Auto ML :-

→ repetitive ml tasks can be automated.

→ Sagemaker Autopilot (AutoML)

Model Building Challenges :-

→ Steps involved in building ML model requires multiple iterations that can often result in increased time to market.
(Ability to reduce the time to market).

→ Lack of ML skillsets (may be from the team)

→ Iterations (data transformations, differ algorithms, hyper-parameter tuning etc.,) to to find a good model.

→ Optimize scarce resources and skillsets.

AutoML :- automates the steps in ML workflow

Benefits :-

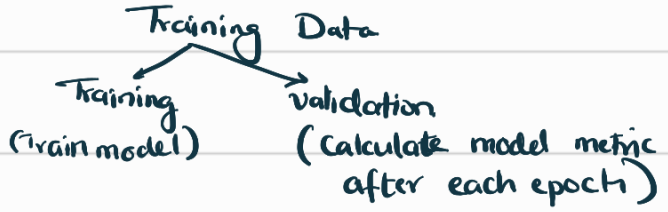
1.) Reduce the time to market by automating data transformation, feature engr, tuning
2.) Non-data science folks can use the ML models
3.) Iterate quickly using ML and automation to perform majority of the tasks in model building workflows.
4.) AutoML + cloud train/tune model in parallel.

Text Clanification Data preparation :-

× No one-hot encoding.

Tokenization → stop words removal → convert into vectors ...

Also, we have to deal with imbalance. { Use synthetic data, resampling techniques, choose different metric (say weighted F1 score), different algorithm (xgboost - grad descent boosted decision trees) }



Training and tuning is time and resource consuming and hence distributed computing (scale out) plays an important benifit.

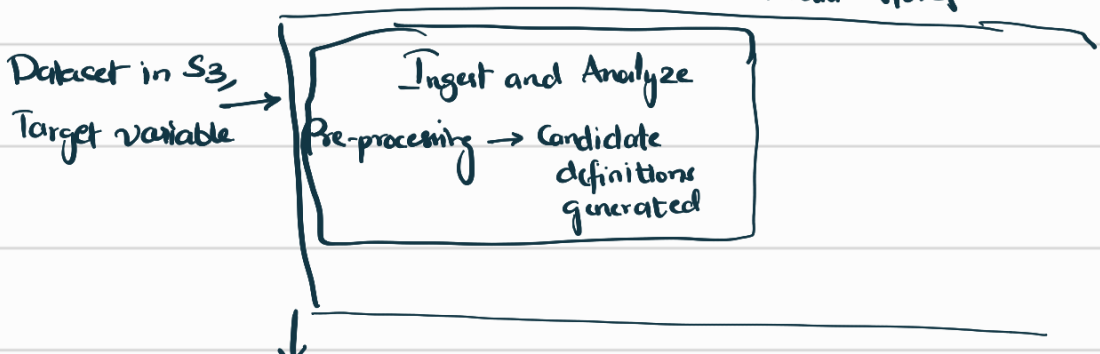
AutoML aims at automating the process of building a model.

→ builds models without ML expertise
→ models at scale without human intervention
→ automate majority of work, then tweak
(data cleaning, feature engr, feature selection)

Level of transparency and control is also be achieved in most cases. In some cases only the best performing model is returned, but it is hard to reproduce manually or explain.

ASM autopilot:

Problem Detection → Data exploration Bias detection → Algorithm selection → Preparation and Transformation {Feature engr. Feature store} → Training, Tuning → Optimal set of parameters with a high performent model.



Analyze data → Calculate statistics → Identifies ML problem → Chooses Algorithm. → Define model candidate pipelines → Generate Feature Engr code → Generate notebooks.

Prepare and Transform {Feature Engineering}

Autopilot transforms the data using generated feature code

Train and Tune Model Tuning

Autopilot.

Model Training & hyper-parameter optimization for each of candidate model.

→ All metrics
→ Leaderboard of model candidates
→ Notebooks
→ Code.

Notebook 1 :- describes what autopilot learned about data, potential issues and insights
Notebook 2 :- candidate generation notebook about candidates, hyperparameters, tuning.

ASM autopilot :-

Autopilot experiment → job.

Interact with → AWS CLI
→ AWS SDK
→ ASM python SDK.

or ASM Studio

ASM SDK on top of sklearn.

Text data → Features

TFIDF

ASM autopilot uses multi-column tfidf vectorizer and ASM autopilot will automatically tune MultiColumnTfidfVectorizer parameters.

How relevant a word is to a doc or collection of documents.

All text into a bag and sample from it without replacement. Form a vector of counts.

Then how important a word is to a document, relative to others in the corpus.

Word importance → statistical measure
→ increases proportionally to number of times a word appears in the doc.
→ decreases by the frequency of the word in the corpus.

t – term / word
d – document
D – corpus

$$tf(t,d) = \frac{f_{t,d}}{\sum_{t' \in d} f_{t',d}}$$

whole corpus → $$idf(t,D) = \log\left(\frac{|D|}{|\{d \in D : t \in d\}|}\right)$$

say food appears in doc, 10 times of 200 words and the word food is in 1000 docs of 10^6 docs

then

$$idf(food, corpus) = \log\left[\frac{10^6}{10^3}\right] = 3$$

$$df(food, doc_1) = \frac{10}{200} = 0.05$$

$$Tfidf(food) = 0.05 \times 3 = 0.15$$

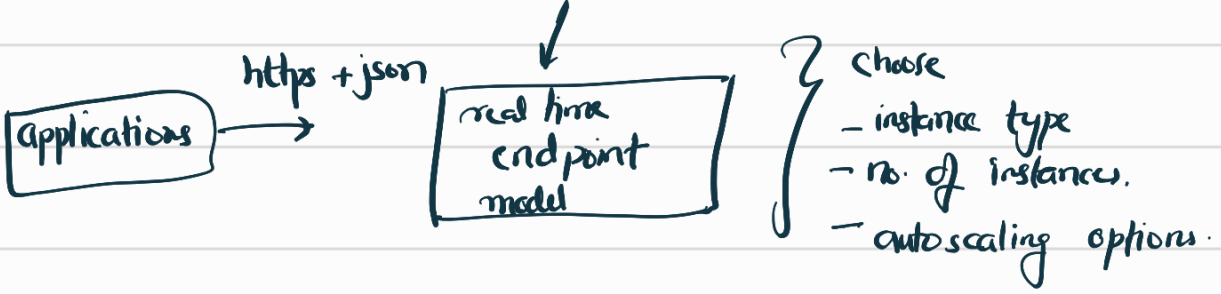
Once you find the model (best performant), if we need to deploy to production (prediction in real time)

- Optimized for low latency of model predictions.

model serving stack → trained model + hosting stack

{ proxy, webserver with trained code }

↓



Inference Pipeline model

Data Transformation Container	Algorithm Container	Inverse Label Transform Container.
Container built from the model we selected and trained during the data transformer sections	a container build from the trained model we selected above from the best HPO training job.	a container that converts intermediate numerical prediction to the label value.