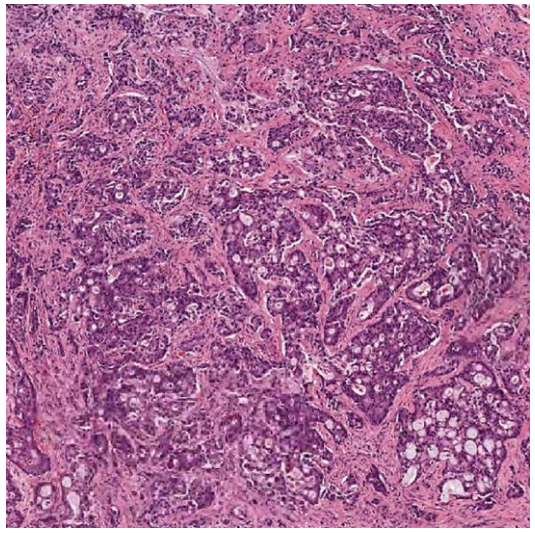
Classification problems and Logistic Regression

Classification

- Detection of cancer in human tissue(cancer / no cancer)
- Prediction of eye color using DNA sequencing



Regression

- Response variable is quantitative
- Temperature forecast for tomorrow
- Company growth prediction for the next quarter

Problem statement

- Efficient recycling requires separation of plastics / glass by their type, use data science to do this automatically.

- Data set: “Glass Identification Data Set from UCI. It contains 10 attributes including id. The response is glass type.”

- The goal is to predict if a glass sample is recyclable or not!

Data set

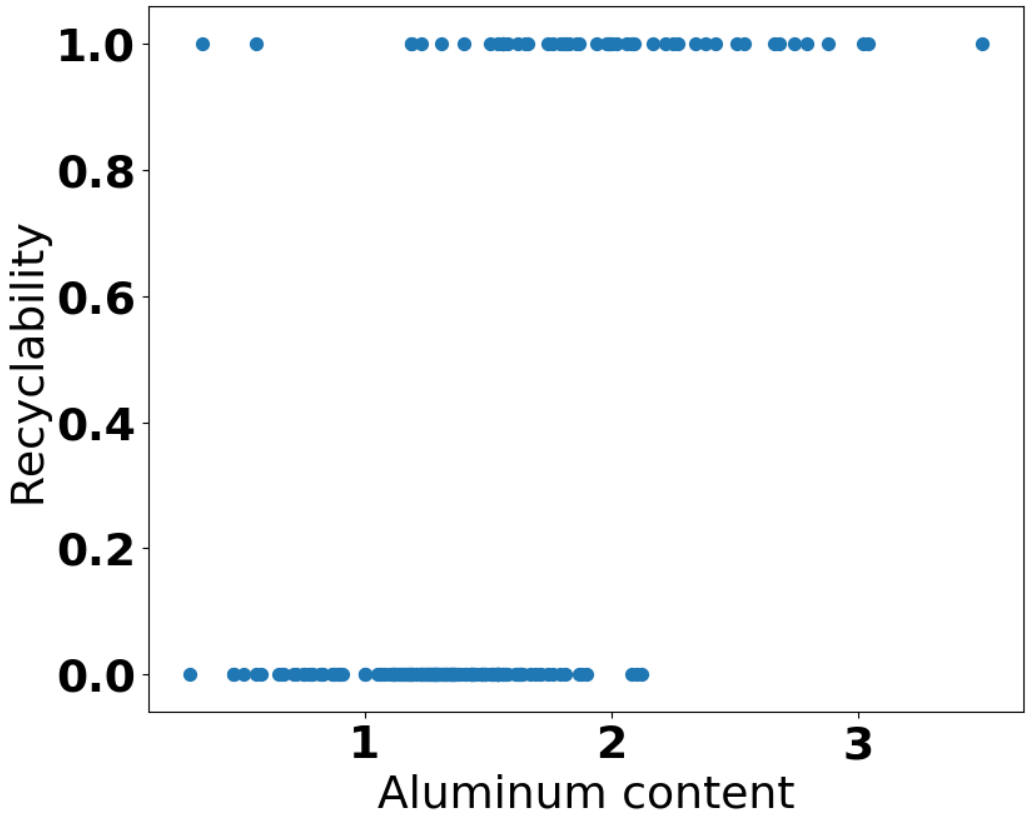
- **9 features:**
  - **ri – refractive index**
  - **al – aluminum**
  - **ca – calcium**

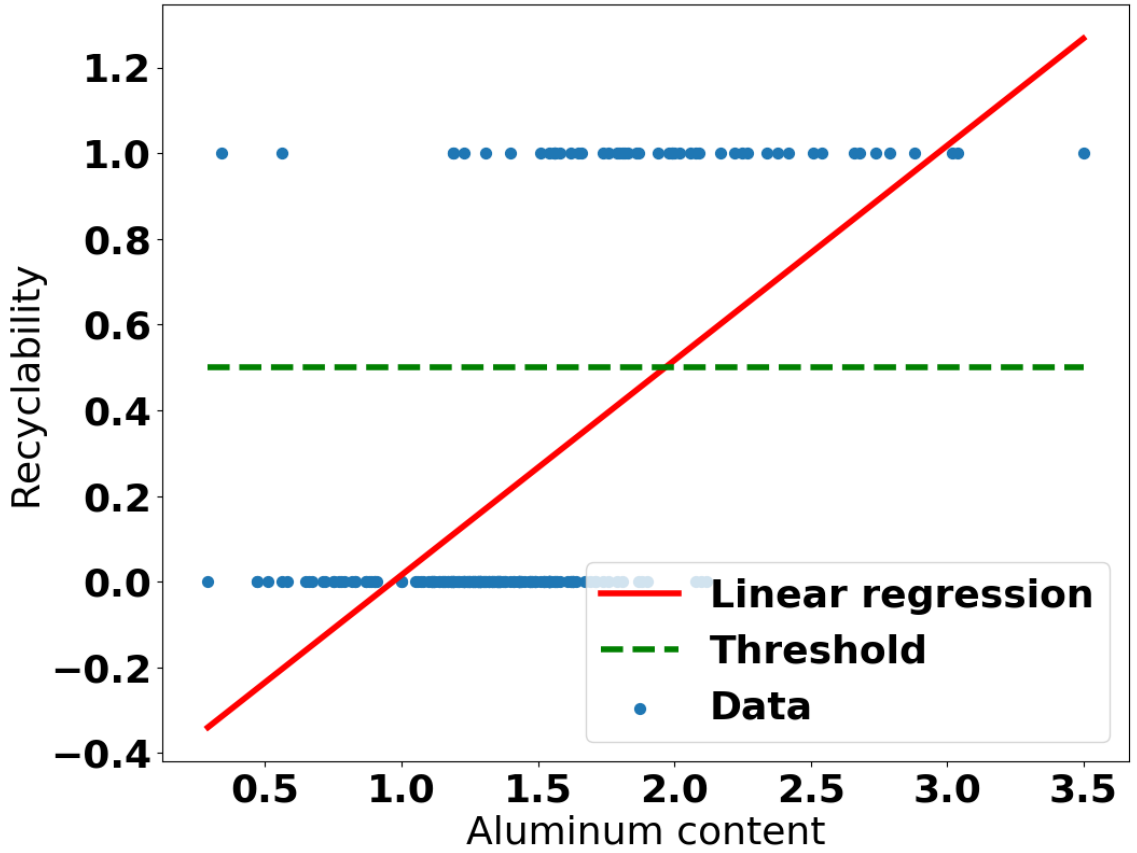
na – sodium
si – silicon
ba – barium

mg – magnesium
k – potassium
fe - iron

```
In [4]: glass.head()
Out[4]:
          ri     na    mg    al     si     k     ca   ba    fe  glass_type
id
22   1.51966  14.77  3.75  0.29  72.02  0.03   9.00  0.0  0.00           1
185  1.51115  17.38  0.00  0.34  75.41  0.00   6.65  0.0  0.00           6
40   1.52213  14.21  3.82  0.47  71.77  0.11   9.57  0.0  0.00           1
39   1.52213  14.21  3.82  0.47  71.77  0.11   9.57  0.0  0.00           1
51   1.52320  13.72  3.72  0.51  71.75  0.09  10.06  0.0  0.16           1
```

In principle we can use regression





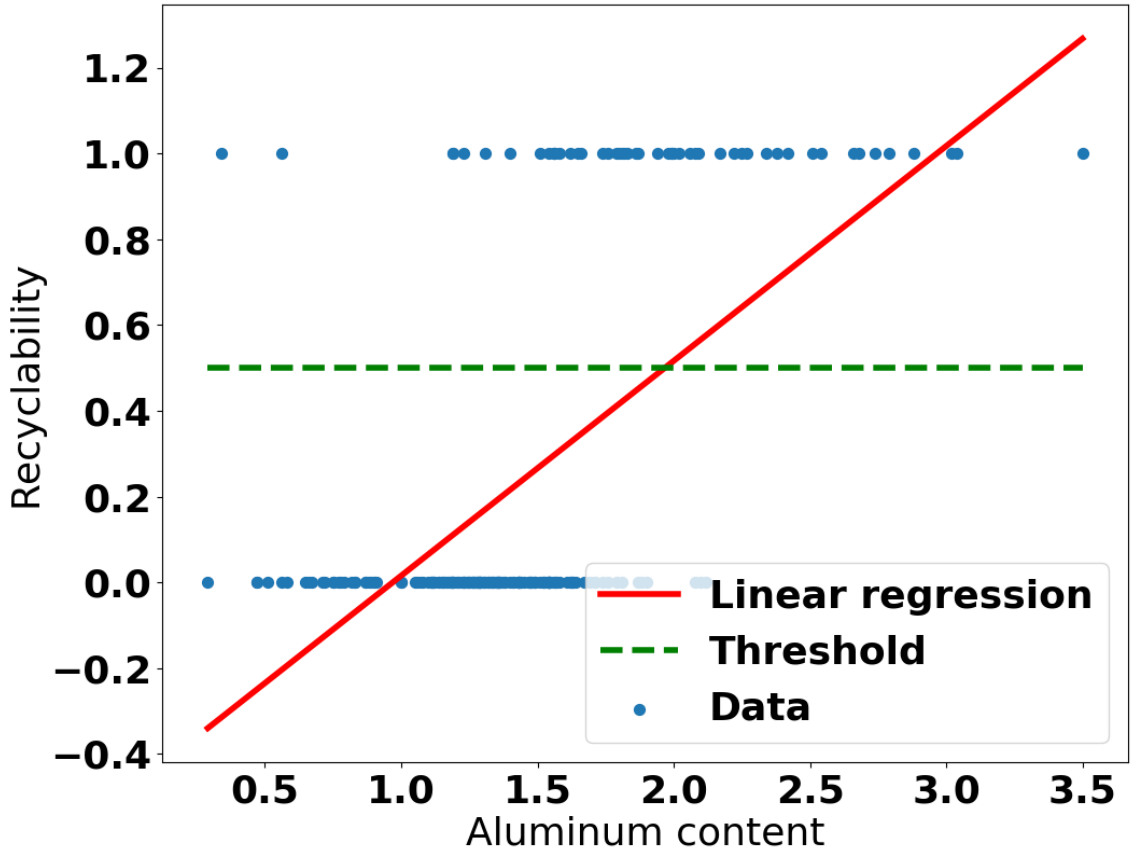
Predicted values can be mapped:

> 0.5 → 1

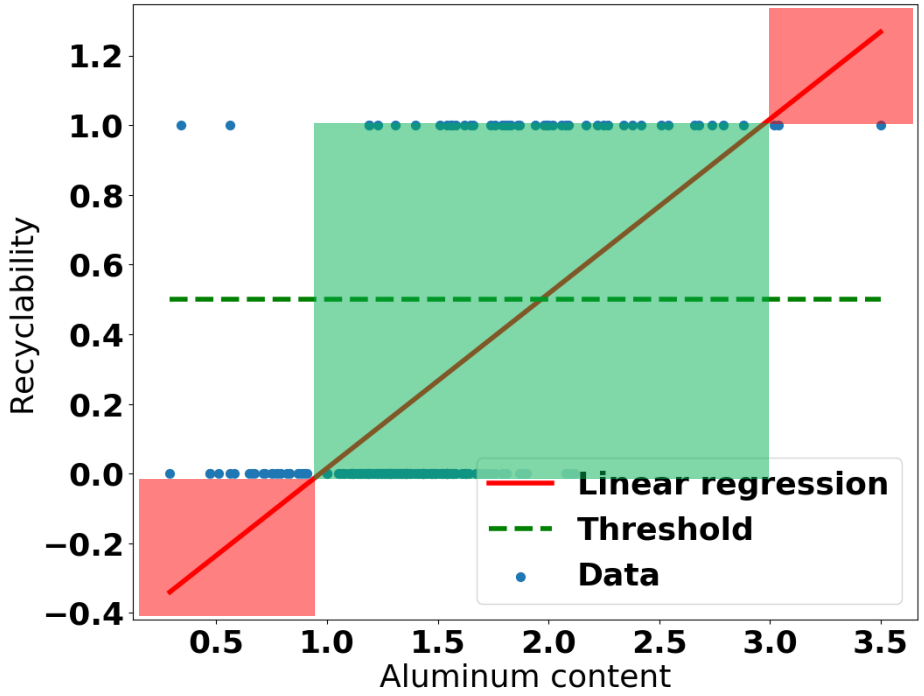
< 0.5 → 0

Key questions:

- What is the significance of the predicted values?
- What happens with more than two labels?



- Cannot predict probability of the sample's label, linear function is not bounded between [0,1].



Why not regression methods for classification:

- Regression methods are all about quantitative independent variables predicting another quantitative dependent variable
  - In classification (logistic regression) the dependent variables is binary (0 or 1)
- Binary data (classification) does not have a normal distribution which typically is required for most types of regression problems
  - In classification, the dependent variable follows the Bernoulli distributions
- Using regression methods the predicted value for the dependent variable can be less than 0 and more than 1 which does not abide by the rules of probability
- Probability often have different shapes in their distribution
  - For instance you can have very high y-values for extreme values (very small or very big) on the x-axis which would look like a ‘U’ shaped distribution. It is difficult to fit a regression line against such shapes.

How does Logistic Regression work?

- Logistic Regression seeks to:
    - **Model** the probability of an event occurring depending on the values of the independent variables (categorical or numerical)
    - **Estimate** the probability that an event occurs for a randomly selected observation versus the probability that the event does not occur
    - **Predict** the effect of a series of variables on a binary response variable
    - **Classify** observations by estimating the probability that an observation is in a particular category

- Probability $P(x)$ Range: [0,1]
- Odds $O(x) = \frac{P(x)}{1-P(x)}$ Range: [0, inf]
- Logit $\log O(x)$ Range: [-inf, inf]

- **Assume:** $\log O(x) = \alpha_1 x + \alpha_0$

- **Linear function:** $\log O(x) = \alpha_1 x + \alpha_0$

- **Inverse Logit:** $P(x) = \frac{e^{\alpha}}{1+e^{\alpha}}$

$$P(x) = \frac{e^{\alpha_1 x + \alpha_0}}{1+e^{\alpha_1 x + \alpha_0}}$$

Estimated Regression Equation

- As in the previous lecture, use the likelihood function to find α_0, α_1:

$$L(\alpha_0, \alpha_1) = \prod_{i:y_i=1} P(x_i) \prod_{i':y_i=1} \left(1 - P\left(x_{i'}\right)\right)$$

Don't worry, this is done by the library of your choice.

Apply logistic regression on the previous data set:

Logistic fit → probability

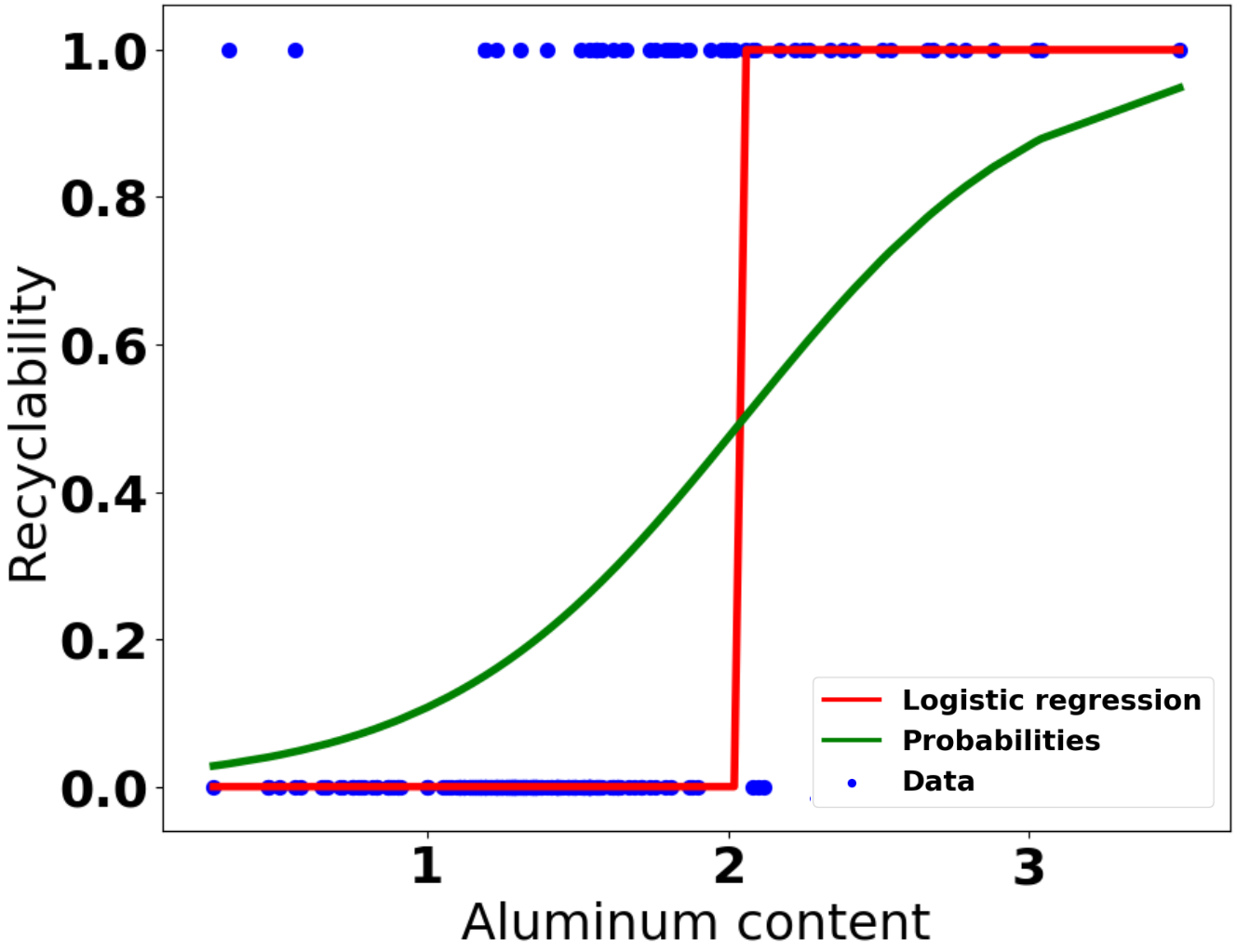
Coefficient α_1:

> 0 → increasing

< 0 → decreasing

Coefficient α_0:

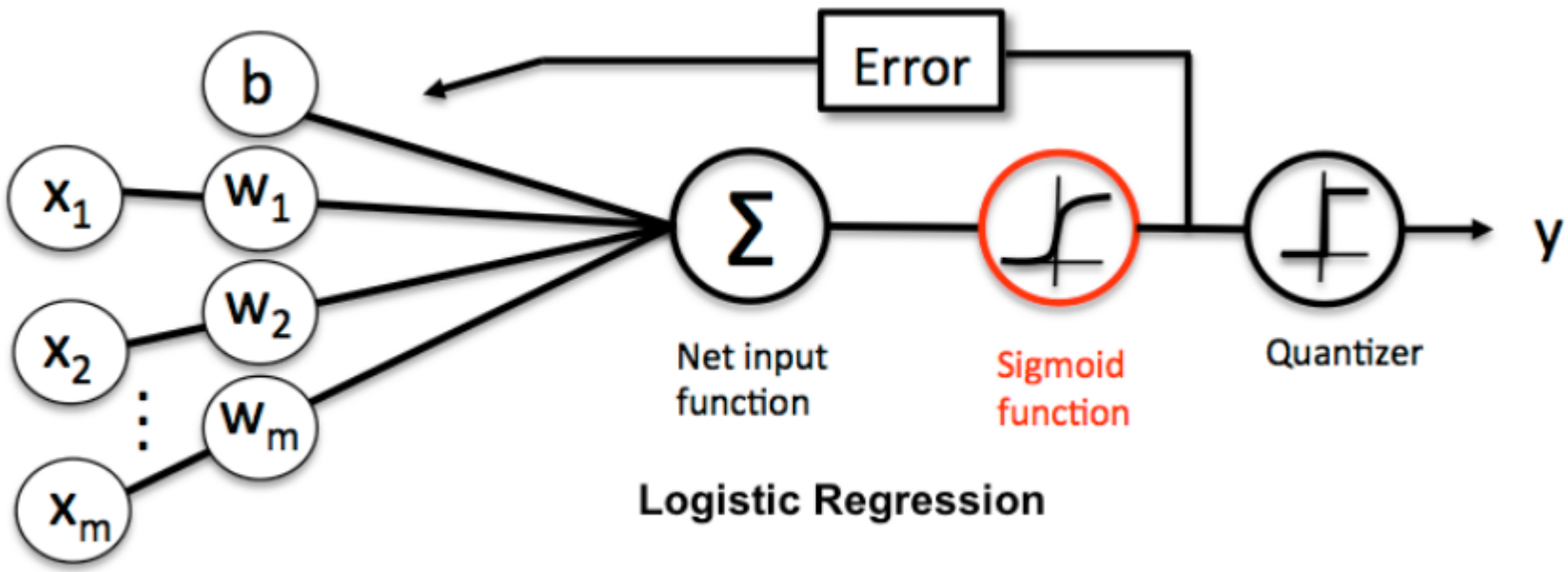
Intercept

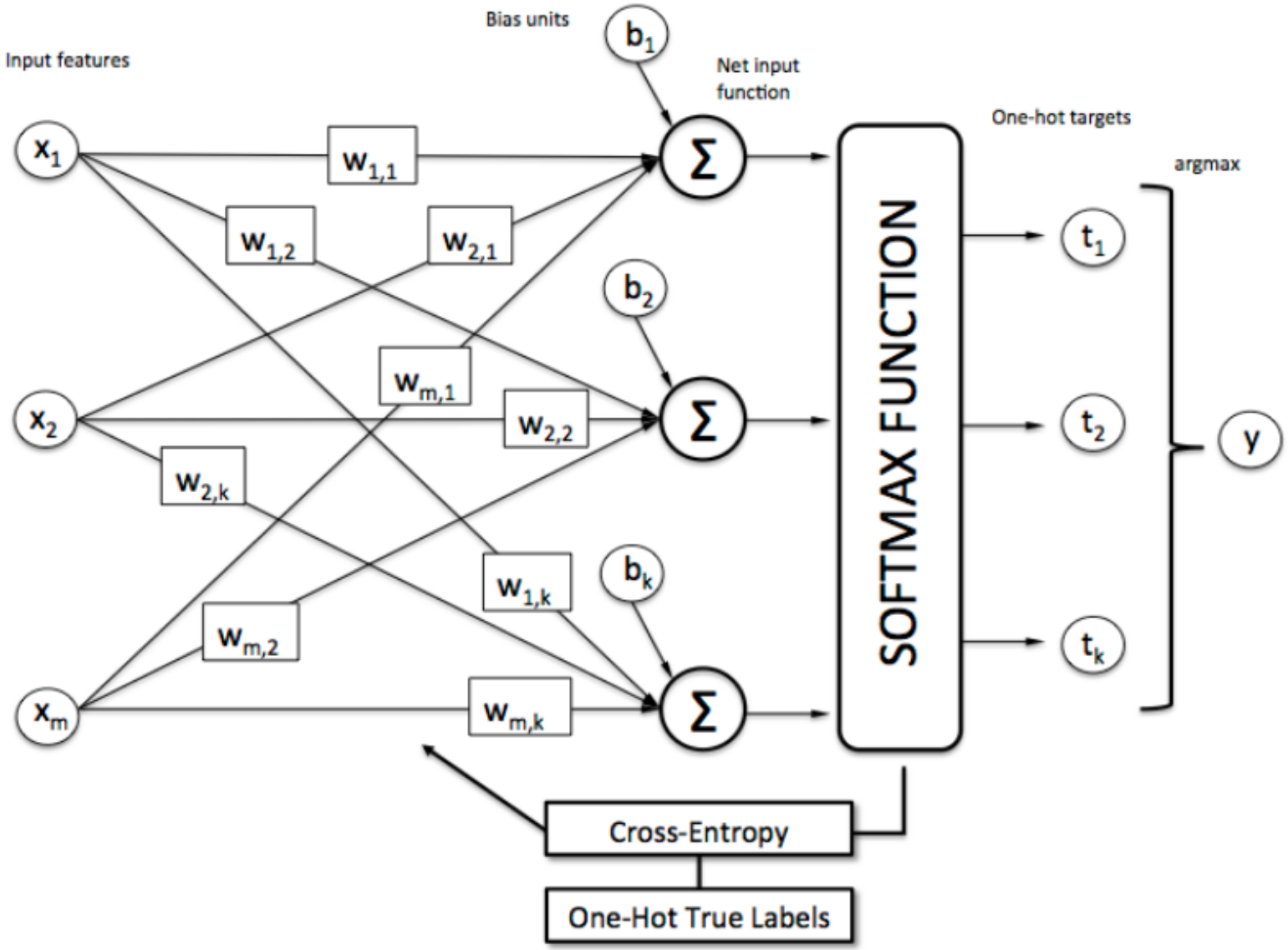


Extension I: What if there are multiple features and two labels?

Assume: $\log O(x) = \alpha_0 + \sum_{i=1}^{n} \alpha_i x_i$, follow the same algorithm!

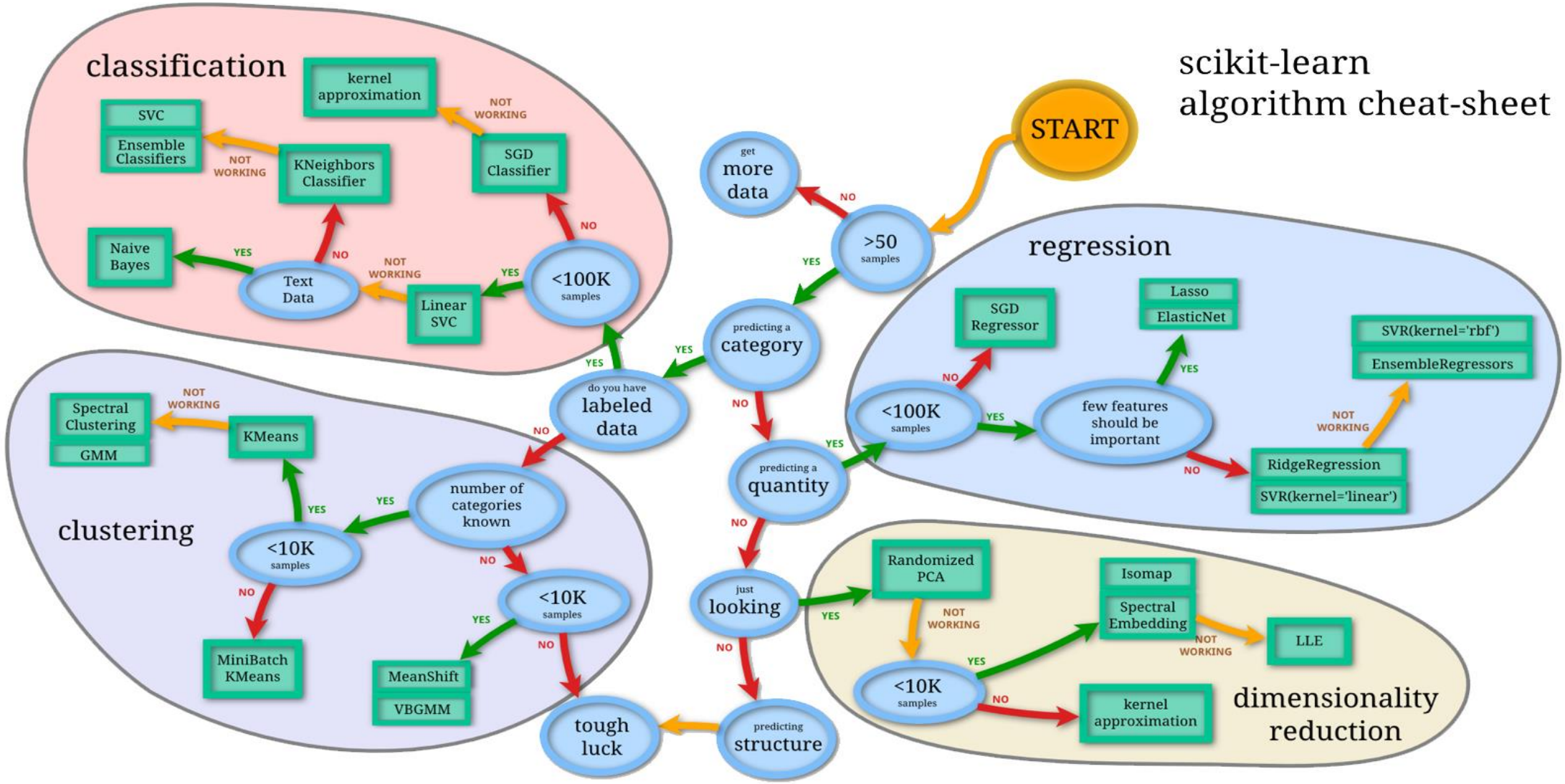
Extension II: What if there are more than two labels?


Error
b
x1
w1
w2
x2
wm
xm
Σ
y
Net input function
Sigmoid function
Quantizer
Logistic Regression


Bias units
Input features
Net input function
One-hot targets
argmax
SOFTMAX FUNCTION
Cross-Entropy
One-Hot True Labels

Why Automated Machine Learning

Machine Learning Complexity

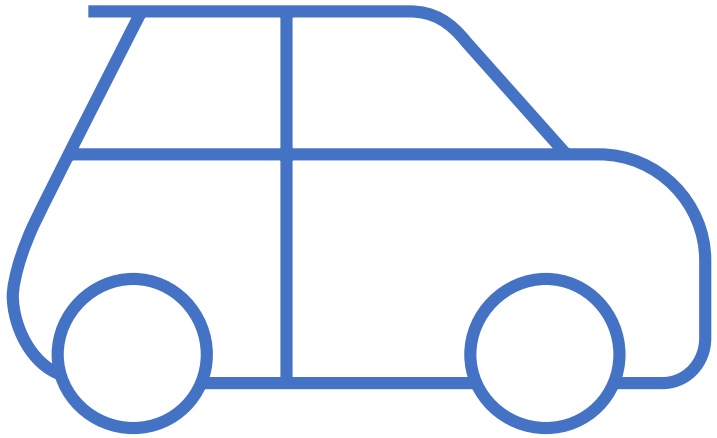


Source: http://scikit-learn.org/stable/tutorial/machine_learning_map/index.html

Energy Forecasting – Traditional Way

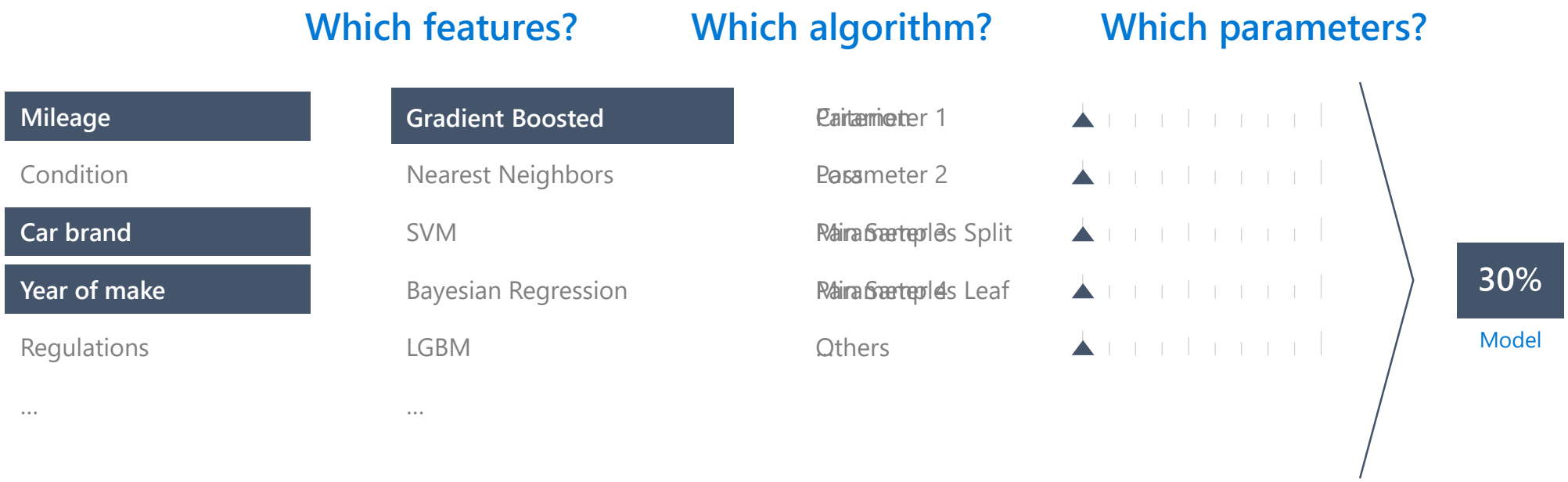
- `1-data-preparation.ipynb` - this Jupyter notebook downloads and processes the data to prepare it for modeling. This is the first notebook you will run.
- `2-linear-regression.ipynb` - this notebook trains a linear regression model on the training data.
- `3-ridge.ipynb` - trains a ridge regression model.
- `4-ridge-poly2.ipynb` - trains a ridge regression model on polynomial features of degree 2.
- `5-mlp.ipynb` - trains a multi-layer perceptron neural network.
- `6-dtree.ipynb` - trains a decision tree model.
- `7-gbm.ipynb` - trains a gradient boosted machine model.
- `8-evaluate-model.py` - this script loads a trained model and uses it to make predictions on a held-out test dataset. It produces model evaluation metrics so the performance of different models can be compared.
- `9-forecast-output-exploration.ipynb` - this notebook produces visualizations of the forecasts generated by the machine learning models.
- `10-deploy-model.ipynb` - this notebook demonstrates how a trained forecasting model can be operationalized in a realtime web service.
- `evaluate-all-models.py` - this script evaluates all trained models. It provides an alternative to running `8-evaluate-model.py` for each trained model individually.

Machine Learning Problem Example



How much is this car worth?

Model Creation Is Typically Time-Consuming



Model Creation Is Typically Time-Consuming

Which features?

- Mileage
- Condition
- Car brand
- Year of make
- Regulations
- …

Which algorithm?

- Gradient Boosted
- Nearest Neighbors
- SVM
- Bayesian Regression
- LGBM
- …

Which parameters?

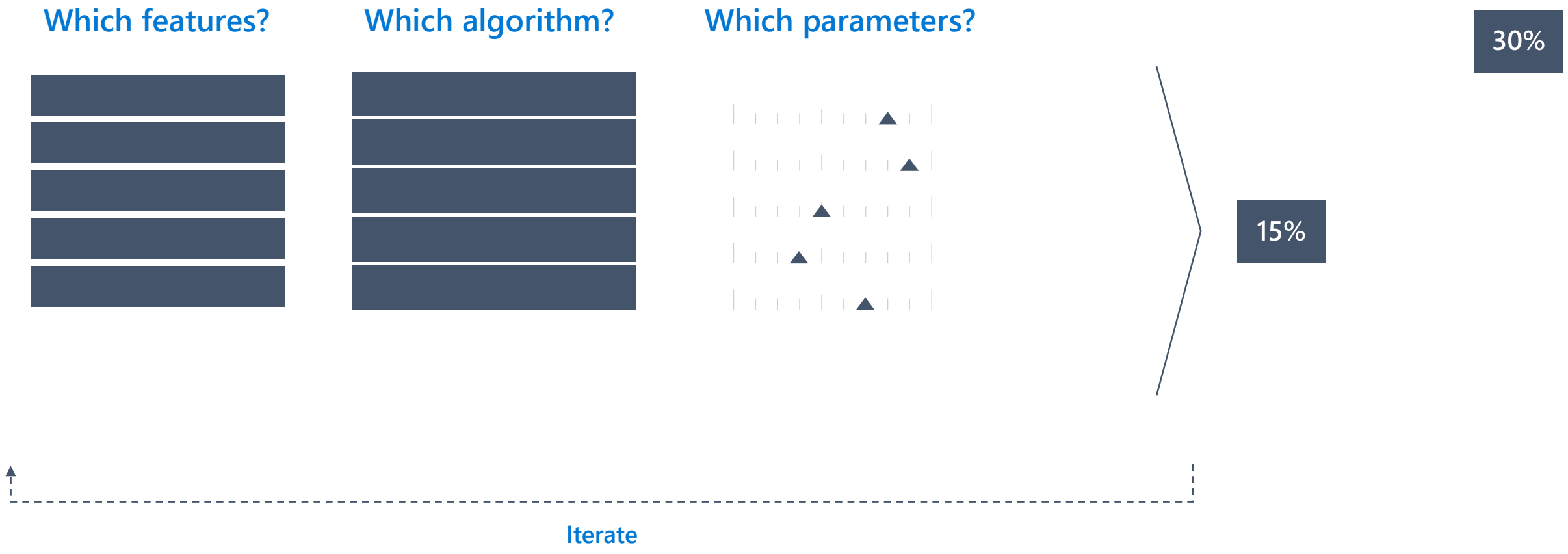
- Number of Neighbors
- Weights
- Min Samples Split
- Min Samples Leaf
- Others

30%

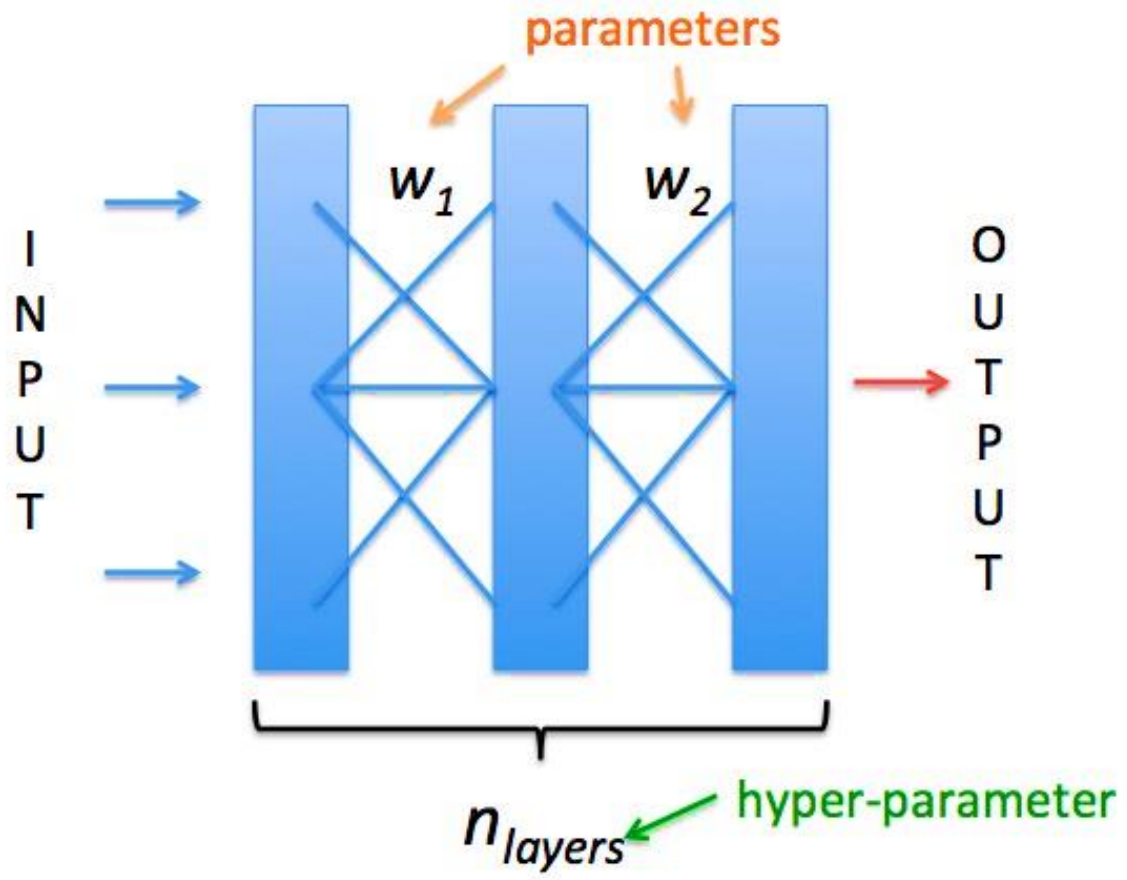
Model

Iterate

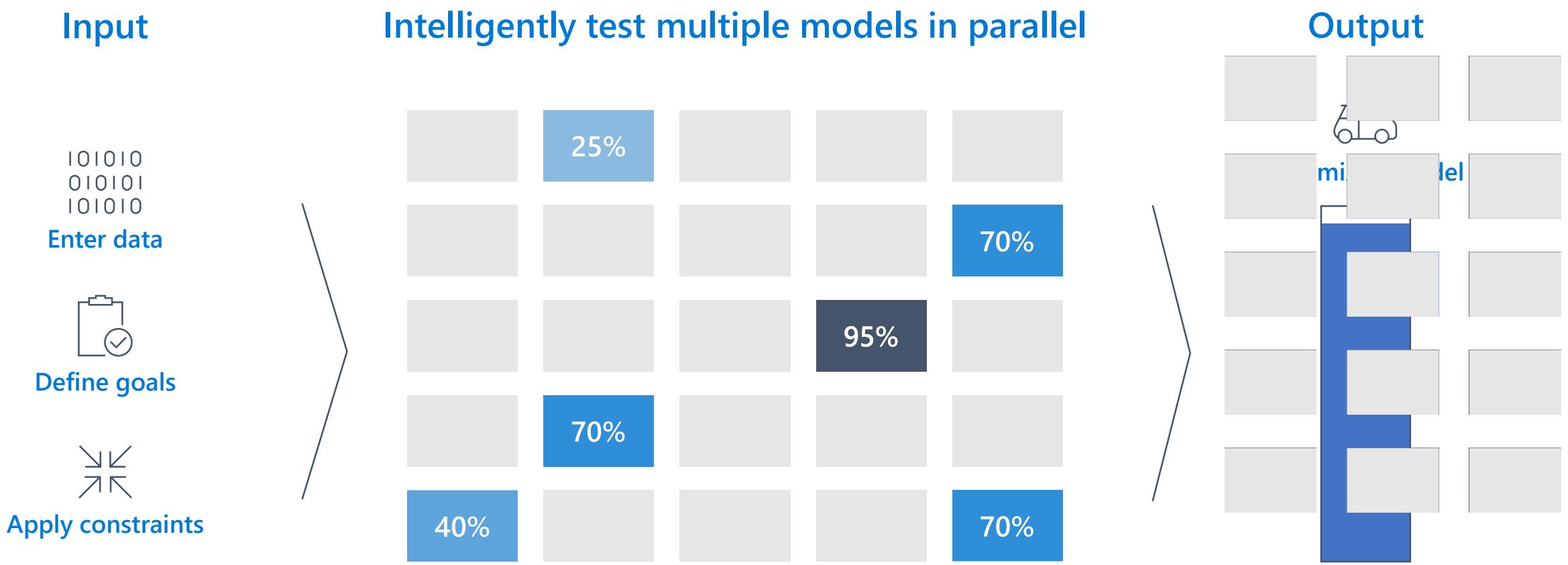
Model Creation Is Typically Time-Consuming



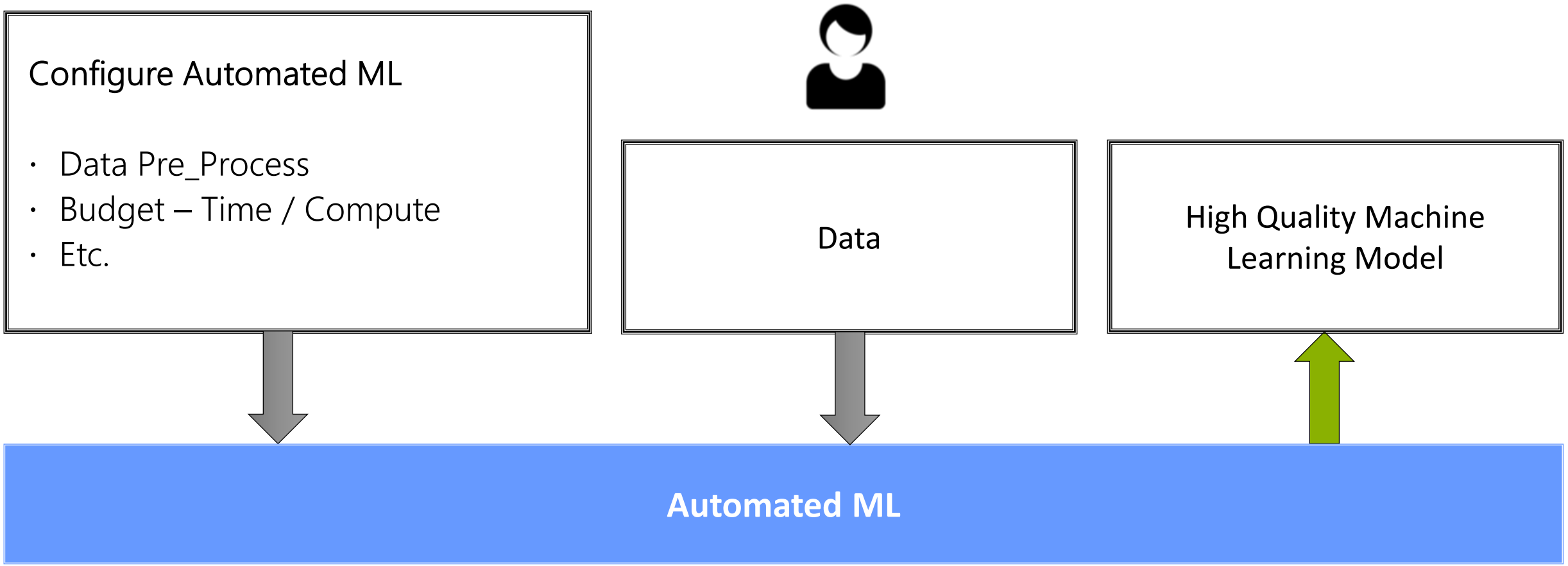
Model Parameters and Hyperparameters



Automated ML Accelerates Model Development



Automated ML



Automated Machine Learning – An Overview

Automated Machine Learning — An Overview

Think Gradient

Jan 20 · 13 min read ★

Since the dawn of the computer age, scientists and engineers have always wondered about infusing computers with the ability to learn, just like humans do. Alan Turing was amongst the first scientists to posit a theory of intelligence that envisaged computers to one day be able to reach a level of intelligence that aims to reach human parity. Since then a number of giant leaps have been made that have pushed the field of Machine Learning forward. We have seen Machine Learning in many cases beating or at least matching specific human cognitive faculties such as in the case of ResNet (a deep Residual Network architecture) surpassing human performance in