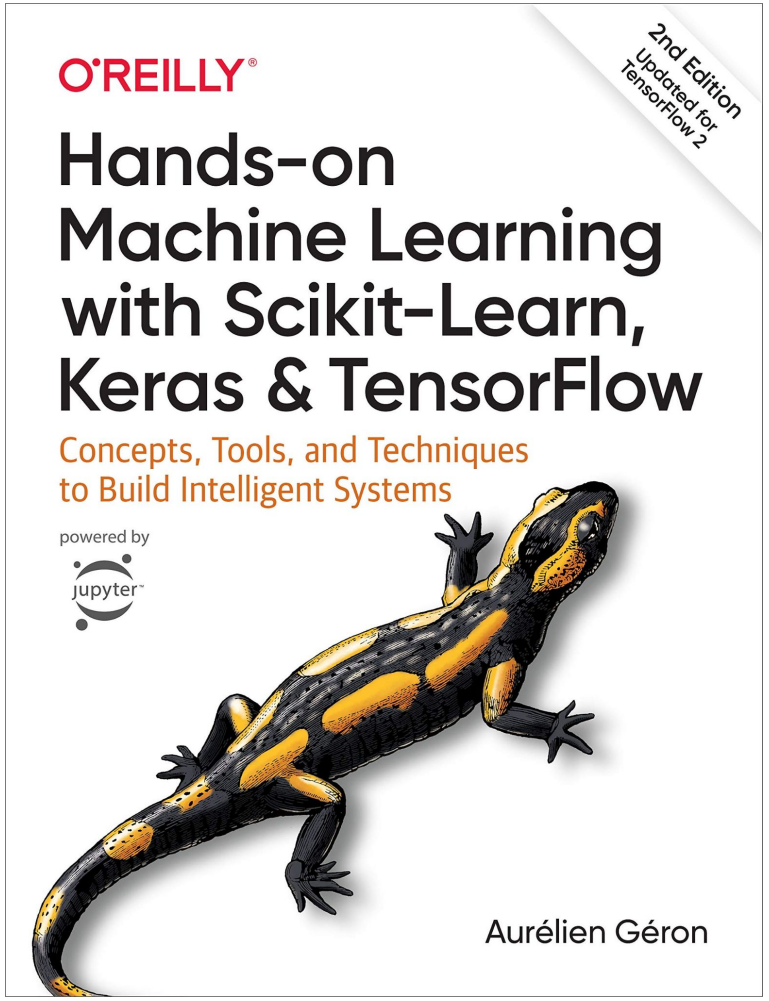
Applied Predictive Analysis

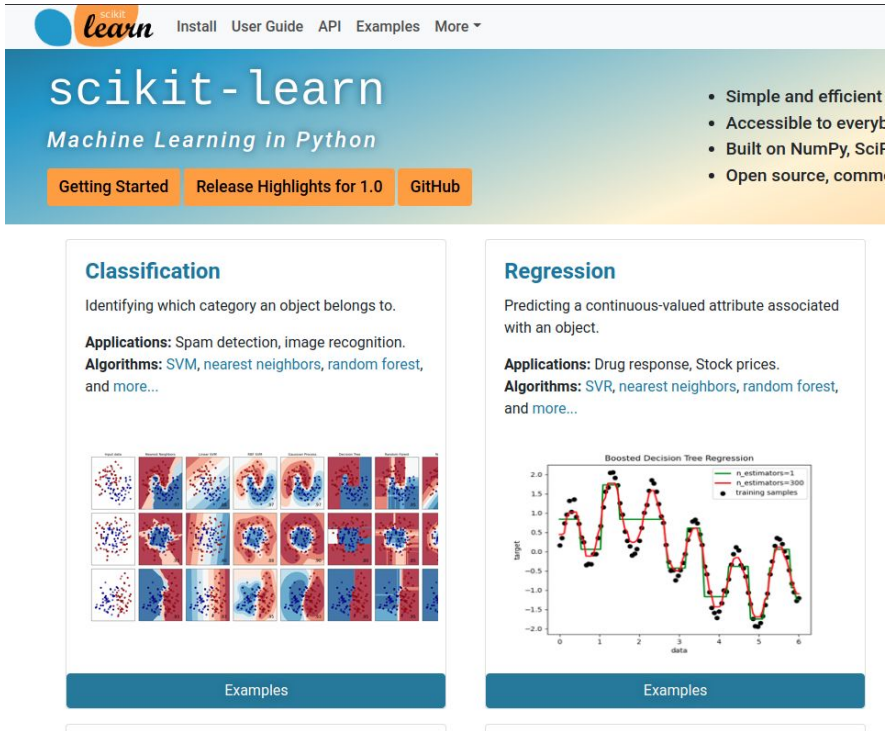
Module 6

Remmy Zen

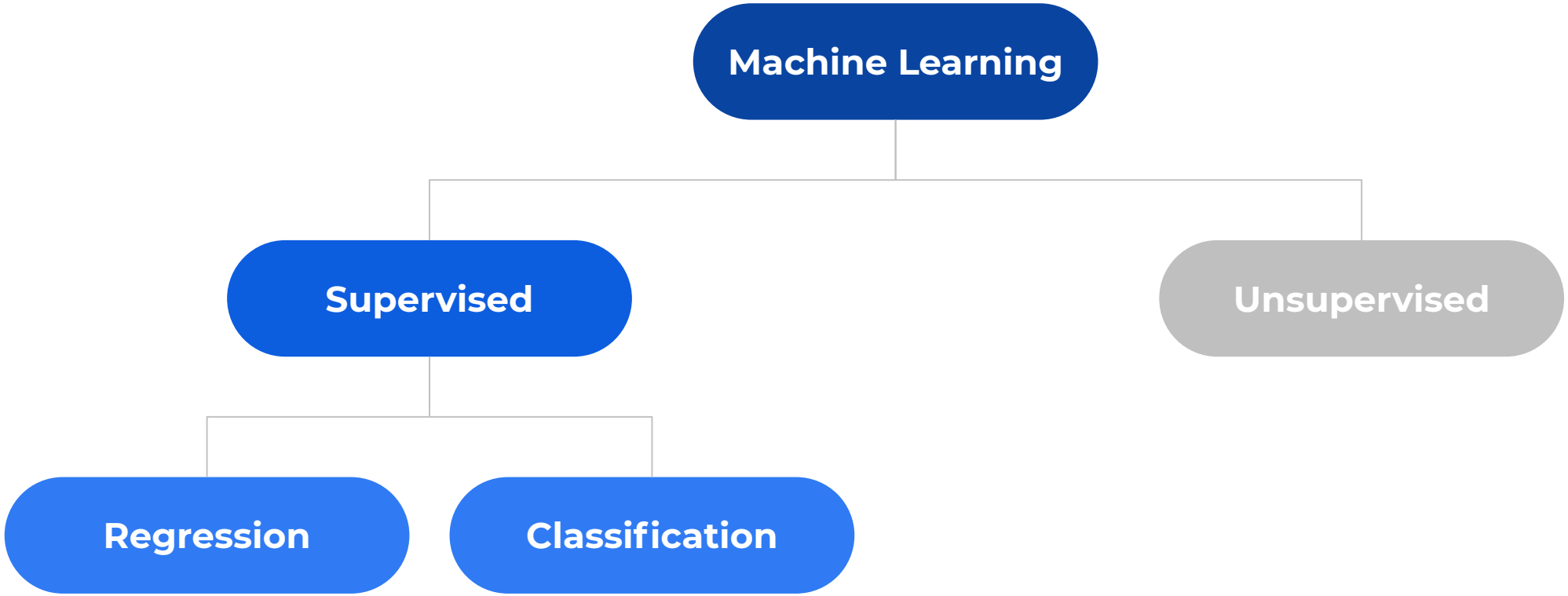
KLASA
By Rajut Indonesia

Recommended Learning Resources



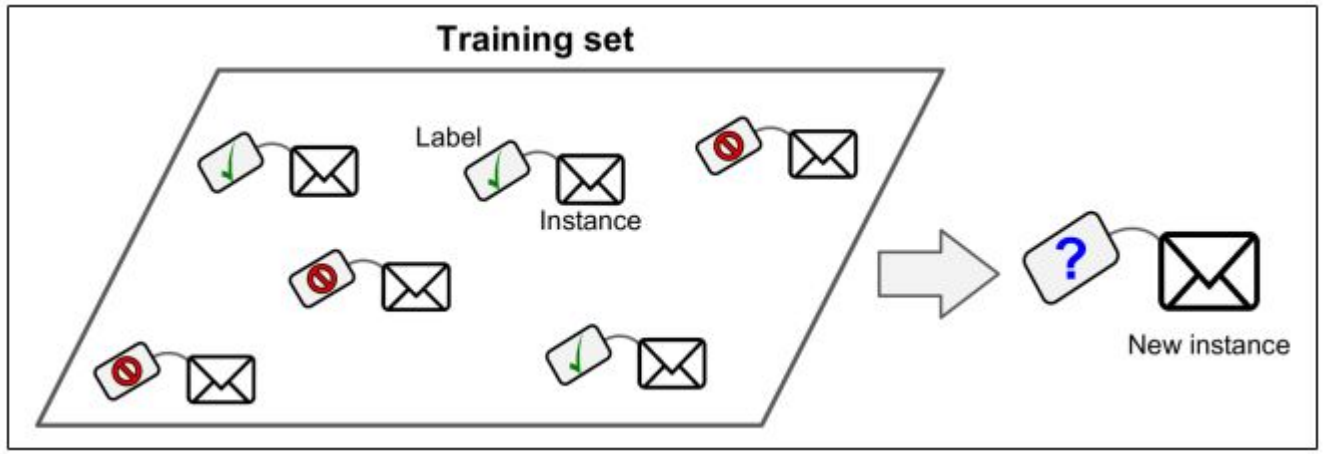


Review: Type of Learning



Supervised Learning

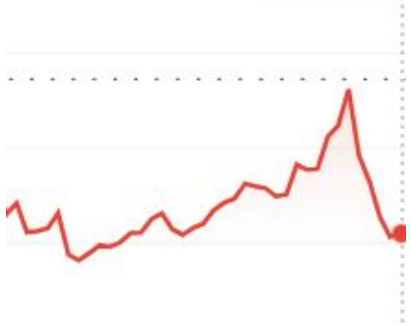
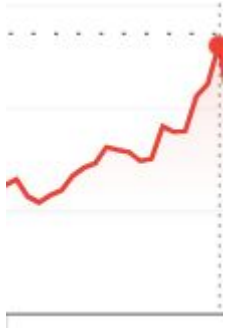
- Approximate/learn the function f(**x**) = **y** from a given data.
- Data (**x**) and target output / label (**y**) is given.
- Type:
  - **Regression**: y is real value
  - **Classification**: y is categorical value



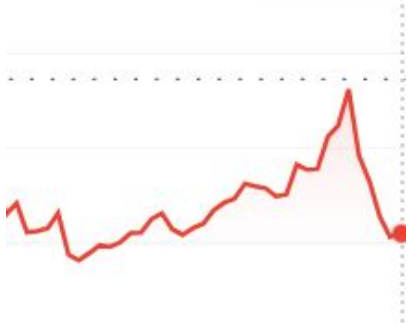
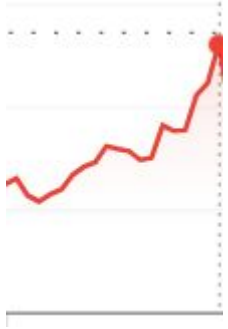
Regression or Classification?

Data **(x)**	Label **(y)**
	**cat**
	**rabbit**
	**dog**

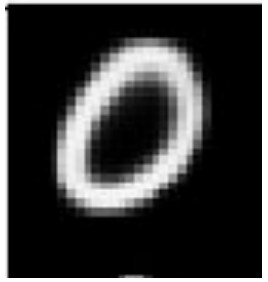
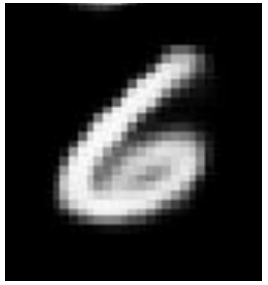
Regression or Classification?

Data **(x)**	Label **(y)**
	**buy**
	**buy**
	**sell**

Regression or Classification?

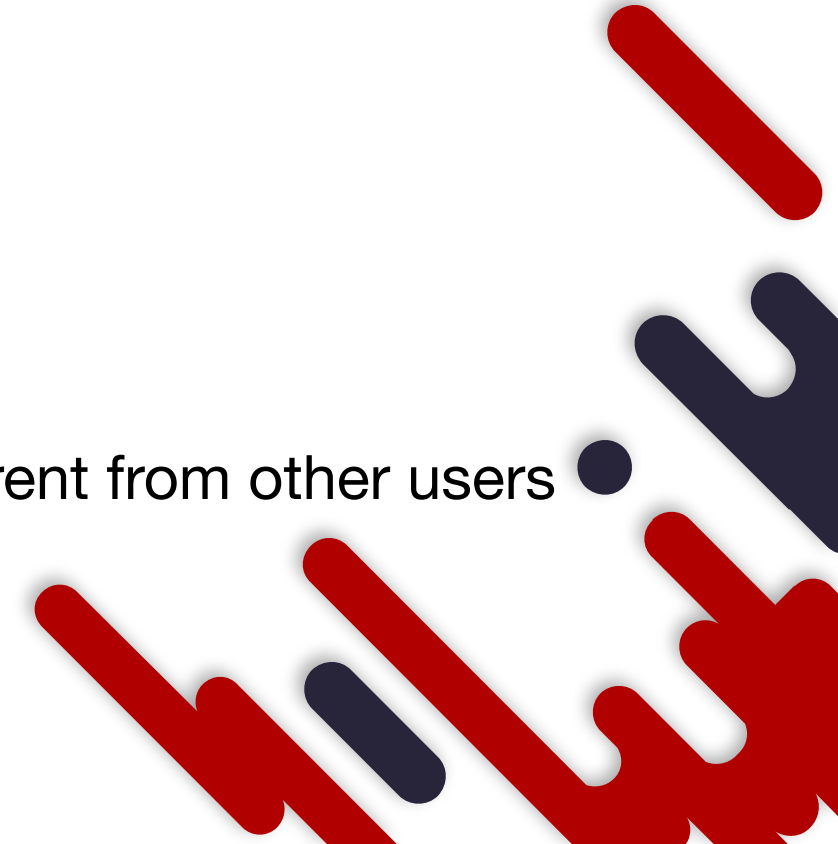
Data **(x)**	Label **(y)**
	**US$ 5.12**
	**US$ 6.33**
	**US$ 12.77**

Regression or Classification?

Data **(x)**	Label **(y)**
	**0**
	**6**
	**7**

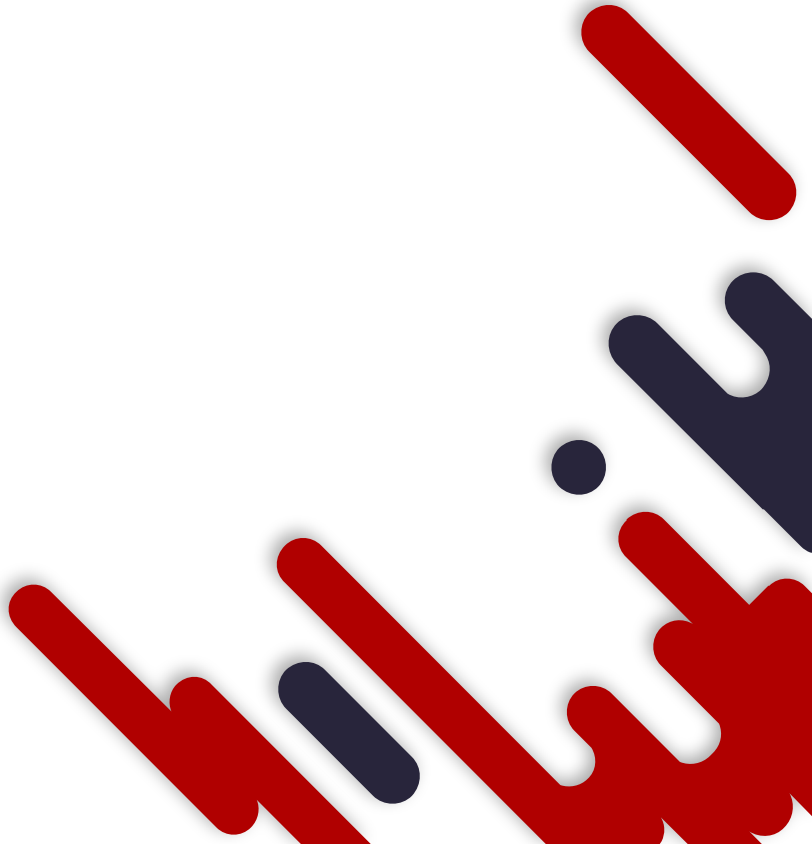
When to use Supervised Learning?

- No human expert.
  - Example: Given a new molecule predict its binding strength.
- Human can perform task but unable to describe how to do it.
  - Example: Handwritten character classification
- Function that changes frequently.
  - Example: predict stock price from past prices
- User requires a customised function
  - Example: Predict important mails for one user is different from other users



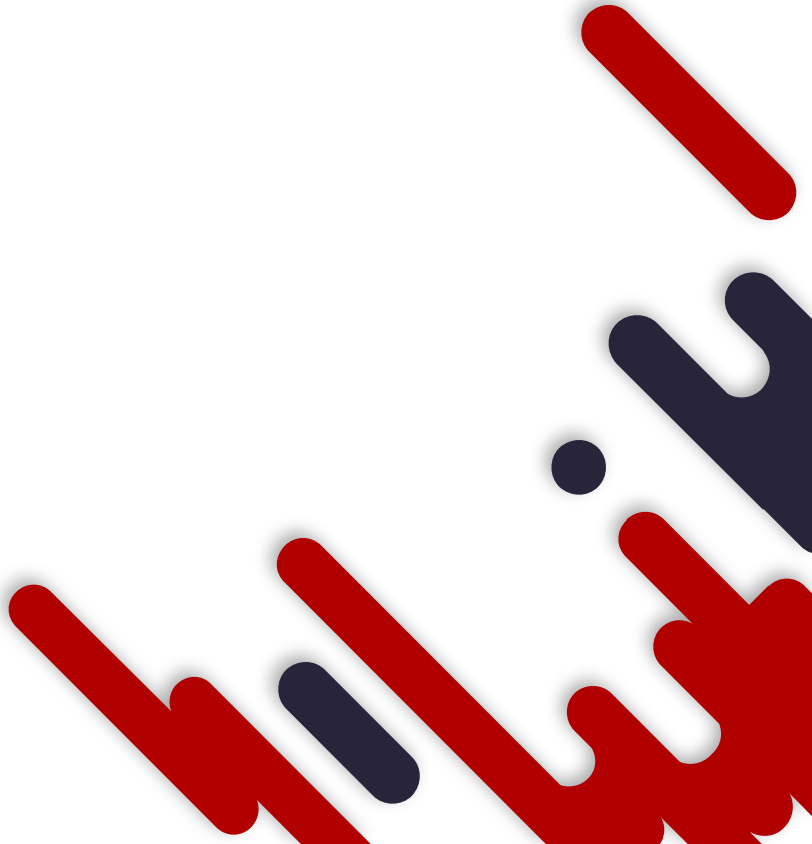
Main Challenges of Supervised Learning

- Insufficient quantity of training data.
- Nonrepresentative training data.
- Poor-quality data.
- Irrelevant features.
- Overfitting/underfitting the training data.



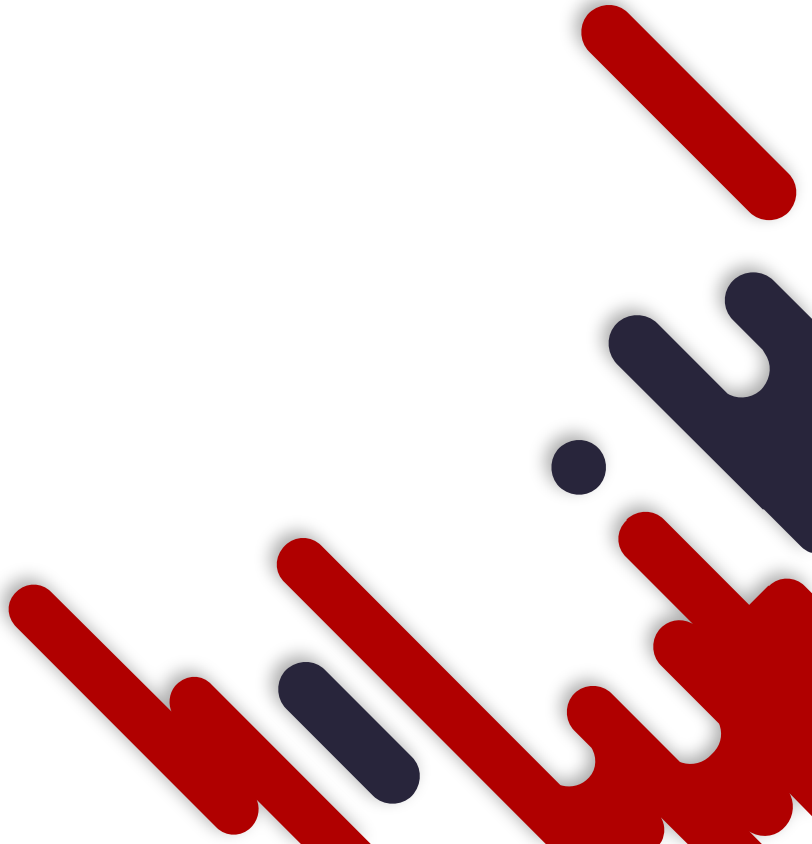
Pipeline

- Data collection
- Data preparation
- Algorithm Selection and Setup
  - Hyperparameter choosing
- Training
- Evaluation



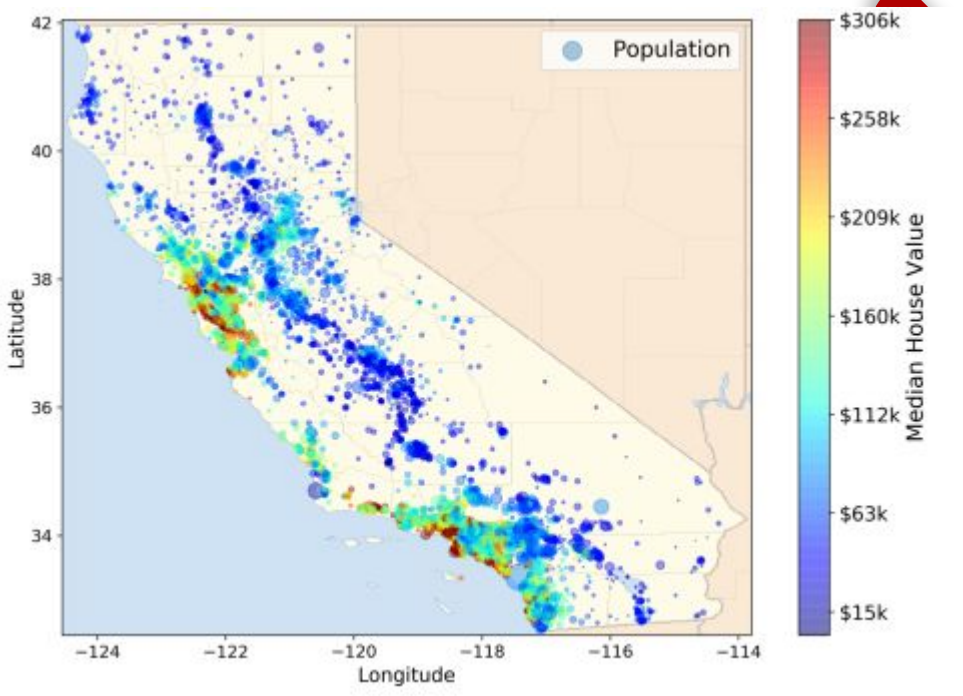
Pipeline

- **Data collection**
- Data preparation
- Algorithm Selection and Setup
  - Hyperparameter choosing
- Training
- Evaluation



Data Collection

- We will use the California Housing Prices that you have seen from Module 3.
- Our goal is to build a **predictive model** of housing prices in the state.
- **Regression** task: predict the house value
- **Binary Classification** task:
  - house value **above** the median
  - house value **below** the median



CALIFORNIA HOUSING

Check the data we have

	MedInc	HouseAge	AveRooms	AveBedrms	Population	AveOccup	Latitude	Longitude	MedianHouseValue
0	8.3252	41.0	6.984127	1.023810	322.0	2.555556	37.88	-122.23	4.526
1	8.3014	21.0	6.238137	0.971880	2401.0	2.109842	37.86	-122.22	3.585
2	7.2574	52.0	8.288136	1.073446	496.0	2.802260	37.85	-122.24	3.521
3	5.6431	52.0	5.817352	1.073059	558.0	2.547945	37.85	-122.25	3.413
4	3.8462	52.0	6.281853	1.081081	565.0	2.181467	37.85	-122.25	3.422

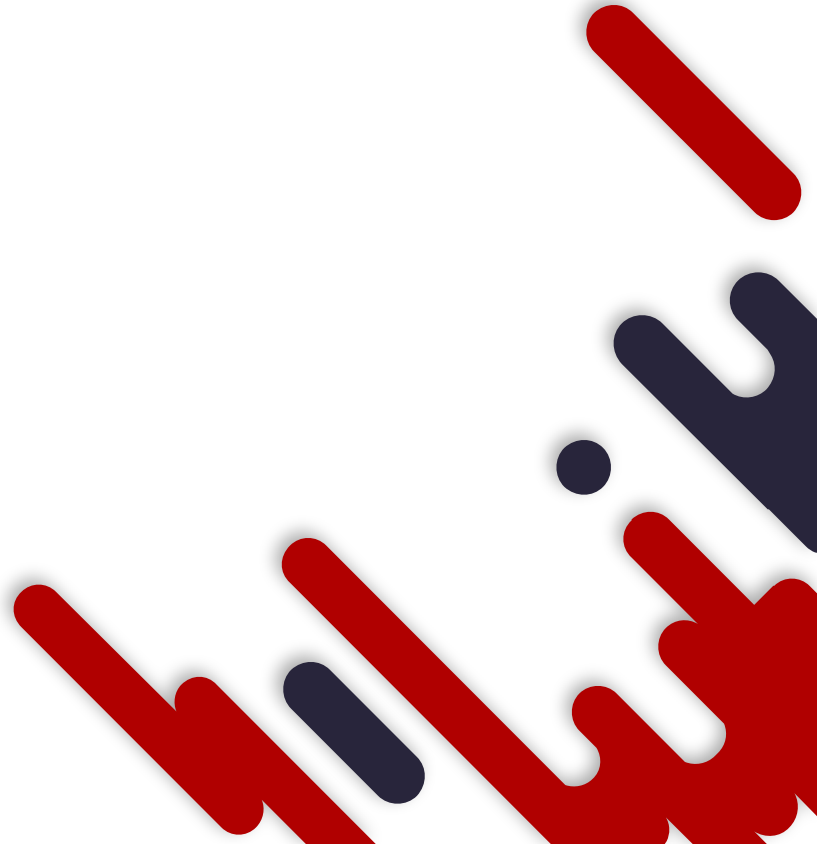
Data Description

- This dataset was derived from the 1990 U.S. census, using one row per census block group
- A block group is the smallest geographical unit for which the U.S. Census Bureau publishes sample data
- A block group typically has a population of 600 to 3,000 people

- **Number of Instances** : 20640
- **Number of Attributes** : 8 numeric, predictive attributes and the target

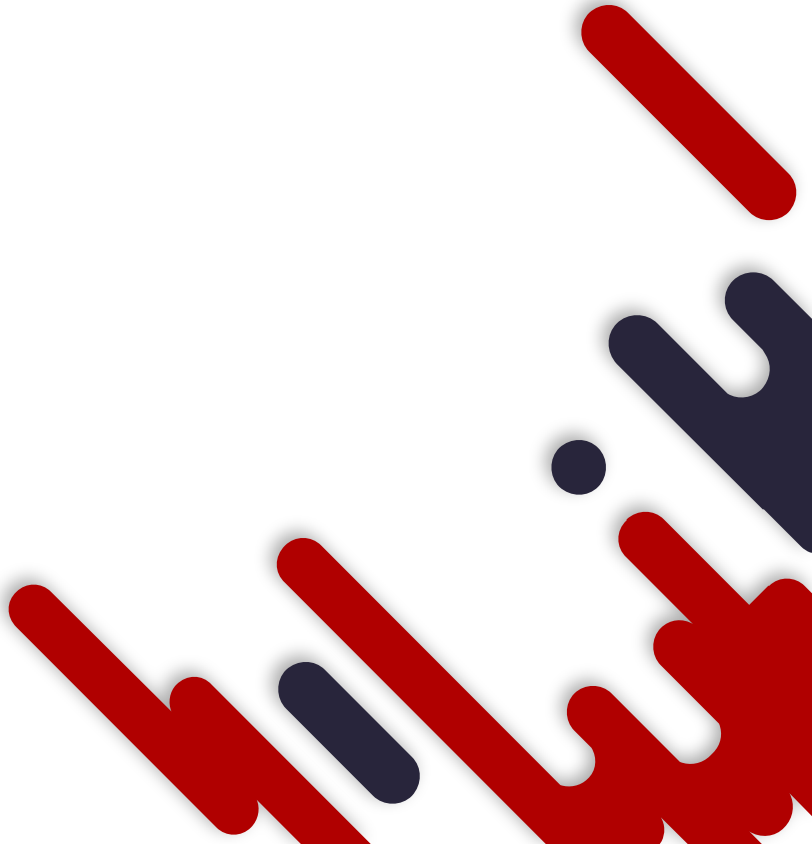
- **Attribute Information:**
  - MedInc median income in block
  - HouseAge median house age in block
  - AveRooms average number of rooms
  - AveBedrms average number of bedrooms
  - Population block population
  - AveOccup average house occupancy
  - Latitude house block latitude
  - Longitude house block longitude
  - MedianHouseValue median house value

Data Collection Hands-On



Pipeline

- Data collection
- **Data preparation**
- Algorithm Selection and Setup
  - Hyperparameter choosing
- Training
- Evaluation

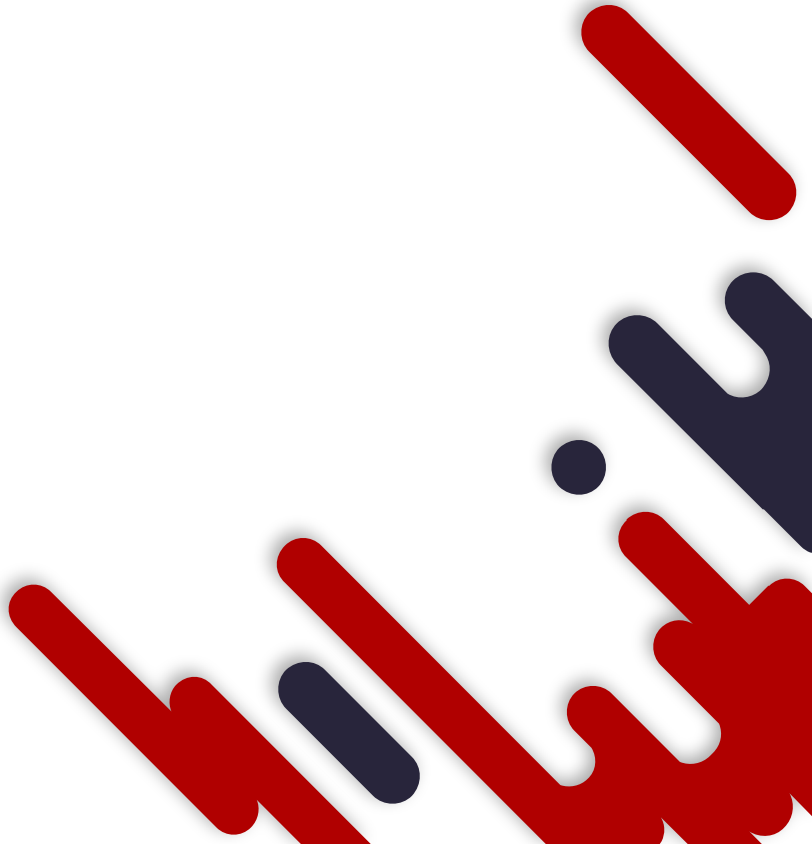


Data Preparation

- Data Cleaning
  - Fix or remove outliers (optional).
  - Fill in missing values (with zero, mean, median, …) or drop the rows
- Feature Selection (optional)
  - Drop the attributes that provide no useful information for the task

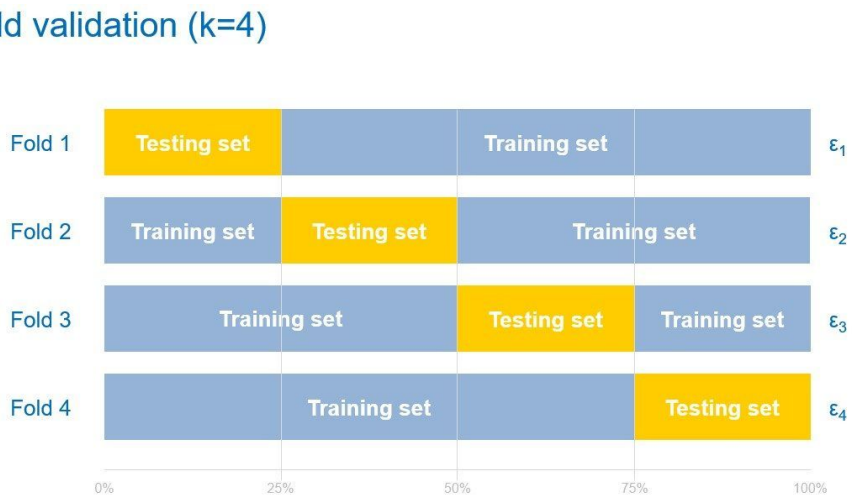
Data Preparation

- Feature engineering, if needed
  - Discretize continuous features.
  - Decompose features (categorical, date/time, etc.).
  - Add transformation of features (e.g. log(x), sqrt(x), …).
    - Example: ocean proximity
  - Aggregate features into new features.
- Feature scaling
  - Standardize or normalize features.

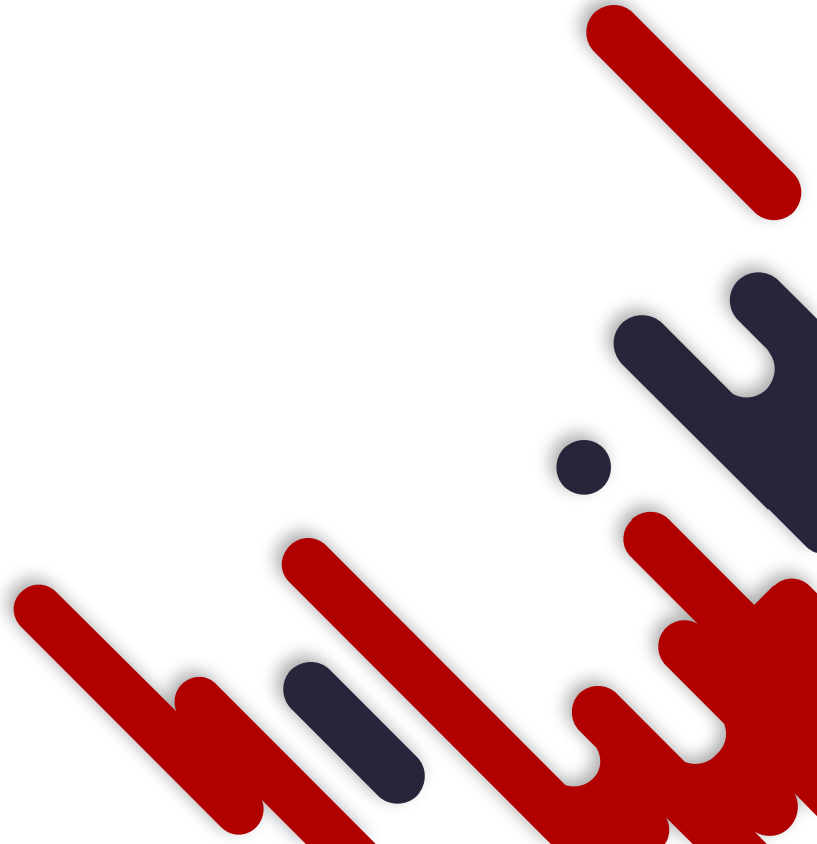


Data Preparation

- We usually split the data into several parts:
  - **Training**: use to train the data.
  - **Validation** (optional): use to tune the hyperparameter.
  - **Testing**: use to test the trained model on unseen data.
- Need to determine the percentage, usually 80:20 training:testing.
- **Holdout**: Split dataset for training and testing once.
- **Cross Validation:**
- **Combination**

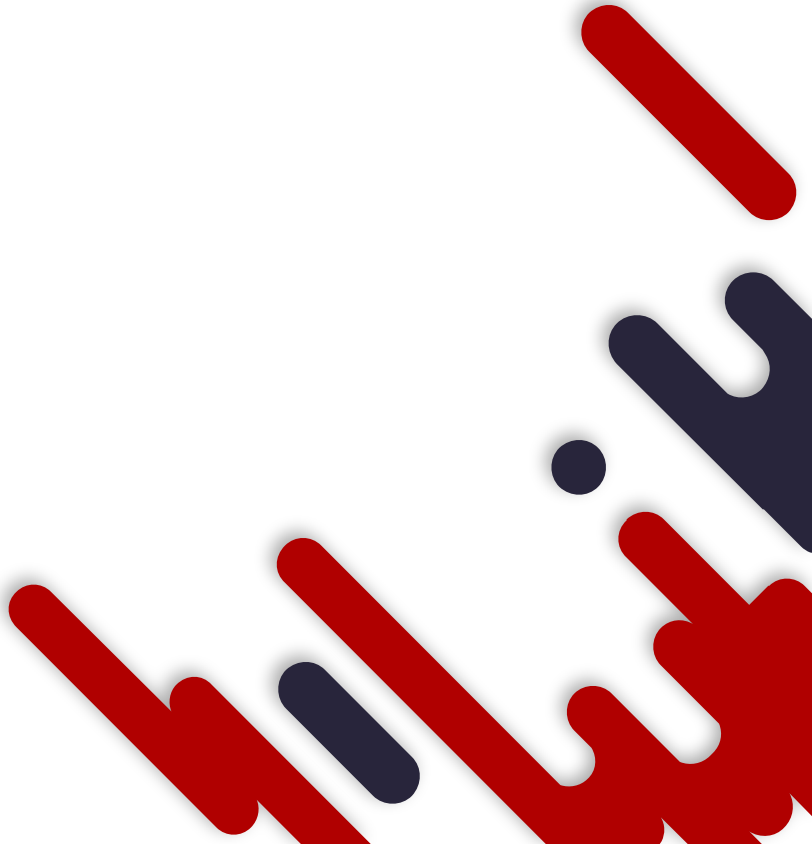


Data Preparation Hands-On



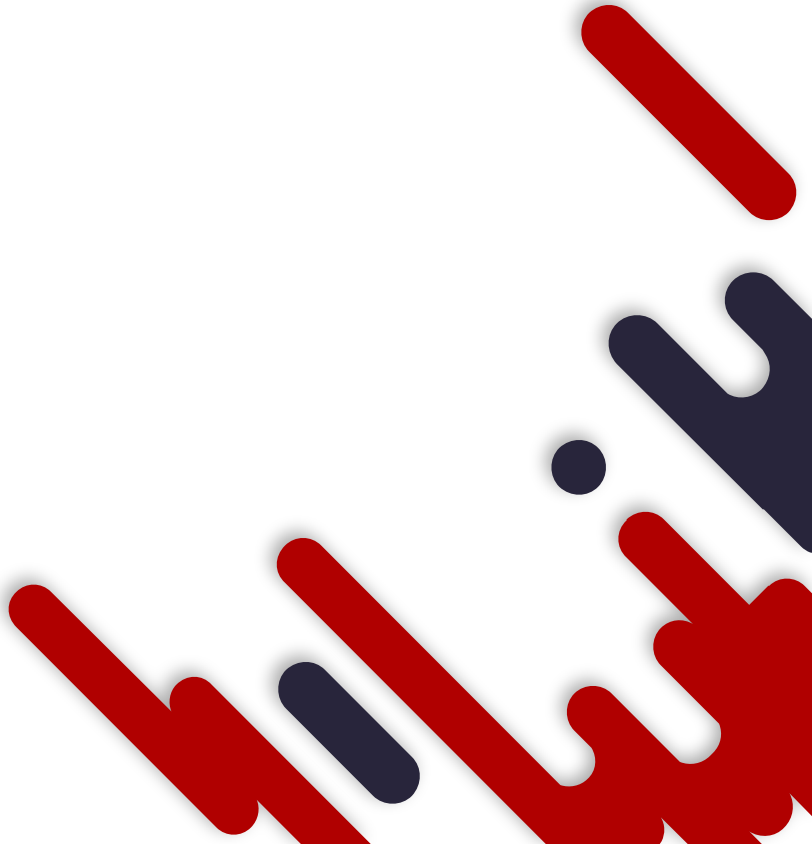
Pipeline

- Data collection
- Data preparation
- **Algorithm Selection and Setup**
  - Hyperparameter choosing
- Training
- Evaluation



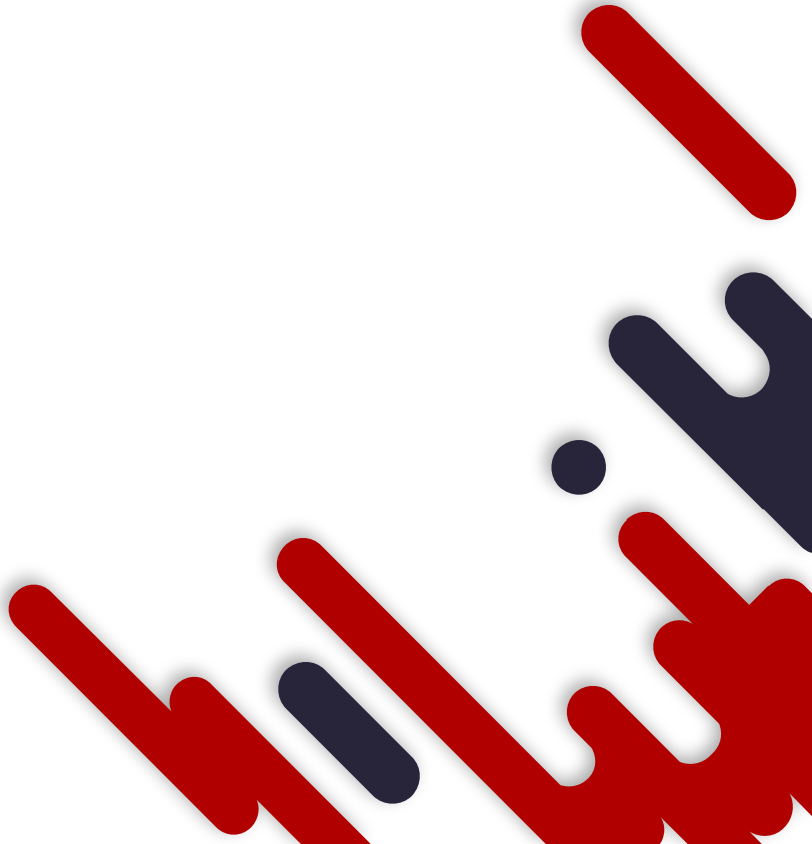
Supervised Learning Algorithms

- Linear regression and logistic regression.
- K-Nearest Neighbours.
- Decision Tree and Random Forests.
- Support Vector Machines
- Neural Networks.



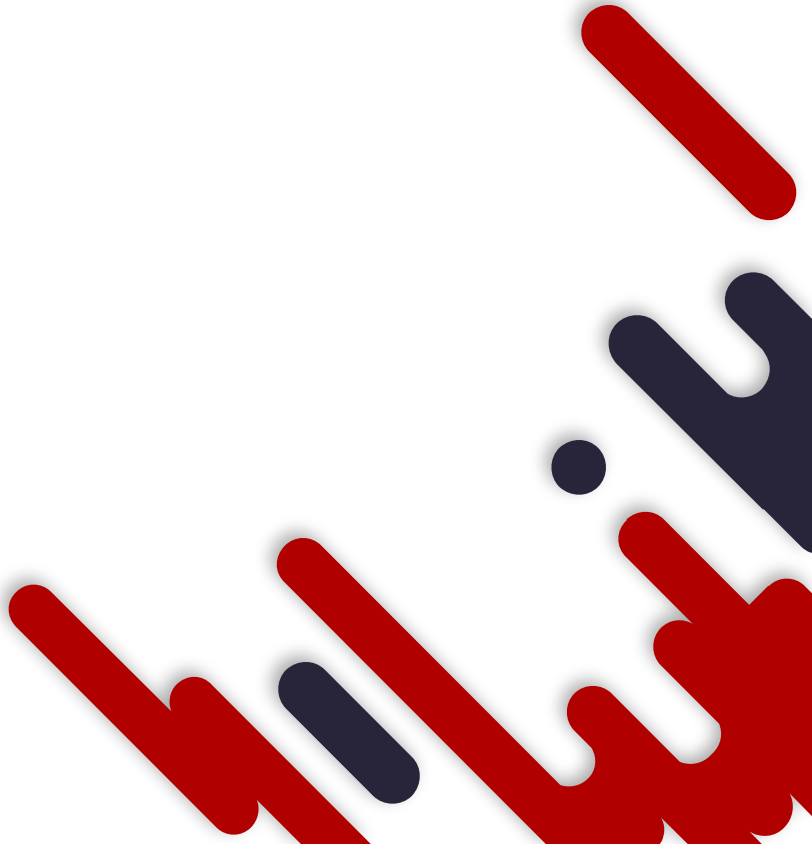
Pipeline

- Data collection
- Data preparation
- Algorithm Selection and Setup
  - Hyperparameter choosing
- **Training**
- Evaluation



Pipeline

- Data collection
- Data preparation
- Algorithm Selection and Setup
  - Hyperparameter choosing
- Training
- **Evaluation**



Regression Performance Metric

- Mean absolute error
- Mean squared error
- Root mean square error
- R^2
- …
- Read more here:

  https://scikit-learn.org/stable/modules/model_evaluation.html#regression-metrics

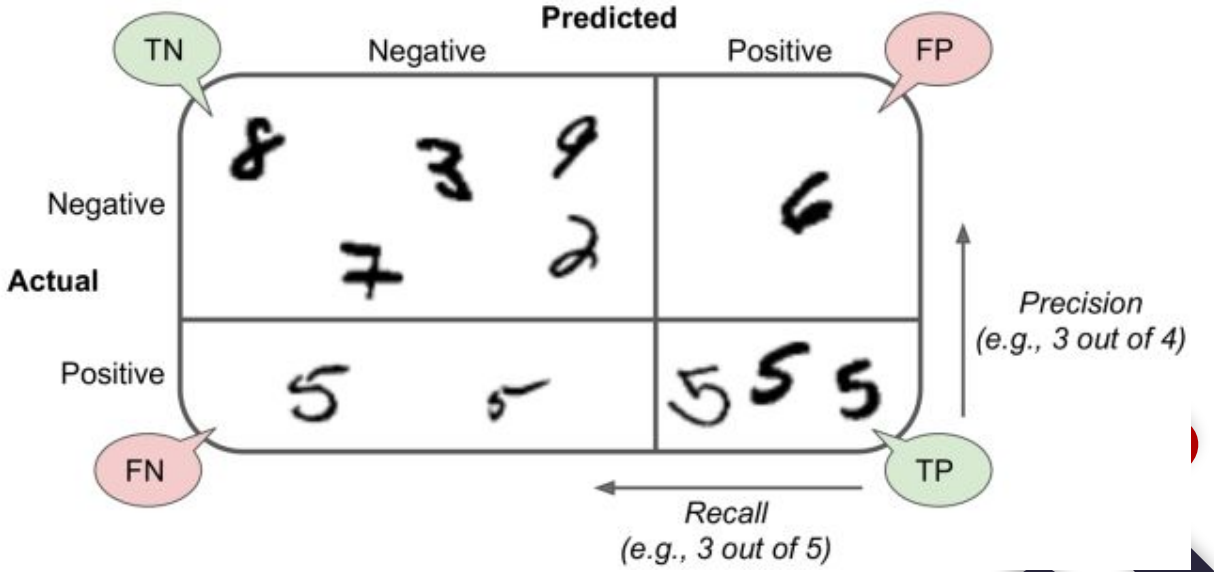
$$\mathrm{MAE}(y, \hat{y}) = \frac{1}{n_{\text{samples}}} \sum_{i=0}^{n_{\text{samples}}-1} |y_i - \hat{y}_i|.$$

$$\mathrm{MSE}(y, \hat{y}) = \frac{1}{n_{\text{samples}}} \sum_{i=0}^{n_{\text{samples}}-1} (y_i - \hat{y}_i)^2.$$

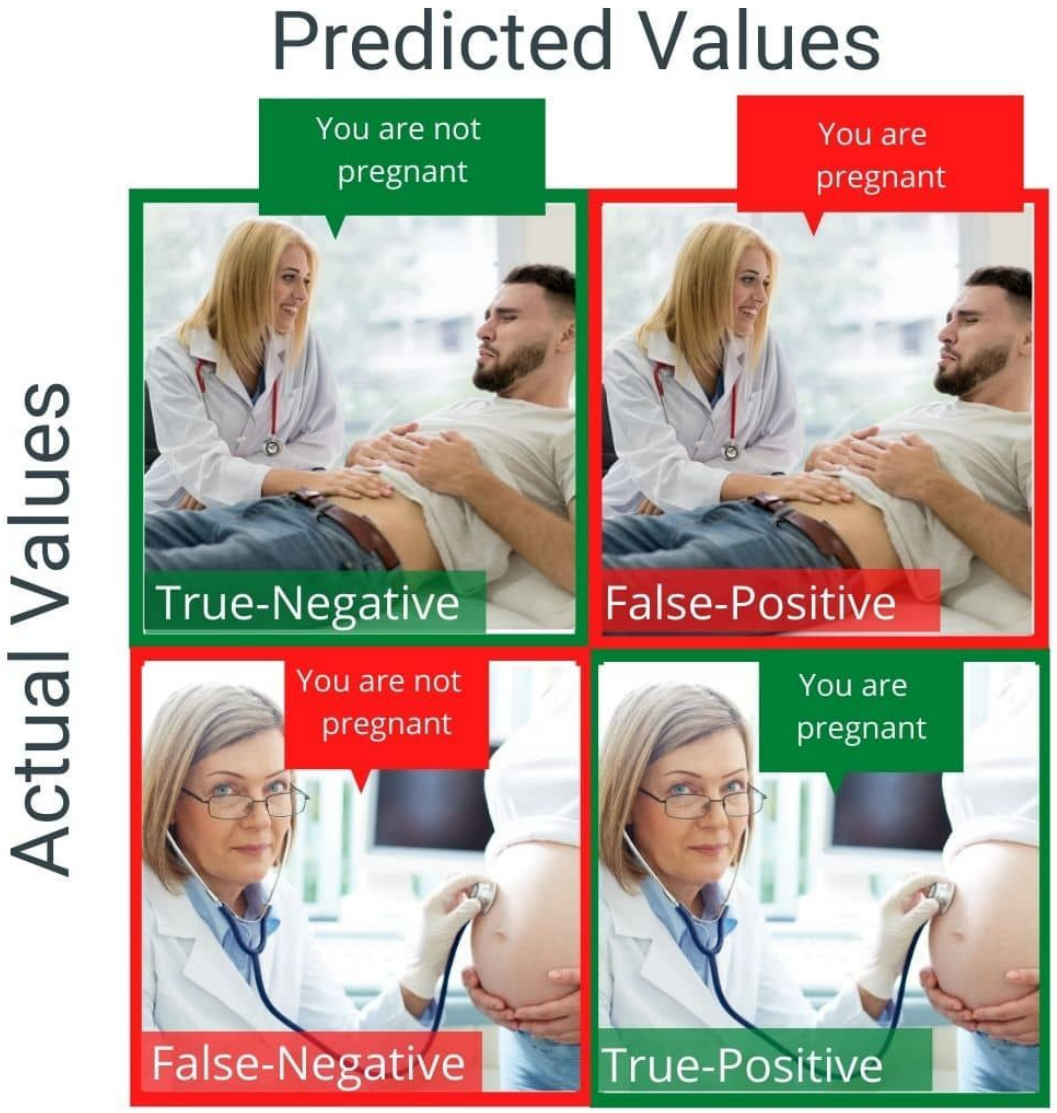
$$R^2(y, \hat{y}) = 1 - \frac{\sum_{i=1}^{n} (y_i - \hat{y}_i)^2}{\sum_{i=1}^{n} (y_i - \bar{y})^2}$$

Classification Performance Metric

- Confusion matrix
- Accuracy = (tp + tn) / (tp + tn + fp + fn)
- Precision = tp / (tp + fp)
- Recall = tp / (tp + fn)
- F1-score = (2 * (prec * rec) / (prec + rec))
- …
- Read more here: https://scikit-learn.org/stable/modules/model_evaluation.html#classification-metrics

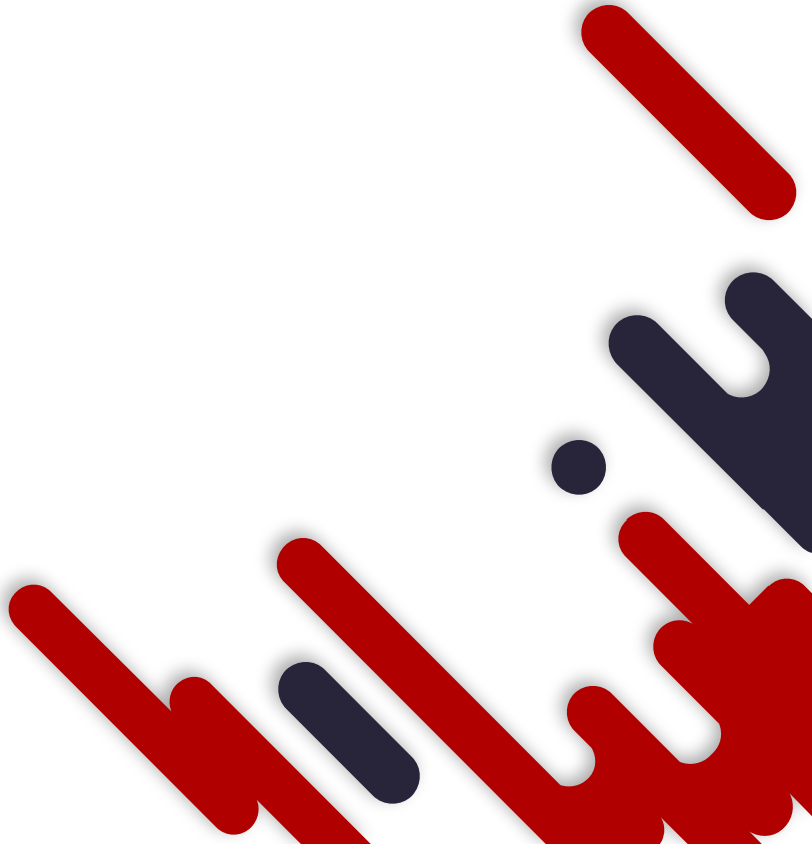


Classification Performance Metric



Supervised Learning Algorithms

- **Linear regression and logistic regression.**
- K-Nearest Neighbours.
- Decision Tree and Random Forests.
- Support Vector Machines
- Neural Networks.



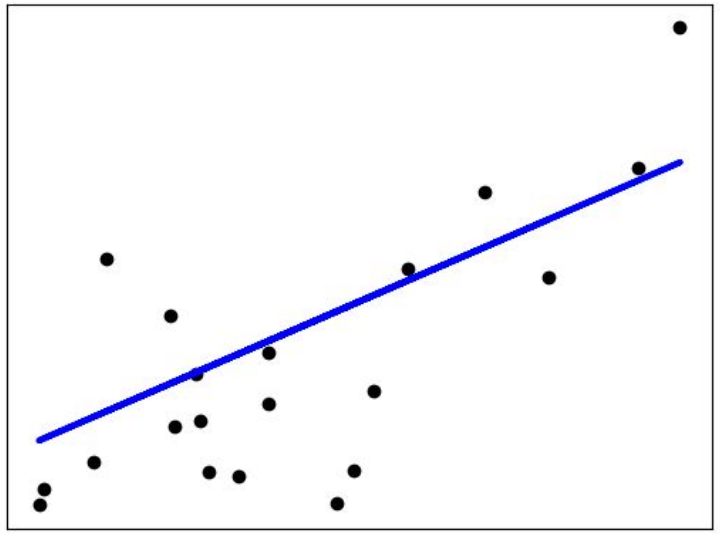
Linear Regression

- For **regression** problem.
- The label is expected to be a linear combination of the inputs.

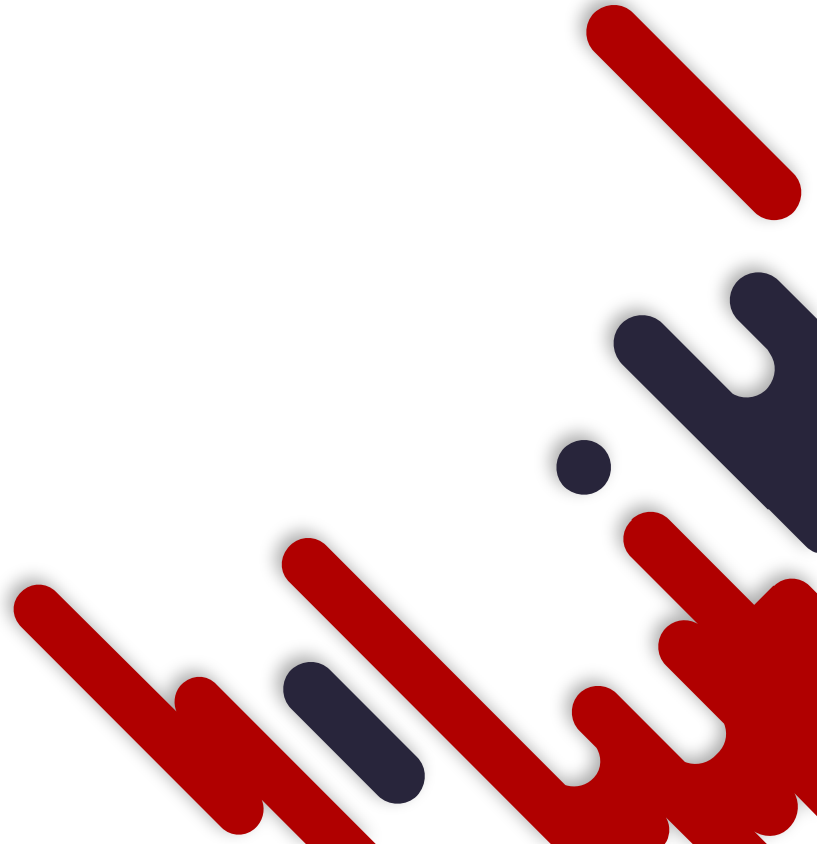
$$\hat{y}(w, x) = w_0 + w_1 x_1 + \ldots + w_p x_p$$

- Simply use **least square** method from linear algebra.

$$\min_w ||Xw - y||_2^2$$



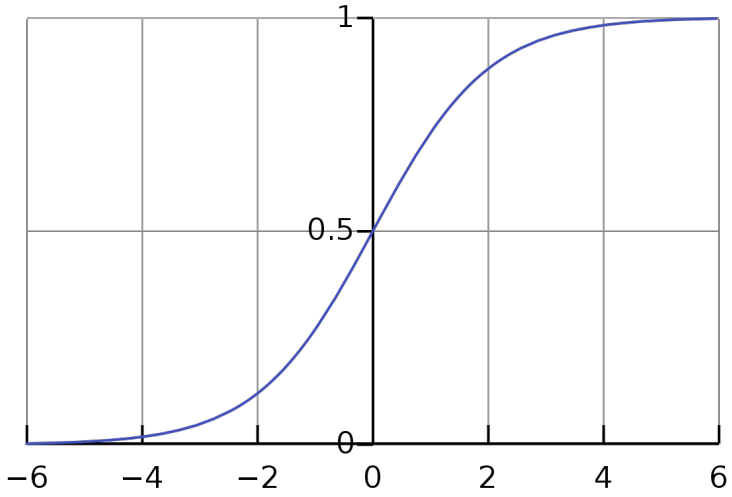
Linear Regression Hands-On



Logistic Regression

- For **classification** problem despite its name.
- Used to estimate probability that a data belongs to a particular label. (what is the probability that this email is spam?)
- Formula $p = \sigma(\mathbf{x}^T W + \mathbf{b})$

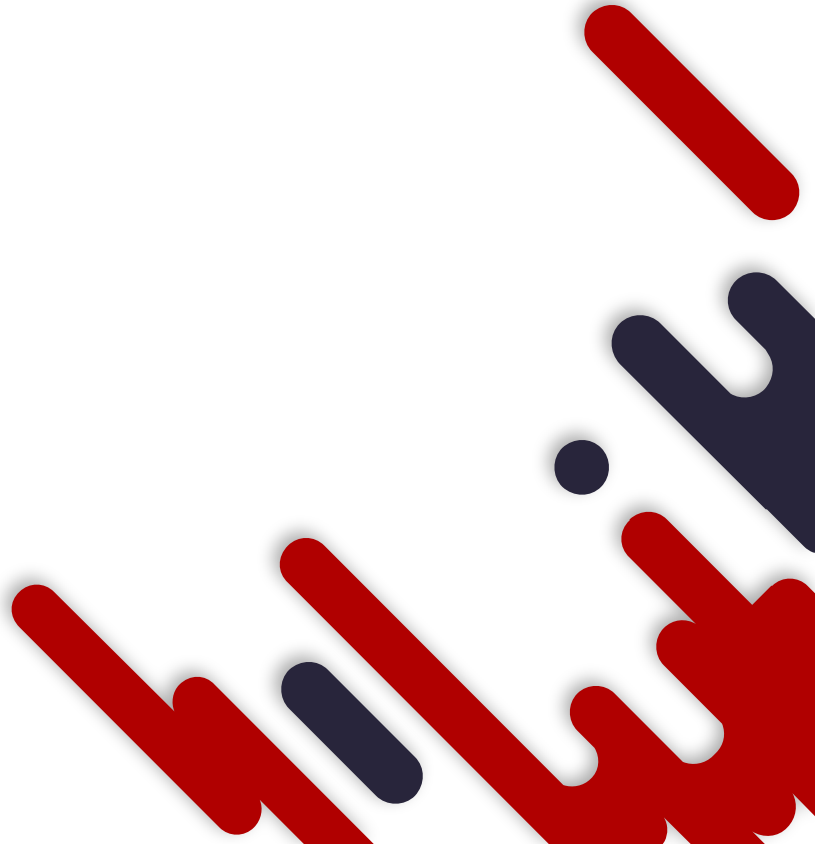
$$y = \begin{cases} 0, \text{ if } p < 0.5 \\ 1, \text{ if } p \geq 0.5 \end{cases}$$



- Optimize the cost function to get W and b.

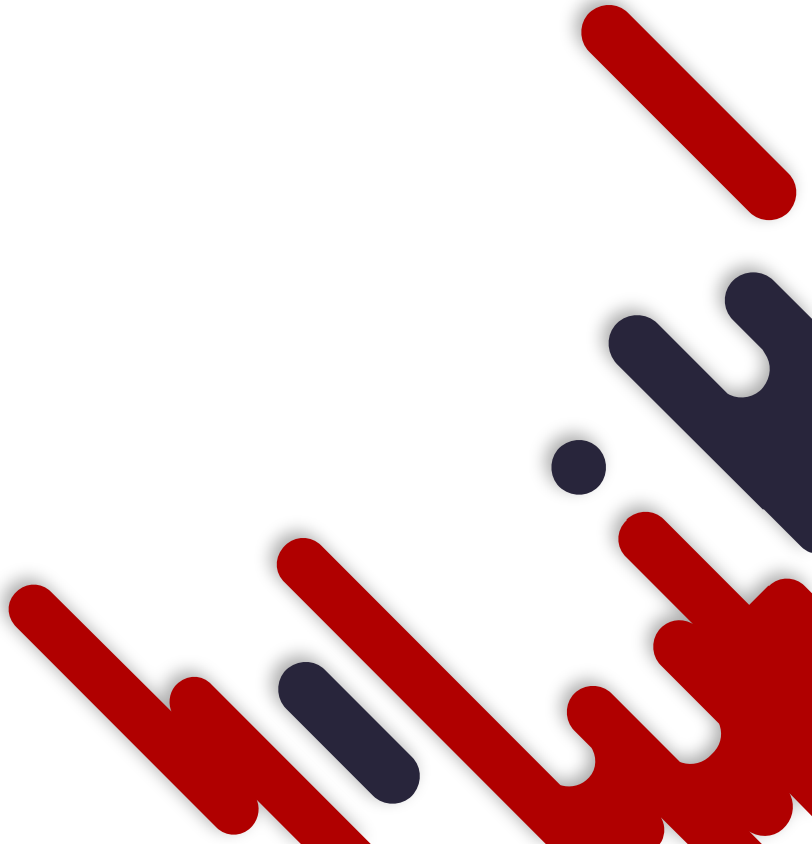
$$c(W, b) = \begin{cases} -\log(p), \text{ if } y = 1 \\ -\log(1 - p), \text{ if } y = 0 \end{cases}$$

Logistic Regression Hands-On



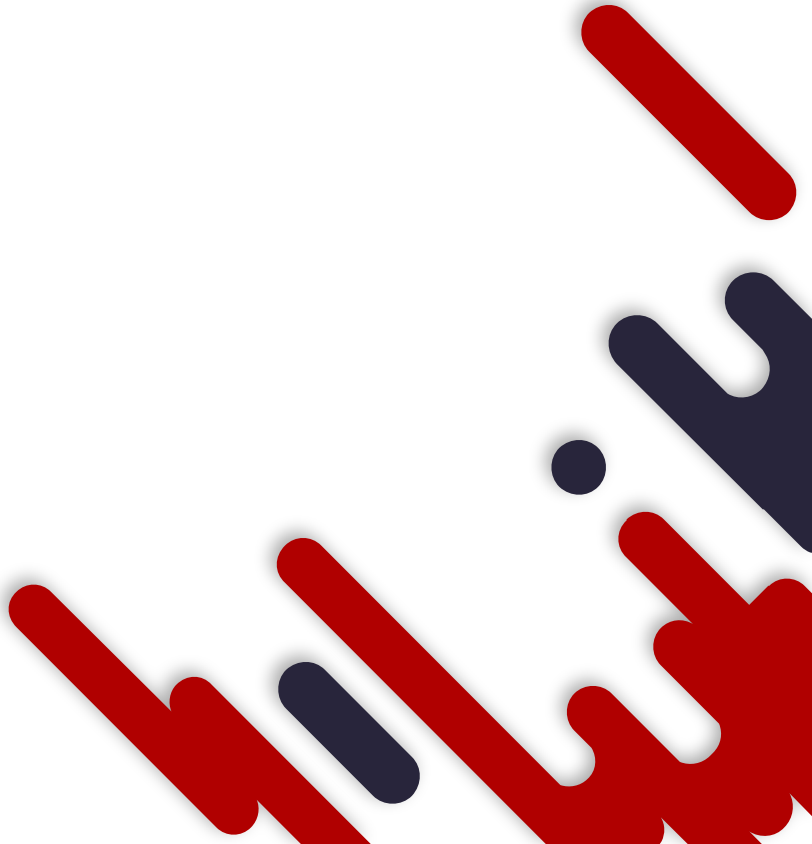
Review Pipeline

- Data collection
- Data preparation
- Algorithm Selection and Setup
  - Hyperparameter choosing
- Training
- Evaluation



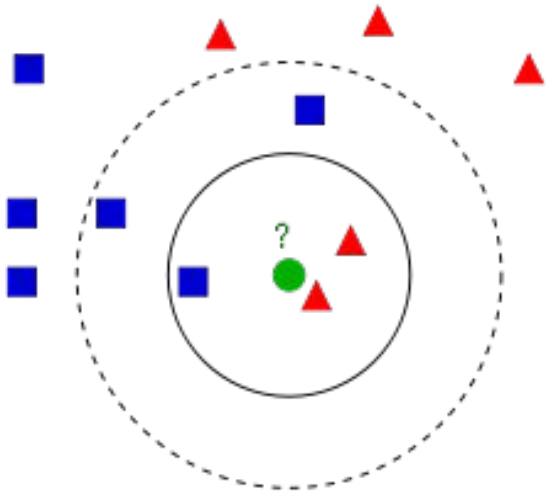
Supervised Learning Algorithms

- Linear regression and logistic regression.
- **K-Nearest Neighbours.**
- Decision Tree and Random Forests.
- Support Vector Machines
- Neural Networks.

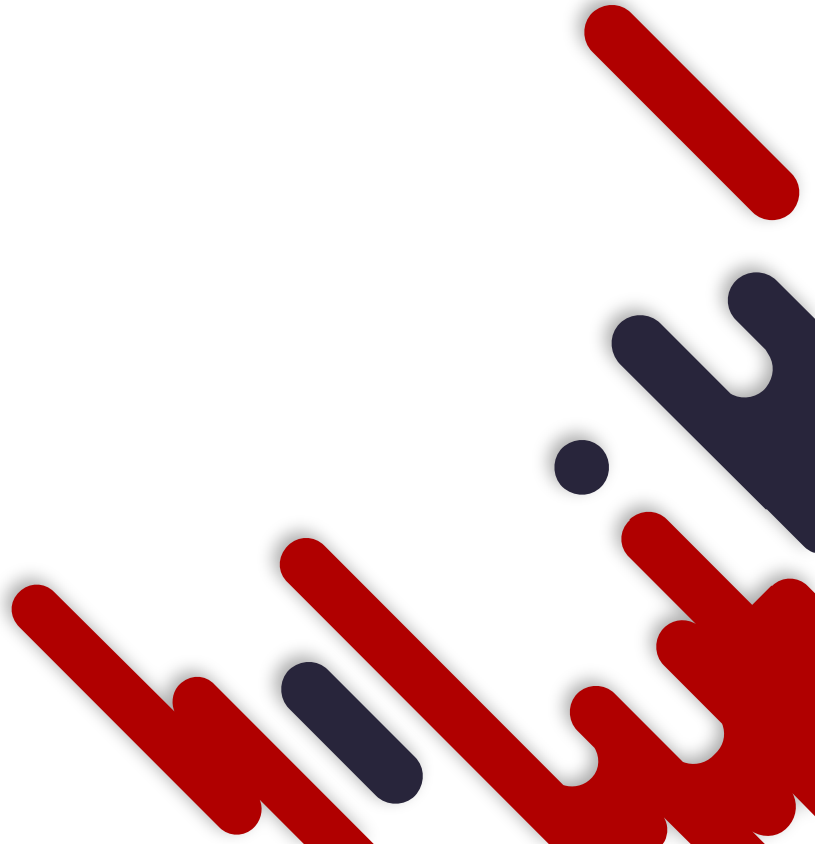


K-Nearest Neighbour (KNN)

- For **classification** or **regression** problem. Classification (regression) is computed from a majority vote (mean value) of its k nearest neighbors.
- Predict the green circle.
- Hyperparameters:
  - k
  - Uniform or distance weight.
  - Different distance metrics.



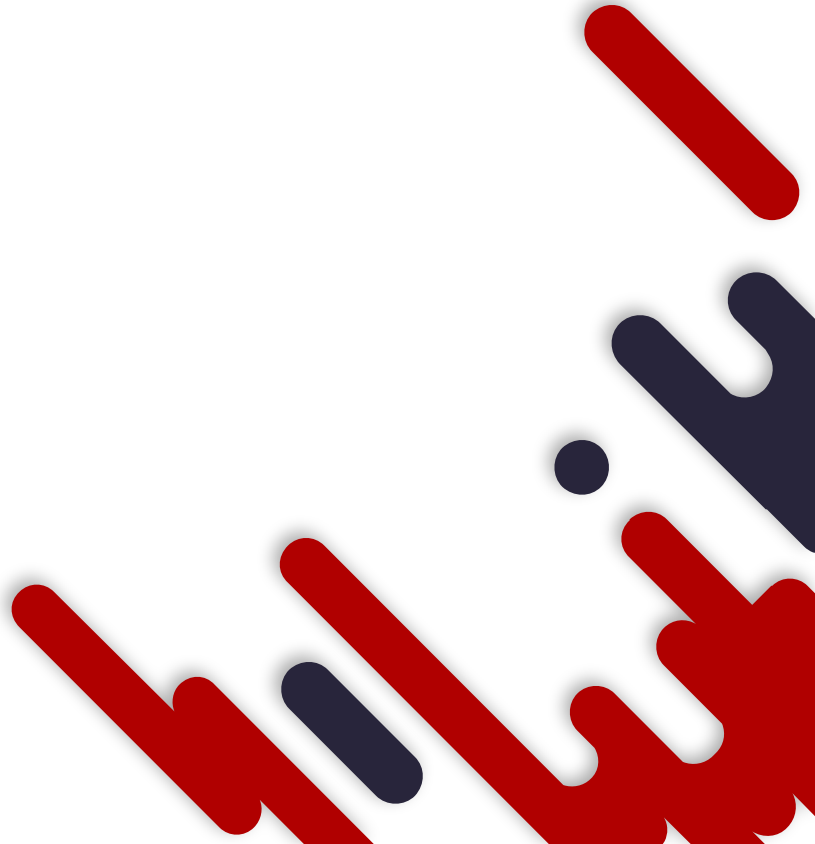
K-Nearest Neighbour (KNN) Hands-On



Hyperparameter Choosing

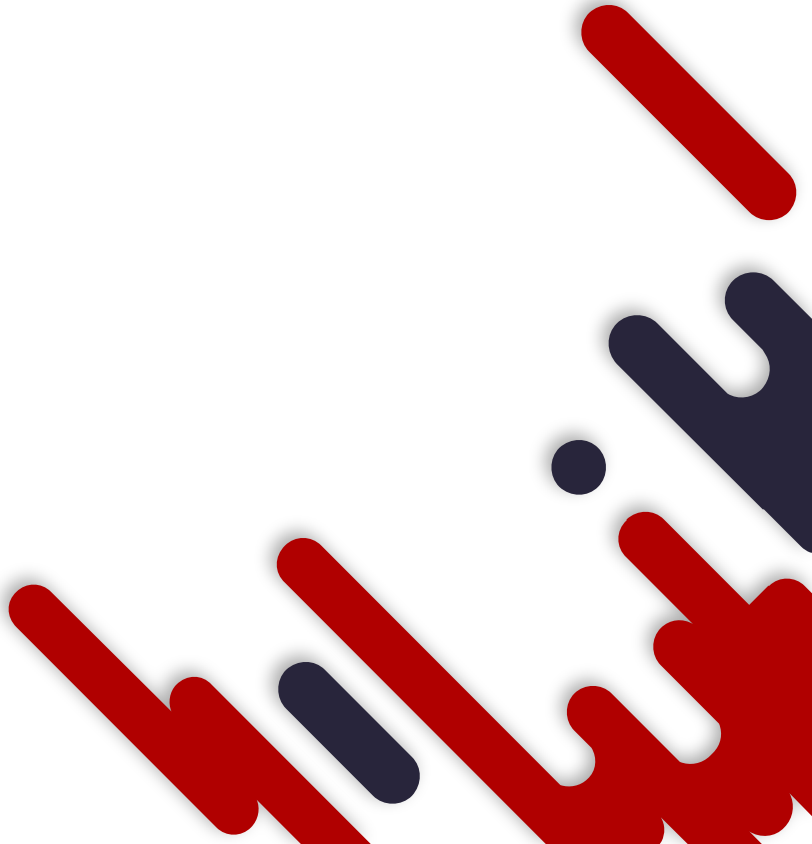
- **Grid search**
  - One option would be to fiddle with the hyperparameters manually, until you find a great combination of hyperparameter values. Very tedious.
  - Used if the search space is small.
  - Use GridSearchCV from Scikit learn.
- **Random search**
  - Try random combinations instead of trying every possibilities.
  - Used if the search space is large.
  - Use RandomizedSearchCV from Scikit learn.

Hyperparameter Hands-On



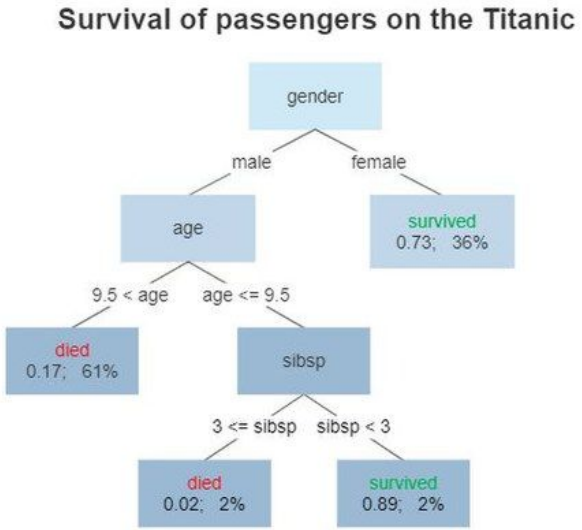
Supervised Learning Algorithms

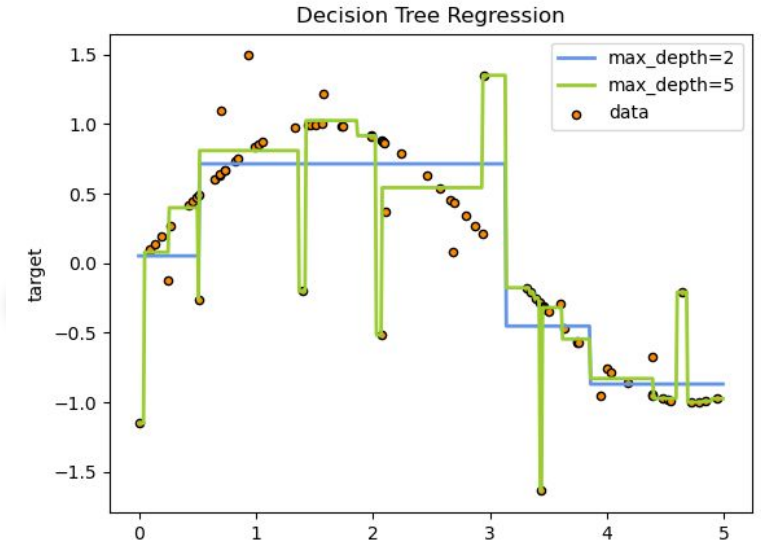
- Linear regression and logistic regression.
- K-Nearest Neighbours.
- **Decision Tree and Random Forests.**
- Support Vector Machines
- Neural Networks.



Decision Tree

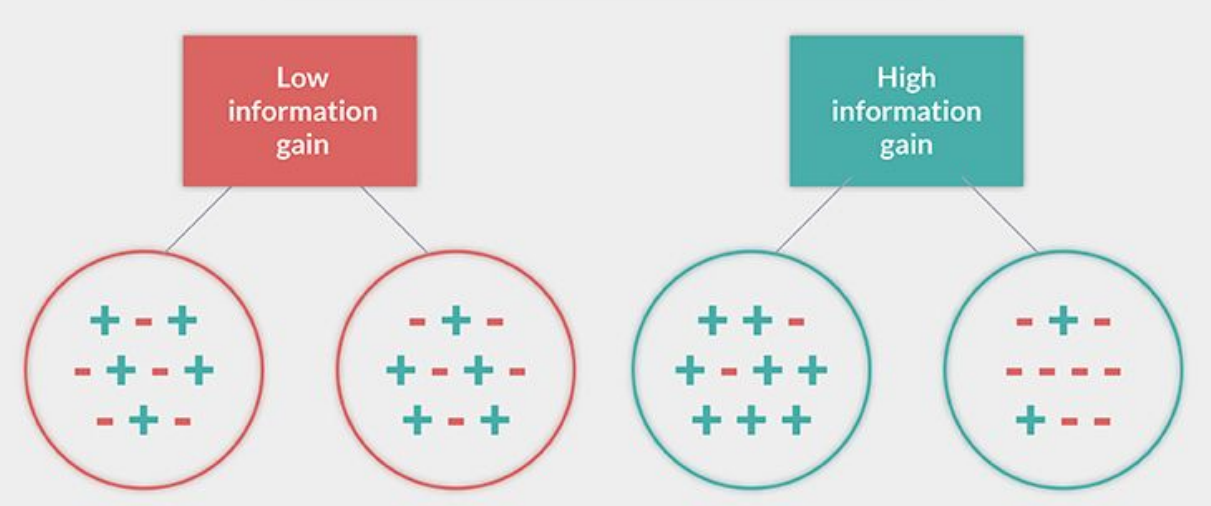
- The goal is to create a model that predicts the label by **learning simple decision rules (if-else)** inferred from the data.
- **Advantages**:
  - Easy to interpret (could be visualised).
  - Less preprocessing needed (does not need data scaling, can handle categorical and numerical easily).
- **Disadvantages**:
  - Prone to overfitting.
  - Unstable (small variation in input, completely different tree)



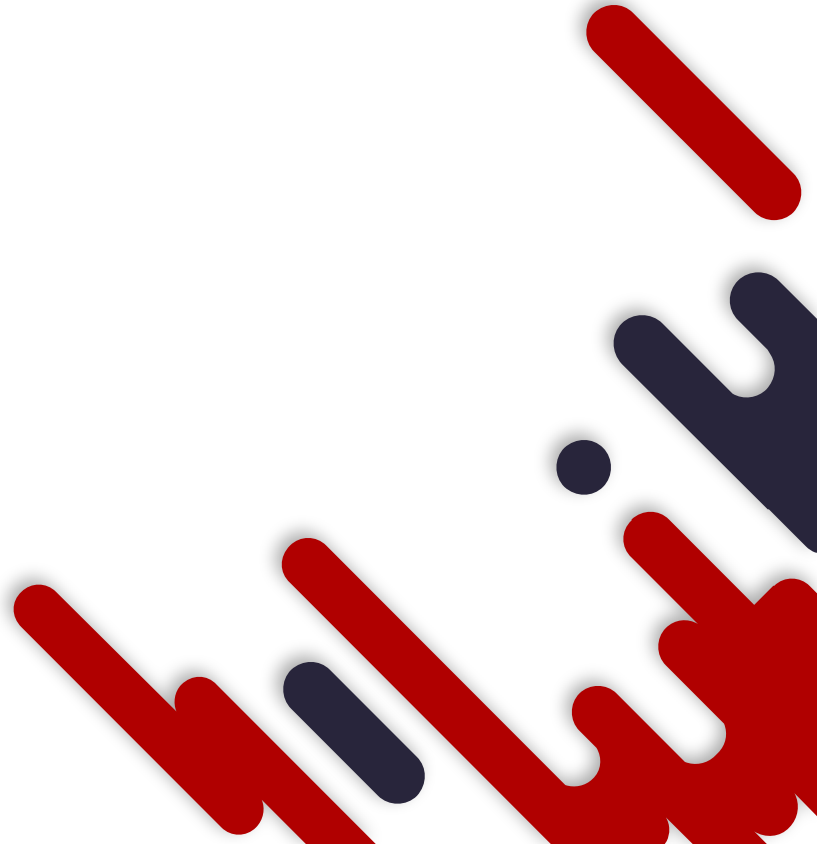


Decision Tree

- Several algorithms to construct decision tree (ID3, C4.5, CART, …)
- Basically, the algorithm works in a **top-down** manner by choosing the best variable **to split.**
- Several **metrics** to split based on information theory (gini index, information gain)
- Keep splitting until target height is achieved or minimum number of samples required to split achieved



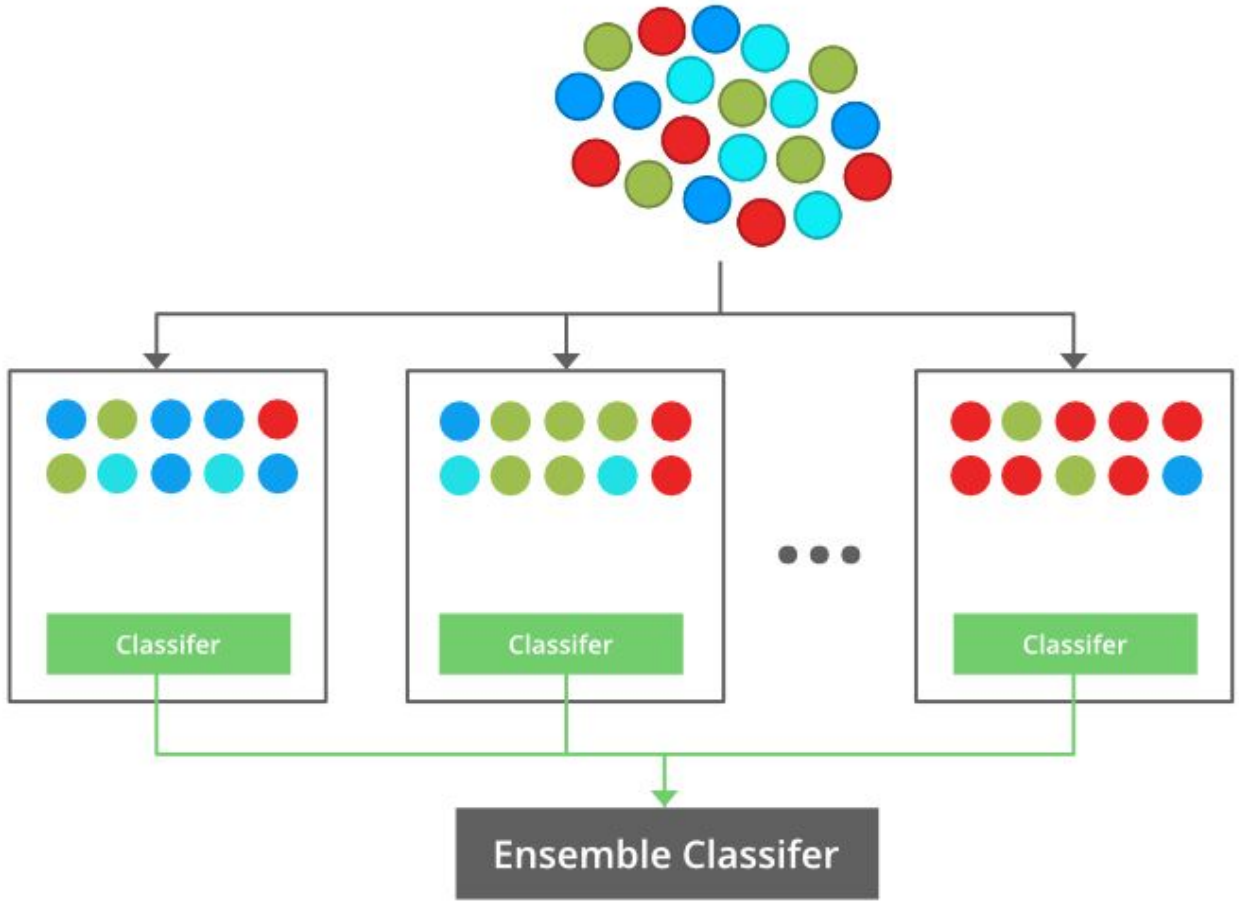
Decision Tree Hands-On



Random Forests

- **Ensemble:** Try to **combine the models that perform best.** The group (or "ensemble") will often perform better than the best individual model.
- **Random forests**: create ensemble of decision trees from a sample drawn with replacement from the training set (**bagging**).
- Sometimes, one tree can only split based on a random set of features.
- **Final prediction**: average prediction or majority vote.
- The injected randomness in forests yield decision trees with somewhat decoupled prediction errors. By taking an average of those predictions, **some errors can cancel out.**

Bagging



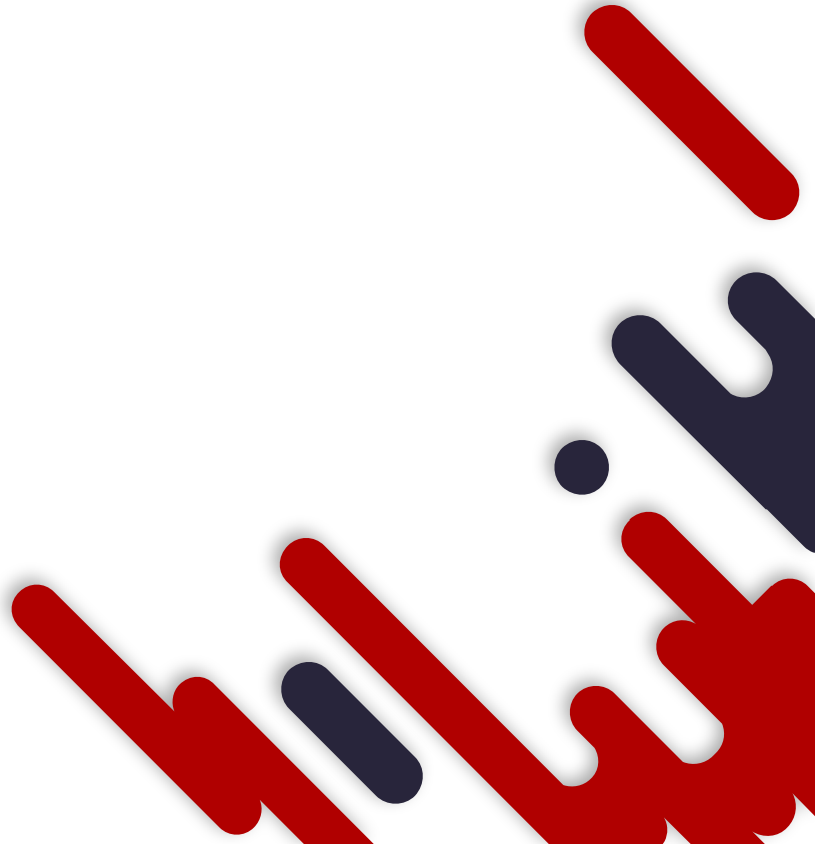
Original Data

Bootstrapping

Aggregating

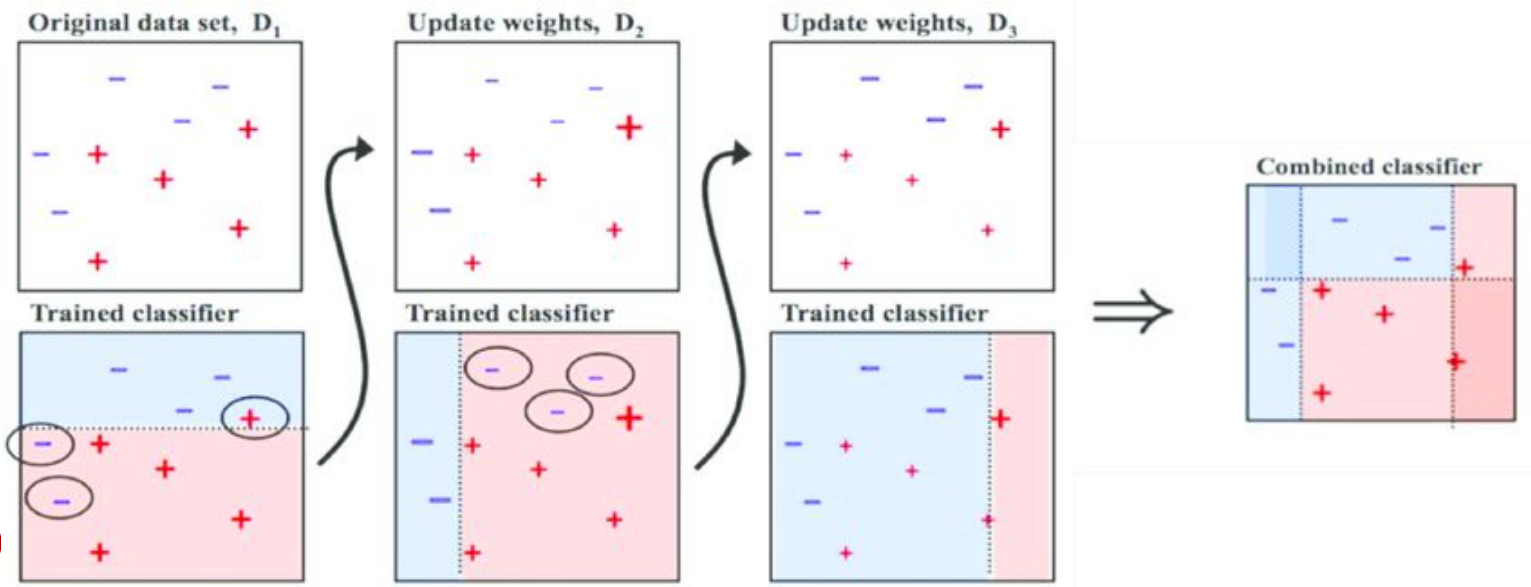
Bagging

Random Forests Hands-On

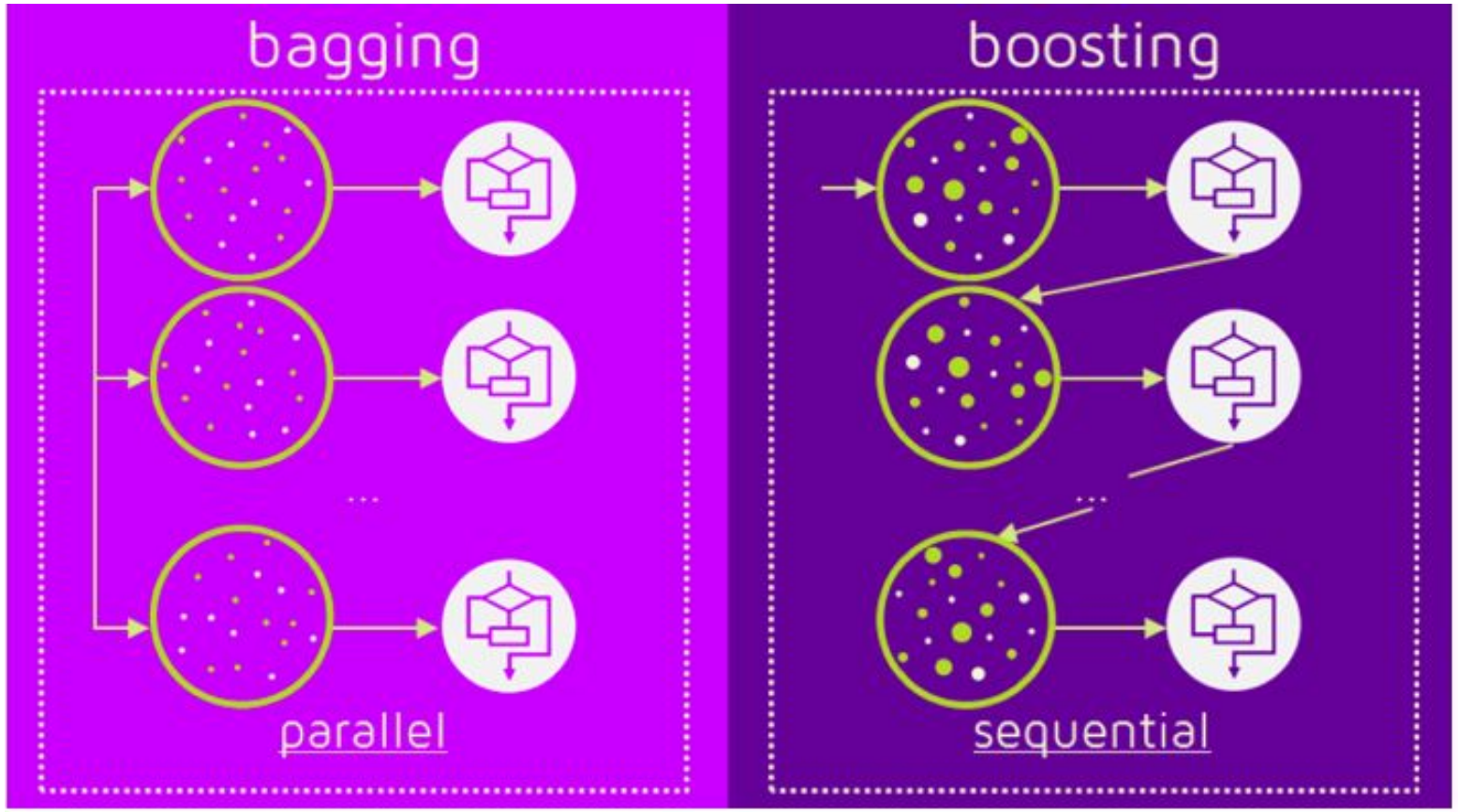


AdaBoost

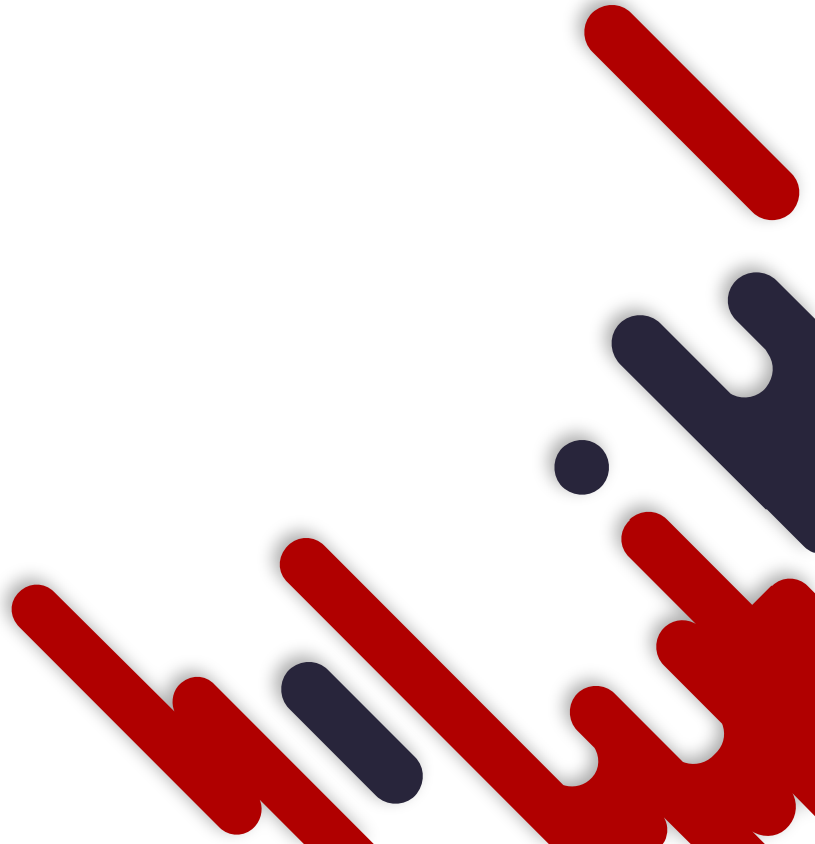
- Adaptive **Boosting**, another type of ensemble.
- The core principle of AdaBoost is to fit a sequence of weak learners on **repeatedly modified versions of the data**.
- **Final prediction**: average prediction or majority vote from all of them.
- Variant: Gradient Boosting (XGBoost) are generally better.



Bagging vs Boosting

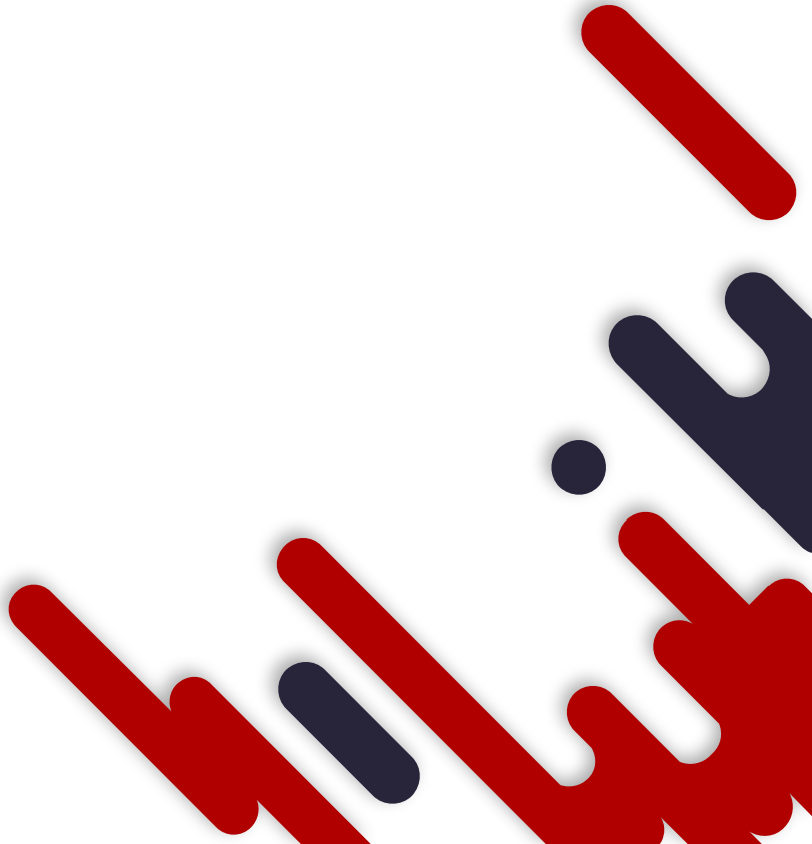


AdaBoost Hands-On



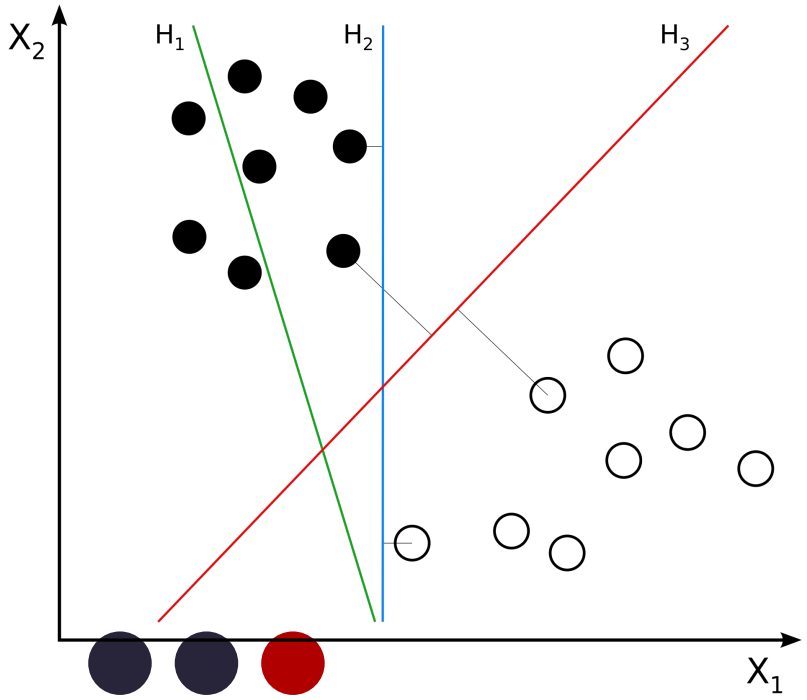
Supervised Learning Algorithms

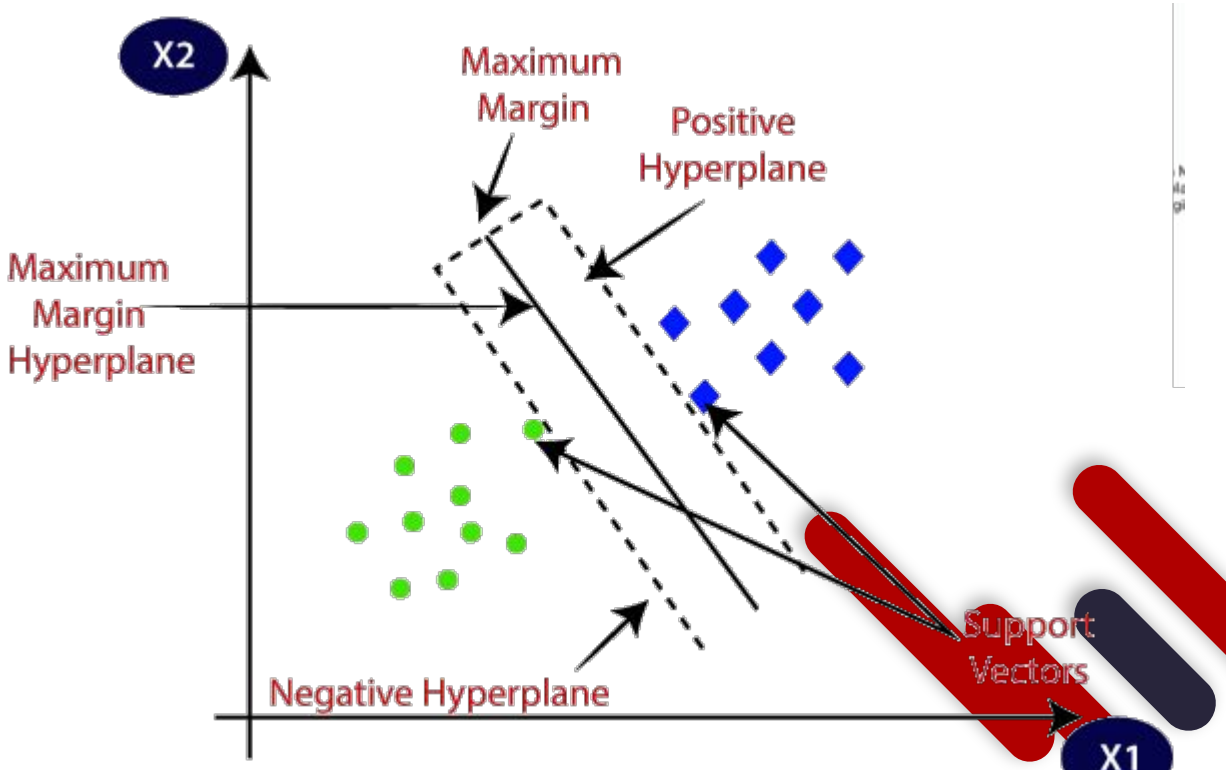
- Linear regression and logistic regression.
- K-Nearest Neighbours.
- Decision Tree and Random Forests.
- **Support Vector Machines**
- Neural Networks.



Support Vector Machine (SVM)

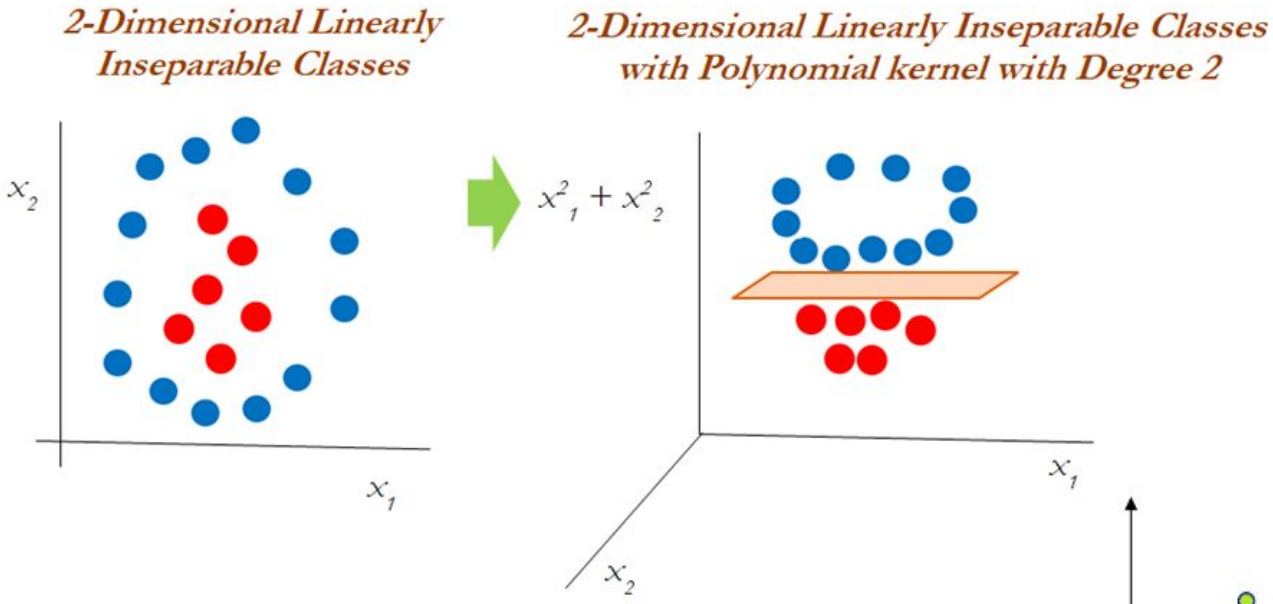
- SVM maps training examples to points in space so as to **maximise the width** of the gap between the two categories.
- New examples are then mapped into that same space and predicted to belong to a category based on which side of the gap they fall.

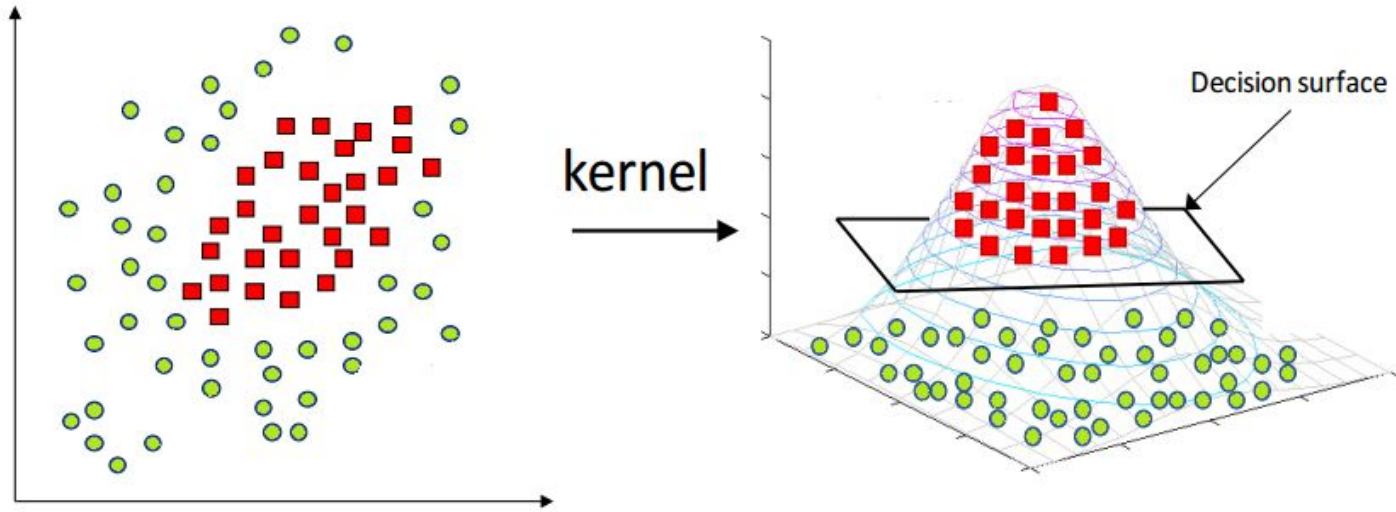




Support Vector Machine (SVM)

- Non-linear with **kernel trick**.

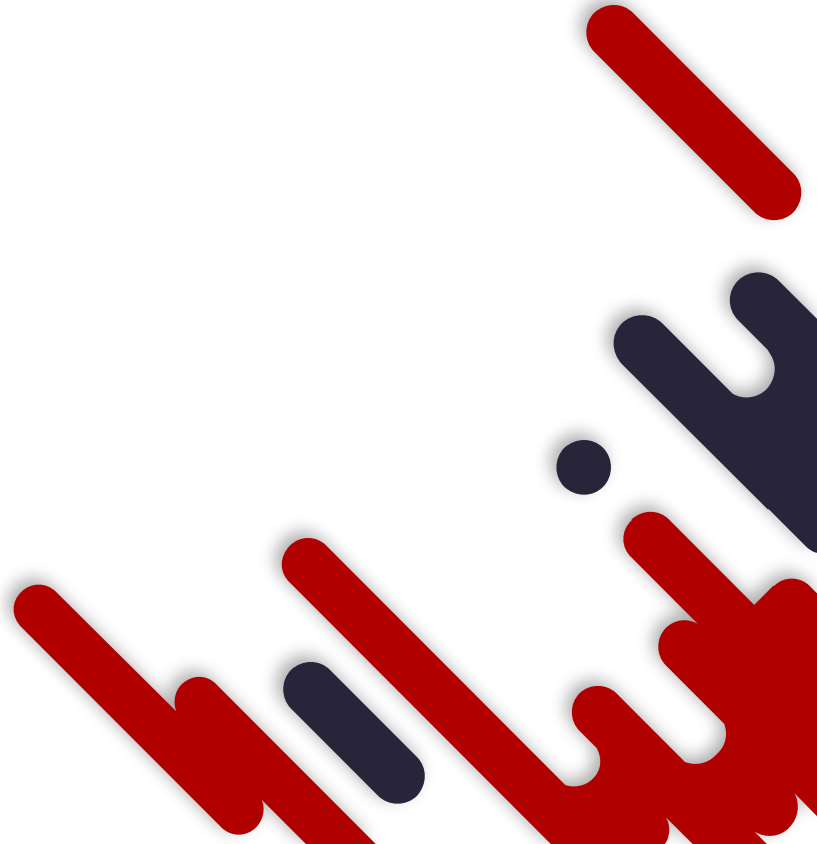




Support Vector Machine (SVM)

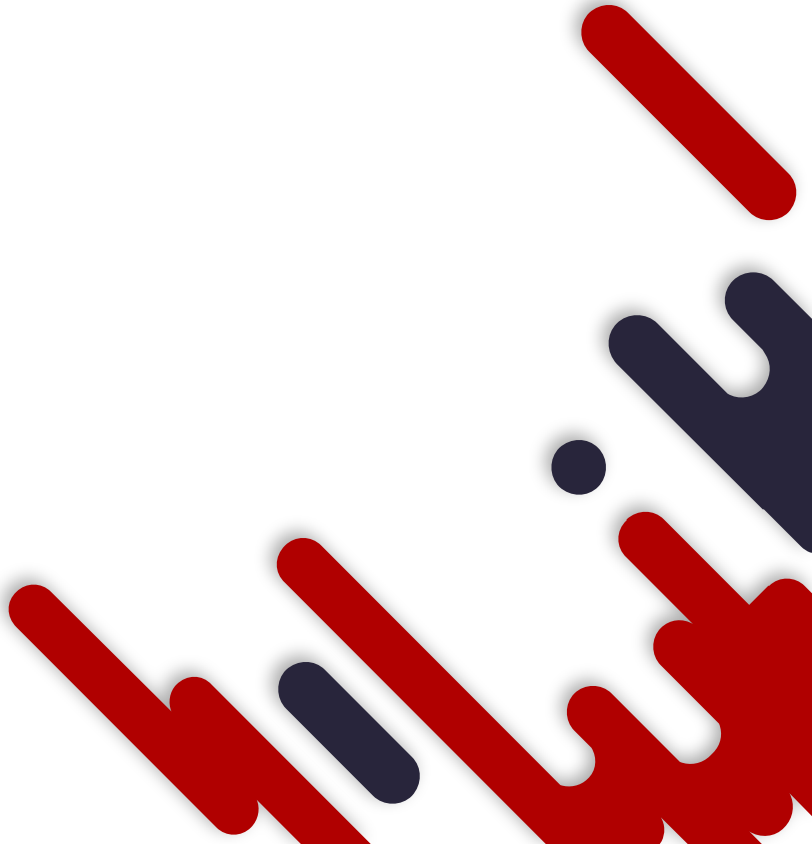
- **Advantages:**
  - Effective in high dimensional spaces.
  - Still effective in cases where number of dimensions is greater than the number of samples
- **Disadvantages**:
  - If the number of features is much greater than the number of samples, avoid overfitting by hyperparameter choosing.
  - SVMs do not directly provide probability estimates.
  - Quite time-consuming.

Support Vector Machine Hands-On



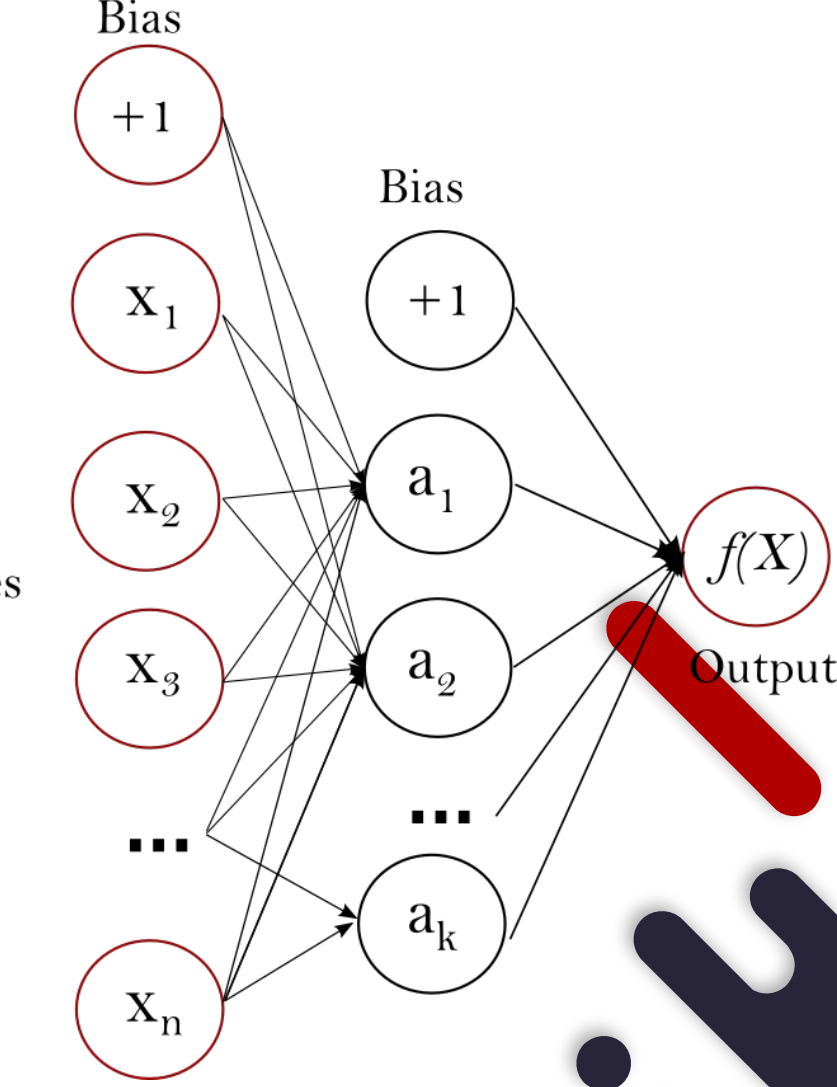
Supervised Learning Algorithms

- Linear regression and logistic regression.
- K-Nearest Neighbours.
- Decision Tree and Random Forests.
- Support Vector Machines
- **Neural Networks.**



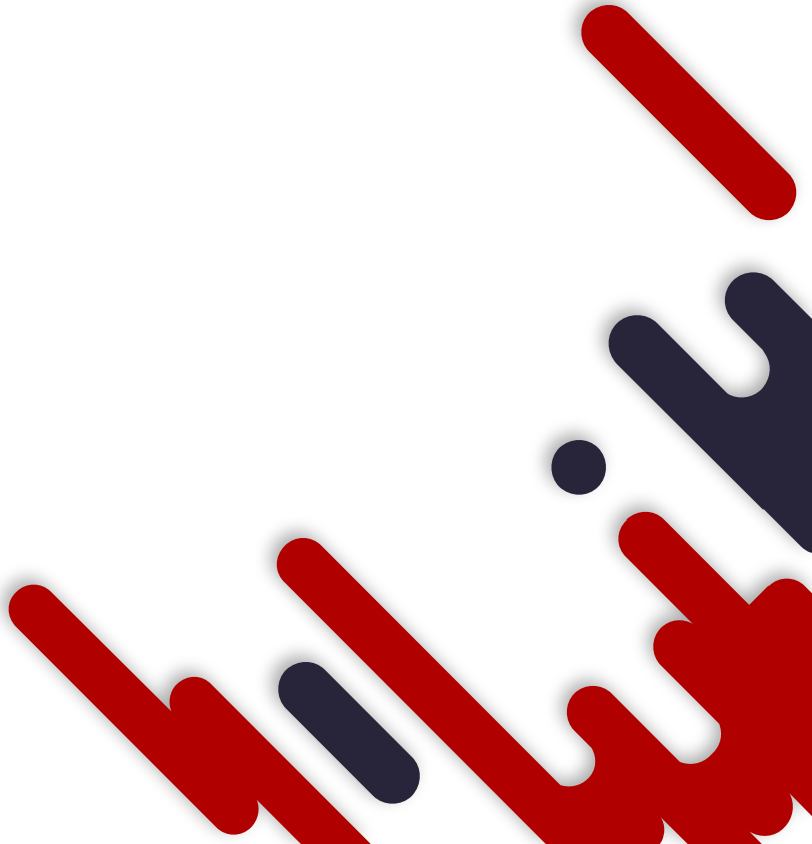
Neural Networks (NN)

- Multilayer perceptron consists of input, hidden, and output layer.
- Each layer consists of several nodes. Each nodes from one layer are connected to the next layer with a weight. There is also a bias vector.
- output of a node: $o = f(W\mathbf{x} + \mathbf{b})$
- f is a non-linear activation function (usually non sigmoid, Relu, tanh, etc.)
- No activation function at the output layer for regression, sigmoid for classification (logistic regression)

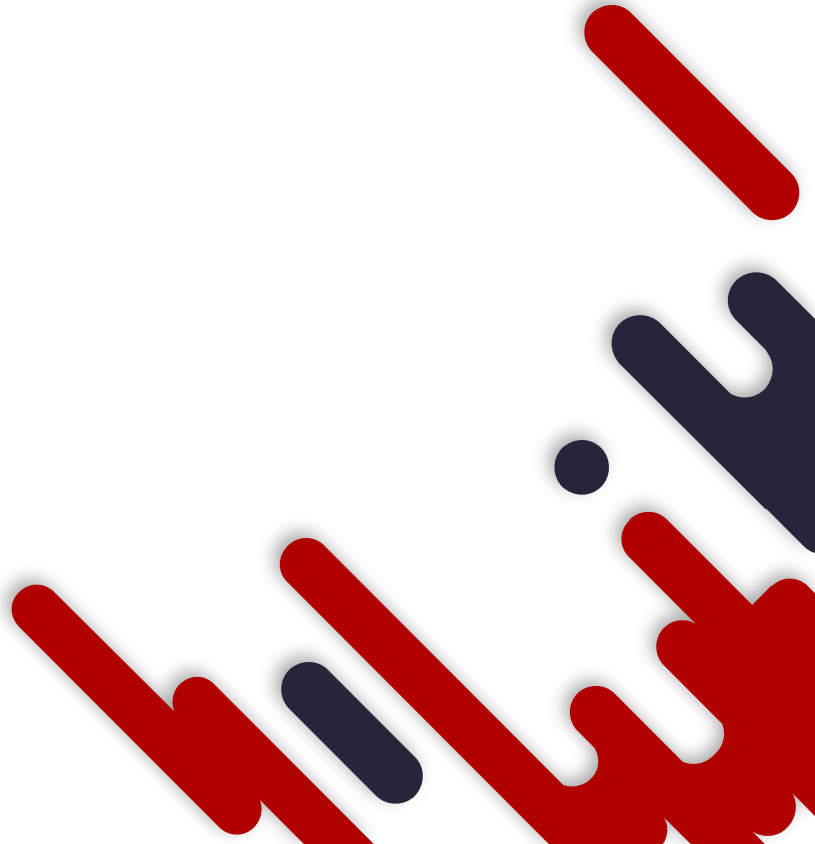


Neural Networks (NN)

- The weights and bias are optimised with gradient descent algorithm based on a defined performance metrics (MAE, MSE, accuracy, …)
- **Advantages**:
  - State-of-the art method especially with deep learning.
  - Easily parallelised with GPU.
- **Disadvantages**:
  - Cannot be interpreted easily (black box).
  - Need a lot of data.

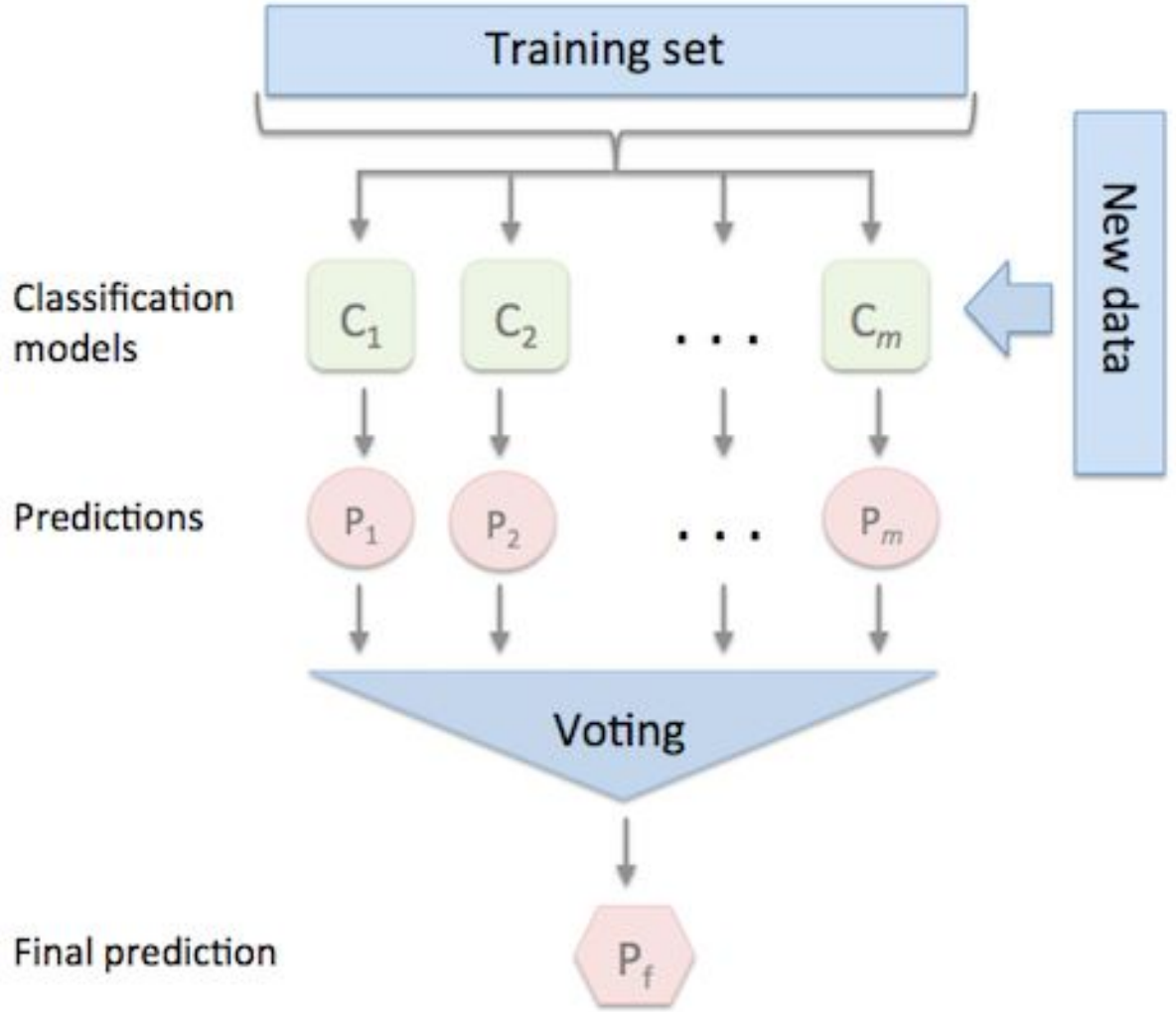


Neural Networks Hands-On



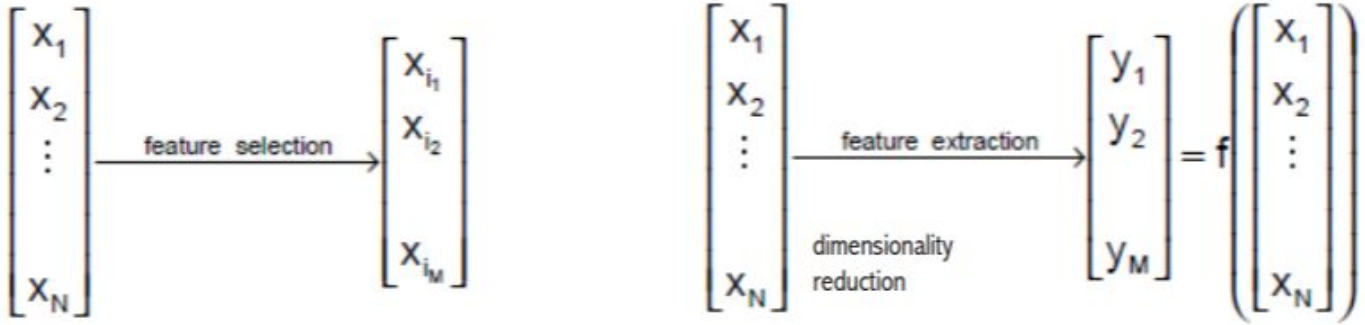
Additional topics: Ensemble

- Hands on: VotingRegressor and VotingClassifier



Additional topics: Feature Selection

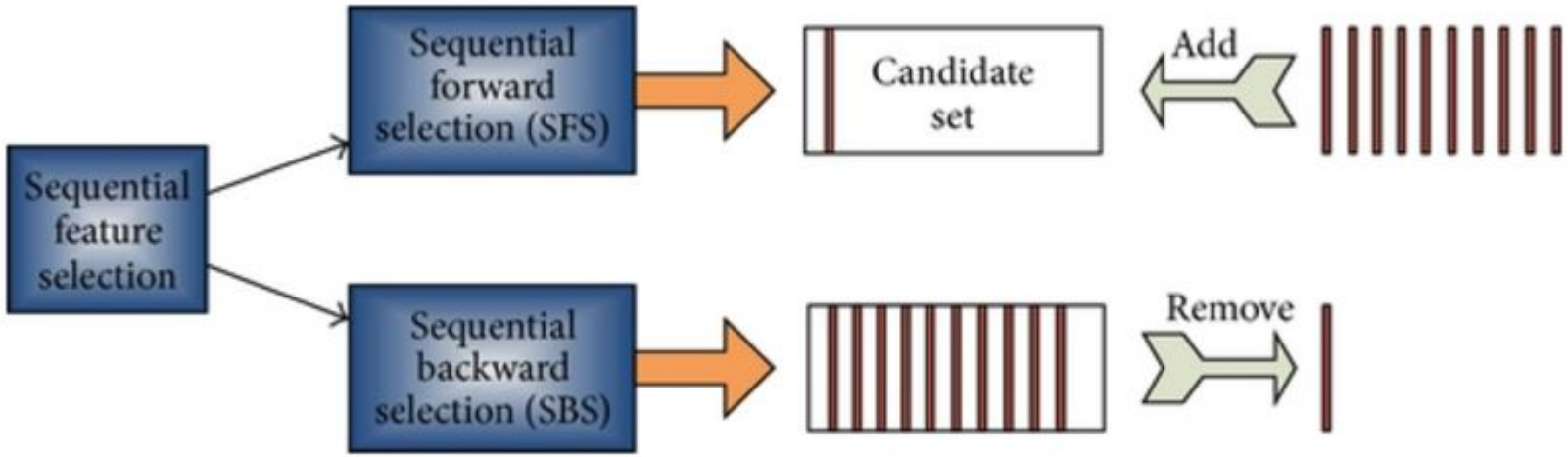
- Given **n features,** select **d < n** features to increase accuracy or minimise error.



- Two methods:
  - **Filter:** Independent of the machine learning algorithm (based on statistics, information theory, etc.)
  - **Wrapper:** Evaluate feature subset based on the machine learning algorithm

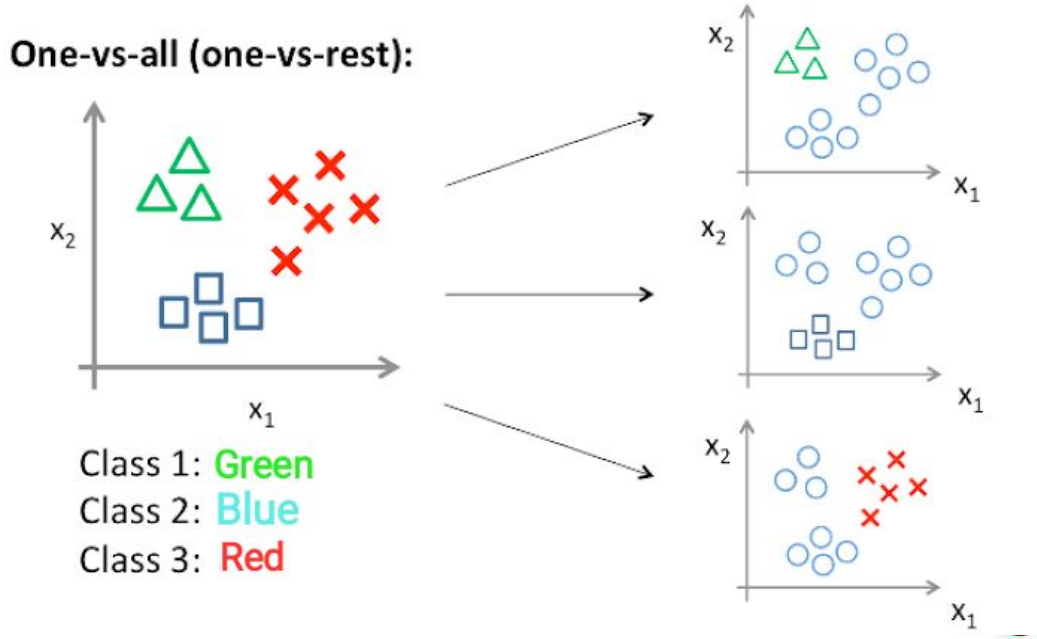
Additional topics: Feature Selection

- **Wrapper methods:**
  - Random search
  - Naive search (one-by-one, select top-K)
  - Sequential forward selection (bottom up)
  - Sequential backward selection (top down)



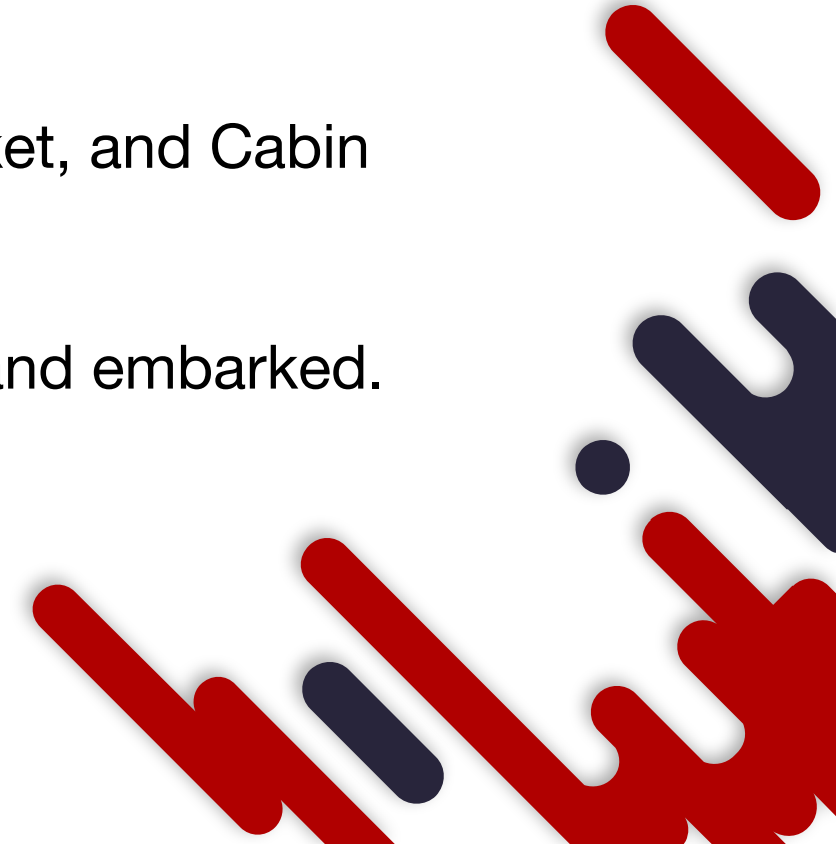
Some topics not covered

- Multi-class classification
  - one-vs-all for SVM and logistic regression
  - Neural network needs to one-hot encoding as its output
- Multi-label classification.
- Other classifiers/regressors: naive bayes, deep learning, ...
- ...



Task

- Compare different classification algorithms for Titanic dataset (see explanation here: https://www.kaggle.com/c/titanic/data). Predict the survival.
- Do data preparation:
  - Separate survival label from the data
  - Drop the following columns: PassengerId, Name, Ticket, and Cabin
  - Fill missing age with its median.
  - One-hot encode the following columns: pclass, sex, and embarked.
  - Scale the features with standard scaler



Task

- Compare different classification algorithms:
  - Logistic Regression.
  - KNN.
  - Tree Classifier.
  - Random Forest Classifier.
  - SVM Classifier.
  - Neural Network Classifier.

Classifier	Accuracy	F1-Score
Logistic regression		
KNN		
Tree		
Random forest		
SVM		
Neural Network		

- Report its accuracy and F1-Score with a table with **5-fold cross validation**.
- Get the best classifier and do a hyperparameter tuning with grid search.