02

Fairness in AI

Introduction to Responsible AI in Practice

In this module, you learn to ...

01 Define (some types of) unfair **bias**

02 Discuss why **fairness is important** and **difficult**

03 Discover some **best practices** on fairness

04 Explore **tools** to study fairness in **datasets** and **models**

05 **Lab:** Using **TensorFlow Data Validation** and **TensorFlow Model Analysis** to Ensure Fairness

Topics

01	Overview of Fairness
02	Tools to Study Fairness in Datasets
03	Tools to Study Fairness in Models
04	Hands-on Lab

Topics

01	Overview of Fairness
02	Tools to Study Fairness in Datasets
03	Tools to Study Fairness in Models
04	Hands-on Lab

Fairness relates to Google's AI Principle #2

1. Be socially beneficial
2. **Avoid creating or reinforcing unfair bias**
3. Be built and tested for safety
4. Be accountable to people
5. Incorporate privacy design principles
6. Uphold high standards of scientific excellence
7. Be made available for uses that accord with these principles

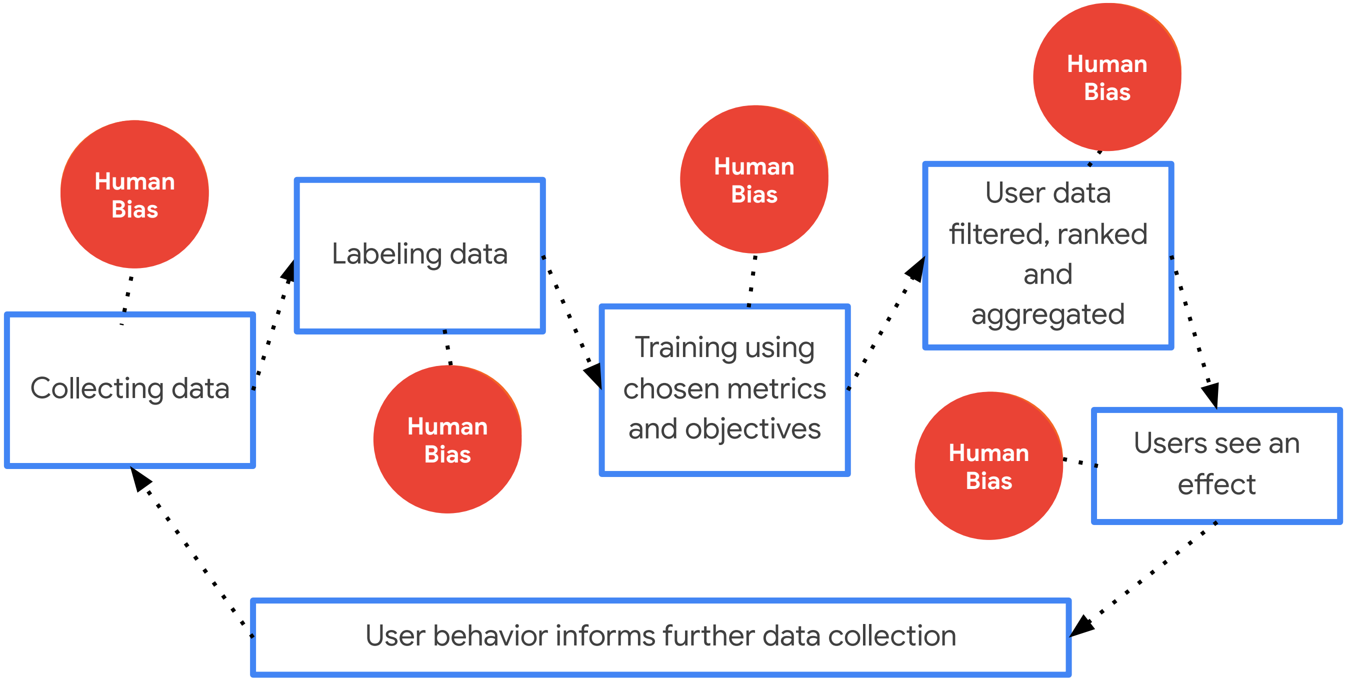
What is bias?

“

Stereotyping, prejudice or favoritism towards some things, people, or groups over others.

Definition from https://developers.google.com/machine-learning/glossary/fairness

What is bias?



AI models are **not** inherently objective.

What types of bias exist?

Reporting	Automation	Selection	Group Attribution	Implicit
Frequency of events, properties, and/or outcomes in a data set does not accurately reflect their real-world frequency.	Tendency to favor results generated by automated systems over those generated by non-automated systems.	A data set's examples are chosen in a way that is not reflective of their real-world distribution.	Tendency to generalize what is true of individuals to an entire group to which they belong.	Assumptions are made based on one's own mental models and personal experiences that do not necessarily apply more generally.

There are over 100 different types of human biases in Wikipedia's catalog of cognitive biases

What types of bias exist?

Reporting	Automation	Selection	Group Attribution	Implicit
Frequency of events, properties, and/or outcomes in a data set does not accurately reflect their real-world frequency.	Tendency to favor results generated by automated systems over those generated by non-automated systems.	A data set's examples are chosen in a way that is not reflective of their real-world distribution.	Tendency to generalize what is true of individuals to an entire group to which they belong.	Assumptions are made based on one's own mental models and personal experiences that do not necessarily apply more generally.

There are over 100 different types of human biases in Wikipedia's catalog of cognitive biases

What types of bias exist?

Reporting	Automation	Selection	Group Attribution	Implicit
Frequency of events, properties, and/or outcomes in a data set does not accurately reflect their real-world frequency.	Tendency to favor results generated by automated systems over those generated by non-automated systems.	A data set's examples are chosen in a way that is not reflective of their real-world distribution.	Tendency to generalize what is true of individuals to an entire group to which they belong.	Assumptions are made based on one's own mental models and personal experiences that do not necessarily apply more generally.

There are over 100 different types of human biases in Wikipedia's catalog of cognitive biases

What types of bias exist?

Reporting	Automation	Selection	Group Attribution	Implicit
Frequency of events, properties, and/or outcomes in a data set does not accurately reflect their real-world frequency.	Tendency to favor results generated by automated systems over those generated by non-automated systems.	A data set's examples are chosen in a way that is not reflective of their real-world distribution.	Tendency to generalize what is true of individuals to an entire group to which they belong.	Assumptions are made based on one's own mental models and personal experiences that do not necessarily apply more generally.

There are over 100 different types of human biases in Wikipedia's [catalog of cognitive biases]

What types of bias exist?

Reporting	Automation	Selection	Group Attribution	Implicit
Frequency of events, properties, and/or outcomes in a data set does not accurately reflect their real-world frequency.	Tendency to favor results generated by automated systems over those generated by non-automated systems.	A data set's examples are chosen in a way that is not reflective of their real-world distribution.	Tendency to generalize what is true of individuals to an entire group to which they belong.	Assumptions are made based on one's own mental models and personal experiences that do not necessarily apply more generally.

There are over 100 different types of human biases in Wikipedia's catalog of cognitive biases

What types of bias exist?

Reporting	Automation	Selection	Group Attribution	Implicit
Frequency of events, properties, and/or outcomes in a data set does not accurately reflect their real-world frequency.	Tendency to favor results generated by automated systems over those generated by non-automated systems.	A data set's examples are chosen in a way that is not reflective of their real-world distribution.	Tendency to generalize what is true of individuals to an entire group to which they belong.	Assumptions are made based on one's own mental models and personal experiences that do not necessarily apply more generally.

There are over 100 different types of human biases in Wikipedia's catalog of cognitive biases

AI Fairness

Decisions made by computers after a machine-learning process may be considered unfair if they were based on variables considered sensitive

Definition from Wikipedia

Why do you need Fairness?

As the impact of AI increases across sectors and societies...

Why do you need Fairness?

As the impact of AI increases across sectors and societies...

Opportunity

To be fairer and more inclusive at a broader scale.

Risk

To have a negative wide-scale impact.

Why is Fairness difficult?

Pre-existing bias

AI models learn from existing data, and an accurate model may learn or even amplify problematic pre-existing biases

Variety of scenarios

Even with the most rigorous and cross-functional training and testing, it is a challenge to build systems that will be fair across all situations.

No standard definition

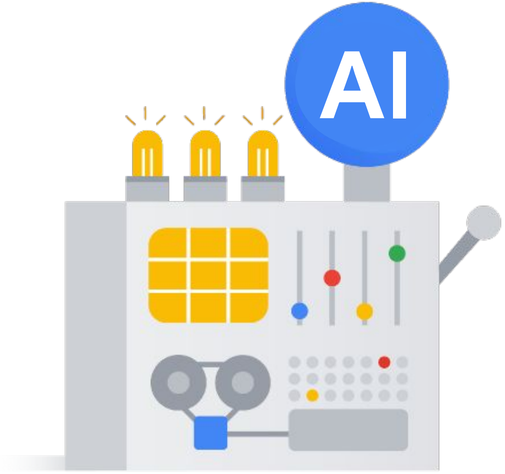
Identifying appropriate fairness criteria for a system requires multidisciplinary considerations, several of which may have tradeoffs.

Incompatibility of fairness metrics

Fairness metrics can be incompatible and impossible to satisfy simultaneously. Fairness needs to be defined contextually for the the given AI problem.

How do you address Fairness issues?

- Fostering an inclusive workflow
- Assessing training datasets for bias
- Engaging with experts to define concrete fairness goals
- Training models to remove / correct bias
- Evaluating models for disparities
- Entrusting adversarial testing to a diverse team
- Continuously testing for unfair outcomes



Topics

01	Overview of Fairness
02	Tools to Study Fairness in Datasets
03	Tools to Study Fairness in Models
04	Hands-on Lab

Tools to study fairness in datasets should allow you to easily examine:

- Missing feature values
- Unexpected feature values
- Data skews

What are good tools to study fairness in datasets?

TF Data Validation

Aequitas

What-if Tool

data-validation : a highly-scalable open-source data validation library

- Scalable calculation of summary statistics of training and test data
- Integration with a data viewer for distributions, statistics, and faceted comparison of feature pairs
- Automated data-schema generation, and schema viewer
- Anomaly detection and viewer for missing features, out-of-range values, wrong feature types, ...

https://www.tensorflow.org/tfx/data_validation/get_started

data-validation : a highly-scalable open-source data validation library

- **Scalable calculation of summary statistics of training and test data**
- Integration with a data viewer for distributions, statistics, and faceted comparison of feature pairs
- Automated data-schema generation, and schema viewer
- Anomaly detection and viewer for missing features, out-of-range values, wrong feature types, ...

```
stats = tfdv.generate_statistics_from_tfrecord(
        data_location=path,
)
```

https://www.tensorflow.org/tfx/data_validation/get_started

data-validation : a highly-scalable open-source data validation library

- **Scalable calculation of summary statistics of training and test data**
- Integration with a data viewer for distributions, statistics, and faceted comparison of feature pairs
- Automated data-schema generation, and schema viewer
- Anomaly detection and viewer for missing features, out-of-range values, wrong feature types, ...

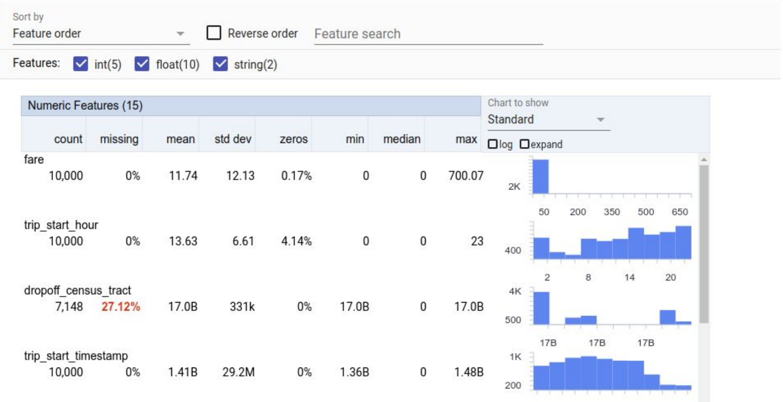
```
# Slice on country feature
# (i.e., every unique value of the feature)
slice_fn1 = slicing_util.get_feature_value_slicer(
        features={'country': None}
)

# Slice on the cross of country and state feature
# (i.e., every unique pair of values of the cross)
slice_fn2 = slicing_util.get_feature_value_slicer(
        features={'country': None, 'state': None}
)

# Slice on specific values of a feature
slice_fn3 = slicing_util.get_feature_value_slicer(
        features={'age': [10, 50, 70]}
)

stats_options = tfdv.StatsOptions(
    slice_functions=[slice_fn1, slice_fn2, slice_fn3]
)
```

https://www.tensorflow.org/tfx/data_validation/get_started

data-validation : a highly-scalable open-source data validation library

- Scalable calculation of summary statistics of training and test data
- **Integration with a data viewer for distributions, statistics, and faceted comparison of feature pairs**
- Automated data-schema generation, and schema viewer
- Anomaly detection and viewer for missing features, out-of-range values, wrong feature types, ...

```
tfdv.visualize_statistics(stats)
```

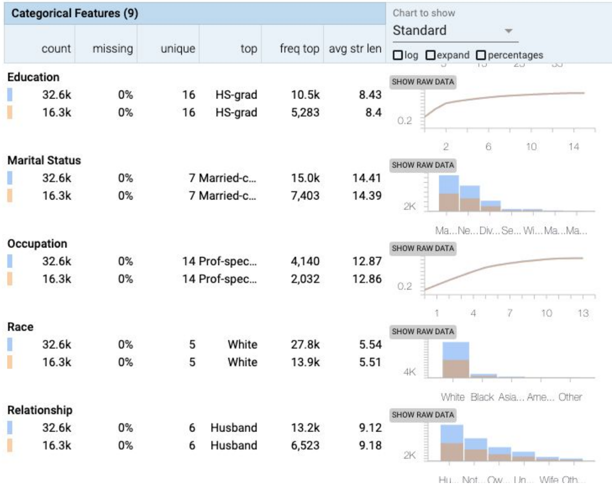


https://www.tensorflow.org/tfx/data_validation/get_started

data-validation : a highly-scalable open-source data validation library

- Scalable calculation of summary statistics of training and test data
- **Integration with a data viewer for distributions, statistics, and faceted comparison of feature pairs**
- Automated data-schema generation, and schema viewer
- Anomaly detection and viewer for missing features, out-of-range values, wrong feature types, ...

```
tfdv.visualize_statistics(stats)
```



https://www.tensorflow.org/tfx/data_validation/get_started

data-validation : a highly-scalable open-source data validation library

- Scalable calculation of summary statistics of training and test data
- Integration with a data viewer for distributions, statistics, and faceted comparison of feature pairs
- **Automated data-schema generation, and schema viewer**
- Anomaly detection and viewer for missing features, out-of-range values, wrong feature types, ...

```
schema = tfdv.infer_schema(stats)
```

```
feature {
  name: "payment_type"
  value_count {
    min: 1
    max: 1
  }
  type: BYTES
  domain: "payment_type"
  presence {
    min_fraction: 1.0
    min_count: 1
  }
}
```

https://www.tensorflow.org/tfx/data_validation/get_started

data-validation : a highly-scalable open-source data validation library

- Scalable calculation of summary statistics of training and test data
- Integration with a data viewer for distributions, statistics, and faceted comparison of feature pairs
- Automated data-schema generation, and schema viewer
- **Anomaly detection and viewer for missing features, out-of-range values, wrong feature types, ...**

```
anomalies = tfdv.validate_statistics(
        statistics=other_stats, schema=schema,
)
```

```
payment_type  Unexpected string values
Examples contain values missing from the
schema: Prcard (<1%).
```

```
options = tfdv.StatsOptions(schema=schema)
anomalous_stats = tfdv.validate_examples_in_csv(
       data_location=input, stats_options=options
)
```

```
tfdv.get_feature(schema,payment_type).skew_comparat
or.infinity_norm.threshold = 0.01
skew_anomalies = tfdv.validate_statistics(
        statistics=stats_1, schema=schema,
        serving_statistics=stats_2,
)
```

https://www.tensorflow.org/tfx/data_validation/get_started

Aequitas : an open-source bias and fairness audit toolkit

Web Audit Tool

Try our Audit Tool to generate a Bias Report

1. Upload Data (or use pre-loaded sample data)
2. Configure (bias metrics of interest and reference groups)
3. Generate the Bias Report

TRY IT OUT!

Python Library

Use our python code library to generate bias and fairness metrics on your data and predictions.

Python Code ›

Command Line Tool

Use our command line tool to generate a report using your own data and predictions.

http://www.datasciencepublicpolicy.org/our-work/tools-guides/aequitas/

Aequitas : an open-source bias and fairness audit toolkit

The Bias Report

Audit Date:	17 Jul 2023
Data Audited:	9769 rows
Attributes Audited:	gender
Audit Goal(s):	Equal Parity - Ensure all protected groups are have equal representation in the selected set. False Positive Rate Parity - Ensure all protected groups have the same false positive rates as the reference group). False Negative Rate Parity - Ensure all protected groups have the same false negative rates (as the reference group).
Reference Groups:	Custom group - The reference groups you selected for each attribute will be used to calculate relative disparities in this audit.
Fairness Threshold:	80%. If disparity for a group is within 80% and 125% of the value of the reference group on a group metric (e.g. False Positive Rate), this audit will pass.

http://aequitas.dssg.io/example.html

Aequitas : an open-source bias and fairness audit toolkit

Audit Results: Summary

Equal Parity - Ensure all protected groups are have equal representation in the selected set.	**Failed**	Details
False Positive Rate Parity - Ensure all protected groups have the same false positive rates as the reference group).	**Passed**	Details
False Negative Rate Parity - Ensure all protected groups have the same false negative rates (as the reference group).	**Passed**	Details

http://aequitas.dssg.io/example.html

Aequitas : an open-source bias and fairness audit toolkit

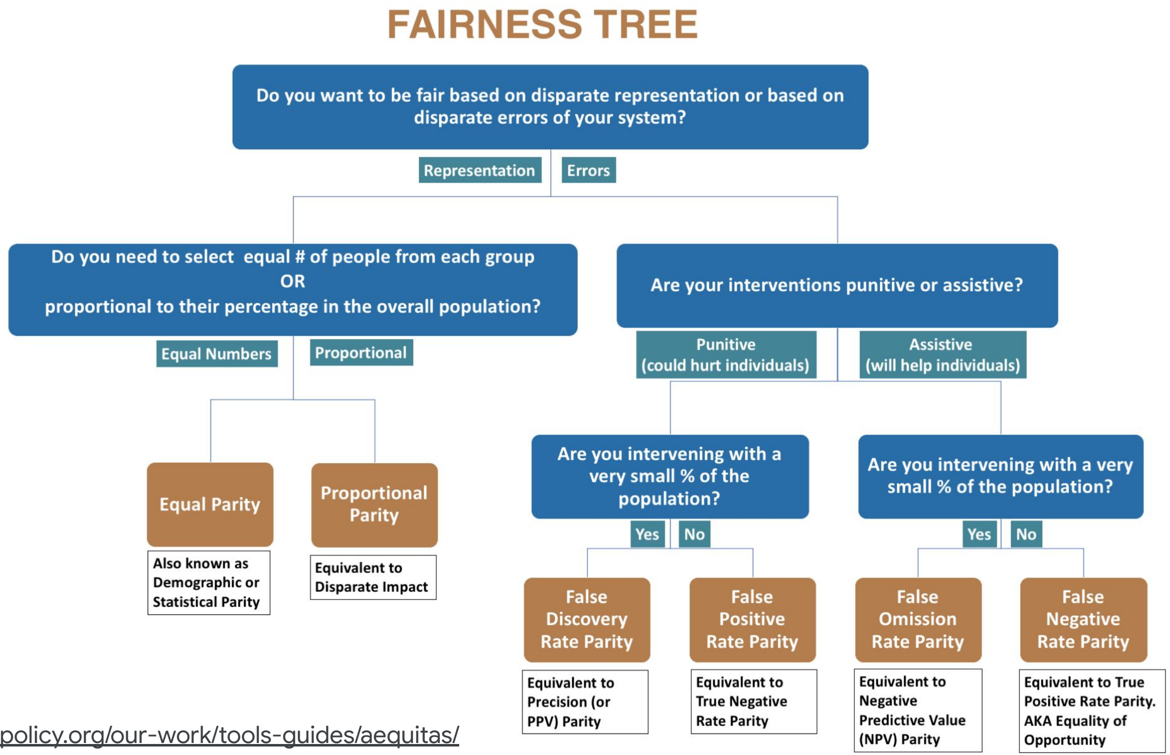
Audit Results: Group Metrics Values

gender

Attribute Value	Group Size Ratio	Predicted Positive Rate	False Positive Rate	False Negative Rate
Female	0.33	0.39	0.97	0.59
Male	0.67	0.61	0.87	0.68

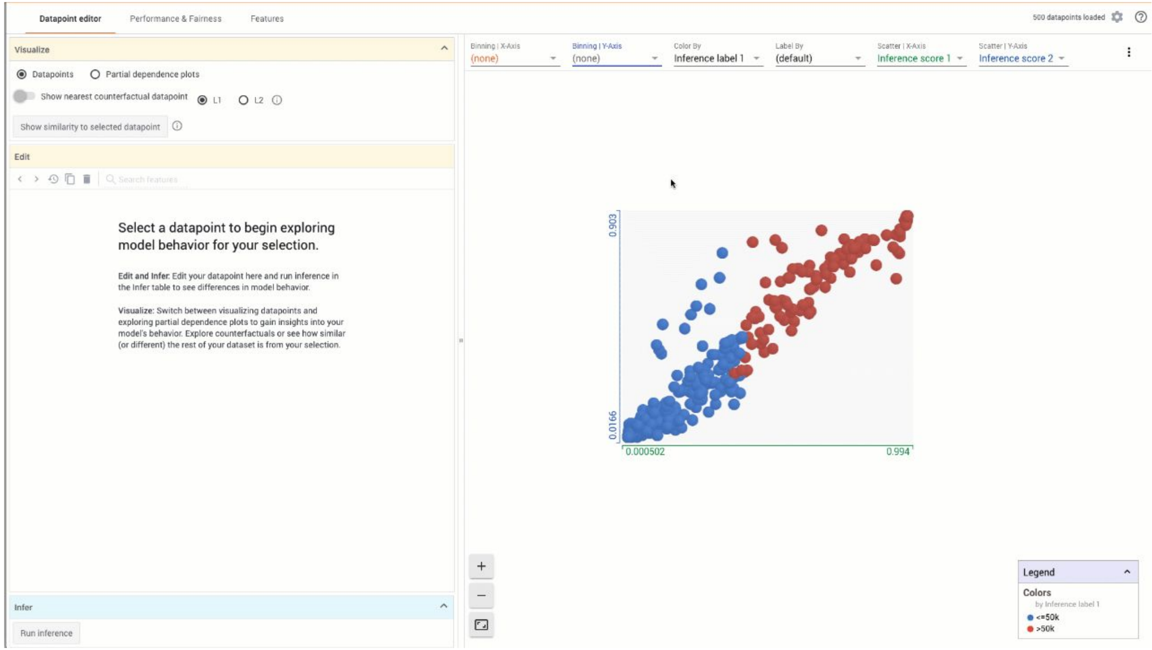
http://aequitas.dssg.io/example.html

Aequitas : an open-source bias and fairness audit toolkit



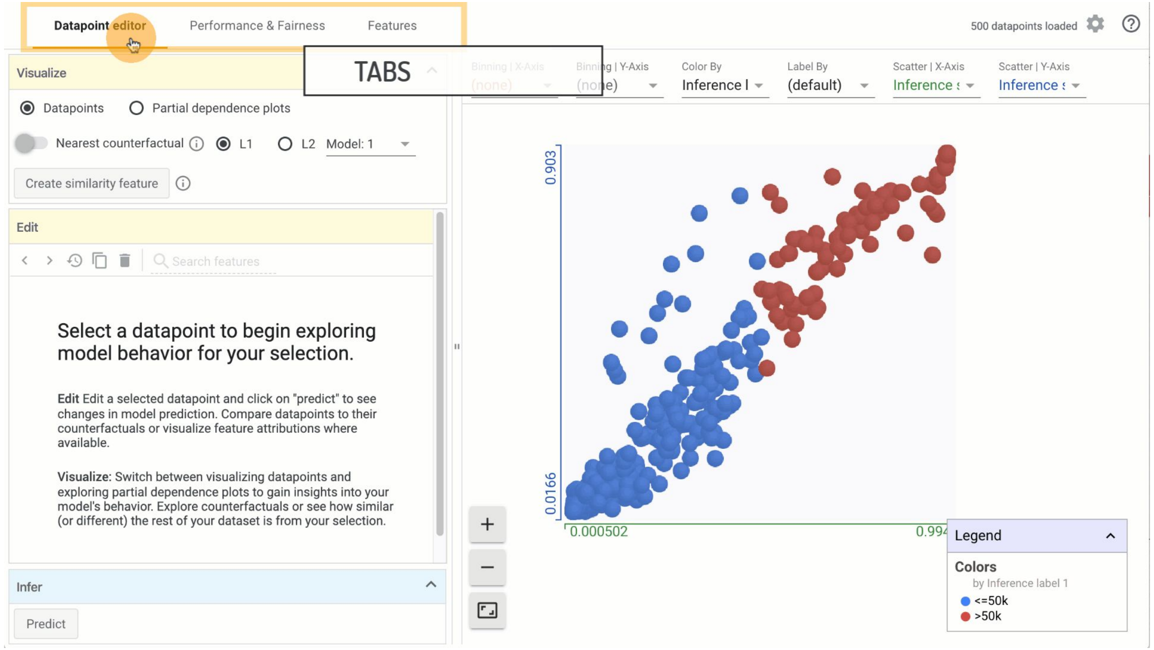
http://www.datasciencepublicpolicy.org/our-work/tools-guides/aequitas/

What-If Tool : open-source tool to visually probe ML models



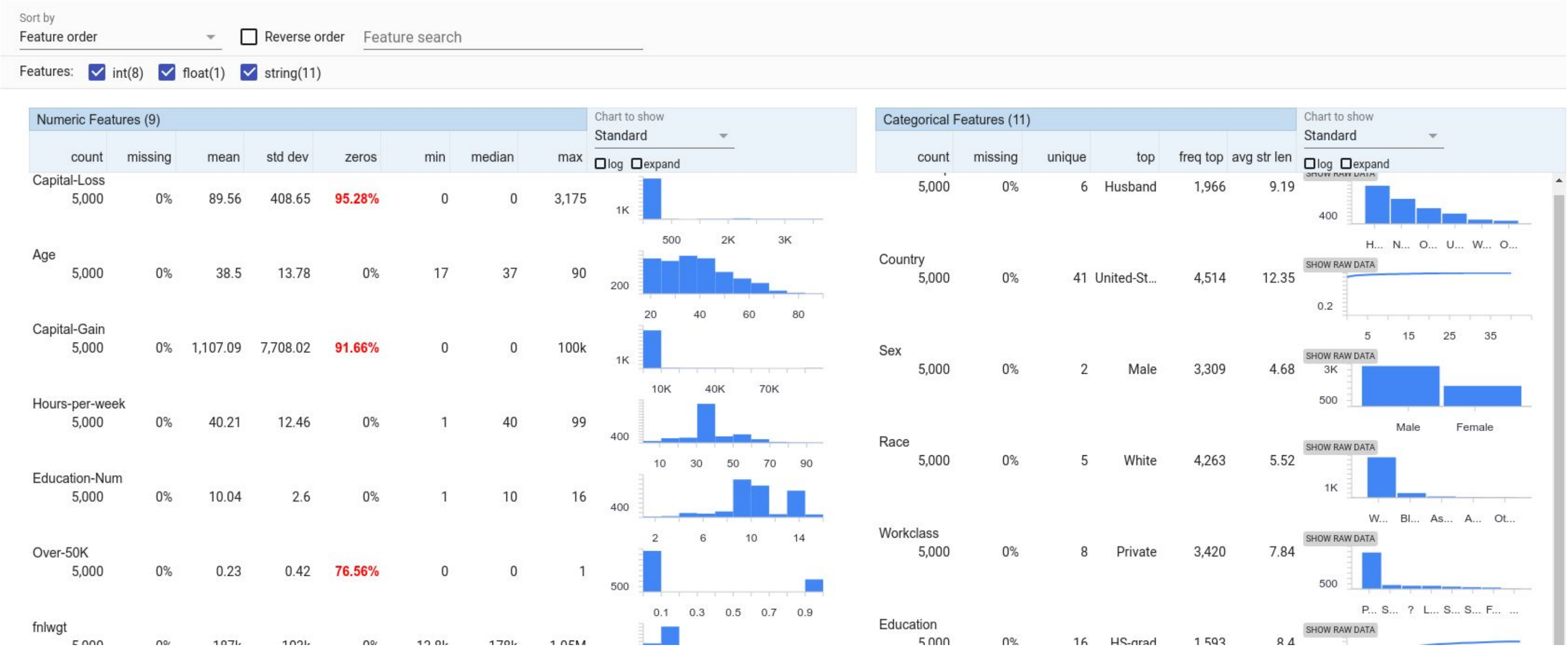
https://pair-code.github.io/what-if-tool/

What-If Tool : open-source tool to visually probe ML models



https://pair-code.github.io/what-if-tool/

What-If Tool : open-source tool to visually probe ML models



https://pair-code.github.io/what-if-tool/

Topics

What are good tools to study fairness in models?

TF Model Analysis

What-if Tool

model-analysis : a highly-scalable open-source model analysis library

TensorFlow

Keras

pandas

scikit learn

...

using pre-calculated predictions

https://www.tensorflow.org/tfx/model_analysis/get_started

model-analysis : a highly-scalable open-source model analysis library

Supported metrics are:

- Regression
- Binary classification
- Multi-class classification
- Multi-label classification
- Micro / Macro average
- Query / Ranking

https://www.tensorflow.org/tfx/model_analysis/get_started

model-analysis : a highly-scalable open-source model analysis library

- Run model analysis on a single serving model
- Validate a candidate model against a baseline
- Compare two models
- Perform fairness analysis with FairnessIndicators

https://www.tensorflow.org/tfx/model_analysis/get_started

model-analysis : a highly-scalable open-source model analysis library

- **Run model analysis on a single serving model**
- Validate a candidate model against a baseline
- Compare two models
- Perform fairness analysis with FairnessIndicators

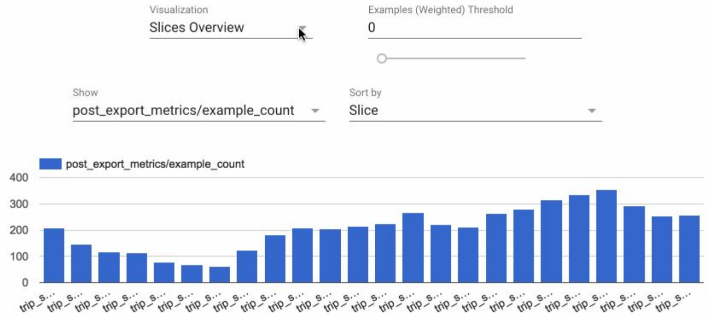
```
from google.protobuf import text_format

eval_config = text_format.Parse("""
model_specs {
  label_key: "label"
  example_weight_key: "weight"
}
metrics_specs {
  metrics { class_name: "AUC" }
  metrics { class_name: "ConfusionMatrixPlot" } # plots
}

slicing_specs {}  # overall slice
slicing_specs {feature_keys: ["age"]}
""", tfma.EvalConfig())

eval_shared_model = tfma.default_eval_shared_model(
    eval_saved_model_path=saved_model_path,
    eval_config=eval_config,
)
eval_result = tfma.run_model_analysis(
    eval_shared_model=eval_shared_model,
    eval_config=eval_config,
    data_location=data_location,
    output_path=output_path)
```

https://www.tensorflow.org/tfx/model_analysis/get_started

model-analysis : a highly-scalable open-source model analysis library

- **Run model analysis on a single serving model**
- Validate a candidate model against a baseline
- Compare two models
- Perform fairness analysis with FairnessIndicators

```
tfma.view.render_slicing_metrics(eval_result)
```



feature	accuracy	accuracy_baseline	auc	auc_precision_recall	average_loss
trip_start_hour:19	0.65672	0.59104	0.66079	0.57315	0.64654
trip_start_hour:14	0.63964	0.65766	0.63072	0.46030	0.63655
trip_start_hour:2	0.64407	0.63559	0.55829	0.46379	0.67816
trip_start_hour:12	0.70536	0.65625	0.71230	0.57907	0.57703
trip_start_hour:0	0.63768	0.66667	0.62093	0.42289	0.62715
trip_start_hour:23	0.66016	0.64844	0.58337	0.44173	0.65142

https://www.tensorflow.org/tfx/model_analysis/get_started

model-analysis : a highly-scalable open-source model analysis library

- Run model analysis on a single serving model
- **Validate a candidate model against a baseline**
- **Compare two models**
- Perform fairness analysis with FairnessIndicators

```
from google.protobuf import text_format

eval_config = text_format.Parse("""
model_specs {
  label_key: "label"
  example_weight_key: "weight"
}
metrics_specs {
  metrics {
    class_name: "AUC"
    threshold {
      value_threshold {
        lower_bound { value: 0.9 }
      }
      change_threshold {
        direction: HIGHER_IS_BETTER
        absolute { value: -1e-10 }
      }
    }
  }
  metrics { class_name: "ConfusionMatrixPlot" } # plots
}

slicing_specs {}  # overall slice
slicing_specs {feature_keys: ["age"]}
""", tfma.EvalConfig())

eval_shared_model = ...
```

https://www.tensorflow.org/tfx/model_analysis/get_started

model-analysis : a highly-scalable open-source model analysis library

- Run model analysis on a single serving model
- **Validate a candidate model against a baseline**
- **Compare two models**
- Perform fairness analysis with FairnessIndicators

```
from google.protobuf import text_format

eval_config = text_format.Parse("""
...
""", tfma.EvalConfig())

eval_shared_models = [
    tfma.default_eval_shared_model(
      model_name=tfma.CANDIDATE_KEY,
      eval_saved_model_path=saved_candidate_model_path,
      eval_config=eval_config),
    tfma.default_eval_shared_model(
      model_name=tfma.BASELINE_KEY,
      eval_saved_model_path=saved_baseline_model_path,
      eval_config=eval_config),
]
eval_result = tfma.run_model_analysis(
    eval_shared_model=eval_shared_models,
    eval_config=eval_config,
    data_location=data_location,
    output_path=output_path)
```

https://www.tensorflow.org/tfx/model_analysis/get_started

model-analysis : a highly-scalable open-source model analysis library

- Run model analysis on a single serving model
- Validate a candidate model against a baseline
- Compare two models
- **Perform fairness analysis with FairnessIndicators**

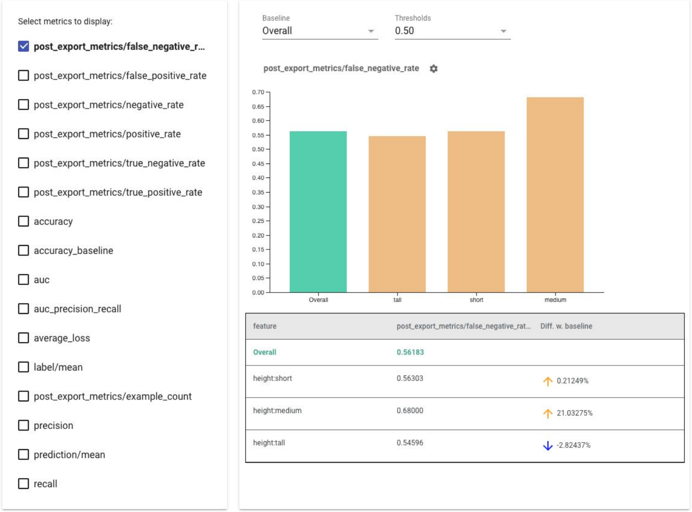
```
from
tensorflow_model_analysis.addons.fairness.post_export_met
rics import fairness_indicators

eval_config = text_format.Parse("""
model_specs {
  label_key: "label"
}
metrics_specs {
  metrics { class_name: "AUC" }
  metrics {
     class_name: "FairnessIndicators"
     config: '{ "thresholds": [0.5, 0.9] }'
  }
  metrics { class_name: "ConfusionMatrixPlot" } # plots
}

slicing_specs {}  # overall slice
slicing_specs {feature_keys: ["age"]}
""", tfma.EvalConfig())

# Let's see how to apply this to a Pandas df
eval_result = tfma.analyze_raw_data(
  data=df,
  eval_config=eval_config,
  output_path=_DATA_ROOT,
)
```

https://www.tensorflow.org/tfx/model_analysis/get_started

model-analysis : a highly-scalable open-source model analysis library

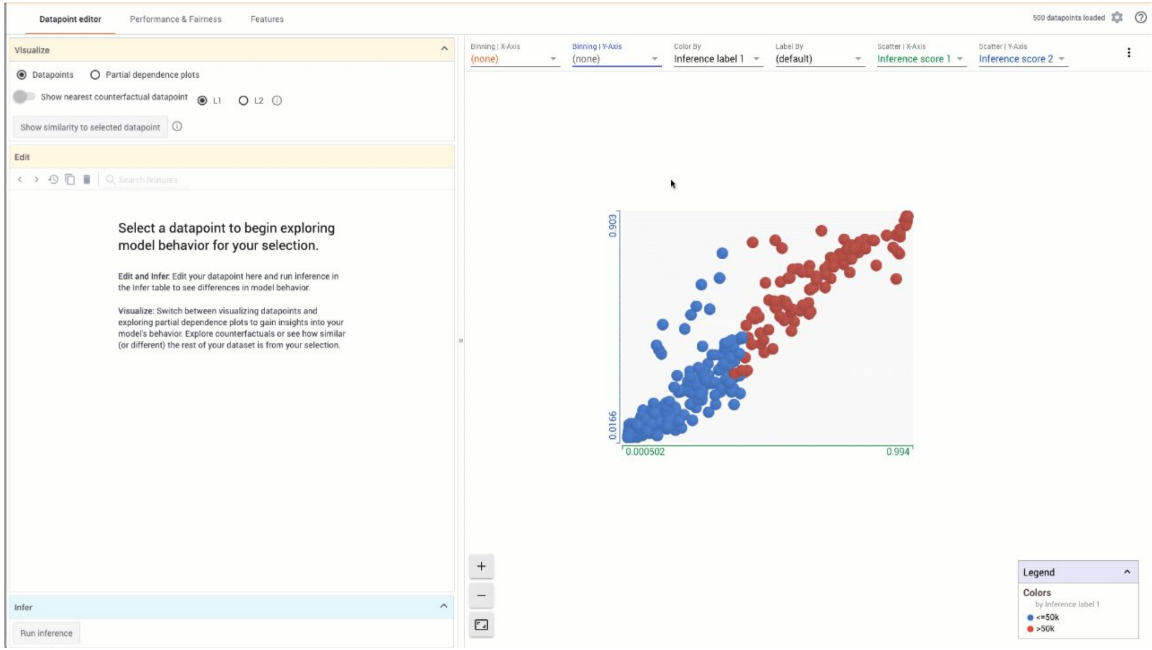
- Run model analysis on a single serving model
- Validate a candidate model against a baseline
- Compare two models
- **Perform fairness analysis with FairnessIndicators**

```
from tensorflow_model_analysis.addons.fairness.view import widget_view

tfma.view.render_slicing_metrics(eval_result)
```



https://www.tensorflow.org/tfx/model_analysis/get_started

What-If Tool : open-source tool to visually probe ML datasets and models



https://pair-code.github.io/what-if-tool/

What-If Tool : open-source tool to visually probe ML models

- Visualize inference results
- Edit a datapoint and see how your model performs
- Explore the effects of single features
- Arrange examples by similarity

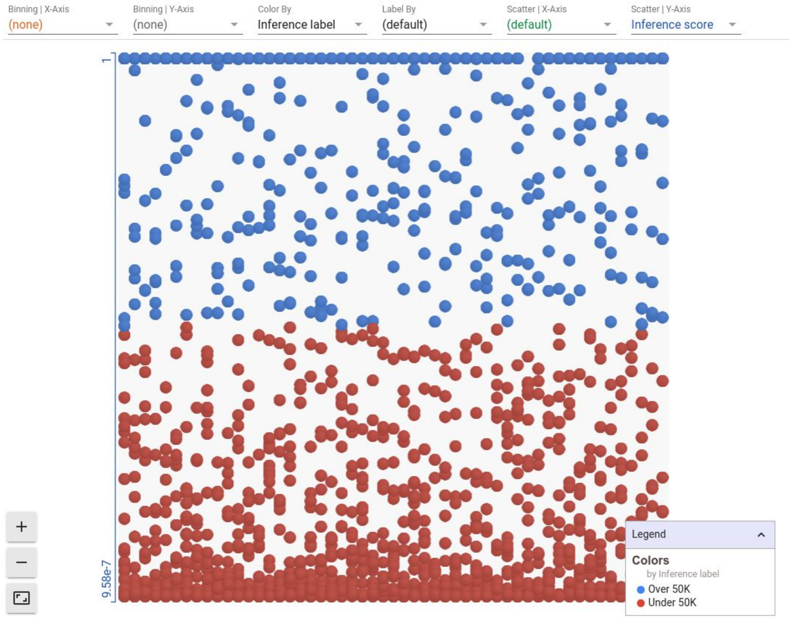
Datapoint Editor

- View confusion matrices and other metrics
- Test algorithmic fairness constraints

Performance and Fairness

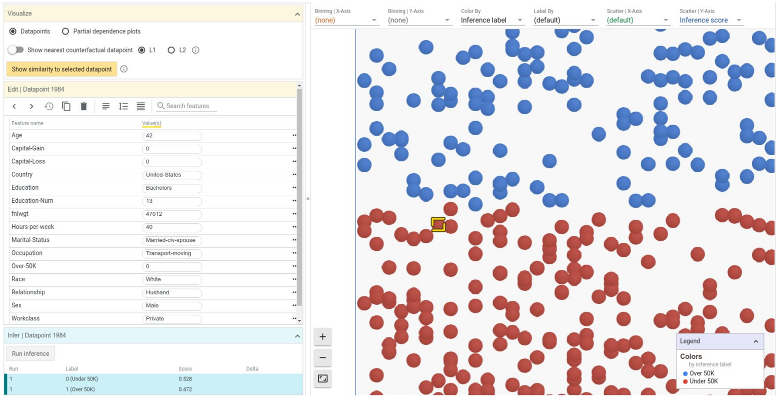
What-If Tool : open-source tool to visually probe ML models

- **Visualize inference results**
- Edit a datapoint and see how your model performs
- Explore the effects of single features
- Arrange examples by similarity
- View confusion matrices and other metrics
- Test algorithmic fairness constraints



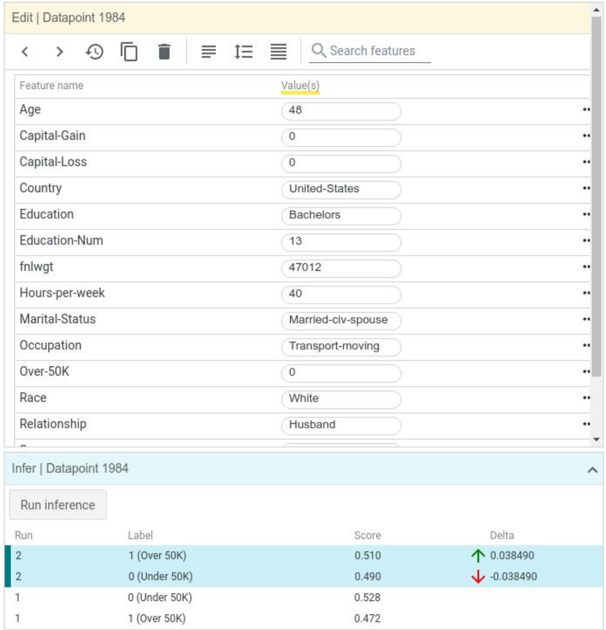
What-If Tool : open-source tool to visually probe ML models

- Visualize inference results
- **Edit a datapoint and see how your model performs**
- Explore the effects of single features
- Arrange examples by similarity
- View confusion matrices and other metrics
- Test algorithmic fairness constraints



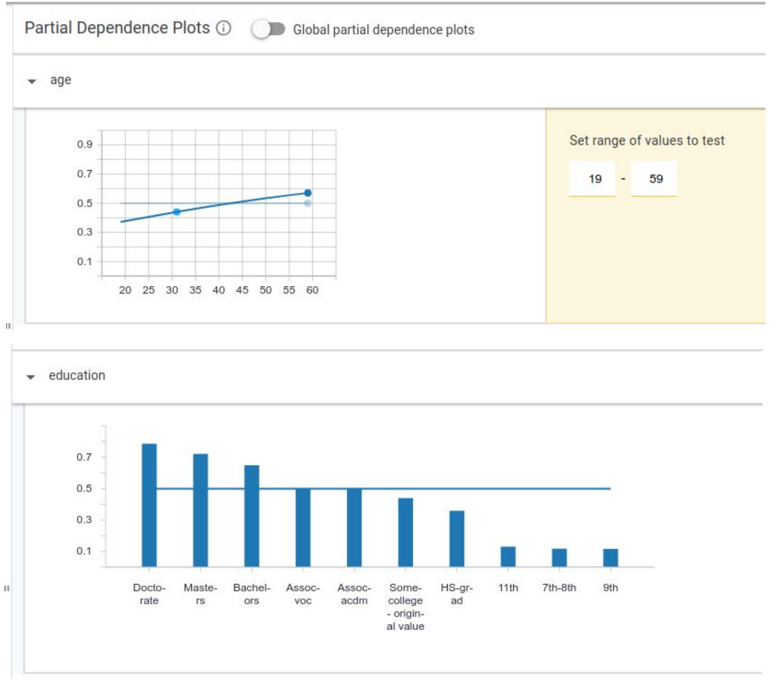
What-If Tool : open-source tool to visually probe ML models

- Visualize inference results
- **Edit a datapoint and see how your model performs**
- Explore the effects of single features
- Arrange examples by similarity
- View confusion matrices and other metrics
- Test algorithmic fairness constraints



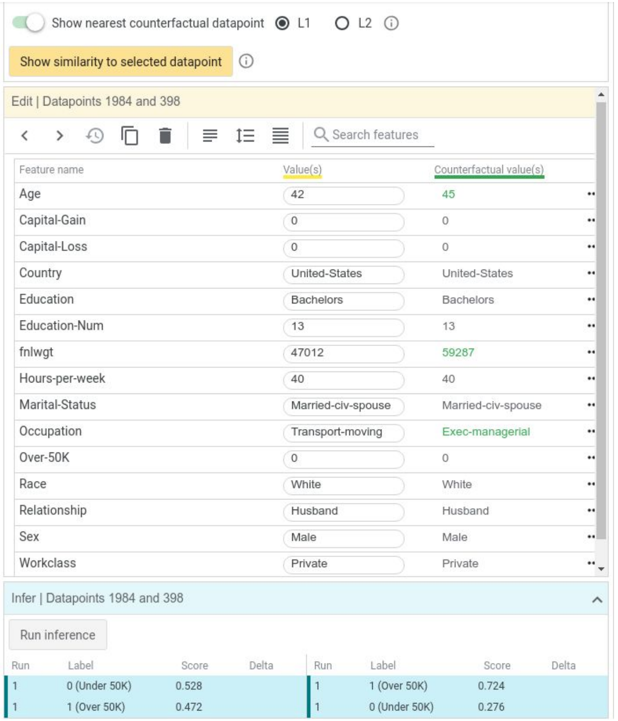
What-If Tool : open-source tool to visually probe ML models

- Visualize inference results
- Edit a datapoint and see how your model performs
- **Explore the effects of single features**
- Arrange examples by similarity
- View confusion matrices and other metrics
- Test algorithmic fairness constraints



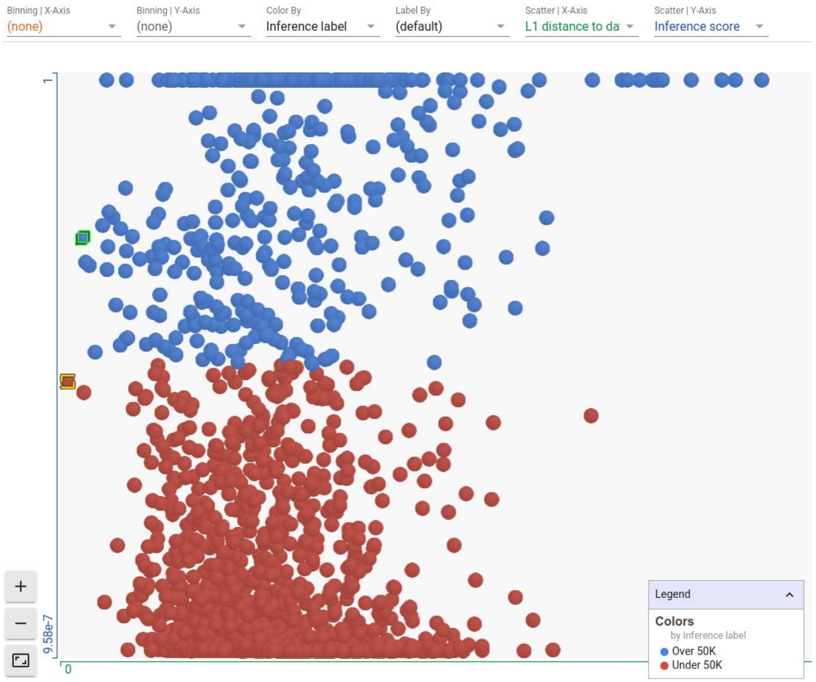
What-If Tool : open-source tool to visually probe ML models

- Visualize inference results
- Edit a datapoint and see how your model performs
- Explore the effects of single features
- **Arrange examples by similarity**
- View confusion matrices and other metrics
- Test algorithmic fairness constraints



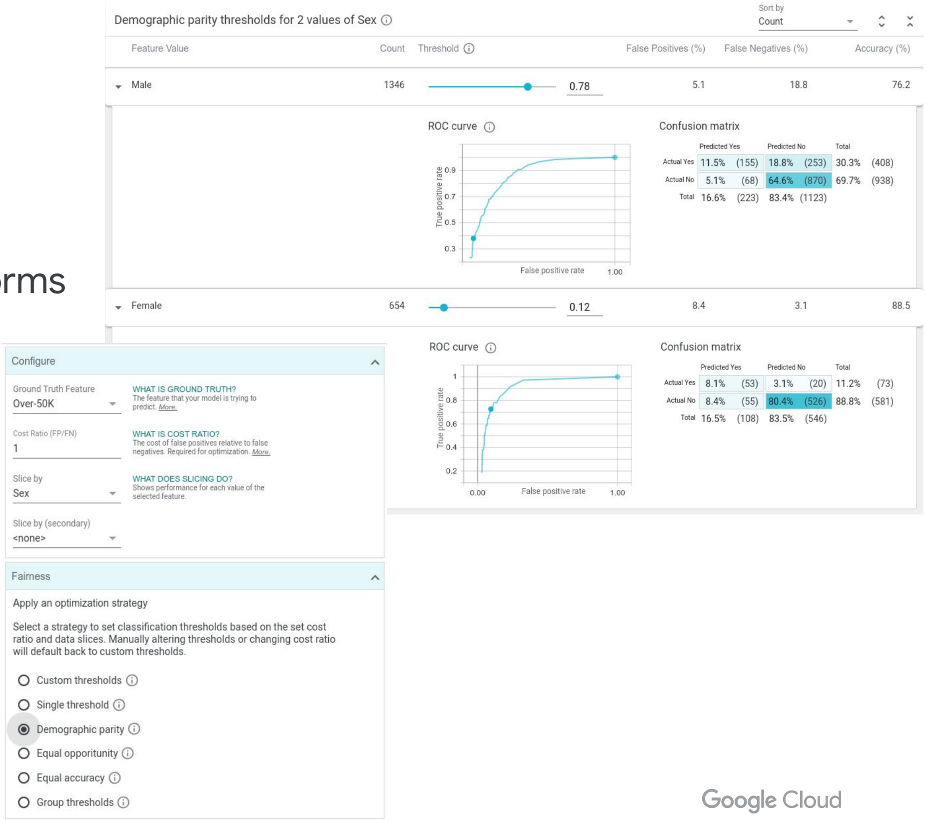
What-If Tool : open-source tool to visually probe ML models

- Visualize inference results
- Edit a datapoint and see how your model performs
- Explore the effects of single features
- **Arrange examples by similarity**
- View confusion matrices and other metrics
- Test algorithmic fairness constraints



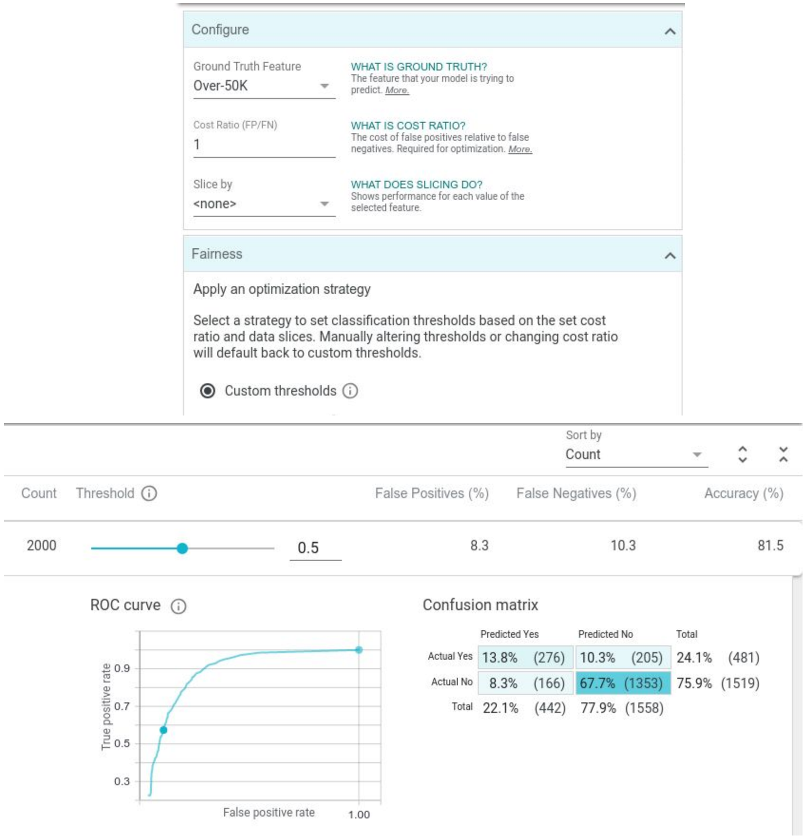
What-If Tool : open-source tool to visually probe ML models

- Visualize inference results
- Edit a datapoint and see how your model performs
- Explore the effects of single features
- Arrange examples by similarity
- View confusion matrices and other metrics
- **Test algorithmic fairness constraints**



What-If Tool : open-source tool to visually probe ML models

- Visualize inference results
- Edit a datapoint and see how your model performs
- Explore the effects of single features
- Arrange examples by similarity
- **View confusion matrices and other metrics**
- **Test algorithmic fairness constraints**



Topics

01	Overview of Fairness
02	Tools to Study Fairness in Datasets
03	Tools to Study Fairness in Models
04	Hands-on Lab

Lab:

Using TensorFlow Data Validation and TensorFlow Model Analysis to Ensure Fairness

Appendix

What are good tools to study fairness in models?

model-analysis | Fairlearn | What-If Tool

Fairlearn : an open-source Python toolkit compatible with scikit-learn

Metrics

e.g. *MetricFrame.overall*

MetricFrame.selection_rate

MetricFrame.by_group.plot.bar

Algorithms

e.g. *CorrelationRemover*

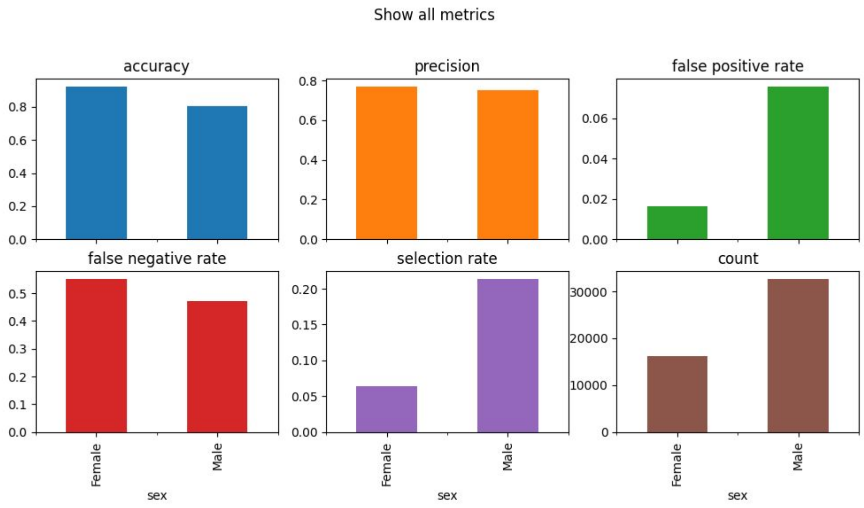
AdversarialFairnessClassifier

AdversarialFairnessRegressor

https://fairlearn.org/v0.8/quickstart.html

Fairlearn : use fairness metrics for assessment

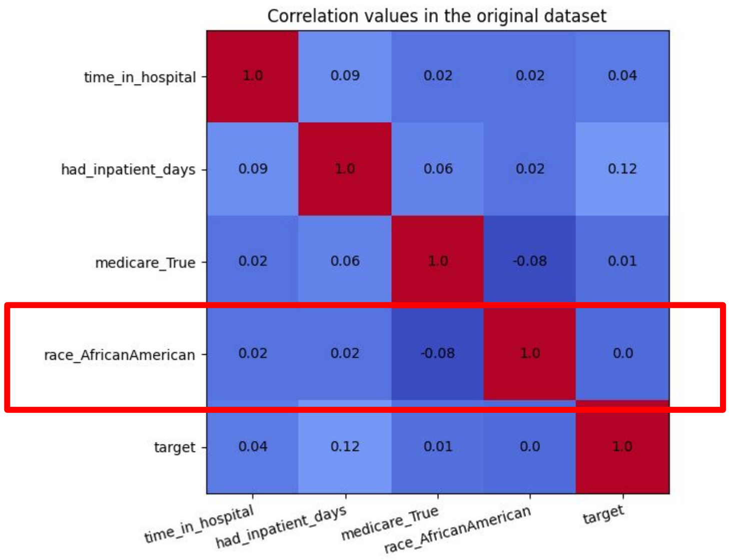
```
from fairlearn.metrics import MetricFrame
from sklearn.metrics import *

data = fetch_openml(data_id=1590, as_frame=True)
X, y_true = ...preprocess data...
classifier = ...train...
y_pred = classifier.predict(X)

metrics = {
    "accuracy": accuracy_score,
    "precision": precision_score,
    "false positive rate": false_positive_rate,
    "false negative rate": false_negative_rate,
    "selection rate": selection_rate,
    "count": count,
}
metric_frame = MetricFrame(
        metrics, y_true, y_pred, sensitive_features=gender,
)
metric_frame.by_group.plot.bar(
    subplots=True,
    layout=[3, 3],
    legend=False,
    title="Show all metrics",
)
```



https://fairlearn.org/v0.8/quickstart.html

Fairlearn : use algorithms for mitigation

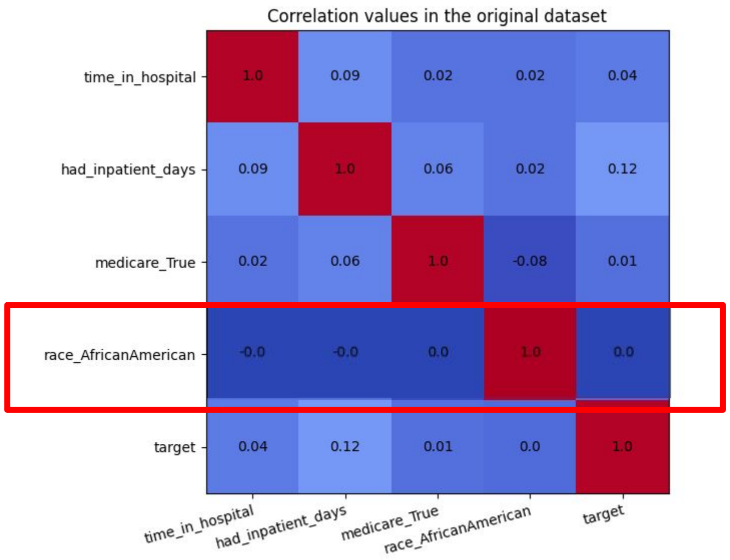
```
from fairlearn.preprocessing import CorrelationRemover
import pandas as pd
from sklearn.datasets import fetch_openml
data = fetch_openml(data_id=43874, as_frame=True)
X = data.data[["race", "time_in_hospital", "had_inpatient_days",
"medicare"]]
X = pd.get_dummies(X)
X = X.drop(["race_Asian",
...                     "race_Caucasian",
...                     "race_Hispanic",
...                     "race_Other",
...                     "race_Unknown",
...                     "had_inpatient_days_False",
...                     "medicare_False"], axis=1)
cr = CorrelationRemover(
        sensitive_feature_ids=['race_AfricanAmerican']
)
cr.fit(X)
X_transform = cr.transform(X)
```



https://fairlearn.org/v0.8/user_guide/mitigation.html

Fairlearn : use algorithms for mitigation

```
from fairlearn.preprocessing import CorrelationRemover
import pandas as pd
from sklearn.datasets import fetch_openml
data = fetch_openml(data_id=43874, as_frame=True)
X = data.data[["race", "time_in_hospital", "had_inpatient_days",
"medicare"]]
X = pd.get_dummies(X)
X = X.drop(["race_Asian",
...                     "race_Caucasian",
...                     "race_Hispanic",
...                     "race_Other",
...                     "race_Unknown",
...                     "had_inpatient_days_False",
...                     "medicare_False"], axis=1)
cr =
CorrelationRemover(sensitive_feature_ids=['race_AfricanAmerican'])
cr.fit(X)
CorrelationRemover(sensitive_feature_ids=['race_AfricanAmerican'])
X_transform = cr.transform(X)
```



https://fairlearn.org/v0.8/user_guide/mitigation.html