CMPT 733
Automated Machine Learning (AutoML)

Instructor Zhengjie Miao

Course website https://coursys.sfu.ca/2025sp-cmpt-733-g1/pages/

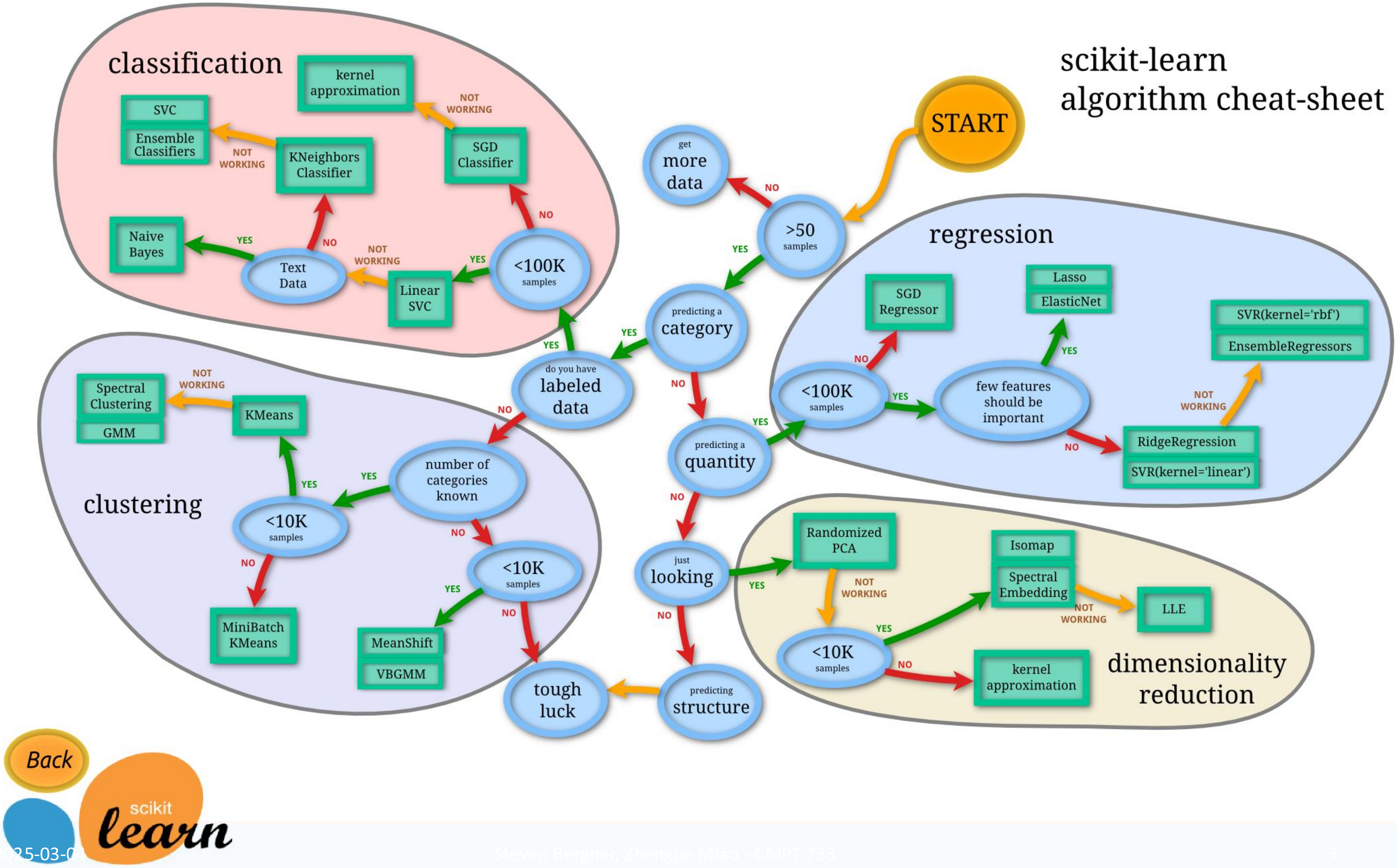
Based on the slides by Lydia Zheng and Jiannan Wang

DL Applications

https://en.wikipedia.org/wiki/Amazon_Alexa

https://didyouknowbg8.wordpress.com/2024/02/24/yolov9-a-leap-forward-in-object-detection-performance/

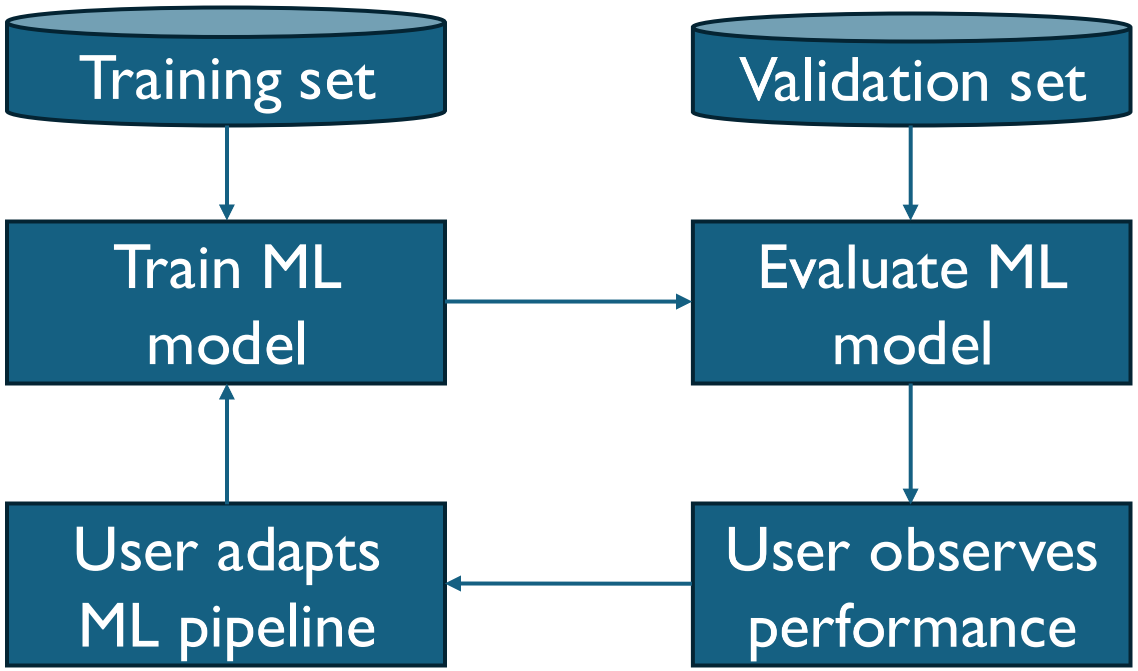
Machine Learning Tasks



Motivation

- Machine learning is very successful
- To build a traditional ML pipeline:
  - Domain experts with longstanding experience
    - Specialized data preprocessing
    - Domain-driven meaningful feature engineering
    - Picking right models
    - Hyper-parameter tuning

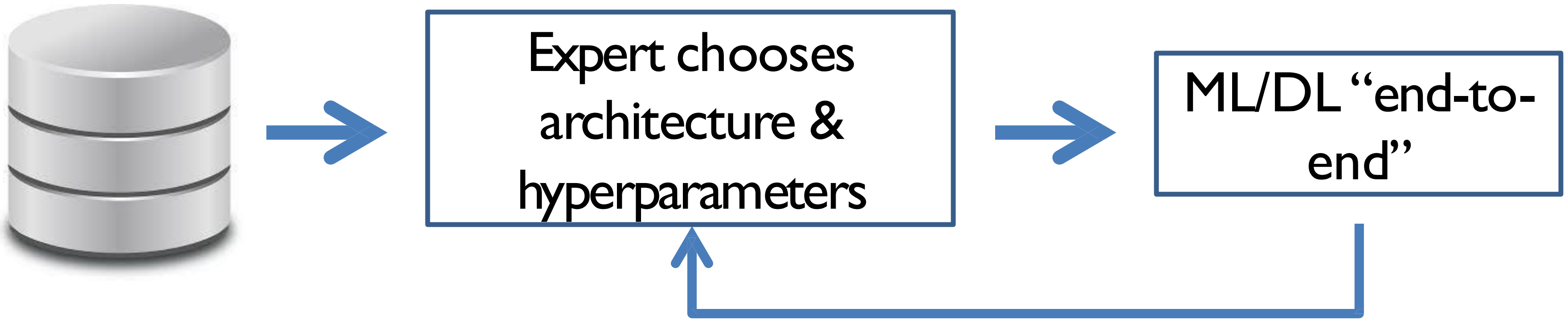
ML Workflow



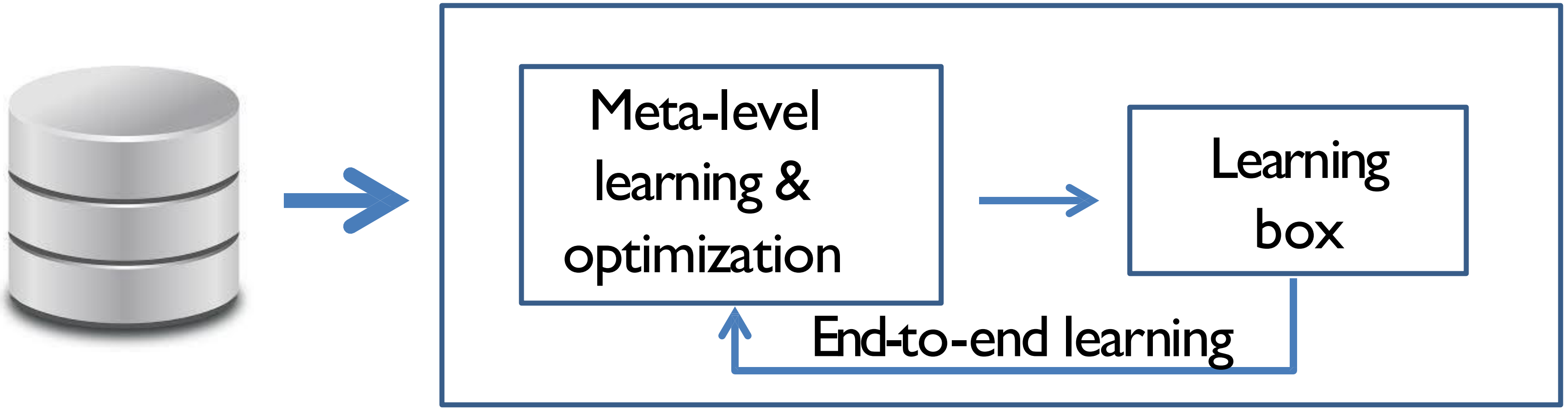
Users indirectly teach machines how to learn.

Classic ML and AutoML

Current ML/DL practice



AutoML: true end-to-end learning



Hutter & Vanschoren: AutoML, Neurips' 18 tutorial

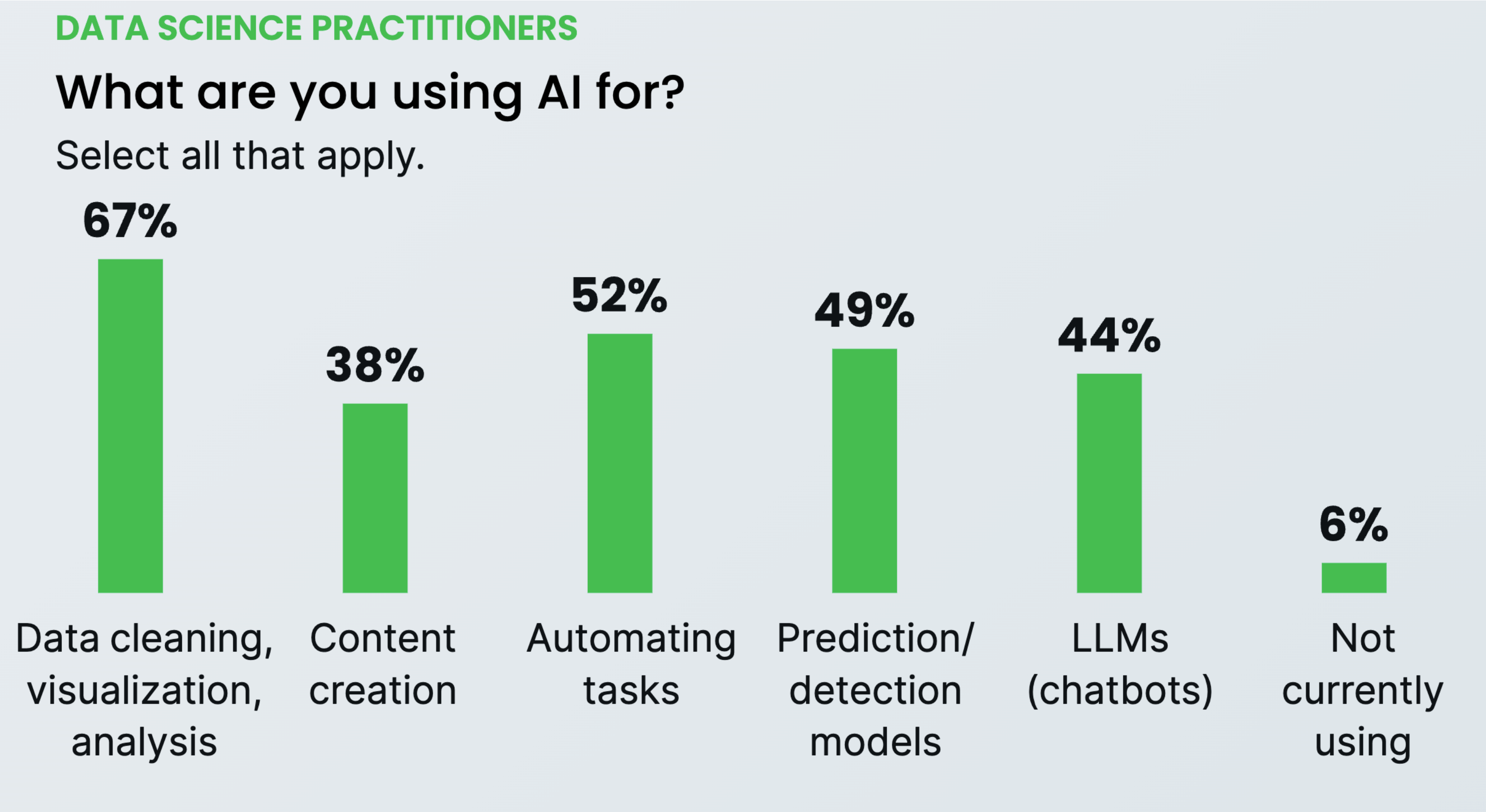
Recap: Trend in 2023

Thinking about your current role, what tasks are most time consuming? (Responses ranked from most to least time consuming)

1. Data Prep
2. Data Cleaning
3. Data Visualization
4. Model Training
5. Model Selection
6. Reporting and Presentations
7. Deploying Models
8. Other

n=1,071

Trend in 2024



https://www.anaconda.com/resources/report/state-of-data-science-report-2024

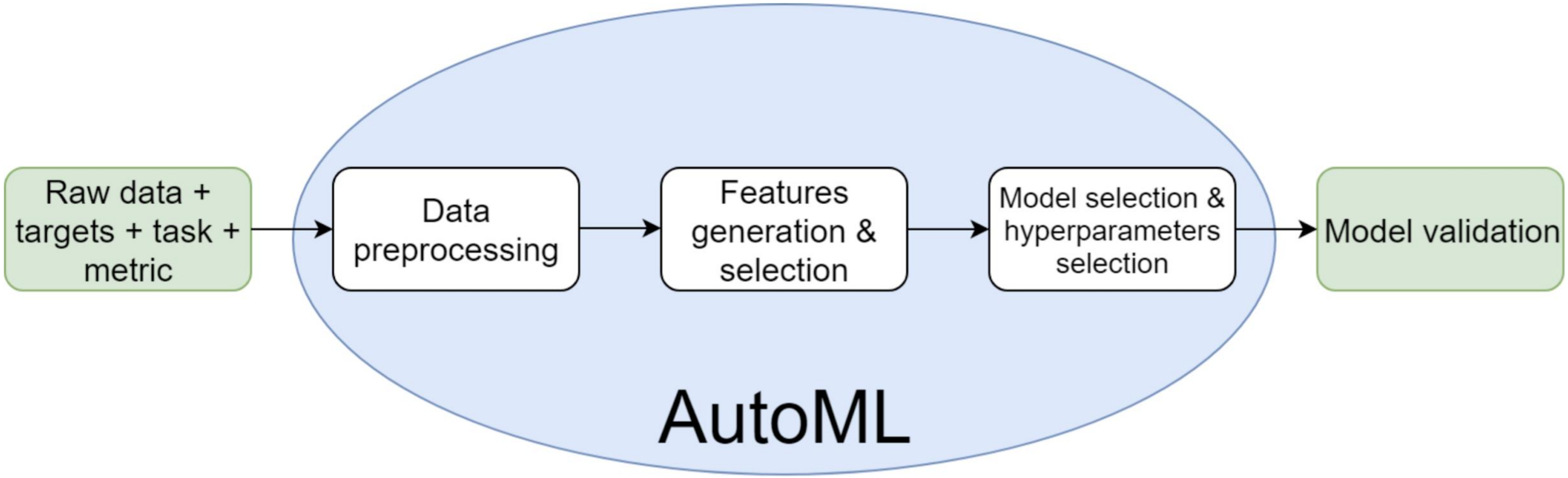
AutoML Vision

- For Non-Experts
  - AutoML allows non-experts to make use of machine learning models and techniques without requiring to become an expert in this field first
  - https://en.wikipedia.org/wiki/Automated_machine_learning

- For Data Scientists
  - AutoML aims to augment, rather than automate, the work and work practices of heterogeneous teams that work in data science.
  - Wang, Dakuo, et al. "Human-AI Collaboration in Data Science: Exploring Data Scientists' Perceptions of Automated AI." Proceedings of the ACM on Human-Computer Interaction 3.CSCW (2019): 1-24.

What is AutoML?

- Automate the process of applying machine learning to real-world problems



Outline

- Auto Feature Selection (Lecture 5)
- Auto Hyperparameter Tuning (part in Lecture 5)
- Auto Feature Generation (This Lecture) Neural Architecture Search (This Lecture)

Auto Feature Generation

Motivation

- The model performance is heavily dependent on quality of features in dataset
- It's time-consuming for domain experts to generate enough useful features

Feature Generation

- Unary operators (applied on a single feature)
  - Discretize numerical features
  - Apply rule-based expansions of dates
  - Mathematical operators (e.g., Log Function)
- Higher-order operators (applied on 2+ features)
  - Basic arithmetic operations (e.g., +, -, ×, ÷)
  - Group-by Aggregation (e.g., GroupByThenAvg, GroupByThenMax)

Katz, Gilad, Eui Chul Richard Shin, and Dawn Song. "Explorekit: Automatic feature generation and selection." 2016 IEEE 16th International Conference on Data Mining (ICDM). IEEE, 2016.

Featuretools

- An open source library for performing **automated feature engineering**
- Design to fast-forward feature generation across multi-relational tables

Concepts

- **Entity** is the relational tables
- An **EntitySet** is a collection of entities and the relationships between them
- **Deep Feature Synthesis (DFS)**
  - Algorithm that creates features by aggregating and transforming data across linked tables
  - Feature Primitives
    - Unary Operator: transformation (e.g., MONTH)
    - High-order Operator: Group-by Aggregation (e.g., GroupByThenSUM)

Entity sets

Customer

Customer_id	Birthdate	MONTH(Birthdate)	SUM(Product.Price)
1	1995-09-28	9	$500
2	1980-01-01	1	...
3	1999-02-02	2	...
...	...	...	...

Unary Operator: MONTH

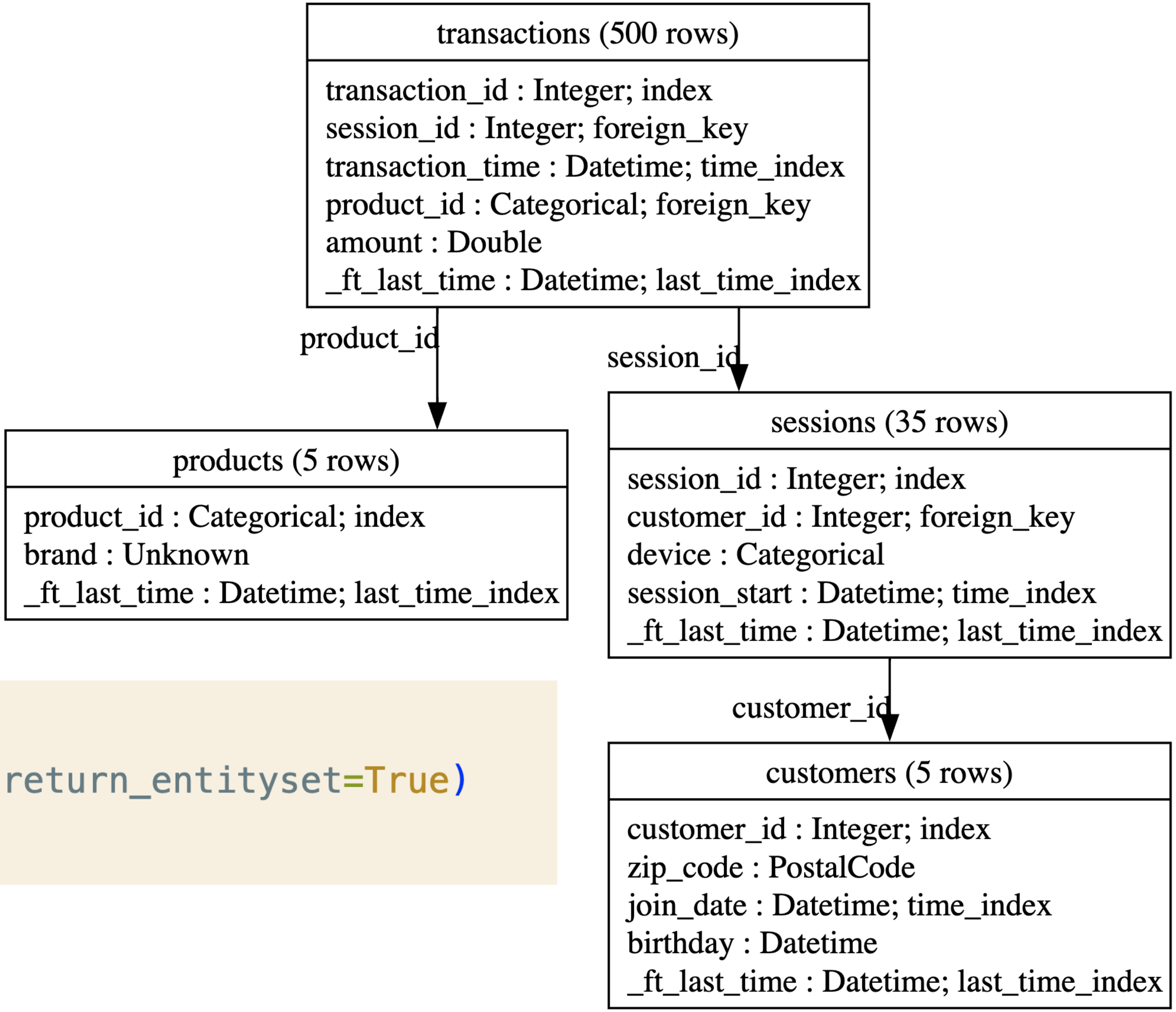
GroupBy ThenSUM:

Product

Product_id	Customer_id	Name	Price
1	1	Banana	$100
2	1	Banana	$100
3	1	Orange	$300
4	2	Apple	$50
...	...	...	...

Feature Primitives

Example




```python
import featuretools as ft
es = ft.demo.load_mock_customer(return_entityset=True)
es.plot()
```

```python
es['transactions'].head()
```

✓ 0.0s

	transaction_id	session_id	transaction_time	product_id	amount	_ft_last_time
298	298	1	2014-01-01 00:00:00	5	127.64	2014-01-01 00:00:00
2	2	1	2014-01-01 00:01:05	2	109.48	2014-01-01 00:01:05
308	308	1	2014-01-01 00:02:10	3	95.06	2014-01-01 00:02:10
116	116	1	2014-01-01 00:03:15	4	78.92	2014-01-01 00:03:15
371	371	1	2014-01-01 00:04:20	3	31.54	2014-01-01 00:04:20

```python
es.relationships
```

✓ 0.0s

```
[<Relationship: transactions.product_id -> products.product_id>,
 <Relationship: transactions.session_id -> sessions.session_id>,
 <Relationship: sessions.customer_id -> customers.customer_id>]
```

```python
# Define Relationship
rel = ft.Relationship(
    es,
    parent_dataframe_name='transactions', parent_column_name='product_id',
    child_dataframe_name='products', child_column_name='product_id'
)
es = es.add_relationship(rel)
```

```python
feature_matrix, features_defs = ft.dfs(entityset=es, target_dataframe_name="customers")
feature_matrix.head(5)
```

✓ 0.4s

	zip_code	COUNT(sessions)	MODE(sessions.device)	NUM_UNIQUE(sessions.device)	CO
customer_id					
5	60091	6	mobile	3	
4	60091	8	mobile	3	
1	60091	8	mobile	3	
3	13244	6	desktop	3	
2	13244	7	desktop	3	

5 rows × 75 columns

```python
feature_matrix, features_defs = ft.dfs(entityset=es, target_dataframe_name="products")
feature_matrix.head(5)
```

✓ 0.0s

	COUNT(transactions)	MAX(transactions.amount)	MEAN(transactions.amount)	MIN(transactions.amount)
product_id				
1	102	149.56	73.429314	6.84
2	92	149.95	76.319891	5.73

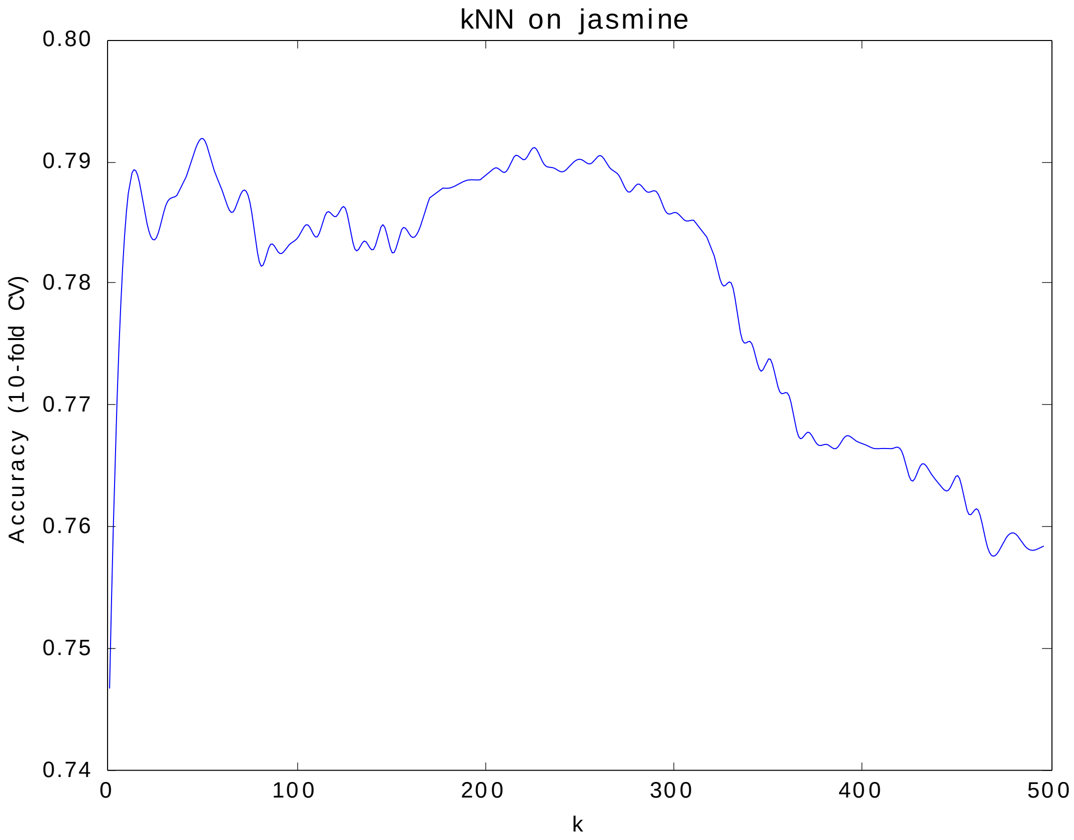
- **Documentation**: docs.featuretools.com
- **GitHub**: github.com/alteryx/featuretools
- **Key Takeaway**: Let Featuretools generate numerous features automatically, so you can focus on **modeling** and **analysis**.

Outline

- Auto Feature Selection (Lecture 5)
- Auto Hyperparameter Tuning (Part in Lecture 5)
- Auto Feature Generation (This Lecture) Neural Architecture Search (This Lecture)

Auto Hyperparameter Tuning

A Simple Example with *k*-NN



Credit to: Marius Lindauer

- *k*-nearest neighbors is one of the simplest ML algorithms
- Size of neighbourhood (*k*) is very important for its performance
- The performance function depending on *k* is quite complex (not at all convex)

Recap: Parameter Tuning and Evaluation

Evaluation

- Ground-truth Label?
- Evaluation Metric?

Parameter Tuning

- Grid Search
- Random Search
- Bayesian Optimization

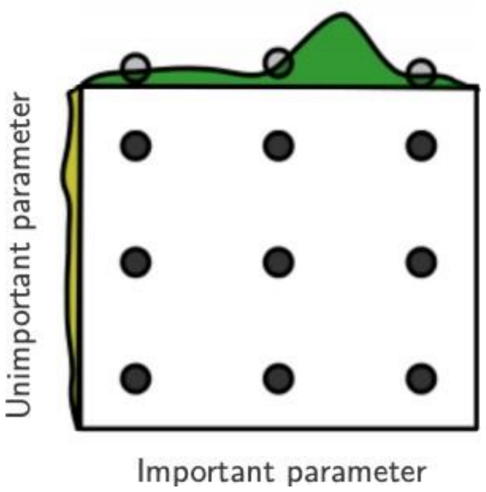
Recap: Grid Search & Random Search

x: # of working hours (1, 2, …,12)
y: # of sleeping hours (1, 2, …,12)

Income(x, y) = Work (x) + Sleep (y)

Grid Search

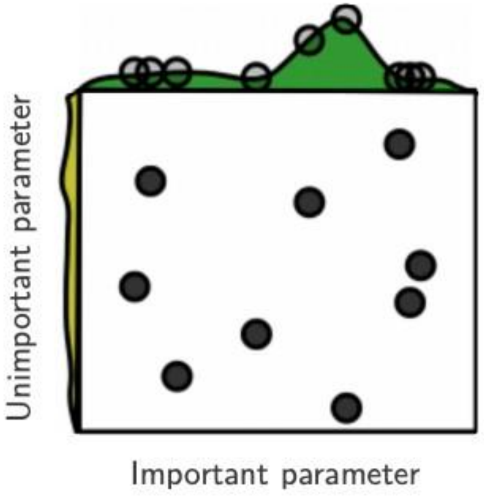
Y



X

Random Search

Y

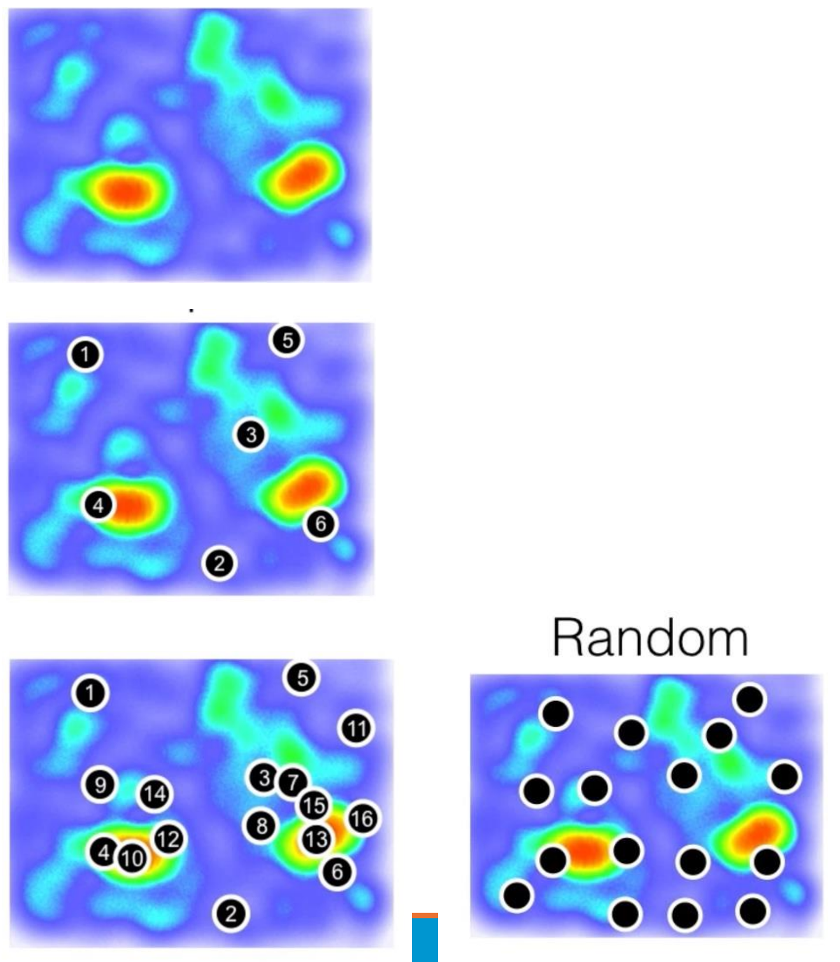


X

Bergstra, James, and Yoshua Bengio. "Random search for hyper-parameter optimization." Journal of machine learning research 13.Feb (2012): 281-305.

Recap: Bayesian Optimization

- **Intuition**
  - Want to find the peak point of objective function (eg. accuracy as a function of parameters)
  - Fit a statistical model to the observed points and pick next best point where we believe the maximum will be
  - Next point is determined by an acquisition function that trades off exploration(objective) and exploitation(uncertainty)



Hyperparameter Optimization (HPO)

Definition: Hyperparameter Optimization (HPO)

Let

- $\boldsymbol{\lambda}$ be the hyperparameters of a ML algorithm A with domain $\boldsymbol{\Lambda}$,
- $\mathcal{L}(A_{\boldsymbol{\lambda}}, D_{train}, D_{valid})$ denote the loss of A, using hyperparameters $\boldsymbol{\lambda}$ trained on D_{train} and evaluated on D_{valid}.

The hyperparameter optimization (HPO) problem is to find a hyperparameter configuration $\boldsymbol{\lambda}^*$ that minimizes this loss:

$$\boldsymbol{\lambda}^* \in \arg\min_{\boldsymbol{\lambda} \in \boldsymbol{\Lambda}} \mathcal{L}(A_{\boldsymbol{\lambda}}, D_{train}, D_{valid})$$

Hutter & Vanschoren: AutoML, Neurips' 18 tutorial

Hyperparameter Gradient Descent

1. Formulation as bilevel optimization problem

$$\min_\lambda \; \mathcal{L}_{val}\left(w^*(\lambda), \lambda\right)$$
$$s.t. \quad w^*(\lambda) \; = \; \operatorname{argmin}_w \mathcal{L}_{train}\left(w, \lambda\right)$$

2. Interleave optimization steps

Hyperparameter gradient step w.r.t. $\nabla_\lambda \mathcal{L}_{val}$

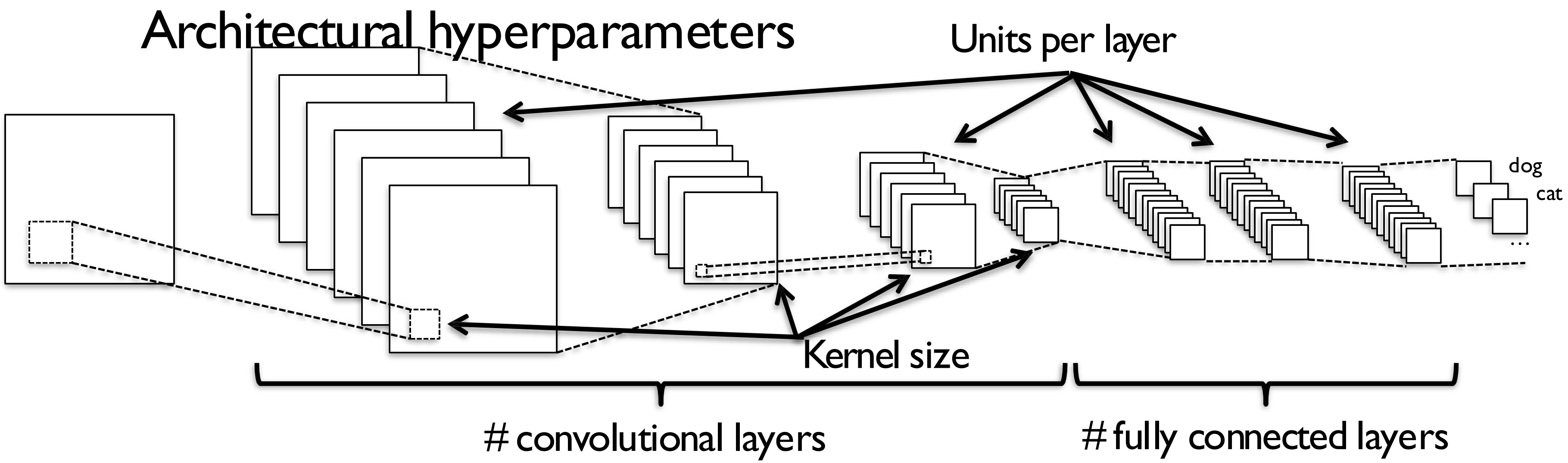
Parameter gradient step w.r.t. $\nabla_w \mathcal{L}_{train}$

Hutter & Vanschoren: AutoML, Neurips'18 tutorial

Neural Architecture Search (NAS)

Challenge in Deep Learning

Performance is very sensitive to many hyperparameters



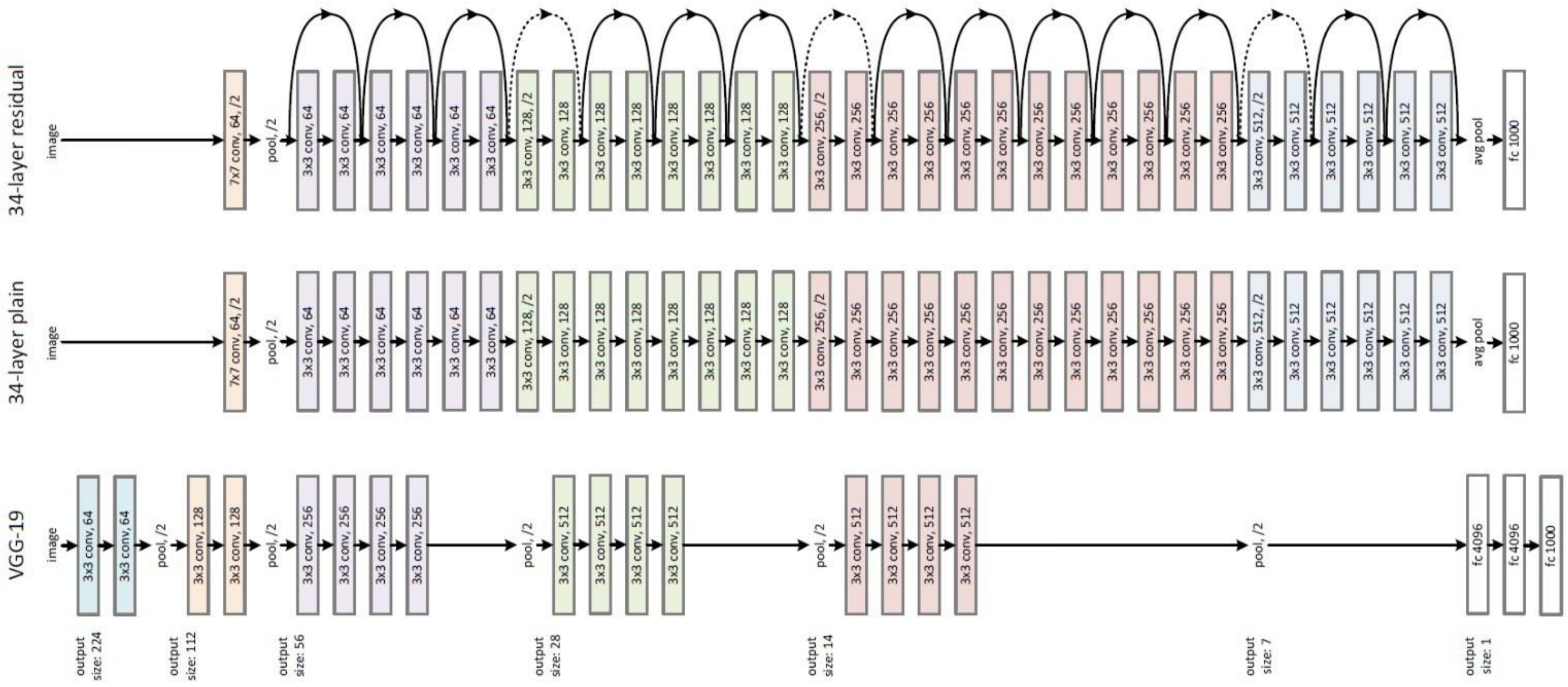
Optimization algorithm, learning rates, momentum, batch normalization, batch sizes, dropout rates, weight decay, data augmentation, …

Easily 20-50 design decisions

Hutter & Vanschoren: AutoML, Neurips'18 tutorial

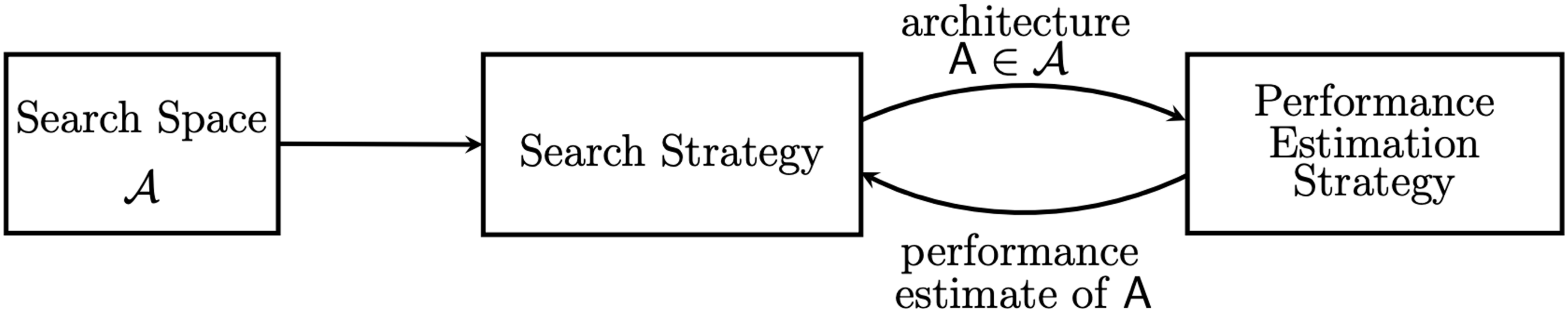
Motivation

How can someone come out with such an architecture?



He, Kaiming, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. "Deep residual learning for image recognition." CVPR 2016

Neural Architecture Search: Big Picture



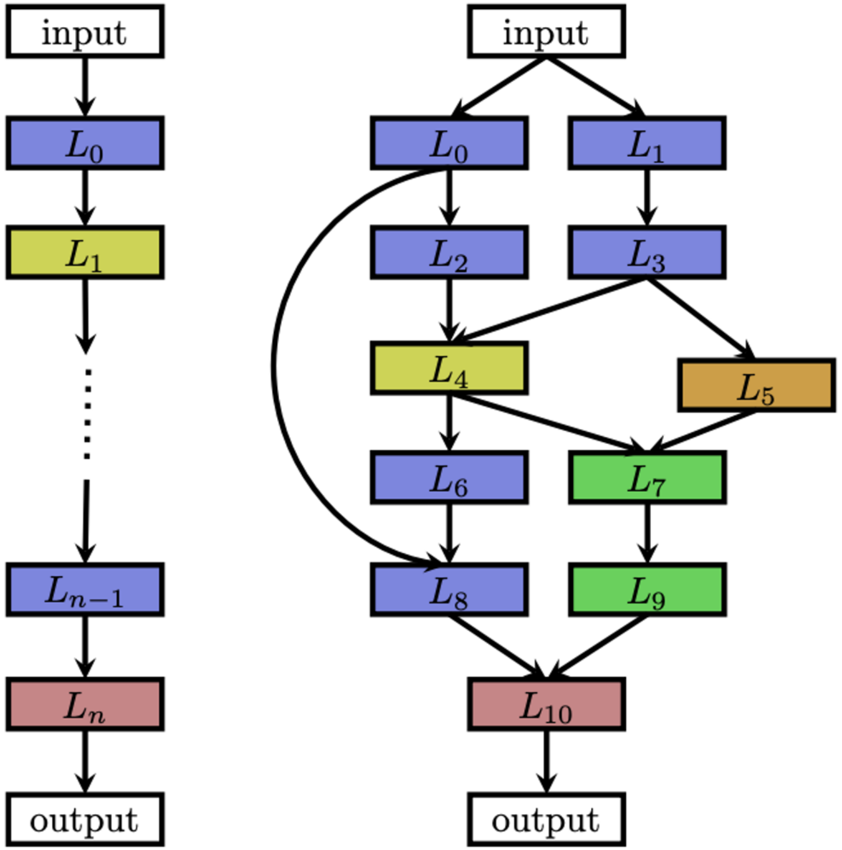
NAS is also Hyperparameter Optimization

Key Challenges:

1. **Vast Search Space**
    - The number of possible neural architectures is huge.
2. **Computational Cost**
    - Training each candidate network is expensive (time, GPU resources).
3. **Overfitting to Benchmark Datasets**
    - Risk of overly specialized solutions.

Search Space

- Define which neural architectures a NAS approach might discover in principle
- May have human bias → prevent finding novel architectural building blocks



Chain-structured Multi-branch

Search Strategy

- Basic Idea
  - Explore search space (often exponentially large or even unbounded)
- Methods
  - Random Search
  - Bayesian Optimization [Bergstra et al., 2013]
  - Evolutionary Methods [Angeline et al., 1994]
  - Reinforcement Learning [Baker et al., 2017]
  - .....

Example: Differentiable Architecture Search (DARTS)

Continuous relaxation of the search space to enable gradient-based optimization.

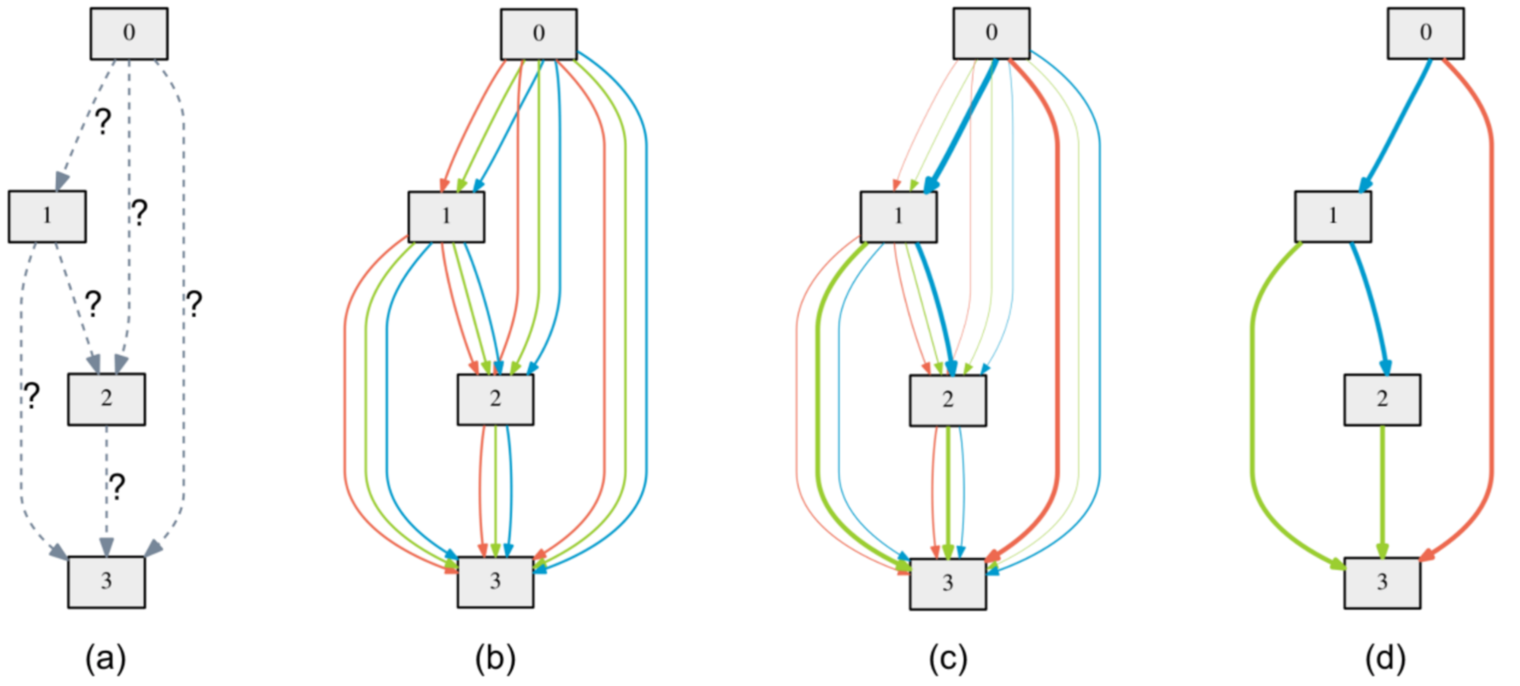
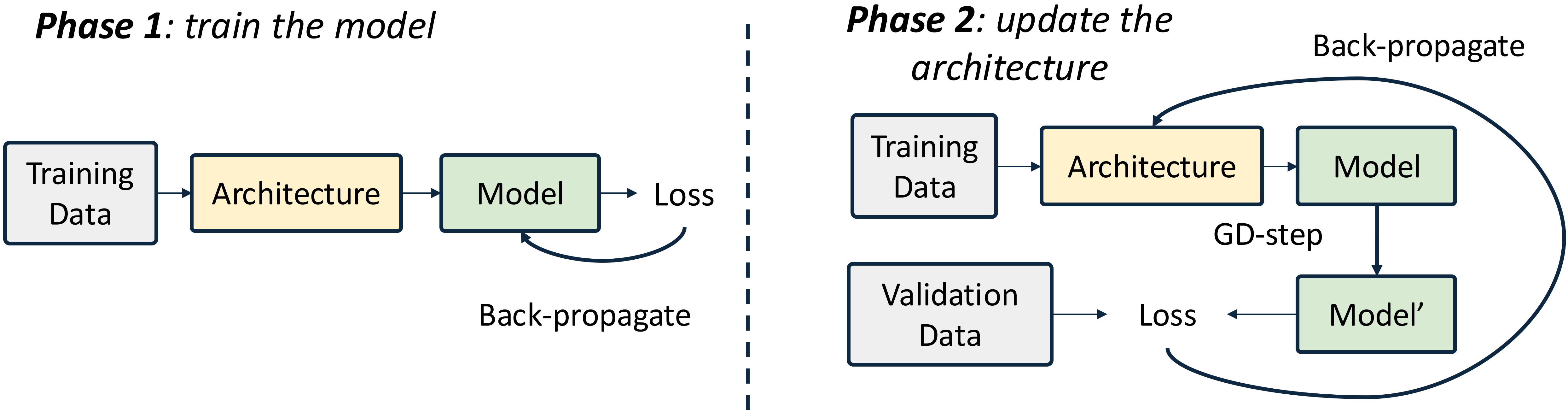


Figure 1: An overview of DARTS: (a) Operations on the edges are initially unknown. (b) Continuous relaxation of the search space by placing a mixture of candidate operations on each edge. (c) Joint optimization of the mixing probabilities and the network weights by solving a bilevel optimization problem. (d) Inducing the final architecture from the learned mixing probabilities.

High level idea of DARTS

- Bi-level optimization



Update the architecture such that the target model performs well on the val. set

Performance Estimation Strategy

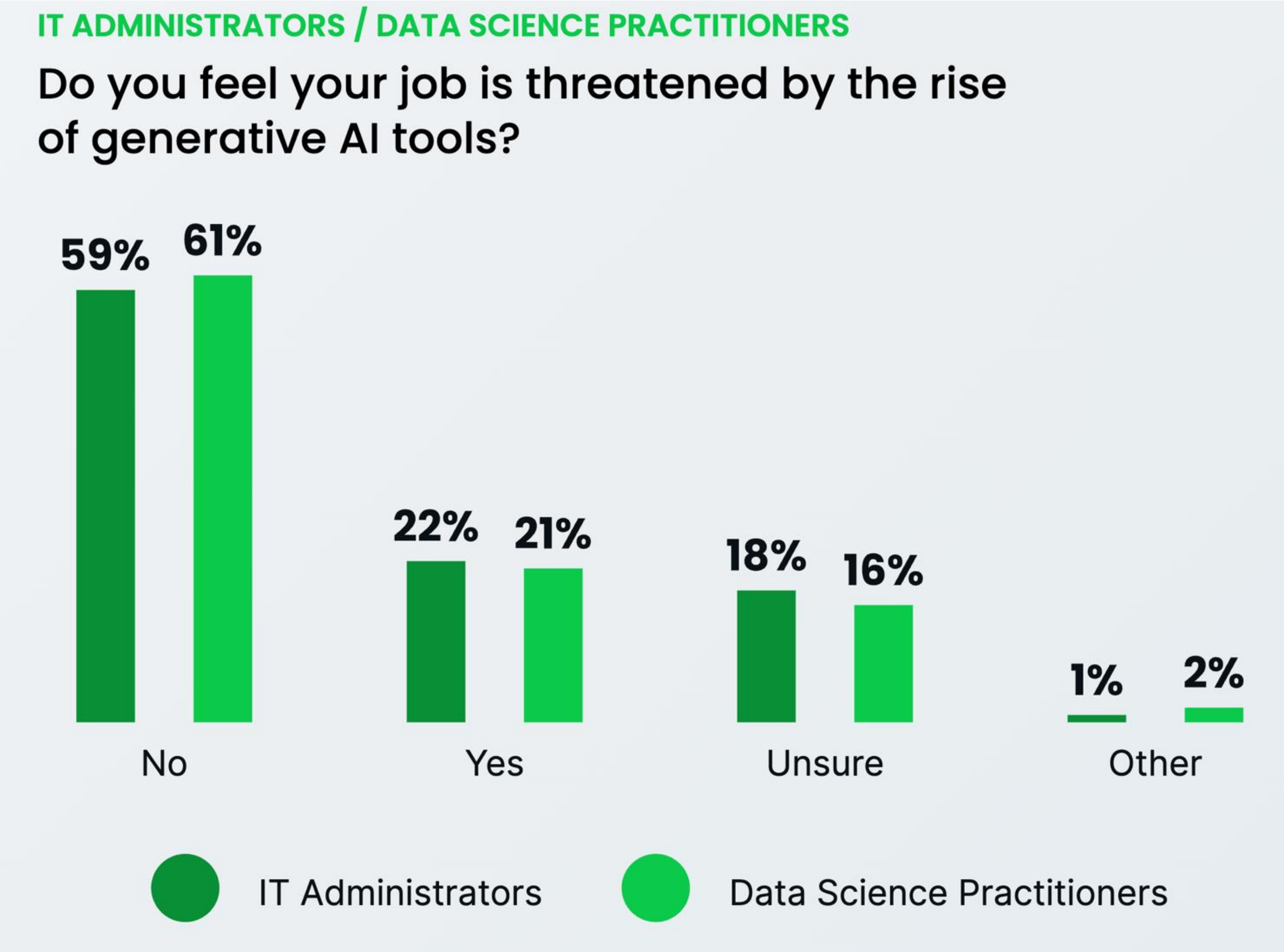
- Basic Idea
  - The process of estimating predictive performance
- Methods
  - Simplest option: perform a training and validation of the architecture on data
  - Initialize weights of novel architecture based on weights of other architectures have been trained before
  - Using learning curve extrapolation [Swersky et al., 2014]
  - ……

H20 Driverless AI Demo

1. Will AutoML software replace Data Scientists?
2. How to approach AutoML as a data scientist?

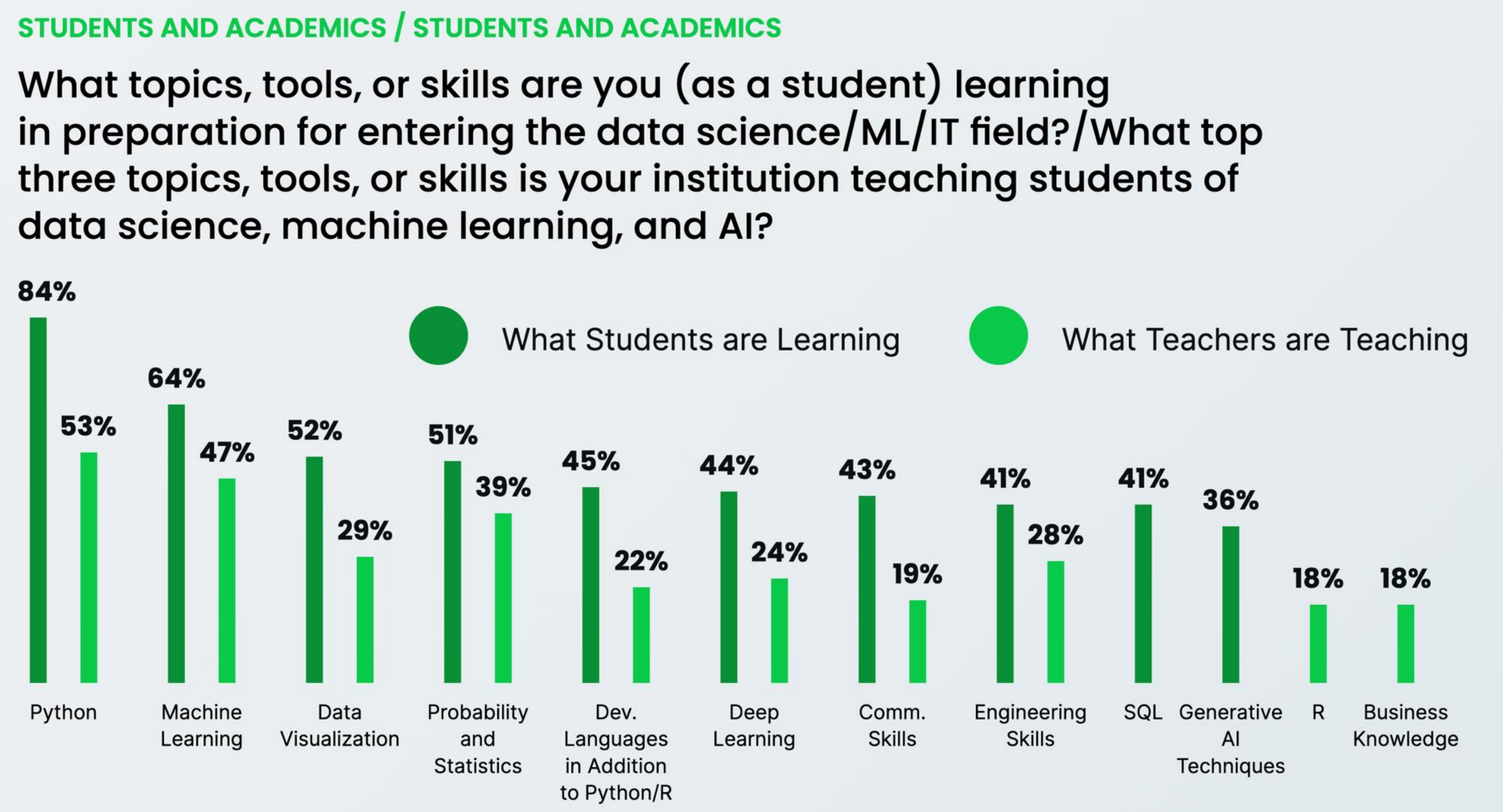
https://www.youtube.com/watch?v=ZqCoFp3-rGc

Trend in 2024



https://www.anaconda.com/resources/report/state-of-data-science-report-2024

Trend in 2024



Summary

What is AutoML and why we need it? How AutoML works?

- **Auto Feature Selection (Lecture 5)**
- **Auto Hyperparameter Tuning (Lecture 5 and this Lecture)**
- **Auto Feature Generation (This Lecture)**
- **Neural Architecture Search (This Lecture)**

CMPT 733
Explainable Machine Learning

Instructor Zhengjie Miao

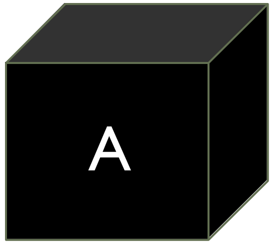
Course website https://coursys.sfu.ca/2025sp-cmpt-733-g1/pages/

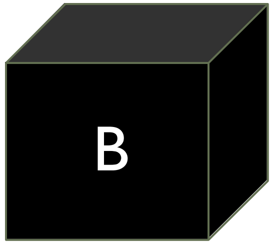
Based on the slides by: Xiaoying Wang and Jiannan Wang

Outline

- Motivation: Why Explainable ML matters?
- Big Picture: Taxonomy
- State-of-the-art Techniques

Evaluation





Bird: 99.0%

Bird: 99.9%

Which model are you going to choose?

Evaluation

A

Because it has wings and a beak

Bird: 99.0%

B

Because it is white and the background is blue

Bird: 99.9%

Which model are you going to choose?

Debugging

What's wrong?

Q: How symmetrical are the white bricks on either side of the building? A: very

Q: How asymmetrical are the white bricks on either side of the building? A: very

Q: How fast are the bricks speaking on either side of the building? A: very

MUDRAKARTA, P.K., TALY, A., SUNDARARAJAN, M. AND DHAMDHERE, K., 2018. DID THE MODEL UNDERSTAND THE QUESTION?. ARXIV PREPRINT ARXIV:1805.05492.

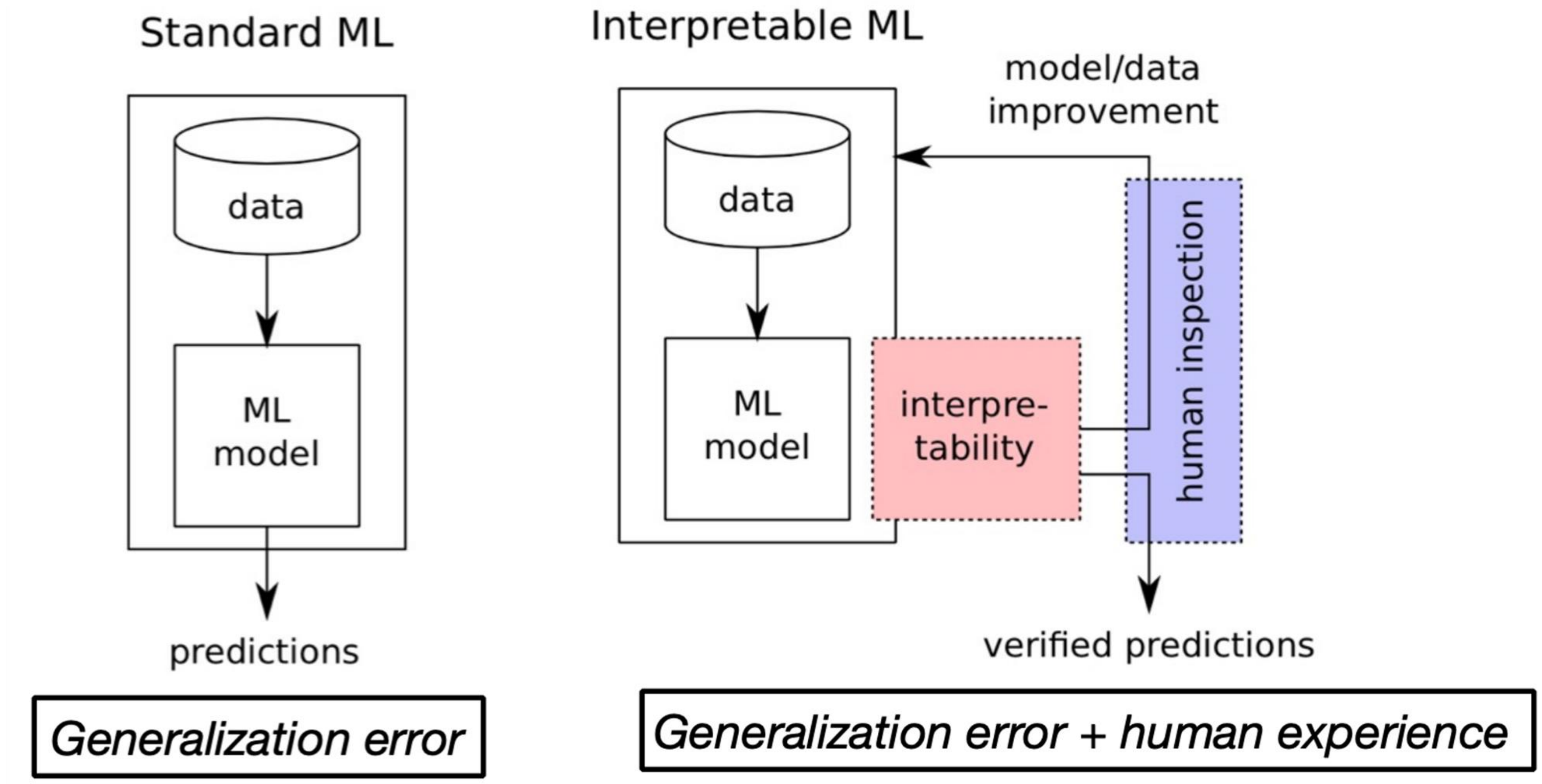
Debugging

How symmetrical are the white bricks on either side of the building?

red: high attribution
blue: negative attribution
gray: near-zero attribution

MUDRAKARTA, P.K., TALY, A., SUNDARARAJAN, M. AND DHAMDHERE, K., 2018. DID THE MODEL UNDERSTAND THE QUESTION?. ARXIV PREPRINT ARXIV:1805.05492.

Improvement



ANON. ICCV'19 TUTORIAL ON INTERPRETABLE MACHINE LEARNING IN COMPUTER VISION

Legal Concerns

SR 11-7: Guidance on Model Risk Management

BOARD OF GOVERNORS
OF THE FEDERAL RESERVE SYSTEM
WASHINGTON, D.C. 20551

DIVISION OF BANKING
SUPERVISION AND REGULATION

SR 11-7
April 4, 2011

TO THE OFFICER IN CHARGE OF SUPERVISION AND APPROPRIATE SUPERVISORY AND EXAMINATION STAFF AT EACH FEDERAL RESERVE BANK

SUBJECT: Guidance on Model Risk Management

Art. 22 GDPR

Automated individual decision-making, including profiling

Outline

Motivation: Why Explainable ML matters?

Big Picture: Taxonomy

State-of-the-art Techniques

Taxonomy

- **Transparent Models**: Linear Regression, Decision Tree, KNN, Bayesian Network…
- **Post-hoc Explanation**
  - **Global Model Explanation**: Permutations, Partial Dependence Plots, Global Surrogate …
  - **Individual Prediction Explanation**: Attribution, Influential Instances, Local Surrogate …

Taxonomy

Transparent Models

Linear Regression, Decision Tree, KNN, Bayesian Network…

Post-hoc Explanation

Global Model Explanation

Permutations, Partial Dependence plots, Global Surrogate …

Individual Prediction Explanation

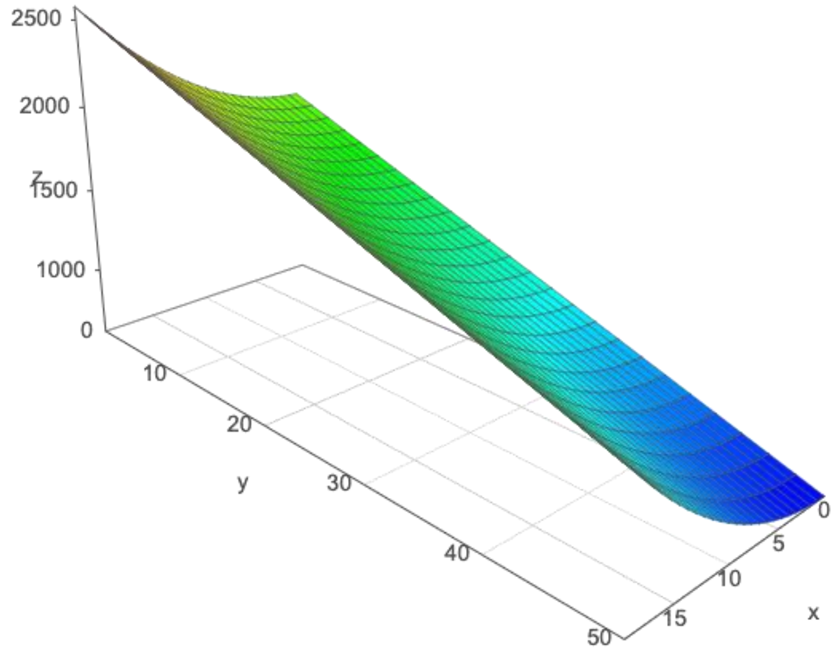
Attribution, Influential Instances, Local Surrogate …

Linear Regression

House rent (z) with respect to its area (x) and distance from SFU (y)

$$z = 2.1x - 2.4y + 1800$$

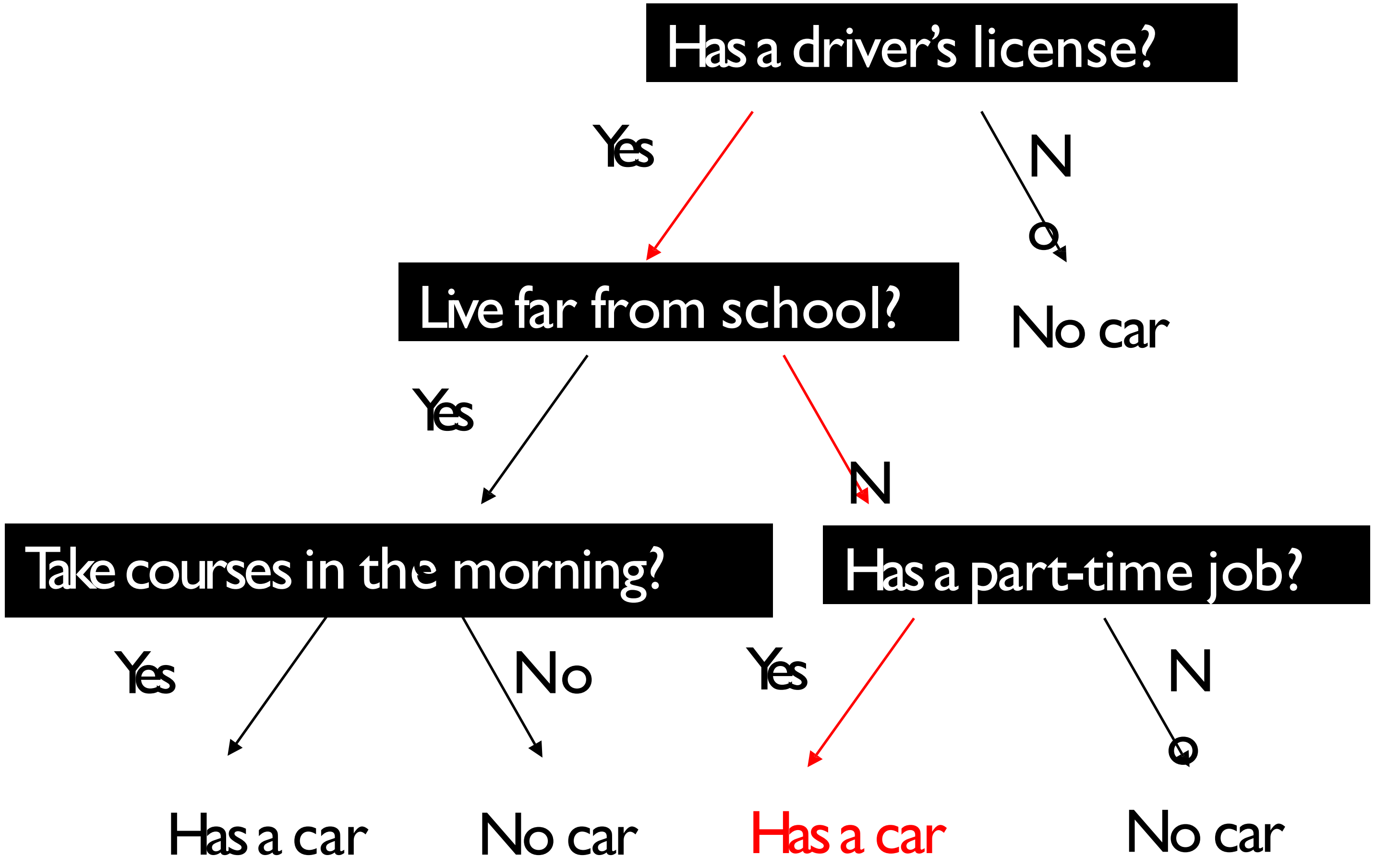
How do area and distance affect the house rent?



Decision Tree

Does a student own a car?

Why does the model predict student A has a car ?



Taxonomy

- Transparent Models
  - Linear Regression, Decision Tree, KNN, Bayesian Network…
- Post-hoc Explanation
  - Global Model Explanation
    - Permutations, Partial Dependence Plots, Global Surrogate …
  - Individual Prediction Explanation
    - Attribution, Influential Instances, Local Surrogate …

Permutations

Main idea: measure the importance of a feature by calculating the increase in the model's prediction error after permuting the feature

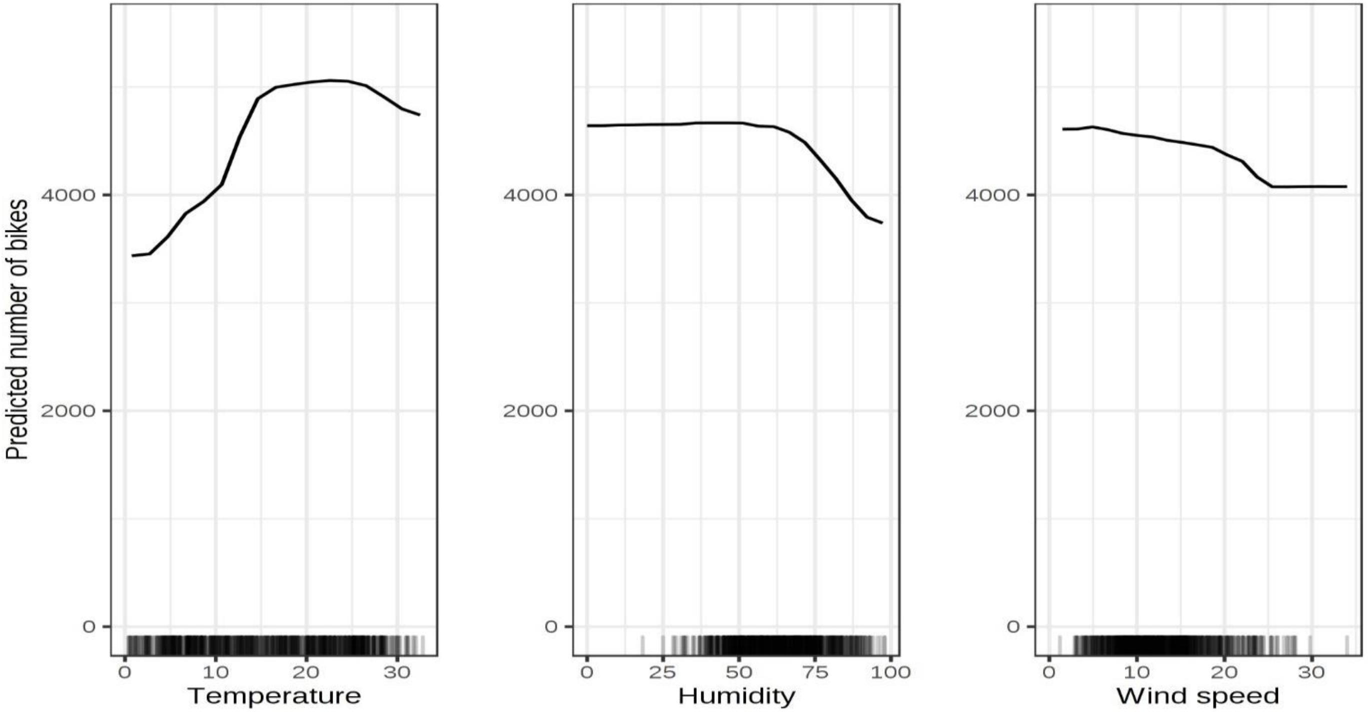
ID	Distance to SFU	# Bathroom	Area	Distance to Bus Stop	…
1	5.0 km	1	670 ft2	300 m	
2	8.2 km	2	970 ft2	120 m	
3	2.3 km	2	880 ft2	1200 m	
…			…		
9999	10.0 km	1	680 ft2	50 m	
10000	7.8 km	1	730 ft2	230 m	

Permutations

- Input: trained model and labeled dataset for evaluation
- Output: relative importance for each feature
- Method:
  - Apply the model on original dataset and get an estimation error E
  - For each feature:
    - Permute feature and apply the model again on the permuted data to get a new estimation error E'
    - The feature importance can be measured by E'-E or E'/E

Partial Dependence Plots

Main idea: show the marginal effect one or two features have on the predicted outcome of a machine learning model



ID	Temperature	Humidity	Wind Speed	Rental#
1	20	30	20	3000
2	25	35	10	2500
3	22	25	15	3300
4	30	20	18	2000
..	..	…	…	…

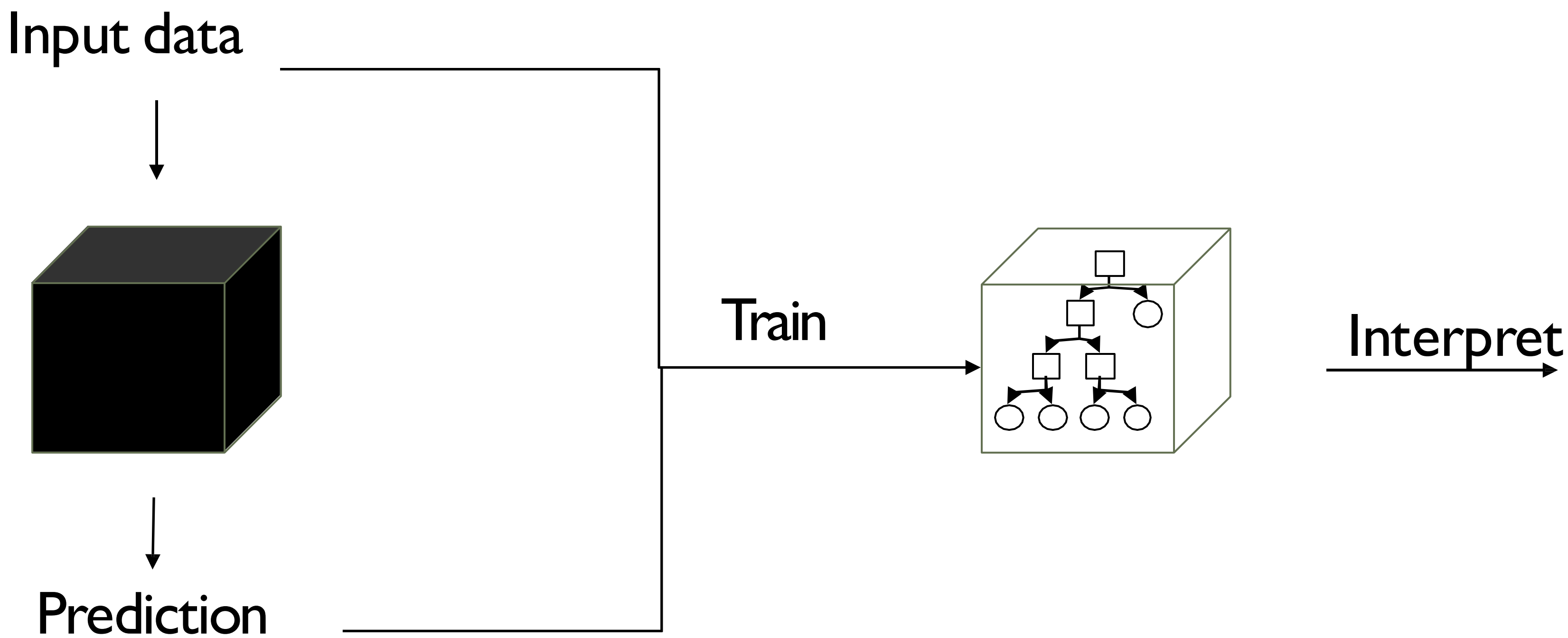
Partial Dependence Plots

Let x_s be the feature set ($|x_s| \in \{1,2\}$) we want to examine, and x_c be the rest of the features used in the model $\hat{f}$:

- Partial dependence function: $\hat{f}_{x_s}(x_c) = E_{x_c}\left[\hat{f}(x_s, x_c)\right] = \int \hat{f}(x_s, x_c)dP(x_c)$
- Can be estimated by: $\hat{f}_{x_s}(x_c) = \frac{1}{n}\sum_{i=1}^{n}\left(x_s, x_c^{(i)}\right)$

Global Surrogate

Main idea: train a transparent model to approximate the predictions of a black box model



Global Surrogate

$\hat{y}^{(i)}$ and $\hat{y}_*^{(i)}$: the target model and surrogate model's prediction for the ith input data

R-squared: measure how good the surrogate model is in approximating the target model

$$R^2 = 1 - \frac{\sum_{i=1}^{n}\left(\hat{y}_*^{(i)} - \hat{y}^{(i)}\right)^2}{\sum_{i=1}^{n}\left(\hat{y}^{(i)} - \hat{y}_{avg}\right)^2}$$

Taxonomy

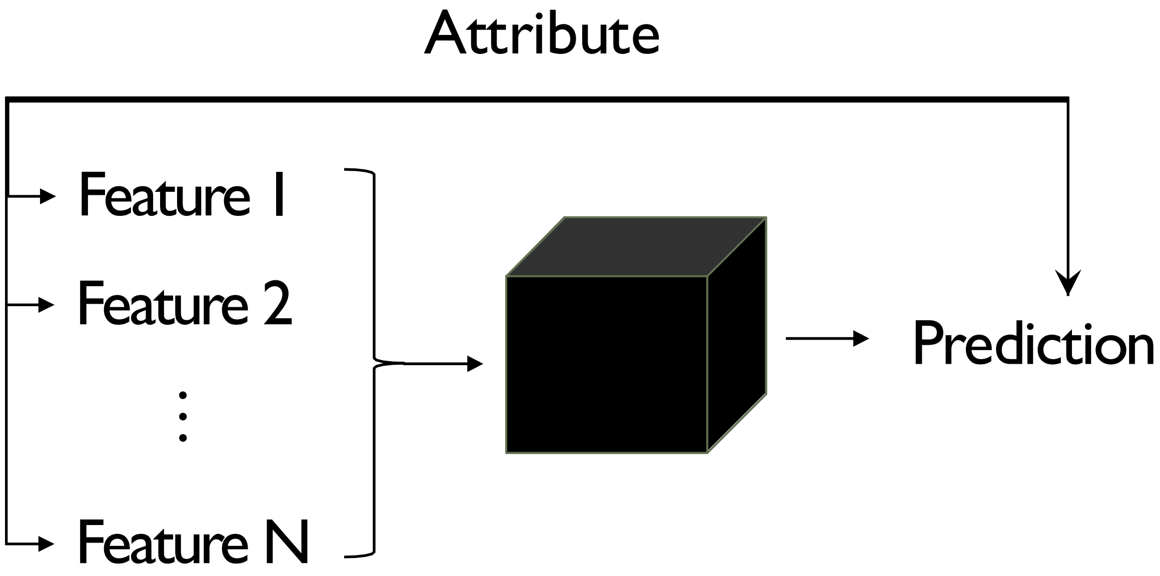
Transparent Models — Linear regression, Decision tree, KNN, Bayesian Network…

Post-hoc Explanation

- Global Model Explanation — Permutations, Partial dependence plots, Global surrogate …
- Individual Prediction Explanation — Attribution, Influential instances, Local surrogate …

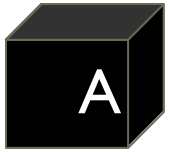
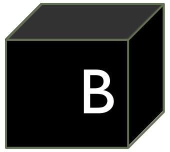
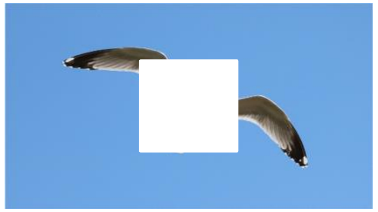
Attribution

- Main idea:
  - Attribute a model's prediction on a sample to its input features
- Approaches
  - Ablation
  - Shapely value
  - …



Attribution (Ablation)

Ablation: drop each feature and attribute the change in prediction to the feature

		
	Bird (99%)	Bird (99%)
	Bird (20%)	Bird (98%)
	Bird (96%)	Bird (35%)

Attribution (Ablation)

- Saliency maps: compute the relative importance of each feature
  - if change/remove this feature, how much is the prediction affected?

Sentiment: an intelligent fiction about learning through cultural clash.

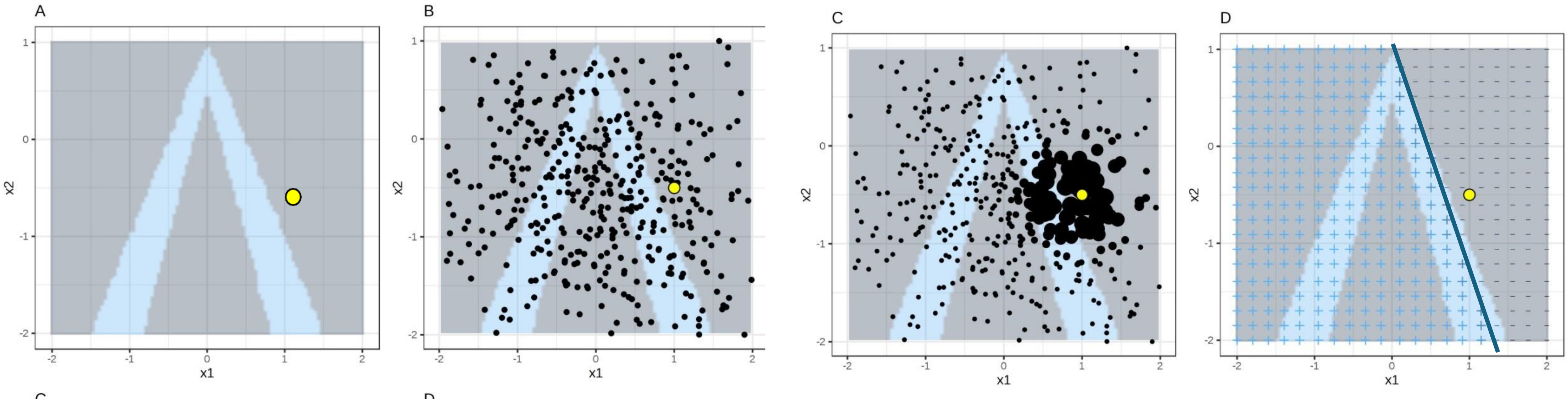
QA: What company won free advertisement due to QuickBooks contest ?

MLM: [CLS] The [MASK] ran to the emergency room to see her patient . [SEP]

[Wallace et al. 2020]

Local Surrogate (LIME)

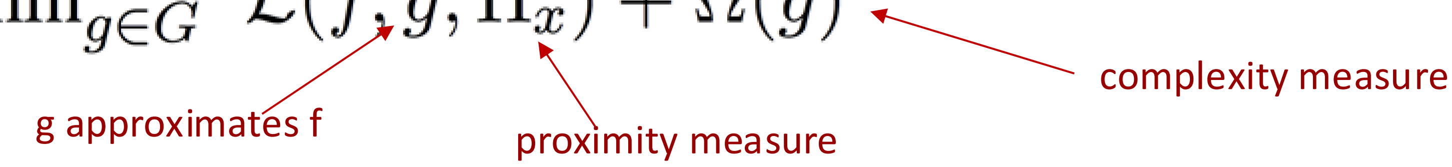
Main idea: Test what happens to the prediction when give variations of data into the machine learning model

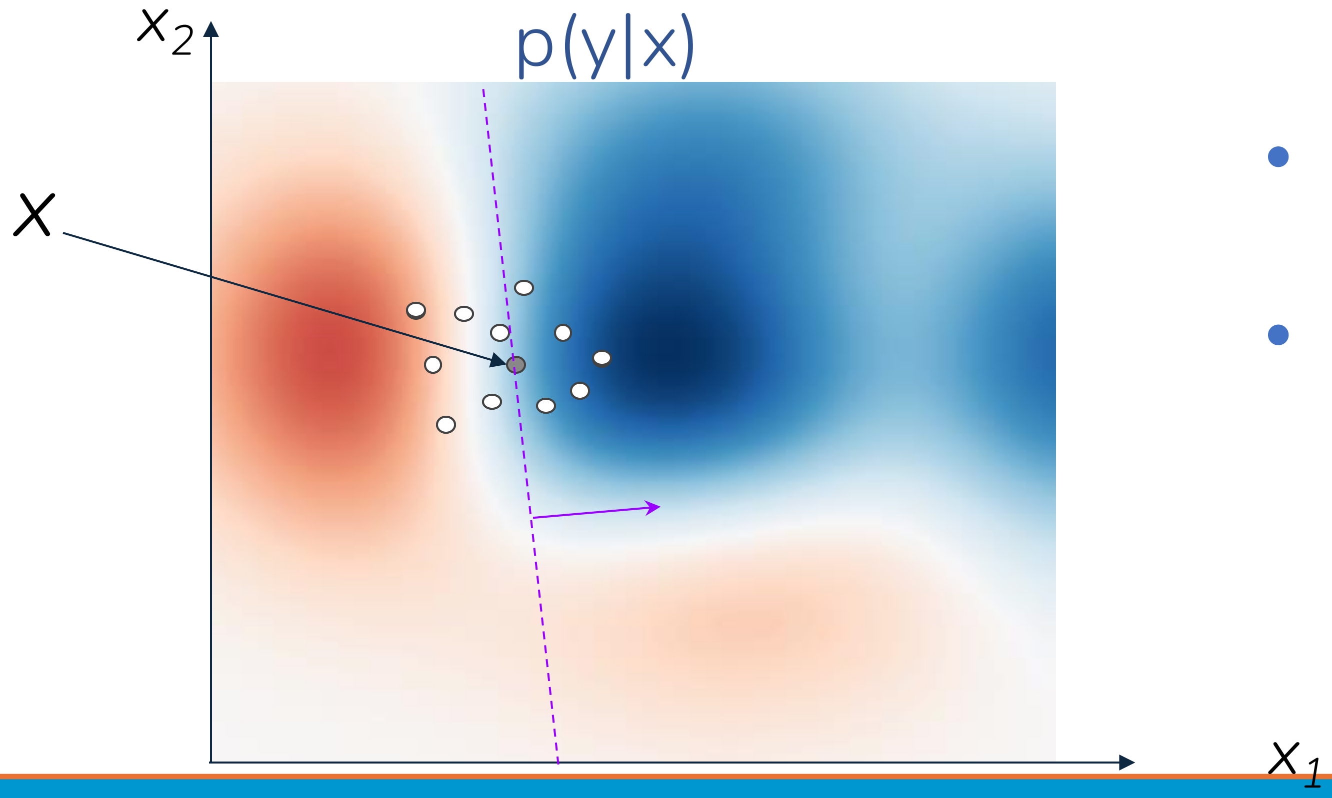


[Ribeiro et al. 2016]

LIME: Local Interpretable Model-Agnostic Explanations

$$\xi(x) = \text{argmin}_{g \in G} \; \mathcal{L}(f, g, \Pi_x) + \Omega(g)$$





- Pick a model class interpretable by humans, e.g., linear regression
- Locally approximate global (blackbox) model
  - Simple model globally bad, but locally good

[Ribeiro et al. 2016]

LIME Sentiment Analysis Example

The movie is mediocre, maybe even bad.	**Negative** 99.8%
The movie is mediocre, maybe even ~~bad~~.	**Negative** 98.0%
The movie is ~~mediocre~~, maybe even bad.	**Negative** 98.7%
The movie is ~~mediocre~~, maybe even ~~bad~~.	**Positive** 63.4%
The movie is ~~mediocre~~, ~~maybe~~ even ~~bad~~.	**Positive** 74.5%
The ~~movie~~ is mediocre, maybe even ~~bad~~.	**Negative** 97.9%

The movie is mediocre, maybe even bad.

Shapley Value

- Classic result in game theory on distributing the total gain from a cooperative game
- Introduced by Lloyd Shapley in 1953 , who won the Nobel Prize in Economics in 2012
- Popular tool in studying cost-sharing, market analytics, voting power, and most recently explaining ML models

Lloyd Shapley in 1980

"A Value for n-person Games". Contributions to the Theory of Games 2.28 (1953): 307-317

Attribution (Shapely Value)

- Shapely value: derive from game theory on distributing gain in a coalition game
- Coalition game: players collaborating to generate some gain, function $val(S)$ represents the gain for any subset S of players
    - Game: prediction task
    - Players: input features
    - Gain: marginalized actual prediction minus average prediction $val_x(S)$ =

$$\int \hat{f}(x_1, x_2, \ldots, x_p) dP_{x \notin S} - E(\hat{f}(X))$$

- Marginal contribution of a feature i to a subset of other features: $val_x(S \cup \{x_i\}) - val_x(S)$

Attribution (Shapely Value)

- Shapely value of a feature i on sample x: weighted aggregation of its marginal contribution over all possible combinations of subsets of other features

$$\sum_{S \subseteq \{x_1, x_2, \ldots, x_p\} \setminus \{x_i\}} \frac{|S|!\,(p - |S| - 1)!}{p!} (val_x(S \cup \{x_i\}) - val_x(S))$$

- Intuition: The feature values enter a room in random order. All feature values in the room participate in the game (= contribute to the prediction). The Shapley value of a feature value is the average change in the prediction that the coalition already in the room receives when the feature value joins them.

Example

- A company with two employees Alice and Bob
  - No employees, 0 profit
  - Alice alone makes 20 units of profit
  - Bob alone makes 10 units of profit
  - Alice and Bob make total 50 units of of profit
- What should be the bonuses be?

All Possible Orders	Marginal for Alice	Marginal for Bob
Alice, Bob		
Bob,Alice		
Shapley Value		

Example

- A company with two employees Alice and Bob
  - No employees, 0 profit
  - Alice alone makes 20 units of profit
  - Bob alone makes 10 units of profit
  - Alice and Bob make total 50 units of of profit
- What should be the bonuses be?

All Possible Orders	Marginal for Alice	Marginal for Bob
Alice, Bob	20	30
Bob,Alice	40	10
Shapley Value	30	20

Attribution (Shapely Value)

- Two challenges when computing shapely value:
  - Exponential time since the permutation
  - Cannot inference on models when some features are not provided
- SHAP (SHapley Additive exPlanations) provide solutions for these two challenges:
  - KernelSHAP: an approximation solution for all models:
    - Sample a subset of feature orders
    - Filling missing features with background dataset provided by user

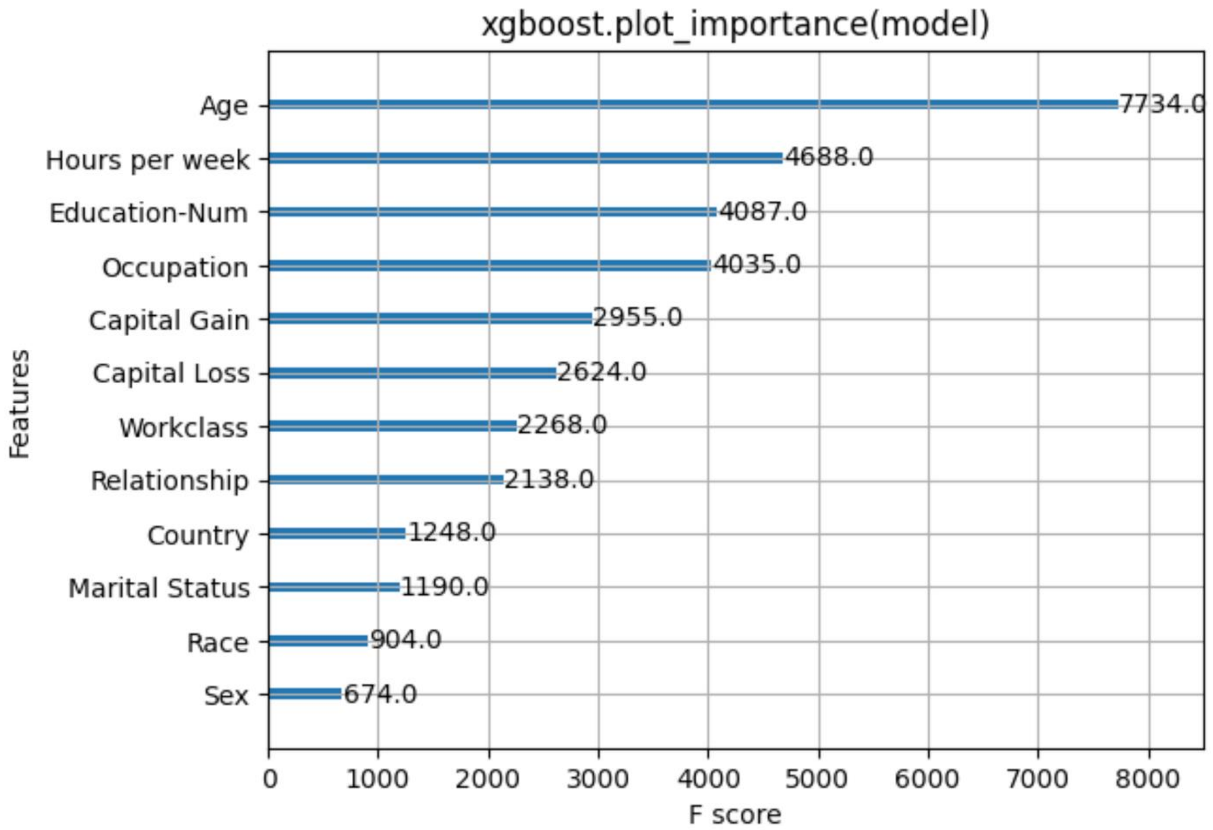
LUNDBERG, SCOTT M., AND SU-IN LEE. "A UNIFIED APPROACH TO INTERPRETING MODEL PREDICTIONS." ADVANCES IN NEURAL INFORMATION PROCESSING SYSTEMS. 2017.

SHAP Example

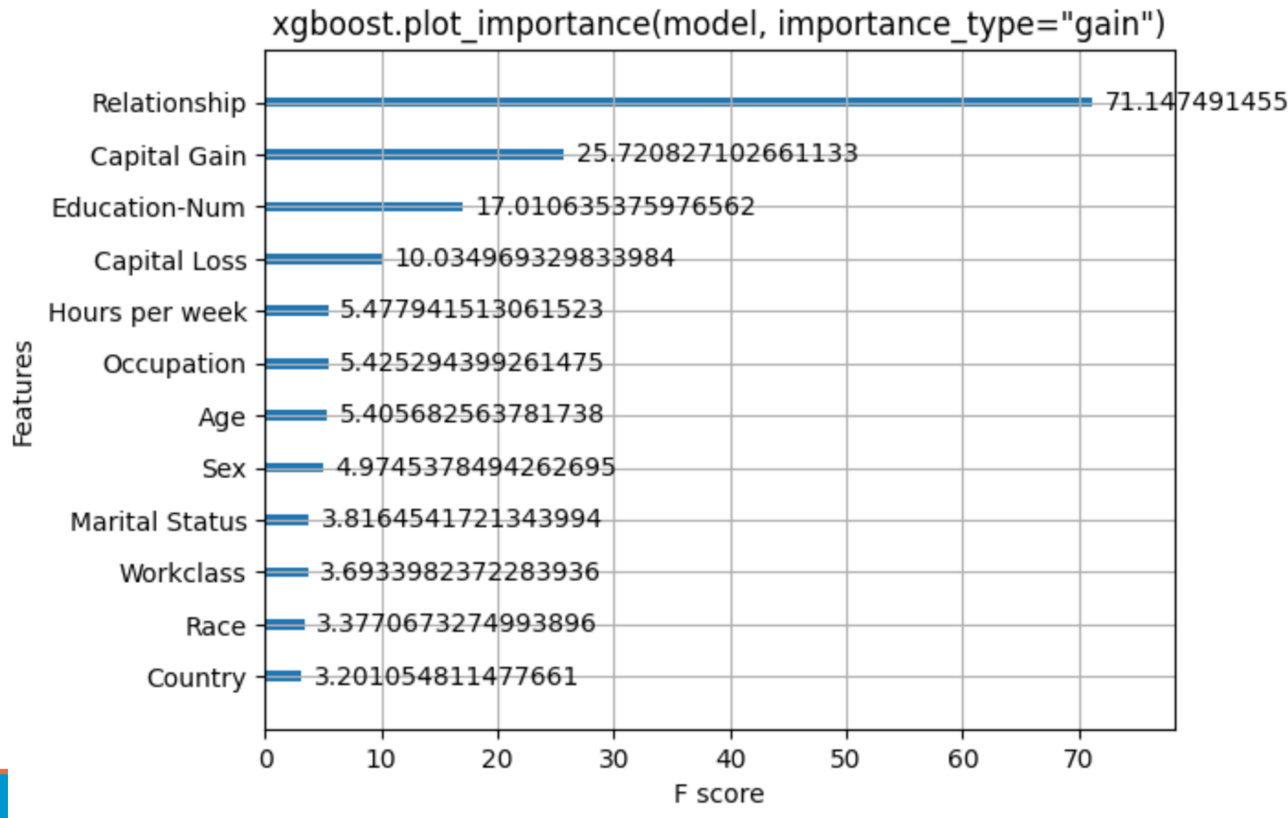
https://github.com/shap/shap/blob/master/notebooks/tabular_examples/tree_based_models/Census%20income%20classification%20with%20XGBoost.ipynb

XGBoost on Adult dataset, classic feature attribution

```
xgboost.plot_importance(model)
pl.title("xgboost.plot_importance(model)")
pl.show()
```

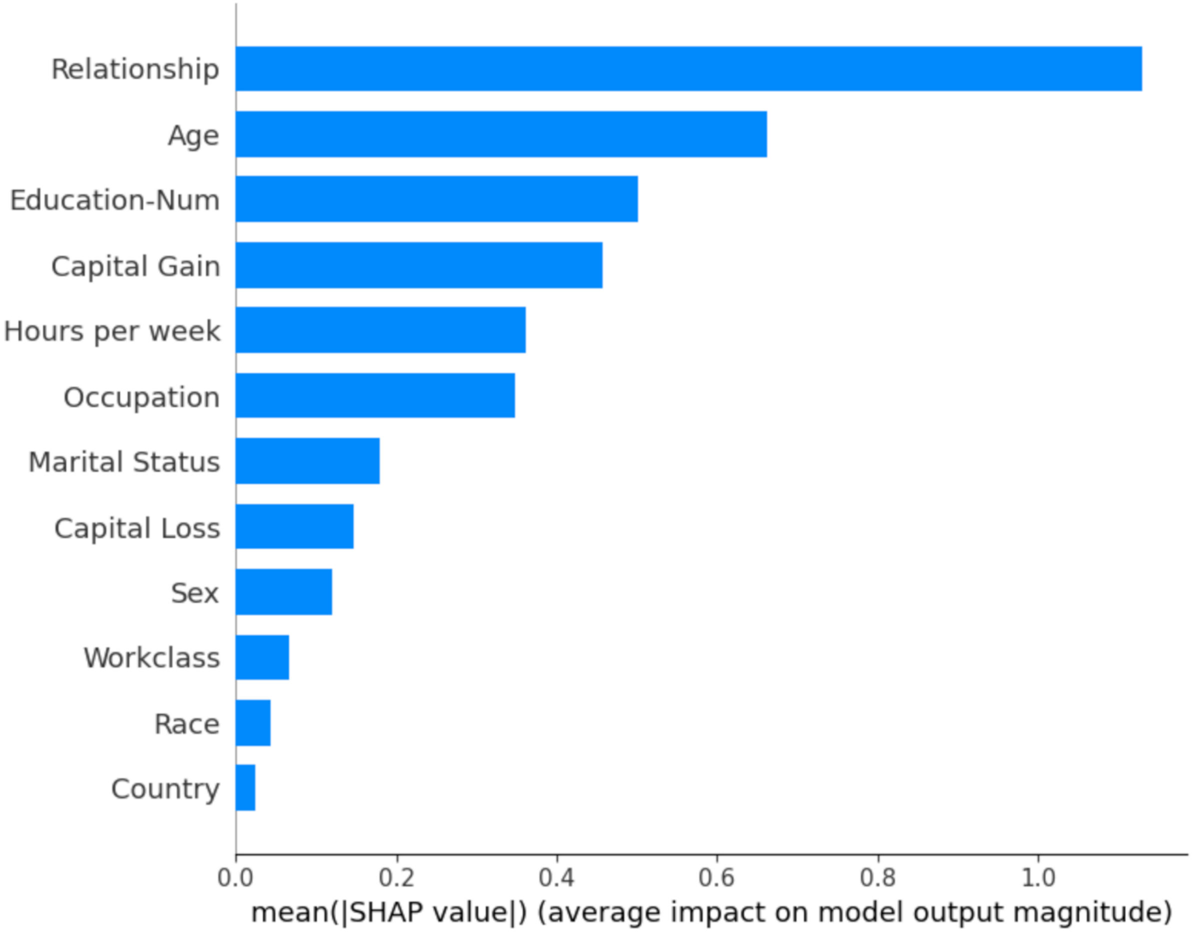



```
xgboost.plot_importance(model, importance_type="gain")
pl.title('xgboost.plot_importance(model, importance_type="gain")')
pl.show()
```



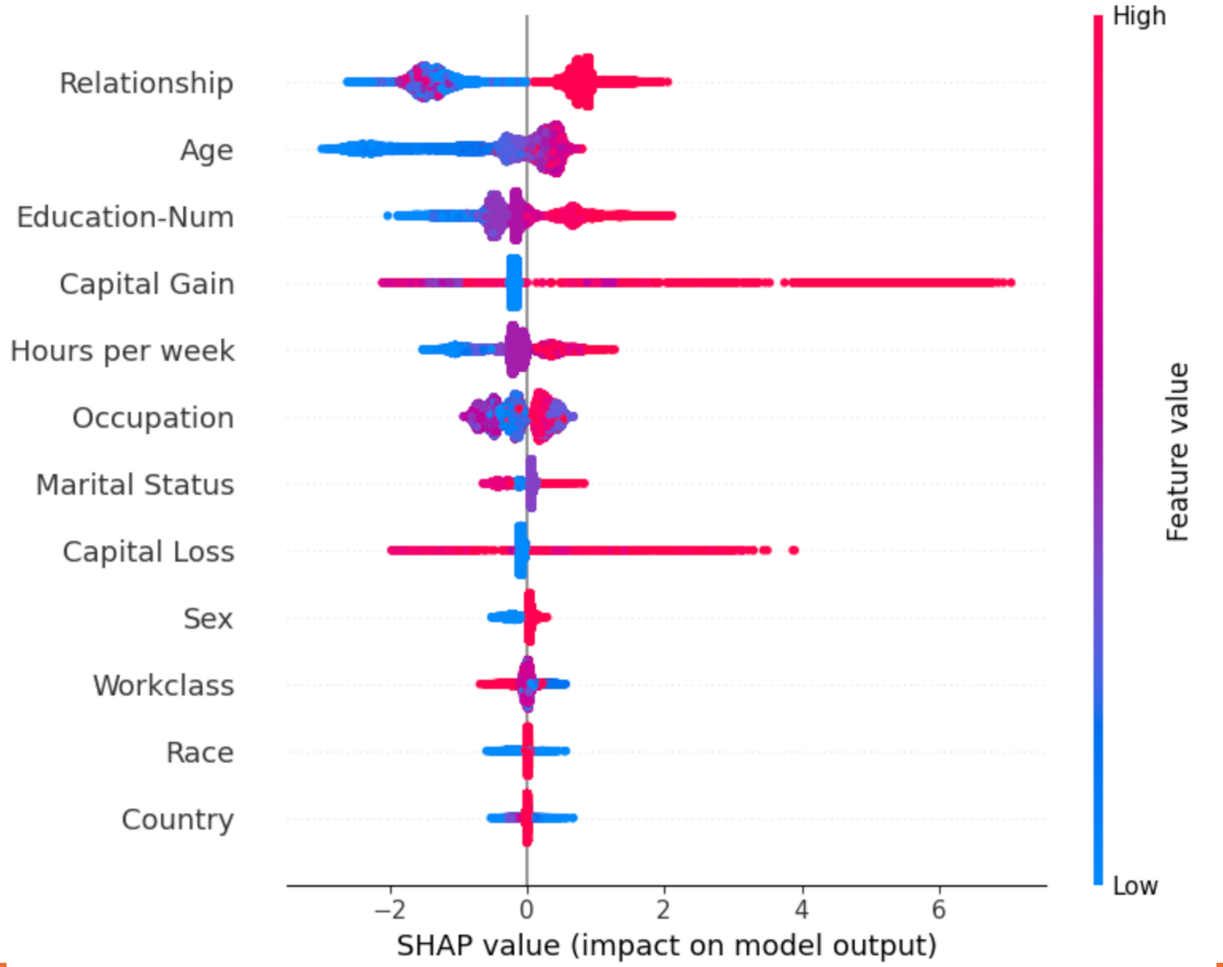
Bar chart of mean importance

```
shap.summary_plot(shap_values, X_display, plot_type="bar")
```



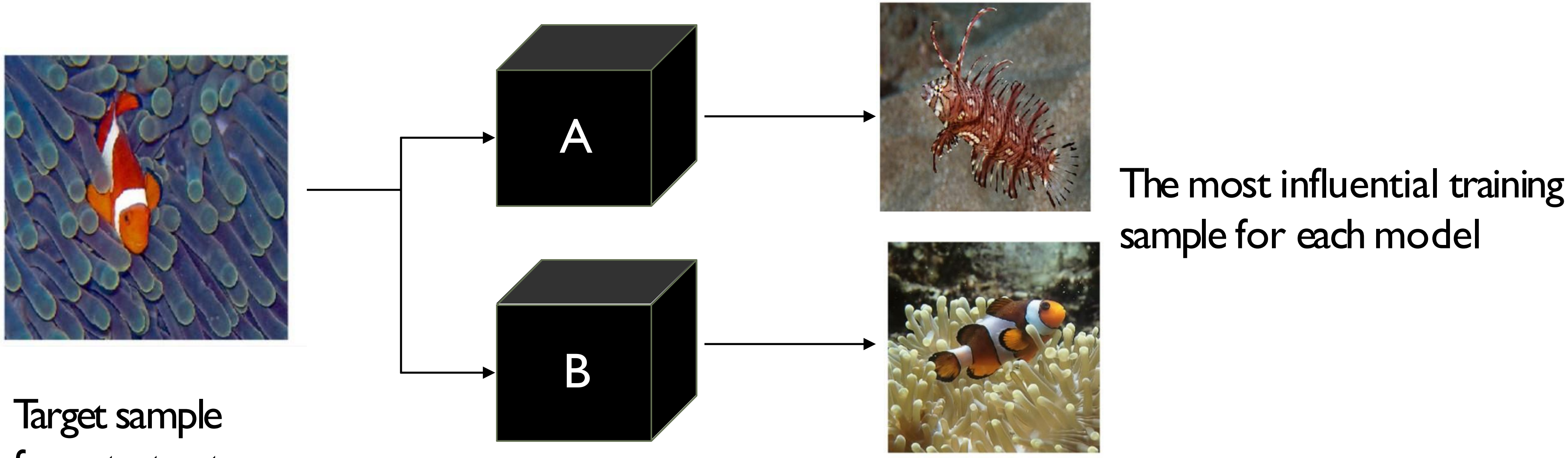
SHAP Summary Plot

```
shap.summary_plot(shap_values, X)
```



Influential Instances

Main idea: debug machine learning model by identifying influential training instances (a training instance is influential when its deletion from training data considerably changes the model's prediction)



Target sample from test set

KOH, PANG WEI, AND PERCY LIANG. "UNDERSTANDING BLACK-BOX PREDICTIONS VIA INFLUENCE FUNCTIONS." ICML'17

Influential Instances

- Naïve approach: deletion diagnostics
  - Train a model on all data instances, predict on test data and choose a target sample, for example: an incorrectly predicted sample with high confidence
  - For each training data, remove the data and retrain a model, predict on target sample and calculate the differences between the prediction and original prediction
  - Get the most influential top K instances (very likely to be mislabeled in this scenario)
  - Train a transparent model to find out what distinguishes the influential instances from the non- influential instances by analyzing their features (optional, for better understand the model)

Evaluation

- Human review: which method that human can get more insight of the model?
- Fidelity: how well does the method approximate the black box model?
- Stability: how much does an explanation differ for similar instances?
- Complexity: computational complexity of the method
- Coverage: the types of models that the method can explain
- …

Available Tools

- LIME https://github.com/ankurtaly/Integrated-Gradients
- SHAP implementation in Python https://github.com/slundberg/shap
  - https://shap.readthedocs.io/en/latest/tabular_examples.html
- Captum: PyTorch model interpretability tool https://github.com/pytorch/captum
- ELI5: a library for debugging/inspecting machine learning classifiers and explaining their prediction https://eli5.readthedocs.io/en/latest/
- Influence function implementation in Python https://github.com/kohpangwei/influence-release

References

- Molnar, Christoph. "Interpretable machine learning. A Guide for Making Black Box Models Explainable", 2021. https://christophm.github.io/interpretable-ml-book/.
- Anon. KDD'19 Explainable AI Tutorial. Retrieved September 13, 2019 from https://sites.google.com/view/kdd19-explainable-ai-tutorial
- Anon. ICCV'19 Tutorial on Interpretable Machine Learning in Computer Vision. Retrieved September 20, 2019 from https://interpretablevision.github.io/

Summary

- **Transparent Models**: Linear Regression, Decision Tree, KNN, Bayesian Network…
- **Post-hoc Explanation**
  - **Global Model Explanation**: Permutations, Partial Dependence Plots, Global Surrogate …
  - **Individual Prediction Explanation**: Attribution, Influential Instances, Local Surrogate …