Predicting Bank Term Deposits With Logistic Regression and PCA: A Mixed Data Approach.

David Agyemfra Atakora

Department of Mathematical Sciences
Montana State University

May 9th, 2024

A writing project submitted in partial fulfillment
of the requirements for the degree

Master of Science in Statistics

APPROVAL

of a writing project submitted by

David Agyemfra Atakora

This writing project has been read by the writing project advisor and has been found to be satisfactory regarding content, English usage, format, citations, bibliographic style, and consistency, and is ready for submission to the Statistics Faculty.

Date	Dr. Mark Greenwood Writing Project Advisor
Date	Dr. Katharine Banner Writing Projects Coordinator

Abstract

Predictor variables are often collected as both categorical and quantitative variables, with categorical variables creating rapid growth in model complexity when creating indicator variables for the levels of each variable, leading to estimation and interpretation challenges when used in models. Principal Component Analysis (PCA) provides a method for reducing the dimension of a suite of variables, retaining much of the original variation. This research investigates the predictive efficacy of logistic regression using the scores from a version of PCA for a mix of categorical and quantitative variables. A dataset from a Portuguese retail bank spanning 2008 to 2013, featuring customer attributes and socio-economic factors, including the impact of the financial crisis, is used to demonstrate the reduction of dimensionality of the suite of predictors and assess the performance of the resulting logistic regression model to classify deposit subscriber based on the derived features. This model is compared to a model using the original variables in terms of its predictive performance.

Contents

1 Introduction

1.1 Motivation

In an era of evolving financial landscapes, predictive modeling has become a cornerstone for financial institutions seeking to optimize customer engagement and enhance business profitability. One such domain is the banking sector, where understanding customer behavior and preferences is paramount for effective product development and targeted marketing strategies. In this context, the ability to predict bank term deposit subscriptions holds is critical, as it enables institutions to tailor their offerings to individual customer needs, optimize resource allocation, and ultimately foster long-term customer relationships (Su et al., 2006). Our approach integrates Principal Component Analysis (PCA) and Logistic Regression, two powerful methods renowned for their efficacy in extracting meaningful patterns from complex datasets and making accurate predictions. Through this endeavor, we seek to not only showcase the practical application of advanced analytics in the financial domain but also contribute to the growing body of knowledge surrounding predictive modeling in banking.

PCA is typically designed for continuous quantitative data, but for mixed data containing both quantitative and categorical variables, an extension called Principal Component Analysis of Mixed Data (PCAmix) can be used to reduce the dimension of the variables. Its use and interpretation are demonstrated with the example data set.

1.2 Background

Grzonka et al. (2016) compared classification methods to predict bank deposit decisions, finding previous campaign effectiveness as the most crucial factor. While a single decision tree achieved the highest true positive rate, random forests yielded

the best overall results. Parlar and Acaravci (2017) employed information gain and Chi-square methods to select key features from the dataset. They compared these methods using Naive Bayes, demonstrating that a reduced set of features enhances classification performance. Bahari and Elayidom (2015) proposed a CRM-data mining framework and examined Naïve Bayes and Neural Networks as classification models. Their findings suggest that Neural Networks offer superior accuracy compared to Naïve Bayes.

A bank term deposit, also known as a fixed deposit or time deposit, is a financial instrument offered by banks and financial institutions where funds are deposited for a specific period at a fixed interest rate. These deposits typically have a predetermined maturity date, and the depositor agrees not to withdraw the funds until the maturity date is reached. In return, the depositor receives interest payments on the deposited amount, which is usually higher than the interest rates offered on regular savings accounts. Bank term deposits are commonly used by individuals and businesses as a low-risk investment option to earn interest on their savings over a fixed period. The institution that collected these data would have been interested in using characteristics of clients to be able to determine which are most likely to use one of these financial instruments.

2 Data

2.1 Data Collection and Description

The dataset utilized in this study is sourced from the "Bank Marketing" dataset, compiled by Moro et al. (2014), and publicly accessible via the UCI Machine Learning Repository. The dataset comprises a total of 41188 observations, organized into 21 variables. Each observation represents a unique interaction between clients and a Portuguese retail bank, while the variables capture diverse attributes and indicators relevant to the banking domain. These variables include both categorical and quantitative types, reflecting aspects such as customer demographics, socio-economic factors, and economic metrics. Among the variables, there exists a binary response variable labeled "y", indicating whether the client has subscribed to a term deposit, with possible responses being "yes" or "no". Additionally, the dataset consists of 20 explanatory variables, with ten that are categorical and ten that are quantitative. Table 1 describes the variables in the dataset.

The dataset comprises several categorical variables with distinct levels. For the variable "job", representing the type of job, there are 12 levels including "admin", "blue-collar", "entrepreneur", "housemaid", "management", retired", "self-employed", "services", "student", "technician", "unemployed", and "unknown". The "marital" variable, indicating marital status, encompasses four levels: divorced, married, single, and unknown (where divorced also includes widowed individuals). Education level, denoted by the "education" variable, encompasses eight levels: "basic.4y", "basic.6y", "basic.9y", "high.school", "illiterate", "professional.course", "university.degree", and "unknown". For "default", indicating whether the client has credit in default, there are three levels: no, yes, and unknown. The "housing" variable, signifying whether the client has a housing loan, includes levels of no, yes, and unknown. Similarly, the "loan" variable,

representing whether the client has a personal loan, has levels of no, yes, and unknown. The "contact" communication type is categorized into two levels: cellular and telephone. The "month" variable, referring to the last contact month of the year, encompasses all twelve months from jan to dec. The "day_of_week" factor indicates the last contact day of the week, includes five levels: mon, tue, wed, thu, and fri. The "poutcome" variable, representing the outcome of the previous marketing campaign, includes three levels: failure, nonexistent, and success. Finally, the output variable "y", indicating whether the client has subscribed to a term deposit, has levels of yes and no. All missing values were represented as unknown and that creates an additional category of many of the categorical variables. Before beginning our data cleaning process, we convert all "unknown" values to "NA" which better describes the values as missing value. In section 2.2, we describe how we handled the missing values.

2.2 Data Cleaning

Various techniques are applied to ensure the quality and consistency of the dataset using the open-source statistical software R (R Core Team, 2023). The "pdays" variable wasn't relevant to our research objective so we took it out. The variable "default" was removed because it had only one level. Including such a variable in the model can lead to issues during model fitting, such as perfect separation. Categorical variables are converted to factor versions of the variables to prepare them for analysis to correctly handle their text coding. We randomly selected 1000 observations for further analysis in order to reduce the computational burden across the various methods explored. This enables analysis and modeling to proceed more swiftly due to the use of a smaller, more manageable subset. By taking a random sample, this ensures that the selected subset is representative of the overall data set. We ended up with a total of 19 variables, including 18 explanatory variables and

Table 1: Variable Description

No	Variable Names	Description	Type
1	age	Age of the client (years)	Quantitative
2	job	Type of job (12 levels)	Categorical
3	marital	Marital status (4 levels)	Categorical
4	education	Education level (8 levels)	Categorical
5	default	Has credit in default? (3 levels)	Categorical
6	housing	Has housing loan? (3 levels)	Categorical
7	loan	Has personal loan? (3 levels)	Categorical
8	contact	Contact communication type (2 levels)	Categorical
9	month	Last contact month of the year (12 levels)	Categorical
10	day_of_week	Last contact day of the week (5 levels)	Categorical
11	duration	Last contact duration (in seconds)	Quantitative
12	campaign	Number of contacts performed during this campaign	Quantitative
13	pdays	Number of days since the client was last contacted from a previous campaign	Quantitative
14	previous	Number of contacts performed before this campaign	Quantitative
15	poutcome	Outcome of the previous marketing campaign (3 levels)	Categorical
16	emp.var.rate	Employment variation rate (quarterly indicator)	Quantitative
17	cons.price.idx	Consumer price index (monthly indicator)	Quantitative
18	cons.conf.idx	Consumer confidence index (monthly indicator)	Quantitative
19	euribor3m	Euribor 3 month rate (daily indicator)	Quantitative
20	nr.employed	Number of employees (quarterly indicator)	Quantitative
21	y	Has the client subscribed a term deposit? (2 levels)	Categorical

one response variable on the 1,000 sampled observations.

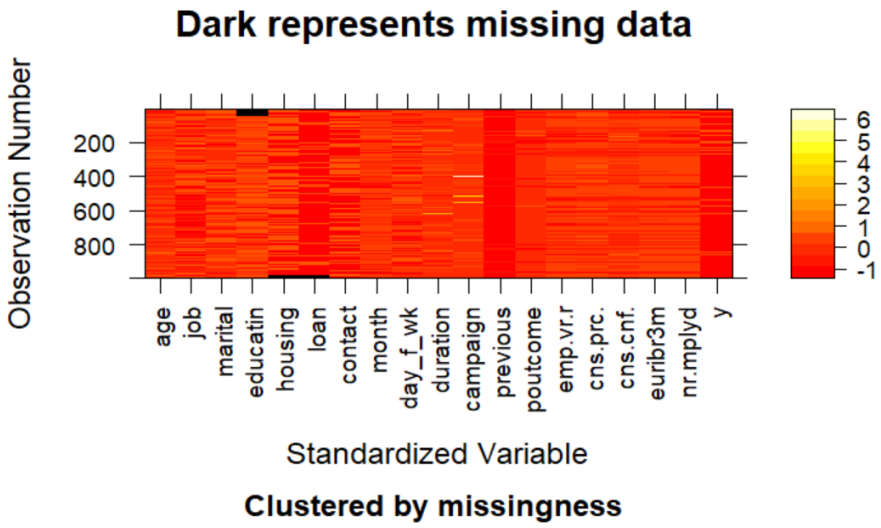

Figure 1: Plot of missingness patterns in 1,000 observation data set.

In Figure 1, the dark parts represent missing values and corresponds to each variable. The purpose of this plot is to visually represent missing data patterns within the dataset. It allows for a quick and intuitive understanding of which variables have missing values and the extent of those missing values, aiding in the data cleaning process. The results show that 83 observations were missing on five variables. Missing values are addressed by employing completion imputation methods, where the missing values are replaced by the mode of the respective variable within their class. This step helps maintain the integrity of the data and prevents bias in subsequent analyses. Figure 1 was made using the R package "mi" by Gelman and Hill (2011).

2.3 Data Splitting

The sample of 1,000 observations was further divided into training and test sets to facilitate model training and evaluation. Here, 80% of the data are randomly selected as the training set (800 observations), while the remaining 20% are allocated to the test set (200 observations). This ensures that the model is trained

on a sufficient amount of data while still retaining a portion for independent evaluation of the trained models.

2.4 Exploratory Data Analysis

In the Exploratory Data Analysis (EDA) phase, we delve into the dataset to gain insights and understand its characteristics better. Alluvial diagrams show how data move between categories across multiple dimensions by creating flows or "alluvia" for each observation or unique combination of results. Quantitative variables are binned into categories and labeled by the mean of the category. These displays help find patterns and relationships in the dataset, especially in the presence of multiple categorical variables. Alluvial plots were created using the R package "easyalluvial" by Koneswarakantha (2023).

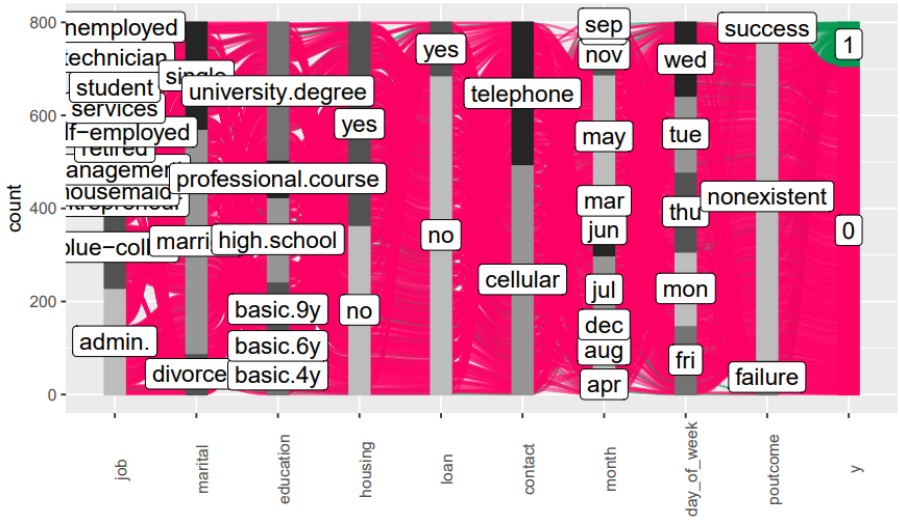

Figure 2: An alluvial plot of the categorical variables against the response variable for the n=800 training observations.

From Figure 2, we can discern the relationships between different levels of each categorical variable and their corresponding response outcomes. The plot visualizes the flow of observations across various levels of categorical predictors, showcasing how these levels contribute to different response outcomes. For the response, "0"

represents a client who has not subscribed to the bank term deposit and "1" represents a client who has subscribed to the bank term deposit. This visualization aids in understanding the distribution of the response variable across different levels of categorical predictors and highlights which levels are associated with higher or lower response rates. As seen, a higher response rate was associated to a client not subscribing to the bank term deposit. It also shows that most of the clients did not subscribe, so the overall success rate (0.1175) is quite low, which eventually will create challenges in the modeling and model evaluation steps.

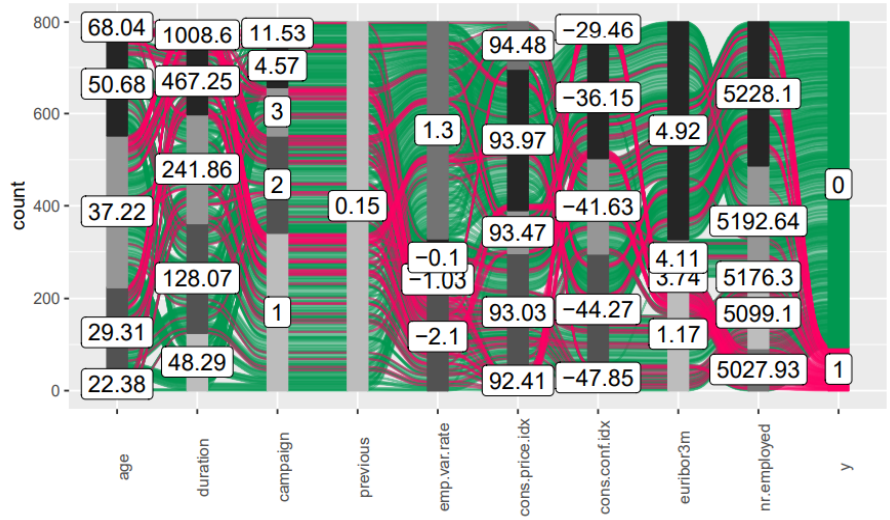

Figure 3: An alluvial plot of the quantitative variables against the response variable for the n=800 training observations.

Figure 3 illustrates the flow of observations across different bins or levels of each quantitative predictor, showcasing how these variables contribute to different response outcomes. The response outcome has the same meaning as explained in Figure 2 even though the colors have been switched.

3 Principal Component Analysis (PCA)

PCA is a widely used tool in data analysis across various fields. Its main objective is to find a new basis to represent data that highlights its underlying structure while filtering out noise. This method is utilized for tasks like dimensionality reduction, data compression, feature extraction, and visualization. By transforming a set of variables into a set of orthogonal variables called principal components, PCA helps extract important information and reduce noise in observations. These principal components represent linear combinations of the original variables and can be interpreted as projections of the data onto a new coordinate system. Essentially, PCA seeks to minimize the average distance between data points and their projections, making it an efficient technique for data analysis and interpretation (Pearson, 1901). This discussion is based on Autio et al. (2023), which is among many potential sources for understanding PCA.

3.1 Theory

First, the process of Principal Component Analysis (PCA) is outlined for analyzing a data set $X = [x_1, ..., x_N]$, structured as an $N \times M$ matrix where each column represents an observation of one of the M variables. Initially, the sample mean vector $\bar{x}$ and sample covariance matrix, $\hat{\Sigma}$, are defined as follows:

$$\bar{x} = \frac{1}{N} \sum_{i=1}^{N} x_i \tag{1}$$

$$\hat{\Sigma} = \frac{1}{N} \sum_{i=1}^{N} (x_i - \bar{x})(x_i - \bar{x})^T = \frac{1}{N} X^{\sim} X^{\sim T} \tag{2}$$

Here, $X^{\sim}$ represents the centered data matrix, derived from subtracting the mean vector from each observation. It is crucial to note the alternative formulation of the covariance matrix, $\hat{\Sigma}^*$, used in some PCA implementations, which employs an

unbiased estimator:

$$\hat{\Sigma}^* = \frac{1}{N-1}\sum_{i=1}^{N}(x_i - \bar{x})(x_i - \bar{x})^T = \frac{1}{N-1}X^{\sim}X^{\sim T}. \tag{3}$$

Moving forward, PCA extracts essential information from observations by computing factor scores as linear combinations of original variables: $y_{1i} = a_1^T(x_i - \bar{x}), \quad (i = 1, ..., N)$, where a_1 denotes the weight vector for these combinations for the first PC. The weight vector a_1 is optimized to maximize the sample variance of the new variable under the constraint $a_1^T a_1 = 1$. Utilizing Lagrange multipliers, the optimization problem is formulated as maximizing the Lagrange function: $L(a_1, \lambda_1) = a_1^T \Sigma a_1 - \lambda_1(a_1^T a_1 - 1)$. The remaining principal components are defined similarly but are constrained to also be orthogonal to one anoth. This framework explains the mathematical underpinnings of PCA, outlining its steps from data preprocessing to the derivation of optimized factor scores. A more detailed proof is found in Kurita (2019).

For each principal component, the weights are eigenvectors of the decomposition of $\hat{\Sigma}$ and the eigenvalues of that decomposition capture the relative variance of each those PCs. The sum of the eigenvalues equals the trace of the covariance matrix being decomposed and so the proportion of the total variance of the original variables accounted for by each PC can be found using this relationship.

3.2 Principal Components

We utilized the quantitative variables from the training dataset to conduct principal component analysis (PCA). We obtained a total of 9 principal components from the dataset.

Table 2 shows the loadings of principal components (PC) for 9 variables. Loadings indicate how much each variable contributes to a particular principal

component. These loadings are standardized/normalized, allowing for a direct comparison of the contribution of each variable across components. Here, we see the first few components (PC1 and PC2) shows more variables having high loadings. For example, PC1 seems to have high loadings on variables like "age", "emp.var.rate", and "euribor3m", suggesting it might capture factors related to economic conditions.

Table 2: Principal Components

Variable	PC1	PC2	PC3	PC4	PC5	PC6	PC7	PC8	PC9
age	-0.02	0.56	0.11	0.71	-0.33	-0.24	0.02	-0.01	0.01
duration	0.01	0.08	-0.77	0.34	0.53	0.11	-0.04	0.00	-0.01
campaign	-0.10	-0.31	0.56	0.49	0.48	0.35	0.00	0.00	-0.01
previous	0.28	-0.26	-0.18	0.21	-0.51	0.56	-0.45	0.03	0.01
emp.var.rate	-0.51	-0.07	-0.08	0.01	-0.10	0.06	-0.08	-0.77	0.34
cons.price.idx	-0.39	-0.22	-0.19	0.12	-0.31	0.25	0.69	0.32	0.08
cons.conf.idx	-0.10	0.68	0.11	-0.29	0.12	0.63	0.03	0.08	0.12
euribor3m	-0.51	0.04	-0.04	-0.03	-0.06	0.05	-0.23	0.00	-0.82
nr.employed	-0.48	-0.03	0.00	-0.02	0.03	-0.18	-0.50	0.55	0.43

3.3 Screeplot and biplot

The screeplot (Figure 4) shows the eigenvalues for each principal component (PC) in the PCA analysis. The eigenvalues are plotted in descending order on the y-axis, which represents the variance explained by each component. Eigenvalues represent the total amount of variance that can be explained by a given principal component. The x-axis represents the component number.

The elbow method looks for the point where the slope of the screeplot changes abruptly, forming an elbow-like shape. By looking for an "elbow" in the curve, we can determine the number of principal components where there are diminishing returns in terms of total variance explained by adding additional components. Often, components to the left of the elbow explain large portions of the variance, while those to the right contribute progressively less. The screeplot for the nine quantitative predictors in the training data shows a sharp elbow at component 2,

suggesting that the first two principal components capture a Substantial portion of the variance and that no additional components are needed here.

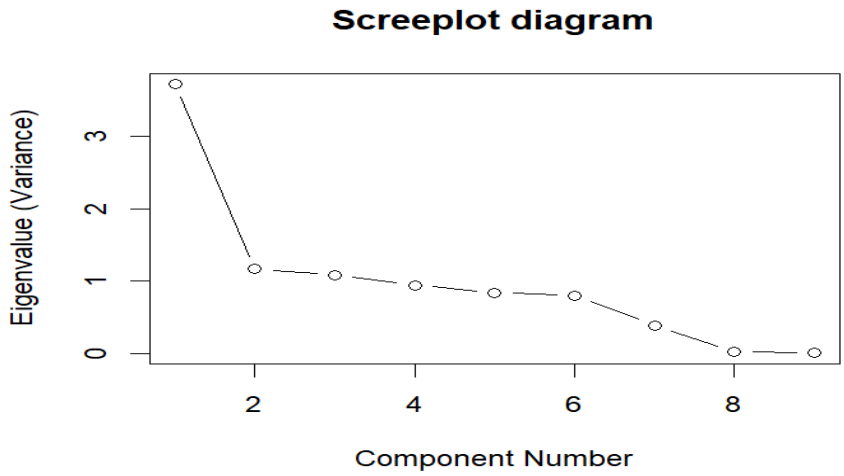

Figure 4: Screeplot of the eigenvalues for each principal component (PC) identified in the PCA analysis in the training data on the quantitative predictors.

The biplot (Figure 5) is a graphical tool used to visualize the relationships between variables and observations (data points) in PCA. The data points (in grey color) represent the observations in the dataset. Their positions on the biplot are determined by their scores on the two principal components (PC1 and PC2) used to create the plot. In this case, points closer together tend to be more similar based on the principal components.

The vectors (arrows) represent the original variables in the PC-space and are based on the eigenvectors (loadings). The direction of an arrow indicates how the original variables contribute to that component. The length of the vector reflects the amount of variance explained by that component. The 'previous' variable appears in the lower right quadrant of the biplot, suggesting a positive association with PC1 and a negative association with PC2. This aligns with the loadings table (Table 2), where 'previous' has a positive value (0.28) for PC1 and a negative value (-0.26) for PC2. In contrast, the "duration" variable is located closer to the origin, indicating a

weaker influence from both PC1 (loading of 0.01) and PC2 (loading of 0.08).

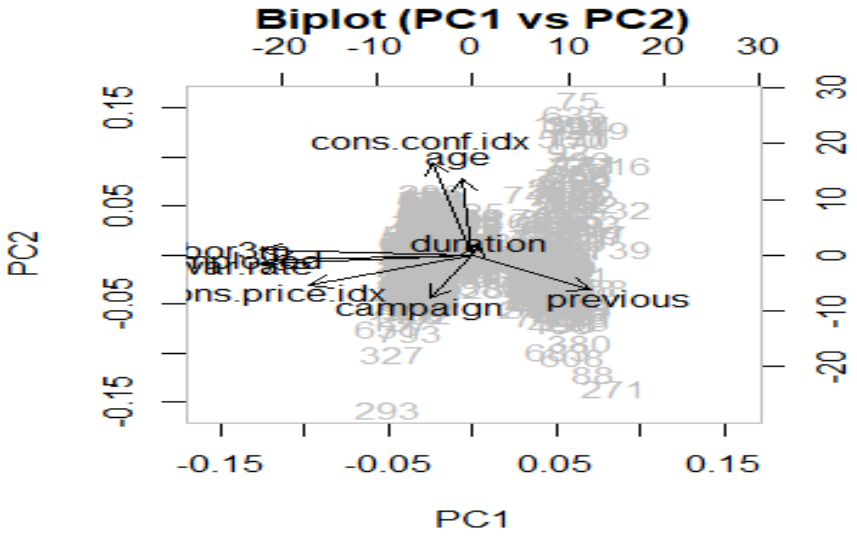

Figure 5: Biplot of the PCA of the quantitative predictors in the training data.

4 Principal Component Analysis of Mixed Data (PCAmix)

Traditional PCA, as discussed previously, is a powerful technique for analyzing datasets consisting solely of quantitative variables. However, real-world data often encompasses a mix of quantitative and qualitative variables. This presents a challenge for standard PCA, as it cannot directly handle categorical data.

To address this limitation, Principal Component Analysis of Mixed Data (PCAmix) emerges as a versatile extension. PCAmix incorporates both quantitative and qualitative variables, allowing for a more comprehensive analysis of datasets with mixed data types. Chavent et al. (2011) proposed PCAmix as a method for generating variables to create a version of PCA that can adapt to conventional PCA (quantitative variables only) or can handle all categorical variables (Multiple Correspondence Analysis) or a mix of quantitative and categorical variables. It resembles conventional PCA in that it creates a single set of orthogonal lower dimensional variables to represent as much variability as possible in the original data set.

4.1 Theory

In this framework, a real matrix Z of dimension $n \times p$ is considered, where n represents the number of observations and p represents the number of variables. The key concept is to project the rows of Z onto axes spanned by vectors by $V_1, ..., V_r$, where r denotes the number of principal components to be retained.

The principal component scores, also known as factor coordinates, are calculated as the projections of the rows of Z onto these axes. These scores are stored in a matrix F of size $n \times r$. The relationship between Z, M (inner product matrix), and V is expressed by $F = ZMV$. The factor scores matrix F is also

equivalent to $U\Lambda$, where U contains the left-singular vectors and Λ is a diagonal matrix containing the singular values. The loadings represent the relationships between the original variables (columns of the data matrix Z) and the principal components.

4.2 Principal Components

The principal components for the predictor variables in the training data set were obtained using the R package "PCAmixdata" by Chavent et al. (2017). The total number of principal components extracted by PCAmix was forty-four (44), which relates to the total number of dimensions implied by the mix of 9 quantitative variables and 9 categorical variables with varying numbers of levels.

Table 3 shows the squared loadings of the first ten principal components obtained from the PCAmix analysis. These loadings indicate the correlation between each variable and the corresponding component. The interpretation of loadings for qualitative variables depends on the levels of the variable. For example, the "marital" status variable has three levels (married, divorced, and single). This can influence how we interpret the strength and direction of the association and other tools are needed to fully understand the new dimensions created from the categorical variables.

Table 3: Squared loadings using PCAmix for the predictors in the training data set.

	dim 1	dim 2	dim 3	dim 4	dim 5	dim 6	dim 7	dim 8	dim 9	dim 10
age	0.00	0.49	0.14	0.00	0.01	0.02	0.03	0.01	0.00	0.00
duration	0.00	0.01	0.00	0.00	0.00	0.05	0.04	0.04	0.00	0.03
campaign	0.03	0.00	0.01	0.01	0.02	0.01	0.15	0.05	0.00	0.17
previous	0.37	0.04	0.02	0.03	0.31	0.02	0.00	0.02	0.02	0.00
emp.var.rate	0.87	0.00	0.01	0.00	0.05	0.00	0.00	0.00	0.01	0.00
cons.price.idx	0.55	0.06	0.05	0.05	0.11	0.01	0.02	0.00	0.02	0.00
cons.conf.idx	0.05	0.22	0.08	0.27	0.01	0.03	0.08	0.01	0.00	0.01
euribor3m	0.88	0.00	0.01	0.00	0.03	0.00	0.00	0.00	0.01	0.00
nr.employed	0.80	0.00	0.02	0.05	0.01	0.00	0.00	0.00	0.02	0.00
job	0.10	0.46	0.52	0.21	0.29	0.51	0.37	0.40	0.39	0.33
marital	0.05	0.24	0.11	0.03	0.00	0.08	0.01	0.05	0.00	0.07
education	0.03	0.13	0.51	0.16	0.18	0.50	0.37	0.13	0.29	0.05
housing	0.00	0.02	0.00	0.04	0.01	0.09	0.00	0.04	0.02	0.08
loan	0.00	0.01	0.00	0.01	0.00	0.02	0.01	0.04	0.01	0.01
contact	0.35	0.02	0.10	0.27	0.01	0.02	0.00	0.02	0.03	0.00
month	0.41	0.32	0.40	0.49	0.29	0.09	0.27	0.37	0.31	0.23
day_of_week	0.01	0.02	0.02	0.07	0.02	0.01	0.11	0.04	0.16	0.26
poutcome	0.40	0.02	0.02	0.12	0.31	0.06	0.00	0.10	0.03	0.02

4.3 PCAmix Analysis Plots

Having identified the principal components (PCs) from our mixed data analysis using PCAmix, the next step is to delve deeper into their characteristics and how they relate to the original data. Visualizations play a crucial role in this exploration. This section presents various plots generated from the PCAmix analysis to aid in interpreting the components and understanding the underlying structure of the data.

This scatterplot (Figure 6) depicts the distribution of observations in the space spanned by the first two principal components (Dimension 1 and Dimension 2) obtained from the PCAmix analysis. The points are colored according to the categories of the response variable "y". The response variable is "Has the client subscribed a term deposit?" (2 levels: Yes or No). Each point represents a data point (client) from the original dataset. The position of a point on the x and y axes reflects its scores on the first two principal components.

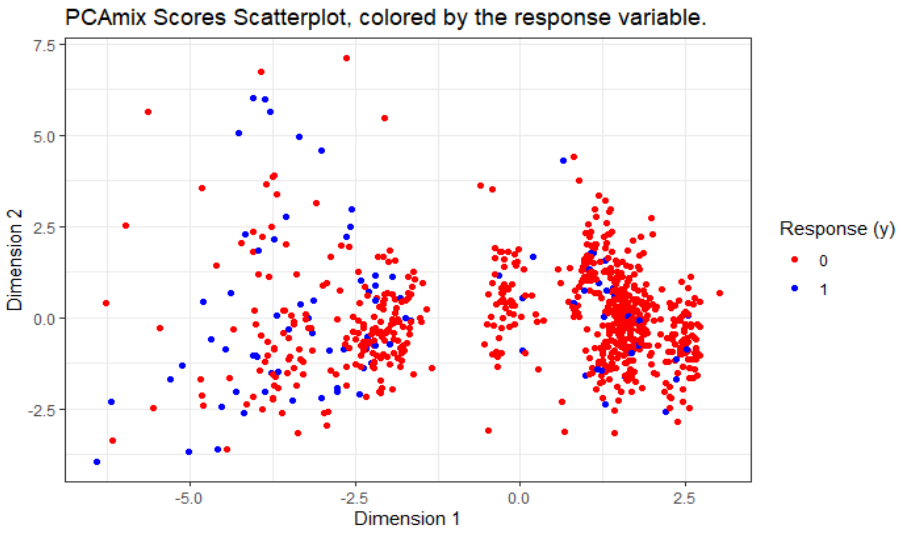

Figure 6: Scatterplot of PCAmix Scores in Dimension 1 and Dimension 2, colored by the response variable indicating whether the client has subscribed to a term deposit.

There seems to be some minor clustering of the data points. The red points (clients who did not subscribe) appear to be concentrated, while the blue points (clients who subscribed) show a little more spread throughout the plot. Coloring by the response variable helps us investigate whether subscription behavior is reflected in the positions of the points within the principal component space.

The correlation circle plot (Figure 7) visually represents the correlations between each numerical variable and the principal components extracted by PCAmix. The distance of a variable from the center of the circle indicates the strength of its correlation with the two principal components (PC1 and PC2) extracted by PCAmix. Variables closer to the center have weaker correlations, while those further away have stronger correlations (either positive or negative). The "duration" variable is closer to the center. This confirms a weaker correlation based on the squared loadings in Table 3 where its squared loading for Dimension 1 was 0.00 and Dimension 2 was 0.01.

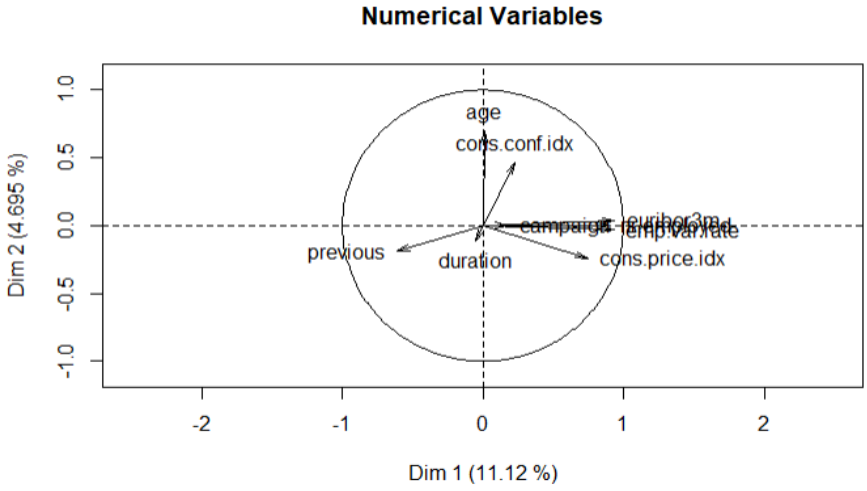

Figure 7: Correlation circle plot illustrating correlations between numerical variables and principal components in PCAmix analysis for predictor variables in the training data.

Figure 8 is a plot from the PCAmix analysis that visualizes the levels of the qualitative variables. The plot represents the categories (levels) of each qualitative variable as points or symbols within the principal component space. The position of a point for a specific level indicates how that level contributes to the overall variation captured by the principal components. From Figure 8, we can also observe some levels of different variables grouped at certain areas. Levels from different variables that cluster together in the PCAmix levels plot likely contribute similarly to the variation captured by the principal components (PC1 and PC2). This suggests that these levels might share some underlying characteristics that influence the data in a similar way. For example, if one variable represents "education level" and another represents "job title," the levels from both variables might indicate that certain education levels are typically associated with specific job titles.

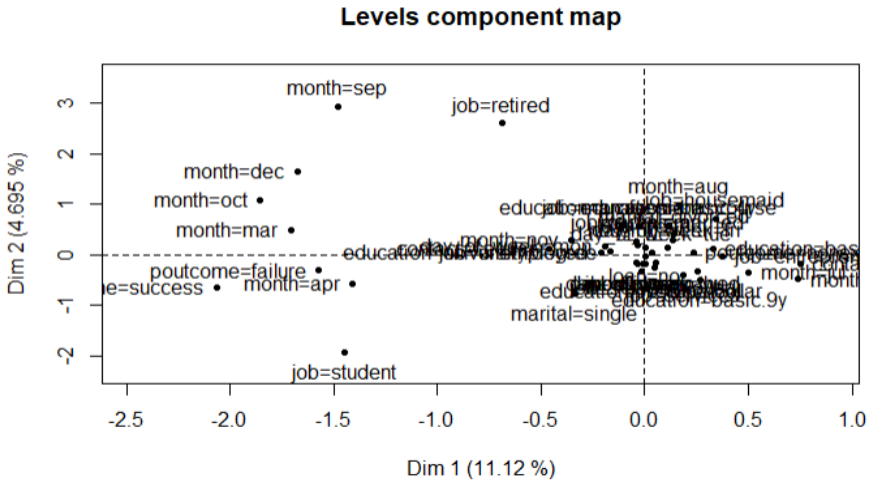

Figure 8: PCAmix Levels Plot: Visualization of qualitative variable levels in the principal component space.

Figure 9 plots all the variables (categorical or numerical) using squared loadings. Each variable's location reflects its "squared loading" on two key components. Squared loadings act like a strength measure - higher values indicate a stronger connection between the variable and a principal component. This applies to both numerical variables (using squared correlations) and categorical variables (using correlation ratios). We can observe that variables or vectors close to any axis indicates a high or low contribute to that axis. For instance, "age" and "job"" are close to the PC2 axis. They represent a higher contribution in the second principal component than the first principal component. In Table 3, "age" has a squared loadings of 0.49 for PC2 and 0.00 for PC2. We can see a similar interpretation for the "job" variable. This has 0.46 for PC2 and 0.10 for PC1. The "month" variable stays within the distance of both axes. This also indicates a contribution for both principal components. The "month" variable has a contribution of 0.41 under PC1 and 0.32 under PC2.

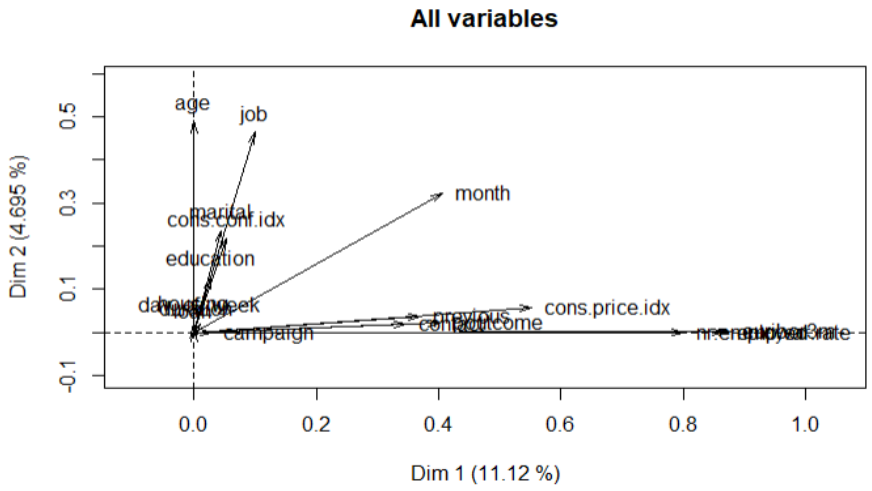

Figure 9: Plot of squared loadings for all variables in PCAmix analysis

Figure 10 shows an alluvial plot of the variables against the first principal component to help understand the main drivers of the scores of the first PC from the PCAmix on the predictors in the training data. We arrange the variables according to their contribution to the first principal component and with a cutoff value of 0.3, arranged from the least contribution on the left and end with the highest contribution on the right side. From the alluvial plot, we can see a lot of direct or straight mapping from the very few variables to the first principal components, which suggests very strong positive relationships for quantitative predictors and clear relationships between the levels of the categorical variables and the PC scores created. The variables in order of importance are "contact", "previous", "poutcome", "month", "cons.price.idx", "nr.employed", "emp.var.rate" and "euribor3m". This suggests that the first PC represents a combination of both quantitative and categorical variables with much contributions coming from the quantitative variables such as the "nr.employed", "emp.var.rate" and "euribor3m". Larger values of the first PC are related to higher values of "nr.employed", "emp.var.rate" and "euribor3m" variables. They are also related to some levels of the "month" variable, in particular, June is related to the highest values on this PC

and May and July often occur with the next highest scores on this PC, while April is associated with the smallest values.

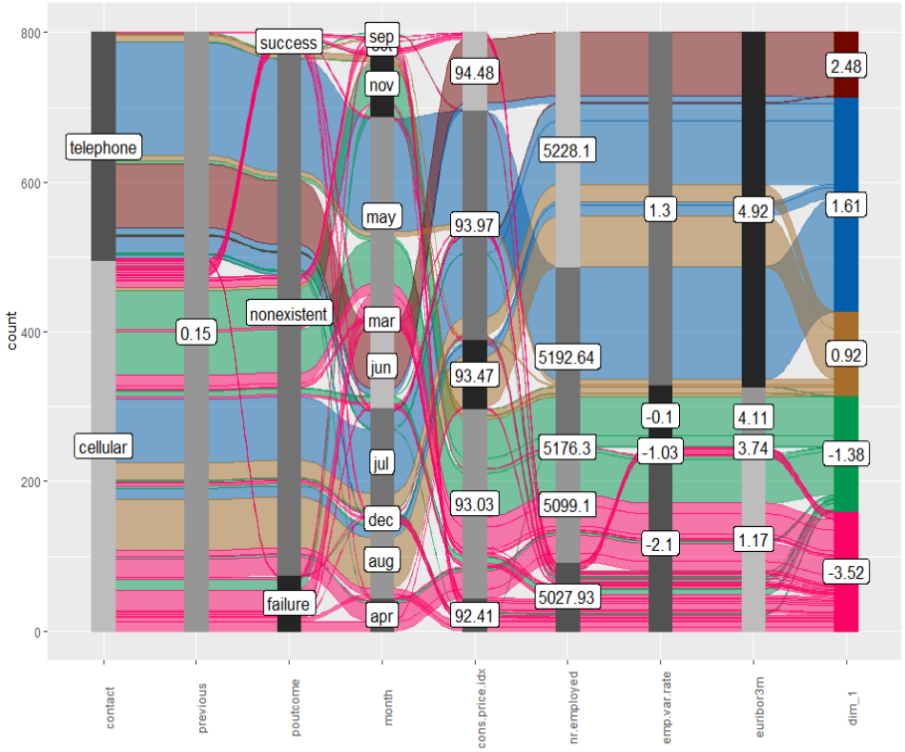

Figure 10: An alluvial plot of the variables with high contribution against the first principal component for PCAmix on the training data.

In Figure 11, we again plot an alluvial plot but the variables are against the second principal component. We also arrange the variables according to the contribution to the principal component with a cutoff of 0.2, which provides 5 variables in Figure 11. A straight path through the plot indicates that a variable is strongly associated with a particular category. The bends in the path indicate potentially less strong relationships. With categorical predictors, the sorting of the levels might lead to crossing but also there might not be as clear a relationship shown in the plot because the squared loadings (Table 3) were lower and this PC explains less variation in the variables than the first PC. The "age" and "job" variable had a contribution of 0.49 and 0.46 respectively and are the only two variables with at least 0.40 contribution to the second principal component. Larger

values of the second PC are related to older age clients in Figure 11. Also, the "retired" level of the "job" variable is related to the largest value of the second PC and "student" tends to occur with the lowest values.

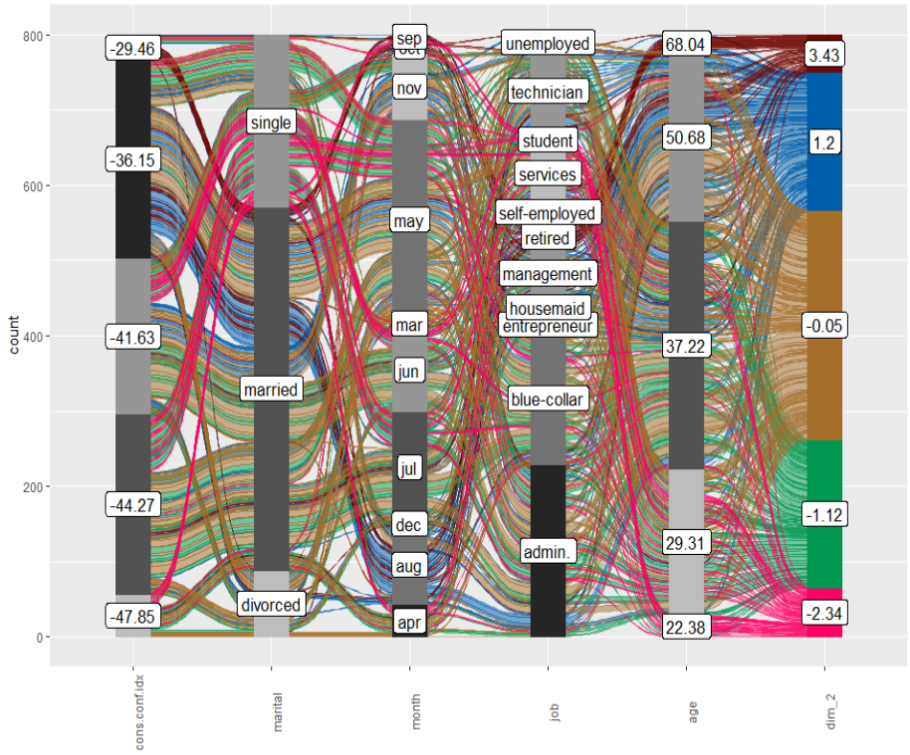

Figure 11: An alluvial plot of the variables against the second principal component for PCAmix on the training data.

5 Logistic Regression

Logistic regression is a statistical model for a binary (success/failure) outcome that is driven by a Bernoulli distribution for the response as a function of the probability of success, π. The probability of success is related to the predictors using the logit link function, which allows us to predict the probability of a specific outcome based on one or more predictor variables. The logit function is:

$$\text{logit}(p) = \log\left(\frac{p}{1-p}\right)$$

where p represents the probability of success. The logit function transforms the probability of success from the range (0,1) to the entire real number line (-∞, ∞), which has units then of log-odds of probability of success and makes it suitable for modeling with the systematic component of the generalized linear model that relates the predictors to the log-odds of success.

We used a logistic regression model because the response variable (Y) has only two levels: "Has subscribed a term deposit" (Yes/No) and we want to predict the probability of a customer subscribing to a bank term deposit. Logistic regression is particularly well-suited for this scenario where the outcome variable has two categories and we're interested in estimating the probability of belonging to one category based on the predictor variables.

The theoretical model is given by:

$$logit(\pi) = \beta_0 + \beta_1(X_1) + ... + \beta_i(X_I) \tag{4}$$

$$subscribe \sim Bernoulli(\pi) \tag{5}$$

In the model, $logit(\pi)$ represents the log-odds of the probability π of a customer subscribing to a term deposit, β_0, β_2, to β_i are the coefficients of the predictor

variables, and X_1 and X_I are the predictor variables, which could be quantitative predictors or indicator variables that were created from the levels of categorical predictors. For a categorical predictor with K levels, $K-1$ indicator variables must be created.

5.1 Logistic regression model using the full data

Using all of the available predictors in the logistic regression model and assuming no interactions are present or needed, the logistic regression model included a combination of 9 quantitative and 9 qualitative predictor variables. This resulted in a model with 45 coefficients, which can make interpretation challenging. Table 4 includes the coefficients from the estimated model and an effects plot (Figure 12). (Fox and Weisberg, 2019) is provided to show the complexity of the model that is fit in this situation. The effects plot shows the estimated probability of success across the values in each predictor holding the other variables at their means (quantitative predictors) or at a weighted average of the results (categorical predictors). While each term can be interpreted conditional on the others (holding others constant), the overwhelming number of coefficients makes the model nearly uninterpretable. It might also be over-fit and suffering from other issues.

Table 4: Coefficients of the Logistic Regression Model (Full Data)

Coefficients	Estimate	Std. Error	z value	Pr(>|z|)
(Intercept)	-438.312	332.029	-1.320	0.18680
age	0.01220	0.01950	0.626	0.53160
jobblue-collar	0.63976	0.63091	1.014	0.31057
jobentrepreneur	0.31086	0.94052	0.331	0.74101
jobhousemaid	-0.09677	1.22270	-0.079	0.93692
jobmanagement	-0.46230	0.67182	-0.688	0.49137
jobretired	1.25613	0.80448	1.561	0.11842
jobself-employed	0.71187	0.75708	0.940	0.34707
jobservices	-1.80772	1.17003	-1.545	0.12234
jobstudent	-0.03233	0.81004	-0.040	0.96817
jobtechnician	0.84830	0.50296	1.687	0.09168
jobunemployed	-2.20543	1.52779	-1.444	0.14887
maritalmarried	-0.74216	0.50354	-1.474	0.14051
maritalsingle	-0.96996	0.59824	-1.621	0.10494
educationbasic.6y	-0.01886	1.34866	-0.014	0.98884
educationbasic.9y	0.65662	0.79365	0.827	0.40805
educationhigh.school	1.94921	0.77638	2.511	0.01205
educationprofessional.course	0.81986	0.86429	0.949	0.34282
educationuniversity.degree	2.07047	0.73127	2.831	0.00464
housingyes	-0.33708	0.33107	-1.018	0.30860
loanyes	0.12199	0.47363	0.258	0.79675
contacttelephone	-0.66377	0.69488	-0.955	0.33946
monthaug	0.01605	1.03183	0.016	0.98759
monthdec	-1.43533	1.76709	-0.812	0.41664
monthjul	0.12411	0.72683	0.171	0.86441
monthjun	-1.24693	1.15697	-1.078	0.28114
monthmar	1.56457	1.21375	1.289	0.19739
monthmay	-1.06725	0.69239	-1.541	0.12322
monthnov	-0.24958	0.99489	-0.251	0.80192
monthoct	-0.72901	1.23992	-0.588	0.55657
monthsep	-1.51716	1.75051	-0.867	0.38611
day_of_weekmon	0.62090	0.51888	1.197	0.23145
day_of_weekthu	-0.02575	0.53969	-0.048	0.96195
day_of_weektue	0.39460	0.51788	0.762	0.44609
day_of_weekwed	0.94692	0.51439	1.841	0.06564
duration	0.00583	0.00064	9.061	<2e-16
campaign	-0.04437	0.08349	-0.531	0.59509
previous	0.87263	0.68468	1.275	0.20248
poutcomenonexistent	1.25169	0.98338	1.273	0.20307
poutcomesuccess	0.75687	0.76760	0.986	0.32412
emp.var.rate	-1.31184	1.36265	-0.963	0.33569
cons.price.idx	3.12246	2.23266	1.399	0.16195
cons.conf.idx	0.14897	0.06420	2.320	0.02032
euribor3m	-1.36417	1.07279	-1.272	0.20351
nr.employed	0.02908	0.02600	1.118	0.26336

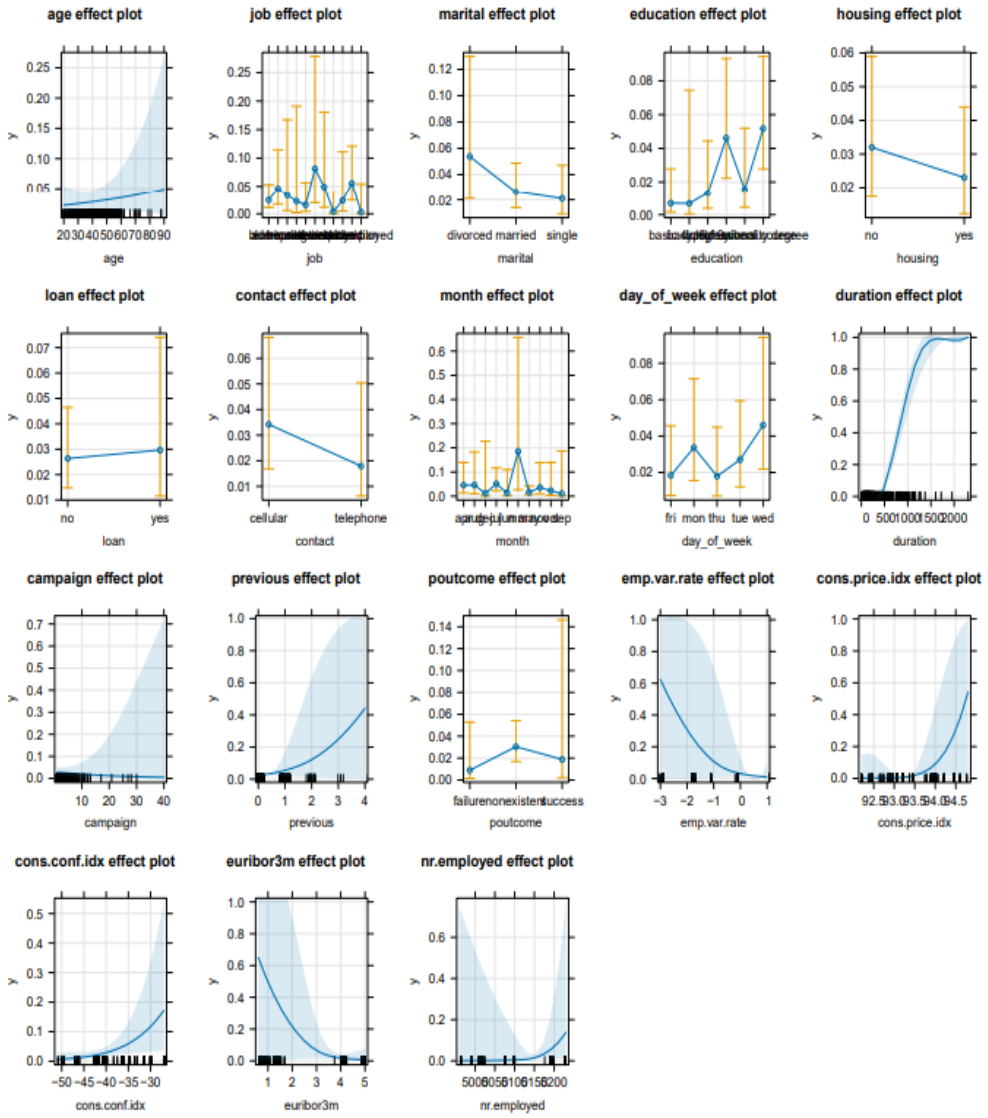

Figure 12: Effect plot of the logistic regression model on the full data.

One possible issue of including so many predictors in the model is multi-collinearity, which is a statistical phenomenon that occurs when two or more predictor variables in a model are highly correlated with each other. Table 5 reports the variance inflation factors and generalized variance inflation factors for this full model. The table has four columns and the second column is the GVIF which shows the VIF values for single degree of freedom predictors and an extension of the VIF for multi-category predictors called the GVIF (Fox and Weisberg (2019)) for each

variable. A higher (G)VIF suggests a stronger potential for multi-collinearity, meaning the variance of that variable's coefficient might be inflated due to its correlation with other variables in the model. The next column is the *df* (degress of freedom) column. This column shows the degrees of freedom associated with each variable. Categorical variables with more categories will have higher *df* values. The last column displays the adjusted VIF scores. These scores take the original GVIF values and adjust them based on the degrees of freedom. This adjustment helps to compare the importance of variables on a more equal footing, especially for categorical variables with different numbers of categories and provides a value that is directly interpreted for all the terms as how many times higher the standard error(s) (SE) of the predictors due to multi-collinearity than they would have been without it present. So even a value of 1.5 would be worrying with the suggestion that the SE is 50% larger than it would have been if no multi-collinearity was present.

Based on a threshold of 10 and using the adjusted VIF, the variables that are highly impacted by multi-collinearity in the model are: "emp.var.rate", "cons.price.idx", "euribor3m" and "nr.employed". Some of the impacts are quite extreme and suggest that the variables should not all be used together. But that high amount of shared information might also suggest that we can usefully reduce the dimensionality of the predictor space to retain the useful shared information in some of these predictors.

Table 5: Variance Inflation Factors (VIF) for the Predictor Variables

Variable	GVIF	Df	GVIF^(1/(2*Df))
age	2.741699	1	1.655808
job	15.632133	10	1.147363
marital	2.060938	2	1.198164
education	5.417034	5	1.184067
housing	1.239478	1	1.113318
loan	1.240734	1	1.113883
contact	3.384086	1	1.839588
month	245.440494	9	1.357609
day_of_week	1.949551	4	1.087031
duration	1.681815	1	1.296848
campaign	1.139730	1	1.067581
previous	8.371005	1	2.893269
poutcome	9.457701	2	1.753664
emp.var.rate	266.800150	1	16.334018
cons.price.idx	104.861310	1	10.240181
cons.conf.idx	6.432055	1	2.536150
euribor3m	182.861780	1	13.522640
nr.employed	226.385271	1	15.046105

A useful summary of the logistic regression models that will help us to compare different approaches to modeling this response is to measure the percentage of variation explained by each model. There are a variety of approaches to finding R-squared values in generalized linear models, but we will use the "r.squaredGLMM" function from the MuMIn package (Bartoń (2023)). For the full model using all predictors, the R-squared is 0.6384, which suggests that this model explains 63.84% of the variation in subscriptions. This suggests that despite the issues with multi-collinearity and nearly impossible number of interpretations, the model is good at explaining the variation in the response.

5.2 Logistic regression model using the first two principal components

In the previous section, we explored a logistic regression model using the full set of original variables to predict customer subscription to bank term deposit. However, a large number of variables, especially those with mixed data types (categorical and quantitative), can lead to model complexity and potentially hinder its interpretability.

To address this challenge, we now turn to use the Principal Components, in particular to PCAmix that can help us condense the original set of variables into a smaller number of uncorrelated components, capturing most of the relevant information. To illustrate this idea, we will use a logistic regression model built using the first two principal components derived from the PCAmix analysis. More components could be extracted from the original 45 dimensional data, but then the model complexity is akin to the results using all the predictors and likely many of those predictors would not be needed. Table 6 shows the coefficients from the estimated model. For this model using only two principal components, the R-squared is 0.1767, which tells us that this model explains 17.67% of the variation in subscriptions. This is quite a bit lower than the model using all the predictors, but only uses two predictors to attain this amount of variation explained.

Table 6: Logistic Regression Coefficients

Coefficients	**Estimate**	**Std. Error**	**z value**	**Pr(>|z|)**
(Intercept)	-2.295	0.1355	-16.93	2.596×10^{-64}
dim_1	-0.3788	0.04951	-7.652	1.977×10^{-14}
dim_2	0.03937	0.06672	0.5901	0.5551

Figure 13 shows the effects plot of the logistic regression model using the first two principal components, which visualizes the relationship between predictor variables (two principal components) and the estimated probability a customer

subscribing to a bank term deposit, suggesting that higher values of the first PC are related to lower probabilities of subscription, controlled for the second PC. Specifically, for two similar clients differing by 1 standard deviation on the first principal component (PC1), we estimate that the client with the higher PC1 score has an estimated mean odds of subscribing to a bank term deposit of exp(-0.3788) = 0.6847 (95% profile likelihood CI: [0.62, 0.75]), controlling for the second principal component (PC2).

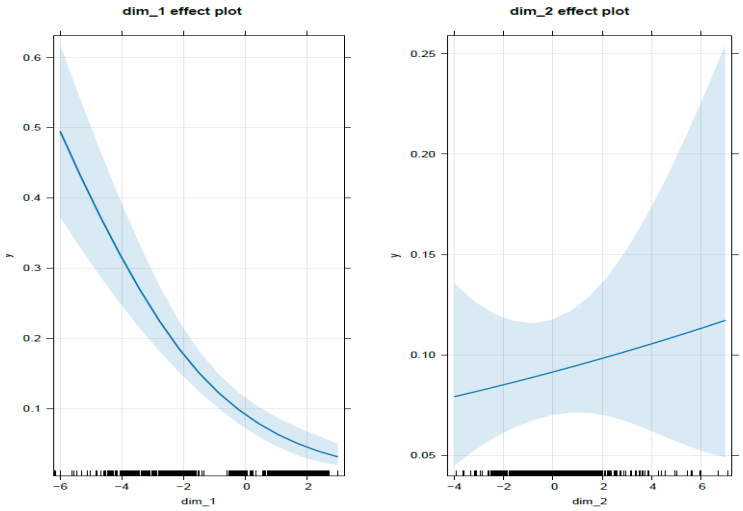

Figure 13: Effect plot of the logistic regression model with the first two principal components.

6 Receiver Operating Characteristic (ROC) Curve Analysis

In the previous section, we built and evaluated a logistic regression model for predicting customer subscription to a bank term deposit. While evaluating the model's performance using metrics like accuracy or error rate is helpful, a more informative approach for classification tasks is often Receiver Operating Characteristic (ROC) curve analysis.

A Receiver Operating Characteristic (ROC) curve is used to study the trade-off between the true positive rate (sensitivity) and false positive rate (1 - specificity) across different threshold values. It helps assess the performance of a binary classifier model. The ROC curves provide a valuable tool to assess the model's ability to discriminate between positive (subscribed) and negative (not subscribed) cases across different classification thresholds.

Figure 14 shows a comparison of two ROC curves for the model with all 18 predictors and the model with the two principal components as the predictor variables in the training data. The plot was done using the R package "pROC" by Robin et al. (2011). Figure 14 (a) has AUC (Area under the curve) of 0.95 and a confidence interval of (0.93, 0.95). Figure 14 (b) also has AUC of 0.74 with a confidence interval of (0.68, 0.74). By comparing both AUC, the full model has better performance. Also, the ROC curve for the full model is closer to the upper left corner than the ROC curve for the two-PCs model. This also indicates that the model with full data performed better. We would expect both models to perform well in the data used to estimate the models.

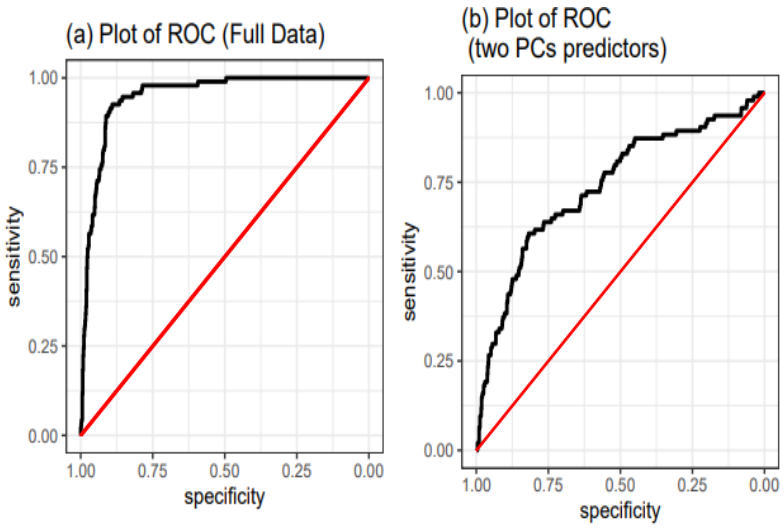

Figure 14: Comparison of ROC Curves: Full Model vs. Two Principal Components in the training data

7 Using the Test Data

We've now built and trained (estimated) both models using the training data. The training data helped us identify patterns and relationships between the features (predictors) and the target variable. However, the real test is how well these models perform on unseen data. This is where the 200 observations in the test data set are used.

Figure 15 shows that the full predictor space logistic model had better predicted probabilities than the predictions from the PC-score logistic model shown in Figure 17. Assuming we define a high probability of success as being greater than 0.5 (> 0.5), the correct classification rate in the test data for the two models are 91.5% for Model 1 and 89.5% for Model 2.

7.1 Model 1: Full Data

We evaluate the performance of our logistic regression model trained that was estimated using all the predictors in the training data by applying that model to doing prediction in the test data. We obtain predicted probabilities for the test data from this model. These predicted probabilities represent the likelihood of each instance belonging to either class 0 or class 1. We then visualize (Figure 15) the predicted probabilities against the true class labels using a stripchart plot. This plot helps us understand how well our model's predictions align with the actual classes in the test data. The x-axis represents the true class labels, while the y-axis represents the predicted probabilities. Points are jittered to avoid overlap, and different colors are used to distinguish between class 0 and class 1.

Figure 15 shows the distribution of predicted probabilities for customer subscription to a bank term deposit, plotted against the actual truth labels (subscribed or not subscribed) for Model 1 (full data). Based on the high AUC

(Figure 16) of 0.92 and narrow confidence interval (0.87, 0.96), we can conclude that Model 1 using all variables exhibits strong performance in differentiating between potential subscribers and non-subscribers. This suggests the model effectively captures the underlying relationships between customer characteristics and their likelihood of subscribing to a term deposit.

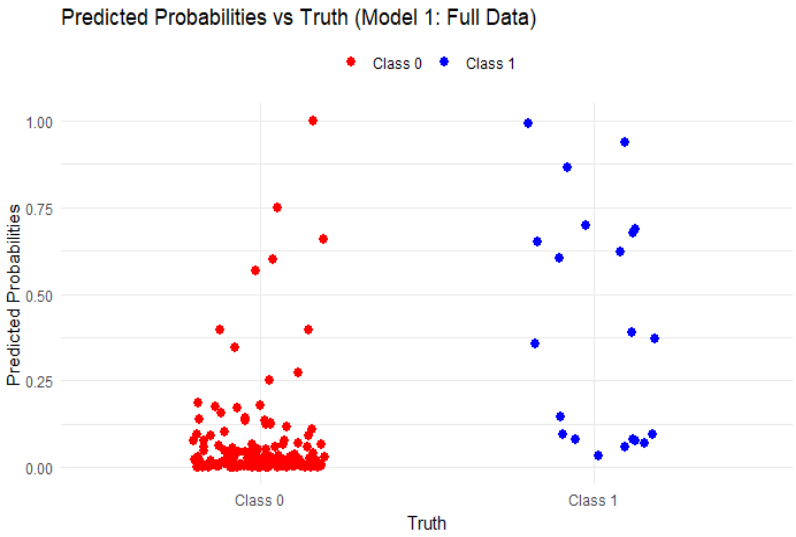

Figure 15: Predicted probabilities vs truth (Model 1: Full Data) in the test data set.

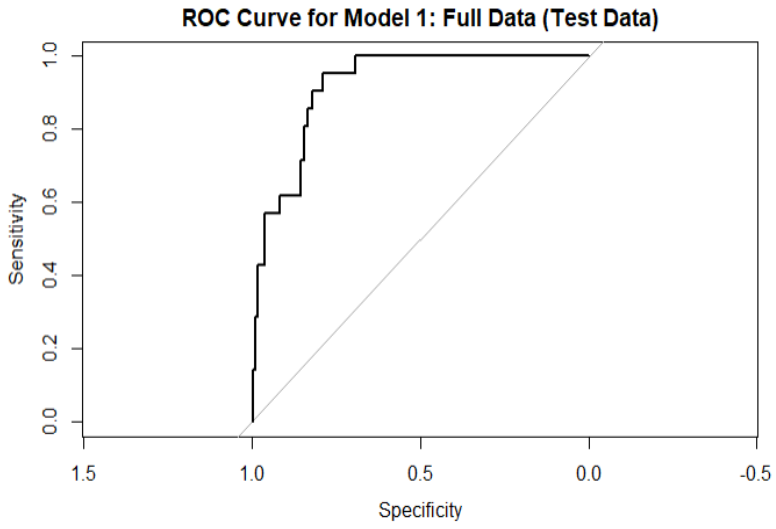

Figure 16: This ROC curve shows the performance of Model 1 (using all variables) in predicting customer subscription to a bank term deposit on the test data.

7.2 Model 2: Model with 2 PCs predictors

Model 2 employs a logistic regression model utilizing two principal components (PCs) as predictors. These principal components are derived from the training data's quantitative and qualitative features using PCAmix. Then predicted scores are generated in the test data using the 'predict.PCAmix' function from PCAmix to create PC scores for the test data set. The logistic model from the training data is used to predict with these new PC scores in the test to assess the potential for this combination of work to predict the probabilities of class membership. The resulting predicted probabilities are then visualized against the true class labels using a stripchart plot (Figure 17). This plot helps in understanding how well the model's predictions align with the actual class labels, providing insights into its performance.

Similarly for Figure 15, Figure 17 also shows the distribution of predicted probabilities for customer subscription to a bank term deposit. The ROC curve for Model 2 demonstrates an Area Under the Curve (AUC) of 0.68 (seen in Figure 18) with a confidence interval from 0.56 to 0.80. Overall, the ROC analysis suggests moderate predictive performance for Model 2 in predicting customer subscription to a bank term deposit on the test data.

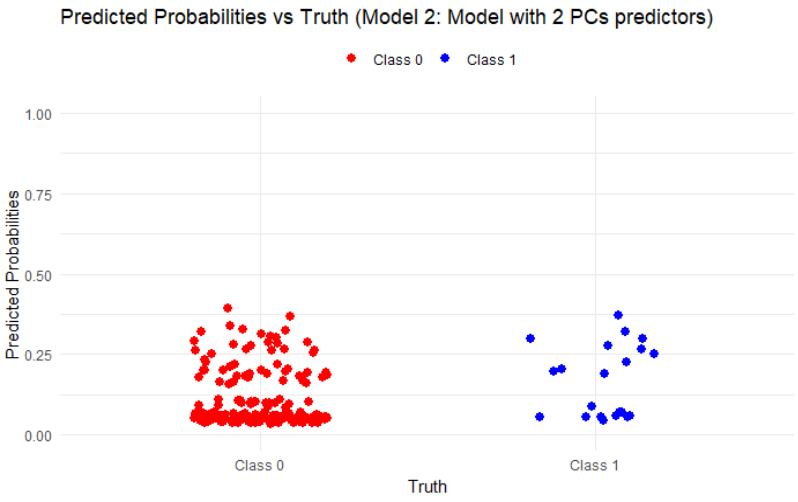

Figure 17: Predicted Probabilities vs Truth (Model 2: Model with 2 PCs predictors) in the test data.

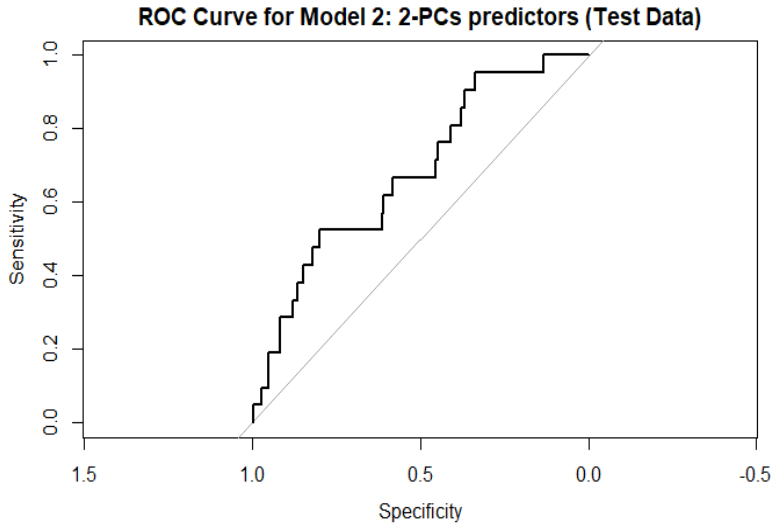

Figure 18: This ROC curve shows the performance of Model 2 (using two principal components) in predicting customer subscription to a bank term deposit on the test data.

Overall, the ROC analysis suggests that Model 1, utilizing all variables, outperforms Model 2, which relies on only two principal components, in predicting customer subscription to a bank term deposit on the test data. Therefore, Model 1 may be preferred for its stronger predictive capability.

8 Conclusion and further studies.

This project investigated the use of logistic regression for predicting customer subscription to a bank term deposit. We explored two approaches or models. Model 1 is a logistic regression model using all predictors (both quantitative and qualitative). Model 2 is a logistic regression model using principal components derived from PCAmix to reduce the dimensionality of the dataset introduced in Chapter 2.

Our analysis revealed some key findings. Model 1 achieved better predictive performance as evidenced by ROC curves and stripchart plots and prediction error rates in the test data set. However, this model also suffered from extremely high complexity due to the large number of features, making interpretation challenging. While it explained a substantial portion of the variation in customer subscription (63.84%), the complexity made it difficult to understand the underlying variables influencing these predictions. Model 2 offered a more interpretable solution by utilizing principal components. While its performance was lower, it demonstrates the potential benefits of dimensionality reduction for model interpretability. It explained a smaller proportion of the variation (17.67%), but the use of principal components provided valuable insights into the key drivers (variables) of customer subscription. In our Model 2 analysis, we found some variables contributing substantial to the two principal components. There were: "nr.employed (Number of Employees)", "emp.var.rate (Employment Variation Rate)", "euribor3m (Euro Interbank Offered Rate, 3-month)", "age" and "job". These variables influences a customer to subscribe to a bank term deposit or not.

For further explorations, we suggest a variety of extensions to this work. We did not consider identifying an optimal number of principal components from PCAmix or using model refinement techniques to decide how many PCs are optimal in the logistic regression model. By exploring the number of principal components

to use in Model 2, we might be better able to find the optimal balance between dimensionality reduction and information retention.

Similarly, we did not engage in model refinement or variable selection in Model 1. By removing un-needed predictors, this could improve the predictive performance in the test data set.

For both approaches, all quantitative predictors were assuming to be linearly related to the response on the link (log-odds) scale and for all variables, we did not explore potential interactions between variables. This is a strong assumption that was made to avoid the high complexity that could have been engaged with 18 predictors. For the PC-score approach, this could be more manageable. In further applications, these modifications could be explored.

Finally, logistic regression is a somewhat restrictive modeling framework and contains assumptions about the relationship between the predictors and the response. It provides a model that can be written out but if the interest is purely in developing a predictive model, approaches such as random forest, gradient boosting machine, neural networks, and support vector machines could be considered. These could be compared to the approaches discussed in the work of Moro et al. (2014).

9 Reference

References

Autio, R., Virta, J., Nordhausen, K., Fogelholm, M., Erkkola, M., and Nevalainen, J. (2023). Tensorial principal component analysis in detecting temporal trajectories of purchase patterns in loyalty card data: Retrospective cohort study. *Journal of Medical Internet Research*, 25:e44599.

Bahari, T. F. and Elayidom, M. S. (2015). An efficient crm-data mining framework for the prediction of customer behaviour. *Procedia computer science*, 46:725–731.

Bartoń, K. (2023). *MuMIn: Multi-Model Inference*. R package version 1.47.5.

Chavent, M., Kuentz, V., Labenne, A., Liquet, B., and Saracco, J. (2017). *PCAmixdata: Multivariate Analysis of Mixed Data*. R package version 3.1.

Chavent, M., Kuentz, V., Liquet, B., and Saracco, J. (2011). Clustering of variables via the pcamix method. In *Joint Conference of the German Classification Society (GfKl), the German Association for Pattern Recognition (DAGM) and the IFCS 2011 Symposium of the International Federation of Classification Societies (IFCS)*, page 1.

Chen, Y.-C. (2018). Statistical inference with local optima. ArXiv preprint arXiv:1807.04431; In submission to Annals of Statistics.

Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977). Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society, Series B*, 485(1):1–38.

Fasy, B. T., Kim, J., Lecci, F., Maria, C., Millman, D. L., and Rouvreau, V. (n.d.).

Tda: Statistical tools for topological data analysis. `https://CRAN.R-project.org/package=TDA`. Accessed: 2019-5-8.

Fasy, B. T. and Wang, B. (2016). Exploring persistent local homology in topological data analysis. In *ICASSP Workshop on Topological Methods in Data Science*.

Fox, J. and Weisberg, S. (2019). *An R Companion to Applied Regression*. Sage, Thousand Oaks CA, third edition.

Gelman, A. and Hill, J. (2011). Opening windows to the black box. *Journal of Statistical Software*, 40.

GrindGIS (n.d.). LIDAR data 50 applications and uses – it is important. `https://grindgis.com/data/lidar-data-50-applications`. Accessed: 2018-11-13.

Grzonka, D., Suchacka, G., and Borowik, B. (2016). Application of selected supervised classification methods to bank marketing campaign. *Information Systems in Management*, 5(1):36–48.

Koneswarakantha, B. (2023). *easyalluvial: Generate Alluvial Plots with a Single Line of Code*. R package version 0.3.2.

Kurita, T. (2019). Principal component analysis (pca). *Computer Vision: A Reference Guide*, pages 1–4.

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

Munroe, R. (n.d.). Correlation. `https://xkcd.com/552/`. Accessed: 2020-2-6.

Parlar, T. and Acaravci, S. K. (2017). Using data mining techniques for detecting the important features of the bank direct marketing data. *International journal of economics and financial issues*, 7(2):692–696.

Pearson, K. (1901). Liii. on lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin philosophical magazine and journal of science*, 2(11):559–572.

R Core Team (2023). *R: A Language and Environment for Statistical Computing*. R Foundation for Statistical Computing, Vienna, Austria.

Robin, X., Turck, N., Hainard, A., Tiberti, N., Lisacek, F., Sanchez, J.-C., and Müller, M. (2011). proc: an open-source package for r and s+ to analyze and compare roc curves. *BMC Bioinformatics*, 12:77.

Su, C.-T., Chen, Y.-H., and Sha, D. (2006). Linking innovative product development with customer knowledge: a data-mining approach. *Technovation*, 26(7):784–795.

Wasserman, L. (2004). *All of Statistics: A Concise Course in Statistical Inference*. Springer, New York, NY.

10 Appendix 1: R-code

```
knitr::opts_chunk$set(echo = TRUE)
```

```
library(ggplot2)
library(dplyr)
library(tidyverse)
library(GGally)
library(ggcorrplot)
library(pander)
library(MVA)
library(MASS)
library(pheatmap)
library(factoextra)
library(arm)
library(caret)
library(gridExtra)
library(tree)
library(ISLR)
library(pander)
library(olsrr)
library(ggResidpanel)
library(PCAmixdata)
library(FactoMineR)
library(mi)
library(effects)
library(car)
```

```
library(easyalluvial)
#remotes::install_github("greenwood-stat/catstats2")
library(catstats2)
library(parcats)
library(PCAmixdata)
library(pROC)
library(GGally)
library(ggthemes)
library(MuMIn)
library(gtsummary)
```

2 Data

2.2 Data Cleaning

```
# Getting the data in R workspace and converting unknown to NA
d <- read.table("bank-additional-full.csv", header=TRUE, sep=";",
na.strings = "unknown")

# Removing pdays variable from the dataset
d <- d[, !colnames(d) %in% "pdays"]

# Randomly getting 1000 observation
total_obs <- nrow(d)
set.seed(123)
random_indices <- sample(1:total_obs, 1000)
```

```
dat <- d[random_indices, ]
head(dat)

# Checking or counting missing values in each variable
na_counts <- numeric(length(colnames(dat)))
for (i in 1:length(colnames(dat))) {
  na_counts[i] <- sum(is.na(dat[, i]))
}
for (i in 1:length(colnames(dat))) {
  cat("Variable:", colnames(dat)[i], "\tCount of NA:", na_counts[i], "\n")
}

# Visualizing the misisng values in each variables
library(mi)
dat %>% as.data.frame() %>% missing_data.frame() %>% image()

# Handling the missing values
Variables with missing values: job, marital, education, default, housing and loan.
We handle these values by using completion imputation techniques.
This means that the missing values have been replaced by the mode of the
respective variable within their class.

## Checking the mode of the covariates with missing values
calculate_mode <- function(x) {
  freq_table <- table(x)
  mode_value <- names(sort(freq_table, decreasing = TRUE))[1]
  return(mode_value)
```

```
}
variables <- c("job", "marital", "education", "default", "housing", "loan")
modes <- sapply(dat[variables], calculate_mode)
print(modes)
pander(modes)

**The provided code replaces NA values with the mode for each categorical variable
in the dataset "dat" and then counts the number of NA values after the replacement.**

variables <- c("job", "marital", "education", "default", "housing", "loan")
for (var in variables) {
  mode_value <- names(sort(table(dat[[var]], useNA = "ifany"), decreasing = TRUE))[1]
  dat[[var]][is.na(dat[[var]])] <- mode_value
  na_count <- sum(is.na(dat[[var]]))
  cat("Variable:", var, "\tCount of 'NA':", na_count, "\n")
}

# Checking (again) all variables with unknown
unknown_counts <- numeric(length(colnames(dat)))
for (i in 1:length(colnames(dat))) {
  unknown_counts[i] <- sum(dat[,i] == "unknown", na.rm = TRUE)
}
for (i in 1:length(colnames(dat))) {
  cat("Variable:", colnames(dat)[i], "\tCount of 'unknown':", unknown_counts[i], "\n")
}
```

```
# Count unique value for each variable
sapply(lapply(dat, unique), length)

# Remove the "default" variable from the dat dataset
To prevent this error;
"Error in ‘contrasts<-‘(‘*tmp*‘, value = contr.funs[1 + isOF[nn]]) :
contrasts can be applied only to factors with 2 or more levels"
dat <- dat[, !colnames(dat) %in% "default"]

# Factor the categorical variables
dat$job <- factor(dat$job)
dat$marital <- factor(dat$marital)
dat$education <- factor(dat$education)
dat$housing <- factor(dat$housing)
dat$loan <- factor(dat$loan)
dat$contact <- factor(dat$contact)
dat$month <- factor(dat$month)
dat$day_of_week <- factor(dat$day_of_week)
dat$poutcome <- factor(dat$poutcome)
dat$y <- factor(dat$y, levels = c("no", "yes"), labels = c(0, 1))
str(dat)
```

```
2.3 Data Splitting

# Splitting data into train and test
training_data <- slice_sample(dat, prop = 0.8)
test_data <- anti_join(dat, training_data)

2.4 Exploratory Data Analysis
# Qualitative variables vs response variable (under training data)
alluvial_wide(training_data[, c("job", "marital", "education", "housing", "loan",
"contact", "month", "day_of_week", "poutcome", "y")],
              fill_by = "last_variable",
              bin_labels = "mean")

# Quantative variables vs response variable (under training data)
alluvial_wide(training_data[, c("age", "duration", "campaign", "previous",
"emp.var.rate", "cons.price.idx", "cons.conf.idx", "euribor3m", "nr.employed", "y")],
              fill_by = "last_variable",
              bin_labels = "mean")


3 Principal Component Analysis (PCA)
# Standard PCA
quant_training_data <- training_data[c("age", "duration", "campaign", "previous",
"emp.var.rate", "cons.price.idx", "cons.conf.idx", "euribor3m", "nr.employed")]
stan_pca <- prcomp(quant_training_data, scale.=T)
stan_pca$rotation[, 1:9]
pander(round(stan_pca$rotation[, 1:9], 2))
```

```
Screeplot diagram
plot(stan_pca$sdev^2, type="b", xlab="Component Number",
ylab="Eigenvalue (Variance)", main="Screeplot diagram")

Biplot
biplot(stan_pca, col = c("gray", "black"), main="Biplot (PC1 vs PC2)")

4 Principal Component Analysis of Mixed Data (PCAmix)
4.2 Principal Components

# PCAmix on training data
split_td <- splitmix(training_data%>% dplyr::select(-y))
pcamix_td <- PCAmix(X.quanti=split_td$X.quanti,
                    X.quali=split_td$X.quali,
                    rename.level=TRUE,
                    graph=FALSE, ndim=45)
summary(pcamix_td)

4.3 PCAmix Analysis Plots
PCAmix Scores Scatterplot, colored by the response variable.

# New dataframe with y from the training dataset and the score_td dataframe
combined_td_df <- cbind(scores_td_df, y = training_data$y)
color_map <- c("0" = "#FF0000", "1" = "#0000FF")
ggplot(combined_td_df, aes(x = dim_1, y = dim_2, color = factor(y))) +
  geom_point() +
```

```
  scale_color_manual(values = color_map) +  # Apply custom color mapping
  labs(x = "Dimension 1", y = "Dimension 2", color = "Response (y)",
       title = "PCAmix Scores Scatterplot, colored by the response variable.") +
  theme_bw()

# Loadings of the numerical variables
plot(pcamix_td, choice = "cor", main = "Numerical Variables")

# Scores of the levels of the categorical variables
plot(pcamix_td, choice="levels")

# Contributions of all variables
plot(pcamix_td, choice="sqload", coloring.var="type", main = "All variables")

# An alluvial plot of the variables with high contribution against the first
principal component for PCAmix on the training data.
combined_new <- cbind(scores_td_df, training_data)
alluvial_wide(combined_new[, c("contact","previous","poutcome","month",
"cons.price.idx","nr.employed","emp.var.rate","euribor3m", "dim_1")],
              fill_by = "last_variable",
              bin_labels = "mean")

# An alluvial plot of the variables against the second principal component
for PCAmix on the training data.
alluvial_wide(combined_new[, c("cons.conf.idx","marital","month","job","age","dim_2")],
              fill_by = "last_variable",
              bin_labels = "mean")
```

5 Logistic Regression

5.1 Logistic regression model using the full data

```
logistic_model <- glm(y ~., data = training_data, family = binomial)
summary(logistic_model)
plot(allEffects(logistic_model), type="response", grid = T)
r.squaredGLMM(logistic_model)
vif(logistic_model)
r.squaredGLMM(logistic_model)
```

5.2 Logistic regression model using the first two principal components

```
logistic_model_pcamix_td <- glm(y ~ dim_1 + dim_2,
                                data =combined_td_df,
                                family = binomial)
summary(logistic_model_pcamix_td)
# R-squared for the logistic model with two principal components
r.squaredGLMM(logistic_model_pcamix_td)
# Effects plot
plot(allEffects(logistic_model_pcamix_td), type="response",
     ylab = "Estimated probability of subscribing",
     grid = T)
```

```
6. Receiver Operating Characteristic (ROC) Curve Analysis
library(gridExtra)
roc_td <- roc(combined_td_df$y, logistic_model_pcamix_td$fitted.values, ci = TRUE)
roc_initial_model <- roc(training_data$y, logistic_model$fitted.values, ci = TRUE)
roc_plot_initial_with_text <- (
  ggroc(roc_initial_model, linewidth = 1) +
  ggtitle("(a) Plot of ROC (Full Data)") +
  geom_segment(aes(x = 1, xend = 0, y = 0, yend = 1), col = "red", alpha = 0.5) +
  theme_bw() +
  coord_fixed()
)
roc_plot_with_text <- (
  ggroc(roc_td, linewidth = 1) +
  ggtitle("(b) Plot of ROC \n (two PCs predictors)") +
  geom_segment(aes(x = 1, xend = 0, y = 0, yend = 1), col = "red", alpha = 0.5) +
  theme_bw() +
  coord_fixed()
)
final_plot <- grid.arrange(roc_plot_initial_with_text, roc_plot_with_text, ncol = 2)
final_plot



7. Using the Test Data
# Calculating the prediction rate
# For Model 1: Full Data
predicted_probabilities_full <- predict(logistic_model, newdata = test_data,
type = "response")
```

```
binary_prediction_full <- ifelse(predicted_probabilities_full >= threshold, 1, 0)
confusion_matrix_full <- confusionMatrix(factor(binary_prediction_full),
factor(test_data$y))
accuracy_full <- confusion_matrix_full$overall['Accuracy']

# For Model 2: Model with 2 PCs predictors
split_td <- splitmix(training_data%>% dplyr::select(-y))
pcamix_td <- PCAmix(X.quanti=split_td$X.quanti,
                X.quali=split_td$X.quali,
                rename.level=TRUE,
                graph=FALSE, ndim=45)
split_test <- splitmix(test_data %>% select(-y))
pred <- predict(pcamix_td, split_test$X.quanti, split_test$X.quali)
pred_test_df <- as.data.frame(pred)
names(pred_test_df) <- c("dim_1", "dim_2")
prediction <- predict(logistic_model_pcamix_td, newdata = pred_test_df,
type = "response")
threshold <- 0.5
binary_prediction <- ifelse(prediction >= threshold, 1, 0)
confusion_matrix_pca <- confusionMatrix(factor(binary_prediction),
factor(test_data$y))
accuracy_pca <- confusion_matrix_pca$overall['Accuracy']
```

7.1 Model 1: Full Data

```
# Predicted probabilities vs truth (Model 1: Full Data) in the test data set.
logistic_model <- glm(y ~ ., data = training_data, family = binomial)
```

```
predicted_probabilities_full <- predict(logistic_model, newdata = test_data,
type = "response")
predicted_probabilities_full_class0 <- predicted_probabilities_full[test_data$y == 0]
predicted_probabilities_full_class1 <- predicted_probabilities_full[test_data$y == 1]
predicted_data_full <- data.frame(Predicted_Probability =
c(predicted_probabilities_full_class0, predicted_probabilities_full_class1),
Truth = rep(c("Class 0", "Class 1"),
c(length(predicted_probabilities_full_class0),
length(predicted_probabilities_full_class1))))

ggplot(predicted_data_full, aes(x = Truth, y = Predicted_Probability, color = Truth))+
  geom_jitter(width = 0.2, height = 0, size = 2.5) +
  labs(title = "Predicted Probabilities vs Truth (Model 1: Full Data)",
       x = "Truth",
       y = "Predicted Probabilities") +
  scale_y_continuous(limits = c(0, 1)) +
  theme_minimal() +
  theme(legend.position = "top") +
  guides(color = guide_legend(title = NULL)) +
  scale_color_manual(values = c("Class 0" = "red", "Class 1" = "blue"))

# ROC Curve for Model 1: Full Data (Test Data)
library(pROC)
predicted_data_full <- data.frame(Predicted_Probability =
c(predicted_probabilities_full_class0, predicted_probabilities_full_class1),
Truth = rep(c("Class 0", "Class 1"), c(length(predicted_probabilities_full_class0),
length(predicted_probabilities_full_class1))))
```

```
plot(roc_data_full, main = "ROC Curve for Model 1: Full Data (Test Data)")
```

7.2 Model 2: Model with 2 PCs predictors

```
# Predicted Probabilities vs Truth (Model 2: Model with 2 PCs predictors)
in the test data.
split_td <- splitmix(training_data%>% dplyr::select(-y))
pcamix_td <- PCAmix(X.quanti=split_td$X.quanti,
                X.quali=split_td$X.quali,
                rename.level=TRUE,
                graph=FALSE, ndim=45)
split_test <- splitmix(test_data %>% select(-y))
pred <- predict(pcamix_td, split_test$X.quanti, split_test$X.quali)
pred_test_df <- as.data.frame(pred)
names(pred_test_df) <- c("dim_1", "dim_2")

# ROC Curve for Model 2: 2-PCs predictors (Test Data)
prediction <- predict(logistic_model_pcamix_td, newdata = pred_test_df,
type = "response")
roc_data <- roc(predictor = prediction, response = test_data$y)
plot(roc_data, main = "ROC Curve for Model 2: 2-PCs predictors (Test Data)")
 ci_values <- ci.auc(roc_data)
 auc_value <- auc(roc_data)
 legend("bottomright",
        legend = paste("AUC =", round(auc_value, 2), " (", round(ci_values[1], 2),
        "-", round(ci_values[3], 2), ")", sep = ""),
        col = "black", lty = 1)
```