Technische Universität München

Business Analytics

Introduction to the DMC

Decision Sciences & Systems (DSS)
Department of Informatics
TU München

Tutorial Business Analytics

Outline

Today's topics:

- Dates & Grading for Data Mining Cup
- Rules of Data Mining Cup
- Steps of Data Mining Cup
- Example dataset + Script
- Presentation of datasets for DMC 1 & 2

Tutorial Business Analytics

Dates & Grading for Data Mining Cups 1 and 2

Date

- 19th of December 2016 at 6 pm until 23rd of January 2017 at 12 am: Data Mining Cups 1 and 2
- 20-23th of December 2016: DMC support tutorial
- 30th of January 2017: Announcement of the (two) best teams per DMC to present their results
- 6st of February 2017: Data Mining Cup Presentation in the final lecture

Grading **for each of the two** DMCs

- Best 25%: +6 points
- Next 25%: +4 points
- Next 25%: +2 points

Note: Only submissions performing at least as good as the “Zero-Rule”-Classifier are taken into account for the ranking.

Tutorial Business Analytics

Rules of Data Mining Cups 1 and 2

Teams

- Team size: 1 – 4 members.
- Teams must be built before the first submission (teams will be fixed after first submission!).
- Each student can only be member of one team within one Data Mining Cup.
- Teams can be different for Data Mining Cups 1 and 2.

Submissions

- Maximum number of valid submissions for each DMC: 10.
- Best ranked submission, **only**, will be taken into account for the ranking.
- For reasons of traceability you must use a fixed seed of 42 (`set.seed(42)`).

Disqualification reasons:

- **Non-reproducible** submissions (submitted predictions **must be reproducible** using the submitted R script)
- **Hard-coded** classifications (even if the best ranked submission is not hard-coded!)
- **Copies** from other groups (disqualification of both teams)

Tutorial Business Analytics

Steps of Data Mining Cups

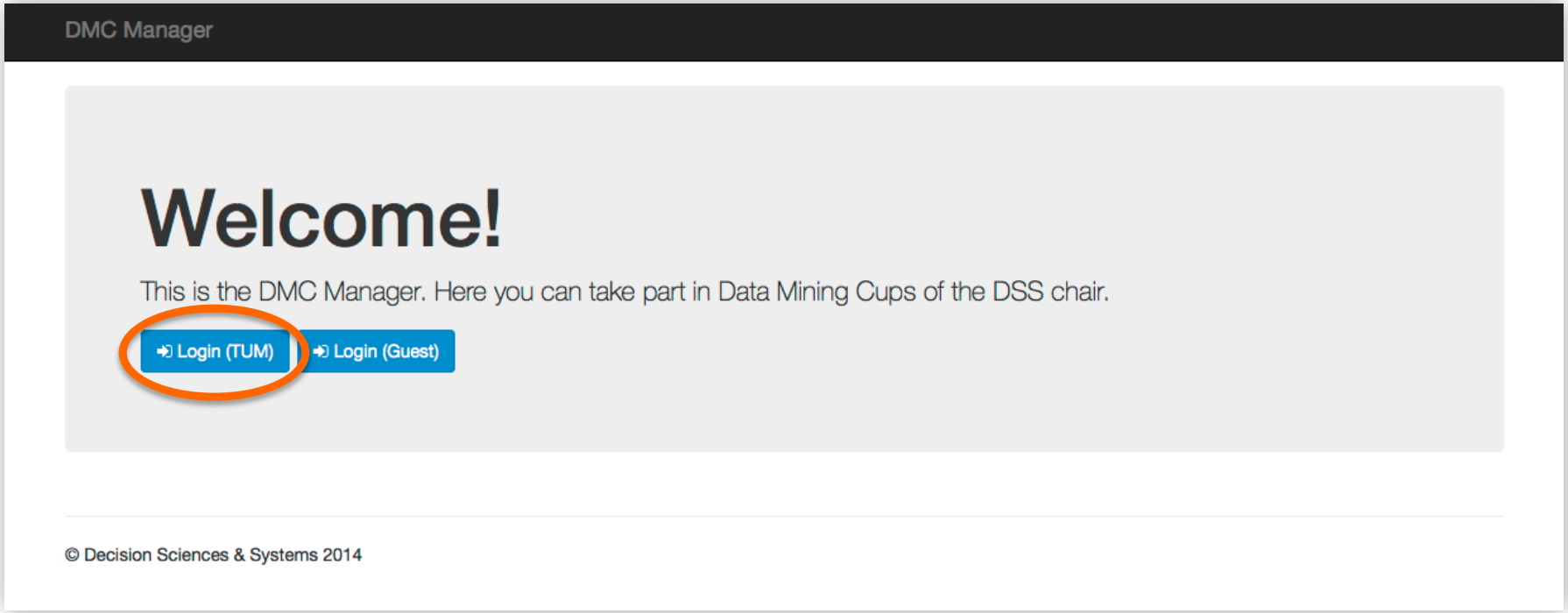
1. Build a Team in the DMC Manager
2. Load & Explore the Data Set
   - Summary statistics
   - Plotting
3. Data Preparation
   - Feature Selection
   - Discretization
4. Training & Evaluation
   - Classification Methods
   - Metrics
   - Resampling Methods
5. Predict Classes in Test Data
6. Export the Predictions
7. Upload the Predictions and the Corresponding R Script on DMC Manager

Source: http://topepo.github.io/caret/

1. Build Team in DMC Manager

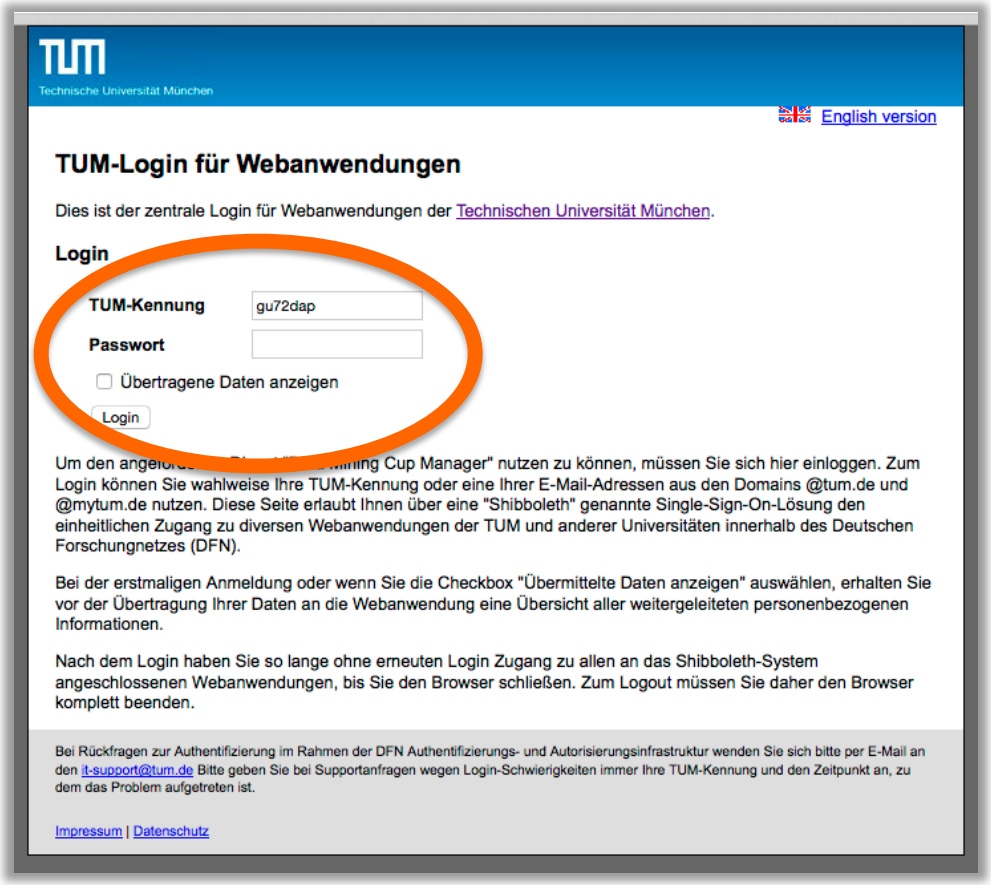
Login with your TUM login data (“TUM Kennung”)

https://dmc.dss.in.tum.de/dmc/



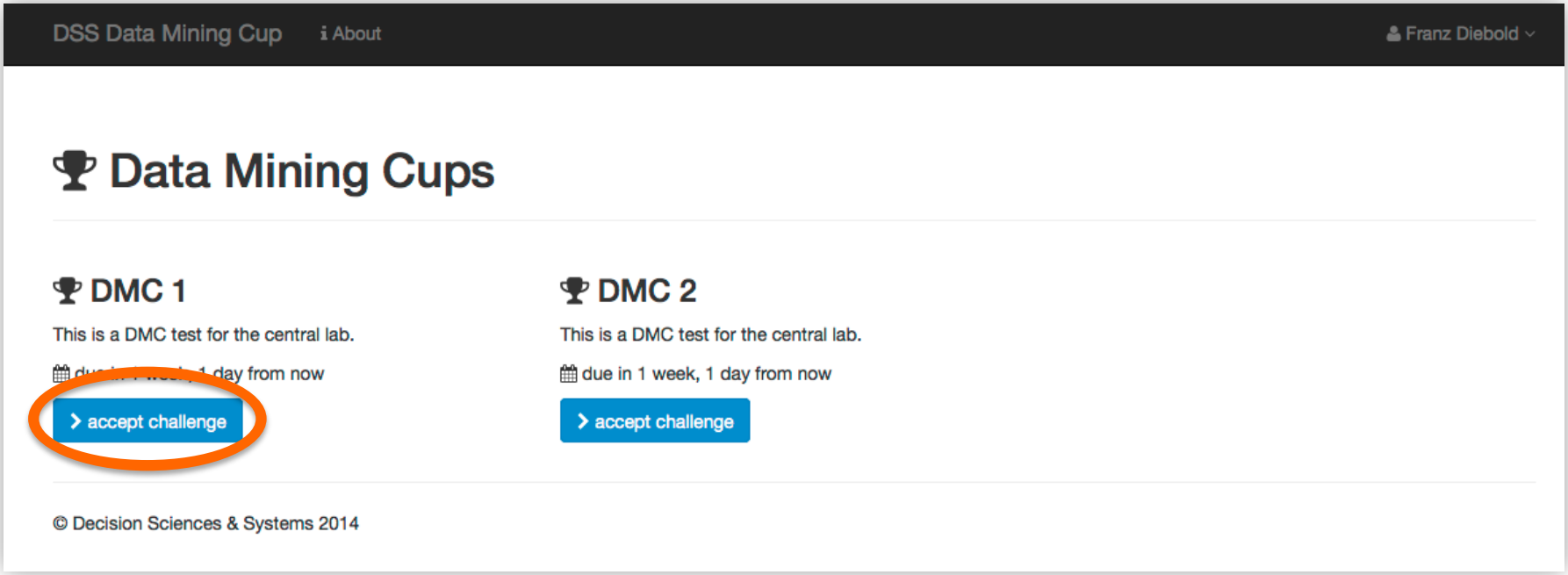
1. Build Team in DMC Manager

Login via “Shibboleth” with your TUM login data (“TUM Kennung”)



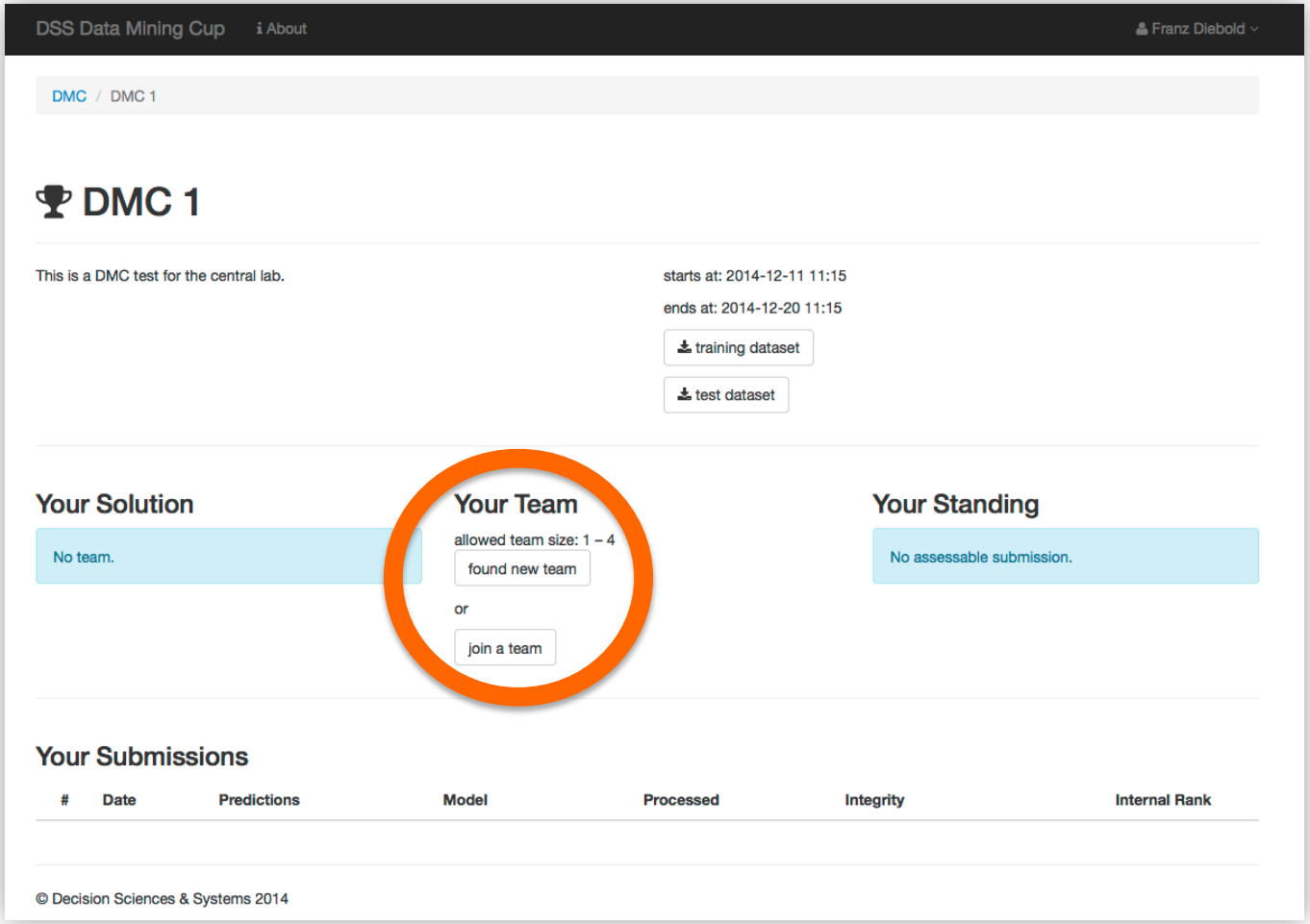
1. Build Team in DMC Manager

Choose the DMC instance in the DMC Manager



1. Build Team in DMC Manager

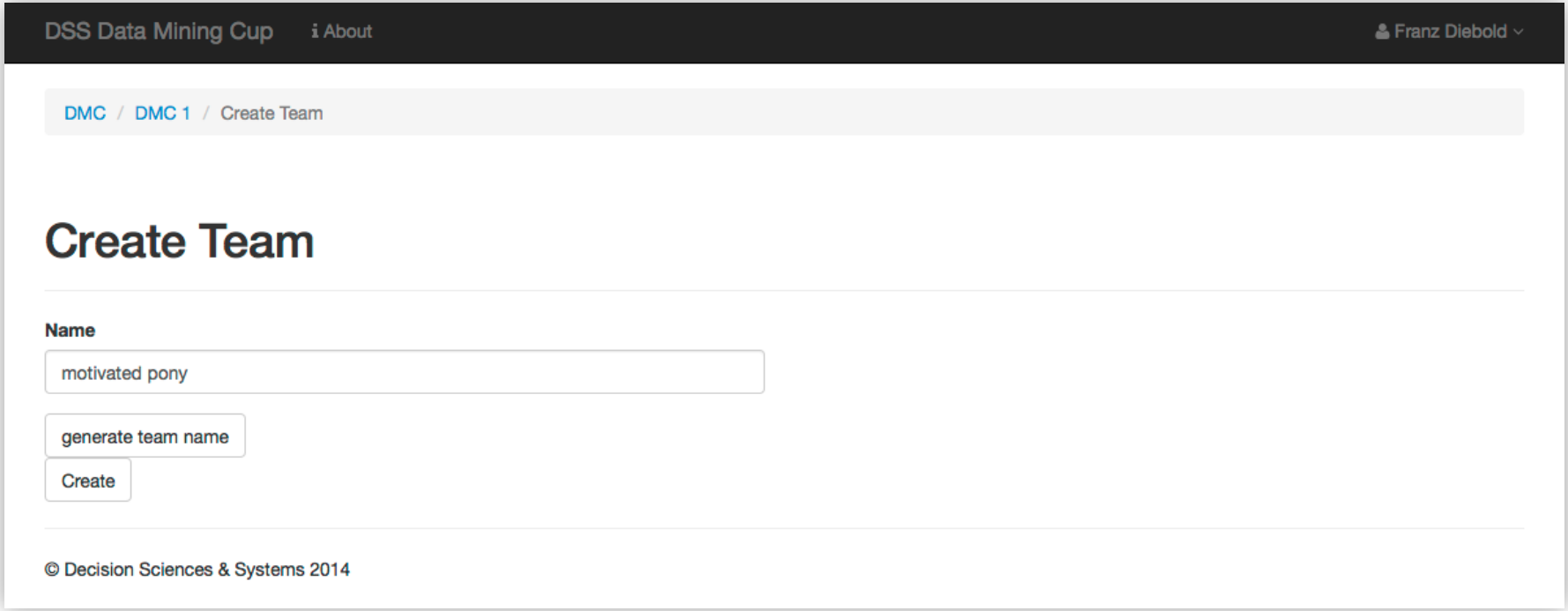
Found new team or join an existing team



1. Build Team in DMC Manager

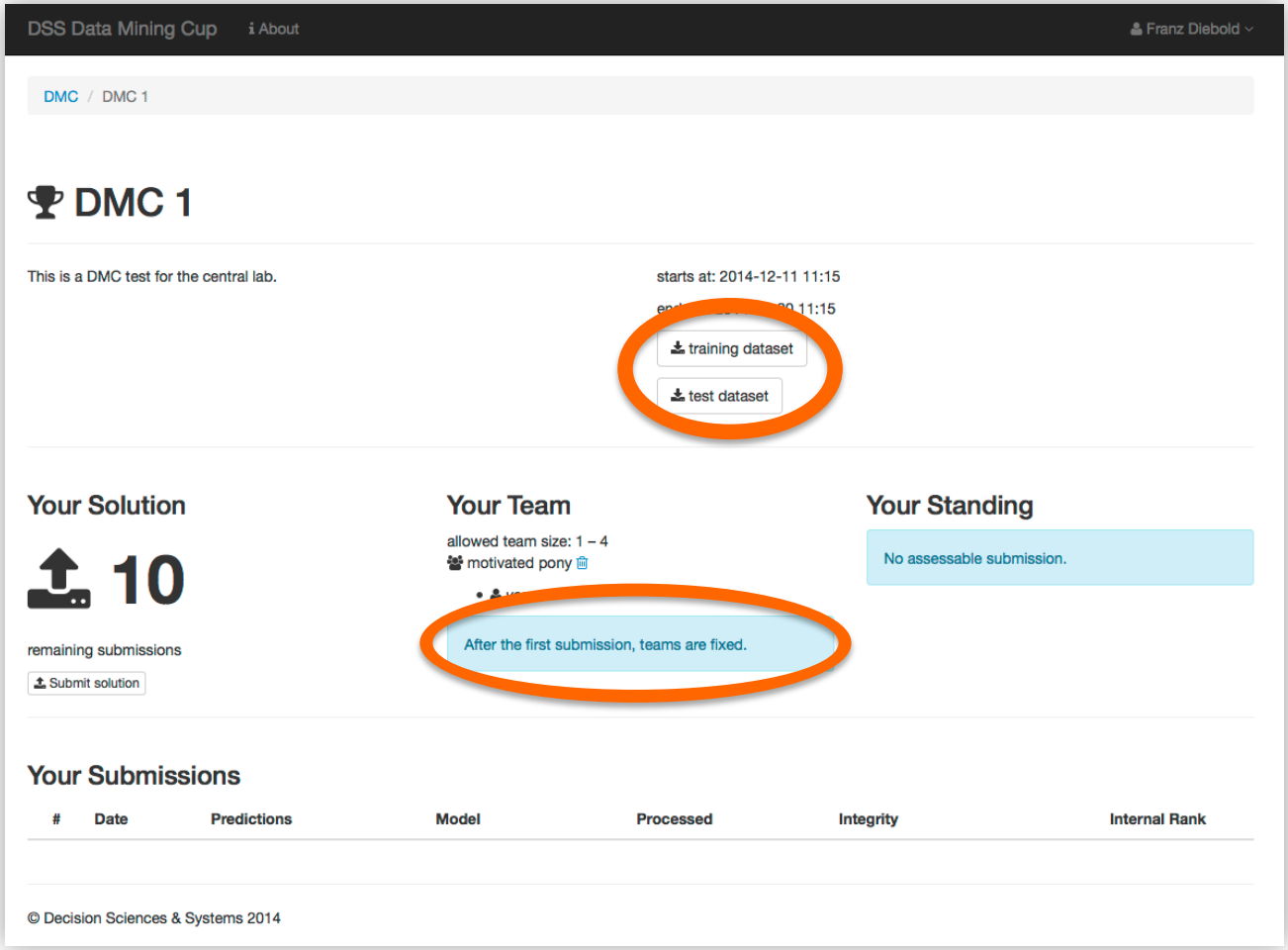
Creating a new team

- Team size: 1-4 members



2. Load & Explore the Data Set

Download the training and test datasets from the DMC Manager

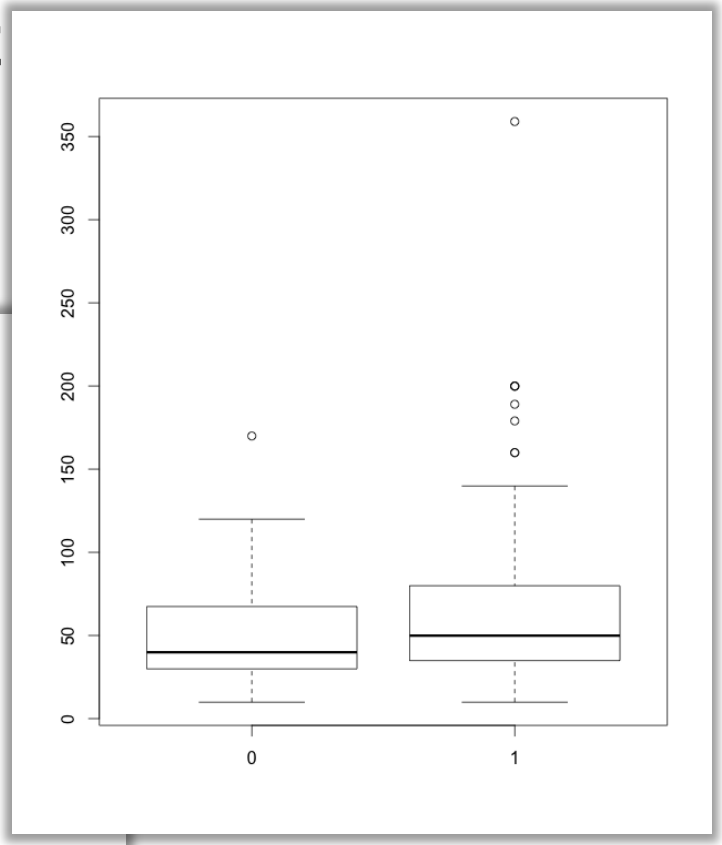


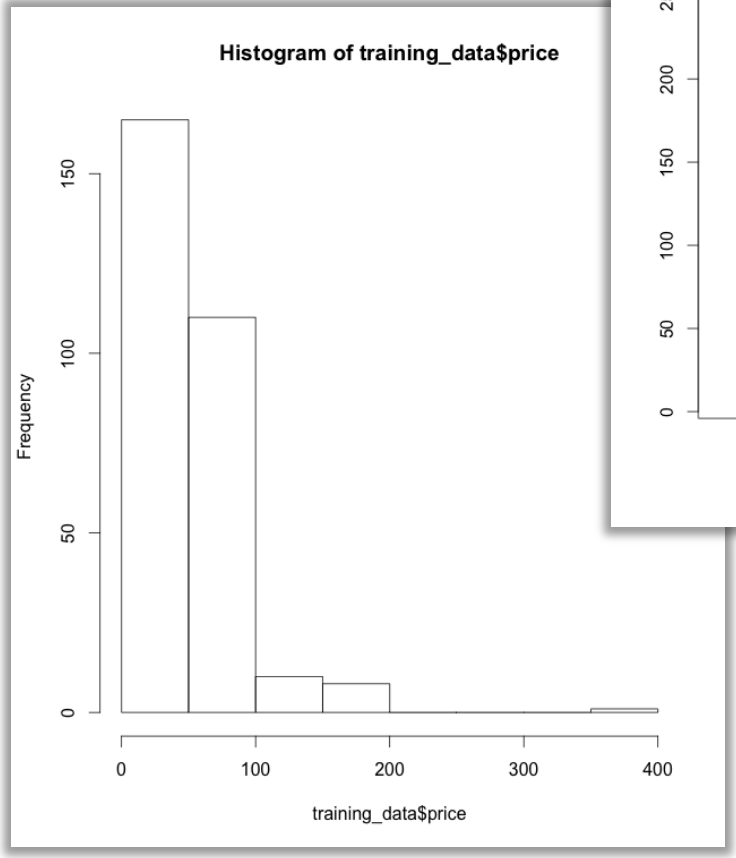
2. Load & Explore the Data Set

Load & Explore in R (compare Tutorial 1)

- Load data sets into R

- Explore the Data Set
  - Get an overview
  - Statistics
  - Plotting





3. Data Preparation

(compare Tutorial 7)

- Possible Data Preparation steps
  - Nominal attributes
  - Ordinal attributes
  - Unified date format
  - Missing values
  - Fix errors and outliers
  - Zero variance and correlation
  - Discretization/Binning
  - Feature Selection

- ALL changes in both training & test dataset!
- Do NOT DELETE any instances in the test data!

4. Training & Evaluation

Classification Methods

Name	***method* Argument in *train* Function**	**Tuning Parameters**
OneRule	OneR	-
Naïve Bayes	nb	fL, usekernel
Logistic Regression	logreg	treesize, ntrees
Decision Trees	J48	C (pruning factor), M
k-Nearest Neighbors	kknn	kmax, distance, kernel
Ensemble Methods	ada, LogitBoost, logicBag	iter; maxdepth; nu, nIter, nleaves, ntrees

```
> model = train(Class~., data=training, method="J48")
```

More classifiers: http://topepo.github.io/caret/modelList.html

Source: http://topepo.github.io/caret/

4. Training & Evaluation

Classification Methods – Tuning Parameters

- `tuneLength`: number of tuning parameter values
- `tuneGrid`: for specific tuning parameter values
  - data frame, where each row is a tuning parameter setting and each column is a tuning parameter

```
> model = train(Class~., data=training, method="J48",
                tuneGrid=data.frame(C=c(0.1, 0.2, 0.3),M=c(2,2,2))
```

Where to find parameters?

http://topepo.github.io/caret/train-models-by-tag.html

Or in R:

```
> getModelInfo()$J48$parameters
```

4. Training & Evaluation

Metrics

Name	*metric* in *train* Function	Description
Accuracy	Accuracy	*=(tp + tn) / (tp + fp + tn + fn)*
Kappa	Kappa	see below
ROC Curve	ROC	area under the ROC curve

```
> model = train(Class~., data=training, method="J48",
        metric="Kappa")
```

Kappa

- Ratio, which compares a classification method with a random classifier
  - < 0: worse than random classifier
  - > 0: better than random classifier

Source: http://topepo.github.io/caret/

4. Training & Evaluation

Resampling Methods

Name	***method* Argument in *trainControl* Function**
Bootstrapping (Holdout method, default)	boot
Repeated K-fold Cross Validation	repeatedcv
Leave-one-out	LOOCV

```
> # 2 x repeated 3-fold cross validation
> fitCtrl = trainControl(method="repeatedcv", number=3, repeats=2)

> model = train(Class~., data=training, method="J48",
                trControl=fitCtrl)
```

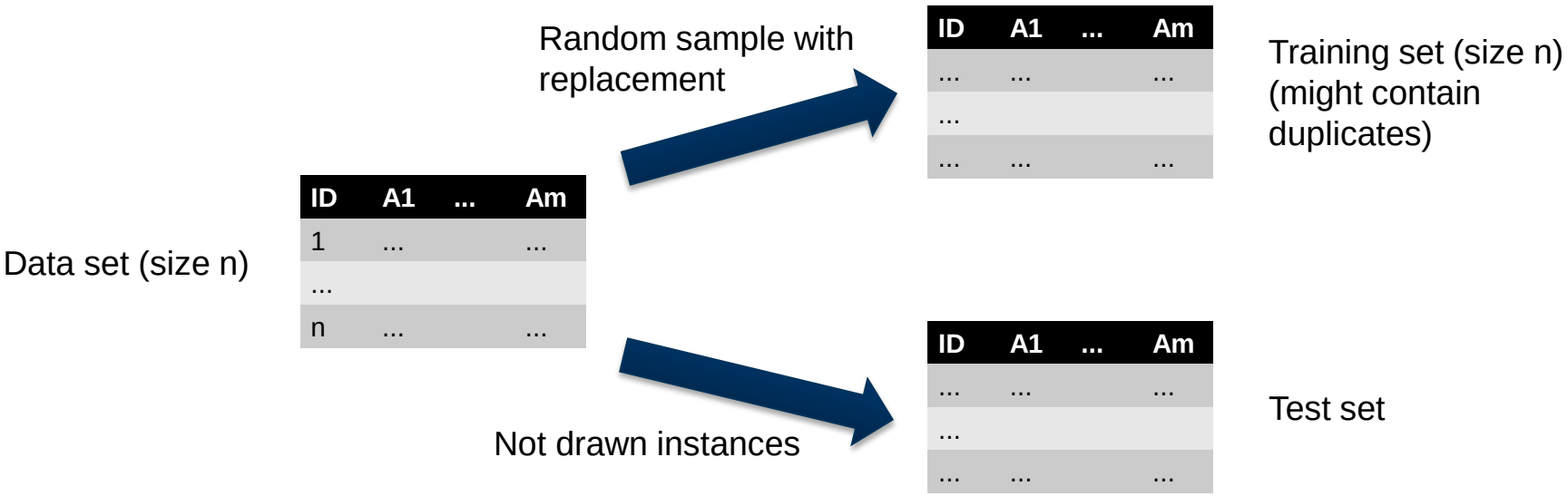
Source: http://topepo.github.io/caret/

4. Training & Evaluation

Bootstrapping

Bootstrapping

- Resampling method



4. Training & Evaluation

Balanced Samples using the “ROSE” package

- „ROSE“ package: http://cran.r-project.org/web/packages/ROSE/index.html
- Balanced samples by over-/under-sampling the minority/majority instances

```
> library(ROSE)
> training_data = ovun.sample(class ~ ., data=training_data,
            method="over", N=10000, na.action="na.pass")$data
```

method	**Description**
over	over-sampling of minority instances
under	under-sampling of majority instances
both	combination of over- and under-sampling

4. Training & Evaluation

Comparing the models

- Can compare several trained models
- The models should be using the same resampling

```
> res = resamples(list(dt = model_dt, nb = model_nb))
> summary(res)

…
Accuracy
     Min. 1st Qu. Median   Mean 3rd Qu.   Max. NA's
dt 0.4457  0.4810 0.4946 0.4910  0.5041 0.5275    0
nb 0.5000  0.5163 0.5246 0.5192  0.5275 0.5275    0
```

5. Predict Classes in Test Data

- Use the trained model to predict the classes in the test dataset.

```
> prediction_classes = predict.train(object=model,
      newdata=test_data, na.action=na.pass)
> predictions = data.frame(id=test_data$id,
      prediction=prediction_classes)
```

6. Export the Predictions

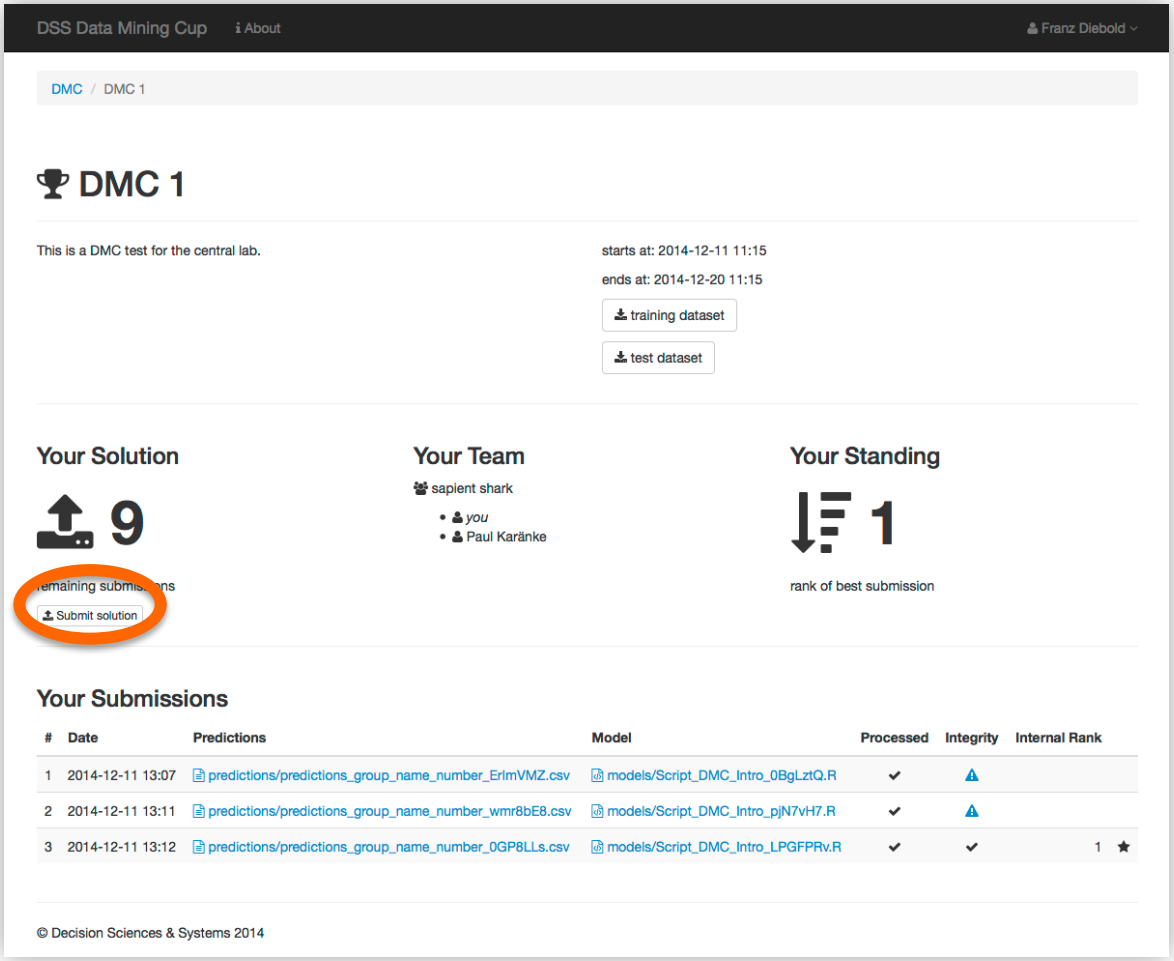
- Export predictions into csv-file
  - Format: id, prediction
  - Must contain all instances of the original test dataset

```
> write.csv(predictions, file="predictions_group_name_number.csv",
        row.names=FALSE)
```

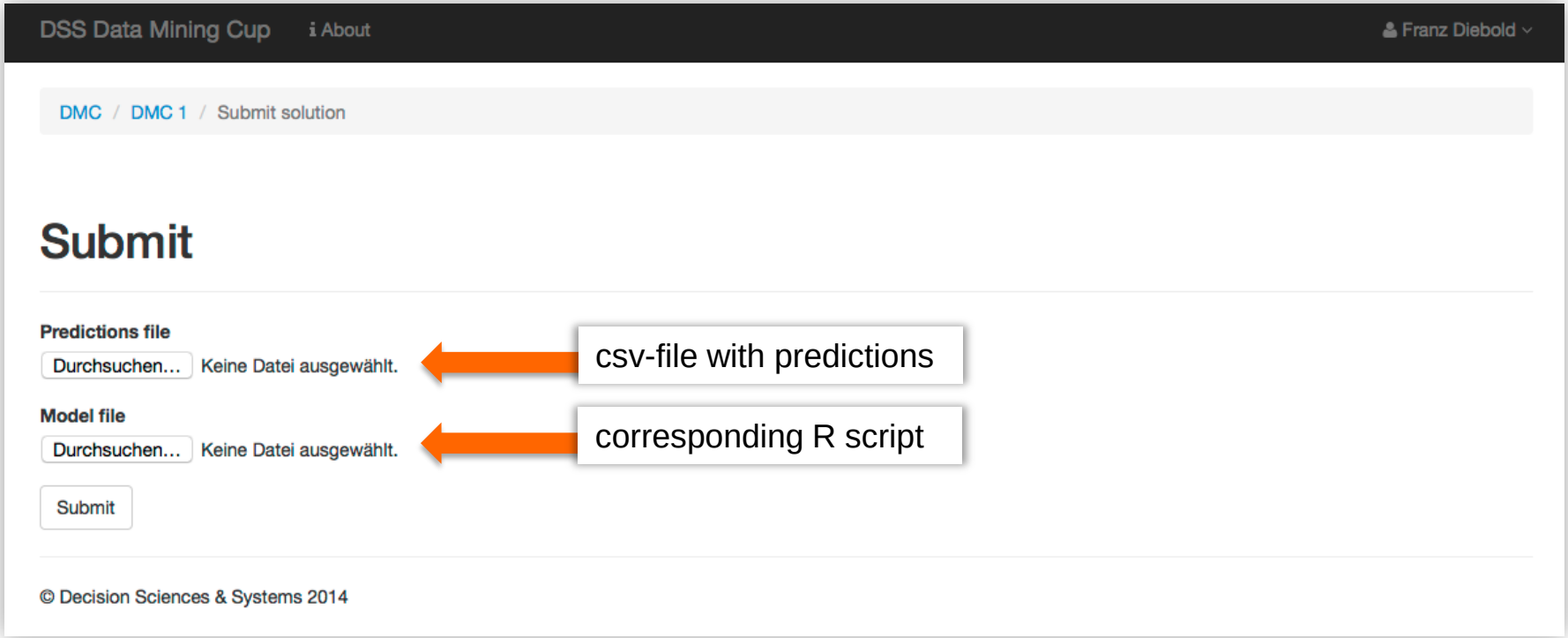
predictions_group_name_number.csv

```
"id","prediction"
130200,"1"
394720,"0"
87847,"1"
228637,"1"
189299,"0”
262991,"1”
…
```

7. Upload the Predictions and the Corresponding R Script on DMC Manager



7. Upload the Predictions and the Corresponding R Script on DMC Manager



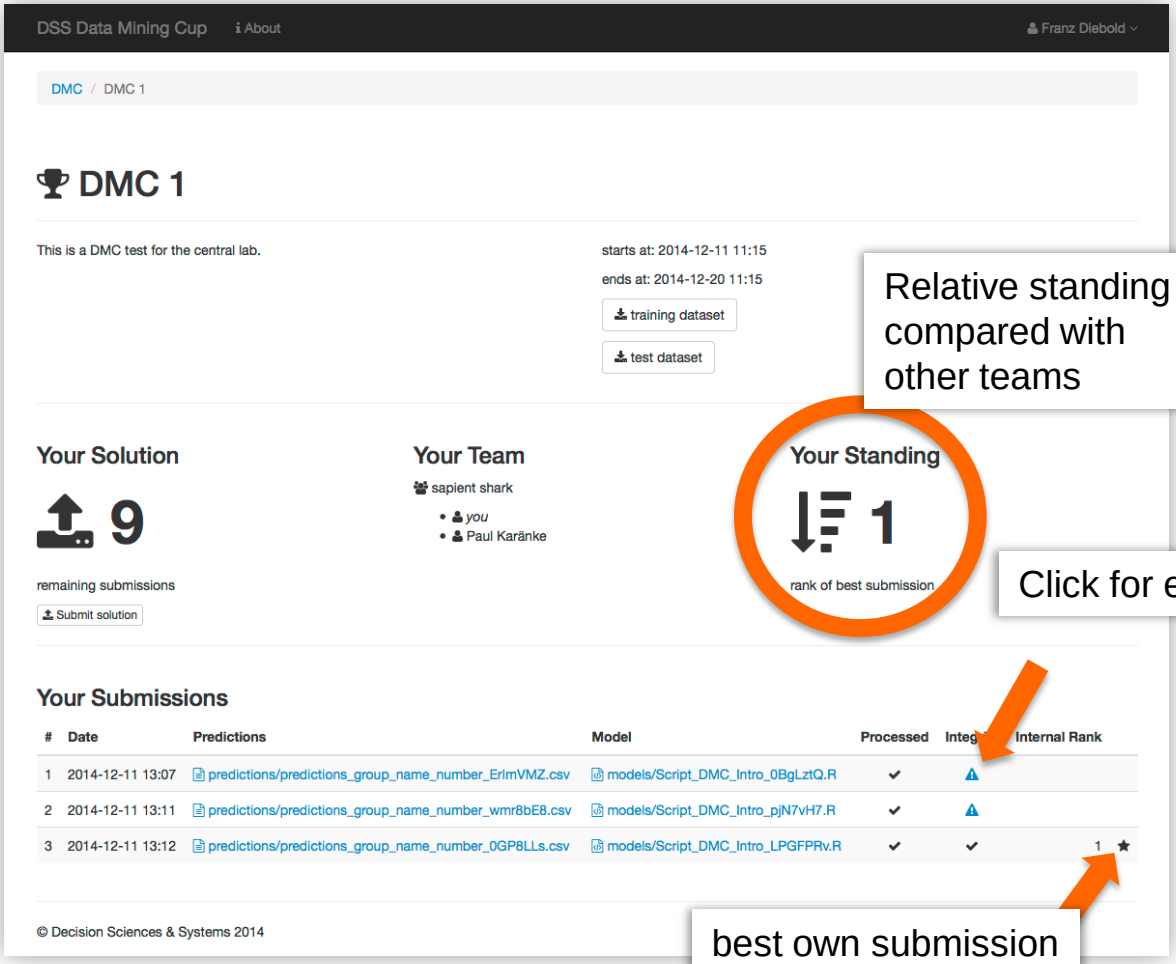
7. Upload the Predictions and the Corresponding R Script on DMC Manager

Submissions & Possible Errors

- Maximum number of submission: 10 (valid submissions)
  - Best submission counts

- Possible errors
  - Wrong column names
  - Unknown IDs (if not in Test Data)
  - Missing IDs (if in Test Data but not in Predictions)
  - Wrong file format
  - …

7. Upload the Predictions and the Corresponding R Script on DMC Manager



Comparing Classifiers

- Classifiers are hard to compare [1]
  - Different datasets
  - Limited collection of publically available datasets
  - Different data preparation
  - Tuning
  - Statistically significant claims
  - Etc.

- No best classifier
  - under certain assumptions, no classifier is better than another one [2]

[1] Salzberg S., On Comparing Classifiers: Pitfalls to Avoid and a Recommended Approach
[2] Wolpert D., On the Connection between In-sample Testing and Generalization Error

Comparing Classifiers

Many studies make mistakes when comparing Classifiers [3]

- Not using statistical tests at all
- Apply unsuitable tests or ignore assumptions
- [3] addresses these problem for…

Comparison of Two Classifiers:

- T-test: checks whether average difference in performance is significant from 0
  - Often inappropriate due to calculating using the averages
  - E.g.: Outliers can have unwanted strong effect on data and increases the variance which decreases the test power
  - Assumes the difference between random variables to be normal distributed (N<30; both often not given)
- Wilcoxon Signed-Ranks Test: non-parametric, ranks the differences in performance and compares them
  - Does not assume normal distribution and is less affected by outliers

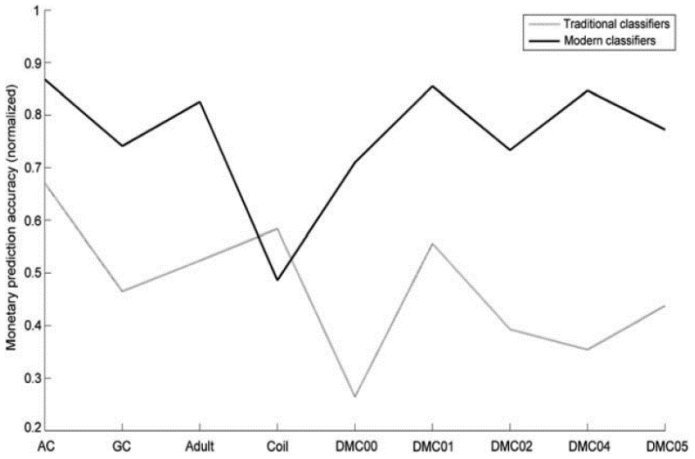
Comparison of Multiple Classifiers

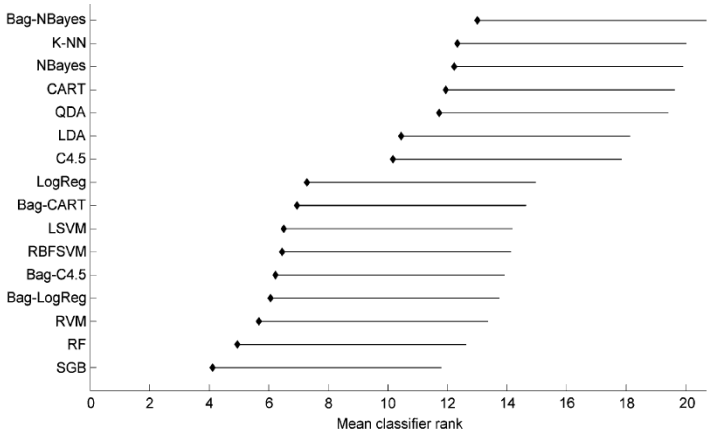
[3] Demsar J., Statistical Comparisons of Classifiers over Multiple Data Sets

Comparing Classifiers

However, there is a number of studies, which can provide useful guidelines on classifier selection

- Modern vs Traditional Classifiers [4]



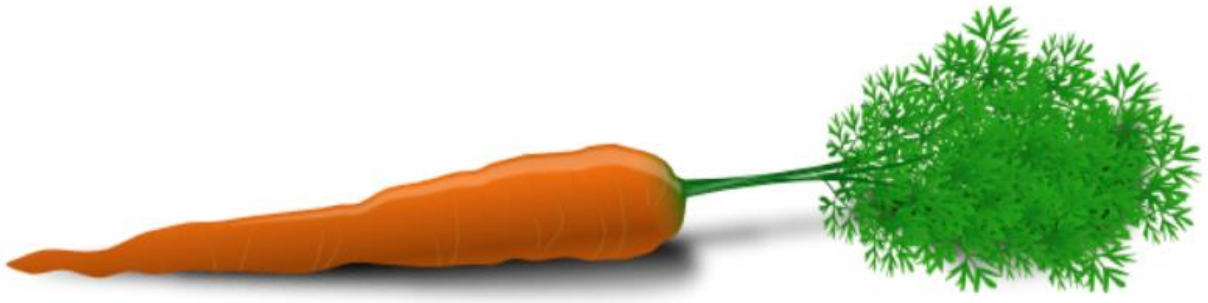


[4] Lessmann S., Voß S., A Benchmarking Study of Novel Versus Established Classification Models

Questions?

Information about the „caret package“

http://topepo.github.io/caret/



Example dataset raw_data_large

Data

- History of purchase of an online shop
- Both information about good and customer

Task

- Predict if there would be a return

Column name	Description	Range of values	Missing values
ID	Order id	Natural number	No
od	Order date	Date	No
dd	Delivery date	Date	Yes
size	Item size	String	No
price	Price of item	Positive real number	No
tax	Tax	Positive real number	No
a6	Salutation	String	No
a7	Date of birth	Date	Yes
a8	State	String	No
a9	Return shipment	{0,1}	No