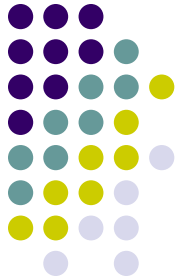
WOLF: Automated Machine Learning WOrkfLow Optimization Framework

Department of Electrical Engineering and Computer Science
University of Kansas

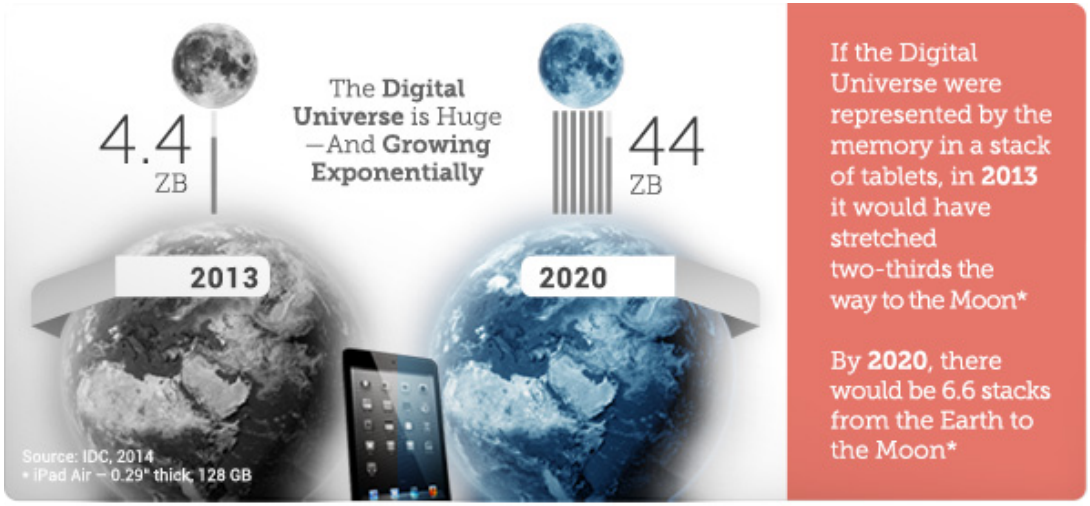



Outline



- Data Science/AI quick overview
- Automation in Model Construction
- WOLF: Automated Machine Learning WOrkfLow Optimization Framework demonstration

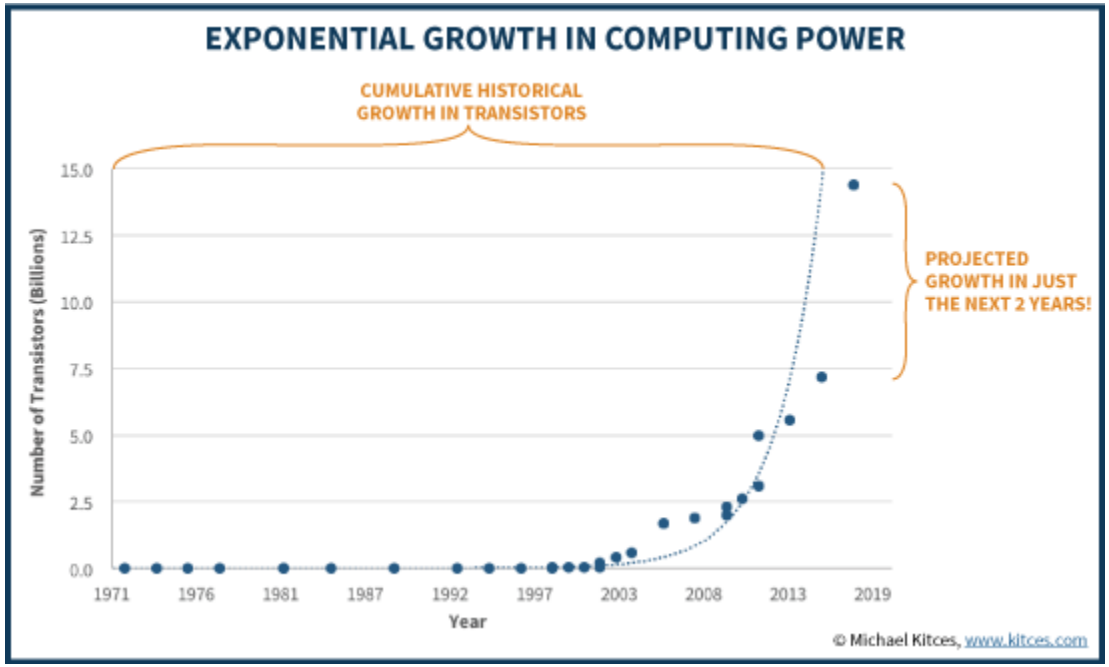
Growth of Data

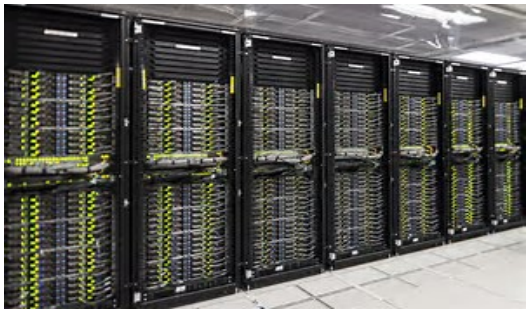


From dawn of civilization until 2003, human totally created about 5 Exabytes of data

[1] Published by IDC, 2014

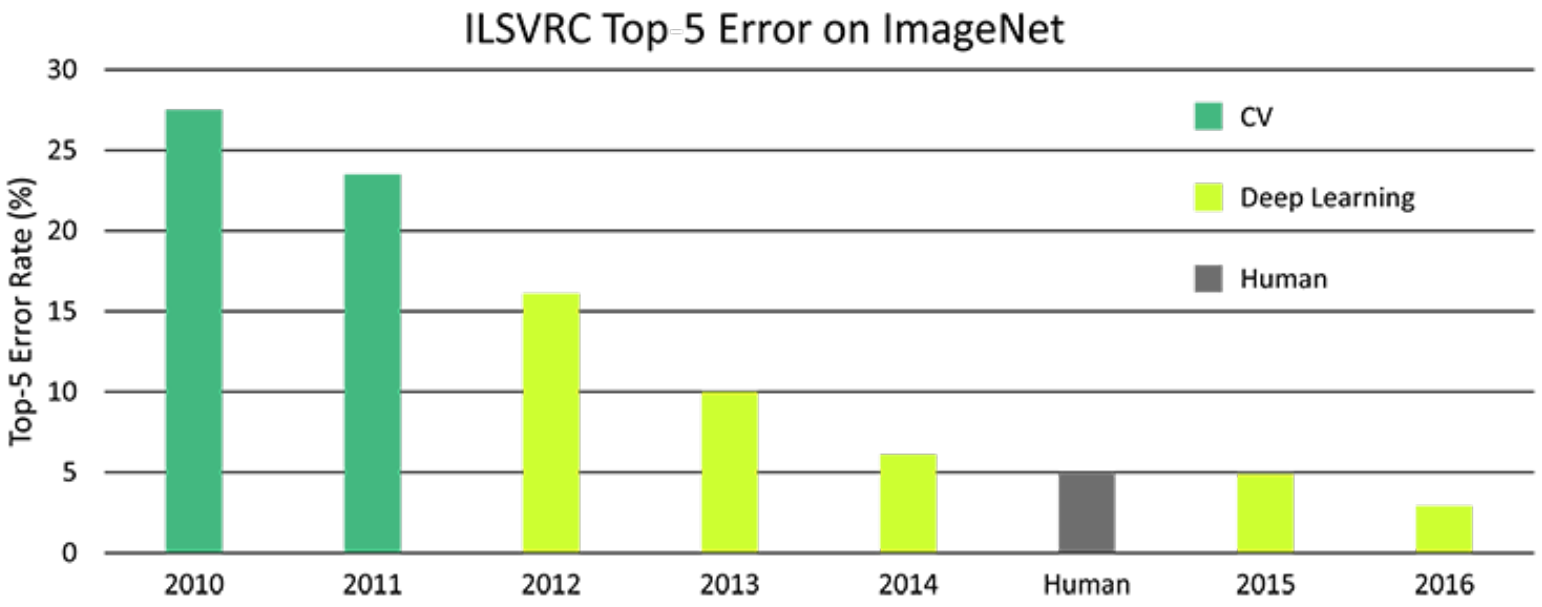
Growth of Computing Power





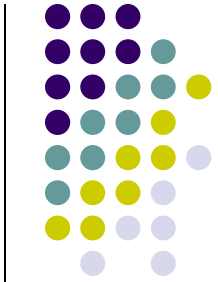
[1] Published by Wikipedia

Improvement in Machine Learning Algorithms



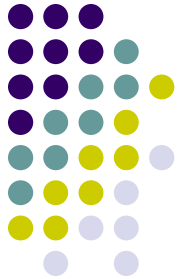
[1] ImageNet Large Scale Visual Recognition challenge (ILSVRC)

Rapid Growth of Data Science/Artificial Intelligence



- Few years ago, task of making self driving cars was years away.
- In the cognitive realm, AI algorithms beating chess champion two decades ago, but now the world champion in Go, which is much more complicated.
- The ability of machines to read lips is better than that of deaf people.

The Fourth Industrial Revolution



- The new paradigm involves using the cloud, big data, advanced machine learning techniques and algorithms, to create the integrated, automated, fast and highly customizable production system of the future.
- AI is the “new electricity”. AI might transform industry, business, technology as electricity did 100 years ago
- “The Fourth Industrial Revolution: what it means, how to respond”, World Economic Forum, 2016 (https://www.weforum.org/agenda/2016/01/the-fourth-industrial-revolution-what-it-means-and-how-to-respond/)

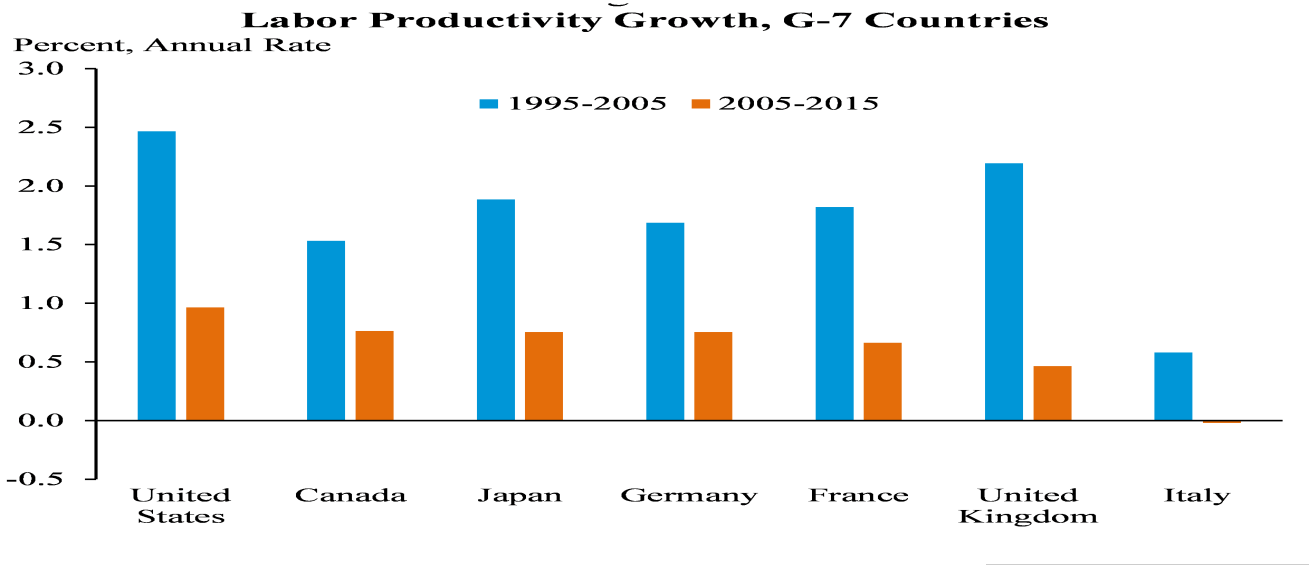
Startup companies in DS/AI

- There are around 1,731 companies in 13 categories across 69 countries, with a total of $13 Billion funding.[1]

[1] Listed by Venture Scanner (startup research firm).

Decline in Productivity

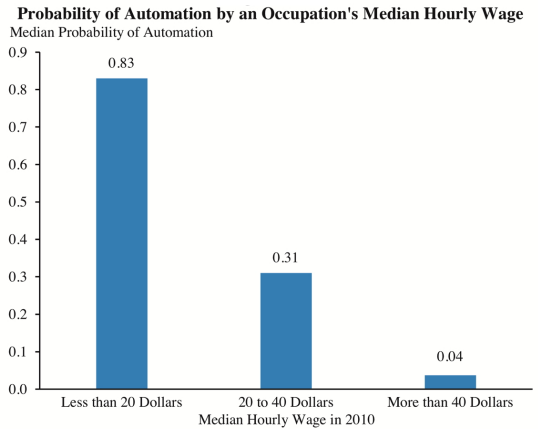
- In the last decade, despite technology's positive push, measured productivity growth has slowed in 30 of the 31 advanced economies

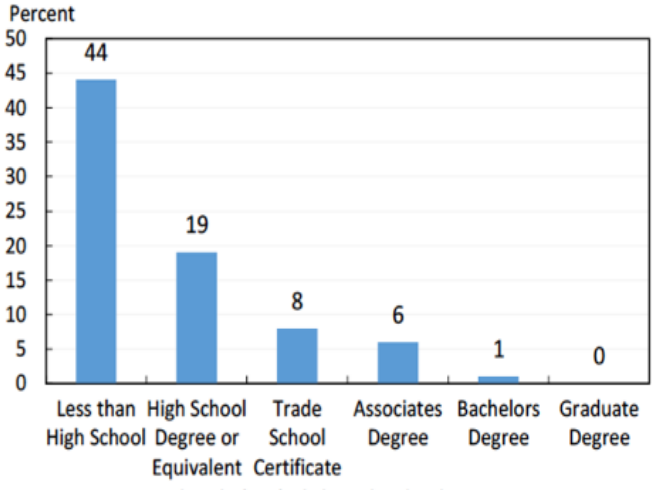


[1] Conference Board, Total Economy Database; CEA calculation.

Jobs more susceptible to AI/Data Science

- Effects of AI on the labor market will continue the trend toward skill-biased change, but researchers differ on possible magnitude of this effect.

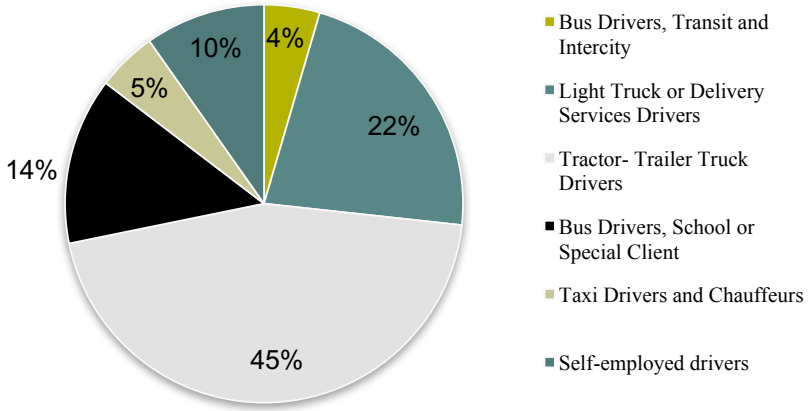




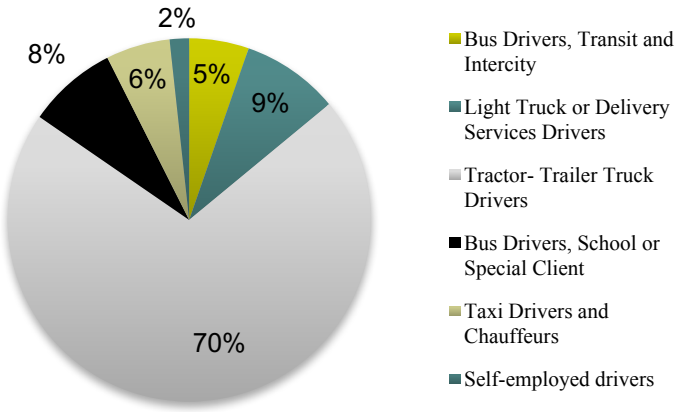
[1] Bureau of Labor Statistics; Fray and Osborne (2013); CEA calculations.
[2] Arntz Gregory, and Zierahn [2016] calculations based on PIAAC 2012.

Case Study: Automated Vehicles

Total Jobs=3,723,930

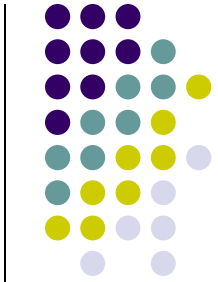


Min. Jobs Threatened=2,196,940



[1] Source: CPS ASEC

Cases Studies of AI/DS Products



- AI/DS products are penetrating into everyone's life
  - Transportation
  - City management
  - Education
  - Health
  - Finance
  - Security

Self-Driving Cars: Government

In response to a petition from Google, the National Highway Transportation and Safety Administration decided that the AI system that controls Google's self-driving car can be considered a driver under federal law.[1]

[1] http://fortune.com/2016/02/10/google-self-driving-cars-artificial-intelligence/

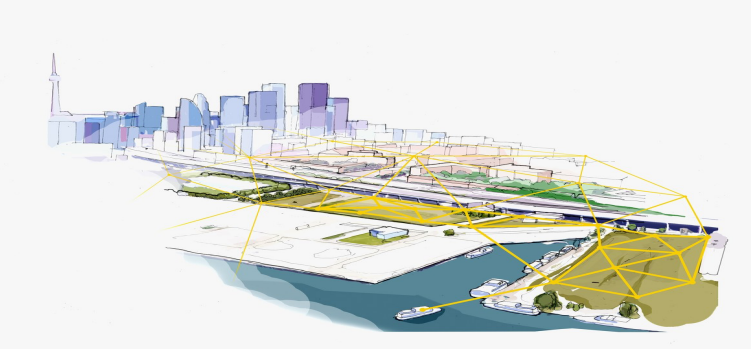
Self-Driving Car: Consumers

- Over 35,000 Tesla owners bought ‘full self-driving’ feature despite still being unavailable. [1]

[1] https://electrek.co/2017/10/10/tesla-autopilot-owners-bought-fully-self-driving-capability/

Reinvent the city: Quayside

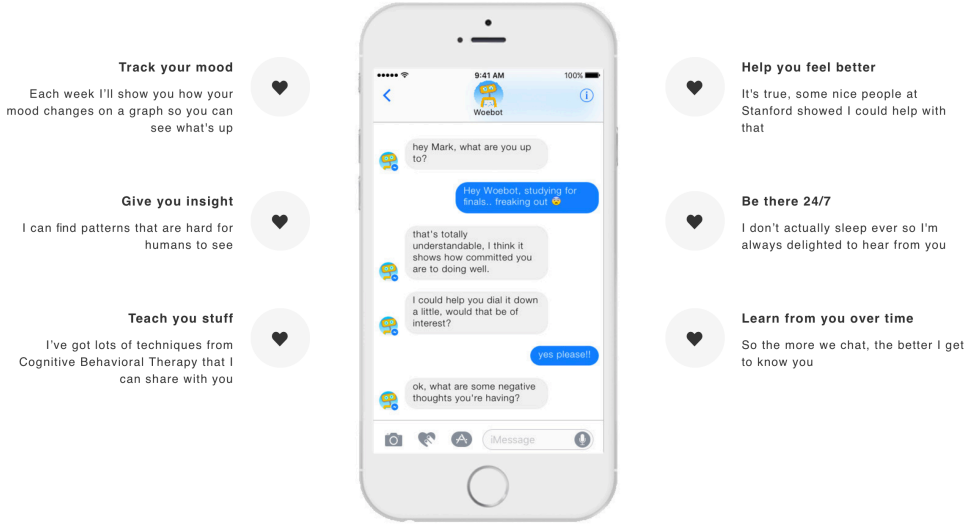
- Google is working with the city of Toronto to build a high tech town. All sorts of sensors will be embedded everywhere possible to collect information about traffic flow, noise levels, air quality, energy usage, travel patterns, and waste output. [1]



[1] https://www.wired.com/story/google-sidewalk-labs-toronto-quayside/

Woebot: A Chatbot That Can Help with Depression

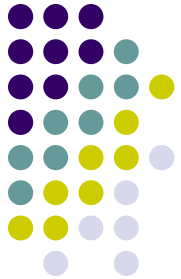
- Woebot was found to reduce the symptoms of depression in students over the course of two weeks.[1]



https://woebot.iohttps://woebot.io

[1] https://www.technologyreview.com/s/609142/andrew-ng-has-a-chatbot-that-can-help-with-depression/

AI Teaching Assistant



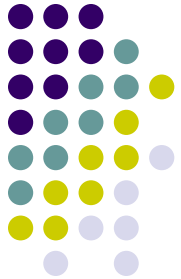
- Jill Watson, an AI teaching assistant who answered students' questions about assignments and due dates, has been used by professor Ashok Goel for his AI class at the Georgia Institute of Technology for a semester. But few realized Ms. Watson was actually a computer. [1]
- It is supported by IBM's Watson analytics system.

[1] https://www.digitaltrends.com/cool-tech/watson-georgia-tech-ta/

AI in Financial Decisions

Image source: https://www.pwc.com/us/en/financial-services/research-institute/artificial-intelligence.html

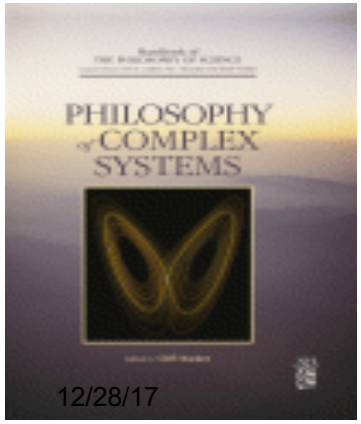
Pressing Needs in DS/AI

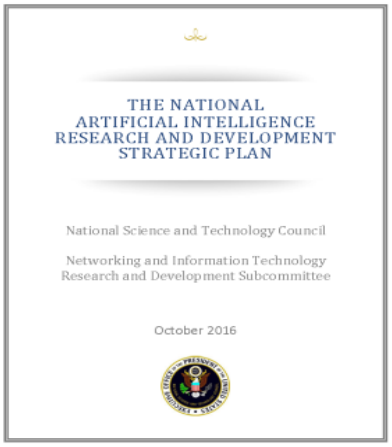


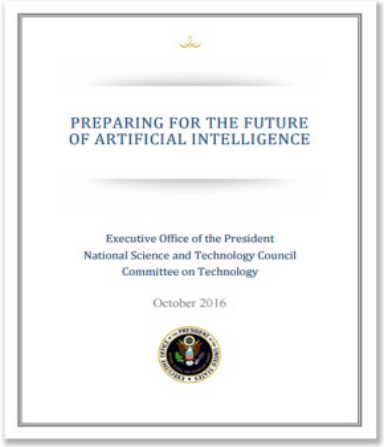
- Ensure DS/AI is safe:
  - Safe AI: Transparency, interpretability, accountability, secure and trustworthy
  - AI Model Commons: Automated ML model construction and lifecycle management

Transparent and Interpretable Machine Learning

- If we can not explain a system, we are not able to know when it fails and the consequence of failures.
- Two White House Reports
  - National AI R&D Strategy: a key research challenge is increasing the "explainability" or "transparency" of AI.
  - Preparing Future of AI: AI-enabled systems must be open, transparent, and understandable.







Credits:
CCC

Safe Machine Learning In Our Society

- Sackler Frontier of Machine Learning
  - Richard Berk (UPenn) argued that there is a unavoidable tradeoff between accuracy, transparency, and fairness in crime justice
  - Karen Yeung (Cambridge) talked about algorithmic regulation and intelligent enforcement

    http://www.lse.ac.uk/accounting/CARR/pdf/DPs/CARR_DP84-Martin-Lodge.pdf#page=54:

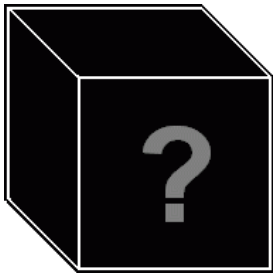
EU Data Protection Regulation

- In April 2016, the European Parliament adopted a set of comprehensive regulations, GDPR.
  - Articles 13 and 14 state that, when profiling takes place, a data subject has the right to "meaningful information about the logic involved." (Goodman et al., 2016)

Transparency in Judicial Decision Making

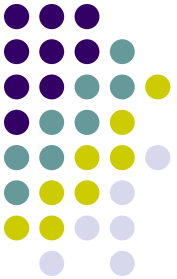
- A Wisconsin man, Eric L. Loomis, who was sentenced to six years in prison based in part on a private company's proprietary software. (The New York Times, May 1, 2017)

My right to due process was violated since I was unable to inspect or challenge the algorithm.



Compas Score:
High risk of violence
High risk of recidivism
High pretrial risk

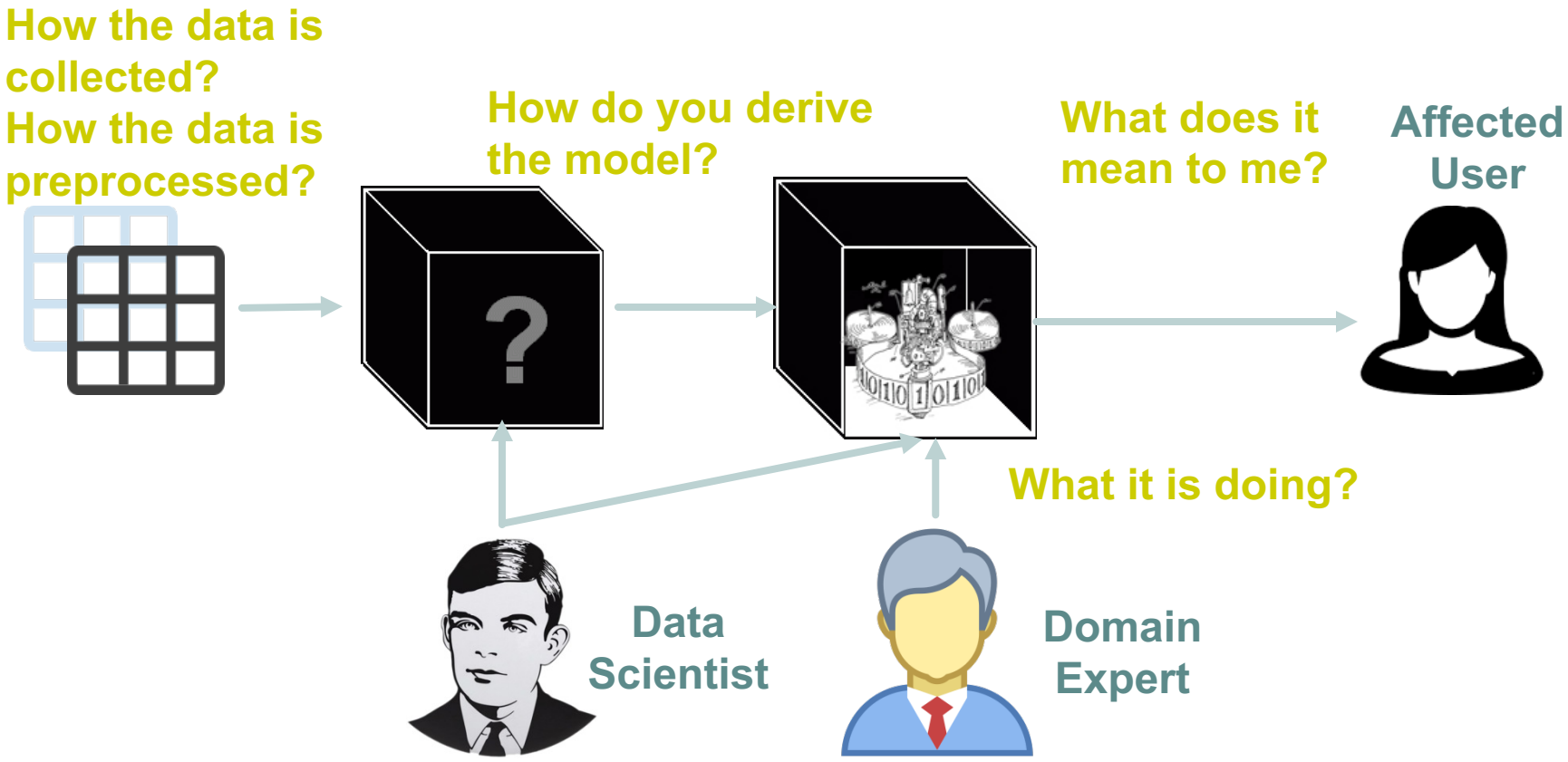
Properties of Predictive Data Analytics Transparency and Interpretability



- It is subjective and context dependent
- It is Not a monolithic concept
- It can be defined at different levels
- Difference in transparency and interpretability
  - The Mythos of Model Interpretability(Lipton, 2016)
    - Transparency: How does the model work?
    - Interpretability: What else can the model tell me?
- Transparency means interpretability and vice versa

Li, Lan, Huan: KDD'17 (tutorial)
Li & Huan, KDD'17 (Constructivism Learning)

Modeling Transparency



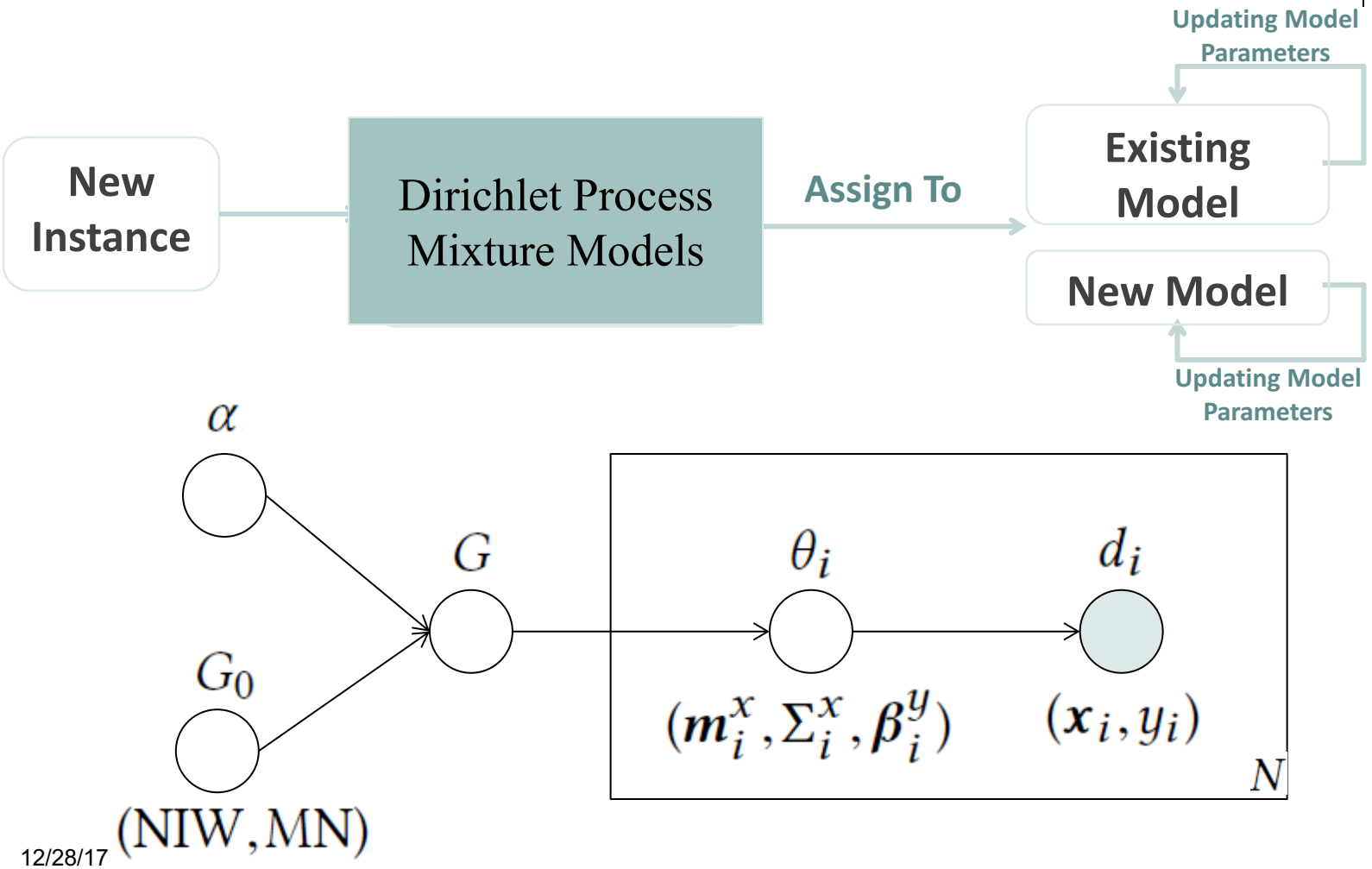
Human Constructivism Learning

- HCL is a philosophical view about how human learns
- It is an instance of epistemology
- HCL significantly influences modern education system

Two Key Processes in HCL

- Human learns by sequentially interacting with external world
  - Assimilation: incorporates new experience into an existing knowledge framework without changing that framework.
  - Accommodation: changes the internal representation of the external world according to the new experience

HCL and Dirichlet Process Mixture models (DPM)



Performance Evaluation

- Average AUC with Synthetic Data Sets

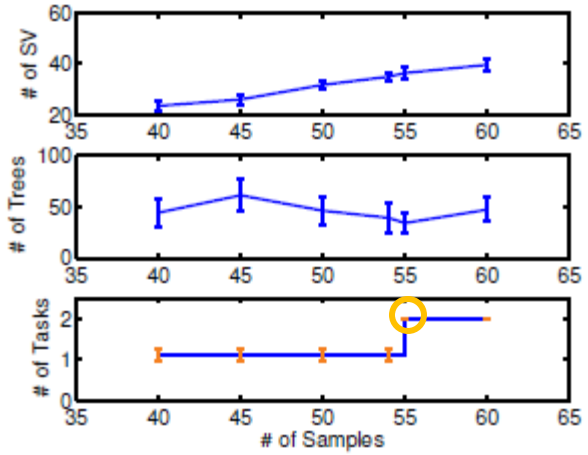
	SVM	RF	EDPMM	sDPMCM	sDPMCM-s
SDS1	0.812	0.801	0.860	0.847	0.856
SDS2	0.787	0.748	0.806*	0.788	0.798
SDS3	0.814	0.789	0.823	0.813	0.822
SDS4	0.823	0.814	0839	0.838	0.852*

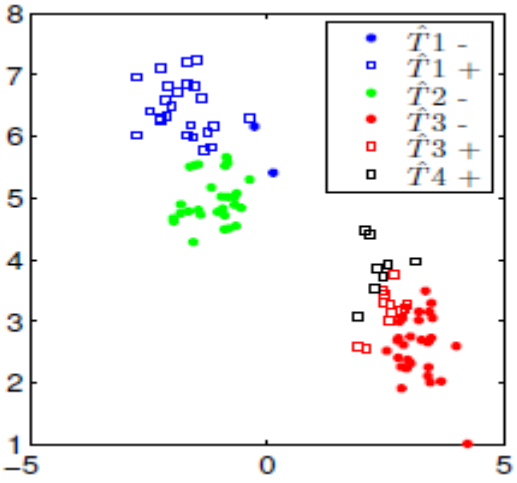
- Average AUC with Real-world Data Sets

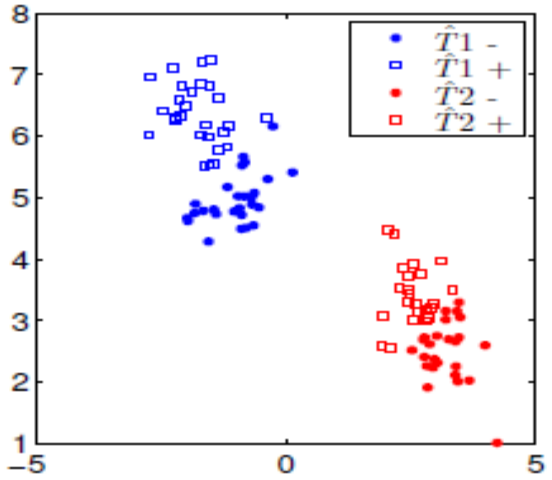
	SVM	RF	EDPMM	sDPMCM	sDPMCM-s
WebKB	0.873	0.896	0.894	0.897	0.910*
School	0.718	0.718	0.676	0.715	0.717
LandMine	0.676	0.670	0.552	0.670	0.687*

*: 5% statistical significance with student t-test

Interpretability Evaluation with Non-Overlapping Tasks

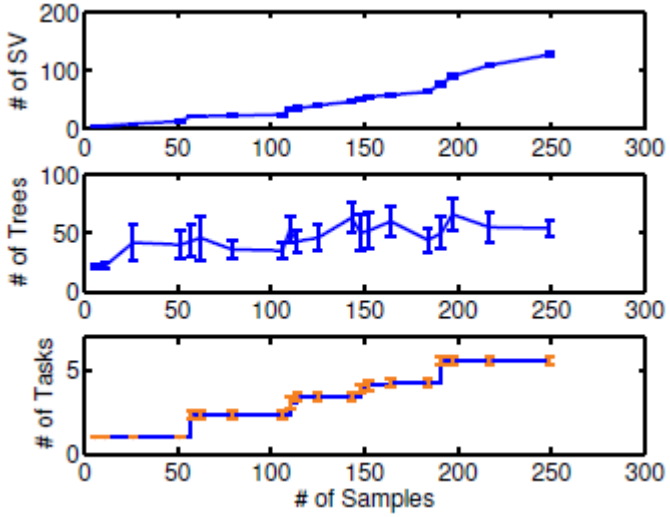


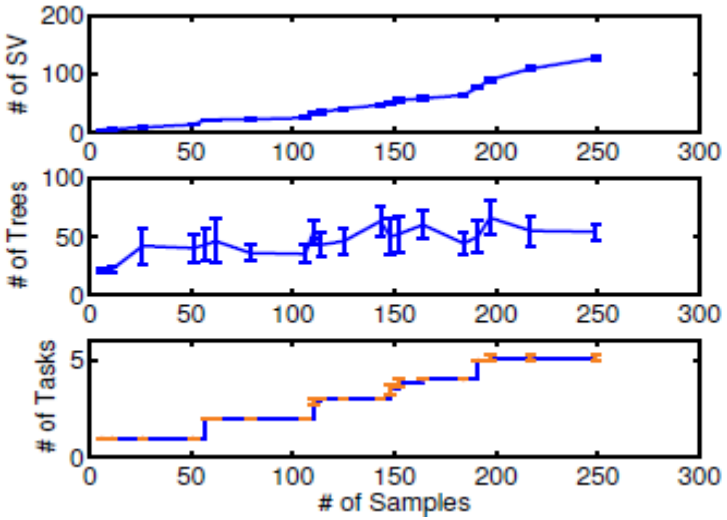


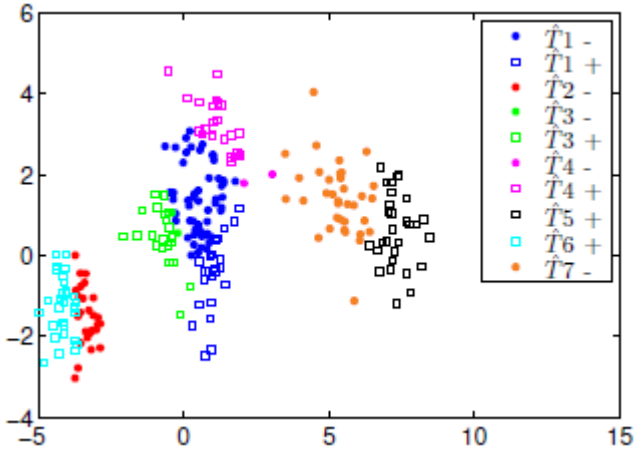


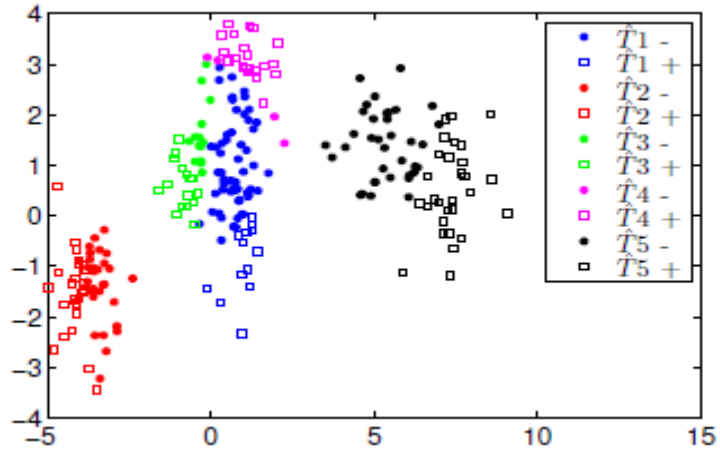
First Instance of the Second Task

Interpretability with Multiple Tasks

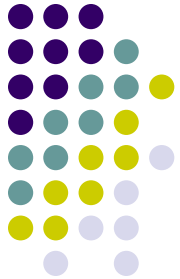


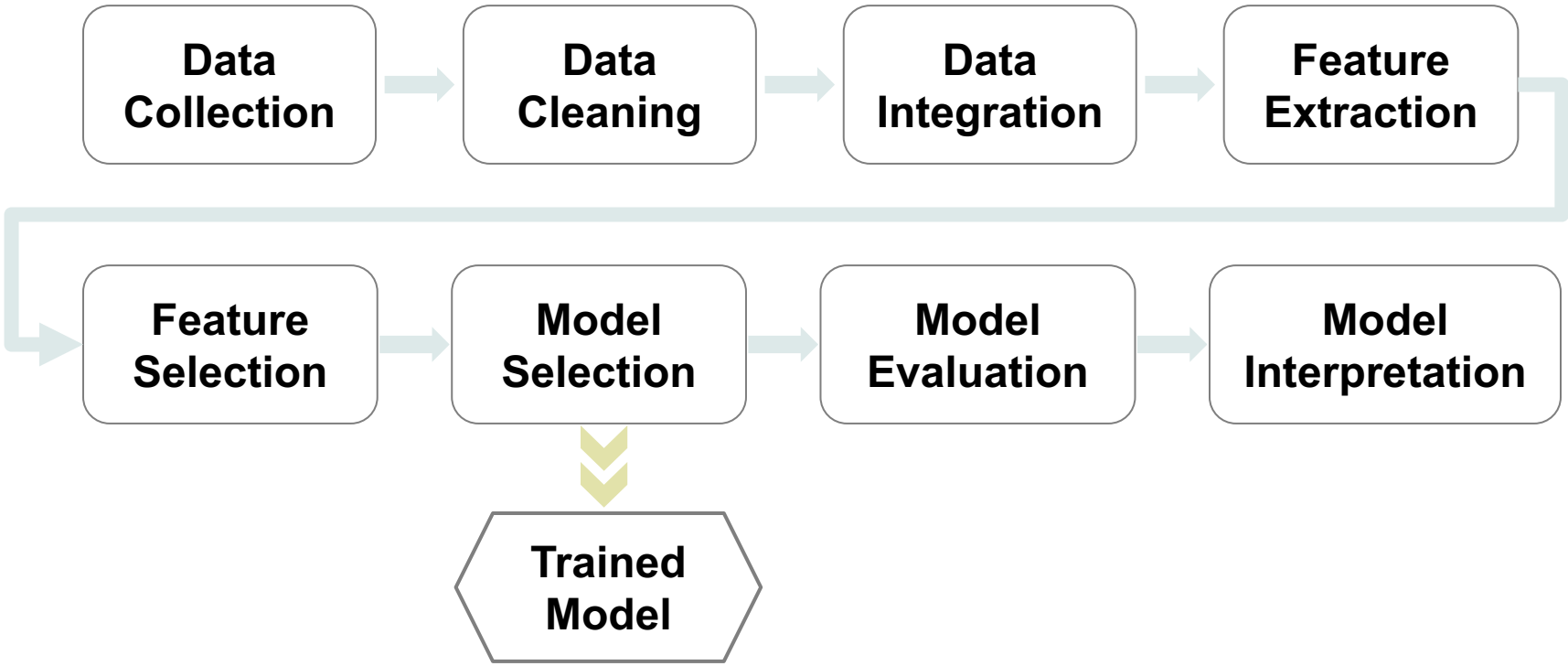




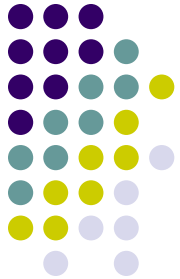


Automation for Machine Learning Workflow



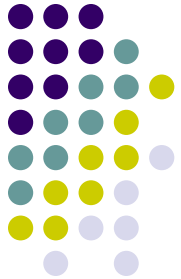


Automated Machine Learning



- Machine learning experts usually perform following tasks:
  - Preprocess the data / select appropriate features.
  - Select appropriate model family.
  - Optimize model hyperparameters.
  - …
- Growth in machine learning applications demands automation for non expert users.
- Automation is also helpful to expert users:
  - Configuration of deep architecture.
  - Importance analysis of hyperparameters

Automated ML



- Research area that targets progressive automation of machine learning.
- Given a target dataset the objective is to provide:
  - Automated hyper parameters settings.
  - Automated algorithms selection
  - Automated configuration and workflow.

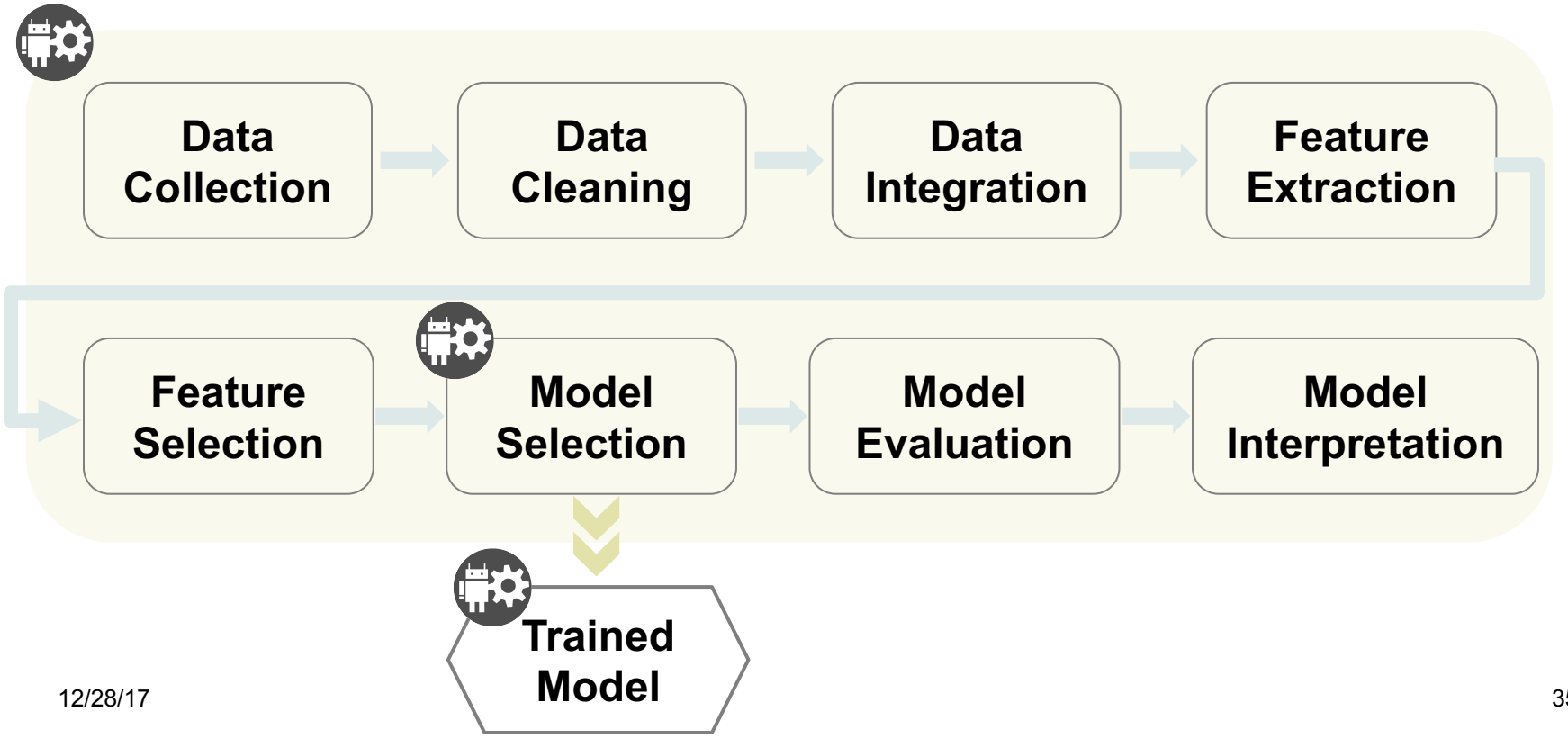
[1] http://www.ml4aad.org/automl/

Techniques for Automated ML

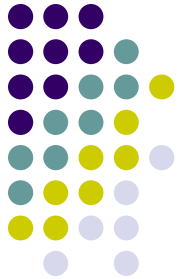
Automated Hyper-parameter Setting and Algorithm Selection

Workflow Configuration and Automated Workflow Optimization

Lifecycle Management of Trained Models

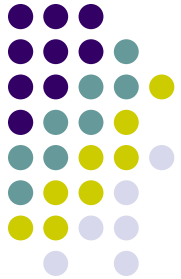


Automated algorithms selection



- Following framework discovers best algorithm with their optimum hyper parameter setting
  - Auto-Weka.
  - Auto-Sklearn.

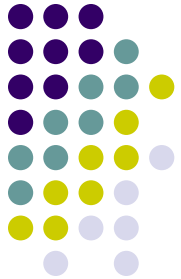
Auto-Weka



- Weka is widely used machine learning platform.
- Auto-Weka was developed for automatic model selection and hyper parameter optimization in Weka.
- Uses Tree based Bayesian optimization (SMAC) to search through joint space of Weka learning algorithms and their respective hyper-parameter settings

[1] Auto-WEKA 2.0: Automatic model selection and hyperparameter optimization in WEKA, Lars Kotthof et al., KDD 2013

Auto-Sklearn



- Uses same Bayesian optimization as Auto-Weka
- But instead, support classification and pre-processing models implemented in scikit-learn library.
- It includes meta-learning step to warm start the optimization procedure by using knowledge of similar datasets.
- Ensemble model construction for robust performance.

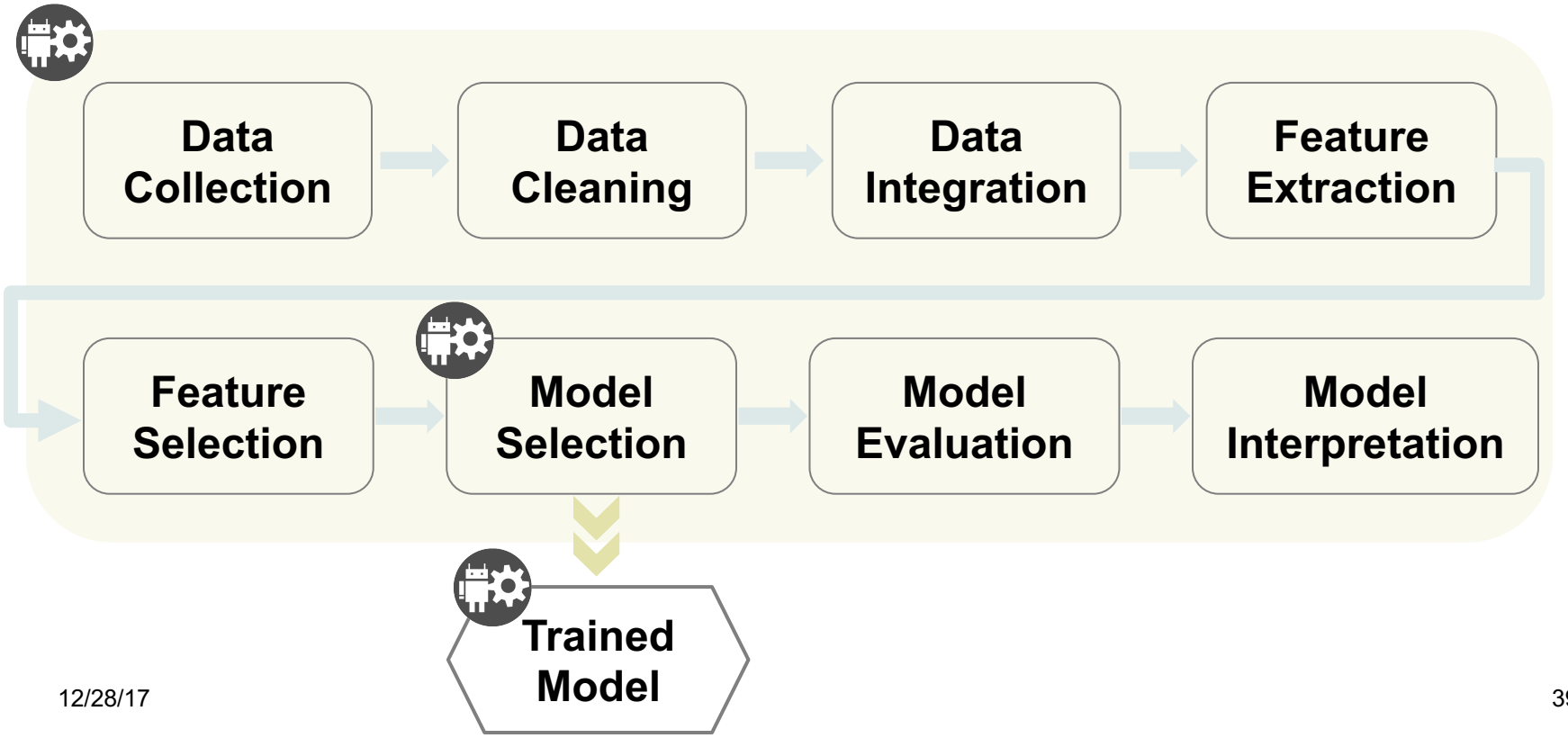
[1] Efficient and robust automated machine learning, Mathias Faurer et al., NIPS 2015

Techniques for Automated ML

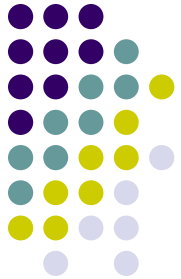
Automated Hyper-parameter Setting and Algorithm Selection

Workflow Configuration and Automated Workflow Optimization

Lifecycle Management of Trained Models

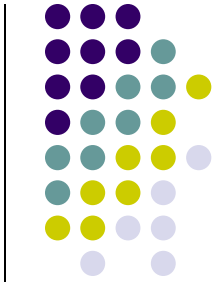


Workflow Configuration



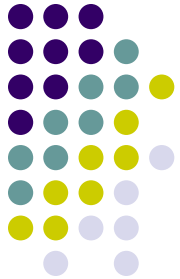
- Workflow configuration employs web UI, graphical UI, or API to:
  - Configure the data pre-processing techniques, feature extraction or selection algorithms, and prediction algorithms used in each transaction of a machine learning workflow.
  - Specify the hyper-parameters used by those techniques, algorithms.

Automated Workflow Optimization



- Automated workflow optimization searches the configuration space of a machine learning workflow, including data pre-processing techniques, feature extraction or selection algorithms, prediction algorithms and their corresponding hyper-parameters, and tries to discover the optimal configuration given a data set.

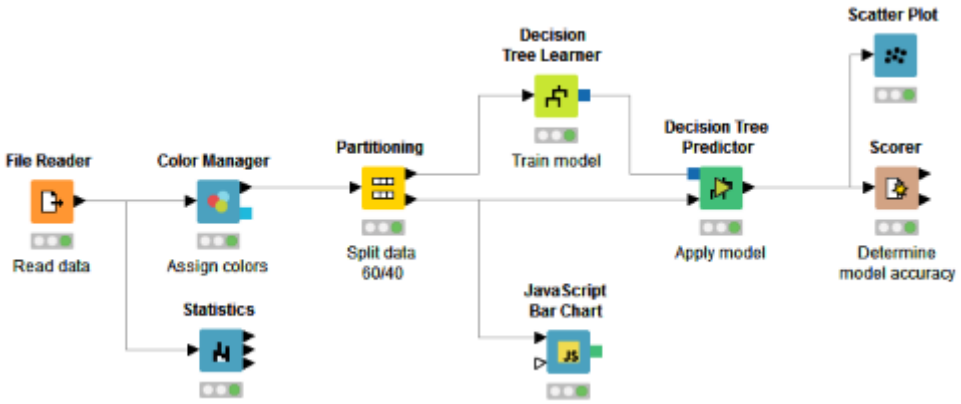
Platforms for Workflow Configuration and Automated Workflow Optimization



- Workflow Configuration
  - KNIME
  - FBLearner Flow
  - Michelangelo
  - OpenML
- Automated Workflow Optimization
  - DataRobot
  - TPOT
  - MLBase

KNIME

- It enables easy visual assembly and interactive execution of a machine learning workflow.
- Basic steps in constructing a workflow: add transactions [1], connect transactions, configure transactions, execute transactions, inspect results

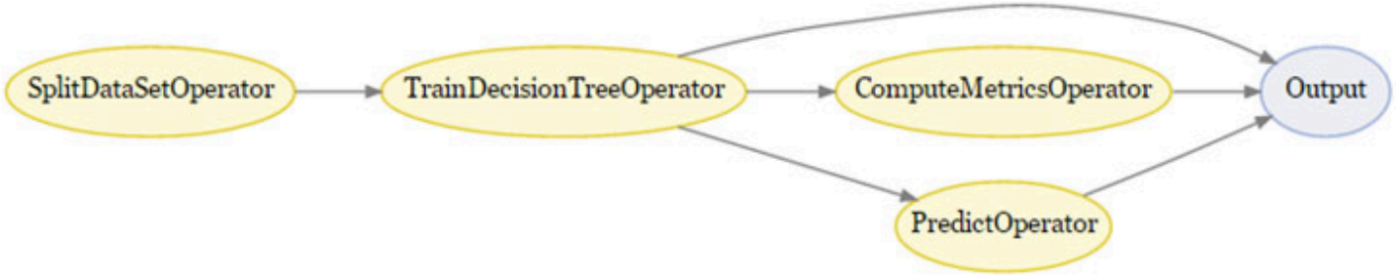


[1] In KNIME , a transaction in a machine learning workflow is referred to as “node”.

Image source: https://www.knime.org/knime-analytics-platform

FBLearner Flow

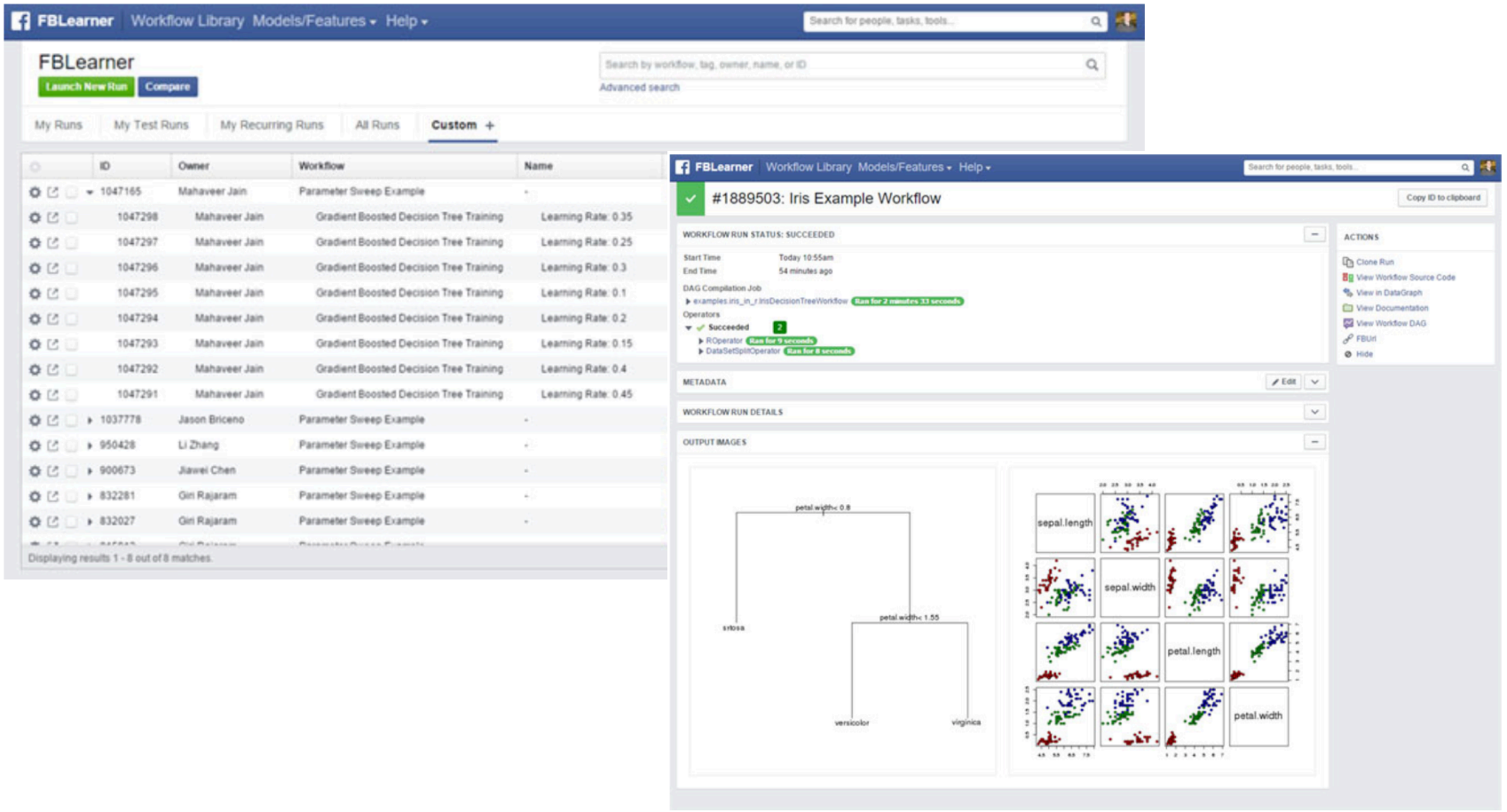
- The platform consists of three core components:
  - A cluster for running distributed workflows.
  - An experimentation management UI for launching experiments and viewing results.
  - Predefined workflows for training the most commonly used machine learning algorithms at Facebook.



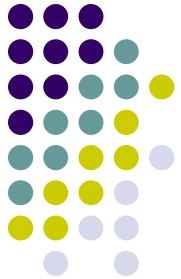
https://code.facebook.com/posts/1072626246134461/introducing-fblearner-flow-facebook-s-ai-backbone/

FBLearner Flow Cont'd

- Experimentation management UI



Michelangelo



- Michelangelo is a workflow management system developed by Uber to standardize and reuse machine learning workflows across teams. It provides both offline and online workflows.
- It employs a feature store to allow teams to share, discover, and use a highly curated set of features for their machine learning problems.

https://eng.uber.com/michelangelo/

DataRobot

- DataRobot is a proprietary data science system built for business users.
- The machine learning algorithms supported by DataRobot cannot be easily extended.

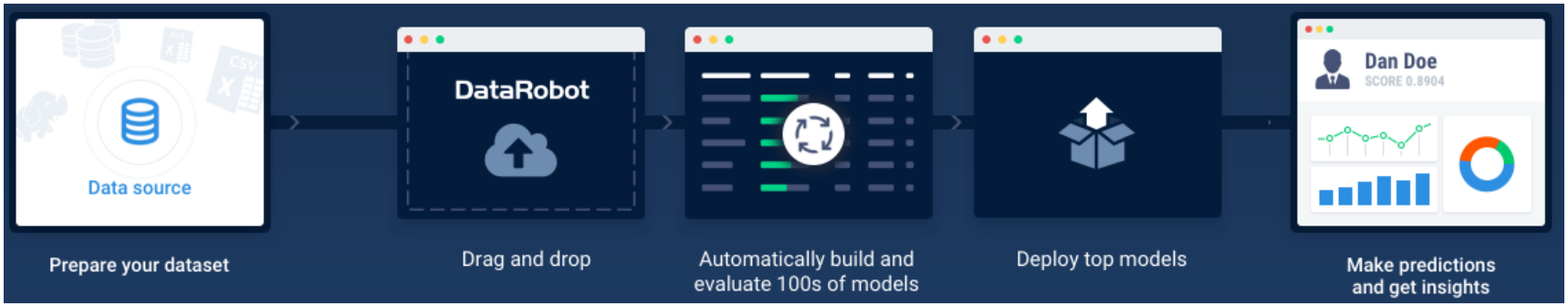
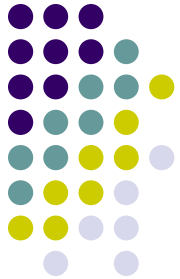


Image source: https://www.datarobot.com/

TPOT

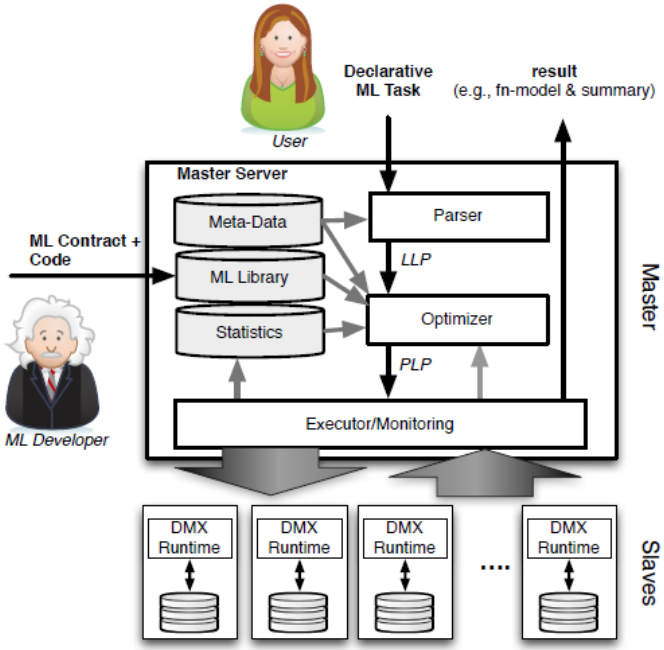


- TPOT discovers the optimal workflow given a data set using genetic programming.
- The discovered workflow is exported as a piece of code.
- The scalability of TPOT may be problematic since its process of finding optimal workflow were not designed for distributed computation.

https://github.com/rhiever/tpot

MLBase

- MLbase is designed to be fully distributed, and it offers a run-time to exploit the characteristics of machine learning algorithms
- The optimizer in MLBase transforms a declarative ML task into a sophisticated learning plan.
- It requires non-negligible efforts to implement an Spark compatible algorithm to achieve distributed computation.



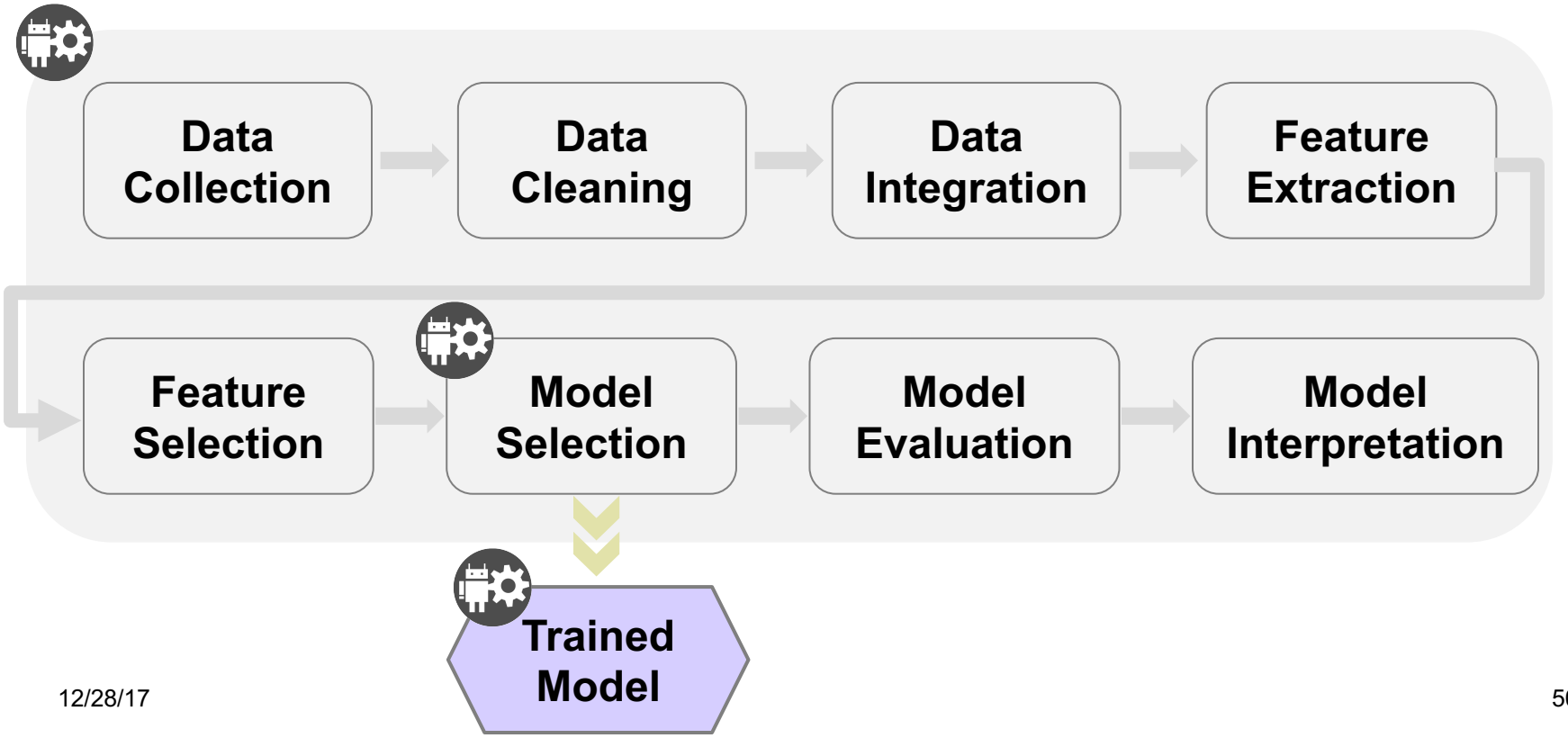
Kraska, T., Talwalkar, A., Duchi, J.C., Griffith, R., Franklin, M.J. and Jordan, M.I., 2013, January. MLbase: A Distributed Machine-learning System. In *CIDR* (Vol. 1, pp. 2-1).

Techniques for Automated ML

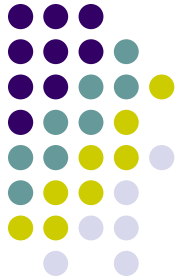
Automated Hyper-parameter Setting and Algorithm Selection

Workflow Configuration and Automated Workflow Optimization

Lifecycle Management of Trained Models



Trained model management



- Life cycle management
- Model reuse
- Model service
- ModelHub
- ModelZoo
- Clipper

Machine learning life-cycle

- Reference models used in similar domains.
- Specify input dataset.
- Create your initial model and hyper-parameter values.
- Test model on testing datasets.
- Evaluate accuracy of model.
- If accuracy is not at a desired level, update hyper-parameters and retest and evaluate again.
- Once model is at a desired level of accuracy, make available for reuse.

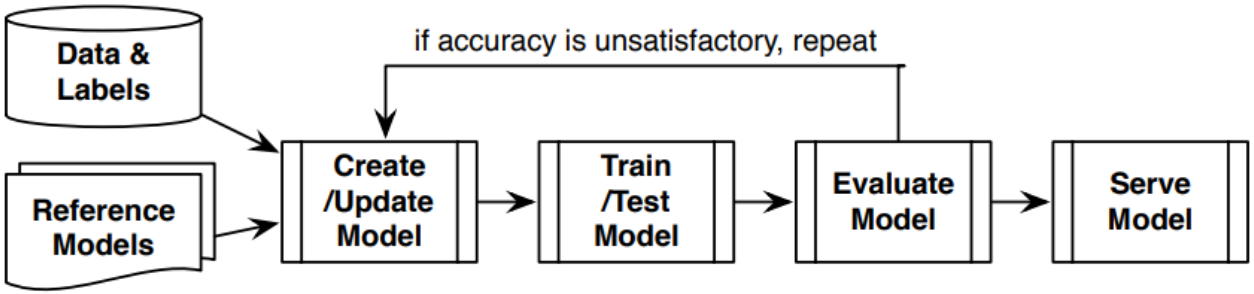
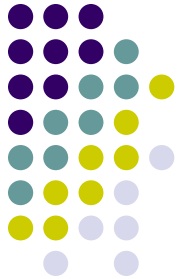


Image source: “Towards Unified Data and Lifecycle Management for Deep Learning” Miao, Ki, Davis, Deshpande; 2016

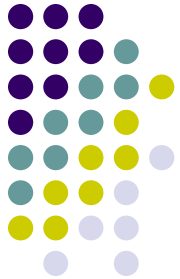
Life-cycle management



- Trained models are stored in a repository.
- These trained models are searchable and reusable.
- Version numbers of each model are stored and all versions should be made available.
- Difference between versions of trained models is the values of the hyper-parameters.

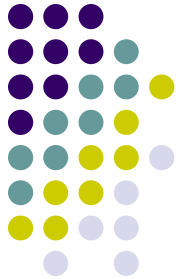
"Towards Unified Data and Lifecycle Management for Deep Learning" Miao, Ki, Davis, Deshpande; 2016

Model reuse



- Take already created models and use them for a new dataset.
- Could use previously trained models and hyper-parameter values.
- Or attempt to keep the machine learning life-cycle going and retrain the model on your dataset. Would create a new version of the model.
- Model can then be used on data in the same domain as your dataset.

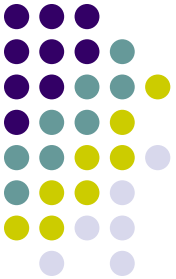
Model service



- A model service is a repository or toolkit used to store models for discovery or reuse.
- Can be used for providing the model algorithm or a trained model for a data set.
- Must be scalable to handle model deployment and predictions.
- Can be thought of as a version control system for machine learning models.
- Should incorporate life-cycle management and reuse of models.

"Towards Unified Data and Lifecycle Management for Deep Learning" Miao, Ki, Davis, Deshpande; 2016
"Clipper: A Low-Latency Online Prediction Serving System" Crankshaw, Wang, Zhou, Franklin, Gonzalez, Stoica; 2017

ModelHub



- Introduced by Hui Miao, Ang Li, Larry S. Davis, and Amol Deshpande in 2016
- Addresses data management, model reuse, and lifecycle management issues related to deep neural network models.
- Provides a high-level domain specific language for model exploration and manipulation.
- Can be used as a version control system for models.
- Versions of models are stored in a repository and search, push, and pull queries can be used.
- These commands are necessary to have the collaboration needed for the reuse of models.

"Towards Unified Data and Lifecycle Management for Deep Learning" Miao, Ki, Davis, Deshpande; 2016

ModelHub

- Integrated with caffe, torch, tensorflow, and other popular systems for deep neural networks.
- Models can be evaluated using different hyperparameters and datasets.

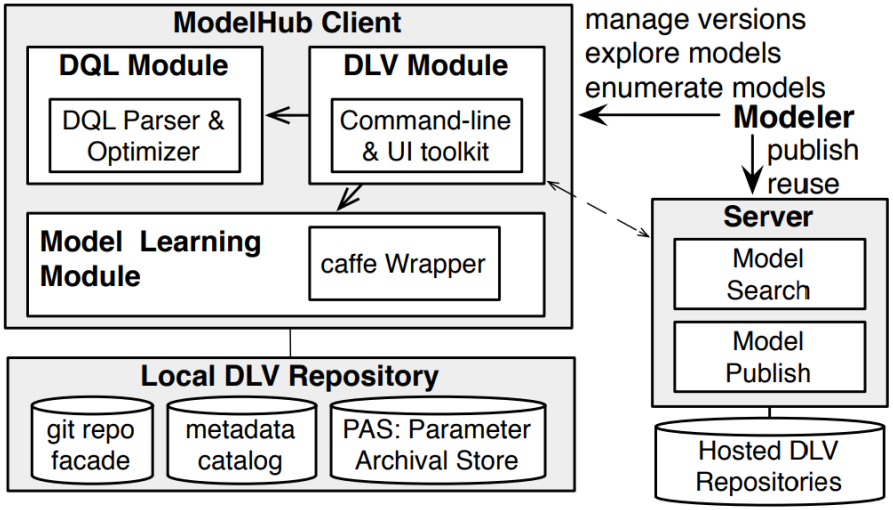


Image source: “Towards Unified Data and Lifecycle Management for Deep Learning” Miao, Ki, Davis, Deshpande; 2016

ModelZoo

- Created by Yangqing Jia in 2016.
- Provides tools to upload and download models built using the Caffe package to and from Github Gists.
- Uses a central wiki page to provide information on the models and give instructions for reuse.
- Community of users that can edit the wiki to include information on their models.

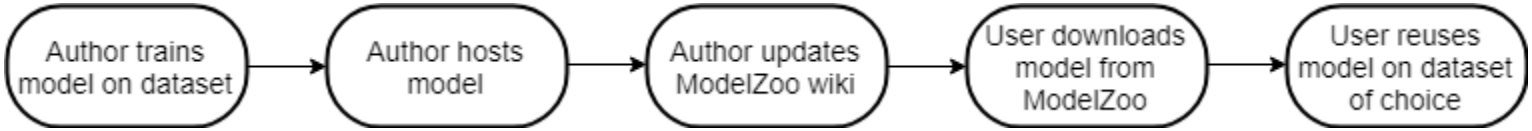
To acquire a model:

1. download the model gist by `./scripts/download_model_from_gist.sh <gist_id> <dirname>` to load the model metadata, architecture, solver configuration, and so on. (`<dirname>` is optional and defaults to caffe/models).
2. download the model weights by `./scripts/download_model_binary.py <model_dir>` where `<model_dir>` is the gist directory from the first step.

"ModelZoo", https://github.com/BVLC/caffe/wiki/Model-Zoo

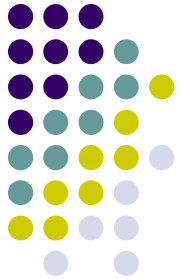
ModelZoo

- Berkeley Artificial Intelligence Research (BAIR) trained models provided by ModelZoo for free use.
- Any other community user can edit the wiki for their own models.
- Can upload model info, trained models, or both.
- It is up to the model author what conditions are put on the models for reuse such as how to cite or licensing information.
- Provide information on how to access the models.
- Files can be hosted anywhere the model author chooses.



"Caffe ModelZoo", http://caffe.berkeleyvision.org/model_zoo.html

Clipper



- Introduced by Crankshaw et al in 2017
- “General-purpose low-latency prediction serving system”
- Aims to improve prediction efficiency and accuracy by caching, batching, and adaptive trained model selection techniques.
- Model selection based on predictions Clipper makes on each model.

“Clipper: A Low-Latency Online Prediction Serving System” Crankshaw, Wang, Zhou, Franklin, Gonzalez, Stoica; 2017

Clipper

- Built on standard frameworks.
- Model abstraction layer.
- Model selection layer.
- Model used in application

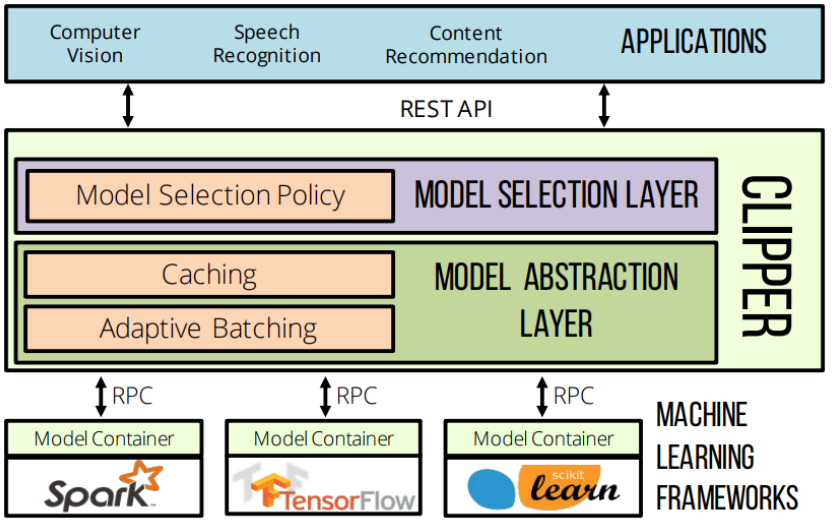
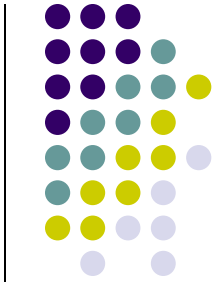


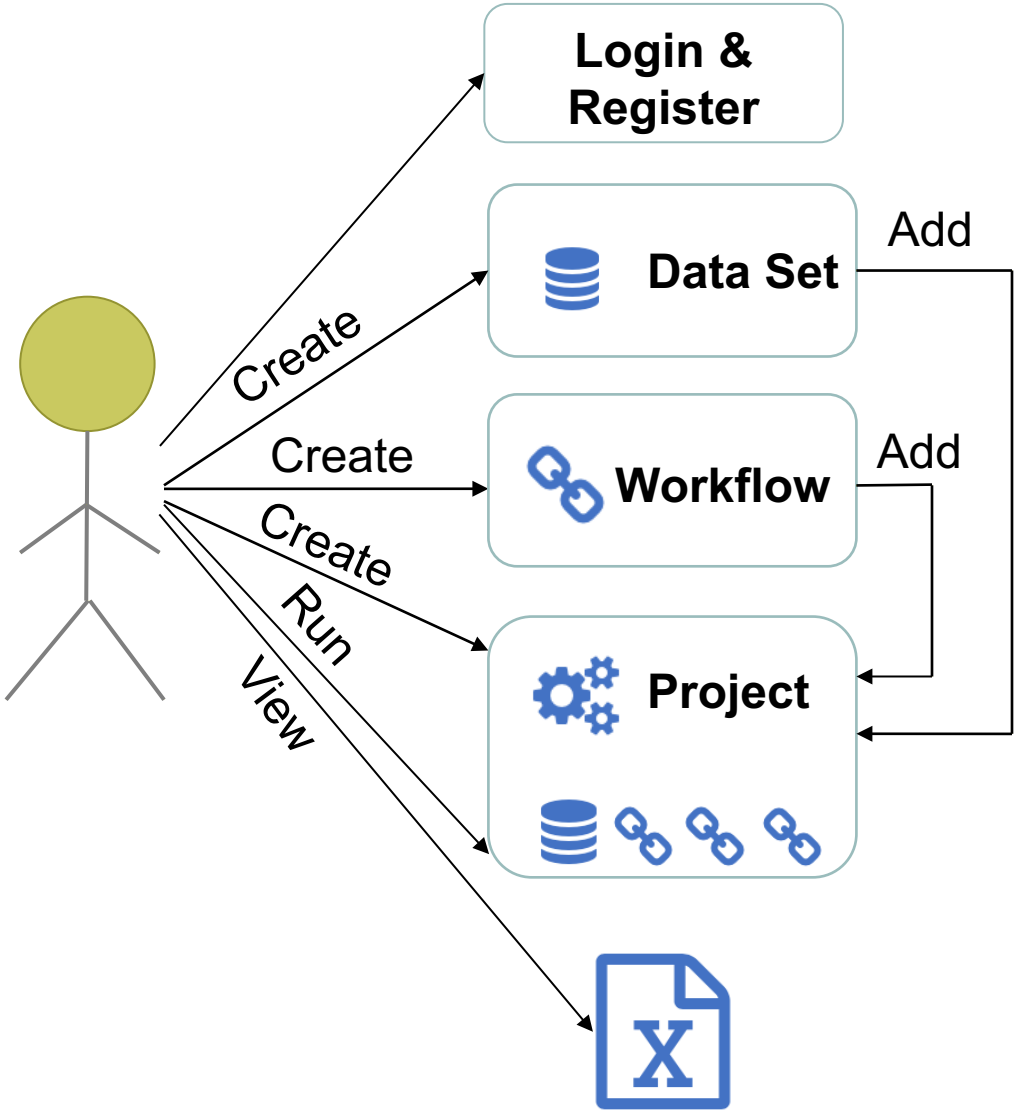
Image source: “Clipper: A Low-Latency Online Prediction Serving System” Crankshaw, Wang, Zhou, Franklin, Gonzalez, Stoica; 2017

WOLF

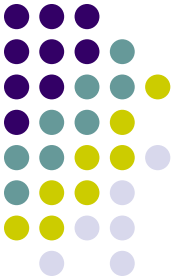


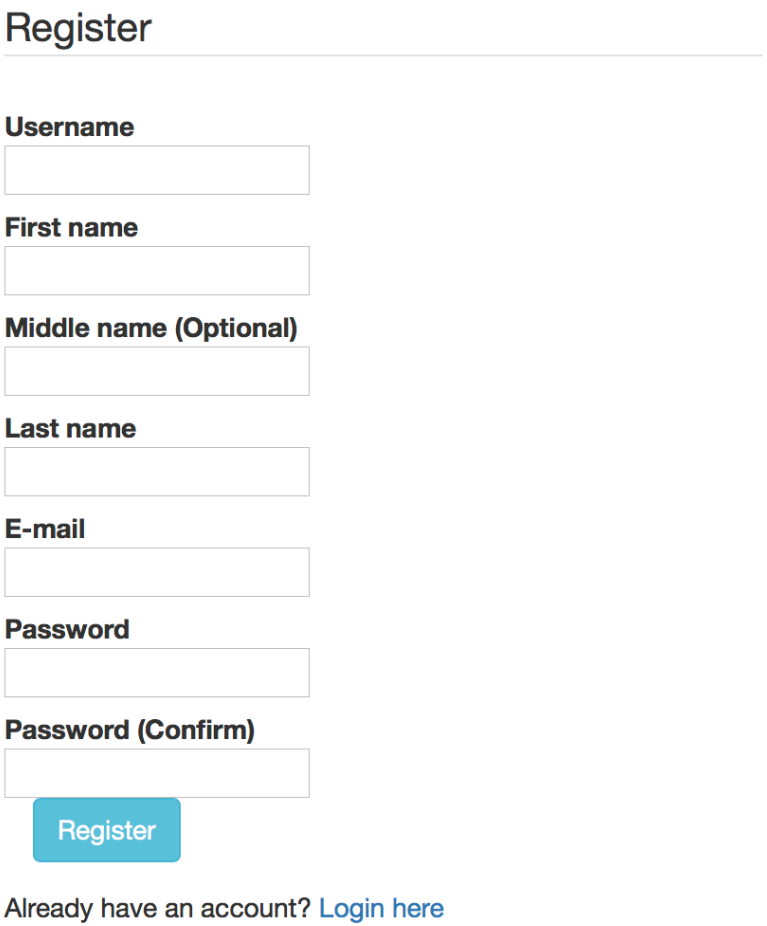
- WOLF is an automated machine learning WOrkfLow Optimization Framework
- Three characteristics of WOLF are:
  - Flexibility
    - WOLF is configured through a plain text file so that we can flexibly change the workflow to use all or partial machine learning components with variable choices of algorithms for each transaction
  - Extensibility
    - WOLF provides the possibility to integrate customized algorithms through API
  - Scalability
    - To speed up the time-consuming model selection and comparison, WOLF is designed to run over clusters/cloud.

Use Cases in WOLF

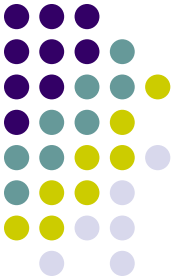


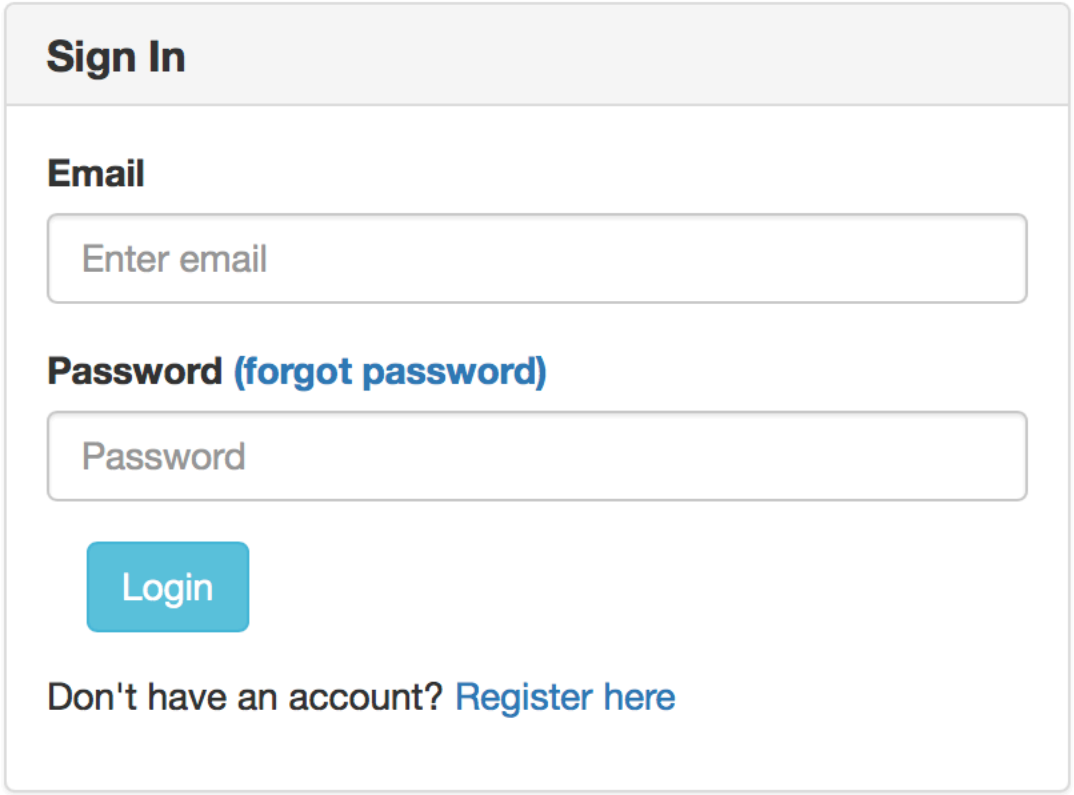
Register



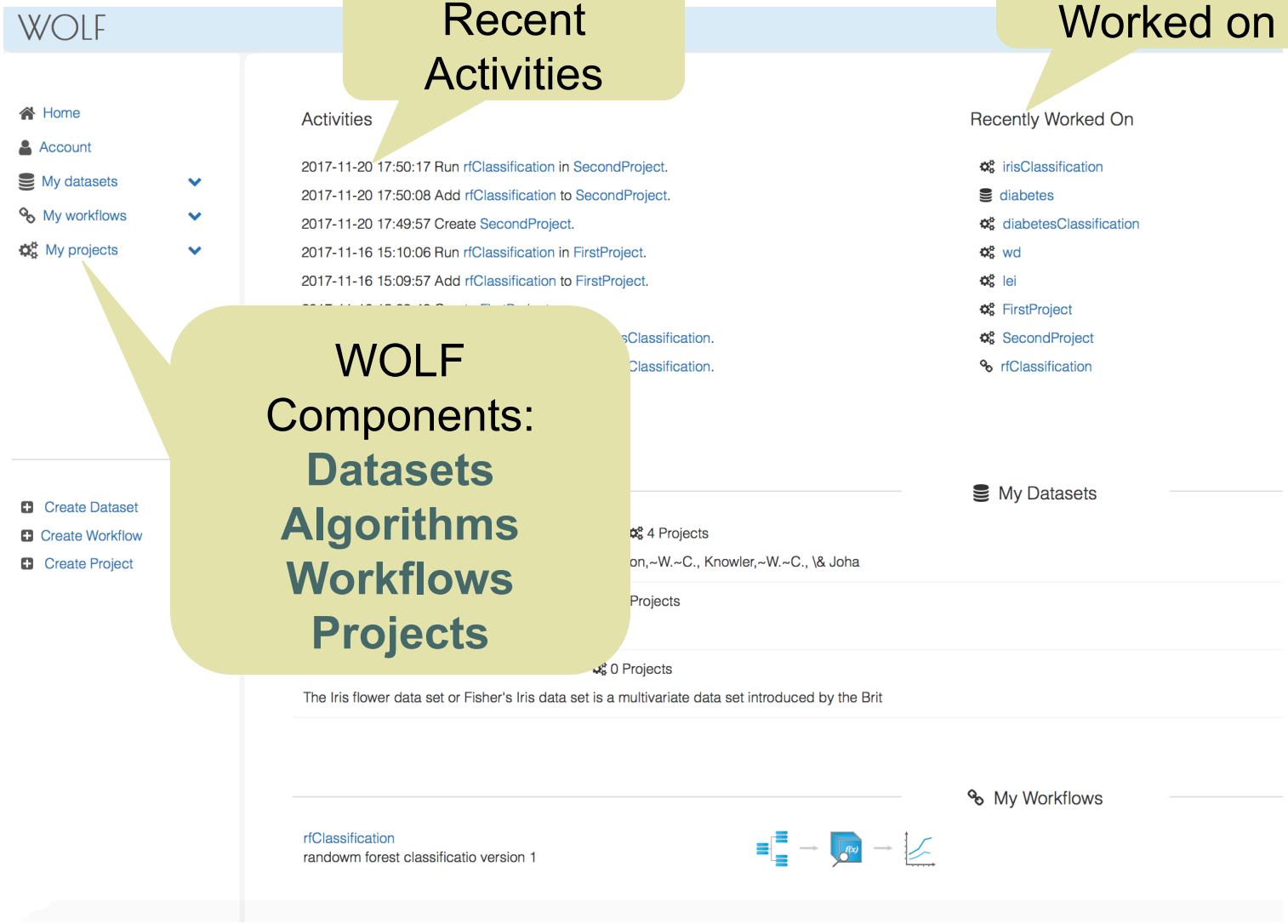

Register
Username
First name
Middle name (Optional)
Last name
E-mail
Password
Password (Confirm)
Register
Already have an account? Login here

Login




Sign In
Email
Enter email
Password (forgot password)
Password
Login
Don't have an account? Register here

Home

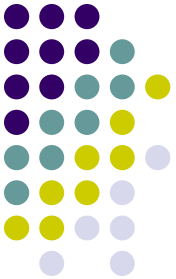


Recent Activities

Components Worked on

WOLF Components:
Datasets
Algorithms
Workflows
Projects

Create a Dataset



Create New Data Set

Data Set Name: DiabetesData

Data Set Description: Diabetes patient records were obtained from two sources: an automatic electronic recording device and paper records. The automatic device had an internal clock to timestamp events, whereas the paper records only provided "logical time" slots (breakfast, lunch, dinner, bedtime). For paper records, fixed times were assigned to breakfast (08:00), lunch (12:00), dinner (18:00), and bedtime (22:00). Thus paper records have fictitious uniform recording times whereas electronic records have more realistic time stamps.

Creator: Michael Kahn

Source: Michael Kahn, MD, PhD, Washington University, St. Louis, MO

Cite Information: @misc{Lichman:2013 , author = "M. Lichman", year = "2013", title = "{UCI} Machine Learning Repository", url = "http://archive.ics.uci.edu/ml", institution = "Un

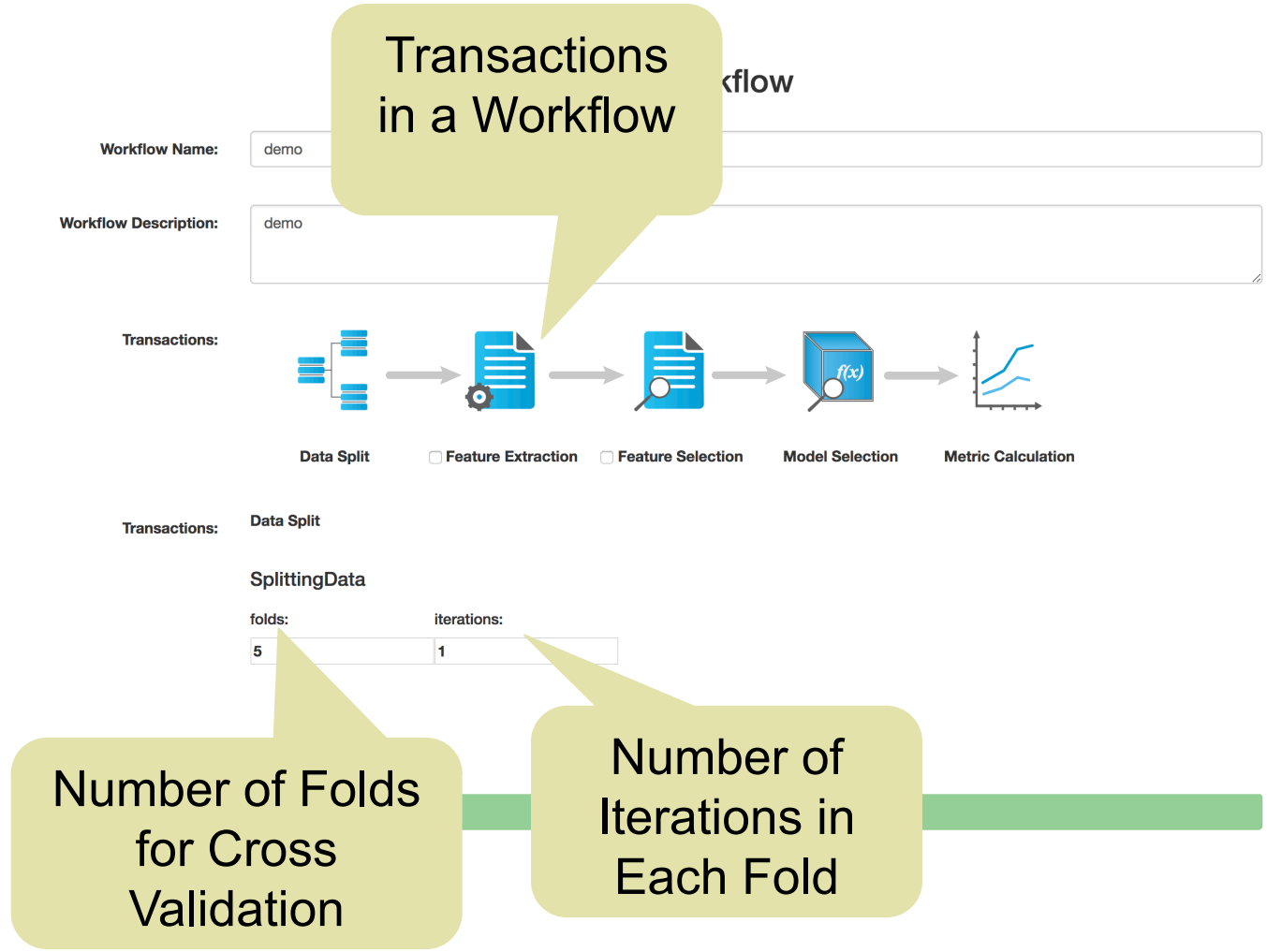
Domain Information: Medical

Upload File: Browse... diabetes.arff

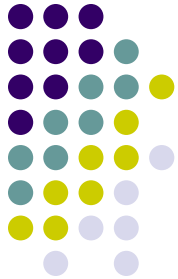
Publish Status: ○ Under Development ◉ Published

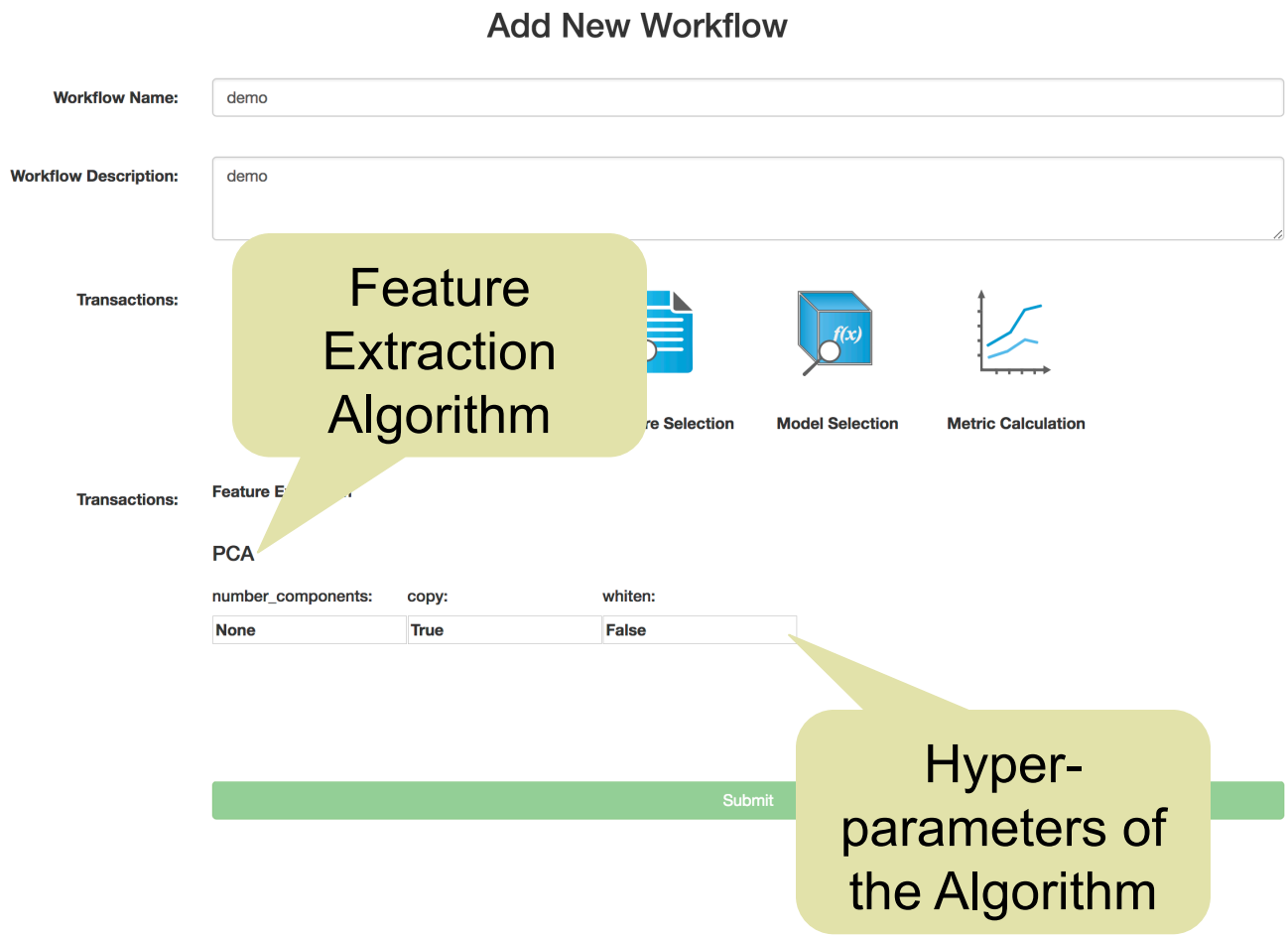
Submit

Create a Workflow: Data Splitting

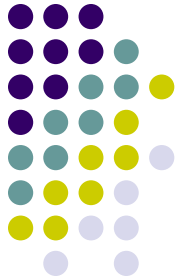


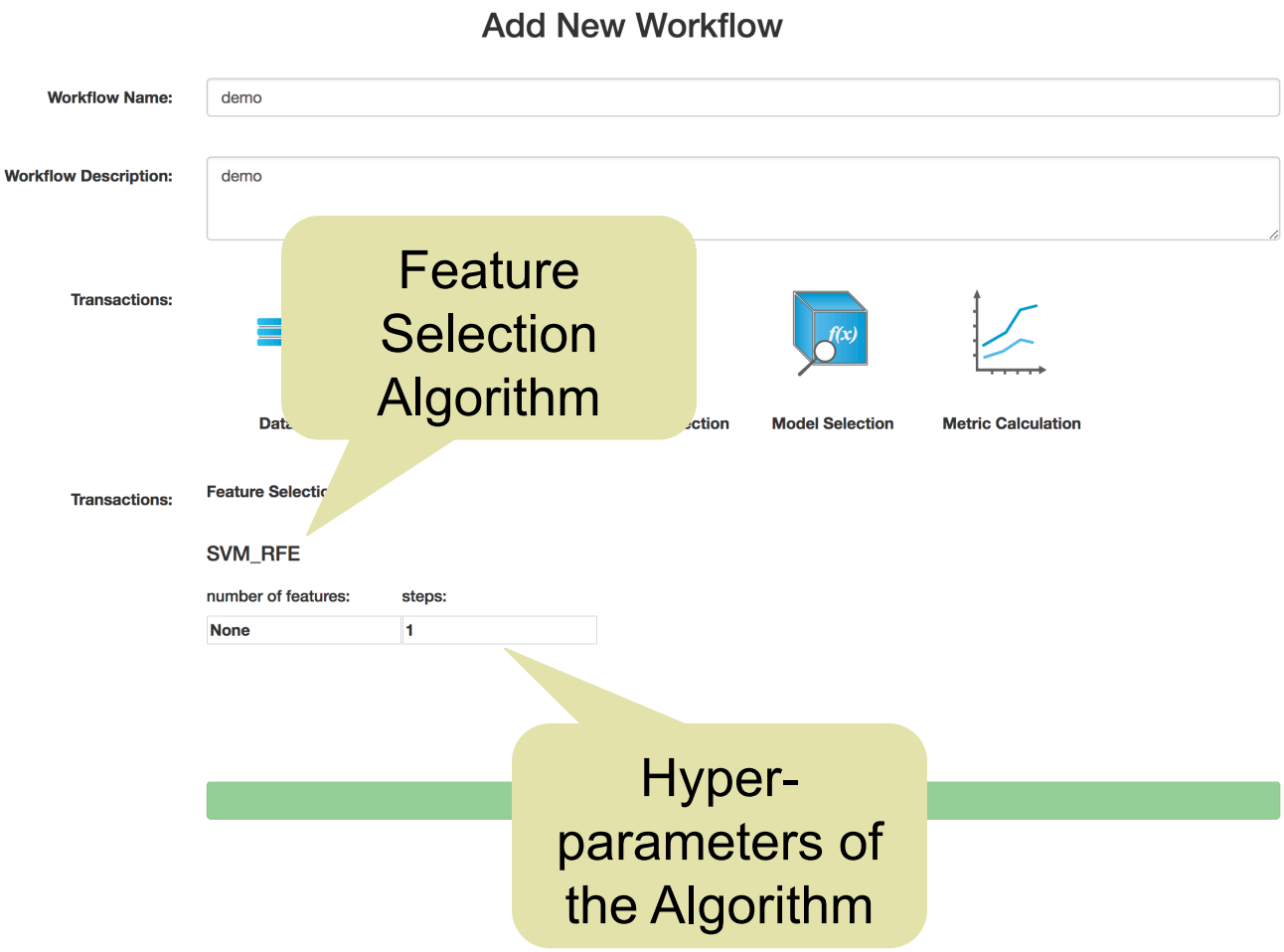
Create a Workflow: Feature Extraction



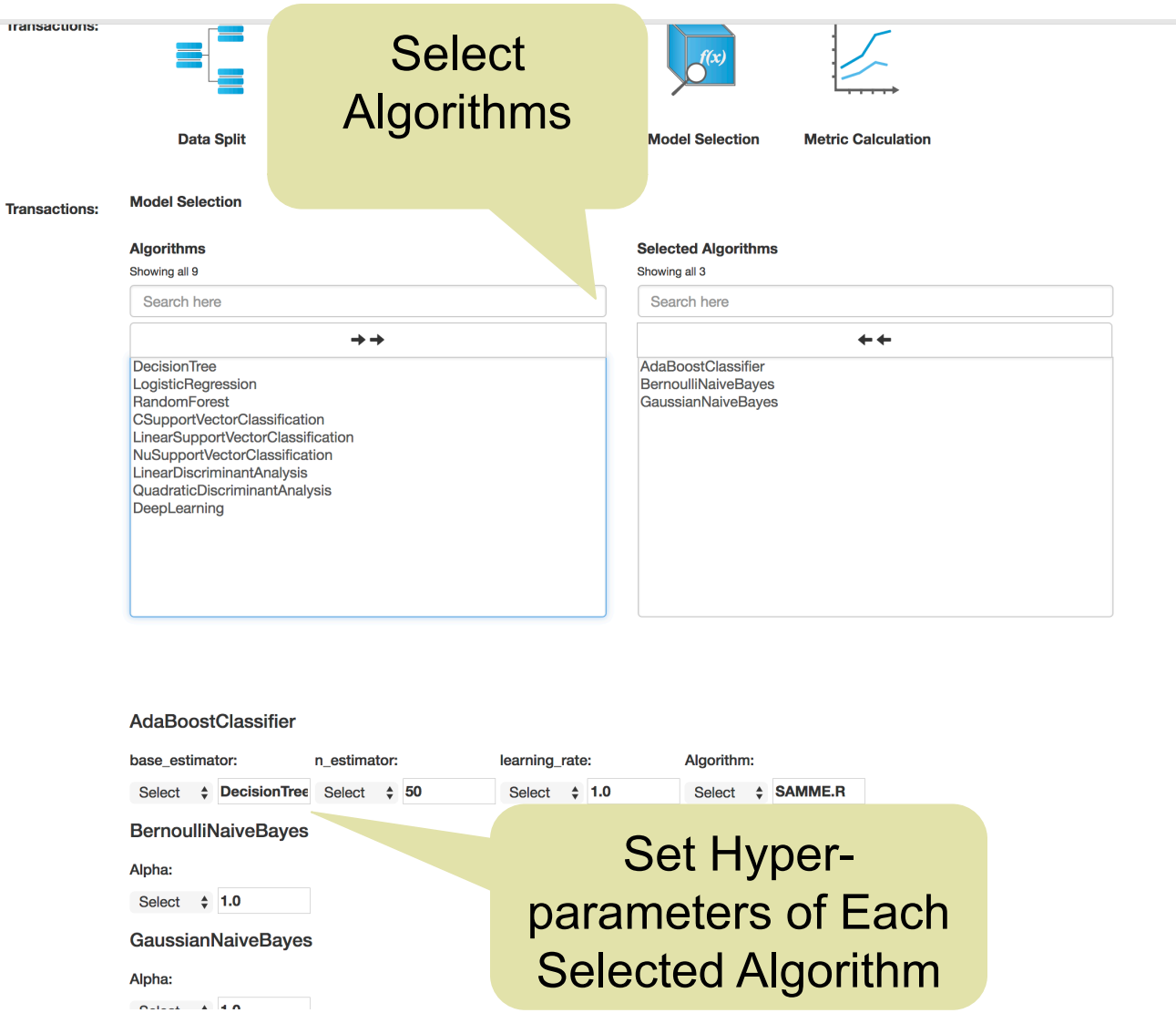


Create a Workflow: Feature Selection

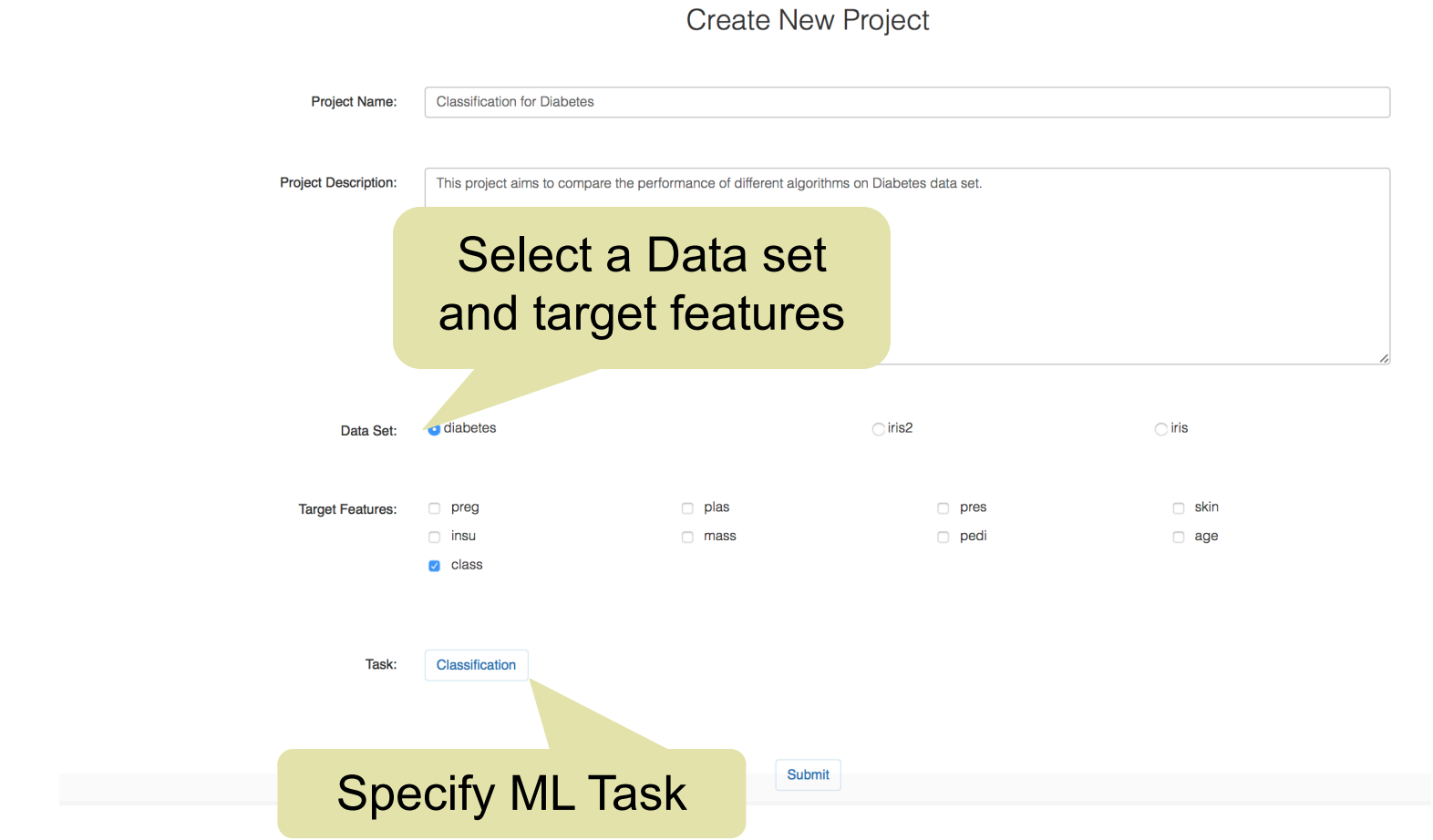




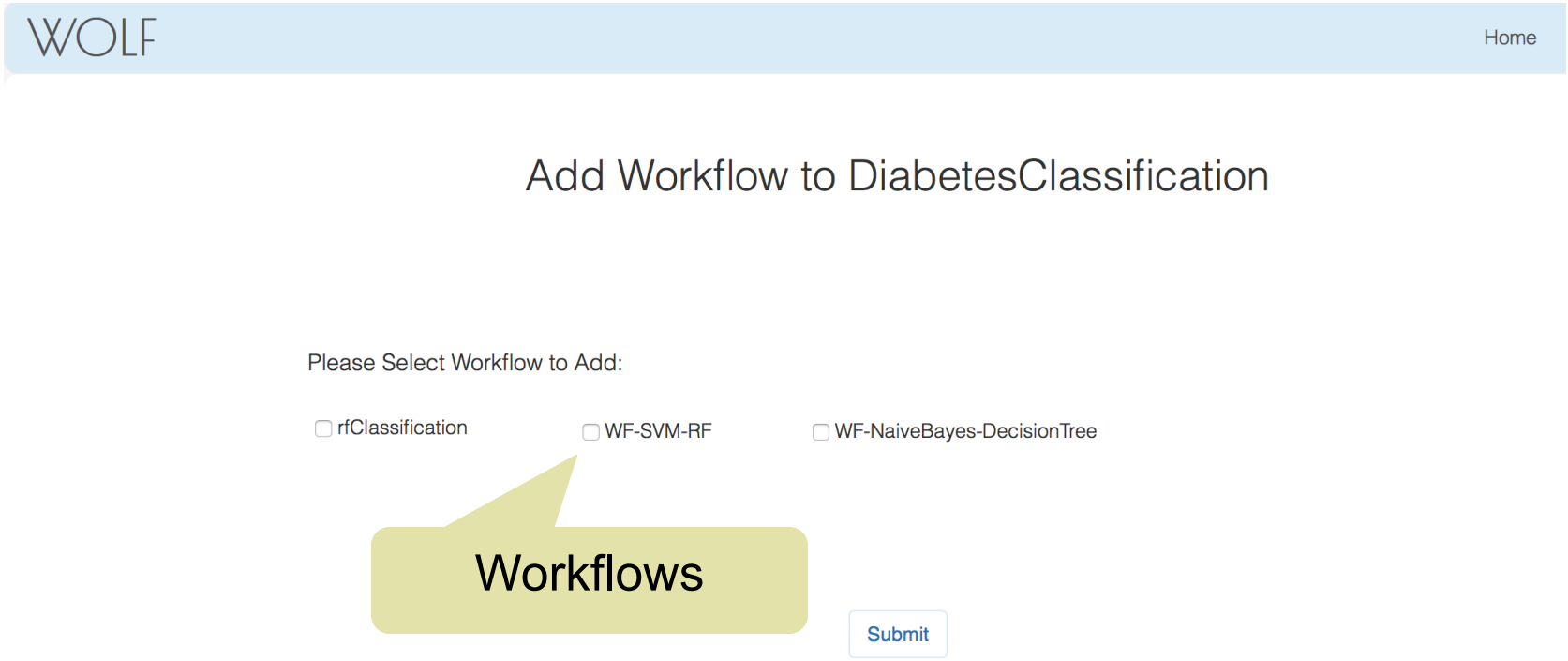
Create a Workflow: Model Selection



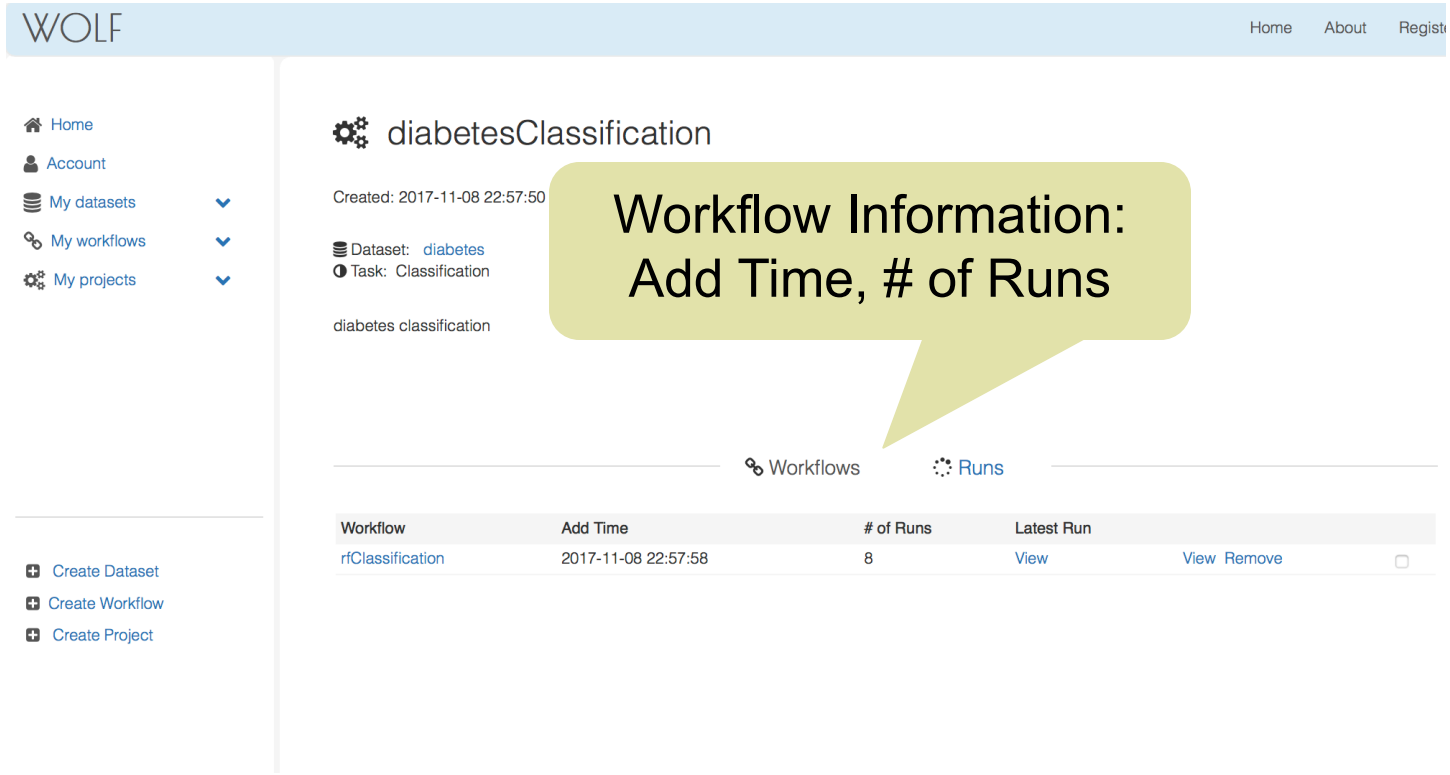
Create a Project



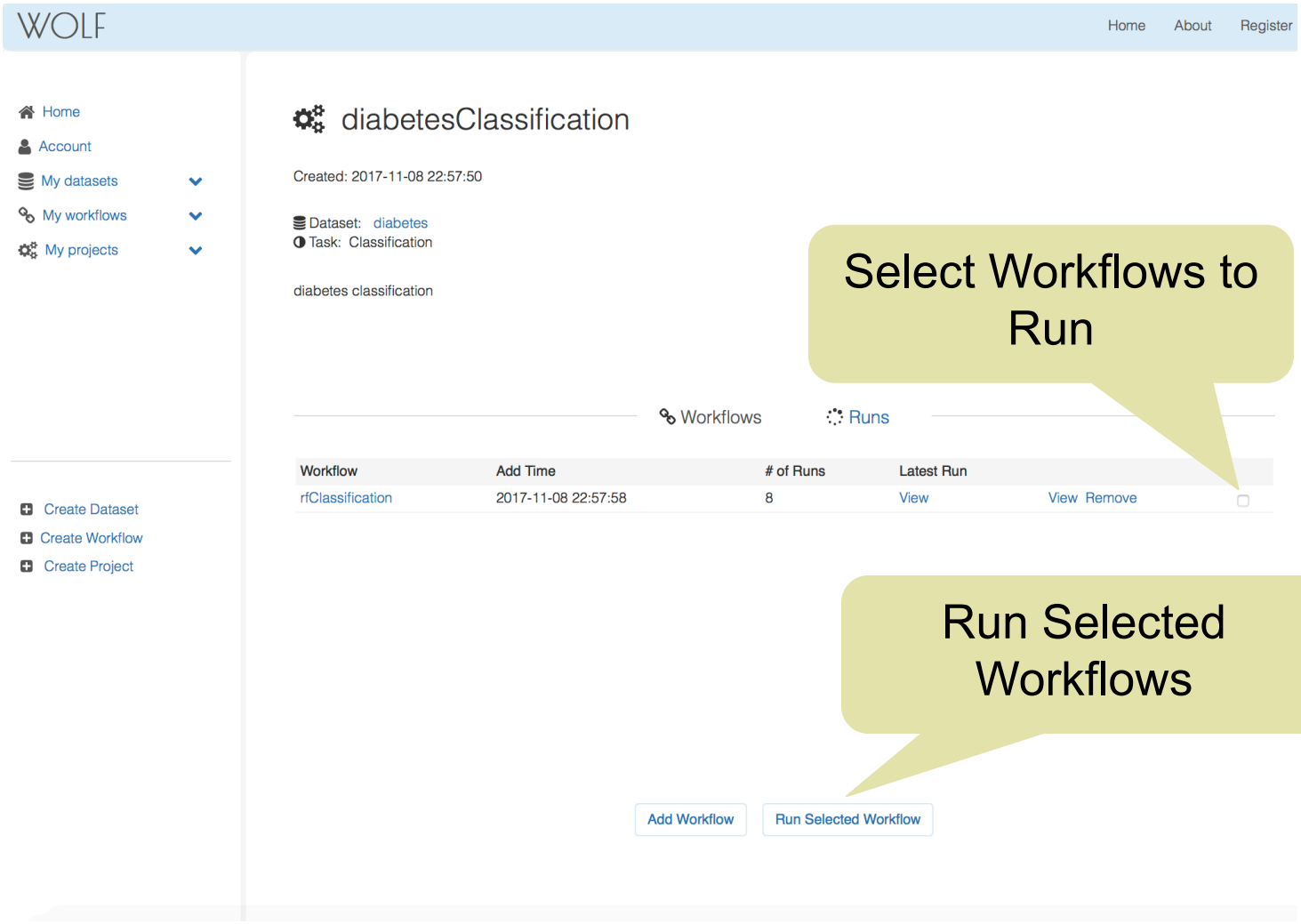
Add a Workflow to a Project



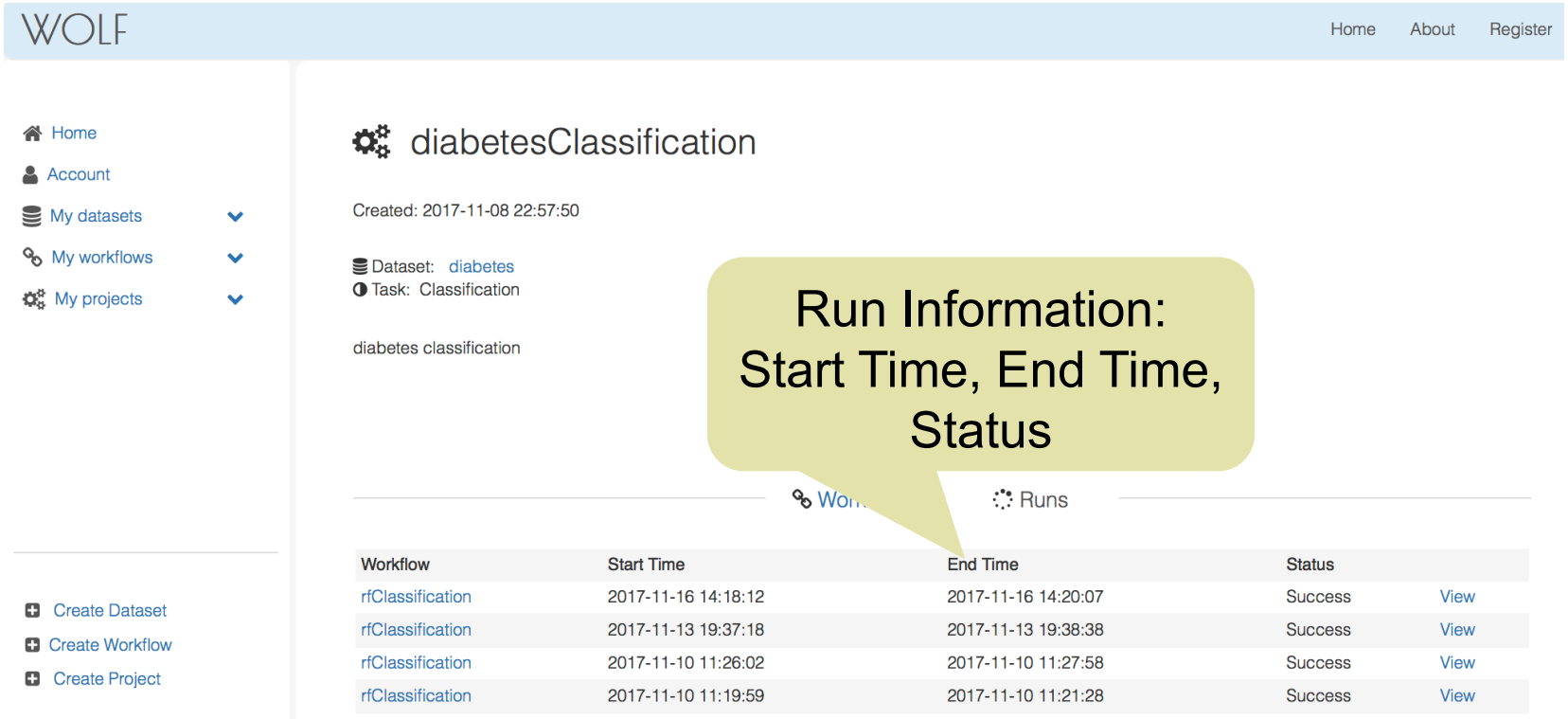
My Project: Workflows



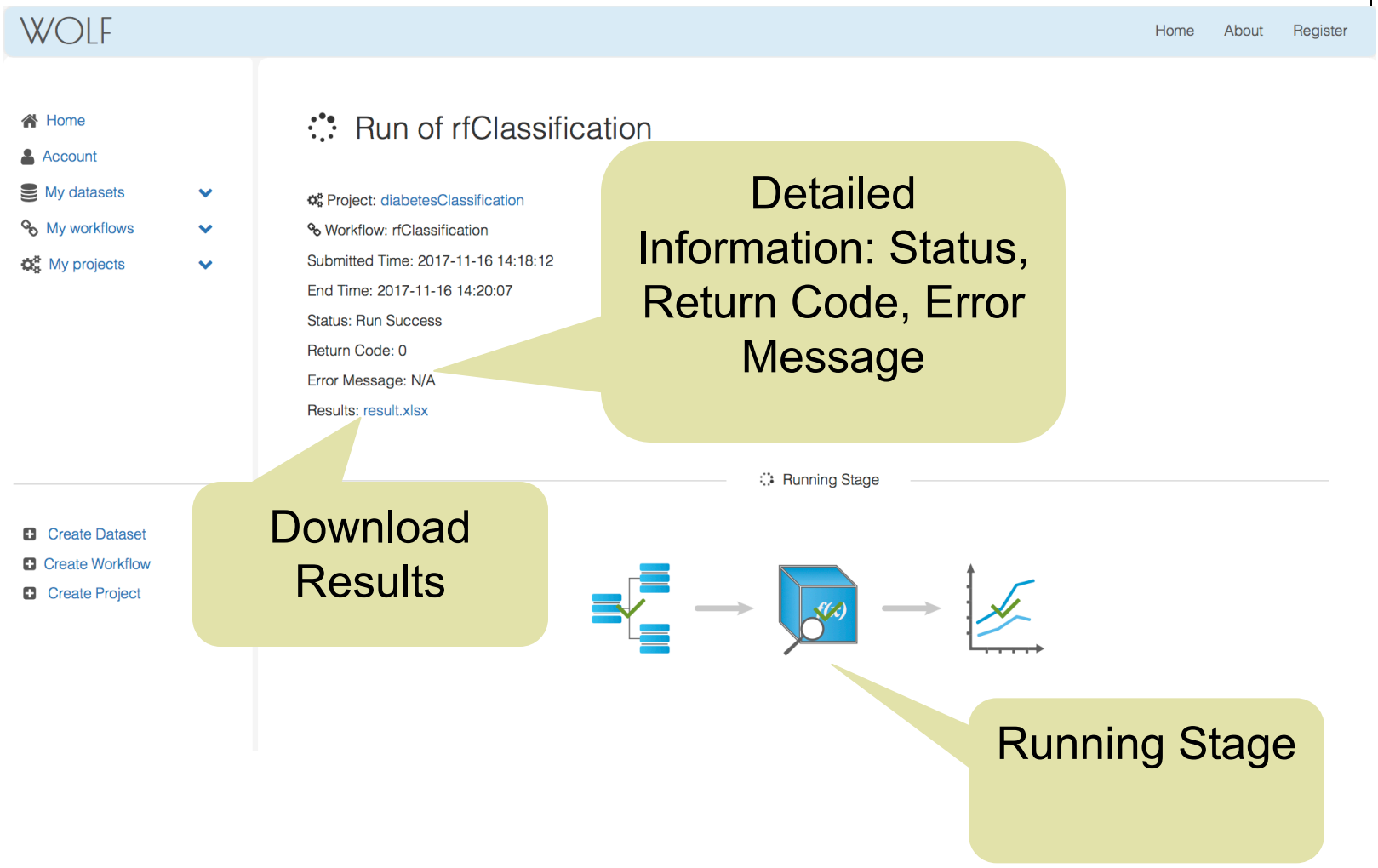
Run a Workflow



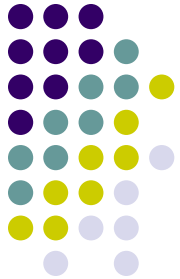
My Project: Runs



Details of a Run

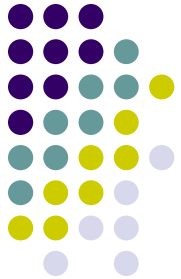


Demo Cases



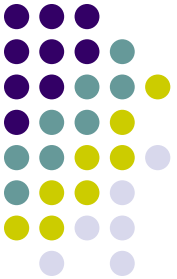
- Case 1: A workflow consisting of Decision tree with Default Hyper-parameter values
- Case 2: A workflow consisting of Decision Tree with a Hyper-parameter having a list a values
- Case 3: A workflow consisting of Decision Tree with a Hyper-parameter having values within a range
- Case 4: A workflow consisting of Decision Tree with a Hyper-parameter having a list a values and a Hyper-parameter having values within a range
- Case 5: A workflow consisting of multiple algorithms

Investigating DNN for Modeling Bioactivity Data



- We investigated
  - How to optimize DNN hyper-parameters to achieve good classification performance for classifying molecules to actives/decoys for virtual screening applications.
  - Compare the performance of DNN to well-established shallow methods for classification tasks:
    - Random Forest
    - Support Vector Machines
    - Naïve Bayes

Bioactivity Datasets used in the study



Activity Class	CHEMBL Target id	Number of active inhibitors	Number of decoys
Carbonic Anhydrase II, Class: enzyme, lyase	CHEMBL205	1,896	18,960
Cyclin-dependent kinase 2, Class: protein kinase	CHEMBL301	740	7,381
hERG (the human Ether-à-go-go-Related Gene), Class: Voltage-gated ion channel	CHEMBL240	748	7,463
Dopamine D4 Receptor, Class: membrane receptor, GPCR	CHEMBL219	546	5,460
Coagulation factor X, Class: enzyme, serine protease	CHEMBL244	1,265	12,610
Cannabinoid CB1 receptor, Class: membrane receptor, GPCR	CHEMBL218	1,911	19,013
Cytochrome P450 19A1, Class: enzyme, cytochrome P450	CHEMBL1978	678	6,762

Actives: Inhibitors with measured potencies (IC50, Ki) equal to or better than 10uM
Decoys: Compounds randomly sampled from a large pool of bioactive molecules against other proteins targets with MW (up to 900)
Source: ChEMBL version 20

Koutsoukas A., Monaghan K.J., Li X. and Huan J. *.Deep-Learning: Investigating feed-forward Deep Neural Networks hyper-parameters and Comparison of Performance to Shallow Methods for Modeling Bioactivity Data.* Journal of Chemical Informatics and Modeling (2016 submitted)

ML Hyper-Parameters Explored

Bernoulli Naïve Bayes		
Hyper-parameters	**Values Explored**	**Parameter**
Alpha	1, 0.5, 0.1	Laplace/Lidstone smoothing parameter
fit_prior	True, False	Class Prior probabilities. In case of false, a uniform prior was used.
Random Forest		
Hyper-parameters	**Values Explored**	**Parameter**
Ntrees	10, 50, 100, 300, 700, 1000	Number of trees
Criterion	Gini, Entropy	Functions used to measure the quality of each split
max_features	sqrt(n_features), log2(n_features)	Number of features considered for each split
Support Vector Machine		
Hyper-parameters	**Values Explored**	**Parameter**
Kernel	rbf	Radial Basis Function
C	1000, 100, 10, 1	Cost
γ	10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}, 10^{-1}	Gamma
Neural Nets		
Hyper-parameters	**Values Explored**	**Parameter**
η	0.05	Learning rate for the Stochastic Gradient Descent ("SGD")
Momentum (μ)	0.9	Weight of the previous update
Epochs	200	Number of training epochs
Batch size	64	mini-batch training size
Hidden Layers	1, 2, 3	Number of hidden layers
Neurons	50, 100, 500, 1000, 1500, 2000	Number of neurons in each hidden layer
Activation Function	ReLU, Sigmoid, Tanh	Neuron activation functions
Regularization	No, Dropout, L2	Regularization techniques
Dropout	{50% input layer, 50% hidden layer}	% of neurons silenced using the Drop-out technique
L2	No, True	L2 regularization, weight = 0.0005
Weight & Bias initiation	Gaussian {stdev: 0.01}	Function used to initiate weights and biases.
Loss function	SoftmaxWithLoss	Function used to minimize loss
Output function	Softmax	Output layer function
Number of output classes	2	Binary Classification

Each dataset was tested several times using different (train_testing)% splits repeated over 10 times

In total:

48 DNN configurations were tested

24 RF configurations

20 SVM configurations

6 NB configurations

The total Number of Calculations performed:

DNN: $7 \times 7 \times 10 \times \mathbf{48} = 23{,}520$

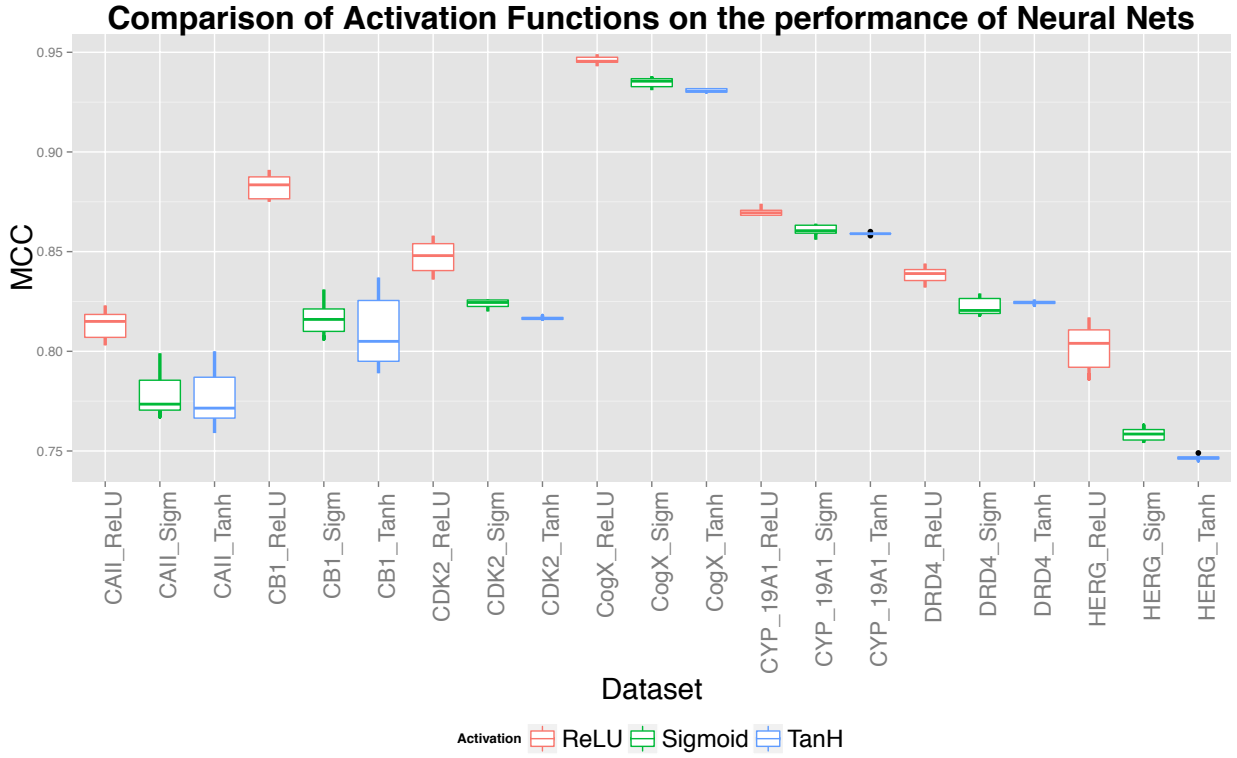
RF: $7 \times 7 \times 10 \times \mathbf{24} = 11{,}760$

SVM: $7 \times 7 \times 10 \times \mathbf{20} = 9{,}800$

NB: $7 \times 7 \times 10 \times \mathbf{6} = 2{,}940$

Koutsoukas A., Monaghan K.J., Li X. and Huan J. ***.Deep-Learning: Investigating feed-forward Deep Neural Networks hyper-parameters and Comparison of Performance to Shallow Methods for Modeling Bioactivity Data.*** **Journal of Chemical Informatics and Modeling (2016 submitted)**

Optimizing Deep Neural Nets: Comparing Neurons Activation Functions

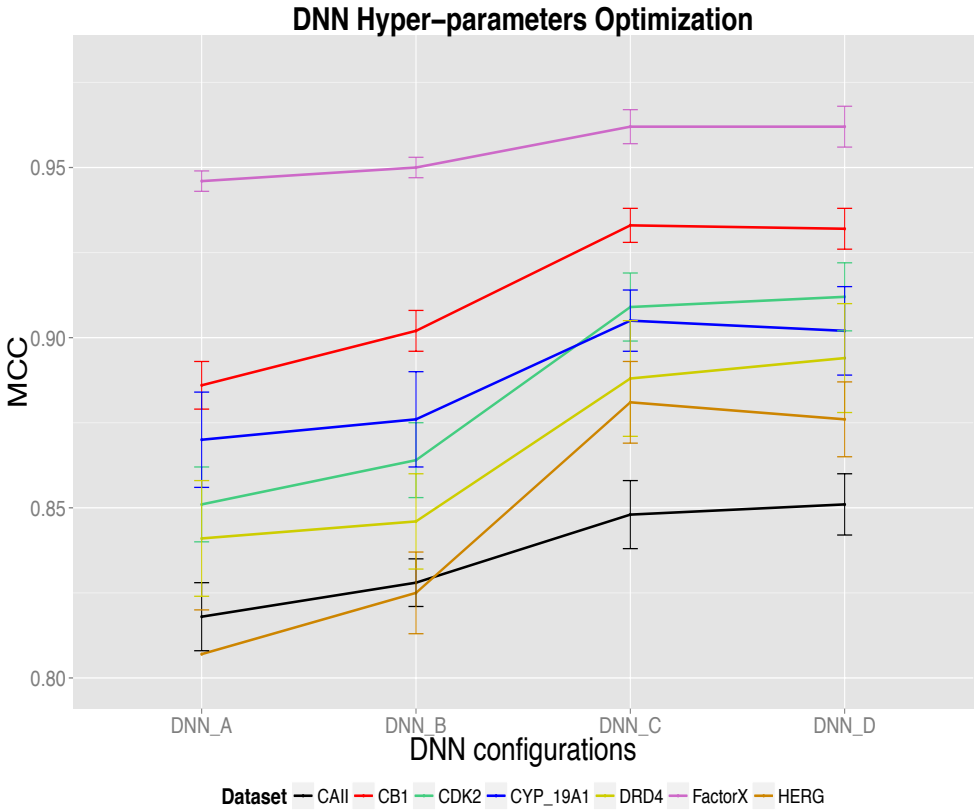


Matthews Correlation Coefficient (MCC) as evaluation metric:

$$MCC = \frac{TP \times TN - FP \times FN}{\sqrt{(TP + FP)(TP + FN)(TN + FP)(TN + FN)}}$$

Koutsoukas A., Monaghan K.J., Li X. and Huan J. *.Deep-Learning: Investigating feed-forward Deep Neural Networks hyper-parameters and Comparison of Performance to Shallow Methods for Modeling Bioactivity Data.* Journal of Chemical Informatics and Modeling (2016 submitted)

Optimizing Deep Neural Nets: Number of hidden layers, Number of Neuronsn and Regularization



DNN_A : Single hidden layer with 1,000 neurons and no regularization

DNN_B: Two hidden layers, with 1,500 neurons in each layer and no regularization.

DNN_C: Two hidden layers, with 2,000 neurons in each layer and drop-out 50% for the input and hidden layers.

DNN_D: Three hidden layers, 2,000 neurons in each layer 50% drop-out for the input and hidden layers.

Koutsoukas A., Monaghan K.J., Li X. and Huan J. ***.Deep-Learning: Investigating feed-forward Deep Neural Networks hyper-parameters and Comparison of Performance to Shallow Methods for Modeling Bioactivity Data.*** **Journal of Chemical Informatics and Modeling (2016 submitted)**

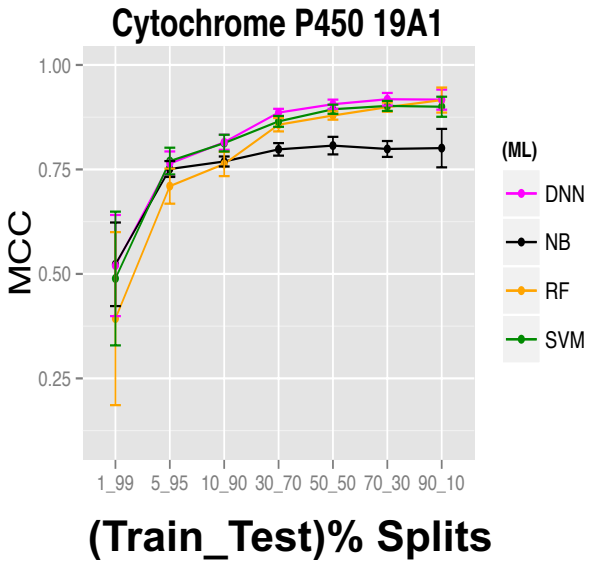
Example of DNN optimization process

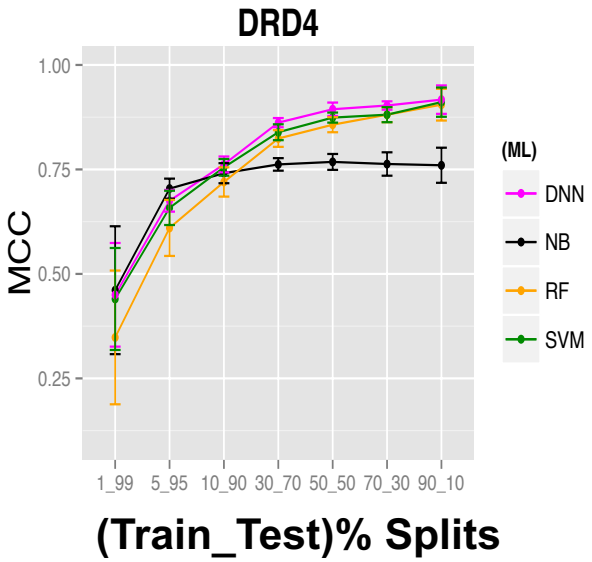
50_50

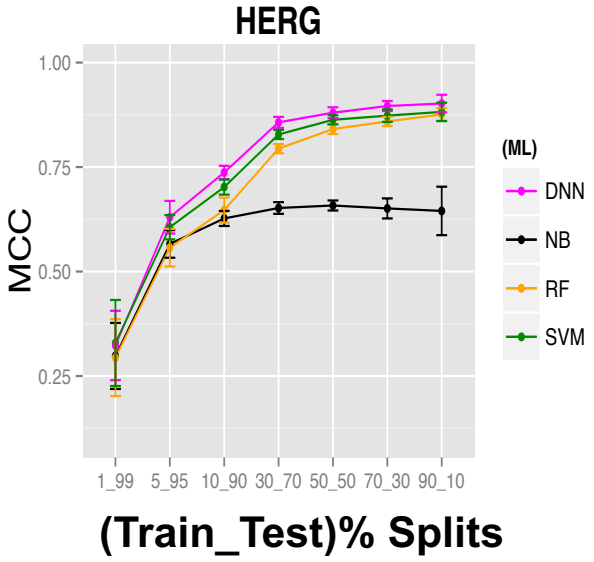
Epochs	Layers	Activation	Input dropout	Dropout	L2	Filler	AvAUC	Std_AUC	AvMCC	Std_MCC	AvFSCORE	Std_FSCORE	AvRecall	Std_Recall	AvPrecision	Std_Precision	AvAccuracy
200	[50]	ReLU	0	0	FALSE	Gaussian	0.964	0.006	0.789	0.012	0.808	0.011	0.810	0.028	0.807	0.020	0.965
200	[100]	ReLU	0	0	FALSE	Gaussian	0.964	0.006	0.786	0.012	0.805	0.011	0.807	0.030	0.804	0.018	0.965
200	[500]	ReLU	0	0	FALSE	Gaussian	0.966	0.005	0.801	0.012	0.819	0.011	0.812	0.030	0.826	0.016	0.968
200	[1000]	ReLU	0	0	FALSE	Gaussian	0.967	0.005	0.807	0.013	0.824	0.012	0.812	0.028	0.837	0.018	0.969
200	[1500]	ReLU	0	0	FALSE	Gaussian	0.968	0.005	0.812	0.014	0.828	0.013	0.813	0.029	0.845	0.016	0.970
200	[2000]	ReLU	0	0	FALSE	Gaussian	0.969	0.005	0.817	0.015	0.832	0.014	0.811	0.031	0.855	0.020	0.971
200	[50]	Sigmoid	0	0	FALSE	Gaussian	0.963	0.005	0.761	0.008	0.782	0.007	0.789	0.024	0.777	0.019	0.960
200	[100]	Sigmoid	0	0	FALSE	Gaussian	0.963	0.005	0.763	0.009	0.784	0.008	0.789	0.023	0.780	0.020	0.961
200	[500]	Sigmoid	0	0	FALSE	Gaussian	0.962	0.005	0.760	0.006	0.781	0.006	0.777	0.028	0.788	0.023	0.961
200	[1000]	Sigmoid	0	0	FALSE	Gaussian	0.962	0.005	0.755	0.012	0.776	0.012	0.772	0.038	0.783	0.031	0.960
200	[1500]	Sigmoid	0	0	FALSE	Gaussian	0.962	0.005	0.757	0.011	0.778	0.010	0.780	0.032	0.779	0.029	0.960
200	[2000]	Sigmoid	0	0	FALSE	Gaussian	0.962	0.005	0.754	0.014	0.774	0.014	0.774	0.048	0.780	0.041	0.959
200	[50]	TanH	0	0	FALSE	Gaussian	0.959	0.006	0.749	0.007	0.771	0.006	0.786	0.023	0.758	0.021	0.958
200	[100]	TanH	0	0	FALSE	Gaussian	0.958	0.006	0.747	0.009	0.770	0.008	0.786	0.023	0.756	0.022	0.958
200	[500]	TanH	0	0	FALSE	Gaussian	0.956	0.006	0.746	0.009	0.769	0.008	0.785	0.022	0.754	0.024	0.957
200	[1000]	TanH	0	0	FALSE	Gaussian	0.956	0.007	0.745	0.009	0.768	0.008	0.784	0.024	0.754	0.023	0.957
200	[1500]	TanH	0	0	FALSE	Gaussian	0.956	0.007	0.747	0.009	0.769	0.008	0.785	0.022	0.756	0.025	0.958
200	[2000]	TanH	0	0	FALSE	Gaussian	0.957	0.007	0.746	0.010	0.769	0.009	0.785	0.023	0.755	0.025	0.957
200	[50, 50]	ReLU	0	0	FALSE	Gaussian	0.963	0.006	0.792	0.014	0.810	0.013	0.806	0.028	0.815	0.019	0.966
200	[100, 100]	ReLU	0	0	FALSE	Gaussian	0.963	0.007	0.800	0.012	0.818	0.011	0.810	0.028	0.827	0.019	0.967
200	[500, 500]	ReLU	0	0	FALSE	Gaussian	0.966	0.006	0.820	0.012	0.836	0.012	0.819	0.026	0.854	0.019	0.971
200	[1000, 1000]	ReLU	0	0	FALSE	Gaussian	0.967	0.005	0.823	0.015	0.838	0.014	0.821	0.029	0.858	0.020	0.972
200	[1500, 1500]	ReLU	0	0	FALSE	Gaussian	0.968	0.005	0.825	0.012	0.840	0.012	0.817	0.029	0.865	0.020	0.972
200	[2000, 2000]	ReLU	0	0	FALSE	Gaussian	0.969	0.005	0.829	0.012	0.843	0.012	0.817	0.028	0.872	0.019	0.973
200	[50, 50]	ReLU	0.5	0.5	FALSE	Gaussian	0.975	0.003	0.866	0.011	0.875	0.012	0.817	0.025	0.942	0.014	0.979
200	[100, 100]	ReLU	0.5	0.5	FALSE	Gaussian	0.976	0.003	0.872	0.012	0.882	0.012	0.834	0.017	0.935	0.008	0.980
200	[500, 500]	ReLU	0.5	0.5	FALSE	Gaussian	0.976	0.003	0.880	0.013	0.889	0.013	0.849	0.025	0.933	0.012	0.981
200	[1000, 1000]	ReLU	0.5	0.5	FALSE	Gaussian	0.977	0.004	0.878	0.012	0.888	0.011	0.857	0.022	0.921	0.015	0.980
200	[1500, 1500]	ReLU	0.5	0.5	FALSE	Gaussian	0.976	0.003	0.878	0.012	0.887	0.011	0.855	0.019	0.923	0.018	0.980
200	**[2000, 2000]**	**ReLU**	**0.5**	**0.5**	**FALSE**	**Gaussian**	**0.977**	**0.003**	**0.881**	**0.012**	**0.890**	**0.011**	**0.853**	**0.020**	**0.931**	**0.014**	**0.981**
200	[50, 50]	ReLU	0	0	TRUE	Gaussian	0.970	0.004	0.821	0.017	0.836	0.016	0.810	0.031	0.865	0.019	0.971
200	[100, 100]	ReLU	0	0	TRUE	Gaussian	0.970	0.005	0.822	0.015	0.836	0.015	0.810	0.026	0.865	0.022	0.971
200	[500, 500]	ReLU	0	0	TRUE	Gaussian	0.971	0.004	0.831	0.014	0.845	0.013	0.815	0.027	0.878	0.024	0.973
200	[1000, 1000]	ReLU	0	0	TRUE	Gaussian	0.971	0.004	0.828	0.015	0.842	0.014	0.814	0.030	0.873	0.024	0.972
200	[1500, 1500]	ReLU	0	0	TRUE	Gaussian	0.971	0.004	0.829	0.014	0.843	0.013	0.812	0.030	0.877	0.021	0.973
200	[2000, 2000]	ReLU	0	0	TRUE	Gaussian	0.968	0.005	0.830	0.012	0.844	0.012	0.817	0.029	0.873	0.019	0.973
200	[50, 50, 50]	ReLU	0.5	0.5	FALSE	Gaussian	0.954	0.066	0.763	0.255	0.770	0.257	0.700	0.234	0.856	0.286	0.969
200	[100, 100, 100]	ReLU	0.5	0.5	FALSE	Gaussian	0.976	0.003	0.864	0.014	0.872	0.014	0.810	0.028	0.946	0.012	0.979
200	[500, 500, 500]	ReLU	0.5	0.5	FALSE	Gaussian	0.975	0.005	0.876	0.013	0.885	0.012	0.834	0.023	0.943	0.010	0.980
200	[1000, 1000, 1000]	ReLU	0.5	0.5	FALSE	Gaussian	0.977	0.004	0.877	0.010	0.886	0.009	0.842	0.016	0.936	0.010	0.981
200	**[1500, 1500, 1500]**	**ReLU**	**0.5**	**0.5**	**FALSE**	**Gaussian**	**0.976**	**0.004**	**0.881**	**0.011**	**0.889**	**0.011**	**0.846**	**0.020**	**0.938**	**0.012**	**0.981**
200	[2000, 2000, 2000]	ReLU	0.5	0.5	FALSE	Gaussian	0.976	0.004	0.876	0.011	0.885	0.011	0.838	0.020	0.938	0.008	0.980
200	[50, 50, 50]	ReLU	0	0	TRUE	Gaussian	0.561	0.127	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.000	0.910
200	[100, 100, 100]	ReLU	0	0	TRUE	Gaussian	0.864	0.185	0.578	0.378	0.588	0.385	0.567	0.372	0.611	0.400	0.953
200	[500, 500, 500]	ReLU	0	0	TRUE	Gaussian	0.972	0.004	0.835	0.013	0.848	0.012	0.816	0.026	0.883	0.020	0.974
200	[1000, 1000, 1000]	ReLU	0	0	TRUE	Gaussian	0.972	0.004	0.836	0.015	0.849	0.014	0.814	0.029	0.888	0.021	0.974
200	[1500, 1500, 1500]	ReLU	0	0	TRUE	Gaussian	0.972	0.004	0.836	0.013	0.849	0.012	0.817	0.031	0.884	0.021	0.974
200	[2000, 2000, 2000]	ReLU	0	0	TRUE	Gaussian	0.972	0.004	0.835	0.014	0.848	0.013	0.814	0.029	0.886	0.021	0.974

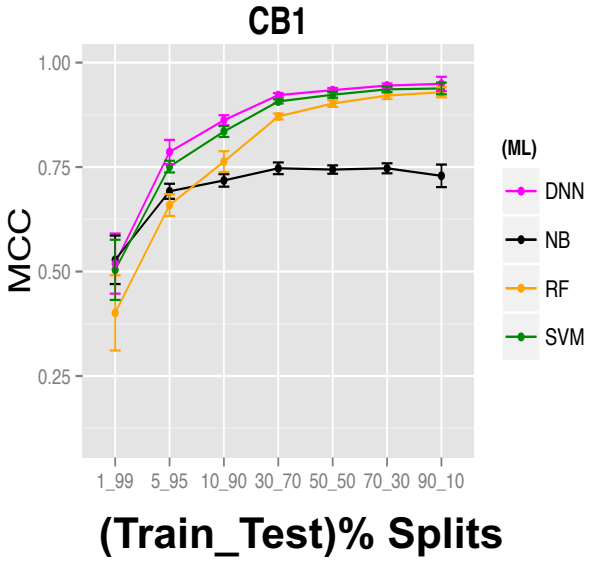
HERG dataset: Results obtained by 48 DNN configurations

Comparison of performance of DNN to Shallow methods: RF, SVM and NB





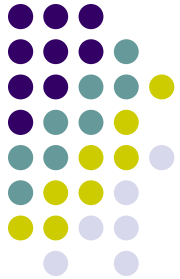




DNN Improves Shallow Methods with Statistical Significance

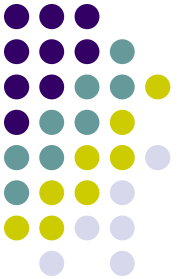
Cannabinoid CB1 receptor	**Mean of MCC diff.**	**Stdev of diff.**	**p-value**
DNN-NB	0.1909	0.0087	9.766e-04
DNN-RF	0.0327	0.008	9.766e-04
DNN-SVM	0.012	0.0037	2.945e-04
Cyclin-dependent kinase 2	**Mean of MCC diff.**	**Stdev of diff.**	**p-value**
DNN-NB	0.1675	0.0146	9.766e-04
DNN-RF	0.0132	0.0078	4e-03
DNN-SVM	0.0175	0.0054	2.913e-03
Dopamine D4 Receptor	**Mean of MCC diff.**	**Stdev of diff.**	**p-value**
DNN-NB	0.1229	0.019	2.945e-03
DNN-RF	0.0337	0.012	9.766e-04
DNN-SVM	0.0169	0.008	9.766e-04
Cytochrome P450 19A1	**Mean of MCC diff.**	**Stdev of diff.**	**p-value**
DNN-NB	0.0994	0.0206	2.945e-04
DNN-RF	0.027	0.01439	2.897e-03
DNN-SVM	0.0176	0.0109	2.945-03
HERG	**Mean of MCC diff.**	**Stdev of diff.**	**p-value**
DNN-NB	0.2229	0.0085	2.929e-03
DNN-RF	0.0394	0.0086	2.897e-03
DNN-SVM	0.018	0.0095	2.945e-03
Coagulation factor X	**Mean of MCC diff.**	**Stdev of diff.**	**p-value**
DNN-NB	0.0947	0.0146	2.929e-03
DNN-RF	0.0061	0.0055	8.267e-03
DNN-SVM	0.0014	0.0055	0.256
Carbonic Anhydrase II	**Mean of MCC diff.**	**Stdev of diff.**	**p-value**
DNN-NB	0.0854	0.0051	2.929e-03
DNN-RF	0.0336	0.0086	9.766e-04
DNN-SVM	0.0051	0.0055	1.405e-02

Quick Wrap-up



- **Rectified Linear Units (ReLU)** is the best activation function
- **2 hidden layers** were found to be sufficient.
- **Dropout regularization** technique helps significantly
- Optimal number of neurons per hidden layer varies between 1000 to 2000.

Software Release



- https://gitlab.ittc.ku.edu/xiaolili/WOLF
- Project website http://pcc.ittc.ku.edu/wolf/

Acknowledgement

- University of Kansas
- National Science Foundation