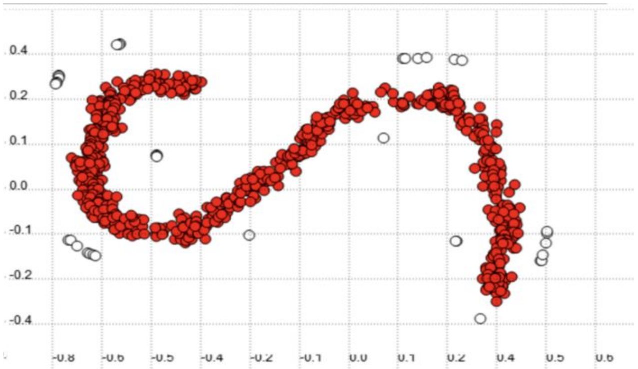
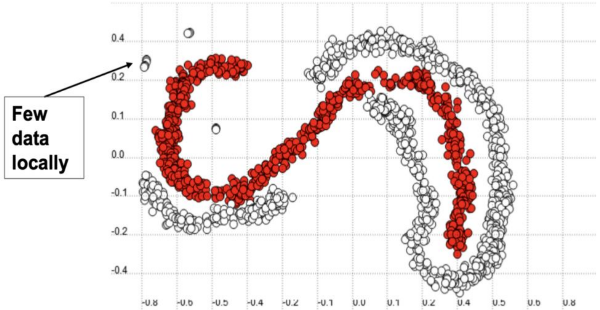
Imbalanced Dataset & Model Evaluation

Imbalanced Dataset

- An unequal distribution of classes
  - Example: In a credit card fraud detection dataset, most of the credit card transactions are not fraud and a very few classes are fraud transactions.
- Types of Imbalance Dataset



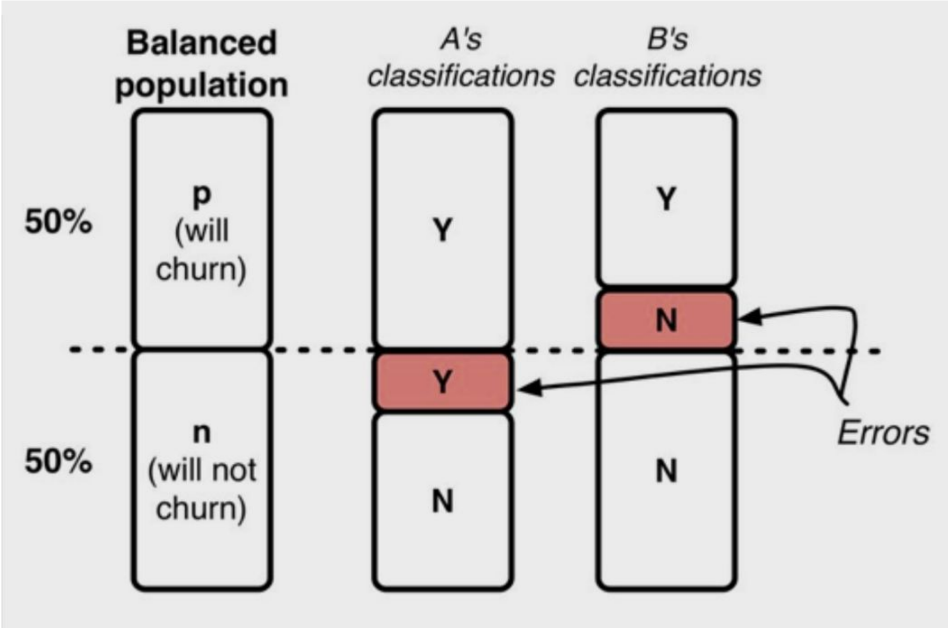
Between Class

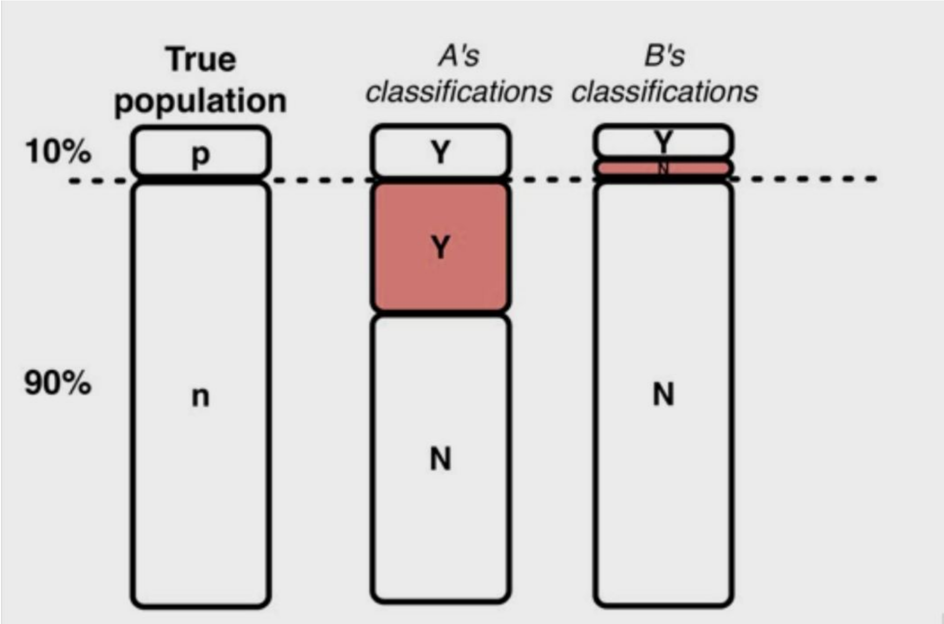


Within Class

Imbalanced Dataset

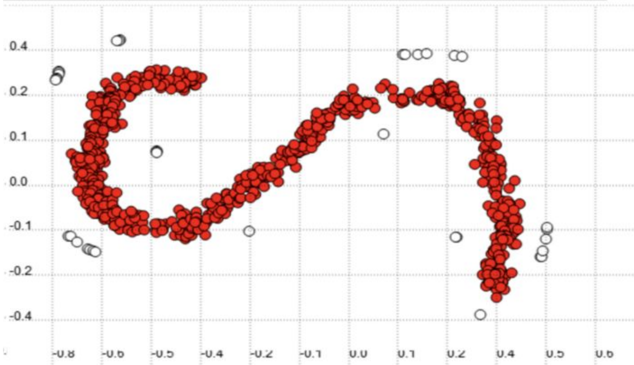
The impacts of balanced dataset in machine learning





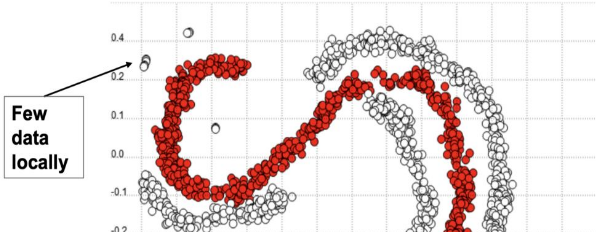
Imbalanced Dataset

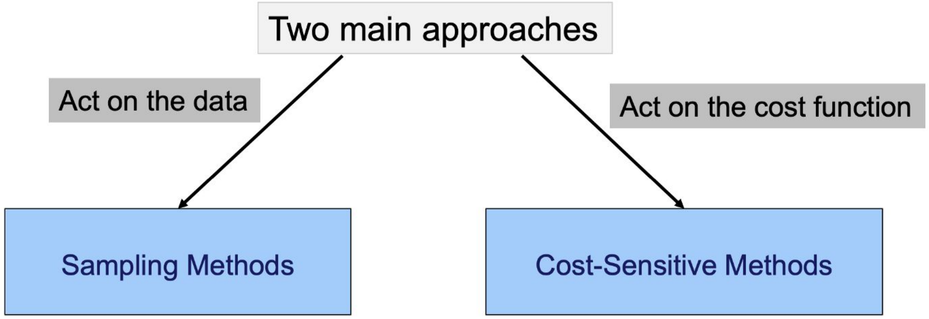
- An unequal distribution of classes
  - Example: In a credit card fraud detection dataset, most of the credit card transactions are not fraud and a very few classes are fraud transactions.
- Types of Imbalance Dataset
- Cause of Imbalance Dataset



Between Class

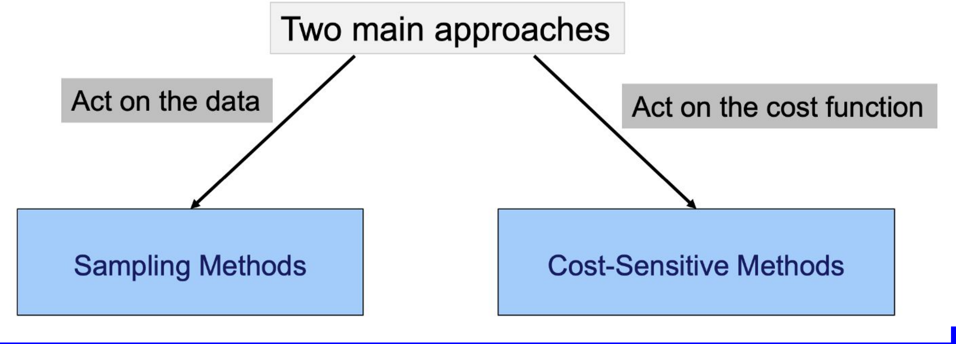
- f the proble
- Dataset
- on Meth





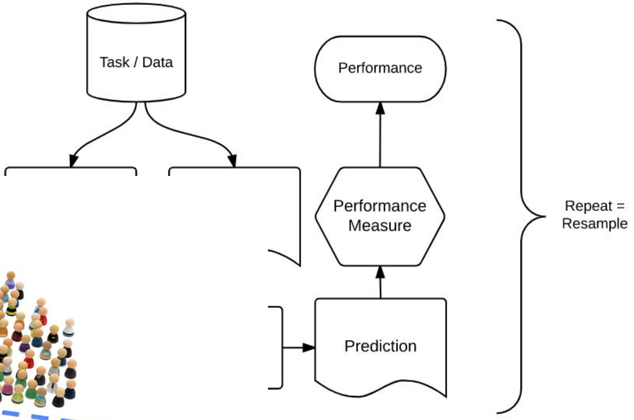
Imbalanced Dataset

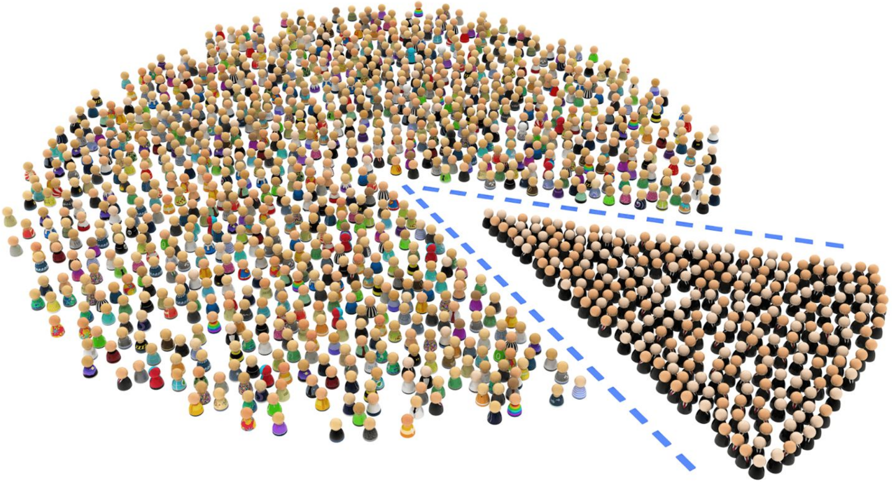
- An unequal distribution of classes
  - Example: In a credit card fraud detection dataset, most of the credit card transactions are not fraud and a very few classes are fraud transactions.
- Types of Imbalance Dataset
- Cause of Imbalance Dataset
  - Biased Sampling
  - Measurement Error
- The imbalance might be a property of the problem domain
- Approaches to handling Imbalanced Dataset
  - Act on Data (Sampling)
  - Act on Cost Function (Evaluation Meth

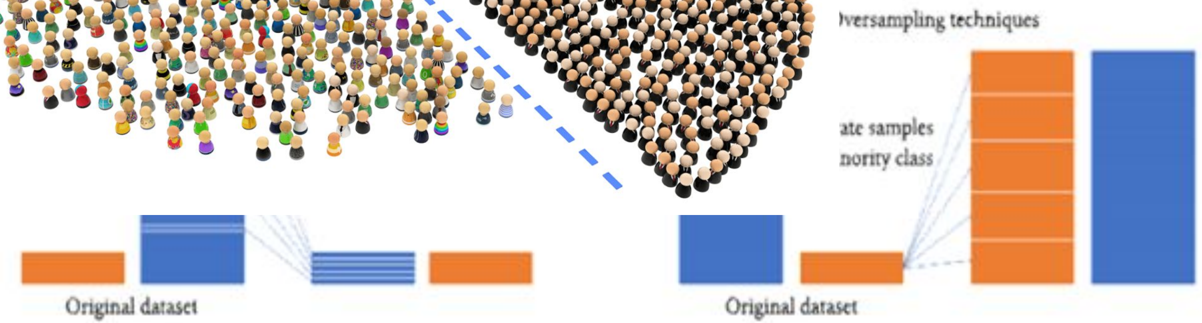


Sampling

- Undersampling (If enough data is available)
    - Remove some data from **Majority Class**
- Oversampling
    - Add new data
    - SMOTE (Synt
    - Random overs
- Resampling





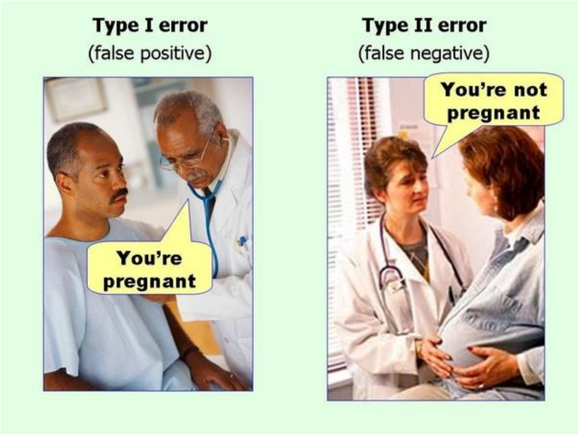


Evaluation Methods

- Upweighting
- Downweighting
- Evaluation Metrics
- Ensemble Method

Evaluation Metrics

- **True Positive**
- **True Negative**
- **False Positive**
- **False Negative**



		Predicted	
Actual		Negative	Positive
	Negative	True Negative	False Positive
	Positive	False Negative	True Positive

Evaluation Metrics

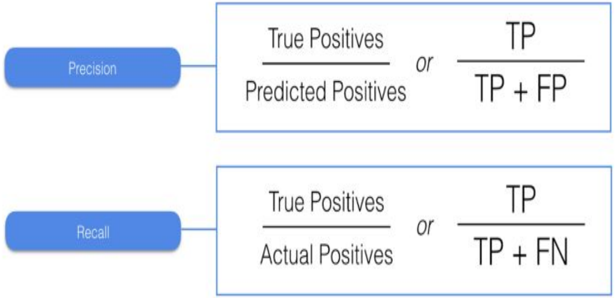
Precision/Specificity: how many selected instances are relevant

Recall/Sensitivity: how many relevant instances are selected

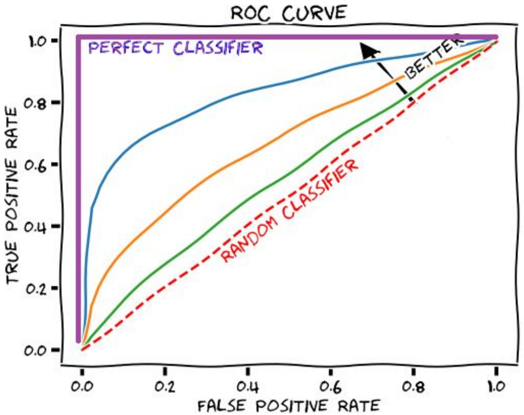
F1 score: harmonic mean of precision and recall

Confusion Matrix: a table showing the relation of predicted and expected result

ROC Curve: true positive vs false positive curve

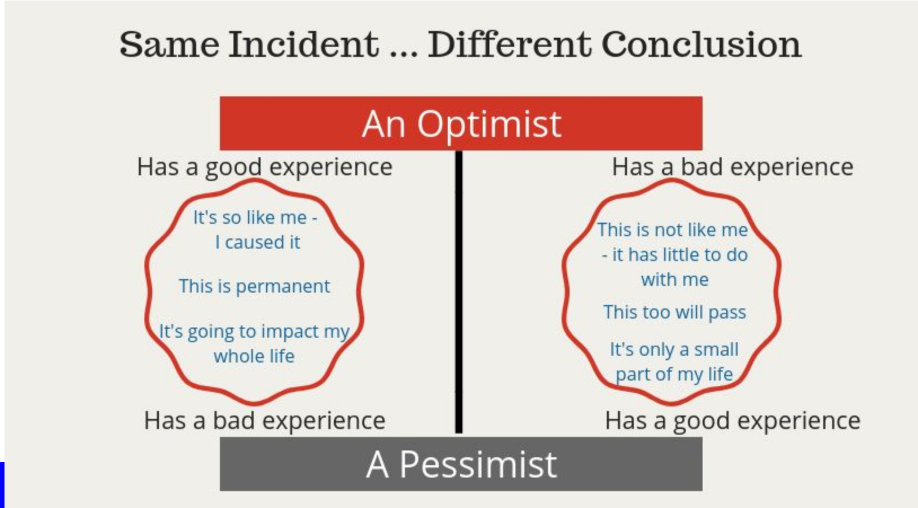


$$F1 = \frac{(1+\beta^2)\times \text{precision}\times \text{recall}}{(\beta^2 \times \text{precision})\times \text{recall}}$$



Model Evaluation (Holdout)

- Randomly partitioned in two independent sets
  - Training set
  - Test set
- Training set is used to train the model
- Test set is used to validate the accuracy of the model
- Estimation is Pessimistic



Model Evaluation (Random Sampling)

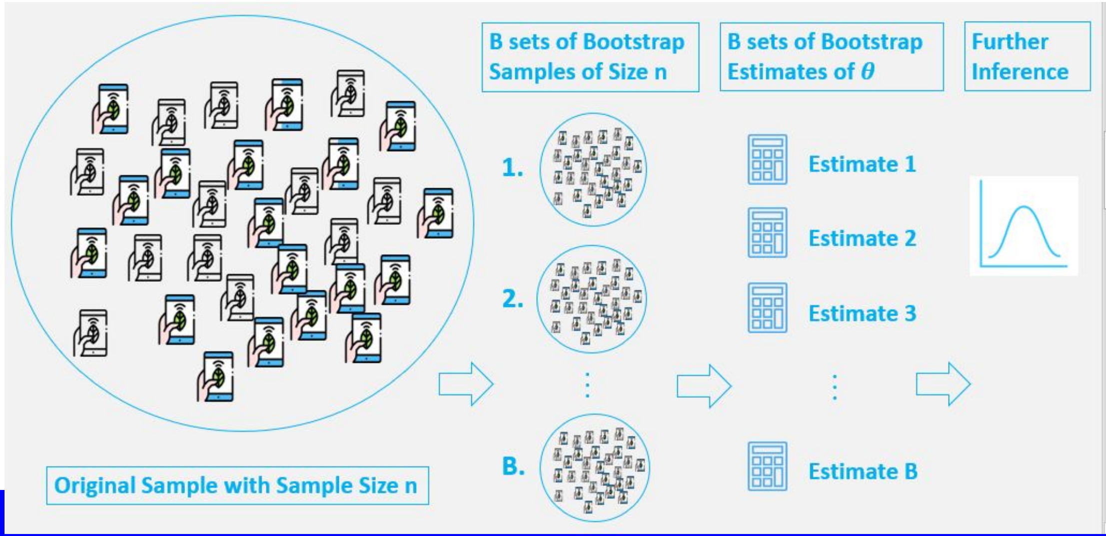
- Variation of Holdout method
- Holdout method is used for some n times
- Average result is considered

Model Evaluation (Cross Validation)

- k-fold
  - Randomly partitioned into k subsets
  - Training performs k times
  - Each time one subset of data is kept test data
  - Other (k-1) subsets are used as training dataset
- Leave-one-out
  - Special case of k-fold
  - Each fold contains only one data tuple
- Stratified cross validation
  - Preserves the data distribution in subsets

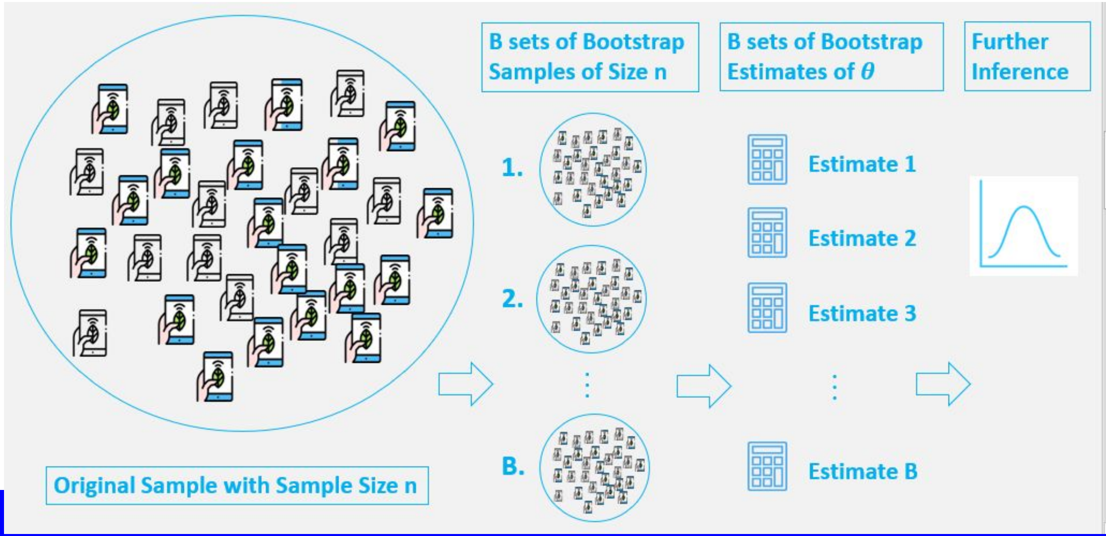
Model Evaluation (Bootstrap)

- Uniformly sample tuple with replacement
- On average, 63.2% data as Training data
- On average, 36.8% data as Test data
- Model accuracy is weighted

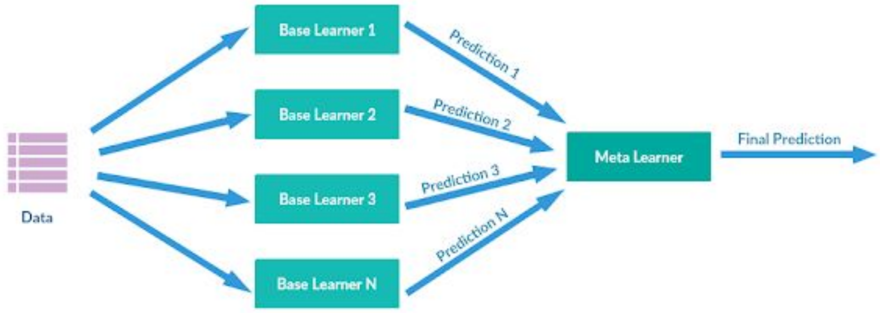


Model Evaluation (Bootstrap - steps)

- A sample from population with sample size n.
- Draw a sample from the original sample data **with replacement** with size n, and replicate B times, each re-sampled sample is called a Bootstrap Sample, and there will totally B Bootstrap Samples.
- Evaluate the **statistic** of θ for each Bootstrap Sample, and there will be totally B estimates of θ.
- Construct a **sampling distribution** with these B Bootstrap statistics and use it to make further statistical inference, such as:
    - Estimating the standard error of statistic for θ.
    - Obtaining a Confidence Interval for θ.



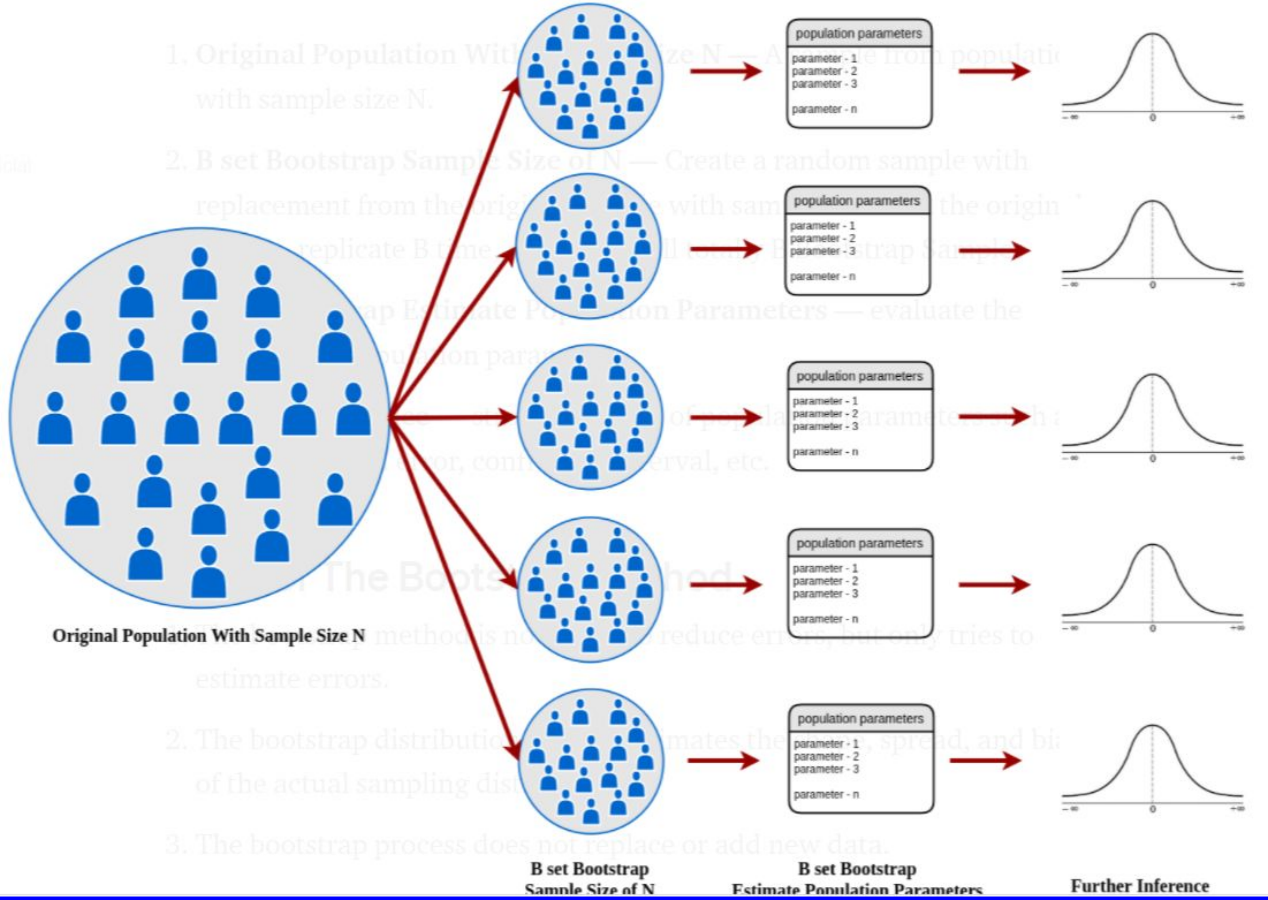
Model Evaluation (Ensemble Method)



- Use a set of classifiers
- Data is sampled for each classifier
- Final prediction based on majority voting
- Techniques:
  - Bagging
  - Boosting

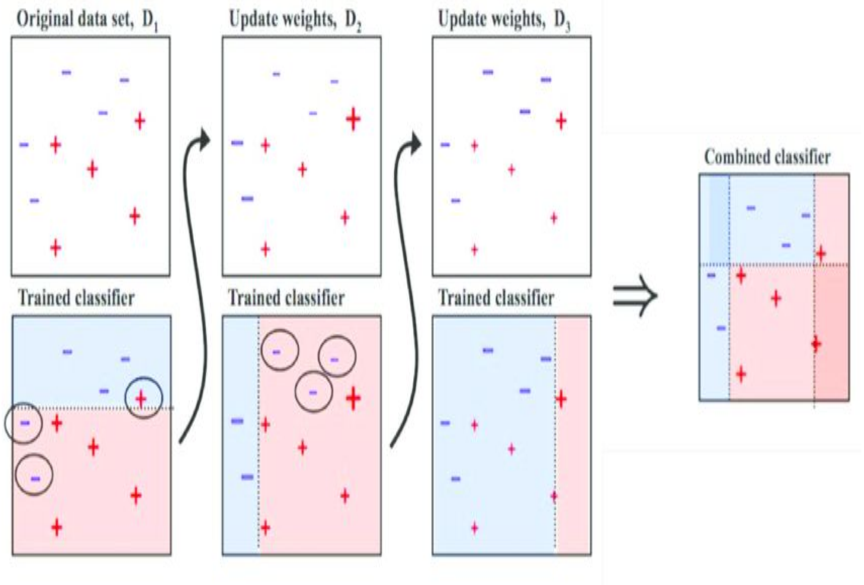
Model Evaluation (Bagging)

- Bagging stands for Bootstrap Aggregation
- Each Training is a bootstrap sample



Model Evaluation (Boosting)

- Weights are assigned to each tuple
- A series of n classifiers are learned
- Weights are updated after each iteration
  - Increased if predicted incorrectly
  - Decreased if predicted correctly
- Adaboost
  - Weights are updated based on error rate
  - Models are weighted based on error



Thank You

And Thanks to:
medium.com & towardsdatascience.com