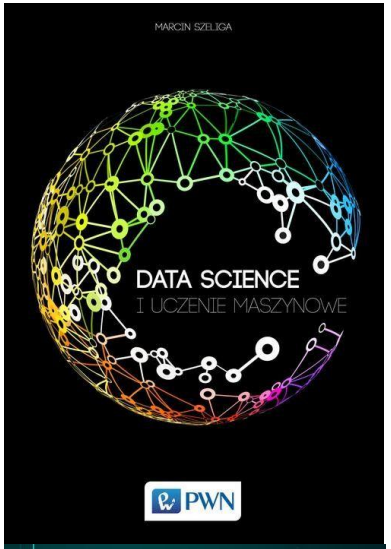
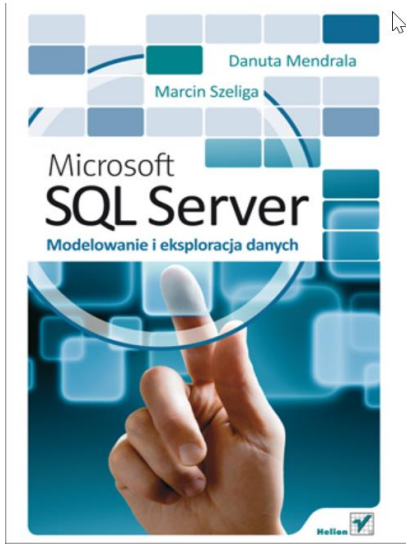
How to win Kaggle competition and get familiar with machine learning?

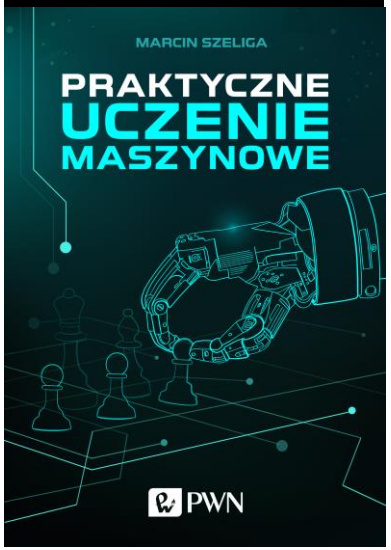
Marcin Szeliga

About me

- Data Philosopher
- Over 20 years of professional experience
- Artificial Intelligence MVP & MCT
- Microsoft Certified Solutions Expert
  - Data Management and Analytics
  - Cloud Platform and Infrastructure
  - Business Intelligence
- Microsoft Certified Solutions Developer
  - Azure Solution Architect
- marcin.szeliga@datacommunity.pl



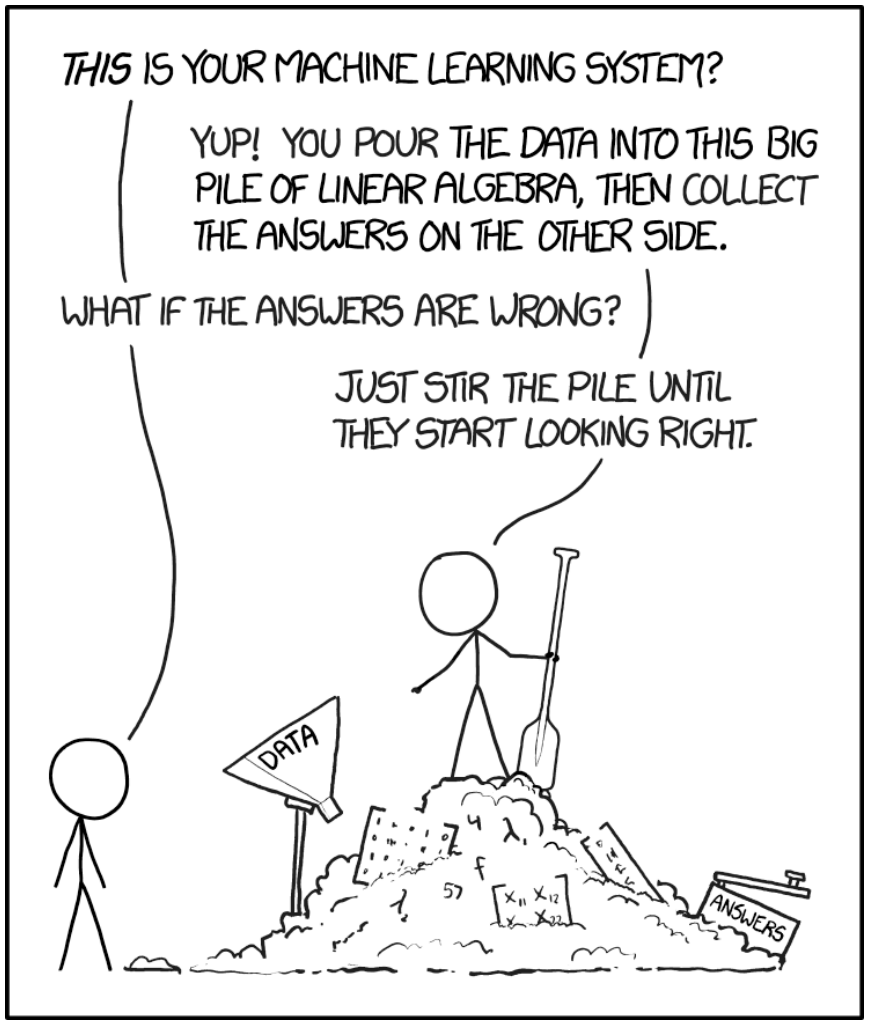




ML data pile

or

Try everything until it works



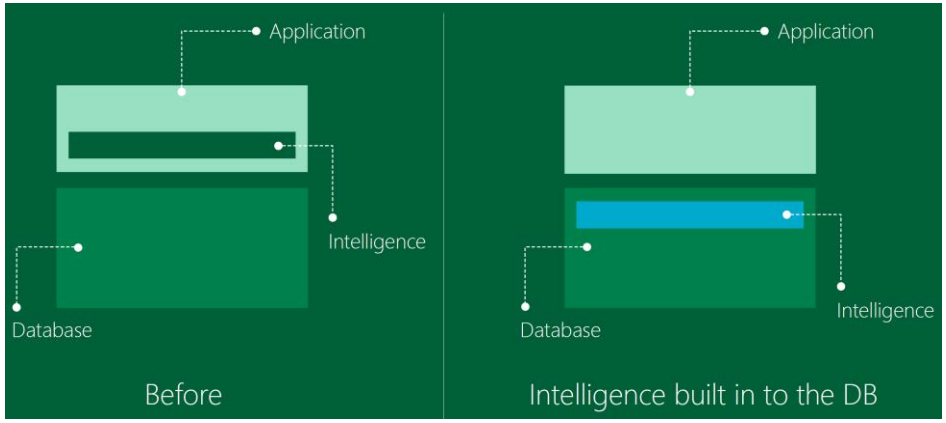
Source: https://xkcd.com/1838/

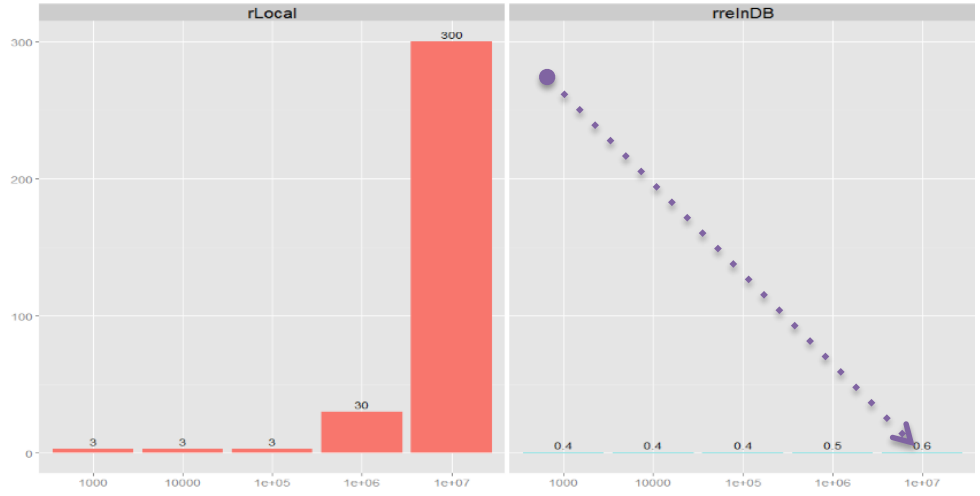
Good tools are prerequisite to success

- Performance
- Scalability
- Security
- Reliability
- Availability
- Ease of use
- …

Discover insights and make better decisions with Machine Learning Server

Transform your business with scalable, enterprise-grade R and Python based data analytics using your data and existing investments.





Load and compress data

```r
# Connection string and compute context
connection_string <- "Driver=SQL Server; Server=MS; Database=Titanic; Trusted_Connection=yes"
local <- RxLocalParallel()
rxSetComputeContext(local)

# Load train data into SQL table
train_file_name <- 'Train.csv'
train_file <- RxTextData(train_file_name)
object.size(train_file)

train_table_name <- "Train"
train_data_table <- RxSqlServerData(table = train_table_name,
                                    connectionString = connection_string)
rxDataStep(inData = train_file,
           outFile = train_data_table,
           overwrite = TRUE)
```

8592 bytes

```
4  CREATE CLUSTERED COLUMNSTORE INDEX CCI_Train
5  ON Train;
```

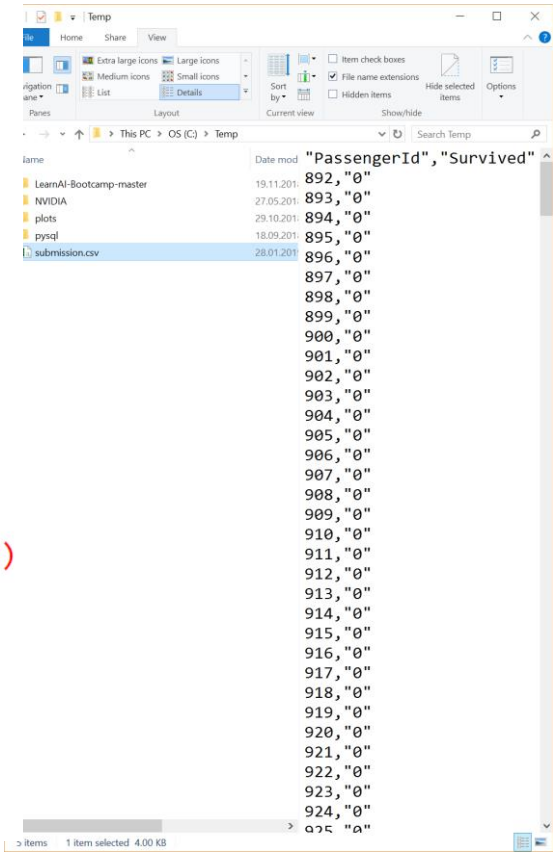
Built a great model and save it in a database

```
DECLARE @model VARBINARY(MAX);

EXEC sp_execute_external_script
    @language = N'R',
    @script = N'
    formula <- "Survived ~ Pclass + Name + Sex + Age + SibSp + Parch + Ticket + Fare + Cabin + Embarked"
    model <- RevoScaleR::rxDForest(formula,
                                   data = train)

    trained_model <- rxSerializeModel(model, realtimeScoringOnly = FALSE)
    ',
    @input_data_1 = N'SELECT * FROM Train',
    @input_data_1_name = N'train',
    @params = N'@trained_model varbinary(max) OUTPUT',
    @trained_model = @model OUTPUT;

INSERT INTO [Models] ([name], model)
VALUES('rxDForest V1', @model);
GO
```

```
> model_metrics
 Accuracy Precision    Recall   F-Score
0.9910213 0.9745223 1.0000000 0.9870968
```

It's time to submit results and win …

```
40  DECLARE @lmodel varbinary(max) =
41      (SELECT model
42      FROM [Models]
43      WHERE [name] =  'rxDForest V1');
44
45  EXEC sp_execute_external_script
46  @language = N'R',
47  @script = N'
48      model <- rxUnserializeModel(lmodel)
49      model_prediction <- rxPredict(modelObject = model,
50                                    data = Test,
51                                    extraVarsToWrite = "PassengerId")
52      model_prediction <- model_prediction[,c(2,1)]
53      colnames(model_prediction) <- c("PassengerId","Survived")
54      model_prediction$Survived <- as.integer(model_prediction$Survived)
55      write.csv(model_prediction, "c:/temp/submission.csv",row.names = FALSE)
56      ',
57  @input_data_1 = N'SELECT * FROM [Test]',
58  @input_data_1_name = N'Test',
59  @output_data_1_name = N'model_prediction',
60  @params = N'@lmodel varbinary(max)',
61  @lmodel = @lmodel
62  WITH RESULT SETS ((PassengerId int, Survived int));
```

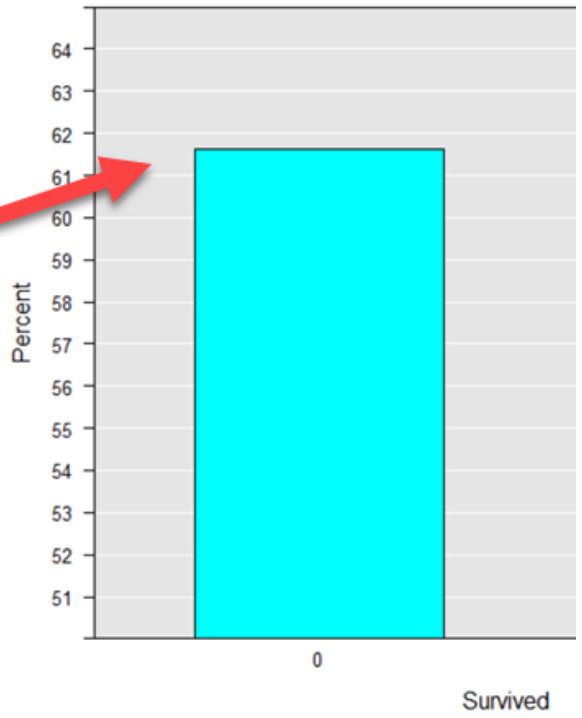


What went wrong ?

8264 ▾401 **Marcin Szeliga**

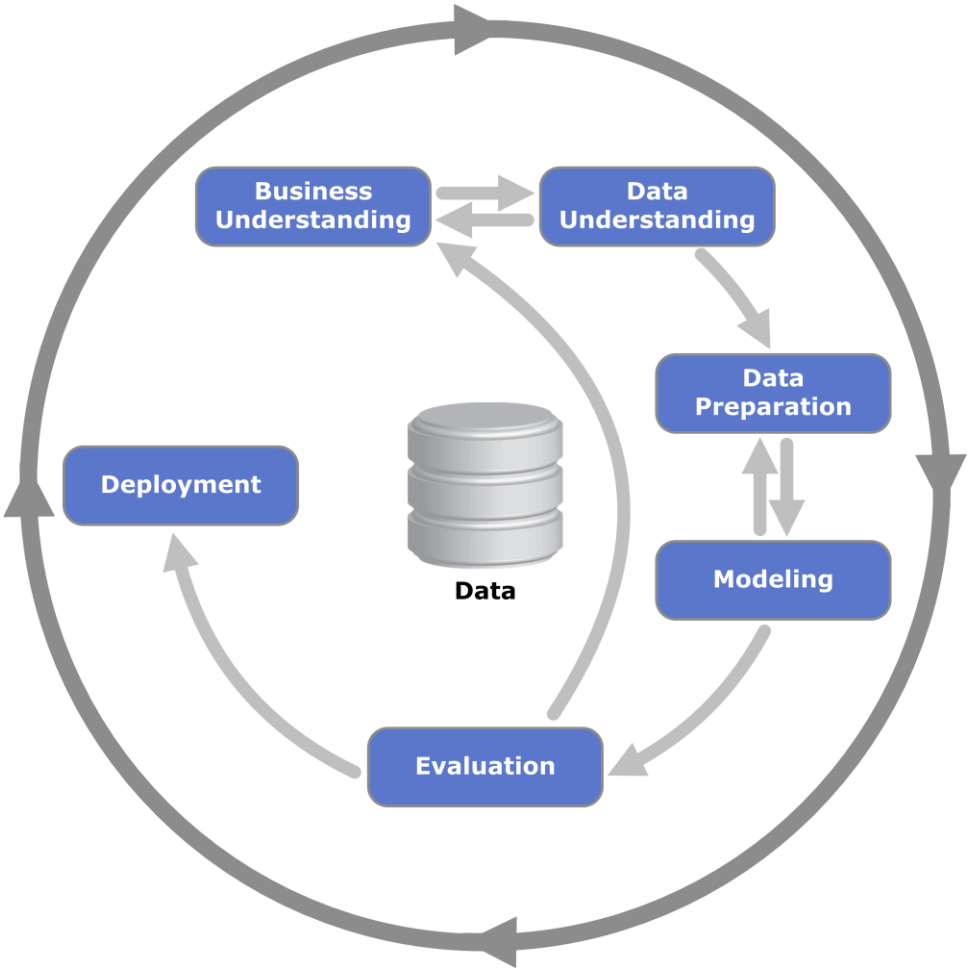
Your Best Entry ↑

Your submission scored 0.62679, which is not an improvement of your best score. Keep trying!



We've reached random guessing accuracy

ML methodology or Know theory of what's happening



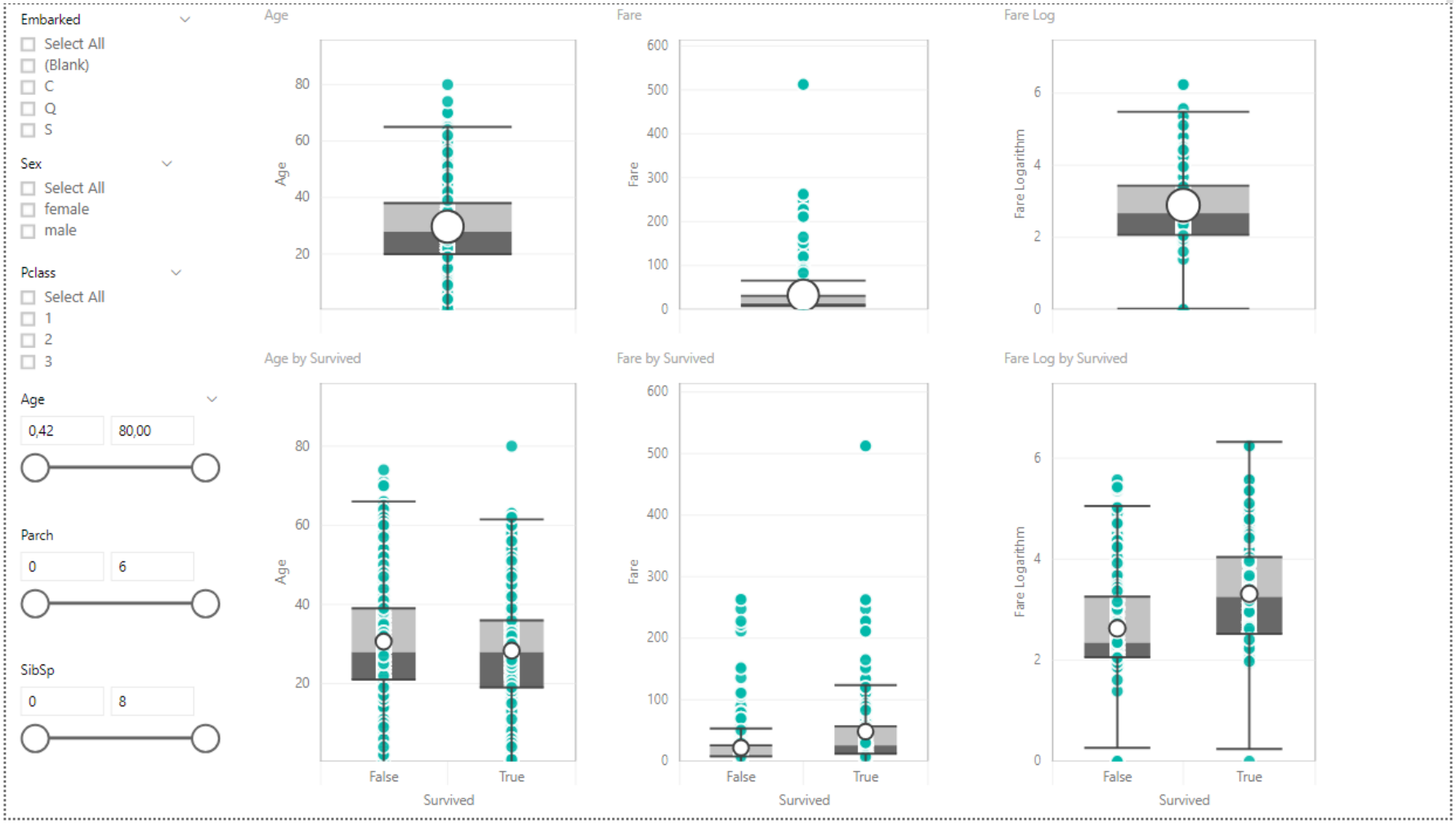
By Kenneth Jensen - own work based on: ftp://public.dhe.ibm.com/software/analytics/spss/documentation/modeler/18.0/en/ModelerCRISPDM.pdf (Figure 1), CC BY-SA 3.0, https://commons.wikimedia.org/w/index.php?curid=24930610

Problem understanding

- On April 15, 1912, during her maiden voyage, the Titanic sank after colliding with an iceberg, killing 1502 out of 2224 passengers and crew
- One of the reasons that the shipwreck led to such loss of life was that there were not enough lifeboats for the passengers and crew
- Although there was some element of luck involved in surviving the sinking, some groups of people were more likely to survive than others

By Willy Stöwer, died on 31st May 1931 - Magazine Die Gartenlaube, en:Die Gartenlaube and de:Die Gartenlaube, Public Domain, https://commons.wikimedia.org/w/index.php?curid=97646

Data understanding



Feature engineering

```
EXEC sp_execute_external_script
@language = N'R',
@script = N'
    library(tidyverse)
    library(stringr)
    library(mice)

    train$Title <- gsub(".*, ([^.]*)\\..*", "\\1", train$Name)
    train$Title <- str_replace_all(train$Title,
                              c("Mlle" = "1", "Capt" = 5, "Col" = 5,
                                "Dr" = 5, "Major" = 5, "Rev"= 5,
                                "Don"= 6, "Jonkheer"= 6, "Lady"= 6, "Sir"= 6,
                                "the Countess"= 6,"Mme"= 2, "Ms"= 2,
                                "Miss" = 1, "Mrs" = 2, "Master" = 3, "Mr" = 4, "Professional" = 5, "6a" = 6))
    train$Title <- as.integer(train$Title)
    train$FamilySize <- train$SibSp + train$Parch + 1
    train$IsAlone <- ifelse(train$FamilySize==1,1,0)
    train$HasCabin <- ifelse(is.na(train$Cabin),0,1)
    train$Fare [train$Pclass == 1 & train$Fare == 0] <- mean(train$Fare[train$Fare != 0 & train$Pclass == 1])
    train$Fare [train$Pclass == 2 & train$Fare == 0] <- mean(train$Fare[train$Fare != 0 & train$Pclass == 2])
    train$Fare [train$Pclass == 3 & train$Fare == 0] <- mean(train$Fare[train$Fare != 0 & train$Pclass == 3])
    train$Fare <- log(train$Fare+0.01)
    train$Sex <- ifelse(train$Sex=="male",0,1)
    train$Embarked[is.na(train$Embarked)] <- "C"
    train$Embarked <- ifelse(train$Embarked=="S",0,
                          ifelse(train$Embarked=="C",1,2))
    temp_data <- mice(train,m=1,maxit=50,meth="pmm",seed=500)
    train <- complete(temp_data,1)
    train$Survived <- factor(train$Survived, levels = c("0","1"))
    train <- subset(train, select = -c(Name, Ticket, Cabin))

    features_table_name <- "Train_Features"
    features_data_table <- RxSqlServerData(table = features_table_name,
                                    connectionString = connection_string)
    rxDataStep(inData = train,
           outFile = features_data_table,
           overwrite = TRUE)
    ',
@input_data_1 = N'SELECT * FROM [Train]',
@input_data_1_name = N'train';
```

Feature engineering

```
> summary(train)
  PassengerId    Survived     Pclass          Sex
 Min.   :  1.0   0:549    Min.   :1.000   Min.   :0.0000
 1st Qu.:223.5   1:342    1st Qu.:2.000   1st Qu.:0.0000
 Median :446.0            Median :3.000   Median :0.0000
 Mean   :446.0            Mean   :2.309   Mean   :0.3524
 3rd Qu.:668.5            3rd Qu.:3.000   3rd Qu.:1.0000
 Max.   :891.0            Max.   :3.000   Max.   :1.0000
      Age            SibSp           Parch
 Min.   : 0.42   Min.   :0.000   Min.   :0.0000
 1st Qu.:20.00   1st Qu.:0.000   1st Qu.:0.0000
 Median :28.00   Median :0.000   Median :0.0000
 Mean   :29.48   Mean   :0.523   Mean   :0.3816
 3rd Qu.:38.00   3rd Qu.:1.000   3rd Qu.:0.0000
 Max.   :80.00   Max.   :8.000   Max.   :6.0000
      Fare          Embarked         Title
 Min.   :1.392   Min.   :0.0000   Min.   :1.000
 1st Qu.:2.071   1st Qu.:0.0000   1st Qu.:2.000
 Median :2.675   Median :0.0000   Median :4.000
 Mean   :2.952   Mean   :0.3636   Mean   :3.082
 3rd Qu.:3.443   3rd Qu.:1.0000   3rd Qu.:4.000
 Max.   :6.239   Max.   :2.0000   Max.   :6.000
   FamilySize        IsAlone          HasCabin
 Min.   : 1.000   Min.   :0.0000   Min.   :0.000
 1st Qu.: 1.000   1st Qu.:0.0000   1st Qu.:0.000
 Median : 1.000   Median :1.0000   Median :0.000
 Mean   : 1.905   Mean   :0.6027   Mean   :0.229
 3rd Qu.: 2.000   3rd Qu.:1.0000   3rd Qu.:0.000
 Max.   :11.000   Max.   :1.0000   Max.   :1.000
```

Feature reduction

- Law of parsimony - **the simplest solution tends to be the right one**
- Feature importance can be assumed (based on correlation) or computed (by training a model)
- Less important features just increase noise and should be removed

```
res.man <- manova(cbind(Pclass, Sex, Age, SibSp, Parch, Fare, Embarked,
                        Title, FamilySize, IsAlone, HasCabin) ~ Survived,
                  data = train)
summary.aov(res.man)
Response Pclass :
            Df Sum Sq Mean Sq F value    Pr(>F)
Survived     1  71.28  71.276  115.03 < 2.2e-16 ***

Response Sex :
            Df  Sum Sq Mean Sq F value    Pr(>F)
Survived     1  60.033  60.033  372.41 < 2.2e-16 ***

Response Age :
            Df Sum Sq Mean Sq F value    Pr(>F)
Survived     1   1993 1993.30  9.4003 0.002235 **

Response SibSp :
            Df  Sum Sq Mean Sq F value Pr(>F)
Survived     1    1.35  1.3503  1.1106 0.2922

Response Parch :
            Df Sum Sq Mean Sq F value Pr(>F)
Survived     1   3.85  3.8531  5.9635 0.0148 *

Response FamilySize :
            Df  Sum Sq Mean Sq F value Pr(>F)
Survived     1    0.64 0.64145  0.2462 0.6199

Response IsAlone :
            Df  Sum Sq Mean Sq F value    Pr(>F)
Survived     1   8.824  8.8239  38.354 9.009e-10 ***

Response Fare :
            Df Sum Sq Mean Sq F value    Pr(>F)
Survived     1  76.59  76.588  96.995 < 2.2e-16 ***

Response Embarked :
            Df Sum Sq Mean Sq F value    Pr(>F)
Survived     1   4.45  4.4540  11.131 0.0008841 ***

Response Title :
            Df  Sum Sq Mean Sq F value    Pr(>F)
Survived     1  390.93  390.93  315.92 < 2.2e-16 ***

Response HasCabin :
            Df  Sum Sq Mean Sq F value    Pr(>F)
Survived     1  15.797 15.7974  99.253 < 2.2e-16 ***
```

Modeling with hyperparameters tuning

```
DROP TABLE IF EXISTS #hiperparameters;
WITH Sequences10 (numTrees,minSplit) AS
    (SELECT 37 AS numTrees, 1 AS minSplit
        UNION ALL
    SELECT numTrees + 3, minSplit + 1
        FROM Sequences10
        WHERE numTrees < 65),
Sequences5 (featureFraction,splitFraction) AS
    (SELECT CAST(0.5 AS DECIMAL (3,2)) AS featureFraction,
    CAST(0.5 AS DECIMAL (3,2)) AS splitFraction
        UNION ALL
    SELECT CAST(featureFraction + 0.1 AS DECIMAL (3,2)),
    CAST(splitFraction  + 0.1  AS DECIMAL (3,2))
        FROM Sequences5
        WHERE featureFraction < 1)
SELECT S1.numTrees, S2.minSplit,
    S3.featureFraction, S4.splitFraction
INTO #hiperparameters
FROM Sequences10 AS S1
CROSS JOIN Sequences10 AS S2
CROSS JOIN Sequences5 AS S3
CROSS JOIN Sequences5 AS S4;
```

```
DECLARE hiperparameters CURSOR FOR
SELECT numTrees, minSplit,
    featureFraction, splitFraction
FROM #hiperparameters;

DECLARE @i INT = 1, @name NVARCHAR (250),
    @numTrees INT, @minSplit INT
DECLARE @featureFraction DECIMAL(5,4),
    @splitFraction DECIMAL(5,4)

OPEN hiperparameters
FETCH NEXT FROM hiperparameters
INTO @numTrees, @minSplit,
    @featureFraction, @splitFraction

WHILE @@FETCH_STATUS = 0
BEGIN
    SET @name = CONCAT('rxFastTrees V',@i);
    PRINT '-------------------------------------'
    PRINT 'Training model ' + @name
    PRINT '-------------------------------------'
    EXEC [BuildModels] @name, @numTrees, @minSplit,
        @featureFraction, @splitFraction;

    FETCH NEXT FROM hiperparameters
    INTO @numTrees, @minSplit,
        @featureFraction, @splitFraction;
    SET @i+=1;
END
CLOSE hiperparameters;
DEALLOCATE hiperparameters;
```

Evaluation by cross validation

```
CREATE PROCEDURE [dbo].[BuildModels]
    (@name NVARCHAR(255), @numTrees INT, @minSplit INT,
     @featureFraction DECIMAL (5,4), @splitFraction DECIMAL (5,4))
AS BEGIN
DECLARE @model VARBINARY(MAX), @accuracy INT

EXECUTE sp_execute_external_script
 @language = N'R'
 ,@script = N'
 library(cvTools)
 formula <- as.formula("Survived ~ Title+Embarked+Fare+Pclass+Sex+Age+IsAlone+Embarked+HasCabin")
 accuracy <- as.numeric()
 kf<-cvFolds(nrow(train), K = 10, type = "random")
 for (i in 1:10) {
  model <- rxFastTrees(formula = formula,
                       data = train[kf$which!=i,],
                       type = "binary",
                       numTrees = numTrees,
                       minSplit = minSplit,
                       learningRate = 0.1,
                       featureFraction = featureFraction,
                       splitFraction = splitFraction,
                       unbalancedSets = FALSE)
  scoreDS <- rxPredict(model, data = train[kf$which==i,], extraVarsToWrite = c("Survived"))
  myForm <- as.formula("~ Survived:PredictedLabel")
  confusion <- rxCrossTabs(myForm,data = scoreDS, returnXtabs = TRUE)
  names(dimnames(confusion)) <- c("actual","predicted")
  tn <- confusion[1, 1]
  fp <- confusion[1, 2]
  fn <- confusion[2, 1]
  tp <- confusion[2, 2]
  model_accuracy <- (tp + tn) / (tp + fn + fp + tn)
  accuracy <- rbind(accuracy, model_accuracy)
 }
 model <- rxSerializeModel(model, realtimeScoringOnly = TRUE)
 accuracy <- as.integer(10000*mean(accuracy))
 '
 ,@input_data_1 = N'SELECT * FROM [dbo].[Train_Features]'
 ,@input_data_1_name = N'train'
 ,@params = N'@numTrees INT, @minSplit INT
    ,@featureFraction DECIMAL(5,4), @splitFraction DECIMAL(5,4)
    ,@model varbinary(max) OUTPUT, @accuracy INT OUTPUT'
 ,@numTrees = @numTrees
 ,@minSplit = @minSplit
 ,@featureFraction = @featureFraction
 ,@splitFraction = @splitFraction
 ,@model = @model OUTPUT
 ,@accuracy = @accuracy OUTPUT

 INSERT INTO [Models] ([Name], Model, Accuracy
    ,numTrees, minSplit, featureFraction, splitFraction)
 VALUES(@name, @model, 1.*@accuracy/100
    ,@numTrees, @minSplit, @featureFraction, @splitFraction);
END
```

Train the best model on all examples and use it for scoring

```
SELECT *
FROM [Models]
WHERE [Accuracy] = (SELECT max([Accuracy]) FROM [Models])
   OR [Accuracy] = (SELECT min([Accuracy]) FROM [Models])
ORDER BY Accuracy DESC;
```

Messages

Name	Model	Accuracy	numTrees	minSplit	featureFraction	splitFraction
rxFastTrees V63	0x626C6F62D185CF4878CDE9758C5BC8F...	85.96	58	1	0.5000	1.0000
rxFastTrees V702	0x626C6F62387B99E70DD32F2136D117BA...	85.96	61	2	0.9000	0.8000
rxFastTrees V703	0x626C6F62FE918296A497A427D0C4C3D1...	85.96	64	2	0.9000	0.8000
rxFastTrees V364	0x626C6F62F64FE9AA705AC047104CACC...	83.15	37	1	1.0000	0.8000
rxFastTrees V365	0x626C6F629A7CB1560ABBFA753FCA338...	83.15	40	1	1.0000	0.8000

525	▲7338	**Marcin Szeliga**		0.81818	6	~10s

Your Best Entry ↑

Your submission scored 0.81818, which is not an improvement of your best score. Keep trying!

198	▲7665	**Marcin Szeliga**		0.83732	7	now

Your Best Entry ↑

You advanced 327 places on the leaderboard!
Your submission scored 0.83732, which is an improvement of your previous score of 0.81818. Great job!

Thank you

- We are in top 2% of the leaderboard
- Nailing result, because some cheat
- This was just a small example, but the same methodology works well in practice
- You can find more examples at https://github.com/szelor/practical-machine-learning
- Stay in touch with me
- https://www.linkedin.com/in/marcinszeliga
- marcin.szeliga@datacommuity.pl