Machine learning algorithms for solving real-world classification and clustering problems

Candidate Name : Aksa Annu George  
MSc Data Science Student Student ID : 13308049

Abstract—This research paper investigates the application of various classification machine learning algorithms on a dataset related to direct marketing campaigns of a Portuguese banking institution. The dataset contains information on phone calls made to clients, where multiple contacts were often required to determine if the client would subscribe to a term deposit. The classification goal is to predict whether a client will subscribe to the term deposit or not. The dataset is available in four versions, including a full dataset with 41,188 examples and 20 inputs, a 10% sample of the full dataset, and older versions with fewer inputs. The datasets are ordered by date, spanning from May 2008 to November 2010. In this research, different classification machine learning algorithms will be explored to predict the subscription outcome. These algorithms include but are not limited to Logistic Regression, Naïve Bayes, Random Forest, and K-Nearest Neighbors (KNN). The performance of each algorithm will be evaluated based on metrics such as accuracy, precision, recall, F1 score, and area under the receiver operating characteristic (ROC) curve. The research aims to compare the performance of these algorithms and identify the most effective approach for predicting term deposit subscriptions. Additionally, feature selection and preprocessing techniques will be explored to optimize the predictive models. The findings of this study will contribute to the understanding of how different classification algorithms can be applied to similar marketing datasets and provide insights for improving marketing campaign strategies.

Keywords: classification, machine learning algorithms, direct marketing, term deposit, predictive modeling

# **Introduction**

Direct marketing campaigns play a crucial role in the success of businesses by targeting potential customers and promoting their products or services. Effective marketing strategies require a deep understanding of customer behavior and preferences. Machine learning techniques have emerged as powerful tools for analyzing and predicting customer responses in marketing campaigns. In this paper, we explore the application of various classification machine learning algorithms on a dataset related to direct marketing campaigns of a Portuguese banking institution.

The dataset used in this study contains information on phone calls made to clients, where the goal was to determine if the client would subscribe to a term deposit. This dataset is available in four versions, including a full dataset with over 41,000 examples and a 10% sample of the full dataset. Additionally, older versions of the dataset with fewer inputs are also included for comparison.

The classification task at hand is to predict whether a client will subscribe to the term deposit or not. This prediction holds significant value for the banking institution as it can guide marketing efforts and optimize campaign strategies. By accurately identifying potential customers who are more likely to subscribe, the institution can allocate resources effectively and enhance conversion rates.

In this research, we aim to apply and compare the performance of various classification machine learning algorithms on the provided dataset. The selected algorithms include Logistic Regression, Naïve Bayes, Random Forest, and K-Nearest Neighbors (KNN). These algorithms are well-established and widely used in classification tasks, each with its own strengths and limitations.

By evaluating and comparing the performance of these algorithms, we seek to identify the most effective approach for predicting term deposit subscriptions. We will assess the algorithms based on several evaluation metrics such as accuracy, precision, recall, F1 score, and area under the receiver operating characteristic (ROC) curve. Additionally, we will explore feature selection and preprocessing techniques to enhance the performance of the predictive models.

The findings of this study will provide insights into the application of classification machine learning algorithms in direct marketing campaigns. Moreover, the results will contribute to the understanding of how different algorithms can be leveraged to improve marketing strategies and enhance customer targeting. The research outcomes will be valuable for the banking institution and can be extended to other industries and domains that rely on direct marketing campaigns.

In the following sections, we will provide an overview of related work in the field, discuss the dataset in detail, describe the methodology employed, present the experimental results, and conclude with discussions on the implications of the findings and potential areas for future research.

# **THE DATA SET**

Dataset was extracted from UCI Machine learning repository. The dataset includes 45211 entries, comprises 17 attributes and need to be preprocessed. Each data row is unique information about bank client data. Given dataset includes different information about the client with count of each client, thereby predicting the response value. The dataset has categorical variables as well as numerical values.

# TABLE 1. DATASET FEATURES

|  |  |  |
| --- | --- | --- |
| Description | Type | Description |
| age | Numerical | Age |
| job | Categorical | type of job |
| marital | Categorical | marital status |
| education | Categorical | Education |
| default | Categorical (binary) | has credit in default?. |
| housing | Categorical (binary) | has housing loan? |
| loan | Categorical (binary) | has personal loan? |
| contact | Categorical | contact communication type |
| month | Categorical | last contact month of year |
| day\_of\_week | Categorical | last contact day of the week |
| duration | Numerical | last contact duration, in seconds |
| campaign | Numerical | number of contacts performed during this campaign and for this client |
| native-country | Categorical | Country |
| pdays | Numerical | number of days that passed by after the client was last contacted from a previous campaign |
| previous | Numerical | number of contacts performed before this campaign and for this client |
| poutcome | Categorical | outcome of the previous marketing campaign |
| y | Categorical (binary) | has the client subscribed a term deposit? |

# **DATA PREPARATION**

In this part, the data is analyzed using the class imbalance process to overcome the imbalance problem. Categorical data is then preprocessed and coded, numerical data is analyzed and scaled, and features are extracted.

## Class Imbalance:

One issue with the dataset is the potential for class imbalance, where instances from one class (for example, term deposit subscribers) may vastly exceed instances from the opposite class (for example, non-subscribers). Class imbalance can have an impact on how well classification algorithms work since they may start to favour the dominant class. We will use a variety of strategies to balance the classes in order to overcome this issue. These methods include combination sampling, SMOTE (Synthetic Minority Over-sampling Technique), over- and under-sampling, and SMOTETomek. (Guest\_Blog, 2023)

A green and purple bar graph

Description automatically generated with low confidence

### Random Over-Sampling

Random oversampling involves randomly duplicating samples from the minority class to make up the same number of samples as the majority class. To make the count of the majority class equal, the minority class in the response is duplicated in the dataset. The number of classes is increased from 7841 to 24719 to match the count of class 0.

### Random Under-Sampling

In random undersampling, samples from the majority class are chosen at random to equal the samples from the minority class. The Response's majority class in the dataset gets downgraded to a minority class. To match the count of class 1, class 0 is reduced from 24719 to 7841.

### Combining Over and Under Sampling

In certain cases, combining the two sample approaches described above has produced more precise modelling outcomes than doing so independently. In this instance, the minority class is somewhat oversampled and the majority class is slightly undersampled. Following the application of combination sampling, the majority class 0 decreases from 24719 to 21972 and the minority class 1 increases from 7841 to 19775.

### SMOTE Over-Sampling

SMOTE chooses neighbouring examples in the feature space, connects the examples with a line, and then adds a fresh sample at a certain location along the line. The minority class count approached the majority after SMOTE was put into place.

### SMOTETomek

SMOTETomek is located between up sampling and down sampling. By utilising both under- and over-sampling, the SMOTETomek hybrid method combines the two approaches. Following the application of SMOTE, both classes' counts reached 24002.

## Handling Null Values:

Null or missing values can hinder the analysis and modeling process. Therefore, it is important to handle them appropriately. We will carefully examine the dataset to identify any missing values and implement suitable strategies to handle them. This may involve techniques such as imputation, where missing values are replaced with estimated values based on other observations, or deletion of instances with missing values, depending on the extent and nature of the missing data.

## Categorical and Numerical Variable Encoding:

The dataset contains both categorical and numerical variables that need to be properly encoded for the classification algorithms. We will address this encoding process using techniques such as label encoding and dummy variable encoding.

### Label Encoding:

We can use label encoding for categorical variables like "sex," "education," or "marital status." Each category in the variable receives a different number label thanks to label encoding. With this method, the categorical values are replaced with a numeric value that falls between 0 and the number of classes minus one. Label encoding helps the algorithms to efficiently process categorical variables by transforming them into numerical representations. (Great Learning Team, 2023)

### Dummy Variable Encoding:

Another common approach for encoding categorical variables is dummy variable encoding, also known as one-hot encoding. In this technique, we create new binary variables for each category in a categorical variable. Each binary variable represents whether the instance belongs to a specific category or not. By leaving out one class, known as the "dummy variable trap," we avoid multicollinearity issues in the data. Dummy variable encoding is particularly useful when there is no inherent ordinal relationship among the categories. (Saxena, 2022)

By applying label encoding and dummy variable encoding to the relevant categorical variables, we ensure that the data is properly represented in a numerical format suitable for the classification algorithms. This encoding process preserves the information within the categorical variables while allowing the algorithms to interpret and utilize them effectively during the training and prediction stages.

We also address the encoding of numerical variables by standardising them, ensuring that all features are on a same scale, in addition to categorical variable encoding. This standardisation stage ensures that the algorithms can effectively take the numerical variables into account during classification and prevents any one feature from controlling the modelling process.

By implementing the appropriate encoding techniques for categorical and numerical variables, we can effectively prepare the data for classification algorithms and improve the accuracy and performance of the predictive models. These encoding steps, along with other data preparation techniques mentioned earlier, form an essential part of the overall preprocessing pipeline for the direct marketing campaign dataset..

## Dimensionality Reduction:

We will use PCA in addition to variable encoding to tackle the problem of dimensionality reduction. PCA is a frequently used method for lowering a dataset's dimensionality while retaining the most crucial data by translating it into a lower-dimensional space. Principal component analysis (PCA) creates new variables, referred to as principal components, that represent the key characteristics of the dataset by determining the directions of maximum variation in the dataset.

Through PCA, we can reduce the number of variables while retaining as much information as possible. This reduction in dimensionality not only simplifies the subsequent analysis but also helps prevent overfitting and computational inefficiency. By selecting the optimal number of principal components based on their explained variance, we can strike a balance between reducing dimensionality and preserving the most relevant information for classification. PCA is performed in Python using the scikit-learn repository. (Towards Data Science, 2022)

A picture containing line, plot, diagram, text

Description automatically generated

With PCA, 32 components were chosen that gives a cumulative variance of 90%.

## Standardization:

Different input variables may have different scales or units of measurement. Standardizing the numerical variables is important to ensure that all features are on a similar scale, as some algorithms are sensitive to the magnitude of the input features. We will employ techniques such as z-score standardization or min-max scaling to normalize the numerical variables, bringing them to a comparable range.

The relevant data preparation stages, such as class imbalance handling, missing value treatment, categorical and numerical variable encoding, dimensionality reduction, and standardisation, will be implemented using Python and its many tools, including pandas and scikit-learn. various libraries offer practical functions and techniques for carrying out various preprocessing operations. We can ensure that the data is correctly structured, reduced in dimension, and standardised for efficient training and evaluation of the classification machine learning algorithms by properly preparing the data.

The following sections will detail the specific techniques and steps undertaken for class imbalance handling, missing value treatment, categorical and numerical variable encoding, dimensionality reduction, and standardization. These preprocessing steps are crucial for creating a robust and reliable predictive model for term deposit subscriptions in the direct marketing campaign dataset.

# **Machine Learning Classification Techniques**

In this paper, we will explore and apply various machine learning algorithms to the direct marketing campaign dataset. These algorithms will help us build predictive models to classify whether a client will subscribe to a term deposit or not. The following machine learning techniques will be employed:

## K-Nearest Neighbors (KNN)

KNN is a non-parametric algorithm that classifies instances based on their proximity to other instances in the feature space. It assigns a label to a new instance based on the majority label of its k nearest neighbors. KNN is simple to implement and can handle both classification and regression tasks. We will employ KNN to assess the similarity between clients and predict their subscription status based on the behavior of similar clients. Python (2022)

## Logistic Regression

Logistic regression is a widely used algorithm for binary classification problems. It models the relationship between the dependent variable and the independent variables using a logistic function to estimate the probability of an event occurring. We will leverage logistic regression to analyze the impact of different features on the likelihood of a client subscribing to a term deposit.

## Naïve Bayes Classifier

Naïve Bayes is a probabilistic classifier that utilizes Bayes' theorem to predict the probability of a certain class given the feature values. It assumes that the features are conditionally independent, which simplifies the computation. Naïve Bayes is particularly effective when dealing with high-dimensional datasets and can provide fast and efficient predictions. We will evaluate the performance of Naïve Bayes in classifying clients based on their characteristics.

## Random Forest

Random Forest is an ensemble learning method that combines multiple decision trees to make predictions. It creates a set of decision trees from randomly selected subsets of the training data and combines their predictions through voting. Random Forest is known for its ability to handle high-dimensional data and capture complex relationships between features. We will utilize this algorithm to improve the accuracy and robustness of our classification models.

We intend to create precise and reliable classification models for projecting the results of the direct marketing campaign by utilising these machine learning approaches. We can learn more about the most efficient method for this particular dataset by comparing the performances of the several algorithms, each of which has its own advantages and presumptions.

# **application of techniques and results**

Algorithms were implemented, and results obtained using HP Pavilion Laptop equipped with intel core i5 with intel iris Xe Graphics and 16.0 GB (15.4 GB usable).

## K-Nearest Neighbour(K-NN)

Using SMOTETomek and SMOTE sampling approach, the results demonstrate that KNN performed at its best. The second-best results came from the oversampling method. The comparably subpar outcomes were caused by undersampling.

A close-up of a graph

Description automatically generated with low confidence

A close-up of a graph

Description automatically generated with low confidence

A close-up of a graph

Description automatically generated with low confidence

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **KNN** | | | | | |
| **Class Imbalance** | Over Sample | Under Sample | Combination Sample | SMOTE | SMOTE-Tomek |
| **Accuracy** | .88 | .76 | .86 | .90 | .90 |
| **F1 Measure** | .88 | .76 | .86 | .90 | .90 |
| **Area Under ROC** | .94 | .845 | .932 | .956 | .956 |

## Logistic Regression

In this case, all algorithms yield comparable outcomes. The only sampling method with a little higher accuracy is SMOTETomek.

A picture containing text, screenshot, diagram, font

Description automatically generated

A picture containing text, screenshot, diagram, font

Description automatically generated

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Logistic Regression** | | | | | |
| **Class Imbalance** | Over Sample | Under Sample | Combination Sample | SMOTE | SMOTE-Tomek |
| **Accuracy** | .81 | .81 | .81 | .81 | .82 |
| **F1 Measure** | .81 | .81 | .81 | .81 | .82 |
| **Area Under ROC** | .889 | .893 | .890 | .894 | ..897 |

.

## Naïve Bayes

In Naïve Bayes, the algorithm accuracies ranged from 69% to 71% .Combination and SMoOTETomek performed slightly better.

A picture containing text, screenshot, font, diagram

Description automatically generated

A picture containing text, screenshot, font, diagram

Description automatically generated

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Naïve Bayes** | | | | | |
| **Class Imbalance** | Over Sample | Under Sample | Combination Sample | SMOTE | SMOTE-Tomek |
| **Accuracy** | .70 | .69 | .71 | .70 | .71 |
| **F1 Measure** | .70 | .68 | .70 | .69 | .70 |
| **Area Under ROC** | .806 | .795 | .811 | .809 | .819 |

## Random Forest

With Random Forest modelling, all the models high accuracies (above 90%). Out of them, oversampling produced best results followed by combination sampling.

A close-up of a graph

Description automatically generated with low confidence

A close-up of a graph

Description automatically generated with low confidence

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Random Forest** | | | | | |
| **Class Imbalance** | Over Sample | Under Sample | Combination Sample | SMOTE | SMOTE-Tomek |
| **Accuracy** | .97 | .81 | .96 | .92 | .93 |
| **F1 Measure** | .97 | .81 | .96 | .92 | .93 |
| **Area Under ROC** | .999 | .893 | .997 | .978 | .979 |

Random forest has the highest accuracy and Roc value compared to other models.

# **conclusion**

In this paper, we have investigated the application of various machine learning techniques to predict the outcome of a direct marketing campaign in a Portuguese banking institution. Through extensive data preparation, including handling class imbalance, null values, categorical and numerical variable encoding, dimensionality reduction using PCA, and standardization, we have prepared the dataset for analysis.

We applied different classification algorithms, including logistic regression, random forest, Naïve Bayes, and K-Nearest Neighbors (KNN), to build predictive models. These algorithms demonstrated promising performance in classifying whether a client will subscribe to a term deposit or not, with accuracies ranging from 69% to 97%.

Based on our analysis, we found that the Random Forest algorithm outperformed other techniques in terms of accuracy, precision, recall, and F1 score. It showed a robust ability to handle high-dimensional data and capture complex relationships between features. However, further analysis and comparison of the algorithms' performance on different evaluation metrics could provide additional insights.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **5Models** | **Sampling Method** | **Accuracy** | **F1 score** | **Area Under ROC** |
| **KNN** | SMOTE and  SMOTETomek | .90 | .90 | .956 |
| **Logistic Regression** | SMOTETomek | .82 | .82 | .897 |
| **Naïve Bayes** | SMOTETomek | .71 | .70 | .819 |
| **Random Forest** | Over sample | .97 | .97 | .999 |

# **future research**

Although we have achieved promising results in this study, there are several avenues for future research that can extend and enhance our findings:

Ensemble Methods: Investigate the use of ensemble methods, such as AdaBoost or Gradient Boosting, to further improve the predictive performance

Feature Engineering: Explore additional feature engineering techniques like Recursive Feature Elimination or L1 regularization to create new informative features that can better capture the underlying patterns and relationships in the data.

Hyperparameter Tuning: Perform a more extensive hyperparameter tuning to optimize the performance of each algorithm. Techniques like grid search or Bayesian optimization can be employed to find the optimal combination of hyperparameters.

Deep Learning Approaches: Evaluate the performance of deep learning models, such as convolutional neural networks (CNNs) or recurrent neural networks (RNNs), in predicting the outcome of the marketing campaign.

Real-time Implementation: Extend the research to real-time implementation, where the predictive models can be integrated into the marketing campaign workflow to provide real-time predictions and guide decision-making.

By pursuing these future research directions, we can further improve the accuracy, interpretability, and practical applicability of the predictive models in the context of direct marketing campaigns. These advancements can contribute to more effective and targeted marketing strategies, resulting in improved campaign outcomes and customer satisfaction.

# **reference**

1. DATASET - UCI Machine Learning Repository. (n.d.). Archive.ics.uci.edu. Retrieved February 13, 2012, from <http://archive.ics.uci.edu/ml/datasets/Bank+Marketing>
2. Sharma, A. (2020). Principal Component Analysis (PCA) in Python Tutorial. <https://www.datacamp.com/tutorial/principal-component-analysis-in-python> (Sharma, 2020)
3. Shafi, A. (2023). Random Forest Classification with Scikit-Learn. https://www.datacamp.com/tutorial/random-forests-classifier-python
4. Guest\_Blog. (2023). 10 Techniques to Solve Imbalanced Classes in Machine Learning (Updated 2023). Analytics Vidhya. https://www.analyticsvidhya.com/blog/2020/07/10-techniques-to-deal-with-class-imbalance-in-machine-learning/
5. Great Learning Team. (2023, May 31). What is Label Encoding in Python | Great Learning. Great Learning Blog: Free Resources What Matters to Shape Your Career! https://www.mygreatlearning.com/blog/label-encoding-in-python/
6. Python, R. (2022). The k-Nearest Neighbors (kNN) Algorithm in Python. realpython.com. https://realpython.com/knn-python/
7. Raschka, S. (2018). Python Machine Learning: Machine Learning and Deep Learning with Python, Scikit-Learn, and TensorFlow (2nd ed.). Packt Publishing.
8. Cole. (2021, December 14). Quickly Compare Multiple Models - Towards Data Science. Medium. <https://towardsdatascience.com/quickly-test-multiple-models-a98477476f0>
9. Brownlee, J. (2019). Naive Bayes Classifier From Scratch in Python. MachineLearningMastery.com. https://machinelearningmastery.com/naive-bayes-classifier-scratch-python/
10. Scikit-learn. (n.d.). Model Evaluation: Quantifying the Quality of Predictions. Retrieved from <https://scikit-learn.org/stable/modules/model_evaluation.html> (Scikit-learn, n.d.)

# **‌appendix**

**SOURCE CODE:**

<https://github.com/aksaannugeorge/Bank_Marketing.git>