Machine Learning Playground App

Content and target of the project

- development of an interactive platform for learning and understanding the functionality of different machine learning algorithms
- realisation in R with shiny
- using the machine learning package “mlr” as backend for the algorithms (165 different algorithms for 5 different task types)
- task types: classification, regression, cluster, multilabel, survival
- using other, more specialized platforms as inspiration

Problems of existing platforms

- too specialized (only NN or xgboost)
- no comparison between different learners (only hyperparameters)
- cluttered / overwhelming at first sight
- ugly
- lack of tryout datasets + bad customization of these
- lack of intro (video tutorial for example)

Examples:

http://playground.tensorflow.org/

http://arogozhnikov.github.io/2016/07/05/gradient_boosting_playground.html

Philosophy

- using as few hardcoded backend elements as possible (the more mlr the better)
- no breaking consequences by further development in mlr (additions/removals of learners, changes to hyperparameters)
- appealing and clean design
- never show the user more than he wants
- assume the user has no idea what he is doing (at first)
- high customization and flexibility for higher skilled users
- minimalism > convenience > aesthetics

Tools

- backend: mlr package https://mlr.mlr-org.com/

  standardized interface for all different kinds of machine learning algorithms with easy, but still high customization abilities

- frontend: css template provided by Templated https://templated.co/

  (license: Creative Commons Attributions 3.0, which means free to use as long as visible credits are given)

- Plots: Plotly https://plot.ly/

Teamwork

- difficult due to lots of work on the (heavily functional) UI with low backend effort (everything already exists in mlr)

- current split:

  Sebastian: frontend + backend

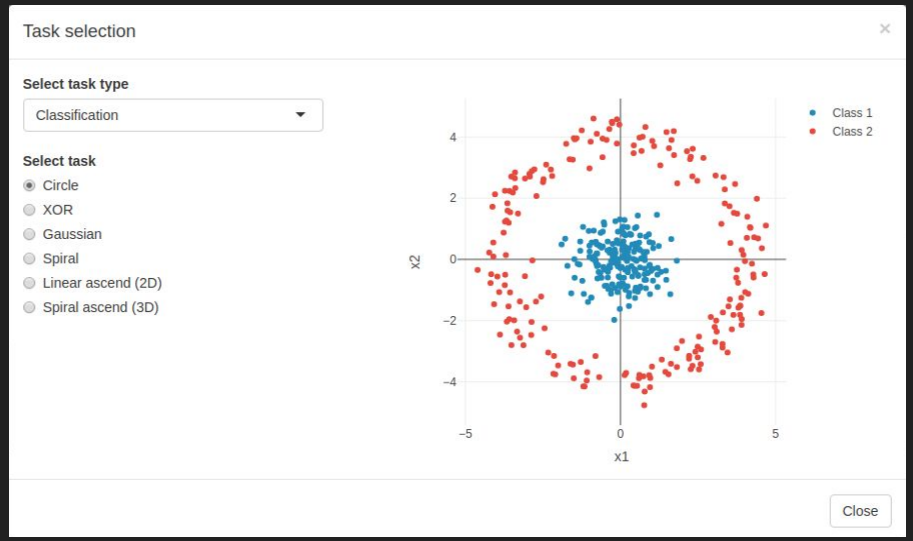
  Yuhao: datasets + plots

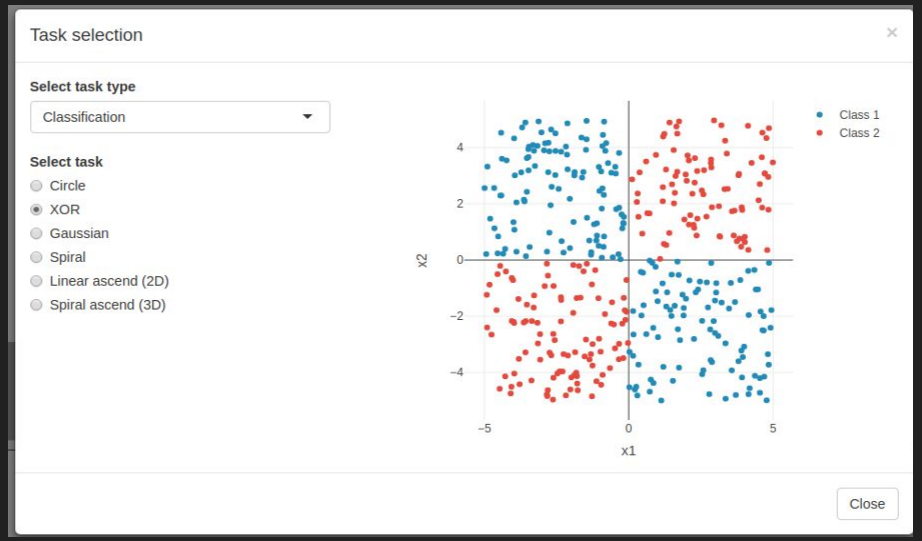
Done so far

(Run the app twice)

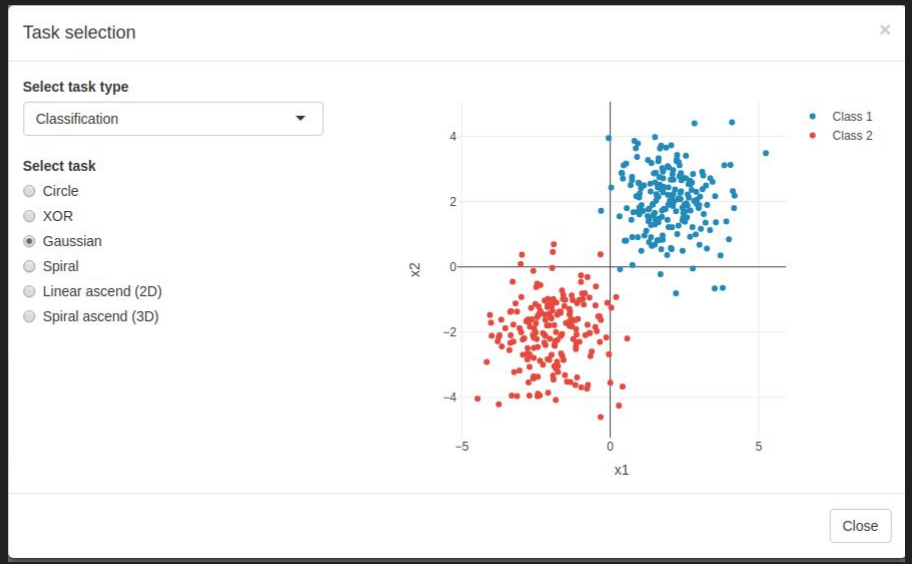
+ datasets on the following slides

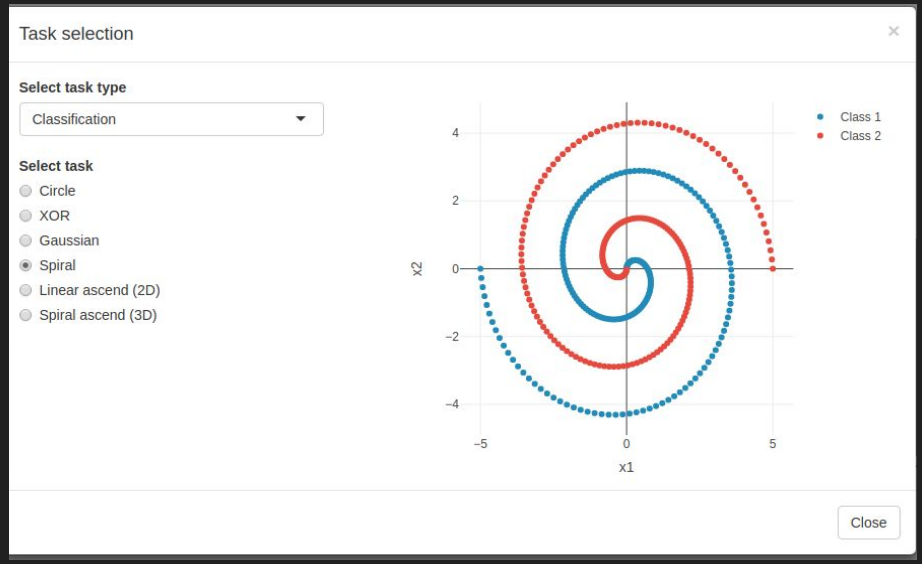
Data sets(Circle,XOR)



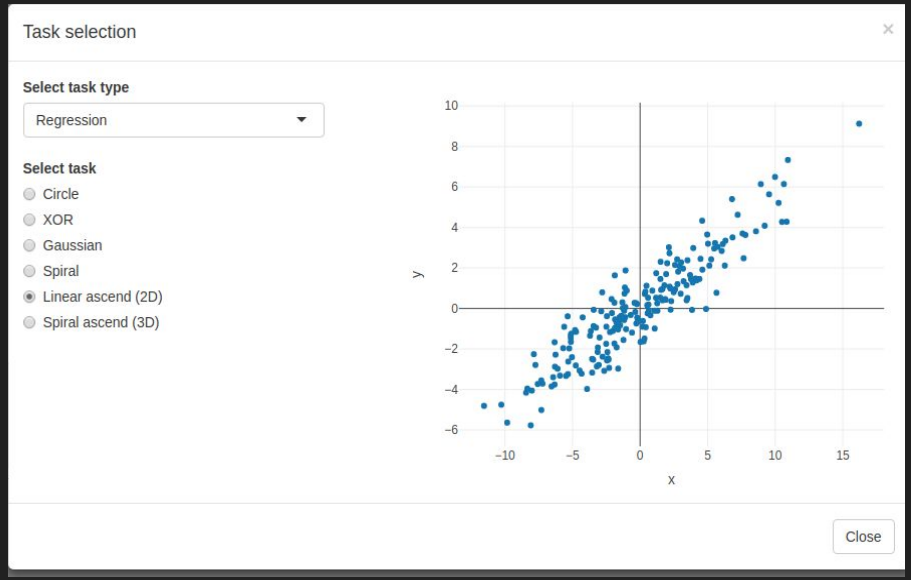


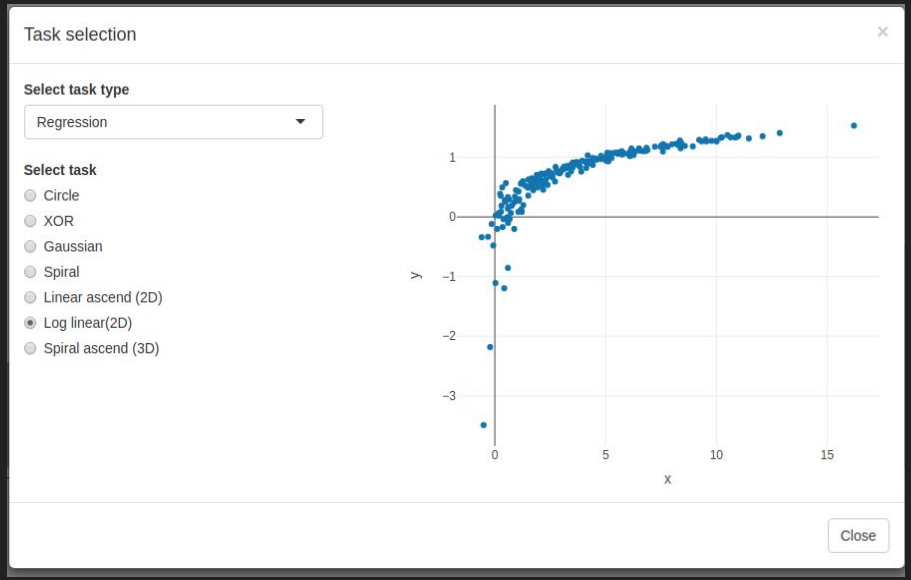
Data sets(Gaussian,Spiral)



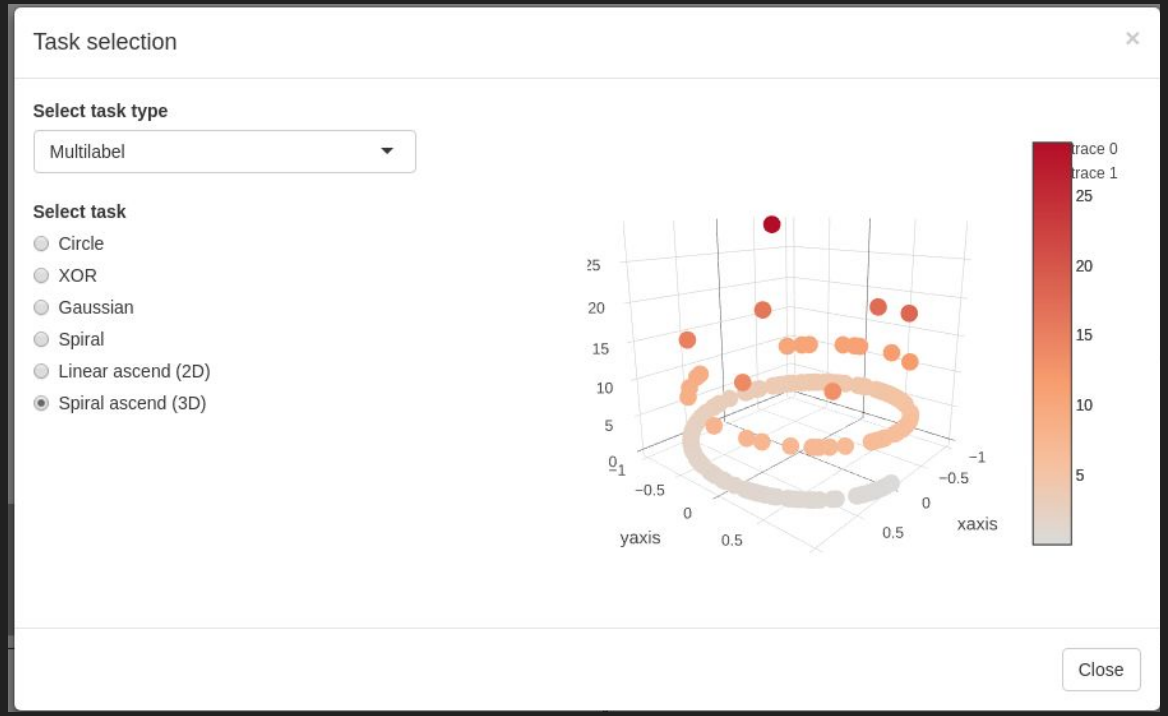


Data sets(Linear ascend (2D))





Data sets(Spiral ascend (3D))



Problems

1. Simulate special 3D data sets (sphere)
2. Soft classification (probabilities instead of classes)
3. Missing UX feedback of low, medium and high skilled users

Todo

- adjustable and !!not-hardcoded!! hyperparameters
- adjustable datasets
- more task types
- better plots
- parallel evaluation of different learners?
- intro video?
- frontend: translating shiny wrappers to low level html-like code (but still in shiny)
- css fine-tuning
- error handling
- testing