Building machine learning models automatically

Julien Simon
Global AI/ML Evangelist, AWS
@julsimon

AutoML

- AutoML aims at automating the process of building a model
- Problem identification: looking at the data set, what class of problem are
  we trying to solve?
- Algorithm selection: which algorithm is best suited to solve the problem?
- Data preparation: how should data be prepared for best results?
- Hyperparameter tuning: what is the optimal set of training parameters?

Scenarios for AutoML

- Build models without any ML expertise
  - Empower more people in your organization: software developers, business people
  - Let experts focus on hard problems

- Experiment and build models at scale
  - Thousands of data sets can be modeled without human intervention
  - Let experts focus on new problems

- Automate 90% of the work, then tweak
  - Data cleaning, feature engineering, feature selection, etc.
  - Let experts focus on high value tasks such domain knowledge, and error analysis.

Transparency and control are important

- Get the best model only
  - Hard to understand it
  - Hard to reproduce it manually

- Get the best model, all candidates, full source code
  - Understand how the model was built
  - Keep tweaking for extra performance

Agenda

- Zero-code AutoML with SageMaker Studio
- AutoML with SageMaker AutoPilot and the SageMaker SDK
- Open source AutoML with AutoGluon

AutoML with Amazon SageMaker

Amazon SageMaker helps you build, train, and deploy models

Prepare

Build

Train & Tune

Deploy & Manage

Web-based IDE for machine learning

Automatically build and train m

Fully managed data processing jobs and data labeling workflows

Collect and prepare training data

One-click collaborative notebooks and built-in, high performance algorithms and models

Choose or build an ML algorithm

One-click training

Set up and manage environments for training

Debugging and optimization

Train, debug, and tune models

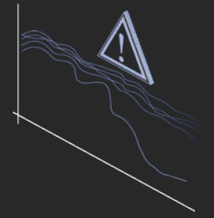
Visually track and compare experiments

Manage training runs

One-click deployment and auto scaling

Deploy model in production

Automatically spot concept drift



Monitor models

Add human review of predictions

Validate predictions

Fully managed with auto-scaling for 75% less

Scale and manage the production environment

Modular service and APIs, from experimentation to production

AutoML with Amazon SageMaker Autopilot

- SageMaker Autopilot covers all steps
  - Problem identification: looking at the data set, what class of problem are we trying to solve?
  - Algorithm selection: which algorithm is best suited to solve the problem?
  - Data preprocessing: how should data be prepared for best results?
  - Hyperparameter tuning: what is the optimal set of training parameters?
- Autopilot is fully transparent
  - An auto-generated notebook shows you how models are built
  - You can run it, reproduce the experiment, and keep tweaking
- Use cases: regression and classification, based on Linear Learner, Factorization Machines, KNN, XGBoost

AutoML with Amazon SageMaker Autopilot

1. Upload the unprocessed dataset to S3
2. Configure the AutoML job
   - Location of dataset
   - Target attribute
   - Completion criteria
3. Launch the job
4. View the list of candidates and the autogenerated notebook
5. Deploy the best candidate to a real-time endpoint, or use batch transform

AutoML with AutoGluon

Apache MXNet

- Open source software library for Deep Learning
- Natively implemented in C++
- Built-in support for many network architectures: FC, CNN, LSTM, etc.
- Symbolic API: Python, Scala, Clojure, R, Julia, Perl, Java (inference only)
- Imperative API: Gluon (Python), with computer vision, natural language processing, and time series toolkits

AutoGluon

https://autogluon.mxnet.io/

- Open source AutoML toolkit
- Tabular data: regression, classification
  - LightGBM, CatBoost, Random Forests, Extra Trees, KNN, linear regression
- Text data: classification
  - Transfer learning based on Gluon NLP models
- Image data: classification, object detection
  - Transfer learning based on Gluon CV models
- Built-in hyperparameter optimization
- Built-in ensemble prediction (bagging & stacking)

AutoML on tabular data in 3 lines of code

```
from autogluon import TabularPrediction as task

predictor = task.fit(train_data=task.Dataset(
            file_path=TRAIN_DATA.csv),
            label=COLUMN_NAME)

predictions = predictor.predict(
              task.Dataset(file_path=TEST_DATA.csv))
```

Demo: AutoGluon on the Boston Housing Dataset

https://gitlab.com/juliensimon/dlnotebooks/-/tree/master/autogluon

Getting started

http://aws.amazon.com/free

https://ml.aws

https://aws.amazon.com/sagemaker

https://github.com/aws/sagemaker-python-sdk

https://github.com/awslabs/amazon-sagemaker-examples

https://gitlab.com/juliensimon/dlnotebooks

https://youtube.com/juliensimonfr