Data Science
Methods for MADS
EBM216A05.2022-2023.1

FREE EDITION*

SUMMARY OF EVERYTHING FROM WEEK 1

Enhanced with a dynamic table of contents.

For the full version Google rug mads madlad

100%

OF THE PROFIT FROM THIS SUMMARY IS DONATED TO THE FOLLOWING NGOs:

SIAN · LGBT Asylum Support · alzheimer nederland · DUTCH CANCER SOCIETY · VOEDSELBANKEN.NL Groningen & regionaal distributiecentrum · Aids Fonds · SOAIDS · STOP AIDS NOW!

This summary helps you study & people in need.
Please do not share it for free, but ask your
friends to buy it and do a good deed.
MADS MADLAD

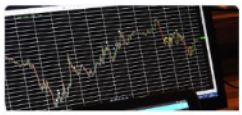
Note from MADS MADLAD:

Thank you for checking out my free summary. When I was writing these I sometimes struggled with this program, but there were no summaries available. This is why I decided to write something that is truly complete with a lot of effort put into it.

It helped me and my friends get good grades, but I always had you in mind, the future reader. When necessary, I always went the extra mile to make my summaries, more readable, organized and complete.

If you feel like it, leave me a review of how the course is going using this summary, it will make my day to hear your feedback.

Check out my other extensive summaries for other MADS courses:



Statistical Learning in Marketing
EBM214A05.2022-2023.1

Data Engineering for MADS
EBM213A05.2022-2023.1

Companies, Brands, and Consumers
EBM215A05.2022-2023.1

Retail & Omnichannel Marketing
EBM880B05.2022-2023.1

Market Models
EBM077A05.2022-2023.1

Data Science Methods for MADS
EBM216A05.2022-2023.1

Digital Marketing Intelligence
EBM079B05.2022-2023.1

Contact info:

If you need help or have an inquiry, contact me: https://www.georgedreemer.com

Connect with me on LinkedIn: https://www.linkedin.com/in/georgedreemer/

Donations:

By no means am I looking for fellow students to send me money! But if you feel like sending me some ETH or BTC, you can do so here:
--> ETH: 0x123e086c6808459e7fC6Ac7F64a77dBA1dDe0149
--> BTC: bc1qgwzc82vph5v8rmzef4ywechjf85772n7m2e22g

MADS MADLAD

wishes you good luck & perseverance.

Grades Testimony:

COURSE CODE	TITLE	SCORE	DATE	RESULT
EBS001A10	Business Research Methods for Pre-MSc	8	21-12-2021	8
EBS002A05	Mathematics for Pre-MSc	9	10-11-2021	9
EBS003A05	Organization Theory & Design for Pre-MSc	7	05-11-2021	7
EBB098A05	Contemporary Theories on Business and Management	6	11-05-2022	6
EBB649C05	Strategic Management B&M	8	15-06-2022	8
EBB617B05	Human Resource Management B&M	8	08-04-2022	8
EBB104A05	Behavioural Decision Making	7	03-11-2021	7
EBB085A05	Marketing Research for E&BE	8	04-04-2022	8
EBS008B10	Research Paper for Pre-MSc Marketing	7	05-07-2022	7
EBM043A05	Business Ethics	8	14-11-2022	8
EBB105B05	Digital Marketing Analytics	8	21-01-2022	8
EBM213A05	Data Engineering for MADS	7	01-11-2022	7
EBM214A05	Statistical Learning in Marketing	8	02-11-2022	8
EBM215A05	Companies, Brands, and Consumers	8	05-11-2022	8
EBM216A05	Data Science Methods for MADS	9	20-01-2023 new	9

Table of Contents

Week 1

Lecture 1: Introduction to Machine Learning

Data Science Process

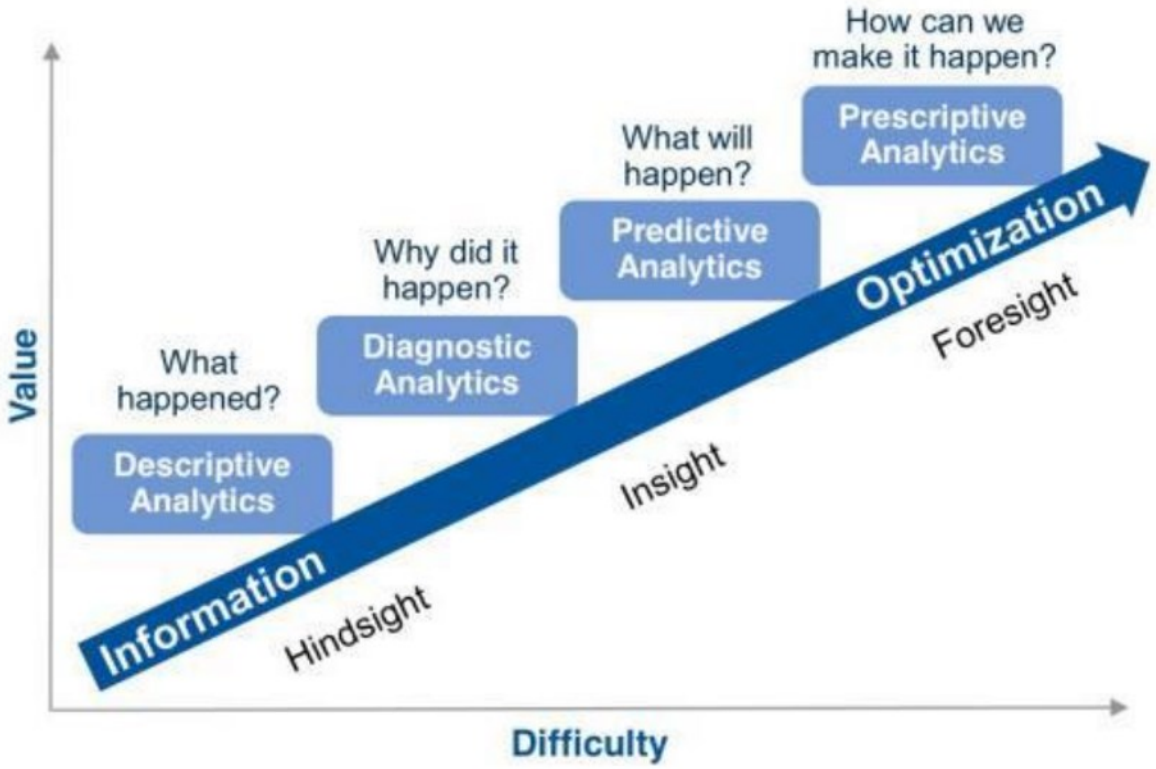
1 DEFINING THE BUSINESS PROBLEM	2 DESIGN THE RESEARCH	3 DATA COLLECTION & PREPARATION	4 EXPLORATIVE ANALYSIS	5 MODELLING	6 IMPLEMEN-TATION	7 MONITORING
Ask question to discover the real problem (manage-ment dilemma, manage-ment question, research question)	Formulate hypothesis, literature research, defining constructs	This includes extracting data from different sources, data cleaning and data trans-formation, including creating new variables based on the research design	Including correlations, statistical tests, histograms, etc.	Model specification (e.g., type and structure), estimation, validation and interpretation	Communicate results to stakeholders using a datadriven storyline and visualisations, deployment, production	Monitoring of the model performance

- **Defining business problem (1)**
  - Ask questions to discover the real problem
    - Management dilemma, questions
    - Research questions
- **Design the Research (2)**
  - Formulate hypotheses
  - Literature research
  - Define Constructs
- **Data Collection & Preparation (3)**
  - Extracting data from sources
  - Data Cleaning
  - Data transformation (e.g., new variables)
- **Explorative Analysis (4)**
  - Correlations, statistical tests, histograms, etc.
- **Modelling (5)**
  - Specification (e.g., type and structure)
  - Estimation
  - Validation
  - Interpretation
- **Implementation (6)**
  - Communicating the results to stakeholders
  - Data-driven storyline & visualization

- **Monitoring (7)**
    - Monitoring the model's performance

Criteria for a good model

- Simple
- Evolutionary
    - Starting simple but building it up
- Complete
    - As complete and simple as possible
- Adaptive
- Robust
    - Able to use it in different circumstances (e.g. during inflation, COVID, etc.)



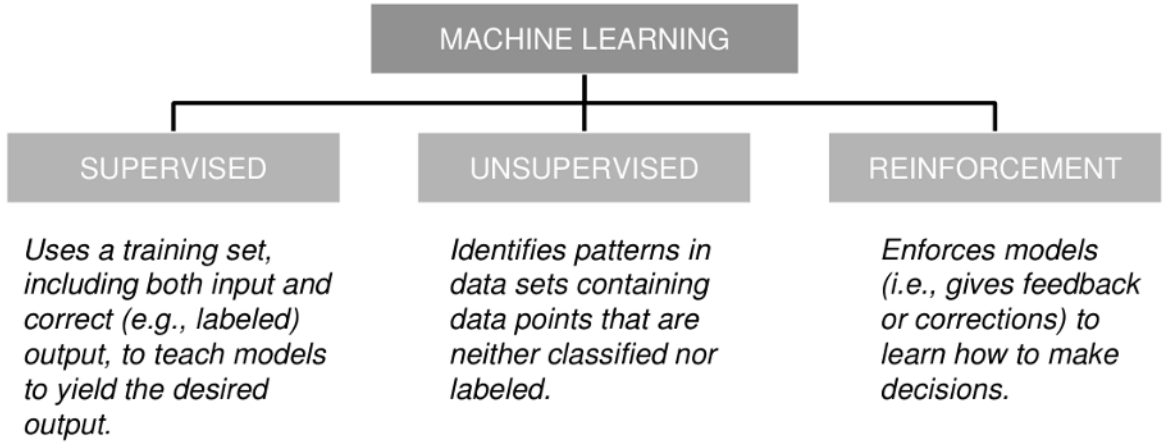
What is (Machine) Learning?

Machine learning is concerned with computer programs that automatically improve their performance through experience.

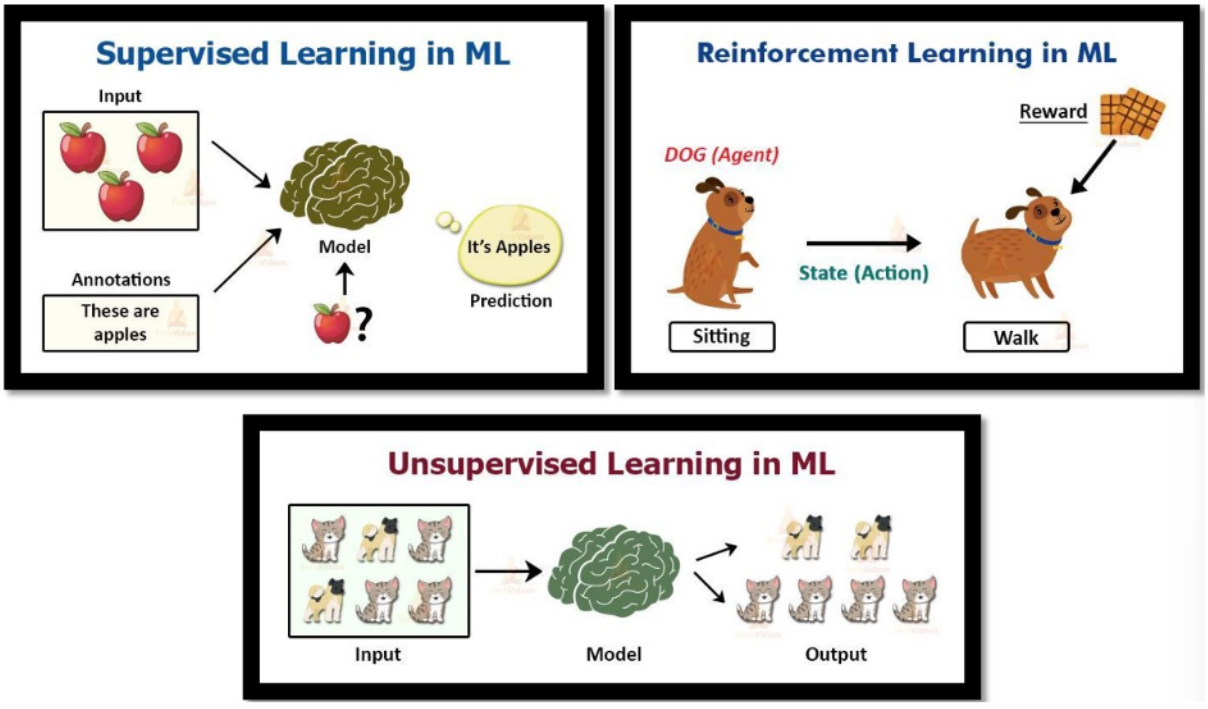
- Branch of AI and CS, which focuses on use of data and algorithms to imitate the way that humans learn.

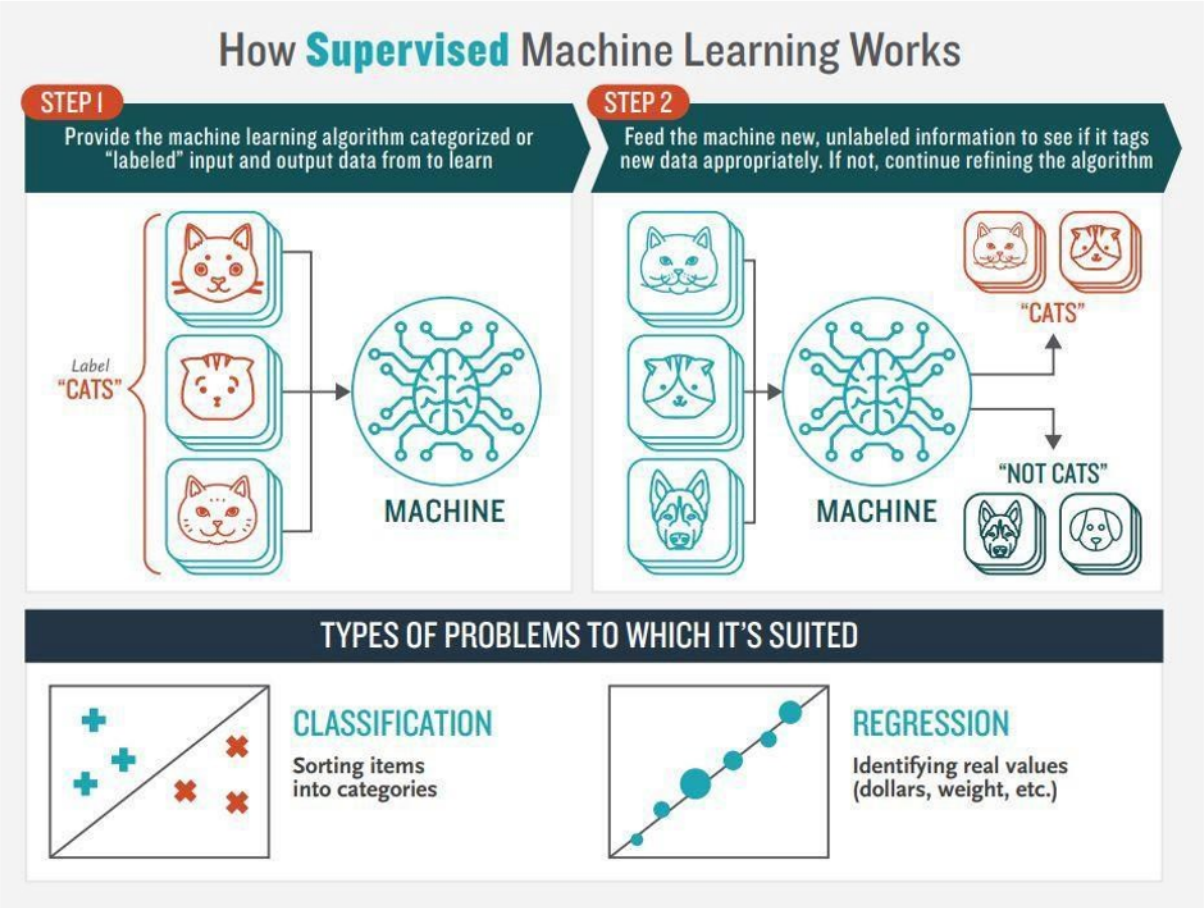
3 Types of ML Models

- **Supervised:** uses a training set, including both input and correct (e.g. labeled) output, to teach models to yield the desired output.
    - **Input + Annotations -> Model -> Prediction**
    - Used for: classificaition (sorting items into categories), regressions
- **Unsupervised:** Identifies patterns in data sets containing data points that are neither classified nor labeled.
- **Reinforcement:** enforces models (gives feedback or corrections) to learn how to make predictions.

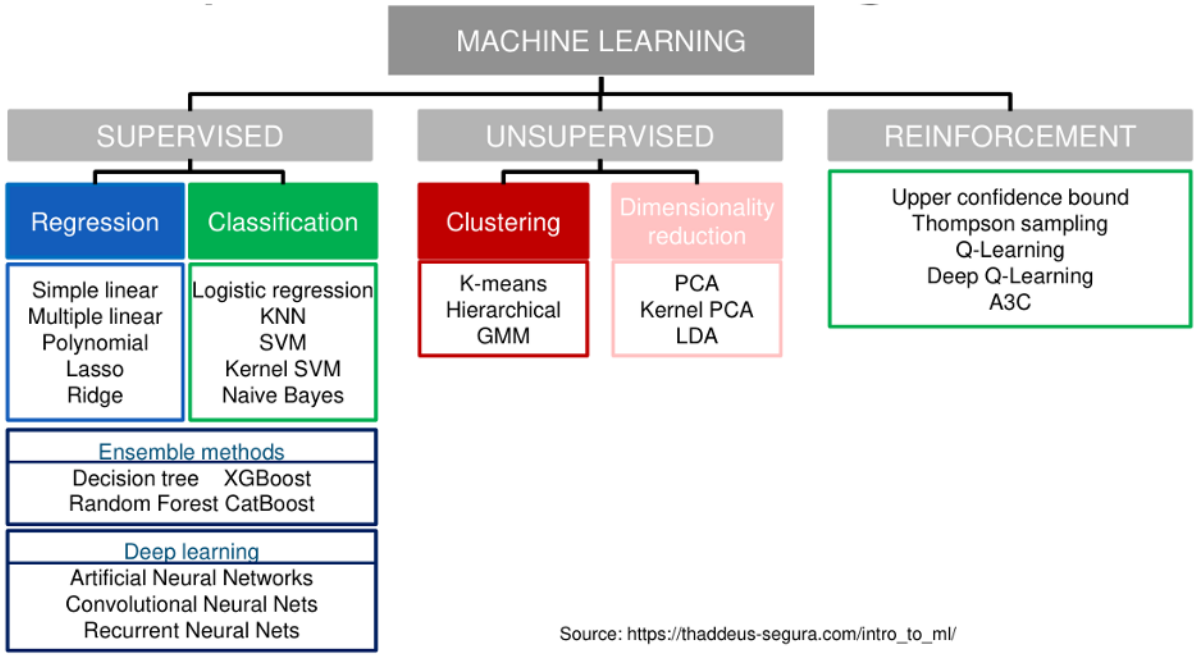


- Visual Examples of the 3 Types:





ML Techniques



Source: https://thaddeus-segura.com/intro_to_ml/

Why ML?

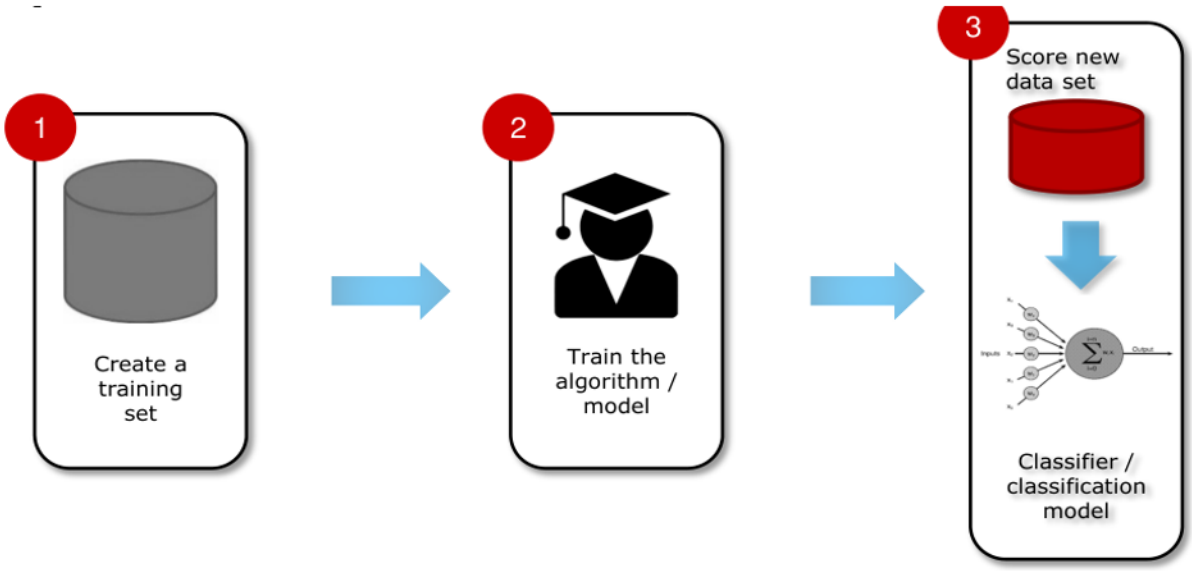
- Develop systems that can automatically adapt and customized themselves to individual users
    - Personalized news, recommendation systems (e.g., Amazon, Netflix), email filters
- Discover new knowledge from large databases (data mining)
    - Market basket analysis (e.g., what products do consumers purchase together?)
    - Make predictions (e.g., which customers will churn, how will revenue develop?)
- Ability to mimic human and replace certain monotonous tasks which require some intelligence
    - Recognizing handwritten characters
    - Categorize unstructured data
    - Automated grading
- Relatively fast and cheap

Statistics vs. ML vs. AI

- **Statistics**
    - Theory-based
    - Focused on testing hypotheses
- **Machine learning**
    - Based on heuristics
    - Focused on improving performance of a learning agent
- **Artificial Intelligence**
    - Machines performing tasks that are characteristic of human intelligence
        - *Examples*: planning, understanding language, recognizing objects/sounds.

ML Modelling Process (3 Steps)

- **Step 1:** Create a training dataset
- **Step 2:** Train the algorithm/model
- **Step 3:** Score new data set + Classification Model



Step 1: Create a training set

Employing a good training set is crucial for any ML project. Without it the algorithm is not trained well and therefore the data won't be classified well.

- ***Minimum requirements:***
  - *Size:* > 10x number of inputs, more in complex non-linear relationships
  - *Good representation:* of all phenomena that can occur
  - *No self-selection effects*

Step 2: Train the algorithm/model

The goal of the algorithm is to obtain relevant insights from the training set.

- Find systematic patterns between variables
  - *Examples:*
    - How can we use customer features to predict fraud?
    - Can we divide customers of InterAmerican into segments?

Step 3: Use trained algorithm to classify new data

The goal is to apply the previously gained insights to new data.

- *Example* Supervised Learning*:*
  - Use trained model to predict fraud (unknown output), based on characteristics of the customer and the situation (known inputs)
- *Example Unsupervised Learning:*
  - Classify new customer of Union in one of the segments that the algorithm identified

ML Model Process – In Practice: Learning to filter spam
Step 1: Creating the training set

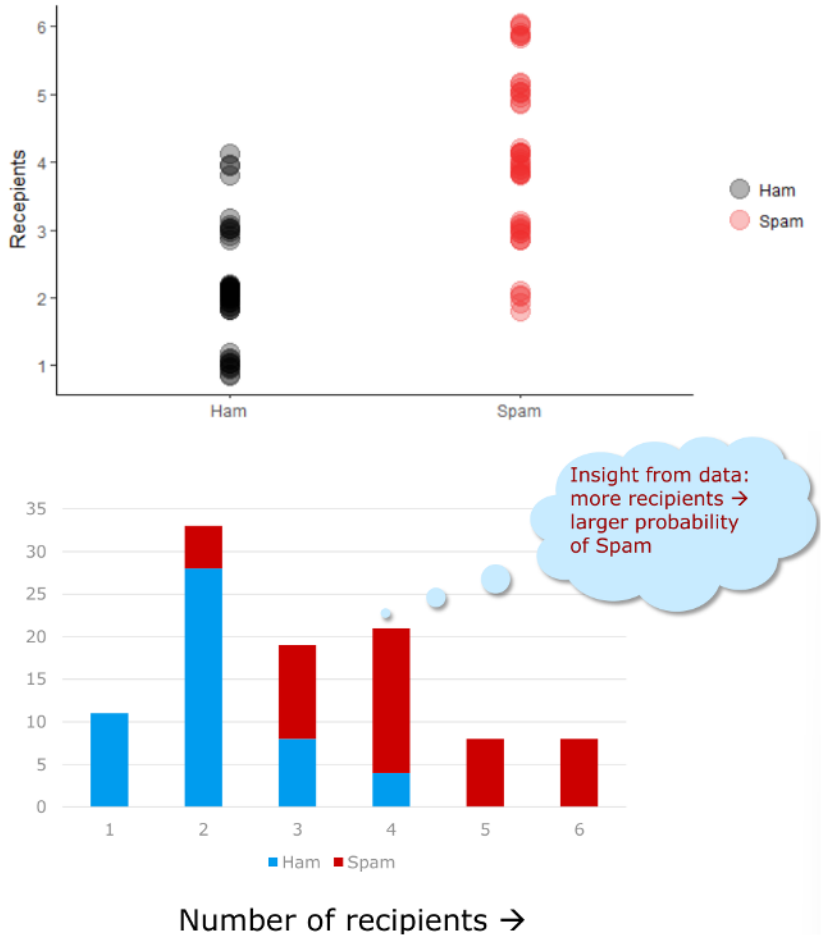
Briefing:

- Spam are all the emails a user does not want to & has not asked to receive
- *Objective:* Identify Spam Emails

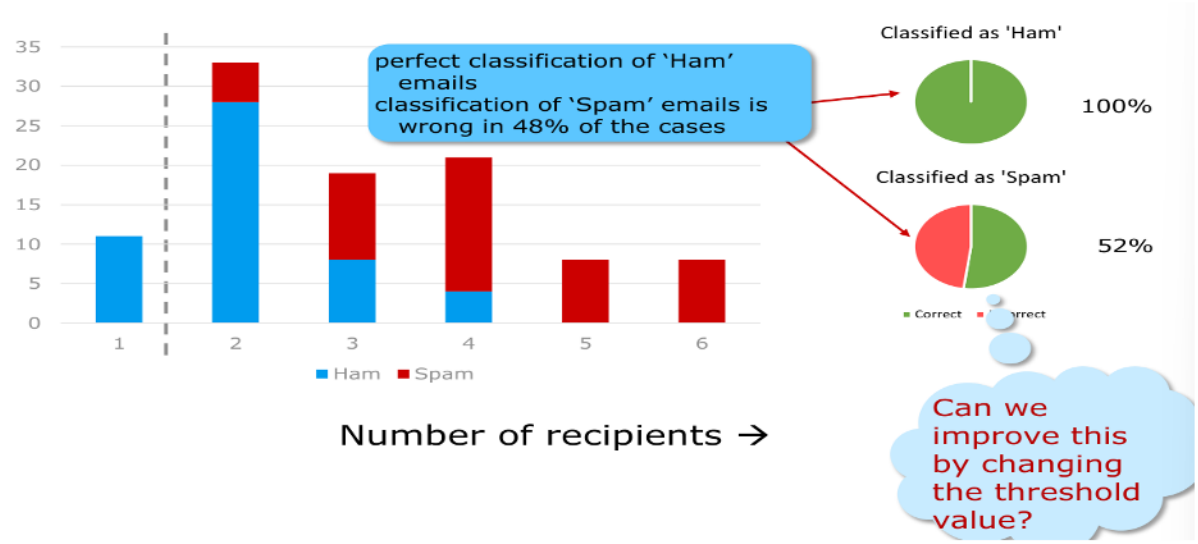
Our data:

- Email type (spam or ham – classified by users)
- Number of recipients
- Email length
- Country (based on IP)
- Customer type
- Wording
- Images (+ host)

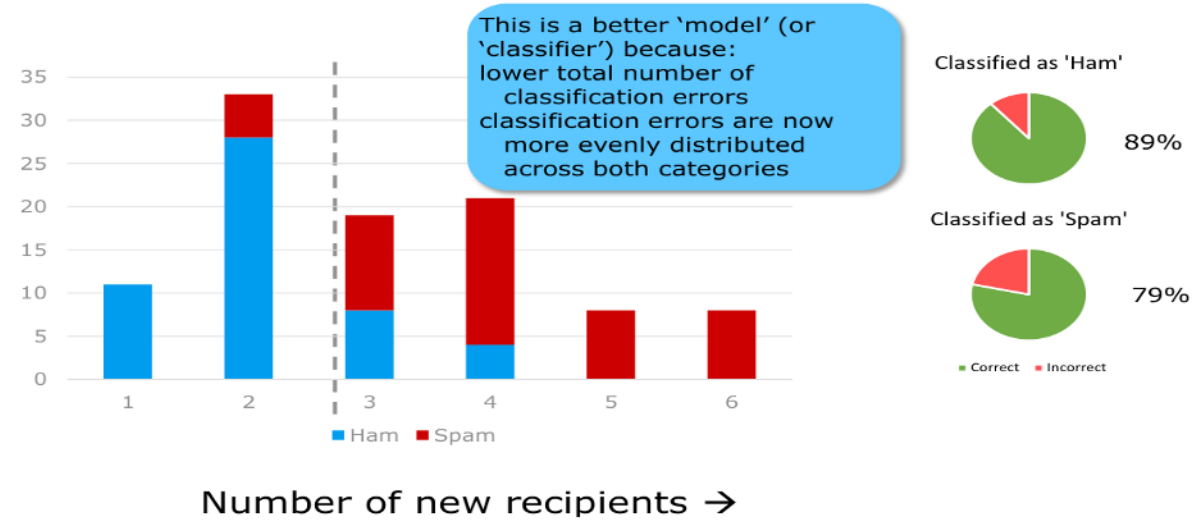
Classification based on number of recipients (1 input)



- First model (or classifier):

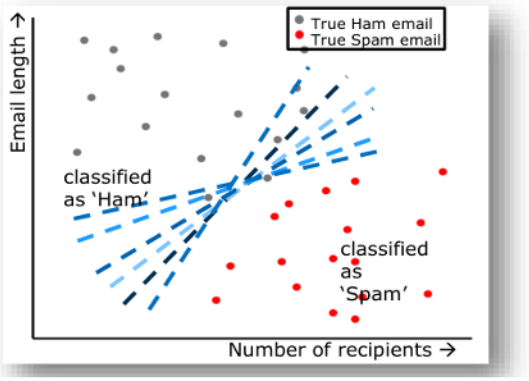


- Second model (better):



Step 2: Learning from the training set – do multiple inputs only recipient number is not robust enough.

› For example number of recipients and email length (number of inputs = 2)

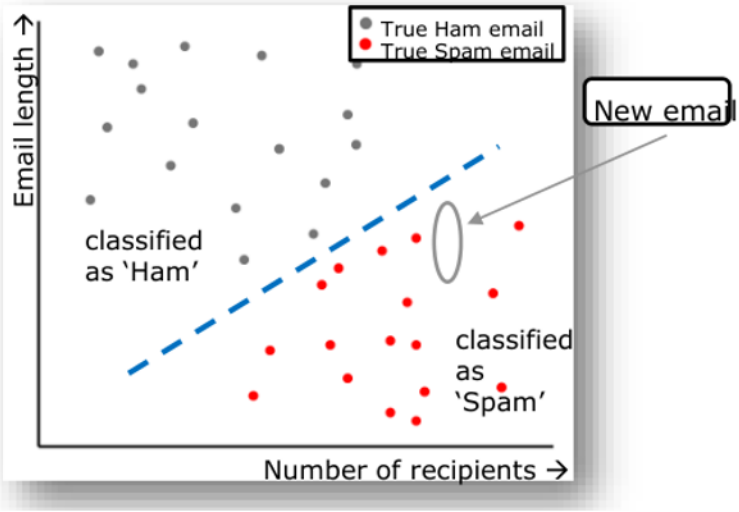


Classification can again be based on a line
classify all emails north-west of it as 'Ham'
classify all emails south-east of it as 'Spam'
Note that this is not a regression line!

- In this case, any of these lines classifies the data perfectly
- 'Training the model' means: finding the best line
- This will be discussed later in this lecture

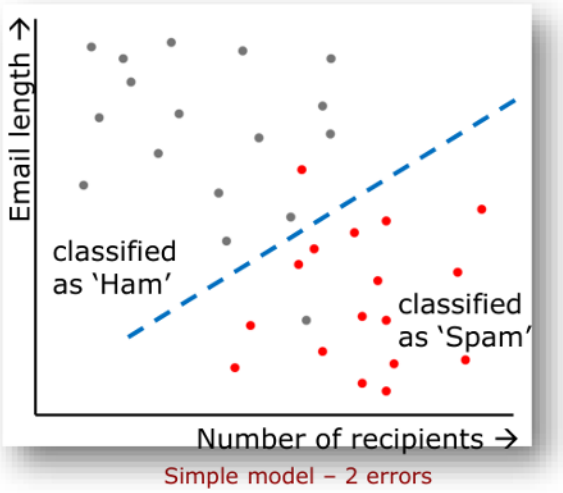
Step 3: Classifying new data

Once the model is trained in step 2 (i.e. when the best line is found), we can use it to classify new data, which has unknown output (Ham/Spam) – new emails.

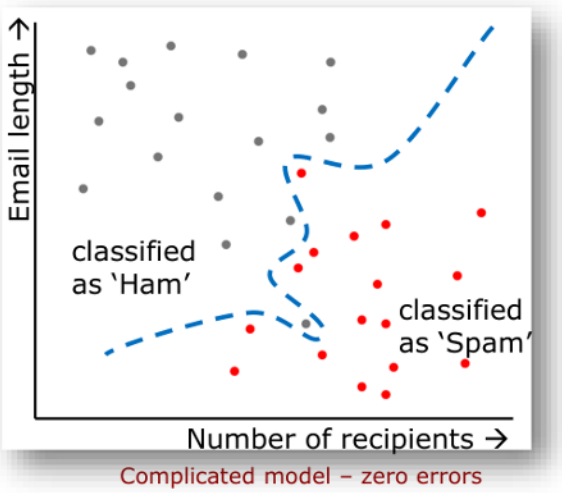


Assessing the ML Process

In most scenarios the reality won't be so perfect.



Simple model – 2 errors

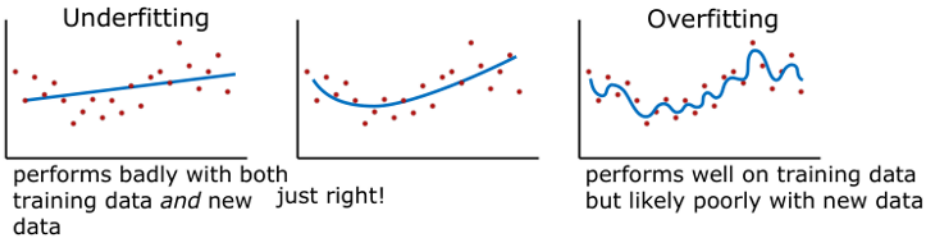


Complicated model – zero errors

The *simpler model* is better, because although it has errors it is likely more suitable for new data.

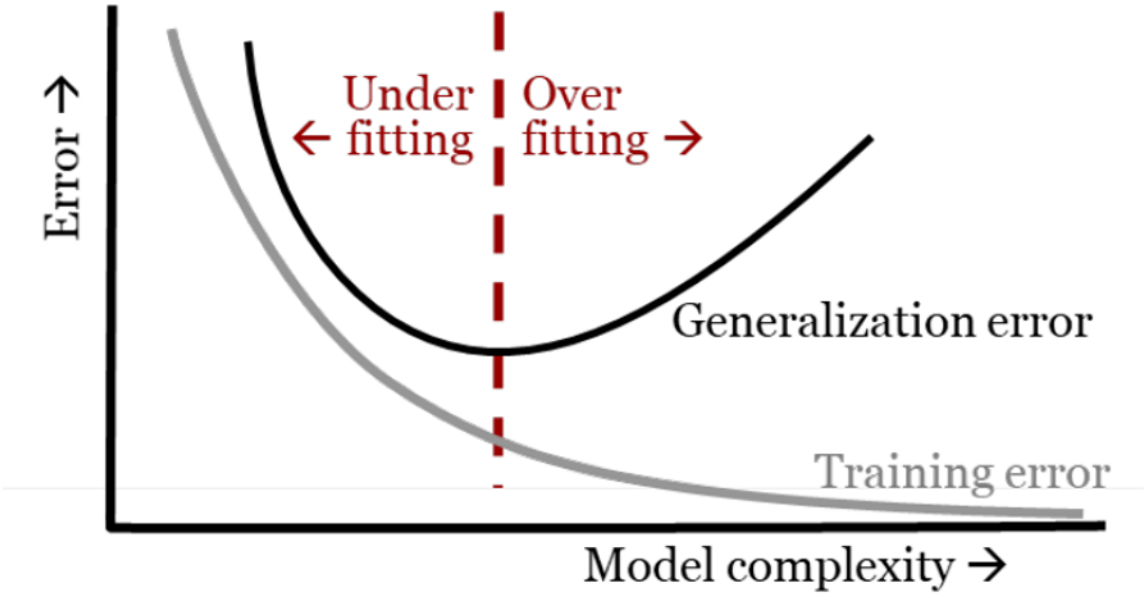
The *complicated model* is an example of **overfitting**, it is perfectly fitted to the training data but that *endangers the usefulness of the model for new data*.

Overfitting & Underfitting



- **Overfitting:** the number of *training errors* decreases, at the cost of increasing the number of errors when the model is applied to new data – also called *generalization error.*
  - The risk of overfitting increases with model complexity.
  - Good fit for training data, bad fit for new data.
  - Lower training error, higher generalization error.
- **Underfitting:** the underlying phenomenon is more complex than what the model can handle.
  - Bad fit for training data, bad fit for new data.
  - High training error, high generalization error.

A good classifier is at the sweet spot where both the training error and the generalization error are small



Measures for assessing model quality

- **To avoid overfitting:** quality of a trained model is assessed by measuring the fit of the model on a *validation set* – contains data that is not used for model training.
- Use the trained model to predict the class to which each observation in the test set belongs to, and determine:
  - **Hit-rate**
  - **Top decile lift**
  - **Lift curve**
  - **GINI coefficient**

Data (pre-)processing

Before starting to estimate models exploring your dataset is crucial.

- Ask yourself questions like:
  - Where do the data come from?
  - How is everything collected?
  - How are the variables measured?
- This gives insights into *what kind of analyses can be done*, what *potential issues and problems there are* (e.g., common method bias, sperrious correlations, missing variables).

› Good data exploration can help tackle issues later on, help improve the final model fit, and understand the model (many machine learning models are "black boxes")

Goal of Data Exploration

Things you want to find out:

- Do all observations make sense?
  - *Examples:*
    - Are there impossible or improbable values? (e.g., age > 120)
    - Are there missing values and is there a good reason?
    - Are variables skewed?
    - Are there (un)expected things going on? (e.g., relations, changes over time)

› Investing time and effort in good data exploration helps resolve problems in the long run.

Steps in Data Exploration

› The steps I typically take when I get a new dataset are:

1. **Discuss with the data provider** where the data comes from, how and when the data are collected, which variables are included, and how everything is measured.
2. Look at the **descriptive statistics** of all of the variables (mean, median, mode, range, std. deviation, skewness)… is it in line with what I expected and are there some impossible or improbable statistics?
3. Make **graphs of individual variables**… do the distributions make sense? How do variables develop over time?
4. Get **statistics on multiple variables** (e.g. correlations)… do they make sense?
5. Make **graphs on multiple variables**: are the joint paterns as expected, do I see outliers or strange paterns here?
6. Start applying simple (regression) models before going to more **complicated methodologies**.

Logistic Regression

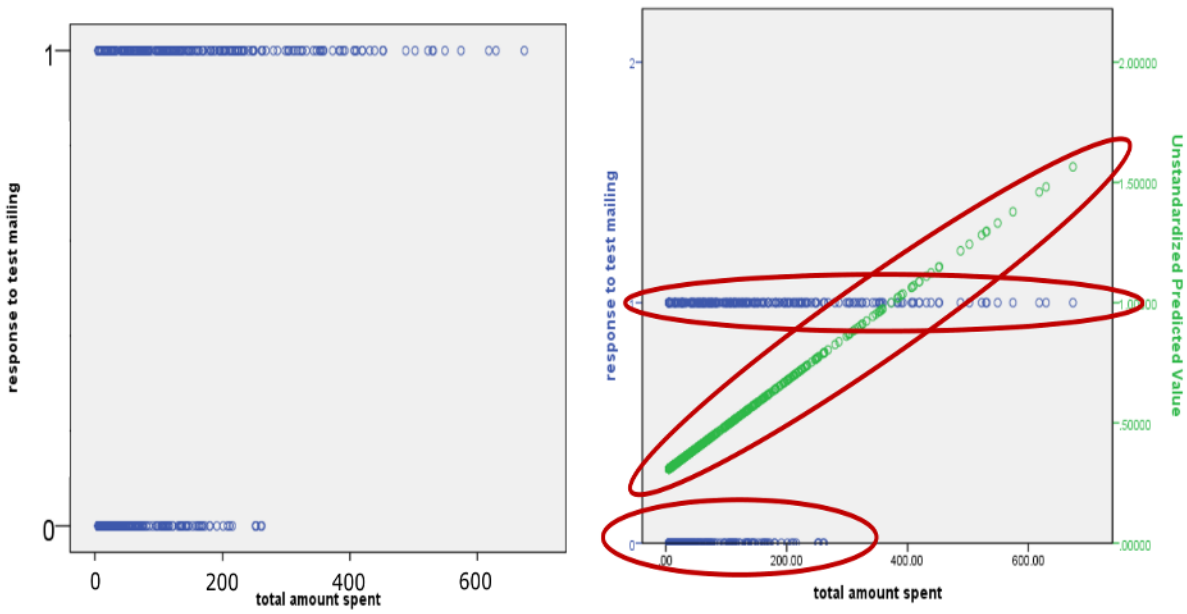
Logistic regression is a supervised classification technique. We use logistic regression when we are dealing with a *continuous independent variables (x)* and a *discrete dependent variable (y).* Furthermore, a logistic regression can be used to make predictions of the *likelihood of an outcome*.

Examples of binary choices

What is the probability that a customer...
... purchases your product?
... cancels his insurance?
... responds to an offer via direct mail?
... files a fraudulent claim?
... adopt the new product that you are launching?
... will pay his bill?

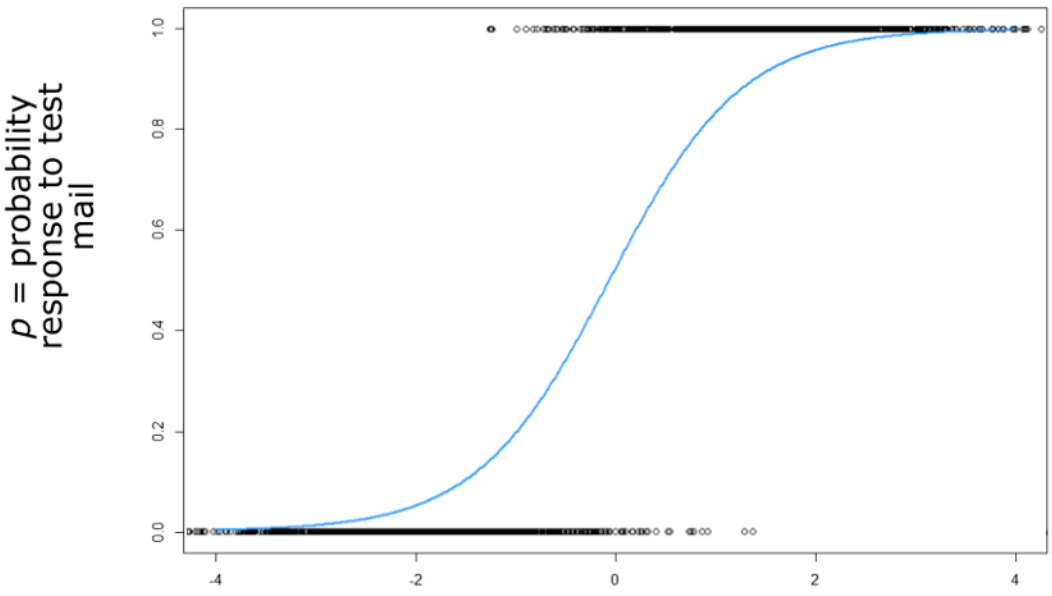
› Many customer-related issues are binary!!

- Why not use linear regression?
  - There is a pattern and no normal distribution
  - Y is 0/1 but the predicted value (x) can be anything



  - The results we get are not very meaningful.

- Now with logistic regression line (better!):



Estimation – Beta's (β)

R chooses the betas in a way to *maximize* the probability (*likelihood*) that the model can generate the observed data points with those β's.

› Maximum likelihood

- Starting point: choose the β's such that the model fits the data as good as possible
- Choose β's such that the probability (likelihood) that this model can generate the observed data points is maximized
  - *≈ when are the calculated probabilities most similar to the observed choices?*
  - *R calculates optimal β's for you based on this principle!*
  - *Numerical optimization*

Interpretation

- In an **OLS regression**, the parameters (*x*) can be interpreted as:
  - One unit increase in *x* results in a β units increase in *y*
- In a **log-regression**, the most important is the sign (- +), which indicates the direction of the effect.
  - When +*x* goes up the chance to *y* (e.g., purchase) increases
  - When -*x* goes up the chance to *y* decreases

➢ If you want to interpret the β in a log-regression you need to *take the exponential of* β, which results in the ***odds ratio***.

- E.g.: if the parameter 'b' for female is .288 in predicting sales, females have exp(.288) = 1.333 higher odds to buy than males, i.e. the odds are 33.3% higher. Example when you would find this result:
  - *20% of males buy and 25% of females buy*
  - *Odds(males) = .20 / .80 = .25 and Odds(females) = .25 / .75 = .33*
  - *Odds ratio = .33 / .25 = 1.333*
- E.g.: if the parameter 'b' for age is -.065 in predicting sales, each 1 unit (i.e. 1 year) increase in age results in a (exp(-.065) - 1 = 0.937 – 1 = -.063) 6.3% decrease in the odds of buying.

In Practice – Titanic Data

- Prepare the data

```
> # Open Titanic data
> Titanic <- read.csv("C:/…/Data titanic.csv")

> # Get descriptives
> summary(Titanic)

> # Create dummy for age is mising
> Titanic$age_missing <- ifelse(is.na(Titanic$age), 1, 0)

> # Set missing age to mean value
> Titanic$age <- ifelse(is.na(Titanic$age), mean(Titanic$age,
na.rm=TRUE), Titanic$age)
```

> In R, you can estimate this with the following function:

- Logistic_regression1 <- glm(survived ~ as.factor(sex) + as.factor(pclass) + age, family=binomial, data=Titanic)

> Where glm stands for "Generalized Linear Models"

> The variable "survived" is the name of our dependent variable

> The variables "sex" and "pclass" and "age" are independent variables.

> The part "family=binomial" tells R that the dependent variable is binary, so R knows it has to estimate a logistic regression model.

> The part "data=Titanic" tells R on which dataset to estimate this model.

- summary(Logistic_regression1)

```
Call:
glm(formula = survived ~ as.factor(sex) + as.factor(pclass) +
    age, family = binomial, data = Titanic)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.5782  -0.6848  -0.4440   0.6692   2.3746

Coefficients:
                    Estimate Std. Error z value Pr(>|z|)
(Intercept)         3.351788   0.300177  11.166  < 2e-16 ***
as.factor(sex)male -2.497729   0.148652 -16.803  < 2e-16 ***
as.factor(pclass)2 -1.190934   0.210305  -5.663 1.49e-08 ***
as.factor(pclass)3 -2.152335   0.194403 -11.072  < 2e-16 ***
age                -0.032436   0.006089  -5.327 9.97e-08 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 1741.0  on 1308  degrees of freedom
Residual deviance: 1227.5  on 1304  degrees of freedom
AIC: 1237.5

Number of Fisher Scoring iterations: 4
```

Deciding on IVs

- Hypothesize which variables are important to be included
- Hypothesize about the direction and functional form
- Estimate the model based on *theory* and *managerial knowledge*

Model Validation (1) – Making Predictions (in R)

As mentioned previously a log-reg model can be used to make prediction of the likelihood of an outcome.

> In R, with the logistic regression model estimated in the previous lecture, we can simply get the predictions with the following function:

```
predictions_model1 <- predict(Logistic_regression1, type = "response", newdata=Titanic)
```

> Here we create a new variable, "predictions_model1", which provides us the probability that the variable "survived" has a value of 1 (vs. 0) according to the estimated model.

Model Validation (2) – 3 Forms of Validation Criteria

We can use the predictions to validate the model – see how good the model is able to make predictions. We will examine *3 forms of validation criteria*:

- Hit Rate
- Top decile lift
- Gini Coefficient

Hit Rate (1) – Interpretation & Calculation

The hit ratio/rate indicates how many observations are correctly predicted.

- Instead of the overall hit ration you can also look at the hit ratio per group (e.g., how many churners and how many retainers are correctly identified)
- In general, when the *estimate probability* is >.50 (*the cut-off*), it is predicted that the outcome is positive (1), otherwise <.50 = negative (0)

		Observed		
		Positive (1)	Negative (0)	
Predicted	Positive (1)	True positive (a)	False positive (b)	a/(a+b) % predicted positves correct
	Negative (0)	False negative (c)	True negative (d)	d/(c+d) % predcited negatives correct
		a/(a+c) % positive correctly predicted (**sensitivity**)	d/(b+d) % negative correctly predicted (**specificity**)	(a+d)/(a+b+c+d) % correct = **hit ratio (or accuracy)**

Hit Rate (2) – How to in R

```
> predicted_model1 <- ifelse(predictions_model1>.5,1,0)

> hit_rate_model1 <- table(Titanic$survived,
  predicted_model1, dnn= c("Observed", "Predicted"))

> hit_rate_model1

> #Get the hit rate
> (hit_rate_model1[1,1]+hit_rate_model1[2,2])/sum(hit_rat
  e_model1)
```

Top Decile Lift (1) – Interpretation & Calculation

The TDL (%) is the ratio of predicted churn rate per group divided by the actual overall churn rate.

- Use the trained model to predict *p*(churn)
- use *p*(churn) to rank all customers (high – low risk)
- divide customers in ten groups → group 1: highest *p*(churn)

$$\text{TDL} = \frac{\text{actual churn rate of group 1}}{\text{actual overall churn rate}} \times 100\%$$

- If **TDL = 1,** model is not better than random selection
- If **TDL = 2**, model is twice as good in predicting churners

Lift table and TDL

	Decile	Upper cut-off	Lower cut-off	Mean predicted	Mean actual	Cum. Churn	Lift
Top churn	1	.885	.459	.551	.542	20.8%	2.08
	2	.459	.378	.411	.420	36.9%	1.61
	3	.377	.315	.342	.354	50.5%	1.36
	4	.315	.270	.292	.305	62.2%	1.17
	5	.270	.232	.251	.272	72.6%	1.04
	6	.232	.195	.214	.219	81.0%	.84
	7	.195	.162	.178	.165	87.3%	.63
	8	.162	.130	.146	.145	92.9%	.56
	9	.130	.093	.111	.121	97.5%	.46
Bottom churn	10	.093	.001	.068	.065	100.0%	.25
				Mean:	**.261**		

.542 / .261 = 2.08

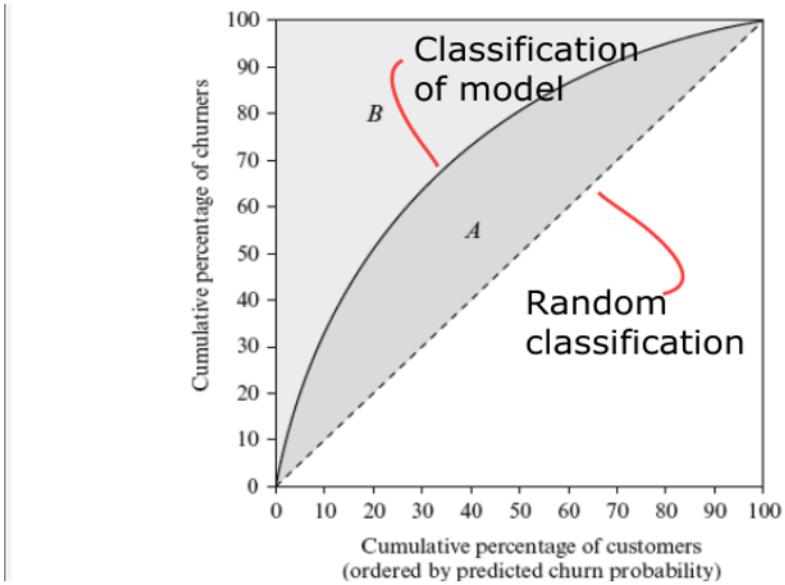
.065 / .261 = .25

Top Decile Lift (2) – How to in R

```
> #TDL table
> library(dplyr)

> decile_predicted_model1 <- ntile(predictions_model1, 10)

> decile_model1 <- table(Titanic$survived, decile_predicted_model1, dnn=
c("Observed", "Decile"))

> decile_model1

> #Get the TDL
> (decile_model1[2,10] / (decile_model1[1,10]+ decile_model1[2,10])) /
mean(Titanic$survived)
```

Top Decile Lift (3) - Lift Curve: Interpretation

The further the solid line is in the upper-lift corner, the better the model.



Top Decile Lift (3) - Lift Curve: How to in R

```
> install.packages("ROCR")
> library(ROCR)

> pred_model1 <- prediction(predictions_model1,
ourdataset$y)
> perf_model1 <- performance(pred_model1,"tpr","fpr")
> plot(perf_model1,xlab="Cumulative % of
observations",ylab="Cumulative % of positive
cases",xlim=c(0,1),ylim=c(0,1),xaxs="i",yaxs="i")
> abline(0,1)
> auc_model1 <- performance(pred_model1,"auc")
```

GINI Coefficient (1) – Interpretation & Calculation

GINI focuses on overall performance, while TDL focuses on predicting top cases.

- **GINI ≈ 0**, model is not better than random selection
- **GINI closer to 1**, the better the prediction power

GINI Coefficient (2) – How to in R

```
> #The Gini is related to the "Area under the Curve" (AUC), namely by: Gini = AUC*2 – 1

> #So to get the Gini we do:

> as.numeric(auc_model1@y.values)*2-1
```

Fit Criteria (1) – Calculation

One way of calculating the *fit criteria* discussed before is estimate the logistic regression model on the sample, and after that calculate the probabilities per customers and with that the fit criteria. (All on the same sample)

- A disadvantage of this is that the model is fitted and optimized on this dataset. If you use more independent variables, the in-sample fit will most likely go up, although the model's predictions may in fact become worse (something that is called 'over fitting').

Fit Criteria (2) – Calculation: Solving overfitting

To solve over fitting, you can use part of the sample to *estimate* the model, and use another part of the sample (or a new sample) to *validate* the model.

The *estimated* log-regression is in this case used to calculate the probabilities to churn for observation in the *validation* sample, which are then use to calculate the *fit criteria*.

- It is more important that the model performs well out-of-sample than in-sample.
- If the model performs much better in-sample, the model is likely over fitted (especially likely when using complicated models on small samples or when sample is unbalanced).

Fit Criteria (3) – How to in R

```
> #Get a 75% estimation sample and 25% validation sample
> set.seed(1234)
> Titanic$estimation_sample <-rbinom(nrow(Titanic), 1,
  0.75)

> #Estimate the model using only the estimation sample
> Logistic_regression2 <- glm(survived ~ as.factor(sex) +
  as.factor(pclass) + age,, family=binomial, data=Titanic,
  subset=estimation_sample==1)
> #Create a new dataframe with only the validation sample
> our_validation_dataset <-
  Titanic[Titanic$estimation_sample==0,]

> #Get predictions for all observations
> predictions_model2 <- predict(Logistic_regression2, type =
  "response", newdata= our_validation_dataset)

> #Calculate the fit criteria on this validation sample
```

Balanced vs. Unbalanced Sample

The fit metrics can be sensitive for the balance of the sample in terms of positive and negative case (e.g., spam or ham).

- E.g., if 90% of you sample is spam, it's easy to have a hit ratio of 90%

› To overcome this:

- Set a different cut-off value (e.g., a predicted probability of >.90 is classified as a positive case, otherwise it is negative)
- Use a balanced sample
- Use a weighted sample
- Be careful interpreting the validation criteria