always hold for real-world data. In such cases, GNB might not perform well and other algorithms might be more appropriate.

0.4.4 Gradient Boosting

Gradient Boosting is a machine learning technique that can be ideal for Bank Marketing data because it has the capability to handle large amounts of data, can work well with non-linear data, and has high accuracy in modelling complex

relationships in the data. Gradient Boosting combines weak models (e.g., decision trees) to create an ensemble model that is stronger than the individual models. This technique is particularly effective for data with a large number of features and high dimensionality, as it can reduce the risk of overfitting the training data. Additionally, Gradient Boosting can be used for both regression and classification problems, making it suitable for a wide range of use cases, including Bank Marketing data.

0.4.5 XGBoost

XGBoost, short for Extreme Gradient Boosting, is a powerful machine learning algorithm that can be used for various tasks such as classification and regression. XGBoost is known to handle large datasets and work well with noisy or incomplete data, which makes it a good choice for bank marketing data.

In bank marketing data, the goal is to predict if a customer will subscribe to a term deposit or not. XGBoost can handle a large number of features and variables and can model complex non-linear relationships between them, which can be beneficial for this type of problem. XGBoost also has an implementation of gradient boosting which is a powerful technique for handling imbalanced datasets, which is a common issue in bank marketing data.

Additionally, XGBoost has built-in feature importance calculation and parameter tuning capabilities, which can help in identifying the most important features and optimizing the model parameters for better performance.

0.5 All models used and accuracy score table

	Models	Score
7	Gradient Boosting	0.914203
6	XGBoost	0.912200
4	Logistic Model	0.909772
0	Random Forest Classifier	0.909499
3	K-Near Neighbors	0.904643
1	Decision Tree Classifier	0.885190
2	Support Vector Machine	0.856055
5	Gaussian NB	0.845038

A Receiver Operating Characteristic (ROC) curve is a graphical representation of the performance of a binary classifier system as the discrimination threshold is varied. The ROC curve plots the true positive rate (sensitivity) against the false positive rate (1-specificity) at various threshold settings.

The ROC curve can be used to evaluate the trade-off between the true positive rate and the false positive rate of the model. A ROC curve plots the true positive rate (sensitivity) against the false positive rate (1-specificity) at various threshold settings. The closer the ROC curve is to the top-left corner, the better the model is at classifying the positive class.

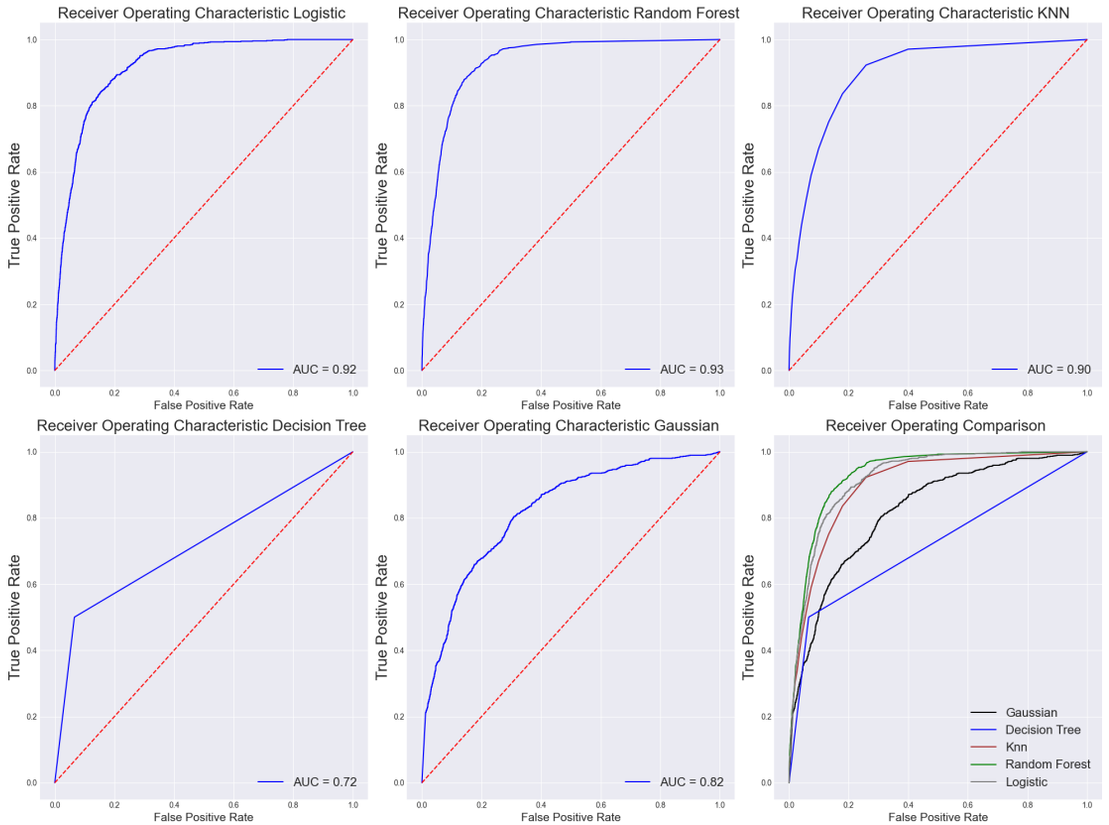

Figure 9: ROC Curve for all models.

The ROC curve would help to visualize the trade-off between the sensitivity (True positive rate) and specificity(1-False positive rate) of the model, which can help to identify the threshold setting that provides the best balance of true positives and false positives.

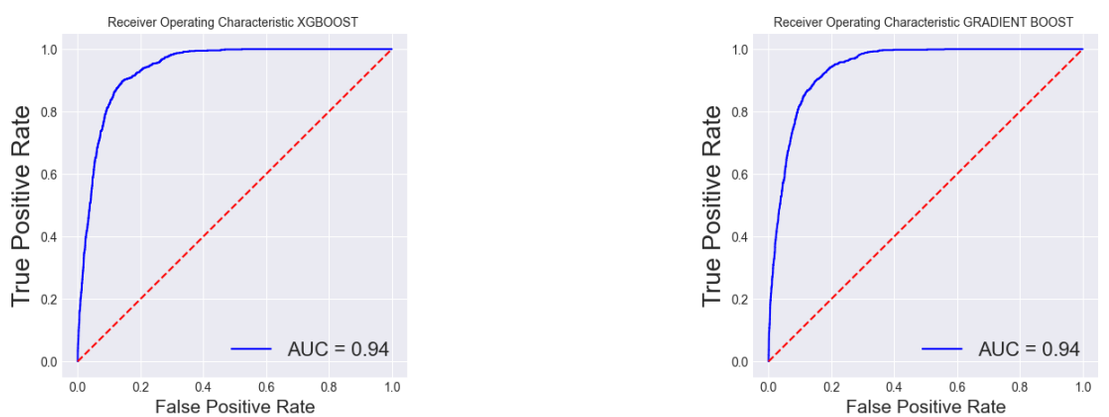

Figure 10: ROC Curve for XGBoost model.

In conclusion, XGBoost's ability to handle large and complex datasets, work well with noisy or incomplete data, handle imbalanced datasets, and built-in feature importance calculation and parameter tuning capabilities make it a good choice for bank marketing data and as indicated it shows an impressive performance.

0.6 Conclusion

The majority of the people in the dataset are married and single. This suggests that the dataset may be skewed towards people who are not divorced or widowed. The biggest three groups represented in the dataset are administrative jobs, blue-collar, and technicians. These professions make up approximately more than 50% of the whole group. This could indicate that the dataset is focused on individuals who are employed in specific types of jobs.
The smallest three groups are represented by the unemployed, students, and an unknown group. The marketing team assumed that these groups might not have savings to deposit. This could be a potential area for future marketing efforts.
The campaign was not targeting self-employed entrepreneurs. This could mean that the department was targeting mainly individuals and not legal entities; also it can mean that deposit offers would be of interest only to individuals and not self-employed or entrepreneurs.
Groups "management", "retired" and "services" were not actively approached despite the fact that these groups might have savings to invest in deposits. This could be a missed opportunity for the department.
Overall, the data suggest that the marketing department has a specific target audience in mind and that there may be opportunities for future marketing efforts to reach a wider range of people.