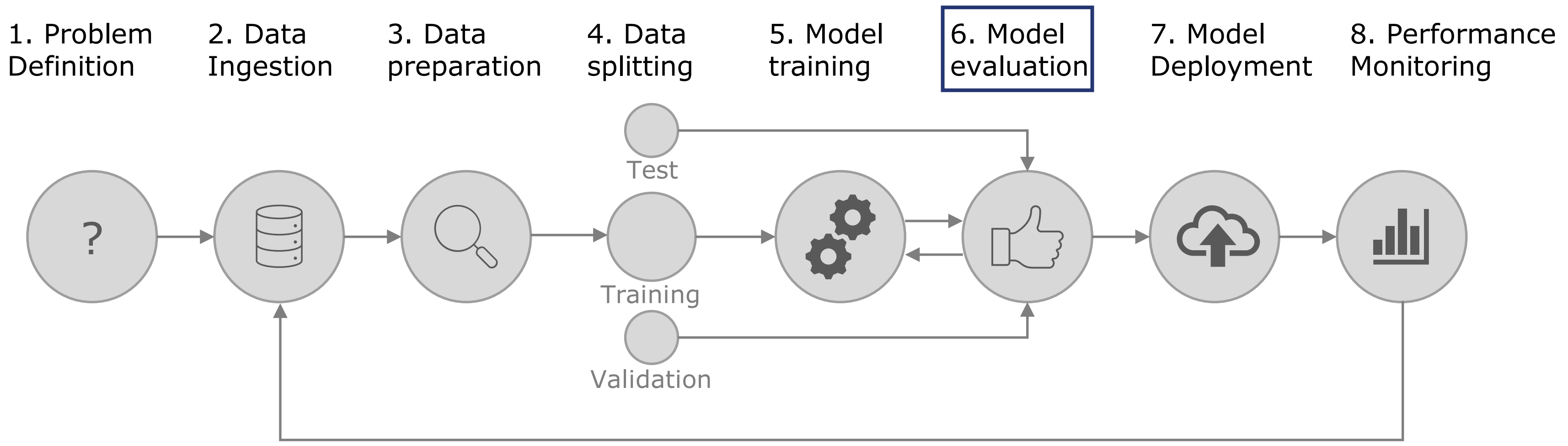
IMC FH KREMS
UNIVERSITY OF APPLIED SCIENCES / AUSTRIA

Machine Learning, Artificial Intelligence, and Big Data Analytics (IL, 4th Semester)

Lecture 2

Focus of this lecture



4 Data splitting= training test cross validation and separating the data

Topics of today

- How to evaluate a ML model
- Coding:
  - Loading data
  - Manipulating with data.table
  - Data splitting and model evaluation

FH KREMS
UNIVERSITY OF APPLIED SCIENCES / AUSTRIA

MAE = Mean absolute error
MAPE = Mean absolute percentage error
RMSE = Root mean squared error
R2score =

Evaluating a model

What to use for evaluating a **regression** model?

- MAE $= \frac{1}{n}\sum_{i=1}^{n}|e_i|$ ← Just the average absolute error (0 means perfect fit)
- MAPE $= \frac{1}{n}\sum_{i=1}^{n}\left|\frac{e_i}{y_i}\right|$ ← The average error in relation to the actual values (0% means perfect fit) it is not about values but percentages
- RMSE $= \sqrt{\frac{1}{n}\sum_{i=1}^{n}|e_i^2|}$ ← The average error but penalizes larger errors more severely (0 means perfect fit)
- R²-score $= 1 - \frac{RSS}{TSS}$ ← The degree to which the model explains the variance in the data (1 means perfect fit. 0 is no better than the mean. <0 is worse than the mean)

- Very easy to compute. R, Python, and Julia also provide built-in functions and usually include these metrics in the model object (from the training data).
- You should know these from the statistics lecture!
- What about **classification**?

Evaluating a model

What to use for evaluating a **Classification** model?

- Back to the spam detection example

Actual	Prediction
No spam	Spam
No spam	No spam
No spam	Spam
No spam	No spam
No spam	No spam
Spam	No spam
Spam	Spam
Spam	Spam
Spam	Spam
Spam	Spam

Evaluating a model

What to use for evaluating a **Classification** model?

- Back to the spam detection example

Actual	Prediction
No spam	Spam
No spam	No spam
No spam	Spam
No spam	No spam
No spam	No spam
Spam	No spam
Spam	Spam
Spam	Spam
Spam	Spam
Spam	Spam

Confusion Matrix

Predicted

Actual		1	0
	1	True positive	False Negative
	0	False Positive	True negative

Evaluating a model

What to use for evaluating a **Classification** model?

- Back to the spam detection example

Actual	Prediction
No spam	Spam
No spam	No spam
No spam	Spam
No spam	No spam
No spam	No spam
Spam	No spam
Spam	Spam
Spam	Spam
Spam	Spam
Spam	Spam

Confusion Matrix

Predicted

Actual		1	0
	1	True positive	False Negative
	0	False Positive	True negative

	Spam	No spam
Spam	4	1
No spam	2	3

Evaluating a model

What to use for evaluating a **Classification** model?

Precision = Y axis
Recall = X axis

- Back to the spam detection example

Actual	Prediction
No spam	Spam
No spam	No spam
No spam	Spam
No spam	No spam
No spam	No spam
Spam	No spam
Spam	Spam
Spam	Spam
Spam	Spam
Spam	Spam

Confusion Matrix

Predicted

Actual		1	0
	1	True positive	False Negative
	0	False Positive	True negative

	Spam	No spam
Spam	4	1
No spam	2	3

Accuracy: What fraction does it get right
(#TP+#TN)/#Total

Precision: When it says 1 how often is it right Sensitivity
#TP/(#TP+#FP)

Recall: What fraction of 1s does it get right Specificity
#TP/(#TP+#FN)

FP Rate: What fraction of 0s are called 1s
#FP/(#FP+#TN)

FN Rate: What fraction of 1s are called 0s
#FN/(#TP+#FN)

F1-Score: $2 * \frac{precision * recall}{precision + recall}$

Evaluating a model

What to use for evaluating a **Classification** model?

- Back to the spam detection example

Actual	Prediction
No spam	Spam
No spam	No spam
No spam	Spam
No spam	No spam
No spam	No spam
Spam	No spam
Spam	Spam
Spam	Spam
Spam	Spam
Spam	Spam

Confusion Matrix

Predicted

Actual		1	0
	1	True positive	False Negative
	0	False Positive	True negative

	Spam	No spam
Spam	4	1
No spam	2	3

Accuracy: What fraction does it get right
(#TP+#TN)/#Total = 7/10 = 70%

Precision: When it says 1 how often is it right
#TP/(#TP+#FP) = 4/6 = 66%

Recall: What fraction of 1s does it get right
#TP/(#TP+#FN) = 4/5 = 80%

FP Rate: What fraction of 0s are called 1s
#FP/(#FP+#TN) = 2/5 = 40%

FN Rate: What fraction of 1s are called 0s
#FN/(#TP+#FN) = 1/5 = 20%

F1-Score: $2 * \frac{precision*recall}{precision+recall} = 0.72$

Evaluating a model

The importance of looking at different metrics

- Imagine the following

Actual	Prediction
No spam	No spam
No spam	No spam
No spam	No spam
No spam	No spam
No spam	No spam
No spam	No spam
No spam	No spam
No spam	No spam
Spam	No spam
Spam	Spam

Predicted

Actual		Spam	No spam
	Spam	TP=1	FN=1
	No spam	FP=0	TN=8

Accuracy: What fraction does it get right
(#TP+#TN)/#Total = 9/10 = 90%

Precision: When it says 1 how often is it right
#TP/(#TP+#FP) = 1/1 = 100%

FP Rate: What fraction of 0s are called 1s
#FP/(#FP+#TN) = 0%

We also need to

Exam question: Given a table compute all the metrics, typical error = verify where is predicted and where is actual
Given that a model has to detect a disease known accuracy, can you say if the model is good or bad?

Evaluating a model

The importance of looking at different metrics

- Imagine the following

Actual	Prediction
No spam	No spam
No spam	No spam
No spam	No spam
No spam	No spam
No spam	No spam
No spam	No spam
No spam	No spam
No spam	No spam
Spam	No spam
Spam	Spam

Predicted

Actual	Spam	No spam
Spam	TP=1	FN=1
No spam	FP=0	TN=8

It says 100% but we predicted that it is SPAM only once which is small, than we need to present Recall and F1-Score

Accuracy: What fraction does it get right
(#TP+#TN)/#Total = 9/10 = 90%

Precision: When it says 1 how often is it right
#TP/(#TP+#FP) = 1/1 = 100%

Recall: What fraction of 1s does it get right
#TP/(#TP+#FN) = 1/2 = 50%

FP Rate: What fraction of 0s are called 1s
#FP/(#FP+#TN) = 0%

FN Rate: What fraction of 1s are called 0s
#FN/(#TP+#FN) = 1/2 = 50%

F1-Score: $2 * \frac{precision * recall}{precision + recall}$ = 0.66

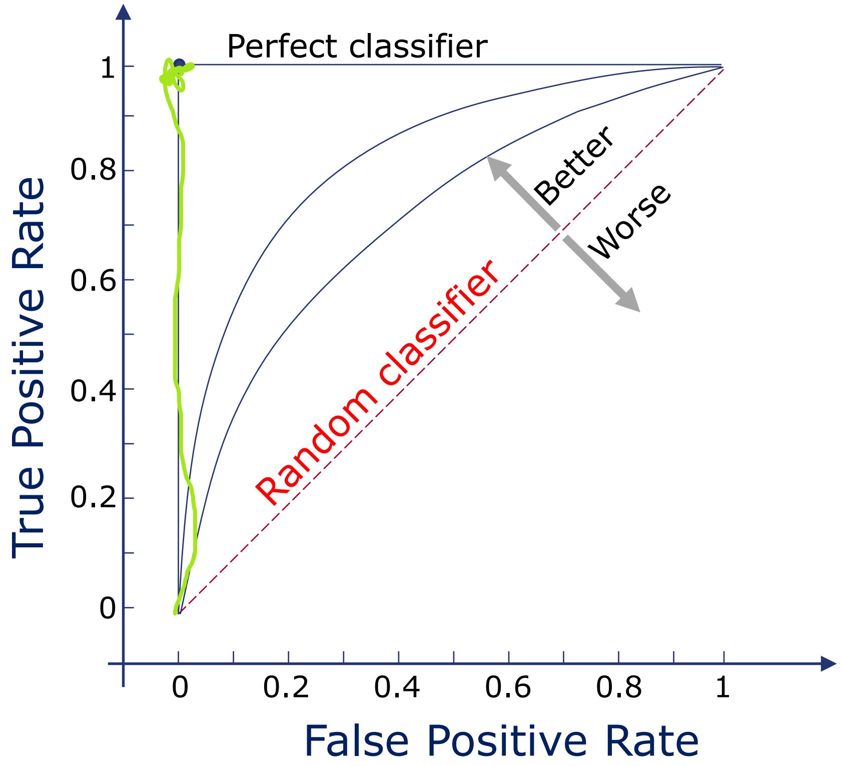
0.6 0.7 is good

Receiver operator characteristic = It plots the false positive rate (FPR), against true positive rate (TPR) X axis = TPR, Y axis = FPR

The ROC curve and the AUC

- Comparing binary classifiers
- True Positive vs. False Positive at various thresholds

perfect

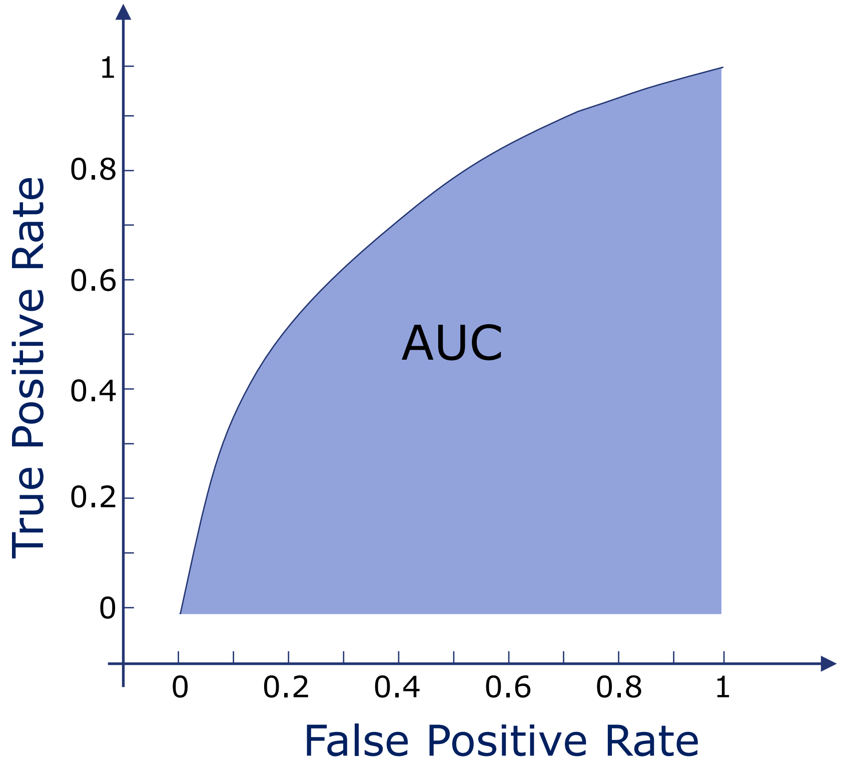


AUC = area under the curve

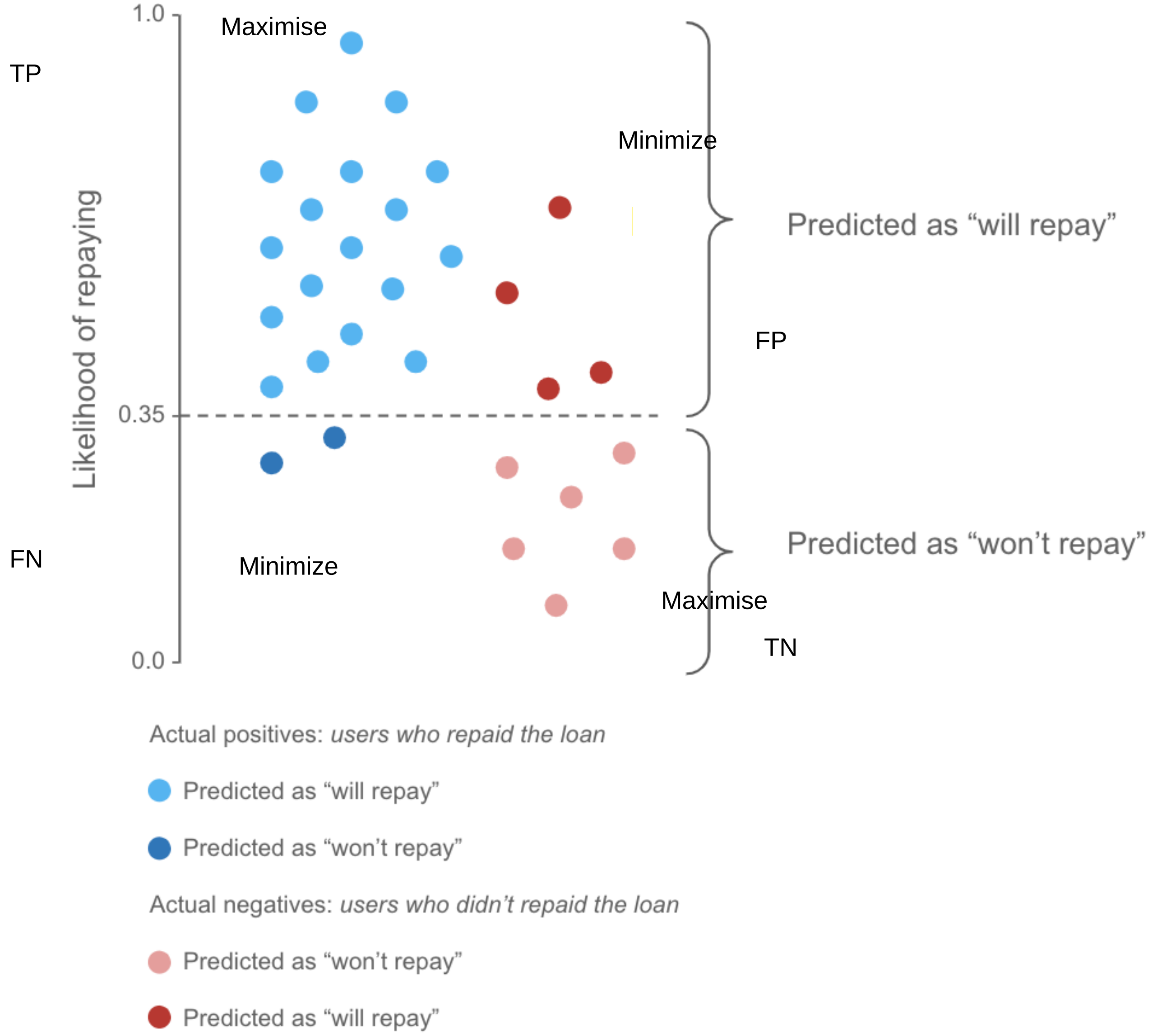
The ROC curve and the AUC

- Comparing binary classifiers
- True Positive vs. False Positive at various thresholds

- 0 < AUC < 1
- The larger the better

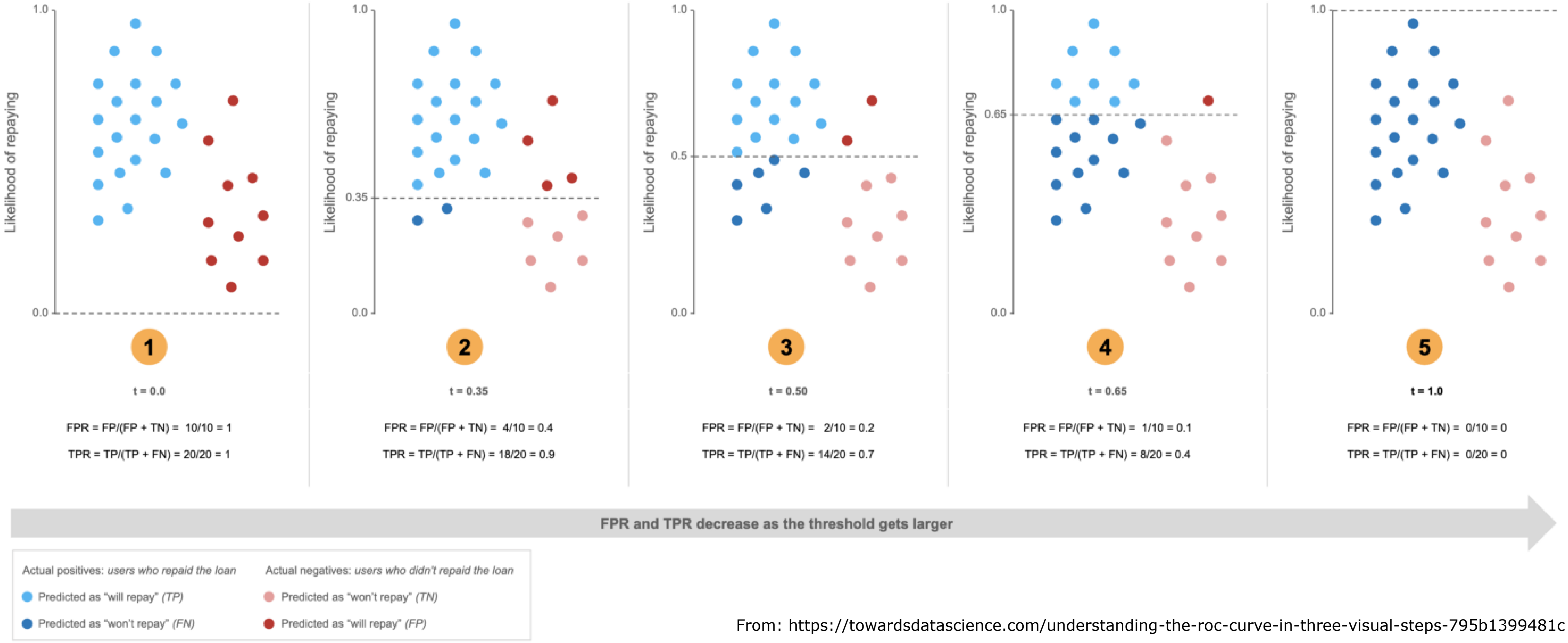


ROC example



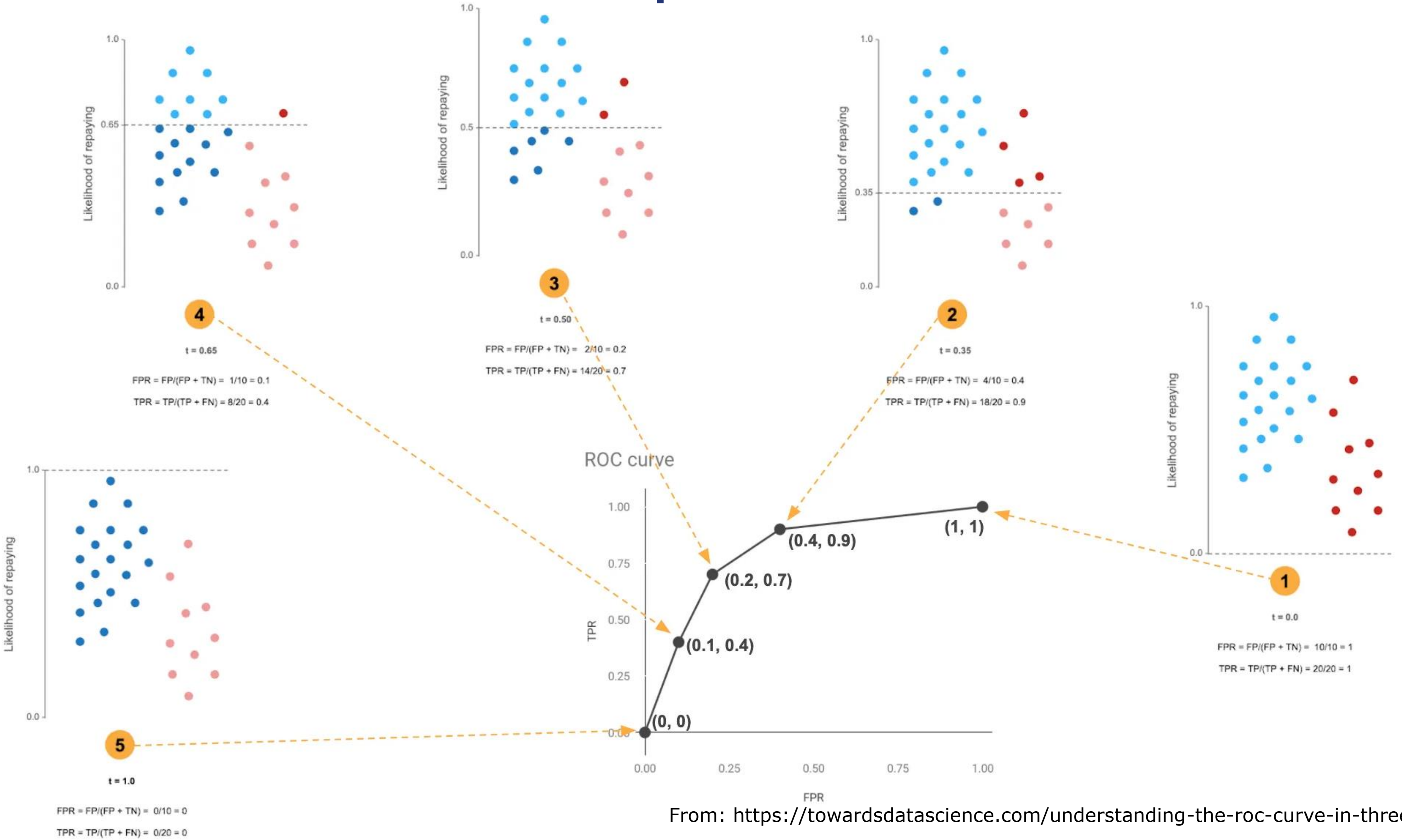
https://towardsdatascience.com/understanding-the-roc-curve-in-three-visual-steps-795b1399481c

ROC example



From: https://towardsdatascience.com/understanding-the-roc-curve-in-three-visual-steps-795b1399481c

ROC example



From: https://towardsdatascience.com/understanding-the-roc-curve-in-three-visual-steps-795b1399481c

Evaluating a model

Summary

Metric	Formula	Meaning	Visual look	range
Accuracy	(#TP+#TN)/#Total	What fraction does it get right	**TP** FN FP **TN** / **TP** **FN** **FP** **TN**	0-$\underline{1}$
Precision	#TP/(#TP+#FP)	When it says 1 how often is it right	**TP** FN FP TN / **TP** FN **FP** TN	0-$\underline{1}$
Recall/ Sensitivity	#TP/(#TP+#FN)	What fraction of 1s does it get right (True Positive Rate – TPR)	**TP** FN FP TN / **TP** **FN** FP TN	0-$\underline{1}$
Specificity	#TN/(#TN+#FP)	What fraction of 0s does it get right (True Negative Rate – TNR)	TP FN FP **TN** / TP FN **FP** **TN**	0-$\underline{1}$
FP Rate	#FP/(#FP+#TN)	What fraction of 0s are called 1s	TP FN **FP** TN / TP FN **FP** **TN**	$\underline{0}$-1
FN Rate	#FN/(#TP+#FN)	What fraction of 1s are called 0s	TP **FN** FP TN / **TP** **FN** FP TN	$\underline{0}$-1
F1-score	$2 * \frac{precision*recall}{precision+recall}$	How "good" are precision and recall		0-$\underline{1}$

Things you should know

100% will be in exam

- What is underfit/overfit. What is the bias-variance tradeoff. How do they relate?
- How does cross-validation work.
- What is bootstrapping and bagging.
- How to evaluate a regression or a classification model
  - RMSE, MAE, …
  - Accuracy, Precision, Recall,…
  - Interpret a ROC curve

bootstrapping = sampling data with replacement from a given dataset
Bagging = training model on different parts of data

- It continues in R

Feedback round

- Scan the barcode from your mobile phone

OR

- go to http://sli.do and insert this code: 19651

and follow my instructions.

Exercise 1
Overfit

1. **Load** the dataset wines.csv (or any other regression dataset from here – i.e., Regression task, numerical variables)
2. **Explore and visualize the dataset** (e.g., how many observations? How many features? Missing values? Are some features irrelevant?)
3. Crete a regression model (i.e., for wine: the quality by using density, chlorides, and volatile acidity).
   1. Split the data into training and test set
   2. Create a linear regression model and polynomial models with increasing degree.
   3. What’s the MAE, the RMSE, and the MAPE on the training and test set for all the models?
   4. When does the model start overfitting? Which degree would you choose?

- **Due date: March 12th, 23.59 CET (Late submission +1week, 6 pts)**
- Comment code and results (or write a notebook).
- Use R or Python