towardsdatascience.com

Automated Machine Learning

Justin Tennenbaum

5-7 minutes

Automated Machine Learning(AutoML) is currently one of the explosive subfields within Data Science. It sounds great for those who are not fluent in machine learning and terrifying for current Data Scientists. The way AutoML has been portrayed in the media makes it seem capable of completely revolutionizing the way we create models by removing the need for Data Scientists. While some companies such as DataRobot aim to fully automate the machine learning process, most in the field are creating AutoML as a tool to increase the production of current Data Scientists, and simplify the process for those entering the field to make it more accessible.

AutoML as a tool to fully automate the process is a great idea on paper, but in the real world it introduces many opportunities for bias and misunderstanding. In the past few years, the machine learning field has began to diverge away from “black-box” models, and instead use simpler models that are easier to interpret. Complex models can be hard to decipher and because of this it is hard to know when a model is introducing bias. AutoML now exacerbates this problem of a black box model, by hiding not only the

mathematics of the model, but also performing the following in the background:

- data cleaning
- feature selection
- model selection
- parameter selection.

With all of the above automated what does that leave for the people using these systems. Other than grabbing a dataset and reading the results, nothing. This level of automation introduces potential problems for both our models and the people interpreting them. While studying to be a Data Scientist three things have become prevalent above all else.

1. Models are only as good as the data are given and can easily introduce bias if the data is flawed.
2. Machines are very good at maximizing the objective function given by a human, whether or not that objective function is accurate or correct.
3. A large portion of studying for Data Science is dedicated to understanding the model's algorithm and its results, without this knowledge it is easy to make incorrect inferences about the models results.

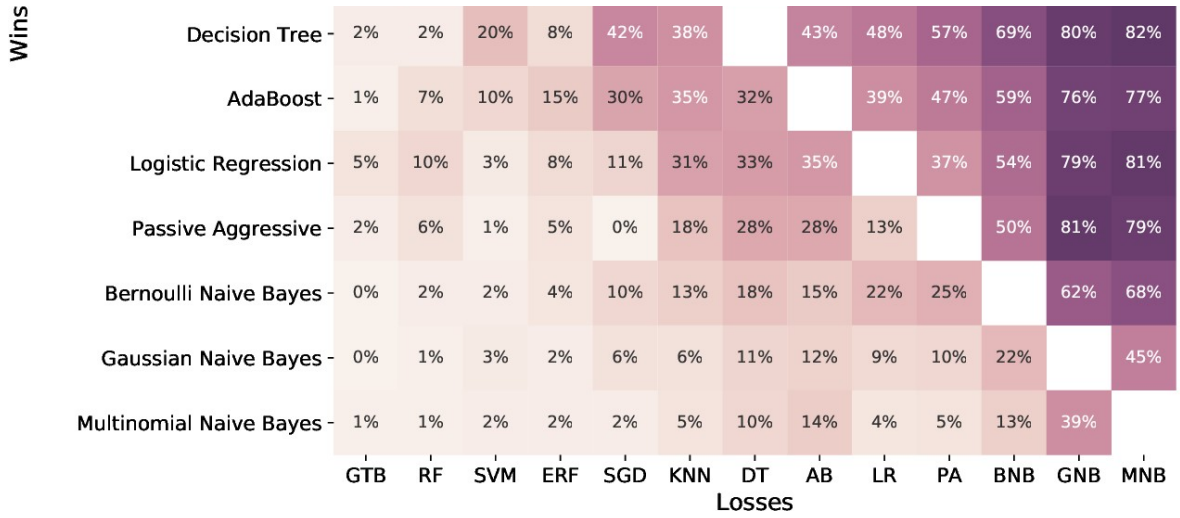
The past few years we have seen multiple reports of models in circulation that continually make incorrect/biased predictions. If there is bias in the real world it can easily be increased by not accounting for it properly in the model. AutoML might cause more harm than good if we use it to fully automate the process. However, full automation is only the goal of a handful of researchers of

companies and there is another side which might benefit the community greatly.

I watched Randall Olsen's talk at Scipy 2018, The Past,Present, and Future of Automated Machine Learning. In his talk he emphasizes that AutoML is not there to replace the Data Scientist, but like I said above, is here to improve the productivity of current data scientists and make the field more accessible to those who have not been studying these algorithms for years.

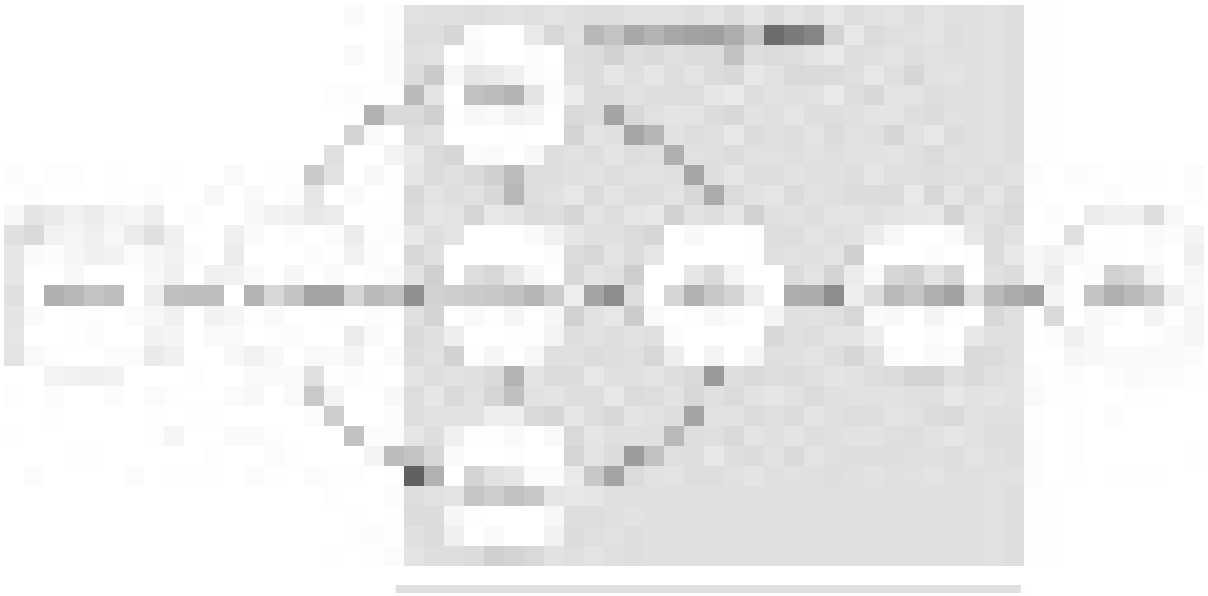
% out of 165 datasets where model A outperformed model B

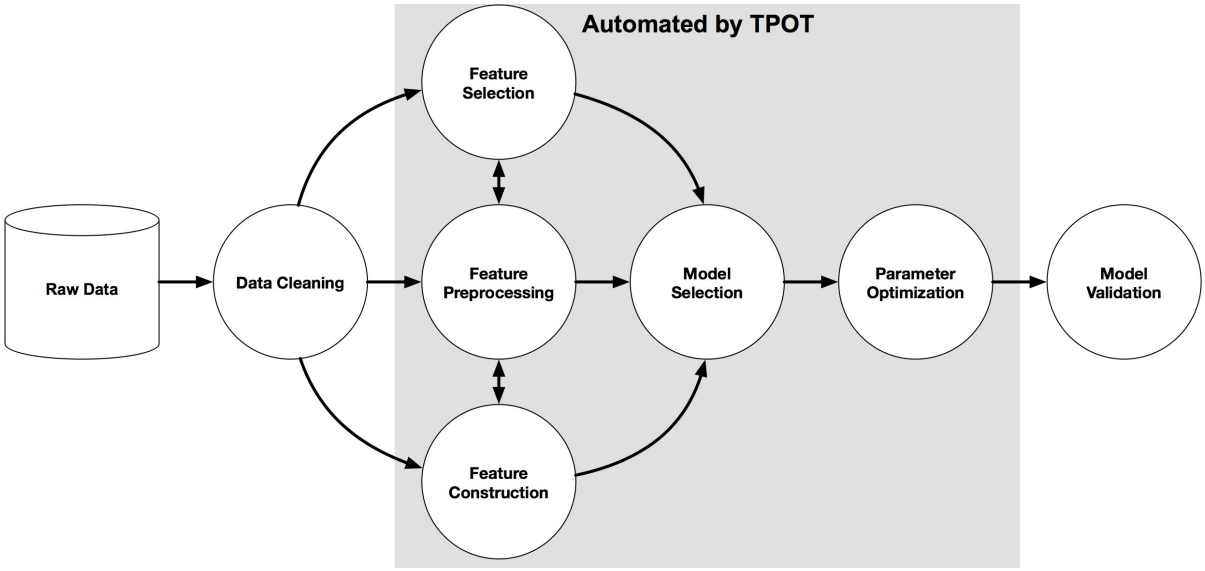
Gradient Tree Boosting		32%	45%	38%	67%	72%	78%	76%	78%	82%	90%	95%	95%
Random Forest	9%		33%	23%	62%	65%	71%	69%	71%	76%	85%	95%	90%
Support Vector Machine	12%	21%		25%	55%	65%	56%	62%	67%	74%	79%	95%	93%
Extra Random Forest	8%	14%	30%		58%	63%	61%	64%	67%	70%	81%	93%	91%
Linear Model trained via Stochastic Gradient Descent	8%	16%	9%	15%		38%	41%	44%	41%	61%	66%	89%	87%
K-Nearest Neighbors	4%	8%	7%	8%	35%		42%	45%	52%	53%	70%	88%	85%



Source: SemanticsScholar.org

As a Data Scientist I could use these applications to quickly overview my data and understand how I might begin the approach to modeling it. The AutoML wouldn't tell me which model to use, but instead lets me know what models and techniques are effective, saving me time spent on Exploratory Data Analysis and Data Cleaning that instead can be spent on fine tuning the model. One example is TPOT, created by Epistasis Labs, which is a AutoML library in Python built on top of Scikit-Learn(a very popular machine learning library). Its design is meant to quickly run analysis of your data and let you know which models, features, and parameters are more effective than others.





Source: https://epistasislab.github.io/tpot/

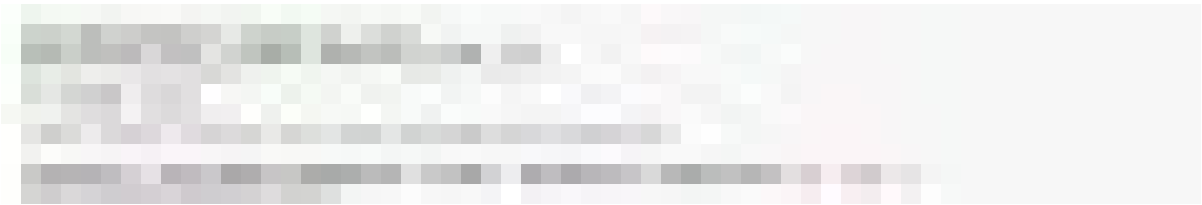
TPOT should be thought of as a place to begin or a model comparison and not as a final product. It opens up ideas and techniques to approach our current dataset in ways we might not have thought of or have had the time to explore deeply. As one can imagine, these algorithms have a very complex sample space/model and it only gets more complicated the bigger a dataset and the more complex the algorithms used. Very high level algorithms are used to navigate this sample space including:

- Meta-Learning
- Bayesian Optimization
- Genetic Programming
- Multi-Arm Bandits

Because they are so complex their runtime is extremely long and can take multiple days depending on the problem. This just reinforces the idea that AutoML will be used as a tool to help data scientists, since these algorithms can run in the background while we work on other aspects of a project. In the next few years these tools will continue to improve, but they won't be replacing Data

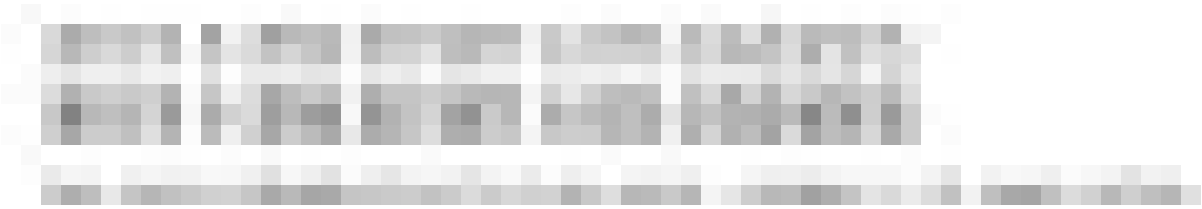
Scientists. Instead they will enhance what we can accomplish and allow machine learning to grow as the field becomes more and more accessible.

I wanted to see how these AutoML libraries ran, so I followed along with the code in Randall Olsen's talk, and ran TPOT on the MNIST dataset, which is a dataset of images containing numbers. It only took a 9 lines of code and I was able to achieve an accuracy rate of 98%.



```
from tpot import TPOTClassifier
from sklearn.datasets import load_digits
from sklearn.model_selection import train_test_split
digits = load_digits()
X = digits['data']
y = digits['target']
X_train,X_test,y_train,y_test = train_test_split(X,y,stratify=y)

classifier = TPOTClassifier(verbosity=2,n_jobs=-1,generations=5,config_dict='TPOT light')
classifier.fit(X_train,y_train)
```



```
Generation 1 - Current best internal CV score: 0.9747317909673437
Generation 2 - Current best internal CV score: 0.9784795011854163
Generation 3 - Current best internal CV score: 0.9829269992494323
Generation 4 - Current best internal CV score: 0.9829269992494323
Generation 5 - Current best internal CV score: 0.9888811346212986

Best pipeline: KNeighborsClassifier(input_matrix, n_neighbors=1, p=2, weights=uniform)
```

TPOT was able to determine that K-Nearest Neighbors using Euclidean Distance and K =1 was the optimal algorithm to classify the MNIST dataset. While this approach was quick, easy, and accurate, there was no need to understand what it was doing or what it was telling us. And by losing that, we lose all interpretation of the model and our ability to determine why we get the numbers we do.

Works Cited:

https://www.youtube.com/watch?v=QrJlj0VCHy