

# Module: Adv. Machine Learning

## Live Session-02

### Agenda:

Tree-based Regression Model

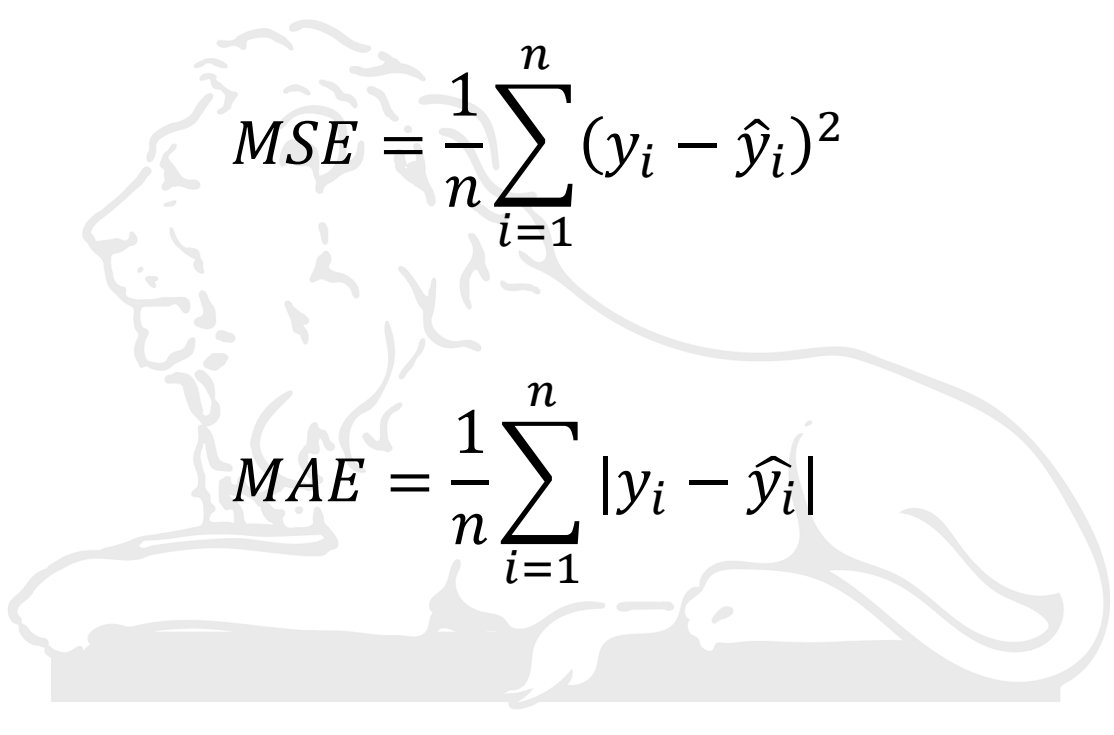
Dealing with Imbalanced

WWW-→ What-When-Why

Bias-Variance Tradeoff

# DT Regression?

DT Regression is similar to DT Classification, however we use **Mean Square Error** (MSE, default) or **Mean Absolute Error** (MAE) instead of *cross-entropy* or *Gini impurity* to determine splits.


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

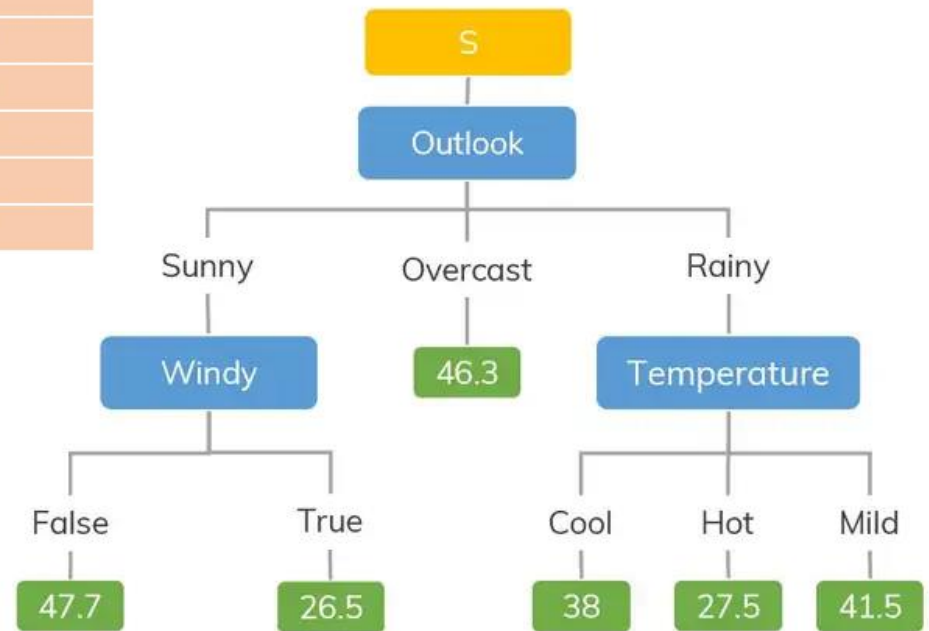
$$MAE = \frac{1}{n} \sum_{i=1}^n |y_i - \hat{y}_i|$$

# DT Regression: Example

Outlook	Temperature	Humidity	Windy	Hours Played
Rainy	Hot	High	False	26
Rainy	Hot	High	True	30
Overcast	Hot	High	False	48
Sunny	Mild	High	False	46
Sunny	Cool	Normal	False	52
Sunny	Cool	Normal	True	23
Overcast	Cool	Normal	True	43
Rainy	Mild	High	False	35
Rainy	Cool	Normal	False	38
Sunny	Mild	Normal	False	48
Rainy	Mild	Normal	True	48
Overcast	Mild	High	True	52
Overcast	Hot	Normal	False	44
Sunny	Mild	High	True	30

# DT Regression: Example

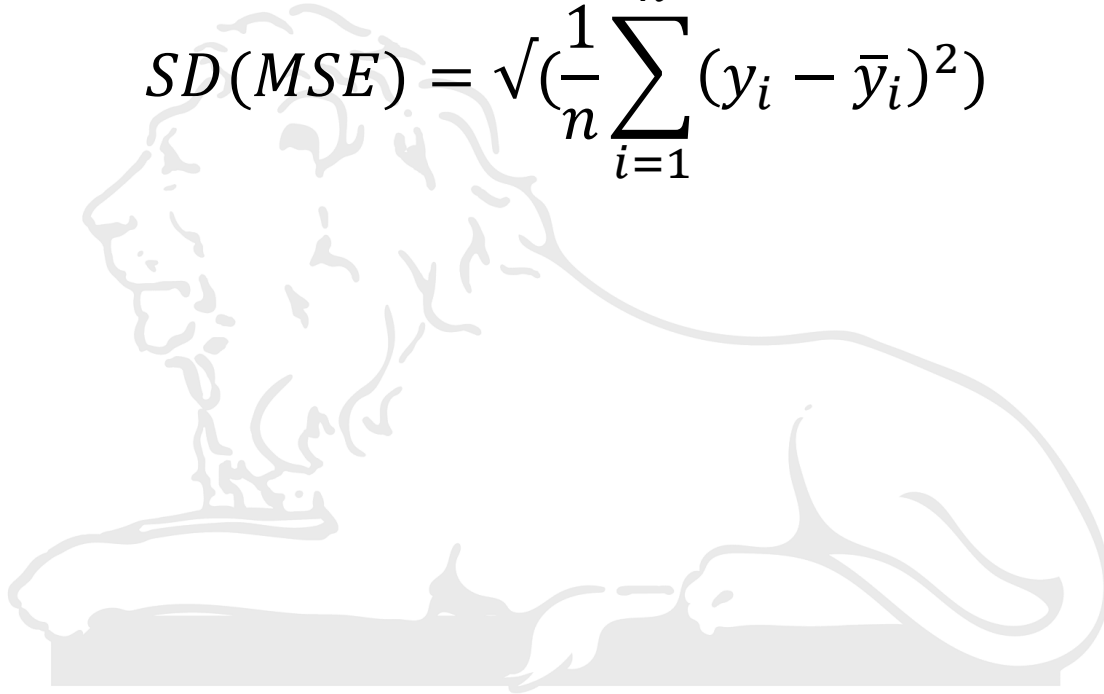
Outlook	Temperature	Humidity	Windy	Hours Played
Rainy	Hot	High	False	26
Rainy	Hot	High	True	30
Overcast	Hot	High	False	48
Sunny	Mild	High	False	46
Sunny	Cool	Normal	False	52
Sunny	Cool	Normal	True	23
Overcast	Cool	Normal	True	43
Rainy	Mild	High	False	35
Rainy	Cool	Normal	False	38
Sunny	Mild	Normal	False	48
Rainy	Mild	Normal	True	48
Overcast	Mild	High	True	52
Overcast	Hot	Normal	False	44
Sunny	Mild	High	True	30



# Algorithm: Step-1

- Calculate the **Standard Deviation** (*SD*) of the current node (let's say *S*, parent node) by using MSE or MAE.

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

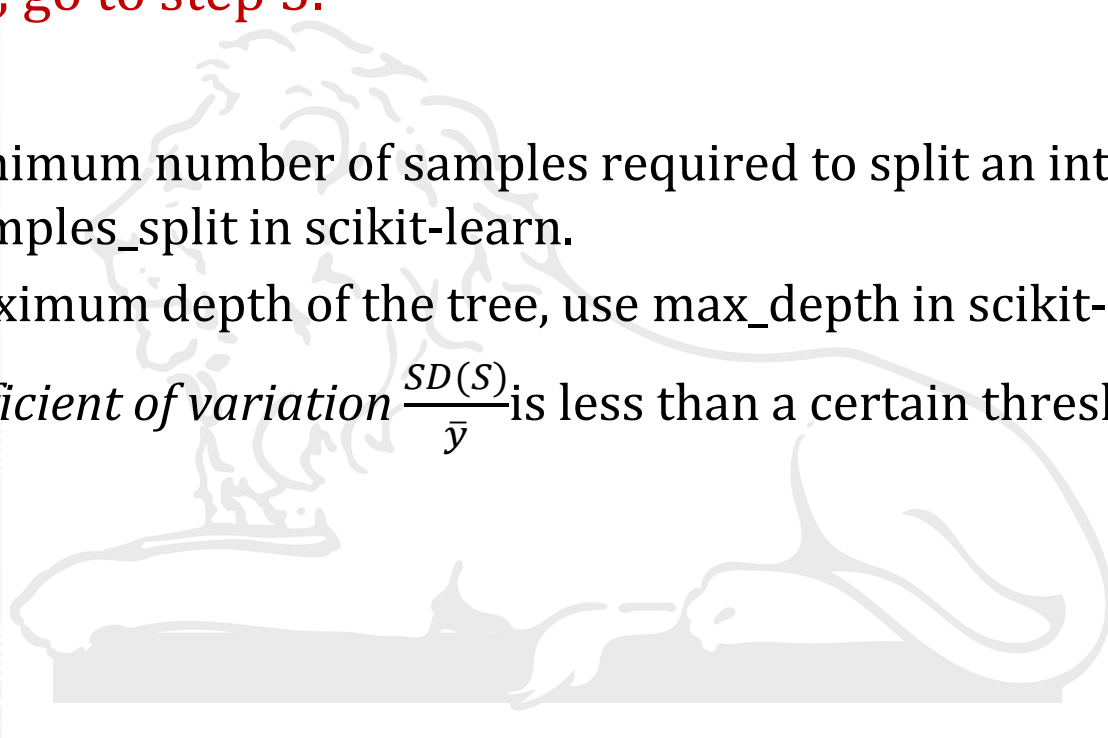


# Algorithm: Step-2

- ▶ Check the **stopping conditions** (we don't need to make any split at this node) to stop the split and this node becomes a leaf node. Otherwise, go to step 3.

## Criteria:

- The minimum number of samples required to split an internal node, use `min_samples_split` in scikit-learn.
- The maximum depth of the tree, use `max_depth` in scikit-learn.
- Its *coefficient of variation*  $\frac{SD(S)}{\bar{y}}$  is less than a certain threshold.



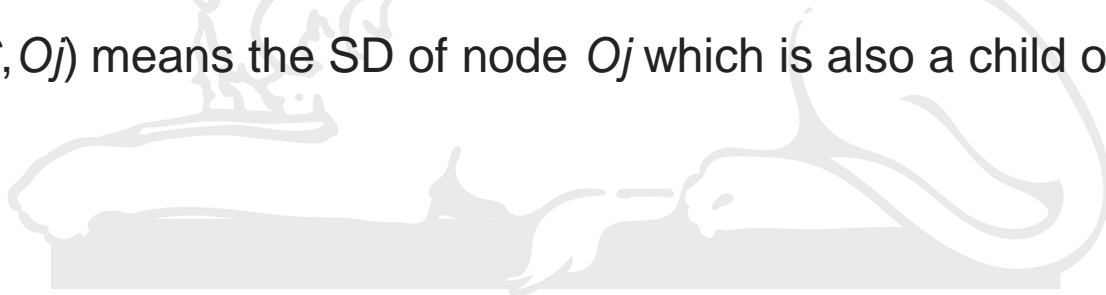
# Algorithm: Step-3

- Calculate the **Standard Deviation Reduction (SDR)** after splitting node  $S$  on each attribute (for example, consider attribute  $O$ ). The attribute w.r.t. the biggest SDR will be chosen!

$$\underbrace{SDR(S, O)}_{\text{Standard Deviation Reduction}} = \underbrace{SD(S)}_{\text{SD before split}} - \underbrace{\sum_j P(O_j|S) \times SD(S, O_j)}_{\text{weighted SD after split}}$$

where  $j$  number of different properties in  $O$ , and  $P(O_j)$  is the probability of  $O_j$  in  $O$ .

Note that,  $SD(S, O_j)$  means the SD of node  $O_j$  which is also a child of node  $S$



## Algorithm: Step-3..

- ▶ Calculate the **Standard Deviation Reduction (SDR)** after splitting node  $S$  on each attribute (for example, consider attribute  $O$ ). The attribute w.r.t. the biggest SDR will be chosen!

$$\underbrace{SDR(S, O)}_{\text{Standard Deviation Reduction}} = \underbrace{SD(S)}_{\text{SD before split}} - \underbrace{\sum_j P(O_j|S) \times SD(S, O_j)}_{\text{weighted SD after split}}$$



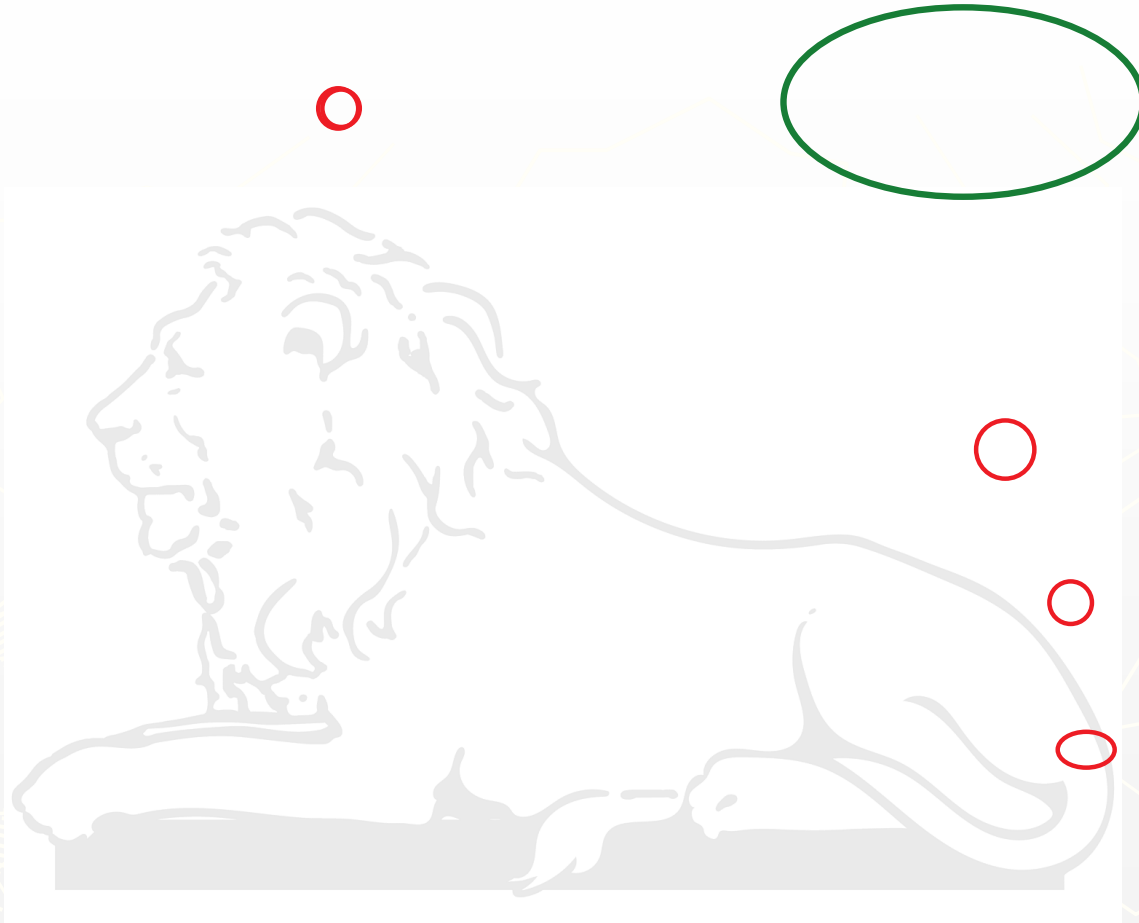


## Algorithm: Step-3..

$$\underbrace{SDR(S, O)}_{\text{Standard Deviation Reduction}} = \underbrace{SD(S)}_{\text{SD before split}} - \underbrace{\sum_j P(O_j|S) \times SD(S, O_j)}_{\text{weighted SD after split}}$$



## Algorithm: Step-3..



## Algorithm: Step-4

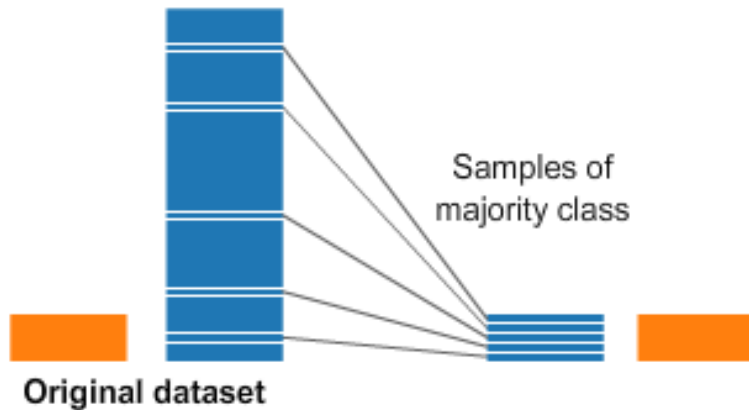
- ▶ After splitting, we have new child nodes. Each of them becomes a new parent node in the next step. Go back to **step-1**



# Imbalanced datasets:

Imbalanced datasets are a special case for classification problem where the class distribution is not uniform among the classes. Typically, they are composed by two classes: The majority (negative) class and the minority (positive) class

**Undersampling**

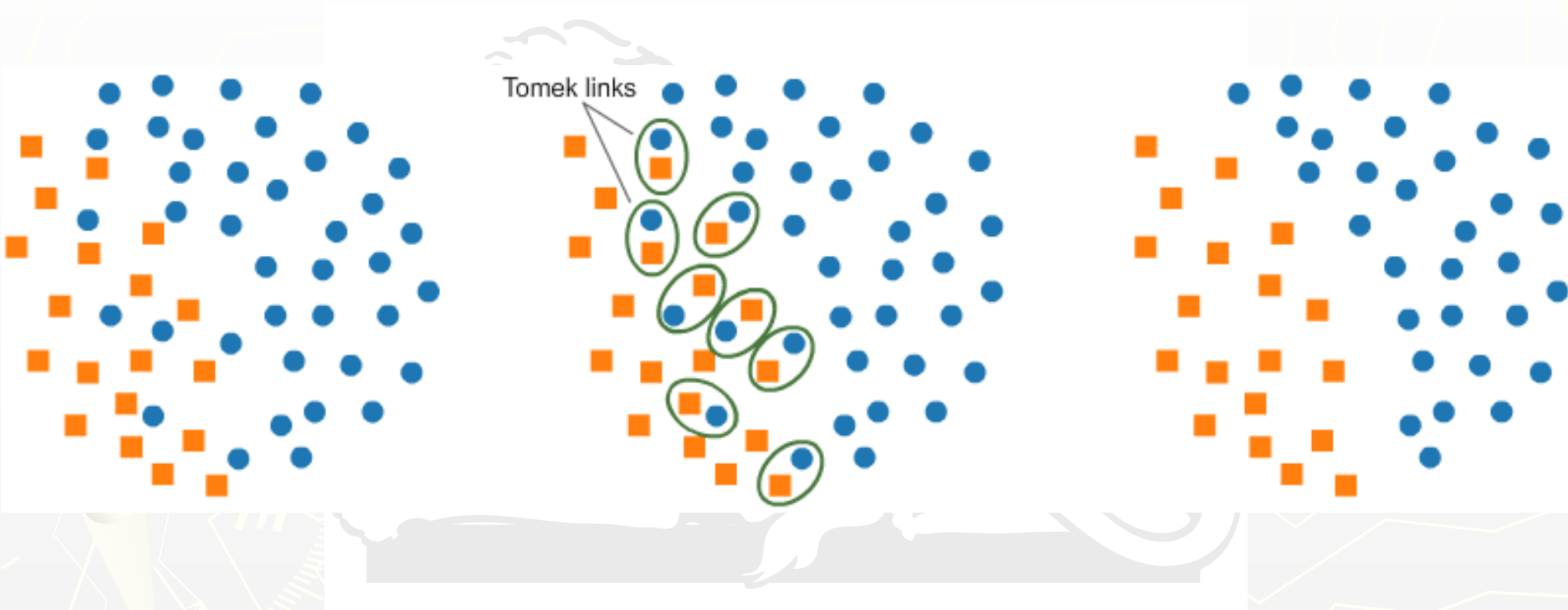


**Oversampling**



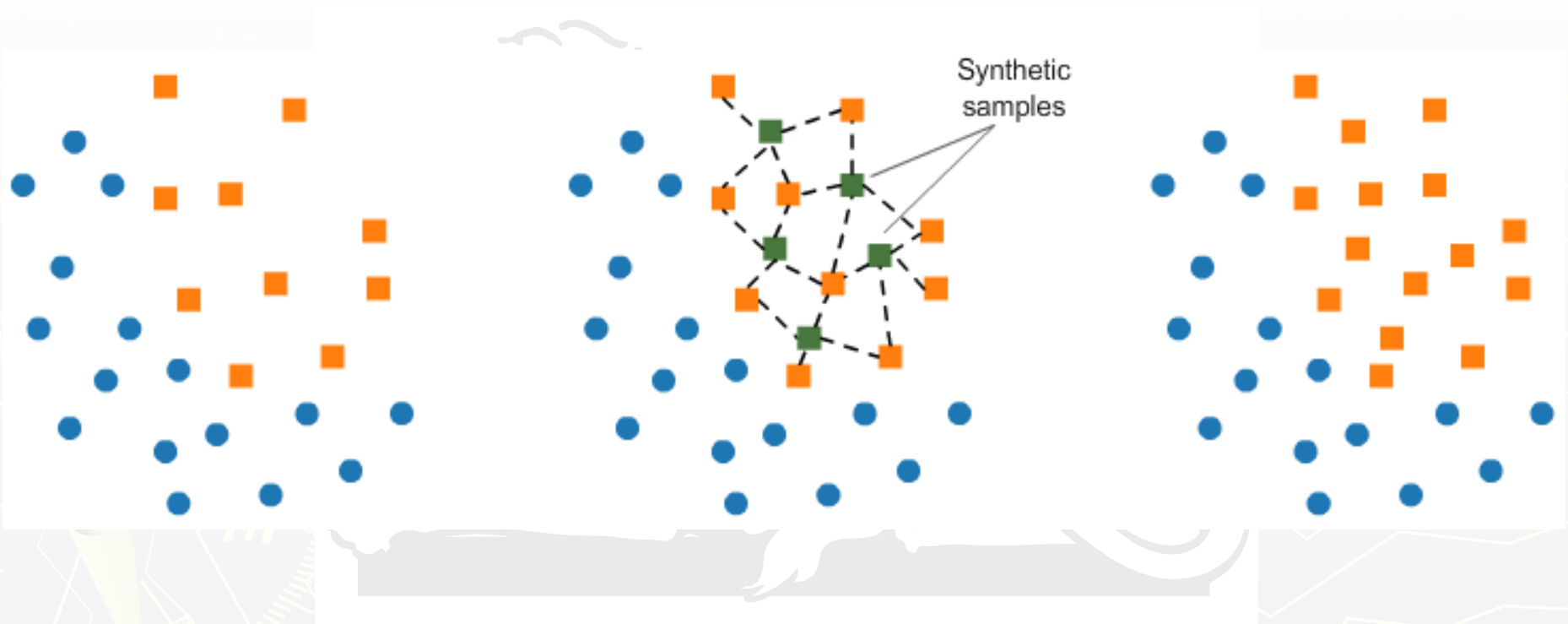
# Undersampling using Tomek Links:

One of such methods it provides is called Tomek Links. Tomek links are pairs of examples of opposite classes in close vicinity:



# Oversampling using SMOTE:

In SMOTE (Synthetic Minority Oversampling Technique) we synthesize elements for the minority class, in the vicinity of already existing elements:



# When, What, Why?

- ▶ No straightforward and sure-shot answer to this question.
- ▶ Beginner can try different algorithms
- ▶ Expert are at some advantage
- ▶ However, there are different factors like
  - the problem statement and the kind of output you want,
  - type and size of the data,
  - the available computational time,
  - number of features,
  - observations in the data,
- ▶ **Here are some important considerations while choosing an algorithm.**

# Factor-1: Speed or Training time

Algorithms	Training time
Linear and Logistic regressions, Linear SVM	easy to implement and quick to run.
Kernel SVM, which involve tuning of parameters, Neural networks with high convergence time, random forests, kNN	need a lot of time to train the data.



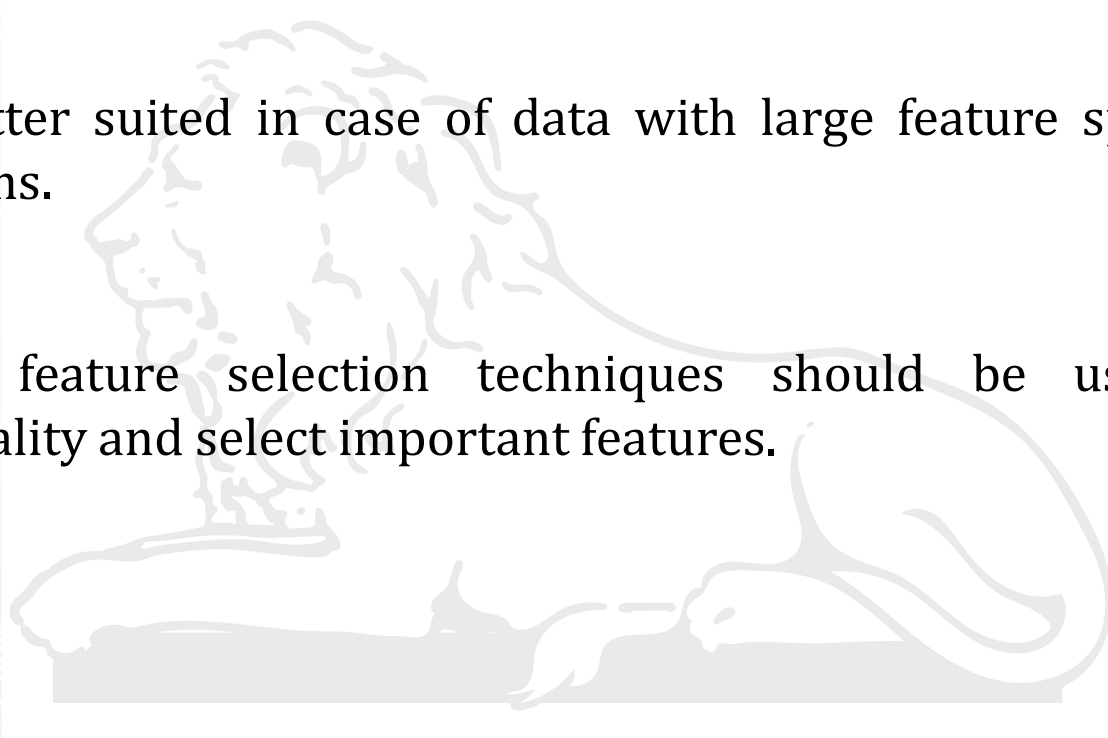


# Factor-2: Linearity

- Many algorithms work on the assumption that classes can be separated by a straight line (or its higher-dimensional analog). Examples include logistic regression and Linear SVM.
- Linear regression algorithms assume that data trends follow a straight line.
- If the data is linear, then these algorithms perform quite good.
- However, not always is the data is linear, so we require other algorithms which can handle high dimensional and complex data structures. Examples include kernel SVM, DT, neural nets.
- The best way to find out the linearity is to either fit a linear line or run a logistic regression or Linear-SVM and check for residual errors. A higher error means the data is not linear and would need complex algorithms to fit.

# Factor-3: # Features

- A large number of features can bog down some learning algorithms, making training time unfeasibly long.
- SVM is better suited in case of data with large feature space and lesser observations.
- PCA and feature selection techniques should be used to reduce dimensionality and select important features.



## Factor 4: Size of the training samples

- ▶ It is usually recommended to have sufficiently large training examples for having a reliable prediction (classification and regression)
- ▶ However, in many real life scenario, the availability of data is a constraint.

### Recommendations

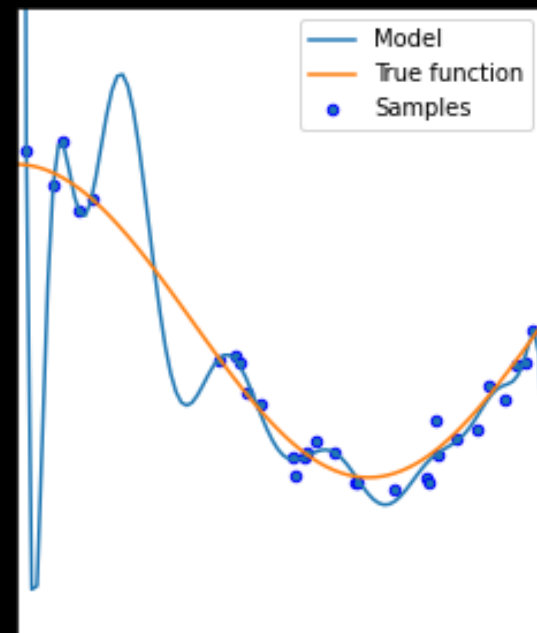
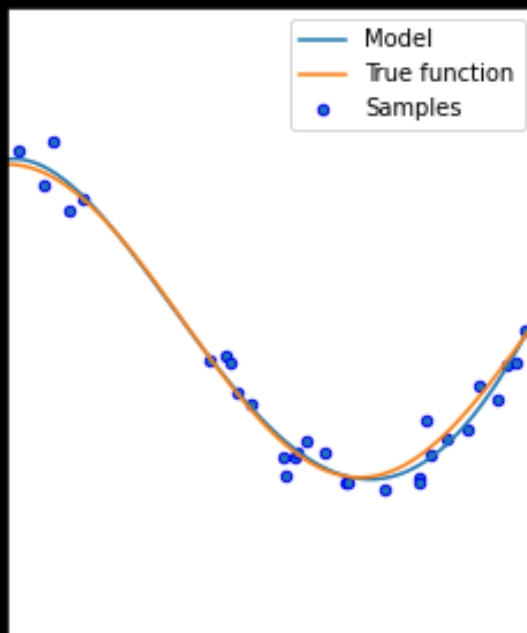
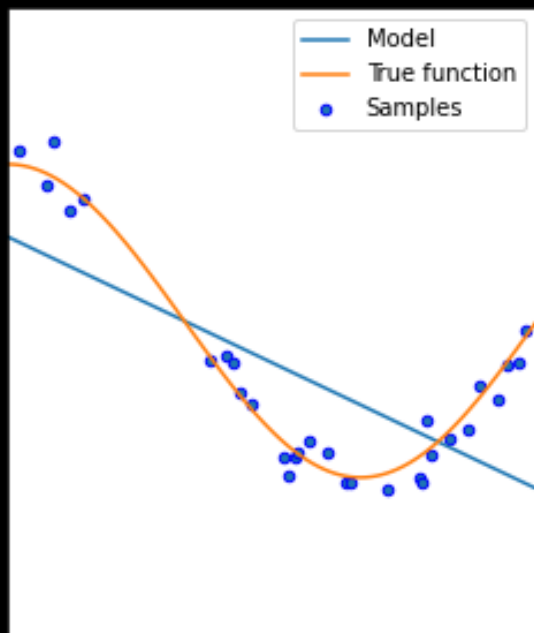
Smaller training data (fewer number of observations and a higher number of features)	like Linear regression, Logistic Regression Linear SVM, LDA
Data is sufficiently large and the number of observations is higher as compared to the number of features, one can go for	KNN, Decision trees, or kernel SVM

# Bias/Variance trade-off

- ▶ A machine learning model's performance is evaluated based on how accurate is its prediction and how well it generalizes on another independent dataset it has not seen.
- ▶ The errors in a machine learning model can be broken down into two parts:
  - Reducible Error
  - Irreducible error
- ▶ Irreducible errors are errors that cannot be reduced even if you use any other machine learning model.
- ▶ Reducible errors, on the other hand, is further broken down into square of bias and variance.
- ▶ Due to this bias-variance, it causes the machine learning model to either overfit or underfit the given data.

# Bias/Variance trade-off

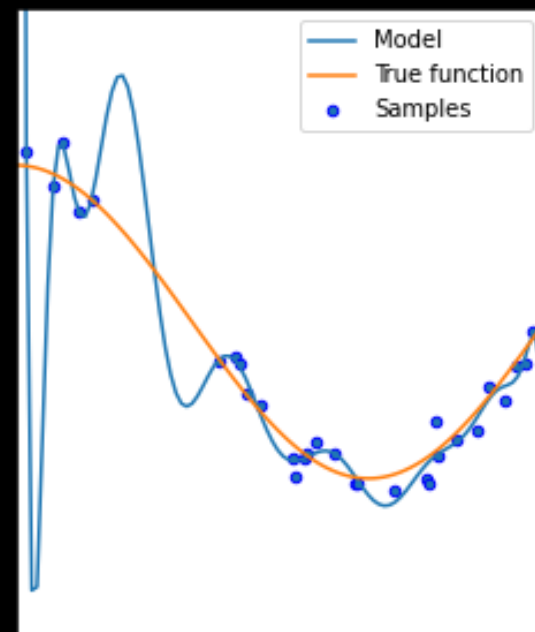
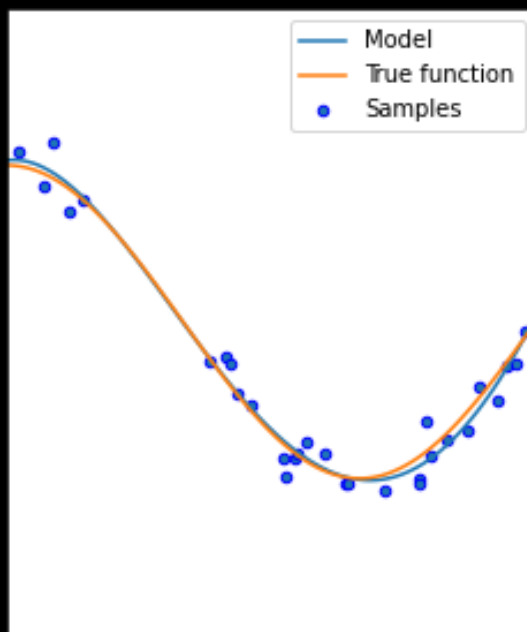
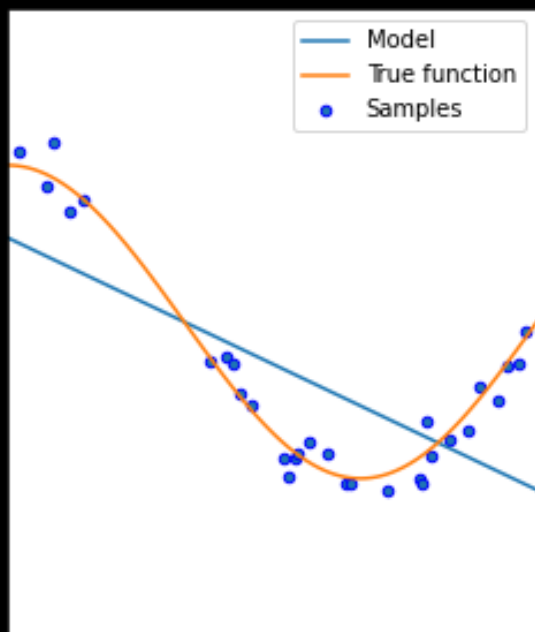
**Bias (underfitting)** is the tendency of an estimator to pick a model for the data that is not structurally correct. A biased estimator is one that makes incorrect assumptions on the model level about the dataset. For example, suppose that we use a linear regression model on a cubic function. This model will be biased: it will structurally underestimate the true values in the dataset, always, no matter how many points we use.



# Bias

Examples of **low-bias** machine learning algorithms include: Decision Trees, k-Nearest Neighbors and Support Vector Machines.

Examples of **high-bias** machine learning algorithms include: Linear Regression, Linear Discriminant Analysis and Logistic Regression.



# Variance

Variance is the amount that the estimate of the target function will change if different training data was used.

Ideally, it should not change too much from one training dataset to the next, meaning that the algorithm is good at picking out the hidden underlying mapping between the inputs and the output variables.

Machine learning algorithms that have a high variance are strongly influenced by the specifics of the training data. This means that the specifics of the training have influences the number and types of parameters used to characterize the mapping function.

**Low Variance:** Suggests small changes to the estimate of the target function with changes to the training dataset.

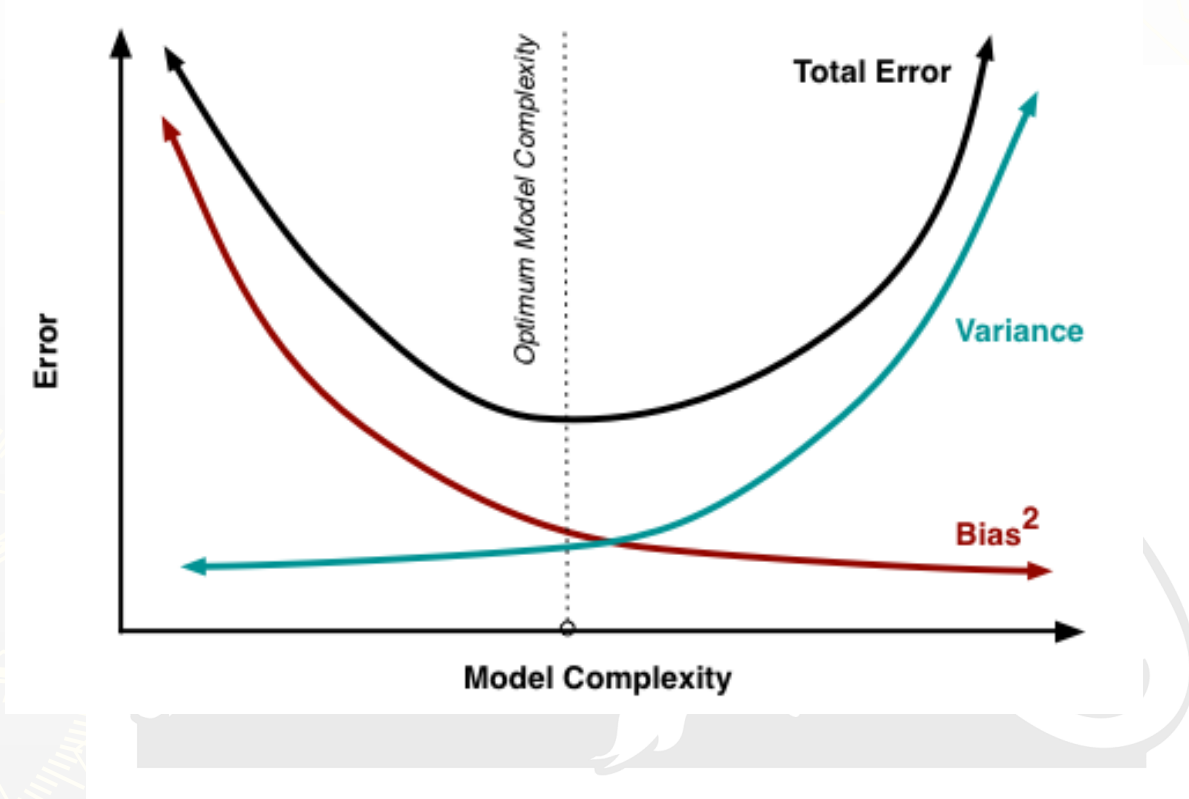
**High Variance:** Suggests large changes to the estimate of the target function with changes to the training dataset.

Examples of **low-variance** machine learning algorithms include: Linear Regression, Linear Discriminant Analysis and Logistic Regression.

Examples of **high-variance** machine learning algorithms include: Decision Trees, k-Nearest Neighbors and Support Vector Machines.

# Trade-off

The goal of any supervised machine learning algorithm is to achieve low bias and low variance. In turn the algorithm should achieve good prediction performance.





# Trade-off

The goal of any supervised machine learning algorithm is to achieve low bias and low variance. In turn the algorithm should achieve good prediction performance.

The k-nearest neighbors algorithm has low bias and high variance, but the trade-off can be changed by increasing the value of  $k$  which increases the number of neighbors that contribute to the prediction and in turn increases the bias of the model.

The support vector machine algorithm has low bias and high variance, but the trade-off can be changed by increasing the  $C$  parameter that influences the number of violations of the margin allowed in the training data which increases the bias but decreases the variance.

[https://colab.research.google.com/drive/1tsxD75mO1U4GZuLHE\\_EYUXZ-dNzQCM71#scrollTo=hWUp7W02DzYr](https://colab.research.google.com/drive/1tsxD75mO1U4GZuLHE_EYUXZ-dNzQCM71#scrollTo=hWUp7W02DzYr)



**THANKYOU**