TU 257 – Fundamentals of Data Science

Data Analytics

L7 – Tuning & AutoML

Brendan Tierney

# Agenda

* Model Tuning
* Automating the process
* --- everything above and before for Assessment-A
* --- everything from now is for Assessment-B
* AutoML

# Model Tuning Previous Examples

* Our Previous Examples all used the default settings



* Each Algorithm has its own settings
* These parameters are often called Hyperparameters
* Lots of testing and Experiments have worked out the best settings to use.
* These work for most cases/scenarios
* But may not work best for all cases/scenarios
* Does not give the best or optimal model

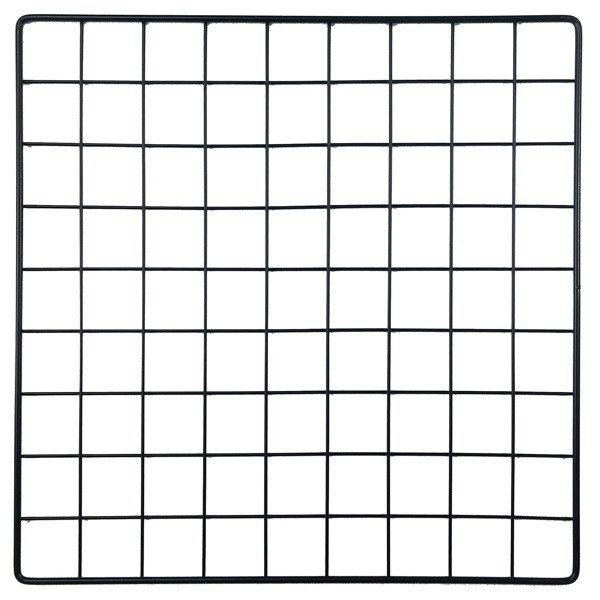
# Model Tuning

* Model Tuning is the process where you try to optimize the model
* By modifying the parameters
* To give a better / more accurate model
* To get better predictions on new data
* Why is this important
* Minor changes can have a big impact
* On € / $ Profit / Loss
* Or reduce fraud / breakages / better health predictions, etc
* Experimentation is needed
* Evaluate the results to see if they are really useful
* How do you know if it works?

# Model Tuning

* Some Algorithms have 10+ parameters
* Each parameter can have 10+, or 100+ possible values
* Search Space becomes huge
* ~~You could do this manually!~~
* Don’t do it manually!
* Use in-built Functions to do this
* But it will take some time, maybe a long, long, long time

# How to do this

* There are 2 main approaches
* Random Grid Search – Randomly select values for parameters from list/range
* Grid Search – Walks through all combinations
* These approaches can be used to find the best combination of Parameters and their Settings
* What’s a Grid?
* It’s a List of Parameters and the Values to be included in the Search
* The Values can be a List of values, or you can give a Range of values
* Or some combination of these

|  |  |  |
| --- | --- | --- |
| #parameters with a list of values a1: [0,1,2,3,4,5] a2: [10,20,30,40,5,60] a3: [105,105,110,115,120,125] |  | #parameters with list & range of values a1: [0,1,2,3,4,5]  a2: list(range(10,60)) #all values between 10 & 60 a3: [105,105,110,115,120,125] |

# Random Grid Search

from sklearn.model\_selection import GridSearchCV, RandomizedSearchCV

param\_grid = {

'n\_estimators': [25, 50, 100, 150],

'max\_features': ['sqrt', 'log2', None],

'max\_depth': [3, 6, 9],

'max\_leaf\_nodes': [3, 6, 9],

}

#RandomizedSearchCV will select a Random selection of values for each parameter.

# This might not be suitable as it might miss important values

random\_search = **RandomizedSearchCV**(RandomForestClassifier(), param\_grid)

random\_search.fit(X\_train, y\_train)

# random random search results

print('Best random search hyperparameters are: '+str(random\_search.best\_params\_)) print('Best random search score is: '+str(random\_search.best\_score\_))

Best random search hyperparameters are: {'n\_estimators': 25, 'max\_leaf\_nodes': 9, 'max\_features': 'log2', 'max\_depth': 6}

Best random search score is: **0.8438924650439015**

Check out this webpage for more RandomizedSearchCV details

[https://scikit-learn.org/stable/modules/generated/sklearn.model\_selection.RandomizedSearchCV.htm](https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.RandomizedSearchCV.html)l

# Grid Search

rfc = RandomForestClassifier()

#GridSearch can take a lot of time! We will only use these 2 parameters as an example forest\_params = [{'max\_depth': list(range(2, 6)),

'max\_features': list(range(3, 8))}]

grid\_search = GridSearchCV(rfc, forest\_params, cv = 10, scoring='accuracy')

#this next command will take some time! grid\_search.fit(X\_train, y\_train)

GridSearchCV(cv=10, estimator=RandomForestClassifier(), param\_grid=forest\_params, scoring='accuracy’)

print('Best hyperparameters are: '+str(grid\_search.best\_params\_)) How does this compare to print('Best score is: '+str(grid\_search.best\_score\_)) RandomGrid Search?

Best hyperparameters are: {'max\_depth': 5, 'max\_features': 6} Can you explain the

Best score is: **0.853106644958161** difference?

[https://scikit-learn.org/stable/modules/generated/sklearn.model\_selection.GridSearchCV.htm](https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html)l



# Automating the Process Why Automate

* To make your life easier
* To make the job easier
* Allows you to concentrate on the important things -> the Business Problem
* No one likes boring, repetitive tasks
* Avoid mistakes due to boring, repetitive tasks
* Things can go wrong when there is so many different tasks and dependencies between these

# How do we automate

* Identify what do we need to do every time
* Can we Automate it in some way
* Writing code is a way to do
* Creating a Notebook with all steps
* Re-run the Notebook – when we have new data
* Using Loops
* Can we really Automate every step?
* Should we automate
* Some legal requirements – See topic later in the semester
* Human oversight is vital
* What happens when the automation goes wrong?

# How do we automate

* Document your code
* Document decisions
* Document outcomes
* Document edge cases
* Etc
* Create Loops
* Integrate Charts
* Integrate Results Time for
* Format the Outputs an
* Make it easier to follow and to understand
* How hands free can you be Example
* Create time to focus on Business Problem

Assessment-A AutoML

Assessment-B

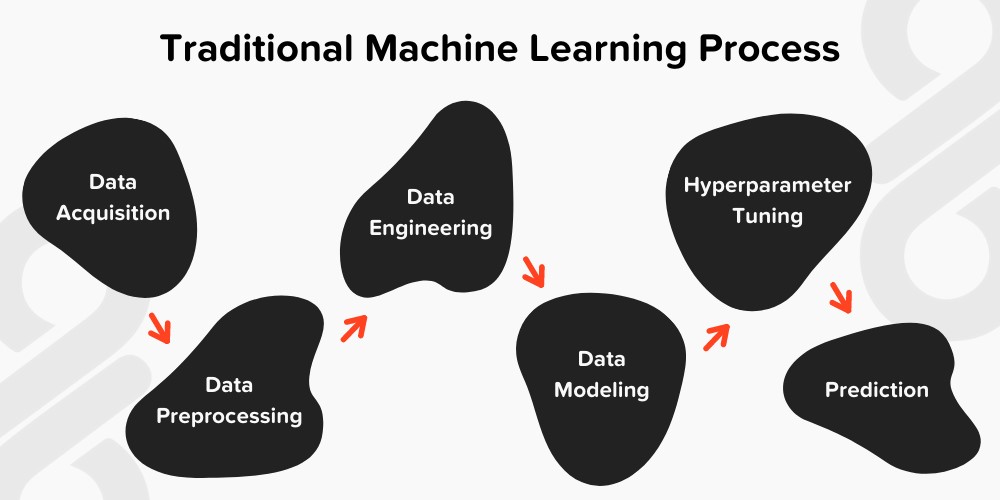
# Automate the Boring Stuff

* We have seen examples of Automation before
* Data Exploration
* Graphs for Data
* Data Preparation
* They are useful up to a point
* AutoML -> Automate Machine Learning
* A very popular “buzz” word over past few years
* Can help to guide the Analytics – but doesn’t give some magic answer
* It can give the wrong result -> just like ChatGPT
* Some Legal aspects

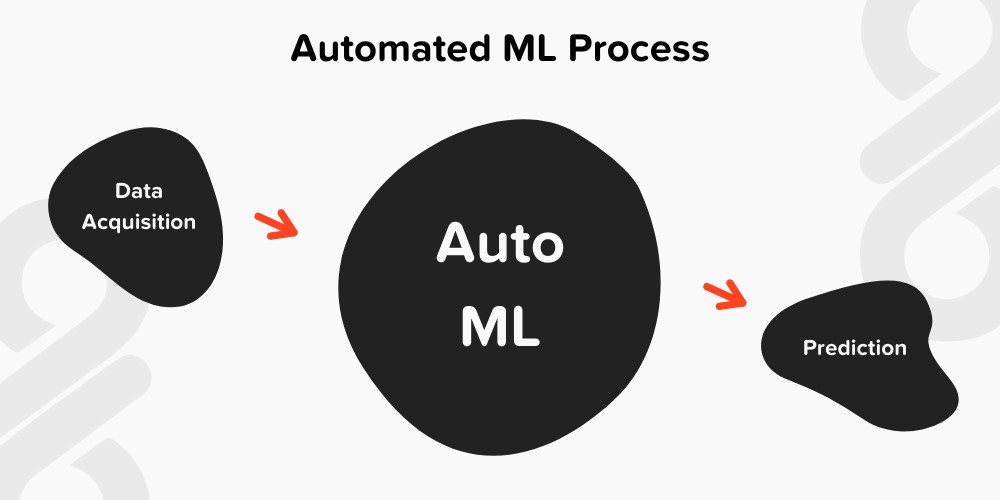
# Pros vs Cons of AutoML

* **Pros**
* Reduce the time it takes to implement traditional ML models
* Reduce human effort by automatically running repetitive tasks
* Reduce human errors
* Save a lot of GPU and CPU processing, resulting in cost and power efficiency
* Anyone without ML knowledge can enjoy the benefits of ML features
* Opens doors for new opportunities to create a platform to provide AutoML apps for easier access to machine learning
* **Cons**
* Human intelligence is neglected in complex problems, which can be more efficient than autoML
* More emphasis on research and automating everything can lead to fewer jobs for data scientists
* ML makes some decisions, like feature engineering, on the basis of domain knowledge which is lacking in the automation process
* AutoML only focuses on supervised tasks that require labelled data as input and overlooks the more challenging tasks of unsupervised and reinforcement learning.

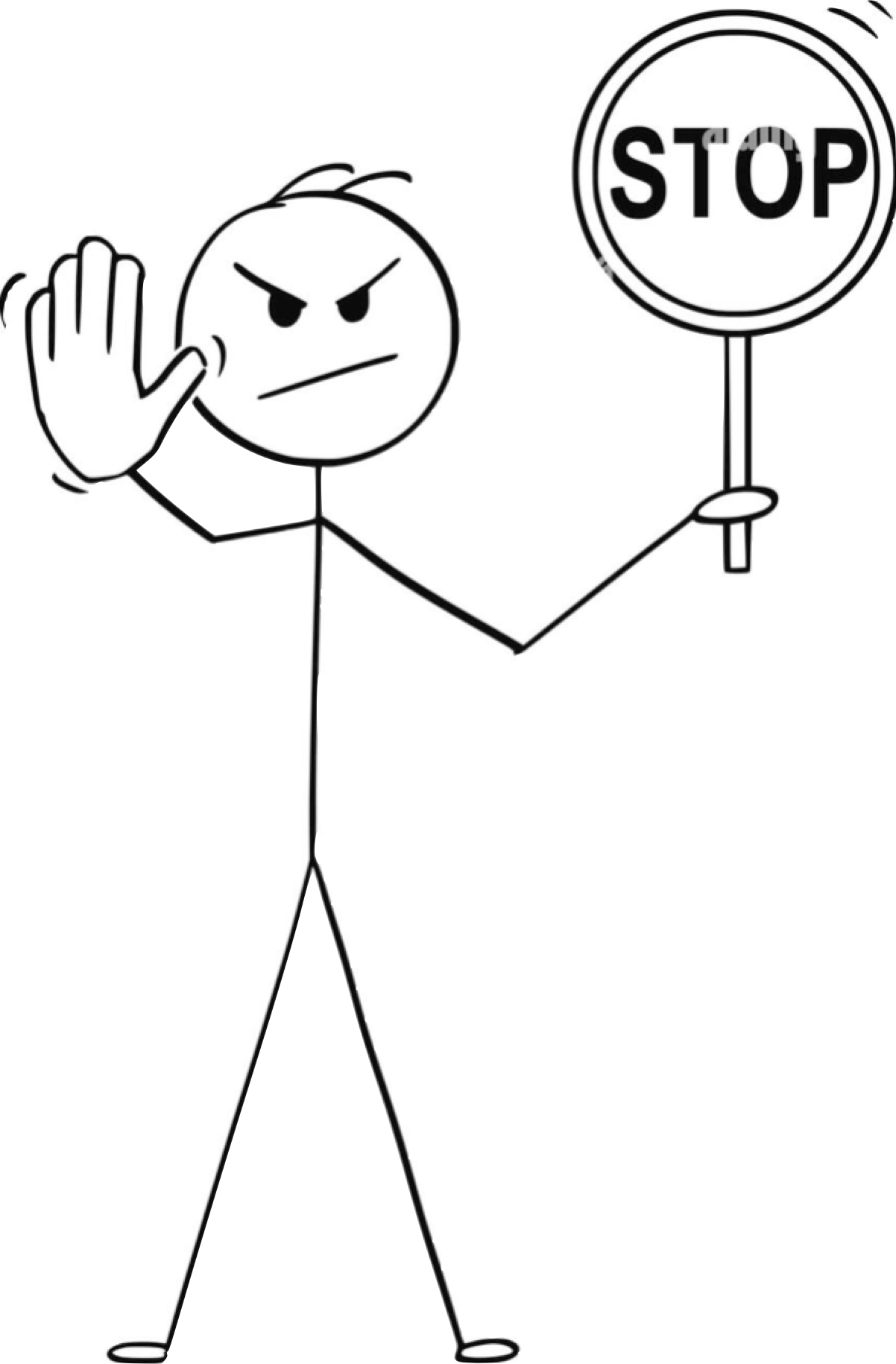
# Traditional ML



# AutoML

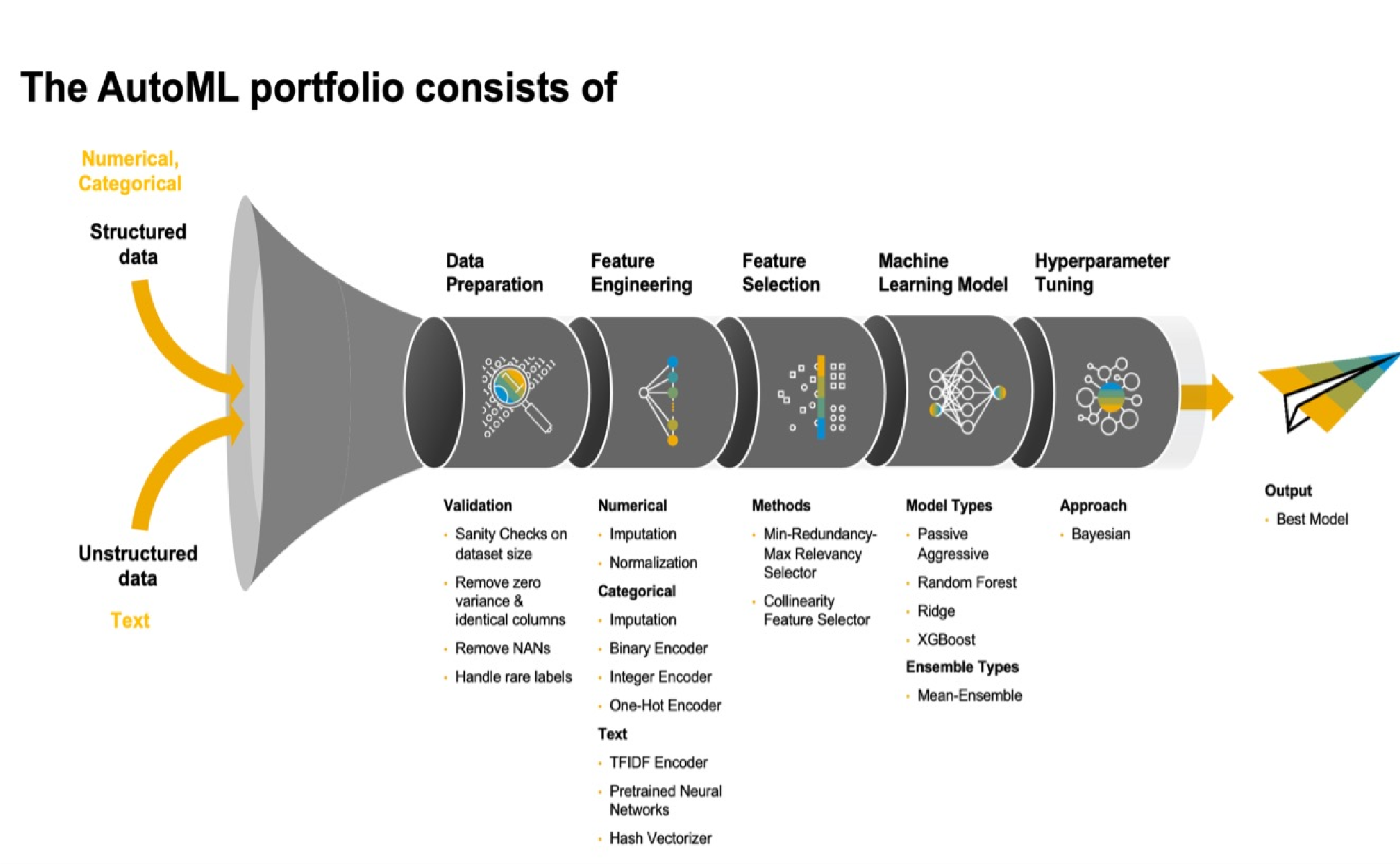


# AutoML - Limitation

* It doesn’t work for all types of Algorithms or Problems
* Typically, suited to Classification
* Yes/No
* 1/0
* Multi-Class e.g. 1, 2, 3, 4
* Some can do Regression
* Not much else -> But a larger percentage of problems are Classification

# What does AutoML do

* Iterates through the process

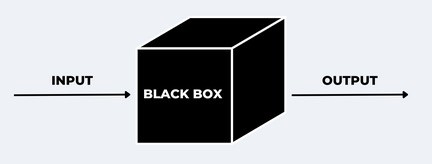


•

Meta Learning is used to iterate back over these steps to improve the results

* Data Preparation
* Feature Engineering
* Feature Selection
* Machine Learning
* Tuning
* Output is “Optimal” model
* Usually based on accuracy scores
* Different Feature subsets selected
* Selects appropriate Algorithms
* Keeps iterating -> for a defined time, or a number of iterations or ….

# Problems with using AutoML

* Cannot fix for bad Business Problem
* Cannot fix bad/poor Data Quality
* Does not explain WHY things have changed, etc
* Rubbish in = Rubbish out
* Human Oversight is needed
* No or Limited Model Explainability
* Legal Implications
* Reinforce Data Biases
* But it could give you a bit of a guide for what you can do Manually -> Human Oversight
* It can be Slow -> But it’s doing lots of work -> It would be slower to write all the code yourself
* This isn’t a bad thing – Just it isn’t a magic solution

# Lots of AutoML soluctions

|  |  |
| --- | --- |
| * [AutoWEK](http://www.cs.ubc.ca/labs/beta/Projects/autoweka/)A * [Auto-sklear](https://automl.github.io/auto-sklearn/master/)n * [Auto-PyTorc](https://github.com/automl/Auto-PyTorch)h * [AutoGluo](https://auto.gluon.ai/stable/index.html)n * [H2O AutoM](http://docs.h2o.ai/h2o/latest-stable/h2o-docs/automl.html)L * [MLBo](https://github.com/AxeldeRomblay/MLBox)X * [TPO](http://epistasislab.github.io/tpot/)T | * [TransmogrifA](https://github.com/salesforce/TransmogrifAI)I * [Amazon Le](https://aws.amazon.com/lex/)x * [AutoKera](https://autokeras.com/)s * [Data Robo](https://www.datarobot.com/)t * [BigML AutoM](https://bigml.com/)L * [Google Cloud AutoM](https://cloud.google.com/automl)L * [Auto-WEK](https://www.cs.ubc.ca/labs/algorithms/Projects/autoweka/)A |

Plus lots, lots more

# Some Blog Posts

* [AutoML, what is it good for? It](https://oralytics.com/2021/03/01/automl-what-is-it-good-for-it-depends/) [Depend](https://oralytics.com/2021/03/01/automl-what-is-it-good-for-it-depends/)s!
* [AutoML –](https://oralytics.com/2021/04/19/automl-using-tpot/) [using](https://oralytics.com/2021/04/19/automl-using-tpot/) [TPO](https://oralytics.com/2021/04/19/automl-using-tpot/)T

|  |
| --- |
| See Installation Tip on next slide |

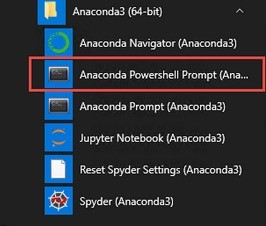
* [AutoML –](https://oralytics.com/2021/04/12/automl-using-autosklearn-in-python/) [using autosklearn in](https://oralytics.com/2021/04/12/automl-using-autosklearn-in-python/) [Pytho](https://oralytics.com/2021/04/12/automl-using-autosklearn-in-python/)n
* [AutoML using Pycare](https://oralytics.com/2022/02/28/automl-using-pycaret/)t
* [OML4Py –](https://oralytics.com/2021/03/29/oml4py-automl-step-by-step-approach/) [AutoML –](https://oralytics.com/2021/03/29/oml4py-automl-step-by-step-approach/) [Step-by-Step](https://oralytics.com/2021/03/29/oml4py-automl-step-by-step-approach/) [Approac](https://oralytics.com/2021/03/29/oml4py-automl-step-by-step-approach/)h

# AutoML install/setup

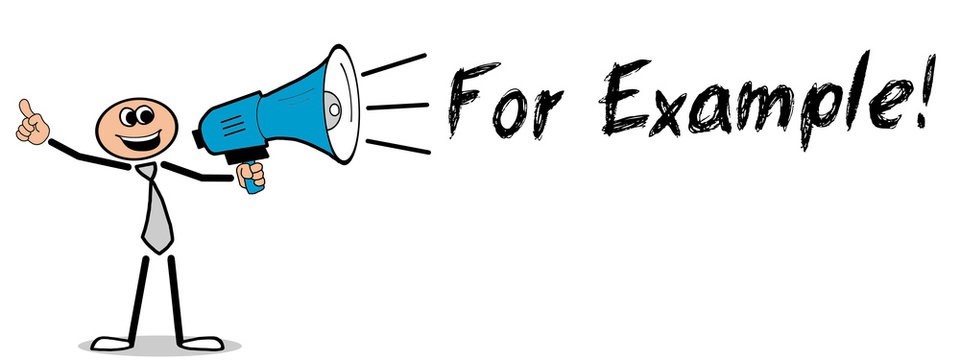
* This can be a little challenging in Anaconda

|  |
| --- |
| Similar needed for autosklearn |

* Some of these AutoML libraries need specific versions of other libraries
* These might not be what you have installed!
* Create a new Anaconda Virtual Environment
* Install the AutoML into it
* Here are some blog posts illustrating this
* [Installing PyCaret in Anacond](https://insaid.medium.com/a-complete-guide-to-pycaret-c07b1e51f698)a
* [Pycaret Installation Documentatio](https://pycaret.gitbook.io/docs/get-started/installation)n
* Although some might work in your current Anaconda environment



tpot - It if isn’t listed in available list of libraries to install, run the following conda install -c conda-forge tpot



Time for an

Example

Any Questions ?

What Now/Next ?