Bias and Explainability-Tabular Data

⚠ Make sure you have performed the steps described in the **Prerequisites** section before beginning this lab.

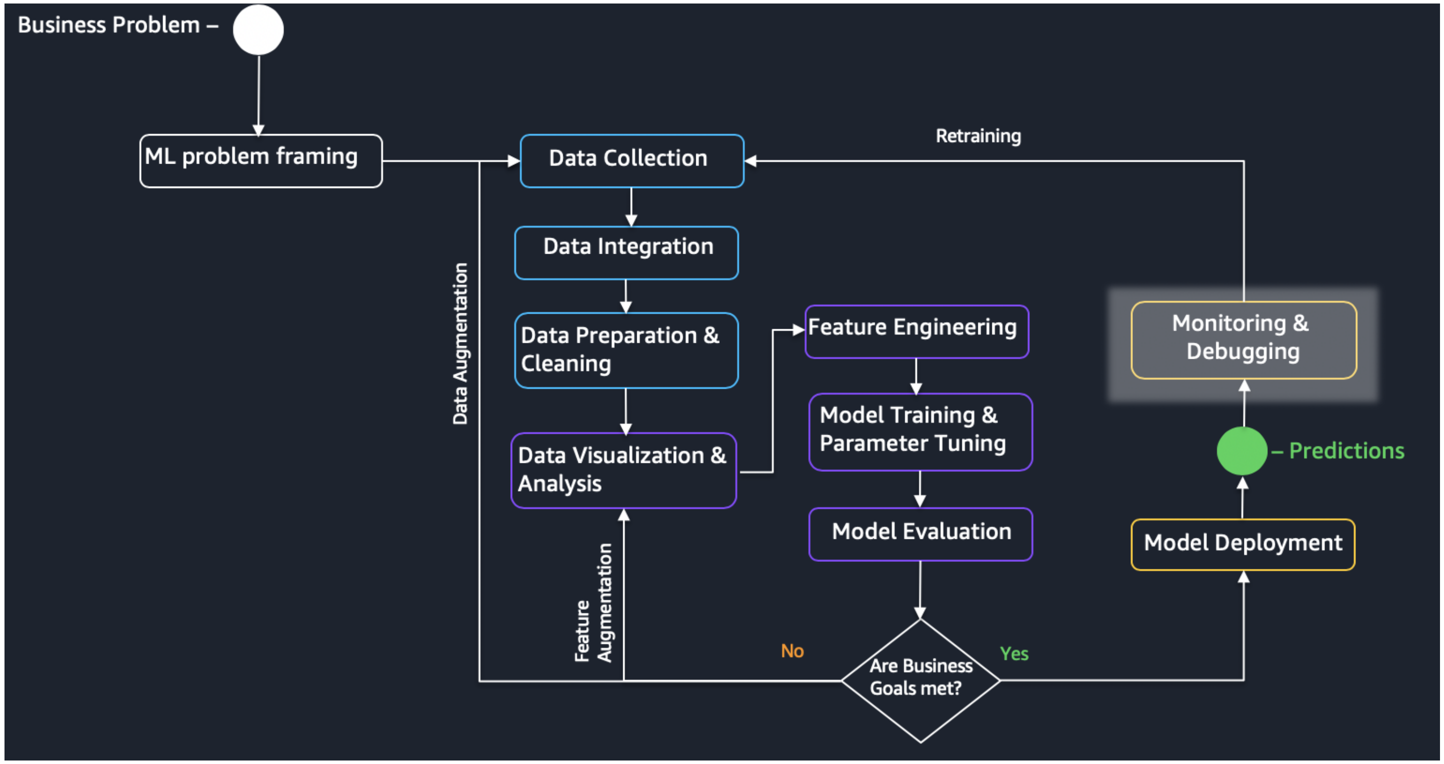
Overview

In this workshop, we demonstrate a end to end ML use case of credit risk prediction with model explainability and bias detection

This lab will walk you through the following SageMaker Clarify functionality

1. Measuring the pre-training bias of a dataset and post-training bias of a model
2. Explaining the importance of the various input features on the model's decision
3. Accessing the reports through SageMaker Studio if you have an instance set up.

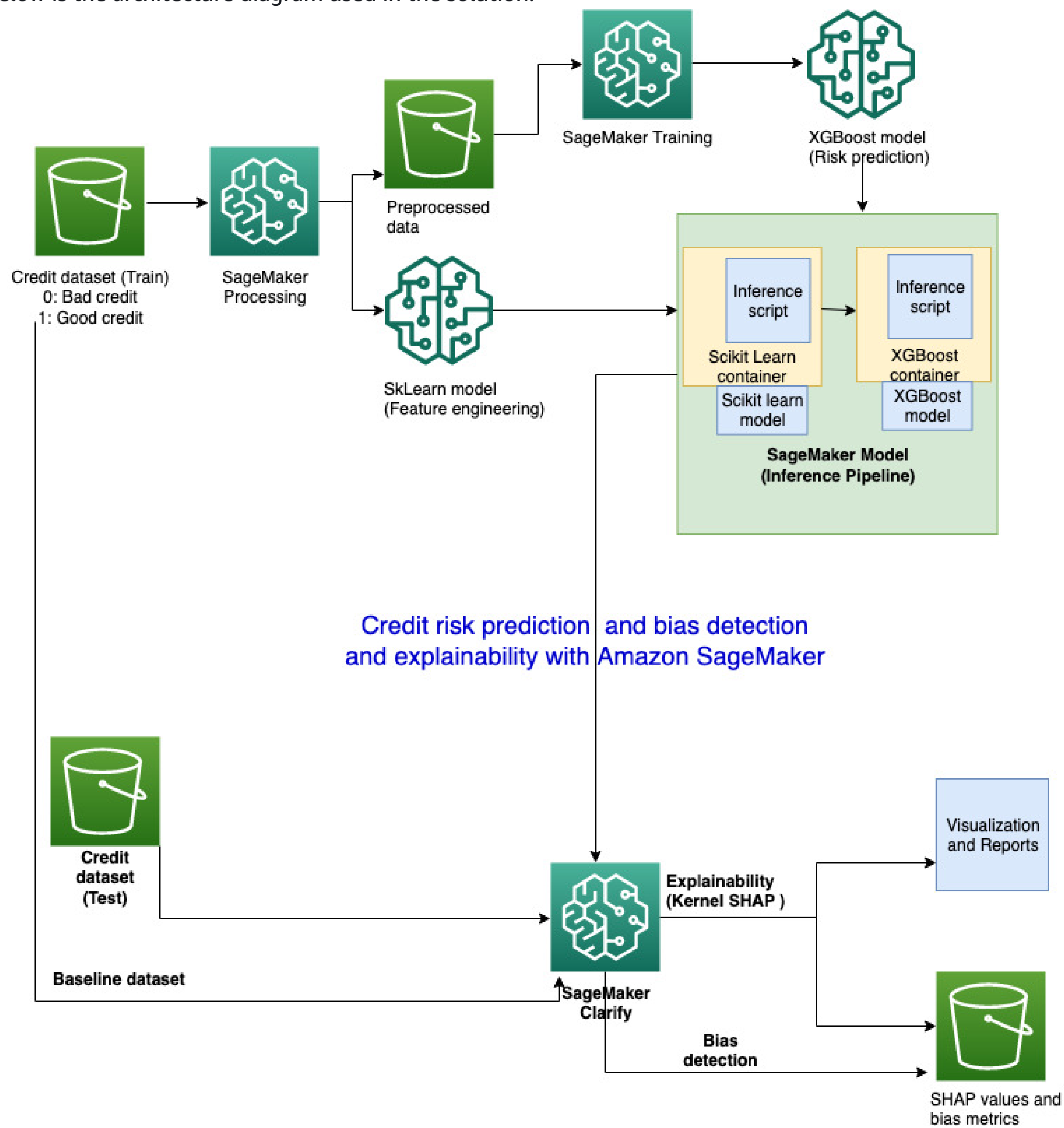
In this tutorial, you will learn highlighted section:



We use a well known open source dataset https://archive.ics.uci.edu/ml/datasets/South+German+Credit+%28UPDATE%29 . We show how to use SageMaker Clarify to run explainability and bias detection on a SageMaker inference pipeline model. We include the explainability and bias reports from Clarify as well as relevant links to further resources on this subject.

SageMaker Processing
Lab 2. Train, Tune and Deploy XGBoost
Lab 3. Bring your own model
Lab 3a. Bring your own Script (TensorFlow)
Lab 3b. Bring your own Script (PyTorch)
Lab3c. Bring your own Container
Lab 4. Autopilot, Debugger and Model Monitor
Autopilot
Debugger
Model Monitor
Lab 5. Bias and Explainability
Bias and Explainability-Tabular Data
Lab 6. SageMaker Pipelines
Lab 7. Real Time ML inference on Streaming Data
Lab 8. Build ML Model with No Code Using Sagemaker Canvas
Lab 9. Amazon SageMaker JumpStart
Lab 10. ML Governance Tools for Amazon SageMaker
Lab 11. SageMaker Notebook Instances
Content preferences
Language

Below is the architecture diagram used in the solution:



The notebook performs the following steps:

1. Prepare raw training and test data
2. Create a SageMaker Processing job which performs preprocessing on the raw training data and also produces an SKlearn model which is reused for deployment
3. Train an XGBoost model on the processed data using SageMaker's built-in XGBoost container

4. Create a SageMaker Inference pipeline containing the SKlearn and XGBoost model in a series
5. Perform inference by supplying raw test data
6. Set up and run explainability job powered by SageMaker Clarify
7. Use open source SHAP library to create summary and waterfall plots to understand the feature importance better
8. Run bias analysis jobs
9. Clean up

Launch the notebook instance

In your SageMaker Studio, in the "File Browser" pane on the left hand side open "sagemaker-clarify/amazon-sagemaker-credit-risk-prediction-explainability-bias-detection" folder , click on the file "credit_risk_explainability_inference_pipelines.ipynb".

You will be prompted to choose a kernel. Choose Image as "Data Science" and Kernel as Python 3 (Data Science).

Downloading and preparing your data

1. Run cell 01-03 by pressing Shift+Enter to Initialize SageMaker and download dataset

2. Prerequisites and Data exploration and Feature engineering

Initialize SageMaker

```
# cell 01
from io import StringIO
import os
import time
import sys
import IPython
from time import gmtime, strftime

import boto3
import numpy as np
import pandas as pd
import urllib

import sagemaker
from sagemaker.s3 import S3Uploader
from sagemaker.processing import ProcessingInput, ProcessingOutput
from sagemaker.sklearn.processing import SKLearnProcessor
from sagemaker.inputs import TrainingInput
from sagemaker.xgboost import XGBoost
from sagemaker.s3 import S3Downloader
from sagemaker.s3 import S3Uploader
from sagemaker import Session
from sagemaker import get_execution_role
from sagemaker.xgboost import XGBoostModel
from sagemaker.sklearn import SKLearnModel
from sagemaker.pipeline import PipelineModel


session = Session()
bucket = session.default_bucket()
prefix = "sagemaker/sagemaker-clarify-credit-risk-model"
region = session.boto_region_name

# Define IAM role
role = get_execution_role()
```

Download data

First, **download** the data and save it in the `data` folder.

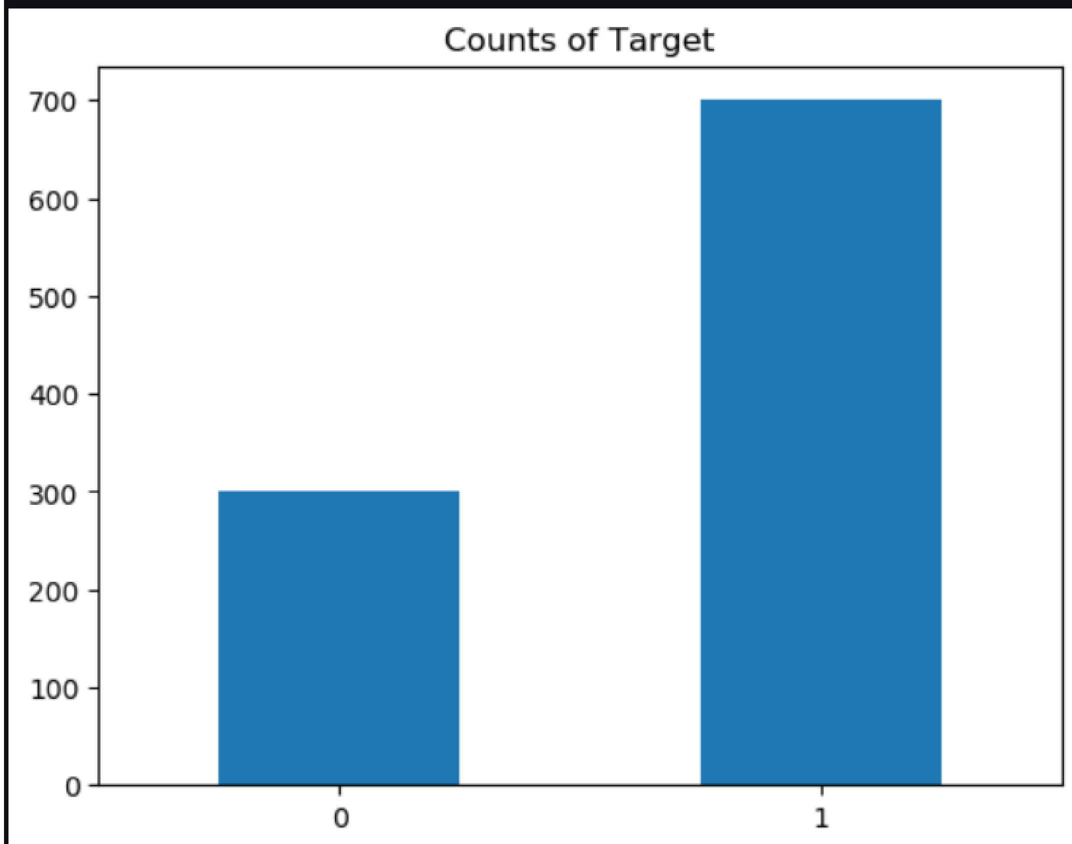
[2] Ulrike Grömping Beuth University of Applied Sciences Berlin Website with contact information: https://prof.beuth-hochschule.de/groemping/.

2. Run the next two cells 04 & 05 to inspect the data

Plotting histograms for the distribution of the different features is a good way to visualize the data.

```
# cell 05
%matplotlib inline
training_data["credit_risk"].value_counts().sort_values().plot(
    kind="bar", title="Counts of Target", rot=0
)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fb2fd211790>
```



3. Next in cell 06-08 we will encode the data and upload it to S3

Encode and Upload Data ¶

Here we encode the training and test data. Encoding input data is not necessary for SageMaker Clarify, but is necessary for XGBoost models.

```
# cell 07
test_raw = S3Uploader.upload(test_filename, "s3://{}/{}/data/test".format(bucket, prefix))
print(test_raw)
```

```
s3://sagemaker-us-east-[redacted]sagemaker/sagemaker-clarify-credit-risk-model/data/test/test.csv
```

```
# cell 08
train_raw = S3Uploader.upload(train_filename, "s3://{}/{}/data/train".format(bucket, prefix))
print(train_raw)
```

```
s3://sagemaker-us-east-1[redacted]sagemaker/sagemaker-clarify-credit-risk-model/data/train/train.csv
```

4. In cell 09,10 &11 We will use SageMaker Processing jobs to perform the preprocessing on the raw data

NOTE: THIS CELL WILL RUN FOR APPROX. 5-8 MINUTES! PLEASE BE PATIENT.

For further documentation on SageMaker Processing, you can refer the documentation here

```
# cell 11
raw_data_path = "s3://{0}/{1}/data/train/".format(bucket, prefix)
train_data_path = "s3://{0}/{1}/data/preprocessed/train/".format(bucket, prefix)
val_data_path = "s3://{0}/{1}/data/preprocessed/val/".format(bucket, prefix)
model_path = "s3://{0}/{1}/sklearn/".format(bucket, prefix)


sklearn_processor.run(
    code="processing/preprocessor.py",
    inputs=[
        ProcessingInput(
            input_name="raw_data", source=raw_data_path, destination="/opt/ml/processing/input"
        )
    ],
    outputs=[
        ProcessingOutput(
            output_name="train_data", source="/opt/ml/processing/train", destination=train_data_path
        ),
        ProcessingOutput(
            output_name="val_data", source="/opt/ml/processing/val", destination=val_data_path
        ),
        ProcessingOutput(
            output_name="model", source="/opt/ml/processing/model", destination=model_path
        ),
    ],
    arguments=["--train-test-split-ratio", "0.2"],
    logs=False,
)
```

```
INFO:sagemaker:Creating processing-job with name sagemaker-clarify-credit-risk-processin-2023-04-15-12-44-50-233
..................................................!
```

Train XGBoost Model

Content preferences
Language

Cell 12-14 to train a model using XGBoost Estimator for SageMaker

SageMaker Training

Now it's time to train the model

NOTE: THIS CELL WILL RUN FOR APPROX. 5-8 MINUTES! PLEASE BE PATIENT.

For further documentation on SageMaker Training, you can refer the documentation here

```
# cell 14
job_name = f"credit-risk-xgb-{strftime('%Y-%m-%d-%H-%M-%S', gmtime())}"

train_input = TrainingInput(
    "s3://{0}/{1}/data/preprocessed/train/".format(bucket, prefix), content_type="csv"
)
val_input = TrainingInput(
    "s3://{0}/{1}/data/preprocessed/val/".format(bucket, prefix), content_type="csv"
)

inputs = {"train": train_input, "validation": val_input}

estimator.fit(inputs, job_name=job_name)
```

```
INFO:sagemaker:Creating training-job with name: credit-risk-xgb-2023-04-15-12-49-14
2023-04-15 12:49:15 Starting - Starting the training job...
2023-04-15 12:49:30 Starting - Preparing the instances for training...
2023-04-15 12:50:14 Downloading - Downloading input data...
```

Create and deploy SageMaker inference pipeline model

Then Create SageMaker Model and deploy the model for inference using cells 15-24

4. Create SageMaker Model

We will be preparing a SageMaker inference pipeline model which can be deployed as an endpoint or used with SageMaker Clarify:

1. Accept raw data as input
2. preprocess the data with the SKlearn model we built earlier
3. Pass the output of the Sklearn model as an input to the XGBoost model automatically
4. Deliver the final inference result from the XGBoost model

To know more, check out the documentation on inference pipelines: https://docs.aws.amazon.com/sagemaker/latest/dg/inference-pipelines.html

Retrieve model artifacts

First, we need to create two Amazon SageMaker Model objects, which associate the artifacts of training (serialized model artifacts in Amazon S3) to the Docker container used for inference. In order to do that, we need to get the paths to our serialized models in Amazon S3. We define the model data location of SKlearn and XGBoost models here.

```
# cell 15
preprocessor_model_data = "s3://{}/{}/{}".format(bucket, prefix, "sklearn") + "/model.tar.gz"

xgboost_model_data = (
    "s3://{}/{}/{}/{}".format(bucket, prefix, "xgb_model", job_name) + "/output/model.tar.gz"
)
```

Content preferences

Language

Amazon SageMaker Clarify

1. Cell 25-30 to create SHAP baseline and setup Clarify configurations

5. Amazon SageMaker Clarify

Pre-requisities :

1. SageMaker Model that can be deployed to a endpoint
2. Input dataset

```
# cell 25
from sagemaker import clarify

clarify_processor = clarify.SageMakerClarifyProcessor(
    role=role, instance_count=1, instance_type="ml.c4.xlarge", sagemaker_session=session
)
```

2. Cell 31 will run SageMaker Clarify Explainability job

Run SageMaker Clarify Explainability job

All the configurations are in place. Let's start the explainability job. This will spin up an ephemeral SageMaker endpoint and perform inference and calculate explanations on that endpoint. It does not use any existing production endpoint deployments.

NOTE: THIS CELL WILL RUN FOR APPROX. 5-8 MINUTES! PLEASE BE PATIENT.

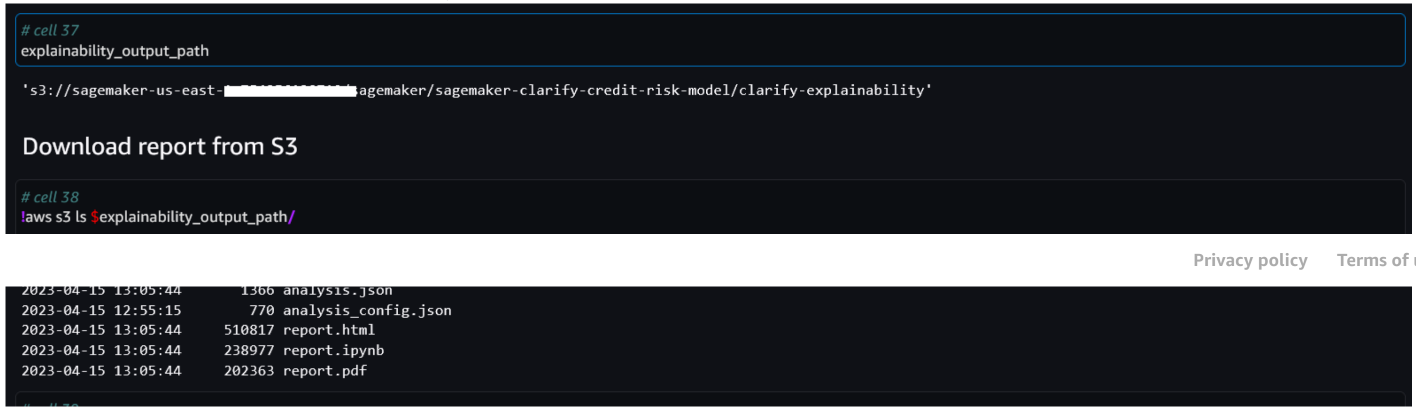
For further documentation on SageMaker Clarify , you can refer the documentation here

```
# cell 31
clarify_processor.run_explainability(
    data_config=explainability_data_config,
    model_config=model_config,
    explainability_config=shap_config,
)
```

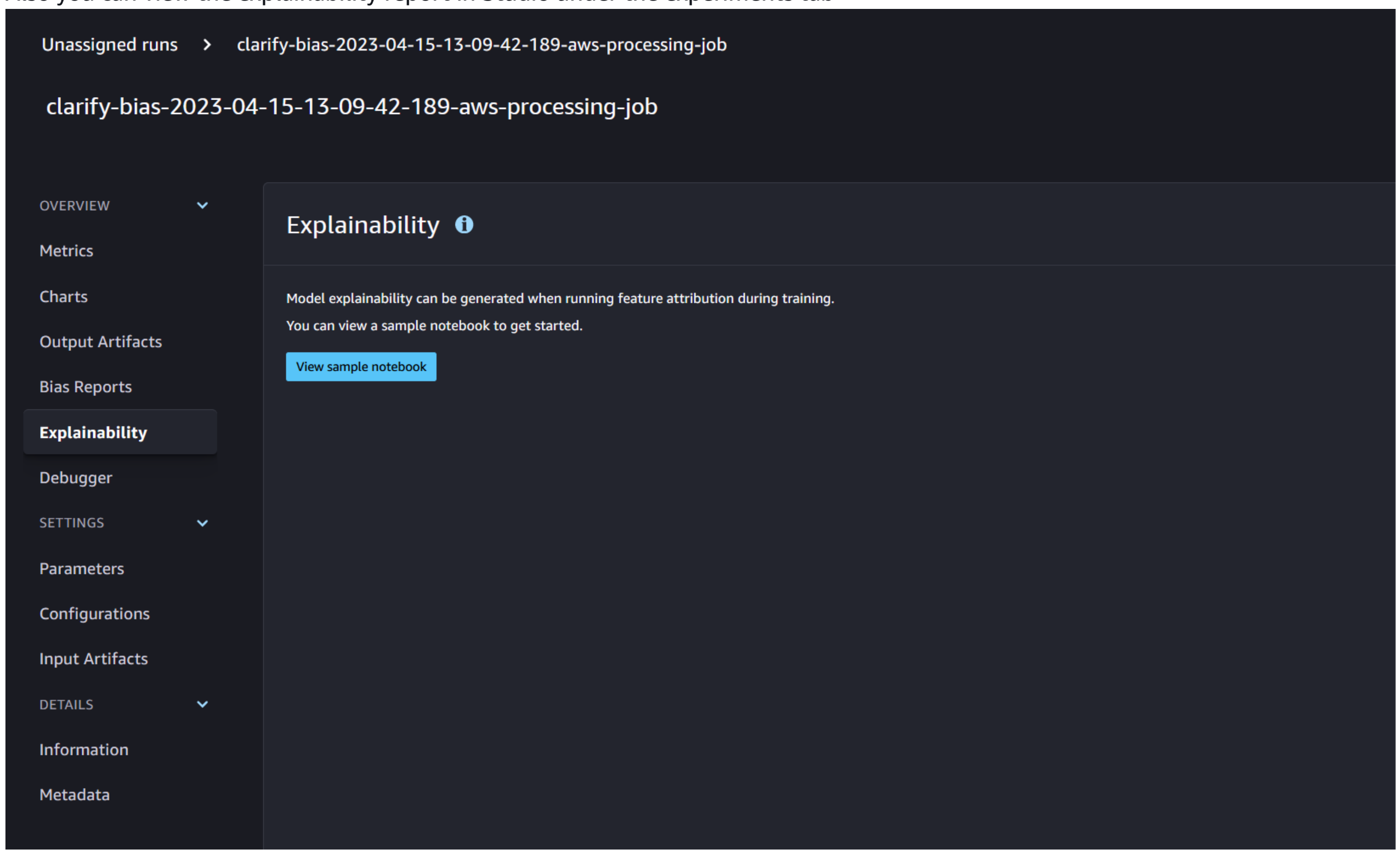
```
INFO:sagemaker.clarify:Analysis Config: {'dataset_type': 'text/csv', 'headers': ['status', 'duration', 'credit_history', 'purpose', 'amount', 'savings', 'employment_duration', 'installment_rate', 'personal_status_sex', 'other_debtors', 'present_residence', 'property', 'age', 'other_installment_plans', 'housing', 'number_credits', 'job', 'people_liable', 'telephone', 'foreign_worker'], 'predictor': {'model_name': 'credit-risk-inference-pipeline-1681563099', 'instance_type': 'ml.c5.xlarge', 'initial_instance_count': 1, 'accept_type': 'text/csv'}, 'methods': {'report': {'name': 'report', 'title': 'Analysis Report'}, 'shap': {'use_logit': True, 'save_local_shap_values': True, 'baseline': [[4, 24, 2, 3, 1258, 1, 3, 4, 3, 1, 4, 3, 27, 3, 2, 1, 3, 2, 1, 2]], 'num_samples': 2000, 'agg_method': 'mean_abs'}}}
INFO:sagemaker:Creating processing-job with name Clarify-Explainability-2023-04-15-12-55-14-155
.......................................2023-04-15 13:01:57,335 logging.conf not found when configuring logging, using default logging configuration.
2023-04-15 13:01:57,336 Starting SageMaker Clarify Processing job
2023-04-15 13:01:57,337 Analysis config path: /opt/ml/processing/input/config/analysis_config.json
2023-04-15 13:01:57,337 Analysis result path: /opt/ml/processing/output
2023-04-15 13:01:57,338 This host is algo-1.
2023-04-15 13:01:57,338 This host is the leader.
```

SageMaker Processing
Lab 2. Train, Tune and Deploy XGBoost
Lab 3. Bring your own model
Lab 3a. Bring your own
Lab 3b. Bring your own Script (PyTorch)
Lab3c. Bring your own Container
Lab 4. Autopilot, Debugger and Model Monitor
Autopilot
Debugger
Model Monitor
Lab 5. Bias and Explainability
Bias and Explainability-Tabular Data
Lab 6. SageMaker Pipelines
Lab 7. Real Time ML inference on Streaming Data
Lab 8. Build ML Model with No Code Using Sagemaker Canvas
Lab 9. Amazon SageMaker JumpStart
Lab 10. ML Governance Tools for Amazon SageMaker
Lab 11. SageMaker Notebook Instances
Content preferences
Language

3. Run cell 32 to 40 to view the explainability report and download the report from s3



4. Also you can view the explainability report in Studio under the experiments tab



Analyze the local explanations of individual predictions by Clarify

SageMaker Processing

Lab 2. Train, Tune and Deploy XGBoost

Lab 3. Bring your own model

Lab 3a. Bring your own Script (TensorFlow)

Lab 3b. Bring your own Script (PyTorch)

Lab3c. Bring your own Container

Lab 4. Autopilot, Debugger and Model Monitor

Autopilot

Debugger

Model Monitor

Lab 5. Bias and Explainability

Bias and Explainability-Tabular Data

Lab 6. SageMaker Pipelines

Lab 7. Real Time ML inference on Streaming Data

Lab 8. Build ML Model with No Code Using Sagemaker Canvas

Lab 9. Amazon SageMaker JumpStart

Lab 10. ML Governance Tools for Amazon SageMaker

Lab 11. SageMaker Notebook Instances

Content preferences

Language

In this section, we will analyze and understand the local explainability results for each individual prediction produced by Clarify. Clarify produces a CSV file which contains the SHAP value for each feature per prediction. Let us download the CSV.

1. Run cell 41 read the shap values and cell 42 to get the base expected value to be used to plot SHAP values
2. Run cell 44 and 45 to join the predictions with the SHAP values and convert the the probability score to binary prediction
3. To filder for bad credit prediction only run cell 46

Filter for bad credit predictions only

Since we interested in explaining negative outcomes (bad credit predictions) only in this use case, we filter the records to keep only the record with prediction as 0.

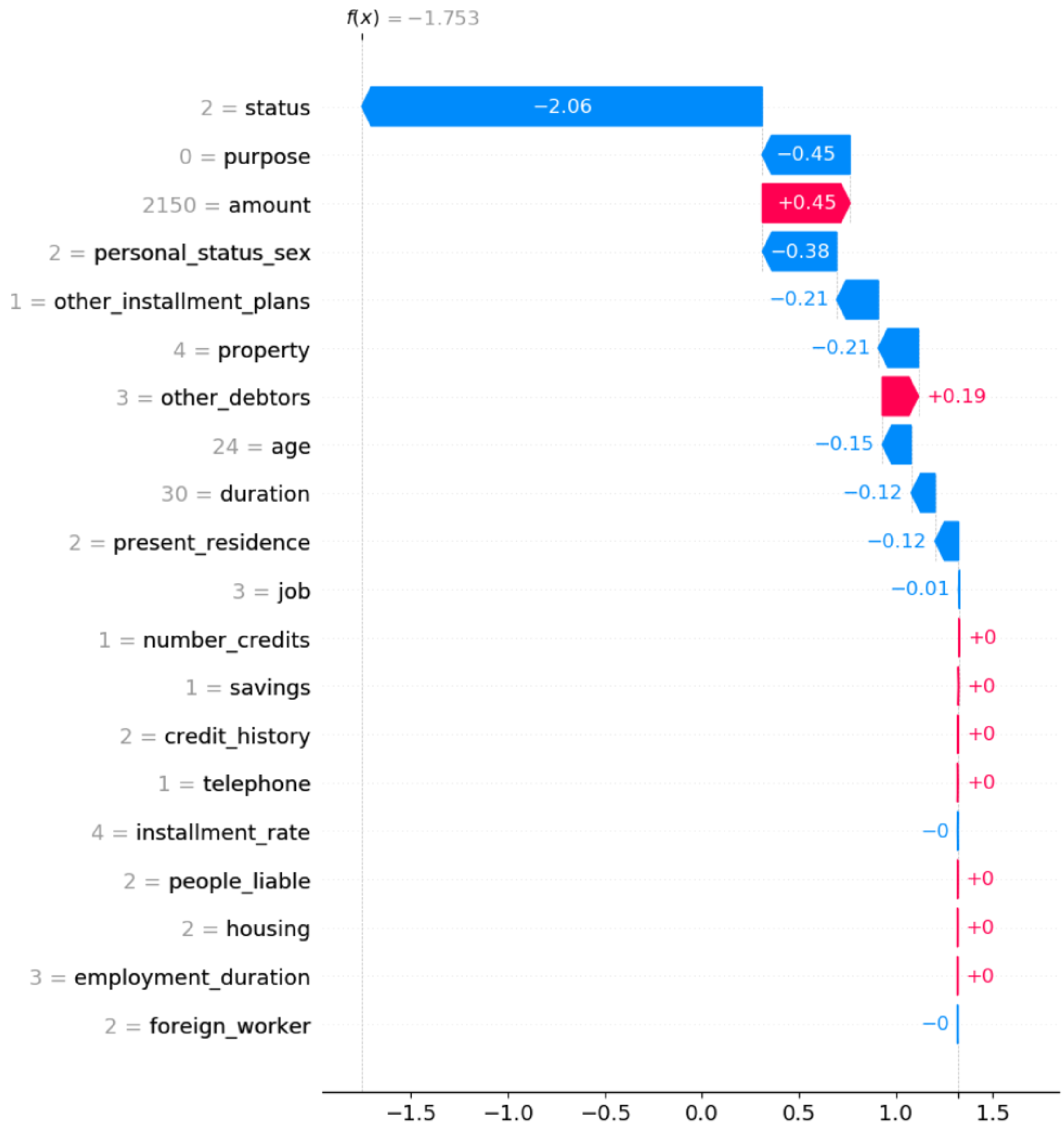
```
# cell 46
bad_credit_outcomes_df = prediction_shap_df[prediction_shap_df.iloc[:, 0] == 0]
bad_credit_outcomes_df
```

	Prediction	probability_score	status_label0	duration_label0	credit_history_label0	purpose_label0	amount_label0	savings_label0	employment_duration_label0	installment_rate_label0	...
14	0	0.381345	-1.735907	0.816273	0.000674	-0.432770	-0.920758	0.008045	0.317587	0.483758	...
17	0	0.461101	-2.051966	0.404304	-0.000794	-0.000543	0.023138	0.004703	0.668121	-0.000457	...
19	0	0.412635	-2.151303	-0.008167	-0.005127	0.003120	0.957666	0.005611	-0.255711	-0.005038	...
22	0	0.339406	-1.976879	-0.239421	0.004627	-0.085219	0.563554	-0.003157	-0.317378	0.002849	...
25	0	0.405300	-1.670402	-0.074422	0.002690	-0.752208	0.357435	-0.001149	0.444318	0.123430	...
27	0	0.385872	-2.248718	-0.000350	0.001519	-0.087650	0.614043	0.000502	-0.000316	-0.002260	...
28	0	0.276659	-2.359466	0.650672	-0.002653	0.002708	0.061002	0.048232	-0.309578	-0.001255	...
32	0	0.421549	-2.084032	0.011939	-0.003016	-0.086260	0.318748	0.000938	0.001740	0.490071	...
39	0	0.398601	-2.149048	0.862798	-0.003274	-0.336365	0.359174	-0.001709	-0.283995	0.090558	...

4. Run cell 47- 50 to plot SHAP explanation for a single bad credit ensemble prediction instance. We will select the prediction instance with the lowest probability

Content preferences

Language



Detect data bias with Amazon SageMaker Clarify

1. Run cell 51-53 to Calculate pre-training and post-training Bias metrics

NOTE: THIS CELL WILL RUN FOR APPROX. 5-8 MINUTES! PLEASE BE PATIENT.

For further documentation on SageMaker Clarify, you can refer the documentation here

```
# cell 53
clarify_processor.run_bias(
    data_config=bias_data_config,
    bias_config=bias_config,
    model_config=model_config,
    model_predicted_label_config=predictions_config,
    pre_training_methods="all",
    post_training_methods="all",
)
```

```
INFO:sagemaker.clarify:Analysis Config: {'dataset_type': 'text/csv', 'headers': ['status', 'duration', 'credit_history', 'purpose', 'amount', 'savings', 'employment_duration', 'installment_rate', 'personal_status_sex', 'other_debtors', 'present_residence', 'property', 'age', 'other_installment_plans', 'housing', 'number_credits', 'job', 'people_liable', 'telephone', 'foreign_worker', 'credit_risk'], 'label': 'credit_risk', 'label_values_or_threshold': [1], 'facet': [{'name_or_index': 'age', 'value_or_threshold': [40]}], 'group_variable': 'personal_status_sex', 'methods': {'report': {'name': 'report', 'title': 'Analysis Report'}, 'pre_training_bias': {'methods': 'all'}, 'post_training_bias': {'methods': 'all'}}, 'predictor': {'model_name': 'credit-risk-inference-pipeline-1681563099', 'instance_type': 'ml.c5.xlarge', 'initial_instance_count': 1, 'accept_type': 'text/csv', 'probability': 0}}
INFO:sagemaker:Creating processing-job with name Clarify-Bias-2023-04-15-13-09-42-189
.....................................2023-04-15 13:16:12,121 logging.conf not found when configuring logging, using default logging configuration.
2023-04-15 13:16:12,122 Starting SageMaker Clarify Processing job
```

2. Run cell 54-62 to view the Bias detection Report at S3 bucket.

Viewing the Bias detection Report

You can view the bis detection report in Studio under the experiments tab

If you're not a Studio user yet, you can access this report at the following S3 bucket.

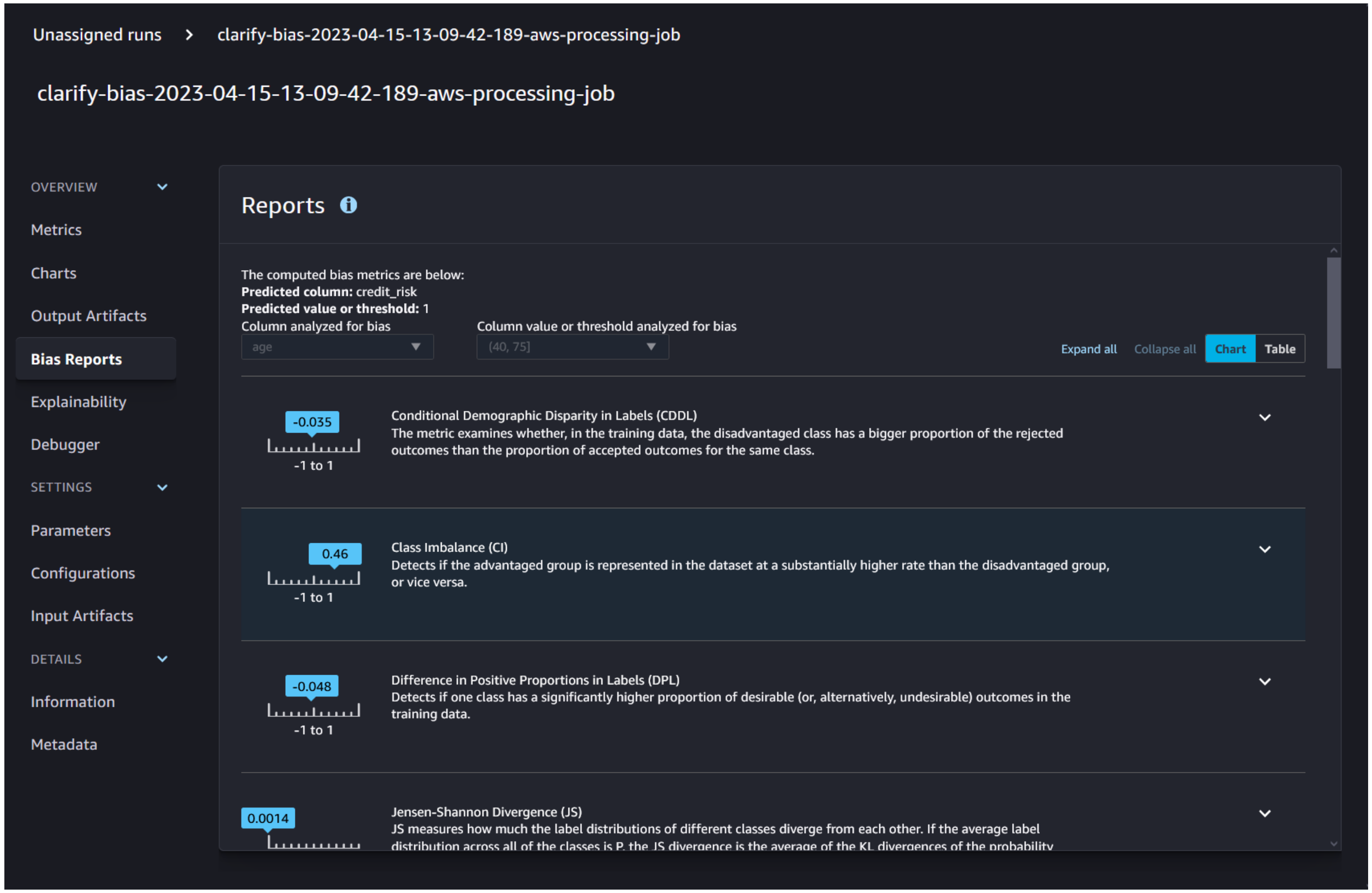
```
# cell 54
bias_report_output_path
```

```
's3://sagemaker-us-east-1[redacted]/sagemaker/sagemaker-clarify-credit-risk-model/clarify-bias'
```

```
# cell 55
run_post_training_bias_processing_job_name = clarify_processor.latest_job.job_name
run_post_training_bias_processing_job_name
```

```
'Clarify-Bias-2023-04-15-13-09-42-189'
```

3. You can also view the bias detection report in Studio under the experiments tab



4. Run cell 63 to look at a couple of pre-training and post-training bias mertics

```
# cell 63
S3Downloader.download(s3_uri=bias_report_output_path + "/analysis.json", local_path="output")

with open("output/analysis.json") as json_file:
    data = json.load(json_file)
    print("pre-training bias metrics")
    class_imbalance = data["pre_training_bias_metrics"]["facets"]["age"][0]["metrics"][1]["value"]
    print("class imbalance: ", class_imbalance)
    DPL = data["pre_training_bias_metrics"]["facets"]["age"][0]["metrics"][2]["value"]
    print("DPL: ", DPL)
    print("\n")
    print("post training bias metrics")
    DPPL = data["post_training_bias_metrics"]["facets"]["age"][0]["metrics"][6]["value"]
    print("DPPL: ", DPPL)
    DI = data["post_training_bias_metrics"]["facets"]["age"][0]["metrics"][5]["value"]
    print("DI: ", DI)
```

```
pre-training bias metrics
class imbalance:  0.456
DPL:  -0.04848093083387206


post training bias metrics
DPPL:  -0.03462346477052358
DI:  1.0436843714955653
```

Clean up

Once you done with this notebook, you can run cell 64 and 65 to remove the hosted endpoint to avoid any charges.

6. Clean Up

Finally, don't forget to clean up the resources we set up and used for this demo!

```
# cell 64
session.delete_endpoint(endpoint_name)
```

```
INFO:sagemaker:Deleting endpoint with name: credit-risk-pipeline-endpoint-1681563099
```

```
# cell 65
session.delete_model(pipeline_model.name)
```

```
INFO:sagemaker:Deleting model with name: credit-risk-inference-pipeline-1681563099
```

Conclusion and Additional Resources to explore

In this lab you walked through the process of using SageMaker Clarify to run Bias and Explainability reports.

- Working toward fairer machine learning
- Fairness Measures for Machine Learning in Finance
- Amazon SageMaker Clarify: Machine learning bias detection and explainability in the cloud
- Amazon AI Fairness and Explainability Whitepaper
- How Clarify helps machine learning developers detect unintended bias

Previous Next