Taming WOLF: Building a More Functional and User-Friendly Framework

Casey Sader

University of Kansas, EECS

OUTLINE

- Introduction
- Objective and Motivation
- Background and Related Tools
- Contributions
- Demonstration
- Future Work
- Conclusion

INTRODUCTION

- WOLF created by Pranav Bahl in 2016
- Select the best hyper-parameters for a model
- Goal is to make WOLF more functional and user-friendly
- Novice user is the target audience

OBJECTIVE

- Improve the functionality of the existing framework
- Create a more user-friendly framework
- Website
- Modern technology – neural network
- Save trained models
- Make predictions
- Feature importance
- Datasets

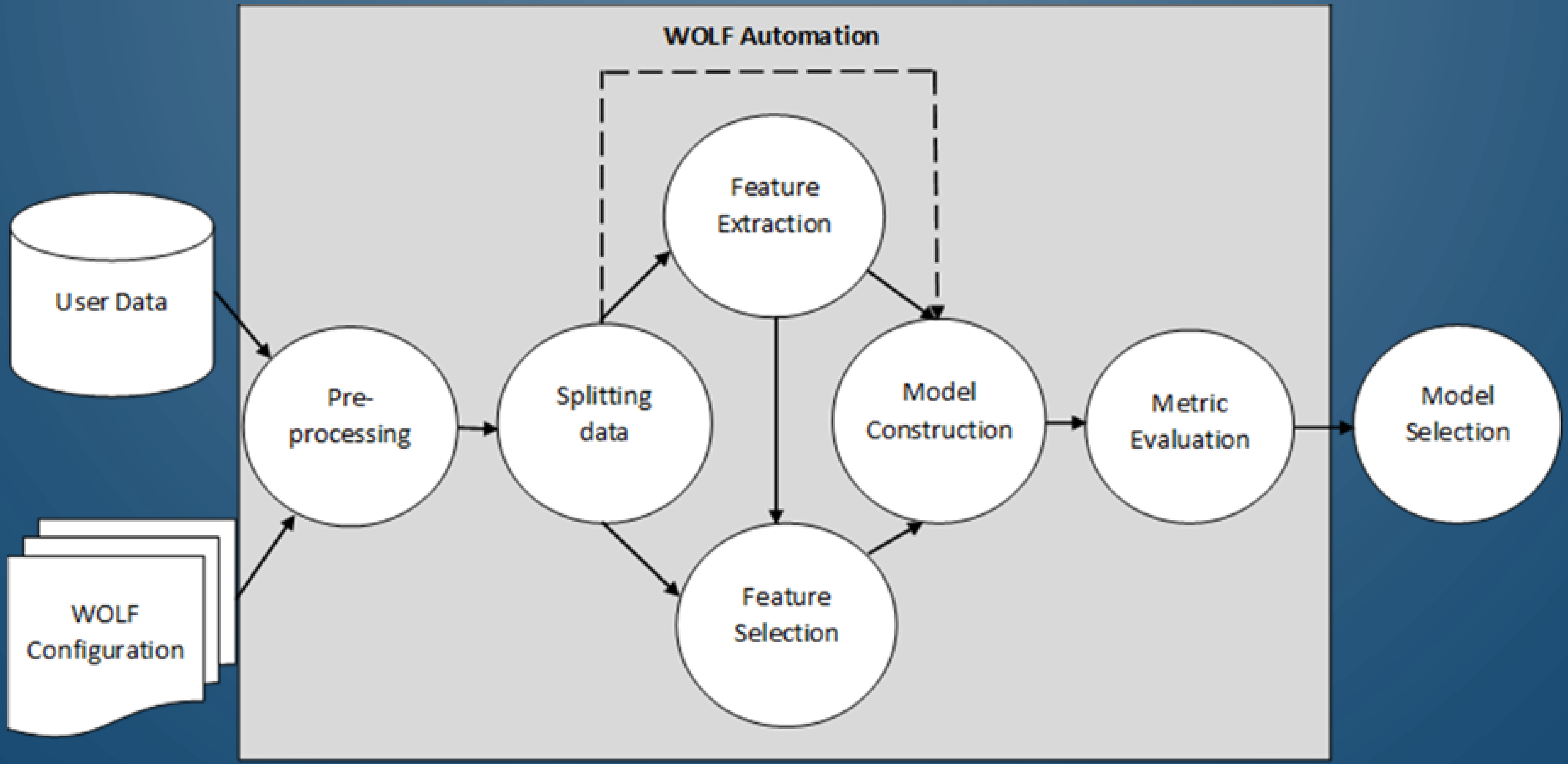
MOTIVATION

- Automate tasks in the machine learning pipeline
- WOLF missing key tasks in the pipeline
- Make WOLF more accessible
- TensorFlow
- Provide a trained model to the user
- Make predictions within the framework
- Understanding of datasets and feature importance
- Provide benchmark datasets

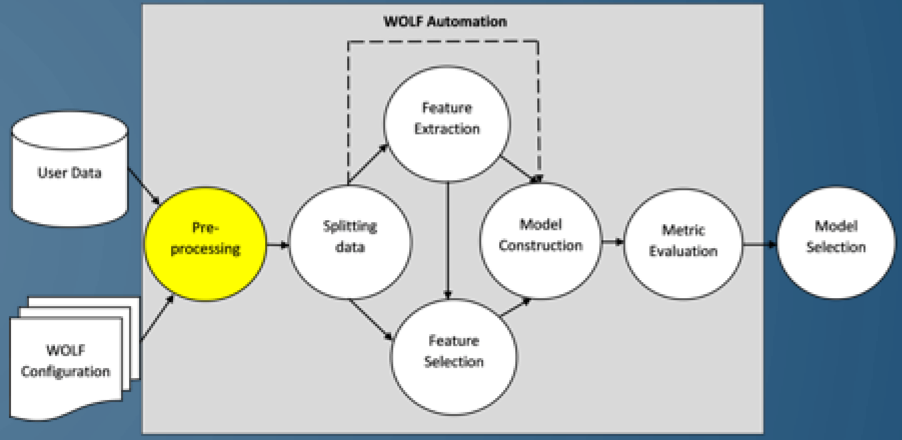
WOLF BACKGROUND

- Allows a user to control each step of the machine learning pipeline
- Each task is called a “transaction”
- Control the pipeline with a configuration file (`yaml` file)
- Select dataset, transactions, and how to run each transaction
- Models implemented using Scikit-Learn
- Possible transactions are pre-processing, data splitting, feature extraction, feature selection, model construction, metric evaluation, and model selection

WOLF BACKGROUND

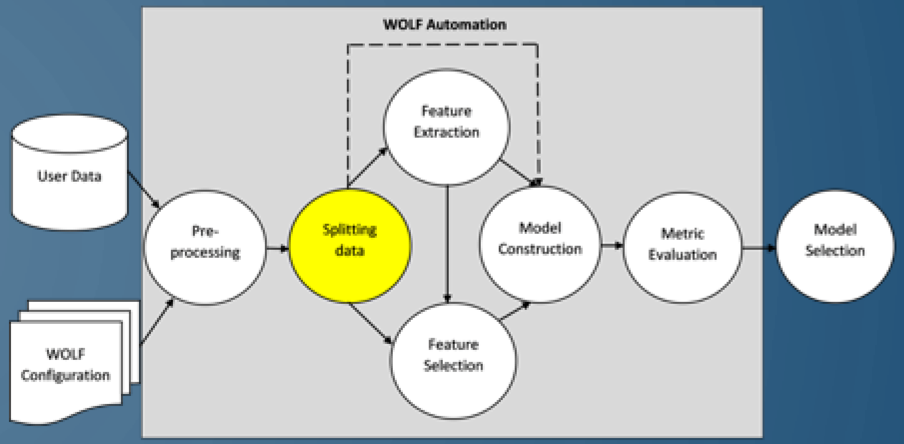


WOLF BACKGROUND



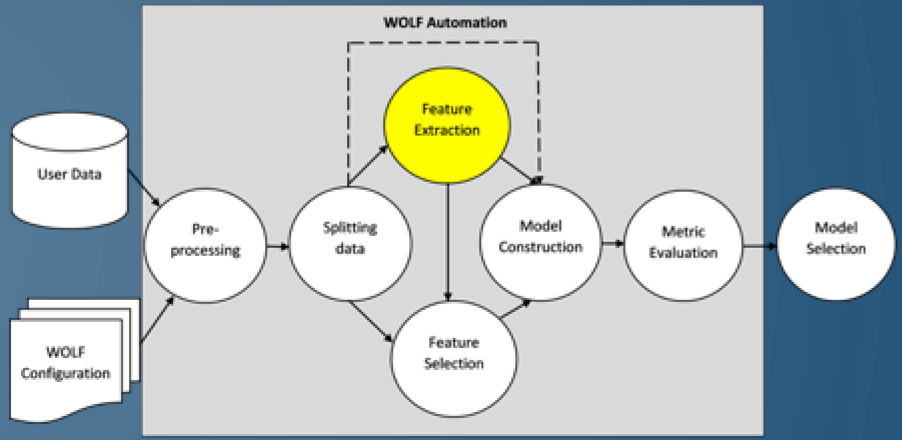
- Pre-processing
  - Performs any step needed to make the dataset complete for WOLF

WOLF BACKGROUND



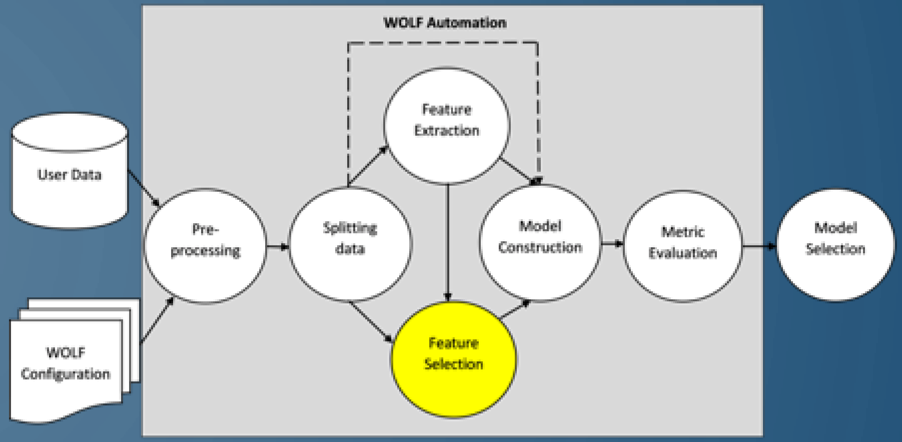
- Pre-processing
- Splitting Data
  - Splits the data into train and test files

WOLF BACKGROUND



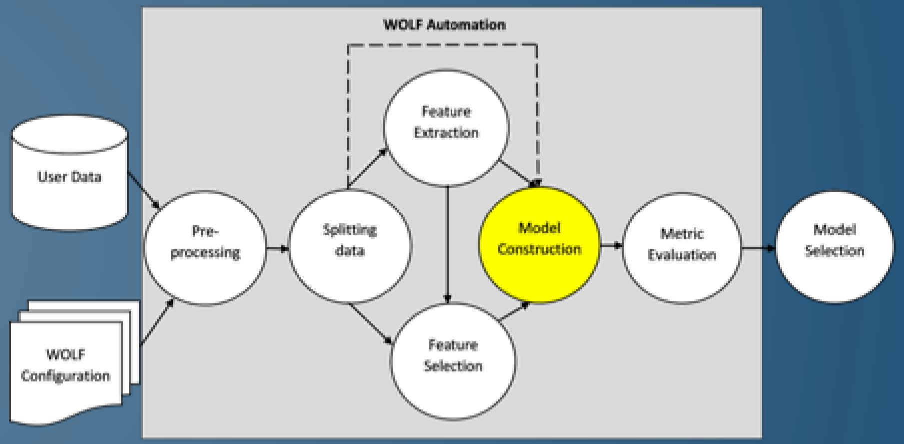
- Pre-processing
- Splitting Data
- Feature Extraction
  - Perform dimensionality reduction

WOLF BACKGROUND



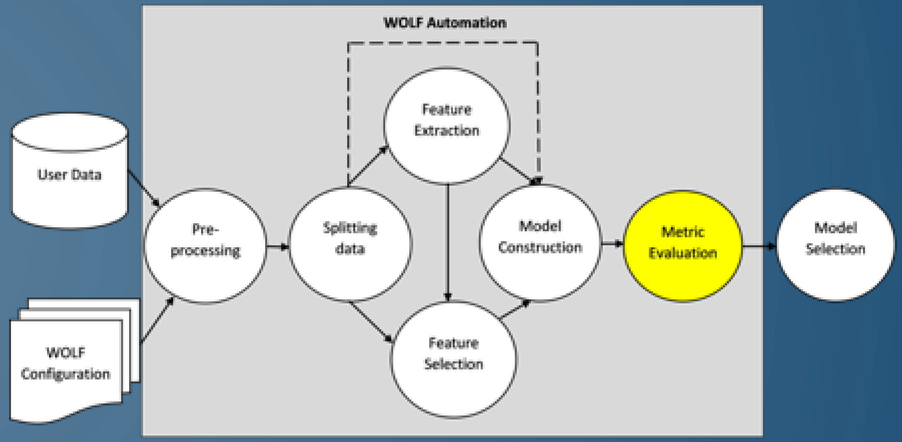
- Pre-processing
- Splitting Data
- Feature Extraction
- Feature Selection
    - Selects features that are noisy or redundant and removes them

WOLF BACKGROUND



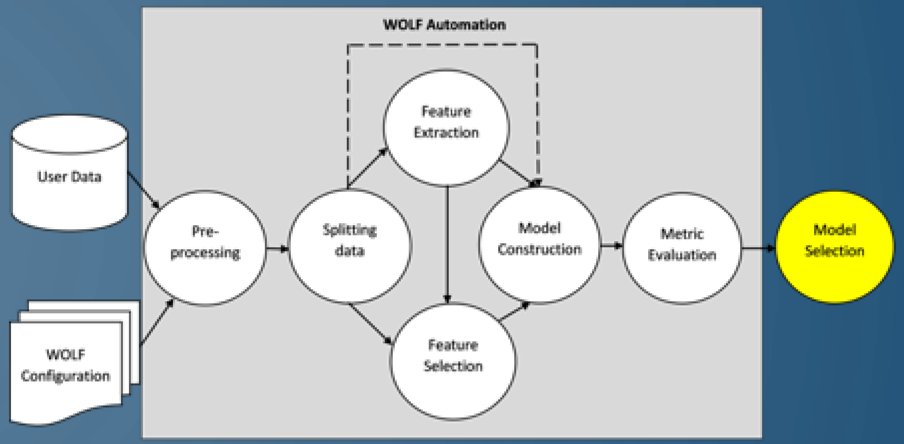
- Pre-processing
- Splitting Data
- Feature Extraction
- Feature Selection
- Model Construction
  - For each machine learning model type and each combination of hyper-parameters that are to be tested on, a model is trained on every train/test set combination

WOLF BACKGROUND



- Pre-processing
- Splitting Data
- Feature Extraction
- Feature Selection
- Model Construction
- Metric Evaluation
  - Accuracy, precision, ROC-AUC, MCC, and F1-Score

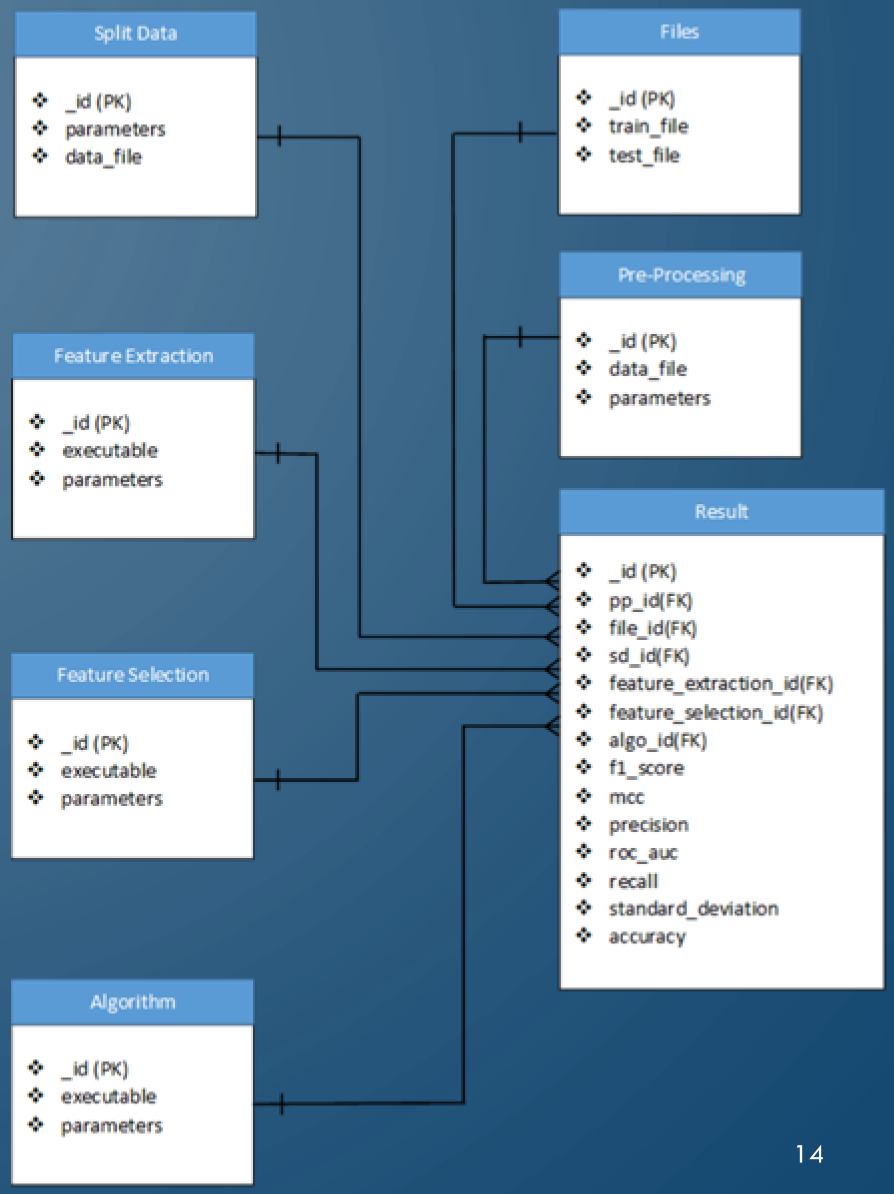
WOLF BACKGROUND



- Pre-processing
- Splitting Data
- Feature Extraction
- Feature Selection
- Model Construction
- Metric Evaluation
- Model Selection
  - Determines the best model and hyper-parameter combination

WOLF BACKGROUND

- MongoDB used to store dataflow
- Every configuration and parameter is stored along with an id value to keep track of each run



WOLF MODELS

- Ada Boost Classifier
- Bernoulli Naïve Bayes
- Gaussian Naïve Bayes
- Decision Tree
- Logistic Regression
- Random Forest
- C-SVM
- Linear SVM
- Nu SVM
- Linear Discriminant Analysis
- Quadratic Discriminant Analysis
- Neural Network (using Caffe)

RELATED TOOLS

- **Auto-WEKA**
  - 2013 at the University of British Columbia
  - Classification using 39 different algorithms
  - Bayesian-optimization used to search hyper-parameters
- **Michelangelo**
  - Created by Uber to perform the machine learning workflow from beginning to end
  - Six steps: “managing data, training models, evaluating models, deploying models, making predictions, and monitoring predictions”
  - Can run on the massive scale needed for Uber

CONTRIBUTIONS

- Website
- Neural network (TensorFlow)
- Saved model and predictions
- Feature importance
- Datasets

WEBSITE (OVERVIEW)

- Allow runs of WOLF for non-ITTC members
- Implement all of the features of the command line version
- Keep the runs of all users separate
- Make WOLF more user-friendly and intuitive
- Work compiled for 2017 IEEE International Conference on Big Data

WEBSITE (MY INDIVIDUAL WORK)

- Contributed to planning out the goals and what a user should expect
- Decided file structure, a project timeline, and the visual layout
- Created and implemented the initial setup of the SQL database
- Enabled user to view and run their own workflows

NEURAL NETWORK

- Deep learning in WOLF used Caffe when this project began
- Caffe no longer the most popular framework for neural networks
- Top two are PyTorch and TensorFlow

NEURAL NETWORK

- PyTorch
  - Framework built using Torch
  - Useful for creating neural networks that are reusable
  - “Reverse-mode auto-differentiation” allows changes to models with little overhead
- TensorFlow
  - Created by Google to perform computations on CPUs and GPUs (and TPUs)
  - Takes advantage of the flow graphs of tensors
  - Can be used for the end-to-end machine learning pipeline
  - Chosen for its large market share and the value it would have

NEURAL NETWORK

- Keras API used for high-level implementation of neural networks
- Runs on top of TensorFlow
- Keras is the second most popular framework on its own
- API function calls similar to Scikit-Learn so the transition in syntax was simpler
- Runs on GPU

NEURAL NETWORK

Parameter flag	Description	Default value
-a	activation function	"relu"
-l	layers	[100,100,100]
-d	input layer dropout	0
-h	hidden layer dropout	0.5
-e	epochs	10
-b	batch size	None
-r	learning rate	0.001

SAVED MODELS

- WOLF was lacking the ability to return a trained model to the user
- WOLF creates a new model for each train/test split pair
- Saving the model is performed using `pickle`
- Each model is stored in the designated folder for the hyper-parameters
- The full path of the best pickled model is given to the user in the “best configuration” tab of the results excel file
- By default, the best model has the highest ROC-AUC score

PREDICTIONS

- Prediction transaction now available in configuration file
- The user will provide the dataset to predict on and the model to use
- Optional parameters: feature to predict, whether to calculate accuracy or not
- The data is read in and put into the proper format for the model
- The pickled model is also loaded in
- Predictions are written to a file which the user can use

FEATURE IMPORTANCE

- Relative importance each feature has on the prediction outcome of a model
- Calculated from the model itself rather than the dataset (as feature extraction and selection attempt to do)
- Feature importances are calculated, sorted, and written to a file
- The number of files is the same as the number of models
- The feature importances of the best model are also in the results excel file

FEATURE IMPORTANCE

- Random forest, decision tree, and ada boost use the built-in attribute `feature_importances_` from Scikit-Learn
- Most other models use the coefficient matrix from Scikit-Learn
- Using Keras, there isn't an attribute or function to call
- Neural networks are not inherently transparent or interpretable
- "Leave one out" strategy
- ELI5

RESULTS FILE

accuracy	algo_name	algo_param	f1_score	mcc	precision	recall	roc_auc	model path
0.7629081	RandomForest	{'c': 'gini', 'd': '10', 'm': '0', 'l': '1', 'p': '1e-7', 's': '2', 't': '50', 'w': 'False'}	0.6306859	0.4634382	0.69188302	0.58287212	0.7211361	/WOLF_CL/output/Splittingdata1/RandomForest_result1/models/RandomForestModel1.pkl

name	importance
plas	0.368909207
mass	0.147325008
age	0.130517471
pedi	0.094459587
insu	0.076882703
preg	0.074408122
skin	0.056018316
pres	0.051479585

model path
/WOLF_CL/output/Splittingdata1/RandomForest_result1/models/RandomForestModel1.pkl

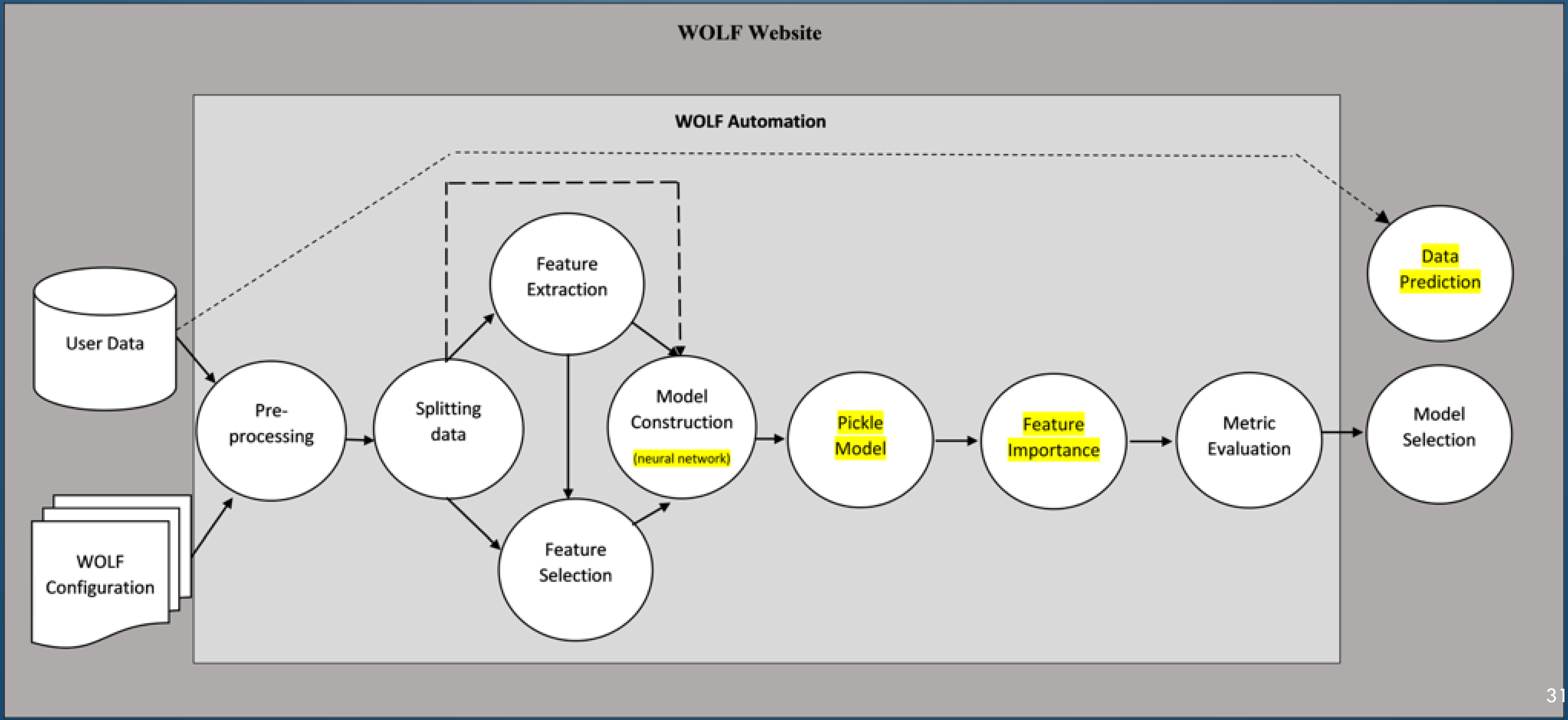
DATASETS

- Used to test the effectiveness of WOLF
- From UCI Machine Learning repository
- Downloaded and converted to arff format
- Datasets are both binary classification and multi-class classification

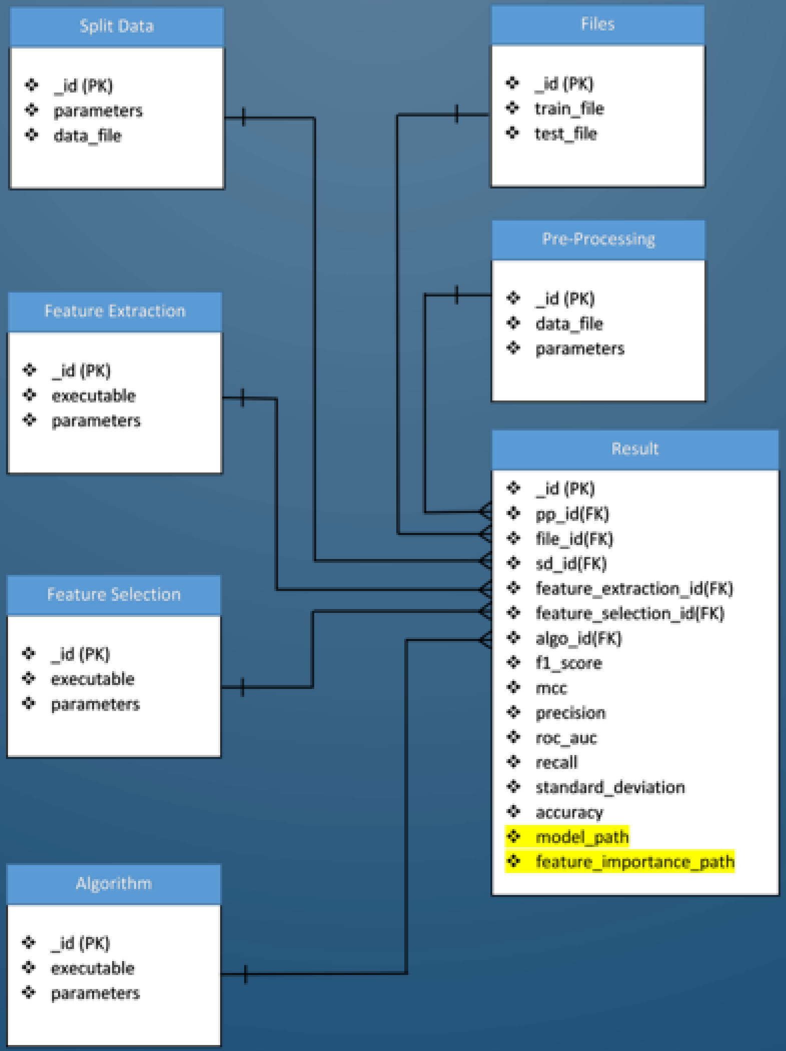
DATASETS

Dataset Name	RF	DT	Lin SVM	Log Reg	BNB	LDA	Ada Boost	NN - Caffe	NN - TF
Bank note auth.	0.9923	0.9810	0.9889	0.9896	0.8419	0.9786	0.996	0.9998	0.9999
Blood Transfusion	0.6279	0.5852	0.5323	0.5495	0.4993	0.5419	0.6181	0.5003	0.5427
Climate Sim. Crashes	0.5501	0.6527	0.7941	0.5773	0.5000	0.7158	0.7535	0.6581	0.7570
Sonar, Mines/Rocks	0.8167	0.6919	0.7692	0.7507	0.5051	0.7358	0.7954	0.6100	0.7662
Default of credit card	0.6540	0.6081	0.5217	0.4999	0.6731	0.6127	0.6388	0.5000	0.5000
Fertility	0.5531	0.4964	0.4960	0.4988	0.5000	0.4878	0.5375	0.5223	0.5333
Voice Rehabilitation	0.7831	0.7351	0.5058	0.5529	0.6854	0.7256	0.7916	0.6344	0.4934
Pima Indians Diabetes	0.7216	0.6412	0.5597	0.7151	0.5035	0.7253	0.7131	0.6864	0.7324
Spambase	0.9323	0.9025	0.8252	0.9215	0.8736	0.8699	0.9345	0.6706	0.8887
Vertebral Column	0.8058	0.7592	0.7261	0.8095	0.6438	0.8017	0.7917	0.8101	0.7619
Wholesale customers	0.9053	0.8520	0.6939	0.8765	0.5000	0.7748	0.8800	0.5000	0.5348

NEW ARCHITECTURE



NEW ARCHITECTURE



FUTURE WORK

- Continued website work (e.g., display results)
- Improvements to feature importance addition
- Image data
- More model types (e.g., regression)
- GPU reservation
- Command line use off of ITTC cluster
- Port to Python 3

Demo

FINAL THOUGHTS

- Learned about writing code that meets industry standards
- Important to be able to add to frameworks and tools easily
- Project can be a guide for future students to continue improving WOLF

PUBLICATIONS

- Sohaib Kiani, Xiaoli Li, Pranav Bahl, Casey Sader, and Jun Huan. WOLF: Machine Learning Workflow Management Framework. 2017.
- Xiaoli Li, Sohaib Kiani, Pranav Bahl, Casey Sader, and Jun Huan. WOLF: Machine Learning WOrkfLow Management Framework. Boston, MA, 2017. url: http://cci.drexel.edu/bigdata/bigdata2017/files/Tutorial3.pdf.

Thank You!

Questions?