Copyright Notice

1. A better understanding of the data and its quality
2. How individual features of the dataset contribute to the ML model
3. High quality ML models require high quality training data. - Models are only going to be as good as the training data.

This week we will learn how to
1. describe the concept of statistical bias and use metrics to measure imbalances in the datasets.
2. detect statistical bias in your data and bias reports.
3. generate feature importance reports - to understand how the individual features in your dataset contribute to the final model.

DeepLearning.AI

aws

Practical Data Science

Statistical Bias and Feature Importance

Machine Learning Workflow

Ingest & Analyze	Prepare & Transform	Train & Tune	Deploy & Manage
Data exploration	Feature engineering	Automated ML	Model deployment
Bias detection	Feature store	Model train and tune	Automated pipelines
Amazon S3 & Amazon Athena	Amazon SageMaker Data Wrangler	Amazon SageMaker Autopilot	Amazon SageMaker Endpoints
AWS Glue	Amazon SageMaker Processing Jobs	Amazon SageMaker Training & Debugger	Amazon SageMaker Batch Transform
Amazon SageMaker Data Wrangler & Clarify	Amazon SageMaker Feature Store	Amazon SageMaker Hyperparameter Tuning	Amazon SageMaker Pipelines

To perform bias detection on your training data sets.

Statistical Bias

Statistical Bias

Statistical Bias: When the data sets cannot completely and accurately represent the underlying problem space.

Bias leads to overestimate or underestimate a paramater.

- Training data does not comprehensively represent the problem space

Learn about Statistical Biases in training data sets which are imbalances in these training data sets.
In these biased data sets some elements of a data set are heavily represented than others.

- Some elements of a dataset are more heavily weighted or represented

Eg. Fraud Detection Model: where majority of the data is non fraudulent. If we use this data to train a model, then it is very unlikely that the model is good at detecting frauds. To address this add more fraudulent data to the model.

Fraud Detection

Biased models

- Imbalances in product review dataset

Statistical Bias Causes

Statistical Bias – Causes

Statistical Bias – Causes

Activity Bias

Social Media Content

Statistical Bias – Causes

Activity Bias

Social Media Content

Societal Bias

Human Generated Content

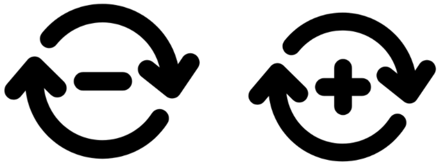
Statistical Bias – Causes

Activity Bias

Social Media Content

Societal Bias

Human Generated Content



Selection Bias

Feedback loop

Statistical Bias – Causes

Activity Bias

Social Media Content

Biases that exist in human generated content, especially on social media.

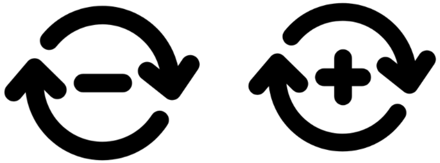
Younger population are more so there are biases in the data collected.

Societal Bias

Human Generated Content

Bias in data generated by humans but not just on social media.

Biases due to preconceived notions that exist in society. Since all of us have unconscious bias.



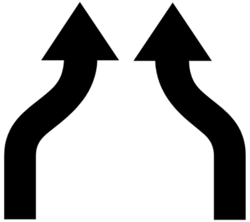
Selection Bias

Feedback Loop

Bias introduced by machine learning system itself when there are feedback loops involved.

You like "Dancing with wolves" for the actress not bcoz of wolves.

Similarly, in CNN Husky vs Warevolves. Detect werewolf because of snow not face.



Data Drift or Data Shift

Data Drift

Independent Vaiable distribution changes

- *Covariant Drift*

When the dist. of labels change

- *Prior probability Drift*

When the relationship b/w the 2 changes.

- *Concept Drift*

Once the model is trained and deployed "data drift" can still happen.

This happens when data distribution significantly varies from the training data

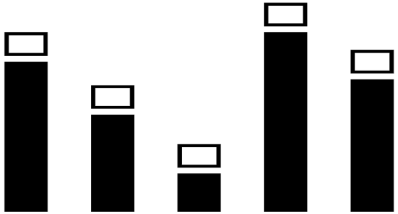
Measuring Statistical Bias

Measure the imbalances and the statistical bias in your data set using specific metrics.

It's important to understand that these metrics are applicable to a particular facet of your dataset. A facet is a sensitive feature in your dataset, that you want to analyze for these imbalances. It is a feature that you want to analyze for imbalances.

Measure Imbalance in Data - Metrics

Class Imbalance is for multi-class feature.
When you have data of a certain class proportionately higher or lower than other classes, then we have Class Imbalance (CI)



Class Imbalance (CI)

- Measures the imbalance in the number of members between different facet values.
- Does a ***product_category*** has disproportionately more reviews than others?

Measure Imbalance in Data - Metrics

Difference in Proportions of Labels (DPL) ✓ ✗

- Measures the imbalance of positive outcomes between different facet values.
- Does a ***product_category*** has disproportionately higher ratings than others

CI looks for disproportionate higher number of total data. DPL looks for disproportionately higher ratings than others.

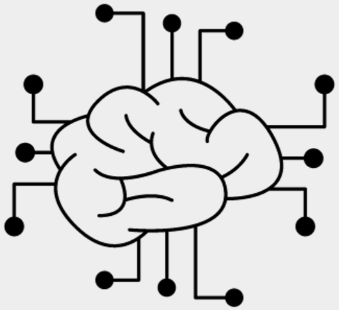
Detecting Statistical Bias

- Amazon SageMaker Data Wrangler
- Amazon SageMaker Clarify

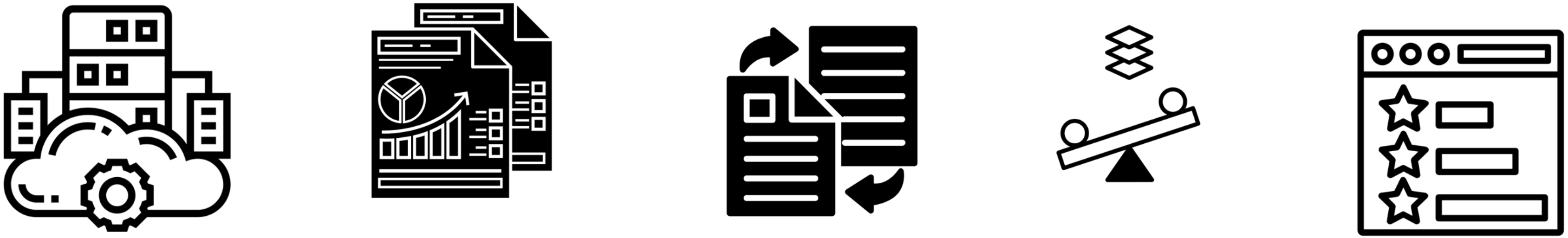
AWS Toolkit

Detect Statistical Bias

Amazon SageMaker Data Wrangler



Detect Statistical Bias - Amazon SageMaker Data Wrangler



Source

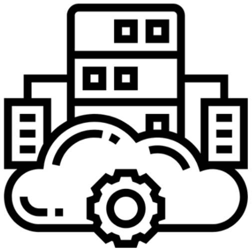
Visualization

Transform

Statistical Bias Report

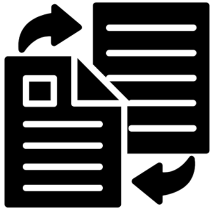
Feature Importance

Detect Statistical Bias - Amazon SageMaker Data Wrangler



Source

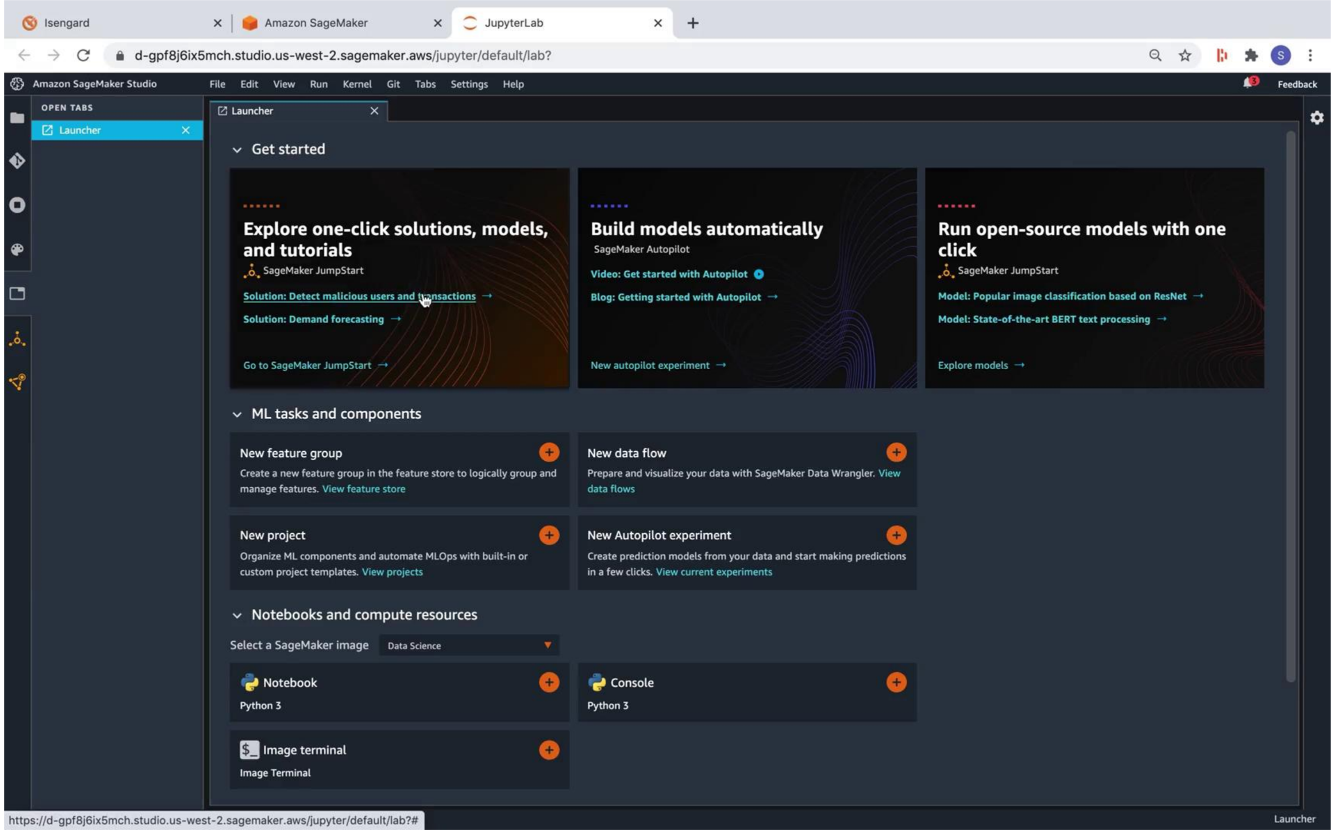
Visualization



Transform

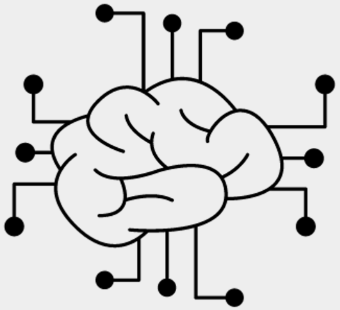
Statistical Bias Report

Feature Importance

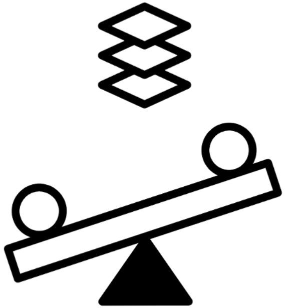

Isengard
Amazon SageMaker
JupyterLab
d-gpf8j6ix5mch.studio.us-west-2.sagemaker.aws/jupyter/default/lab?
Amazon SageMaker Studio
File Edit View Run Kernel Git Tabs Settings Help
Feedback
OPEN TABS
Launcher
Get started
Explore one-click solutions, models, and tutorials
SageMaker JumpStart
Solution: Detect malicious users and transactions →
Solution: Demand forecasting →
Go to SageMaker JumpStart →
Build models automatically
SageMaker Autopilot
Video: Get started with Autopilot
Blog: Getting started with Autopilot →
New autopilot experiment →
Run open-source models with one click
SageMaker JumpStart
Model: Popular image classification based on ResNet →
Model: State-of-the-art BERT text processing →
Explore models →
ML tasks and components
New feature group
Create a new feature group in the feature store to logically group and manage features. View feature store
New data flow
Prepare and visualize your data with SageMaker Data Wrangler. View data flows
New project
Organize ML components and automate MLOps with built-in or custom project templates. View projects
New Autopilot experiment
Create prediction models from your data and start making predictions in a few clicks. View current experiments
Notebooks and compute resources
Select a SageMaker image
Data Science
Notebook
Python 3
Console
Python 3
Image terminal
Image Terminal
https://d-gpf8j6ix5mch.studio.us-west-2.sagemaker.aws/jupyter/default/lab?#
Launcher

Detect Statistical Bias

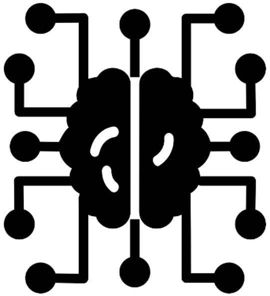
Amazon SageMaker Clarify



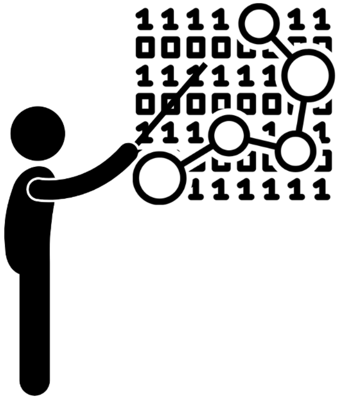
Detect Statistical Bias - Amazon SageMaker Clarify



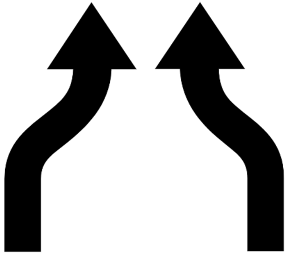
Statistical Bias Report



Model Bias Report

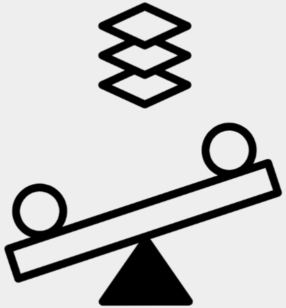


Explainability

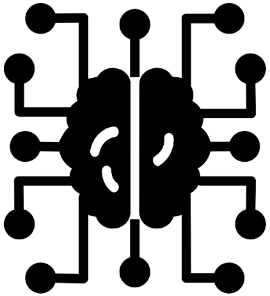


Drift

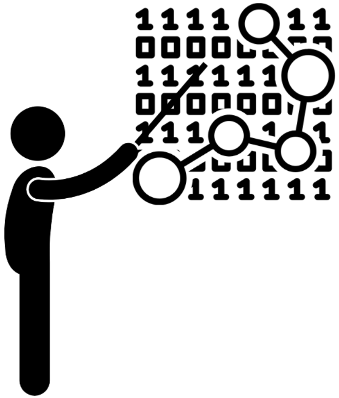
Detect Statistical Bias - Amazon SageMaker Clarify



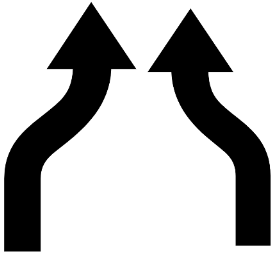
Statistical Bias Report



Model Bias Report



Explainability



Drift

Detect Statistical Bias - Amazon SageMaker Clarify

```
from sagemaker import clarify

clarify_processor = clarify.SageMakerClarifyProcessor(
    role=role,
        instance_count=1,
        instance_type='ml.c5.2xlarge',
        sagemaker_session=sess)

bias_report_output_path = << Define S3 path >>
```

Distributed cluster size

Type of each instance

S3 path where the bias report is saved to

S3 location to store bias report

Once you have the Clarify library, construct the object, SageMaker Clarify Processor using the library. SageMaker Clarify Processor is a construct that allows you to scale the bias detection process into a distributed cluster. By using two parameters, instance type and instance count, you can scale up the distributed cluster to the capacity that you need. Instant count represents the number of nodes that are included in the cluster, and instance type represents the processing capacity of each individual node in the cluster. The processing capacity is measured by the node's compute capacity, memory, and the network I/O.

Detect Statistical Bias - Amazon SageMaker Clarify

```
bias_data_config = clarify.DataConfig(
          s3_data_input_path=...,
          s3_output_path=...,
          label='sentiment',
          headers=df_balanced.columns.to_list(),
          dataset_type='text/csv')
```

Data Configuration

Next step is to configure the data config object on the Clarify library. The data config object represents the details about your data. So as you can expect, it has the input and output location of your data, in S3, as well as the label that you're trying to predict, using that dataset. In this case here, that label that we are trying to predict is sentiment.

Detect Statistical Bias - Amazon SageMaker Clarify

```
bias_config = clarify.BiasConfig(
    label_values_or_threshold=[...],
    facet_name='product_category')
```

Bias Configuration

Next, you configure the bias config object on Clarify library. The bias config object captures the facet or the featured name that you are trying to evaluate for bias or imbalances. In this case, you're trying to find out imbalances in the product category feature. The parameter label values or threshold defines the desired values for the labels. So if the sentiment feature is your label, what is the desired value for that label? That value goes into the parameter label values or threshold. Once you have configured those three objects, you are ready to run the pre-training bias method on the Clarify processor.

Detect Statistical Bias - Amazon SageMaker Clarify

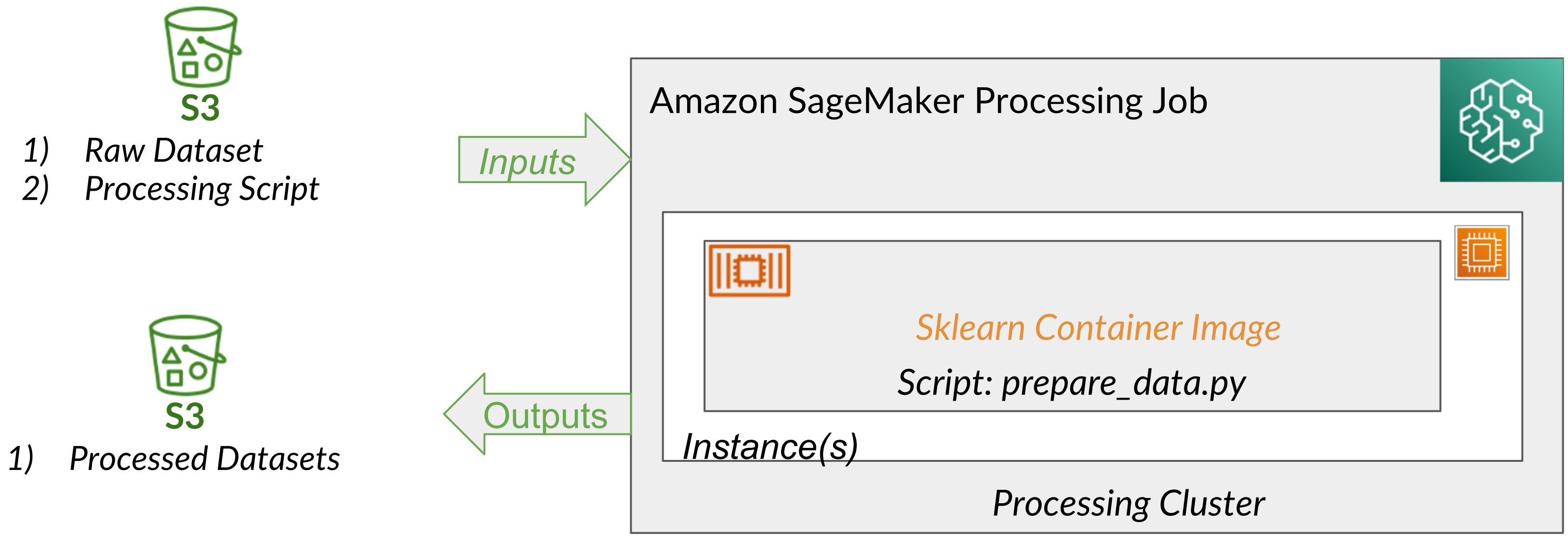
```
clarify_processor.run_pre_training_bias(
    data_config=...,
    data_bias_config=...,
    methods=["CI", "DPL", ...],
    wait=<<False/True>>,
    logs=<<False/True>>)
```

Pre training bias job

In addition to specifying the data config and the data bias config that you already configured, you can also specify the methods that you want to evaluate for bias. So, these methods are basically the metrics that you've already learned about to detect bias. The metrics here are the CI, the class imbalance, and the DPL. You can also specify a few other methods here as well. The wait parameter specifies whether this bias detection job should block the rest of your code or should it be executed in the background. Similarly, logs parameter specify whether you want to capture the logs or not. Once the configuration of the pre-training bias method is done, you launch this job.

Amazon SageMaker Processing

Execute preprocessing, post processing, model evaluation



In the background, SageMaker Clarify is using a construct called SageMaker Processing Job to execute the bias detection at scale. SageMaker Processing Jobs is a construct that allows you to perform any data-related tasks at scale. These tasks could be executing pre-processing, or post-processing tasks, or even using data to evaluate your model. As you can see in the figure here, the SageMaker Processing Job expects the data to be in an S3 bucket. The data is collected from the S3 bucket and processed on this processing cluster which contains a variety of containers in the cluster. By default, containers for Sklearn, Python, and a few others are supported. You can also have the opportunity to bring your own custom container as well. Once the processing cluster has processed the data, the transformed data or the processed data is put back in the S3 bucket.

Detect Statistical Bias - Amazon SageMaker Clarify

```
clarify_processor.run_pre_training_bias(
    data_config=...,
    data_bias_config=...,
    methods=["CI", "DPL", ...],
    wait=<<False/True>>,
    logs=<<False/True>>)
```

Result ?

The result will actually be a very detailed report on the bias on your dataset that has persisted in S3 bucket. You can download the report and review in detail to understand the behavior of your data.

Detecting Statistical Bias - Two Approaches

- Amazon SageMaker Data Wrangler

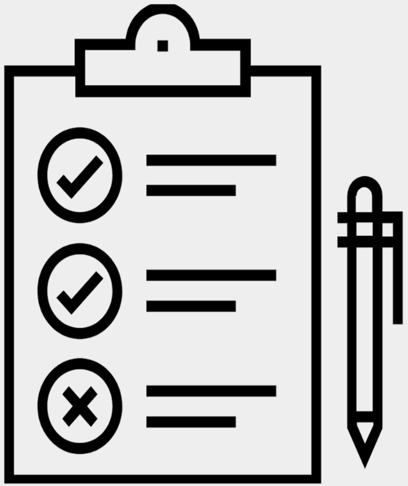
Data Wrangler provides a UI based visual experience for Statistical Bias.
- connect to multiple data sources
- make selections from drop downs and button click.
Data Wrangler is only using a subset of your data to detect bias in that data set.

- Amazon SageMaker Clarify

On the other hand, SageMaker Clarify provides you with a more of an API-based approach.
Additionally, Clarify also provides you with the ability to scale out the bias detection process.
It uses a construct called processing jobs that allow you to configure a distributed cluster to execute your bias detecting job at scale.

Feature Importance

SHAP



Feature Importance

- Explains the features that make up the training data using a score (importance).
- How useful or valuable the feature is relative to other features.
- Predict the sentiment for a product → Which features play a role?

Feature Importance

- **Open Source Framework - SHapley Additive exPlanations**

Feature Importance

- **Open Source Framework - SHAP**
  - Shapley values based on game theory.

Feature Importance

- **Open Source Framework - SHAP**
  - Shapley values based on game theory.
  - Explain predictions of a ML model
    - Each feature value of training data instance is a player in a game
    - ML prediction is the payout

Feature Importance

- **Open Source Framework - SHAP**
  - Shapley values based on game theory.
  - Explain predictions of a ML model
    - Each feature value of training data instance is a player in a game
    - ML prediction is the payout
  - Local vs global explanations

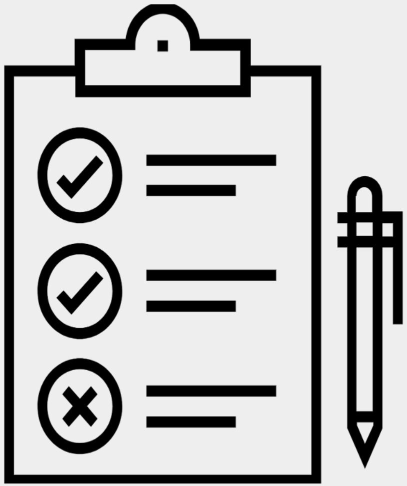
Feature Importance

- **Open Source Framework - SHAP** SHapley Additive exPlanations
  - Shapley values based on game theory. Consider a game in which many players are involved and there is a specific outcome to the play that could be either a win or a loss. Shapley values allow you to attribute the outcome of the game to the individual players involved in the game.
  - Explain predictions of a ML model Individual players -----> Individual features.
    The outcome of the play would be the machine learning model prediction.
    - Each feature value of training data instance is a player in a game
    - ML prediction is the payout
  - Local vs global explanations Local Explanation --> Focuses on indicating how an individual feature contributes to the final model.

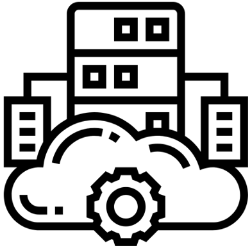
    Global Explanation -> takes a much more comprehensive view in trying to understand how the data in its entirety contributes to the final outcome.
  - SHAP can guarantee consistency and local accuracy.
    Extensive in nature, in that it considers all possible combinations of feature values along with all possible outcomes for your ML model.
    Usually time incentive but provides consistency and local accuracy.

Feature Importance

Amazon SageMaker Data Wrangler

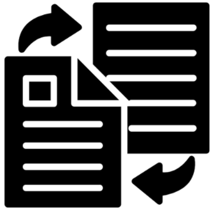


Feature Importance - Amazon SageMaker Data Wrangler

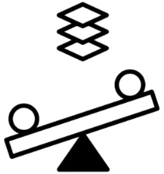


Source

Visualization



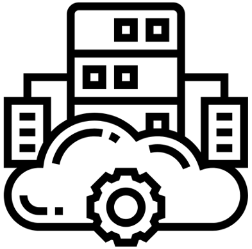
Transform



Bias Report

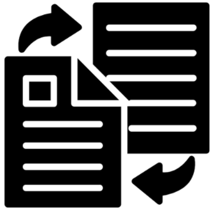
Feature Importance

Feature Importance - Amazon SageMaker Data Wrangler

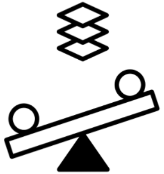


Source

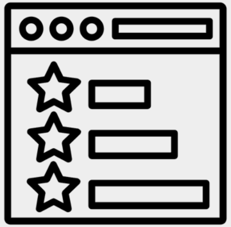
Visualization



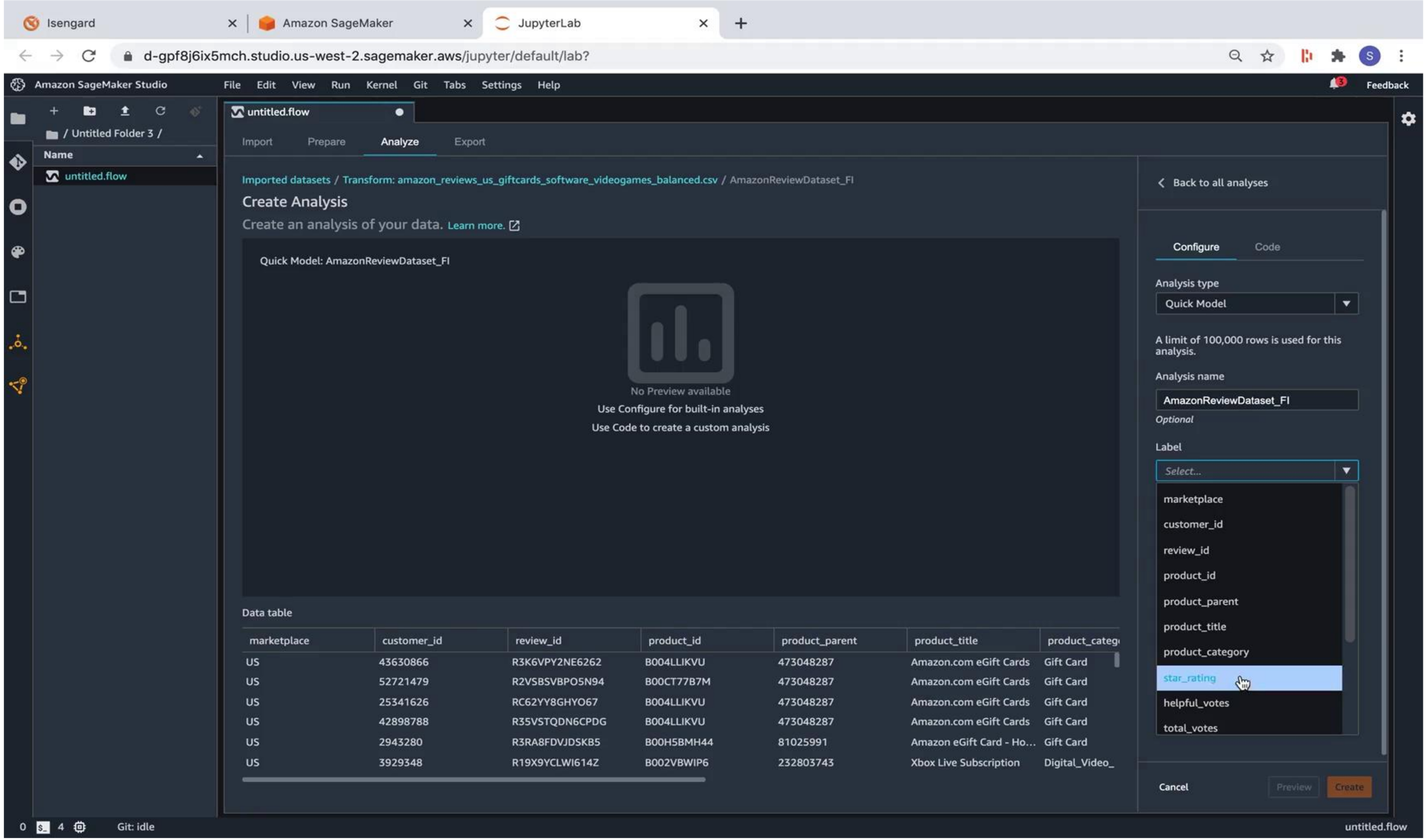
Transform



Bias Report



Feature Importance



DeepLearning.AI

aws