Learning from data

Classification

February 3, 2020

The story so far...

- **Setup:** a dependent (response) variable, and a set of independent (predictor) variables.
- Fit a linear regression model to predict the dependent (response) variable.

The story so far...

- **Setup:** a dependent (response) variable, and a set of independent (predictor) variables.
- Fit a linear regression model to predict the dependent (response) variable.

There's catch: Above procedure cannot be implemented whenever the dependent (response) variable is a qualitative variable.

The story so far...

- **Setup:** a dependent (response) variable, and a set of independent (predictor) variables.
- Fit a linear regression model to predict the dependent (response) variable.

There's catch: Above procedure cannot be implemented whenever the dependent (response) variable is a qualitative variable.

Why? Simply because the predictors are all numerical, and we cannot expect a qualitative prediction from the linear relationship of numerical predictors.

The story so far...

- **Setup:** a dependent (response) variable, and a set of independent (predictor) variables.
- Fit a linear regression model to predict the dependent (response) variable.

There's catch: Above procedure cannot be implemented whenever the dependent (response) variable is a qualitative variable.

Why? Simply because the predictors are all numerical, and we cannot expect a qualitative prediction from the linear relationship of numerical predictors. So, why not encode the qualitative variables as numbers.

The story so far...

- **Setup:** a dependent (response) variable, and a set of independent (predictor) variables.
- Fit a linear regression model to predict the dependent (response) variable.

There's catch: Above procedure cannot be implemented whenever the dependent (response) variable is a qualitative variable.

Why? Simply because the predictors are all numerical, and we cannot expect a qualitative prediction from the linear relationship of numerical predictors. So, why not encode the qualitative variables as numbers. There are some technical problems.

Predicting a qualitative variable

Predicting a qualitative variable

- Credit card companies often have to predict if the lender is going to default on her/his payments based on the information about variables like individual's income, past history of loan repayments, socio–economic status, etc.

Predicting a qualitative variable

- Credit card companies often have to predict if the lender is going to default on her/his payments based on the information about variables like individual's income, past history of loan repayments, socio–economic status, etc.

- Doctors often have the task to arrive at diagnosis using predictors like body temperature, blood pressure, and several other measurements. For instance, using predictors like above to diagnose if the individual suffered a stroke, or overdosed on a drug, or suffered an epileptic seizure.

Predicting a qualitative variable

- Credit card companies often have to predict if the lender is going to default on her/his payments based on the information about variables like individual's income, past history of loan repayments, socio–economic status, etc.

- Doctors often have the task to arrive at diagnosis using predictors like body temperature, blood pressure, and several other measurements. For instance, using predictors like above to diagnose if the individual suffered a stroke, or overdosed on a drug, or suffered an epileptic seizure.

Encoding a qualitative variable as a numerical variable has unintentional effects, like ordering of the categories.

Predicting a qualitative variable

- Credit card companies often have to predict if the lender is going to default on her/his payments based on the information about variables like individual's income, past history of loan repayments, socio–economic status, etc.

- Doctors often have the task to arrive at diagnosis using predictors like body temperature, blood pressure, and several other measurements. For instance, using predictors like above to diagnose if the individual suffered a stroke, or overdosed on a drug, or suffered an epileptic seizure.

Encoding a qualitative variable as a numerical variable has unintentional effects, like ordering of the categories.

Leads us to the classification setting.

What are we up to?

- n individuals – each is associated with certain quantitative measurements called the predictors or features, say the p measurements of i-th individual are given as $(x_{i1}, x_{i2}, \ldots, x_{ip})$

What are we up to?

- n individuals – each is associated with certain quantitative measurements called the predictors or features, say the p measurements of i-th individual are given as $(x_{i1}, x_{i2}, \ldots, x_{ip})$
- Each individual is also categorised using a qualitative variable: dependent variable

What are we up to?

- n individuals – each is associated with certain quantitative measurements called the predictors or features, say the p measurements of i-th individual are given as $(x_{i1}, x_{i2}, \dots, x_{ip})$
- Each individual is also categorised using a qualitative variable: dependent variable
- Goal: given predictor measurements $(x_1, x_2, \dots, x_p)$ of certain individual, we wish to model y, the category this individual belongs to – Classification problem

A mathematician's retort

- Usual linear regression type of models will invariably fail. (Why?)

A mathematician's retort

- Usual linear regression type of models will invariably fail. (Why?)
- Modelling a function which takes only finitely many values is not difficult.

A mathematician's retort

- Usual linear regression type of models will invariably fail. (Why?)
- Modelling a function which takes only finitely many values is not difficult. For example, if y takes only two values 0 or 1, then we can set:

$$y = 1_{\{(x_1,\ldots,x_p)\in B\}}$$

or, in general, if y takes J distinct values $(a_1, \ldots, a_J)$ then we can set

$$y = a_1 \, 1_{\{(x_1,\ldots,x_p)\in B_1\}} + \cdots + a_J \, 1_{\{(x_1,\ldots,x_p)\in B_J\}}$$

where $B_1, \ldots, B_J$ are disjoint subsets of $\mathbb{R}^p$.

A mathematician's retort

- Usual linear regression type of models will invariably fail. (Why?)
- Modelling a function which takes only finitely many values is not difficult. For example, if y takes only two values 0 or 1, then we can set:

$$y = 1_{\{(x_1, \ldots, x_p) \in B\}}$$

or, in general, if y takes J distinct values $(a_1, \ldots, a_J)$ then we can set

$$y = a_1 \, 1_{\{(x_1, \ldots, x_p) \in B_1\}} + \cdots + a_J \, 1_{\{(x_1, \ldots, x_p) \in B_J\}}$$

where $B_1, \ldots, B_J$ are disjoint subsets of $\mathbb{R}^p$.

Issues with the proposed solution:

- y could be a qualitative variables, and thus the above method would fail.
- How does one incorporate statistical error (noise)?

Binary classifier: logistic regression

Binary classifier: the setup

When the dependent (response) variable is a categorical variable which takes only two possible values (binary), there is a more methodical way of building a classifier.

Binary classifier: the setup

When the dependent (response) variable is a categorical variable which takes only two possible values (binary), there is a more methodical way of building a classifier.

Consider the German Credit dataset (different from the “Credit” dataset of Assignment-1).

Binary classifier: the setup

When the dependent (response) variable is a categorical variable which takes only two possible values (binary), there is a more methodical way of building a classifier.

Consider the German Credit dataset (different from the “Credit” dataset of Assignment-1).

Goal is to build a model through which a bank manager can decide whether to approve an applicant’s loan application based on the information on those variables.

Binary classifier: the setup

When the dependent (response) variable is a categorical variable which takes only two possible values (binary), there is a more methodical way of building a classifier.

Consider the German Credit dataset (different from the “Credit” dataset of Assignment-1).

Goal is to build a model through which a bank manager can decide whether to approve an applicant’s loan application based on the information on those variables.

Of those several variables, we focus on the variable called “Credit.Amount” (amount of credit applicant has applied for) as an illustrative example for this analysis.

Binary classifier: the setup

When the dependent (response) variable is a categorical variable which takes only two possible values (binary), there is a more methodical way of building a classifier.

Consider the German Credit dataset (different from the “Credit” dataset of Assignment-1).

Goal is to build a model through which a bank manager can decide whether to approve an applicant’s loan application based on the information on those variables.

Of those several variables, we focus on the variable called “Credit.Amount” (amount of credit applicant has applied for) as an illustrative example for this analysis.

Once again, we divide the dataset into two portions: training and test

Once again, we divide the dataset into two portions: training and test

Let us code “creditability” as 0 or 1 valued with 1 denoting creditable meaning loan is approved, and 0 otherwise.

Once again, we divide the dataset into two portions: training and test

Let us code “creditability” as 0 or 1 valued with 1 denoting creditable meaning loan is approved, and 0 otherwise.

Writing X for the variable “Credit.Amount”, the binary classifier is based on modelling:

$$\mathbb{P}(Y = 1|X = x)$$

- We could model: $\mathbb{P}(Y = 1|X = x) = \beta_0 + \beta_1 x$

Once again, we divide the dataset into two portions: training and test

Let us code "creditability" as 0 or 1 valued with 1 denoting creditable meaning loan is approved, and 0 otherwise.

Writing X for the variable "Credit.Amount", the binary classifier is based on modelling:

$$\mathbb{P}(Y = 1|X = x)$$

- We could model: $\mathbb{P}(Y = 1|X = x) = \beta_0 + \beta_1 x$
- We could model: $\mathbb{P}(Y = 1|X = x) = e^{\beta_0+\beta_1 x}$

We write $p(x) = \mathbb{P}(Y = 1|X = x)$.

We write $p(x) = \mathbb{P}(Y = 1|X = x)$.

Possibly the right quantity to consider is:

$$\log\left(\frac{p(x)}{1 - p(x)}\right)$$

We write $p(x) = \mathbb{P}(Y = 1|X = x)$.

Possibly the right quantity to consider is:

$$\log\left(\frac{p(x)}{1 - p(x)}\right)$$

It's also referred to as **log-odds**, which can take any real value.

We write $p(x) = \mathbb{P}(Y = 1|X = x)$.

Possibly the right quantity to consider is:

$$\log\left(\frac{p(x)}{1 - p(x)}\right)$$

It's also referred to as **log-odds**, which can take any real value.

We therefore, model

$$\log\left(\frac{p(x)}{1 - p(x)}\right) = \beta_0 + \beta_1 x$$

We write $p(x) = \mathbb{P}(Y = 1|X = x)$.

Possibly the right quantity to consider is:

$$\log\left(\frac{p(x)}{1 - p(x)}\right)$$

It's also referred to as **log-odds**, which can take any real value.

We therefore, model

$$\log\left(\frac{p(x)}{1 - p(x)}\right) = \beta_0 + \beta_1 x$$

In other words,

$$p(x) = \frac{e^{\beta_0 + \beta_1 x}}{1 + e^{\beta_0 + \beta_1 x}}$$

Setup: Given a quantitative measurement (dependent or predictor) on an individual, we wish to predict if the individual belongs to one group or the other (binary classifier)

Setup: Given a quantitative measurement (dependent or predictor) on an individual, we wish to predict if the individual belongs to one group or the other (binary classifier)

Model:

$$p(x) = \frac{e^{\beta_0+\beta_1 x}}{1 + e^{\beta_0+\beta_1 x}}$$

Setup: Given a quantitative measurement (dependent or predictor) on an individual, we wish to predict if the individual belongs to one group or the other (binary classifier)

Model:

$$p(x) = \frac{e^{\beta_0+\beta_1 x}}{1 + e^{\beta_0+\beta_1 x}}$$

Goal: Estimate β_0 and β_1

Setup: Given a quantitative measurement (dependent or predictor) on an individual, we wish to predict if the individual belongs to one group or the other (binary classifier)

Model:

$$p(x) = \frac{e^{\beta_0+\beta_1 x}}{1 + e^{\beta_0+\beta_1 x}}$$

Goal: Estimate β_0 and β_1

Method: Maximum likelihood estimation

Maximum likelihood estimation

Note that the model is built on the training dataset, therefore estimation is performed on the training dataset.

Maximum likelihood estimation

Note that the model is built on the training dataset, therefore estimation is performed on the training dataset.

Let us denote the training dataset as: $(x_1, y_1), \ldots, (x_n, y_n)$, where x_i's are predictor values, and y_i are known values of the qualitative variable for the i-th individual.

Maximum likelihood estimation

Note that the model is built on the training dataset, therefore estimation is performed on the training dataset.

Let us denote the training dataset as: $(x_1, y_1), \ldots, (x_n, y_n)$, where x_i's are predictor values, and y_i are known values of the qualitative variable for the i-th individual.

Let $\mathcal{Y}_0$ be the subset of training dataset for which $y_i = 0$, and similarly, $\mathcal{Y}_1$ be the subset of training dataset for which $y_i = 1$.

Maximum likelihood estimation

Note that the model is built on the training dataset, therefore estimation is performed on the training dataset.

Let us denote the training dataset as: $(x_1, y_1), \ldots, (x_n, y_n)$, where x_i's are predictor values, and y_i are known values of the qualitative variable for the i-th individual.

Let $\mathcal{Y}_0$ be the subset of training dataset for which $y_i = 0$, and similarly, $\mathcal{Y}_1$ be the subset of training dataset for which $y_i = 1$.

The likelihood for this training sample is:

$$L(\beta_0, \beta_1) = \prod_{i \in \mathcal{Y}_1} p(x_i) \prod_{i \in \mathcal{Y}_0} [1 - p(x_i)]$$

The maximum likelihood estimates are those $\widehat{\beta_0}$ and $\widehat{\beta_1}$, which maximise $L(\beta_0, \beta_1)$

Maximum likelihood estimation

Finding those $\widehat{\beta_0}$ and $\widehat{\beta_1}$, which maximise

$$L(\beta_0, \beta_1) = \prod_{i \in \mathcal{Y}_1} \frac{e^{\beta_0 + \beta_1 x_i}}{1 + e^{\beta_0 + \beta_1 x_i}} \prod_{i \in \mathcal{Y}_0} \frac{1}{1 + e^{\beta_0 + \beta_1 x_i}}$$

is analytically a difficult exercise.

Maximum likelihood estimation

Finding those $\widehat{\beta_0}$ and $\widehat{\beta_1}$, which maximise

$$L(\beta_0, \beta_1) = \prod_{i \in \mathcal{Y}_1} \frac{e^{\beta_0 + \beta_1 x_i}}{1 + e^{\beta_0 + \beta_1 x_i}} \prod_{i \in \mathcal{Y}_0} \frac{1}{1 + e^{\beta_0 + \beta_1 x_i}}$$

is analytically a difficult exercise.

Remark: Likelihood estimation is a very general approach, applicable to almost every setting of estimation. Least squares estimation in linear regression is a special case of likelihood estimation.

Maximum likelihood estimation

Finding those $\widehat{\beta_0}$ and $\widehat{\beta_1}$, which maximise

$$L(\beta_0, \beta_1) = \prod_{i \in \mathcal{Y}_1} \frac{e^{\beta_0 + \beta_1 x_i}}{1 + e^{\beta_0 + \beta_1 x_i}} \prod_{i \in \mathcal{Y}_0} \frac{1}{1 + e^{\beta_0 + \beta_1 x_i}}$$

is analytically a difficult exercise.

Remark: Likelihood estimation is a very general approach, applicable to almost every setting of estimation. Least squares estimation in linear regression is a special case of likelihood estimation.

Interpretation of $\widehat{\beta_1}$: A unit increase in x, changes the odds by a factor of $e^{\widehat{\beta_1}}$.

Prediction

- Given a new value of x_0 (predictor), we wish to now classify the individual into one of the categories.

Prediction

- Given a new value of x_0 (predictor), we wish to now classify the individual into one of the categories.
- Compute

$$\mathbb{P}(Y = 1|X = x_0) = p(x_0) = \frac{e^{\widehat{\beta_0}+\widehat{\beta_1}x_i}}{1 + e^{\widehat{\beta_0}+\widehat{\beta_1}x_i}}$$

Prediction

- Given a new value of x_0 (predictor), we wish to now classify the individual into one of the categories.

- Compute

$$\mathbb{P}(Y = 1 | X = x_0) = p(x_0) = \frac{e^{\widehat{\beta_0} + \widehat{\beta_1} x_i}}{1 + e^{\widehat{\beta_0} + \widehat{\beta_1} x_i}}$$

- If $p(x_0) > 0.5$ (say), then we classify x_0 into the category $Y = 1$, else into the category $Y = 0$.

Note: The thresholding at 0.5 is our choice. If we have some prior information then we may want to tweak this a little.

Testing: statistical significance of the model

The hypothesis of interest:

$$
\begin{aligned}
H_0 &: \beta_1 = 0 \\
H_1 &: \beta_1 \neq 0
\end{aligned}
$$

- The estimators are tested using the standard Wald's Z-test.
- Or, one could also use the *likelihood ratio (LR)* test.

Testing: statistical significance of the model

The hypothesis of interest:

$$
\begin{aligned}
H_0 &: \beta_1 = 0 \\
H_1 &: \beta_1 \neq 0
\end{aligned}
$$

- The estimators are tested using the standard Wald's Z-test.
- Or, one could also use the *likelihood ratio (LR)* test.

Testing: goodness of the model

- The testing phase includes predicting the classification for given set of x_i's, for which the true classification is already know.
- We count the number of 1's which are correctly identified as 1, and the number of 1's which are incorrectly identified as 0.
- Similarly, we count the number of 0's which are correctly identified as 0, and the number of 0's which are incorrectly identified as 1.

A word of caution

A word of caution

Before we even perform the binary classification, it is important to see if the categorical variable indeed has an impact on the predictor variable.

A word of caution

Before we even perform the binary classification, it is important to see if the categorical variable indeed has an impact on the predictor variable.

For instance, if the task is to determine whether an individual patient suffered a stroke or not, given only the measurement of body temperature. It is important to first find out if there's any statistically significant effect of stroke on body temperature.

A word of caution

Before we even perform the binary classification, it is important to see if the categorical variable indeed has an impact on the predictor variable.

For instance, if the task is to determine whether an individual patient suffered a stroke or not, given only the measurement of body temperature. It is important to first find out if there's any statistically significant effect of stroke on body temperature.

We apply the standard 2-sample t-test.

German Credit Database

The variables:

```
> names(G.credit)
 [1] "Creditability"                   "Account.Balance"
 [3] "Duration.of.Credit..month."      "Payment.Status.of.Previous.Credit"
 [5] "Purpose"                         "Credit.Amount"
 [7] "Value.Savings.Stocks"            "Length.of.current.employment"
 [9] "Instalment.per.cent"             "Sex...Marital.Status"
[11] "Guarantors"                      "Duration.in.Current.address"
[13] "Most.valuable.available.asset"   "Age..years."
[15] "Concurrent.Credits"              "Type.of.apartment"
[17] "No.of.Credits.at.this.Bank"      "Occupation"
[19] "No.of.dependents"                "Telephone"
[21] "Foreign.Worker"
```

German Credit Database

The variables:

```
> names(G.credit)
 [1] "Creditability"                  "Account.Balance"
 [3] "Duration.of.Credit..month."     "Payment.Status.of.Previous.Credit"
 [5] "Purpose"                        "Credit.Amount"
 [7] "Value.Savings.Stocks"           "Length.of.current.employment"
 [9] "Instalment.per.cent"            "Sex...Marital.Status"
[11] "Guarantors"                     "Duration.in.Current.address"
[13] "Most.valuable.available.asset"  "Age..years."
[15] "Concurrent.Credits"             "Type.of.apartment"
[17] "No.of.Credits.at.this.Bank"     "Occupation"
[19] "No.of.dependents"               "Telephone"
[21] "Foreign.Worker"
```

We are going to focus on the predictor “Credit.Amount” and the response as “Creditability”.

German Credit Database

First, let us test if there is indeed any effect of "Creditability" on the "Credit.Amount":

```
> t.test(Credit.Amount ~ Creditability, data=G.credit)

        Welch Two Sample t-test

data:  Credit.Amount by Creditability
t = 4.2643, df = 421.86, p-value = 2.477e-05
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
  513.5476 1391.8201
sample estimates:
mean in group 0 mean in group 1
       3938.127        2985.443
```

This is Welch's *t*-test which is a slight modification of the standard two sample *t*-test.

German Credit Database

Some more evidence of effect of “Creditability” on the “Credit.Amount”:

```
> wilcox.test(Credit.Amount ~ Creditability, data=G.credit)

        Wilcoxon rank sum test with continuity correction

data:  Credit.Amount by Creditability
W = 116520, p-value = 0.005918
alternative hypothesis: true location shift is not equal to 0
```

German Credit Database

Some more evidence of effect of “Creditability” on the “Credit.Amount”:

```
> wilcox.test(Credit.Amount ~ Creditability, data=G.credit)

        Wilcoxon rank sum test with continuity correction

data:  Credit.Amount by Creditability
W = 116520, p-value = 0.005918
alternative hypothesis: true location shift is not equal to 0
```

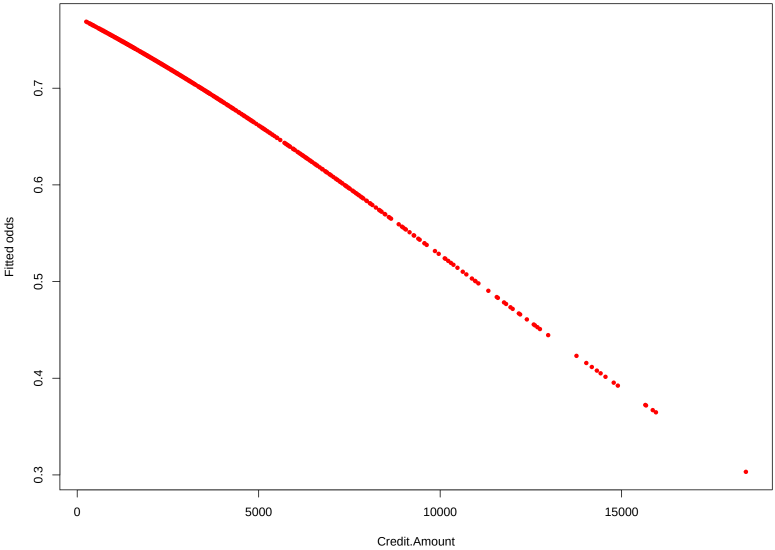
Conclusion: Indeed creditable group has a different distribution of “Credit.Amount” as compared to that of the non–creditable group. Implying that a binary classification indeed is a good idea.

Logistic regression: Creditability $\sim$ Credit.Amount

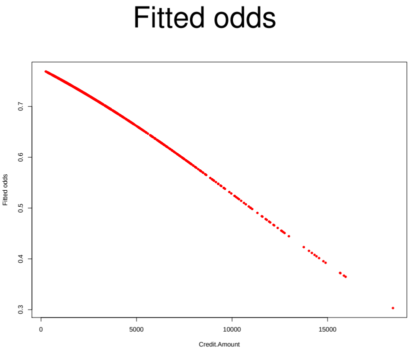
```
> library(gmodels)
> creditability.on.cramount <- glm(Creditability~Credit.Amount,data=G.credit,
+                                  family=binomial)
> summary(creditability.on.cramount)

Call:
glm(formula = Creditability ~ Credit.Amount, family = binomial,
    data = G.credit)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.7022  -1.3674   0.7688   0.8269   1.4158

Coefficients:
                Estimate Std. Error z value Pr(>|z|)
(Intercept)    1.229e+00  1.083e-01  11.348  < 2e-16 ***
Credit.Amount -1.119e-04  2.355e-05  -4.751 2.02e-06 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 1221.7  on 999  degrees of freedom
Residual deviance: 1199.1  on 998  degrees of freedom
AIC: 1203.1

Number of Fisher Scoring iterations: 4
```

****interpretation****

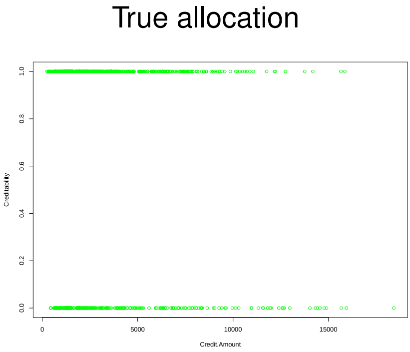
Predictions



Compare



Fitted odds
Fitted odds
0.3
0.4
0.5
0.6
0.7
0
5000
10000
15000
Credit.Amount



True allocation
Creditability
0.0
0.2
0.4
0.6
0.8
1.0
0
5000
10000
15000
Credit.Amount

Compare

Total Observations in Table: 1000

creditability.f	prediction.for.CrAm.c No	Yes	Row Total
No	20	280	300
	6.667%	93.333%	30.000%
Yes	9	691	700
	1.286%	98.714%	70.000%
Column Total	29	971	1000

Overal, the prediction seems to work fairly well, in that the classifier could correctly predict 71.1% times.

Specifically, the model seems to perform very well when predicting 1s, but it performs poorly in predicting the 0s.

Multiple logistic regression

When we have more than one predictor

When we have more than one predictor

Consider the same dataset, but now we consider two predictors: “Credit.Amount” and “Duration.of.credit..month.”.

When we have more than one predictor

Consider the same dataset, but now we consider two predictors: “Credit.Amount” and “Duration.of.credit..month.”.

More generally, we could be dealing with many more predictors, but that’s not an issue!

When we have more than one predictor

Consider the same dataset, but now we consider two predictors: “Credit.Amount” and “Duration.of.credit..month.”.

More generally, we could be dealing with many more predictors, but that’s not an issue!

Say, we have measurements on p predictors or features, and we wish to classify each point into one or the other group (binary classification).

When we have more than one predictor

Consider the same dataset, but now we consider two predictors: “Credit.Amount” and “Duration.of.credit..month.”.

More generally, we could be dealing with many more predictors, but that’s not an issue!

Say, we have measurements on p predictors or features, and we wish to classify each point into one or the other group (binary classification).

We shall again be interested in modelling:

$$p(x_1, \dots, x_p) = \mathbb{P}\left(Y = 1 | X_1 = x_1, \dots, X_p = x_p\right)$$

Once again, we propose:

$$\log\left(\frac{p(X_1,\ldots,X_p)}{1-p(X_1,\ldots,X_p)}\right)=\beta_0+\beta_1X_1+\cdots+\beta_pX_p$$

Once again, we propose:

$$\log\left(\frac{p(X_1,\ldots,X_p)}{1-p(X_1,\ldots,X_p)}\right) = \beta_0 + \beta_1 X_1 + \cdots + \beta_p X_p$$

Estimation: maximum likelihood estimation, i.e., we look for those values of $\beta_0, \beta_1, \ldots, \beta_p$ which maximise

$$L(\beta_0,\beta_1,\ldots,\beta_p) = \prod_{i\in\mathcal{Y}_1} p(x_{1,i},\ldots,x_{p,i}) \prod_{i\in\mathcal{Y}_0} \left[1 - p(x_{1,i},\ldots,x_{p,i})\right]$$

Few remarks:

- Interpretation of $\widehat{\beta}_i$ for $i = 1, \ldots, p$: Keeping all other variables constant, a unit increase in the i-th feature would change the odds by a factor of $e^{\widehat{\beta}_i}$.

Few remarks:

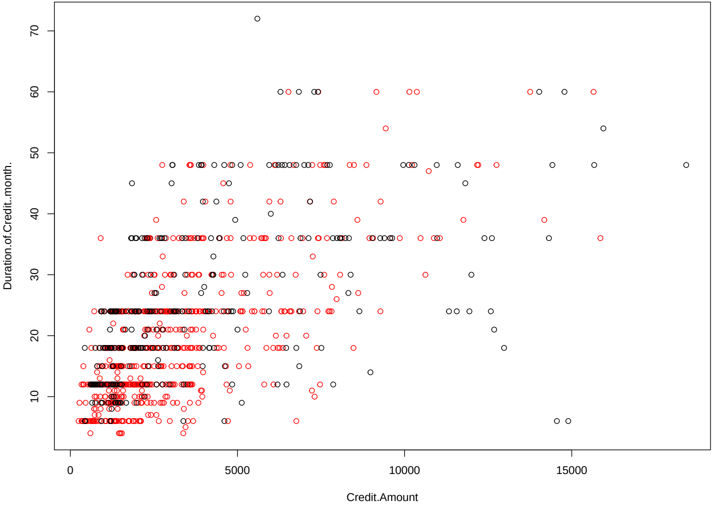
- Interpretation of $\widehat{\beta}_i$ for $i = 1, \ldots, p$: Keeping all other variables constant, a unit increase in the i-th feature would change the odds by a factor of $e^{\widehat{\beta}_i}$.
- It is desirable that different features give indeed different information $\Rightarrow$ multicollinearity.

Few remarks:

- Interpretation of $\widehat{\beta}_i$ for $i = 1, \ldots, p$: Keeping all other variables constant, a unit increase in the i-th feature would change the odds by a factor of $e^{\widehat{\beta}_i}$.
- It is desirable that different features give indeed different information $\Rightarrow$ multicollinearity. This is again tested by the same VIF

German Credit Database

The plot for the three variables looks like:



Multiple Logistic Regression: output

```
> summary(creditability.on.cramdur)

Call:
glm(formula = Creditability ~ Credit.Amount + Duration.of.Credit..month.,
    family = binomial, data = G.credit)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-1.8249  -1.2734   0.7164   0.8533   1.5020

Coefficients:
                            Estimate Std. Error z value Pr(>|z|)
(Intercept)                1.670e+00  1.466e-01  11.390  < 2e-16 ***
Credit.Amount             -2.300e-05  3.059e-05  -0.752    0.452
Duration.of.Credit..month. -3.412e-02  7.282e-03  -4.685  2.8e-06 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 1221.7  on 999  degrees of freedom
Residual deviance: 1176.6  on 997  degrees of freedom
AIC: 1182.6

Number of Fisher Scoring iterations: 4
```

****interpret****

Predictions

Total Observations in Table: 1000

	prediction.for.CrAmDur.c		
creditability.f	No	Yes	Row Total
No	37 12.333%	263 87.667%	300 30.000%
Yes	29 4.143%	671 95.857%	700 70.000%
Column Total	66	934	1000

Overall efficiency of the model has actually reduced a bit as compared to the logistic model with single predictor “Credit.Amount”.

Although the model still performs poorly in predicting the 0s, but it is better than the single predictor model.

However, the model’s efficiency is classifying correctly identifying the 1s has considerably reduced.

Comparing the two binary classifiers

Predicting creditability using "Credit.Amount"

Total Observations in Table: 1000

creditability.f	prediction.for.CrAm.c: No	Yes	Row Total
No	20	280	300
	6.667%	93.333%	30.000%
Yes	9	691	700
	1.286%	98.714%	70.000%
Column Total	29	971	1000

Predicting creditability using "Credit.Amount" and "Duration"

Total Observations in Table: 1000

creditability.f	prediction.for.CrAmDur.c: No	Yes	Row Total
No	37	263	300
	12.333%	87.667%	30.000%
Yes	29	671	700
	4.143%	95.857%	70.000%
Column Total	66	934	1000

Which one would we choose?

Comparing the two binary classifiers

Predicting creditability using "Credit.Amount"

```
Total Observations in Table:  1000

                | prediction.for.CrAm.c
creditability.f |        No |       Yes | Row Total |
----------------|-----------|-----------|-----------|
             No |        20 |       280 |       300 |
                |    6.667% |   93.333% |   30.000% |
----------------|-----------|-----------|-----------|
            Yes |         9 |       691 |       700 |
                |    1.286% |   98.714% |   70.000% |
----------------|-----------|-----------|-----------|
   Column Total |        29 |       971 |      1000 |
----------------|-----------|-----------|-----------|
```

Predicting creditability using "Credit.Amount" and "Duration"

```
Total Observations in Table:  1000

                | prediction.for.CrAmDur.c
creditability.f |        No |       Yes | Row Total |
----------------|-----------|-----------|-----------|
             No |        37 |       263 |       300 |
                |   12.333% |   87.667% |   30.000% |
----------------|-----------|-----------|-----------|
            Yes |        29 |       671 |       700 |
                |    4.143% |   95.857% |   70.000% |
----------------|-----------|-----------|-----------|
   Column Total |        66 |       934 |      1000 |
----------------|-----------|-----------|-----------|
```

Which one would we choose?

Sensitivity: The ability to correctly identify the actual "positives"

Specificity: The ability to correctly identify the actual "negatives"

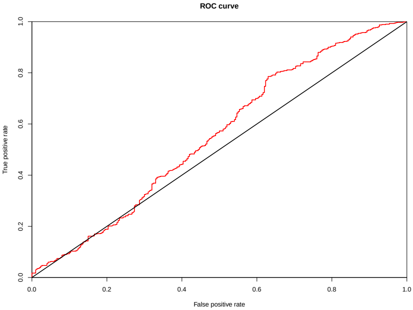
ROC curve and AUC

Receiver operating characteristic (ROC) curve, historically, originated from communications theory, as a simple tool to plot **true positive rate (sensitivity)** and **false positive rate (specificity)** for various values of the threshold.

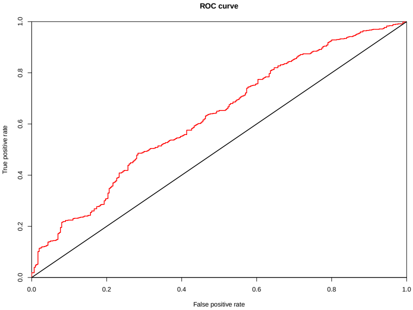
ROC curve and AUC

Receiver operating characteristic (ROC) curve, historically, originated from communications theory, as a simple tool to plot **true positive rate (sensitivity)** and **false positive rate (specificity)** for various values of the threshold.

ROC for predicting creditability using "Credit.Amount"



ROC for predicting creditability using "Credit.Amount" and "Duration"



ROC curve and AUC

ROC curve provides compelling visual depiction of the goodness of models.

However, it is often preferable to be able to quote a single, which is the **area under the curve (AUC)**.

This is the area under the ROC curve. A good model should have a high value of AUC (close to 1).

For the two models we considered, the AUC were:
simple logistic: 0.5548
multiple logistic: 0.6257

Another approach: KNN classifier