R-squared indicates goodness of it. However it is biased: with increasing variables that are not necessarily predictors, R-squared will **always grow.**

Adjusted R-squared has a penalization factor: the greater the amount of variables, the greater the penalization.

\* = p-value < 0.1 10% chance you might be wrong

\*\* = p-value < 0.05 5% chance you might be wrong

\*\*\* = p-value < 0.01 1% chance you might be wrong

Backwards Elimination method

Sometimes, strictly removing a variable when P-value > SL but the values are very close becomes an arbitrary and dubious decision.

**If after optimization, adjusted R-squared increases:**Removal of the variable is acceptable as it improves the model.

**If the Akaike criterion decreases:**The model is better, because this parameter is relative to two models. It deals with the trade-off between the goodness of fit of the model and the simplicity of the model.

The magnitude of the coefficient of a variable **is not proportional** to its contribution to the change of the dependent variable.

**If the units are the same, then they are comparable.  
If not, then they can only be compared in the same unit of the variable.***‘’X has a bigger impact on profit per unit of X, than…’’*

Also, we can only infer the contribution of a variable when we consider all others remain unchanged.

Logistic Regression rules

When we apply a sigmoidal function to a linear regression function, we get a logistic regression. The sigmoidal function avoids insignificant extrapolations by creating asymptotes near the extremes of the data set (at 0 and 1, for example).

Applying the sigmoidal function transforms our interpretation of the y-axis from discrete nominal values to **continuous probability values**.

In a way, logistic regression is applied to calculating DL50, IC50 in pharma, toxicology, binding affinities. However, these studies have a continuous set of values, and are not approximated to 0/1 switches.

**Type I error:** false positive. Should occur, did not. Soft warning.  
**Type II error:** false negative. Shouldn’t occur, did. Dangerous warning.

When doing a logistic regression, assessment of a good model is done by looking at the confusion matrix. **A good model will have less type II errors (bottom-left) than type I (top-right).**

**Useful calculations:**

**Accuracy rate (AR)** of the model = Correct observations / Total  
*The correct observations are the pairs (i, î) which are (0,0) and (1,1)*

**Error rate (ER)** of the model = Wrong observations / Total = **1 – AR**

In logistic LR, the coefficients of the variables cannot be compared to the change they input in the dependent variable:

* In linear regressions, the dependent variable is y and it varies linearly with the independent variables, as long as the units remain the same 🡪 **CAN BE COMPARED**
* In logistic regressions, the y-term is sigmoidal and represented by . This is not linear dependency, so the contributions cannot be interchanged linearly.

However, **between independent variables**, the contributions can be compared to one another. Once again, the units must be the same.

The z-score is a standardized coefficient, independent of the scale of its variable:

The z-score is harder to interpret, but given that is has its own standardized unit, it is more efficient in comparing the contribution of a given IV to the DV.

**HOW TO USE GRETL – logistic example**

1. Load data set
2. Define dummy variables for categorical values
3. Run logit binary
4. Use all IVs that could be regressors of the DV.
5. For dummy variables, the *baseline* is the subvariable that is **not** included in the model. However, they will not influence the final result.
6. Now we do the backwards elimination. Remove the worst value and re-iterate.  
   ***Validations of a good model:***
   1. *Adjusted R-squared must improve.*
   2. *Number of cases correctly predicted must improve.*
   3. *Type II errors (mostly) must decrease*

**If** there’s a calculated variable on the set of data and the p-value is insignificant, it could mean two things:

1. *The calculated variable is not a predictor of a change in the DV.*
2. *The calculation is incorrect and not a good model for the IV, which could actually contribute to a change in the DV.*
3. When all variables with no significance are excluded and we start going for \* (p < 0.1), we need to be careful in removing them. **At this stage, the validation techniques are what define if an elimination is good for the model.** If the change in the criteria are not sufficient enough, than the decision must be done through critical thinking and arbitrary.

**Independent variable transformations:**

Applying a square root transform **inflates smaller numbers but stabilises bigger ones**. So you can think of it as pushing small residuals at low x values away from the fitted line and squishing large residuals at high X values towards the line.

A polynomial term–a quadratic (squared) or cubic (cubed) term turns a linear regression model into a curve. But because it is X that is squared or cubed, not the Beta coefficient, it still qualifies as a linear model. This makes it a nice, straightforward way to model curves without having to model complicated non-linear models.

There are **three main situations that indicate a linear relationship may not be a good model**.

1. Most important is the theoretical one. There are some relationships that a researcher will hypothesize is curvilinear. Clearly, if this is the case, include a polynomial term.

2. The second chance is during visual inspection of your variables. This is one of those reasons for always doing univariate and bivariate inspections of your data before you begin your regression analyses. (You always do this, right?) A simple scatter plot can reveal a curvilinear relationship.

3. Inspection of residuals. If you try to fit a linear model to curved data, a scatter plot of residuals (Y axis) on the predictor (X axis) will have patches of many positive residuals in the middle, but patches of negative residuals at either end (or vice versa). This is a good sign that a linear model is not appropriate, and a polynomial may do better.

Logarithms are good for normalizing big sets of values. If the value of an x variable is too high, a change of 1 units does not affect the value as much as a change in a small value. This would lead to big x values not varying the model too much and we would say they are not significant in that interval. **This is wrong:** a logarithmic approach allows to say that any value change in any interval of data, big or small the value may be, has the same contribution to a change in the model.  
  
*If there are 0 values, remember that x must be inputted with x+1. It doesn’t change the final model and only avoids error values.*

**Creating derived variables:**

When a variable misrepresents our data, it is better to derive it into another form. For example, bank balance masks a lot of other parameters (saving efficiency, more earnings, etc) and it can be represented in the form of:

**Beware of collinearity!** Because the new variable derives from two variables also used in the model, they are connected and therefore introduce bias in the model. Therefore, we need to exclude the other two variables.

**How to check for multicollinearity?**

If a variable appears as not significant, it might be due to multicollinearity. To check for this, use the **VIF test**.

If collinearity values are not >10 but are relatively high, try to remove one possible related variable and check again. If the values are back to 1, the issue disappeared and we can know which variables are related.

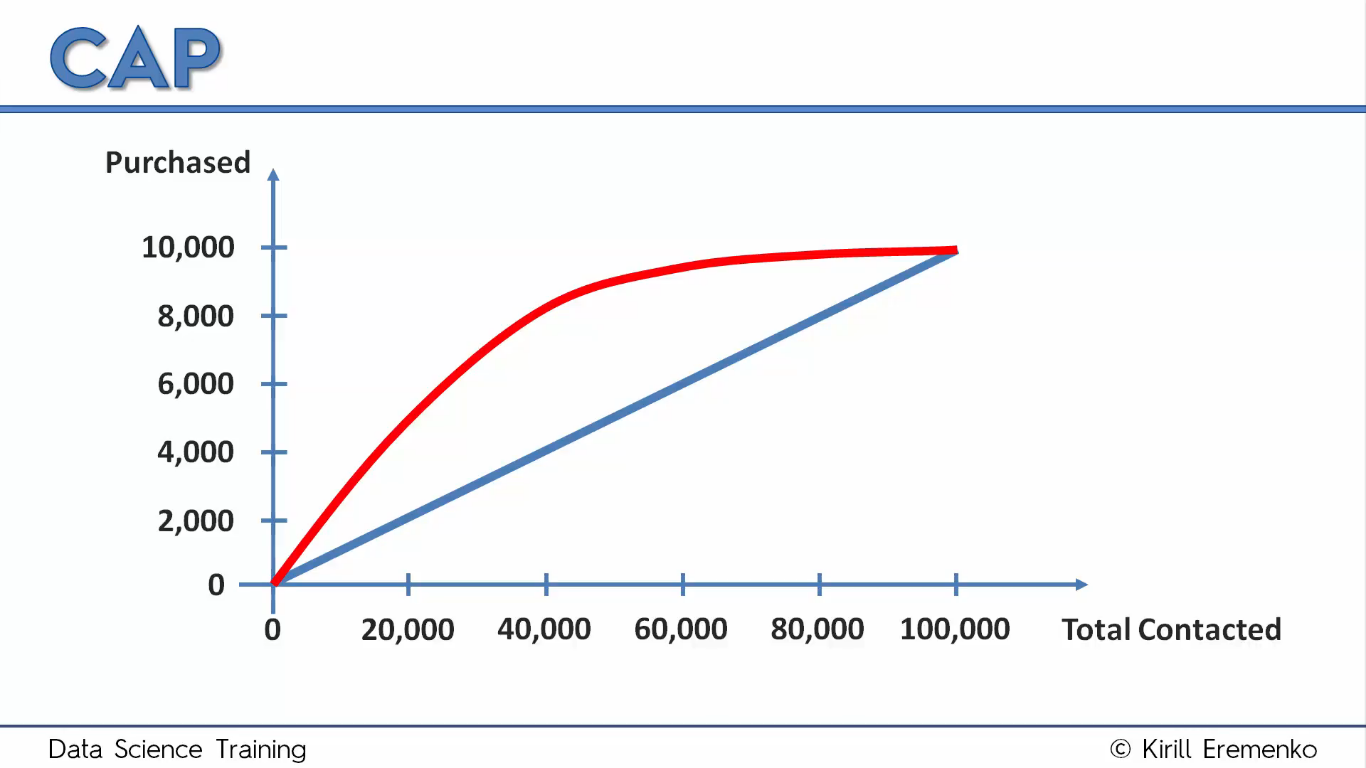
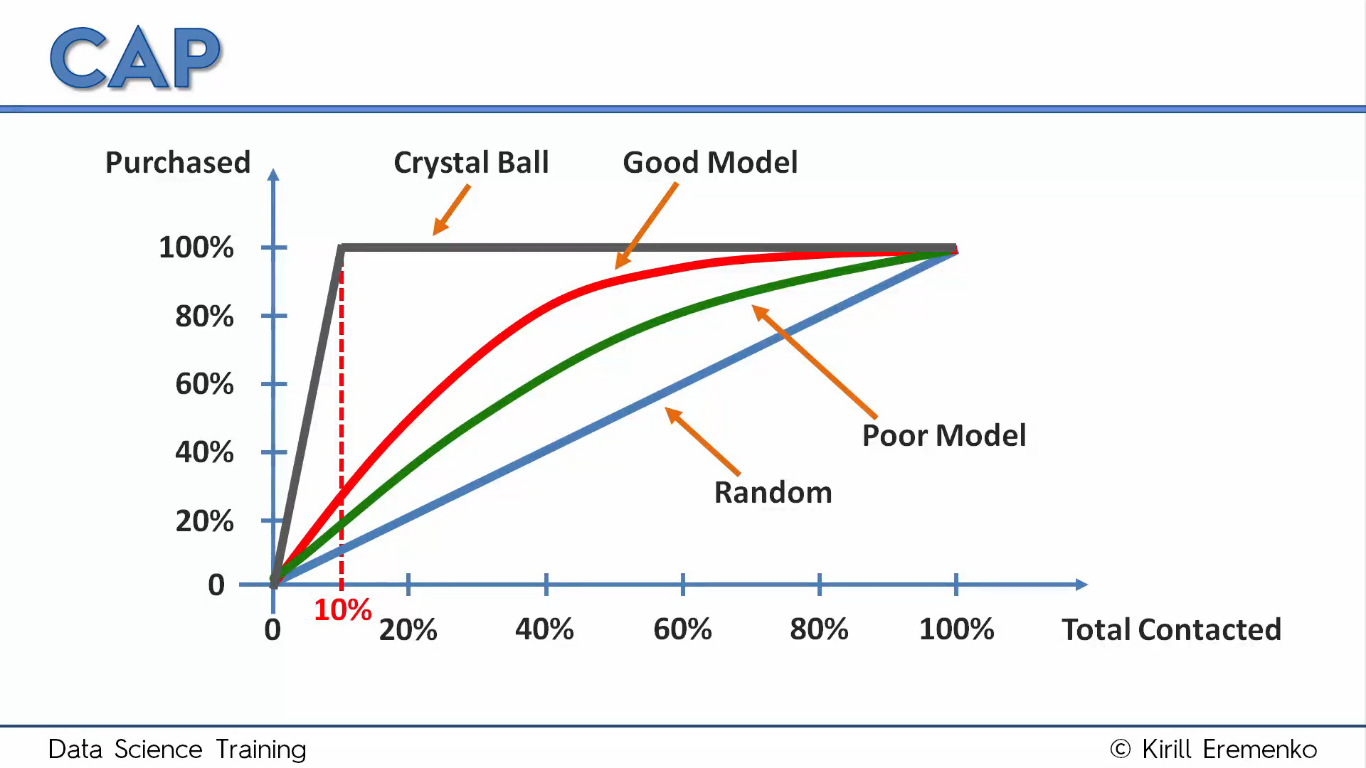
The **correlation matrix** indicates individual correlations between pairs of variables. Anything above 0.7-0.9 is highly correlated and needs to be dealt with. Anything below 0.3 can be ignored.

When the program tweaks the x variables to yield a y variable, it fixes all xn while changing x1. **If x1 and x2 are correlated**, no amount of iterations will enable the model to change x1 without tweaking x2 at the same time.

**How to assess the model**

The accuracy rate is not sufficient due to the **accuracy paradox**: if the model stops predicting any events, some type I/II errors may not occur anymore and the accuracy actually goes up.

The **Cumulative Accuracy Profile (CAP)** is better. In this example, *y = 0.1x*

It maximizes the amount of successes (y-axis) with the least number of x-variables. The CAP model allows to reduce the amount of experiments one needs to do to reach a significant y-value.

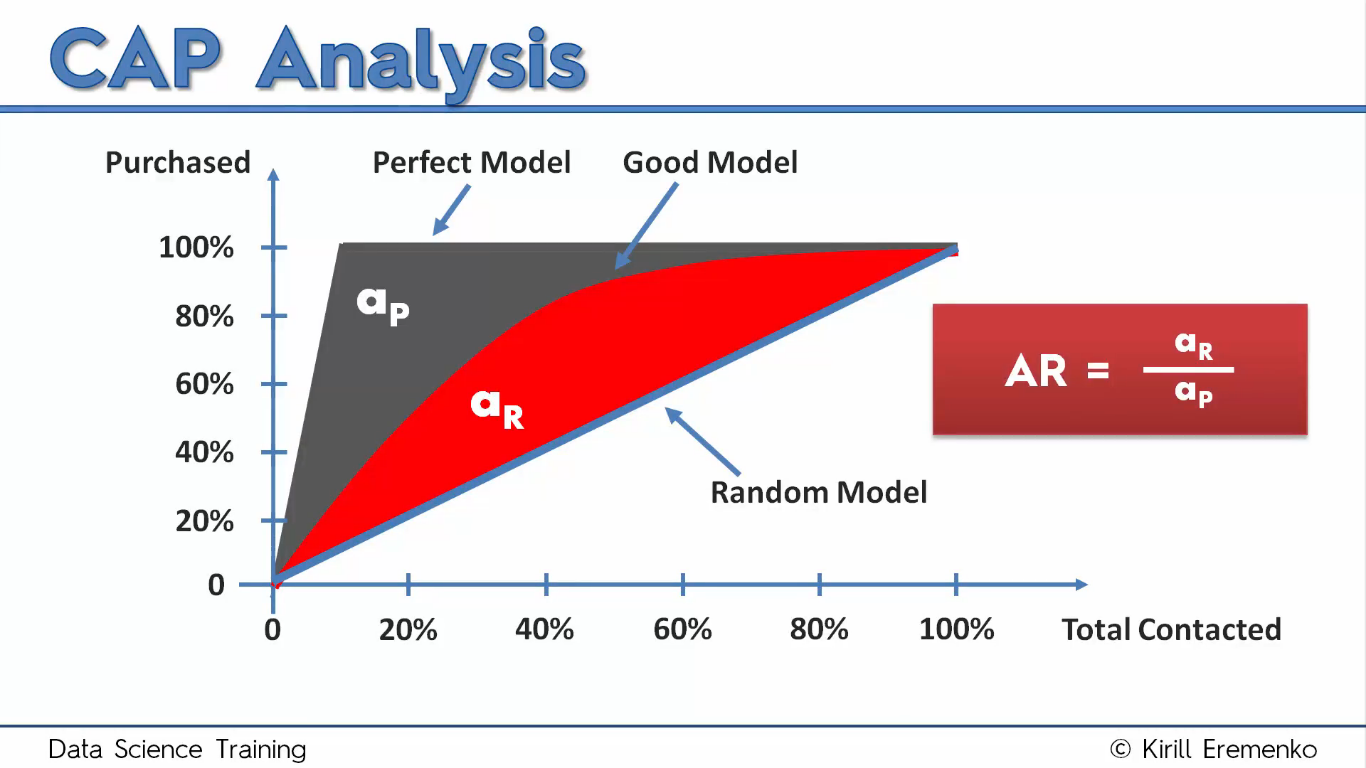
Different CAP curves can be modelled. The one which deviates the most from the linear tendency is the one that maximizes y with the least x-iterations.

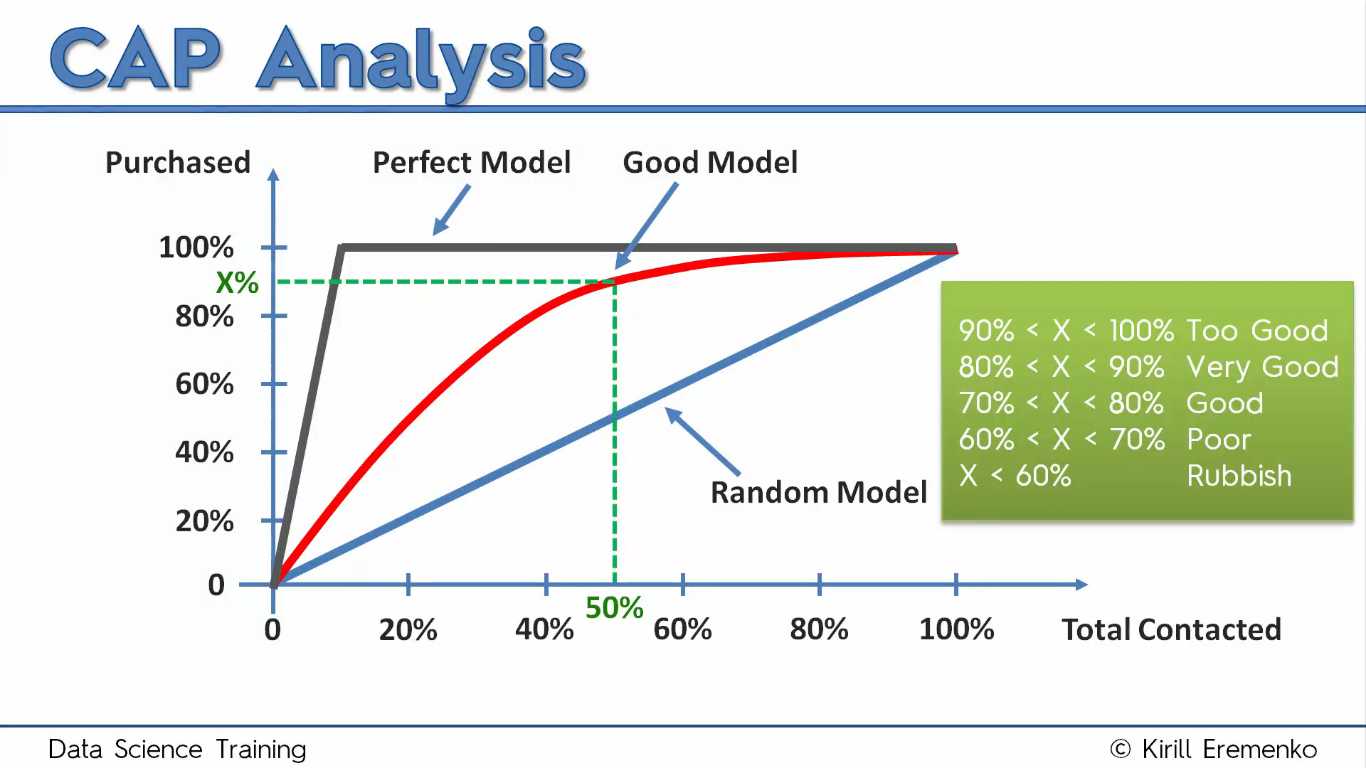
*The ROC curve is similar to the CAP curve but whereas   
the CAP curve relates the hit rate to the rate of all alarms   
the ROC curve compares it with the rate of false alarms.*

**How to build CAP curves in Excel**

1. Get the probabilities from the forecast option of the model in Gretl
2. Save as a variable, export, save and load onto Excel
3. Filter probability by largest to smallest 🡪 this is our target
4. Use the Excel that I made

***How to assess the model:***

1. Calculate the accuracy ratio by using the formula.
2. If no statistical tools are available to automatically do this for you, just calculate the Y50, which is the y-axis value at which X = 50%, just like in pharmacology. **80% is what we are aiming for. Over 90% is not good, because it might be a result of overfitting.**



A **post-facto variable** (a variable that ‘’looks into the future’’) might also be at stake, which is a variable that biases your model (for example, the CreditScore goes to 0 after someone left the bank, so anyone with 0 CS with have Exit = 1, which is a biased correlation)

***How to avoid overfitting:***

1. Calculate and compare two variables:

AR taken from the big test sample. This accuracy rate might be biased, because the model was derived from the same population in which it was tested on. To see if it’s a good model, we need to fit it to a smaller, more heterogeneous sample (**test**).

This is the test population. They are not a portion of the original data set, but a **different** data set (they can be from the same set, but no overlap can be present).  
**If the AR ratio is ~1.0, then the model is good and no overfitting occurred.**

***To finalize, test your model on new data:***

1. Add the new data to the previously existing dataset, **but without the result** (the value we want to see predicted).
2. Load the data to gretl and do the same as above. *The new data set will not be input in the model because we removed the result!*
3. When you have the model, do the forecast for your new data. Do not use pre-forecast data.
4. Save the new prediction variable and export it with RowNumber and ^P.
5. Copy the success results onto the new Excel file, so you have 3 columns:

*RowNumber Exited P\_hat*

1. Paste the data into the CAP template.

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Model deteriorated?**

* Overfitting **Did not change much?**

Good

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Model ruggedness?**

* Low number of data points **AR did not change much?**

Good

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Other alternatives:**

**1.** Instead of applying just one model to different data sets, **run different models on the same data.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Model 1 | Model 2 | Model 3 | Model 4 | Model **n** |
| AR/X50? | AR/X50? | AR/X50? | AR/X50? | AR/X50? |

**From the AR or X50, choose the best model and then apply it to the test data.**

**2.** Test your model over time. **Choose one model, and see if it applies to different temporal data.**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Month 1 | Month 2 | Month 6 | Month 12 | Month **n** |
| Same AR/X50? | Same AR/X50? | Same AR/X50? | Same AR/X50? | Same AR/X50? |

**If the model gets worse, population tendency is changing.**

**Drawing insight from the CAP curve:**

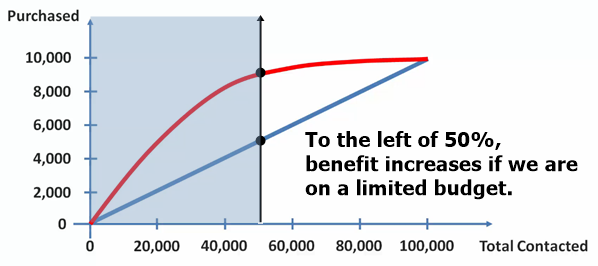
* **Likelihood score**

It is the ^P values we have in our data set. Highest is better. We can determine which percentage of customers we want to contact based on their likelihood to perform an action.

* **Budget constraint**

The CAP curve allows to target the customers of highest percentage, which allows to contact a limited amount of people with a limited budget with the highest success.

You can calculate benefit:



* **Efficiency**

When budget constraints are not an issue, it is good to simplify the business strategy and be smart about it. Instead of contacting all customers, just contact those who will yield the highest profit, it is economic efficiency.

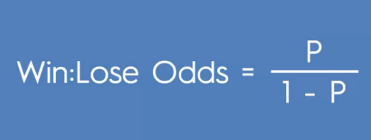
When we see significant kinks on the model curve (lower slopes), that is an hint for **a cut-off point**. To the left of that kink lies efficiency.

**How to interpret coefficients on Logistic Regression**

They are compared indirectly: by the change they produce in the **odds ratio.**

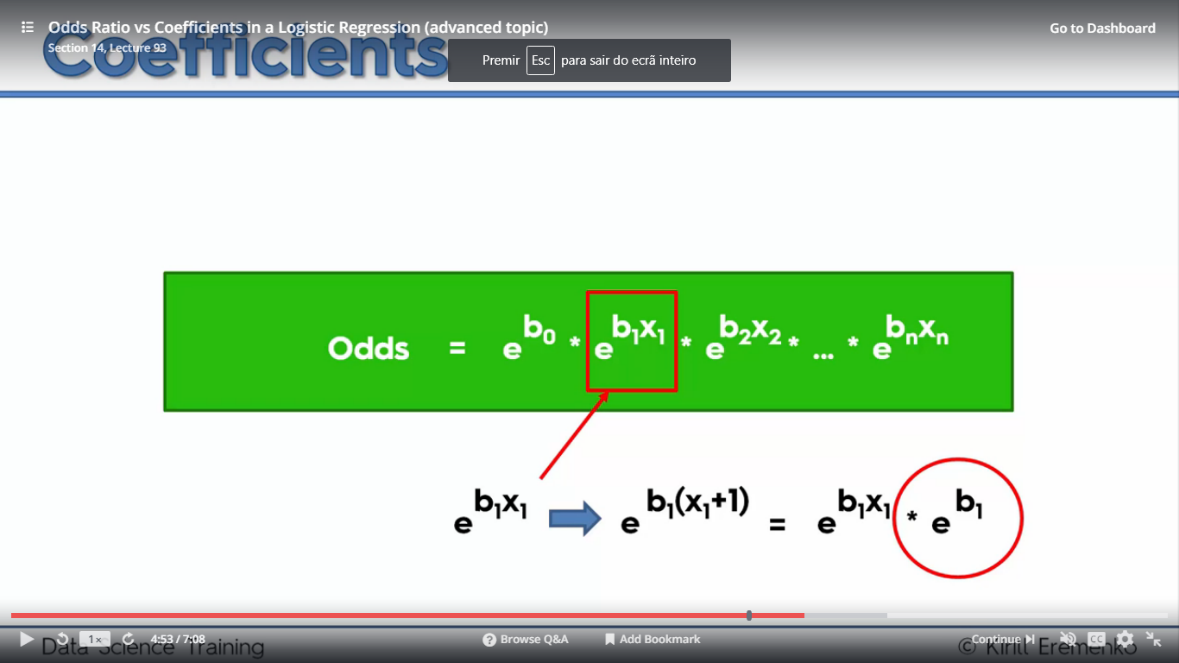
**Odds ratio**

Losing:Winning Odds 🡪 L:W 🡪 **L/W** Probability of winning =  **=**



*As the probability increases (x), the odds (y) will increase exponentially.*

**Coefficients**



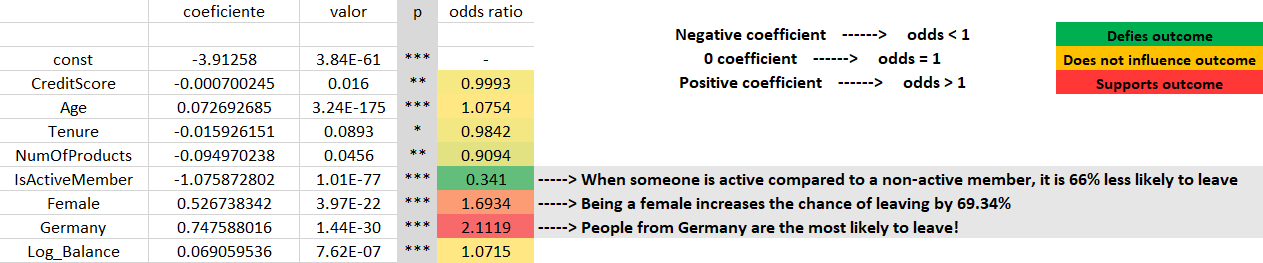
*Increasing an independent variable xi by* ***1*** *unit will increase the odds by a multiplicative factor of* ***ebi****.*

**Example:** in a model, age has the coefficient 0.07. If we increase age by 1, the odds will increase by e0.07 = 1.075.

*If someone is 1 year older than anyone, their odds of leaving the bank increase by x1.075!*

1. Copy the odds ratio from gretl and do an Excel with: **coefficient, p-value, odds ratio**.
2. Do a conditional heat map for the odds ratio values, so you can see which are the most contributive in the model.

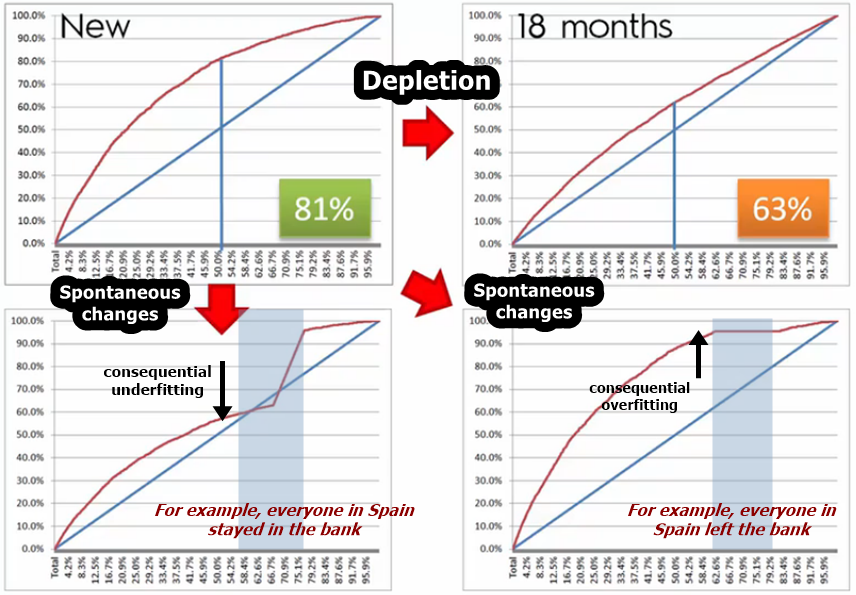
***0.8 < ODDS < 1.2 is doesn’t have a significant impact.***



**Model deterioration**

Possible reasons are:

* **Additional factors:** changes in the way the system behaves that are not accounted in the model could worsen the predictions.
* **Changes in behavior:** something that happened in the system that change the behavior of our observation sample (people, for example).
* **Changes in process:** changes in the way the system works requires a new predictive model.
* **Changes in existing factors:** the best example is if your customer base continues the same but they get older, and your company does not follow trends so your model and business strategy become outdated. The distribution of age itself changes as well, so new data needs to be considered.
* **Competitor:** works like an additional factor that skews the tendency of your system.
* **Changes in industry:** once again, inability to follow trends results in outdated models. Sometimes new things appear and just change the interests of the industry. Bitcoin for example.
* **Changes in regulations:** for example, the call-to-action was by physical mailing. If you switch to e-mails, it gets easier to connect, but the model also needs to change.
* **Changes in product:** each product might have its own different model. The product depends on its environment, if you change it, the model required is also different.
* **Depletion:** this occurs over time, and your model can get addicted to a constant customer base. Your data changes over time but the model doesn’t: this leads to deviations not accounted in a static model.
* **Spontaneous changes:** quick changes that can’t be predicted. In a short period, they can render the model completely outdated because the outcome is affected.



**How to rebuild a deteriorated model**

The ARR process is hierarchical: the first fails, move on to the next.

**A|ASSESS**

Look at your model periodically. Run a fresh data set and see if it fits well (about 10% of your total sample).

*If it fails…*

**R|RETRAIN**

You don’t have to do the model building from scratch: keep all the variables the model was composed of. Use a fresh sample and re-do the model, as the coefficients will change. Then deploy the retrained model and assess it for shorter periods of time at first to check its adequacy.

*If it fails…*

**R|REBUILD**

Redo the model from scratch, testing all available variables. One quick way is to just remove some variables from the previous one, it might work and it is faster.

**If you do model maintenance, you can avoid all this**

Run a Champion-Challenger setup: test the old (champion) and the new model (challenger) side by side. See which one performs better over time with real life data: you can split the population over both, or use the sample partial dataset on both and see which is the best CAP curve.

Frequency of maintenance depends on the industry can be from 1-6 months, for example.