Spam classification

Statistical Learning project

Sofia Bazzan, Diletta Pasin

Università degli Studi di Padova

Dipartimento Matematica

Aim of the study

GOAL

The goal is to detect if an email is spam or not.

DATA

The dataset was obtained from *Kaggle.com*. It's made of 4601 observations described by 57 features.

FEATURE'S TYPES

Variables can be divided in the following groups:

- **Word frequency**: There are 48 **continuous real attributes** between 0 and 100 of the type word_freq_WORD that represents the percentage of words in the e-mail that match WORD.
- **Char frequency**: 6 of the attributes are frequency of characters in the mail (char_freq_.3B (;), char_freq_.28 (), char_freq_.5B ([), char_freq_.21 (!), char_freq_.24 ($), char_freq_.23 (#))
- **capital_run_length_average**: 1 continuous real attribute of type, which denotes the average length of uninterrupted sequences of capital letters.
- **capital_run_length_longest**: 1 continuous real attribute of type, that is equal to the length of longest uninterrupted sequence of capital letters.
- **capital_run_length_total**: 1 continuous real attribute that indicate the total number of capital letters in the e-mail.
- **class**: 1 nominal 0,1 class attribute of type spam that denotes whether the e-mail was considered spam (1) or not (0).

Preprocessing of data

1. Study of missing values and duplicates
2. Study of inconsistencies
3. Dataset scaling
4. Study of correlations

1 Missing values and duplicates

First we checked the presence of NA values in our dataset and our research gave 0 missing values.

Moving on, we also need to determine if there are any duplicate entries. We found some duplicates, so we identified the number of duplicated rows, which was 391 and we removed them utilizing the *unique* command, which retains only the distinct rows of the dataframe.

Study of inconsistencies 2

To see if there were inconsistencies in our dataset we checked for rows where the average frequency of a word is larger than its longest frequency, or where any word's frequency surpasses the maximum value of 100

```
result <- df[df$word_freq_average > df$word_freq_longest, ]

if (nrow(result) > 0) {
  print("There are lines where word_freq_average is longer than word_freq_longest.")
} else {
  print("There are no lines where word_freq_average is longer than word_freq_longest.")
}
```

```
## [1] "There are no lines where word_freq_average is longer than word_freq_longest."
```

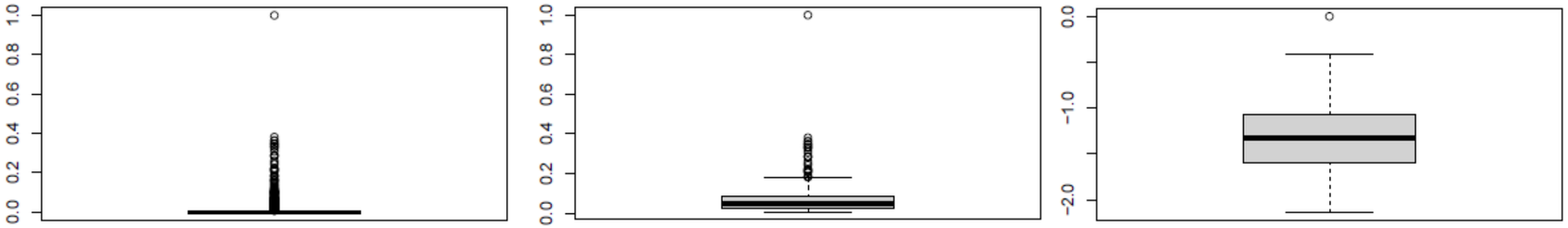
```
columns_to_exclude <- c("capital_run_length_average", "capital_run_length_longest",
                        "capital_run_length_total","class")

all_column_names <- colnames(df)
columns_to_check <-all_column_names[!all_column_names %in% columns_to_exclude]

all_column_names <- colnames(df)
columns_to_check <- all_column_names[!all_column_names %in% columns_to_exclude]

has_greater_than_100 <- any(df[,columns_to_check] > 100)
has_greater_than_100
```

```
## [1] FALSE
```

3 Dataset scaling

Since the variables representing frequencies range between 0 and 100 while others do not, we transformed all variables of the dataset in a common scale between 0 and 1

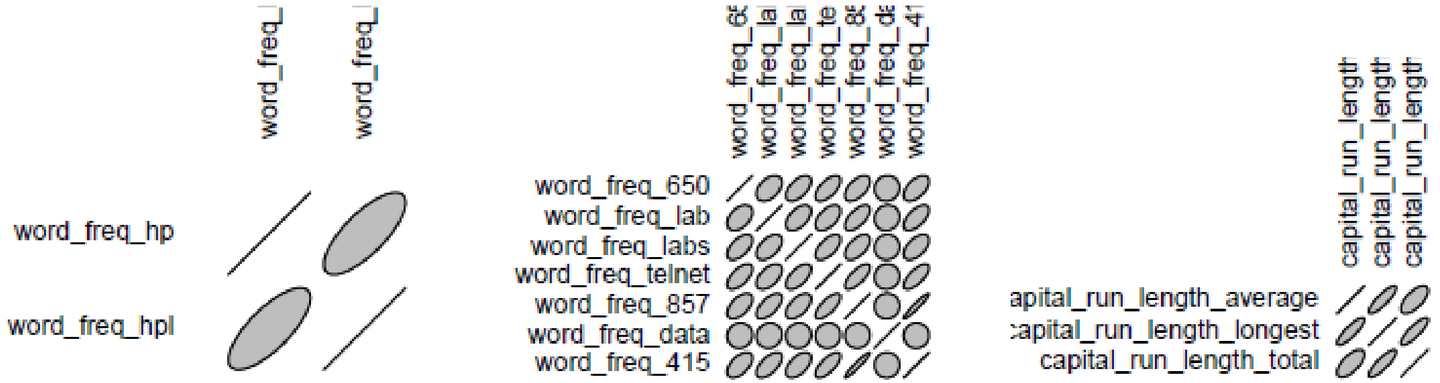
We also applied a logarithmic transformation to spread out the data, which was useful in our case as could be seen in the following boxplots



Study of correlation 4

To see if there were highly correlated variables we construct the correlation matrix and after setting a threshold of 0.6 we found 3 groups of variables with a correlation over the threshold:

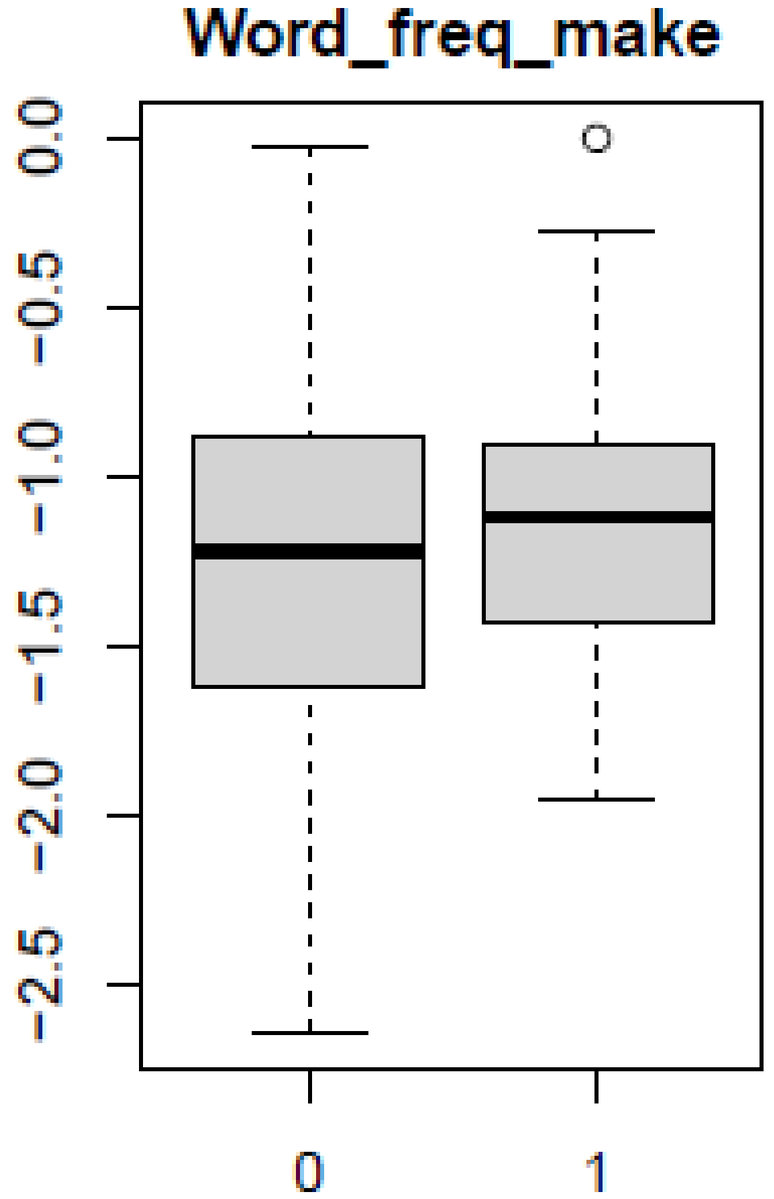
- word_freq_hpl and word_freq_hp
- word_freq_telnet, word_freq_85, word_freq_650, word_freq_labs, word_freq_857, word_freq_415, word_freq_lab
- capital_run_length_longest, capital_run_length_total, capital_run_length_average

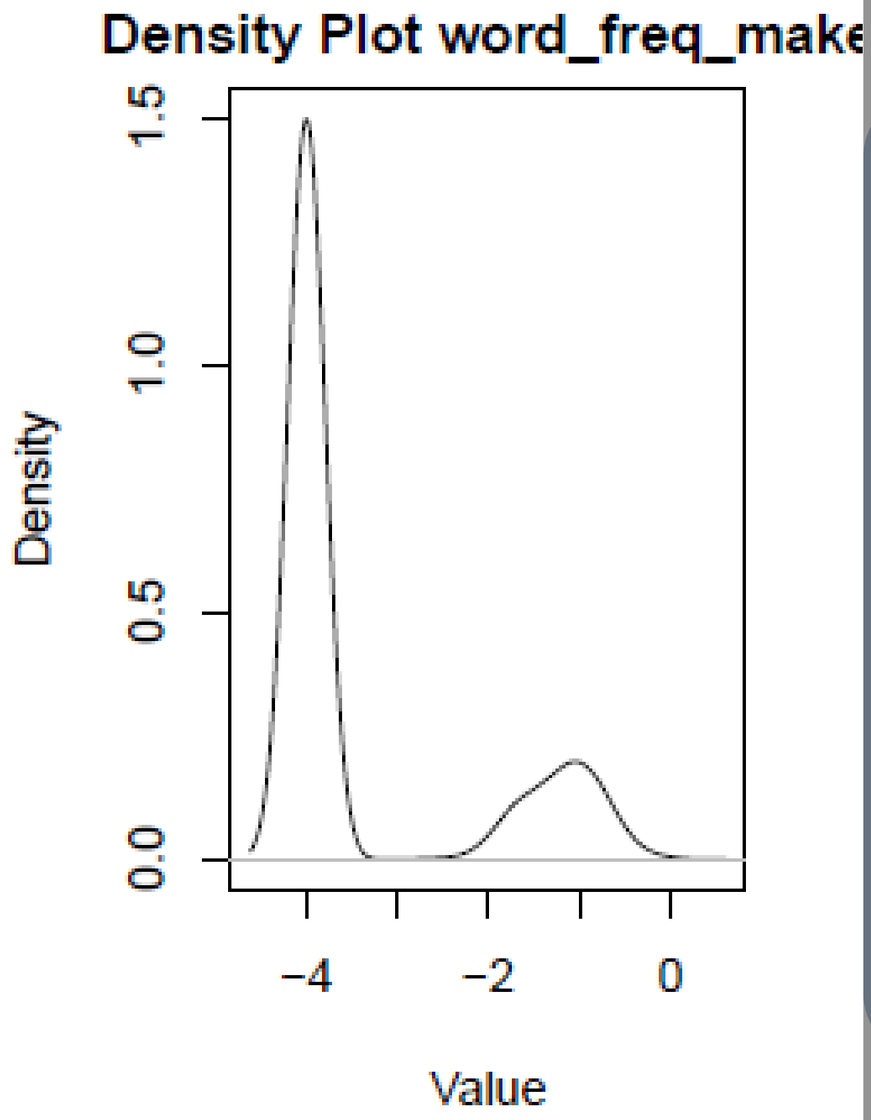



Data visualization

Boxplots

First we wanted to show the boxplot of the data divided by class. In this plot is possible to see the variable *word_freq_make* for the value greater than -4 which rapresents frequencies greater than 0 after the logarithmic transformation.



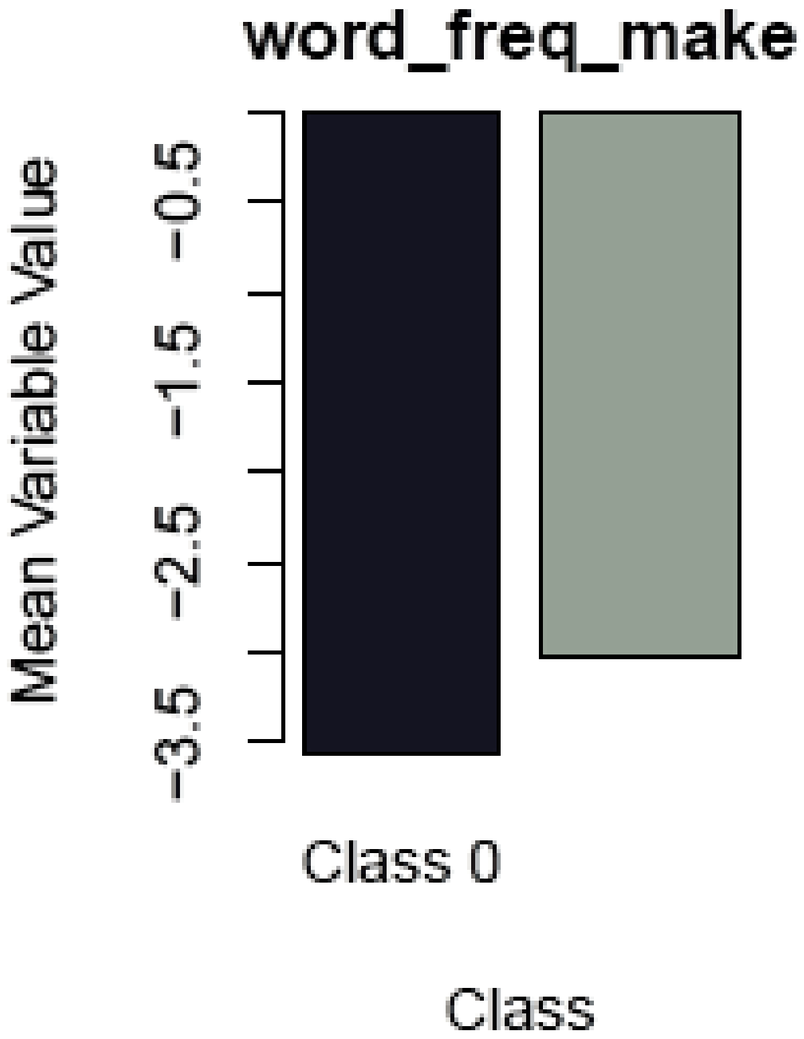


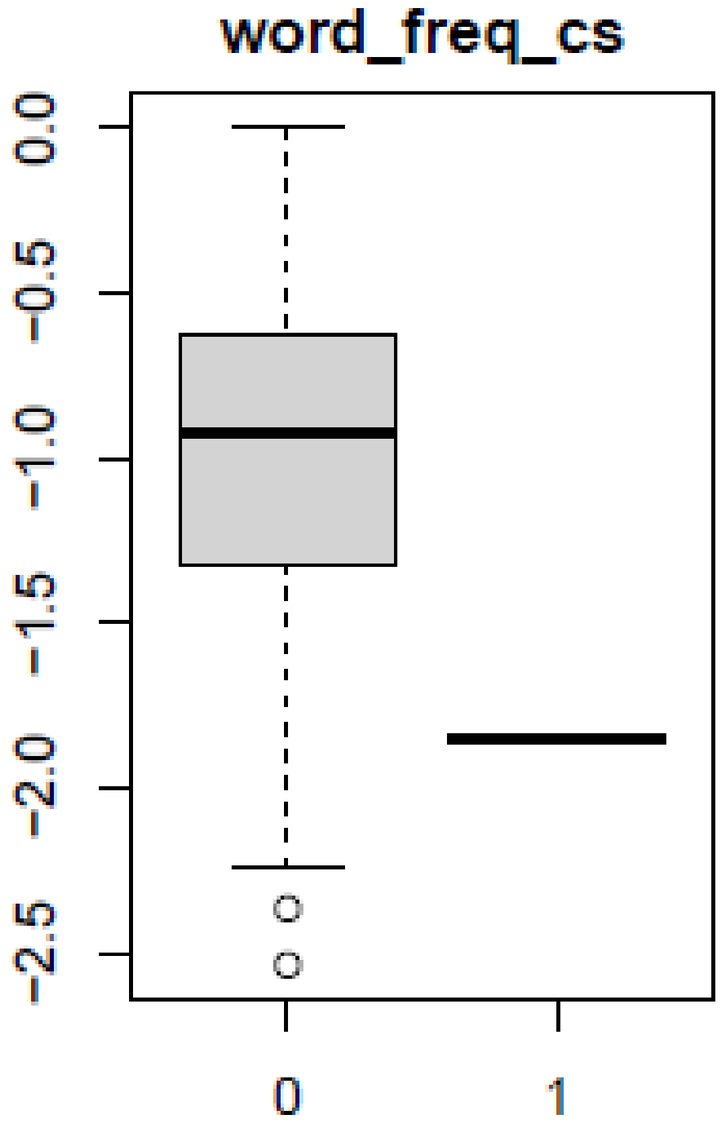
Density plot

In order to see if our data were close to a normal distribution we also decided to create density plot for some of our variables. Is possible to see that the data, as expected, didn’t follow a normal distribution because of the high number of zero values.

Barplots

In addition to the techniques we've explored so far, we've also chosen to utilize bar plots for our data visualization. Bar plots offer a clear and concise way to compare different categories or groups. We'll be using them to make straightforward comparisons that help us better understand the distribution and relationships within our data.



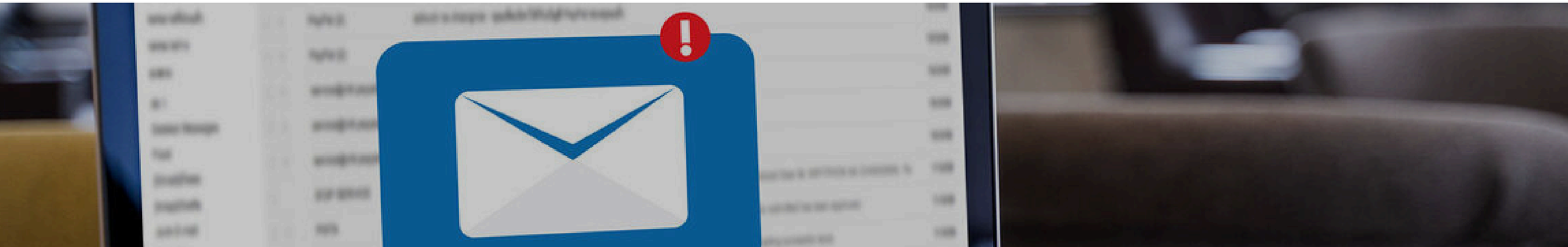


Observations

After looking at the plots of different variables we notice something really important: one of the variables, *word_freq_cs*, presented a rank deficiency that can be clearly seen from the box plot.

Models

- **Logistic model**
- **Ridge regression**
- **Lasso regression**
- **Linear Discriminant Analysis (LDA)**
- **Quadratic Discriminant Analysis (QDA)**
- **Naive Bayes**
- **KNN**



COMPLETE LOGISTIC MODEL

The logistic regression model works to estimate the probability of an e-mail to be spam based on predictor variables, utilizing the **logistic function** (also known as the sigmoid function), which maps the linear combination of predictors to a probability range from 0 to 1.

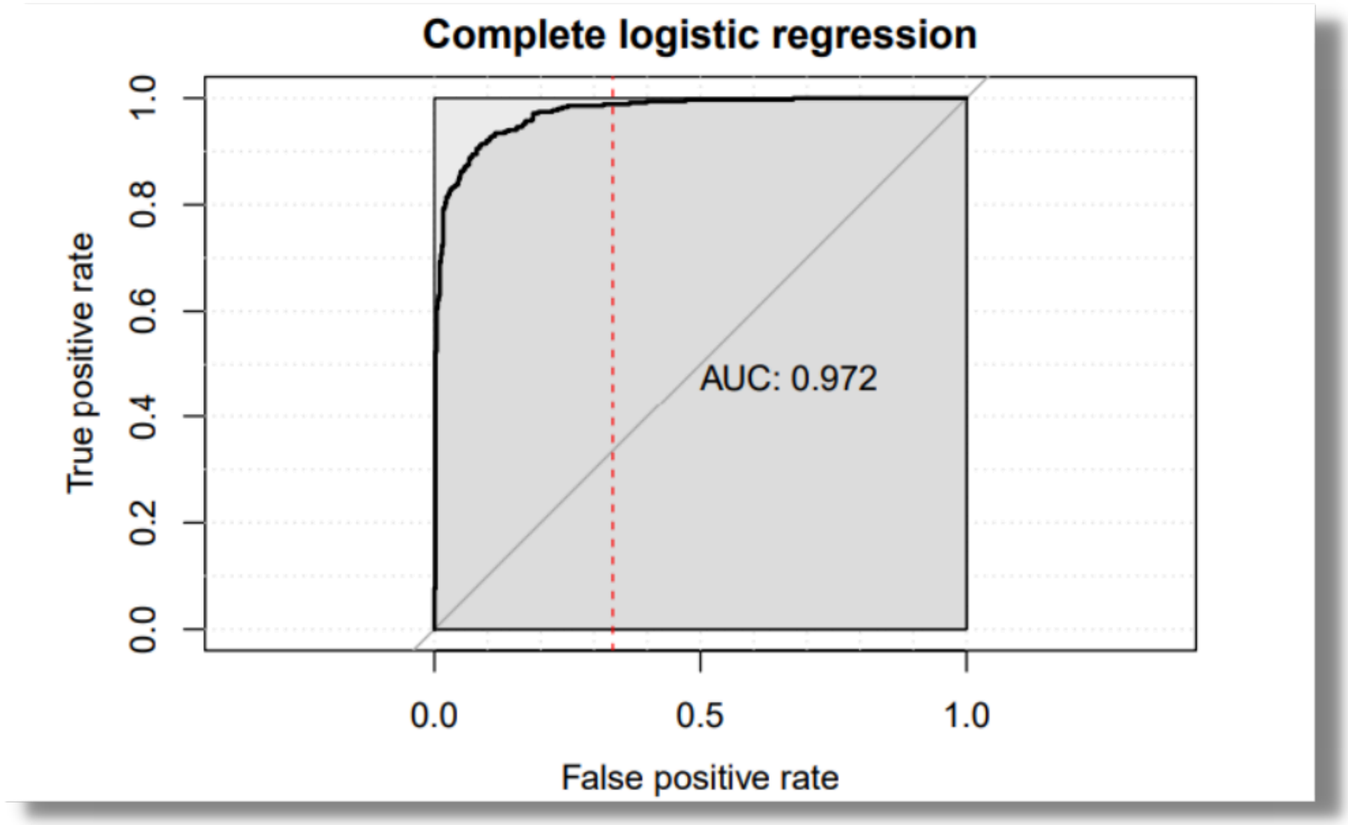
word_freq_make	word_freq_address
1.195232	1.526227
word_freq_all	word_freq_3d
1.444433	1.082468
word_freq_our	word_freq_over
1.319158	1.250389
word_freq_remove	word_freq_internet
1.196218	1.361647
word_freq_order	word_freq_mail
1.190788	1.387656
word_freq_receive	word_freq_will
1.377975	1.368008
word_freq_people	word_freq_report
1.454683	1.132521
word_freq_addresses	word_freq_free
1.090451	1.220686
word_freq_business	word_freq_email
1.420143	1.420461
word_freq_you	word_freq_credit
1.494753	1.107448
word_freq_your	word_freq_font
1.569138	2.124416
word_freq_000	word_freq_money
1.385355	1.207466
word_freq_hpl	word_freq_george
1.101441	1.354861
word_freq_data	word_freq_415
1.095592	1.136955
word_freq_85	word_freq_technology
1.076417	1.382367

word_freq_1999	word_freq_parts
1.320319	1.118492
word_freq_pm	word_freq_direct
1.094253	1.128928
word_freq_meeting	word_freq_original
1.148886	1.074648
word_freq_project	word_freq_re
1.097739	1.184117
word_freq_edu	word_freq_table
1.074002	1.071803
word_freq_conference	char_freq_.3B
1.127301	1.392345
char_freq_.28	char_freq_.5B
1.307599	1.151098
char_freq_.21	char_freq_.24
1.171699	1.426844
char_freq_.23	capital_run_length_total
1.988878	2.545399

BASED ON THE VIF, ALL PREDICTOR VARIABLES ARE STATISTICALLY SIGNIFICANT.

ROC CURVE

The **ROC** (Receiver Operating Characteristic) curve visually illustrates the effectiveness of a binary classification model. It graphically depicts the relationship between the true positive rate (sensitivity) and the false positive rate (1 - specificity) across different threshold settings.

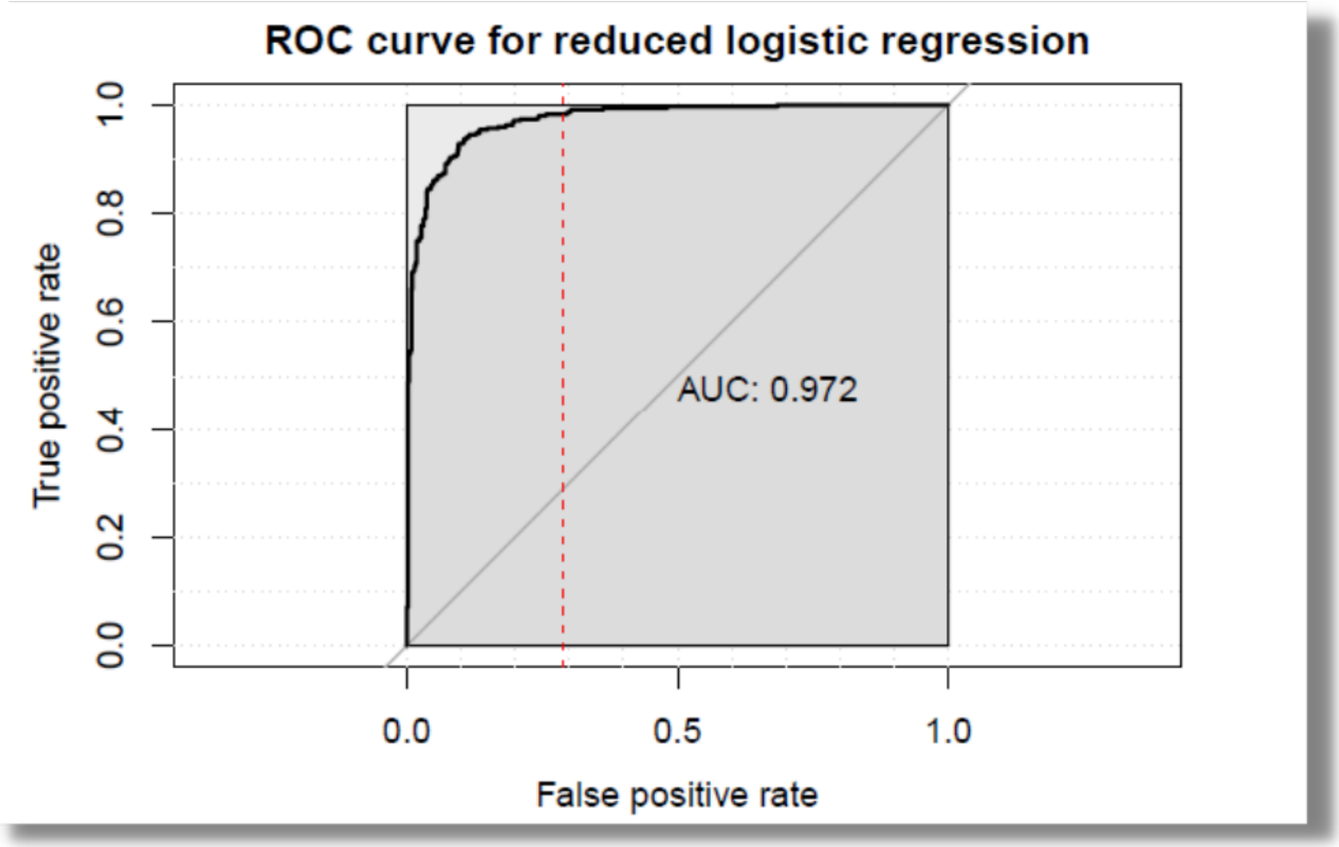


PRECISION:

97,5 %

REDUCED MODEL

Since we have a very high number of predictors, we decided to implement a reduced logistic regression model obtained by removing from the complete model the variables with a high *p-value* (> 0.1): these are in fact the ones that are less relevant in our analysis. So removed from the model 23 predictors.



PRECISION:

96,3 %

BACKWARD FEATURE SELECTION

Studying the AIC, we found out that removing one of the predictors the AIC of the new model increases, so we don’t do backwards elimination.

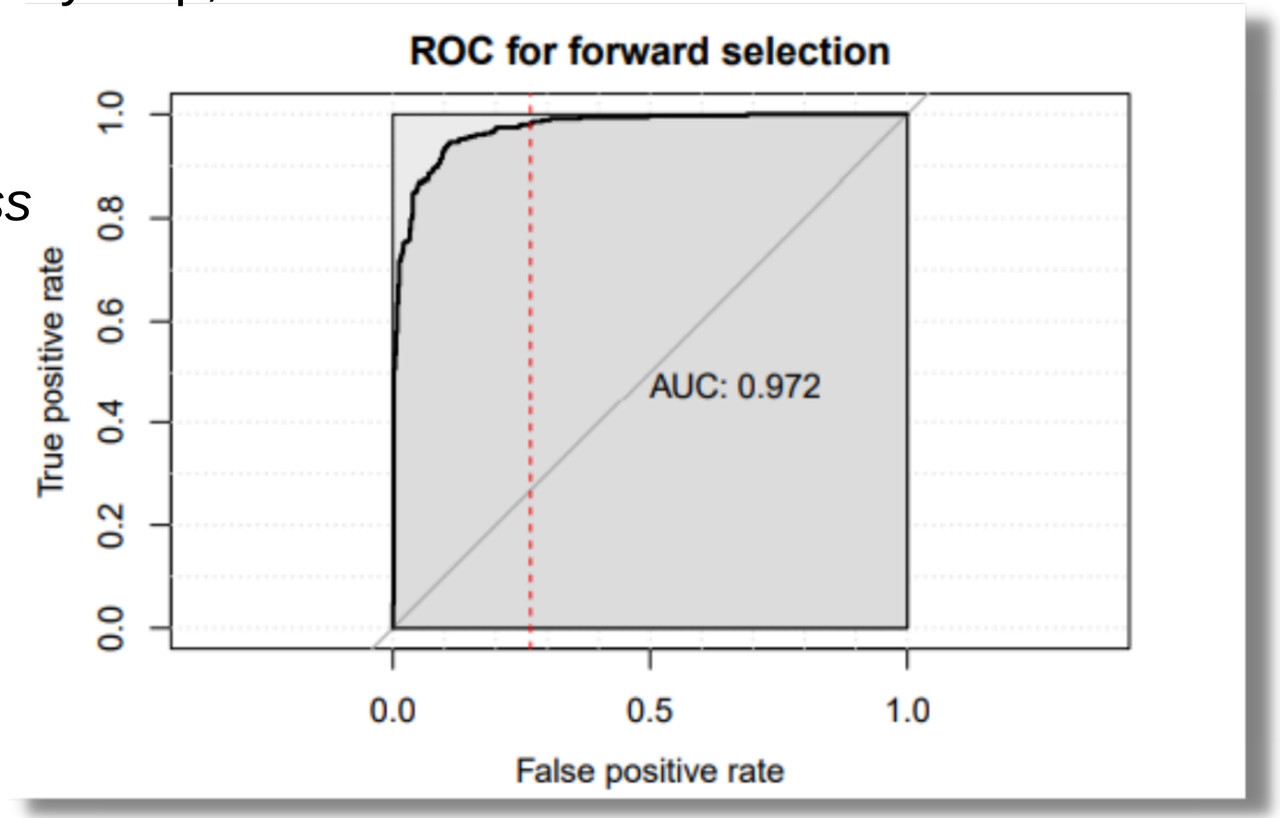
We can interpret this by saying that the model we started with had already taken out at least the same number of predictors as the best model we would have gotten using the removal method.

FORWARD STEPWISE SELECTION

We use the AIC also to compute the forward selection. We started from the intermediate model and we calculated the AIC of each model obtained by adding one of the discarded variables.

We found out that, step by step, we had to add:

- *word_freq_credit*
- *word_freq_over*
- *word_freq_business*



PRECISION:

96,1 %

SHRINKAGE METHODS

These methods are designed to address multicollinearity and overfitting.

Ridge

This method leads to more consistent coefficient estimates, mitigates the influence of multicollinearity, and improves the model's overall performance.

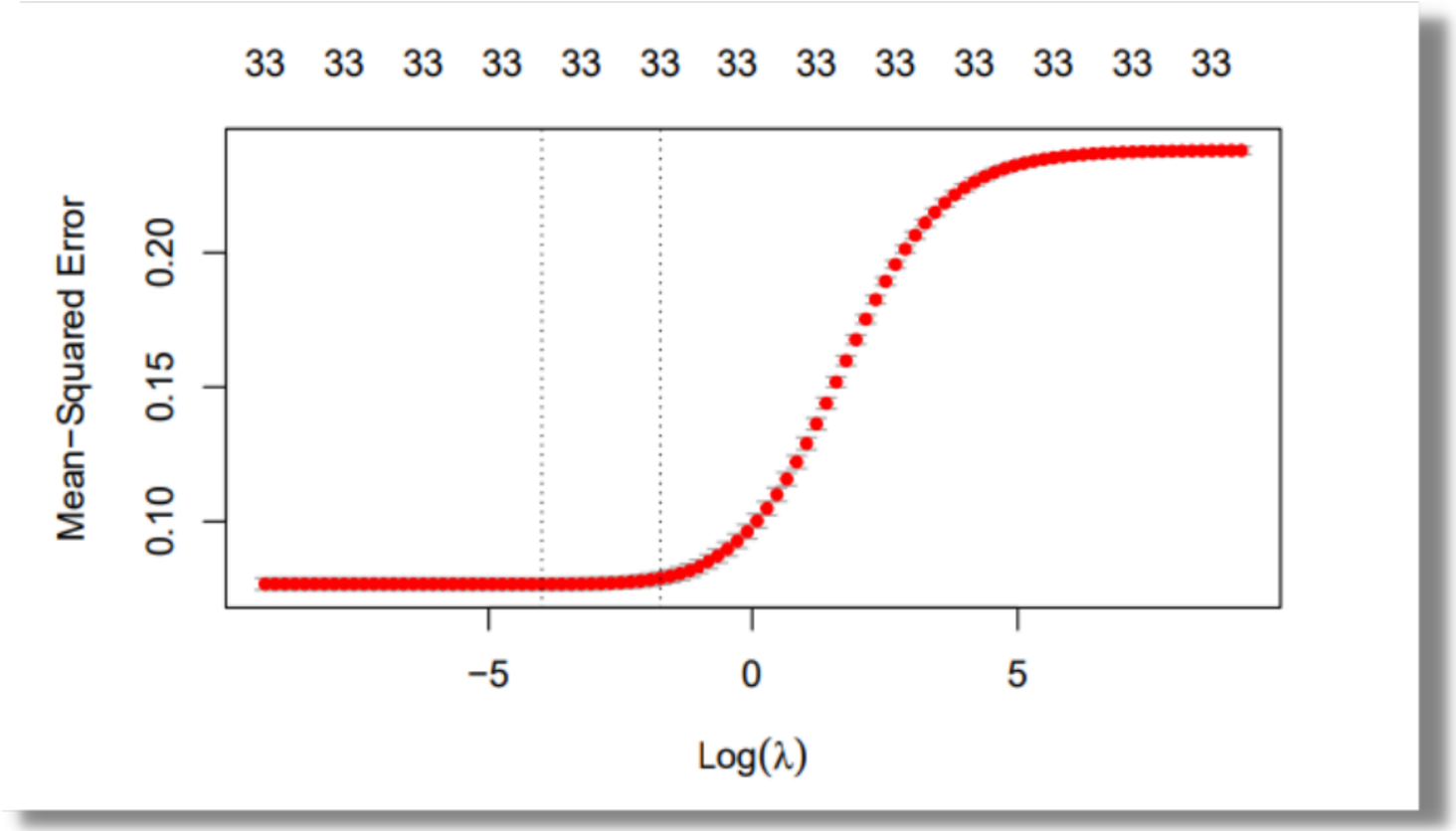
Lasso

Lasso Regression efficiently discerns and eliminates unimportant predictors from the model, offering a solution that retains only the most significant predictors.

RIDGE REGRESSION

HYPERPARAMETER LAMBDA SELECTION.

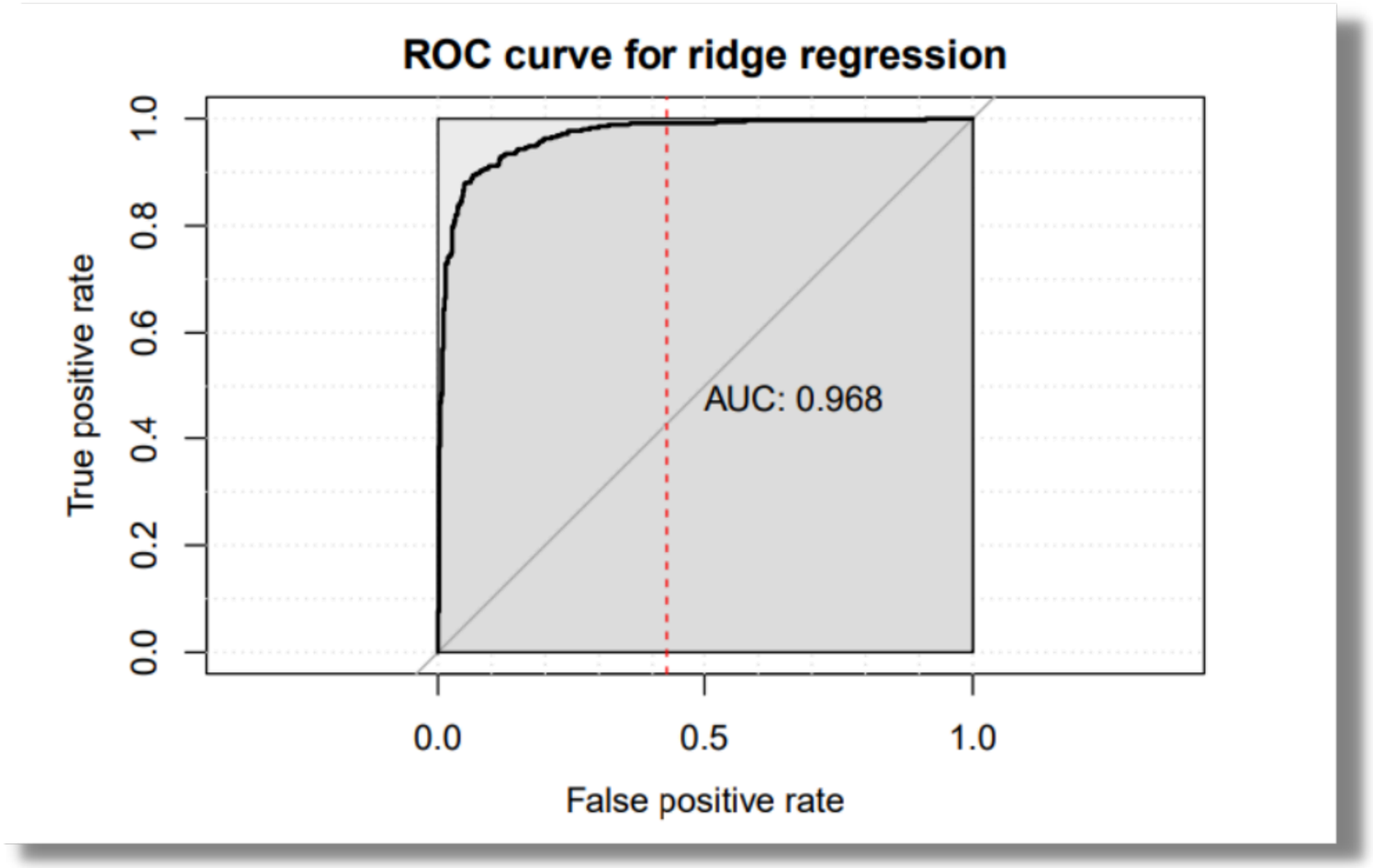
For the implementation of the model we need to create a grid for the hyperparameter *lambda* that is used into the model.



THE GRAPH ILLUSTRATES THE PATHS OF PREDICTOR COEFFICIENTS AS YOU CHANGE THE VALUE OF THE REGULARIZATION PARAMETER

RIDGE REGRESSION (CTND)

ROC CURVE:



PRECISION:

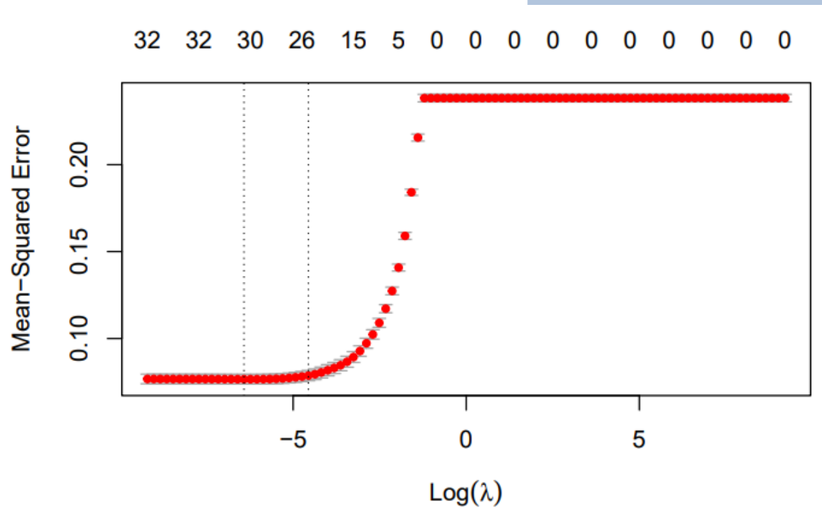
98,9%

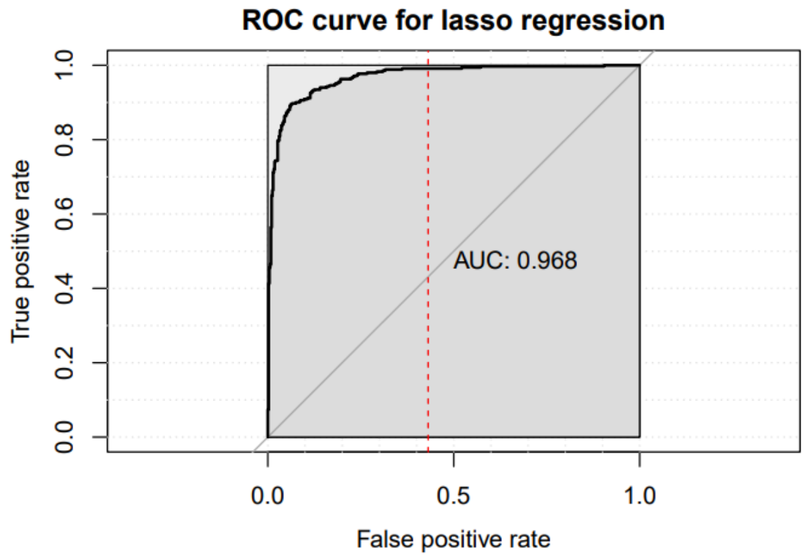
LASSO REGRESSION

The lambda value that minimizes the misclassification error results in a model where 8 features' coefficients are reduced to 0. These features are:

- *word_freq_receive*
- *word_freq_email*
- *word_freq_parts*
- *char_freq_.5B*
- *char_freq_.23*
- *word_freq_credit*
- *word_freq_over*
- *word_freq_business*

We have presented the ROC curve achieved using the lambda that minimizes the misclassification error.





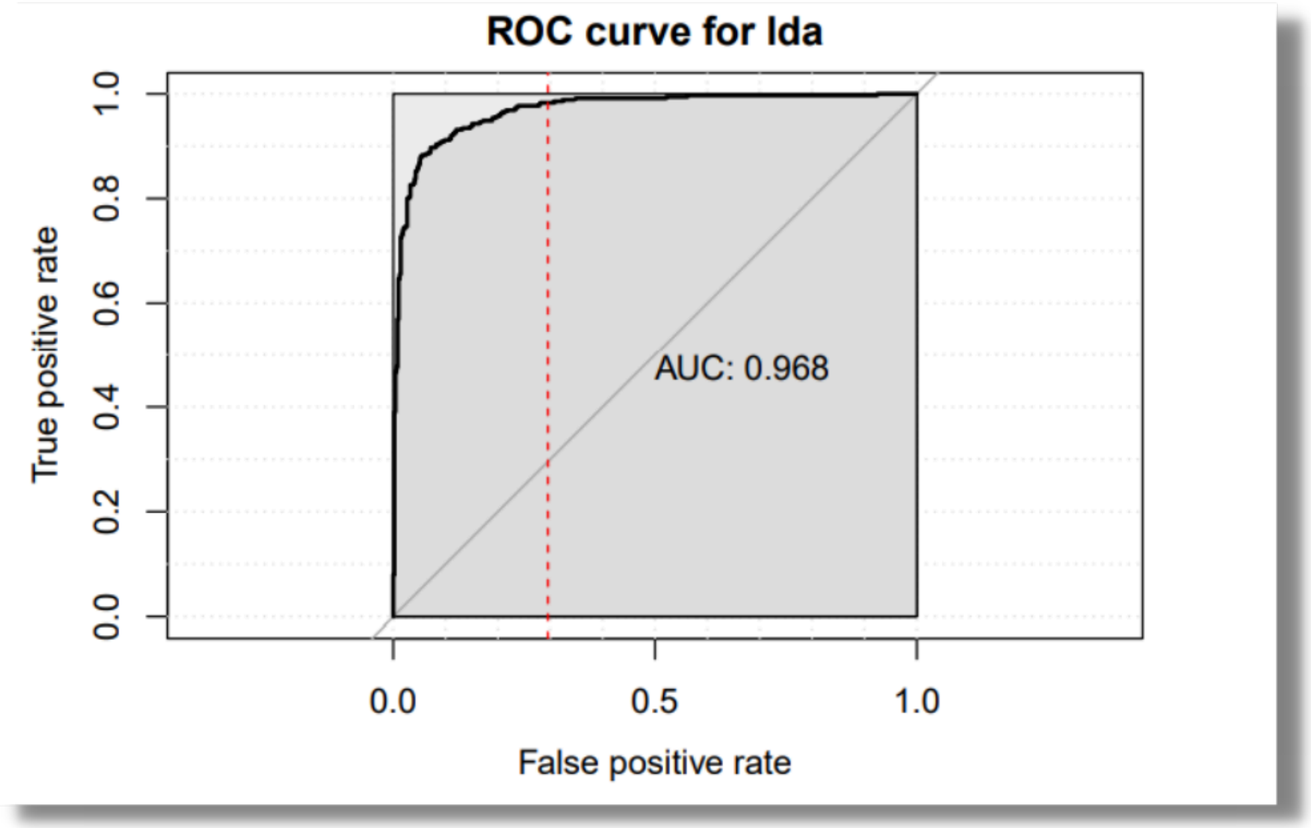
PRECISION:

97,3%

LINEAR DISCRIMINANT ANALYSIS

Although the normal condition of the predictors is not present, we still tried to implement the LDA:

As we can see from the ROC curve, the results were however more than satisfactory:



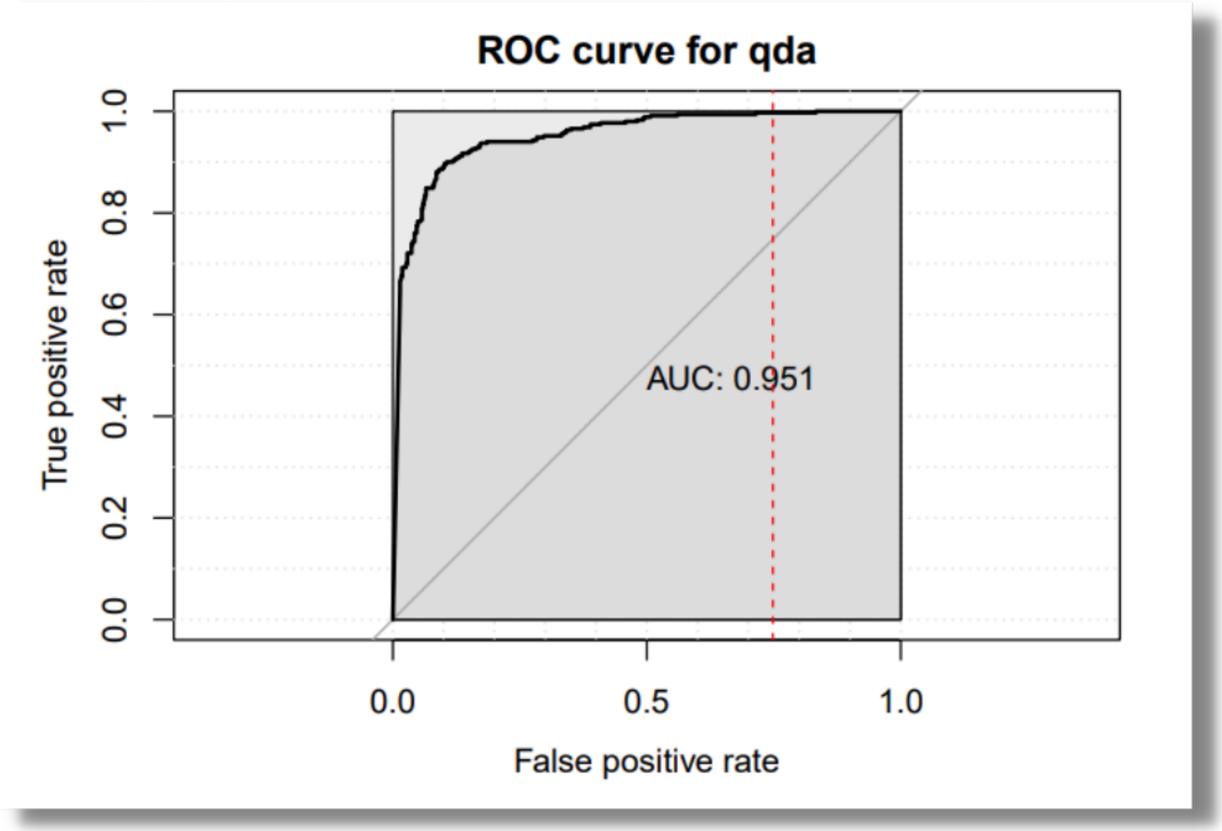
PRECISION:

97,3%

QUADRATIC DISCRIMINANT ANALYSIS

QDA relaxes the assumption of equal covariance matrices among classes and permits distinct variances and covariances for each class.

QDA performed slightly worst than LDA:

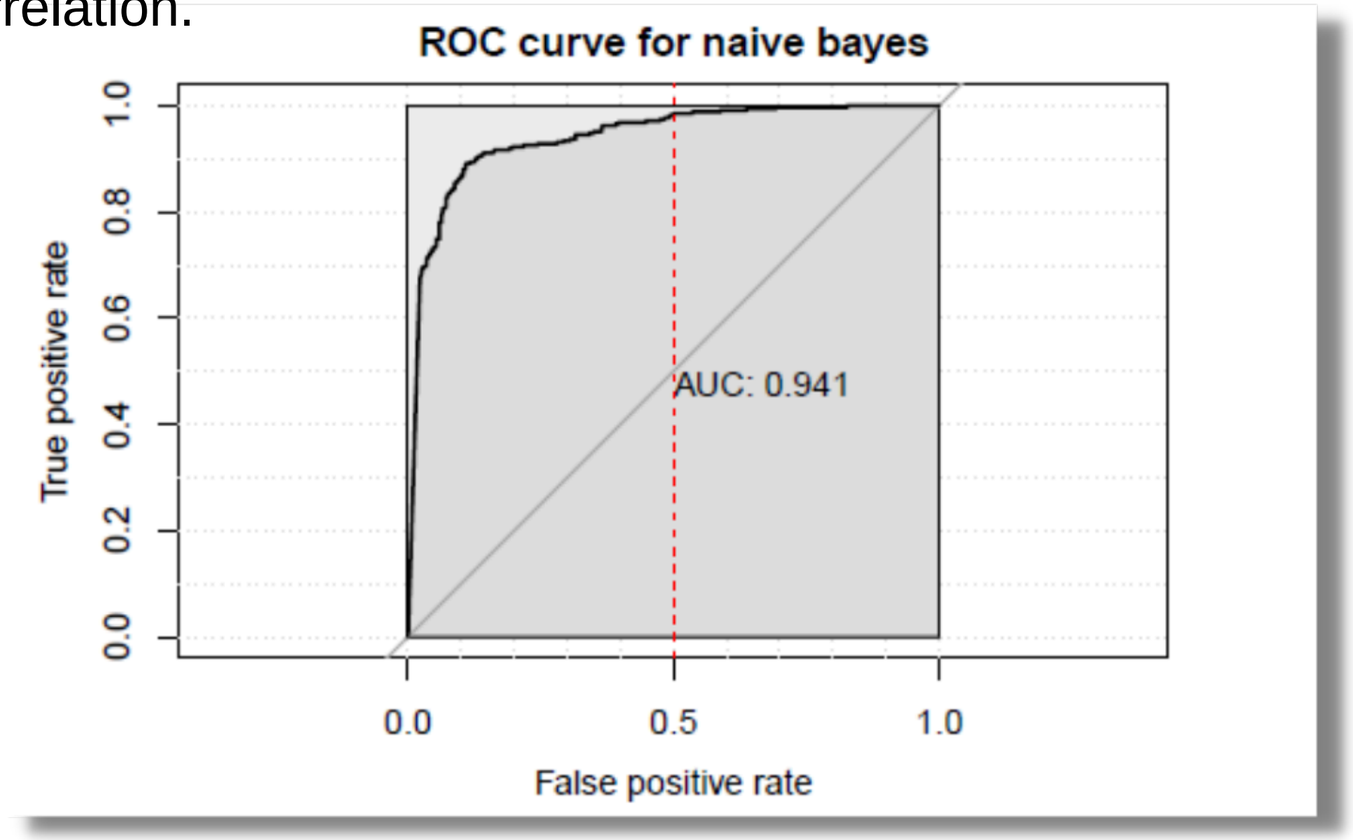


PRECISION:

89,6%

NAIVE BAYES

Naive Bayes is a straightforward technique used for making decisions between two choices. It involves analyzing various features associated with each choice. The term “naive” come from the assumption that these features are independent of each other, although in reality, they might have some level of correlation.

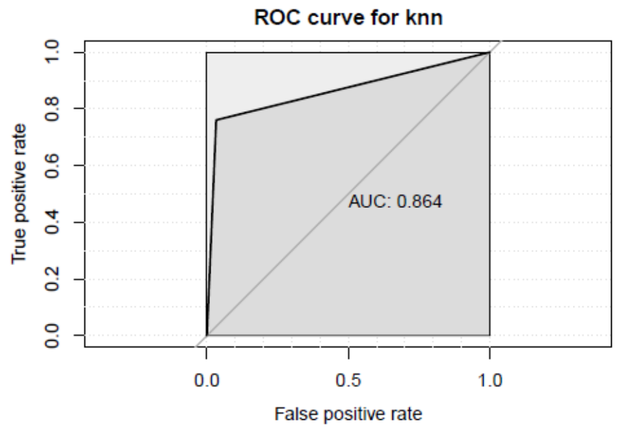


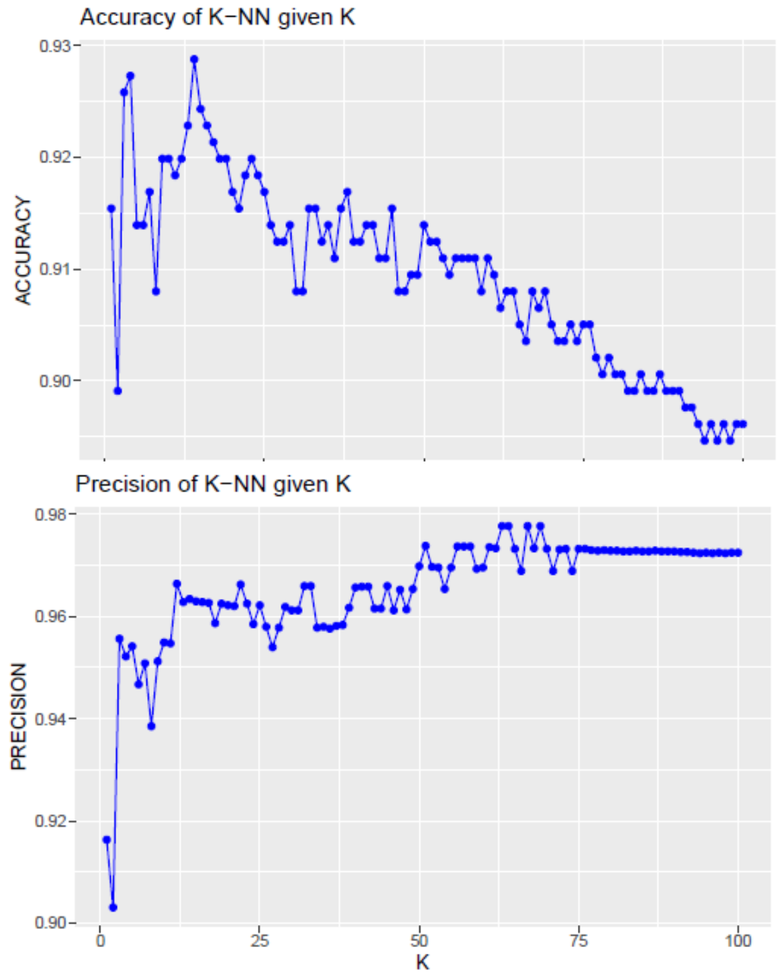
PRECISION:

89,2%

KNN

K-Nearest Neighbors is a method that predicts outcomes based on the similarity between data points. For a given new data point, KNN identifies its “k” then assigns a class or predicts a value for the new point based on the majority class or average value of its nearest neighbors.



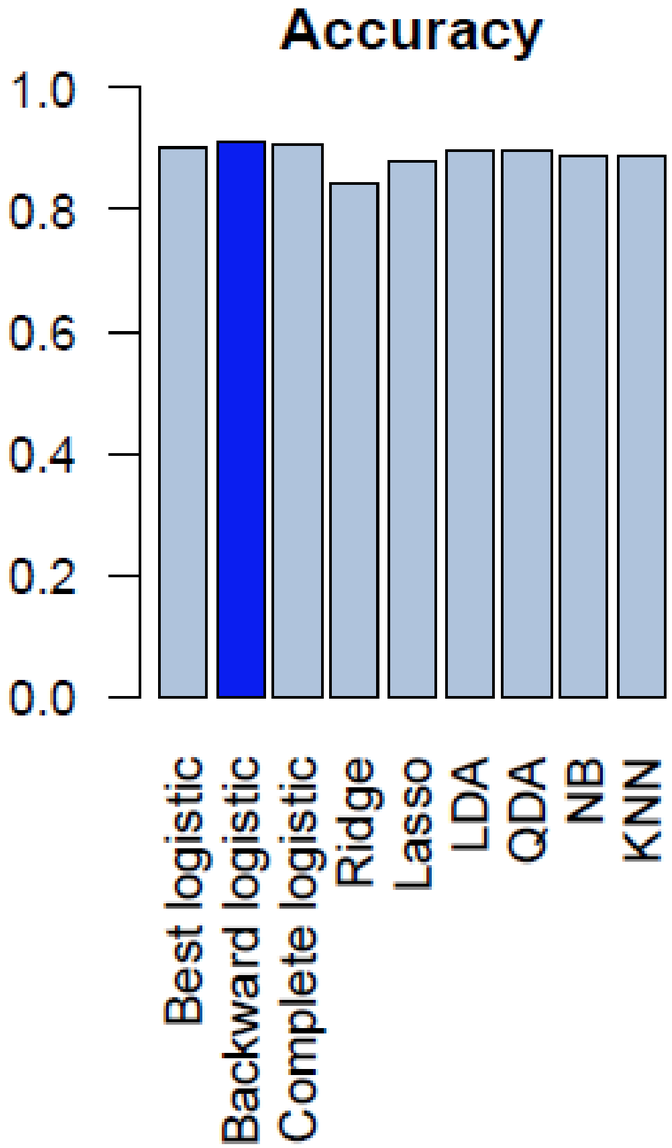


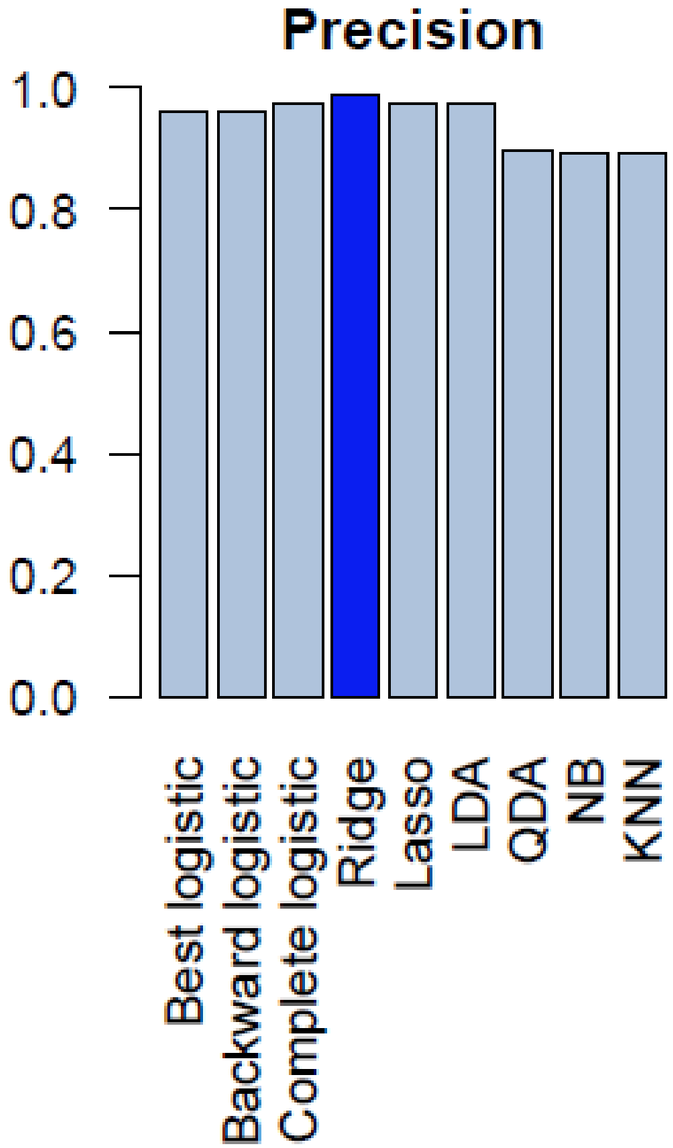
BECAUSE WE OPTIMIZE FOR PRECISION, WE CHOOSE K=63

Comparing models

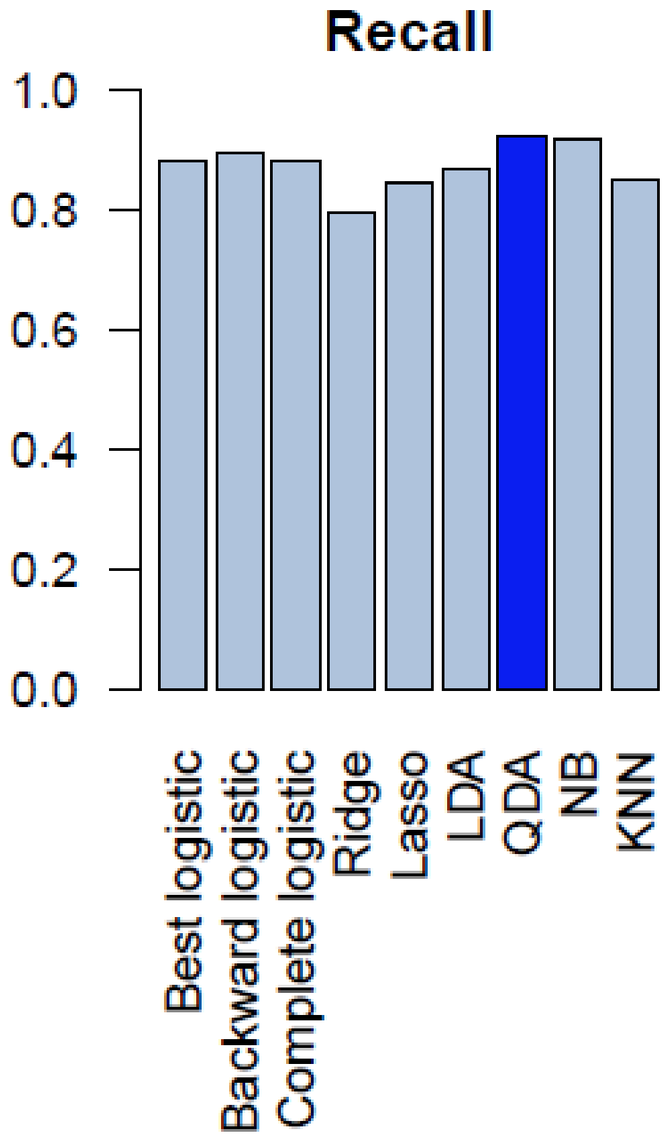
Model	Accuracy	Precision	Recall	F1_score	AUC
Best logistic	0.9038	0.9613	0.8839	0.9210	0.9717
Logistic without high p-value	0.9121	0.9633	0.8941	0.9275	0.9718
Complete logistic	0.9086	0.9756	0.8805	0.9256	0.9725
Ridge	0.8444	0.9898	0.7941	0.8812	0.9680
Lasso	0.8800	0.9735	0.8445	0.9044	0.9679
LDA	0.8979	0.9735	0.8675	0.9175	0.9678
QDA	0.8967	0.8961	0.9244	0.9100	0.9512
NB	0.8895	0.8921	0.9163	0.9040	0.9415
KNN	0.8895	0.8921	0.8515	0.8713	0.8665

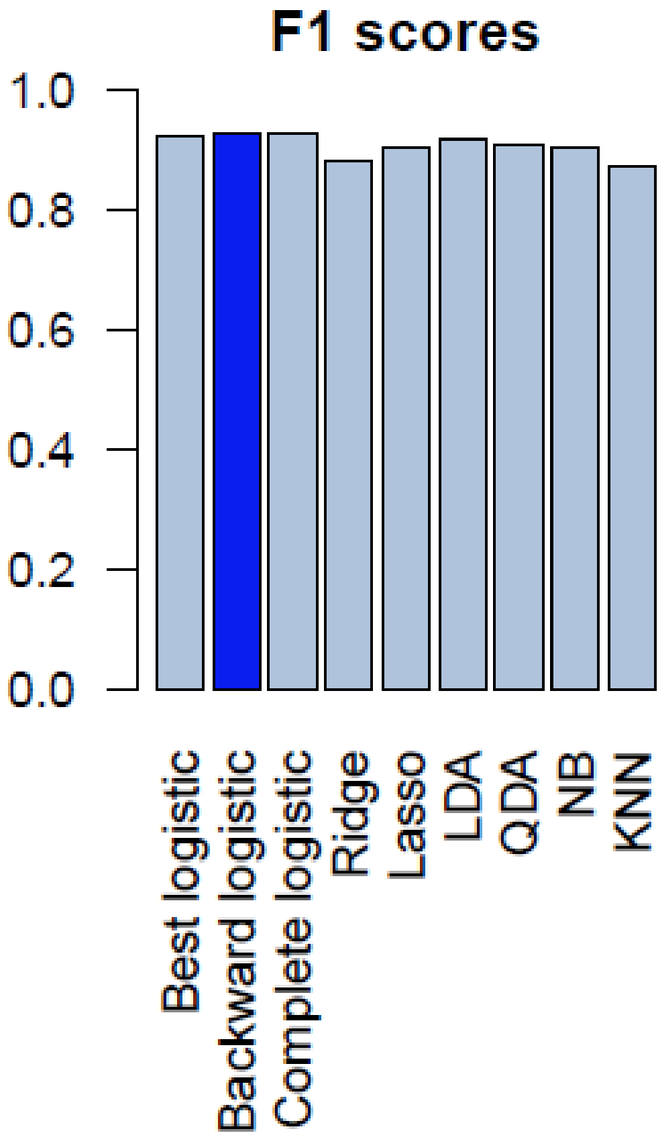
Comparing models



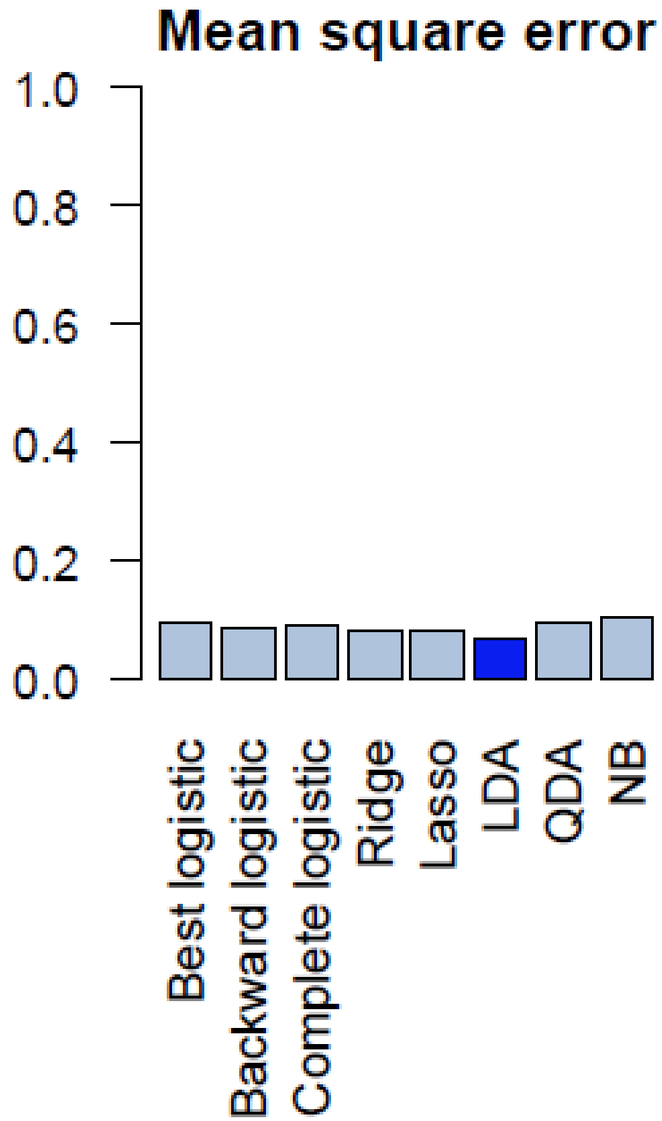


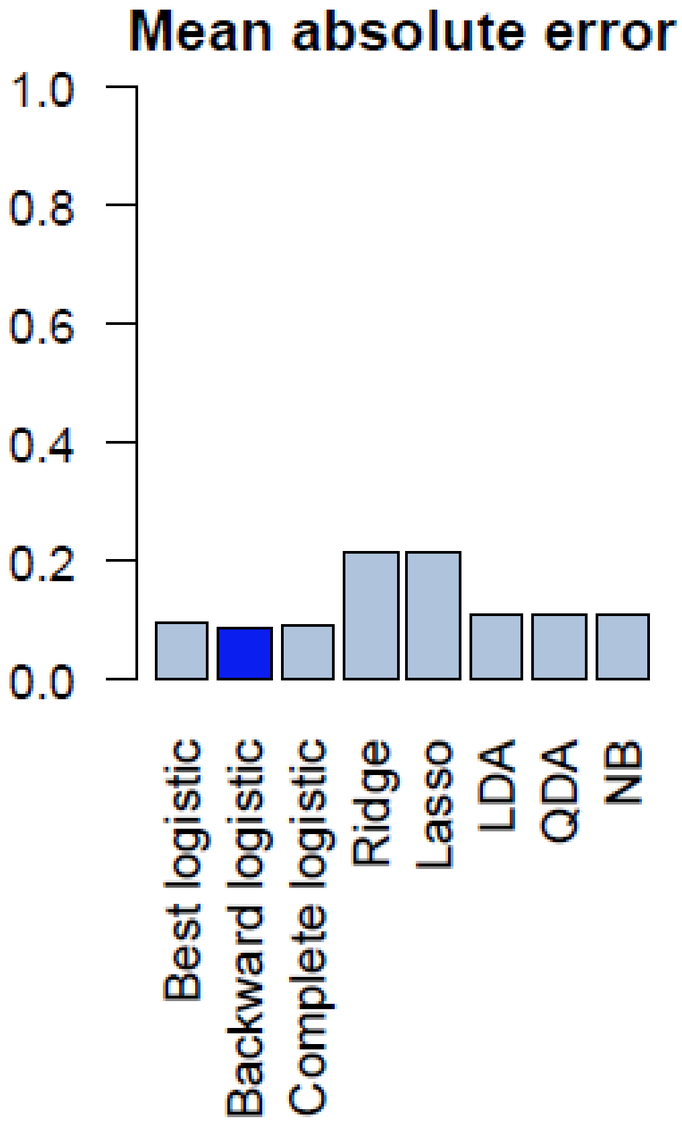
Comparing models



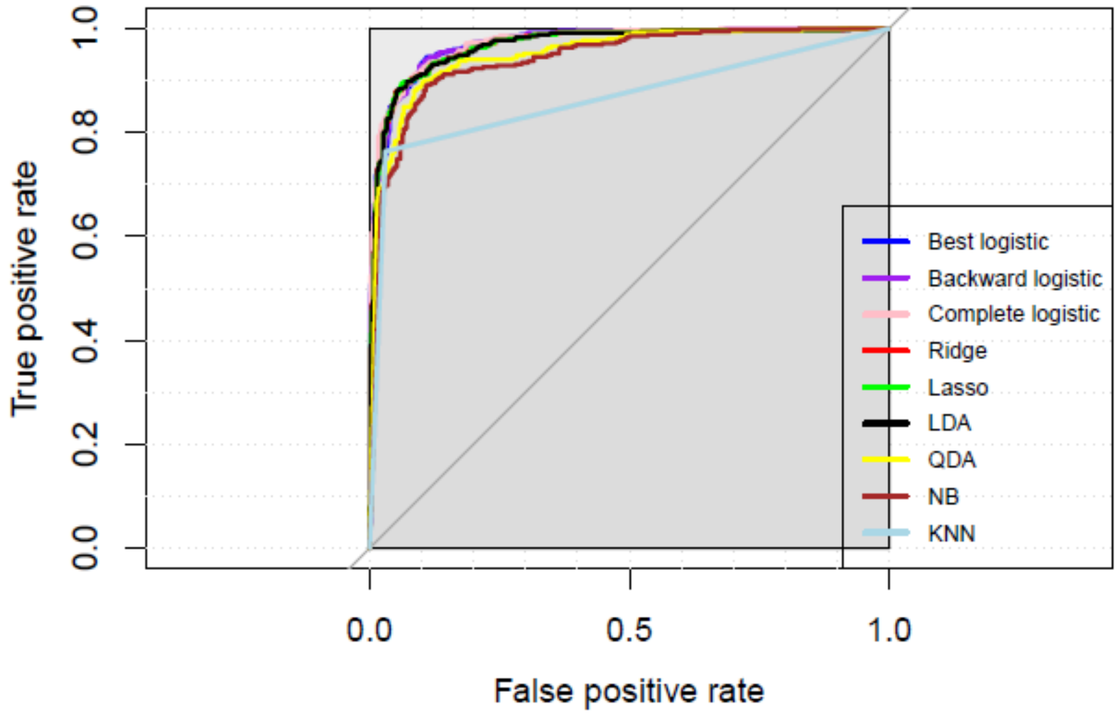


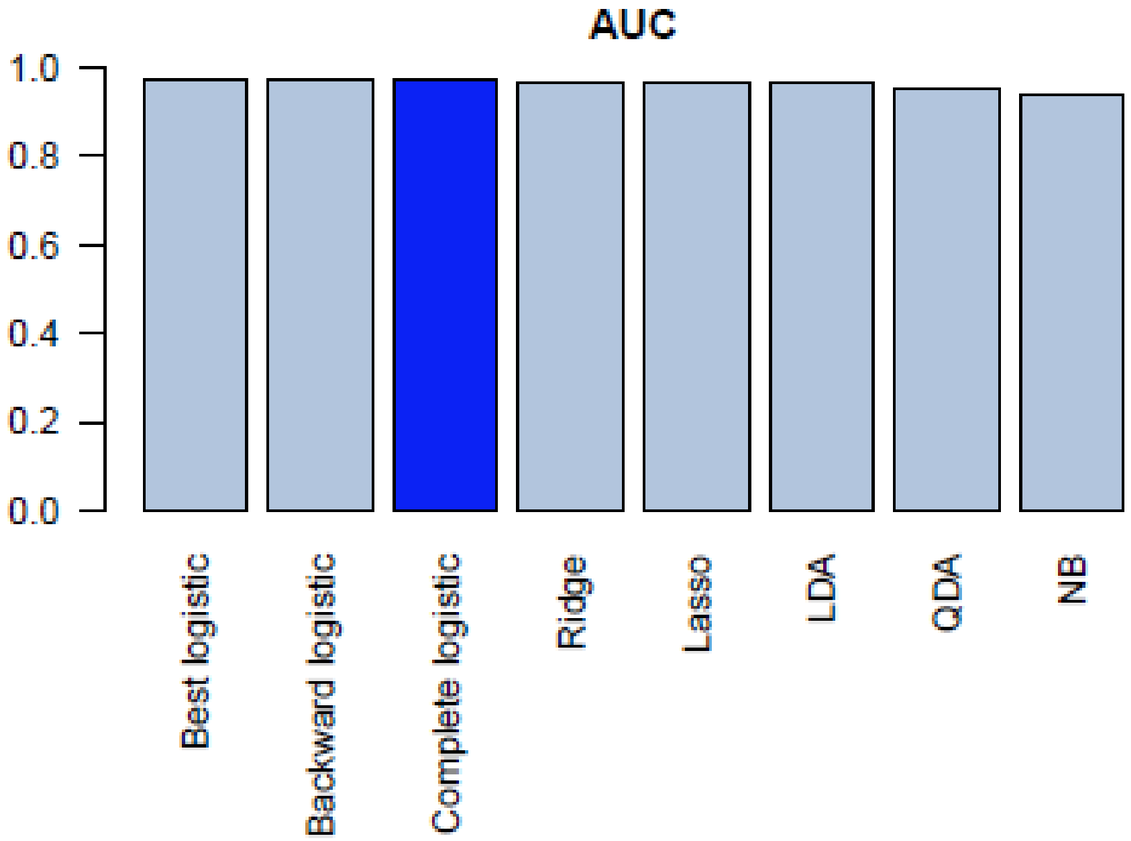
Residual analysis





ROC curve


True positive rate
0.0
0.2
0.4
0.6
0.8
1.0
0.0
0.5
1.0
False positive rate
Best logistic
Backward logistic
Complete logistic
Ridge
Lasso
LDA
QDA
NB
KNN


AUC
1.0
0.8
0.6
0.4
0.2
0.0
Best logistic
Backward logistic
Complete logistic
Ridge
Lasso
LDA
QDA
NB

Conclusions

1

A significant aspect of our study was accurately categorizing spam emails and from our analysis is possible to see that the obtained scores of accuracy, precision, recall, and F1-scores reflect an **impressive level of performance across all employed methods**. In particular the **reduced logistic method** and the **Ridge regression** gave very notable results, one because of it’s ability to balance and the other in terms of precision.

2

Another important goal of the project was to see if the frequency of certain words or characters in the text of a mail could be a good indicator of spam content. For what concerns this part is possible to say that the achieved results are particulary strong so the **different frequencies could be a good choice of variables for spam classification**. Maybe a way to improve the research could be investigating more on which words or characters can immediatly signal that a mail is spam but also the one considered in the dataset gave pretty good results.


Thank
you for the
attention