Forecasting Patient Enrollment for Clinical Trials

Final Presentation
September 11th, 2020

01
Introduction and Goals
Introduction to the project and the background of clinical trials

02
Data
Overview on the different data sources, data crawling and their storage

03
Preprocessing
Introducing our Preprocessing Pipeline with Custom Transformers

04
Hyperparameter Tuning
Fine tuning and selecting models to improve the prediction performance

05
Results
Reviewing the prediction capability of our model

06
Insights and Outlook
Analysis of the results and suggestions for further research

Introduction and Goals

- Who we are
- Project Management
- Scope of the Project
- Goals
- Introduction to Clinical Trials

The Team

Carolin Holtermann
Master Data Science

Giang Hoang
Master Business Informatics

Luka Biedebach
Master Data Science

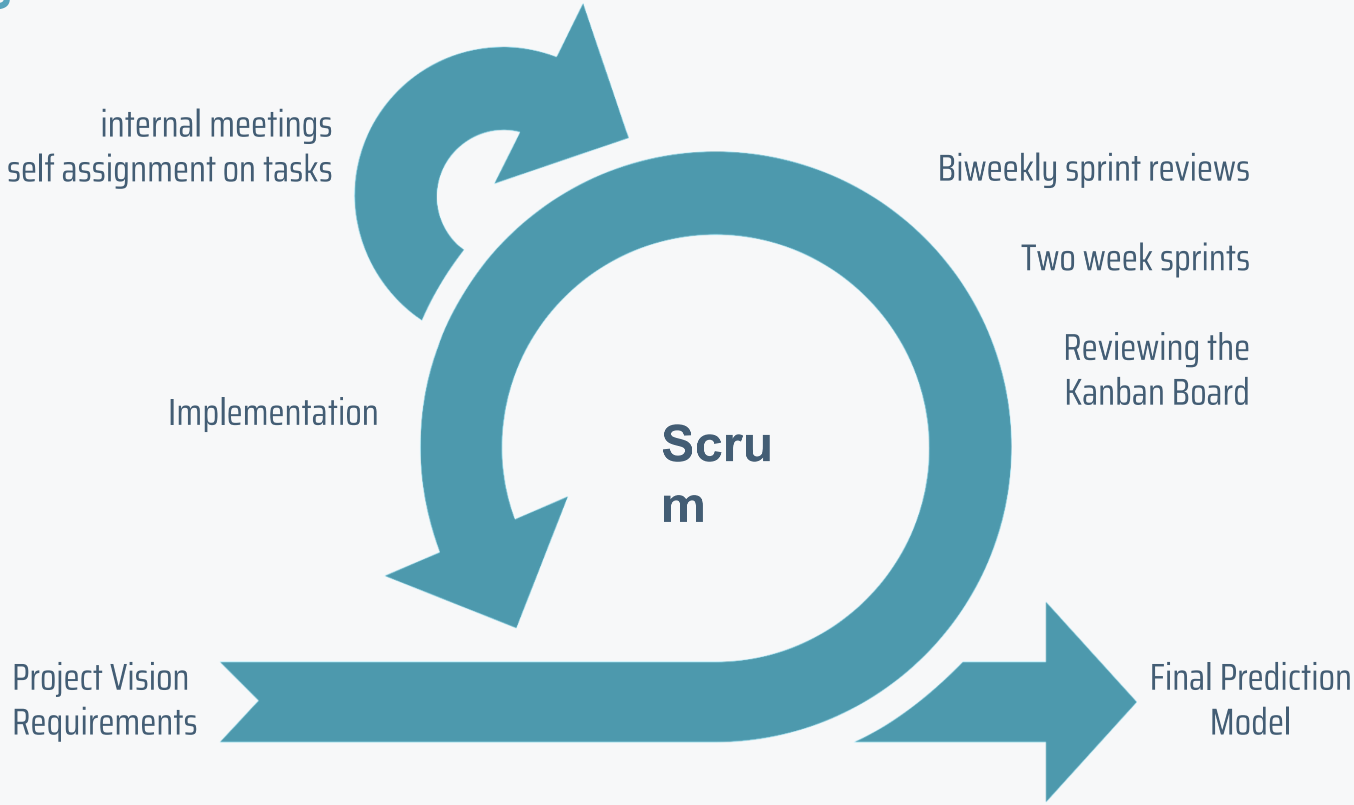
Stefan Sousa
Master Business Informatics

Wei-Yi Chen
Master Business Informatics

Project Management

Niklas
=Project Owner

Tommi
=Project Owner



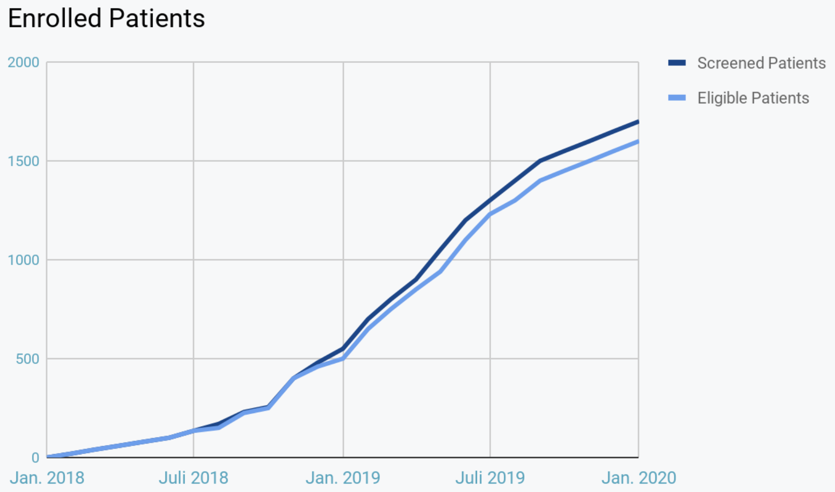
Project Outline

Challenge of pharmaceutical companies to decide:

- Which countries participate in a study? How many participants?
- How long study will take?

Goals of our project :

- Understand factors that influence patient enrolment during trial
- Build model to estimate enrolment duration



Project Scope

Information Gathering
How to clinical trials work and what do we need to predict their duration?

Data Gathering
Find data sources and learn how to crawl them

Data Storage
Create a MongoDB and learn how to work with it

Preprocessing
Write custom transformers for every group of features

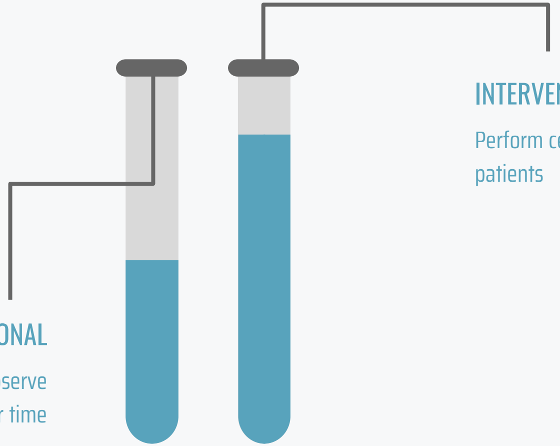
Building a Pipeline
Combining all required steps in a pipeline

Hyperparameter Tuning
Come up with different ideas on how to increase the prediction accuracy

Improving the Model
Tuning the hyperparameters in a optimization framework

Clinical Trials

- Study with the goal to acquire new medical knowledge about diseases, therapies, drugs and devices
- Pharmaceutical companies are legally obliged to successfully conduct clinical trials before releasing a new drug or device



INTERVENTIONAL

Perform certain treatment to patients

OBSERVATIONAL

Measure and observe patients over time

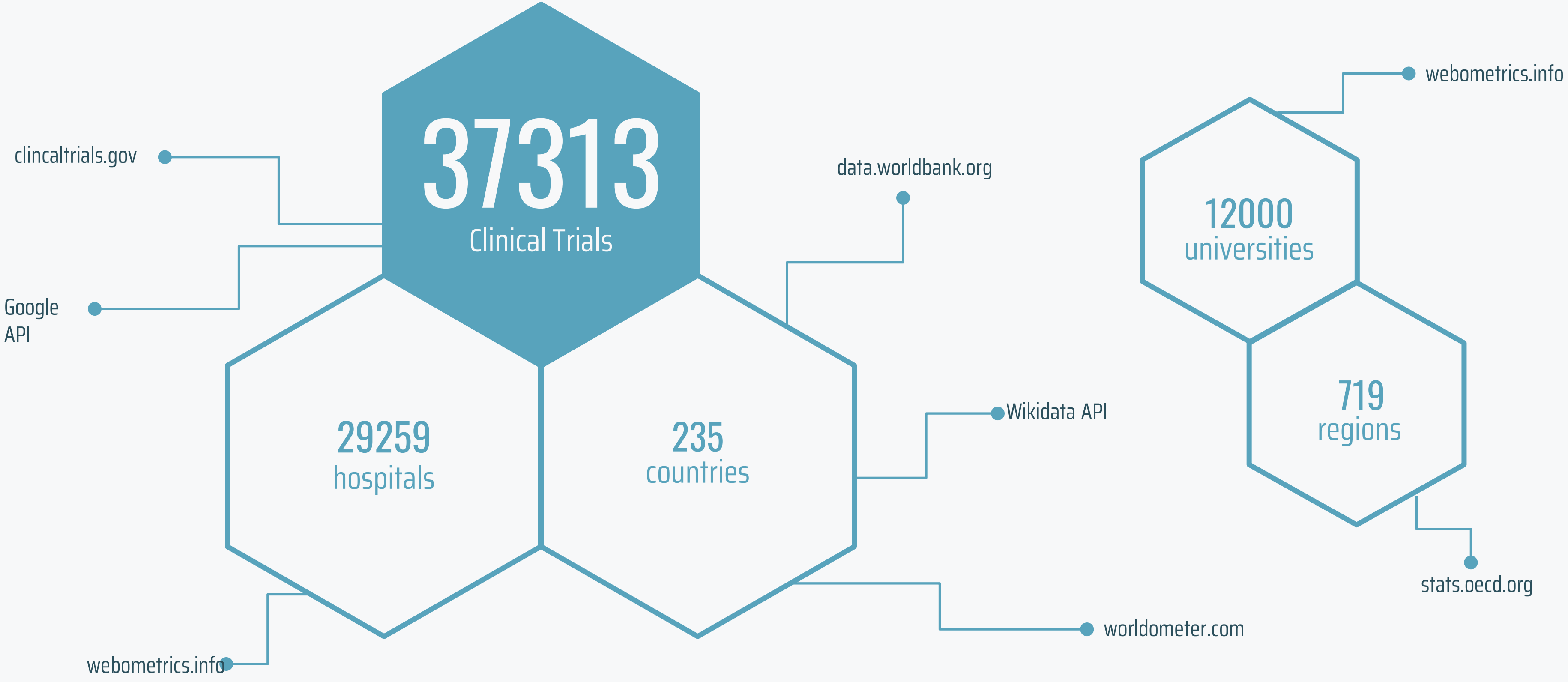
Phases

	PHASE 0	PHASE 1	PHASE 2	PHASE 3	PHASE 4
	Lab Studies	Safety and dosage	Efficacy and side effects	Efficacy and monitoring of adverse reactions	Post-Market Safety Monitoring
Typical Number of Participants		20 to 100	Up to several hundred	300 to 3000	Several thousand
Typical Duration		Several months	Up to 2 years	1 to 4 years	

Data

- Data Sources and Data Gathering
- Analyzing the Data

Data Sources



Data Gathering - Clinical Trials Data

- From https://clinicaltrials.gov/
  - largest clinical trials database
  - run by the United States National Library of Medicine (NLM) at the National Institutes of Health
- Filters:
  - Status = Completed
  - Intervention Type = Drug
  - Study Type = Interventional
  - Phase = Phase 2, 3 and 4

_id: ObjectId("5e907c32dd30d42bdfcf9970")
Rank: 1
NCTId: "NCT00000134"
OrgFullName: "Johns Hopkins Bloomberg School of Public Health"
OrgClass: "OTHER"
BriefTitle: "Studies of the Ocular Complications of AIDS (SOCA)--Cytomegalovirus Re..."
OfficialTitle: "Cytomegalovirus Retinitis Retreatment Trial"
BriefSummary: "To compare the relative merits of three therapeutic regimens in patien..."
StudyType: "Interventional"
OverallStatus: "Completed"
Phase: Array
StartDate: "December 1992"
StartDateType: Array
StatusVerifiedDate: Array
CompletionDate: "March 1995"
CompletionDateType: Array
Condition: Array
ConditionAncestorId: Array
ConditionAncestorTerm: Array
ConditionBrowseBranchAbbrev: Array
ConditionBrowseLeafName: Array
ConditionBrowseLeafRelevance: Array
ConditionMeshId: Array
ConditionMeshTerm: Array
LeadSponsorName: Array
LeadSponsorClass: Array
CollaboratorName: Array
CollaboratorClass: Array
EligibilityCriteria: Array
EnrollmentCount: 279
EnrollmentType: "Actual"
HealthyVolunteers: "No"

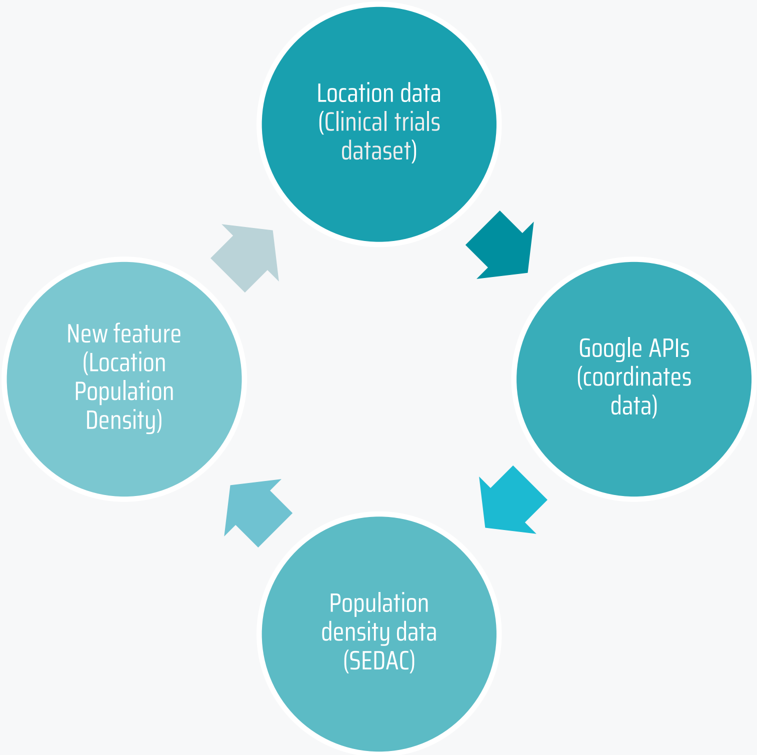
Clinical Trials: Raw Features

Organisational Information	• OrgFullName, OrgClass, StudyType • LeadSponsorName, LeadSponsorClass, CollaboratorName, CollaboratorClass
Disease related features	• Condition, ConditionBrowseLeafName and ConditionBrowseLeafRelevance • ConditionAncestorId/ConditionAncestorTerm • ConditionMeshId/ConditionMeshTerm
Study Design related features	• EligibilityCriteria, HealthyVolunteers, Gender, StdAge • DesignAllocation, DesignInterventionModel, DesignPrimaryPurpose • InterventionName • IsFDARegulatedDrug • Enrollmentcount
Study Location	• LocationFacility, LocationCity, LocationCountry
Other Features	• ArmGroupLabel • Keyword

Additionally Crawled Features

Country level	worldshare, GDP, fertility, capital city, life expectancy, median Age, …	- World bank - Wikidata - Worldometer
Regional level	regional age structure, regional unemployment rate, location density,…	- OECD - Google API
Site level	university world rank, hospital world rank, hospital coordinates, facility zip code,…	- CSIC - Google API

Data Enrichment - Clinical Trials Data

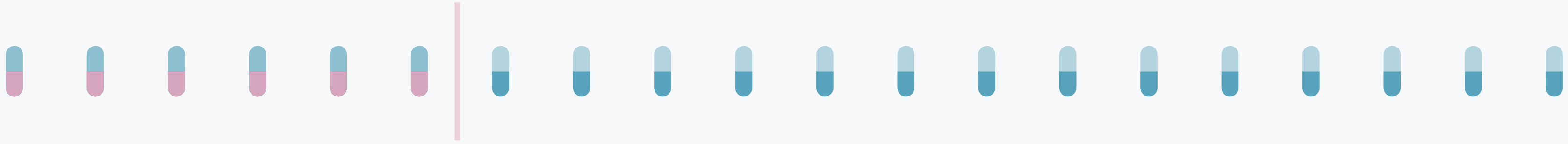


Target variable: Enrollment Duration

- Time needed to enroll the patients
- Artificially created from the StartDate and CompletionDate

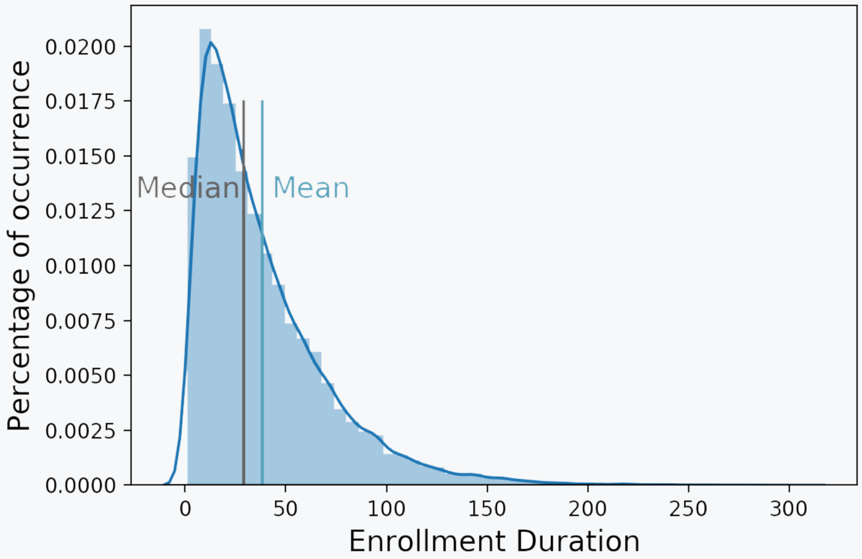
- A lot of studies don't clearly distinguish between the study and the enrollment duration
- Some features include this information in a free text field
- Estimated 30% of the study duration:
    - according to a medical paper[1]
    - according to lightweight NLP processing

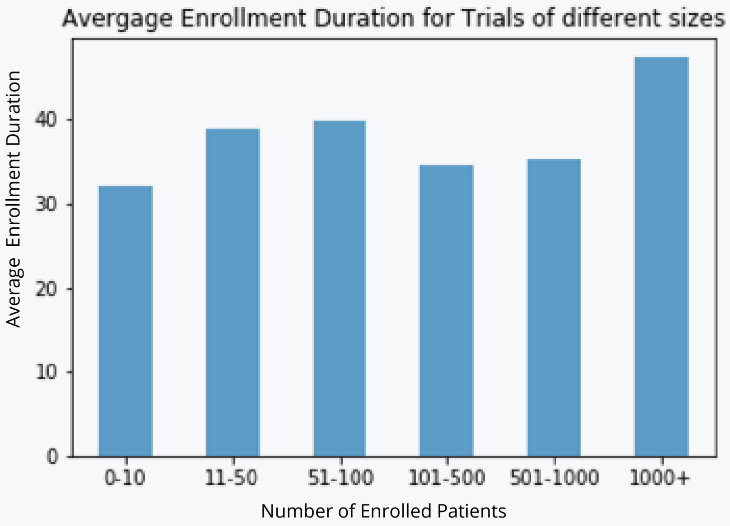
Study Duration



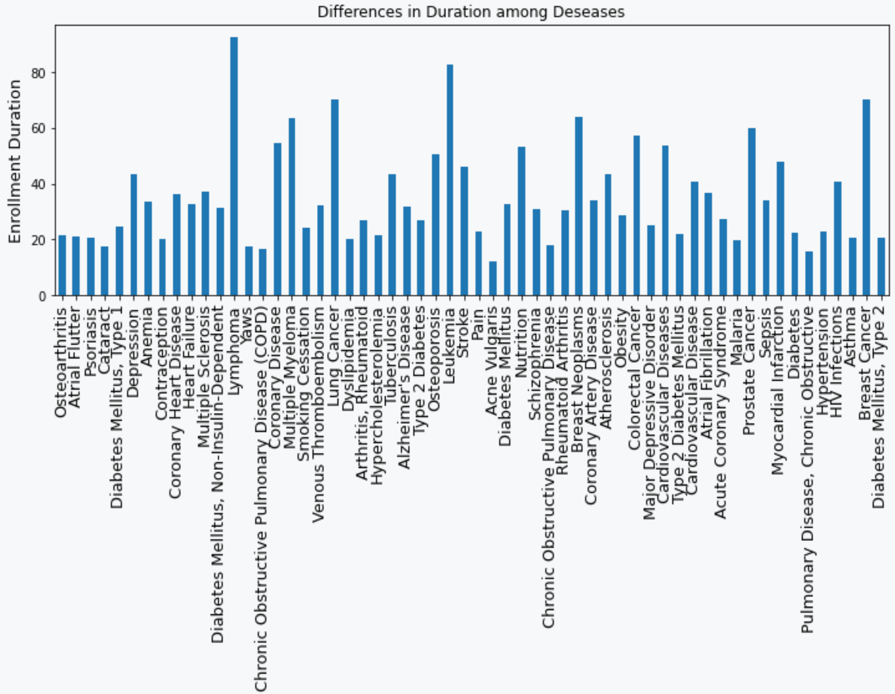
[1] https://www.cognizant.com/whitepapers/patients-recruitment-forecast-in-clinical-trials-codex1382.pdf

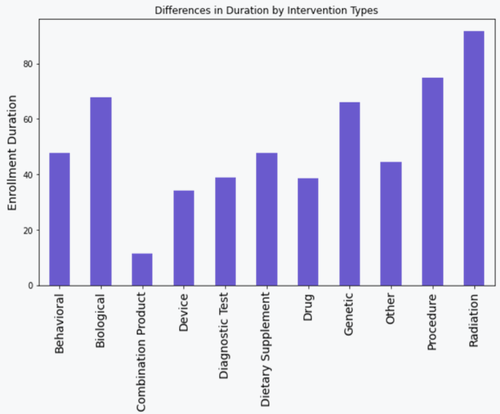
Data Analysis: Target variable

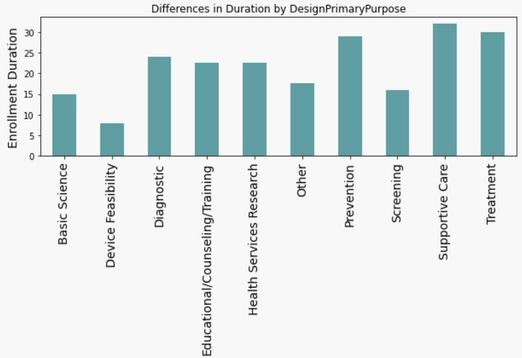




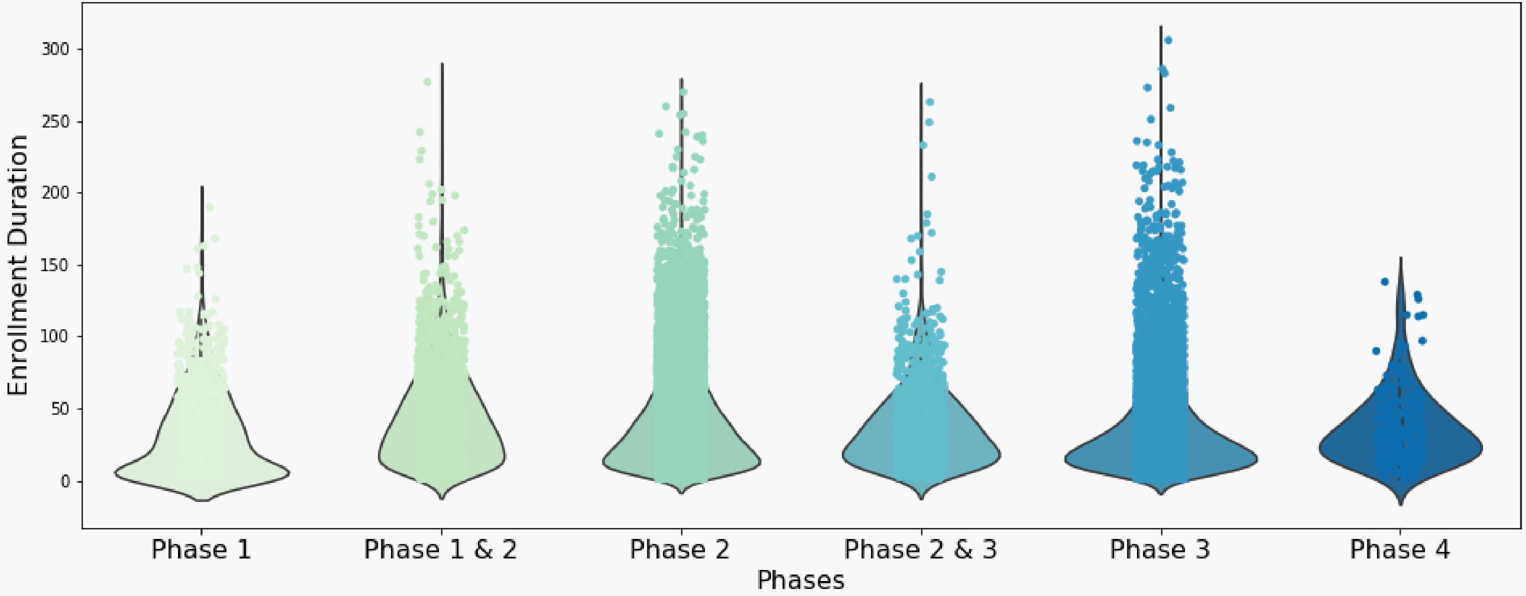
Data Analysis: Correlations with the Target Variable





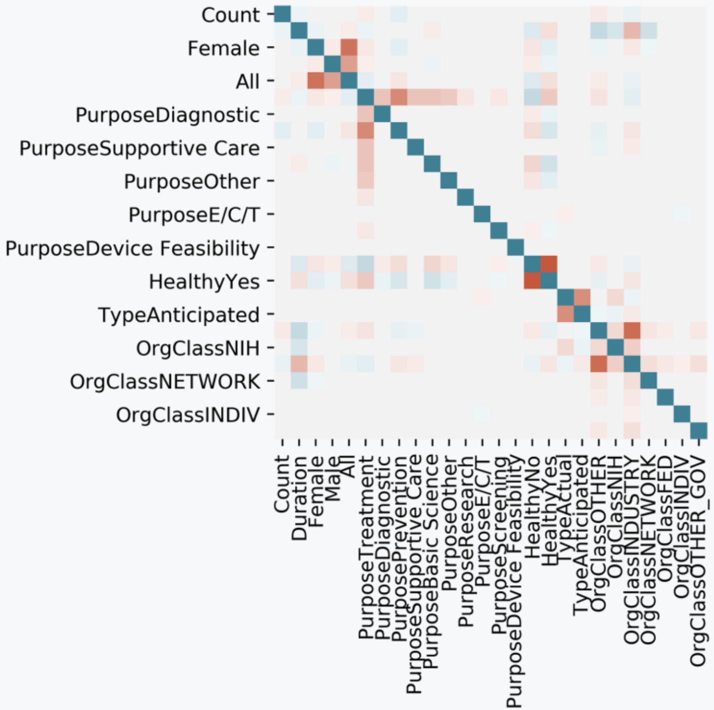


Data Analysis: Distribution of Phases

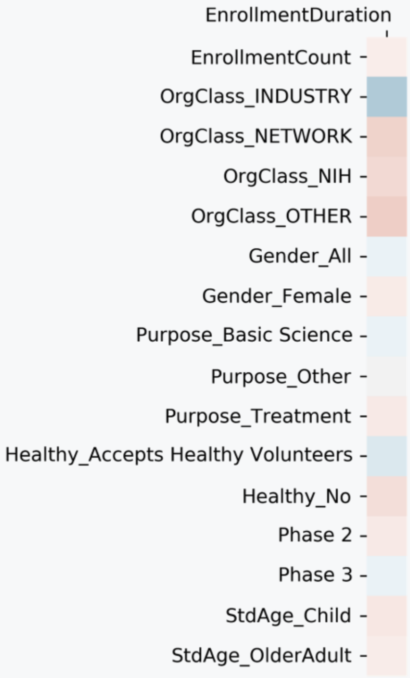


Data Analysis: Correlation

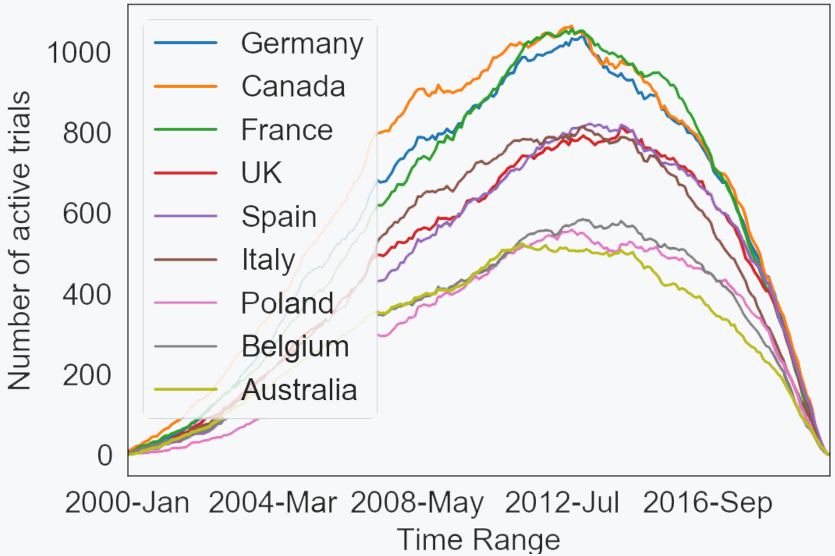
Correlation of features

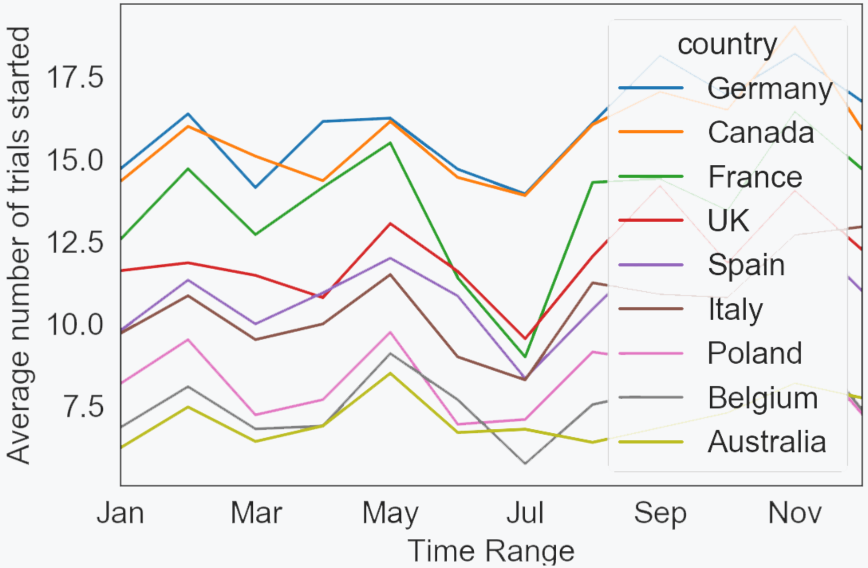


Correlation of features with target variable



Data Analysis: StartYear





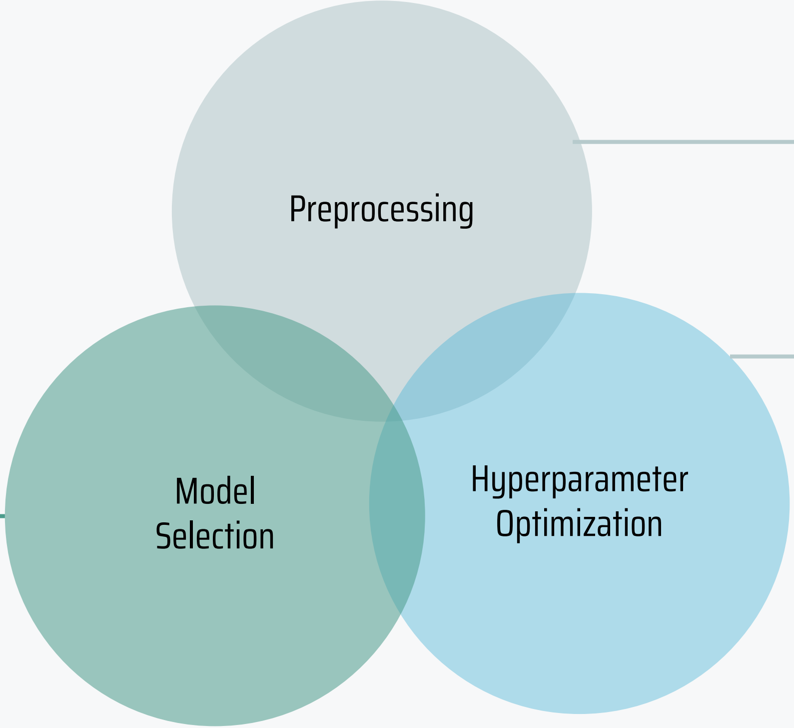
Preprocessing

- Framework
- Transformers and Pipeline
- Custom Transformers Examples
- Transformer Overview
- Improved Correlation

Framework

Make sure all required preprocessing steps are taken in the right order with custom transformers

Easily compare different machine learning algorithms on the same data set



Efficiently optimize the hyperparameters of our models

Missing Values

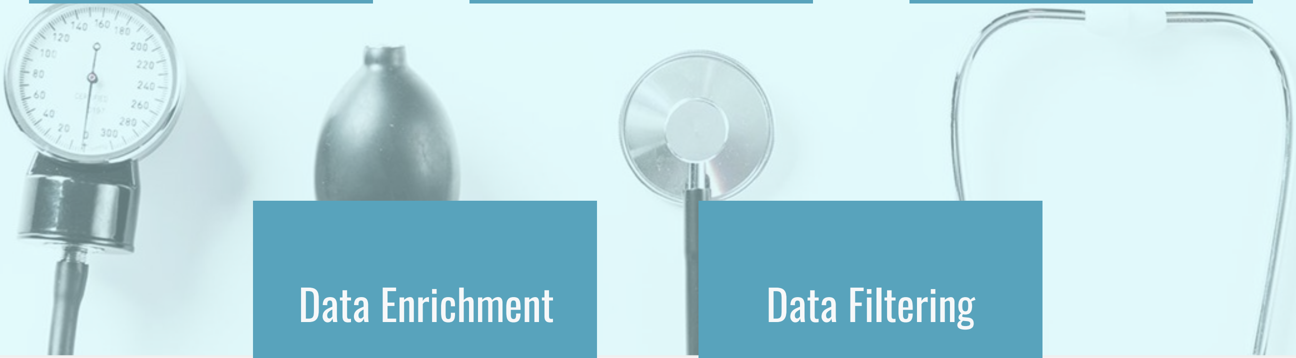
Which features are critical for analysis and how much data has them filled?

Feature Selection

Not all features we retrieved are usable or relevant

Data Types

Extract single values from arrays and save in the correct data type



Data Enrichment

Creating new features (e.g Duration) and collecting additional data

Data Filtering

Dropping trials before 1990, ...

Transformers

- Used for data preparation
- Methods:
  - fit: find parameters from training data (if needed)
  - transform: apply to training or test data
- Possibility to implement custom transformers

Two types of custom transformers:

- Transformers, which apply to a group of similar features
- Specific Transformers for individual features

sklearn.base.TransformerMixin

class `sklearn.base.TransformerMixin` [source]

Mixin class for all transformers in scikit-learn.

Methods

`fit_transform`(self, X[, y])	Fit to data, then transform it.

`__init__`(*self, /, *args, **kwargs*)

Initialize self. See help(type(self)) for accurate signature.

`fit_transform`(*self, X, y=None, **fit_params*) [source]

Fit to data, then transform it.

Fits transformer to X and y with optional parameters fit_params and returns a transformed version of X.

Transformers and Pipeline

Why transformers?

- Can be easily used in Pipelines
- Pipelines makes it easier to understand the workflow and combine different steps
- Allow to apply the same transformation steps (with the same parameters) on training data and new data

```
pipeline = Pipeline([
        ('features', FeatureUnion([
            ('categoricals_single', Pipeline([
                ('extract', FeatureSelector(CAT_SINGLE_FEATS)),
                ('cat_fill', MissingStringsTransformer(strategy='most_frequent')),
                ('single_one_hot_encoding', SingleOneHotEncoder()),
                ('excluder', FeatureExcluder(CAT_SINGLE_FEATS))
            ])),
            ('categoricals_top1', Pipeline([
                ('extract', FeatureSelector(CAT_MULTIPLE_TOP_FEATS1)),
                ('multiple_one_hot_encoding', MultipleTopOneHotEncoder(strategie="top", top=30)),
                ('excluder', FeatureExcluder(CAT_MULTIPLE_TOP_FEATS1))
            ])),
            ('counting_features', Pipeline([
                ('extract', FeatureSelector(TO_COUNT_FEATS)),
                ('counter', DistinctCounter()),
                ('excluder', FeatureExcluder(TO_COUNT_FEATS))
            ])),
            ('textual_features', Pipeline([
                ('extract', FeatureSelector(TEXTUAL_FEATS1)),
                ('keyword_extractor', TextualFeatureTransformer(n_keywords=25)),
                ('excluder', FeatureExcluder(TEXTUAL_FEATS1))
            ]))
        ])),
        ('patients_distribution', PatientsDistributionTransformer()),
        ('location_transformation', LocationDataTransformer(df_dbcountry,
                                    transformer='totalCombine', strategy='weighted', mean='worldwide')),
        ('excluder', FeatureExcluder(ALL_FEATURES))
    ])
```

Example 1: Feature Transformation Groups

- **Categorical features with lists**, e.g. StdAge and CollaboratorClass
- Custom Transformer applying One Hot Encoding

Phase	Std Age	Phase=Phase 1	Phase=Phase 2	Phase=Phase 3	Std Age=Adult	Std Age=Child	Std Age=Older Adult
['Phase 2']	['Child', 'Adult', 'Older Adult']	0	1	0	1	1	1
['Phase 2']	['Adult', 'Older Adult']	0	1	0	1	0	1
['Phase 3']	['Child', 'Adult', 'Older Adult']	0	0	1	1	1	1
['Phase 3']	['Child']	0	0	1	0	1	0

Example 2: Feature Transformation Groups

- **Special form of Multiple One Hot Encoder**, e.g. Condition and ConditionAncestorTerm
  - 1083 different conditionAncestorTerms
  - 9836 different conditions
- Conclusion: too many possible values to one hot encode

Approaches:

i) take Top X Conditions/conditionAncestorTerms

ii) take all Conditions/conditionAncestorTerms which are involved in more than X trials


```
{
  "Breast Cancer": 505,
  "Asthma": 381,
  "Prostate Cancer": 325,
  "Diabetes Mellitus, Type 2": 321,
  "HIV Infections": 276,
  "Multiple Myeloma": 250,
  "Schizophrenia": 248,
  "Pain": 239,
  "Hypertension": 228,
  "Type 2 Diabetes Mellitus": 220,
  "Diabetes": 218,
  "Lung Cancer": 207,
  "Rheumatoid Arthritis": 205,
  "Leukemia": 200,
  "Colorectal Cancer": 196,
  "Lymphoma": 180,
  "Healthy": 163,
  "Alzheimer's Disease": 148,
  "Major Depressive Disorder": 146,
```

Example 3: Textual Feature Transformer

- **Free text features** as list or single value, e.g. InterventionName, OrgFullNa

InterventionName	OrgFullName
[Ganciclovir, Foscarnet]	Johns Hopkins Bloomberg School of Public Health

```
University
Inc.
Center
Cancer
National
Institute
Hospital
Research
Medical
Health
```

- Initialization parameters:
    - NLP processing steps
    - number of keywords
    - stopwords to exclude
- Transformation steps:

    i. Apply NLP cleaning steps (lower case, remove punctuation & special chars, remove stopwords …)

    ii. Apply Tokenization & extract top n most frequent keywords

    iii. Apply one hot encoding for keywords

Example 4: Country Data Transformer

- adds information on the countries involved in the study

2 facilities

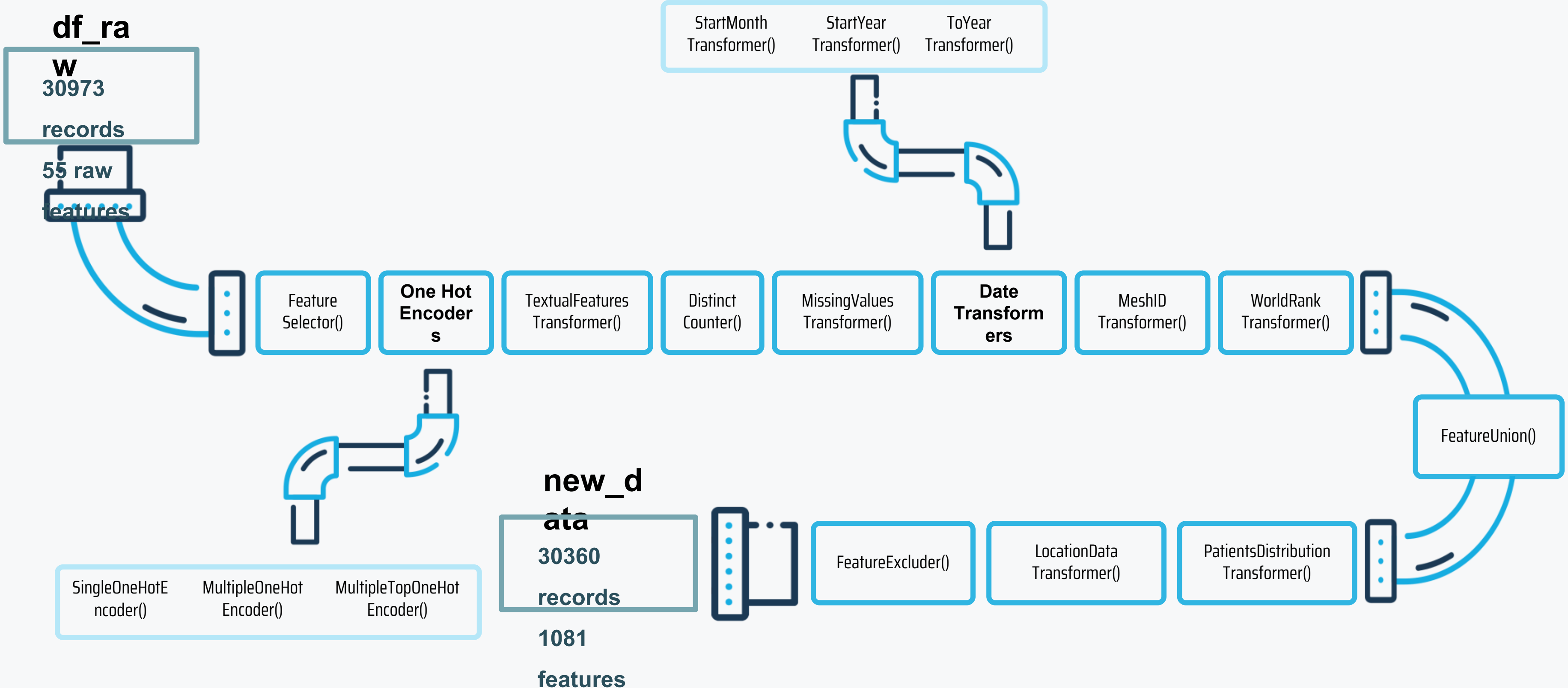
Population: 331 million
Life Expectancy: 78
GDP: 20.54 trillion
Unemployment rate: 3%
Hospital Beds: 2
Health Expenditure: 17
Size: 9.1 million km²
Urban Population: 83

1 facility

Population: 83 million
Life Expectancy: 80
GDP: 3.95 trillion
Unemployment rate: 5%
Hospital Beds: 8
Health Expenditure: 11
Size: 357,386 km²
Urban Population: 76

I. Simple Approach: take average of both countries
II. Weighted Approach: take number of facilities into account → USA counts 2/3

Transformer Overview



Improved Correlation

- After applying the preprocessing steps to the raw features, the correlation to the target variable could be improved
- Features with a correlation of > 20% with target variable:

Before Preprocessing: **3**

After Preprocessing: **18**

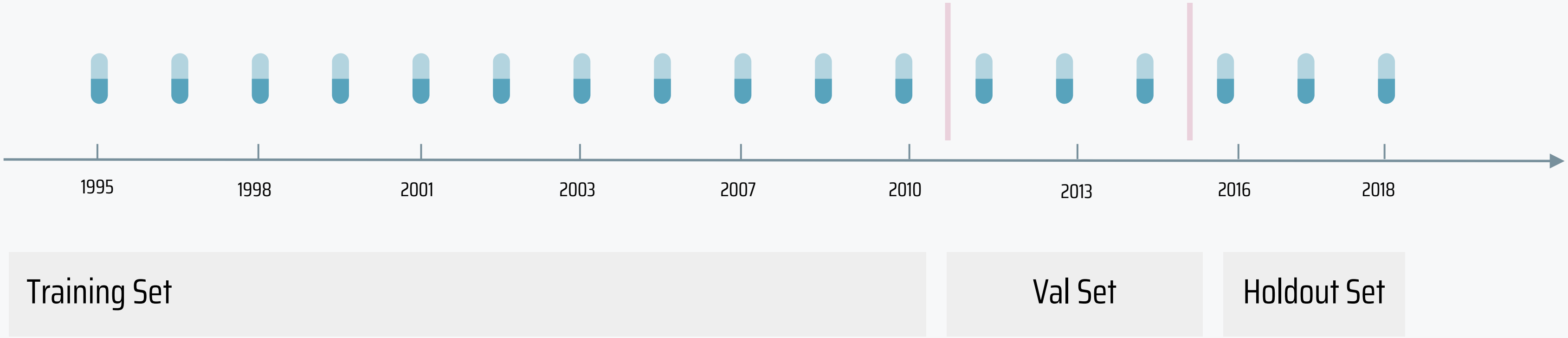
Feature correlation of >0.2 with target variable

Hyperparameter Optimization

- Data splits
- Hyperopt setups

Which split fits?

Time Series Split



- Sort values by StartYear
- Split by time series, training data 72%, validation data 18%, holdout set 10%
- Take “newest” trials as hold-out data

Hyperopt: Distributed Hyperparameter Optimization

Hyperopt: Python library for serial and parallel optimization over all search spaces.

General setup for hyperopt:

- Define space search
- Define objective function
- Select search algorithm: Tree of Parzen Estimators (TPE)
- Setup database to store all searched evaluations

HYPEROPT

Hyperopt: Customized Transformers and Functions

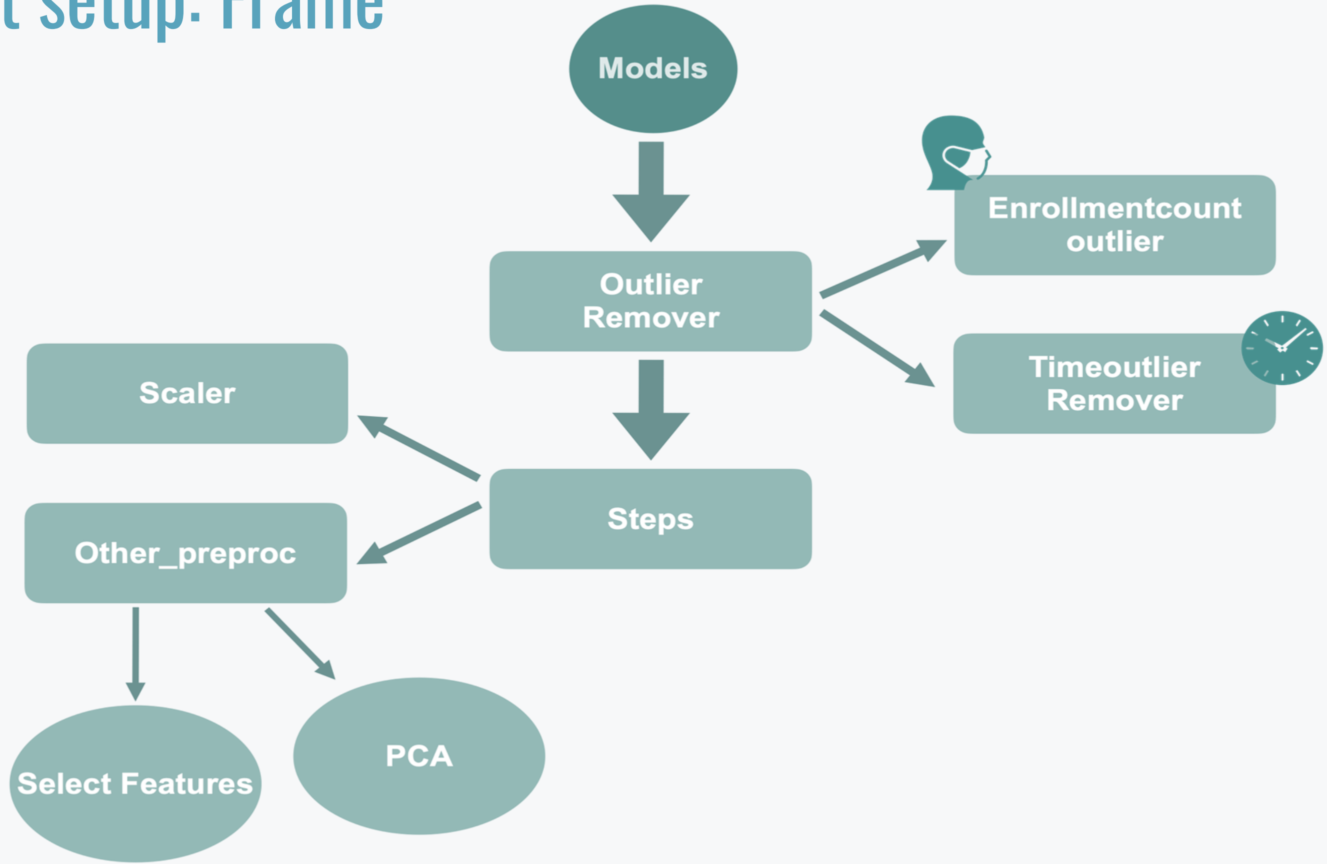
Transformers:

- **TimeOutlierRemover**: removes all records outside a defined time window
- **EnrollmentOutlierRemover**: removes records whose enrollment count lies statistically outside
- **FeatureSelectorTransformer**: identifies the most relevant features with lightGBM algo

Function:

- **Scalers**: Standard Scaler, MinMax Scalers, Normalizer
- **PCA**: create new numpy values and cut down the dimensions

HyperOpt setup: Frame



HyperOpt setup - Define space search

- Regression: Random Forest; XGBoost and LightGBM
- Classification: Naive bayes; SVM; Random Forest; XGBoost and LightGBM

=> Try to use almost all available parameters for each model; with 4 optional transformers

Revised final pipeline: Used condition transformer

Regression

Optional params:
1. Timeoutlier remover
2. Enroll_outlier remover
3. Scaling
4. Dimensionality reduction

Regression modified

General:
Target encoder
LightGBM:
Target encoder or Categorical encoder

Classification

Label encoder beforehand

Optional params

Classification modified

General:
Target encoder
LightGBM:
Target encoder or Categorical encoder

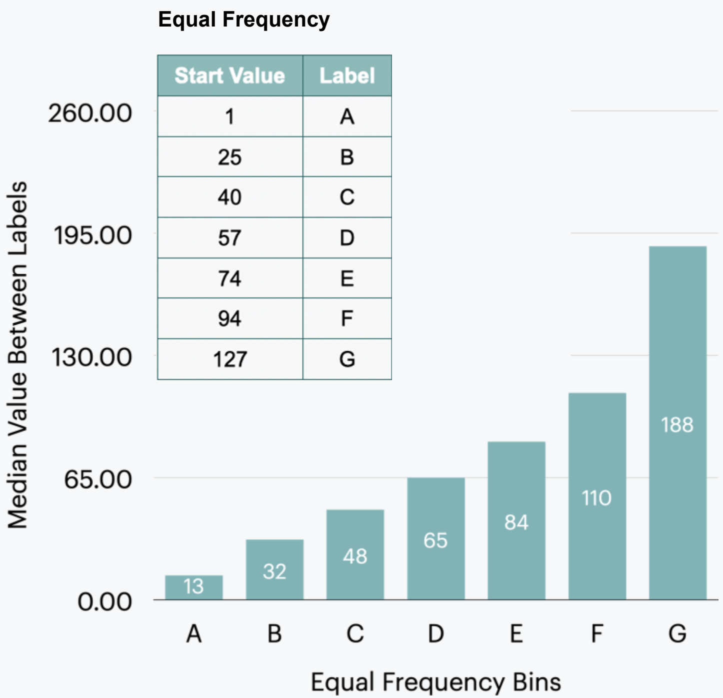
HyperOpt setup - Define objective function

- Objective function
  - Regression: MAE
  - Classification: modified
    - Label options:

      → Equal width

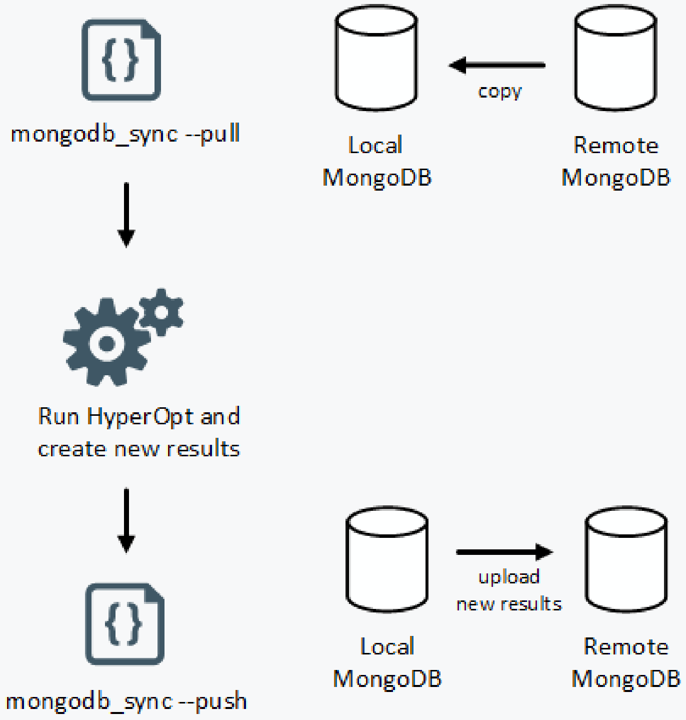
      → Equal frequency
    - Modified objective function

      → Take median values



Start Value	Label
1	A
25	B
40	C
57	D
74	E
94	F
127	G

HyperOpt setup - Parallel search using MongoDB



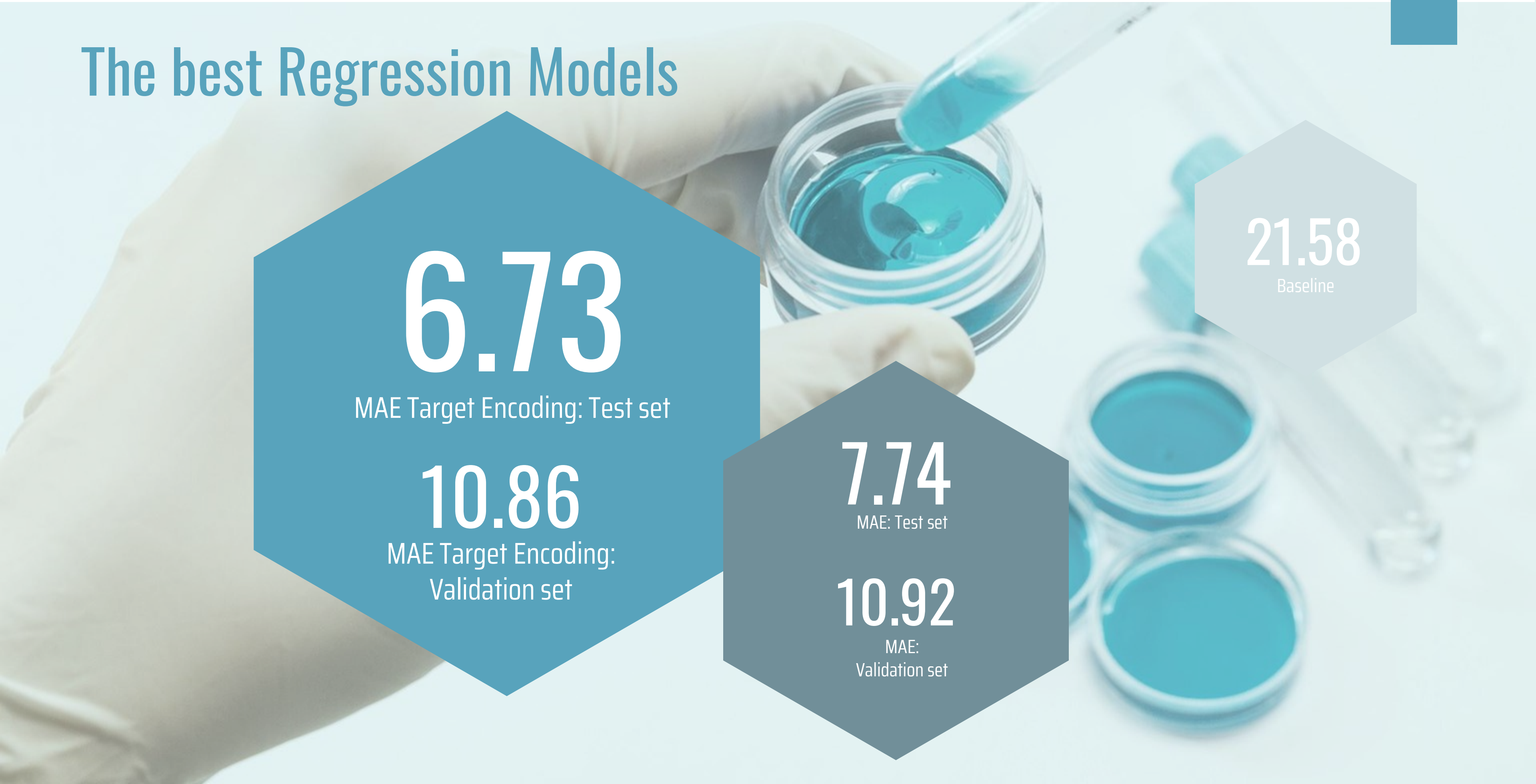
1226

Hyperopt Runs

Results

- Evaluating the models
- Regression vs. Classification
- Feature Importance
- Demo

The best Regression Models



6.73

MAE Target Encoding: Test set

10.86

MAE Target Encoding: Validation set

7.74

MAE: Test set

10.92

MAE: Validation set

21.58

Baseline

The best Classification Models

9.63

MAE: Test set

15.92

MAE for
Validation set

10.24

MAE for Target Encoding:Test set

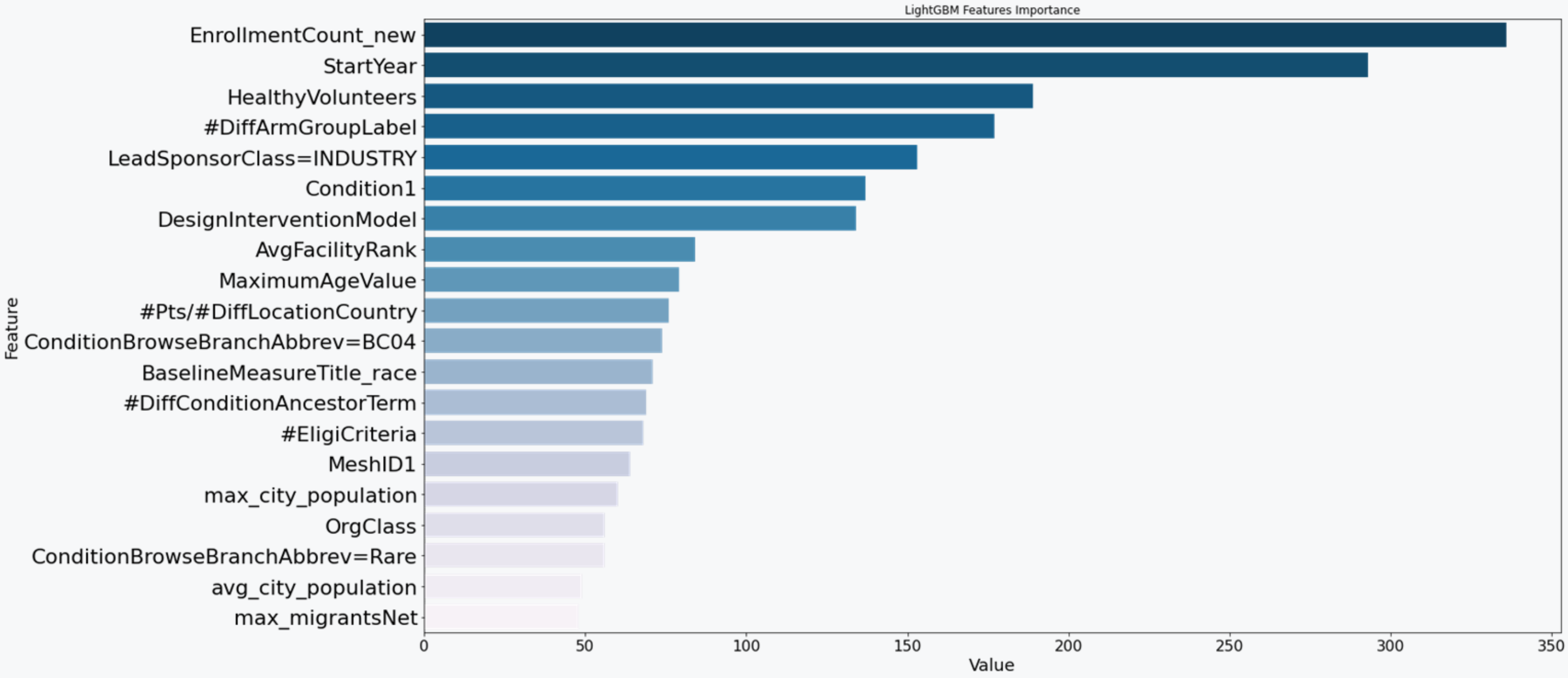
13.26

MAE for Target Encoding:
Validation set

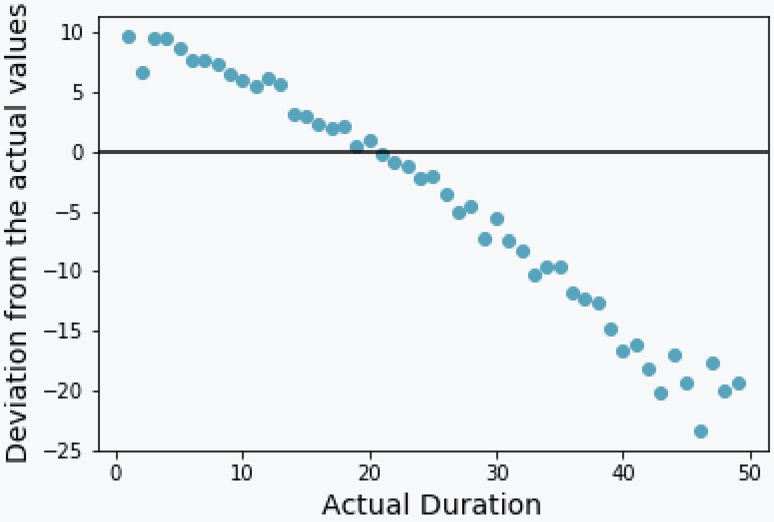
27.42

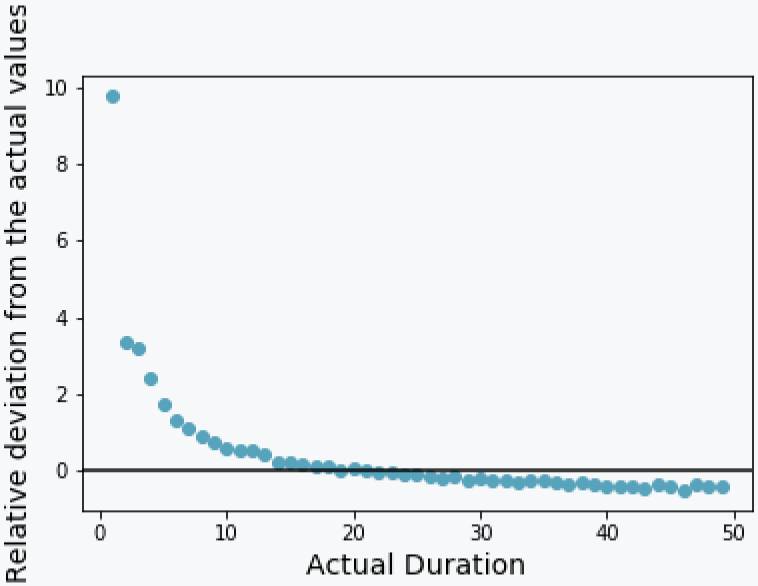
Baseline

Feature Importance

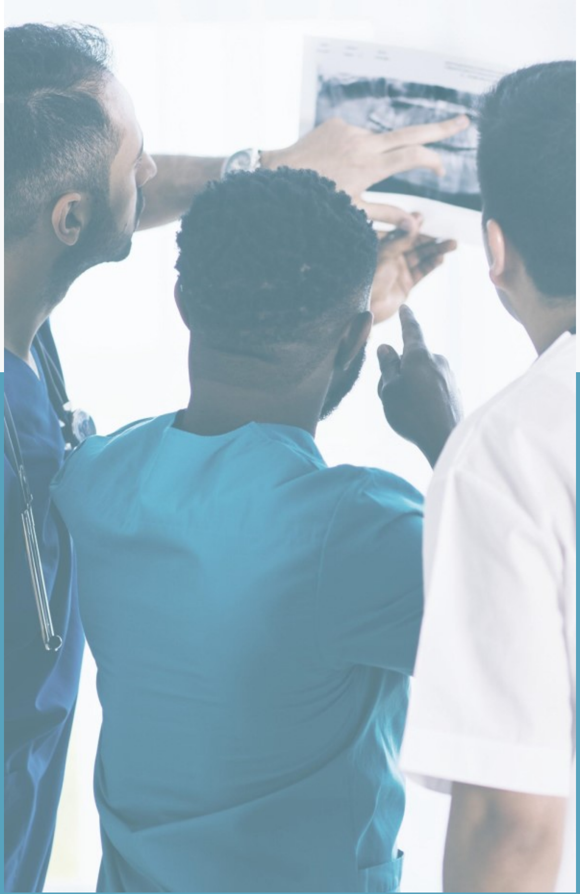


Results Analysis





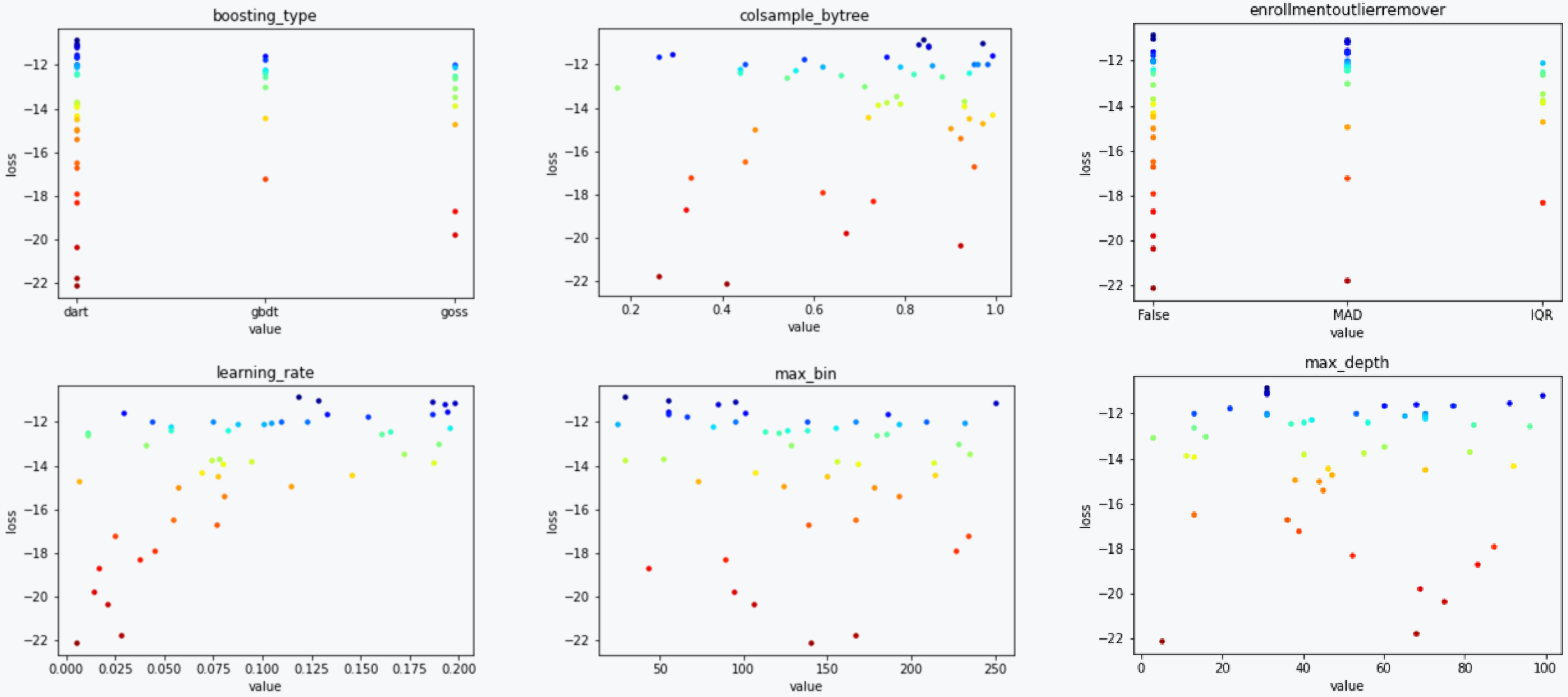
Demo



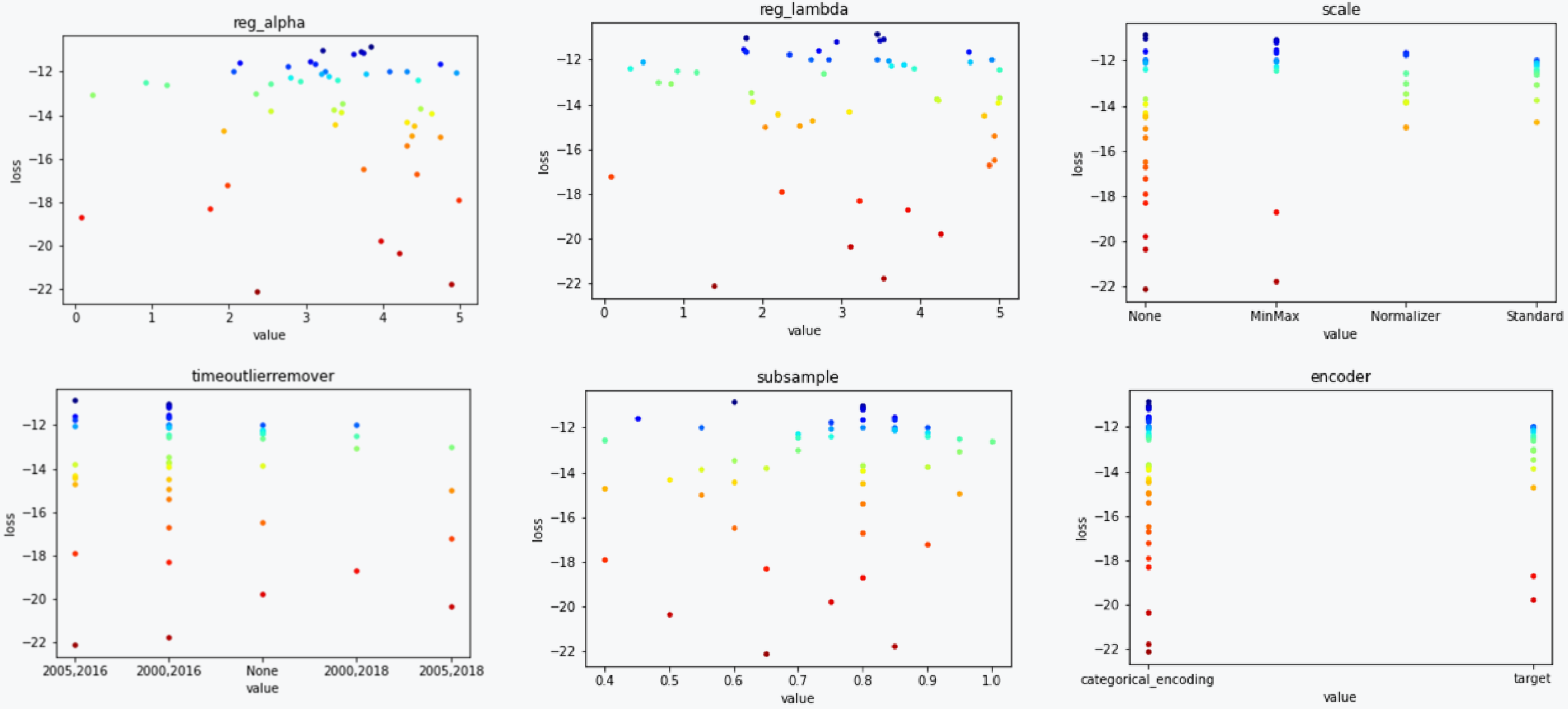
Insights and Outlook

- Insights of parameters
- Insights during the project
- Suggestions for further research

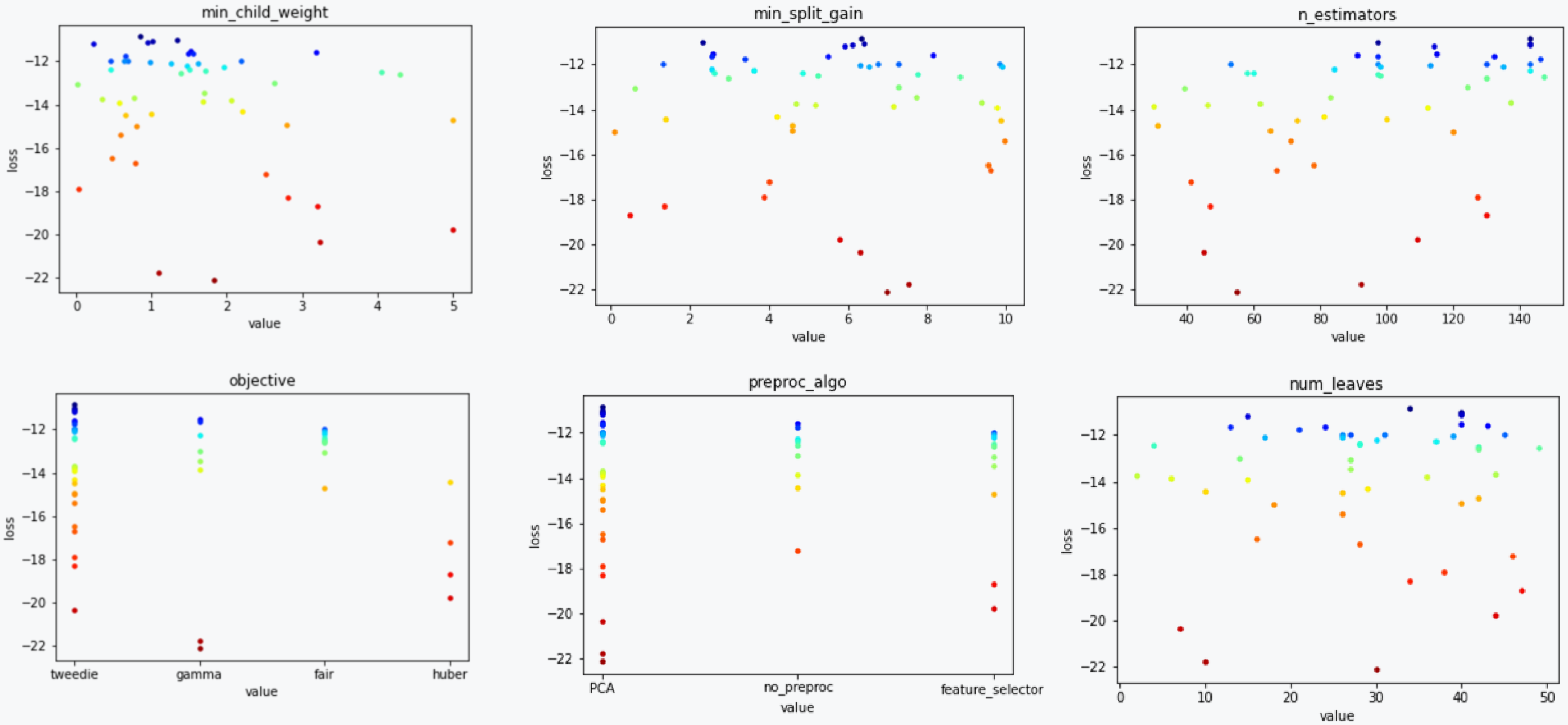
HyperOpt Insights: LightGBM modified regression



HyperOpt Insights: LightGBM modified regression



HyperOpt Insights: LightGBM modified regression



Suggestions for Future Improvements

- Use deeper NLP methods to extract the enrollment duration and Screening Numbers from the free text
- Further enrich the data: add more hospital data, data on other facilities or condition related informations
- Use trials data from other clinical trials databases (e.g. chinese database with ~6900 completed studies and similar fields)
- Try models we did not yet consider in model selection (e.g. a Neural Network)

Conclusion

- The prediction capability of the model is limited to a half year on average

→ The data might not fit the data generating process: The key drivers of enrollment may not be captured in our data

- The model confirms that enrollment count is the most important feature
- Try to keep dimensionality low
- Extensive data enrichment pays off
- Also try naive approaches, in our case they worked!

Thank you!