COLLEX - Collecting **L**abels from **Ex**perts

Final Presentation - July 28, 2017

Ahmet Gündüz
Gunnar CS Koenig
Julia M Moosbauer
Franz MJ Pfister

Outline

BD Lab

- Introduction & Background
- Declaration of Transparency
- Technology
- Sprint Overview
- Results
  - Software Engineering
  - Pre-Processing / Cleaning
  - Model Building & Statistical Evaluation
- Key Learnings & Outlook
- Appendix: Sprint Details

Introduction & Background

Parkinson’s Disease

- Second most common neurodegenerative Disease worldwide
- Typical Symptoms:
  - Bradykinesia
  - Rigor
  - **Rest Tremor**
  - Gait Disorders
  - Non-motor symptoms: Depression, urinary dysfunction, autonomous disorders

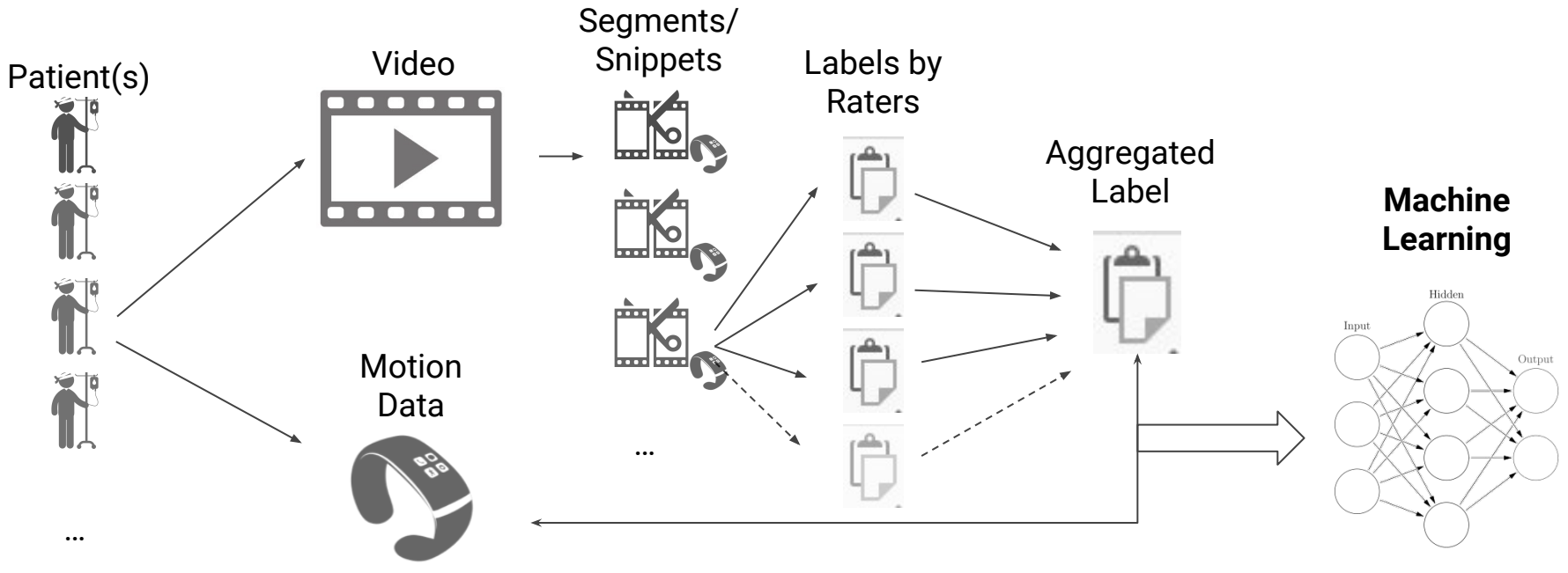
Purpose & Goal - Specific Use Case

- Application in biomedical research: Parkinson‘s Disease (PD)
- (Objective) analysis of PD Symptoms, e.g. rest tremor, is a key success factor in management of the disease
- Many scientific players to solve that problem
- Giant amounts of motion data
- Problem: No / little amount of labels[1]

→ Solution: Fitting **various machine learning models** enabled by collection of large amounts of **expert labels** and their **meta-information** (e.g. confidence)

[1] Ching et al 2017, Opportunities and obstacles for deep learning in biology and medicine, published online

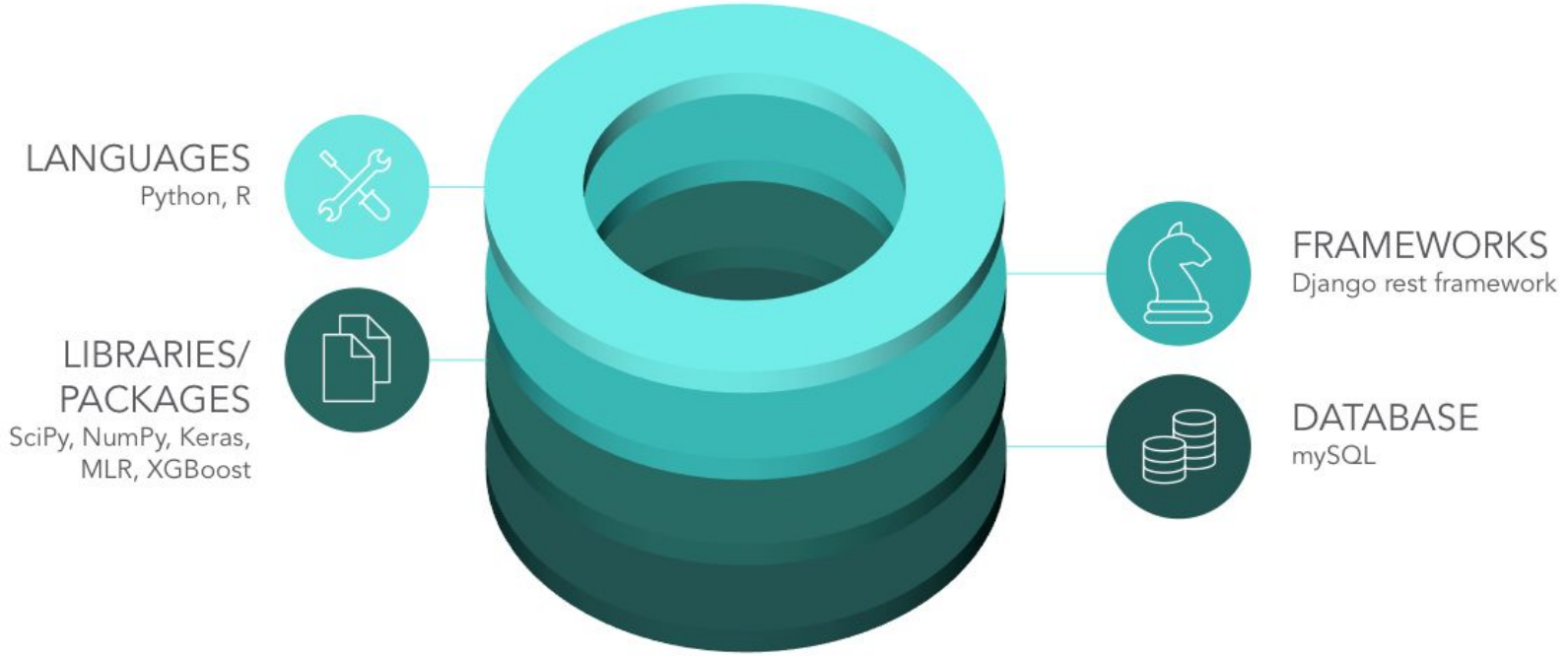
Workflow



Source: © Freepik

Technology

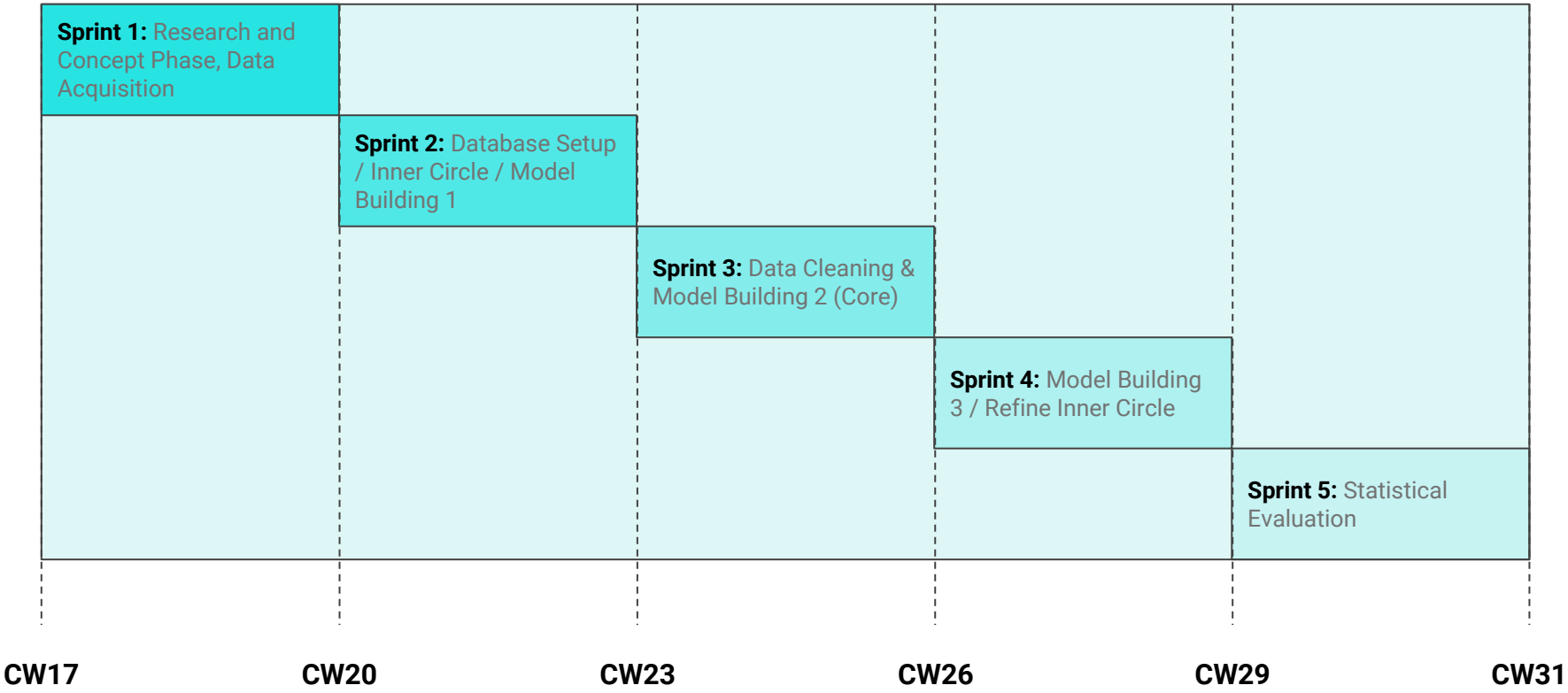
Technology used



Sprint Overview

of the Big Data Lab

Sprint Overview

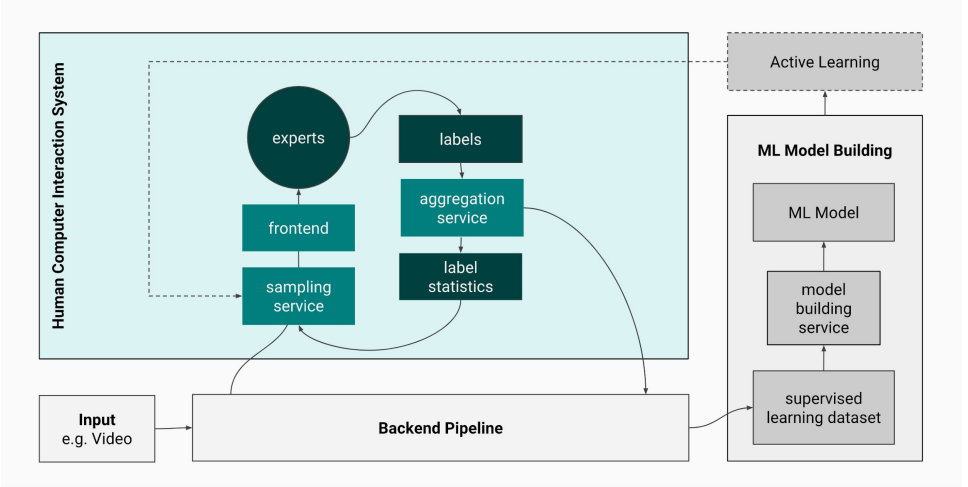


Sprint 1: Research and Concept Phase

Goals:

- Definition of Core Technologies
- Acquisition & Preparation of Clinical Research Data
- Component Structure of Project
- Assignment of team roles and responsibilities
- Discussion with supervisors and implementation of feedback

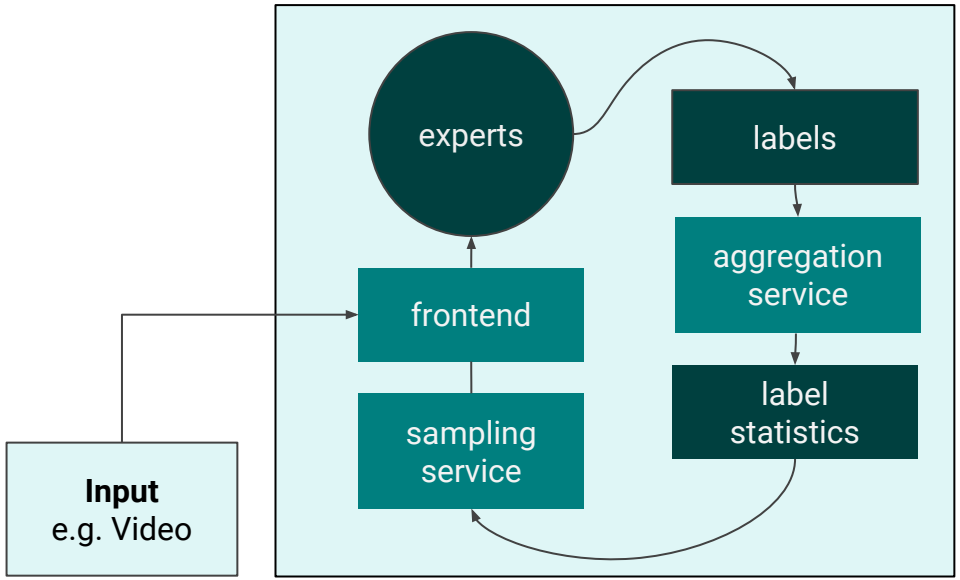
Achievements:



Sprint 2: Database Setup / Inner Circle / Model Building 1

Goals:

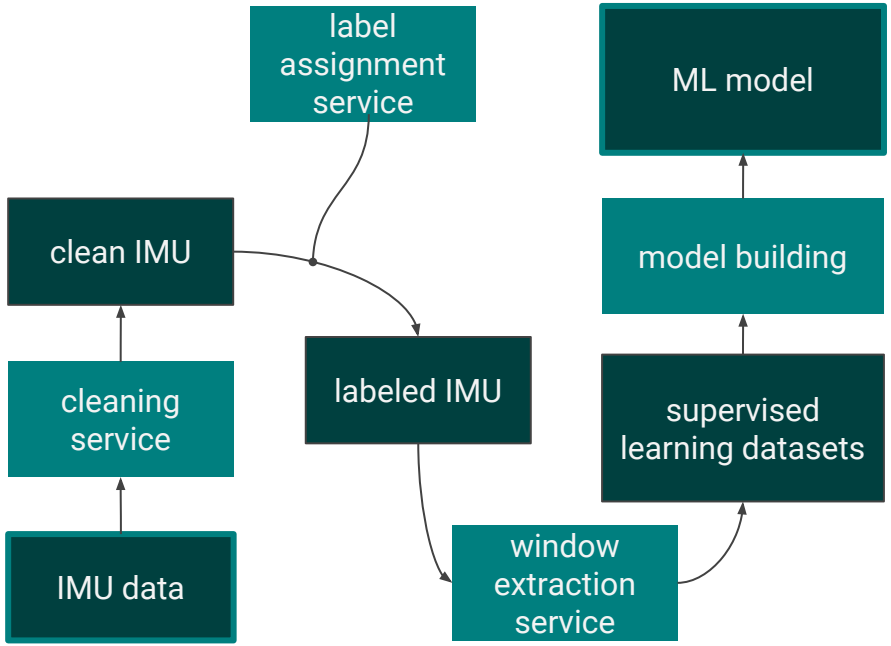
- Prototype Inner Circle of the System
  - Process Input Data
  - Setup prototype database
  - Decision on API
- Research on Machine Learning Technology
- Evaluate Backend-Pipeline
- Implementation of first API version
- **Authentication Service**



Sprint 3: Data Cleaning & Model Building 2 (Core)

Goals:

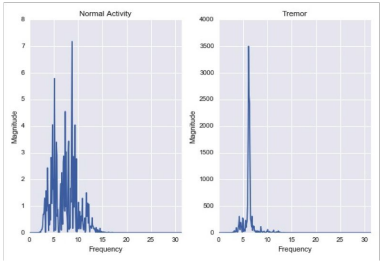
- **Connecting the components**
- **Cleaning Service:** Preprocessing of IMU data for Data Science analysis (e.g. read-in, Spectrogram, FFT, Noise reduction, DWT)
- **Machine Learning Component:** Implement Several Machine Learning Models using Test Data
  - Random Forests
  - KNN
  - Trees

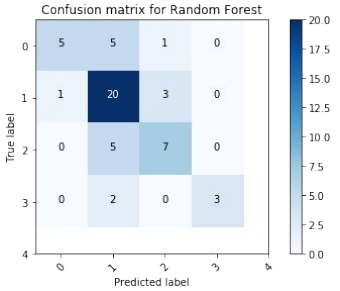


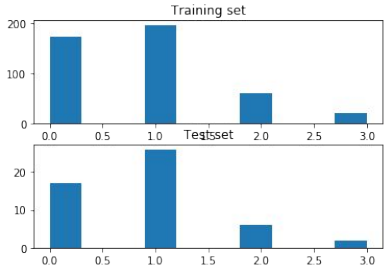
Sprint 3: Data Cleaning & Model Building 2 (Core)

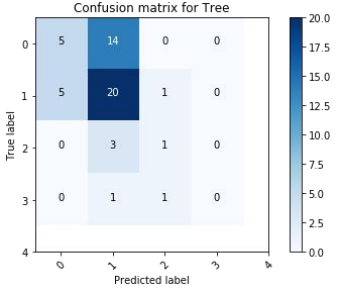
Goals:

- **Connecting the components**
- **Cleaning Service:** Preprocessing of IMU data for Data Science analysis (e.g. read-in, Spectrogram, FFT, Noise reduction, DWT)
- **Machine Learning Component:** Implement Several Machine Learning Models using Test Data
  - Random Forests
  - KNN
  - Trees





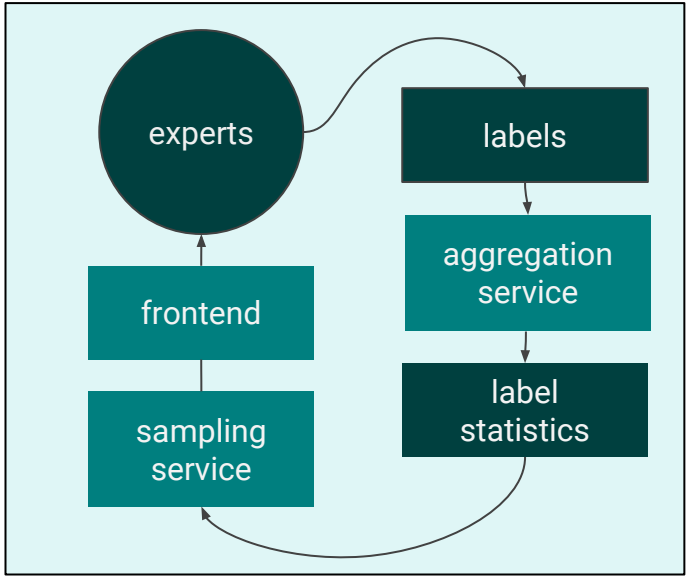




Sprint 4: Model Building 3 / Refine Inner Circle

Goals:

- Fine-tune current machine learning models
- Add machine learning models, e.g.
  - Logistic / Linear / Spline Regression
  - Boosting (AdaBoost, XGBoost)
  - Support Vector Machine (SVM)
  - Multi Layer Perceptron (MLP)
  - Convolutional Neural Networks (CNN)
  - Long Short Term Memory (LSTM)
- Gathering more labels from more experts
- Refine Inner Circle & Iterations



Sprint 5: Wrap-Up / Statistical Evaluation

Goals:

- Refinement of ML techniques
  - parameter tuning
  - development of cost-sensitive approaches
- Statistical Evaluation of
  - Machine Learning Models
  - Label Statistics
  - Rater Statistics
- Write final report and presentation

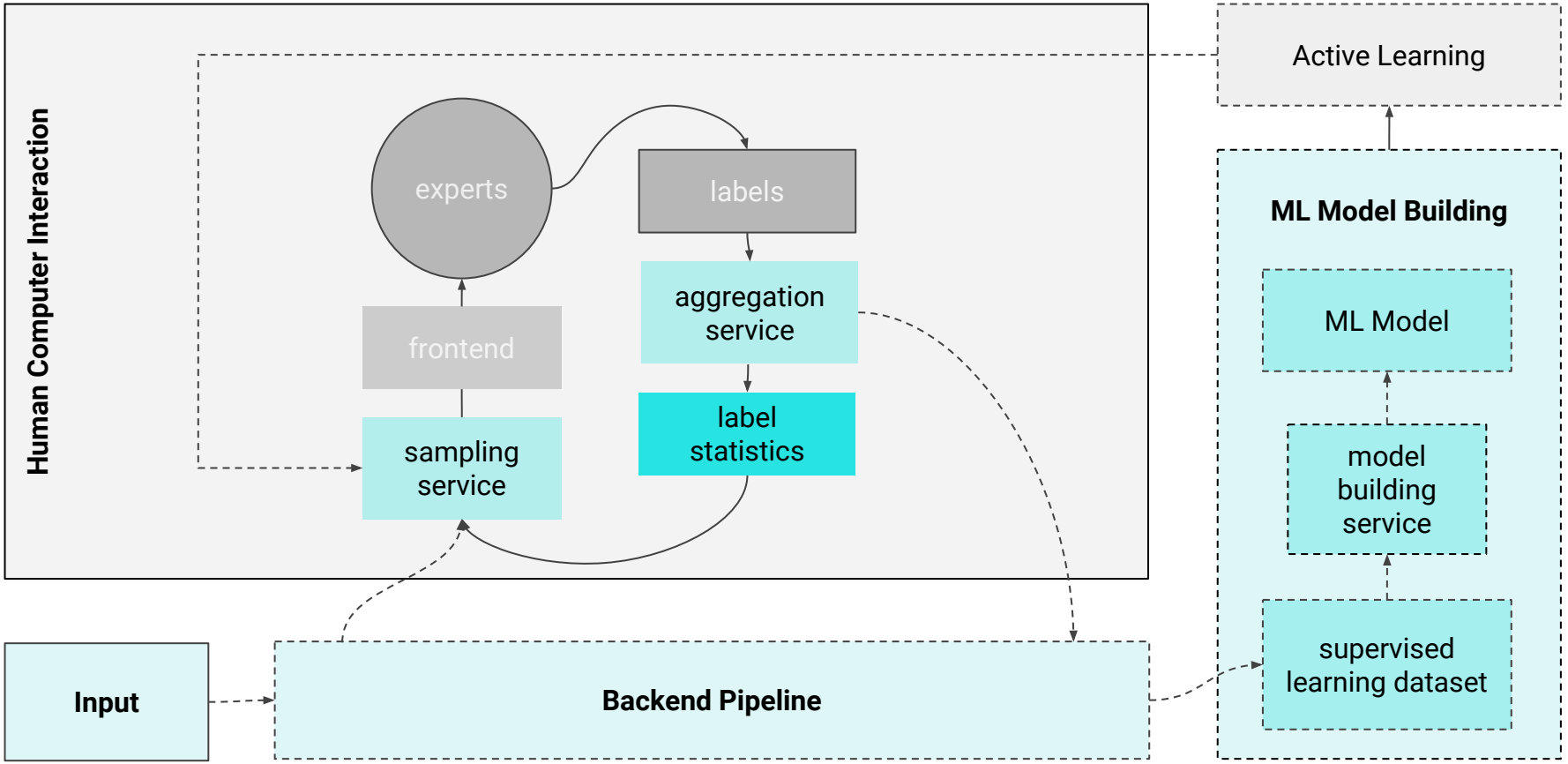
```
Confusion Matrix and Statistics

          Reference
Prediction  0  1  2  3  4
         0 52  6  0  0  0
         1 14 27  2  1  0
         2  4  1 10  2  0
         3  0  2  0  4  0
         4  1  0  0  0  2

Overall Statistics

               Accuracy : 0.7422
                 95% CI : (0.6574, 0.8154)
    No Information Rate : 0.5547
    P-Value [Acc > NIR] : 9.076e-06

                  Kappa : 0.595
 Mcnemar's Test P-Value : NA

Statistics by Class:

                     Class: 0 Class: 1 Class: 2 Class: 3 Class: 4
Sensitivity            0.7324   0.7500  0.83333  0.57143  1.00000
Specificity            0.8947   0.8152  0.93966  0.98347  0.99206
Pos Pred Value         0.8966   0.6136  0.58824  0.66667  0.66667
Neg Pred Value         0.7286   0.8929  0.98198  0.97541  1.00000
Precision              0.8966   0.6136  0.58824  0.66667  0.66667
Recall                 0.7324   0.7500  0.83333  0.57143  1.00000
F1                     0.8062   0.6750  0.68966  0.61538  0.80000
Prevalence             0.5547   0.2812  0.09375  0.05469  0.01562
Detection Rate         0.4062   0.2109  0.07812  0.03125  0.01562
Detection Prevalence   0.4531   0.3438  0.13281  0.04688  0.02344
Balanced Accuracy      0.8136   0.7826  0.88649  0.77745  0.99603
```

Results

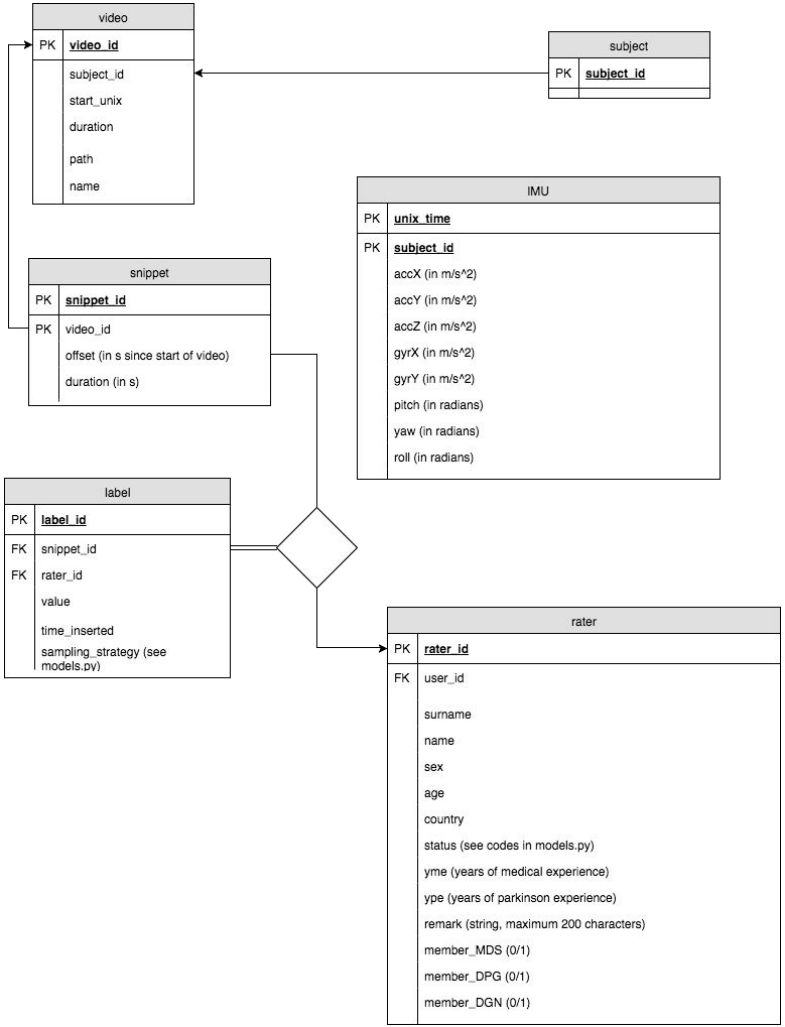
of the Big Data Lab

Software Engineering

- Spark/Flink vs Pandas
- SQL vs NoSQL
- Django vs Flask



Known: System Overview



Relational Database Model

- Backend of Collex System:
  - Flask vs Django
- Framework for Data Preprocessing
  - Spark/Flink vs numpy/pandas
- Data Model
- API specs
- Storage of IMU data
- Storage of Video data

Overview of Decisions made

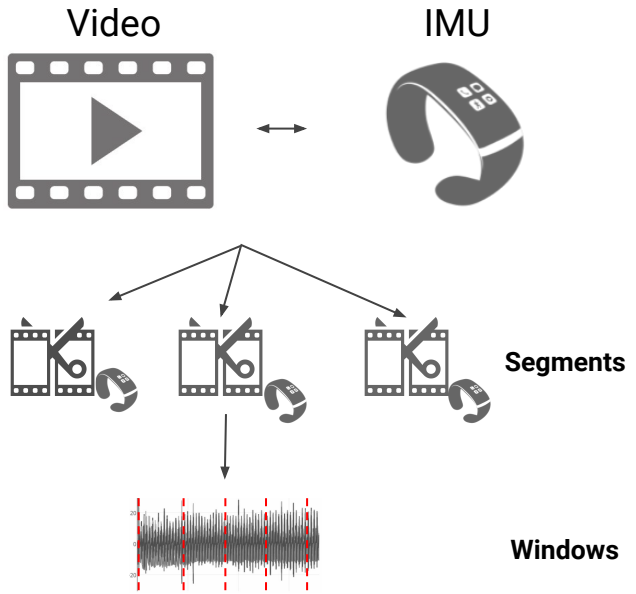
Preprocessing / Cleaning

Challenges:

- Missing Data
- Timestamp correction
- IMU data from different watches
- Matching IMU data with Video segment

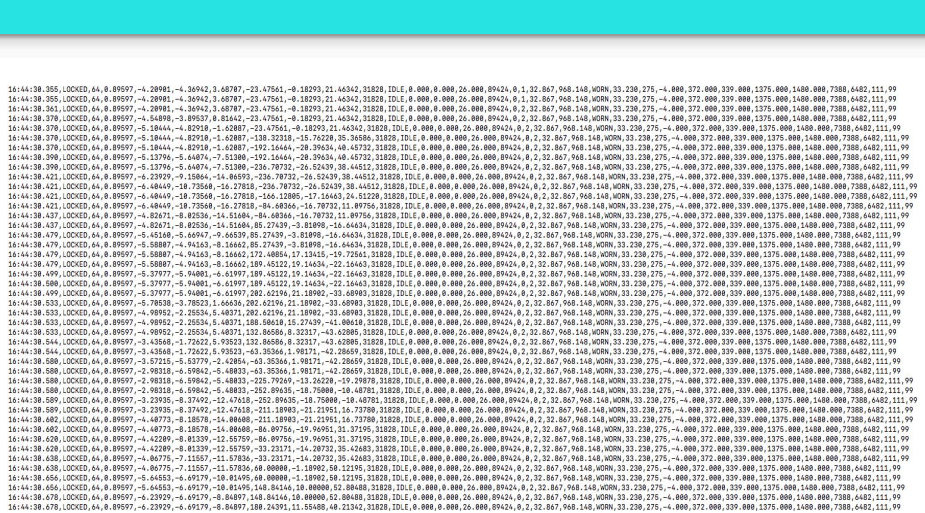
Label Collection

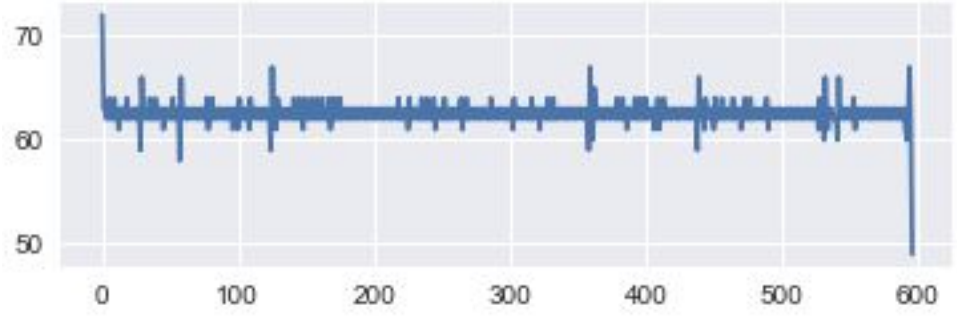
- Raw Data from 19 Patients with Parkinson's Disease (about 10min each), in total n segments of motion data, equals to about **11.400s** of raw motion data
- accounts for **1.275** windows
- Labels from 4 Expert Raters for 254 snippets each, accounts to **1.016** collected labels

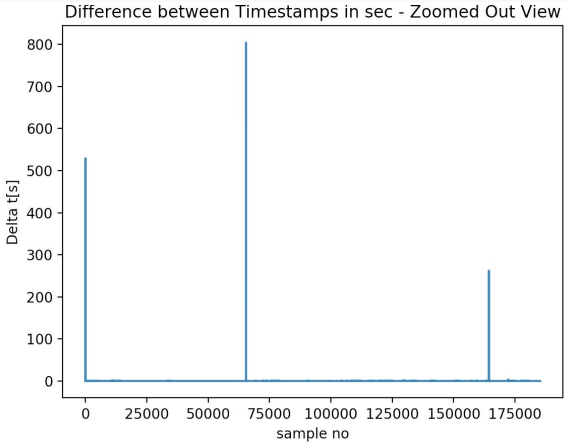


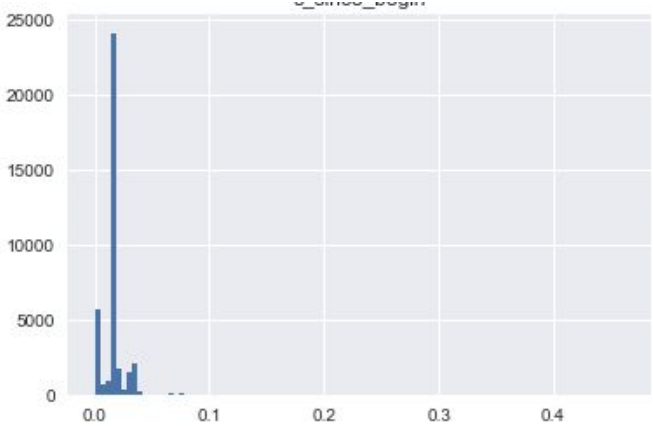
Raw IMU Data - Issues

16:44:30.370,
16:44:30.370,
16:44:30.370,
16:44:30.370,
16:44:30.390,
16:44:30.390,
16:44:30.421,
16:44:30.421,
16:44:30.421,
16:44:30.421,
16:44:30.437,
16:44:30.437,
16:44:30.479,
16:44:30.479,
16:44:30.479,
16:44:30.479,
16:44:30.479,
16:44:30.499,
16:44:30.500,
16:44:30.499,
16:44:30.533,
16:44:30.533,
16:44:30.533,
16:44:30.533,
16:44:30.544,
16:44:30.544,
16:44:30.580,
16:44:30.580,
16:44:30.580,
16:44:30.580,
16:44:30.589,
16:44:30.589,
16:44:30.602,
16:44:30.602,
16:44:30.620,
16:44:30.620,



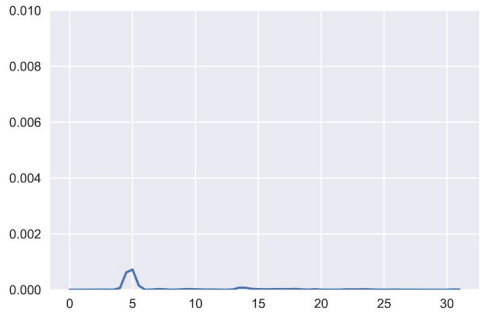


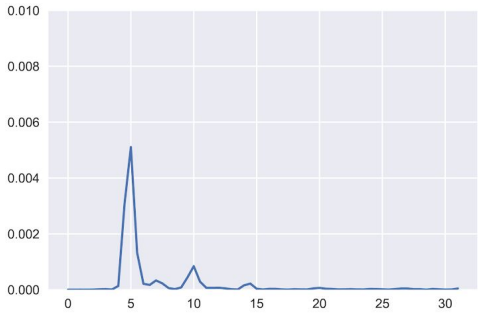


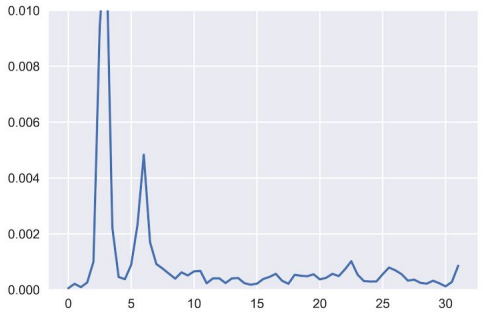


Cleaning & Preprocessing

- Cleaning of Raw Data: Duplicate Timestamps, Transmission Errors, Matching
- Preprocessing of Raw Data (Discrete Wavelet Transform, PSD, Periodogram, Welch Method, Kalman Filtering)







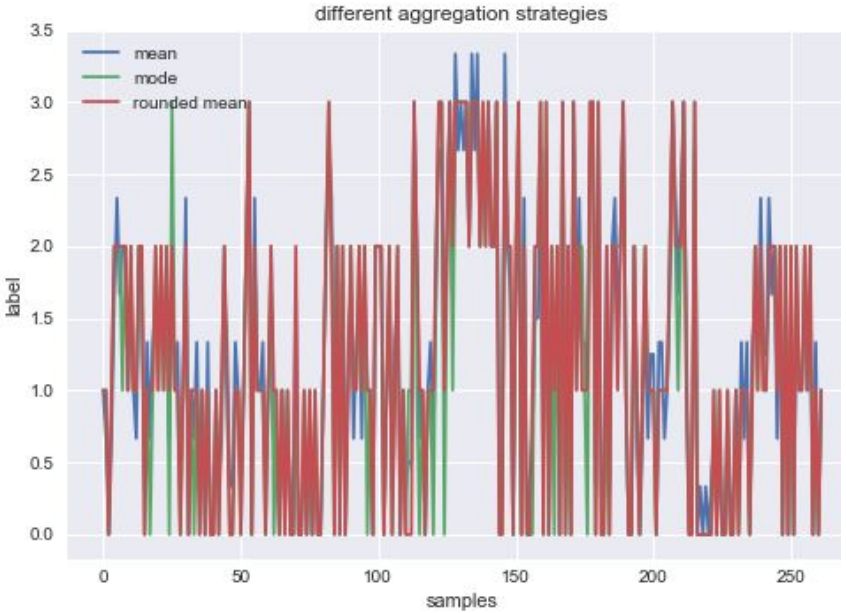
Feature Engineering

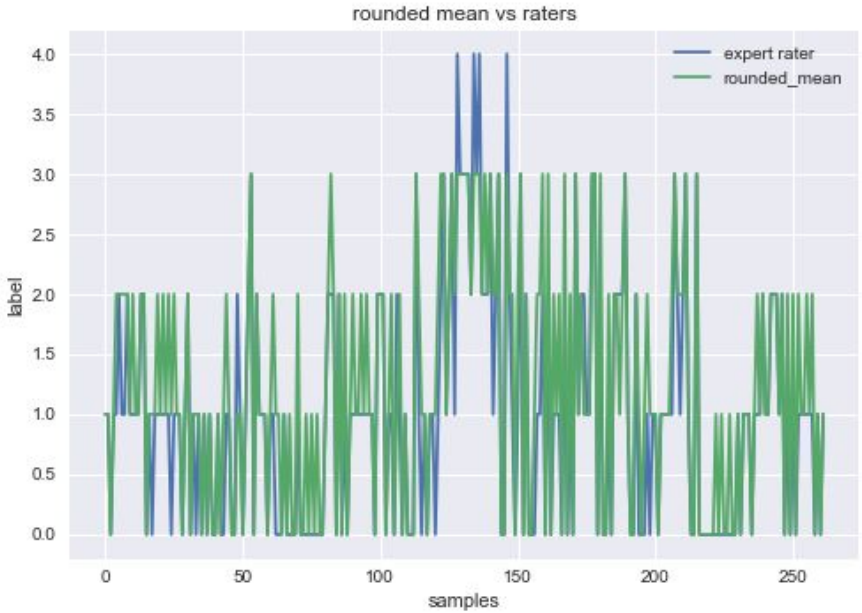
- Energy of each segment
- Energy between 3-7 Hz (characteristic frequency band of rest tremor)
- Maximum energy
- Maximum energy between 3-7 Hz
- Power Spectral Density in the frequency bands between 0 and 31 Hz at a 0.5Hz step size

Label Engineering

- 1 Random Rater
- Mean
- Mode
- Aggregation of Agreement
- Cost-Sensitivity

Aggregation Strategies

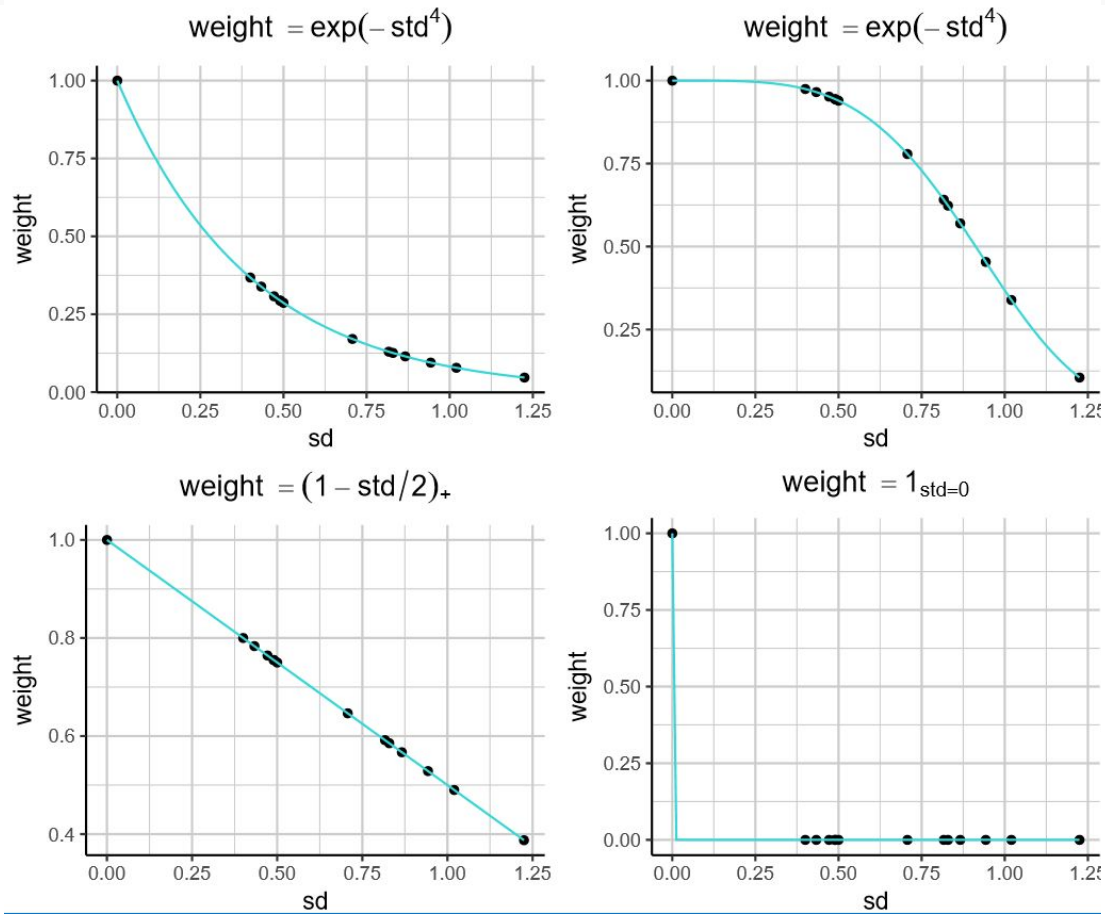




Cost-sensitive learning

- Different ways to do cost-sensitive learning
  - Thresholding
  - **Direct weighting**
  - **Rebalancing the data**
- Rebalancing is done by rejection resampling
  - with a probability of **1 - weight** reject the observation
  - otherwise add observation to data set
- How to choose weights?

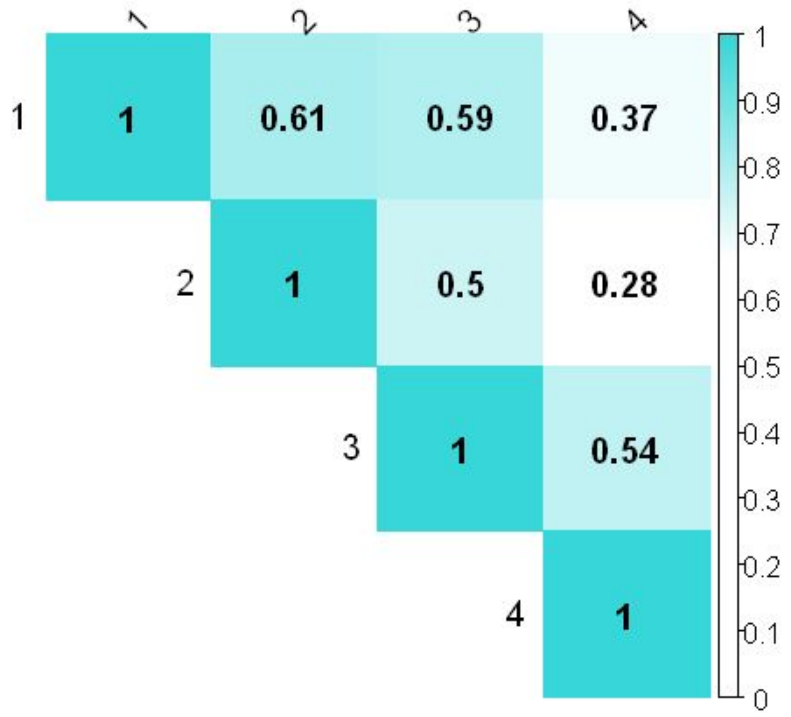
Weighting strategies



Model Building & Statistical Evaluation

Challenges:

- class imbalance
- label noise
- few data



Interrater Agreement

Comparison of different machine learning approaches

Model	1RR	1ER	SSD	Mod	Avg	Agr
Linear Regression*	-	-	.61	-	.58	1.06
Log. Regr. L1/L2**	.31	.31	.48	.31	-	.71
Random Forest**	.68	.64	.61	.69	.41*	**.84**
AdaBoost**	.64	.72	.73	.66	-	**.81**
XGBoost**	.63	.69	-	.67	-	**.83**
SVM**	.51	.61	.58	.51	-	.71
FF Neural Network**	.42	.51	-	.40	.42	.53
CNN**	.52	.52	-	.40	.41	.53
LSTM (raw)**	-	-	-	-	.34	-

Table 3: Overview of different Model Performances. Abbreviations: SLA = Best Performance Small Label Amount, LLA = Best Performance Large Label Amount, 1RR = 1 Random Rater Baseline, 1ER = 1 Expert Rater, SSD = Small Standard Deviation, Mod = Mode, Avg = Average / Mean, Agr = Agreement, WCS = Weighted / Cost-Sensitive Approach, *MSE, **Accuracy, in **bold**: best performance

How do raters influence results of the ML model?

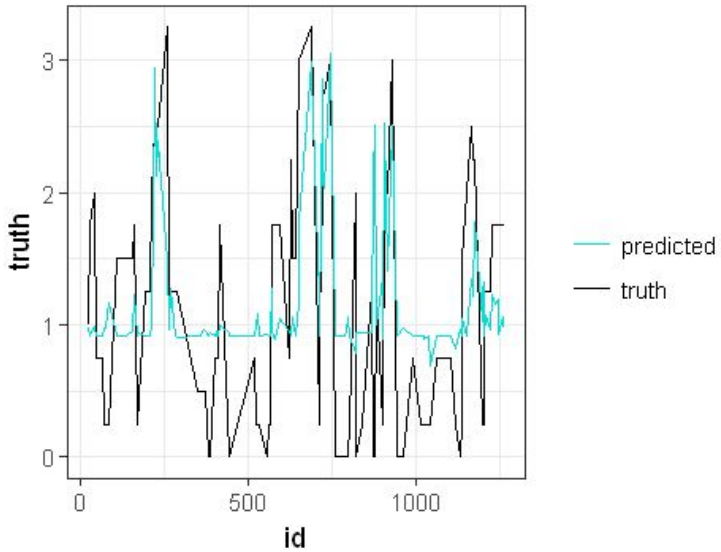
- in reality: train and evaluate on labels of **one** rater
- → high variance in rater's results
- results of one model (same hyperparameters, same evaluation method)

	Student	**Doctor 1**	**Expert**	**Doctor 2**
RF classification*	0.688	0.742	0.652	0.640

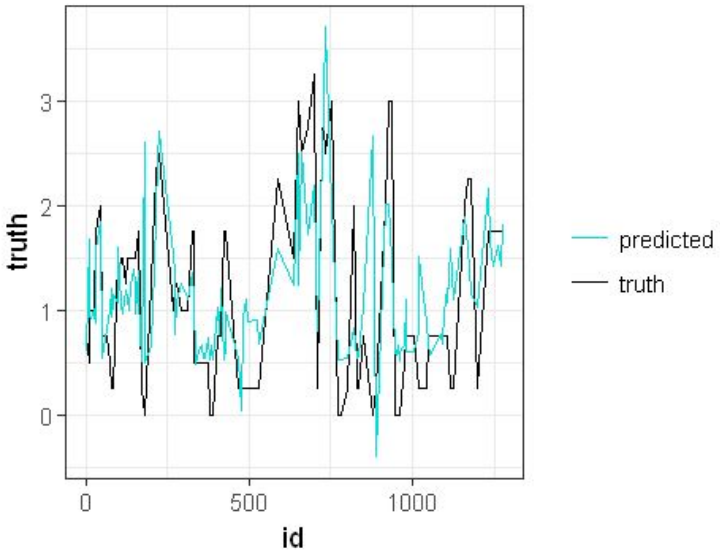
- Which setting to choose?
- How to find something near to the "ground truth"?

* evaluated by 10-fold CV

Linear / Logistic / Spline Regression



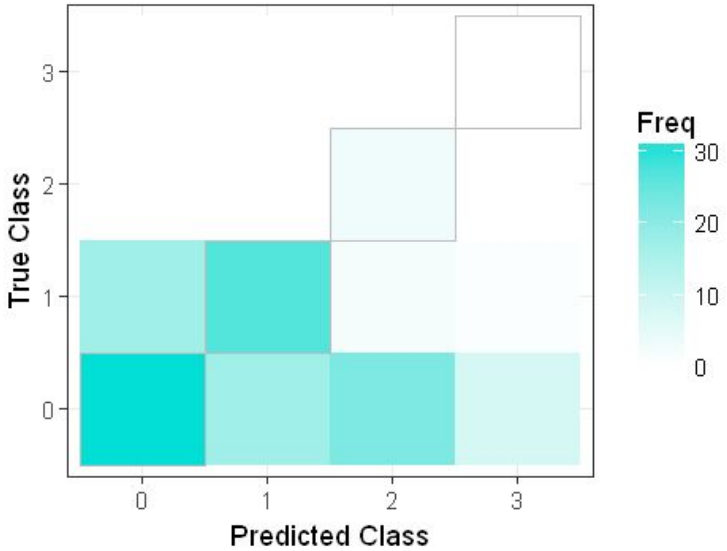
Linear Regression (MSE 0.558)



Spline Regression (MSE 0.468)

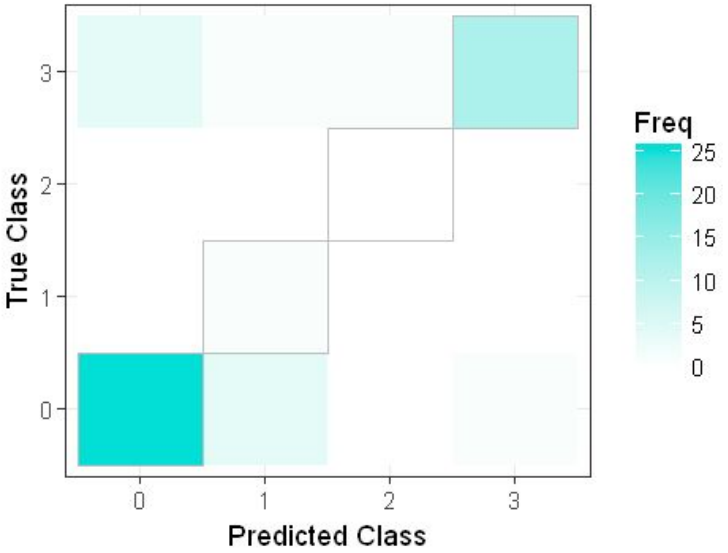
SVM - without Tuning

- Kernel SVMs: Gaussian Kernel
- Without Tuning: ACC = 0.477 (Mode)



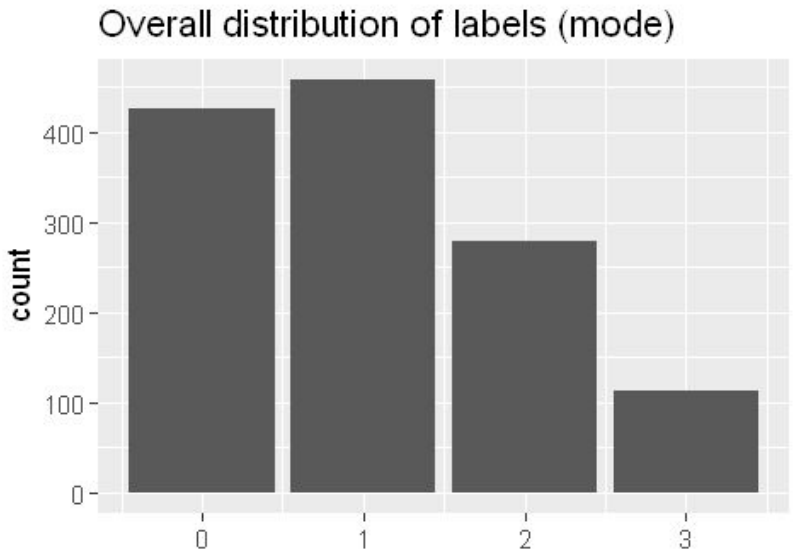
SVM - Tuning

- With Tuning: ACC = 0.529 (Cost-Parameter C and Kernel Width by Grid Search)

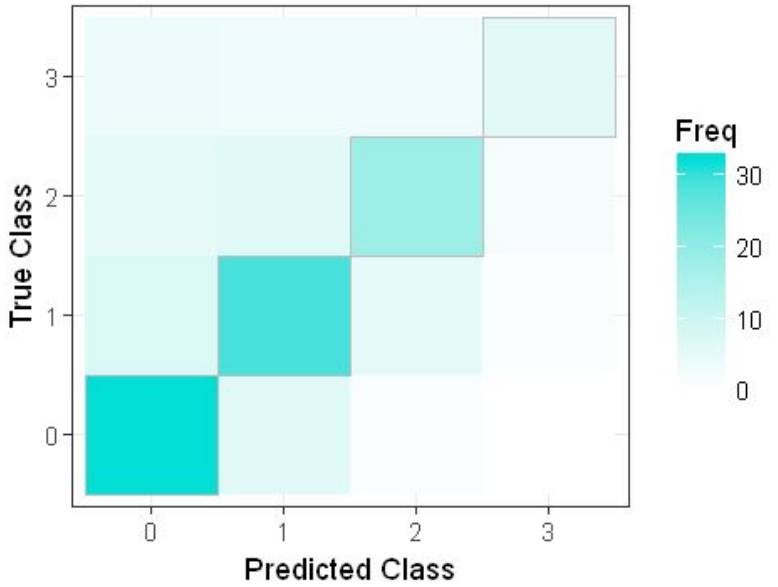


SVM - Challenges for Dataset

- Number of Features
  - no accuracy gain
- Class Imbalance
  - SMOTE: ACC = 0.577



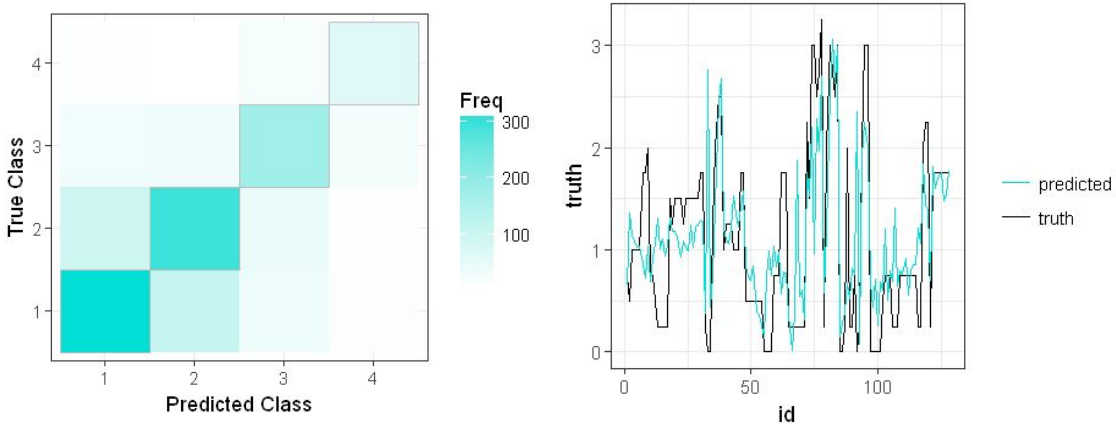
AdaBoost



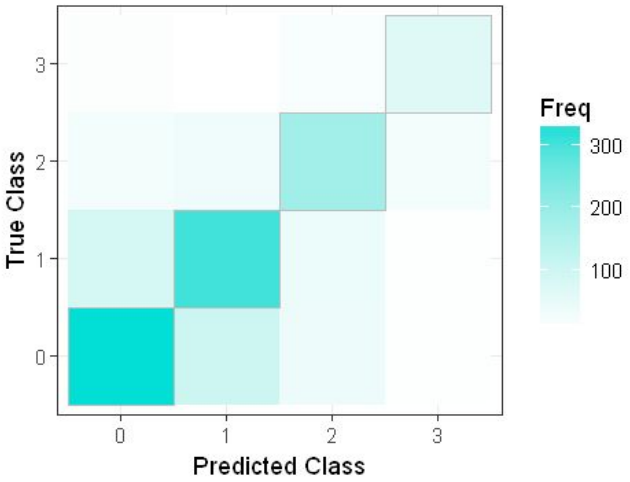

```
Resample Result
Task: mode
Learner: classif.ada.multiclass
Aggr perf: acc.test.mean=0.654,mmce.test.mean=0.346
Runtime: 79.4886
```

XGBoost

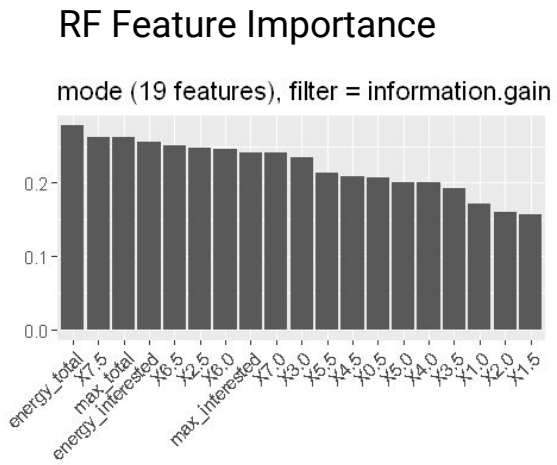
- Classification with tree boosters (acc 0.673)
- Regression with linear boosters (mse 0.39)
- Tuning by grid search



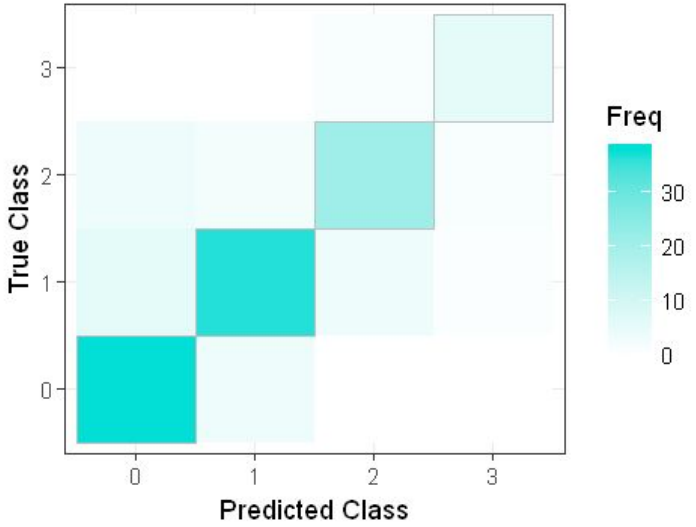
Random Forest for mode




```
Confusion Matrix and Statistics

          Reference
Prediction   0   1   2   3
         0 332  85  27   8
         1 102 306  31   1
         2  37  42 180  16
         3   7   7  27  67

Overall Statistics

               Accuracy : 0.6941
                 95% CI : (0.668, 0.7193)
    No Information Rate : 0.3749
    P-Value [Acc > NIR] : < 2e-16

                  Kappa : 0.5612
 Mcnemar's Test P-Value : 0.05879

Statistics by Class:

                     Class: 0 Class: 1 Class: 2 Class: 3
Sensitivity            0.6946   0.6955   0.6792  0.72826
Specificity            0.8494   0.8395   0.9059  0.96534
Pos Pred Value         0.7345   0.6955   0.6545  0.62037
Neg Pred Value         0.8226   0.8395   0.9150  0.97858
Precision              0.7345   0.6955   0.6545  0.62037
Recall                 0.6946   0.6955   0.6792  0.72826
F1                     0.7140   0.6955   0.6667  0.67000
Prevalence             0.3749   0.3451   0.2078  0.07216
Detection Rate         0.2604   0.2400   0.1412  0.05255
Detection Prevalence   0.3545   0.3451   0.2157  0.08471
Balanced Accuracy      0.7720   0.7675   0.7926  0.84680
```



Random Forest cost-sensitive




```
                 Accuracy : 0.7969
                   95% CI : (0.7167, 0.8628)
      No Information Rate : 0.375
      P-Value [Acc > NIR] : < 2.2e-16

                    Kappa : 0.7074
   Mcnemar's Test P-Value : NA
```

	Class: 0	Class: 1	Class: 2	Class: 3
Sensitivity	0.9048	0.7708	0.7000	0.75000
Specificity	0.8837	0.9125	0.9388	0.97500
Pos Pred Value	0.7917	0.8409	0.7778	0.66667
Neg Pred Value	0.9500	0.8690	0.9109	0.98319
Precision	0.7917	0.8409	0.7778	0.66667
Recall	0.9048	0.7708	0.7000	0.75000
F1	0.8444	0.8043	0.7368	0.70588
Prevalence	0.3281	0.3750	0.2344	0.06250
Detection Rate	0.2969	0.2891	0.1641	0.04688
Detection Prevalence	0.3750	0.3438	0.2109	0.07031
Balanced Accuracy	0.8942	0.8417	0.8194	0.86250

Cost-Sensitive Approach & Agreement Aggregation

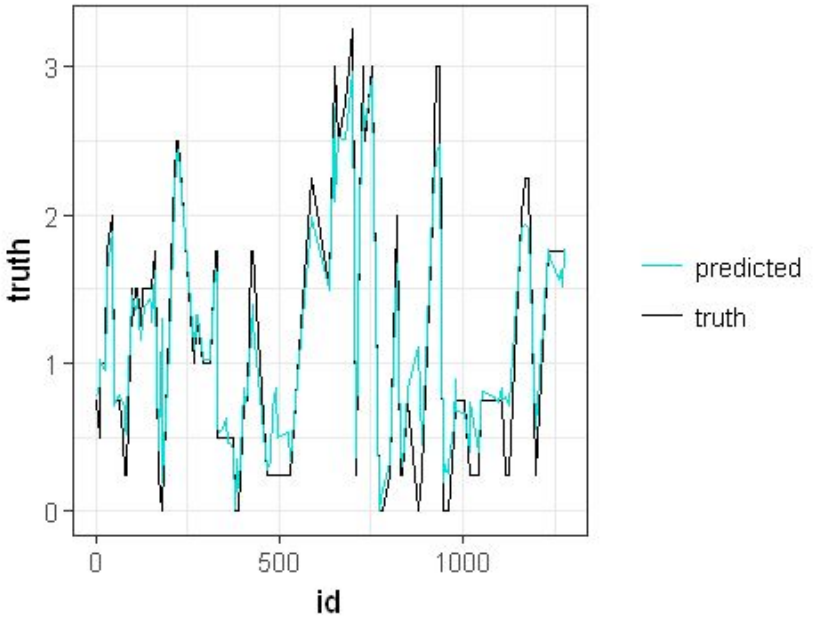
- tested with Random Forest
- Full agreement portion: 19.4%

Model	Accuracy
(a) Small Dataset	.84
(b) Large Dataset	.62

Model	Accuracy
RF 1 randomly picked rater	.65
RF mode of all raters	.69
RF /w cost-sensitivity	.80

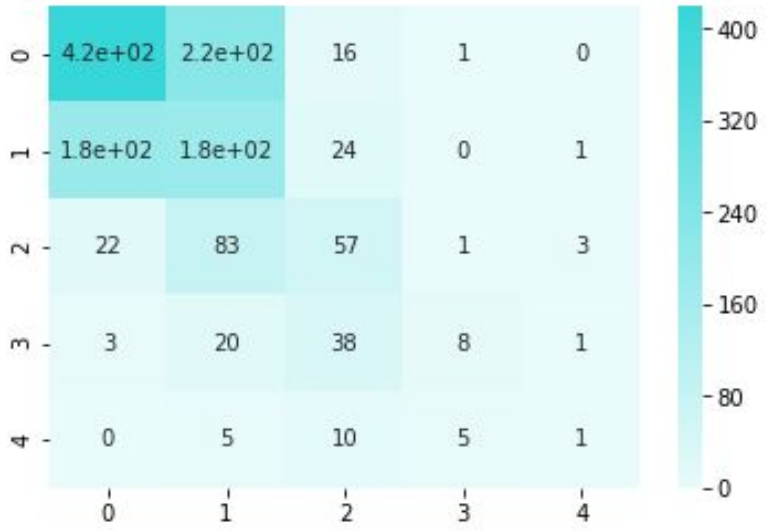
Random Forest regression

- MSE of 0.417
- even there weighting according to raters confidence achieved good results



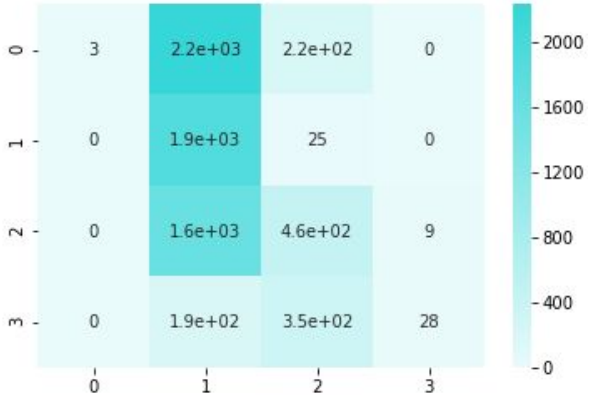
Feed Forward NN

- lower performance than classical machine learning methods
- aggregation approach slightly improved performance

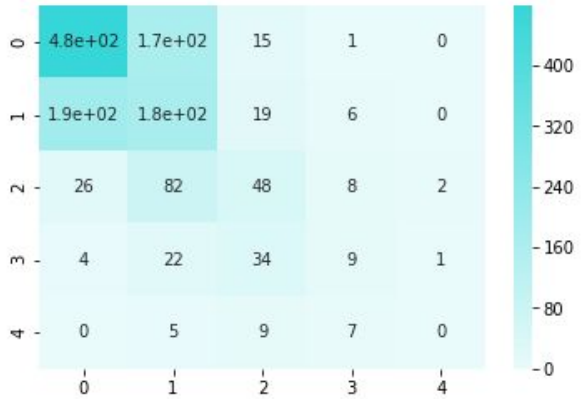


Deep Learning: CNN & LSTM

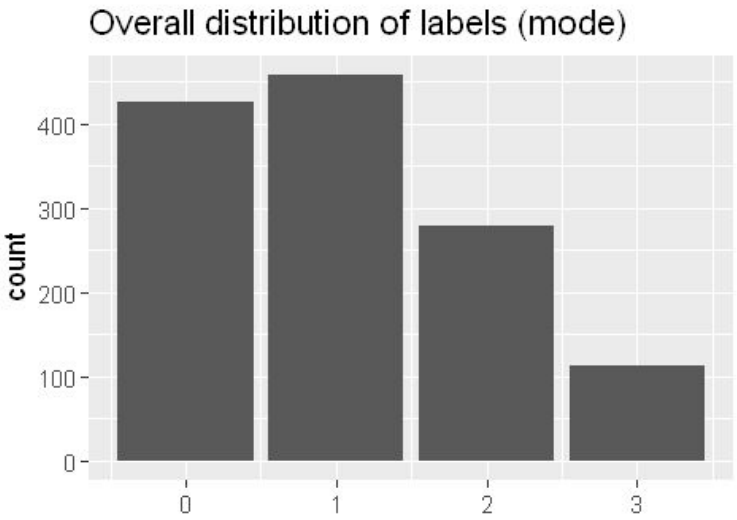
- too little data
- difficult to choose hyperparameters
- method not as robust, sensitive to variance ir labels



LSTM



CNN



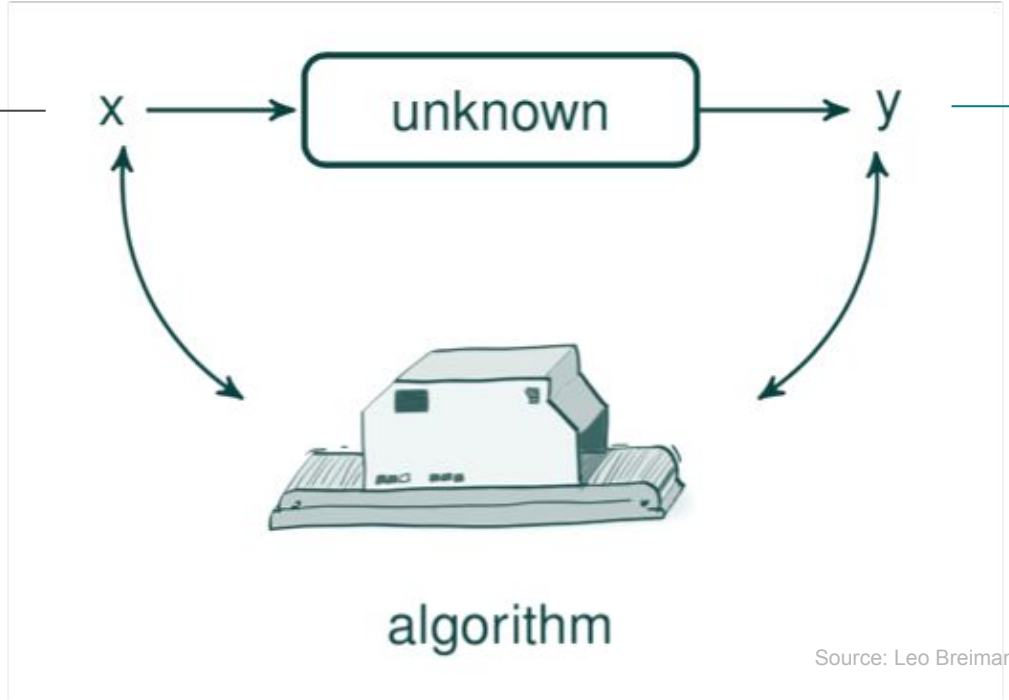
Class Imbalance

→ SMOTE 5nn
→ class weights

Conceptual Considerations (1)

Feature Engineering

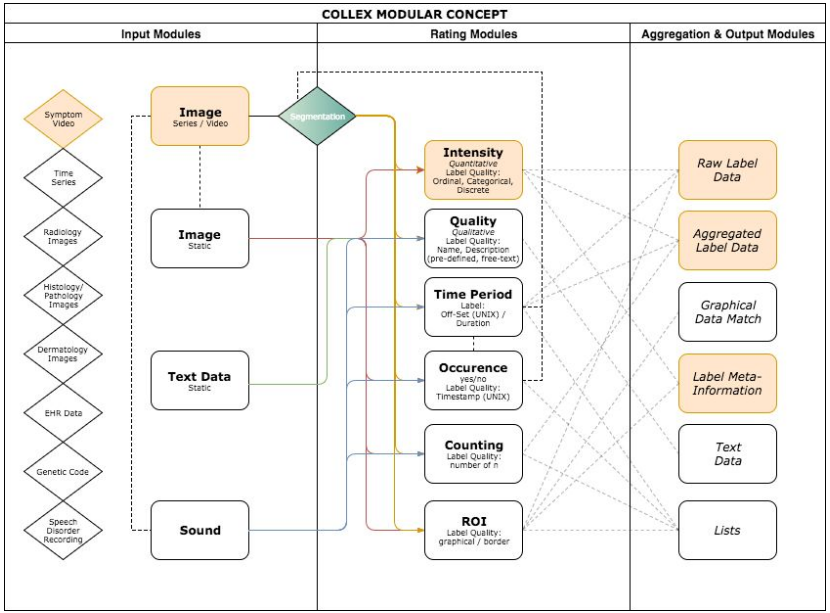
- Manual/Automatic feature engineering
- dimensionality reduction
- filtering
- data augmentation
- outlier removal
- balancing of classes
- ...

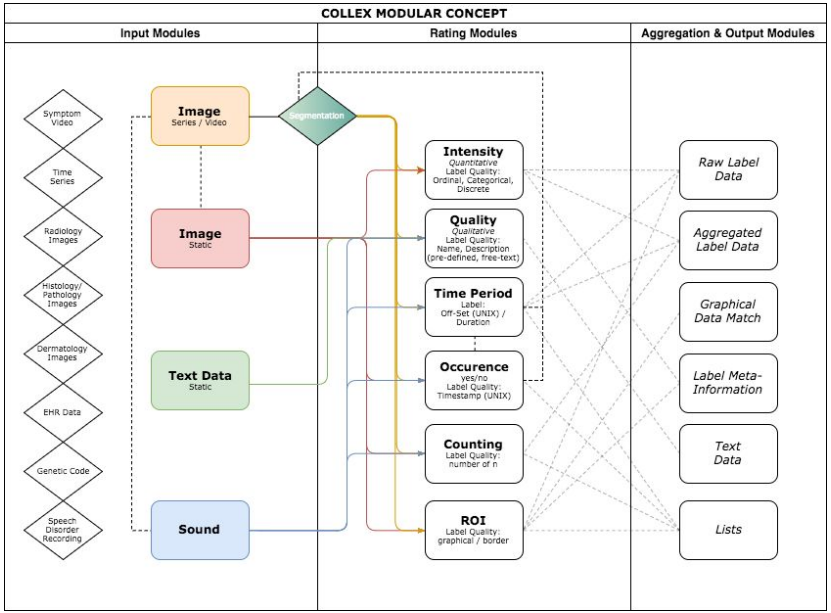


Label Engineering

Source: Leo Breiman (2001)Statistical Modeling: The Two Cultures

Conceptual Considerations (2)





Summary & Key Learnings

- **Aggregation of multiple labels** is a very helpful method to infer ground truth
- **Aggressive Sample Weighting** is gaining accuracy
- **Cost-Sensitive Approach**, which takes the costs for each label within one segment into account is gaining accuracy
- Large **Inter-Rater Disagreement** / Variability
- Inter-Rater Variability is **very valuable** in inferring ground truth
- **Quality of Labels** over **Quantity of labels** for medical data?

Outlook

- Add Active Learning Component
- Refining the current use case
  - Collecting more labels from more experts
  - Collecting more data (more severe cases)
- Other use cases

is a formalized framework system to collect **large-scale, high-quality, correctly labeled training data** from experts for machine learning applications in the biomedical domain

Team Members & Roles

Ahmet Gündüz: Software Development, Preprocessing, Deep Learning

Gunnar CS König: Software Development, Preprocessing, Deep Learning

Julia M Moosbauer: Software Development, Statistics, Model Building

Franz MJ Pfister: Software Development, Model Building, Domain Knowledge

Thank you for your attention.

COLLEX // A-Team

Ahmet Gündüz
Gunnar CS Koenig
Julia M Moosbauer
Franz MJ Pfister

SCHÖN KLINIK

DATA SCIENCE@LMU STATISTICS + INFORMATICS