XCCELERATE

Data Science / Machine Learning - Intermediate

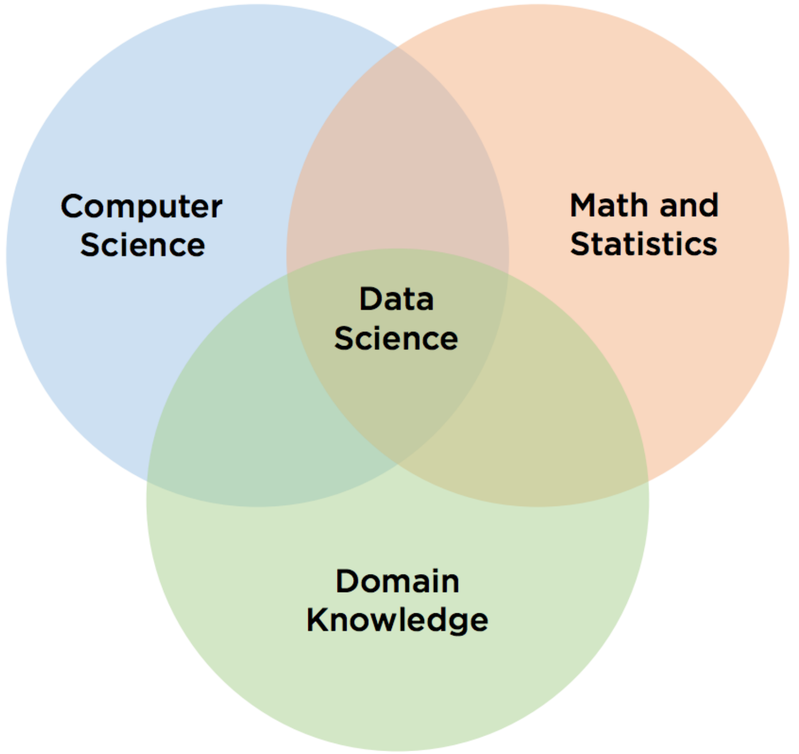
What is Data Science?

Joins statistics and programming in applied settings.

Analysis of diverse data

Deriving insights from data to make decisions or actionable steps.

These insights help business understand customers and competition.



What do Data Scientists do?

Understand the business and ask the right questions
Identify key business variables
Define success metrics and business requirements?

Acquire data from relevant sources:
Data collection: ingest the data into analytic environment
Data exploration: explore the data to check quality and adequac

Modeling
Pre-processing: set up data pipeline to prepare data
Feature engineering: create features from raw data
Model evaluation: perform model fitting
Model selection: explore to find the optimal model

Operationalization:
Deploy model to production
Model consumption: use model to make predictions (scoring)
Business validation: To verify if business requirements are met.

Iterate

Data sciences compared.

Compare

Process	Data Engineering	Business Intelligence	Business Analytics	Data Science
Integrate Data Sources	X			X
Build Data Pipelines	X			X
Process and Transform Data	X			X
Store Data	X			X
Dashboards/Reports		X	X	X
Exploratory Analytic		X	X	X
Statistical Modelin			X	X
Machine Learnin			X	X
Business Recommendations		X	X	X
Business Action			X	X

Machine Learning

What is Machine Learning?

Arthur samuel (1959) Machine learning is a field of computer science that gives computers the ability to learn without being explicitly programmed.[1]

Tom M. Mitchell (1998)"A computer program is said to learn from experience *E* with respect to some class of tasks *T* and performance measure *P* if its performance at tasks in *T*, as measured by *P*, improves with experience *E*."[16]

Machine Learning

What is Machine Learning?

Algorithms that do the learning without human intervention.

Learning is done based on examples (aka dataset).

Goal:
learning function f: x [?] y to make correct prediction for new input data
Choose function family (logistic regression, support vector machines)
Optimize parameters on training data: Minimize Loss ∑(f(x) - y)2

Course Outline -

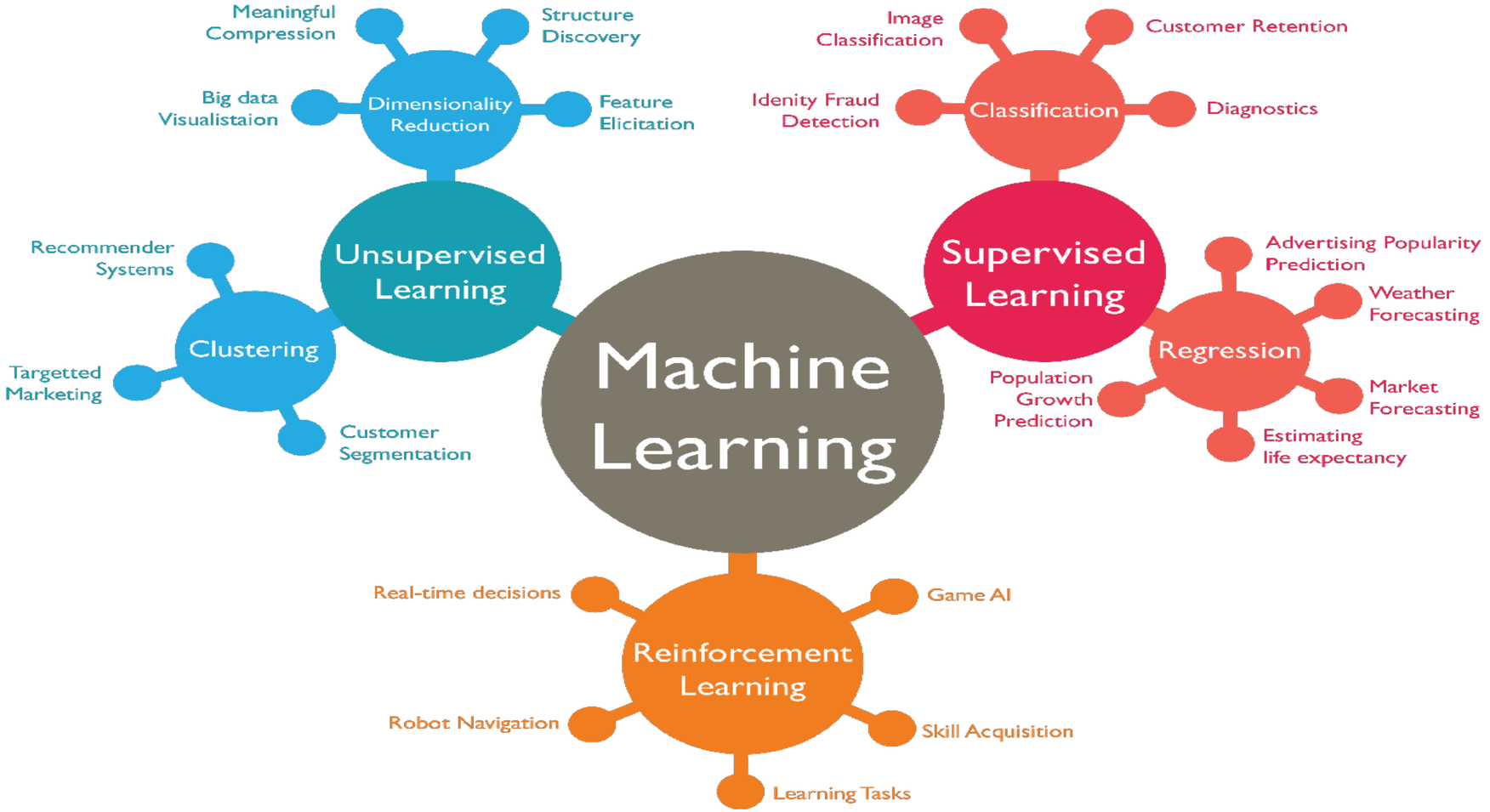
Fundamentals	Machine Learning Track	Business Intelligence Track	Data Engineering Track	Projects
Python Numpy and Pandas Visualization libraries API /JSON Web scraping Linear algebra, Matrices, Vectors Essentials / operations research Analysis Project	Fundamentals Statistics Machine Learning Feature Engineering, PCA Dimensionality reduction	BI Fundamentals Data cleaning in Excel Data analysis in Excel Visualizing in Excel. Project	Data Storage Database Management systems SQL Data Lakes Tableau Prep Tool	Collaborative Project Capstone Project
	Image recognition Recommender systems Text mining Projects	Introduction to Tableau Tableau for data cleaning and exploration Data mining with Tableau Dashboards/Visualization Project	Data Warehousing concepts Hadoop and Big data Google Big Query Azure Data Bricks Analytics	

Course Outline – Intermediate Machine Learning

Image Recognition (4 days)	Text Mining(3 days)	Recommender systems (3 days)
Machine Learning Project Presentations Machine Learning Revision Model selection & accuracy improvements Data pre-processing tips Code Templates Intro to Ensemble Learning Self Explore – xgBoost (1pm) Exercises (4 pm onwards – for ensemble) Deep Learning Intro (2:30) Deep Learning using Keras Convolutional Neural Networks Image recognition using CNN Day 2- exercise.	- Introduction to Natural Language processing - Using NLTK - Tokenize sentences - Remove stop words - Identify bigrams - Auto-summarizing Text	Intro to Recommender systems Product to Product relationships - User to Product relationships - User to User relationships - Types of systems - Collaborative Filtering Exercises/Projects
Saving your models. Using pre-trained models Transfer learning Tensor flow – image recognition Cloud based image recognition. Self Explore – tensorflow.js, facial recognition, yolo3. Exercises	Abstract Extraction - Beautiful Soup - Typical ML workflow - K means clustering - Classification - K Nearest Neighbors	- Latent Factor Analysis - Content Based Filtering - Nearest Neighbors model - Ratings Matrix - Association Rules learning - Apriori Algorithm Exercises / Projects

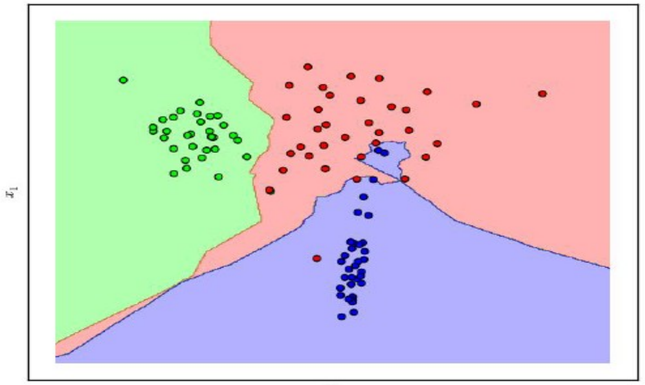
Projects
Collaborative Project Capstone Project

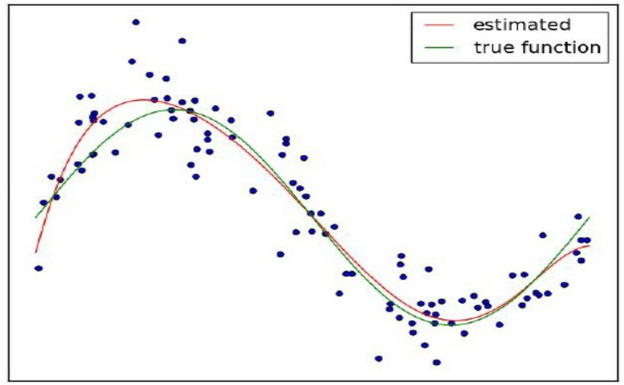
Machine Learning types



Outcomes

- **Classification**
  - Given labeled input data (with two or more labels), fit a function that can determine for any input, what the label is.
- **Regression**
  - Given continuous input data fit a function that is able to predict the continuous value of input given other data.





Parametric and Nonparametric

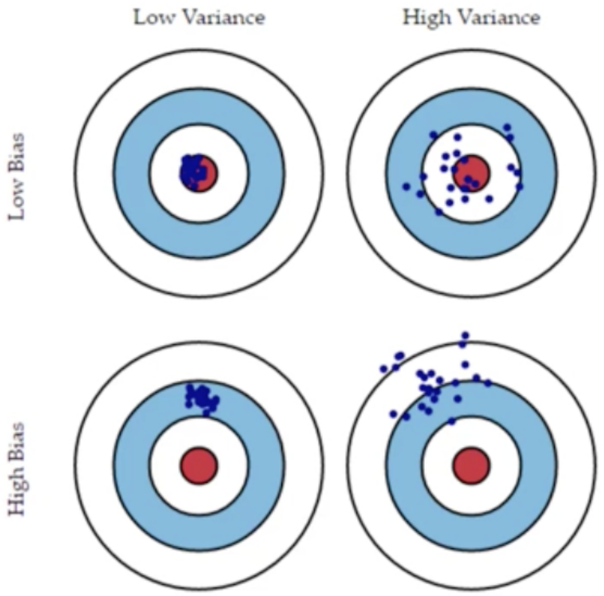
- **Parametric**
  - Rely on set parameters of fixed size independent of the training samples.
  - Make large assumptions about the mapping of the input variables(form) to the output variable
  - Functional form is a linear combination of input variables.
  - Eg. Linear / Logistic regression, LDA, Perceptron.
  - Pros: Simpler, faster, less data. Cons: Constrained, limited complexity, Poor fit
- **Non-Parametric**
  - Do not make assumption about form of the function,
  - They learn from lots of data
  - Have no prior knowledge of data.
  - Eg: Decision trees, Naïve Bayes, SVM, Neural networks
  - Pros: Flexibility, Power, Performance. Cons: More data, Slower, Overfitting

Supervised and Unsupervised

- **Supervised**
  - Input variables and Output variables supplied to map the algorithm from input to output.
  - Examples:
  - Linear regression for regression problems
  - Random forests for both classification and regression problems
  - Support vector machines for classification.
- **Unsupervised**
  - Only input variables and no corresponding output variables
  - Algorithms devise their own structure and relationship in the data
  - Clustering – segments or groups
  - Association – discover rules which describes large portions of your data.
- **Semi-supervised**
  - Some data is labeled, so a mix of supervised and unsupervised can be used.

BIAS and Variance

- Imagine that the center of the target is a model that perfectly predicts the correct values.
- As we move away from the bulls-eye, our predictions get worse and worse.

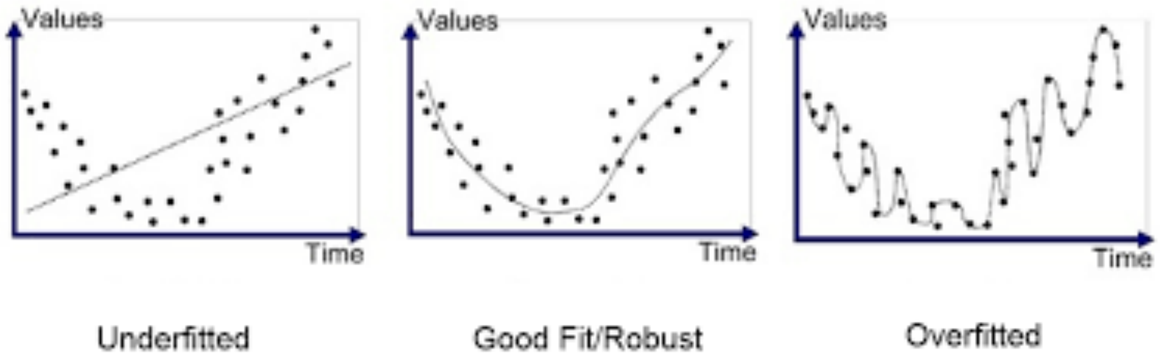


- **BIAS:** assumptions made by the model to make target function easier to “learn”.
  - **HIGH BIAS = OVERLY GENERALIZED MODEL**
- **VARIANCE:** is the amount that the estimate of the target function will change if different training data was used
- **IRREDUCIBLE ERROR**

Trade-off between Bias and Variance

Over fitting and Under fitting

- Over fitting
  - Learns the training data too well
  - Negatively impacts the model ability to generalize
  - Most common problem
  - Techniques to handle
    - Resampling technique ; k-fold cross validation.
    - Have a separate validation test.
- Under fitting
  - Neither model the training data nor generalise to new data
  - Choose a different algorithm



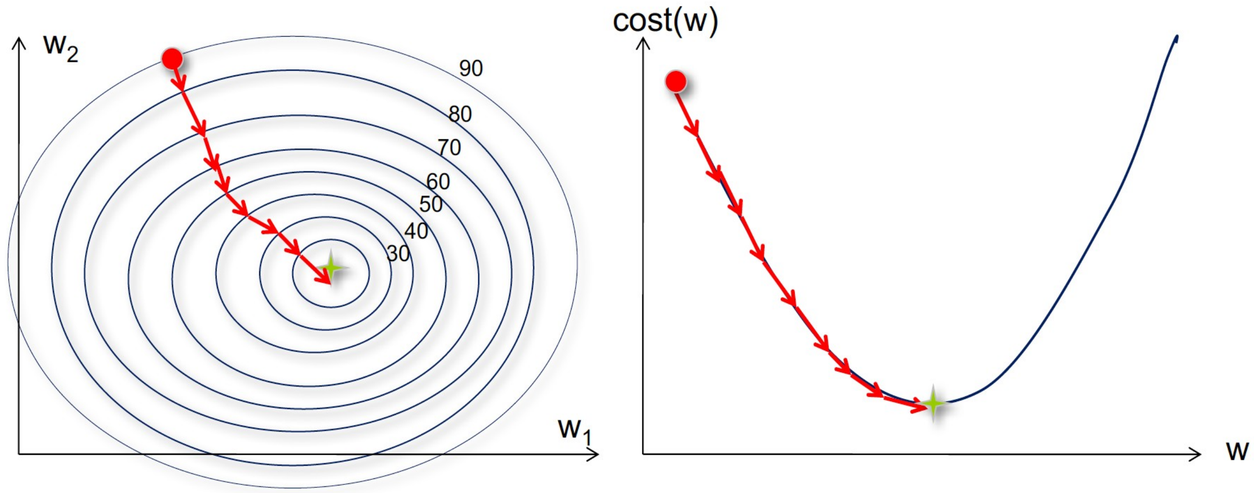
Gradient Descent

Gradient descent is an optimization algorithm used to find the values of parameters (coefficients) of a function (*f*) that minimizes a cost function (*cost*).

Gradient descent is best used when the parameters cannot be calculated analytically (e.g. using linear algebra) and must be searched for by an optimization algorithm.

Gradient Descent is an approach to adjust the weights by little steps.

In each step, the weights are adjusted toward the direction of gradient descent of the Cost Function.



Types of Gradient Descent

Batch gradient descent refers to calculating the derivative from all training data before calculating an update.

Stochastic gradient descent refers to calculating the derivative from each training data instance and calculating the update immediately (SGD)

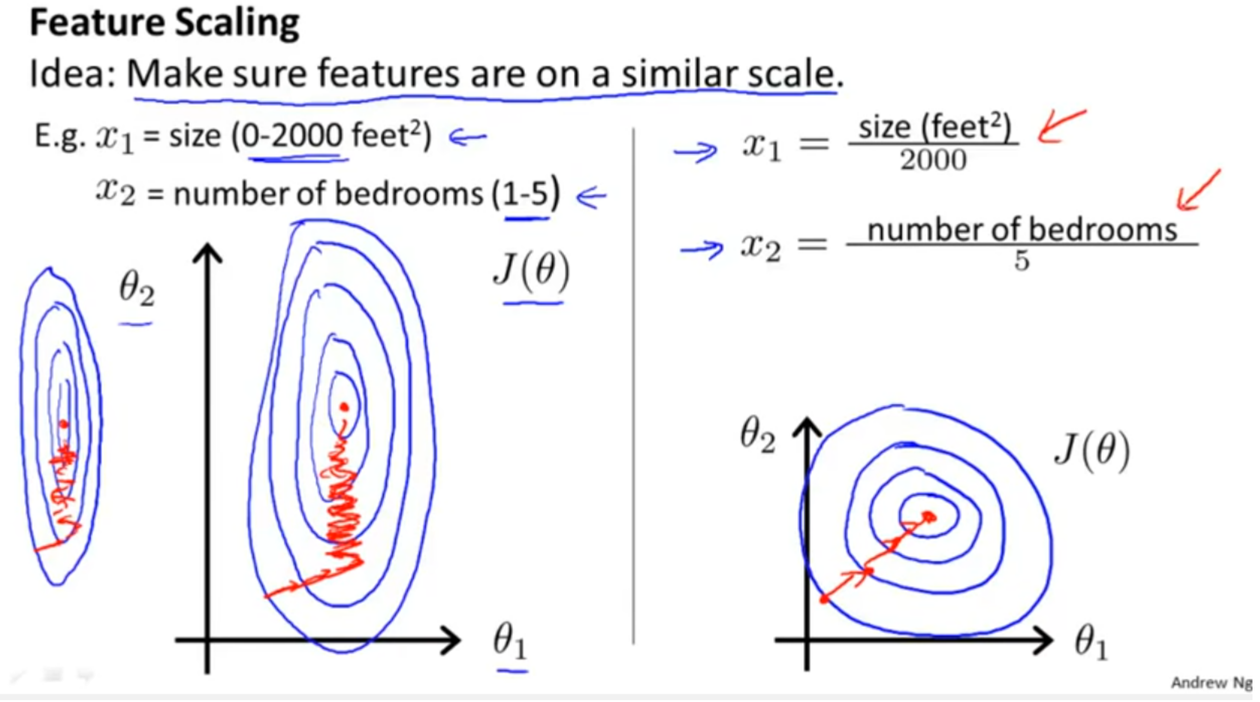
Data Pre-processing receipes

Pre-processing libraries or API
Fit – prepare the params to transform once
Transform – prepare for modeling

Re-Scale	**Standardise**	**Normalise**	**Binarize**
Modules: MinMaxScaler Attributes are scaled between 0 and 1 • Regression • Neural networks(Gradient Descent) • K-nn	Standard Gaussian distribution with a mean of 0 and a standard deviation of 1 **Modules: StandardScalar** • Linear regression • Logistic regression • LDA	Each observation to have a length of 1.(unit norm) **Modules: Normalizer** **Sparse datasets with lots of zeros and varying scales.** • Neural networks • K-nn	**Modules:** Binarizer Probabilities to crisp values

Feature Scaling

	A	B	N
1	**Algorithm**	**Problem Type**	**Features might need scaling?**
2	KNN	Either	Yes
3	Linear regression	Regression	No (unless regularized)
4	Logistic regression	Classification	No (unless regularized)
5	Naive Bayes	Classification	No
6	Decision trees	Either	No
7	Random Forests	Either	No
8	AdaBoost	Either	No
9	Neural networks	Either	Yes



https://scikit-learn.org/stable/auto_examples/preprocessing/plot_all_scaling.html

Supervised Learning:

- Look at some examples (labeled data) and find a way to predict future (unlabeled) examples
- the target variable ("labels") contains the ground truth we want to predict
- by comparing predictions with the ground truth, we know how well we're doing

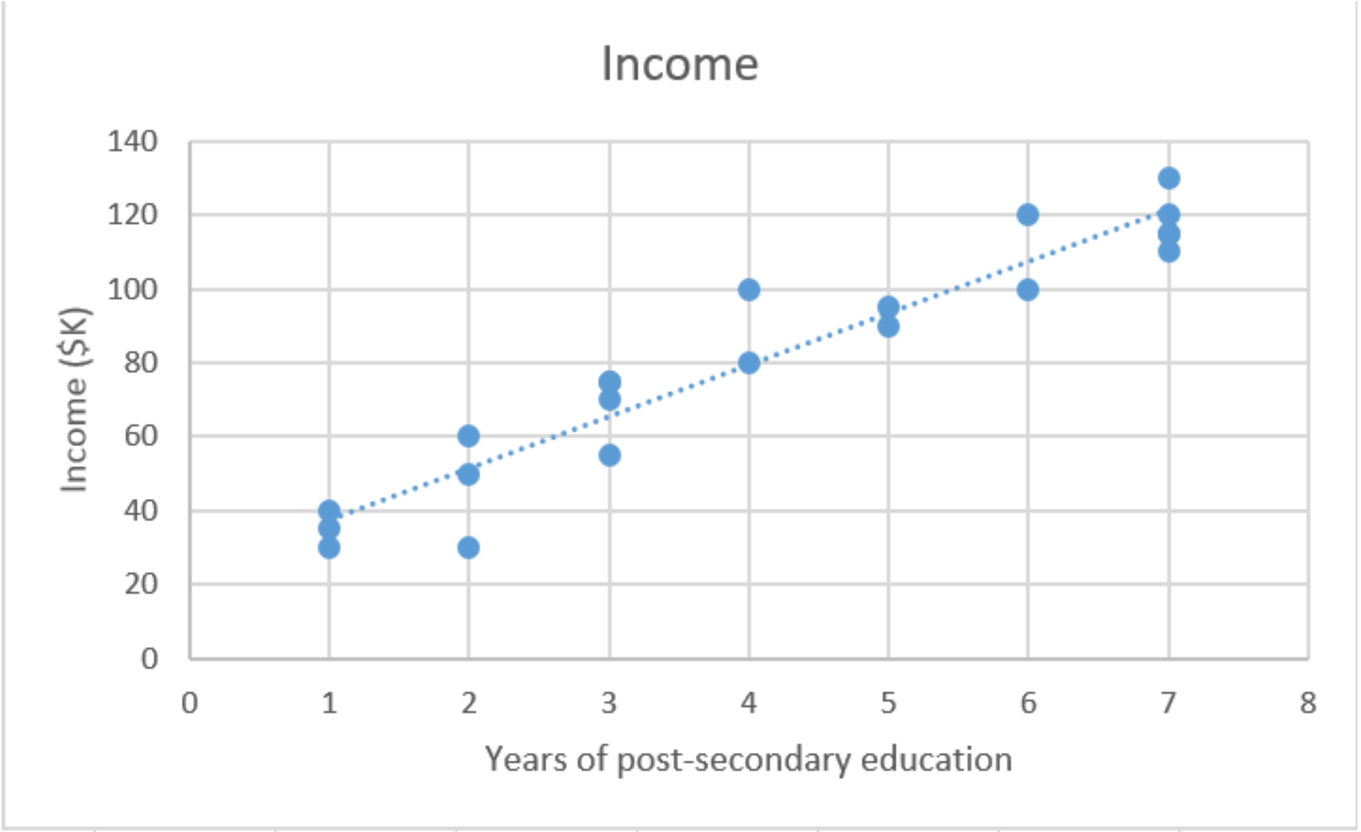
Summary: We are trying to predict a variable (called labels, target variable, response variable or dependent variable) using other variables (called features, explanatory variables, covariates, attributes or independent variables).

- regression algorithms predict a number (numeric target)
- classification algorithms predict a category (categorical target)
- Sometimes regression refers to a family of ML algorithms. For example, linear regression is a regression algorithm but logistic regression is a classification algorithm!

Linear Regression:

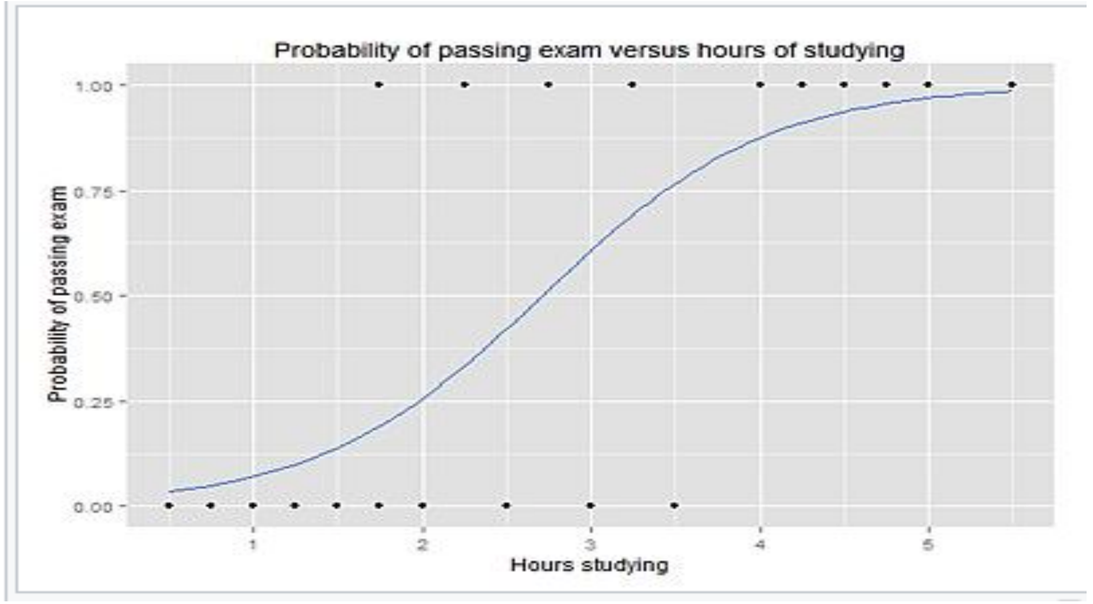
Linear Regression

- predict a continuous numerical value. *How much will that house sell for?*
- *Predict income levels based on years of experience*



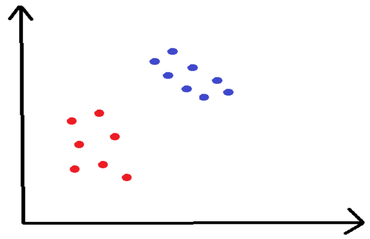
Logistic Regression

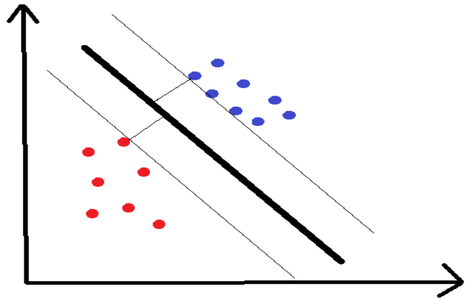
– Classification
  - Is this email SPAM or NOT
  - *Will users click on that Ad?*
  - *Who is the person in the picture?*

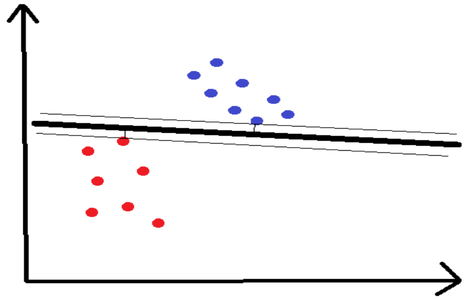


For Multi class – use Linear Discriminant Analysis (LDA)

Support Vector Machines



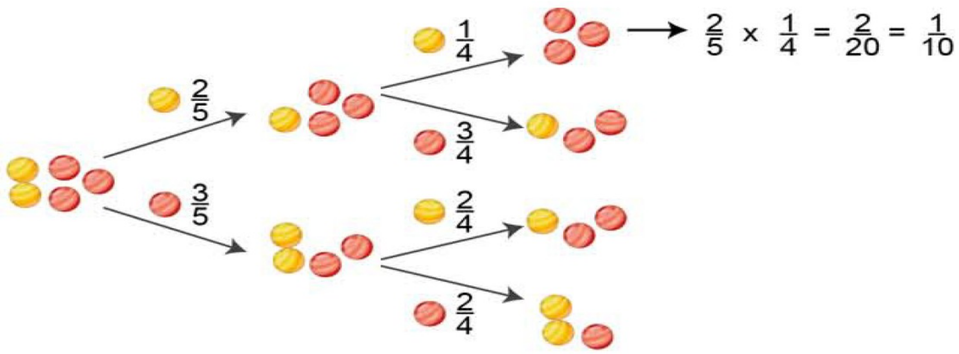




- Is this an image of a cat or a dog?
- Is this review positive or negative?
- Are the dots in the 2D plane red or blue?

Naïve Bayes

- Works with small amount of data, handle multiple classes
- Using Bayesian decision theory to predicate the possibility of every classes based on the conditional probability
  - $P(c_i|w) = P(w|c_i)P(c_i)/P(w)$
- Assumption: all features are independent (often not quite right, hence so called "Naïve"
- Sensitive to how the input data is prepared

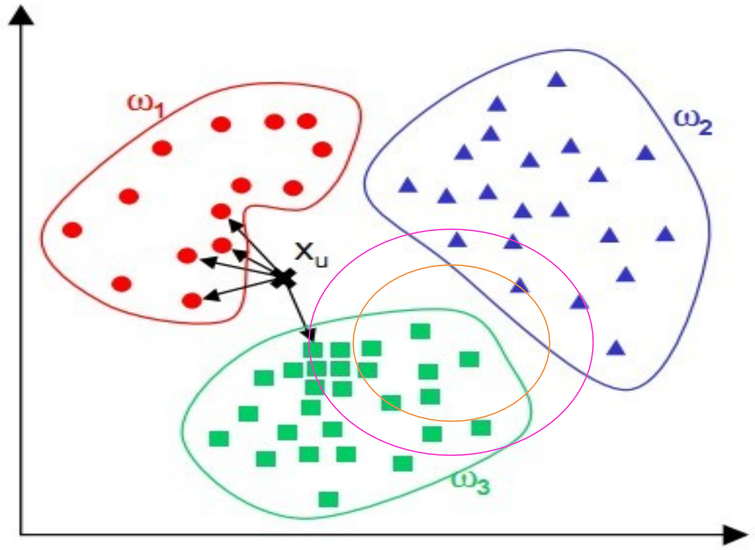


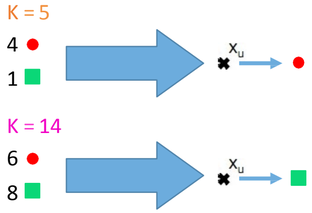
$$P(C_\# A) = \frac{P(A|C_\#) * P(C_\#)}{P(A)}$$

$$P(C_\#|A) = \frac{P(A|C_\#) * P(C_\#)}{\sum P(A|C_\#) * P(C_\#)}$$

K-Nearest Neighbors

In KNN, data points are categorized and when determining the category of a new data point, the K nearest points are used in this process.





The distance of the points can be calculated with the following fomula:

- For points in two dimensional space:

$$d = \sqrt{(x_0 - y_0)^2 + (x_1 - y_1)^2}$$

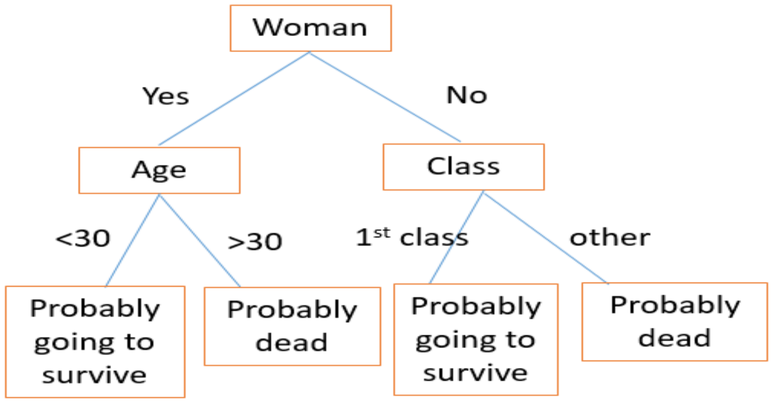
- For points in n-dimensional space:

$$d = \sqrt{\sum_{i=1}^{n} (x_i - y_i)^2}$$

Decision Trees

A normal tree

A decision tree

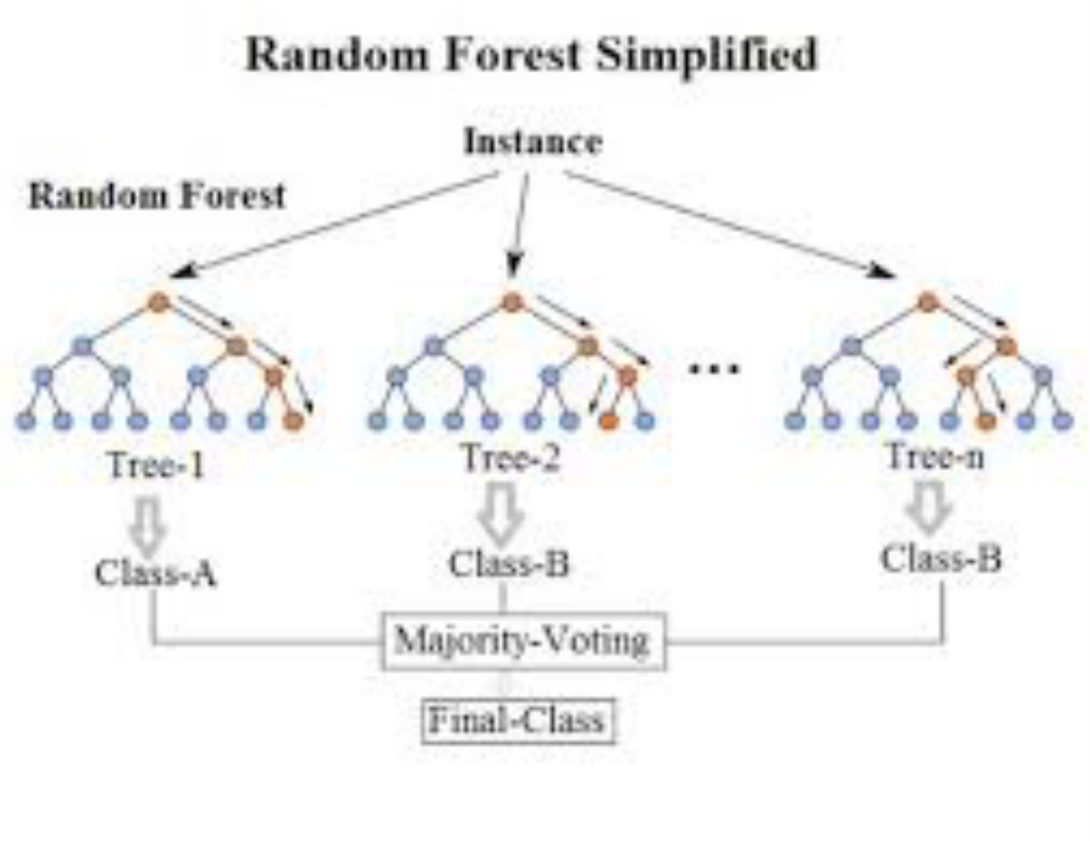


Random Forests

A model comprised of many models is called an **ensemble model**, and this is usually a winning strategy.

Number of features that can be split on at each node is limited to some percentage of the total

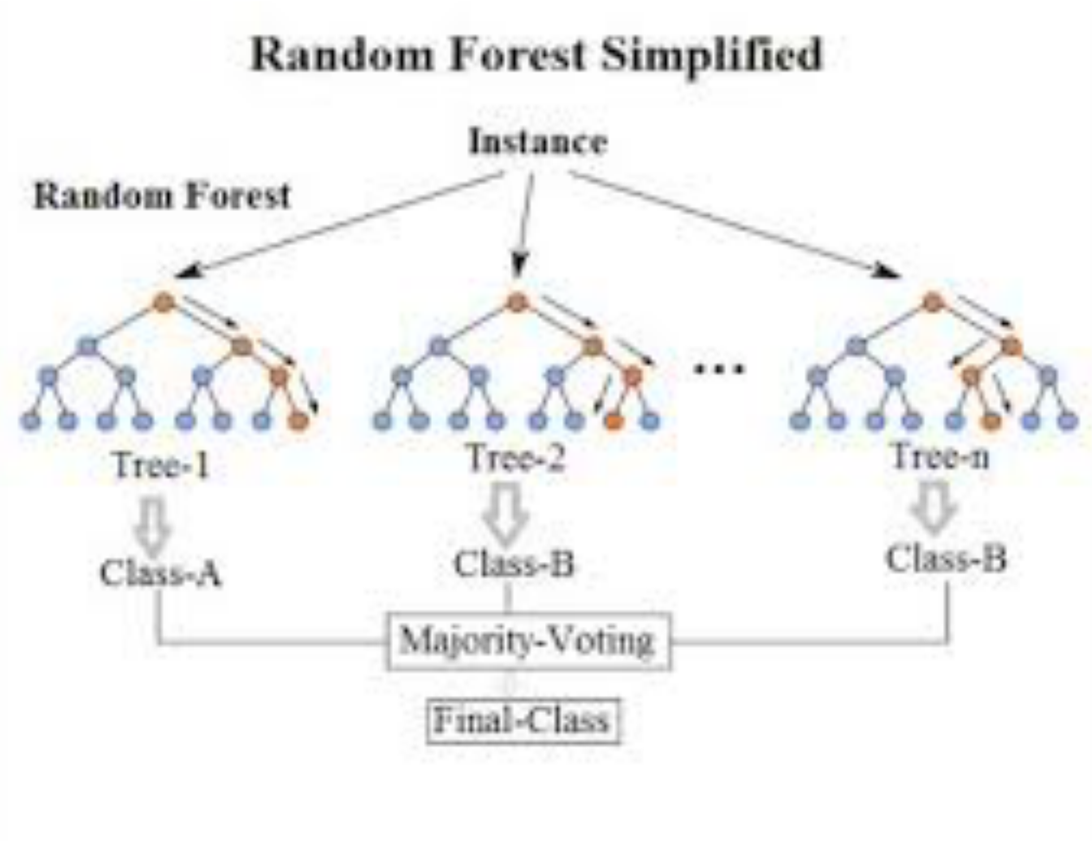
Each tree draws a random sample from the original data set when generating its splits, adding a further element of randomness that prevents overfitting



Ensemble learning

Random Forest is the most popular ensemble learning algorithm

Uses something called as Bagging (or bootstrap aggregation)

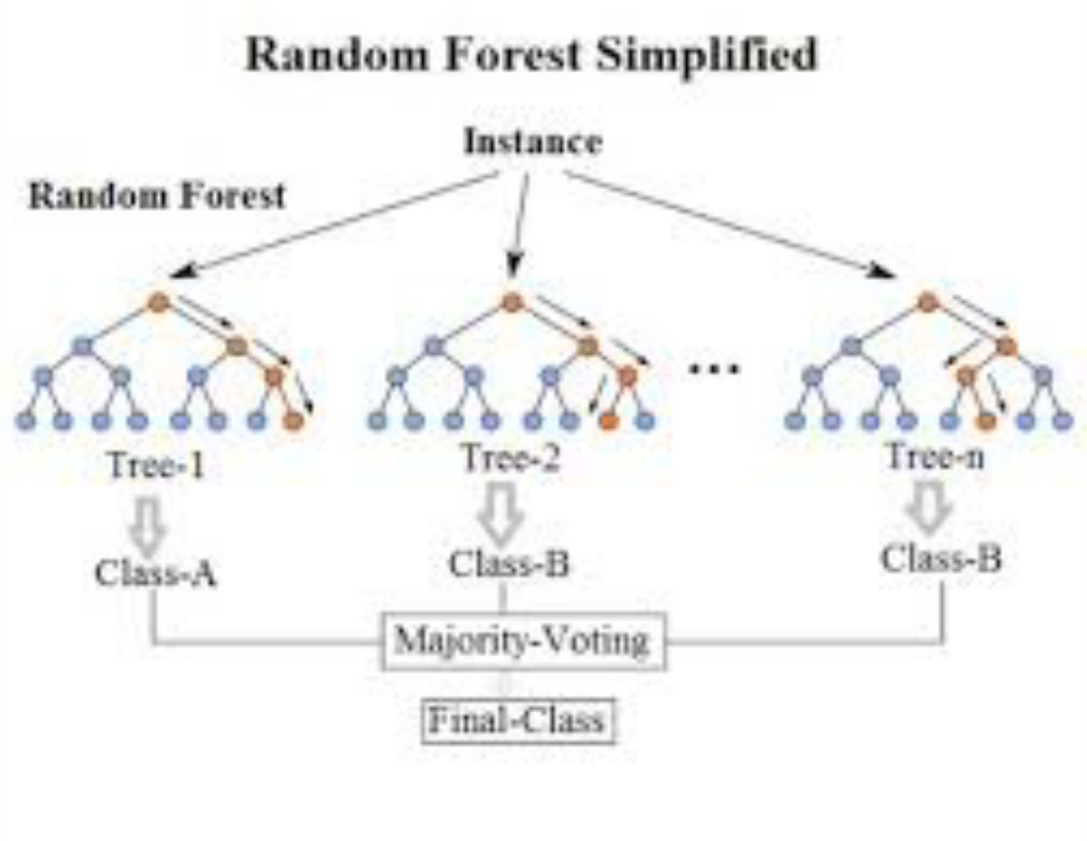


Bootstrap Method

Estimate a quantity from a data sample

Uses something called as Bagging (or bootstrap aggregation)

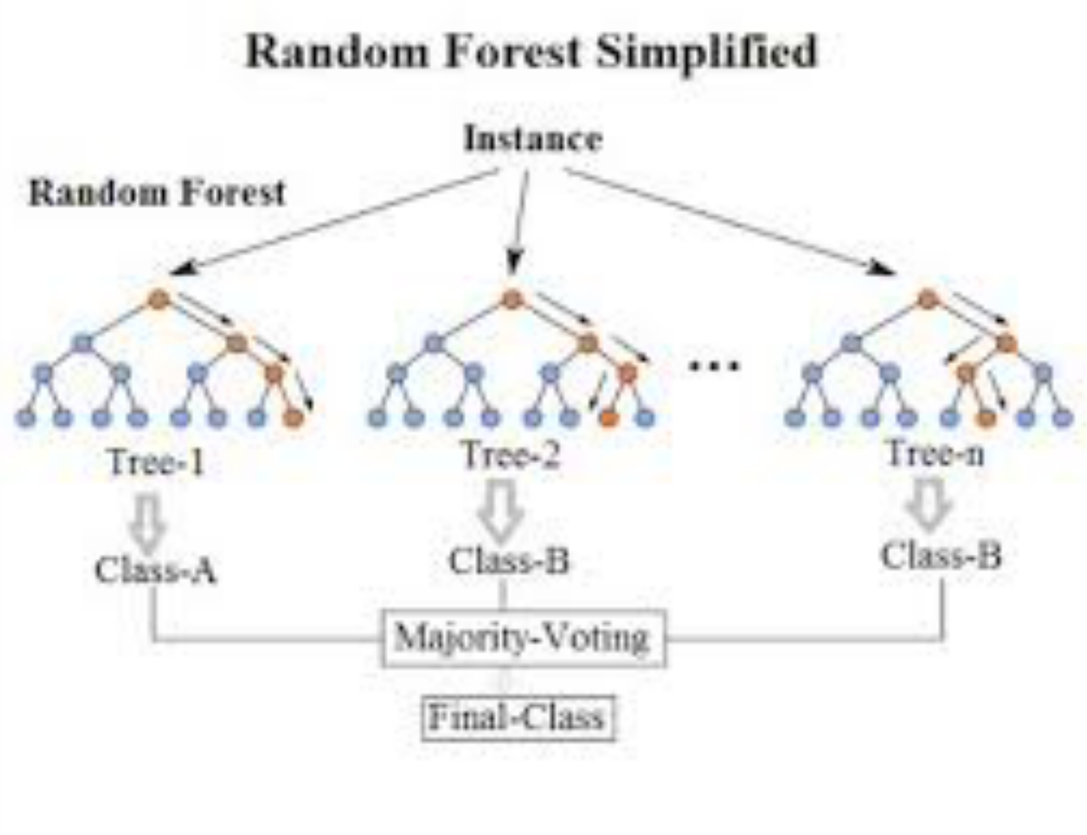
- Create many (e.g. 1000) random sub-samples of our dataset with replacement (meaning we can select the same value multiple times).
- Calculate the mean of each sub-sample.
- Calculate the average of all of our collected means and use that as our estimated mean for the data.



Bootstrap Aggregation

Also called bagging

Bootstrap Aggregation (or Bagging for short), is a simple and very powerful ensemble method. An ensemble method is a technique that combines the predictions from multiple machine learning algorithms together to make more accurate predictions than any individual model



Create many (e.g. 100) random sub-samples of our dataset with replacement.

Train a CART model on each sample.

Given a new dataset, calculate the average prediction from each model.

if we had 5 bagged decision trees that made the following class predictions for an input instance: blue, blue, red, blue and red, we would take the most frequent class and predict blue

Boosting

Boosting is an ensemble technique that attempts to create a strong classifier from a number of weak classifiers

- Build a model from training data using weights.
- Predictors are made sequentially.
- Create a second model which corrects errors from the first model.
- Each model has a stage value
- Repeats until the training set Is predicted perfectly or max models reached.

AdaBoost – most popular for binary classification.

For example, 5 weak classifiers may predict the values 1.0, 1.0, -1.0, 1.0, -1.0.

From a majority vote, it looks like the model will predict a value of 1.0 or the first class.

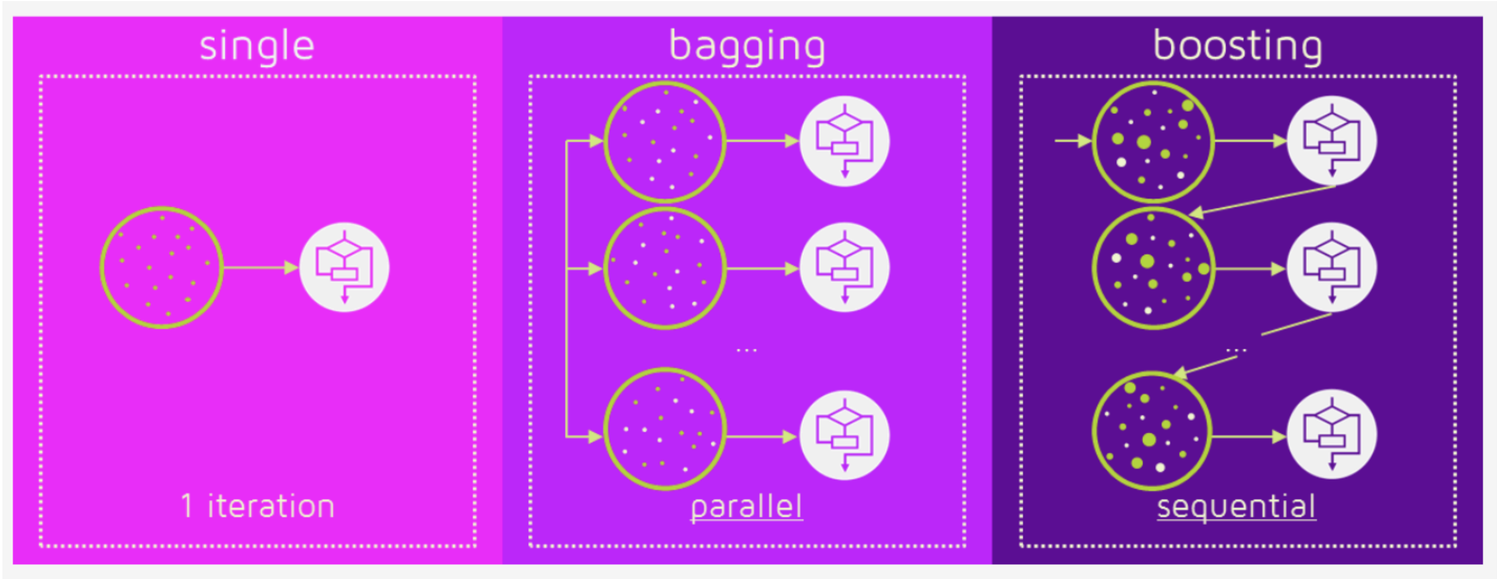
These same 5 weak classifiers may have the stage values 0.2, 0.5, 0.8, 0.2 and 0.9 respectively.

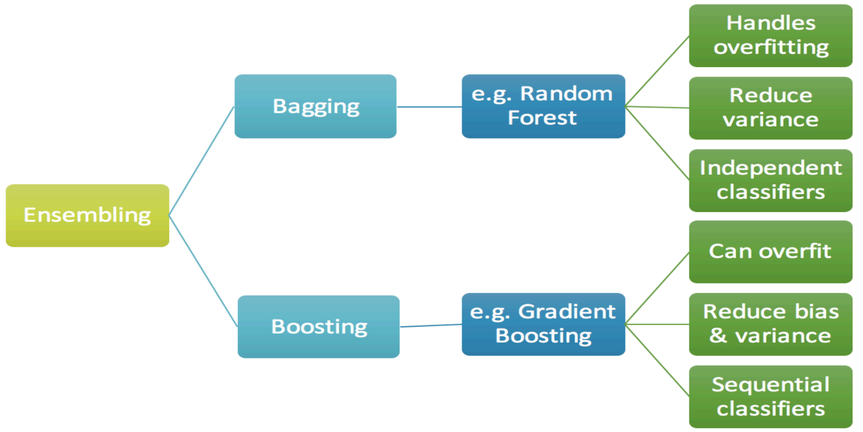
Calculating the weighted sum of these predictions results in an output of -0.8, which would be an ensemble prediction of -1.0 or the second class.

Bagging and Boosting

- Bagging creates variance in data by sampling and replacing data.
- Each hypothesis has it own weight
- Runs in parallel and then votes finally.

- Boosting utilises weighted averages to make weak learners into stronger learners.
- Boosting is about teamwork
- Runs one at a time
- It weights each model and tracks which data samples are successful and which are not. Ones with more errors are trained further.
- Models with better outcomes have stronger pull on final output.





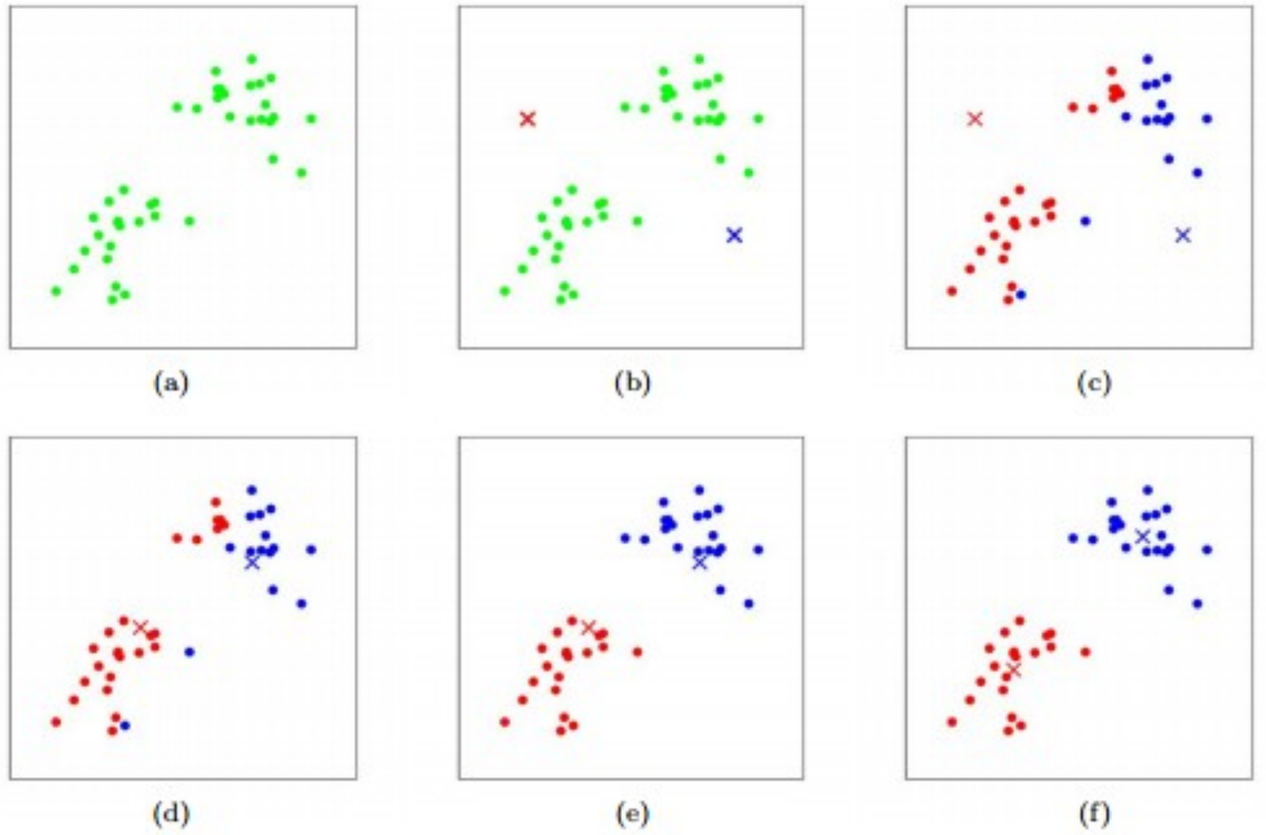
UnSupervised Learning

– look at unlabeled data and find general patterns

– more subjective and difficult to interpret

K-means clustering

- K-means is a type of clustering algorithm that find k clusters for a given dataset. k is a user defined parameter.
- Each cluster is described by a point named centroid, which is the center of the cluster
- The algorithm works like this:
    1. Create k centroid (often randomly)
    2. Each point in the dataset are assign to a cluster whose centroid is closest the point.
    3. Calculate the new centroid for each cluster by take the mean value of the points in the cluster.
    4. Iterate step 2 & 3 until none of the points change its cluster (or no centroid is changed).



Dimensionality Reduction

Dimensionality reduction looks a lot like compression. This is about trying to reduce the complexity of the data while keeping as much of the relevant structure as possible

To select the most significant principal components, we look at how much of the data's variance they capture and order them by that metric.

Another way of thinking about this is that PCA remaps the space in which our data exists to make it more compressible. The transformed dimension is smaller than the original dimension.

Re-inforcement Learning

Learning Agent

A finite set of states

A set of actions available

Transition between states

Rewards with each transition

Immediate and future rewards

Memorylessness

Exploratory Data analysis

- Slice and dice data to get a feel for it
- Visualise the data looking at patterns
- Bivariate statistics and plots can hint at relationships between variables
- Use statistical summaries to see if the data makes sense or if outliers are present
- Examine missing values

Visual EDA using pandas and matplotlib – scatter_matrix, Head, Info, Describe, Names , data types, Columns etc

Machine learning ⊆ artificial intelligence

ARTIFICIAL INTELLIGENCE

Design an intelligent agent that perceives its environment and makes decisions to maximize chances of achieving its goal.
Subfields: vision, robotics, machine learning, natural language processing, planning, …

MACHINE LEARNING

Gives "computers the ability to learn without being explicitly programmed" (Arthur Samuel, 1959)

SUPERVISED LEARNING

Classification, regression

UNSUPERVISED LEARNING

Clustering, dimensionality reduction, recommendation

REINFORCEMENT LEARNING

Reward maximization

Step to machine learning

- Gathering data
- Exploring / Preparing that data
- Choosing a model
- Training
- Evaluation
- Hyper-parameter tuning
- Prediction.

Performance Tuning

- Type of problem
- Data quality and distribution
- Algorithm used for the model
- Data used to train
- Weights used to train (learnt)
- Parameters used to train (hyper-parameters)

Measuring Accuracy

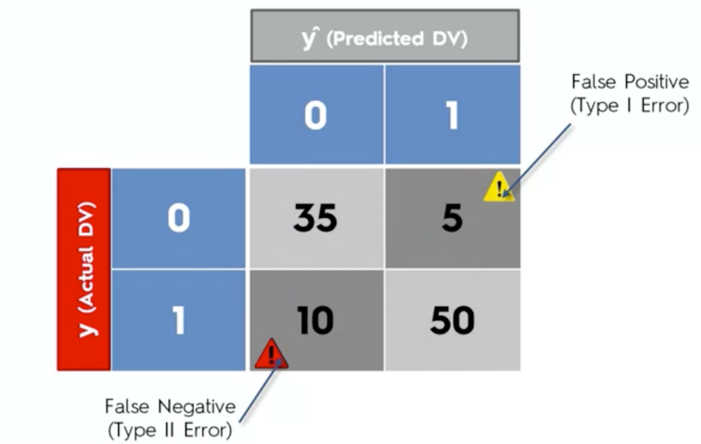
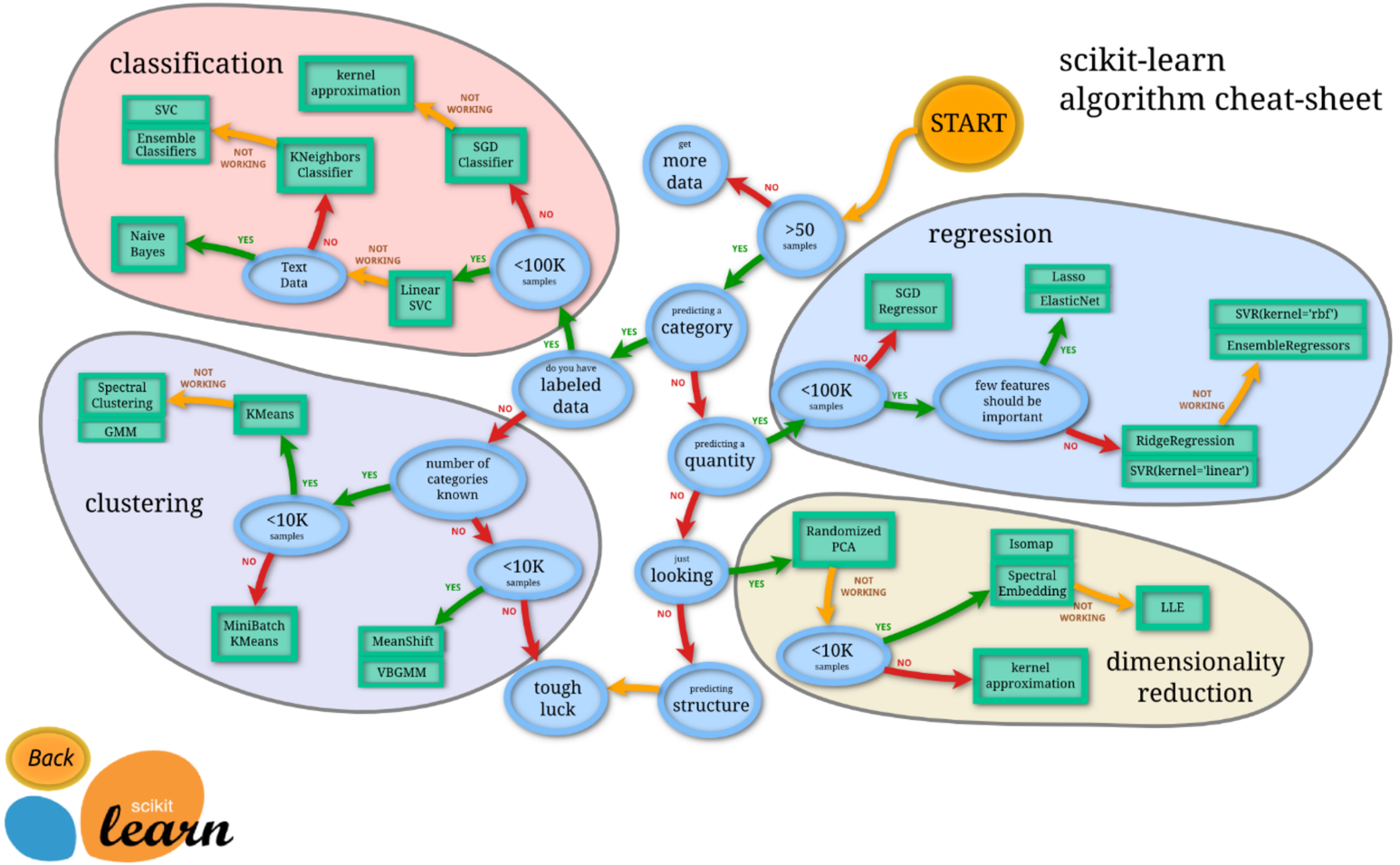


Table 3
Measures for multi-class classification based on a generalization of the measures of Table 1 for many classes C_i: tp_i are true positive for C_i, and fp_i – false positive, fn_i – false negative, and tn_i – true negative counts respectively. μ and M indices represent micro- and macro-averaging.

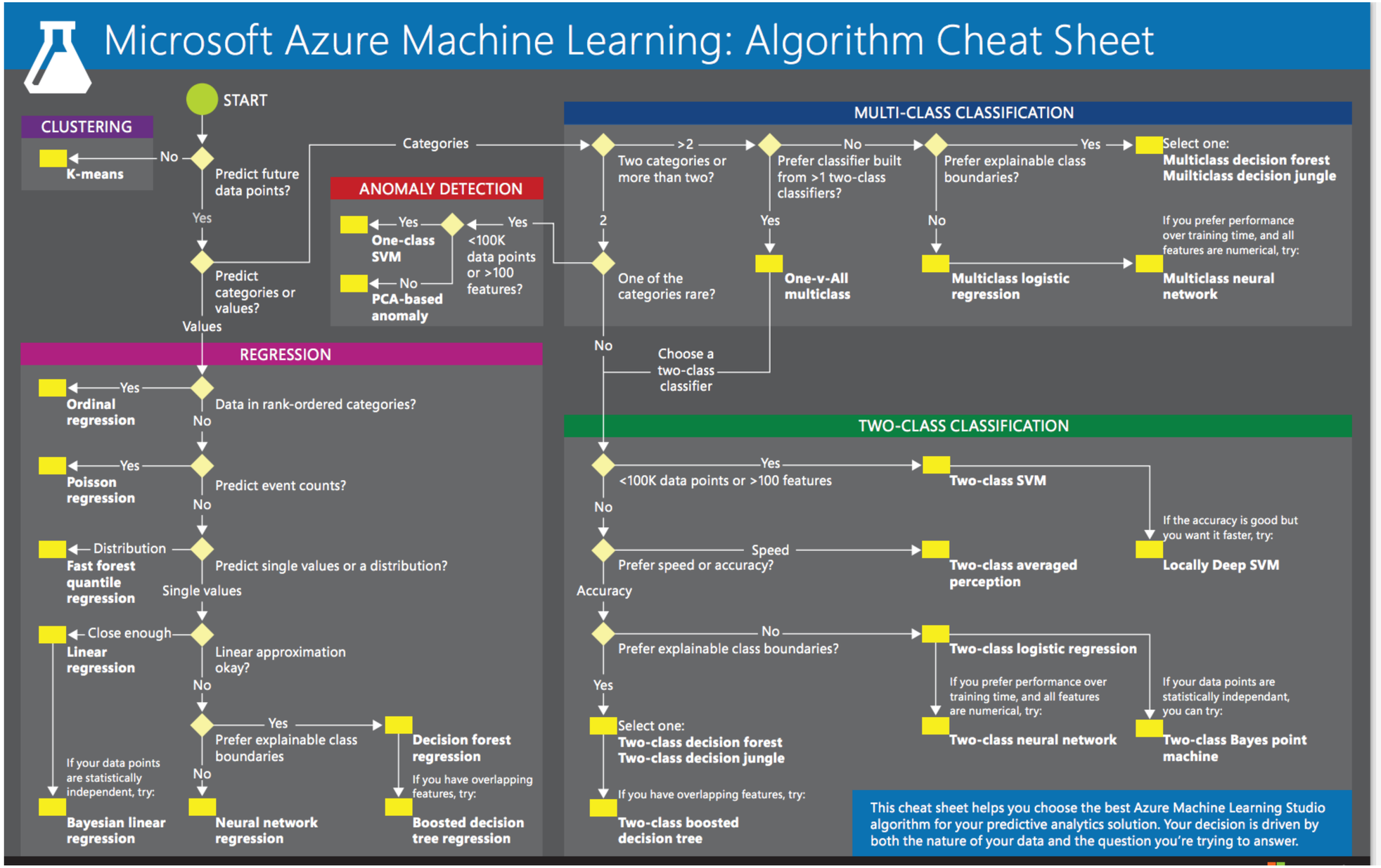
Measure	Formula	Evaluation focus
Average Accuracy	$\frac{\sum_{i=1}^{l}\frac{tp_i+tn_i}{tp_i+fn_i+fp_i+tn_i}}{l}$	The average per-class effectiveness of a classifier
Error Rate	$\frac{\sum_{i=1}^{l}\frac{fp_i+fn_i}{tp_i+fn_i+fp_i+tn_i}}{l}$	The average per-class classification error
$Precision_\mu$	$\frac{\sum_{i=1}^{l}tp_i}{\sum_{i=1}^{l}(tp_i+fp_i)}$	Agreement of the data class labels with those of a classifiers if calculated from sums of per-text decisions
$Recall_\mu$	$\frac{\sum_{i=1}^{l}tp_i}{\sum_{i=1}^{l}(tp_i+fn_i)}$	Effectiveness of a classifier to identify class labels if calculated from sums of per-text decisions
$Fscore_\mu$	$\frac{(\beta^2+1)Precision_\mu Recall_\mu}{\beta^2 Precision_\mu+Recall_\mu}$	Relations between data's positive labels and those given by a classifier based on sums of per-text decisions
$Precision_M$	$\frac{\sum_{i=1}^{l}\frac{tp_i}{tp_i+fp_i}}{l}$	An average per-class agreement of the data class labels with those of a classifiers
$Recall_M$	$\frac{\sum_{i=1}^{l}\frac{tp_i}{tp_i+fn_i}}{l}$	An average per-class effectiveness of a classifier to identify class labels
$Fscore_M$	$\frac{(\beta^2+1)Precision_M Recall_M}{\beta^2 Precision_M+Recall_M}$	Relations between data's positive labels and those given by a classifier based on a per-class average

https://www.sciencedirect.com/science/article/pii/S0306457309000259

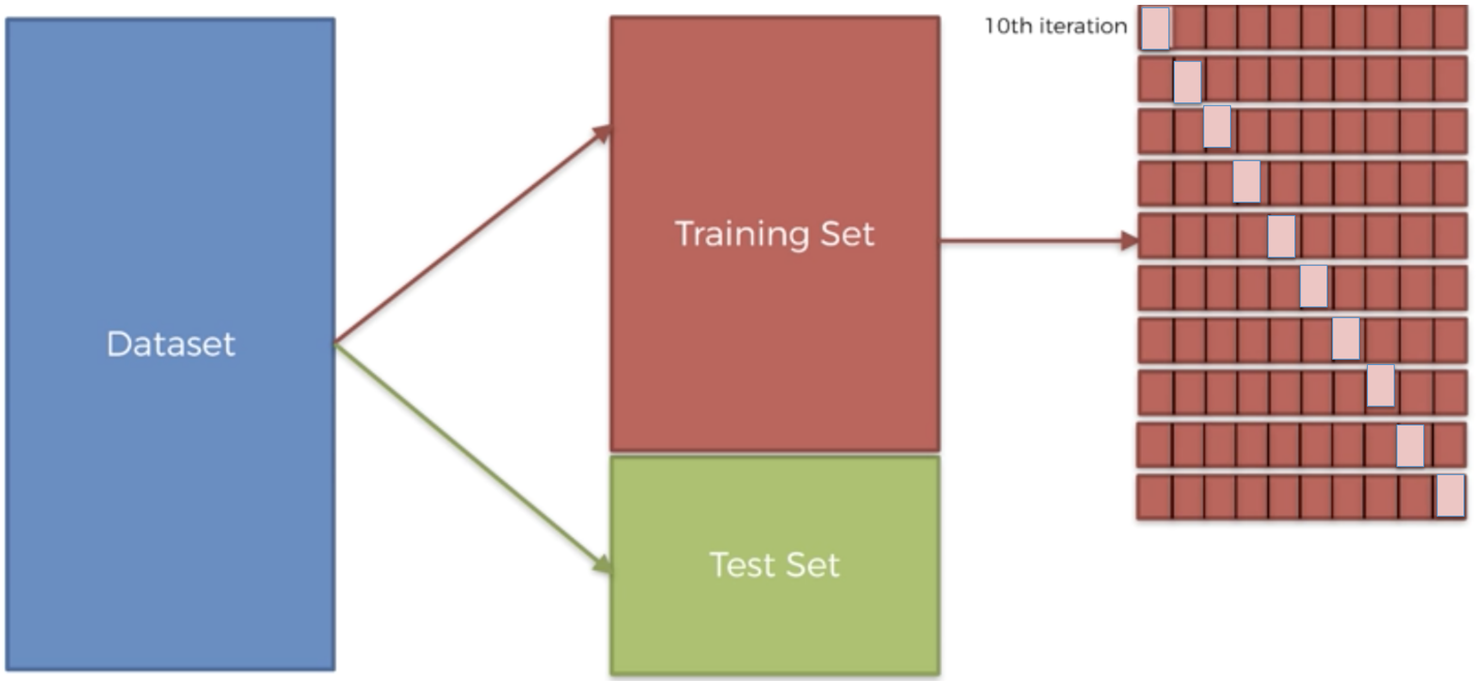
Choosing a ML Algorithm



Choosing a ML Algorithm



K-Fold Cross Validation



k = 10
randomly partitioned into k equal samples
Single subsample for training
Cross-validation repeats k times(folds)
Each k subsample used only once for validation
Mean of the results can be used to measure results

```
From sklearn.model_selection import cross_val_score
# will contain the 10 accuracies.
Accuracies =  cross_val_score(estimator = "classifier", X= X_train, y = y_train,
cv=10)
#Take the mean of all accuracies.
Accuracies.mean()
Accuracies.std()
```

HyperParameter tuning

Grid search is arguably the most basic hyperparameter tuning method.

With this technique, we simply build a model for each possible combination of all of the hyperparameter values provided, evaluating each model, and selecting the architecture which produces the best results

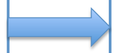
```
From skilearn.model_selection import GridSearchCV
Based on the model you are using.

parameters = {'kernel':('linear', 'rbf'), 'C':[1, 10]}

grid_search = GridSearchCV(estimator=classifier, param_grid=parameters,
scoring="accuracy", cv = 10, n_jobs=-1)
Grid_search = grid_search.fit(X_train, y_train)
Best_accuracy = grid_search.best_score_
Best_parameters = grid_search.best_params_
```

Data Science Steps

Know your data
- Summary
- Stats
- Visualizations

Clean your data
Augment your data
Categorise your problem
- Input
- Output

Understand constraints
Find the algorithms which would work well with your data.



Train the chosen algorithms with your data

Do a performance evaluation.

Do k-fold cross validation

Try hyper-parameter tuning to improve model prediction

Deploy and Productionise your model.

Don’t forget - Business requirements, rules and regulations and stakeholder management!