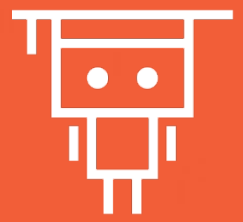


MACHINE LEARNING
UNIVERSITY

Responsible AI

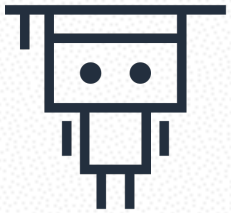
Fairness & Bias in ML – DAY 2

Learning Outcomes

- Fundamental understanding of Machine Learning:
  - Concepts & terminology
- Practical ML skills and techniques:
  - Train, tune, test and evaluate simple ML models
  - Check data and ML model for bias
- How to identify and mitigate bias and fairness issues in ML

Course Schedule

Day One	Day Two	Day Three
Fundamentals of Machine Learning	Data Processing	Bias Mitigation during Model Training
Introduction to Fairness & Bias Mitigation in ML	ML Algorithm Selection, Model Build & Evaluation	Bias Mitigation during Post-Processing
Model Formulation & Data Collection	Fairness Criteria	Bias Mitigation for Models in Production
Exploratory Data Analysis	Bias Mitigation during Pre-Processing	Explainability

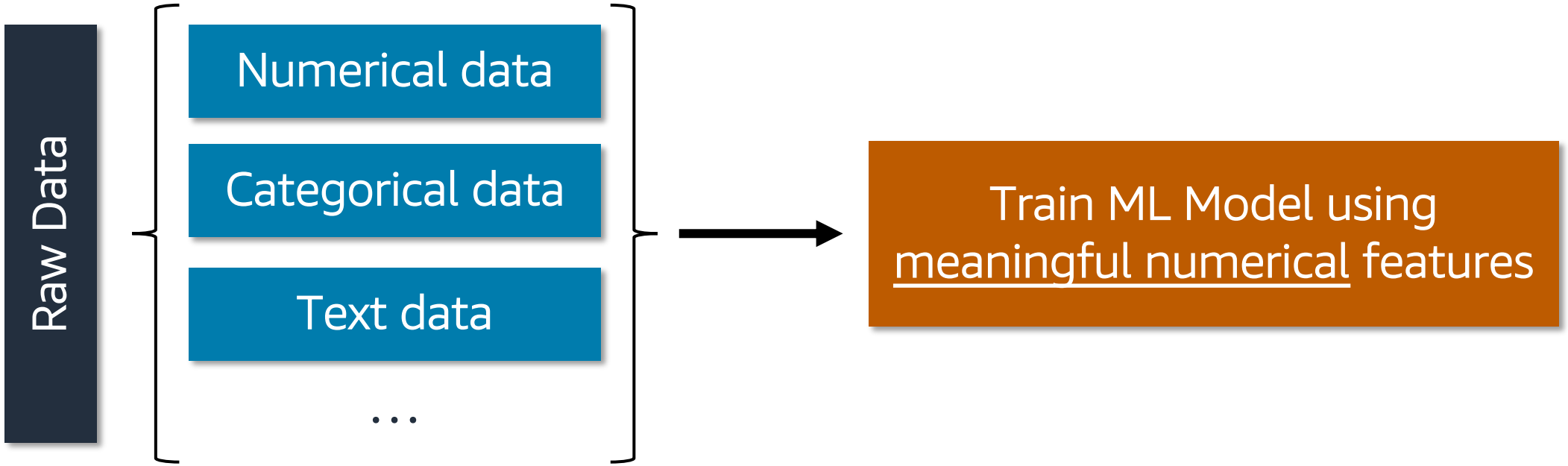


MACHINE LEARNING
UNIVERSITY

Data Processing

Data Processing

Using domain and data knowledge to prepare features as inputs for ML models from the raw data provided.



Data Processing

- There is no single way to pre-process data; it will depend on the ML problem, the data itself and the intended algorithm.

- Example data processing techniques:
  - » Encoding data
  - » Standardizing data
  - » Filling missing values
  - » Up-sampling data
  - » Redacting/coarsening data
  - » Renaming features
  - » Combining features
  - » …

Redacting Data: Preserving Privacy

- Combination of features can enable re-identification of individuals; to mitigate redact or coarsen (aggregate) information
  - » Combination ZIP, sex and birthday can uniquely identify approx. 87% of US population
  - » K-anonymity: determines level of protection (re-identification granularity)

Name	Age	Sex	ZIP	Diagnosis
Tom	61	M	1001	Cancer
Peter	67	M	1500	Heart Disease
Lisa	50	F	1301	Heart Disease
Julie	45	F	1906	HIV

Fictional Healthcare Dataset (not anonymized)

Redacting Data: Preserving Privacy

- Combination of features can enable re-identification of individuals; to mitigate redact or coarsen (aggregate) information
  - Combination ZIP, sex and birthday can uniquely identify approx. 87% of US population
  - K-anonymity: determines level of protection (re-identification granularity)

Name	Age	Sex	ZIP	Diagnosis
*	60-70	M	1***	Cancer
*	60-70	M	1***	Heart Disease
*	40-50	F	1***	Heart Disease
*	40-50	F	1***	HIV

Fictional Healthcare Dataset (2-way anonymized)

Encoding Categorical Features

Categorical (also called discrete) features

Example: isEmployed ∈ {false, true}, ageGroup ∈ {20-40, 40-60}

» Most ML models require converting categorical features to numerical ones.

Encode/define a mapping: Assign a number to each category.

- **Nominals:** Categories are unordered, e.g., color ∈ {green, red, blue}. No natural numerical representation for classes.
- **Ordinals:** Categories are ordered, e.g., size ∈ {L > M > S}. We can assign L->3, M->2, S->1.

Encoding Categorical Features

OneHotEncoder: sklearn converts categorical features into new "dummy"/indicator features – `.fit(), .transform()`

» Does not automatically name the new features.

	color	size	price	classlabel
0	green	S	10.1	shirt
1	red	M	13.5	pants
2	blue	L	15.3	shirt

```
from sklearn.preprocessing import OneHotEncoder

ohe = OneHotEncoder(sparse=False)
pd.DataFrame(ohe.fit_transform(df[['color']]))
```

	0	1	2
0	0.0	1.0	0.0
1	0.0	0.0	1.0
2	1.0	0.0	0.0

Let's encode one (or more) categorical fields.

Encoding Categorical Features

OrdinalEncoder: sklearn encoder, encodes categorical features as an integer array – `.fit(), .transform()`

» Encodes categorical features (alphabetic default) to integers (0 to n_categories - 1).

```
from sklearn.preprocessing import OrdinalEncoder

oe = OrdinalEncoder([['S','M','L']])
df[['size']] = oe.fit_transform(df[['size']])
```

	color	size	price	classlabel
0	green	S	10.1	shirt
1	red	M	13.5	pants
2	blue	L	15.3	shirt

Let's encode one (or more) categorical fields.

	color	size	price	classlabel
0	green	0.0	10.1	shirt
1	red	1.0	13.5	pants
2	blue	2.0	15.3	shirt

Feature Scaling

- Motivation: Many algorithms are sensitive to features being on different scales, like metric-based algorithms (KNN, K Means) and gradient descent-based algorithms (regression, neural networks)
  - » Note: tree-based algorithms (decision trees, random forests) do not have this issue

- Solution: Bring features to the same scale
  - » Mean/variance standardization
  - » MinMax scaling

Standardization in sklearn

StandardScaler: sklearn scaler, scaling values to be centered around mean 0 with standard deviation 1 - `.fit(), .transform()`

Transform: $x_{scaled} = \frac{x - x_{mean}}{x_{std}}$

```
from sklearn.preprocessing import StandardScaler
std_sc = StandardScaler()

raw_data = np.array([[-3.4],[4.5],[50],[24],[3.4],[1.6]])
scaled_data = std_sc.fit_transform(raw_data)
print(scaled_data.reshape(1,-1))
```

```
[[-0.90560498 -0.47848383  1.98151777  0.57580257 -0.53795639 -0.63527514]]
```

MinMax Scaling in sklearn

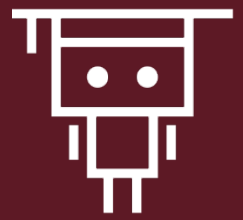
MinMaxScaler: sklearn scaler, scaling values so that minimum value is 0 and maximum value is 1 - `.fit(), .transform()`

Transform: $x_{scaled} = \frac{x - x_{min}}{x_{max} - x_{min}}$

```
from sklearn.preprocessing import MinMaxScaler
minmax_sc = MinMaxScaler()

raw_data = np.array([[-3.4],[4.5],[50],[24],[3.4],[1.6]])
scaled_data = minmax_sc.fit_transform(raw_data)
print(scaled_data.reshape(1,-1))
```

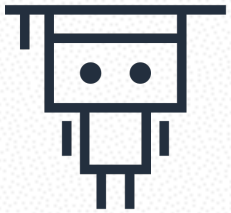
```
[[0.         0.14794007 1.         0.51310861 0.12734082 0.09363296]]
```



Notebook: MLA-RESML-DATAPREP.ipynb

MLA-RESML-DATAPREP.ipynb Notebook

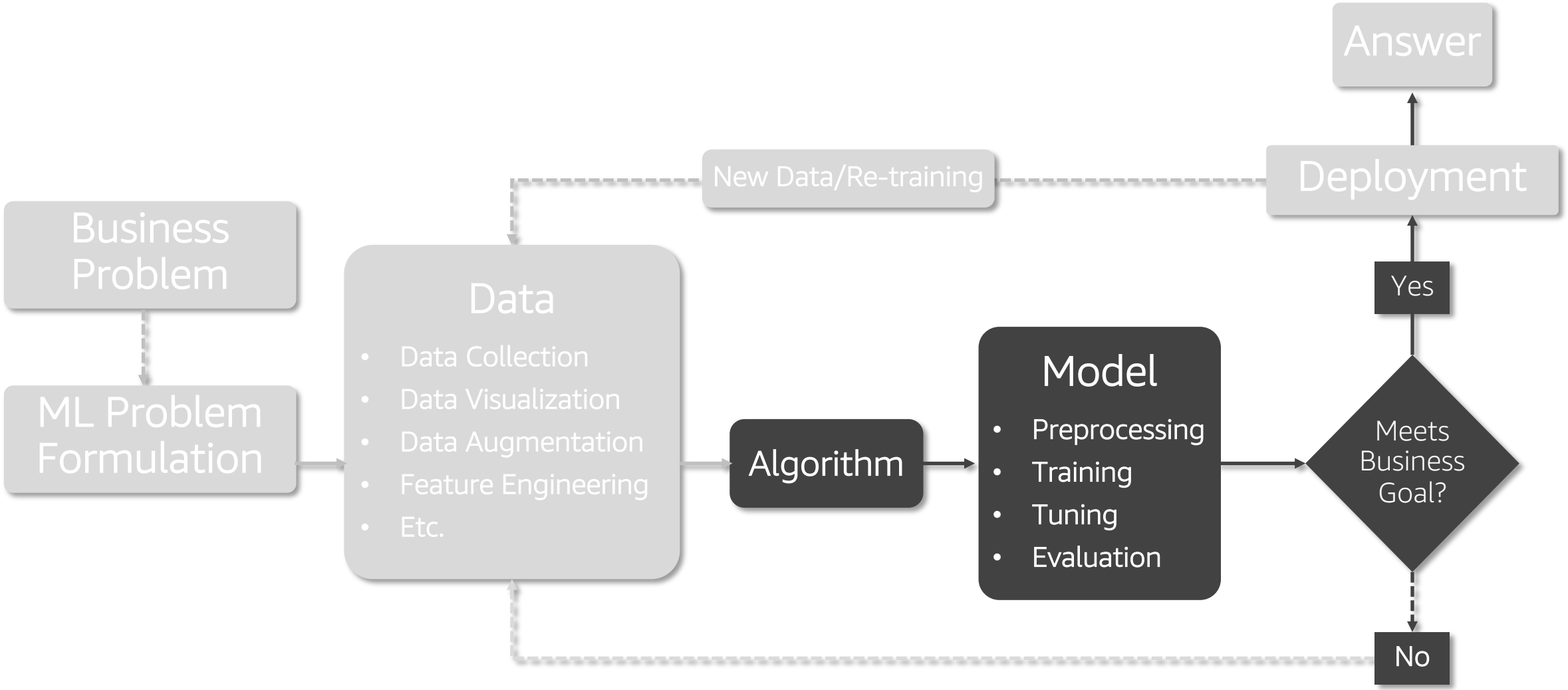
- This notebook shows how to prepare data for a ML model (transforming everything into numerical values).
- We will also have a look at missing values and feature selection.
- We will see a mix of data types (numerical, categorical data).

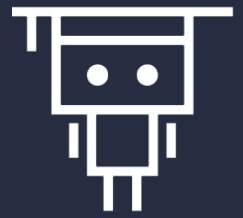


MACHINE LEARNING
UNIVERSITY

ML Algorithm Selection, Model Build & Evaluation

Bias Mitigation throughout the ML lifecycle…





Algorithm Selection

ML Algorithm

- Depending on the problem type, have a range of possible algorithms we can chose from:

	Regression (Quantity)	Classification (Category)	Clustering
ML Algorithms	• Linear Regression • K-Nearest Neighbors • Neural Nets • Decision Trees …	• Support Vector Machine • K-Nearest Neighbors • Neural Nets • Decision Trees …	• PCA • Collaborative Filtering • K-Means …

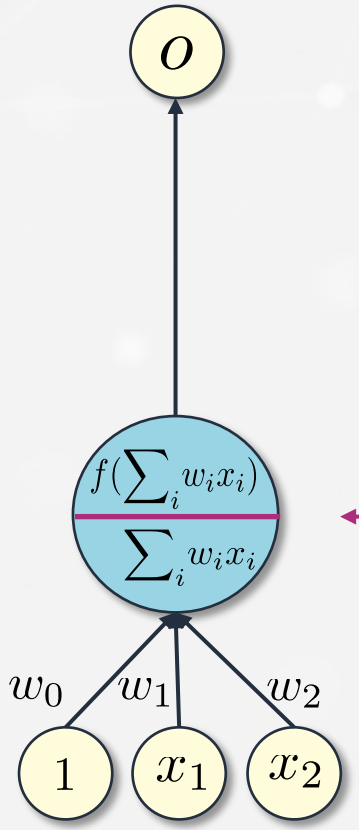
Algorithm Selection

- Different requirements can be solved with different algorithmic approaches, consider:
  - How frequently to predict?
  - How quickly predictions required (latency)?
  - How much training data available?
  - Distribution of training data?
  - Do I need to explain predictions (explainability)?
  - …

Not all models are easy to explain...

In certain settings, *accuracy-interpretability trade offs* may exist too.

Artificial Neuron



Output

Activation function

(sum)

(weights)

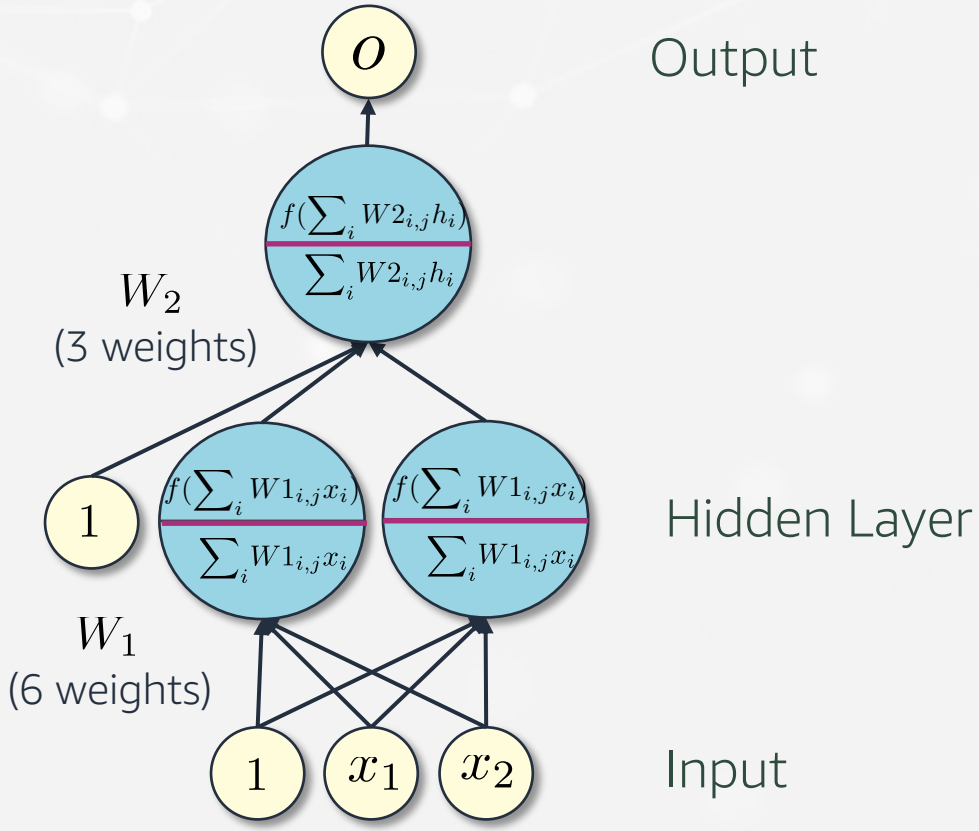
Input

Artificial Neuron: Given $\{x_i\}$ predict y, where $y \epsilon \{0,1\}$:

$$y = f(w_0 + w_1 x_1 + ... + w_q x_q)$$

where f is a nonlinear activation function (sigmoid, tanh, ReLU, ...)

Multilayer Perceptron



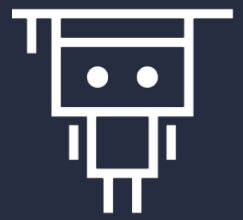
A neural network consisting of input, hidden and output layers.

Each layer is connected to the next layer.

An activation function is applied on each hidden layer (and output layer).

More details

MACHINE LEARNING UNIVERSITY



Model Training

Model Training Steps

- Every parametric model has 3 main components:
  - » ML Algorithm
  - » Loss/Cost Function: Function, C(w), that compares prediction to true value, a.k.a. “the model error”
  - » Optimization Method: Method to minimize the error
- Goal of any parametric model is to find weights (parameters) that minimize the loss on the data. Evaluate on test (holdout) set.

ML Algo: Regression Example

Label | Features

HS	Age	BMI	Smoker	Ethnicity
80	32	28	Y	White
65	21	23	N	Hispanic
…	…	…	…	…

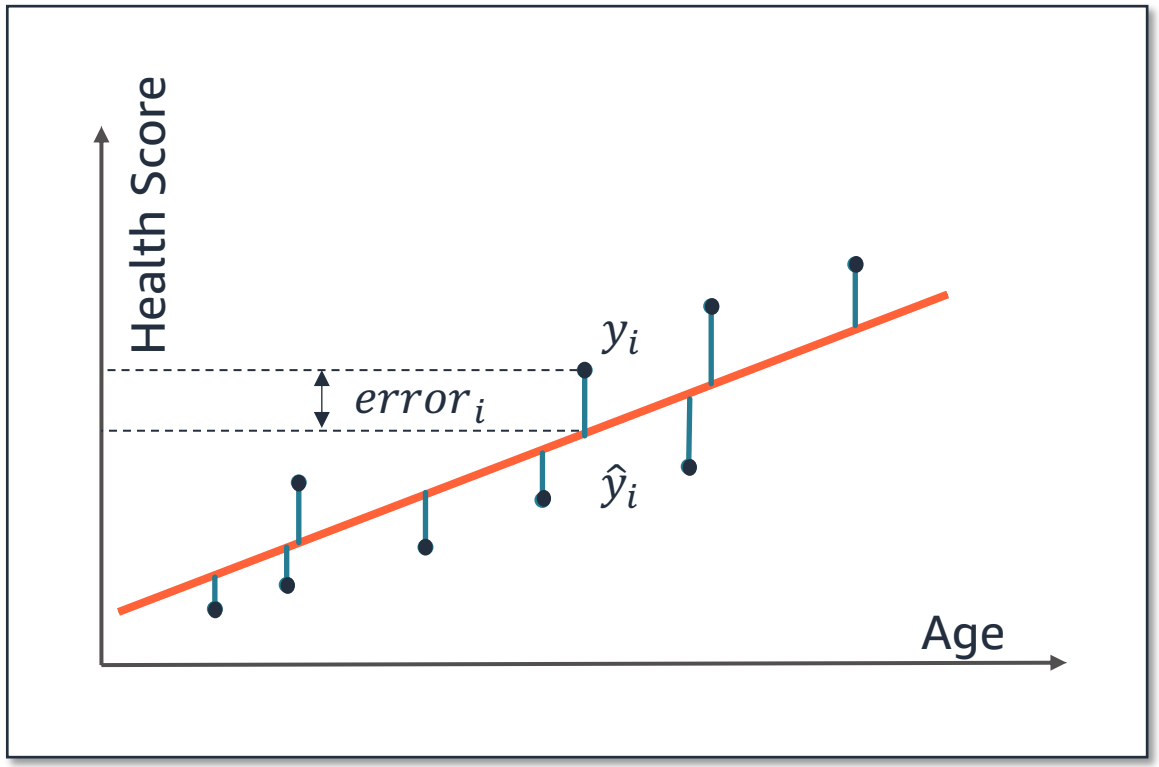


Fig.: Predicted Healthcare Spend

Regression Loss Function

Sum of Squared Errors (SSE) - a numeric value that measures the **performance of a regressor** when model output is a continuous numerical value between –infinity and +infinity:

Regression Loss Function

Sum of Squared Errors (SSE) - a numeric value that measures the **performance of a regressor** when model output is a continuous numerical value between –infinity and +infinity:

$$C(w_i) = \sum_{i}^{n} (y_i - \hat{y}_i)^2 = \sum_{i}^{n} (y_i - (w_0 x_0 + w_1 x_1 + \cdots))^2$$

where y: true value $\in \{-\infty, +\infty\}$, $\hat{y}$: predicted value, n: number of data points

To improve your regression model, minimize SSE.

ML Algo: Classification Example

Label | Features

Approved	Age	Income	Smoker	Ethnicity
+	32	28k	Y	White
-	21	23k	N	Hispanic
…	…	…	…	…

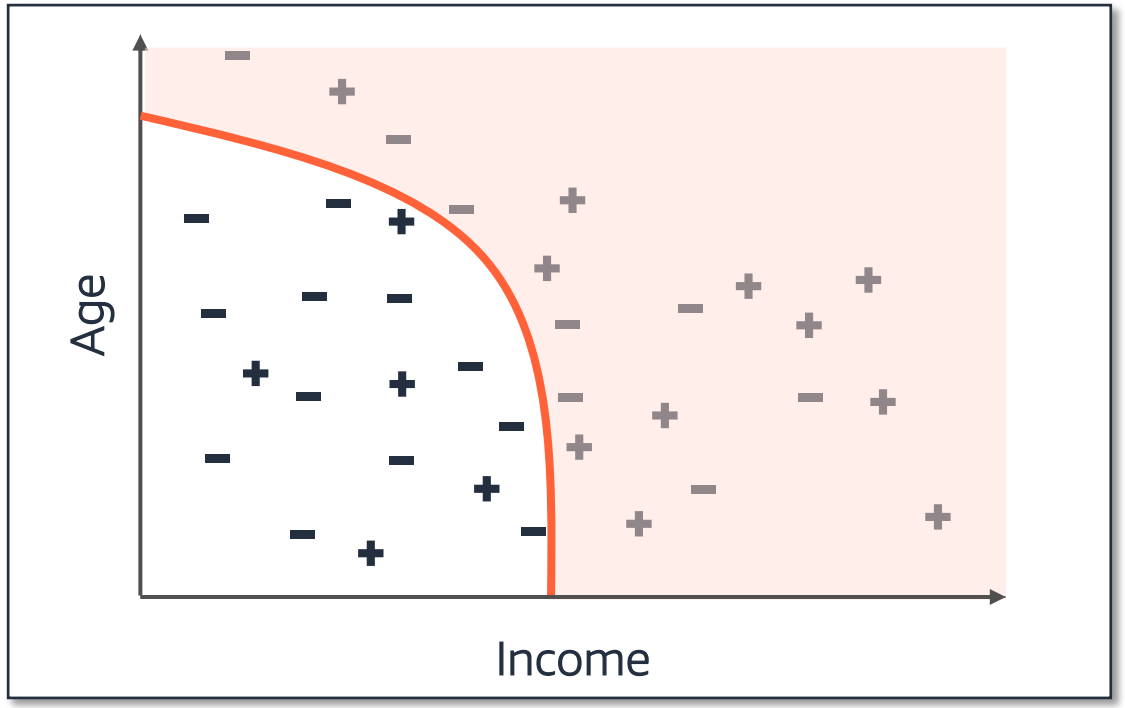


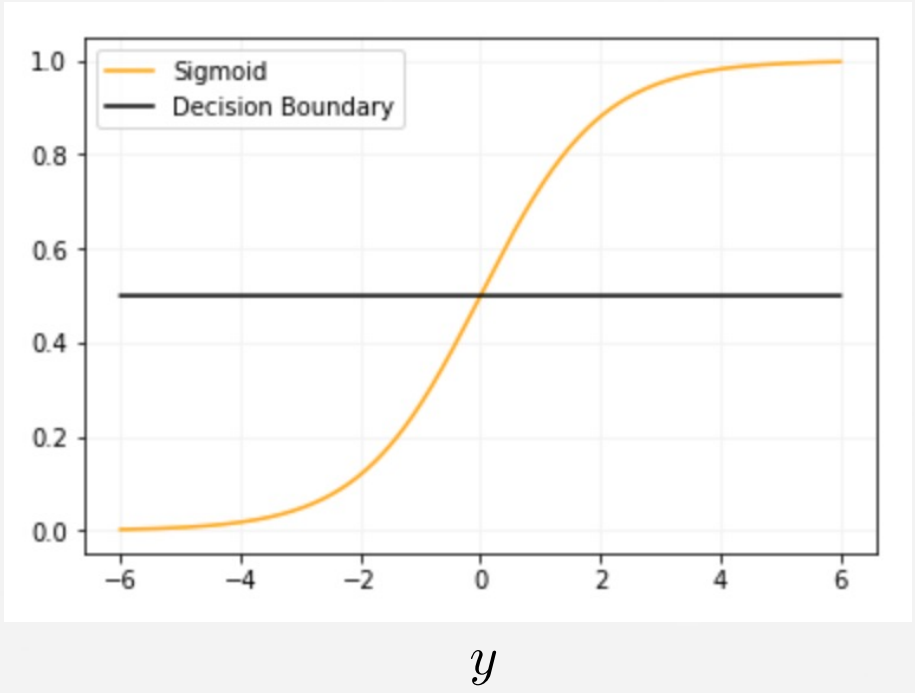
Fig.: Bank Loan Ground Truth

+ Approved
− Not Approved

MACHINE LEARNING UNIVERSITY

Logistic Regression

Idea: Apply **Sigmoid function** to linear regression to create classifier.



$$sigmoid(y) = \frac{1}{1 + e^{-y}}$$

"squishes" y values to the 0 –1 range.

Our **logistic regression** equation becomes:

$$sigmoid(w_0 + w_1 x_1 + ... + w_q x_q)$$

Use a "**Decision boundary**" at 0.5

- if $sigmoid(y) < 0.5$, round down (class 0)
- if $sigmoid(y) \geq 0.5$, round up (class 1)

Classification Loss Function

Log-Loss is a numeric value that measures the **performance of a binary classifier** when model output is a probability between 0 and 1:

Classification Loss Function

Log-Loss is a numeric value that measures the **performance of a binary classifier** when model output is a probability between 0 and 1:

$$C(w_i) = -\sum_j y_j - \log(\hat{y}_j) = -\sum_i^n y_i \, log(\hat{y}_i) + (1 - y_i) \, log(1 - \hat{y}_i)$$

where j: class ∈ {0, 1}, $\hat{y}$: probability of being class 1

$$\hat{y} = sigmoid(w_0 x_0 + w_1 x_1 + \ldots)$$

To improve Logistic Regression, minimize Log-Loss.

What is Optimization?

- Optimizing: Minimizing loss function.
  - » Regression - find the weight(s), ***w***, that result in the lowest error value.
  - » Classification - find splitting rules that minimizes impurity or error.

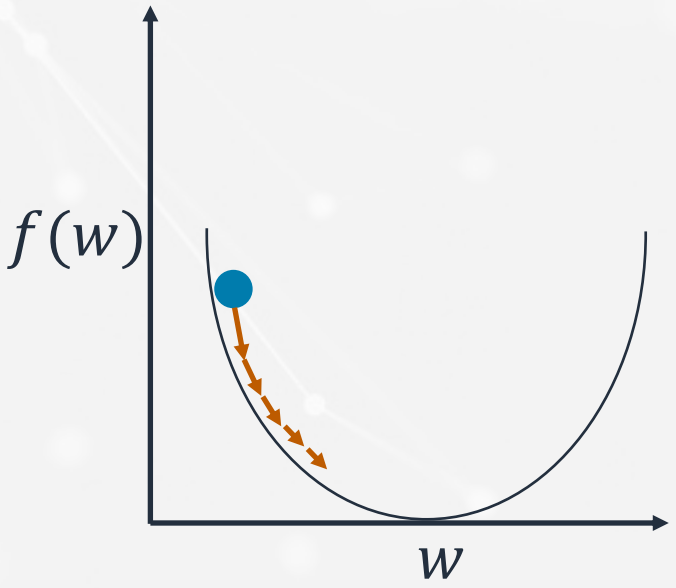
- Optimization methods: Methods to update/find weights.
  - Exhaustive search (trying all possible w's)
  - Gradient descent (iteratively updating w's)
  - Normal equation

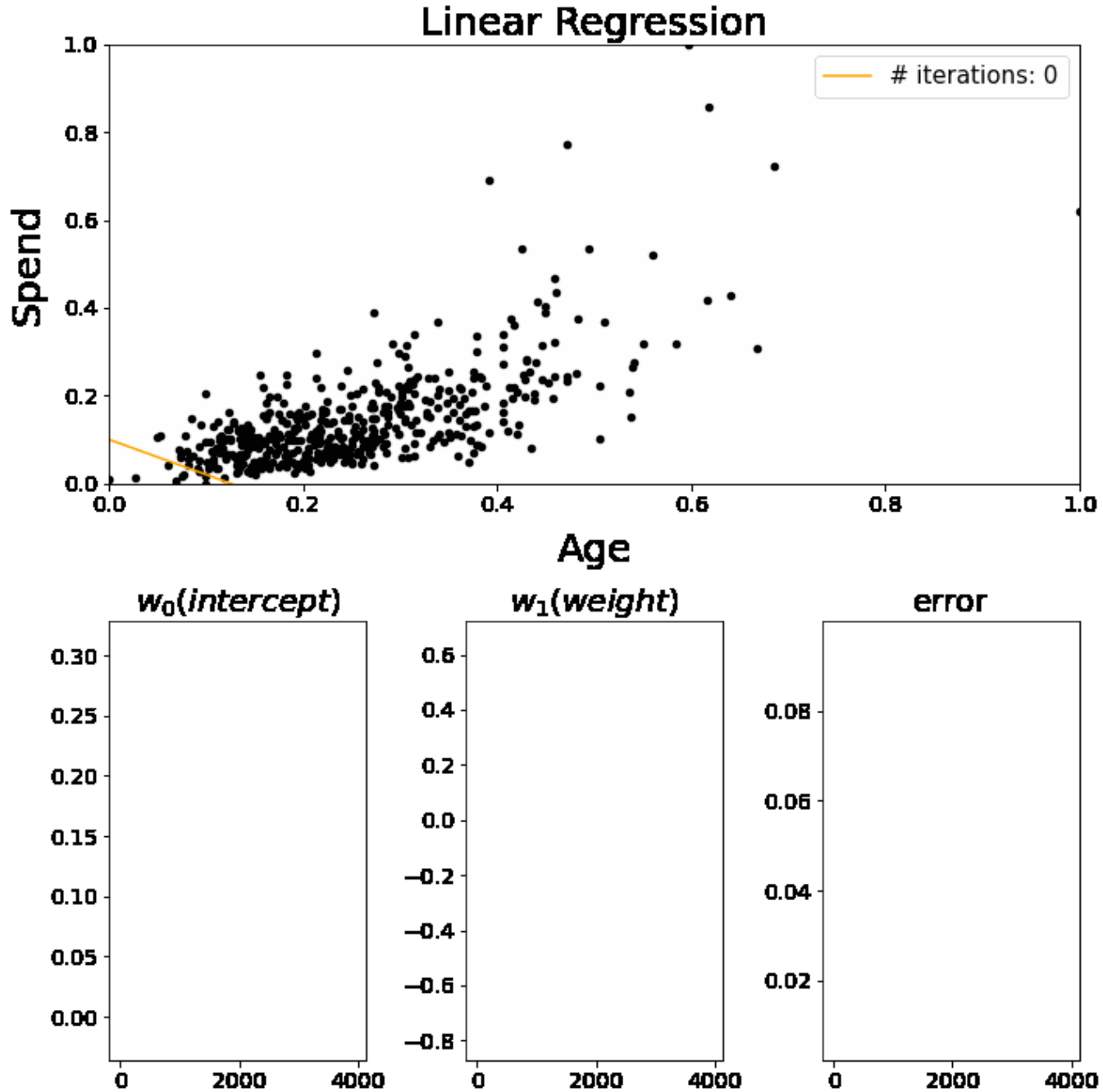
...

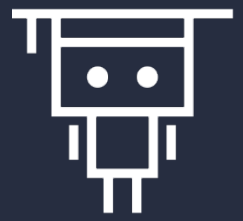
Gradient Descent Method

- Gradient Descent **method** uses gradients to find the **minimum** of a function **iteratively**.
  - » Taking **steps** (proportional to the gradient size) towards the minimum, in the **opposite** direction of the gradient.

- Gradient Descent Algorithm:
  - » Start at an initial point w
  - » Update: $w_{new} = w_{current} - step\ size\ * gradient$

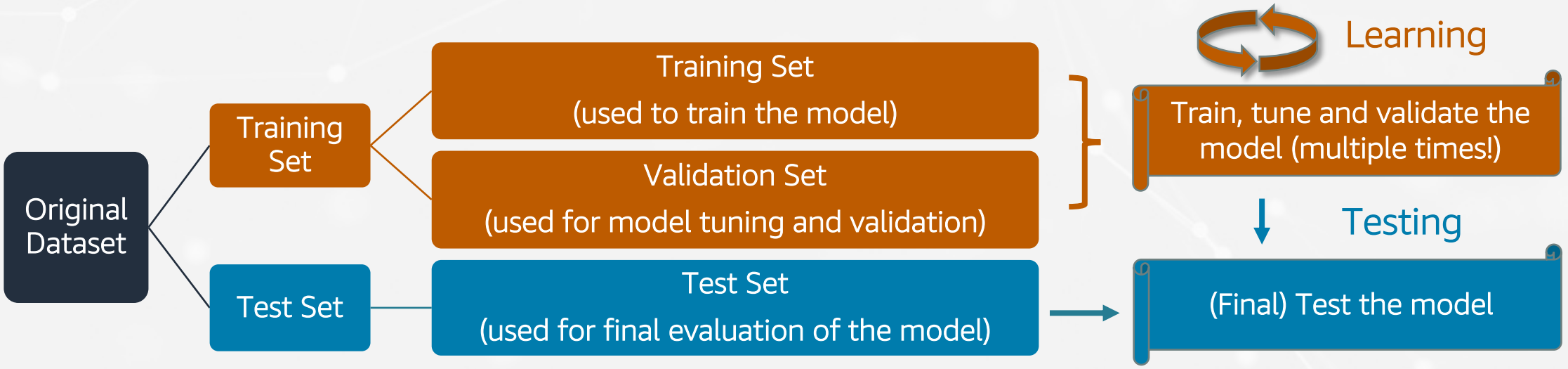






Model Performance Evaluation

Train-Test-Validation Split

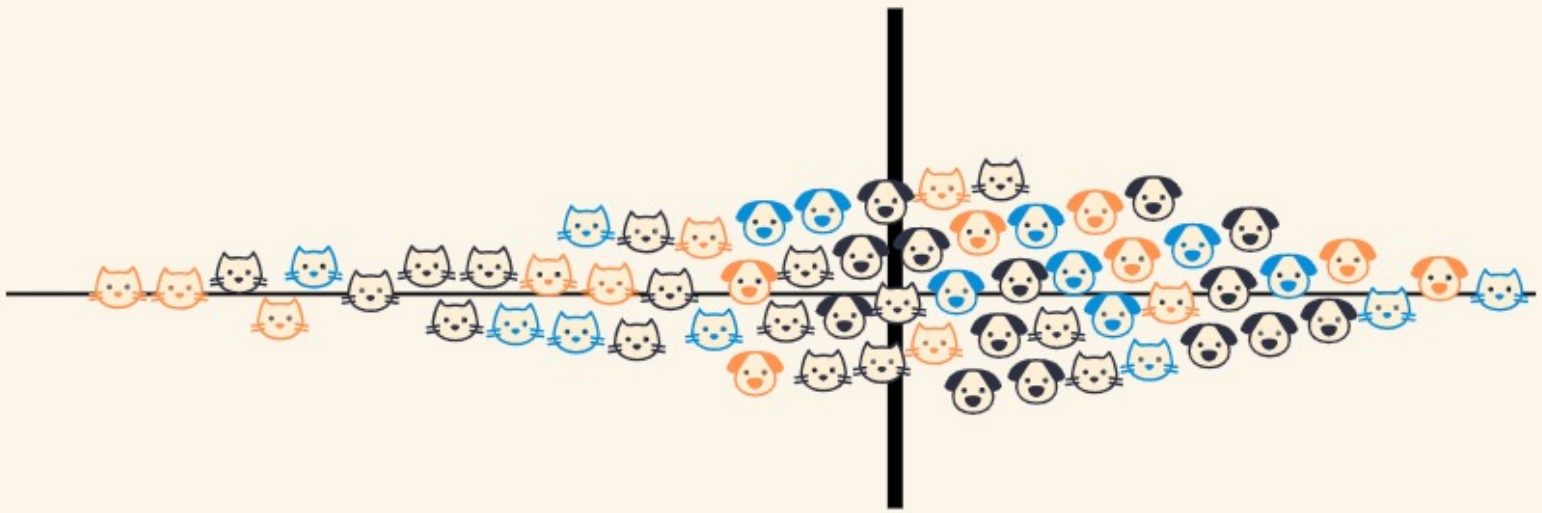


- Shuffle the dataset before the split!
- IID data in all 3 splits & balance across sensitive attributes.
- Sklearn splitting method doesn't allow to specify sensitive attribute up-front.

Summary

Model Features: None Weight Fluffiness Both

You may have noticed that the test accuracy of the "just fluffiness" model was higher than that of the "both features" model, despite the validation set selecting the latter model as the best. This occurrence of the validation performance not exactly matching the test performance might happen, yet it is not a bad thing. Remember that the test performance is not a number to optimize over — it is a metric to assess future performance. It allows us to estimate, with confidence, that our model can distinguish between cats and dogs with 87.5% accuracy.



Model Evaluation

- Models should be fair and as perform well.
  - **Fairness**: Model should produce similar outcomes for similar groups (or individuals).
  - **Performance**: Model should perform above a certain accuracy threshold or below error threshold.

- Models that don't meet the above criteria, should be re-evaluated.

Evaluation Example for Regression

1. Find distance (squared, absolute, …) between your prediction and labels.

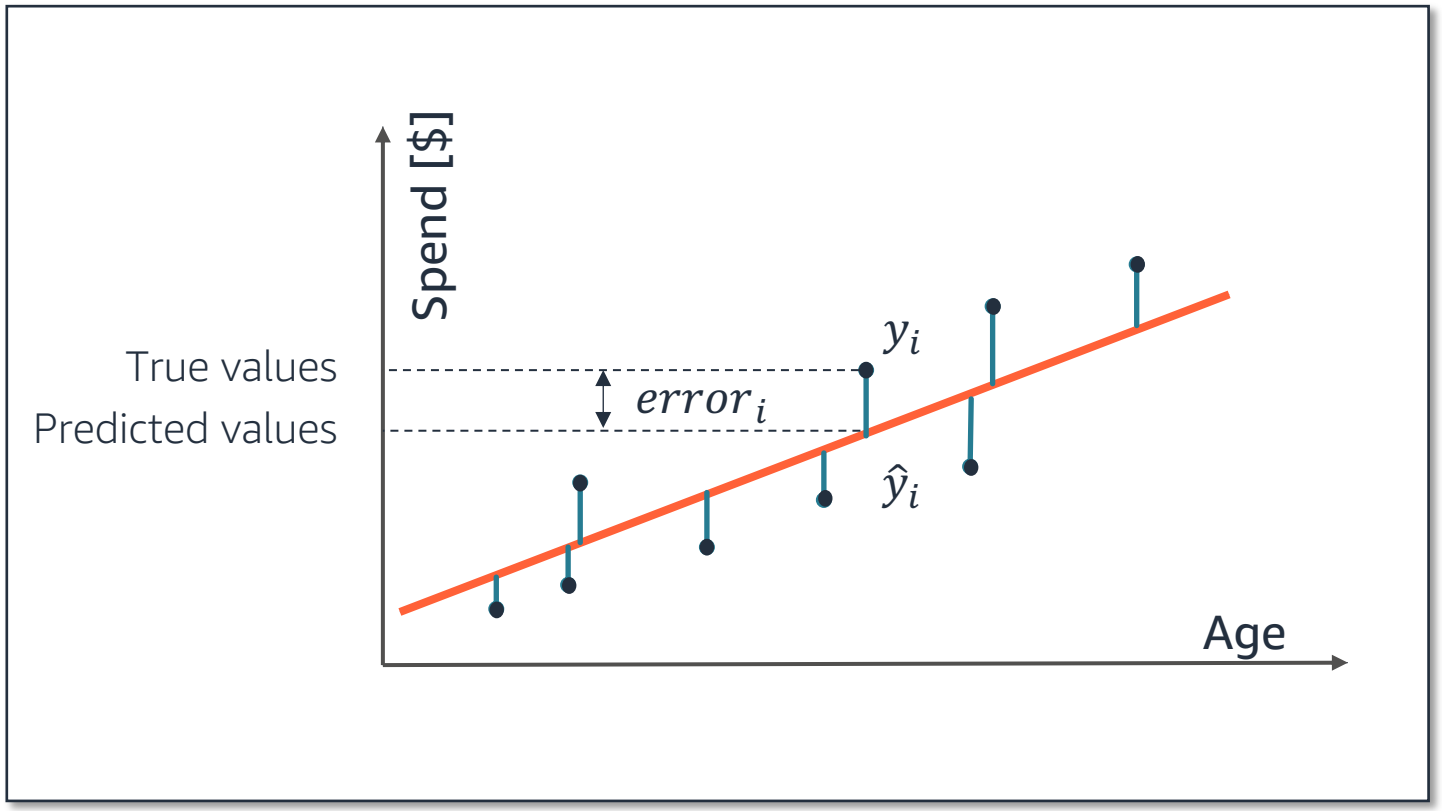


Fig.: Predicted Health Care Spend

Evaluation Example for Regression

1. Find distance (squared, absolute, …) between your prediction and labels.

2. Aggregate up into 1 value across whole dataset (e.g. mean error across squared distances, MSE).

$$\mathbf{MSE} = \frac{1}{N}\sum_{i}(y_i - \hat{y}_i)^2$$

N: number of datapoints

We are not yet talking about group membership of the datapoints here!

Evaluation Example for Classification

True State \ Prediction	1 (positive)	0 (negative)
1 (positive)	True Positive 11	False Negative 5
0 (negative)	False Positive 8	True Negative 12

1. Compare true state to predicted state.

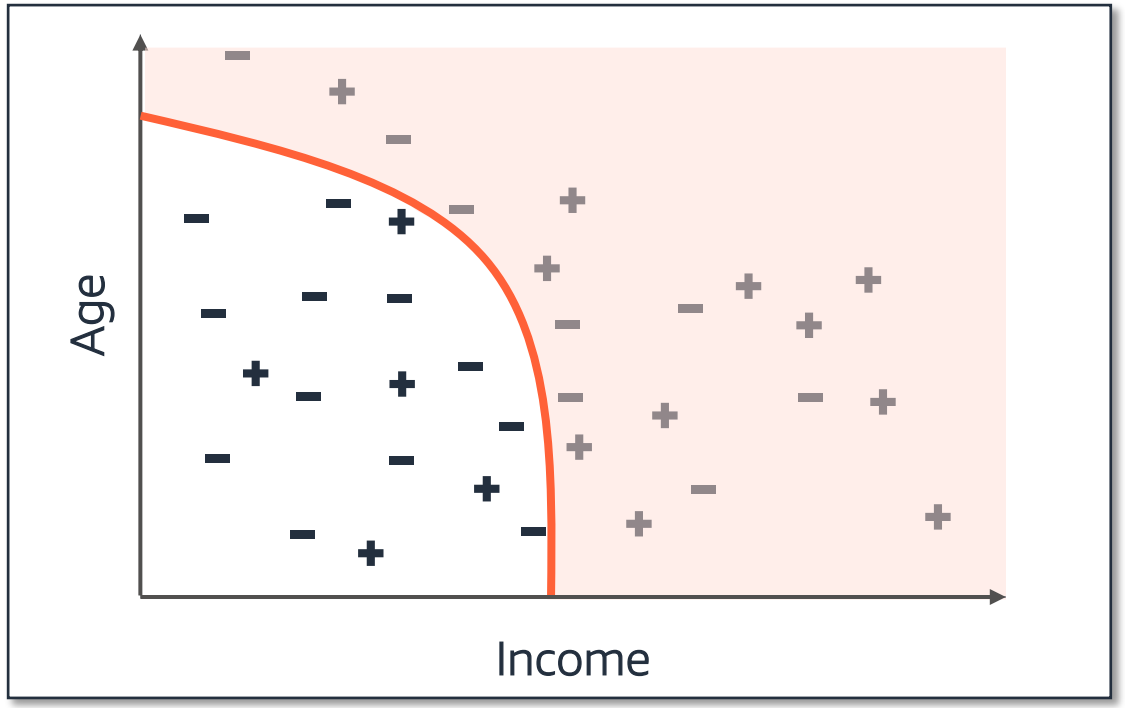
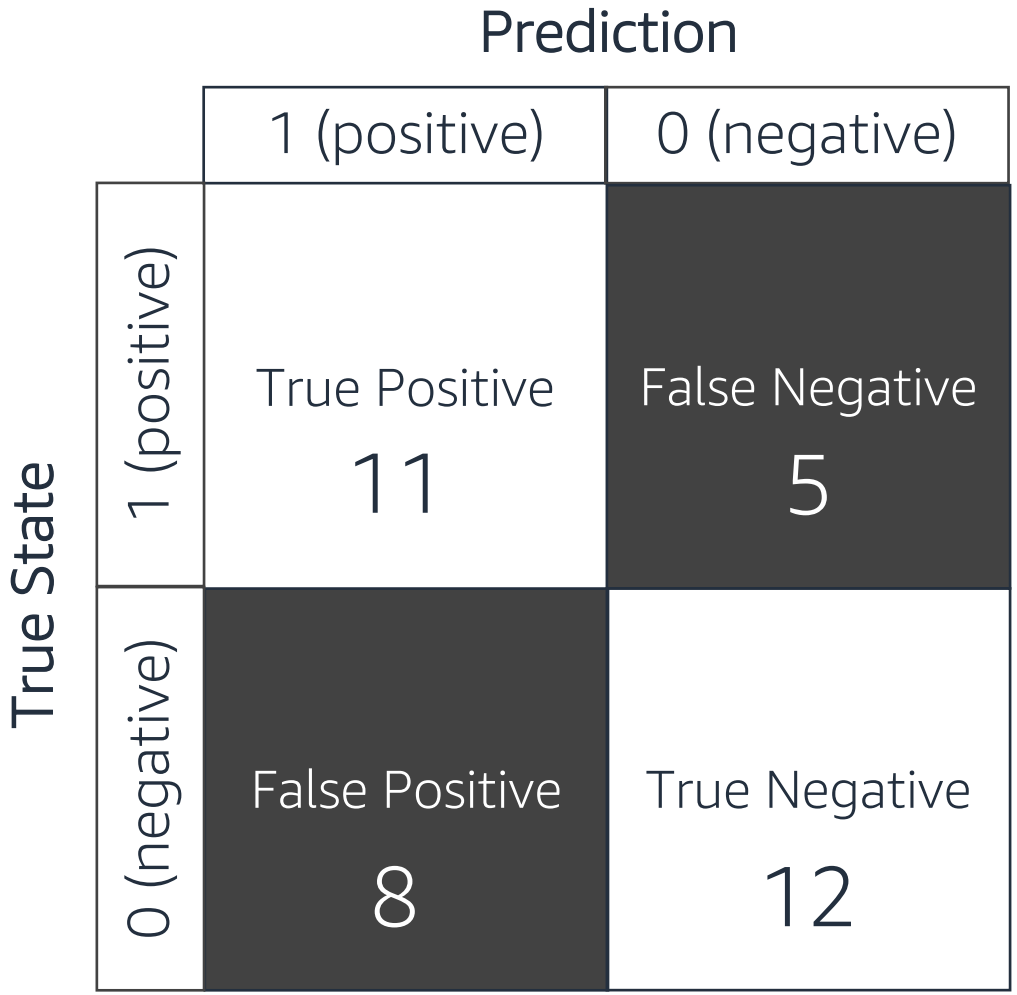


Fig.: Bank Loan Ground Truth

+ Approved
− Not Approved

MACHINE LEARNING UNIVERSITY

Evaluation Example for Classification



1. Compare true state to predicted state.
2. Aggregate up into 1 value across whole dataset by creating ratios (e.g. correct predictions over total: accuracy).

$$\boldsymbol{Accuracy} = \frac{\boldsymbol{TP} + \boldsymbol{TN}}{\boldsymbol{TP} + \boldsymbol{FP} + \boldsymbol{TN} + \boldsymbol{FN}}$$

E.g.: $$\frac{11 + 12}{11 + 5 + 8 + 12} = 0.64$$

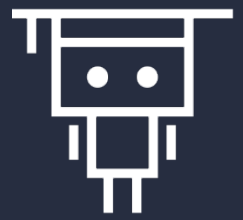
Considerations

- How can we be sure that this performance will be upheld in production?

→ Train-Test-Validation Split

- How can we include group membership in the evaluation?

→ Quantifying Bias



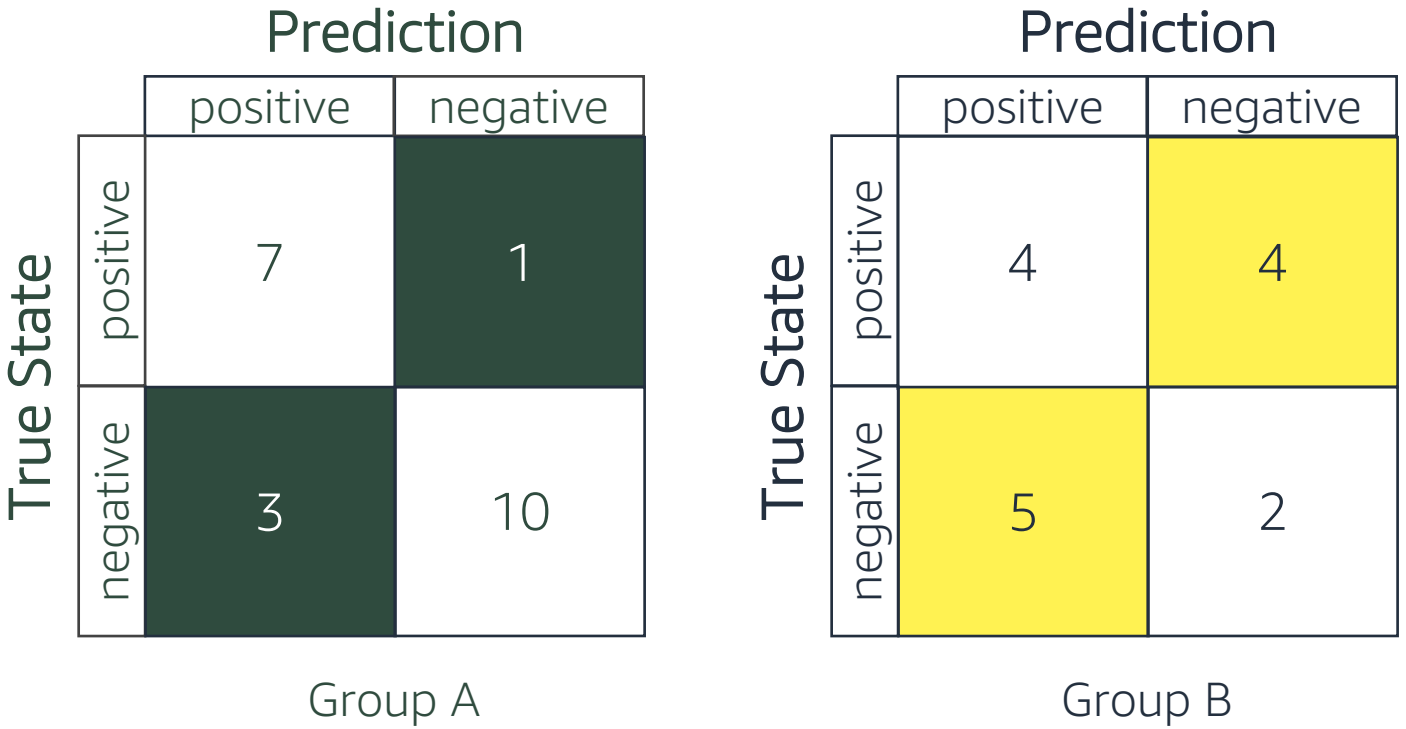
Quantifying Model Bias

Quantifying Bias

- Common evaluation methods (if not calculated group-wise), don't consider how different groups could be affected differently.

→ we need fairness-specific model performance measures that can consider group membership

Fairness Measure Example

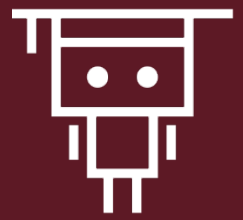


Accuracy difference (AD):
Difference in accuracy for 2 groups.

The closer to 0, the better!

$$A_A = \frac{7 + 15}{7 + 1 + 15 + 1} = 0.81 \qquad A_B = \frac{3 + 1}{3 + 5 + 4 + 1} = 0.4$$

AD = 0.81 – 0.4 = **0.41**



Notebook: MLA-RESML-LOGREG.ipynb

Notebook Content

- This notebook shows how to build a logistic regression (and how to derive a logistic regression from a linear regression).
- We will use the logistic regression model to predict classes.
- You will also see techniques to evaluate model performance (accuracy difference & DPPL).

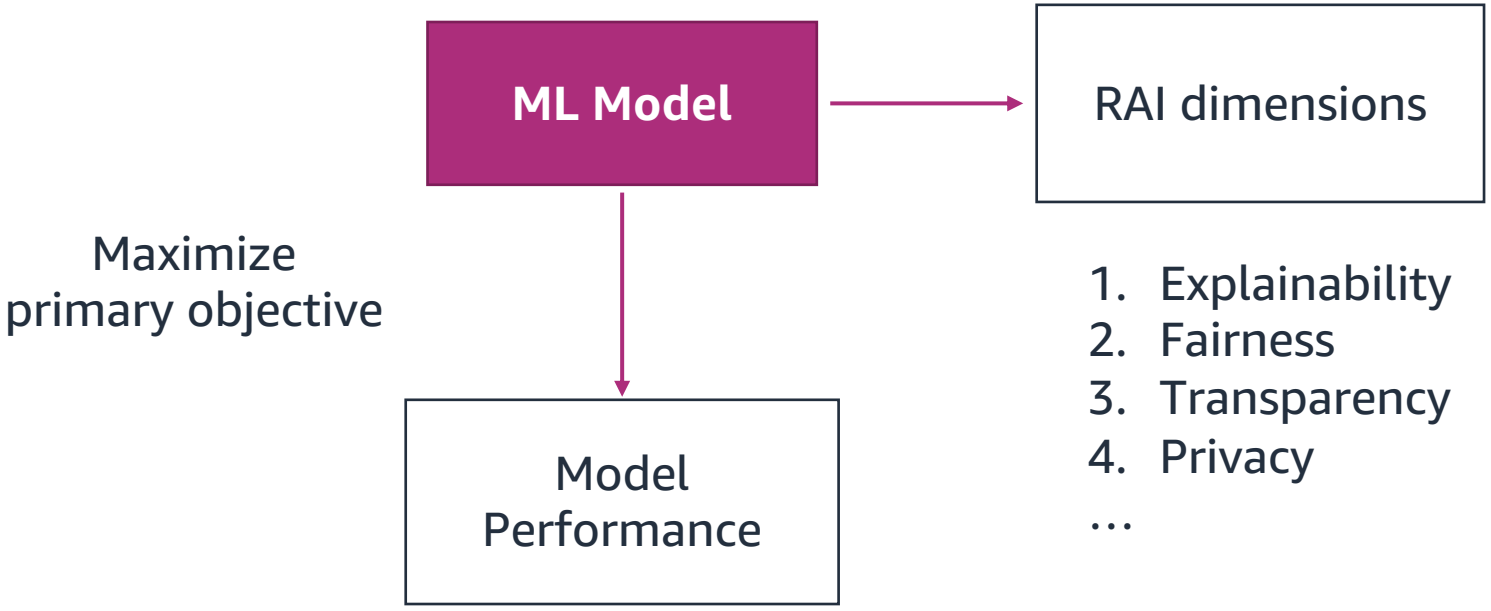
No ”one fits all” metric

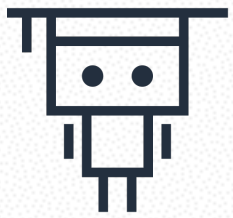
- Easier to measure a particular metric (e.g. disparity in outcomes), than fairness (no universally accepted definition & use case dependent).

- No “one fits all” (bias) metric! We need to look at different measures.
  - » If you find discrepancies debug the model
  - » Useful in spotting issues (and preventing misconceptions)

Tradeoffs in RAI & ML metrics

- Fairness metrics as well as general ML metrics can be at odds with each other (fair for individual ≠ fair for group, accuracy vs. bias mitigation, …)





MACHINE LEARNING
UNIVERSITY

Fairness Criteria

Fairness Criteria

- Fairness achievable through *mitigation of unwanted bias* (harmful *disparities* in system behavior or downstream *impact* on subpopulations)

- Fairness criteria describe connection between sensitive attribute and true/predicted labels mathematically

S. Barocas and M. Hardt (2017)

Fairness Criteria ≠ Social Equity

- S. Barocas and M. Hardt (2017) distinguish :
  - **Independence, Separation, Sufficiency** (mutually exclusive)
  - *Variations, relaxations* and related metrics exist.

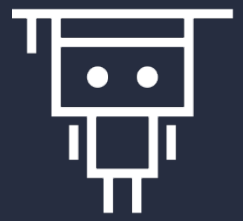
- Criteria do not necessarily map to established legal or social understandings of equity [Stanford Computational Policy Lab].

- Do not just optimize fairness criteria when training and hope that the problem gets better!

How to satisfy Fairness Criteria

- Fairness criteria help us understand how to prepare data, tweak the way models learn or how to make adjustments to model predictions with the goal to *reduce unwanted bias*.

- Use fairness criteria to measure and/or mitigate bias at different stages of the ML lifecycle:
  - **Pre-processing**
  - **In-processing**
  - **Post-processing**



Probability Basics

Probability Basics

- Probability, $Pr(A)$, of an event A, is the sum of the probabilities of the outcomes which make up A.

$$Pr(A) = \frac{\text{\# outcomes that make up event A}}{\text{total \# of possible outcomes}}$$

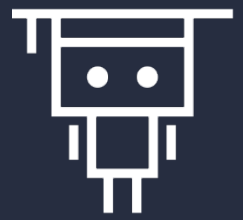
Basic Probability Intro

Conditional Probability

- Conditional probability, $Pr(B|A)$, of an event B is the probability that the event B will occur given the knowledge that event A has already occurred.

$$Pr(B|A) = \frac{\text{probability of A and B occurring}}{\text{probability of A}}$$

Conditional Probability Intro



Independence

Independence/Statistical Parity

- Having a certain attribute, A, is not related to the true outcome, Y.

→ probability for positive or negative true outcome, has to be equal:

$$Pr(Y = y \mid A = 0) = Pr(Y = y \mid A = 1)$$

- To measure bias of a dataset or estimator:

$$disparity = Pr(Y = y \mid A = 0) - Pr(Y = y \mid A = 1)$$

Independence/Statistical Parity

- There is no disparity if:

$$Pr(Y = y \mid A = 0) - Pr(Y = y \mid A = 1) = 0$$

where $\boldsymbol{A = 1}$ is the advantaged group.

- In practice, it will be hard to find a dataset/build a model that meets this.
- Relaxation with threshold, ϵ: Difference in Demographic Parity:

$$|Pr(Y = y \mid A = 0) - Pr(Y = y \mid A = 1)| \leq \epsilon$$

Independence/Statistical Parity

- Relaxation with threshold, ϵ, also known as **Disparate Impact (DI)**:

$$\frac{Pr(Y = y \mid A = 0)}{Pr(Y = y \mid A = 1)} \leq \tau$$

where $\boldsymbol{A = 1}$ is the advantaged group.

Feldman et. al (2015)

Independence/Statistical Parity

- For regression, we consider score ranges (prediction scores, r, should be statistically independent of the sensitive attribute):

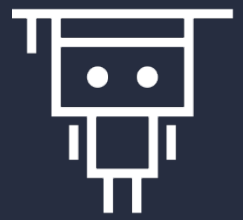
$$|Pr(R \geq r \mid A = 1) - Pr(R \geq r \mid A = 0)| \leq \epsilon$$

Difference in Mean

- Difference in mean predictions per group

$$mean(\hat{y}_{A=0}) - mean(\hat{y}_{A=1})$$

- The closer to 0, the fairer an outcome.



Separation

Separation

- Prediction, $\hat{Y}$, is conditionally independent of the attribute, A, given the true outcome, Y.

  → true positive rate or false positive rate have to be equal for both attribute values, $A = 0$ and $A = 1$:

$$Pr\left(\hat{Y} = \hat{y} \,\middle|\, A = 1, Y = y\right) = Pr\left(\hat{Y} = \hat{y} \middle| A = 0, Y = y\right)$$

- Requires y to reflect what society considers fair.

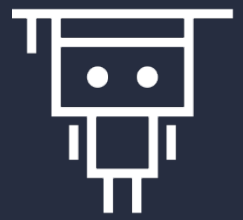
Separation

- Prediction, $\hat{Y}$, is conditionally independent of the attribute, A, given the true outcome, Y.

→ true positive rate or false positive rate have to be equal for both attribute values, $A = 0$ and $A = 1$:

$$Pr\left(\hat{Y} = \hat{y} \,\middle|\, A = 1, Y = y\right) = Pr\left(\hat{Y} = \hat{y} \middle| A = 0, Y = y\right)$$

- Relaxation: $y = 1$ to equalize TPR (equality of opportunity), $y = 0$ to equalize FPR

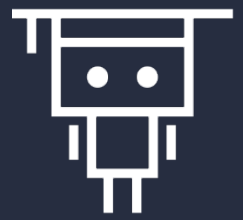


Sufficiency

Sufficiency

- Sufficiency is a target-based test:
  - Requires that the **true outcome (target) is conditionally independent** of the sensitive attribute, given the prediction value.
  - Uses **numerical values** (for classifiers, use predicted probabilities, not class value)
  - Usually only possible to test for model developers (as the prediction that is shown to users/customers is a decision, not a probability value)

- Ex.: In lending, applicants with similar predicted probabilities should have same rate of acceptance across different groups



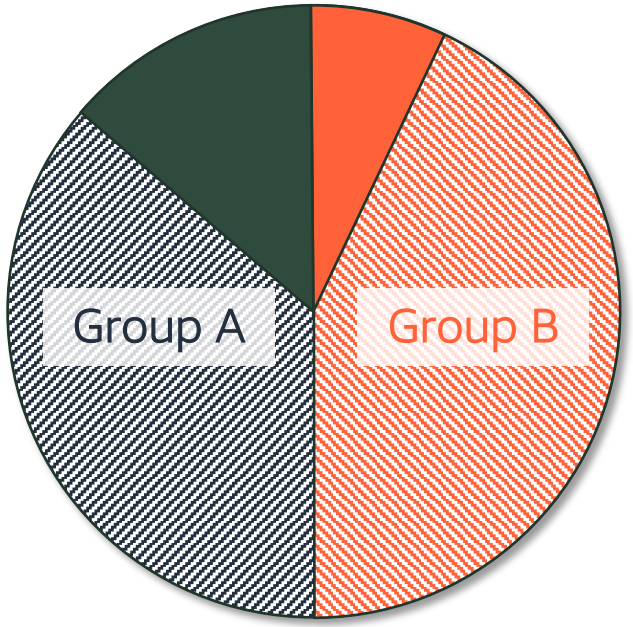
General Limitations of all Fairness Criteria

General limitations

- Optimizing for one criteria can lead to making another criteria worse.
- Combined attributes can lead to accumulated bias (**intersectional fairness**, e.g. race & gender)
- Potential **legal restrictions** for use of sensitive attributes → mindful; certain fairness criteria require sensitive attribute to be known.
- **Insufficiency** of criteria (lazy solutions exist).

Incompatibility of Criteria

Separation: Everyone that can pay back, gets approved (same TPR).

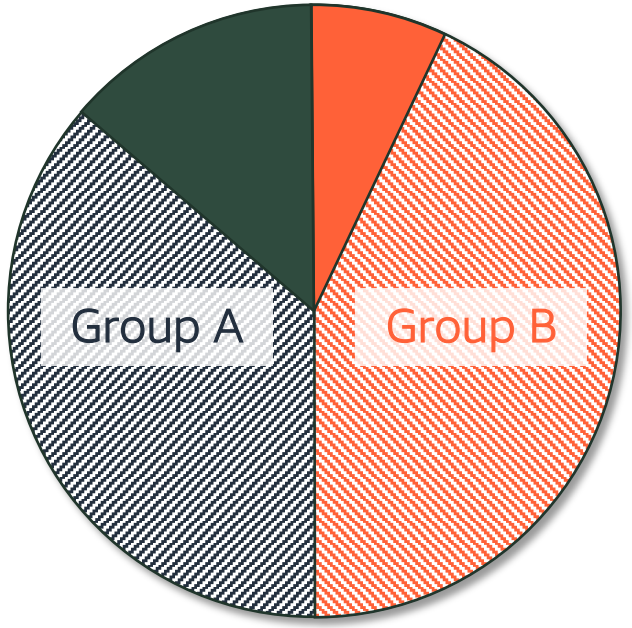


Cannot pay back

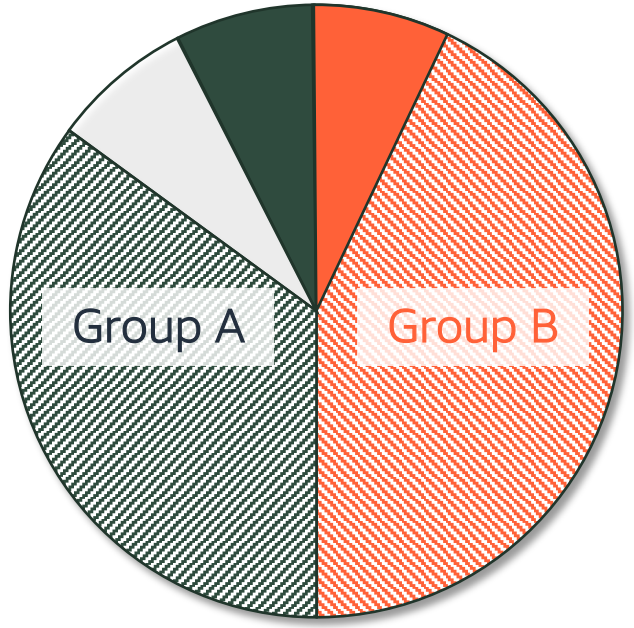
Can pay back

Incompatibility of Criteria

Separation: Everyone that can pay back, gets approved (same TPR).



Independence: Same fraction in both groups gets approved.



Cannot pay back

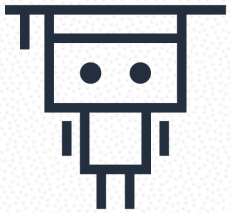
Can pay back

Impossible to achieve simultaneously!

Non-ideal world

- Most fairness methods assume existence of ideal world and measure deviations from it
- Certain methods fail to account for non-ideal world behavior
- Carefully chose metric and consider:
  - » Harms of misguided solutions
  - » Responsibilities of decision makers
  - » …

Algorithmic Fairness from a Non-ideal Perspective
(Lipton et al., 2020)



MACHINE LEARNING UNIVERSITY

Pre-Processing for Bias Mitigation

Pre-Processing Methods

- Collection of pre-processing methods that modify data or labels before the model is trained to mitigate bias in dataset:
  - **Suppression** (delete sensitive feature – beware of proxies)
  - **Re-weighting** (make group membership & outcome independent)
  - **Transformation** of data (generate synthetic features or labels)
  - …

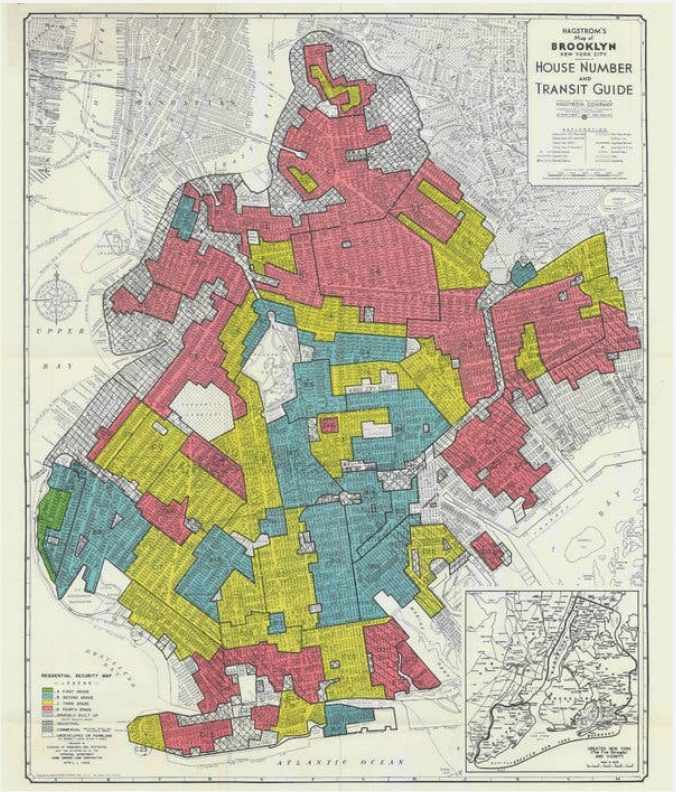
Suppression

- “Fairness through unawareness” is ineffective due to the existence of proxies (rich data sources).

  Dwork et al (2011)

- Deleting features makes it impossible to disentangle effects downstream.

- Check whether legal requirements impact ability to use certain features or whether they need to, or should be, removed.



Redlining in banking industry (law forbade to use race, but ZIP code acted as proxy)

Re-weighting

- If two events (*here*: outcome & value of attribute) are independent then we can multiply probability of the attribute value occurring with probability of a certain outcome occurring

$$Pr(A = a, Y = y) = Pr(A = a) \times Pr(Y = y)$$

- Re-arrange for re-weighting factor:

$$W := \frac{Pr(A = a) \times Pr(Y = y)}{Pr(A = a, Y = y)}$$

Re-weighting Example

- Calculate all possible permutations of attributes and outcomes. E.g.: $Sex = female$ reduces chance of positive outcome → need to increase weight

$$\frac{Pr(Sex = female) \times Pr(Y = +)}{Pr(Sex = female, Y = +)} = \frac{\frac{5}{10} \times \frac{6}{10}}{\frac{2}{10}} = 1.5$$

Sex	Ethnicity	Highest Degree	Job Type	Cl.	Weight
m	native	h. school	board	+	0.75
m	native	univ.	board	+	0.75
m	native	h. school	board	+	0.75
m	non-nat.	h. school	healthcare	+	0.75
m	non-nat.	univ.	healthcare	-	2
f	non-nat.	univ.	education	-	0.67
f	native	h. school	education	-	0.67
f	native	none	healthcare	+	1.5
f	non-nat.	univ.	education	-	0.67
f	native	h. school	board	+	1.5

Calders et al. (2009)

Re-weighting Example Code

Calculate weights with custom function (or fairness Python library), then apply weights in models that allow sample weights (e.g. Logistic Regression, KNN, …)

```
lr = LogisticRegression()
lr.fit(X_train, y_train, sample_weight = weights)
```

Equivalent to up-sampling but more efficient as we don't create duplicate datapoints.

Disparate Impact for Transformation

Ratio of success rate for datapoints with disfavored attribute value to the success rate of the favored attribute value, Disparate impact:

$$\frac{Pr(Y = 1 | A = 0)}{Pr(Y = 1 | A = 1)} \leq \tau$$

Success and favored attribute value are denoted as 1.

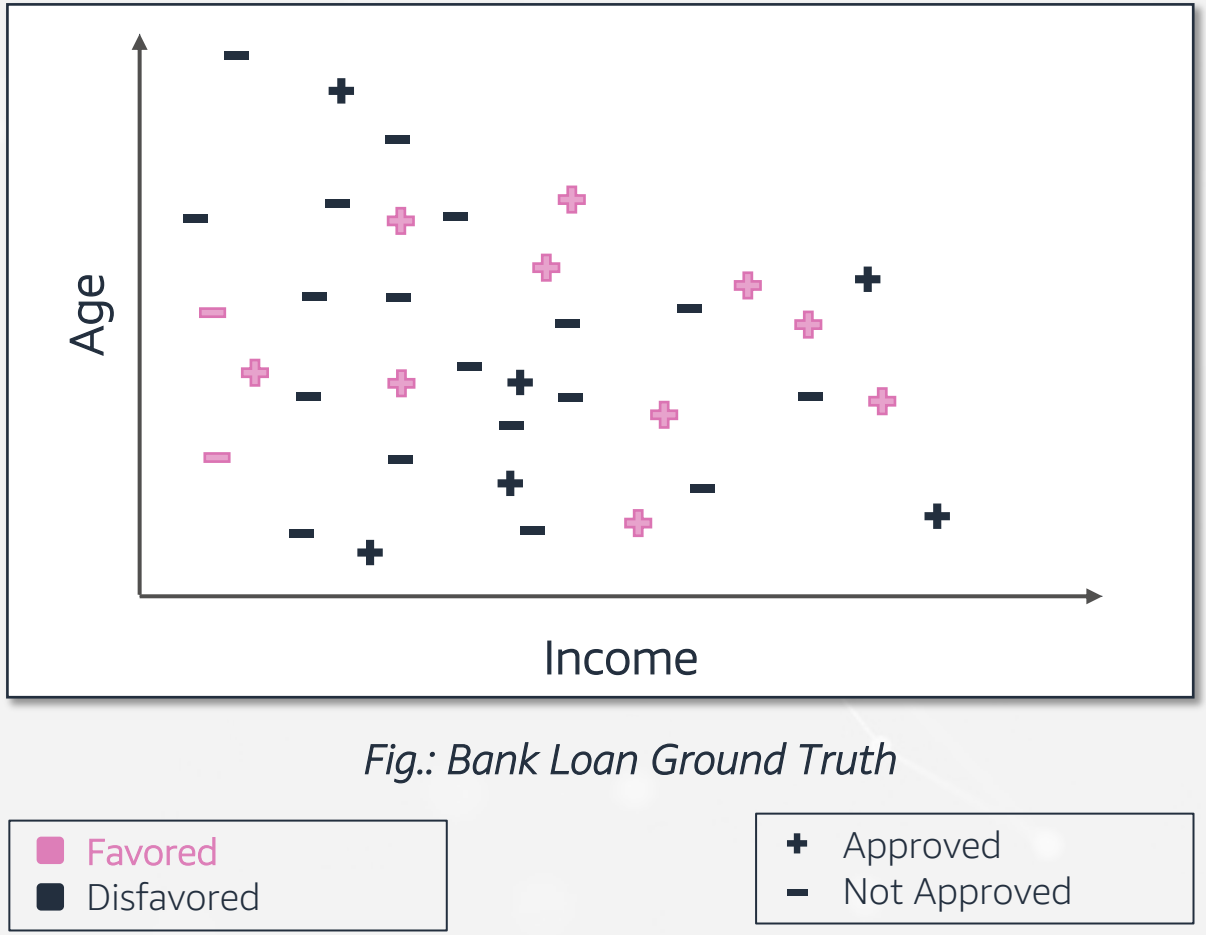


Fig.: Bank Loan Ground Truth

[Feldman et. al (2015)]

Disparate Impact for Transformation

- Success with favored attribute value:

$$Pr(Y = 1 | A = 1) = \frac{10}{12}$$

- Success with unfavored attribute value:

$$Pr(Y = 1 | A = 0) = \frac{6}{24}$$

- Disparate impact: $DI = 0.3$

Four-fifths rule:
Adverse impact if DI ≤ 0.8

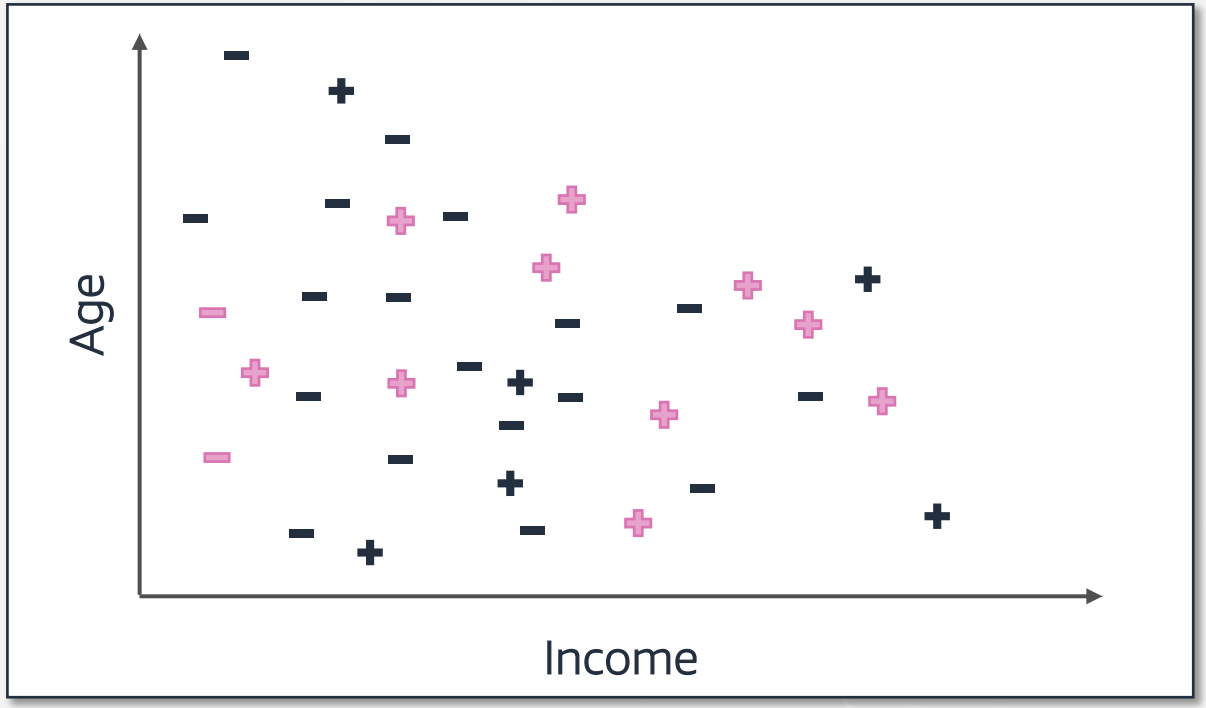


Fig.: Bank Loan Ground Truth

Disparate Impact Removal

- Theorem: A classifier is considered free of disparate impact if the value the sensitive attribute assumes cannot be predicted from X (remaining features).

- In order to reduce DI as much as possible:
  1. Set ϵ threshold
  2. Adjust distributions of X such that A cannot be predicted at ϵ-threshold:
     - » Combinatorial (rank-preserving repair)
     - » Counterfactual repair
     - » …

DI Removal: Example SAT

1. Split groups based on sensitive attribute (→ sub-group) & order.

2. Split sub-groups based on pos./neg. outcome (→ outcome sub-groups).

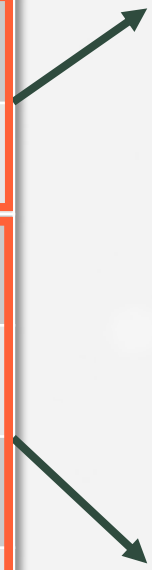
3. Replace feature value of disfavored group with median feature value of favored outcome sub-group.

Gender	Score	Admission
F	1400	1
F	1300	0
M	1400	0
M	1500	1
M	1400	0
M	1500	1

Algorithmic Fairness in ML (Duke)

DI Removal: Example SAT

Gender	Score	Admission
F	1400	1
F	1300	0
M	1400	0
M	1500	1
M	1400	0
M	1500	1



Gender	Score	Admission
F	1300	0
F	1400	1

Gender	Score	Admission
M	1400	0
M	1400	0
M	1500	1
M	1500	1

DI Removal: Example SAT

Gender	Score	Admission
F	1400	1
F	1300	0
M	1400	0
M	1500	1
M	1400	0
M	1500	1

Gender	Score	Admission
F	1300	0
F	1400	1

Gender	Score	Admission
M	1400	0
M	1400	0
M	1500	1
M	1500	1

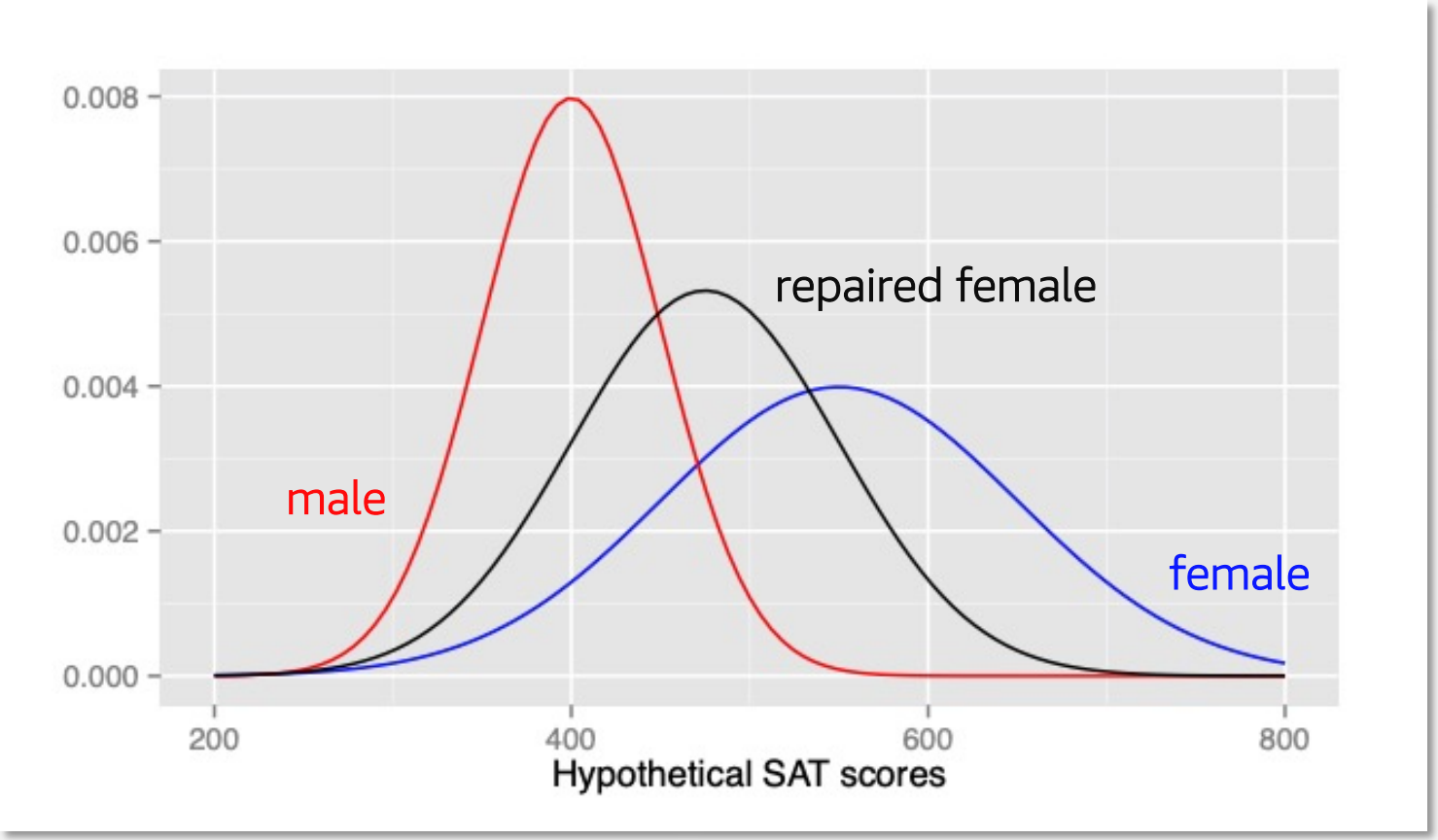
1400

1500

Gender	Score	Admission
F	1400	0
F	1500	1

The code is implemented such that the median is taken as left of center. E.g. [1,2,3,4] == 2, not 2.5.

DI Removal: Example SAT



In practice, use SMOTE to generate synthetic data for underrepresented individuals (e.g. women with high SAT scores) to increase prevalence.

Feldman et al. (2015) Certifying and removing disparate impact

Disparate Impact as Bias Quantifier

Ratio of success rate for datapoints with disfavored attribute value to the success rate of the favored attribute value

$$\frac{Pr(Y = 1|A = 0)}{Pr(Y = 1|A = 1)} \leq \tau$$

$$DI = \frac{{}^{11}/_{24}}{{}^{7}/_{12}} = 0.78$$

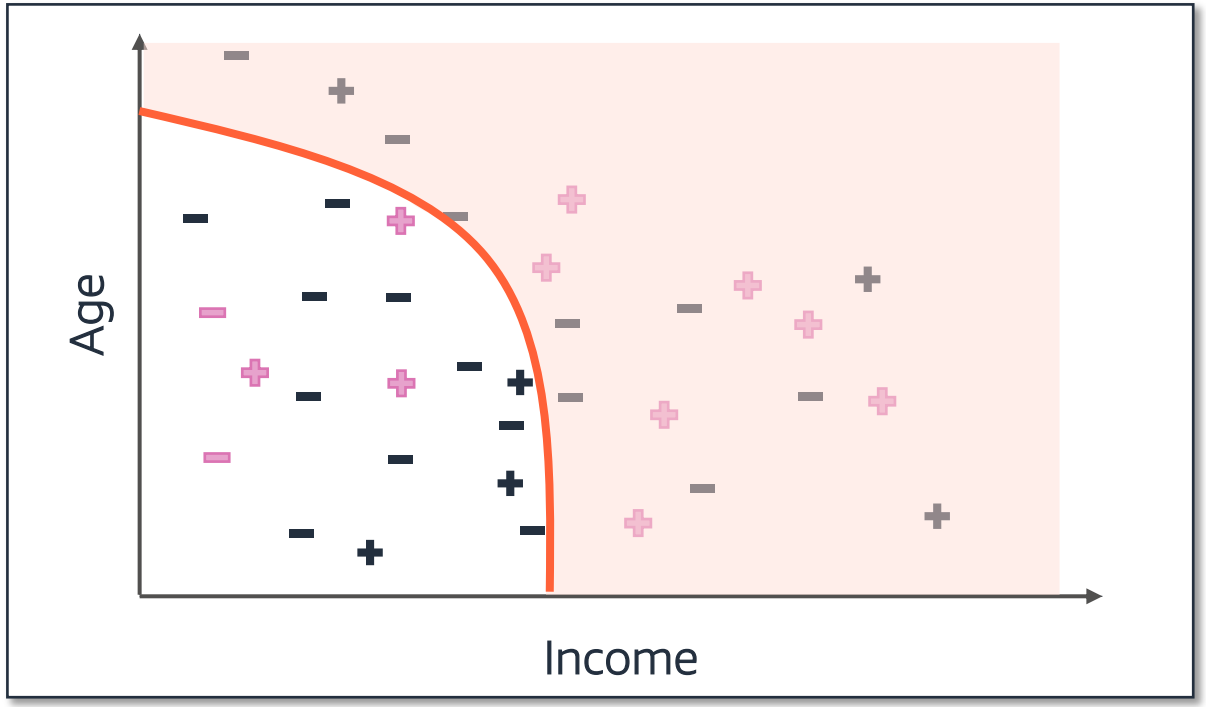
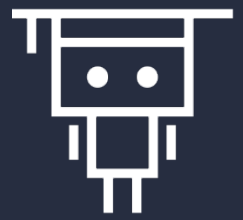


Fig.: Bank Loan Approval Predictions

[Feldman et. al (2015)]



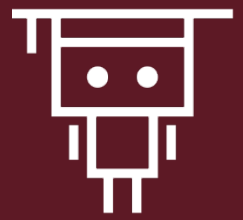
Limitations Pre-Processing Methods

Limitations

- Trade-off between accuracy and fairness improvement (e.g. applying weights to data decreases accuracy of model)

- Once information is removed, we cannot re-engineer back; might be better to not lose information early into process:
  - » Can't be sure if all proxies captured
  - » Synthetic inputs can be far from natural distributions

- Allows 'laziness' (accept top 50% of favored group and random 50% of unfavored group)



Notebook: MLA-RESML-DI.ipynb

MLA-RESML-DI.ipynb Notebook

- Shows how to quantify disparate impact and the implementation of a basic disparate impact remover (with different repair levels).

- Uses logistic regression in the model.