Office Hours

BUILD AN AI MODEL TO FORECAST TRENDS AND
DETECT OPPORTUNITIES IN RETAIL BANKING

DRUCKER LEADERSHIP ACADEMY
It's an Art

Tangible

NAAMII

Machine Learning

Extract information from data to make predictions

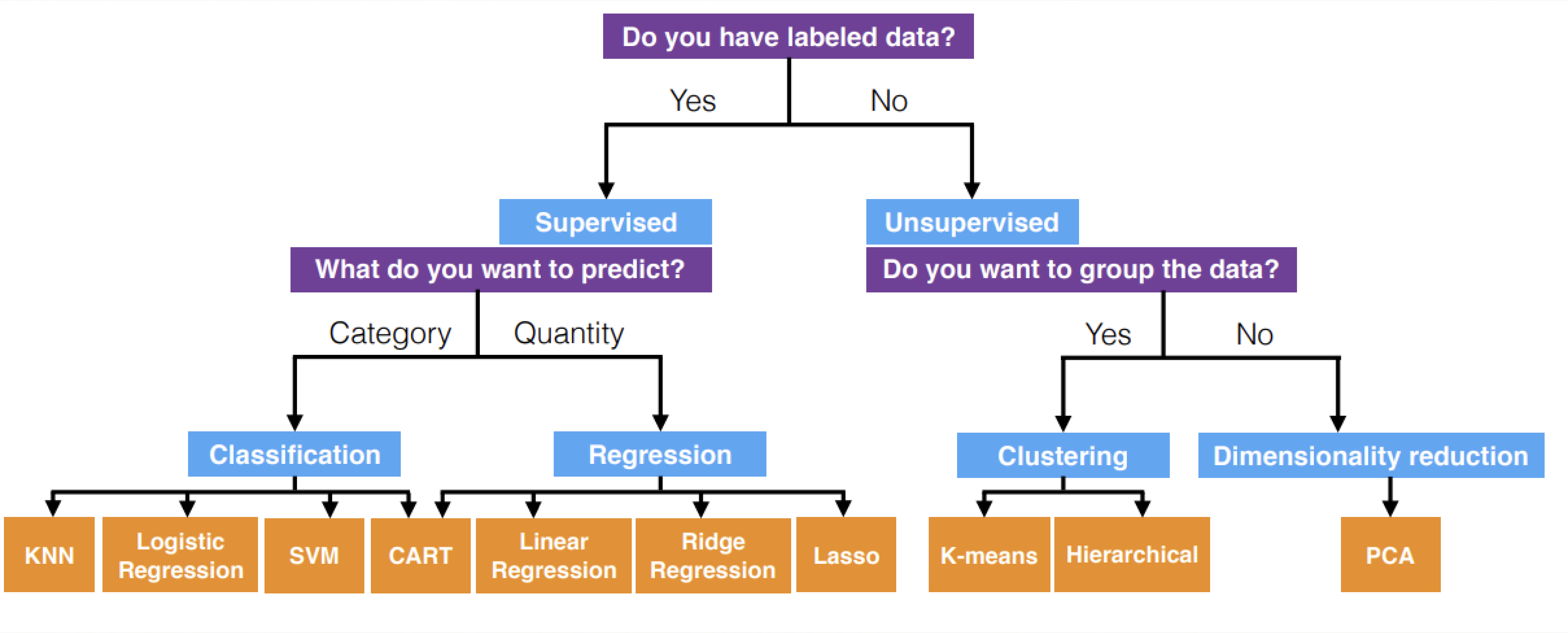
Learning Methods:

1. **Supervised Learning Models**: predict something, given some other things

a. **regression** - predict a real scaler(number) or vector value.
b. **classification** - predict a value from a finite set {TRUE, FALSE}
c. **forecasting** - predict a future value, given a current and past values

2. **Unsupervised learning** models create a model of the data

Machine Learning



Example Problems

Supervised Learning: Classification

A bank has historical data on 10,000 past loans, including applicant income, credit score, and debt level. We also have a label for each: 'Defaulted' or 'Repaid'. We need a model to automatically approve or reject new loan applications.

Supervised Learning: Regression

A real estate agency wants to suggest listing prices for homeowners. We have a dataset of recently sold houses with features like square footage, number of bedrooms, and neighborhood rating. We want to predict the exact dollar amount a new house will sell for.

Example Problems

Unsupervised Learning: Clustering
Our marketing team has a database of 50,000 customers with data on their age, spending habits, and browsing history. We do not have pre-defined segments (no labels). We want to discover natural groupings so we can send targeted email campaigns to different 'types' of shoppers.

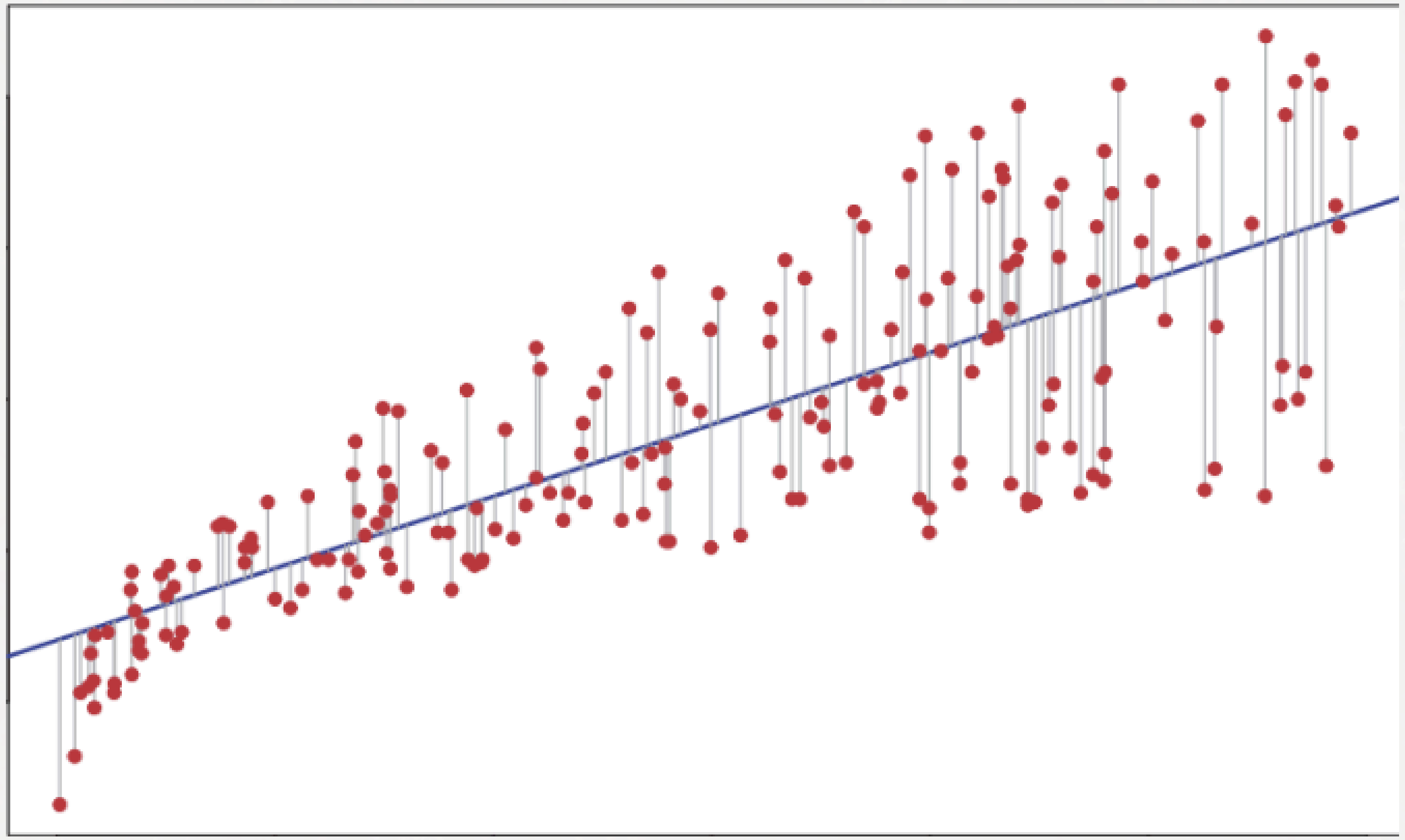
Unsupervised Learning: Dimensionality Reduction
We are analyzing genetic data where every patient has 20,000 different gene markers. It is impossible to visualize this or run standard models because the dataset is too wide and slow. We need to compress these 20,000 features into just 2 or 3 'summary' features that retain the most important variations, so we can plot it on a simple 2D graph.

Linear Regression

Simple Linear Regression:
Predict a quantitative response Y on the basis of single predictor variable X.

$$Y = \beta_0 + \beta_1 X + \epsilon$$

B0and B1 are the coefficients or parameters of the linear model. In this case they represent the intercept and slope terms of a line.



Linear Regression

Least Squares:

Typically, how well a linear model is fit to the data is measured using least squares.

Residual for the i-th sample:

$$\epsilon^{(i)} = y^{(i)} - \hat{y}^{(i)}$$

Residual sum of squares (RSS):

$$\text{RSS} = \epsilon^{(1)2} + \epsilon^{(2)2} + \ldots + \epsilon^{(n)2} = \sum_{i=1}^{n}(y^{(i)} - \hat{y}^{(i)})^2$$

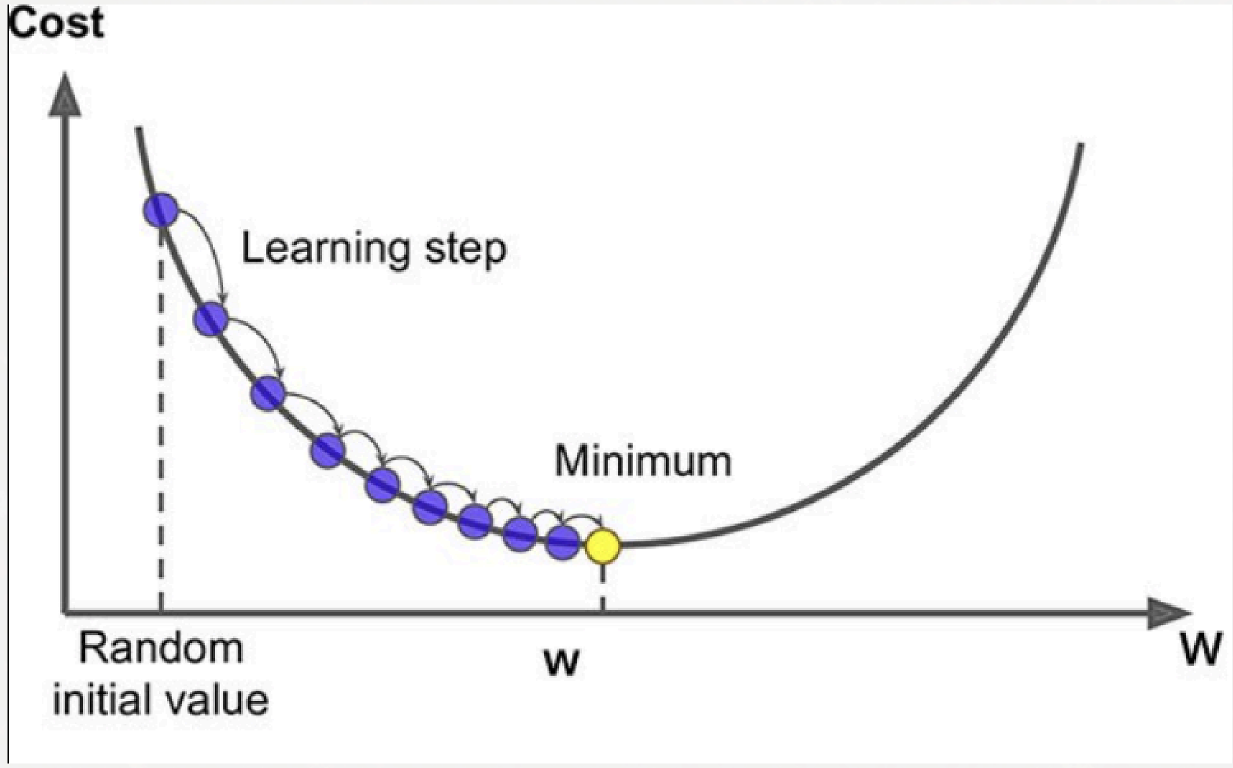
Gradient Descent:

Modify the weights of the equation using :

w = w – α * (dJ/dw)

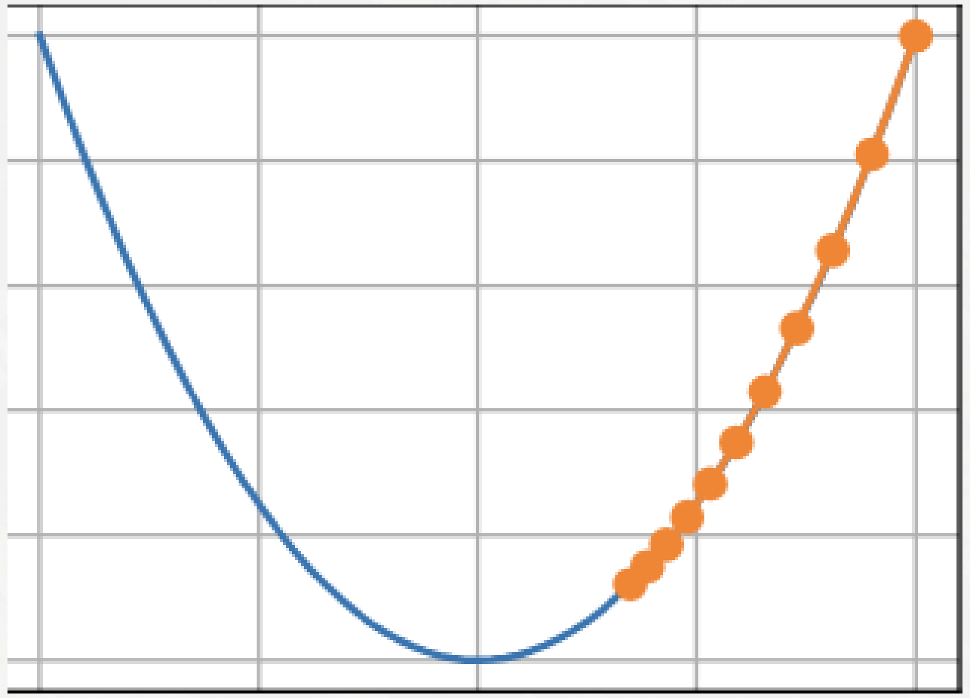
where,

w = weights, α = learning rate, (
dJ/dw) = gradient of the loss function w.r.t. to the weights

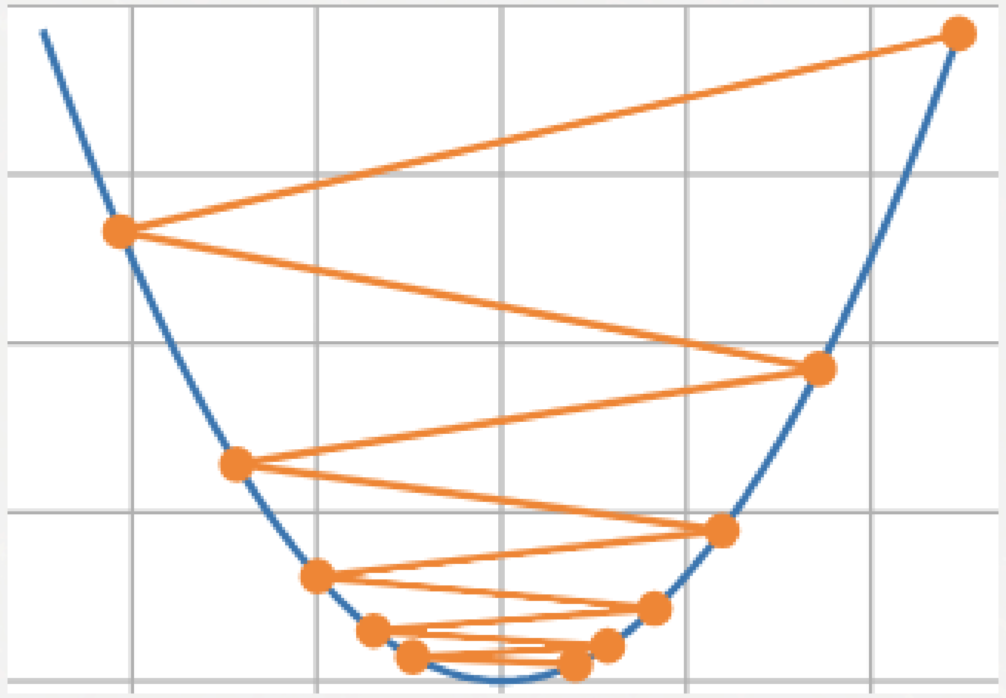


Linear Regression

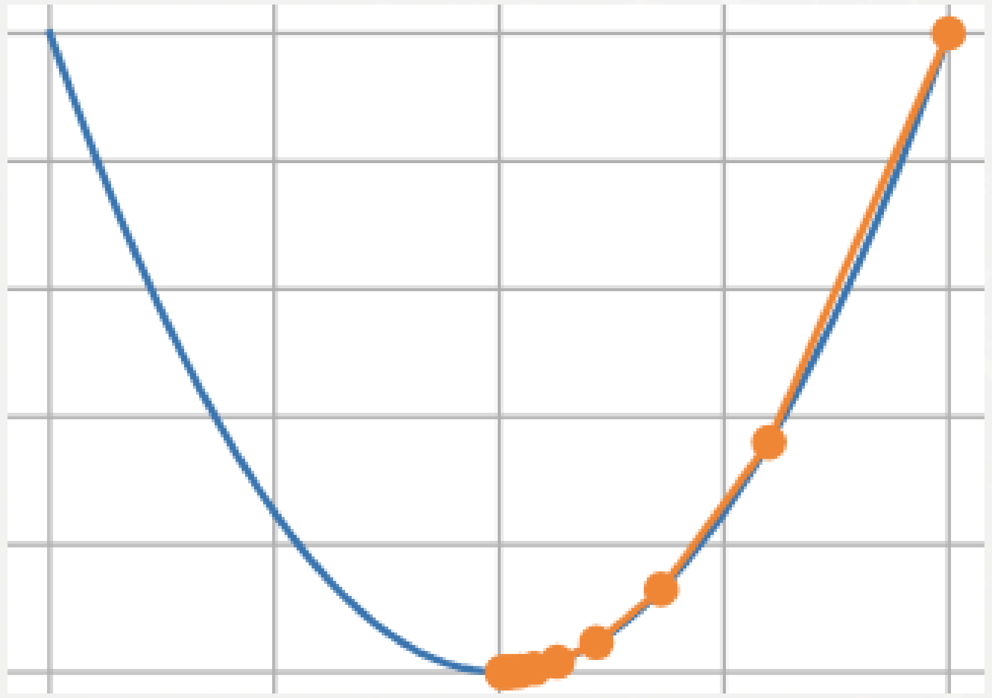
Specifying a learning rate
If too high, the model will oscillate around the loss.
If too low, it will take too long to convege.



Too low.



Too high



Just the right amount

Linear Regression

Assumptions in Linear Regression:

Related to data:
Linearity: The relationship between x and y is **linear**
If the relationship is not linear, model won't be a good fit for data and parameter estimates will be meaningless. Use polynomial regression.

No multicolinearity: Two or more of the independent variables are not highly correlated. The weights of the equation for an independent imply how much the dependent variable changes by if all the other independent variables are kept constant. If, mulitple variables are correlated it is not possible to keep the other variables constant.

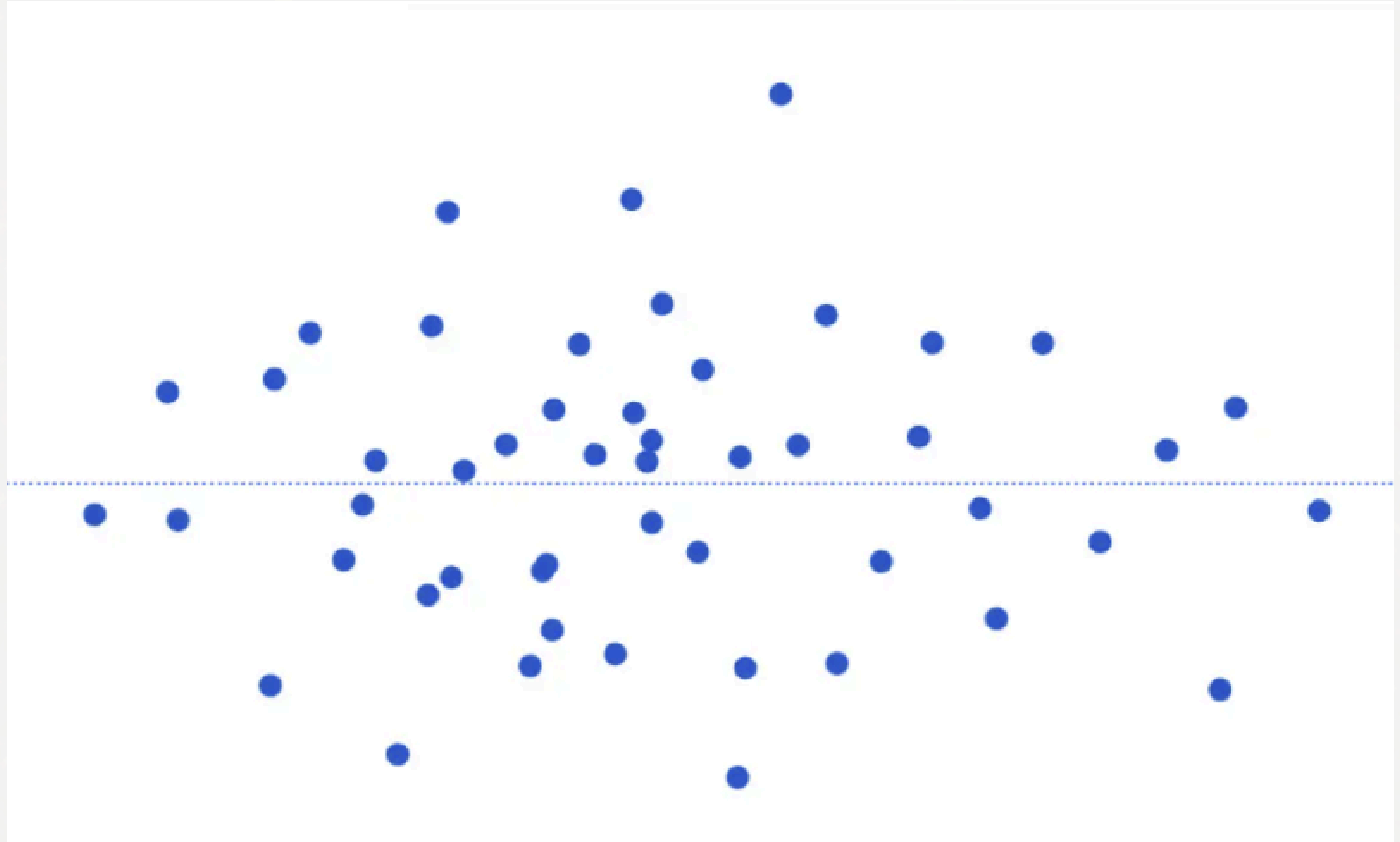
Linear Regression

Assumptions related to residuals/error:

Residuals are normally distributed:
Errrors follow a symmetric bell curve, which is necessary for reliable confidence intervals.

Residuals have the same variance
The spread of errors remains constant across all predicted values (Homoscedasticity), avoiding "funnel" shapes in plots.

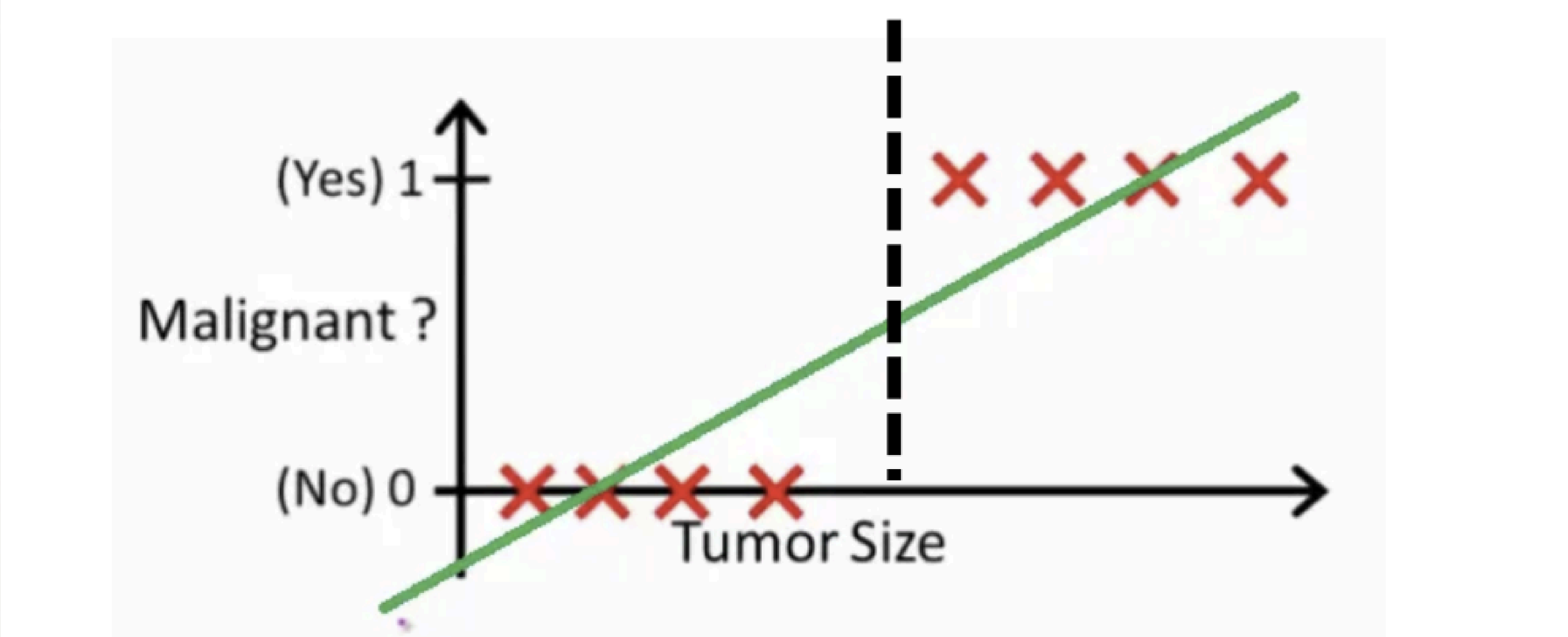
Residuals are not correlated
Error terms are independent of each other; the value of one error does not influence the next.



Linear Regression

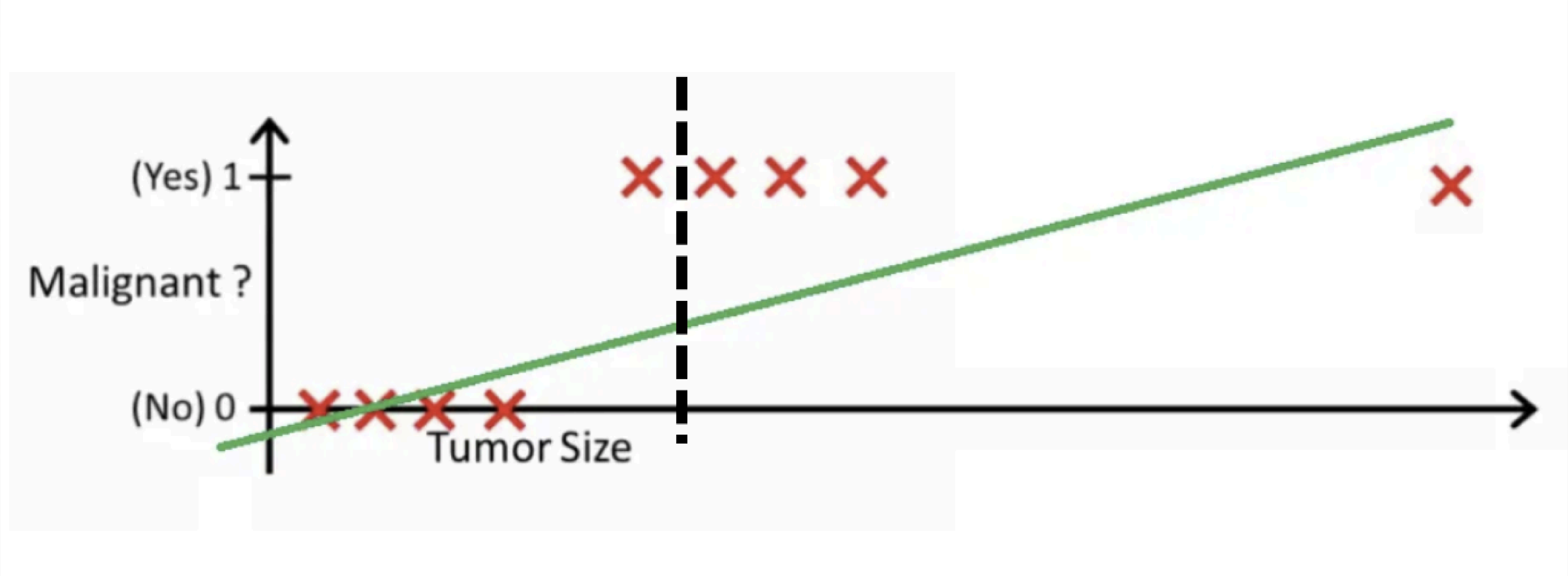
For classification we need to select from a set of values.

We can use a **decision threshold** with linear regression to classify. Any value above a certain threshold can be considered to be positive and below it can be considered negative.



Linear Regression

Linear regression is sensitive to outliers.



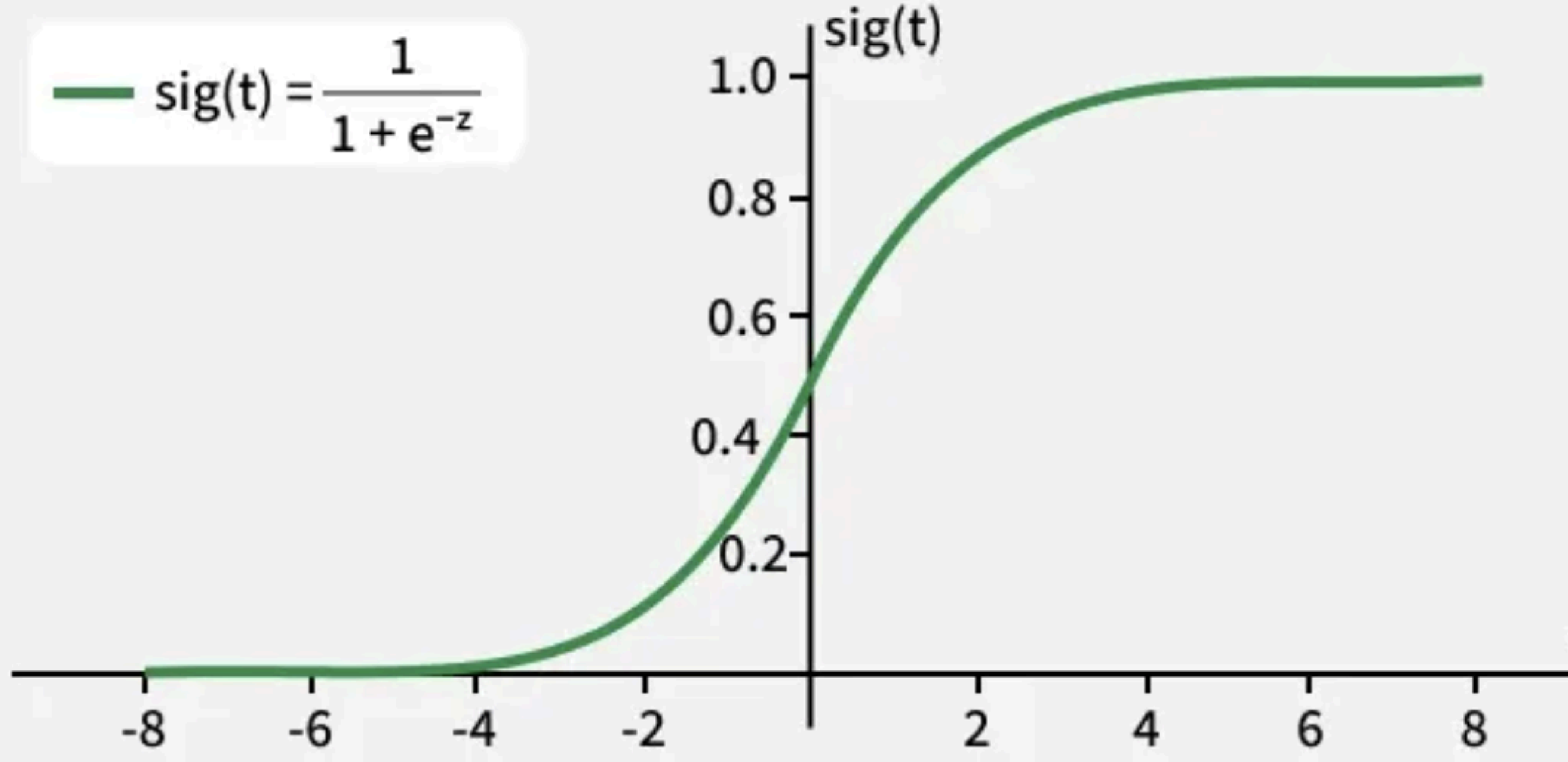
Logistic Regression

Use a sigmoid function to bound the output between 0 and 1.

1. Gives result between 0 and 1.
2. Less sensitive to outliers.

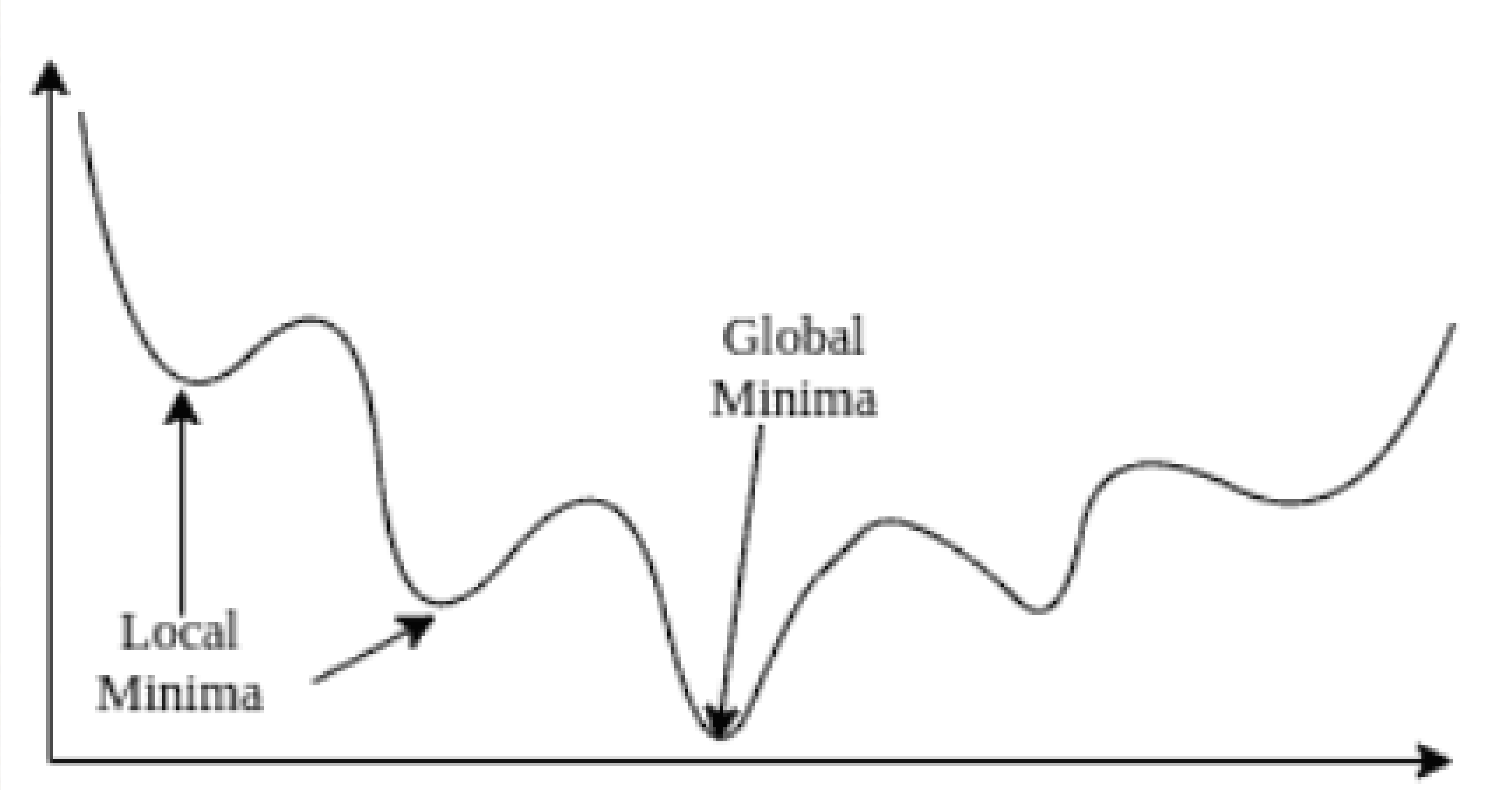
$$z = w \cdot X + b$$

$$\sigma(z) = \frac{1}{1+e^{-z}}$$



Logistic Regression

Squares Error for logistic regression is not convex



Logistic Regression

Train the model using binary cross entropy loss:

$$\text{Log Loss} = -\frac{1}{N}\sum_{i=1}^{N} y_i \log(y_i') + (1 - y_i)\log(1 - y_i')$$

where:

- N is the number of labeled examples in the dataset
- i is the index of an example in the dataset (e.g., (x_3, y_3) is the third example in the dataset)
- y_i is the label for the ith example. Since this is logistic regression, y_i must either be 0 or 1.
- y_i' is your model's prediction for the ith example (somewhere between 0 and 1), given the set of features in x_i.

Train Test Split

Goal: Develop a model that performs well for new, unseen data

Divide the give data set into 2 parts:

1. training set - train the model using this.
2. testing set - evaluate the models.

Defining target(y-variable)

Must be **actionable**. If we predict it, can we do something about it? Predicting age of the customer is not as useful as predicting whether the loan customers are likely to miss payments.

Avoid **Data Leakage**: The y variable must be an outcome that happens after the events captured in the input variable have occured. If, we are predicting whether a loan will be approved or not, we can’t use loan approved date.

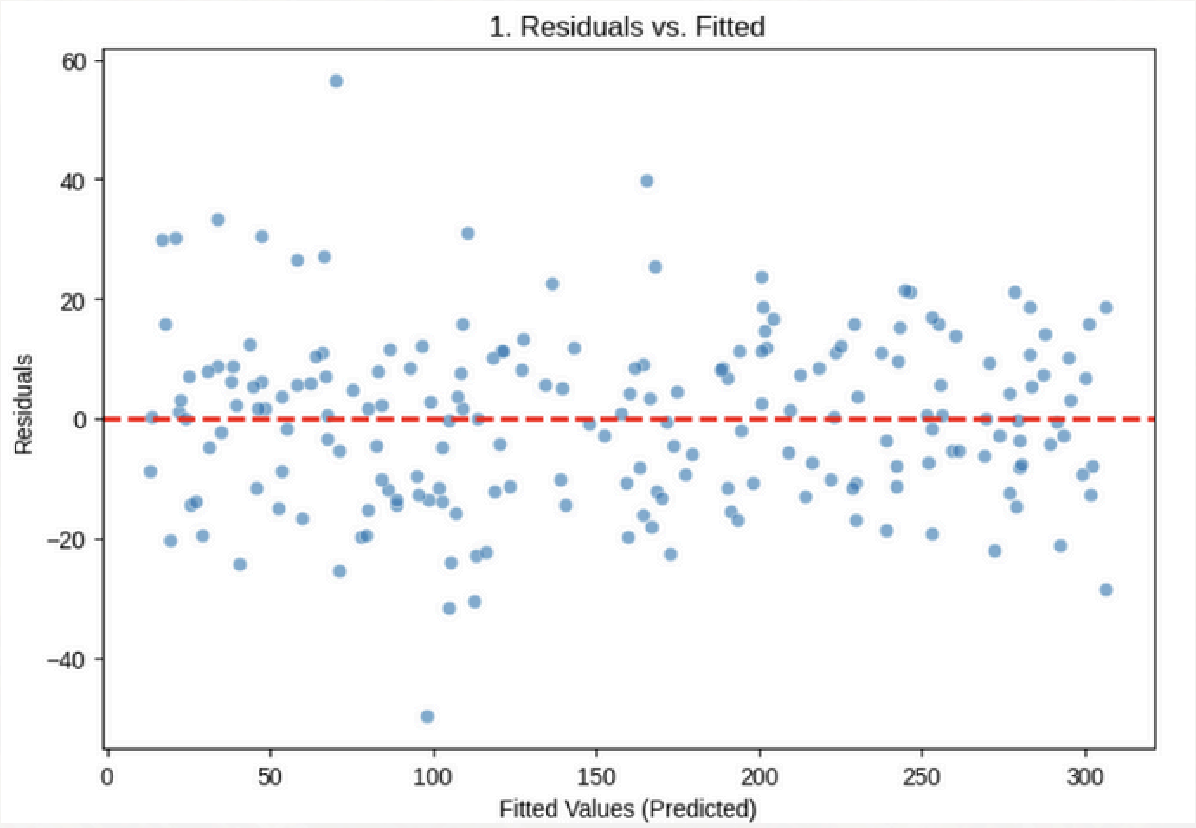
The variable being predicted must have **high variance**. If, no variance there is **nothing to predict**. If, low variance then there are many examples of one kind and too little of another kind, there will be **class imbalances**.

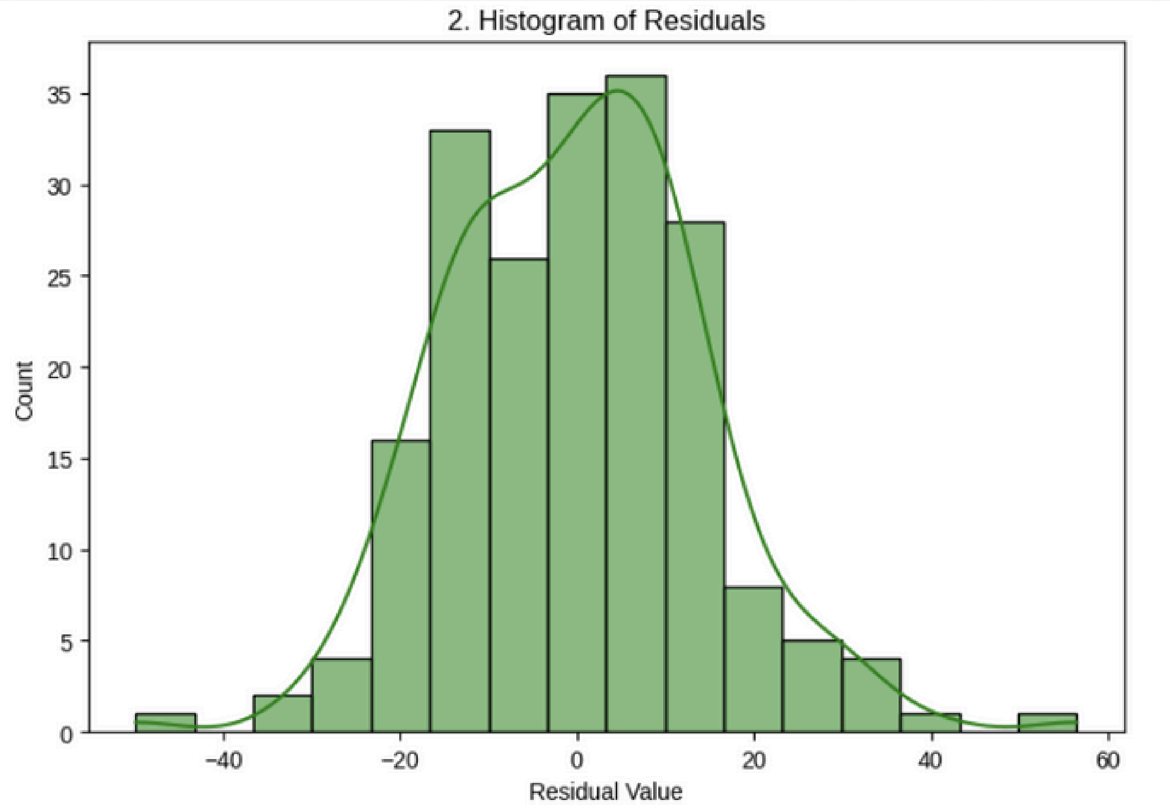
Model Evaluation

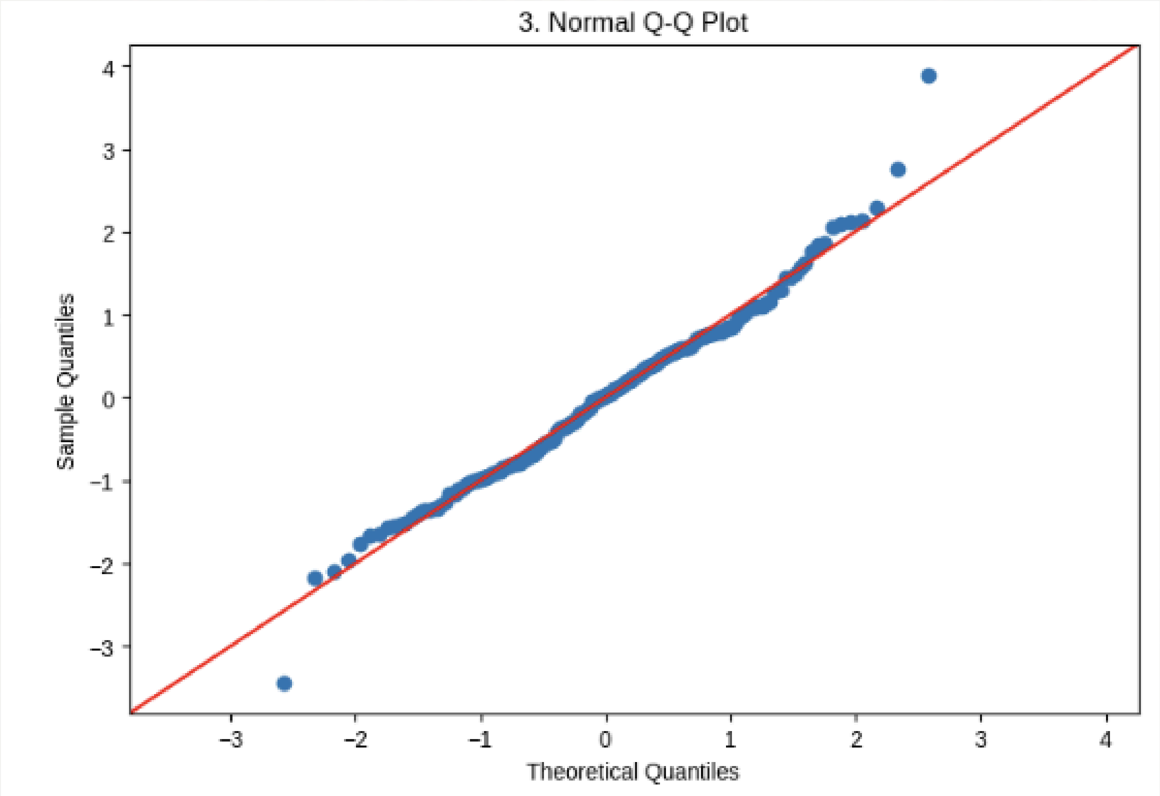
Residual Plots:
Linear regression tries to fit a line that produces the **smallest difference between predicted and actual values**, where these **differences are unbiased as well.**

After, your residual plot looks correct, evaluate on other metrics.

Model Evaluation







Model Evaluation(Linear Regression)

Mean Squared Error (MSE)

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

Root Mean Squared Error (RMSE)

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2}$$

Mean Absolute Error (MAE)

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|\hat{y}_i - y_i|$$

Model Evaluation (Logistic Regression)

Accuracy

Number of correct predictions from all predicted results.

False Positives: Type I Error

False Negatives: Type II Error

Suitable for equal distribution of classes on the classification as opposed to imbalanced datasets; as even if the accuracy may be high, the performance of the model might be bad on the class with smaller number of samples.

Model Evaluation (Logistic Regression)

Precision:
Measures the proportion of the positive predictions that are actually positive

Precision = **Number of True positives (TP) / (Number of true positives(TP) + Number of false positives(FP))**

How correct is a positive prediction

Sensitivity / Recall:
Measures the models ability to predict the positives out of actual positives correctly.

Recall = **Number of true positives (TP) / (Number of true positives (TP) + Number of false negatives(FN))**

Specificity:
Measures the model's ability to predict the negatives out of the actual negatives correctly

Specificity = TN / (TN + FP)

F1-Score:
Harmonic Mean of precision and recal
F1 = **2 * P* R / (P + R)**

Model Evaluation (Logistic Regression)

Confusion Matrixis a simple table used to measure how well a classification model is performing.
This helps you understand where the model is making mistakes so you can improve it. It breaks down the predictions into four categories:

- True Positive (TP): The model correctly predicted a positive outcome i.e the actual outcome was positive.
- True Negative (TN): The model correctly predicted a negative outcome i.e the actual outcome was negative.
- False Positive (FP): The model incorrectly predicted a positive outcome i.e the actual outcome was negative. It is also known as a Type I error.
- False Negative (FN): The model incorrectly predicted a negative outcome i.e the actual outcome was positive. It is also known as a Type II error.

	Predicted Positive	Predicted Negative
Actual Positive	True Positive (TP)	False Negative (FN)
Actual Negative	False Positive (FP)	True Negative (TN)

Model Evaluation(Logistic Regression)

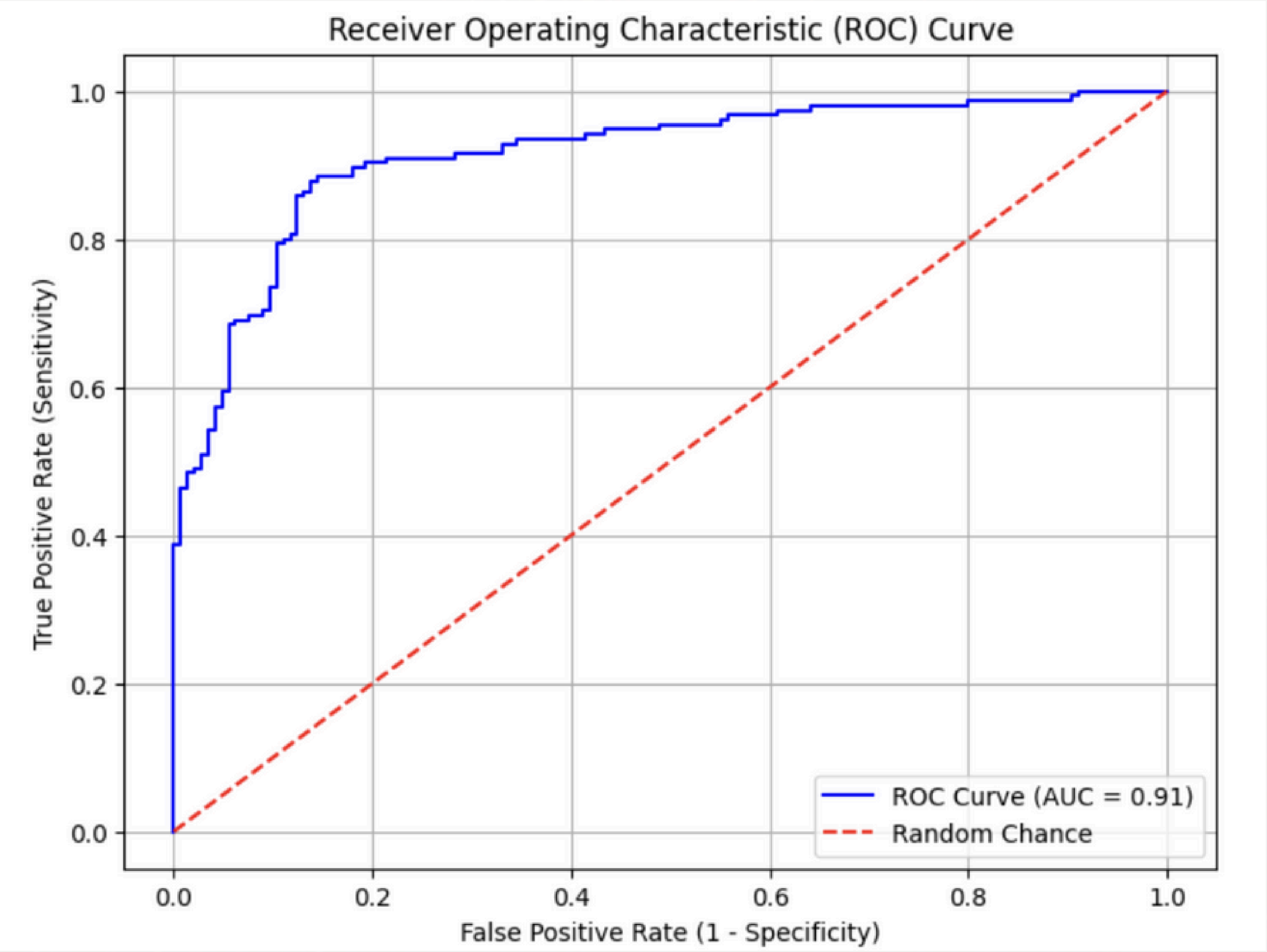
ROC curve is a graph used to check how well a binary classification model works. It helps us to understand how well the **model separates the positive cases** like people with a disease **from the negative cases** like people without the disease **at different threshold level.**
Does this by plotting **True Positive Rate (TPR) / Senstivity or Recall and False Positive Rate(FPR) / Specificity.**

Model Evaluation(Logistic Regression)

AUC ROC Curve in Machine Learning

ROC Curve: TPR vs FPR -- different thresholds. trade-off between the sensitivity and specificity of a classifier.

AUC (Area under the curve): meausre the area under the ROC curve. A higher AUC value indicates better model performance as it suggests a greater ability to distingush classes. If you pick one random Positive example and one random Negative example, what is the probability that your model gave a higher score to the Positive one?



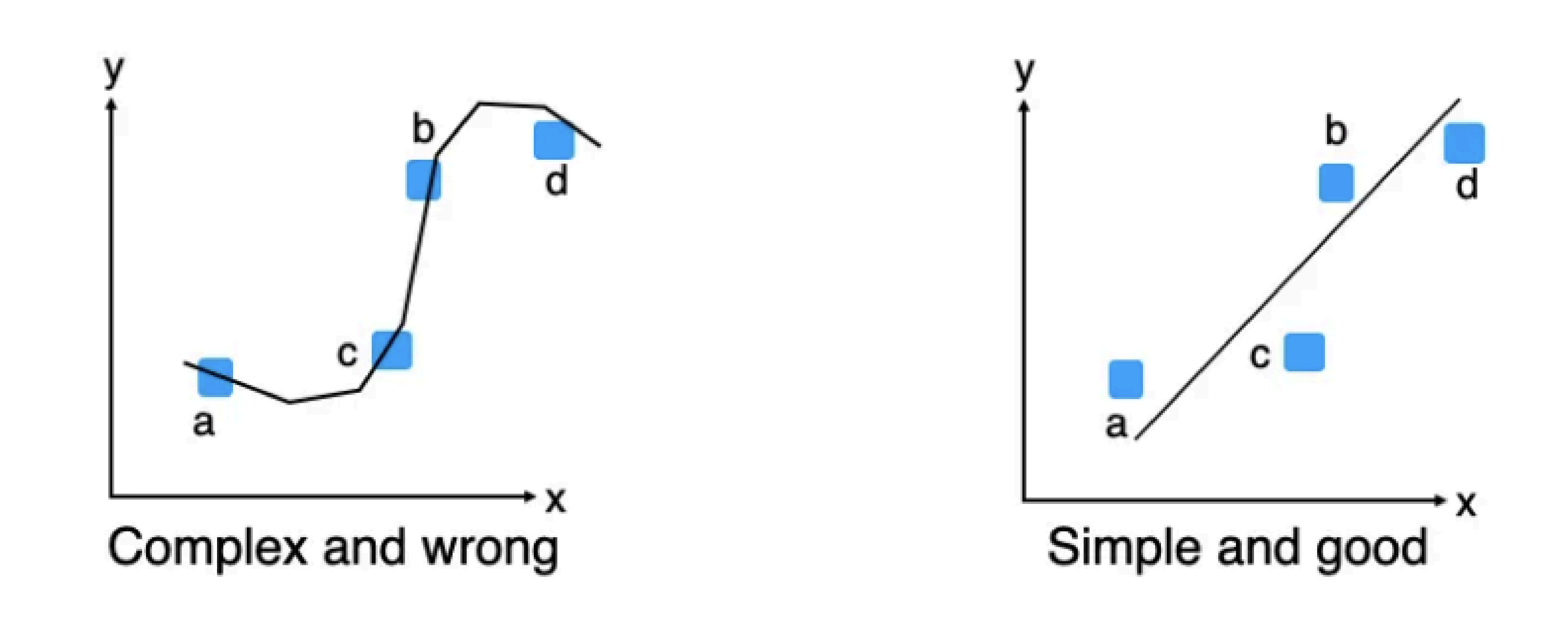
Model Evaluation

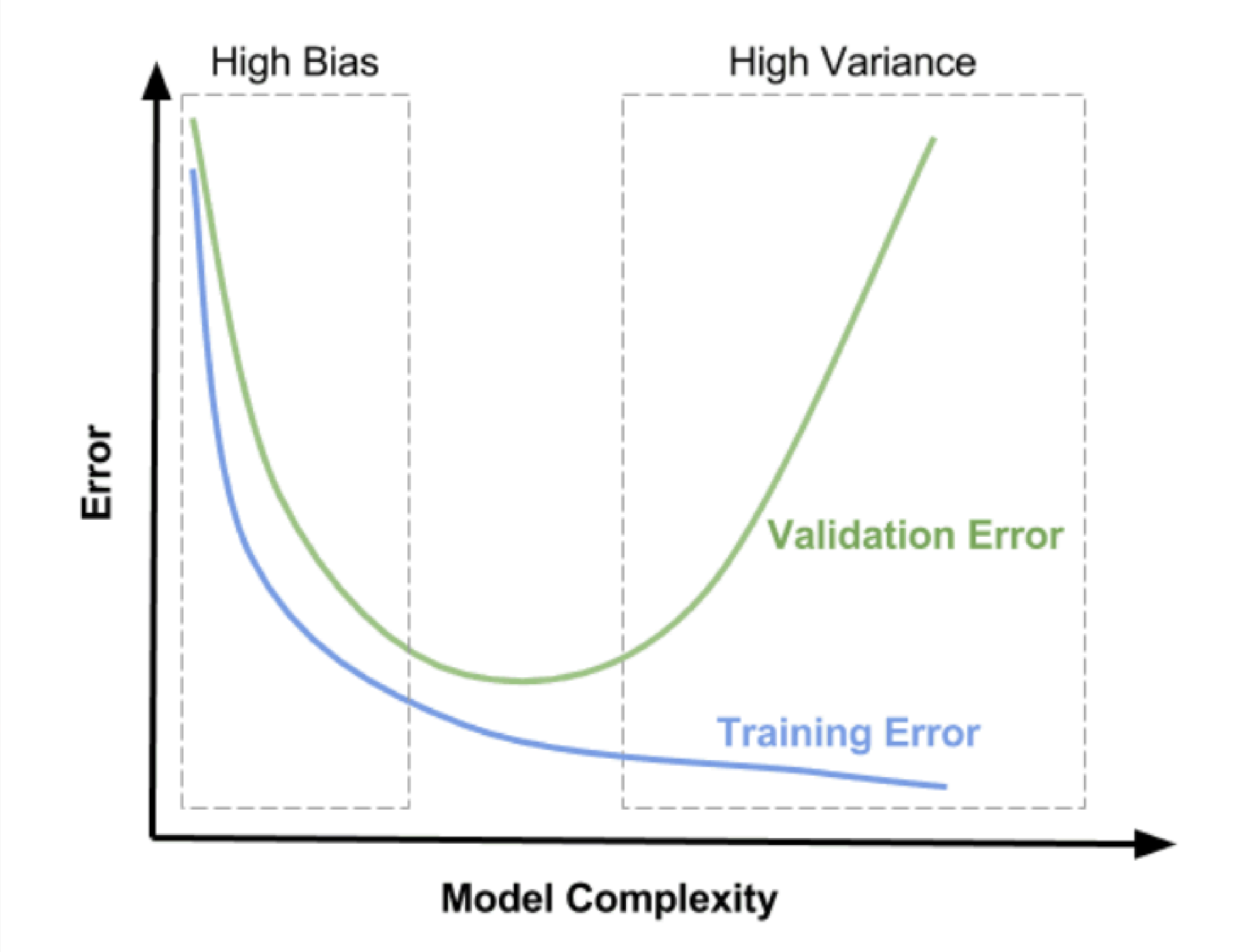
Bias and Variance

- Bias is an error from erroneous assumptions in the learning algorithm. High bias can cause an algorithm to miss the relevant relations between features (also called variables, dimensions) and target outputs ***(under-fitting)***.

- Variance is an error from sensitivity to small fluctuations in the training set. High variance can cause an algorithm to model the random noise in the training data, rather than the intended outputs (**over-fitting**).

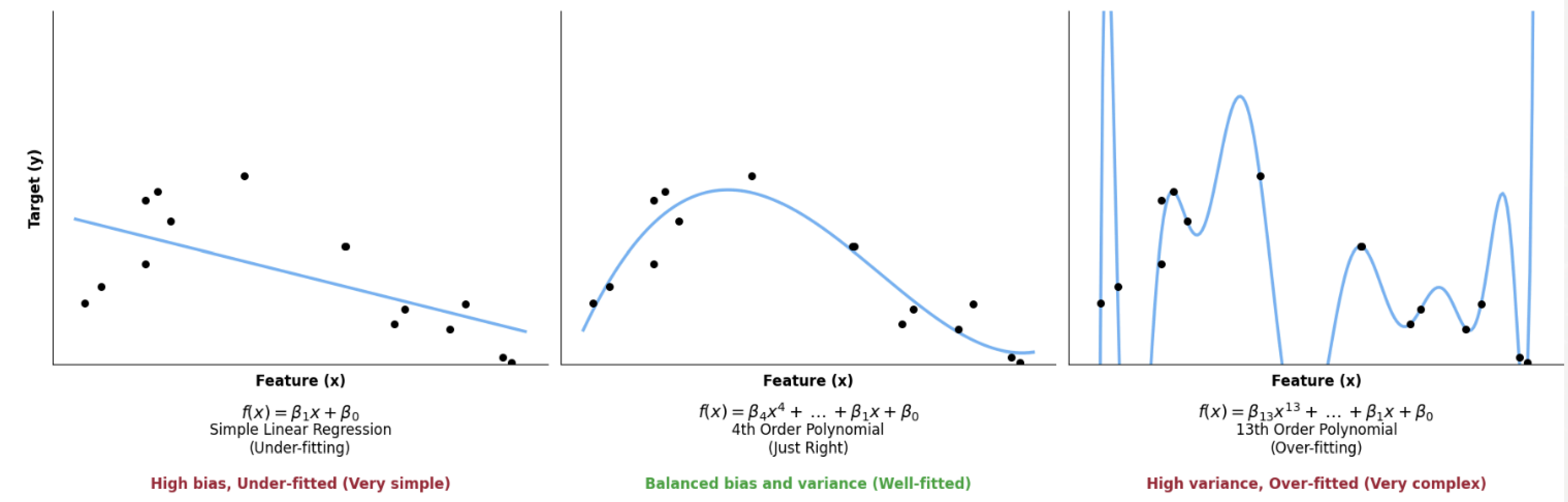
Model Evaluation





Use the model complexity to error plot to find the best model.

Regularization



Additional term in the loss function used to prevent the model from overfitting.

L2 Regularization (Ridge Regression)

$$\text{minimize: } RSS + Ridge = \sum_{i=1}^{n}\left(y_i - \left(\beta_0 + \sum_{j=1}^{p} x_{ij}\beta_j\right)\right)^2 + \alpha \sum_{j=1}^{p} \beta_j^2$$

where

β_j^2 is the squared coefficient for variable x_j.

$\sum_{j=1}^{n} \beta_j^2$ is the sum of these squared coefficients for every variable we have in our model. This does **not** include the **intercept** β_0 (this is a convention).

α is a constant for the *strength* of the **regularization parameter**. The higher this value, the greater the impact of this new component in the loss function.

L1 Regularization (Lasso Regression)

$$\text{minimize: } RSS + Lasso = \sum_{i=1}^{n}\left(y_i - \left(\beta_0 + \sum_{j=1}^{p} x_{ij}\beta_j\right)\right)^2 + \alpha\sum_{j=1}^{p}|\beta_j|$$

where

$|\beta\_j|$ is the absolute value of the β coefficient for variable x_j

- Lasso can result in sparse models with few coefficients, as some coefficients can become zero and eliminate from the model. This makes it easier to interpret the lasso model as compared to the ridge.
- Larger penalties result in coefficient values closer to zero, which is ideal for producing simpler models.
- Lasso performs both variable selection and regularization to enhance the prediction accuracy and interpretability of the model.

Office Hours

BUILD AN AI MODEL TO FORECAST TRENDS AND DETECT OPPORTUNITIES IN RETAIL BANKING

DRUCKER LEADERSHIP ACADEMY
It's an Art

Tangible

NAAMII