Stockholms universitet

Logistic Regression for Spam Filtering

Niclas Englesson

Kandidatuppsats i matematisk statistik
Bachelor Thesis in Mathematical Statistics

Kandidatuppsats 2016:9
Matematisk statistik
Juni 2016

www.math.su.se

Matematisk statistik
Matematiska institutionen
Stockholms universitet
106 91 Stockholm

Stockholm University

Mathematical Statistics
Stockholm University
Bachelor Thesis **2016:9**
http://www.math.su.se

Logistic Regression for Spam Filtering

Niclas Englesson*

June 2016

Abstract

Unsolicited bulk emails, also known as spam emails, are a regular occurrence for anyone who uses email. *Spam filtering* is a way to distinguish between spam emails and regular emails. The goal with spam filtering is to determine whether an email is spam or not spam, then filtering out the spam emails, resulting in a spam-free in-box for the user.

Logistic regression is a statistical method that can be utilized for spam filtering. It is sensible that spam emails typically share a certain type of characteristics. Words that recurrently show up in spam emails can be used as predictor variables in the logistic regression model. Other email characteristics, such as special formatting, tables, links, may also be used as predictor variables. More on this in section **??**.

This report looks into what determines the probability of an email being a spam email by using logistic regression. We will examine if certain characteristics alter the probability of an email being a spam email or not. We will also test which model best predict the probability of an email being spam.

The study initially contain 12 variables that potentially could alter the probability of an email being spam. After variable selection, the number of explanatory variables decrease to 5 in the chosen model. The variables shown to be insignificant are excluded from the model while all the significant variables are kept in the model. It appears that factors such as *winner*, *mult_sent*, *prior_corr* and *sender_spam* have a significant effect on determining the probability of an email being spam, and thus should be used in a spam filter.

*Postal address: Mathematical Statistics, Stockholm University, SE-106 91, Sweden. E-mail: niclas.englesson@gmail.com. Supervisor: Gudrun Brattström and Jan-Olov Persson.

Bachelor Thesis in Mathematical Statistics
Logistic Regression for Spam Filtering

Niclas Englesson

May 25, 2016

Contents

Abstract

Unsolicited bulk emails, also known as spam emails, are a regular occurrence for anyone who uses email. *Spam filtering* is a way to distinguish between spam emails and regular emails. The goal with spam filtering is to determine whether an email is spam or not spam, then filtering out the spam emails, resulting in a spam-free in-box for the user.
Logistic regression is a statistical method that can be utilized for spam filtering. It is sensible that spam emails typically share a certain type of characteristics. Words that recurrently show up in spam emails can be used as predictor variables in the logistic regression model. Other email characteristics, such as special formatting, tables, links, may also be used as predictor variables. More on this in section 3.2.2.
This report looks into what determines the probability of an email being a spam email by using logistic regression. We will examine if certain characteristics alter the probability of an email being a spam email or not. We will also test which model best predict the probability of an email being spam.
The study initially contain 12 variables that potentially could alter the probability of an email being spam. After variable selection, the number of explanatory variables decrease to 5 in the chosen model. The variables shown to be insignificant are excluded from the model while all the significant variables are kept in the model. It appears that factors such as *winner*, *mult_ sent*, *prior_ corr* and *sender_ spam* have a significant effect on determining the probability of an email being spam, and thus should be used in a spam filter.

1 Introduction

Unwanted electronic messages, also known as *spam*, is electronic junk mail. These kinds of emails are usually trying get the recipient to buy some product or service[1]. An estimation shows that close to 80% of all the email traffic is spam [6]. A *spam filter* is a software that keeps spam emails from entering the in-box. Hence, it predicts if an email is considered spam or no-spam, and decides if the email should be displayed in the in-box or be junked.[2]

1.1 Aim

The aim of this paper is to build a model that can predict for the outcome if an email is *spam* or *no-spam*. The paper will try to give answers to the following questions:

- How can we construct a spam filter, given the data set?
- What factors alter the probability of an email being a spam-email?
- How does one create a model that can predict if an email is spam?
- What is the risk of the model making false predictions?
- What improvements can be made to the study?

If the model were to make false predictions it could result in the spam filter letting in spam-emails or junking non-spam-emails, which is not preferable. Therefore we want to create the best possible model where inaccurate predictions are scarce, if not non-existent.

1.2 Problem Description

The ordinary approach towards regression requires nearly normal residuals. However, there are circumstances when this is impossible. An important case is when the response is categorical with only two levels. Throughout this paper the *Logistic Regression Model* is used. Logistic regression is an approach for demonstrating the relationship between a binary dependent variable, and several explanatory variables. With the dependent variable being binary, it can only take on two values, which in this case is "Spam", or "No Spam". However, the explanatory variables can be both numerical and categorical.
In this paper, an email data set is introduced, for which the goal will be to build a spam filter. The fundamentals of logistic regression are covered and modelling the probability of an event. The data set is obtained from a regular email in-box (my personal *www.hotmail.com* in-box) with the goal of discovering characteristics of spam emails. The data set is created manually. Therefore it is important to clarify the definition of spam that has been used when collecting for data: spam emails are *unsolicited* and *not personal*. See section 3.2.4 on

how to determine if an observation is marked as *spam*.
Logistic regression is used to come up with a model that tells what factors are significant for determining if an email is spam *and* predicts if an email in question is in fact *Spam* or *not-spam*.

1.3 Disposition of the paper

Section 1 introduces the aim with this paper and describes the problem. Section 2 covers all the theoretical knowledge needed for this study. Later in the report it will be described how data is processed. Thereafter, in section 4, we will look more closely into which factors are significant/insignificant, after which a final model will be built. Lastly, section 5 and 6 will conclude with a discussion regarding the final model and its prediction accuracy. What flaws there are with the study are later mentioned and suggestions are made on how one could make the study more accurate.

2 Theory

This section will look into some of the theory that is necessary for the development of the prediction models. The vast majority of the theory is cited Agresti *Categorical Data Analysis* [1]. Additional references will be presented in section **References** .

2.1 Logistic regression

Logistic regression is a statistical method used to demonstrate if a binary response variable Y is dependent on one or more independent variables $X = (X_1, \cdots, X_n)$. It is a tool for for building a model in situations where there is a two-level categorical response variable, in contrast to a numerical response variable, where multiple linear regression would be more appropriate. Like multiple regression, logistic regression is a type of GLM[3] with the difference being the categorical response variable.
The outcome of a GLM is usually denoted by Y_i, where i stands for observation number i.

In this report, Y_i will denote if an email is spam or not; $(Y_i = 1)$ for spam, and $(Y_i = 0)$ for non-spam. The independent variables X will take on the following form; x_{ij} denotes the value for variable j for observation number i.The outcome Y_i takes on value $(Y_i = 1)$ with probability π_i and $(Y_i = 0)$ with probability $(1 - \pi_i)$.

The logistic regression model links the probability of an email being spam (π_i) to the prediction variables $(x_{1i}, \cdots, x_{ij})$ through a framework very similar to that of multiple regression. Since the response is binary, we need to find a suitable transformation in order to make the regression model work. A natural transformation for π_i is the *logit transformation*:

$$logit(\pi_i) = ln(\frac{\pi_i}{1 - \pi_i}) \tag{1}$$

The logistic regression model is given by:

$$ln(\frac{\pi_i}{1 - \pi_i}) = \alpha + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_j x_{ij} \tag{2}$$

Note that since the probability of an email being *spam* (π_i) is a number between zero and one, the $log(\frac{\pi_i}{1-\pi_i})$ can take on any real number:

$$0 \leq \pi_i \leq 1 \quad \Longrightarrow \quad -\infty < ln(\frac{\pi_i}{1 - \pi_i}) < +\infty$$

The relation between $P(Y_i = 1)$ is obtained by solving 2 for π_i. We get:

$$P(Y = 1|X = x) = \pi_i = \frac{\exp(\alpha + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_j x_{ij})}{1 + \exp(\alpha + \beta_1 x_{i1} + \beta_2 x_{i2} + \cdots + \beta_j x_{ij})} \tag{3}$$

$$= \frac{\exp(\alpha + \boldsymbol{\beta x})}{1 + \exp(\alpha + \boldsymbol{\beta x})}$$

Equation 3 is the *logisic regression model* that will be utilized throughout this paper.

We define the odds as

$$\Omega = \frac{\pi_i}{1 - \pi_i} \tag{4}$$

where the odds is the probability of the outcome *spam* divided with the probability of the outcome *no spam*. By taking the logarithm on both sides we get equation 2. The logistic regression coefficients correspond to the change in the log odds, for each variable respectively. The exponentiated form of the coefficients correspond to the odds ratio.

2.2 Odds-ratio

Equation 4 tells us that if the probability of the outcome *spam* is between $[0, 1]$, the odds will be non-negative. We also see that if $\Omega > 1$ the probability of *spam* is greater than the probability of *no-spam*. An example is the odds ratio for *prior correspondence*.
Let Ω_{pc} denote the odds for an e-mail being *spam* for e-mails *with* prior correspondence with the sender, and Ω_{npc} denote the odds for an e-mail being *spam* for e-mails *without* prior correspondence with the sender. The odds ratio is defined as:

$$\theta = \frac{\Omega_{pc}}{\Omega_{npc}} = \frac{\pi_{pc}/(1 - \pi_{pc})}{\pi_{npc}/(1 - \pi_{npc})} \tag{5}$$

where $\pi_{(n)pc}$ is the probability for an e-mail being spam for e-mails *with* (*without*) prior correspondence with the sender. $\theta > 1$ implies that the probability π_{pc} is greater than π_{npc}. If $\theta = 1$ the outcome does not depend on whether you had prior correspondence with the sender or not.

2.3 Purposeful selection

In order to develop a model that can be used for spam filtering we need a clear approach. When developing a model using Logistic regression, one can use *Purposeful selection* [7]. It is a procedure that contains 7 steps:

Step 1: Purposeful selection begins with individual examination of the independent variables. Starting by performing simple logistic regression on every factor to the outcome Y respectively. Here we are not so strict when deciding what $p - value$ is required for including a certain variable in the model. Limits such as 0.25 are more likely to be used than the regular 0.05 limit. This is due to the fact that we might exclude variables that later show to have impact in a model together with other variables. Excluding these variables is undesirable.

Step 2: Now we include all the variables in a multivariate logistic regression model. Include all the predictors with a p-value less than 0.25. Look at the Wald Statistic of the predictor variables and exclude variables at a preferred significance level, such as 0.05. Now, compare the new smaller model with the full model by performing a likelihood ratio test.

Step 3: Compare the estimated coefficients in the smaller model with the bigger model. We are particularly interested in the variables whose parameter estimates differ a lot. $\Delta\hat{\beta}_i = |(\hat{\theta}_i - \hat{\beta}_i)/\hat{\beta}_i| > 0.2$ is used as an indication that the parameter estimates differ to much. $\hat{\beta}_i$ denotes the parameter estimate for variable i in the full model and $\hat{\theta}_i$ is the corresponding parameter estimate in the smaller model. If $\Delta\hat{\beta}_i > 0.2$ the variable is removed.

Step 4: One by one, add the variables that were insignificant in *step 1* to the smaller model and check its level of significance. This is an important step to identify possible association terms. Variables that independently were not significant may still be jointly dependent with other variables of Y.

Step 5: Examine the included variables in the model. The levels of the categorical variables shall be reasonable (logical direction of the categories) and the continuous variables should have a linear relationship with the logit.

Step 6: Now we must consider all the interactions between variables that may be present. An interaction between variables means that the impact a variable has on Y is not constant over different levels of the other the other variable. Even interactions between previously excluded variables are tested. If an interaction term is to be added to the model, it must be motivated both mathematically and realistically in order to be included. We add interaction terms, one at a time, and check if it is statistically significant or not, in terms of a low p-value. The interaction terms that make the 0.05 limit gets added to the model. Now we repeat step two, but consider the main effects as fixed. The selected model after *Step 6* is called the *preliminary final model.*

Step 7: We must now test how well the final model can explain variation in data. See section 2.4 below.

2.4 Model fit and diagnostics

The purpose of a statistical model is to describe variation in data while being simple and understandable. We will preform a handful of test the different models to see how well they fit data. We will also test their prediction capacity and make a comparison between models. This section will present the tests that are used in this study.

2.4.1 Likelihood-ratio test

Consider a statistical model M_0 with a certain set variables and interaction terms included. Goodness of fit of M_0 can be formalized as an hypothesis test between M_0 and a larger model M_1:

$$H_0 : M_0 \text{ holds}$$

$$H_1 : M_1 \text{ holds, but not } M_0.$$

We test this by using the deviance, which is obtained by taking the likelihood ratio test between M_0 and M_1:

$$G^2(M_0) = -2(l_0 - l_1)$$

where l_0 is the maximized log-likelihood function under the null hypothesis and correspondingly l_1 is the maximized log-likelihood function under the alternative hypothesis.
The likelihood ratio statistic G^2 is asymptotically χ^2-distributed under the null[4]:

$$-2(l_0 - l_1) \overset{H_0}{\approx} \chi^2_{df}$$

where *df* denotes the degrees of freedom, which is obtained by taking the difference between the number of parameters in M_1 and M_0.

2.4.2 The Hosmer-Lemeshow test

Suppose we have n estimated probabilities of n different events. We group them in order of size so that the first column is the lowest estimated probability and the n'th column is the largest estimated probability. The Hosmer-Lemeshow test categorizes the estimated probabilities in e.g. $g = 10$ groups, where the first group contains the lowest 10% of the estimated probabilities and the last group contains the highest 10% of the estimated probabilities. The Pearson chi-squared statistic is then used to compare between observed and estimated values, which asymptotically results in a χ^2-squared statistic, $\hat{C}$, with $(g-2)$ degrees of freedom.
If $\hat{C}$ is significant then the model in question does *not* fit data very well. More details regarding this test can be found in chapter 5.2.2 in Hosmer and Lemeshow (2013).

2.4.3 AIC

One of the most common ways to compare between models is to use the *Akaike Information Criterion*, AIC. It compares between models that are built from the same data set. This means that it doesn't actually say anything about the goodness of fit of a model as a whole, but makes a comparison to how well the model describes the variation in data compared to the full model. AIC is defined as:

$$AIC = 2p - 2ln(L) = -2(ln(L) - p)$$

where p is the number of parameters in the model and L is the maximized likelihood function for that same model. Given a number of models, the preferred model is the one with *lowest* AIC. A model is likely to have a low AIC when it has a high likelihood and few parameters.

2.4.4 ROC

A way of finding out how well a model can make predictions is by creating a *receiver operating characteristic* (ROC) curve. Let

$$\hat{y}_i = \begin{cases} 1 \text{ if } \hat{\pi}(x_i) > \pi_0 \\ 0 \text{ otherwise} \end{cases}$$

where $\hat{y}_i$ is the predicted value of y_i (the outcome) and π_0 is a *cutoff* limit. Agresti (page.228) makes the following definitions:

$$\textit{Sensitivity} = P(\hat{y} = 1 | y = 1) \text{ and } \textit{Specificity} = P(\hat{y} = 0 | y = 0).$$

A *receiving operating curve*, ROC, is a curve that plots *Sensitivity* against *Specificity* for all possible values of π_0 To be more precise, the ROC curve plots

Sensitivity against 1-*Specificity*. This creates a concave curve that connects the points (0.0) and $(1,1)$ in the XY-plane. The area under the curve reflects the predictive ability the model in question has, where a larger area reflects a better predictive ability.
The area 0.5 is created when we calculate the area under a straight line that goes from (0.0) to $(1,1)$. Obtaining the area 0.5 means predictions of the model where no better than random guessing (*Agresti* page.229).

Hosmer and Lemeshow (2013 page.177) provide guidelines on how to interpret different areas under the ROC-curve (AUC):

$$\text{If} = \begin{cases} 0.5 < AUC < 0.7 & Poor \\ 0.7 < AUC < 0.8 & Acceptable \\ 0.8 < AUC < 0.9 & Exellent \\ 0.9 < AUC & Outstanding. \end{cases}$$

One has to keep in mind the disadvantage with using the AUC as an index of prediction capability is that it doesn't take into account the number of parameters in the model. We prefer a model with few parameters because it is easier to interpret.

2.4.5 Generalized R-squared

There is a risk of using to many parameters in the model *if* we only use AUC when measuring for predictive capacity. Therefore we need to introduce an adjusted R^2-measure. The R^2-measure and AIC both take into account the number of variables in a model, but they differ in other aspects.
In linear regression, R^2 explains how much of the variance can be explained by the independent variables under linear conditions, whereas AIC is a trade of between goodness of fit of the model and model complexity. The R^2-measure is adjusted so that it can go both go up and down when a variable is added, depending on if it adds or doesn't add explanatory power to the model. Although, AIC does not have to change when adding a variable to a model, it changes with the predictors' composition.
The formula of the R^2-measure is given by:

$$R^2 = 1 - \frac{\sum_i (y_i - \hat{y}_i)^2}{\sum_i (y_i - \bar{y}_i)^2}$$

where y_i is the observed value, $\hat{y}_i$ is estimated value and $\bar{y}_i = \frac{\sum_i y_i}{n}$ is the mean. However, in logistic regression, this is not very useful since our outcome y is binary. We need to introduce another adjusted R^2-measure that uses the likelihood function. Mittböck and Schemper (1996) make the following definition:

$$R^2_E = 1 - \frac{l_1}{l_0}$$

where l_1 is the maximized log-likelihood for the model and l_0 is the maximized log-likelihood for the a model with only an intercept. The R^2 available in the statistical software of SAS[6], can be defined by the following equation:

$$R^2_{sas} = 1 - exp(2(l_0 - l_1)/n)$$

where n is the sample size. This definition is analogous to that used by Mittböck and Schemper.[5] The defined R^2_{sas} cannot attain a value of 1, which is a disadvantage with the measure. For this reason Nagelkerke (1991) proposed the following adjusted measure[5]:

$$R^2_{adj} = R^2_{sas}/(1 - exp(2l_0/n))$$

For more information on this measure, see Mittböck and Schemper (1996)[8].

2.4.6 Cross Validation

When evaluating a model we want to asses how well it predicts on subsets of data. Cross Validation is an algorithm to estimate predictive errors. It is based on leaving one observation out of the data set and building a model from the remaining observations, then predicting on that observation and measuring the prediction error.
The ROC curve is used to assess binary response models (logistic regression models). It is done by fitting a model to the data set and using Cross validated predicted probabilities to provide a ROC analysis. " *The cross validated predicted probability for an observation simulates the process of fitting the model ignoring the observation and then using the model fit to the remaining observations to compute the predicted probability for the ignored observation*" [11].

2.4.7 AIC or R-Squared?

When building logistic regression models, it is possible that the measures R^2 and AIC speak for different models. For example, R^2 could be lower for one model, that is better according to AIC, than for another bigger model. The question is if the model with low R^2 and low AIC should be picked, or if the model with higher R^2 values *and* higher AIC should be picked?
It depends if the goal with the model selection is model parsimony or predictive power of the model. If model parsimony is preferred, then use the model with low AIC. If predictive power is more important, then use the model with higher R^2.
So which model do we pick? Usually the answer is the same regardless of looking at AIC or R^2, but in some cases when comparing models with very similar R^2 values the answer can be different.

3 Processing Data

The data set that will be used in this study contains 353 observations of emails that were collected from my personal in-box. The goal is to build a spam filter with this data set by using logistic regression. The data set is obtained manually and extends from 05/10-2015 to 20/02-2016.

3.1 Description of data

In the data set there are two types of variables; categorical and numerical. A numerical variable can take on any value in its defined range, while a categorical variable can only take on the different categories. A binary variable is a categorical variable with only two outcomes. It represents occurrences of an event happening. The binary variables are encoded to take on values $X_i = 1$ if the event occurred, and $X_i = 0$ otherwise. An example would be the outcome variable Y, if an email is spam or not, then Y takes on the value $Y_i = 1$, and if it is non-spam, it takes on $Y_i = 0$.
The goal is to create a useful model that discerns *spam emails* from *regular emails* using the characteristics of the email; the predictor variables.

3.2 Predictor variables

Variable	Notation	Description	Type
Outcome	Y	Tells if the email is spam/no-spam.	Binary $Y = j,\ j = 0, 1$
dollar	X_1	Indicates if the email contains the word "dollar".	Binary $X_1 = j,\ j = 0, 1$
winner	X_2	Indicates if the email contains the word "winner$".	Binary $X_2 = j,\ j = 0, 1$
password	X_3	Indicates if the email contains the word "password".	Binary $X_3 = j,\ j = 0, 1$
inherit	X_4	Indicates if the email contains the word "inherit".	Binary $X_4 = j,\ j = 0, 1$
re_subject	X_5	Indicates if "RE" was included in the start of the email (SV in Swedish).	Binary $X_5 = j,\ j = 0, 1$
attach	X_6	If there was an attachment, such as an image or a document.	Binary $X_6 = j,\ j = 0, 1$
multiple_sent	X_7	If the email was listed to more than one person.	Binary $X_7 = j,\ j = 0, 1$
cc	X_8	Indicates if someone was CCed on the email.	Binary $X_8 = j,\ j = 0, 1$
prior_corr	X_9	Indicates if there has been two way prior correspondence with the sender.	Binary $X_9 = j,\ j = 0, 1$
sender_spam	X_{10}	Indicates if the sender previously has sent spam.	Binary $X_{10} = j,\ j = 0, 1$
format	X_{11}	Indicats if the email contain any special formatting, such as bolding, tables.	Binary $X_{11} = j,\ j = 0, 1$
exclamation	X_{12}	Numerical variable of how many exclamation marks were used.	Numerical X_{12}

Table 1: *Table with brief descriptions of the variables given in the data set.*

3.2.1 Reasoning behind obtaining data

The table on the previous page makes brief definitions of the explanatory variables as well as the response variable. Inspiration for the selection of variables was found in [4].
When collecting for observations, we use the definition presented in the following subsection for deciding if an email is spam. Also, since most of the emails in the data set are in Swedish, we allow for all possible translations of the variables $X_1 \cdots X_4$. For example, if an email contains the word "kr", or "SEK", then the variable X_1 is marked as $X_1 = 1$ for that observation. The same reasoning is used for all the variables in the data set.

3.2.2 Explanation of variables

Since data is collected manually it is important to give clear definitions of each variable. The following subsection will explain each variable in depth:

Outcome; Tells if the email in question of in fact *spam* or *no-spam*. What can be defined as spam? The definition we go with in this report is the same one as mentioned in the introduction; *spam*, is electronic junk mail. Spam emails are most commonly trying get the recipient to buy some product or service, see [10] in list of references. See section 3.2.4 on how to determine if an email is spam.

dollar; Indicates if the email contains the word "dollar". The same reasoning goes for any possible translation of dollar to another currency. Since most of the emails in the data set are in Swedish, words like "SEK" or "kr" will be more encountered more often. If the email contains for example "SEK", then the variable X_1 is marked as $X_1 = 1$ for that observation.

winner; Indicates if the email contains the word "winner". Any possible translation of the word winner will also mark the variable as $X_2 = 1$. Sometimes spam emails are trying to lure recipients into thinking they won something, with the underlying incentive of making money. Therefore we wish to test if an email containing the word *winner* alters the probability of Y.

password; Indicates if the email contains the phrase "password". The same reasoning applies to this variable: possible translations of password will also mark the variable as $X_3 = 1$. Spam sometimes tries to get recipients passwords, which is the reason why we wish to test this variable.

inherit; Indicates if the email contains the word "inherit" or "inheritance". A common spam email fraud is a false promise of inheritance. The email may say that a long-lost relative of the recipient has died and they are the only heir. We want to test if the word inherit can be linked to spam. As we will see in later when presenting different models, observations with $X_4 = 1$ are scarce and we therefore we can't make accurate assumptions regarding this variable.

re_subject; Indicates if "RE" was included in the start of the email. "RE" is for *replies*. If you reply to an email, "RE" will be inserted in from of the original message. So *if* an observation is marked with $X_5 = 1$ the email is a response from someone and it is unlikely to be spam.

attach; Indicates if there is something attached to the email. It can be files such as images or document. We want to test if emails attached with files are more or less likely to be spam emails.

multiple_sent; Indicates of the email was sent to more than one person. Spam emails are usually sent multiply to many people, but non-spam can also be sent to multiple people. We want to test if this has an effect on the probability of an email being spam.

cc; If someone is CCed in the email, the variable X_7 is marked as $X_7 = 1$ for that observation. Presumably, if someone is CCed in the email it in not likely to be a spam email.

prior_corr; Indicates if there has been two way prior correspondence with the sender. For instance, if you sent an email to *Carl@example.com* and Carl replied, then the variable takes on the value $X_9 = 1$. Most likely, if you have had prior correspondence with someone they will not sent you a spam email. Although, sometimes people's email accounts are hacked and used to send spam emails to the contacts. A spam of that kind could be that the email contains a story about the sender being robbed on vacation and he is asking you to wire money. Therefore we wish to test if prior correspondence alters the probability of Y.

sender_spam; Indicates if the sender previously sent spam emails. Most spam that settle in an inbox is spam from senders that already sent you spam. Electronic flyers with advertisements for example. It is likely that if the sender already sent you spam then the email in question is also spam.

format; Indicates of the email has any special formatting. It could be anything such as bold text, tables, images etc. If the email only contains "normal" texting then the variable X_11 is marked with $X_11 = 0$.

exclamation; States how many exclamation marks there are in the email. This is the only numerical variable. We wish to test if more exclamation marks increases the probability of an email being spam. Sometimes advertisements contain many exclamation marks to make the add sell more.

3.2.3 Variables with few observations

In some of the predictors there are very few observations for one of the levels. This causes a problem with too big standard errors. The variable *inherit* has to few observations in the category $X_3 = 1$. Hence, accurate estimations cannot be made with this variable and it is therefore excluded from the models.

3.2.4 Determining if an email is Spam

When collecting for data, we need to have a clear definition of which emails should be encoded with *spam*. The main definition of spam used in this paper is emails that are *unsolicited*. However, if for example a long-lost relative sends you an email it is not considered spam, even though it is unsolicited. Spam is generally in the form of email advertisement.
For the data set used in this paper, an observation is marked as *spam* if the email in question is *unsolicited*, with the exceptions such as the one mentioned above. When sticking to this definition, deciding for if an observation is spam is self-evident.
An example of what is perhaps the most common spam email is electronic flyers. They are *unsolicited* and not personal. Therefore they are considered as spam in this paper. It is important to emphasize that determining for if an email is spam is *not* done by the predictor variables.

3.2.5 Summary of data

A brief summary of the data set is given in table 2 below.

Table 2: **Summary of the observations in the data set**

Data Set Name	WORK.IMPORT	
Observations	353	
Variables	14	
Outcome=Spam		
Y	Frequency	Percent
0	154	43.63
1	199	56.37

For more tables, see Appendix A.

4 Modelling the Probability

We begin by looking at the 12 variables presented in table 1. The aim is to build a model that can predict the probability of an email being *spam*. The selected model should be able to predict data *well*, while still being simple and easy to interpret. We will use a few different methods to obtain a handful of models, which we will compare and analyse. When using *purposeful selection*, presented in chapter 2.3, we get the model presented in the following section.

4.1 First model

Note that sometimes we use limits such as 0.1 instead of 0.05 when doing *purposeful selection*.

Step 1: In the first step we examine each of the independent variables individually. Since the data set is quite large, most of the variables were shown to be significant on the 0.25 level. The only two factors with a $p-value$ over 0.25 are *re_sub* and *inherit*.

Step 2: We now include all the variables with a p-value of less than 0.25 in a multivariate logistic regression model. The result is presented in table 3. We call this "large" model M_0.

Table 3: **Result of the multivariate logistic regression model with the 10 remaining variables from *Step 1*.** Denote this model with M_0

Parameter	estimate	std. error	p-value
intercept	-5.3914	1.4477	0.0002
dollar	1.8593	1.3330	0.1631
winner	5.6806	2.9548	0.0545
password	0.9230	1.2683	0.4668
re_sub	-10.2969	270.3	0.9696
attach	1.1980	1.7859	0.5023
mult_sent	5.2588	1.3256	< 0001
cc	3.1767	2.2839	0.1643
prior_corr	-6.8900	1.8573	0.0002
sender_spam	5.3473	1.3256	< 0001
format	0.9292	0.9567	0.3314
exclamation	-0.2721	0.2162	0.2851

We look at the significance level of each predictor variable of M_0 and exclude all the variables on a 0.1 significance level. The parameters that make the 0.1 mark will create a new model. As presented in table 4, the new smaller model only contains 4 variables.

Table 4: **Result of the multivariate logistic regression model after excluding variables in *Step 2*.** Denote this model with M_1

Parameter	estimate	std. error	p-value
intercept	-4.3722	1.1811	0.0002
winner	5.3234	2.9167	0.0680
mult_sent	5.2701	1.2132	< 0001
prior_corr	-5.5903	1.1678	0.0002
sender_spam	5.4588	1.2014	< 0001

We compare M_0 and M_1 by performing a likelihood ratio test:

$$G^2(M_1) = 2(l_0 - l_1) = 2(441,6 - 435,3) = 12.6 > \chi^2_{0.05}(df = 4) = 9.49$$

This indicates that we can reject the hypothesis of " M_1 holds".

Step 3: We now compare the parameter estimates of M_0 and M_1. By looking at the tables we can see that the variable that differs the most is *prior_corr*. We test its $\Delta\hat{\beta}_i = |(\hat{\theta}_i - \hat{\beta}_i)/\hat{\beta}_i| = \frac{(-5.5903+6.89)}{6.89} = 0.1886 > 0.2$ and draw the conclusion that it does not differ too much. Since *prior_corr* is the variable that differed the most of the 4 variables in M_1, and it did not differ too much, one can draw the conclusion that no estimate differs too much.

Step 4: We now revisit the variables that were excluded from M_0 in *step 1* and plug them into M_1 one at a time. No large deviations were found when adding the variables *re_sub* and *inherit* to M_1. The parameter estimates stay roughly the same.

Step 5: Firstly, we note that M_1 only contains binary variables. We see that the factors *winner*, *mult_sent* and *sender_spam* all have positive parameter estimates. In logistic regression, a positive coefficient corresponds to a positive association between the response variable and the predictor variables. The estimated coefficient of *winner* is 5.3234. Suppose we use M_1 as the software for a spam filter and an incoming email contains the word "winner", the positive coefficient indicates that the incoming email has an increased probability of it being spam, which seems reasonable.
Similar interpretation goes for *mult_sent* and *sender_spam*. It is to expect that if the email was sent to more people than you it increases the probability of it being spam. Also, if the sender previously sent spam, it is not unreasonable that a new email from that same sender is also likely to be spam.
The factor *prior_corr* is the only variable with a negative coefficient. It means that if you receive an email from someone, with whom you have had previous two way email correspondence with, then it lowers the probability of the email being spam.

Step 6: We now consider all possible interaction terms between variables in

M_1 and M_0. We add them to the model one by one and test their significance. As three-way (or higher) interactions are generally hard to motivate in realistic terms, we stick to two-way interactions only.
We find that, out of the $\binom{10}{2} = 45$ possible two-way interaction terms, the only two interactions that are fairly significant are *dollar*password* and *dollar*format*, with p-values 0.1212 and 0.6998 respectively. Since *dollar*format* is the only one with $p - value$ 0.1 we repeat *step 2* by fitting it in the model. We have now created a model with one interaction term, presented in table 5, where all factors are significant on the 0.1 level[6].

Table 5: **Model M_1 after adding the interaction term *dollar*format* *Step 2.*** Denote this model with M_{11}

Parameter	estimate	std. error	p-value
intercept	-4.2914	1.1386	0.0002
winner	5.1466	3.0088	0.0872
mult_sent	4.7618	1.1862	< 0001
prior_corr	-5.5648	1.2436	0.0002
sender_spam	4.8708	1.2192	< 0001
*dollar*format*	1.4485	0.8953	0.1017

In order for an interaction term to be included in a model, it must be motivated both mathematically and realistically. As we see in table 5, *dollar*format* is significant on the 0.1 level[5]. Also, even if *dollar* and *format* are insignificant independently, we can still consider their interaction to be significant. If an email contains special formatting such as boldings, images etc, and that same email contains the word dollar (or *SEK*, *kr*, \$) , then chances are that email is an electronic flyer. In this report, electronic flyers are considered spam, with the motivation that they are trying to get the recipient to buy a product or service.
So even if an email contains the word dollar, *or* has special formating, it does not alter the probability of the email being spam. Their joint effect however, an email that both has special formatting and contains the word dollar, seem to have some effect on the probability of the email being spam.

Step 7: After the 6 previous steps we arrive at the model M_{11} as *the preliminary final model*. Before testing how well M_{11} can explain variation in data, we consider other models.

4.2 Alternative ways of model selection

We want to build a suitable model for data given the explanatory variables in table 1. The method used in the following section is "stepwise selection" in SAS[7]. The different methods are explained below:

Backward selection: We begin with a logistic regression model containing all the explanatory variables. Then make a hypothesis test to see which variables are significant on the 10% level. The variables that are least significant are removed from the model, this procedure is repeated until all the variables in the model are significant on the 10% level.

Forward selection: Here we start in the other end, meaning we begin with a model that only contains the intercept. Variables are added to the model, one at a time, and tested on the 10% level. The variables that are significant stay in the model and the other variables are not added to the model. Repeat this procedure until no more significant variables can be added.

Stepwise selection: is a mix of the two previous selections methods. We can for example choose to begin with a model that only contains the intercept. Then you add the most significant variables to the model. Thereafter we remove the variables that do not make the 10% cut off. These two steps are repeated until no more variables can be added or removed from the model.

For the model selection above, the statistical software SAS was used for the calculations. We used all the explanatory variables individually as well as two way interaction terms between all of them. The reason for not using higher order interactions is because they are hard to interpret. We found that *Stepwise selection* and *Forward selection* gave the same final model, while *Backward selection* gave a slightly different model.

Type of selection, final model and maximum likelihood estimates for the variables are presented below.

4.2.1 Backward selection

When performing backward selection we use the command *PROC logistic* in SAS and choose $slstay = 0.10$. This tells SAS to only keep variables significant on the 10%-level. The model that SAS gave is presented in table 6 below.

Table 6: **Final model from *Backward selection.*** Denote this model with M_{BWD}

Parameter	estimate	std. error	p-value
intercept	-4.3722	1.1811	0.0002
winner	5.3234	2.9167	0.0680
mult_sent	5.2701	1.2132	< 0.001
prior_corr	-5.5903	1.1678	< 0.001
sender_spam	5.4588	1.2014	< 0.001

4.2.2 Forward and Stepwise selection

As mentioned above, Forward selection and Stepwise selection both obtained the same final model. In SAS, the command *PROC logistic* was used with $slstay = 0.10$ and $slentry = 0.10$. SAS will only keep variables that makes the 10%-significance level and will only exclude variables over the 10%-level. The final model is presented in table 7 below.

Table 7: **Final model from *Forward selection* and *Stepwise selection.*** Denote this model with M_{FWD}

Parameter	estimate	std. error	p-value
intercept	-4.9761	1.2237	0.0002
winner	5.1976	2.8353	0.0668
mult_sent	5.0466	1.1752	< 0.001
prior_corr	-5.3507	1.2429	< 0.001
sender_spam	5.1821	1.1951	< 0.001
format	1.2952	0.7900	0.1011

4.3 Low number of parameters

The goals of model selection is to build a simple and interpretable model that can make accurate predictions. A model with few parameters is simple and easy to interpret. Since the previous model selection methods has provided models that have 4 or more parameters we want to find an alternative to the previous models, but with less parameters. In SAS, the command SELECTION=SCORE uses an algorithm to find models with the highest likelihood score (chi-square) statistic for a desired model size. When using this command, while setting the model size to 3 parameters, we get the following model:

Table 8: **Model with highest chi-squared statistic containing 3 parameters.** Denote this model with M_{small}

Parameter	estimate	std. error	p-value
intercept	-2.6673	0.7063	0.0002
prior_corr	-5.3473	0.9209	< 0.001
mult_sent	4.3626	0.8174	< 0.001
*sender_spam*format*	5.8168	0.9192	< 0.001

4.4 Comparing between models

We now have 4 preliminary final models. The model that was obtained by performing *purposeful selection*, the two models that were developed in SAS by using Forward and Backward selection, and simple model with only 3 parameters. We will now determine how well the different models perform when testing for *ability to describe data* and for *predictive capacity*.

Hosmer Lemeshow test is performed on all models by following the procedure described in section 2.4.2.

As presented in the table below, three out of the four tested models were significant. We want a simple model with a high p-value for the Hosmer Lemeshow

Table 9: **Hosmer Lemeshow test results for all models.**

Hosmer Lemeshow	M_{11}	M_{BWD}	M_{FWD}	M_{small}
χ^2-value	8.1948	15.4521	20.5677	13.1570
p-value	0.2242	0.0170	0.0022	0.0105

statistic. The model M_{11} seems to perform better than the other models when looking at the p-values presented in table 9.

A *Reciever operating curve* is fitted to the models and the area under the curve (AUC) is calculated. The results are presented in figure 2 and figure 3 in appendix A. Notably, the models differ very little, and it is only in the third decimal place. Moreover, all the models have a $AUC > 0.9$, which is the limit for *Outstanding* according to Hosmer and Lemeshow, see section 2.4.4.

Generalized R^2 is calculated for each model individually. When looking at

Table 10: **The R^2-measured for all models.**

Type of R^2	M_{11}	M_{BWD}	M_{FWD}	M_{small}
R^2_{sas}	0.7110	0.7086	0.7108	0.6934
R^2_{adj}	0.9532	0.9500	0.9530	0.9296

table 10 it appears that model M_{11} seems to perform slightly better than the other models. Note that since the adjustment between R^2_{sas} and R^2_{adj} is independent of the model selection, comparisons based on R^2_{sas} and R^2_{adj} will surely give the same result.

In table 11 below, the results are summarized together with the *AIC* value of each model. It shows that model M_{11}, which was obtained through *purposeful selection*, performs slightly better than the other models in *all* tests. However, this is not unexpected since it has the most complex structure of the models, with 5 parameters, of which one is a two-way interaction term.

Table 11: **Summary of model diagnostics**

	M_{11}	M_{BWD}	M_{FWD}	M_{small}
R^2_{sas}	0.7110	0.7086	0.7108	0.6934
R^2_{adj}	0.9532	0.9500	0.9530	0.9296
AUC	0.9969	0.9954	0.9966	0.9908
AIC	57.476	58.344	57.640	74.284
Hosmer Lemeshow p-value	0.2242	0.0170	0.0022	0.0105

Since model M_{11} doesn't perform drastically better than the other three models, the question becomes whether the complexity outweighs the simplicity in one of the simpler models. All tests give similar results for all models, which speaks for selecting a simpler model. Although, the structure in model M_{11} is not very different from the other models. It has the same amount of parameters as M_{FWD}, and even model M_{small}, which is supposed to be the simplest model, contains a two-way interaction term.

Considering we are building a model for spam filtering, we want to eliminate the possibility of the model letting in spam or junking non-spam. Also, since all the models have relatively similar structure, we decide to go with the model that performed best in all the diagnostics-tests: M_{11}. Moreover, M_{11} was the only model that wasn't significant in the Hosmer Lemeshow test, which is an additional reason for choosing it as our final model.

4.4.1 Selected model

Now that we have selected model M_{11} as our final model we calculate the odds ratio for the variables in table 5. See section 2.2 on how to calculate odds ratios. The results can be found in appendix A, figure 1. When looking at the table of the odds ratios estimates, we can see that the odds ratio increases by 17083.9% if the email contains the word winner compared to if it doesn't. We also see that if the email has been sent to more people it increases the odds ratio by 11595.5% compared to if the email was only sent to one person. Moreover, the odds ratio also increases by 129.419% if the sender previously sent spam, compared to if the sender never sent spam.

The most influential variable in model M_{11} is *prior_ corr*. From figure 1 we can draw the conclusion that if you have had two-way correspondence with the sender, then the odds of the email being spam is decreased by 99.6% compared to if you haven't had two-way correspondence with the sender.
All the point estimates for the odds ratios in model M_{11} are *very* significant in some direction (depending on which variable). This is because M_{11} only contains strong and influential variables. Consider for example *prior_ corr*: realistically it is *highly* unlikely that a person with whom you have had email correspondence with would sent you a spam email. This may occur only if for example the senders email was hacked, or if the sender is a company that you have had correspondence with that uses the same email address to sent electronic flyers, etc. Table 12 is a contingency table for spam and the variable *prior_ corr*. When looking at the 139 emails where there was correspondence with the sender, only 13 were spam.

Table 12: **Table of Spam frequencies by *prior_ corr***

	Prior correspondence		
	no	yes	Total
Spam	186	13	199
not spam	28	126	154
Total	214	139	353

Going back and looking at those observations in the data set, 12 were shown to be electronic flyers from a company and one was a spam email from a hacked address. This suggests that the variable *prior_ corr* is very effective at distinguishing if an email is spam or not. This variable alone would make accurate predictions about emails in a spam filter.

The results after performing Cross Validation on model M_{11} are presented in figure 4 in appendix A. Note that the estimated cross validated area is smaller than the area under from the original ROC curve. As presented in the table, the AUC drops from 0.9969 to 0.9938 when cross validation is used. The small drop indicated that the model M_{11} predicts data well when using cross validation.

5 Conclusion

The purpose of this paper was to build a model that can be used as a spam filter. After analysing data, a final model was obtained through *purposeful selection*. Based on the results of the previous section we see that all the tested models have a good predictive capacity. Most values in table 11 suggest that all the models seem to fit data well. Considering the ROC plots, that showed that all models have an *outstanding* predictive ability, we cannot reject the possibility to use one of the models for prediction purposes in a spam filter. Also interesting is how little the models differ, considering they have different structures. The chosen model M_{11}, which is considered to have a more complex structure than the other models, wins by *very* little compared to smaller models. We still chose M_{11} as our final model because the simplicity of a smaller model does not outweigh the fact that M_{11} is the only model that is not significant in the *Hosmer Lemeshow* test.
Having chosen model M_{11} as our final model for a spam filter we can make some conclusion on what alters the probability of an email being spam. We see that in order maximise the probability of an email being spam it should contain the word *winner*, it should be sent to more people and the sender should have previously sent you spam. Also, if the email has special formatting *and* has contains the word dollar, it also increases the probability. For an email not to be spam we need to look at if there has been prior two-way correspondence with the sender. If so, then the probability of the email being spam is decreased drastically.

6 Discussion

All models are simplifications of reality, to put in words of George E.P Box [3], *essentially, all models are wrong, but some are useful*. There is always a risk that a model is too much simplified and makes false predictions.
Seemingly, all models that were built in section 4 show good ability to predict data. Out of the four models M_{11} performs best and is therefore chosen as the final model. The diagnostic tests suggest that M_{11} fits data very well. Although, it is not flawless and could make false predictions.
If a model makes false predictions in a spam filter two things can go wrong: it can let in spam into the in-box, or it can junk regular emails. The latter error is clearly worse than the first error. If the filter would let i a spam email from time to time it would not be a big disaster. On the other hand, a spam filter that junks an important email is not desirable. For example, if an email is estimated to be spam with a 75% probability by the filter, then it is probably wiser to let that email into the mail box than to junk it. This makes sure that if that email was in fact not spam, then it ends up in the in-box. Even though it was more likely to be spam we still choose to take the risk because the consequence of letting in a spam email is not huge, while junking a normal email could lead to serious consequences for the user. Therefore it is recommended to only have

the filter junk emails that are estimated to be spam with a 95% probability or higher. This strategy makes sure as good as all regular emails end up in the in-box, with the downside of letting in a few spam emails once in a while.

Another aspect to consider is that M_{11} is built from data that is collected from my personal in-box, and may therefore not work for everyone. A spam filter should in theory work for any email address, regardless of who the user is. When building a spam filter for an email service that manages many accounts, more time should be spent on adding additional variables. One could also use transformations to help include skewed variables into the logistic regression model.
An example is an indicator variable that flags an email which contains a link that has been included in previous spam emails. The variable is marked as 1 if such a link is found and makes as 0 if not.
In order to utilize this predictor variable, a data base that holds links found in spam emails would be needed. In this report, access to such information is limited. Therefore we could not implement this variable when building a model.

In addition to adding better predictive variables, building a separate logistic regression model for each mail account would build an improved spam filter. The model would be customized for the emails in each persons in-box, respectively.

For what seemed to be an extremely challenging task of classifying spam emails, we have made very good progress in the field. Simple email characteristics, such as inclusion of the word dollar, the formatting, and other variables, provide useful information for spam classification. Many improvements can be made, from including more variables and having a better data set to performing the necessary programming to make a logistic model into a filter. Completing such tasks is conceivable, and by doing so, one could build an exceptional spam filter.

References

[1] Alan Agresti and Maria Kateri. *Categorical data analysis*. Springer, 2011.

[2] Nikhila Arkalgud. Logistic regression for spam filtering. 2008.

[3] George EP Box. Science and statistics. *Journal of the American Statistical Association*, 71(356):791–799, 1976.

[4] Mine Çetinkaya-Runde David M Diez, Christopher D Barr. Logistic regression, 2014.

[5] Mary B. Barton Ernest S. Shtatland, Sara Moore. Why we need an r2 measure of fit in proc logistic, 2012.

[6] Joshua Goodman and Wen-tau Yih. Online discriminative spam filter training. In *CEAS*, pages 1–4, 2006.

[7] David W Hosmer Jr and Stanley Lemeshow. *Applied logistic regression*. John Wiley & Sons, 2004.

[8] Martina Mittlböck, Michael Schemper, et al. Explained variation for logistic regression. *Statistics in medicine*, 15(19):1987–1997, 1996.

[9] PennState. Multinomial logistic regression models. *Analysis of Discrete Data*, 2016.

[10] Margaret Rouse. Ube, unsolicited bulk email. *TechTarget*, 2006.

[11] SAS.com. Roc analysis using validation data and cross validation, 2012.

[12] D Sculley and Gordon V Cormack. Filtering email spam in the presence of noisy user feedback. In *CEAS*. Citeseer, 2008.

Notes

[1]A bulk email, is a formal term for *spam*. The most common type of UBE is unsolicited commercial email (UCE): bulk email that is trying to get the recipient to buy some product or service. [10]

[2]Additional references on spam filtering are [2] and [12]

[3]GLM-generalized linear model

[4]See *Agresti* 2002 page 79

[5]see page 2 of [5], found in list of references.

[6]Note that the p-value of dollar*format is 0.1017, which is just above 0.1. We still choose to keep it in the model.

[7]SAS is the statistical software that is used throughout the this report, [9]

A SAS Printouts

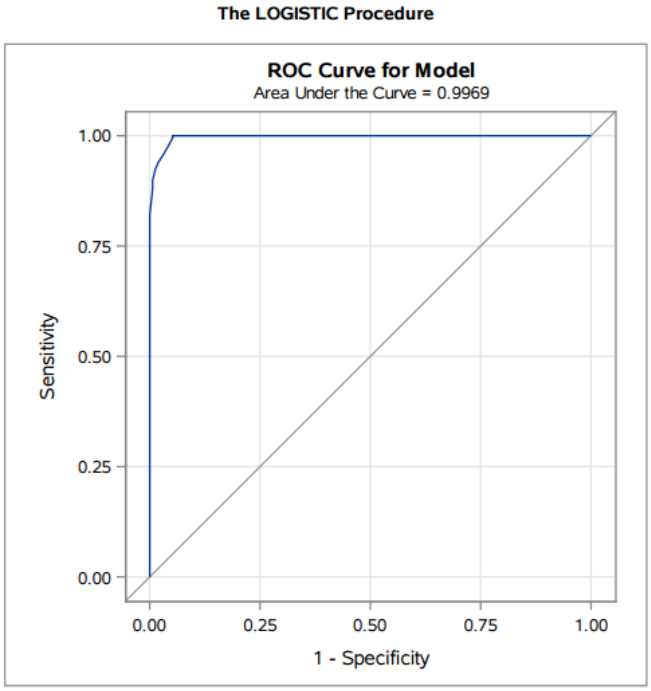
Figure 1: **Table showing parameter estimates and odds ratio estimates of model M_{11}.**

The LOGISTIC Procedure

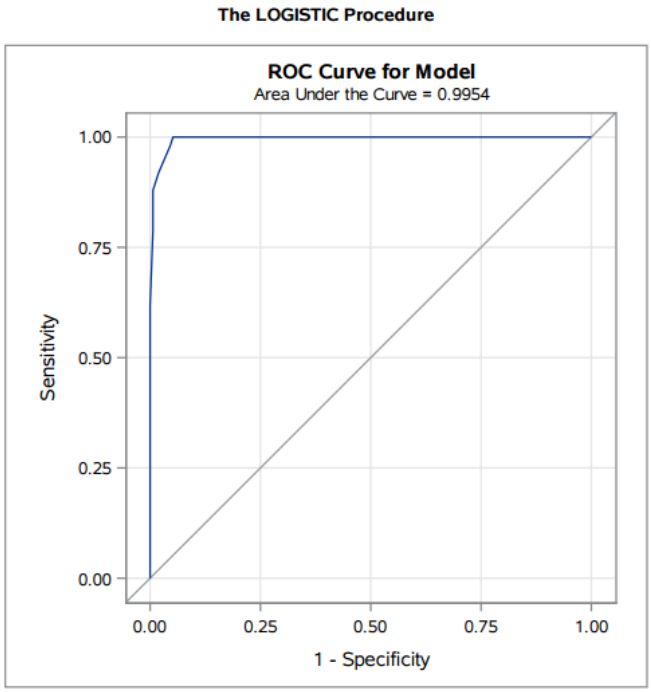
Analysis of Maximum Likelihood Estimates					
Parameter	DF	Estimate	Standard Error	Wald Chi-Square	Pr > ChiSq
Intercept	1	-4.2914	1.1386	14.2053	0.0002
winner	1	5.1466	3.0088	2.9257	0.0872
mult_sent	1	4.7618	1.1862	16.1153	<.0001
prior_corr	1	-5.5648	1.2436	20.0249	<.0001
sender_spam	1	4.8708	1.2192	15.9602	<.0001
dollar*format	1	1.4485	0.8953	2.6177	0.1057

Odds Ratio Estimates			
Effect	Point Estimate	95% Wald Confidence Limits	
winner	171.839	0.472	>999.999
mult_sent	116.955	11.438	>999.999
prior_corr	0.004	<0.001	0.044
sender_spam	130.419	11.955	>999.999

Figure 2: **Graph of ROC curves for models M_{11} and M_{BWD}.**

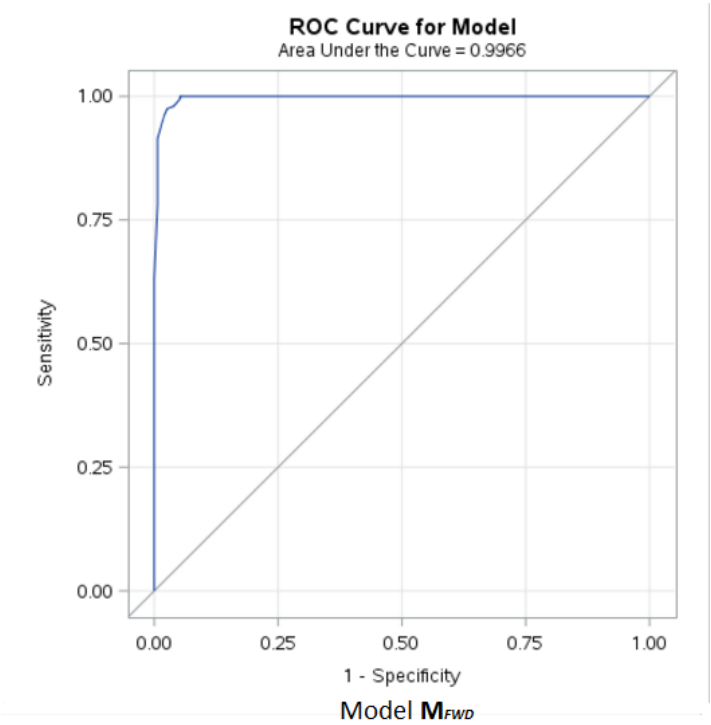


Model M_{11}



Model M_{BWD}

Figure 3: **Graph of ROC curves for models M_{FWD} and M_{small}.**



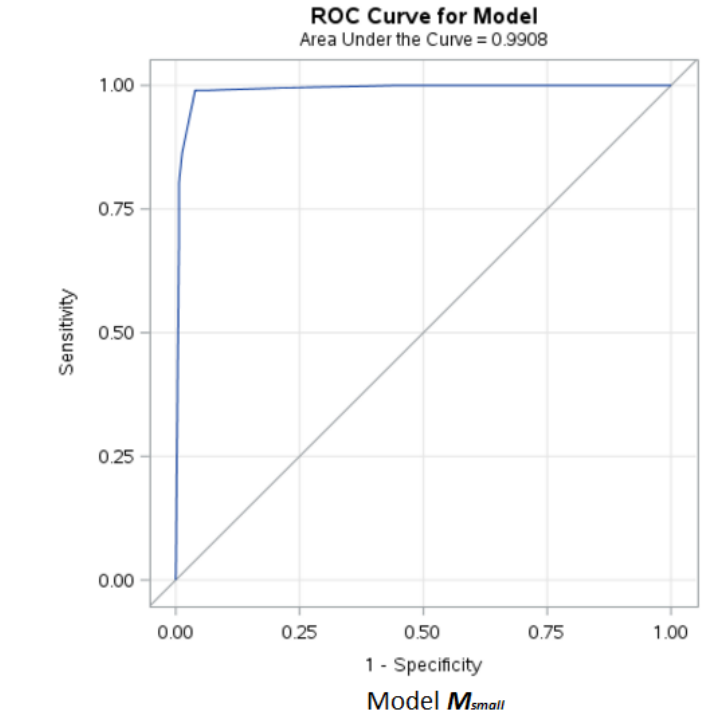
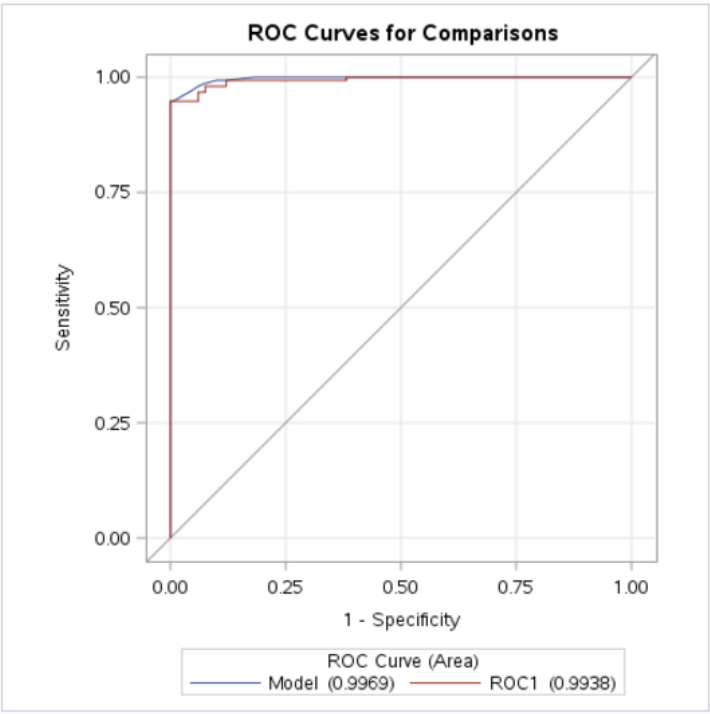


Figure 4: **Graph of ROC curves for testing Cross validation on model M_{11}.**



ROC Association Statistics							
	Mann-Whitney						
ROC Model	Area	Standard Error	95% Wald Confidence Limits		Somers' D	Gamma	Tau-a
Model	0.9969	0.00147	0.9940	0.9998	0.9938	0.9954	0.4902
ROC1	0.9938	0.00299	0.9879	0.9997	0.9876	0.9876	0.4872

ROC Contrast Test Results			
Contrast	DF	Chi-Square	Pr > ChiSq
Reference = Model	1	3.5523	0.0595

B Tables

This sections contains a number of tables presenting the data set.

Table 13: **Table of Spam frequencies by *winner***

	Winner		
	no	yes	Total
not spam	147	7	154
spam	58	141	199
Total	205	148	353

Table 14: **Table of Spam frequencies by *dollar***

	Dollar		
	no	yes	Total
not spam	143	11	154
spam	72	127	199
Total	215	138	353

Table 15: **Table of Spam frequencies by *attach***

	attach		
	no	yes	Total
not spam	101	53	154
spam	197	2	199
Total	298	55	353

Table 16: **Table of Spam frequencies by *sender_spam***

	sender_spam		
	no	yes	Total
not spam	130	24	154
spam	21	178	199
Total	151	202	353

Table 17: **Table of Spam frequencies by *password***

	password		
	no	yes	Total
not spam	130	24	154
spam	156	43	199
Total	286	67	353

Table 18: **Table of Spam frequencies by *inherit***

	inherit		
	no	yes	Total
not spam	154	0	154
spam	198	1	199
Total	352	1	353

Table 19: **Table of Spam frequencies by *cc***

	cc		
	no	yes	Total
not spam	112	42	154
spam	199	0	199
Total	311	42	353

Table 20: **Table of Spam frequencies by *format***

	format		
	no	yes	Total
not spam	120	34	154
spam	15	184	199
Total	135	218	353

Table 21: **Table of Spam frequencies by *mult_sent***

	mult_sent		
	no	yes	Total
not spam	113	41	154
spam	25	174	199
Total	138	215	353