Introduction to Data Science

Lesson 5
Linear and Logistic Regression

Marija Stankova Medarovska, PhD
marija.s.medarovska@uacs.edu.mk

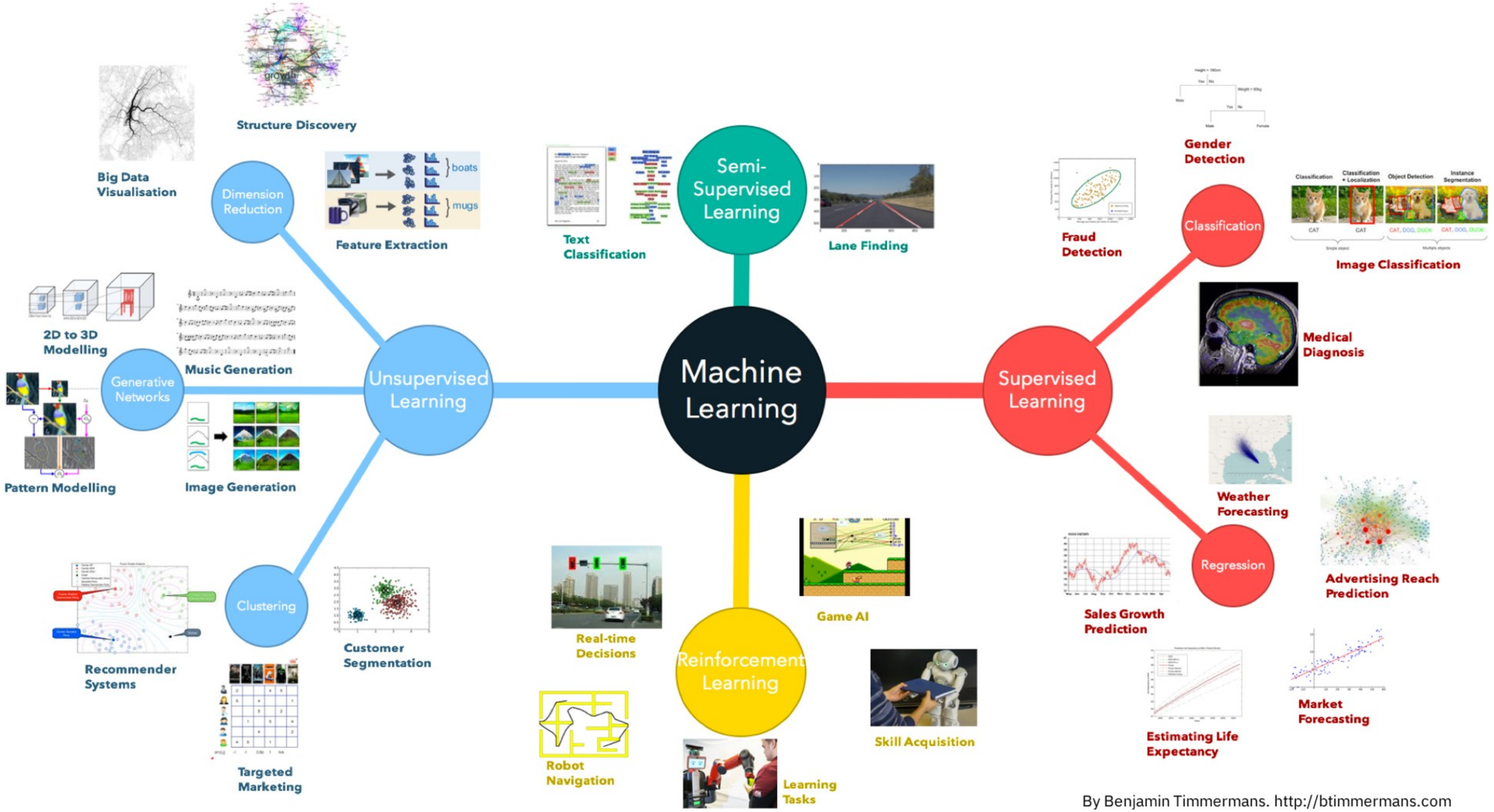
Choosing a machine learning algorithm

Which algorithm you use for a task will depend on:

- The type of problem you are trying to solve
- The type of data you have access to

Note that it's possible to have data ill-suited for the problem of interest. In this case, algorithms won't save you.

Choosing a machine learning algorithm



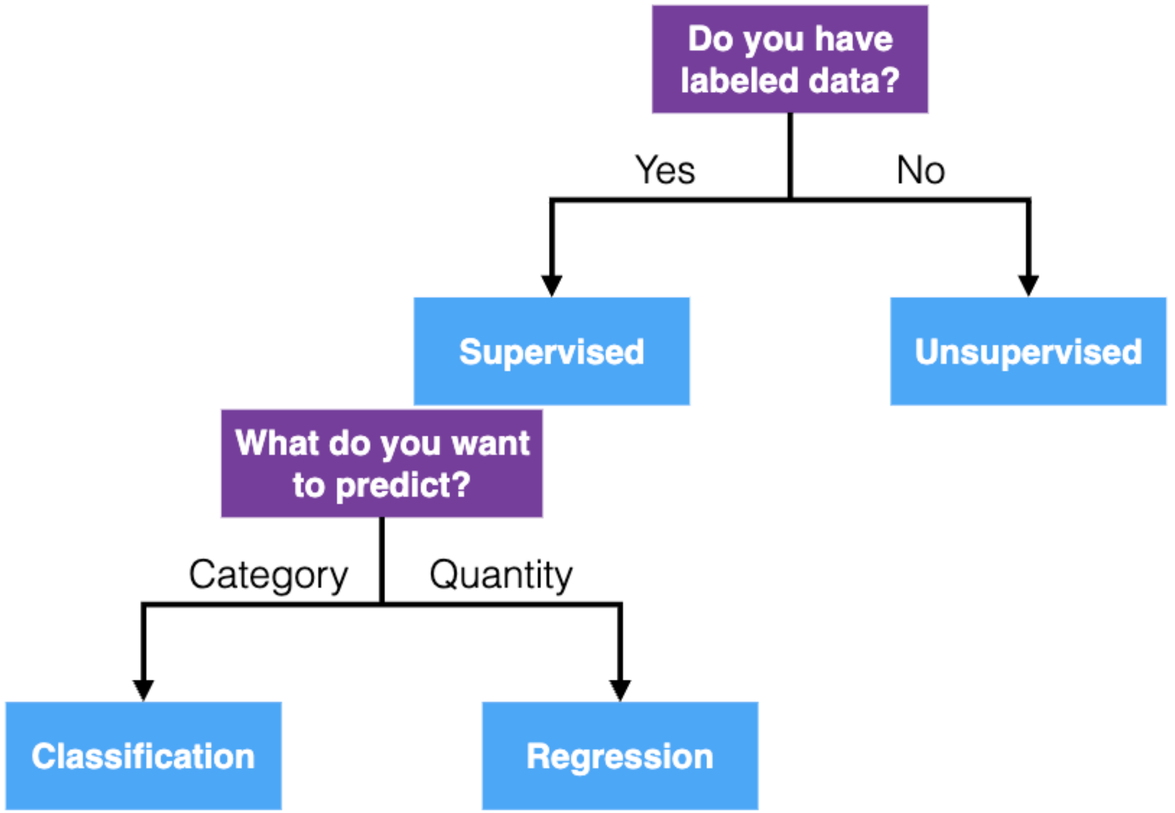
Recall: Supervised vs. Unsupervised Learning

Supervised learning:

- Builds a machine learning model to predict an output from inputs
- Goal: *generalize* to new data

Unsupervised learning:

- Learns structure from data without supervising output
- Goal: *understand relationships* between variables or among instances



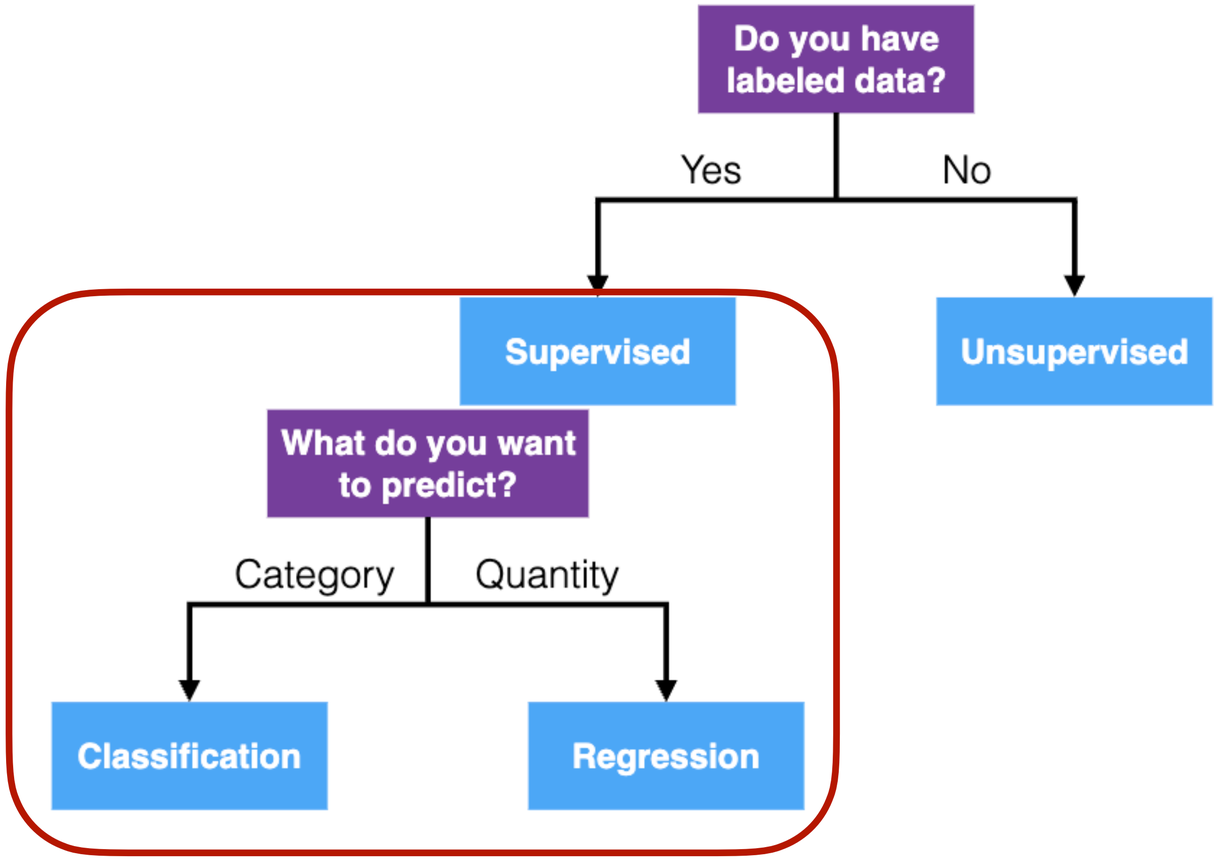
Recall: Supervised vs. Unsupervised Learning

Supervised learning:

- Builds a machine learning model to predict an output from inputs
- Goal: *generalize* to new data

Unsupervised learning:

- Learns structure from data without supervising output
- Goal: *understand relationships* between variables or among instances



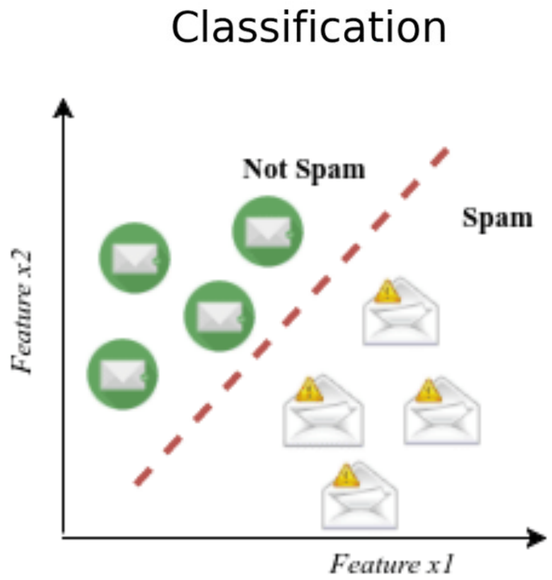
Recall: Classification vs. Regression

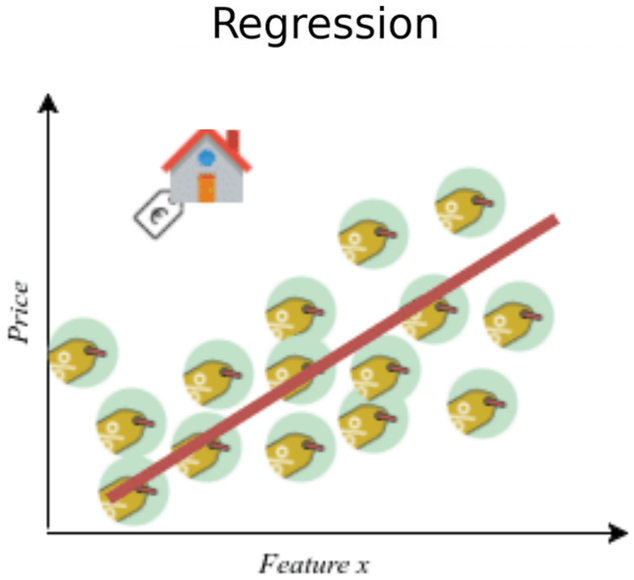
Classification:

- Output is qualitative (categorical)
- e.g. predict whether a credit card transaction is fraudulent (Yes/No)

Regression:

- Output is quantitative (numerical)
- e.g. predict the value of a house (258 000$)





Recall: Assessing Model Performance

There are a number of metrics used to assess model performance on supervised tasks (regression and classification).

Key point: We want to know how good predictions are when we apply our method to previously unseen data.

Why? The ability to generalize to unseen data is what makes these methods useful.*)*

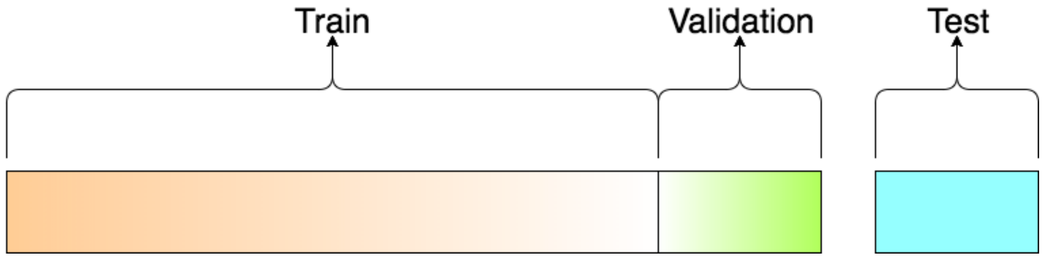


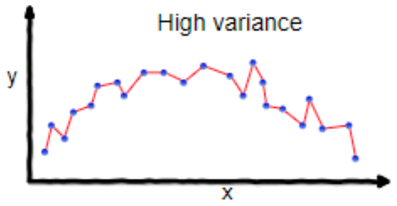
Figure: (1) Training data: instances used to learn the model; (2) Validation data: instances used to estimate error for parameter-tuning or model selection; (3)Test data: instances used to measure performance on unseen data (how well the model generalizes) - not available to the algorithm during any part of the learning process

Recall: Overfitting and underfitting

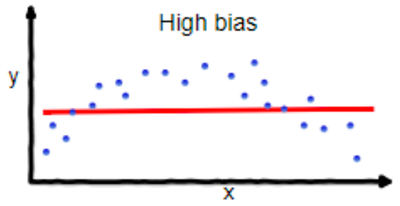
Generally, more complex methods have more variance and less bias.

Less complex methods have more bias and less variance.

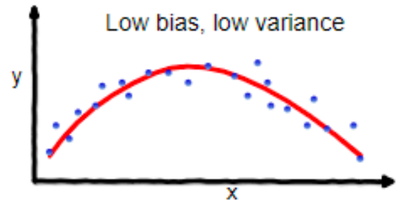
The best method for a task will balance the two types of error to achieve the lowest test error.



overfitting

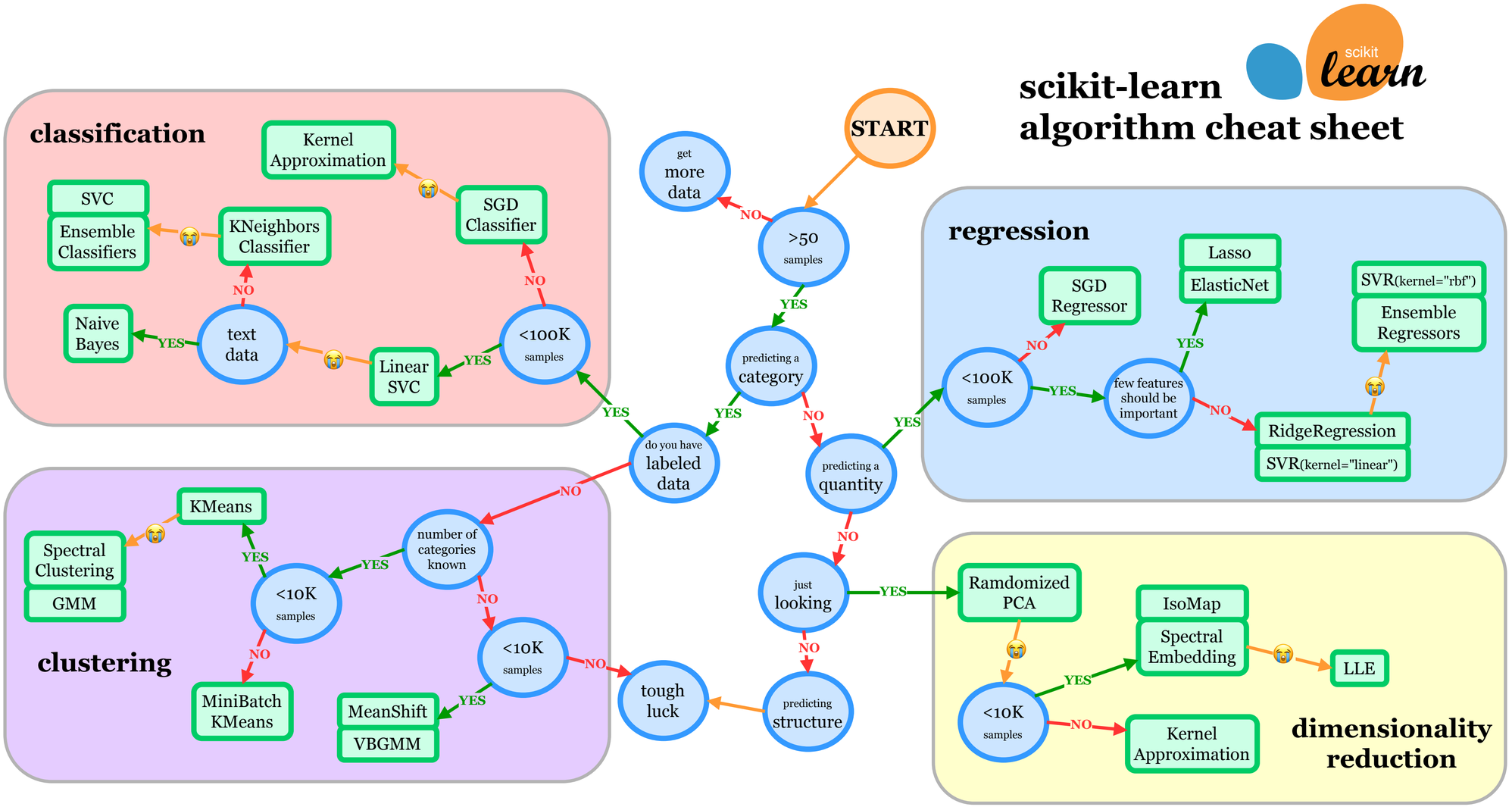


underfitting



Good balance

Choosing a machine learning algorithm*



[1] https://scikit-learn.org/1.5/machine_learning_map.html

Linear Regression

Linear Regression

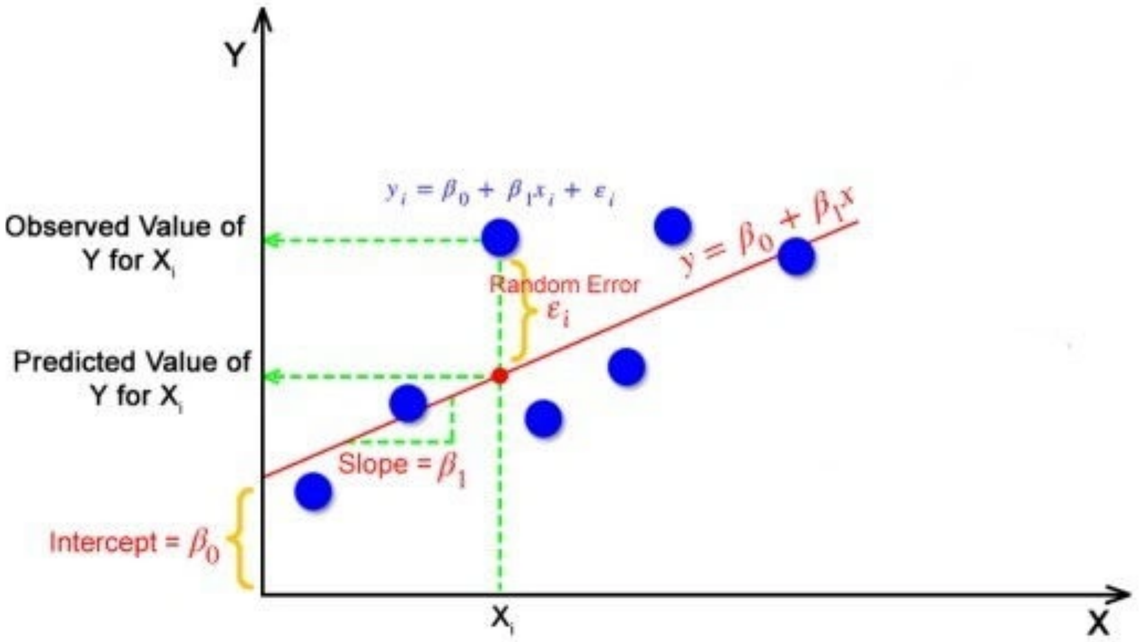
Linear regression is a supervised learning method, used to predict quantitative output values.

Simple linear regression: Predict a quantitative response Y on the basis of a single feature (predictor variable) X.

Assumes there is an approximately linear relationship between X and Y.

$$Y \approx \beta_0 + \beta_1 X$$

β_0 and β_1 are the *coefficients* (or *parameters*) of the linear model. In this case they represent the intercept and slope terms of a line.



Linear Regression

Estimate βs using training data

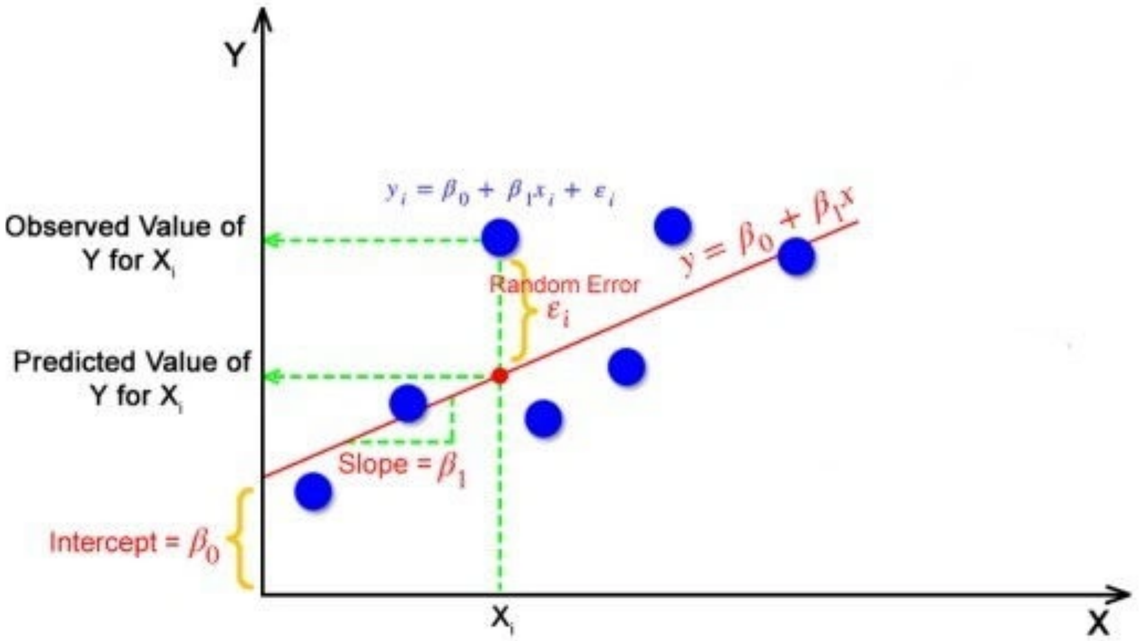
Once βs are estimated, we denote them using "hats".

For a particular realization of X, aka $X = x$, the predicted output is denoted "y-hat":

$$\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x$$

Goal: Pick β_0 and β_1, such that the model is a good fit to the training data.

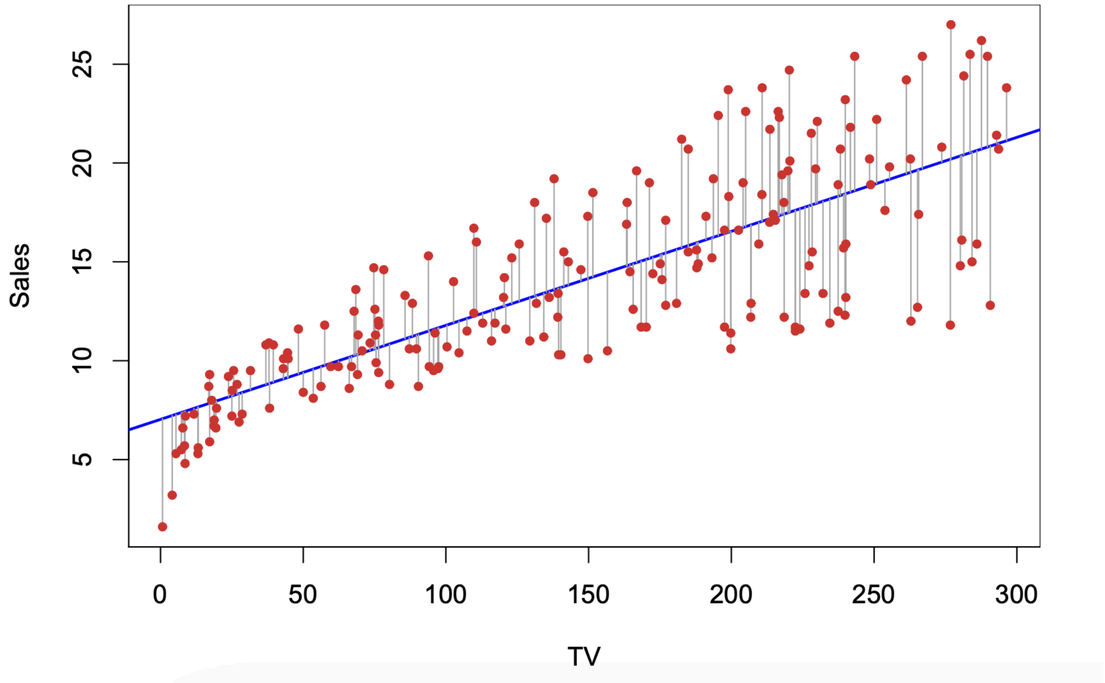
$$y^{(i)} \approx \hat{\beta}_0 + \hat{\beta}_1 x^{(i)}, \quad i = 1, \ldots, n$$



Linear Regression

Two related questions:

- How do we estimate the coefficients? (aka “fit the model”)
- What is a “good fit” to the data?



Example: The Advertising data set. The plot displays sales, in thousands of units, as a function of TV budget, in thousands of dollars, for 200 different markets

Least Squares

A common method to find optimal parameters (coefficients) for the regression model is using *least squares.*

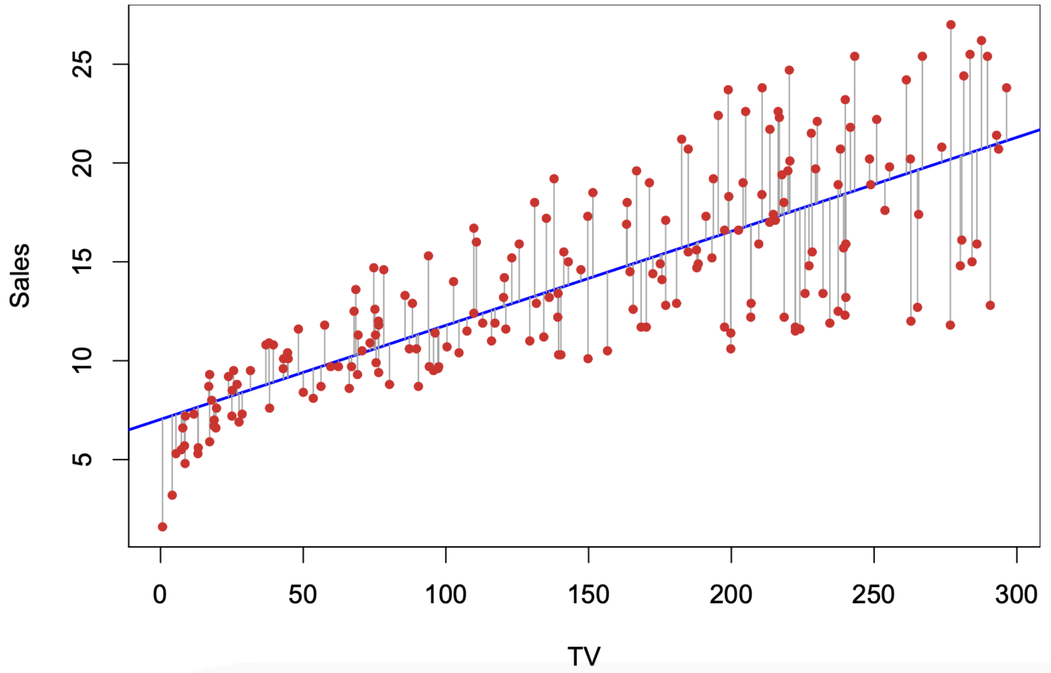
Residual is the difference between the i-th observed response value and the i-th response value that is predicted by our linear model

Residual for the i-th sample:

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

Residual sum of squares (RSS):

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



Example: In the Advertising data set plot. each grey line segment represents a residual.

Least Squares

The model fit using least squares finds β_0 and β_1 that minimize the RSS.

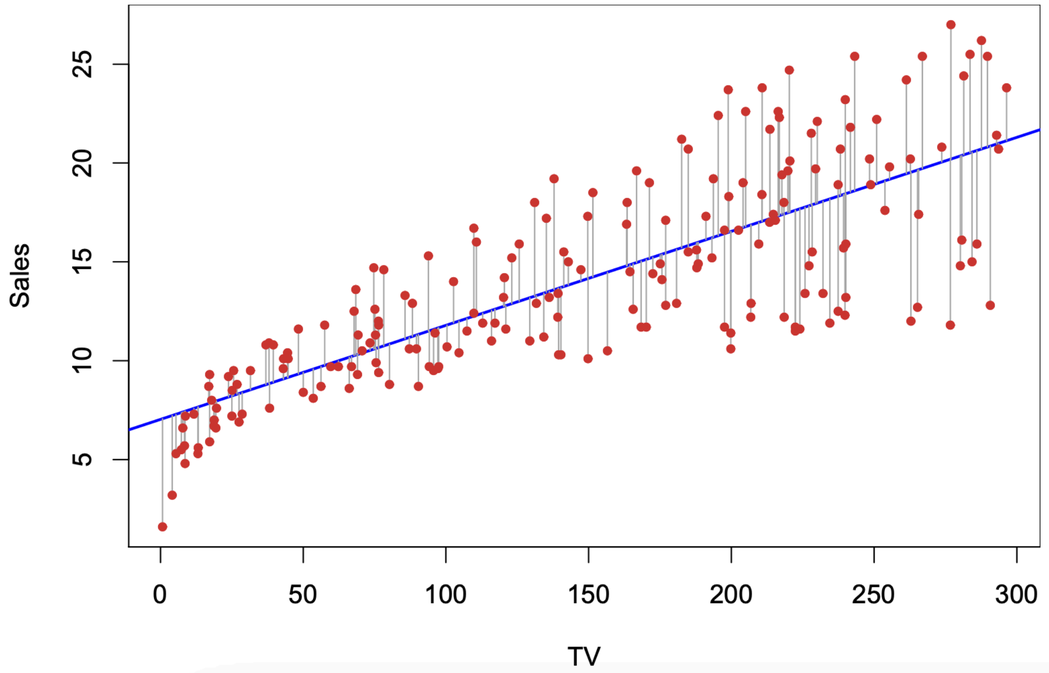
$$\hat{\beta}_0, \hat{\beta}_1 = \arg\min_{\beta_0, \beta_1} \left[\sum_{i=1}^{n} \left(y^{(i)} - \left(\beta_0 + \beta_1 x^{(i)} \right) \right)^2 \right]$$

Recall that the extrema of a function can be found by setting its derivative to zero, and verified to be minima via the second derivative

Least squares coefficient estimates for simple linear regression

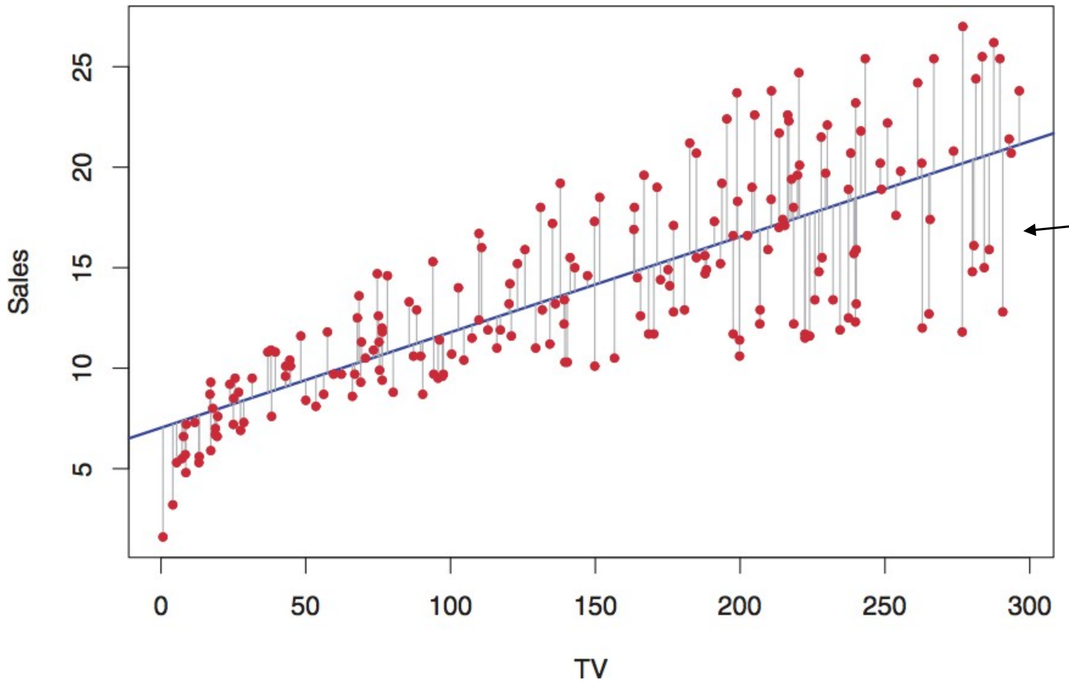
$$\hat{\beta}_1 = \frac{\sum_{i=1}^{n} \left(x^{(i)} - \bar{x} \right) \left(y^{(i)} - \bar{y} \right)}{\sum_{i=1}^{n} \left(x^{(i)} - \bar{x} \right)^2}, \qquad \hat{\beta}_0 = \bar{y} - \hat{\beta}_1 \bar{x}$$

sample means



Example: The Figure displays the simple linear regression fit to the Advertising data, where $\hat{\beta}_0 = 7.03$ and $\hat{\beta}_1 = 0.0475$. In other words, according to this approximation, an additional $1,000 spent on TV advertising is associated with selling approximately 47.5 additional units of the product

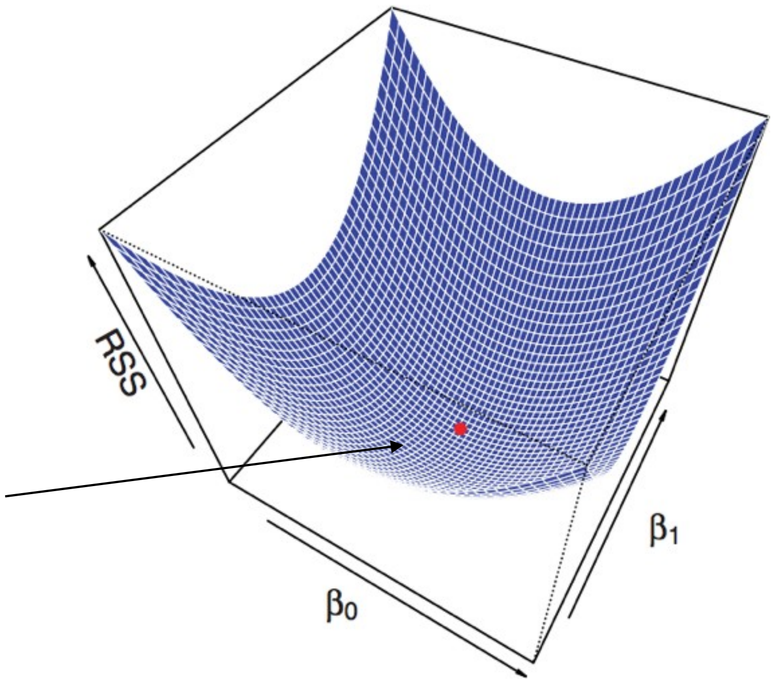
Least Squares



$$\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x$$

Residual sum of squares (RSS) is the sum of the squares of all vertical gray lines.

As we vary the βs, RSS changes. Least squares finds βs that minimize RSS.



How good is the model fit?

In order to assess the accuracy of the model, we can use several metrics:

- *Mean Squared Error (MSE)*

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

higher MSE means worse fit

- *Root Mean Squared Error (RMSE)*

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

higher RMSE means worse fit

- *Mean Absolute Error (MAE)*

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

higher MAE means worse fit

How good is the model fit?

In order to assess the accuracy of the model, we can use several metrics:

- *Residual standard error* (RSE)

$$\mathrm{RSE} = \sqrt{\frac{1}{n-2}\mathrm{RSS}} = \sqrt{\frac{1}{n-2}\sum_{i=1}^{n}\left(y^{(i)} - \hat{y}^{(i)}\right)^2}$$

higher RSE means worse fit

- R^2 statistics

$$R^2 = \frac{\mathrm{TSS} - \mathrm{RSS}}{\mathrm{TSS}} = 1 - \frac{\mathrm{RSS}}{\mathrm{TSS}}$$

TSS is the "total sum of squares" $\sum_{i=1}^{n}(y^{(i)} - \bar{y})^2$

higher R^2 means better fit

Multiple Linear Regression

What if our dataset contains multiple input dimensions X_j?

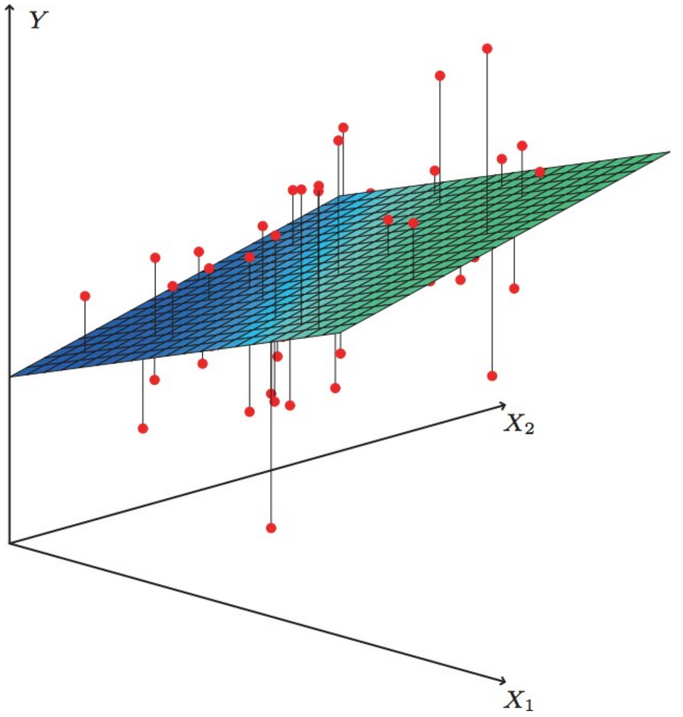
Predict the response variable using more than one feature (predictor variable):

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p + \epsilon$$

Here, X_j represents the j-th feature.

We interpret β_j as the average effect on Y of one unit increase in X_j, holding all other predictors fixed.

The parameters are estimated using the same least squares approach that we saw in the context of simple linear regression.



Example: In a three-dimensional setting, with two input variables and one response, the least squares regression line becomes a plane. The plane is chosen to minimize the sum of the squared vertical distances between each observation (shown in red) and the plane.

Additivity of features

Linear regression makes several highly restrictive assumptions that are often violated in practice.

One of them, *additivity* means each feature X_j affects Y independently of the value of other features

If this is not true, we can extend linear regression by including *interaction effects* between multiple variables.

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \beta_3 X_1 X_2 + \epsilon$$

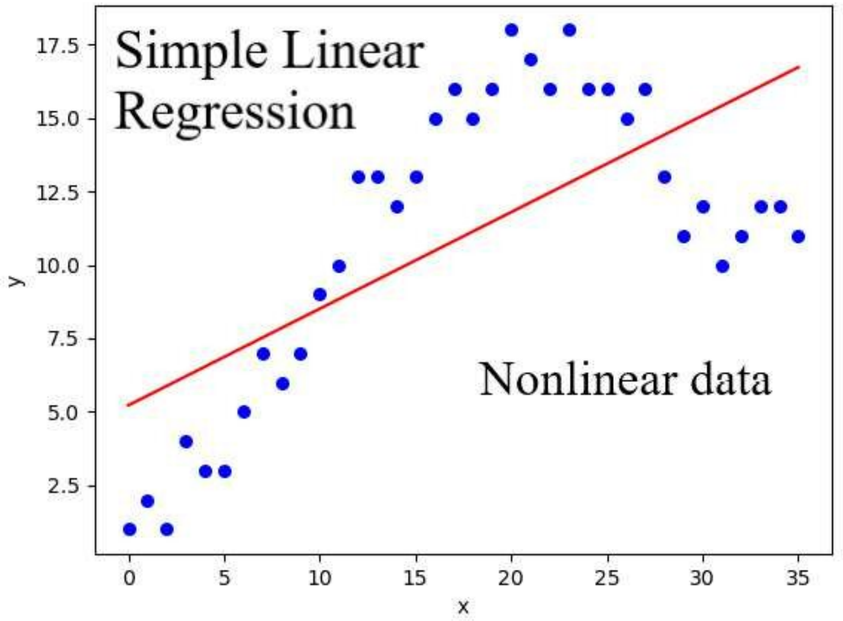
main term

interaction term

Nonlinear data

Furthermore, the linear regression model assumes that there is a linear (straight-line) relationship between the input and the response.

If the true relationship is far from linear, then virtually all of the conclusions that we draw from the fit are suspect. In addition, the prediction accuracy of the model can be significantly reduced.



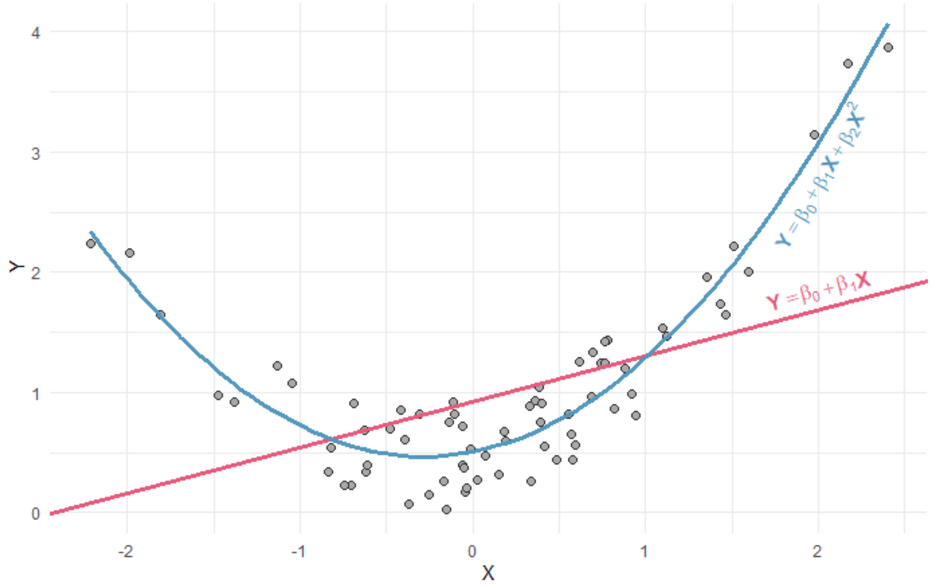
Example: Linear regression against nonlinear data

Nonlinear data

A simple way to extend linear regression to model non-linear relationships is via *polynomial regression*.

$$Y = \beta_0 + \beta_1 X_1 + \beta_2 X_1^2 + \epsilon$$

This is still a linear model; it can be fit via least squares.



Example: Linear vs quadratic model to fit nonlinear data

Collinearity

Collinearity refers to the situation in which two or more features are highly correlated with each other.

It can reduce the accuracy of the models and pose problems with model interpretability

Solutions include dropping one of the problematic variables from the regression or combining the collinear variables together into a single feature

If the collinearity exist between three or more variables, its called *multicollinearity*

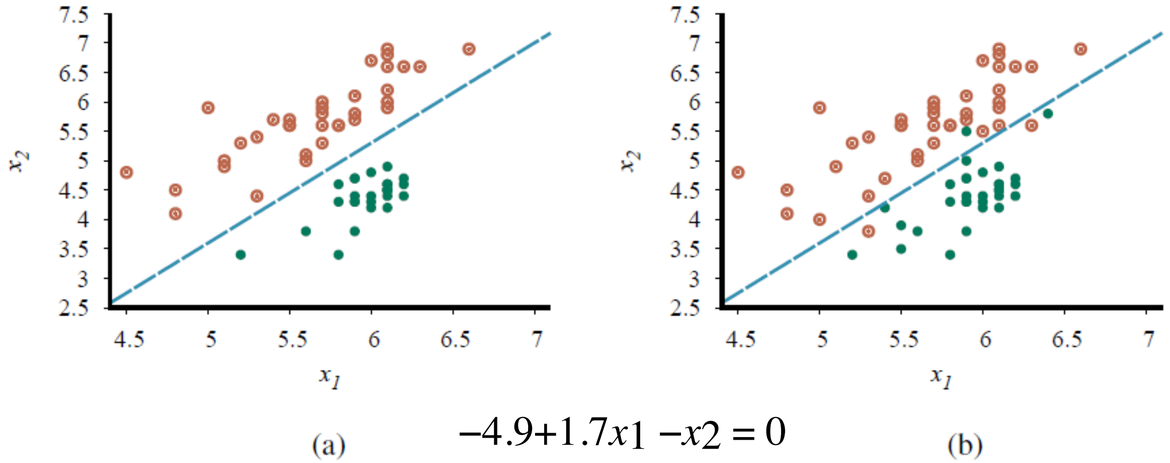
Example: Correlation matrix can be used to detect collinearity. For multicollinearity, the variance inflation factor (VIF) can be calculated. *As a rule of thumb, a VIF value that exceeds 5 or 10 indicates a problematic amount of collinearity.*

Linear classification with a hard threshold

While linear regression is primarily designed for continuous outcomes, it can be used fo classification tasks (better suited algorithms are available for this)

In case of classification, a *decision boundary* is a line (or a surface, in higher dimensions) that separates the classes.

The model will output a continuous value (not necessarily 0 or 1), which is the predicted output



Example: *(a) Plot of two seismic data parameters, body wave magnitude x_1 and surface wave magnitude x_2, for earthquakes (open orange circles) and nuclear explosions (green circles) occurring between 1982 and 1990 in Asia and the Middle East b) The same domain with more data points. The earthquakes and explosions are no longer linearly separable.*
The explosions, which we want to classify with value 1, are below and to the right of this line; they are points for which $-4.9 + 1.7x_1 - x_2 > 0$, while earthquakes have $-4.9 + 1.7x_1 - x_2 < 0$.

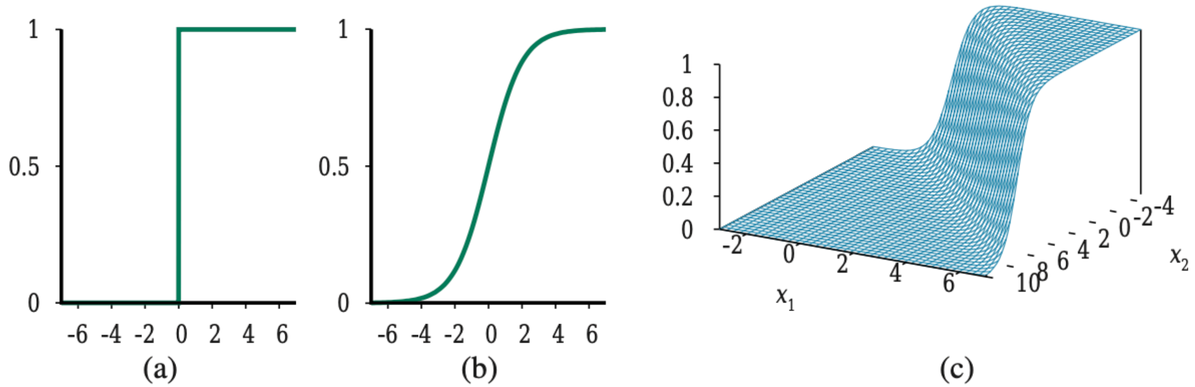
Logistic Regression

Logistic regression

Logistic regression is a supervised learning method used for classification

Unlike linear regression, which predicts continuous values, logistic regression predicts the probability that a given input belongs to a particular class.

$$Pr(Y \text{ belongs to class } 1 \,|X)$$



Example: *(a) The hard threshold function Threshold(z) with 0/1 output. Note that the function is non differentiable at 0. It also creates completely confident prediction of 1 or 0, even for examples that are very close to the boundary*
(b) The logistic function, also known as the sigmoid function. Notice that the output, being a number between 0 and 1, can be interpreted as a probability of belonging to the class labeled 1.

Logistic regression

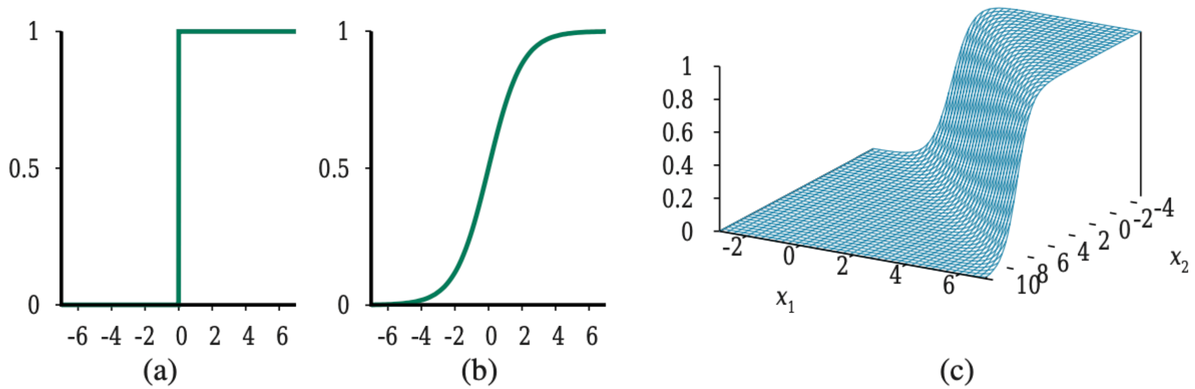
Logistic regression uses the logistic function (or sigmoid function) to constrain the output between 0 and 1, which makes it suitable for predicting probabilities.

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

The logistic regression model is:

$$Pr(Y=1|X) = \sigma(\beta_0 + \beta_1 X) = \frac{e^{\beta_0+\beta_1 X}}{1+e^{\beta_0+\beta_1 X}}$$

This can be extended to multiple logistic regression, when there are multiple input variables.



Example: *(a) The hard threshold function Threshold(z) with 0/1 output. Note that the function is non differentiable at 0. It also creates completely confident prediction of 1 or 0, even for examples that are very close to the boundary*
(b) The logistic function, also known as the sigmoid function. Notice that the output, being a number between 0 and 1, can be interpreted as a probability of belonging to the class labeled 1.

Maximum likelihood estimation

Recall that for linear regression, βs are estimated using least squares on the training data.

For logistic regression, there is no closed form solution for βs obtainable by taking the derivative and setting to zero.

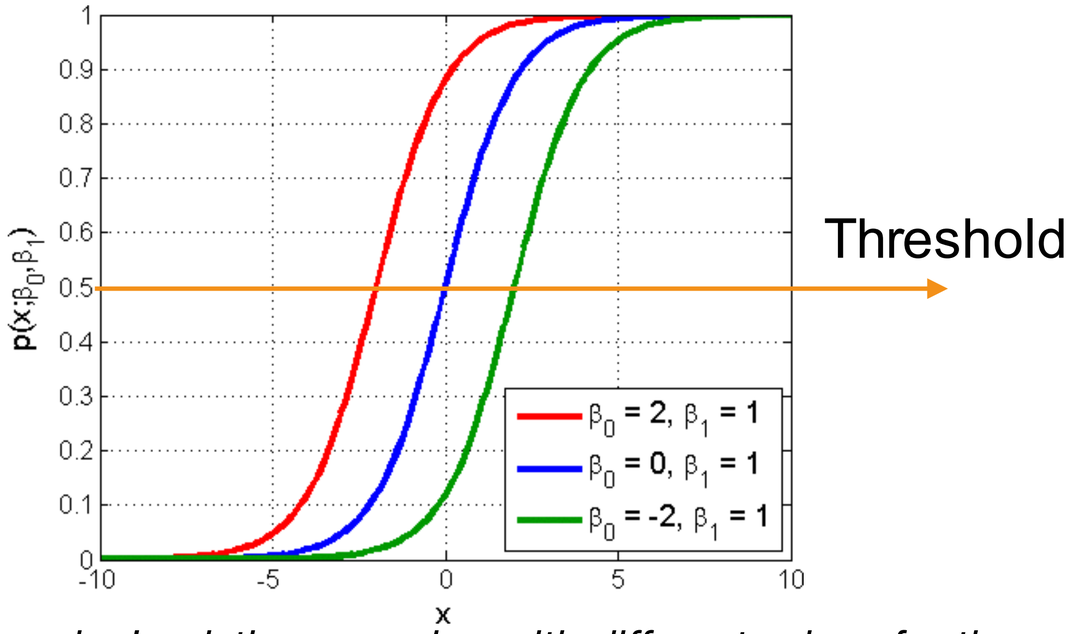
Instead, βs are estimated using *maximum likelihood estimation*.

Likelihood function:

$$\ell(\beta_0, \beta_1) = \prod_{i:y_i=1} p(x_i) \prod_{i':y_{i'}=0} (1 - p(x_{i'})).$$

The estimates β_0 and β_1 are chosen to *maximize* this likelihood function.

For large datasets *gradient descent* is more suitable (in the next lectures).



Example: *Logistic regression with different values for the coefficients. The threshold is the value against which this probability is compared to determine the class label. E.g. using a threshold of 0.5, for output values larger than 0.5, we predict class 1 and for values small than 0.5 class 0.*

How good is the model fit?

Classifier performance can be summarized in a table called the *confusion matrix*.

Good performance is when TP, TN are large and FP, FN are small.

Can be computed for training, validation, and test sets. Test set informs you about model generalizability.

		Predicted	
		Negative **(N)** -	Positive **(P)** +
Actual	Negative -	True Negative **(TN)**	**False Positive (FP) Type I Error**
	Positive +	**False Negative (FN) Type II Error**	True Positive **(TP)**

Example: *The layout of a 2 × 2 confusion matrix for a classifier with 2 class labels showing the names of the correct predictions (main diagonal) and errors (off-diagonal) entries.*

How good is the model fit?

Some commonly used metrics for evaluating classifiers, which are fundamentally summaries of the confusion matrix:

Accuracy is the count of correct decisions divided by the total number of decisions made. It performs poorly on imbalanced datasets (TP + TN) / (TP + FP + TN + FN).

Precision is TP / (TP + FP), which is the accuracy over the cases predicted to be positive.

Recall (or Sensitivity or True Positive Rate) measures the proportion of actual positives that are correctly identified by the model TP / (TP + FN).

Specificity (or True Negative Rate) measures the proportion of actual negatives that are correctly identified by the model TN / (TN + FP).

F-measure (F1 score) = 2 * (precision * recall) / (precision + recall)

		Predicted	
		Negative **(N)** -	Positive **(P)** +
Actual	Negative -	True Negative **(TN)**	**False Positive (FP) Type I Error**
	Positive +	**False Negative (FN) Type II Error**	True Positive **(TP)**

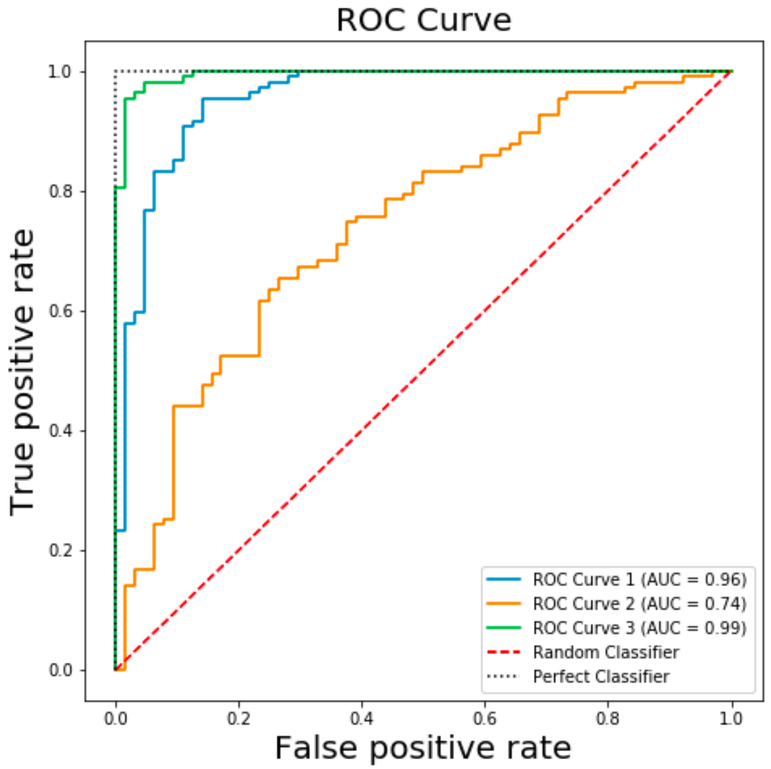
Example: *The layout of a 2 × 2 confusion matrix for a classifier with 2 class labels showing the names of the correct predictions (main diagonal) and errors (off-diagonal) entries.*

How good is the model fit?

The *Receiver Operating Characteristic (ROC)* curve illustrates the performance of a classifier as its decision threshold for converting is varied. It plots the true positive rate TP / (TP + FN) against the false positive rate FP / (FP + TN) at various threshold settings.

Area Under the Curve (AUC) is the size of the area under the ROC curves and quantifies the overall performance of the classifier. It represents the probability that the classifier will rank a randomly chosen positive instance higher than a randomly chosen negative instance.

decision threshold = from which value to convert a predicted probability from the classifier to a positive label.



The ROC curves and AUC scores of several example classifiers. AUC values range from 0 to 1, with higher values indicating better performance. An AUC of 0.5 suggests random performance, while an AUC of 1 represents perfect discrimination.

Appendix