aws machine learning

Automatically build, train, and tune models with AutoML from AWS

Yevgeniy Ilyin, AWS Senior Solutions Architect

By the end of today's session, you will be able to...

1

...use a simple drag and drop UI to do data exploration, data processing and feature engineering.

2

...use AutoML to automatically build, train, and tune the best machine learning pipelines for your tabular datasets.

3

...use SageMaker Python SDK to implement an AutoML-pipeline in your Jupyter notebook

Use this workshop to see the detailed steps of the demos we will go over

AWS Machine Learning Low-Code Immersion Day

aws machine learning

Agenda

- Overview of low-code and no-code capabilities within SageMaker
- Amazon SageMaker Autopilot
- Use cases for Amazon SageMaker Autopilot
- Hands-on workshop
  - *Multi-class classifier (Healthcare and Life Sciences dataset)*
  - *Binary classifier (Financial Services dataset)*
  - *Time series (cross-industry)*
  - *Use SageMaker Autopilot with Python SDK*
- Resources & Documentation

Typical ML Workflow

PREPARE	BUILD	TRAIN & TUNE	DEPLOY & MANAGE
Data Exploration, Data Preparation & Feature Engineering	Model Development	Model Training & Optimization	Live or Batch Predictions: Model Hosting & Monitoring

Common challenges with Machine Learning

1. Requires deep expertise to prepare the data, and build the models
2. Experimentation is time-consuming & resource intensive
3. Data scientists are oversubscribed, and needs are only increasing
4. Ramp time from business analysts turning citizen data scientists

aws machine learning

Low-Code and No-Code ML Can Support Faster Experimentation

"Build the best feature engineering pipelines in a matter of days rather than months."

"Automatically build, train and tune hundreds of models in parallel and pick the best performing."

"Use pre-trained models and reach production with a full-blown ML solution for your own use cases in 4–6 weeks instead of 3–4 months."

Low-Code / No-Code Machine Learning from AWS

Amazon SageMaker Studio

A dedicated workspace for data engineers, data scientists and ML Ops teams to collaborate and bring ML to market faster

PREPARE → **Data Exploration, Data Preparation & Feature Engineering**

BUILD → **Model Development**

TRAIN & TUNE → **Model Training & Optimization**

DEPLOY & MANAGE → **Live or Batch Predictions: Model Hosting & Monitoring**

1 **Amazon SageMaker Data Wrangler**
A faster, visual way to aggregate and prepare data for machine learning

2 **Amazon SageMaker Autopilot**
AutoML capability that automatically prepares your data, as well as builds, trains, and tunes the best machine learning models for your tabular datasets

3 **Amazon SageMaker JumpStart**
Pre-built solutions and a model zoo of pre-trained and easily tunable state-of-the-art models for Computer Vision, and Natural Language Processing

Many deployment options

Collaboration

Amazon SageMaker Canvas

A dedicated no-code workspace for data analysts to generate ML-powered predictions

4 **Amazon SageMaker Canvas**
A visual point-and-click interface that allows analysts to generate accurate ML predictions on their own — without requiring any machine learning experience or having to write a single line of code.

Today's Use Cases using SageMaker Autopilot

1 Predict Patient Hospital Readmission

- Dataset: **Tabular** *from clinical care at 130 US hospitals*
- SageMaker Features: Data Wrangler + Autopilot

2 Predict Loan Defaults

- Dataset: **Tabular** *from Lending Club Loan Dataset*
- SageMaker Features: Data Wrangler + Autopilot

3 Predict weekly average temperature

- Dataset: **Time series** *from NOAA Global Historical Climatology Network Dataset*
- SageMaker Features: Data Wrangler + Autopilot

+ demo four how to use SageMaker Python SDK to run Autopilot workflow

aws machine learning

Amazon SageMaker Data Wrangler

How SageMaker Data Wrangler Works

Amazon SageMaker Data Wrangler

A faster, visual way to aggregate and prepare data for machine learning

Select & Query

Select and query data from a variety of data sources such as Amazon Athena, Amazon Redshift, AWS Lake Formation, and Amazon S3

Visualize

Graphically understand data and detect outliers with pre-configured visualization templates

Cleanse & Enrich

Cleanse and explore data, perform feature engineering with built-in data transforms, and detect statistical bias with SageMaker Clarify

Understand

Use a sample dataset to quickly estimate model performance, accuracy, and diagnose potential issues

Operationalize

Export data preparation workflow to a notebook or code script to bring the workflow into production

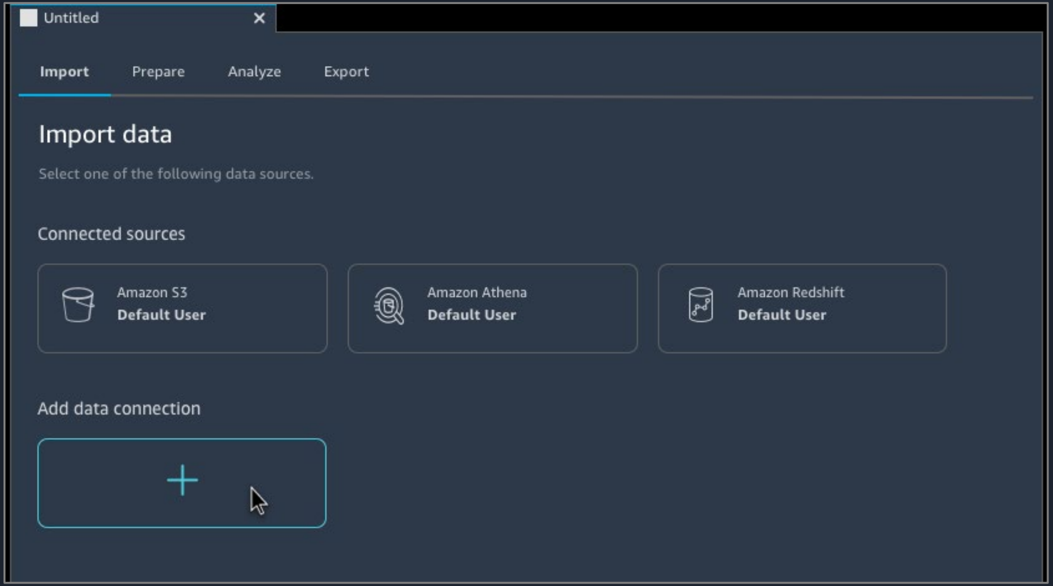
Integrate data preparation workflow with Amazon SageMaker Pipelines

Export prepared data into Amazon SageMaker Feature Store

Import data from a feature store such as Amazon SageMaker Feature Store

aws machine learning

Quickly select and query data

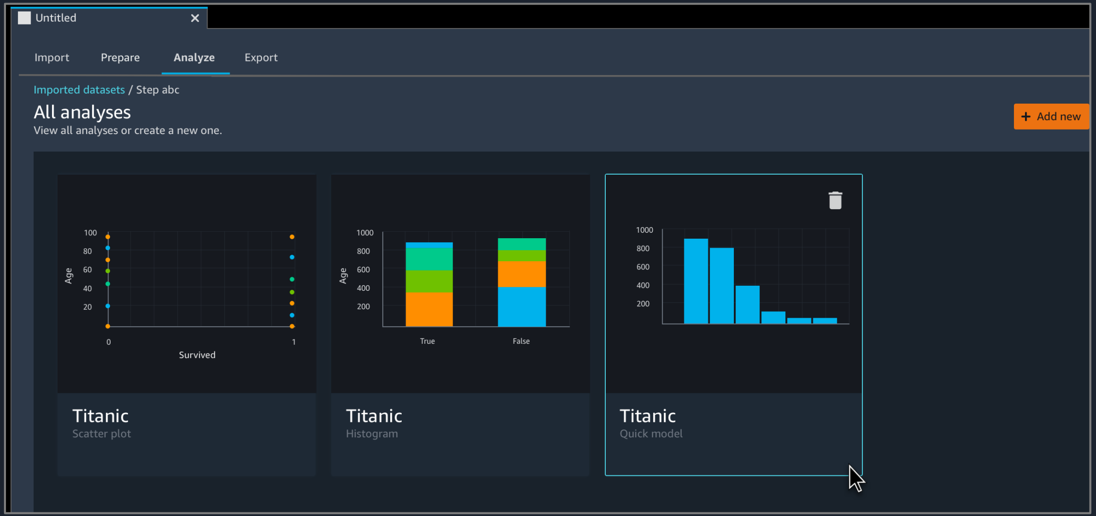


Select data from Amazon Athena, Amazon Redshift, AWS Lake Formation, Amazon S3, Snowflake, and features from SageMaker Feature Store

Write queries for data sources before importing data over to Data Wrangler

Import data in various file formats, such as CSV files, Parquet files, and database tables directly into Amazon SageMaker

Understand your data visually

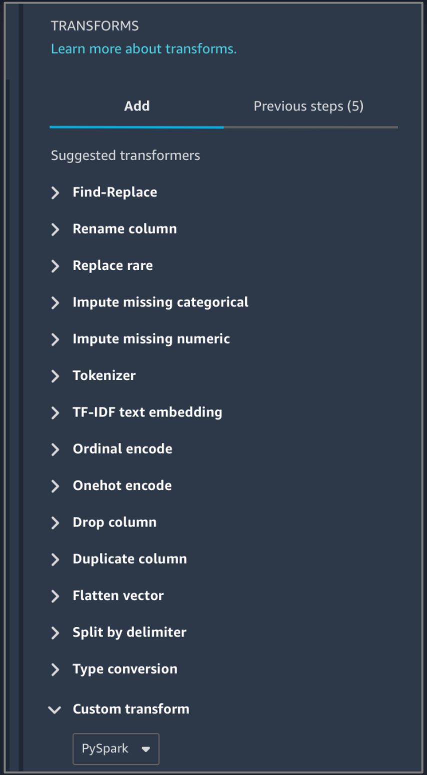


Intuitively understand data with a set of pre-configured visualizations

Pre-configured visualization templates include histograms, scatter plots, box and whisker plots, line plots, and bar charts

Ability to customize the templates, and script your own visualizations with Altair

Easily transform data



Transform your data without writing a single line of code using over 300 built-in data transformations

Built-in data transformations include convert column type, rename column, and delete column

Author custom transformations in PySpark, SQL, and Pandas

aws machine learning

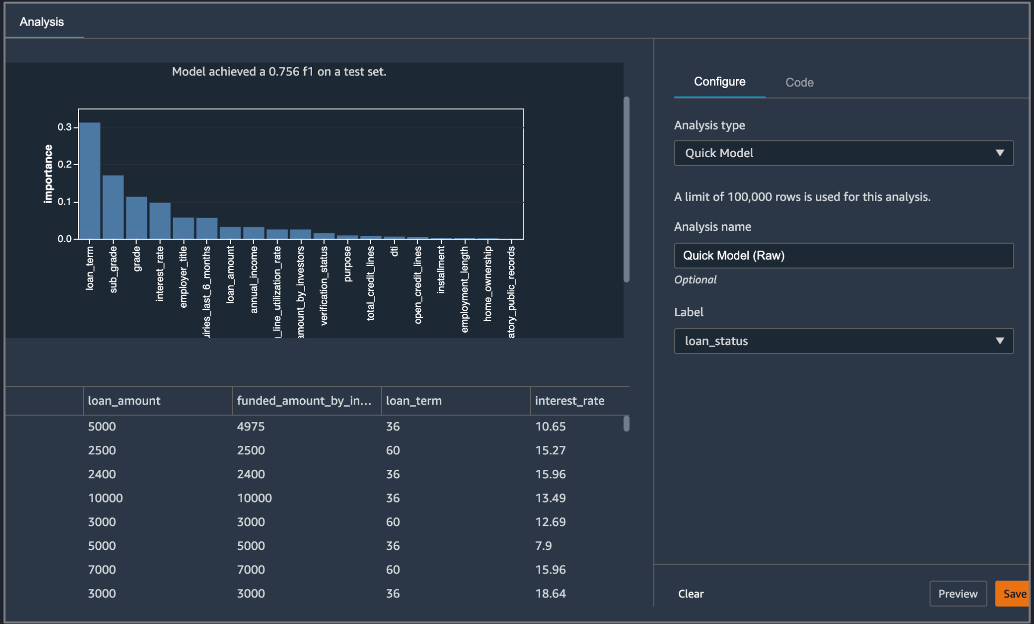
Quickly estimate model accuracy or leakage

Identify inconsistencies in data preparation workflows and diagnose issues before ML models are deployed into production

Select subsets of data to identify errors

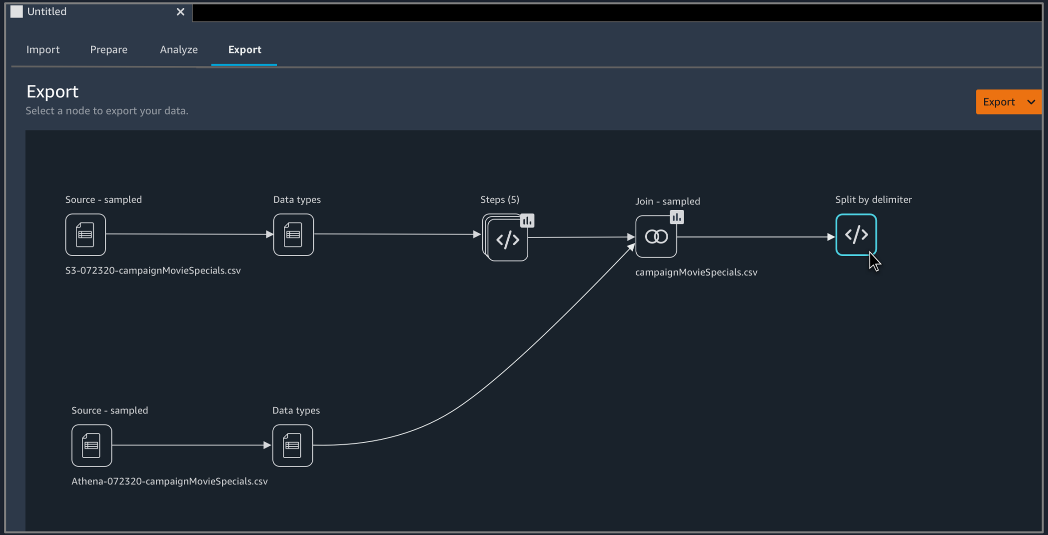
Identify which features are contributing to model performance relative to others

Determine if more feature engineering is needed to improve model performance



aws machine learning

Deploy data preparation workflows into production



Export data preparation workflows directly to S3, or Python code

Integrate your workflow with SageMaker Pipelines to automate model deployment and management

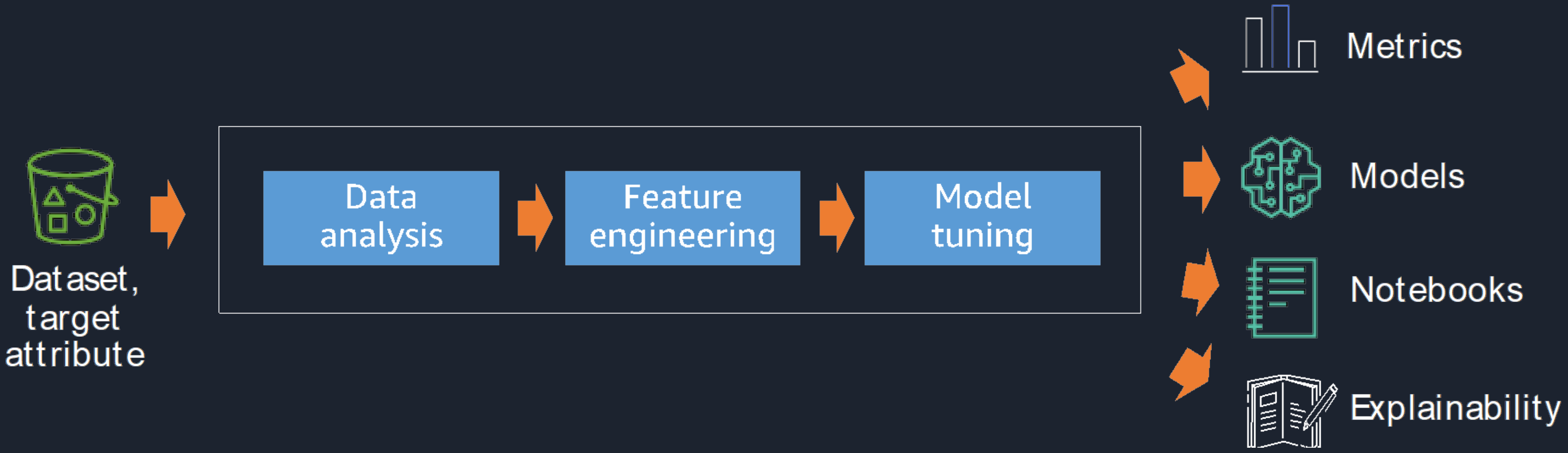
Publish created features to SageMaker Feature Store for reuse and syndication across teams and projects

aws machine learning

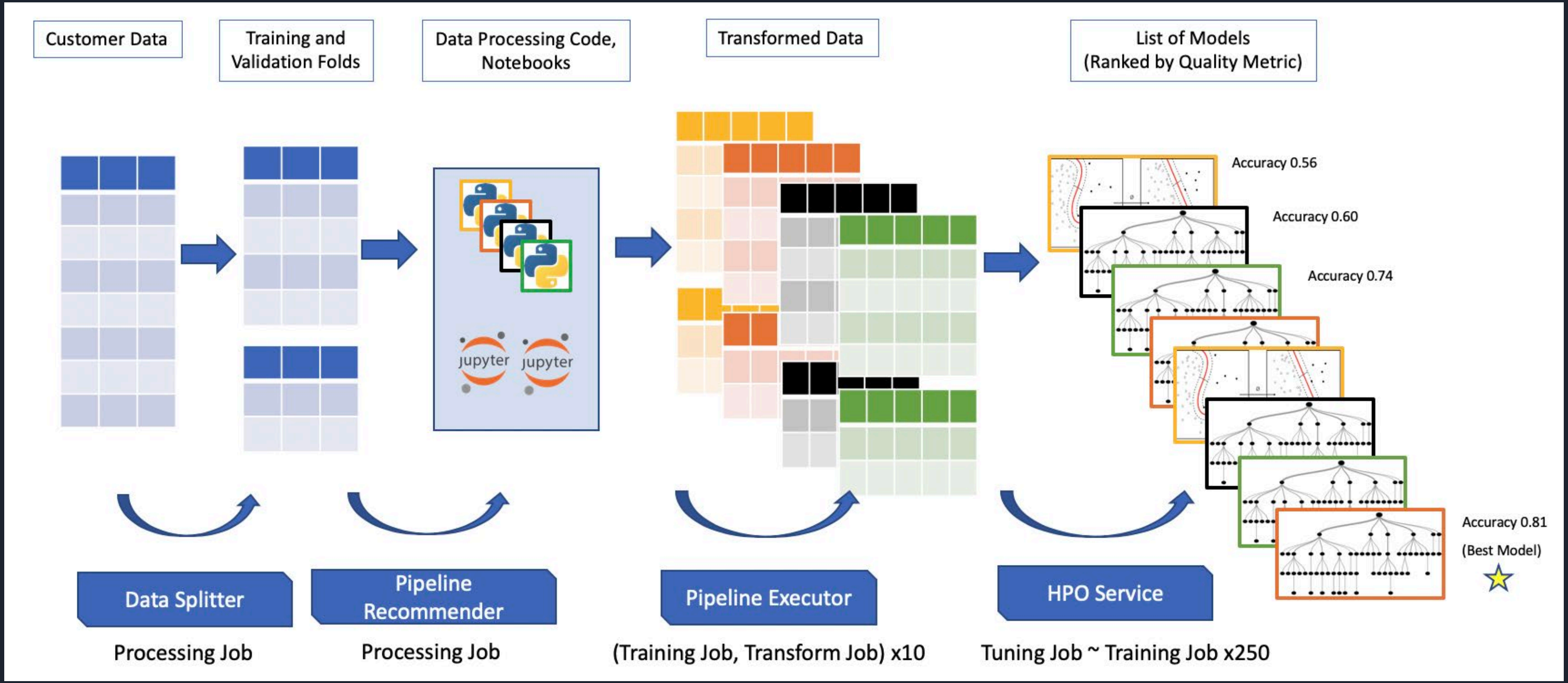
Amazon SageMaker Autopilot

 |

How SageMaker Autopilot Works

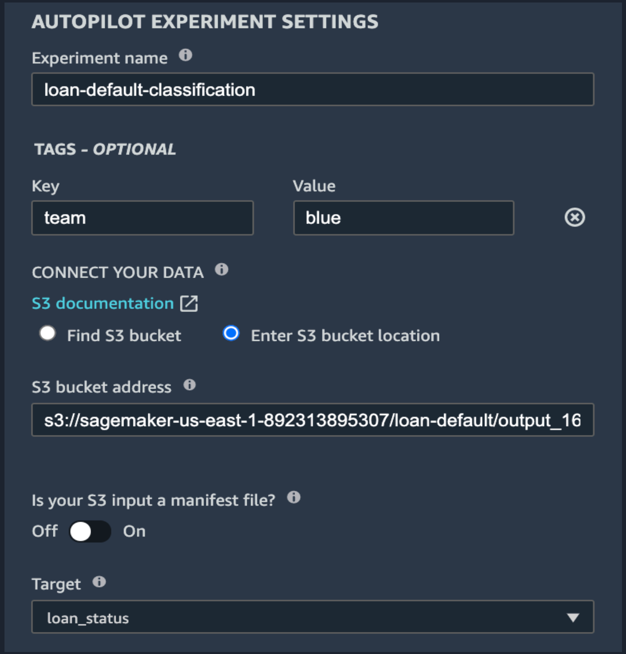


SageMaker Autopilot workflow



aws machine learning

Kick Off an Autopilot Job with a Few Clicks



Only S3 location and target variable required

Optional control points:

- dry-run vs complete mode
- setting problem type
- security settings

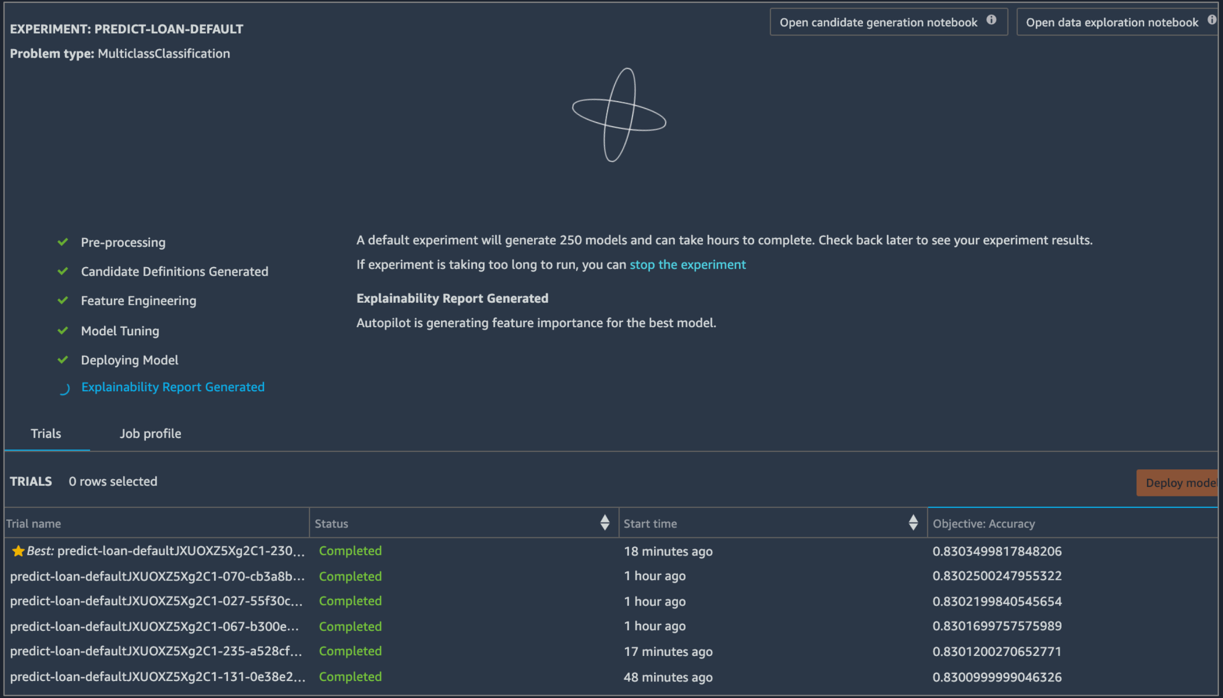
API level control points:

- number of candidate models to build
- maximum time to take
- model evaluation metric (accuracy, F1, RMSE)

 |

aws machine learning

Review the Performance of Candidates in Autopilot



Evaluate progress as each steps runs

Review the performance and lineage of each experiment

Understand explainability via a detailed report

Get access to the data exploration, and candidate notebooks

aws machine learning

A Completely White-Box Experience

Column Analysis

The AutoML job analyzed the `31` input columns to infer each data type and select the feature processing pipelines for each training algorithm. For more details on the specific AutoML pipeline candidates, see Amazon SageMaker Autopilot Candidate Definition Notebook.ipynb.

Percent of Missing Values

Within the data sample, the following columns contained missing values, such as: `nan`, white spaces, or empty fields.

SageMaker Autopilot will attempt to fill in missing values using various techniques. For example, missing values can be replaced with a new 'unknown' category for `Categorical` features and missing `Numerical` values can be replaced with the **mean** or **median** of the column.

We found **0 of the 31** of the columns contained missing values.

Suggested Action Items

Dataset Exploration Notebook:

- Dataset statistics: row-wise and column-wise
- Suggested remedies for common data issues

The following tunable hyperparameters search ranges are recommended for the Multi-Algo tuning job:

```
from sagemaker.parameter import CategoricalParameter, ContinuousParameter, IntegerParameter

ALGORITHM_TUNABLE_HYPERPARAMETER_RANGES = {
    'xgboost': {
        'num_round': IntegerParameter(2, 512, scaling_type='Auto'),
        'max_depth': IntegerParameter(2, 32, scaling_type='Auto'),
        'eta': ContinuousParameter(1e-3, 1.0, scaling_type='Logarithmic'),
        'gamma': ContinuousParameter(1e-6, 64.0, scaling_type='Logarithmic'),
        'min_child_weight': ContinuousParameter(1e-6, 32.0, scaling_type='Logarithmic'),
        'subsample': ContinuousParameter(0.5, 1.0, scaling_type='Linear'),
        'colsample_bytree': ContinuousParameter(0.3, 1.0, scaling_type='Linear'),
        'lambda': ContinuousParameter(1e-6, 2.0, scaling_type='Logarithmic'),
        'alpha': ContinuousParameter(1e-6, 2.0, scaling_type='Logarithmic'),
    },
    'linear-learner': {
        'wd': ContinuousParameter(1e-7, 1.0, scaling_type='Logarithmic'),
        'l1': ContinuousParameter(1e-7, 1.0, scaling_type='Logarithmic'),
        'learning_rate': ContinuousParameter(1e-5, 1.0, scaling_type='Logarithmic'),
        'positive_example_weight_mult': CategoricalParameter(['balanced', '0.01', '1', '100']),
    },
}
```

Fully runnable model candidate notebook:

- data transformers
- featurization techniques applied
- override points:
    - algorithms considered
    - evaluation metric
    - hyper-parameter ranges
    - model search strategy
    - instances used

aws machine learning

Demo

Data Wrangler and Autopilot

[AWS Machine Learning Low-Code Immersion Day]

Demo 1: Multi-class classifier

1

Predict Patient Hospital Readmission

Dataset

Tabular
from clinical care at 130 US hospitals

SageMaker Features

Data Wrangler
+
Autopilot

Use Case Details

- Hospital readmission is an important contributor to total medical expenditures and is an emerging indicator of quality of care. Diabetes, similar to other chronic medical conditions, is associated with increased risk of hospital readmission. hospital readmission is a high-priority health care quality measure and target for cost reduction, particularly within 30 days of discharge.

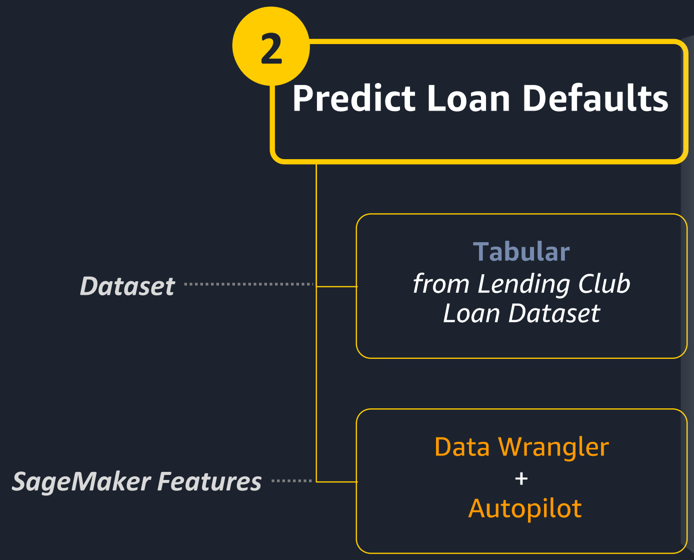
Dataset

- The data set represents 10 years (1999-2008) of clinical care at 130 US hospitals and integrated delivery networks. It includes over 15 features representing patient and hospital outcomes.
- The data set contains ~70,000 rows and 15 columns.

Process Details

- Data Wrangler to prepare data, perform exploratory data analysis (EDA) and feature engineering.
- Autopilot to train and tune optimal multi-class classifier.

Demo 2: Binary classifier



Use Case Details

- Lending loans to 'risky' applicants is the largest source of financial loss (called credit loss). The credit loss is the amount of money lost by the lender when the borrower refuses to pay or runs away with the money owed.
- Borrowers who default cause the largest amount of loss to the lenders, thus is important to predict who will.

Dataset

- The data is coming from the Lending Club Loan and contains complete loan data for all loans issued through the 2007–2011, including the current loan status and latest payment information.
- 39717 rows, 22 feature columns and 3 target labels.

Process Details

- Data Wrangler to prepare data, perform exploratory data analysis (EDA) and feature engineering.
- Autopilot to train and tune optimal binary classifier.

 |

aws machine learning

Demo 3: Timeseries prediction

3 **Predict weekly average temperature**

Dataset

Time series

from NOAA Global Historical Climatology Network Dataset

SageMaker Features

Data Wrangler
+
Autopilot

Use Case Details

- The simplest and arguably a logical way to predict future behavior of a system is to learn from it's past.
- Predict weekly average temperature based on previous observations.

Dataset

- NOAA Global Historical Climatology Network Daily Dataset: The dataset contains daily observations over global land areas.
- Filtered to one single weather station - CA21220. The dataset contains the daily values from the year 1955 to 2021.

Process Details

- Data Wrangler to prepare data, perform exploratory data analysis (EDA) and feature engineering for timeseries
- Autopilot to train and tune optimal model for predicting average temperature.

Demo 4: Use SageMaker Autopilot with Python SDK

[SageMaker Autopilot notebook example]()

2. Get detailed information on model candidates:

Python
```python
candidates = sm.list_candidates_for_auto_ml_job(AutoMLJobName=auto_ml_job_name, SortBy='Fi
index = 1
for candidate in candidates:
  print (str(index) + "  " + candidate['CandidateName'] + "  " + str(candidate['FinalAutoM
  index += 1
```

1. Create AutoML job:

Python
```python
auto_ml_job_name = 'automl-dm-' + timestamp_suffix
print('AutoMLJobName: ' + auto_ml_job_name)

import boto3
sm = boto3.client('sagemaker')
sm.create_auto_ml_job(AutoMLJobName=auto_ml_job_name,
                      InputDataConfig=input_data_config,
                      OutputDataConfig=output_data_config,
                      RoleArn=role)

AutoMLJobName: automl-dm-28-10-17-49
```

3. Deploy the best model candidate:

Python
```python
model_arn = sm.create_model(Containers=best_candidate['InferenceContainers'],
                            ModelName=model_name,
                            ExecutionRoleArn=role)

ep_config = sm.create_endpoint_config(EndpointConfigName = epc_name,
                                      ProductionVariants=[{'InstanceType':'ml.m5.2xlarge',
                                                           'InitialInstanceCount':1,
                                                           'ModelName':model_name,
                                                           'VariantName':variant_name}])

create_endpoint_response = sm.create_endpoint(EndpointName=ep_name,
                                              EndpointConfigName=epc_name)
```


aws machine learning

What’s new in SageMaker Autopilot

[SageMaker Autopilot experiments run up to 2x faster]

SageMaker Autopilot use ml.m5.12xlarge instances (48 vCPUs, 192 GiB memory) to reduce the number of default trials needed from 250 to 100.

Smaller datasets (< 100MB) – up to 45% faster
Medium datasets (>100MB < 1GB) – up to 40% faster
Large datasets (> 1Gb) – up to 40% faster

 |

aws machine learning

What's next?

1. Browse available resources and documentation

2. Run the same demos yourself! Go to [AWS Machine Learning Low-Code Immersion Day]

3. Reach out with any additional questions at: low-code-no-code-ml@amazon.com

aws machine learning

Resources & documentation

Documentation

Introduction to Amazon SageMaker

Introduction to SageMaker Data Wrangler – Documentation

Introduction to SageMaker Autopilot - Documentation

Blog posts & papers

[Make batch predictions with Amazon SageMaker Autopilot]

[Amazon SageMaker autopilot: a white box AutoML solution at scale - Amazon Science]

[Develop and deploy ML models using Amazon SageMaker Data Wrangler and Amazon SageMaker Autopilot]

aws machine learning

Workshops

[Workshop Demos – AWS Online Tech Talks]

[Amazon SageMaker examples Github]

[AWS Machine Learning Low-Code Immersion Day]

aws machine learning

Thank you

