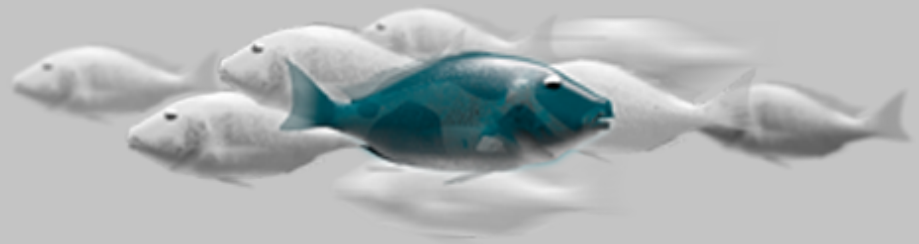
EDEM



Master Data Analytics | 3a Edición

Artificial Intelligence Intro IV

Jose Peris Adsuara

Jose Peris Adsuara

Head of Ai @Bit2Me
Data analytics @Edem

bit 2 me

EDEM
Escuela de Empresarios

- **CTO // Digital Product Manager @aiwannapay.com**
  aiwanna
  Sep 2019 – Present · 2 yrs 4 mos

  The most complete SaaS solution for HORECA channel. Scan, order and pay from your smartphone.
  Artificial Intelligence for Small and Medium enterprises. Developing AI algorithms and display augmented analytics. Reduce your costs, increase sales and automate your repetitive tasks

- **Senior Consultant**
  straiqr.ai GmbH · Part-time
  Jun 2021 – Present · 7 mos

  Straiqr is a German company that is changing fashion rules through Metaverse, nft´s, digital fashion & AI
  We create digital fashion for the digital generation.We believe in a world where digital garment and clothing design is authentic and true to you. STRAIQR Intelligent Shopping is ready - now - in an easy to navigate store, where accurate delivery and sustainability is a consideration at every step of the way.

- **Teacher // Machine Learning and AI consulting and for MÁSTER EN DATA ANALYTICS & MASTER FINANZAS**
  EDEM Escuela de Empresarios
  Apr 2019 – Present · 2 yrs 9 mos
  Valencia y alrededores, España

  -Coordinating AI and ML contents , team building
  -Machine Learning and Deep Learning teacher:
  -Master Data Analytics
  -Master Finanzas
  -EMBA Executive
  -Webinar " IA aplicada a pymes"
  -Online

- **Data scientist // CEO & co-founder**
  tailor
  Jul 2017 – Sep 2019 · 2 yrs 3 mos
  Valencia, Spain

  Our mision is to improve the online shopping experience in fashion industry.
  Tailored by Big Data is an spanish Startup born in Valencia wich is reasearching in AI.
  Developing algorithms for fashion and trend industry .Our project has been developed under Garaje de Lanzadera incubator program & Órbita accelerator program
  https://lanzadera.es/
  http://www.programaorbita.com
  http://www.tailoredbybigdata.com
  Deep Learning - Machine Learning - Computer vision

2021:
-Winner Star Starups contest "Star days" hackathon

2019:
-Included in PlayBook platform, powered by Plug and Play Ventures

2018:
-Selected for Lanzadera start-up accelerator program
-Selected for Programa Órbita start-up accelerator program
-Selected as speaker in VI Mobile e-commece Madrid

// Index

(very) Important Concepts

ETL& Feature Engineering

AI CONCEPTS

Concepts: **missing values**

Column 0	age	years_seniority	income	parking_space	attending_party	entree	pets	emergency_contact
Tony	48	27		1	5	shrimp		Pepper
Donald	67	25	86	10	2	beef		Jane
Henry	69	21	95	6	1	chicken	62	Janet
Janet	62	21	110	3	1	beef		Henry
Nick		17		4				
Bruce	37	14	63		1	veggie		NA
Steve	83		77	7	1	chicken		n/a
Clint	27	9	118	9		shrimp	3	None

AI CONCEPTS

Concepts: how to handle **missing values?**

1 // Delete columns that are missing too many values to be useful.

2// Delete rows that are missing critical values.

3// Replace missing values with the mean (numerical)

4// Replace missing values with the median (categorical)

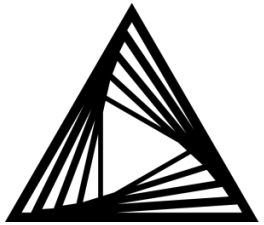
5// Replace missing values with an interpolated estimate.

6// Replace missing values with a constant

7// Understand the context and be creative

AI CONCEPTS

Concepts: how to handle **missing values?**



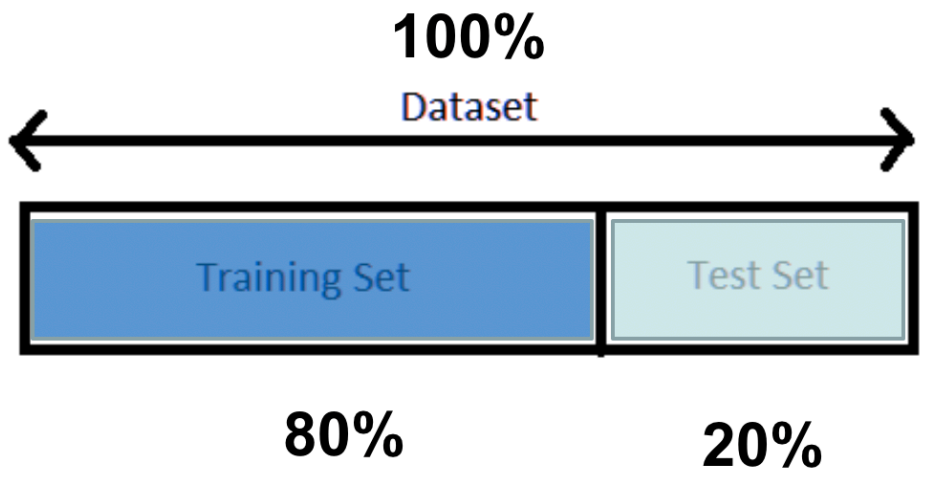
Titanic Missing Values

AI CONCEPTS

Concepts: **train & test**

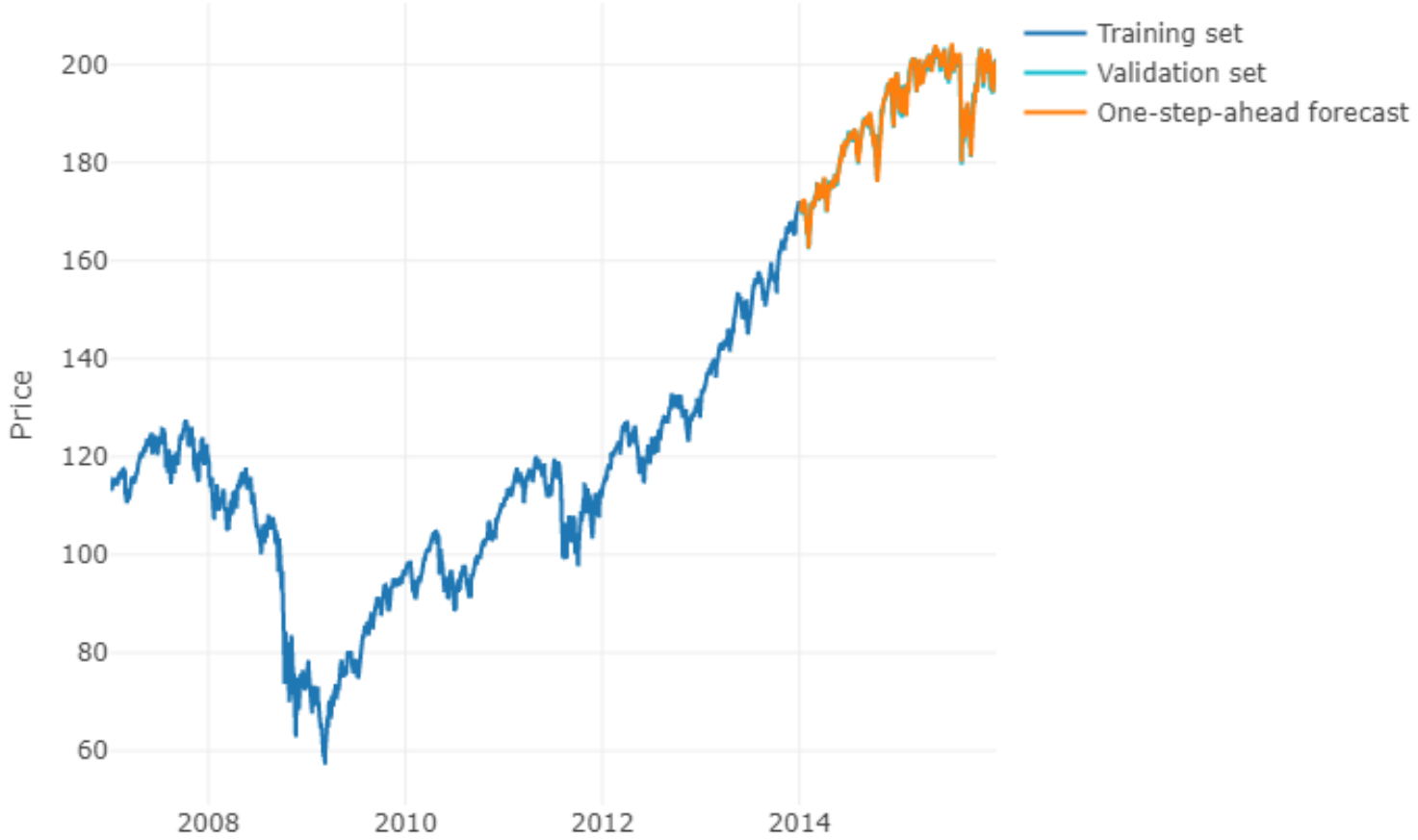
Concepts: train & test

In machine learning, a common task is the study and construction of **algorithms that can learn from and make predictions on data.**
Such algorithms work by making data-driven predictions or decisions, through building a mathematical model from input data.
The data used to build the final model usually comes from multiple datasets. In particular, three data sets are commonly used in different stages of the creation of the model.

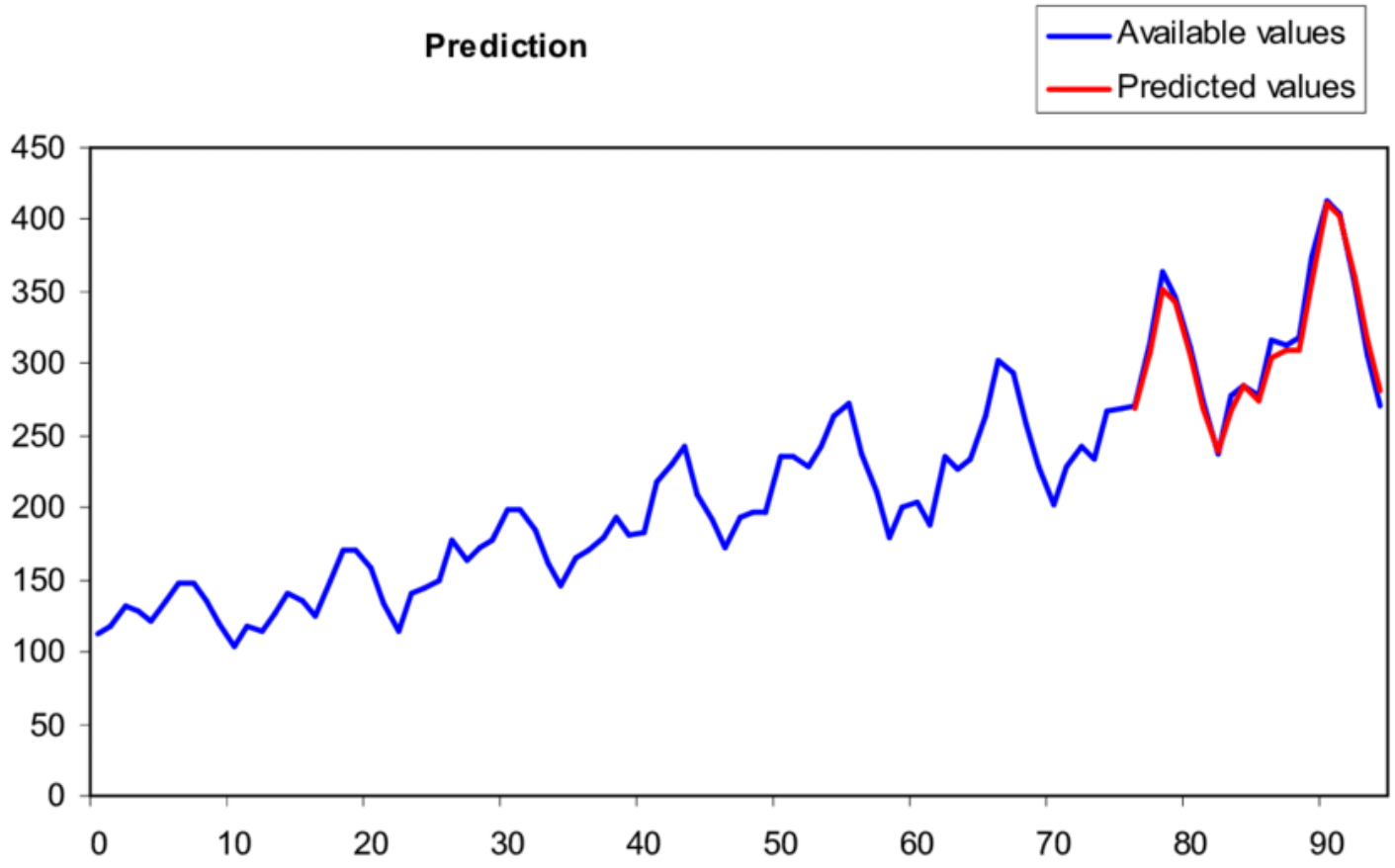


AI CONCEPTS

Concepts: train & test



Concepts: train & test



Cross validation

Cross validation

Cross-Validation is a very powerful tool. **It helps us better use our data**, and it gives us much more information about our algorithm performance.
In complex machine learning models, it's sometimes easy not pay enough attention and use the same data in different steps of the pipeline. This may lead to good but not real performance in most cases, or, introduce strange side effects in others. We have to pay attention that we're confident in our models.
Cross-Validation helps us **when we're dealing with non-trivial challenges** in our Data Science projects. There are to main purposes to use it:

1) **To avoid overfitting.**
2) **To Compare different machine learning methods**

Cross validation: missclassify

We want to use the variables (Chest Pain, Good Blood Circulation, etc.)…

Chest Pain	Good Blood Circ.	Blocked Arteries	Weight	Heart Disease
No	No	No	125	No
Yes	Yes	Yes	180	Yes
Yes	Yes	No	210	No
…	…	…	…	…

…to predict if someone has heart disease.

Chest Pain	Good Blood Circ.	Blocked Arteries	Weight	Heart Disease
No	No	No	125	No
Yes	Yes	Yes	180	Yes
Yes	Yes	No	210	No
…	…	…	…	…

Cross validation: missclassify

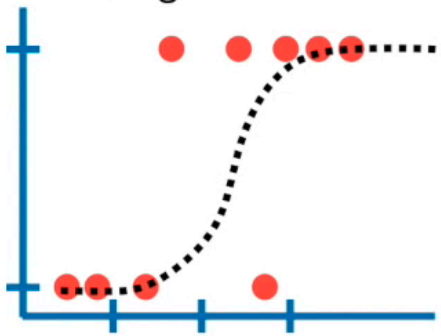
Chest Pain	Good Blood Circ.	Blocked Arteries	Weight	Heart Disease
No	No	No	125	No
Yes	Yes	Yes	180	Yes
Yes	Yes	No	210	No
...	...	...	...	...

…and predict if they have heart disease or not.

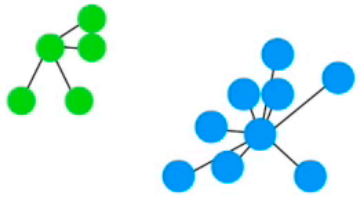
Chest Pain	Good Blood Circ.	Blocked Arteries	Weight	Heart Disease
Yes	No	No	168	???

Cross validation: missclassify

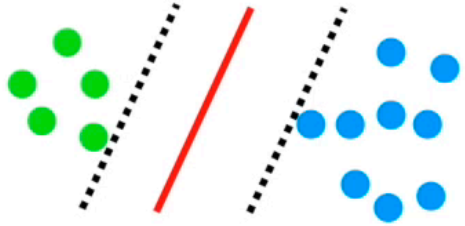
We could use Logistic Regression…



…or K-nearest neighbors…

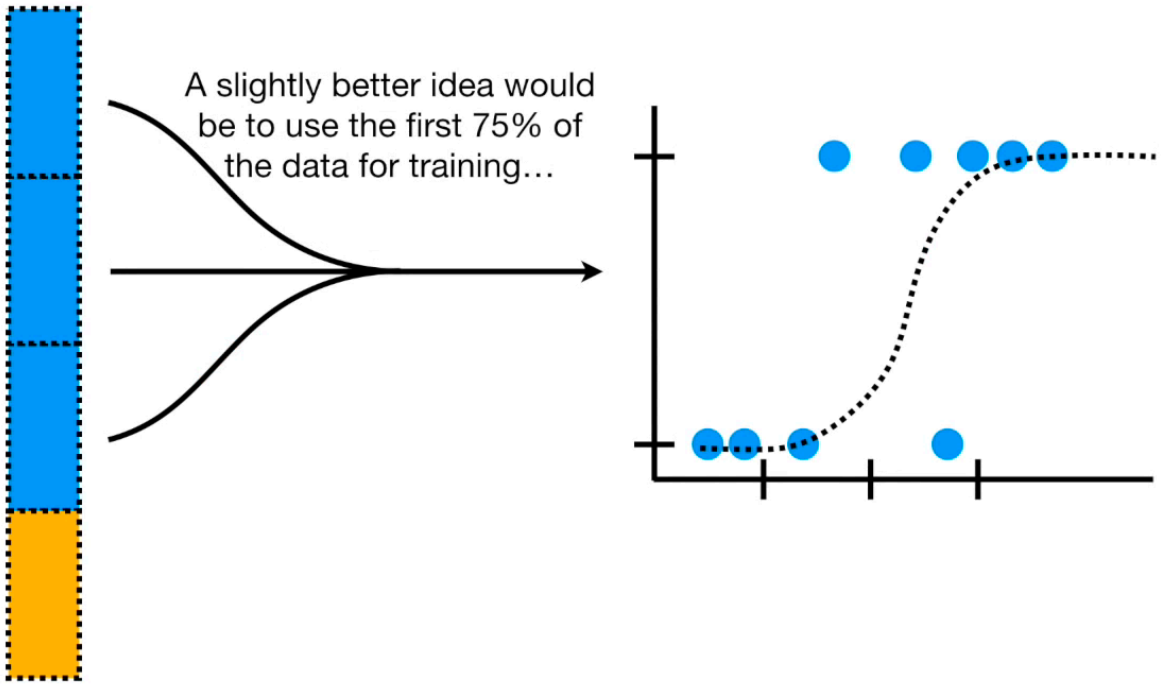


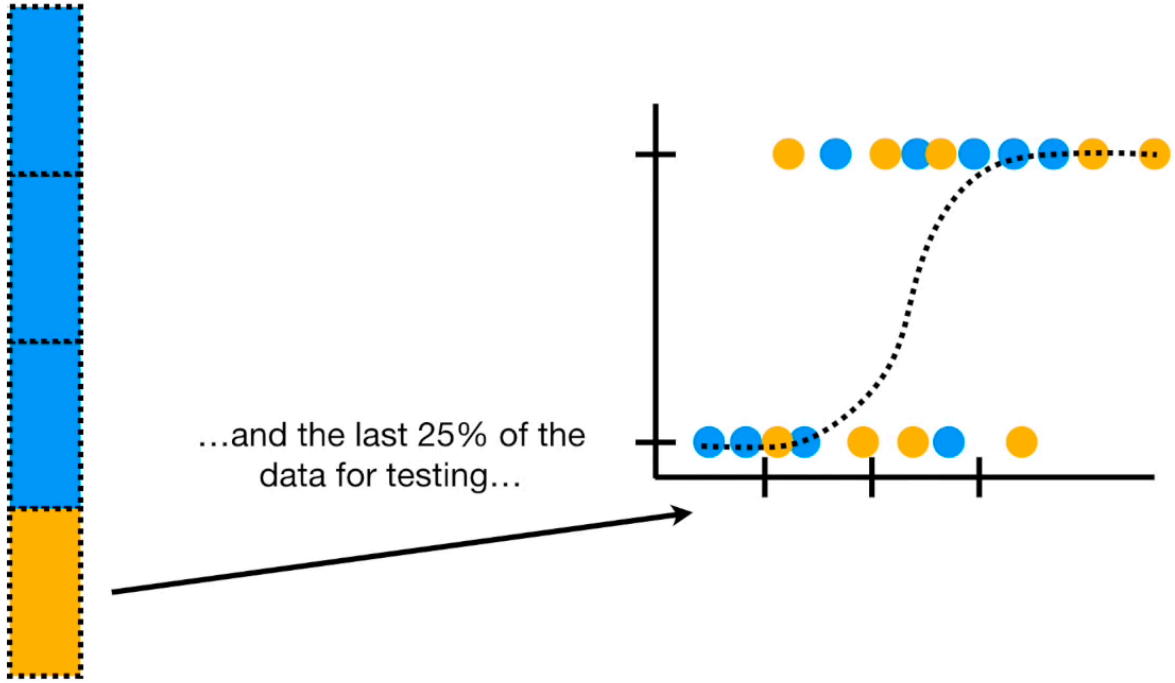
…or support vector machines (SVM)…



Cross validation allows us to compare different machine learning methods and get a sense of how well they will work in practice.

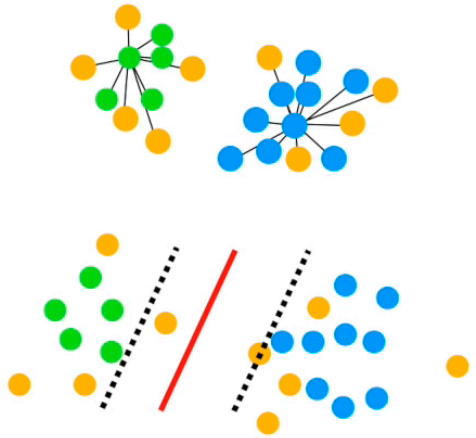
Cross validation: missclassify

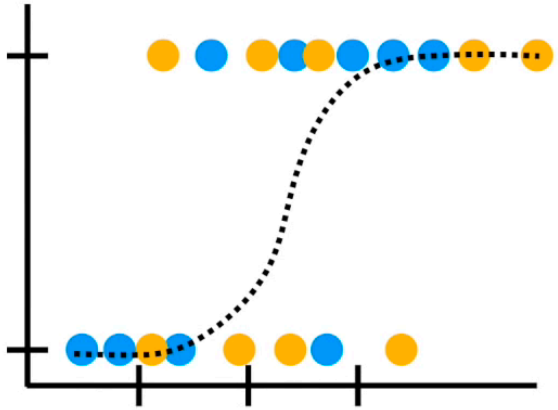


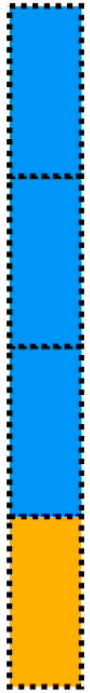


Cross validation: missclassify

We could then compare methods by seeing how well each one categorized the test data.

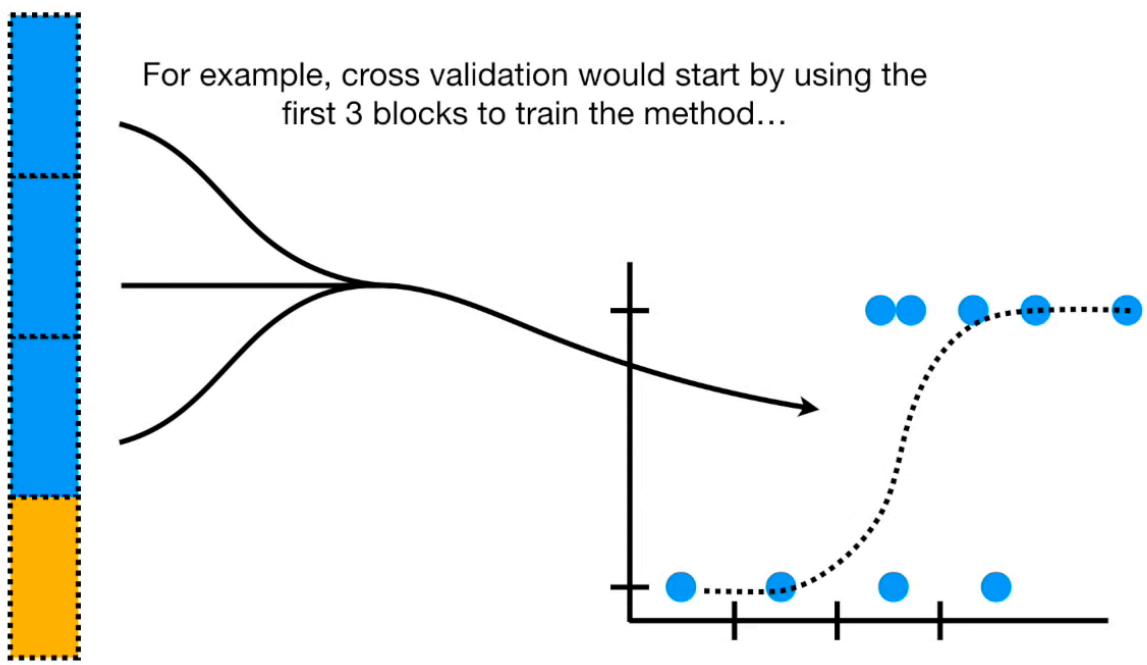


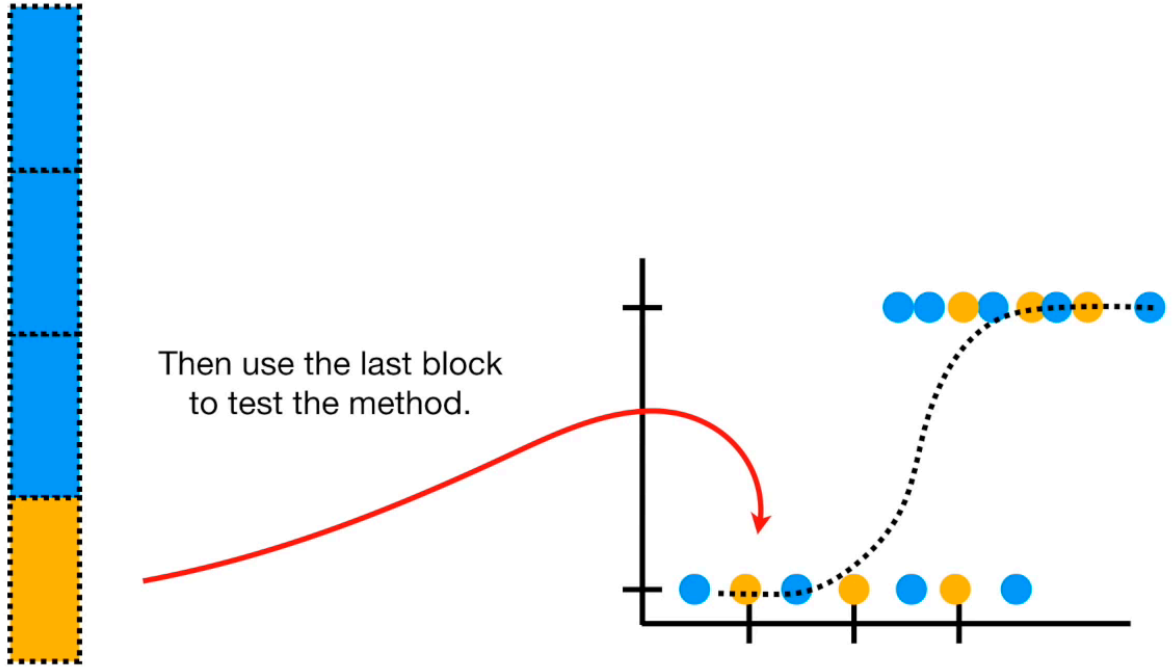




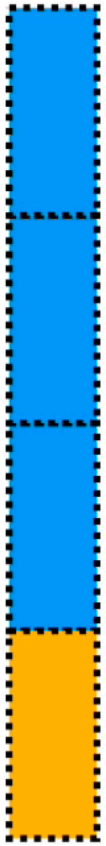
But how do we know that using the first 75% of the data for training and the last 25% of the data for testing is the best way to divide up the data?

Cross validation: missclassify





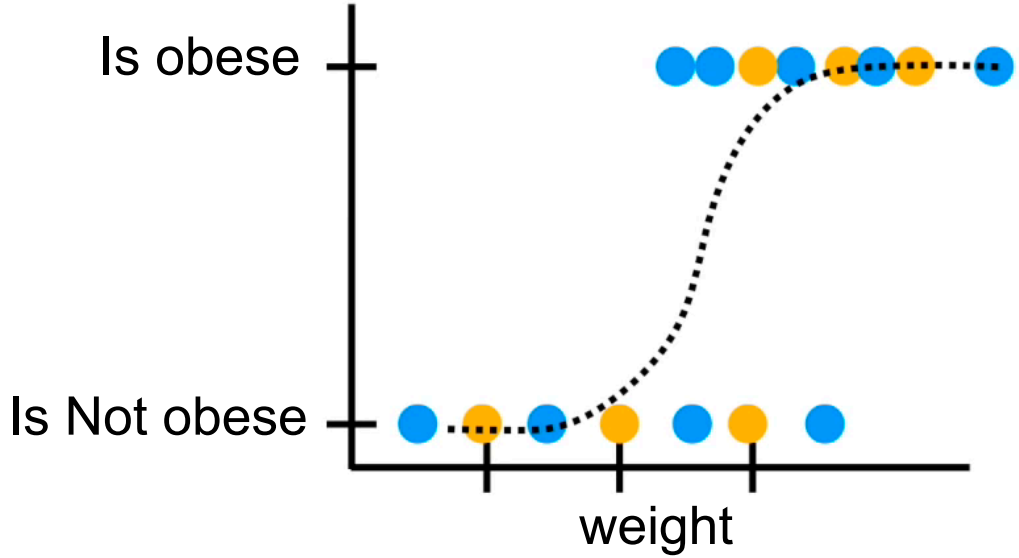
Cross validation: missclassify



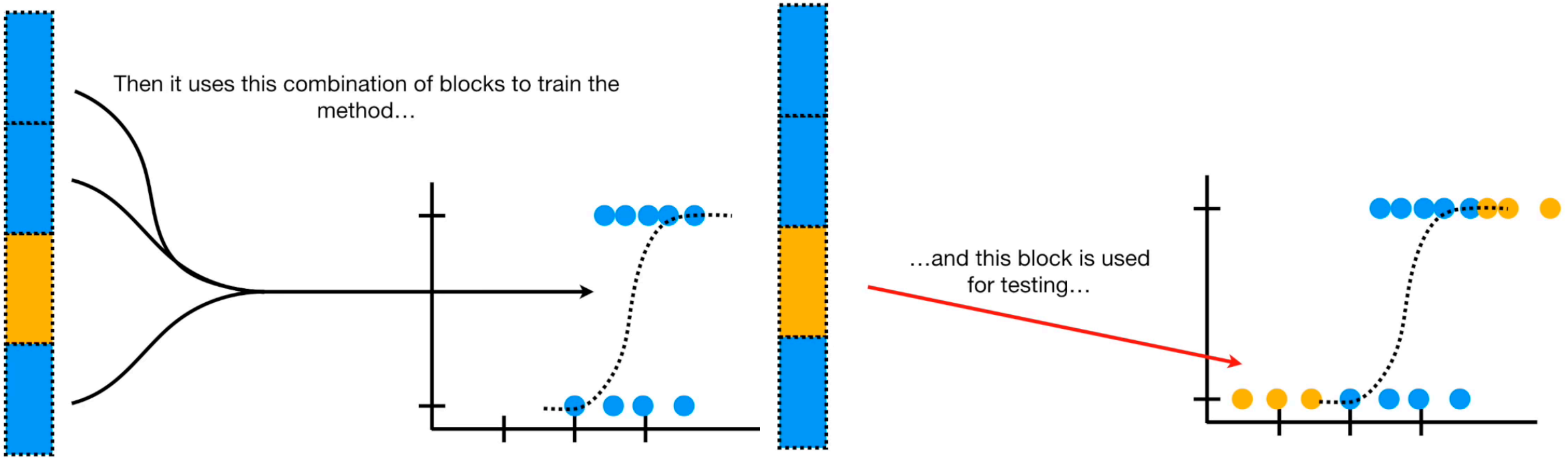
…and then it keeps track of how well the method did with the test data….

Test data categorization…

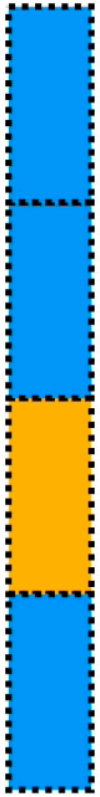
Correct	Incorrect
5	1



Cross validation: missclassify



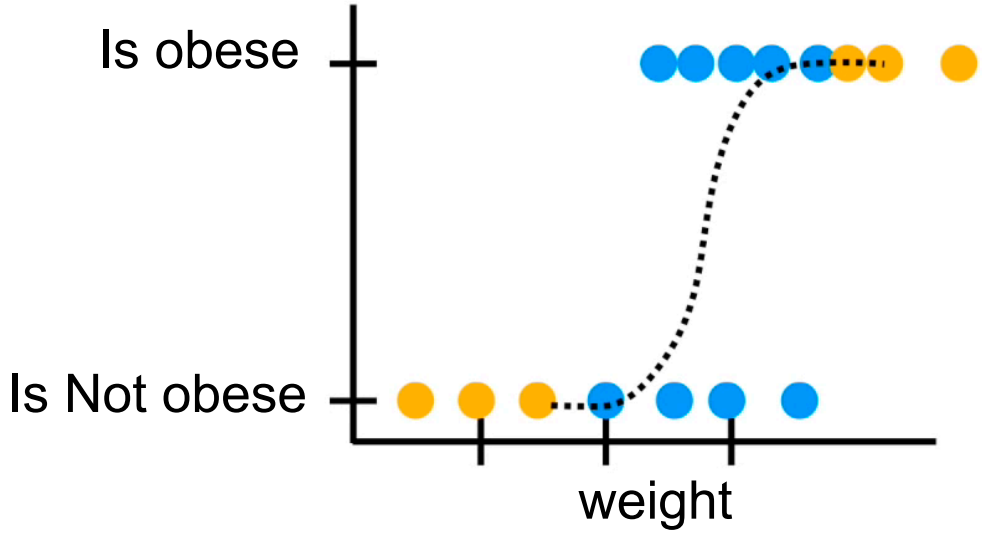
Cross validation: missclassify



…and then it keeps track of how well the method did with the test data….

Test data categorization…

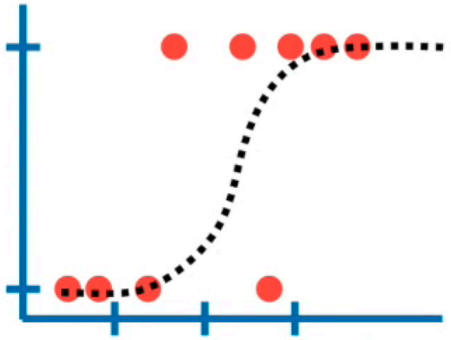
Correct	Incorrect
4	2



Cross validation: missclassify

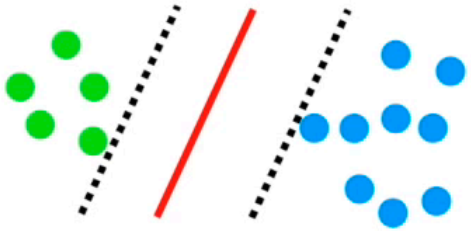
In the end, every block of data is used for testing and we can compare methods by seeing how well they performed.

Logistic Regression



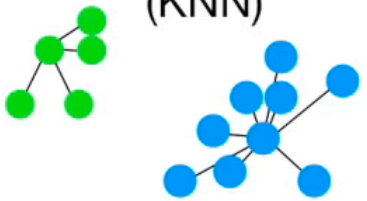
Correct	Incorrect
16	8

Support Vector machines (SVM)



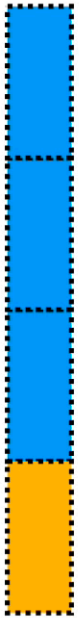
Correct	Incorrect
18	6

K-nearest neighbors (KNN)



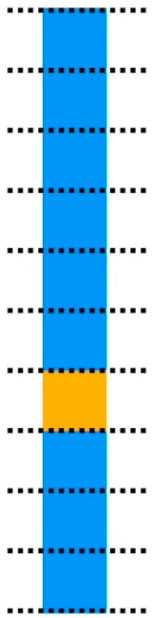
Correct	Incorrect
10	12

Cross validation: missclassify



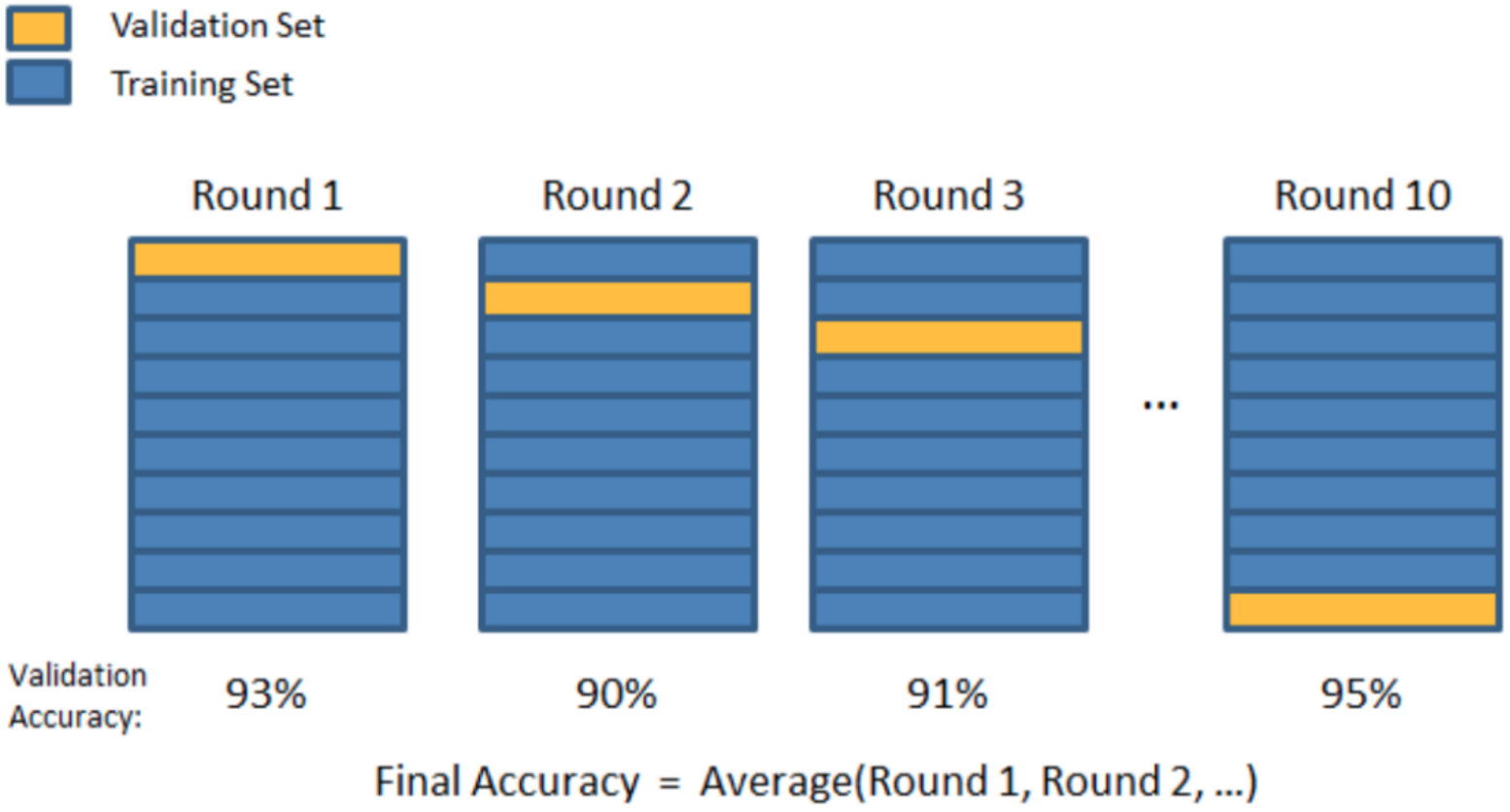
NOTE: In this example, we divided the data into 4 blocks. This is called **Four-Fold Cross Validation**.

However, the number of blocks is arbitrary.



That said, in practice, it is very common to divide the data into 10 blocks. This is called **Ten-Fold Cross Validation**.

Cross validation: conclusion



Cross validation: Exercise

train data

Training data - 24:0:2 - X-Partitioner (cross validation:)

File Edit Hilite Navigation View

Table "vehicle.data" – Rows: 677 | Spec – Columns: 19 | Properties | Flow Variables

Row ID	Col0	Col1	Col2	Col3	Col4	Col5	Col6	Col7	Col8	Col9	Col10
Row1	0.391	0.308	0.611	0.162	0.11	0.132	0.242	0.543	0.167	0.357	0.211
Row3	0.435	0.308	0.583	0.24	0.176	0.132	0.209	0.571	0.167	0.357	0.158
Row4	0.261	0.423	0.417	0.441	0.615	0.943	0.242	0.543	0.167	0.371	0.584
Row6	0.522	0.385	0.458	0.301	0.198	0.075	0.268	0.457	0.167	0.357	0.242
Row7	0.37	0.385	0.361	0.231	0.198	0.132	0.163	0.629	0.083	0.4	0.168
Row8	0.283	0.038	0.306	0.157	0.154	0.094	0.065	0.8	0	0.129	0.058
Row9	0.435	0.423	0.806	0.406	0.165	0.17	0.464	0.286	0.417	0.4	0.379
Row10	0.283	0.115	0.417	0.17	0.154	0.132	0.137	0.686	0.083	0.171	0.121
Row11	0.37	0.038	0.361	0.14	0.088	0.075	0.072	0.8	0	0	0.095
Row12	0.326	0.5	0.472	0.293	0.231	0.075	0.261	0.486	0.167	0.429	0.263
Row13	0.348	0.346	0.625	0.175	0.121	0.151	0.261	0.514	0.167	0.371	0.226
Row14	0.457	0.615	0.542	0.432	0.264	0.057	0.405	0.314	0.333	0.514	0.347
Row15	0.5	0.846	0.875	0.424	0.198	0.132	0.601	0.171	0.5	0.686	0.511
Row16	0.348	0.115	0.153	0.022	0.055	0.075	0.039	0.886	0	0.157	0.037
Row18	0.674	0.808	0.833	0.358	0.154	0.151	0.68	0.143	0.583	0.786	0.5
Row19	0.609	0.885	0.833	0.485	0.242	0.151	0.627	0.171	0.583	0.729	0.511
Row20	0.239	0.538	0.486	0.214	0.187	0.075	0.275	0.486	0.167	0.386	0.237
Row21	0.239	0.154	0.181	0.074	0.132	0.057	0.072	0.829	0	0.1	0.058
Row22	0.457	0.385	0.333	0.301	0.242	0.094	0.248	0.486	0.167	0.343	0.205
Row23	0.304	0.231	0.417	0.192	0.154	0.094	0.203	0.571	0.083	0.257	0.179
Row24	0.565	0.769	0.903	0.502	0.209	0.17	0.601	0.171	0.5	0.671	0.479
Row25	0.261	0.462	0.556	0.218	0.187	0.132	0.229	0.543	0.167	0.429	0.205
Row26	0.217	0.115	0.194	0.066	0.11	0.075	0.105	0.771	0.083	0.1	0.068
Row27	0.739	0.808	0.806	0.432	0.198	0.17	0.693	0.143	0.667	0.7	0.521
Row29	0.152	0.192	0.319	0.109	0.088	0.094	0.222	0.571	0.167	0.171	0.2
Row30	0.348	0.385	0.625	0.245	0.187	0.17	0.281	0.486	0.167	0.471	0.226
Row31	0.326	0.346	0.514	0.205	0.121	0.113	0.183	0.6	0.083	0.343	0.184
Row32	0.435	0.077	0.361	0.218	0.132	0.075	0.196	0.571	0.083	0.143	0.168

test data

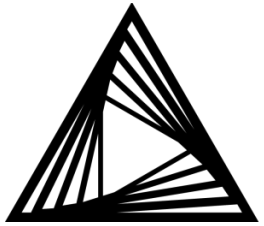
Test data - 24:0:2 - X-Part

File Edit Hilite Navigation View

Table "vehicle.data" – Rows: 169 | Spec – C

Row ID	Col0	Col1	Col2	Col3	Col4	Col5
Row0	0.478	0.577	0.597	0.323	0.275	0.151
Row2	0.674	0.654	0.917	0.459	0.209	0.151
Row5	0.739	0.923	0.917	0.297	0.033	0.075
Row17	0.565	0.308	0.514	0.406	0.242	0.075
Row28	0.63	0.462	0.625	0.389	0.187	0.075
Row37	0.37	0.577	0.639	0.882	0.868	0.887
Row47	0.261	0.346	0.361	0.079	0.077	0.075
Row48	0.348	0.538	0.569	0.188	0.187	0.17
Row54	0.609	0.885	0.889	0.354	0.066	0.075
Row56	0.522	0.423	0.778	0.397	0.176	0.132
Row57	0.348	0.538	0.611	0.127	0.088	0.17
Row59	0.261	0.231	0.389	0.066	0.055	0.057
Row69	0.543	0.615	0.611	0.502	0.297	0.094
Row71	0.522	1	0.944	0.537	0.253	0.17
Row82	0.457	0.115	0.389	0.1	0.077	0.094
Row84	0.348	0.269	0.5	0.367	0.319	0.094
Row91	0.457	0.192	0.611	0.236	0.088	0.132
Row92	0.674	0.731	0.833	0.38	0.132	0.132
Row95	0.522	0.654	0.944	0.467	0.198	0.151
Row96	0.348	0.346	0.556	0.205	0.165	0.075
Row103	0.413	0.192	0.278	0.114	0.165	0.057
Row104	0.63	0.462	0.597	0.41	0.198	0.057
Row106	0.565	0.5	0.903	0.459	0.187	0.17
Row111	0.457	0.577	0.597	0.253	0.187	0.151
Row117	0.783	0.769	0.958	0.511	0.242	0.189
Row124	0.261	0.462	0.431	0.201	0.176	0.113
Row126	0.348	0.346	0.417	0.192	0.165	0.094
Row127	0.261	0.308	0.361	0.223	0.198	0.377
Row136	0.283	0.577	0.486	0.14	0.121	0.151

SVM: **breast cancer**



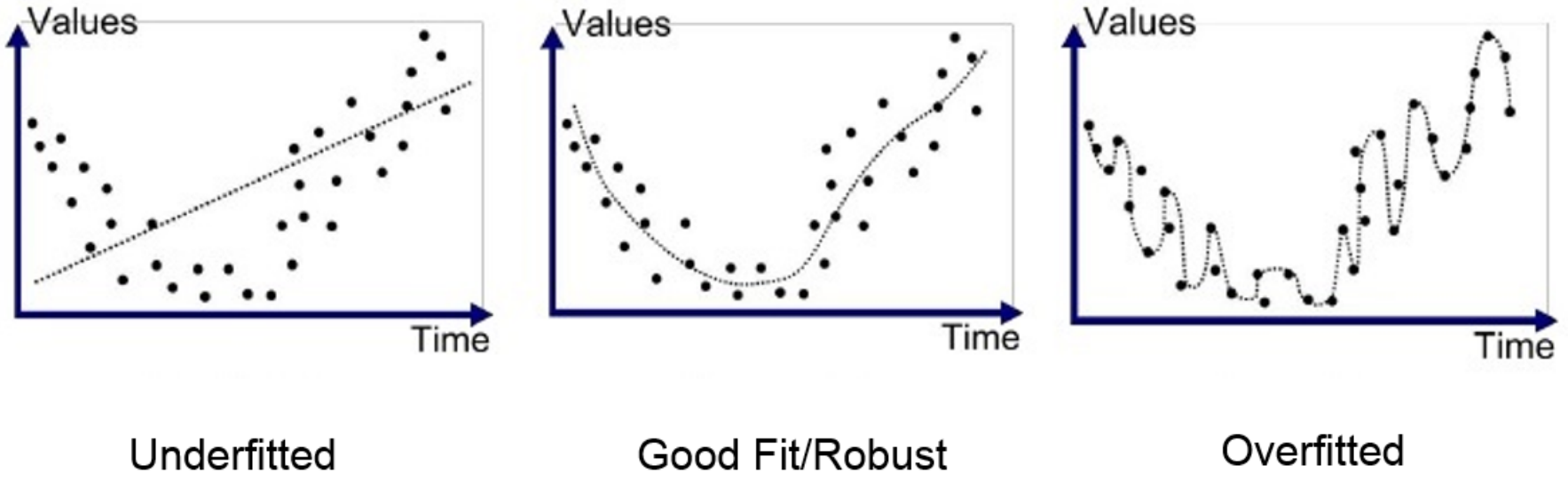
Cross Validation

AI CONCEPTS

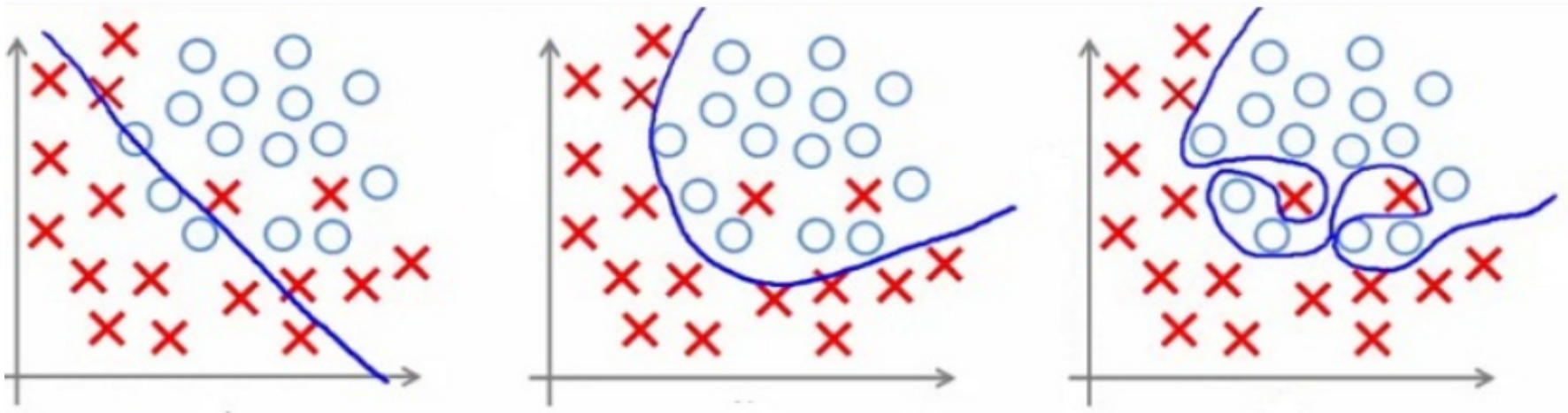
Concepts: overfitting

Concepts: overfitting in forecasting

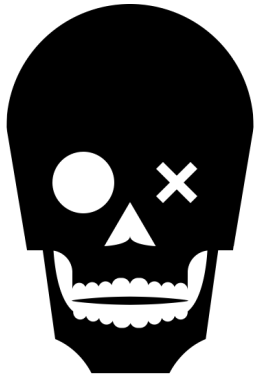
Overfitting is the **case** where the overall cost is really small, but the **generalization of the model is unreliable**.
This is due to the model learning “too much” from the training data set



Concepts overfitting in classification



Concepts: overfitting



How to Prevent Overfitting?

1) Train and test split. Large datasets required
2) Cross-validation. Cross-validation is a powerful preventative measure against overfitting.
3) Data augmentation: we can apply data augmentation to artificially increase the size of our dataset
4) Feature selection: we should only select the most important features for training so that our model doesn't need to learn for so many features and eventually overfit
5) Train with more data. It won't work every time, but training with more data can help algorithms detect the signal better.
4) Dropout.
5) Early stopping.
6) Regularization.
7) Ensembling.

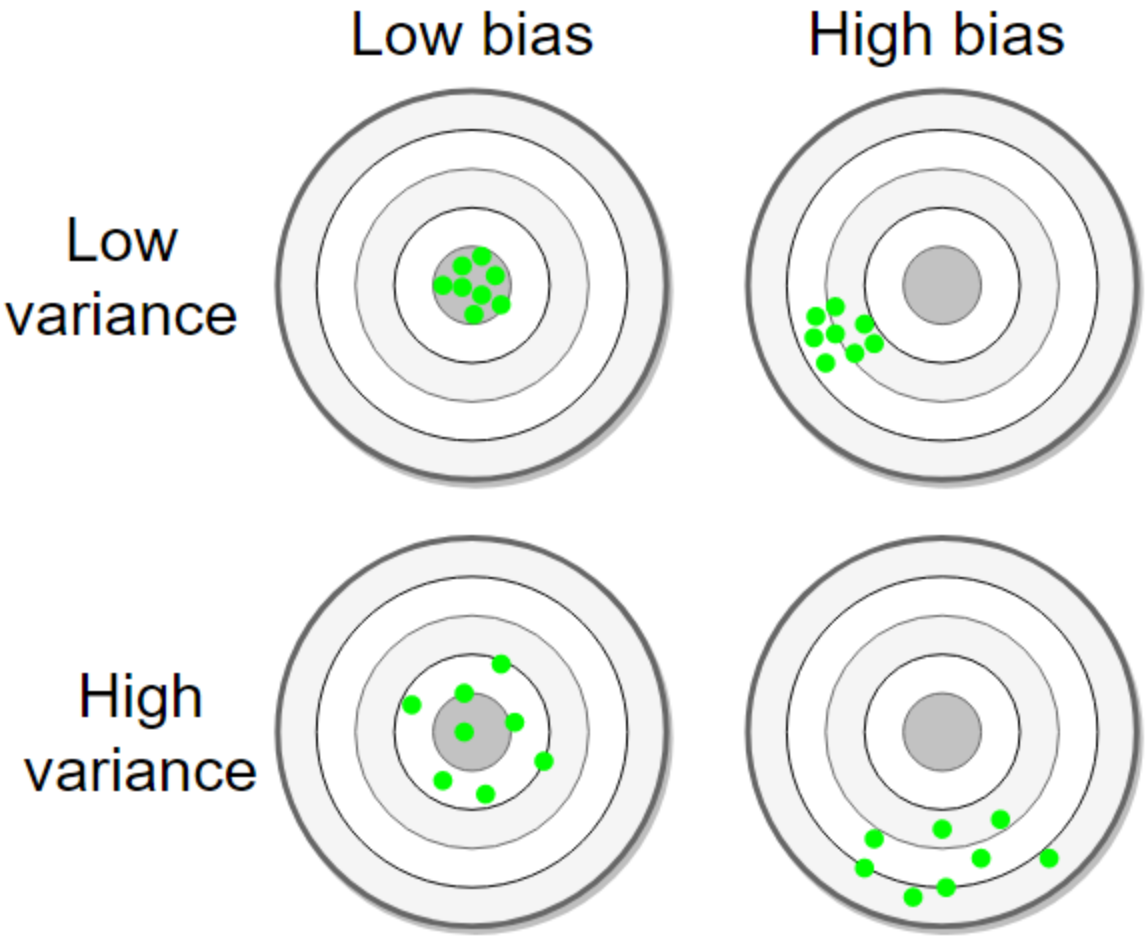
AI CONCEPTS

Concepts: Bias and variance

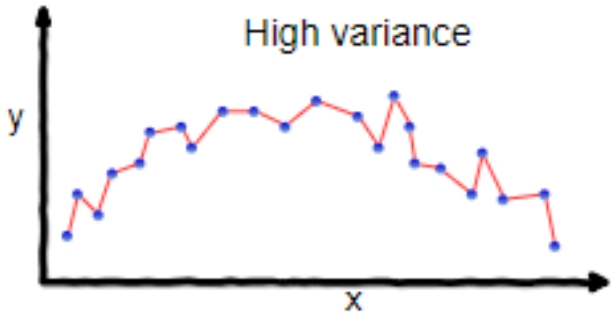
Concepts: Bias and variance

Bias is the simplifying assumptions made by the model to make the target function easier to approximate.

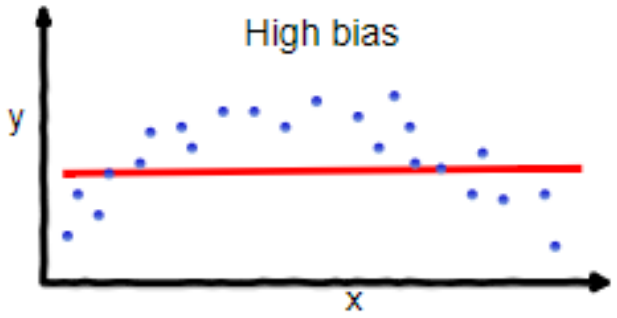
Variance is the amount that the estimate of the target function will change given different training data



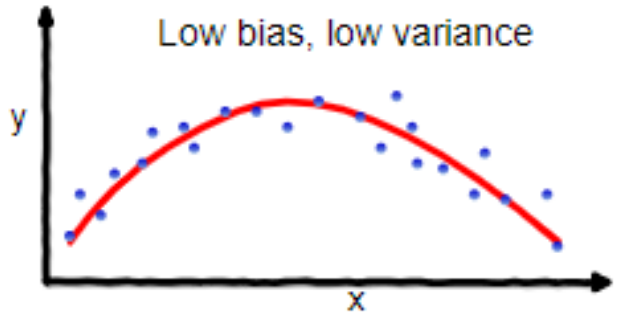
Concepts: Bias and variance



overfitting



underfitting



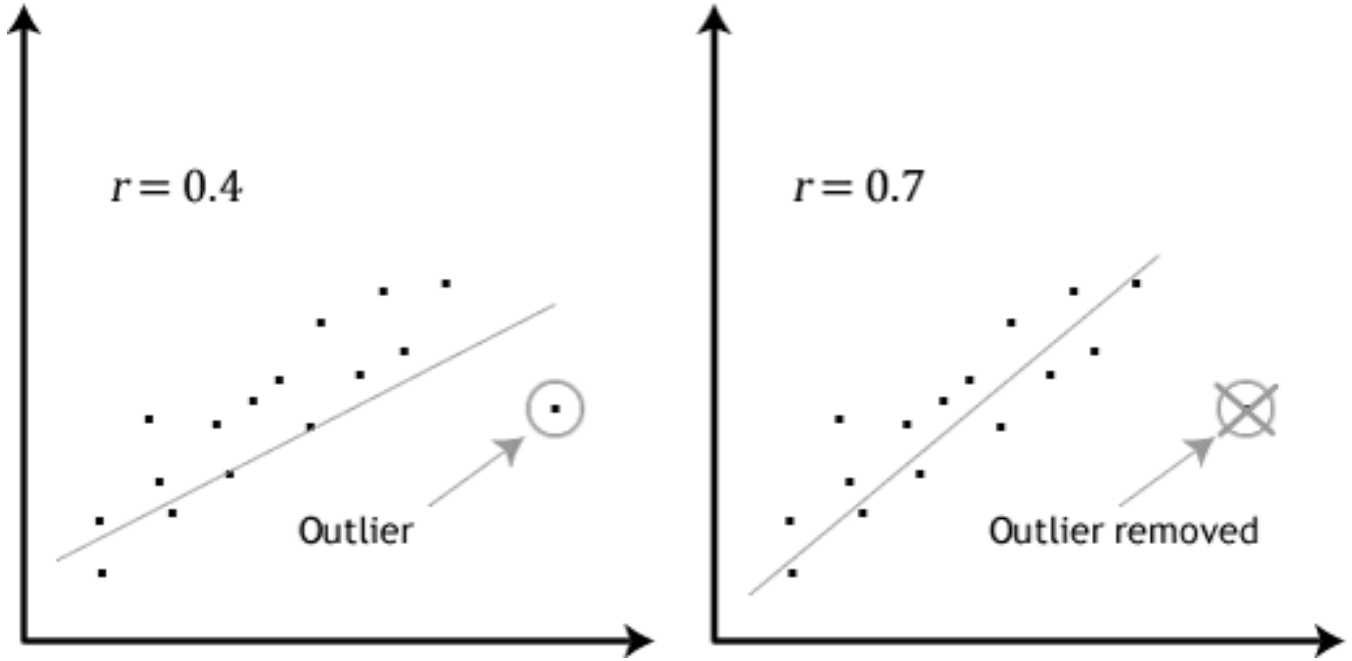
Good balance

AI CONCEPTS

Concepts: Outlier

Concepts: Outlier

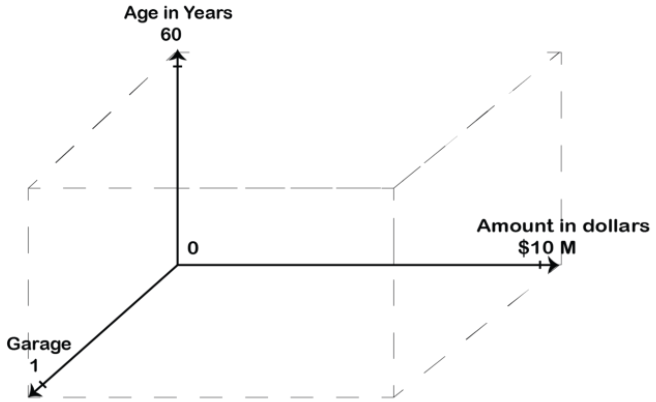
Machine learning algorithms are **very sensitive to the range and distribution of attribute values.**
Data outliers can spoil and mislead the training process resulting in longer training times, less accurate models and ultimately poorer results.



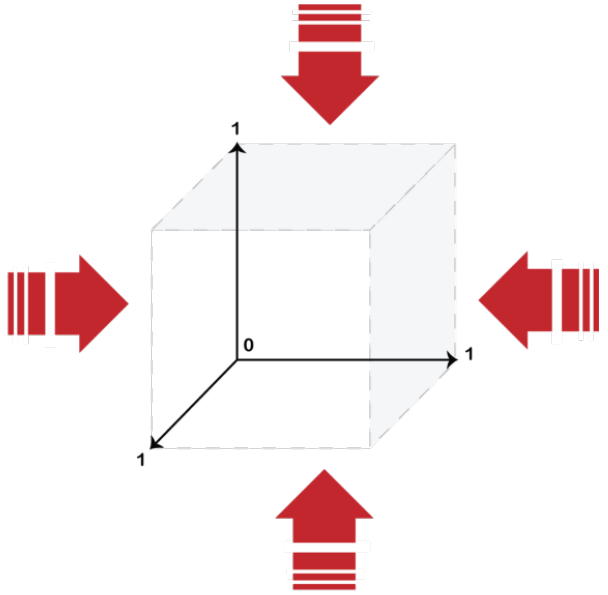
AI CONCEPTS

Concepts: Feature Scaling, Normalization & Standarization

Concepts: Normalization



I. Unscaled

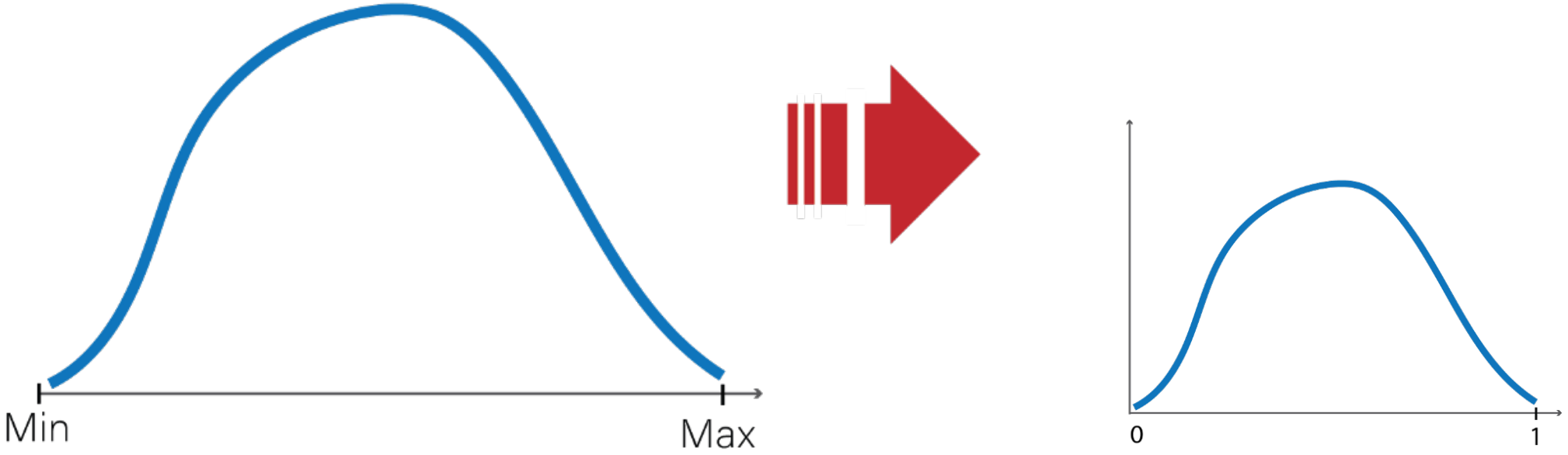


II. Scaled

“Age in years” indicates the age of the house, “Amount in Dollars” indicates the listed price of the house, and “Garage” is a flag feature that indicates if the house has a garage or not.

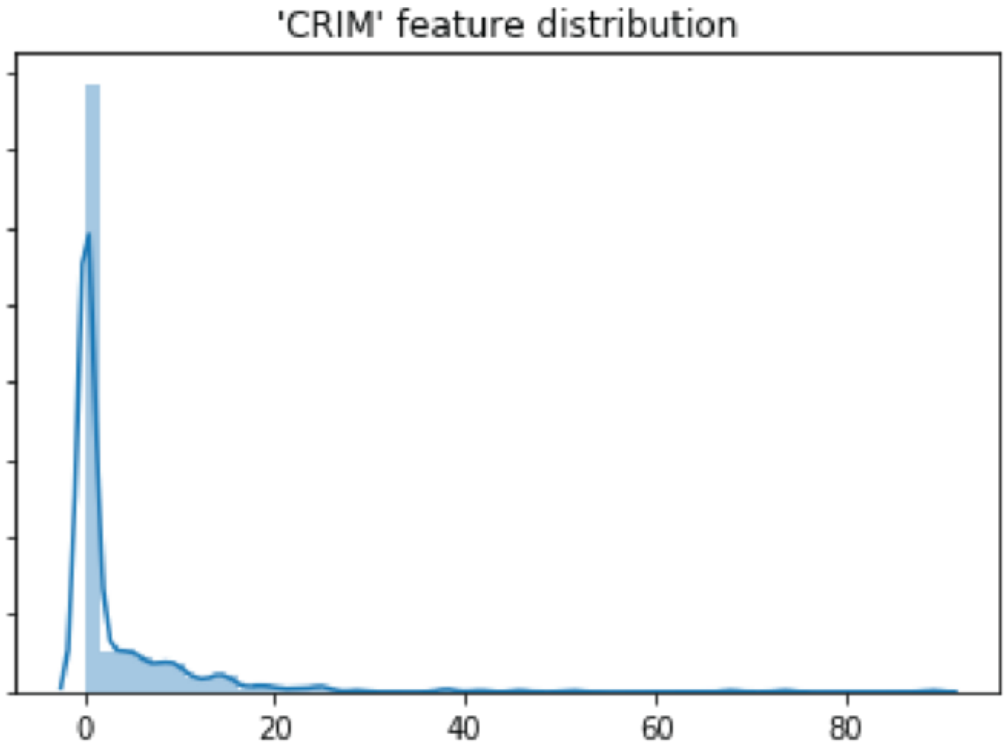
AI CONCEPTS

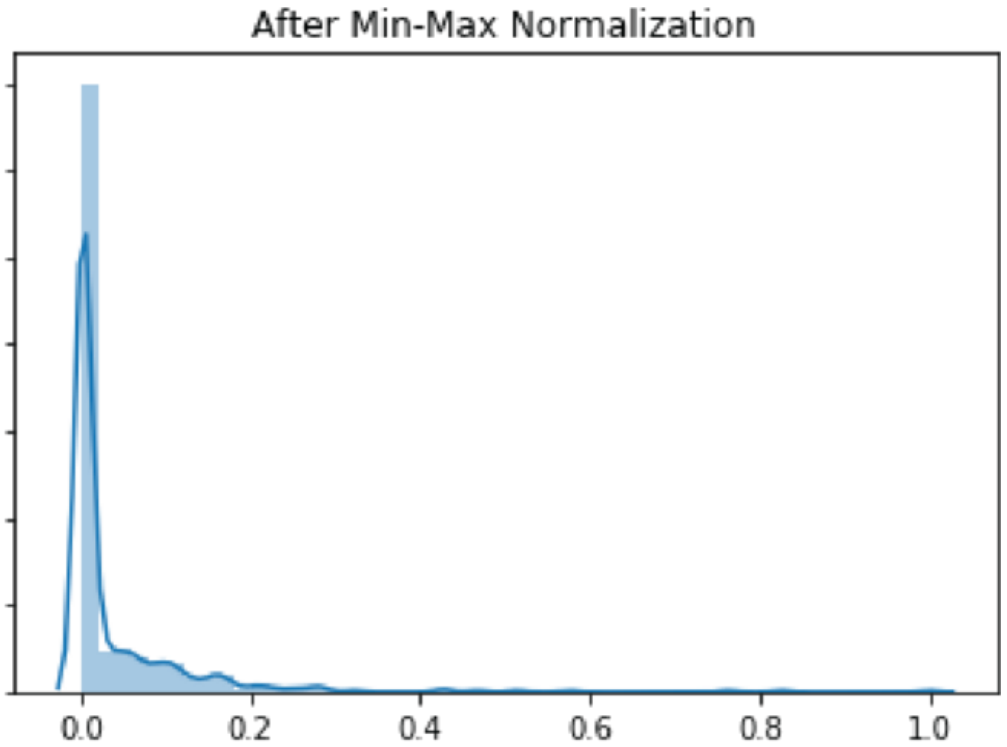
1// Normalization(scaling) transforms features with different scales to a fixed scale of 0 to 1. This ensures that no specific feature dominates the other.



AI CONCEPTS

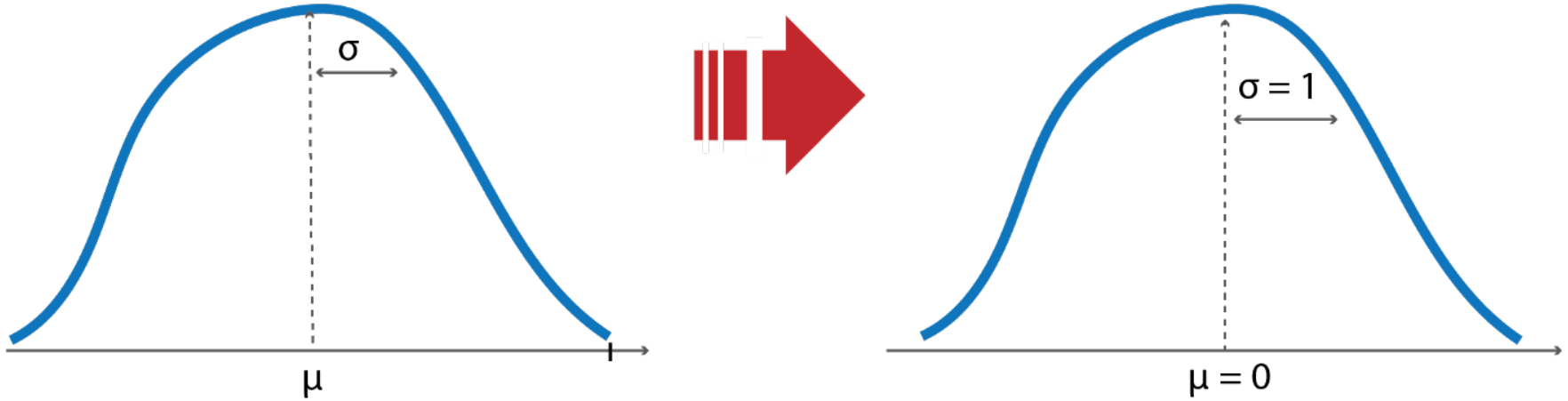
Normalization allows us to transform all the features with varying scales to a common scale but does not handle outliers well.





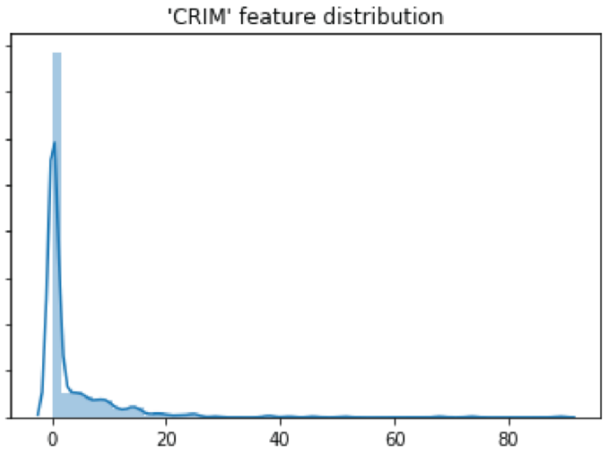
AI CONCEPTS

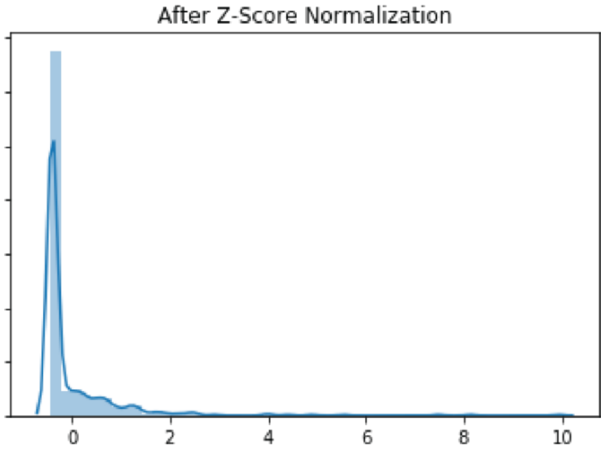
2// Standardization — Standardization transforms features such that their mean (μ) equals 0 and standard deviation (σ) equals 1. The range of the new min and max values is determined by the standard deviation of the initial un-normalized feature.

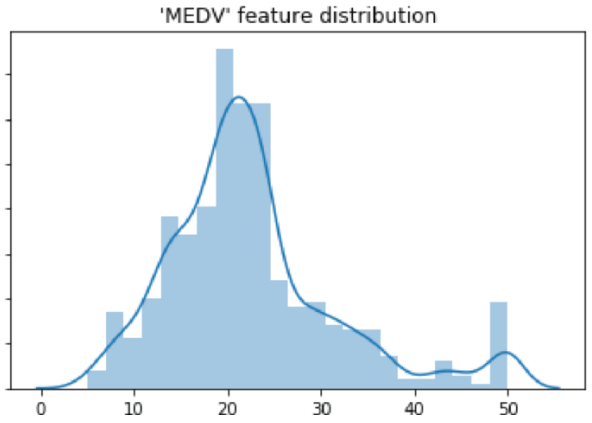


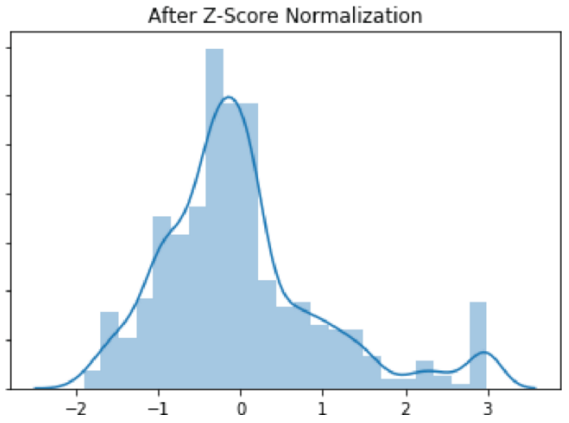
AI CONCEPTS

Unlike normalization, the mean and standard deviation of a feature is more robust to new data than the min and max values. Standardization is more effective if the feature has a Gaussian distribution, to begin with. Observe what happens when you standardize the '*CRIM*' feature which has a right-skewed distribution.









Whereas, if you do the same on the '*MEDV*' feature, which has a Gaussian or Gaussian-like distribution, the z-score transformation is more effective. 'MEDV' is the median value of owner-occupied homes (in $1000's).

Normalization or Standardization, which one is better?
The answer is: you guessed it right, *it depends*. It is sometimes good to perform selective scaling of the numerical features, but the better option is to try out different combinations to data scaling and then comparing the performances of the model.

The different combinations could be:
a. Simply normalizing all the features
b. Simply standardizing all the features
c. Selectively normalizing features with Non-Gaussian distribution and standardizing features with Gaussian distribution.

AI CONCEPTS

Normalization allows us to transform all the features with varying scales to a common scale but **does not handle outliers well.** On the contrary, s**tandardization is more robust to outliers**, new data and facilitates faster convergence of loss function for some algorithms.
Therefore, standardization is typically preferred over Normalization.

Algorithms	Feature Scaling
Algorithnms that work with distances	
K-Mean Clustering	Yes
K-Nearest Neighbour (KNN)	Yes
Support Vector Machine (SVM)	Yes
Principal Component Analysis (PCA)	Yes
Linear Discriminant Analysis (LDA)	Yes
Gradient Descent based algortihms	
Regularized Linear Regression	Yes
Regularized Logistic Regression	Yes
Neural Networks	Yes
Tree based algorithms	
Decision Trees	No
Random Forests	No
XG-Boosts	No

AI CONCEPTS

Concepts: One hot encoder

One hot encoder

123

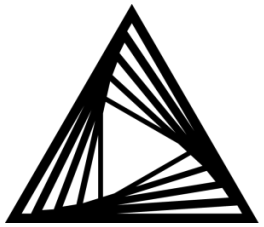
However, sometimes the best features are **categorical and do not fit into a linear regression model**. One-hot encoding is a great tool for turning some of these **categorical features into multiple binary features**; the presence or absence of the individual categorical unit can then be fit into the linear regression.

SALE_DATE	BOROUGH
2019-01-01	C
2019-01-01	D
2019-01-02	B
2019-01-02	C
2019-01-02	C

SALE_DATE	BOROUGH_C	BOROUGH_D	BOROUGH_B	BOROUGH_E	BOROUGH_A
2019-01-01	1	0	0	0	0
2019-01-01	0	1	0	0	0
2019-01-02	0	0	1	0	0
2019-01-02	1	0	0	0	0
2019-01-02	1	0	0	0	0

Linear regression: one hot encoder

123



One hot Encoder

Unsupervised

ML // UNSUPERVISED

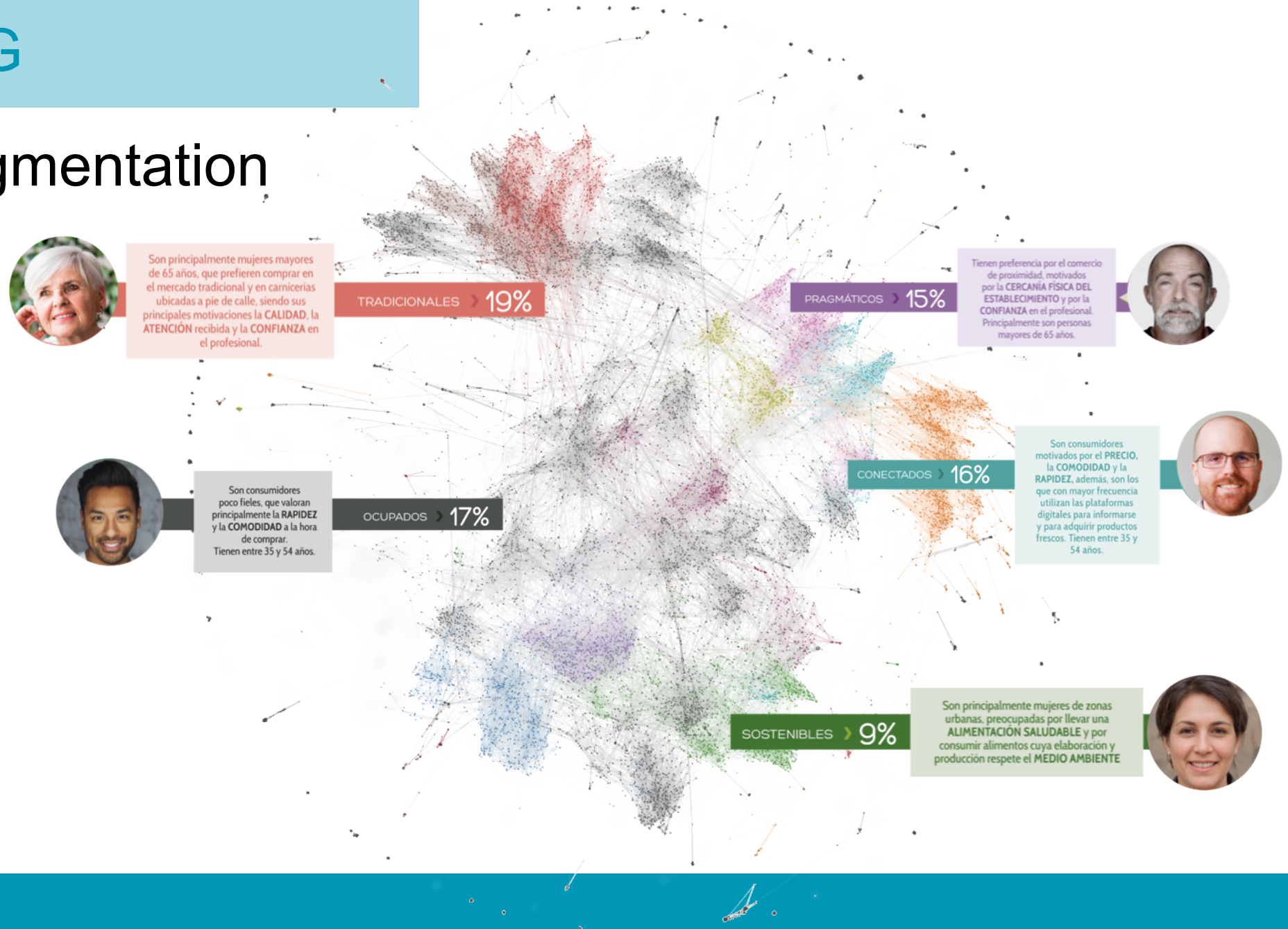
Clustering

CLUSTERING

“Understanding our customers is the key providing them a good service and sustain a profitable business.”

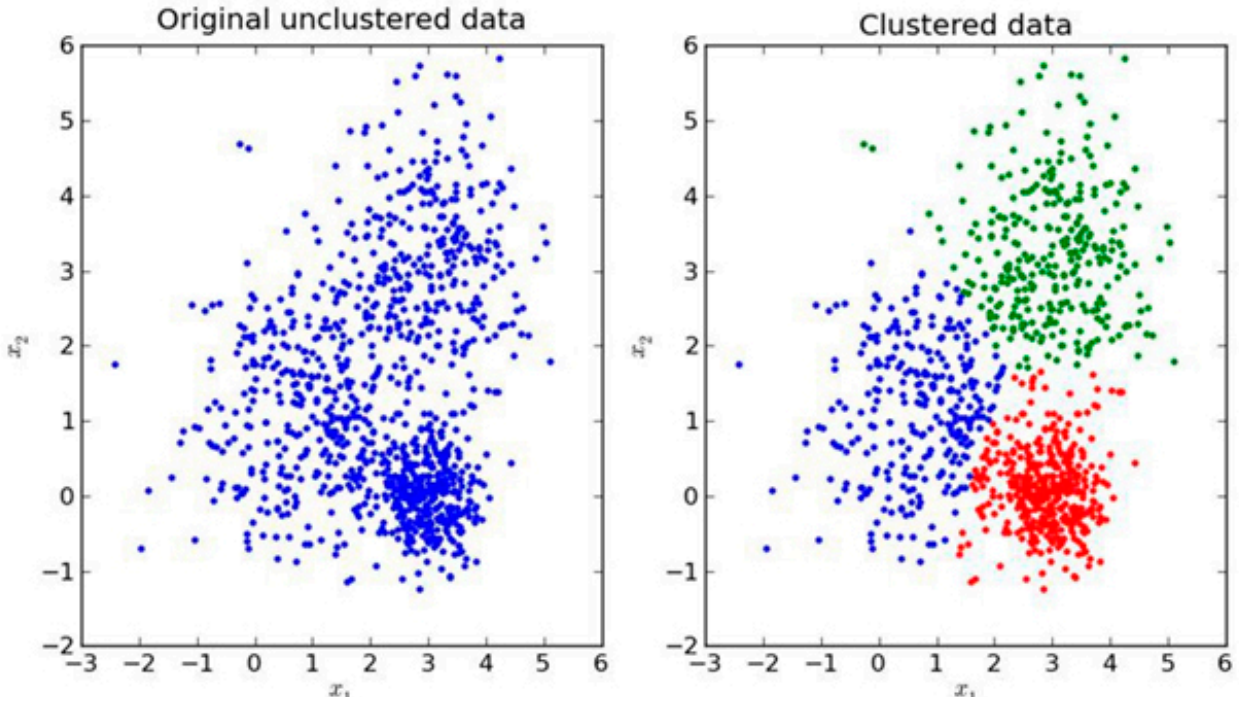
CLUSTERING

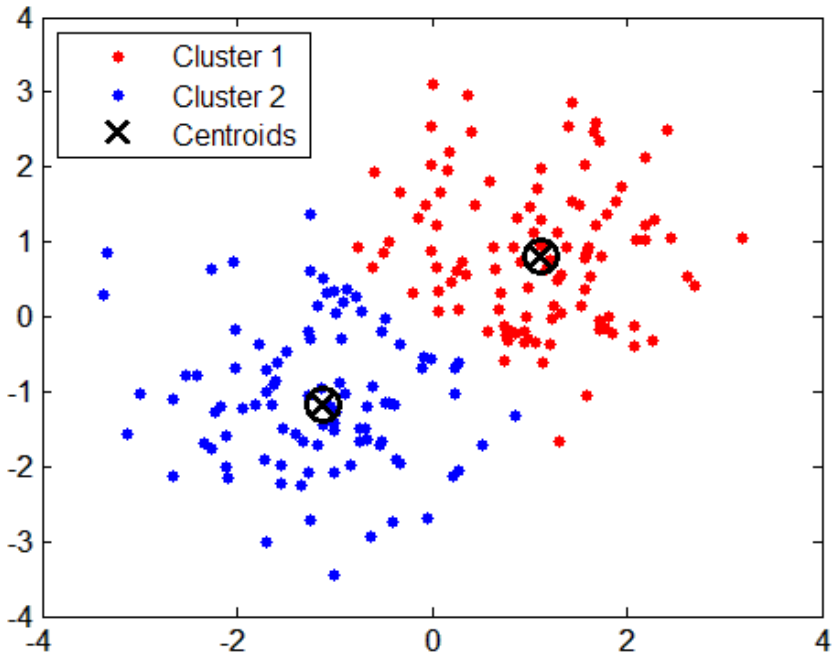
Customer segmentation



CLUSTERING

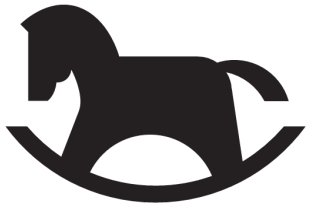
Clustering is the task of **dividing the population or data points into a number of groups such that data points in the same groups are more similar to other data points in the same group than those in other groups.**





CLUSTERING

In simple words, the aim is to segregate groups with similar traits and assign them into clusters.
The goal of clustering is to find groups in the data, with the number of groups represented by the variable K. The algorithm works iteratively to assign each data point to one of K groups based on the features that are provided. In the reference image below, K=2, and there are two clusters identified from the source dataset.

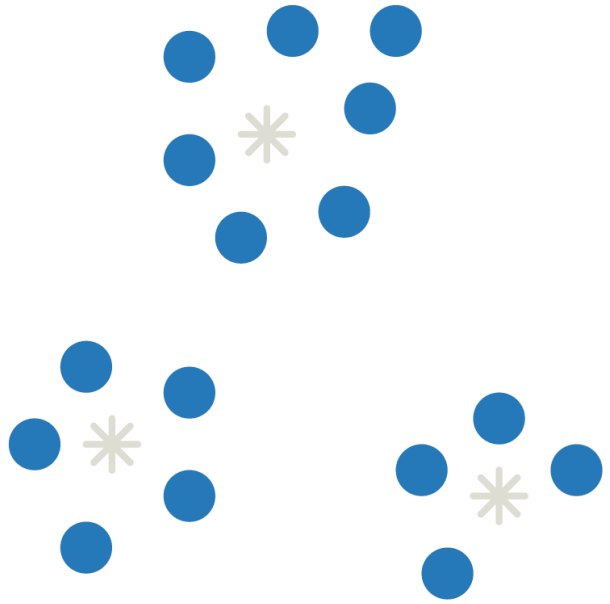


Clustering Playground

CLUSTERING // K-means

K-means: 123

-distance to cluster´s center
-<u>number of clusters is known</u>
-fast clustering of large datasets



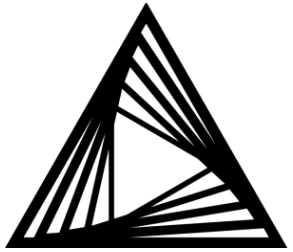
Pros:

- **interpretability of the clusters may not be required**
- **quick solution is sufficient to generate insights for most cases**.
- **Big data problem as the algorithm can be scaled easily**

Cons:

- **The dataset contains a lot of categorical variables.**
- **Outliers** may skew the clusters substantially

Knime Exercise k-means



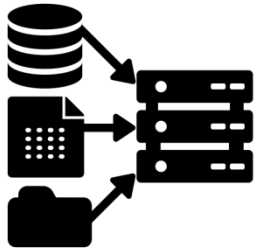
https://hub.knime.com/knime/spaces/Examples/latest/04_Analytics/03_Clustering/01_Performing_a_k-Means_Clustering

CLUSTERING // K-means

Exercise k-means: **credit card purchases**

Exercise k-means: **credit card purchases**

The sample Dataset summarizes the usage behavior of about **9000 active credit card holders** during the last 6 months. The file is at a customer level with **18 behavioral variables.**



CC GENERAL.csv

CLUSTERING // K-means

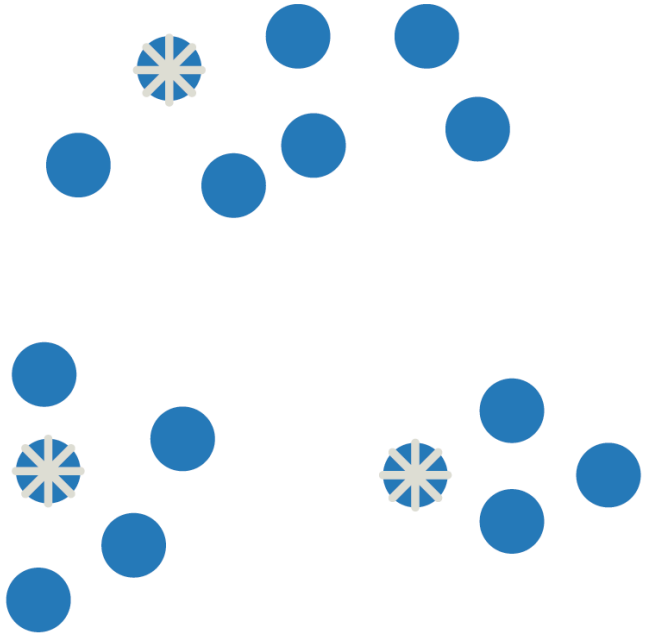
variables

Following is the Data Dictionary for Credit Card dataset :-
CUST_ID : Identification of Credit Card holder (Categorical)
BALANCE : Balance amount left in their account to make purchases
(BALANCE_FREQUENCY : How frequently the Balance is updated, score between 0 and 1 (1 = frequently updated, 0 = not frequently updated)
PURCHASES : Amount of purchases made from account
ONEOFF_PURCHASES : Maximum purchase amount done in one-go
INSTALLMENTS_PURCHASES : Amount of purchase done in installment
CASH_ADVANCE : Cash in advance given by the user
PURCHASES_FREQUENCY : How frequently the Purchases are being made, score between 0 and 1 (1 = frequently purchased, 0 = not frequently purchased)
ONEOFFPURCHASESFREQUENCY : How frequently Purchases are happening in one-go (1 = frequently purchased, 0 = not frequently purchased)
PURCHASESINSTALLMENTSFREQUENCY : How frequently purchases in installments are being done (1 = frequently done, 0 = not frequently done)
CASHADVANCEFREQUENCY : How frequently the cash in advance being paid
CASHADVANCETRX : Number of Transactions made with "Cash in Advanced"
PURCHASES_TRX : Number of purchase transactions made
CREDIT_LIMIT : Limit of Credit Card for user
PAYMENTS : Amount of Payment done by user
MINIMUM_PAYMENTS : Minimum amount of payments made by user
PRCFULLPAYMENT : Percent of full payment paid by user TENURE : Tenure of credit card service for user

CLUSTERING // K-medoids

K-medoids: **123** **category**

-center coincide with data points
-is an algorithm based on partition.
-for fast clustering of categorical data (One hot encoding)
-to scale large datasets



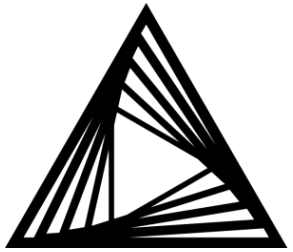
Pros:

- **selects the most centered member belonging to the cluster.**
- is sensitive to outliers or noise.

Cons:

- **It may obtain different results for different runs on the same dataset because the first k medoids are chosen randomly.**

Knime Exercise k-medoids



https://hub.knime.com/knime/spaces/Examples/latest/04_Analytics/03_Clustering/02_Performing_a_k-Medoids_Clustering

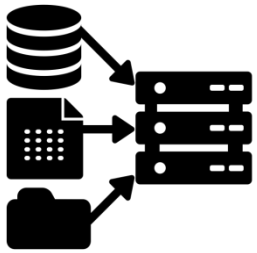
Exercise k-medoids: **on-line shopping intention**

Exercise k-medoids: **on-line shopping intention**

The dataset consists of feature vectors belonging to 12,330 sessions. The dataset was formed so that each session would belong to **a different user** in a 1-year period to avoid any tendency to a specific campaign, special day, user profile, or period.

The dataset consists of **10 numerical and 8 categorical attributes**.

The 'Revenue' attribute can be used as the class label**.**



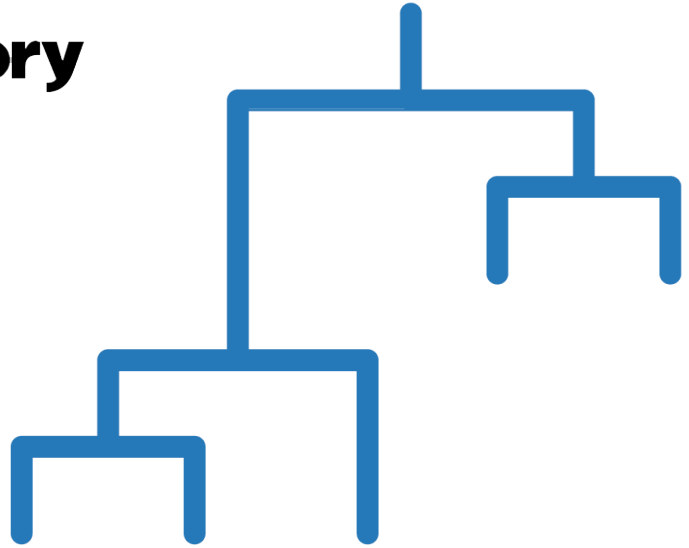
Online_shoppers_intention.csv

Exercise k-medoids: **on-line shopping intention**

Administrative", "Administrative Duration", "Informational", "Informational Duration", "Product Related" and "Product Related Duration" represent the number of different types of pages visited by the visitor in that session and total time spent in each of these page categories.
The values of these features are derived from the URL information of the pages visited by the user and updated in real time when a user takes an action, e.g. moving from one page to another.
The "Bounce Rate", "Exit Rate" and "Page Value" features represent the metrics measured by "Google Analytics" for each page in the e-commerce site. The value of "Bounce Rate" feature for a web page refers to the percentage of visitors who enter the site from that page and then leave ("bounce") without triggering any other requests to the analytics server during that session. The value of "Exit Rate" feature for a specific web page is calculated as for all pageviews to the page, the percentage that were the last in the session. The "Page Value" feature represents the average value for a web page that a user visited before completing an e-commerce transaction.
The "Special Day" feature indicates the closeness of the site visiting time to a specific special day (e.g. Mother's Day, Valentine's Day) in which the sessions are more likely to be finalized with transaction. The value of this attribute is determined by considering the dynamics of e-commerce such as the duration between the order date and delivery date.
For example, for Valentina's day, this value takes a nonzero value between February 2 and February 12, zero before and after this date unless it is close to another special day, and its maximum value of 1 on February 8.
The dataset also includes operating system, browser, region, traffic type, visitor type as returning or new visitor, a Boolean value indicating whether the date of the visit is weekend, and month of the year.

CLUSTERING // Hierarquical

Hierarquical clustering: category



-grouping objects into a binary hierarquical tree
-number of clusters is unknown
-you want visualization to guide your selection

Pros

- Categorical data
- Finding **outliers** and aberrant groups
- Displaying the results using an easy-to-read **dendrogram**
- Flexibility because of the different dissimilarity functions available (i.e. complete linkage, single linkage, average linkage, minimum variance, etc.) that each give very different results

Cons

- As the amount of data increases, it becomes very time/resources intensive as it processes each data point iteratively, looping through the entire dataset every time. Hierarchical clustering **does not scale well.**

CLUSTERING // K-modes

K-modes: category

-Categorical

-An extension to k-means **by replacing means of clusters by modes**

-to scale large datasets

-minimize a dissimilarity measure: it counts the number of “features” that are not the same

Pros

- When the dataset contains **categorical data exclusively**

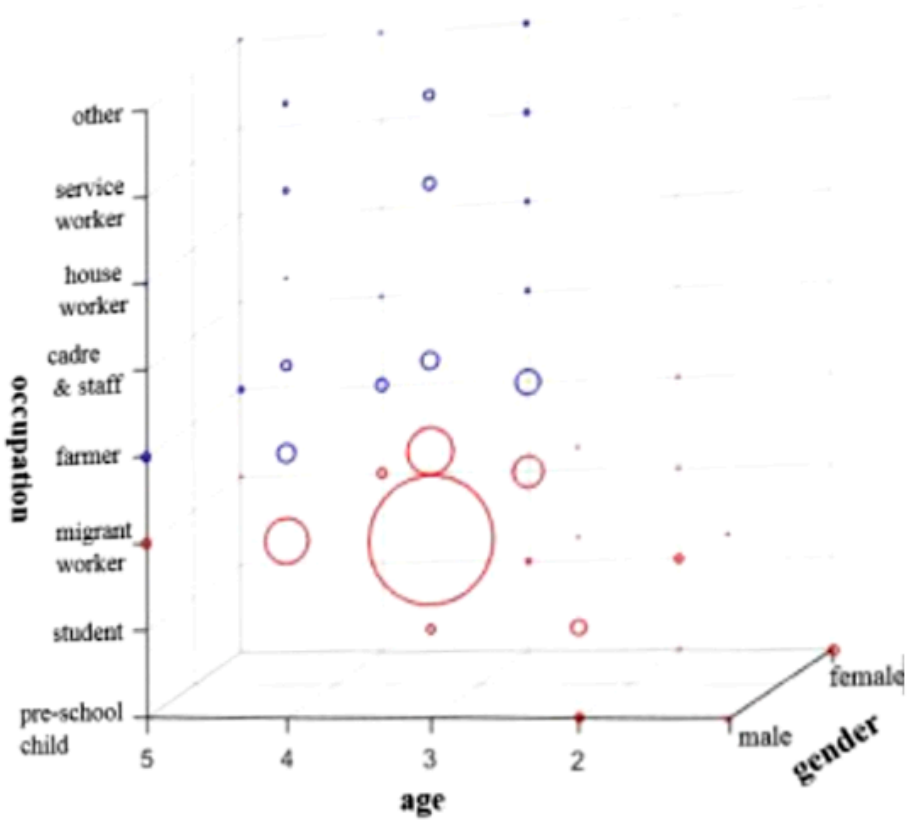
Cons

- **Data types are mixed not working efficiently**
- **If a given category is particularly prevalent, this may become an issue** as the algorithm will not take it into account when clustering.

CLUSTERING // K-prototypes

K-Prototypes: 123 category

-neural network based 2D map
-to visualize dimensional data in 2D, 3D
-to scale large datasets
-The k-prototypes algorithm combines k-modes and k-means and is able to cluster mixed numerical / categorical data.



Pros

- **Large Mixed Dataset categorical & numerical**

Cons

- Needing a quick solution as it is not among the scikit-learn supported algorithms (requires the use of PyClustering or custom code to implement)
- It may be unclear what weighs to give to the categorical variables

CLUSTERING // Use cases

Paper Clustering Starbucks

Paper Starbucks Clustering

Geo Blink

CLUSTERING // Use cases

Use **case**

Sample Customer Cluster Analysis Result

The following chart shows the results of a three-dimension cluster analysis performed on the customer base of an e-commerce site. This analysis resulted in the discovery of four customer personas.

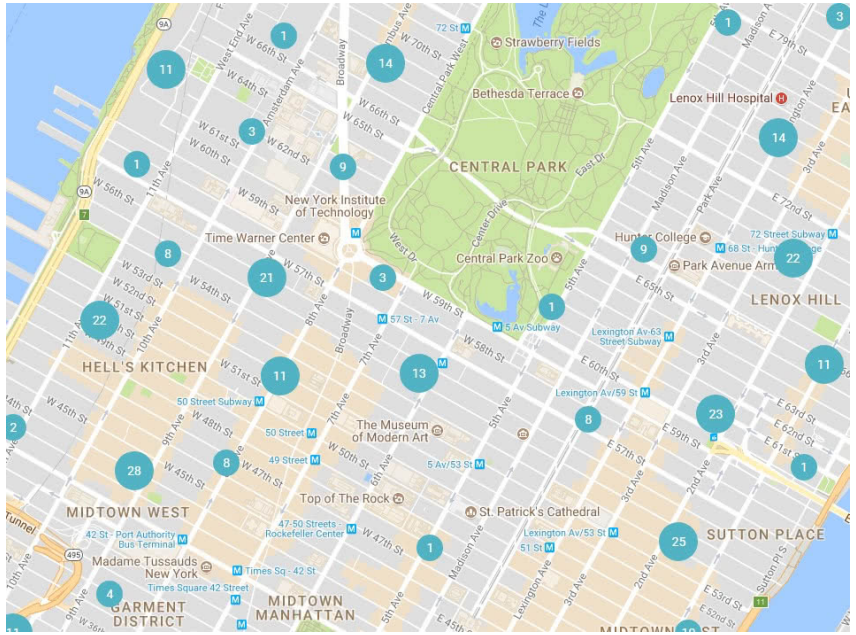
Customers		Days Since Last Purchase	Number of Purchases (Past 12 Months)	Net Revenue (Past 12 Months)
High Spenders	922	9	4	$154
Mid Spenders	581	54	3	$121
Risk of Churn	807	192	2	$70
Low Spenders	1,361	192	2	$4
	3,671	**447**	**3**	**$87**

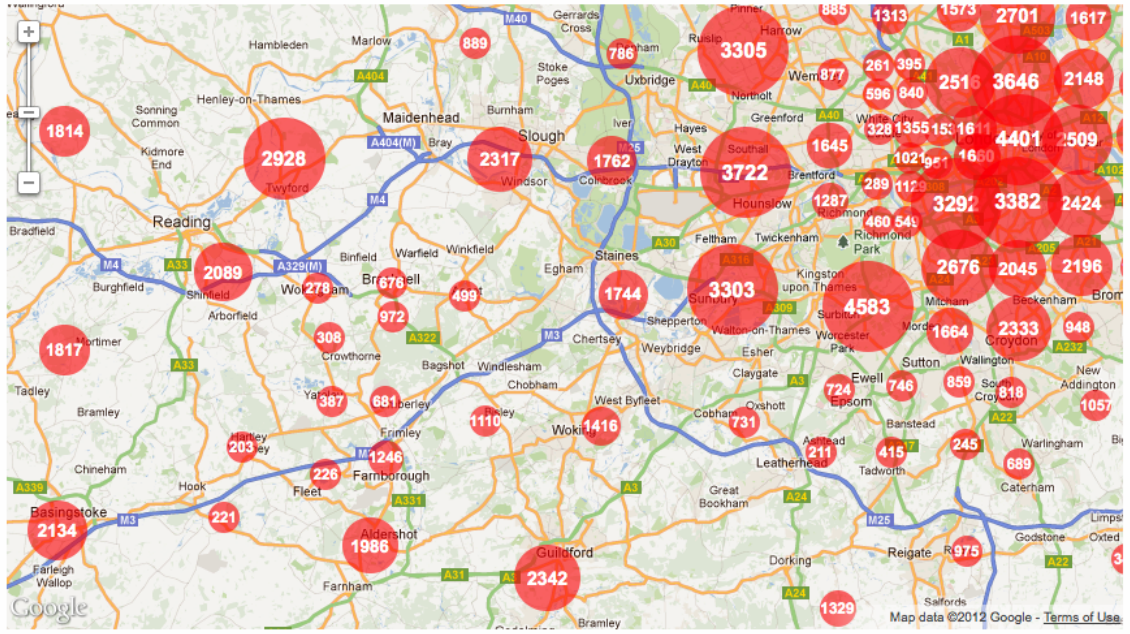
CLUSTERING // Use cases

1. Location intelligence

This technology makes it possible to collect and enrich a great variety and volume of data, such as information from GPS, commercial transactions, sociodemographic and economic data, data related to pedestrian traffic or points of sale. All of them undergo an exhaustive geospatial analysis.

The information is then displayed in a simple interface in the form of a map that allows users to visually **conceptualize the factors that affect the performance of their business in different locations.**





CLUSTERING // Use cases

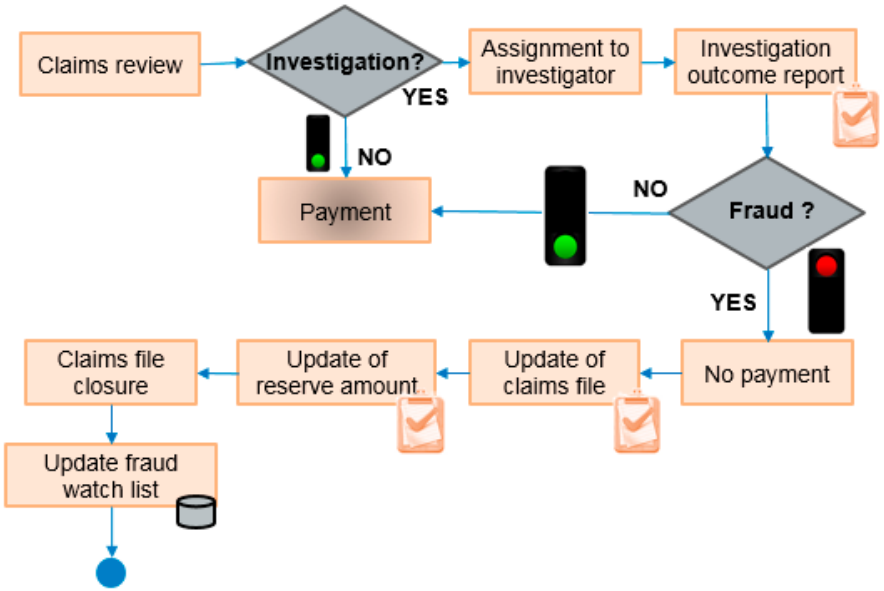
2. Video Games - Similar players

analyzing player stats has always been a critical element of the sporting world, and with increasing competition, machine learning has a critical role to play here. as an interesting exercise, if you would like to create a **fantasy draft team and like to identify similar players based on player stats**, k-means can be a useful option.

CLUSTERING // Use cases

3. insurance fraud detection

Machine learning has a critical role to play in fraud detection and has numerous applications in automobile, healthcare, and insurance fraud detection. utilizing past historical data on fraudulent claims, **it is possible to isolate new claims based on its proximity to clusters that indicate fraudulent patterns.** since insurance fraud can potentially have a multi-million dollar impact on a company, the ability to detect frauds is crucial.



CLUSTERING // Use cases

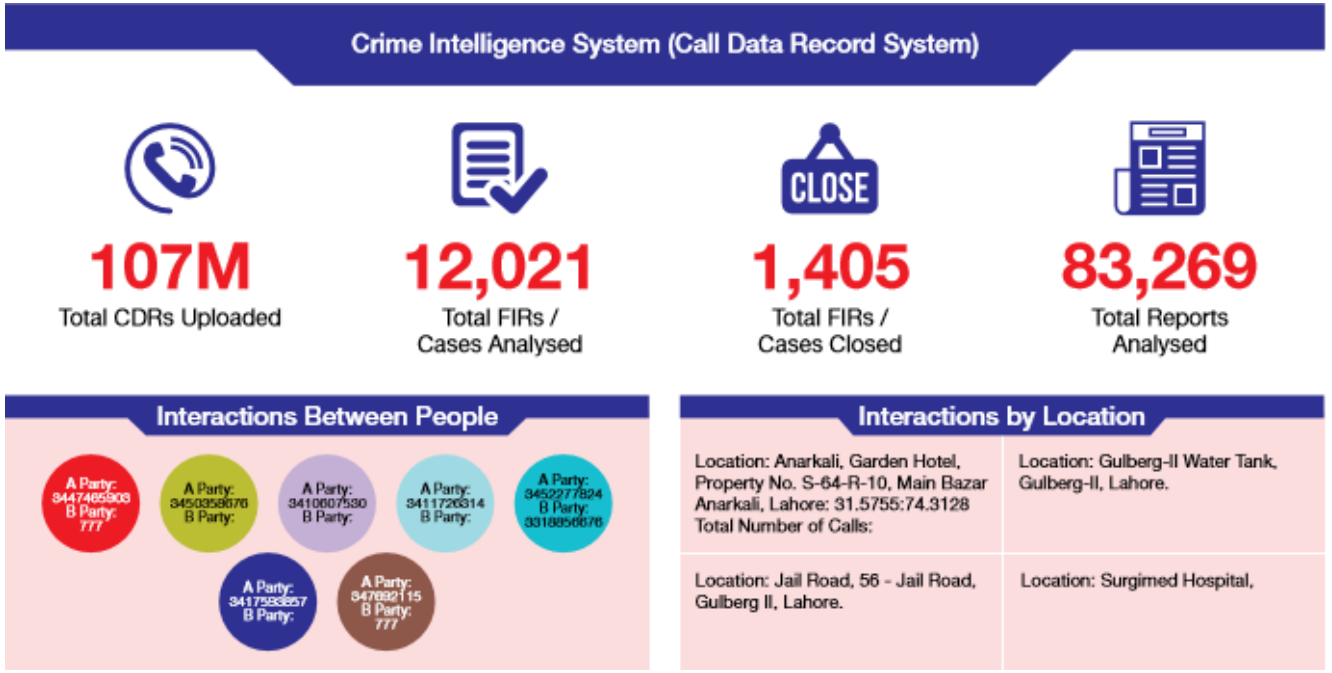
4. cyber-profiling criminals

Cyber-profiling is the process of collecting data from individuals and groups to identify significant co-relations. the idea of cyber profiling is derived from criminal profiles, which provide information on the investigation division **to classify the types of criminals who were at the crime scene.**

CLUSTERING // Use cases

5. call record detail analysis

A call detail record (cdr) is the information captured by telecom companies during the call, sms, and internet activity of a customer. this information provides **greater insights about the customer's needs when used with customer demographics. You can cluster customer activities for 24 hours** by using the unsupervised k-means clustering algorithm. it is used to understand segments of customers with respect to their usage by hours.

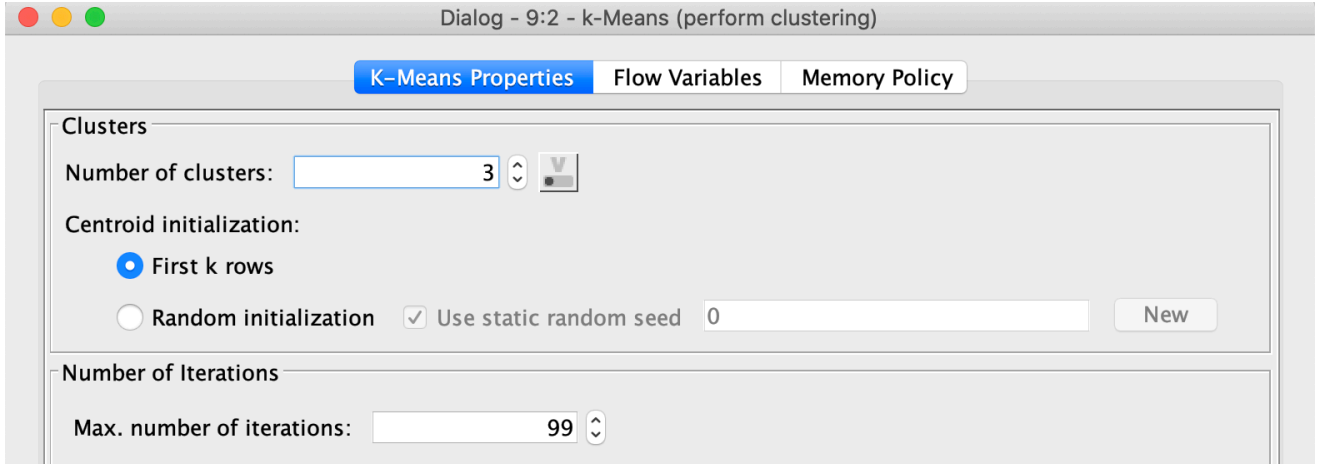


CLUSTERING // K

Number of cluster determination
The number of cluster is determined with 2 metrics :

- Silhouette scores : the higher is better.
- “Elbow Method” where we are looking for significant drops in SSE before it starts to flatten.

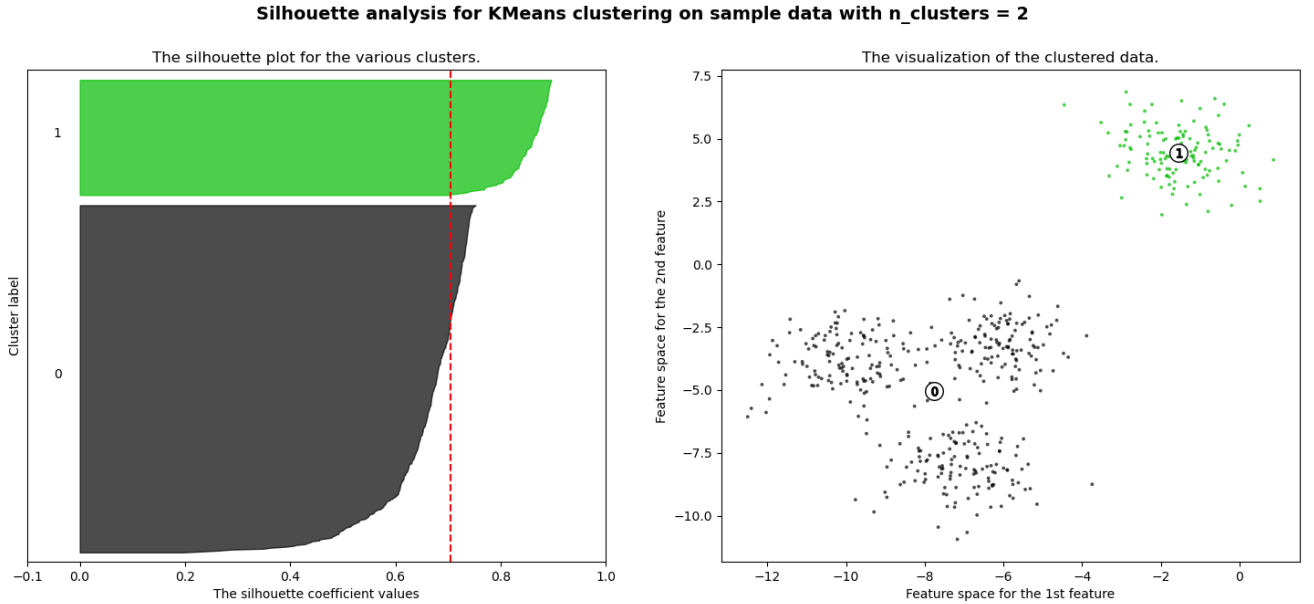
Clustering is an unsupervised machine learning algorithm that groups data into k number of clusters. **The number of clusters is user-defined and the algorithm will try to group the data even if this number is not optimal for the specific case.**



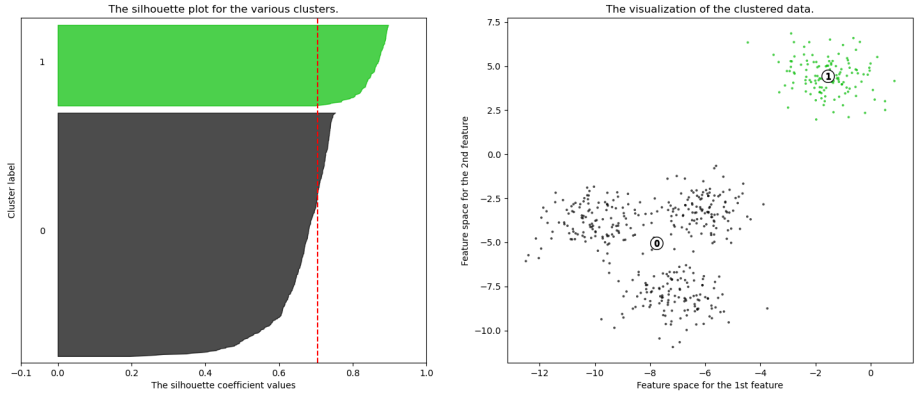
Silhoutte score

Silhouette analysis can be used t**o study the separation distance between the resulting clusters.** The silhouette plot displays a measure of how close each point in one cluster is to points in the neighboring clusters and thus provides a way to assess parameters like number of clusters visually. **This measure has a range of [-1, 1].**

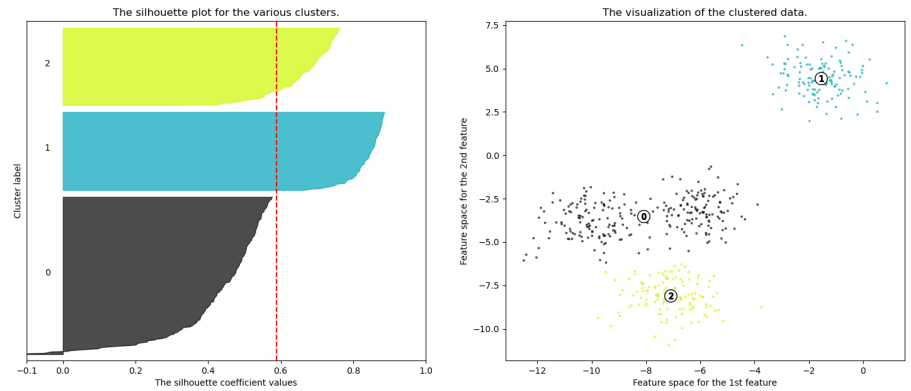
Silhouette coefficients (as these values are referred to as) near +1 indicate that the sample is far away from the neighboring clusters. A value of 0 indicates that the sample is on or very close to the decision boundary between two neighboring clusters , and negative values indicate that those samples might have been assigned to the wrong cluster.



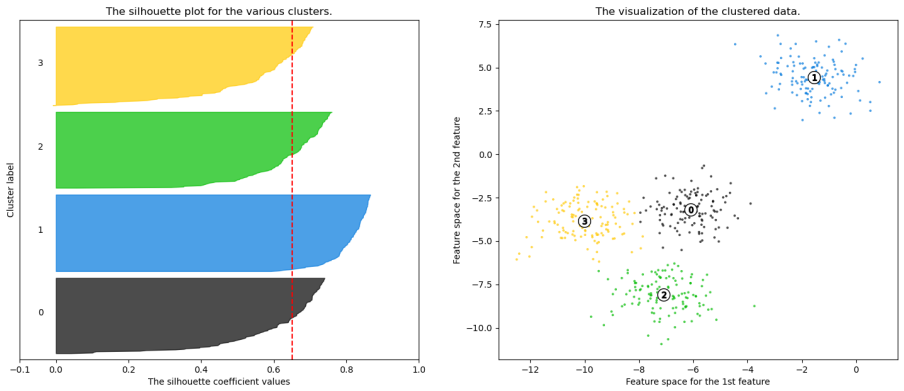
Silhouette analysis for KMeans clustering on sample data with n_clusters = 2



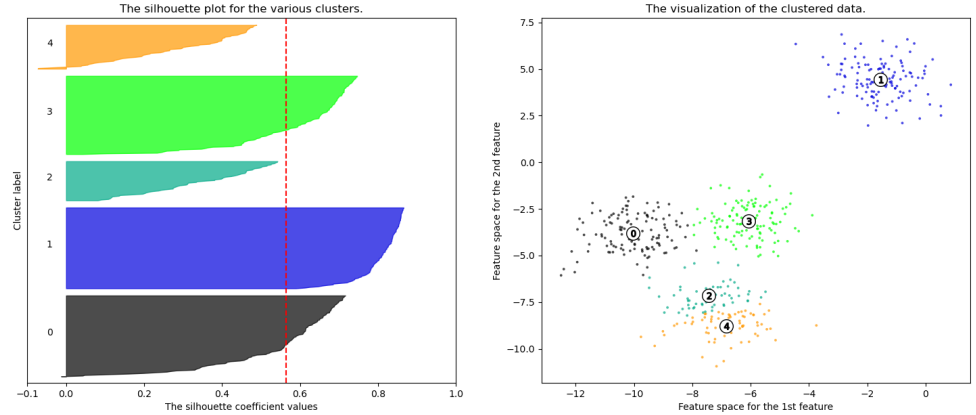
Silhouette analysis for KMeans clustering on sample data with n_clusters = 3



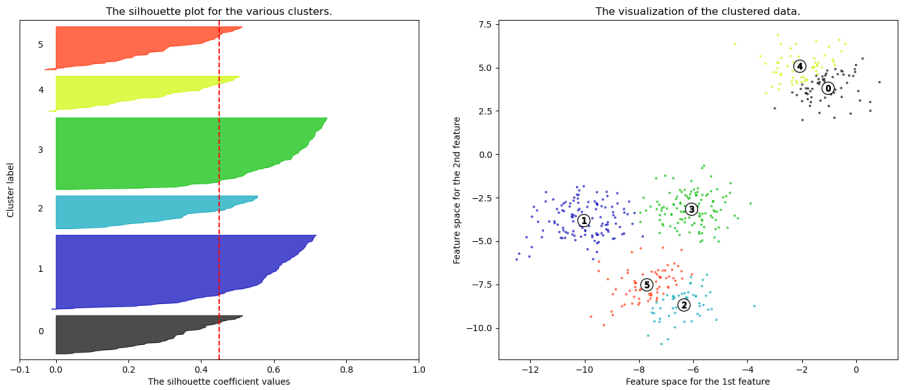
Silhouette analysis for KMeans clustering on sample data with n_clusters = 4



Silhouette analysis for KMeans clustering on sample data with n_clusters = 5



Silhouette analysis for KMeans clustering on sample data with n_clusters = 6

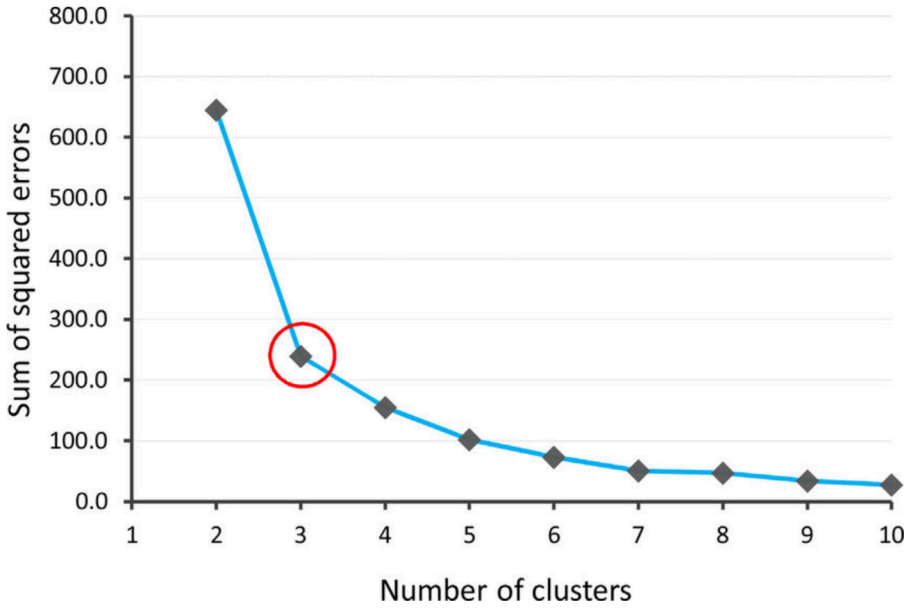


Out:

```
For n_clusters = 2 The average silhouette_score is : 0.7049787496083262
For n_clusters = 3 The average silhouette_score is : 0.5882004012129721
For n_clusters = 4 The average silhouette_score is : 0.6505186632729437
For n_clusters = 5 The average silhouette_score is : 0.56376469026194
For n_clusters = 6 The average silhouette_score is : 0.4504666294372765
```

The Elbow Method

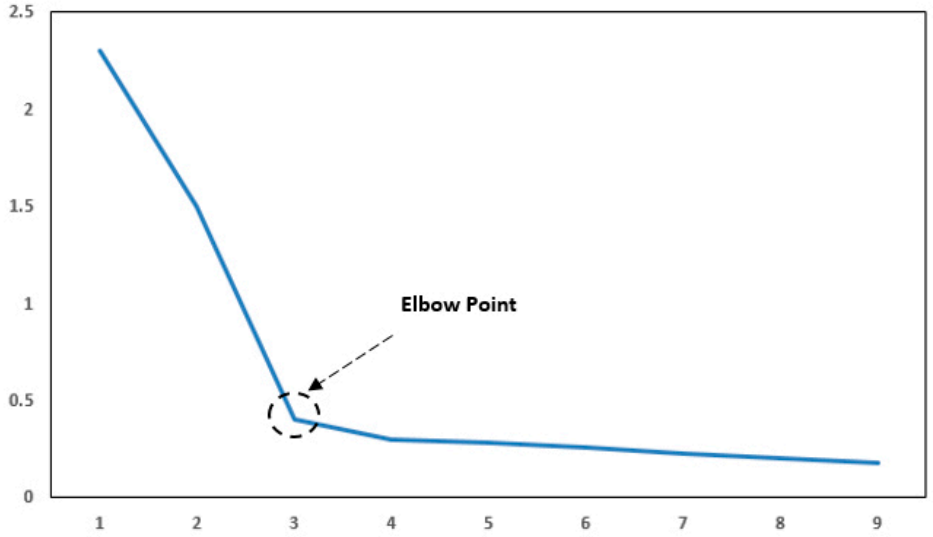
Therefore we have to come up with a technique that somehow **will help us decide how many clusters we should use for the K-Means model.**
The Elbow method is a very popular technique and the idea is to run k-means clustering for a range of clusters k (let's say from 1 to 10) and for each value, we are calculating the sum of squared distances from each point to its assigned center(distortions).

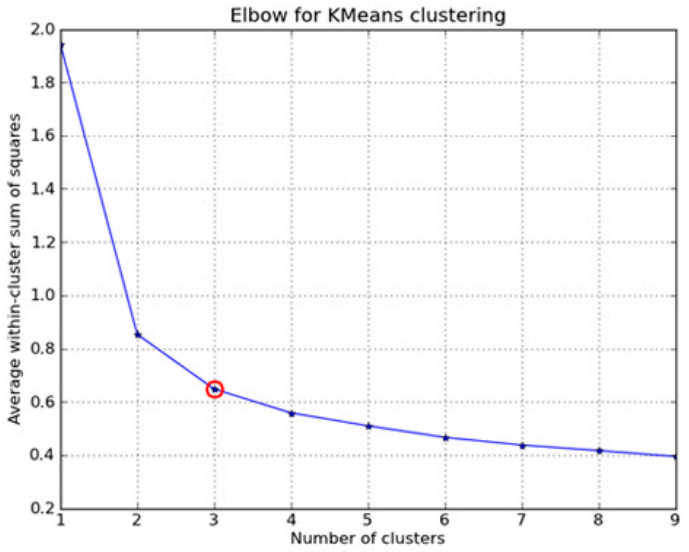


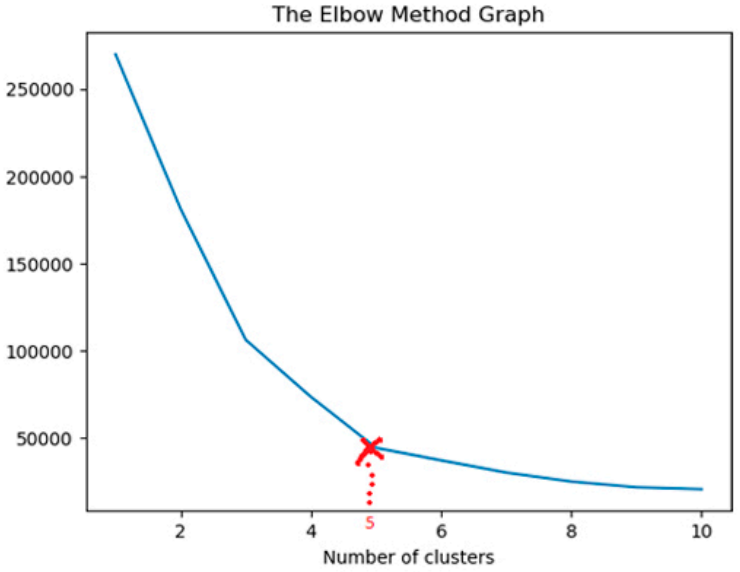
Result of the elbow method to determine optimum number of clusters.

The Elbow Method

When the distortions are plotted and the plot looks like an arm then the **“elbow”(the point of inflection on the curve) is the best value of k.**

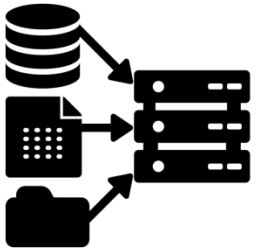






CLUSTERING // K

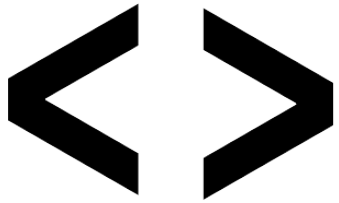
Google Colab The Elbow Method & Silhoutte score



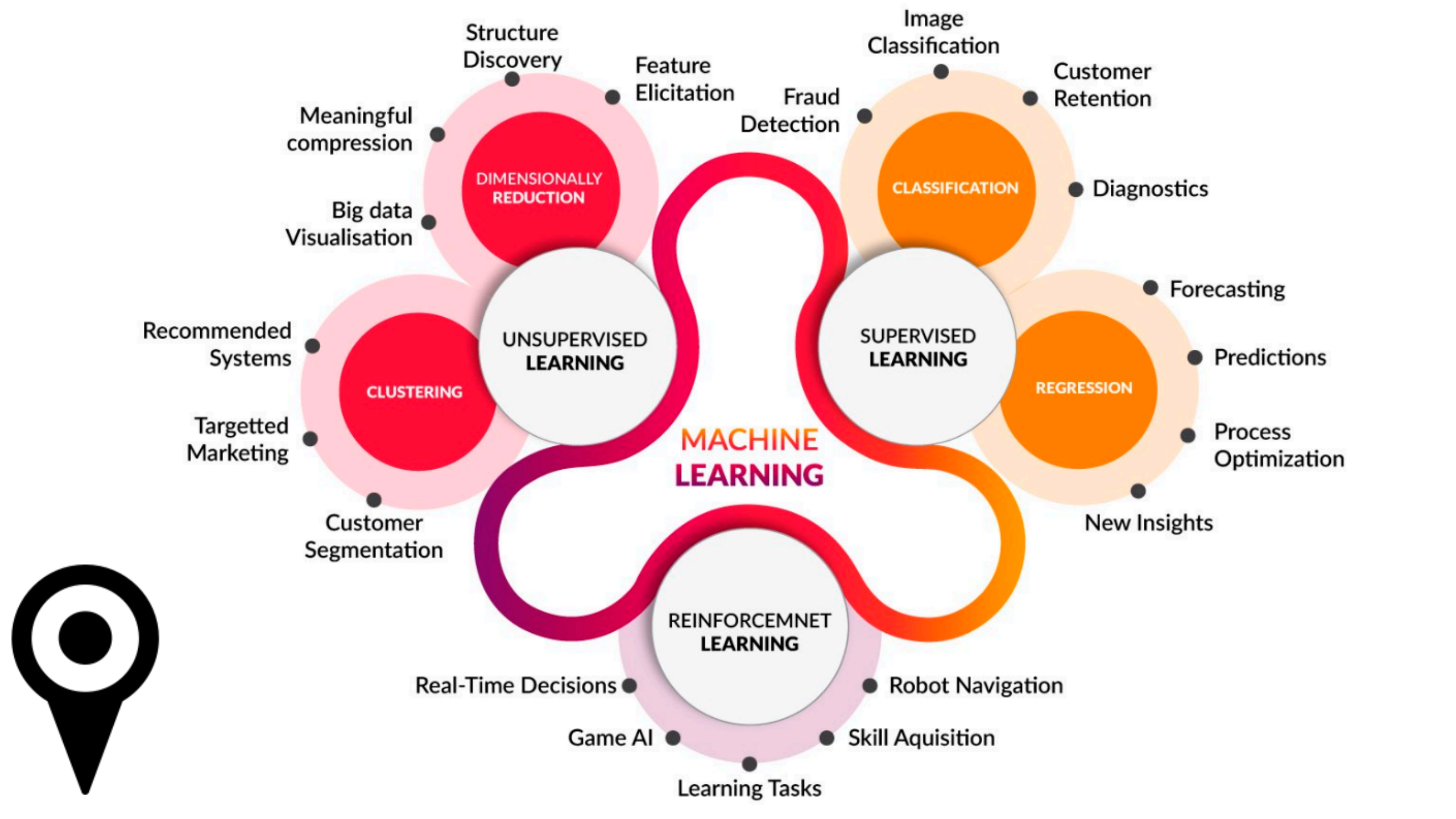
Credit card clean.csv

CLUSTERING // K

Google Colab The Elbow Method & Silhoutte score

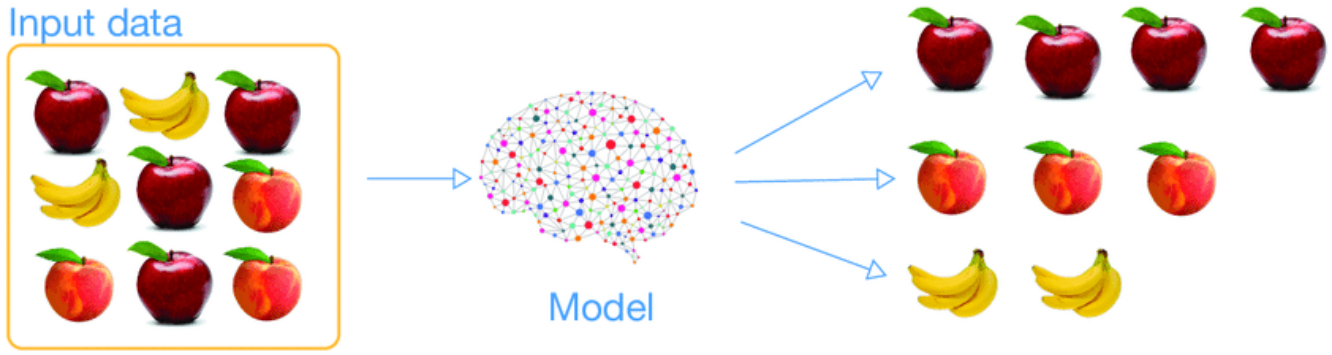


Credit_card_k.pynb


MACHINE LEARNING
UNSUPERVISED LEARNING
SUPERVISED LEARNING
REINFORCEMNET LEARNING
DIMENSIONALLY REDUCTION
CLUSTERING
CLASSIFICATION
REGRESSION
Structure Discovery
Feature Elicitation
Meaningful compression
Big data Visualisation
Image Classification
Customer Retention
Fraud Detection
Diagnostics
Recommended Systems
Targetted Marketing
Customer Segmentation
Forecasting
Predictions
Process Optimization
New Insights
Real-Time Decisions
Robot Navigation
Game AI
Skill Aquisition
Learning Tasks

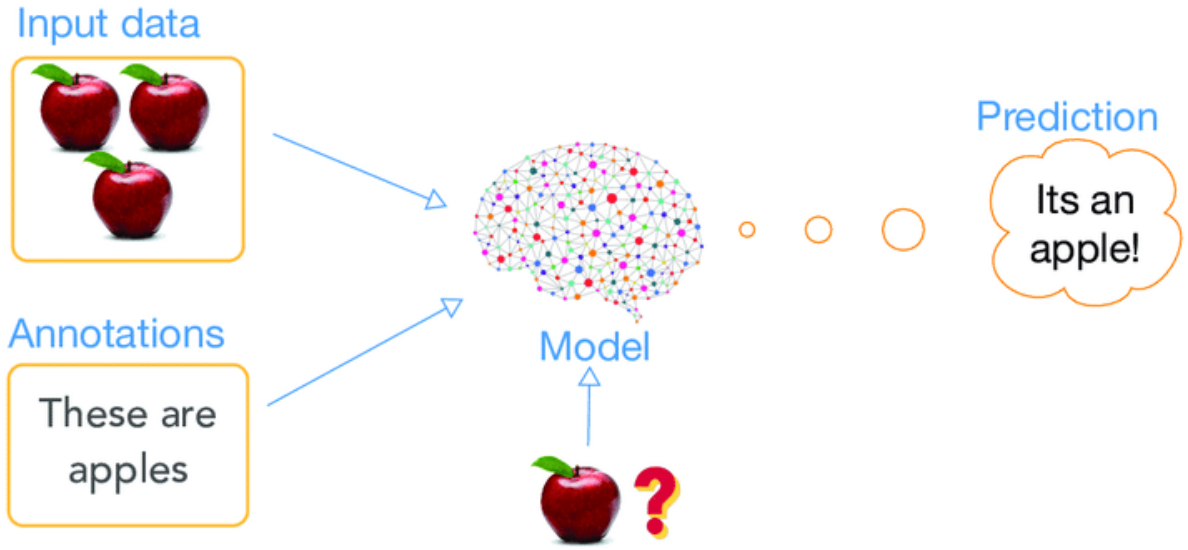
INTRO

Non labeled data



Unsupervised learning

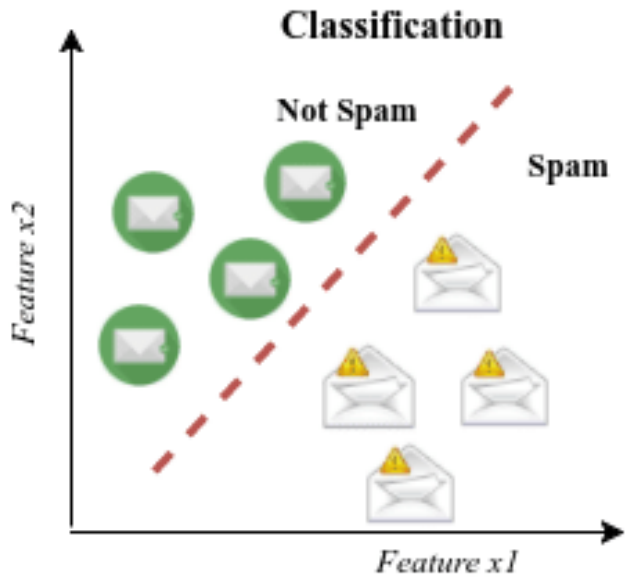
Labeled data

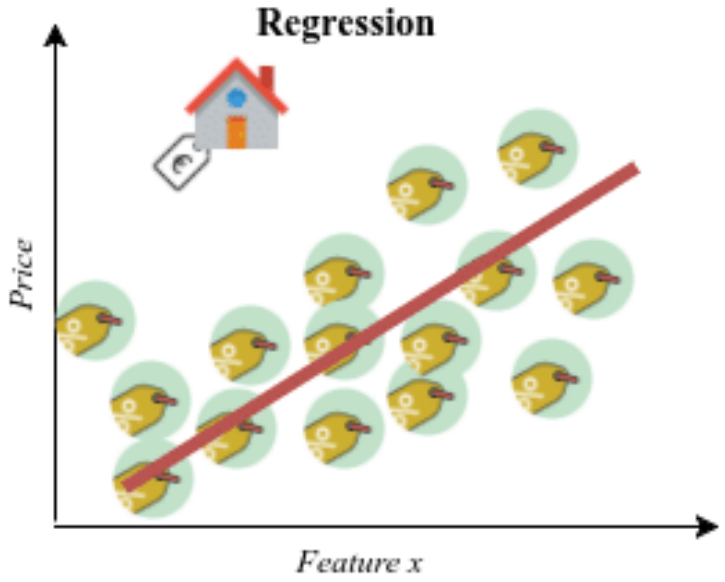


Supervised learning

INTRO

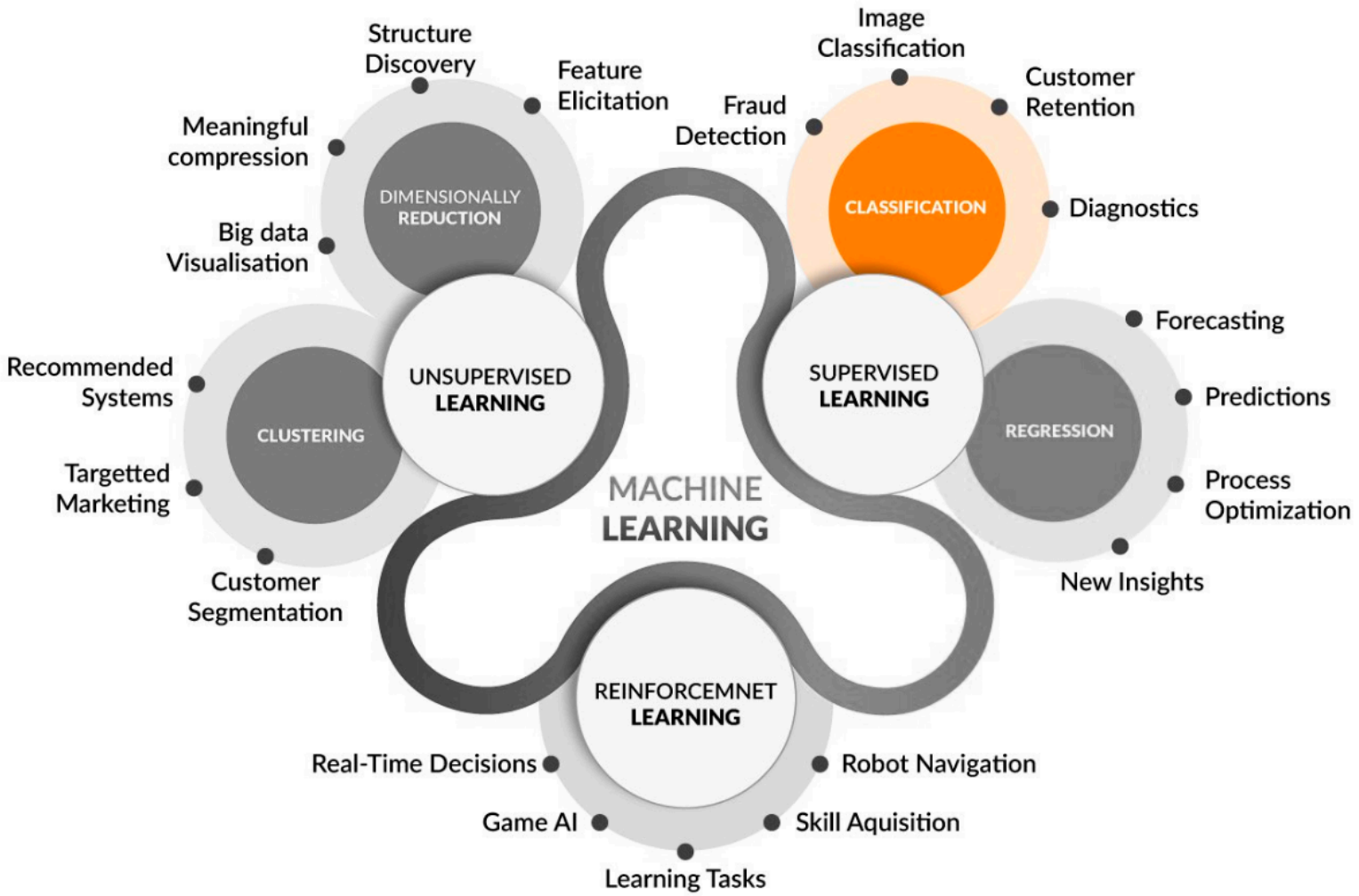
What the algos **can do** for us? **classify / predict / group**





Supervised

ML // SUPERVISED



Supervised Learning

1 // Input data is called training data and **has a known label or result** (such as spam/not-spam or a stock price at a time.)

2 // A model is prepared through a **training process** in which it is required to make predictions and is corrected when those predictions are wrong.

3 // The training process continues until the model achieves a **desired level of accuracy** on the testing data.

Example problems are **classification and regression.**
Example algorithms include: **Logistic Regression and the Back Propagation Neural Network**.

ML // SUPERVISED // classification

Logistic regression

123 category

Logistic regression

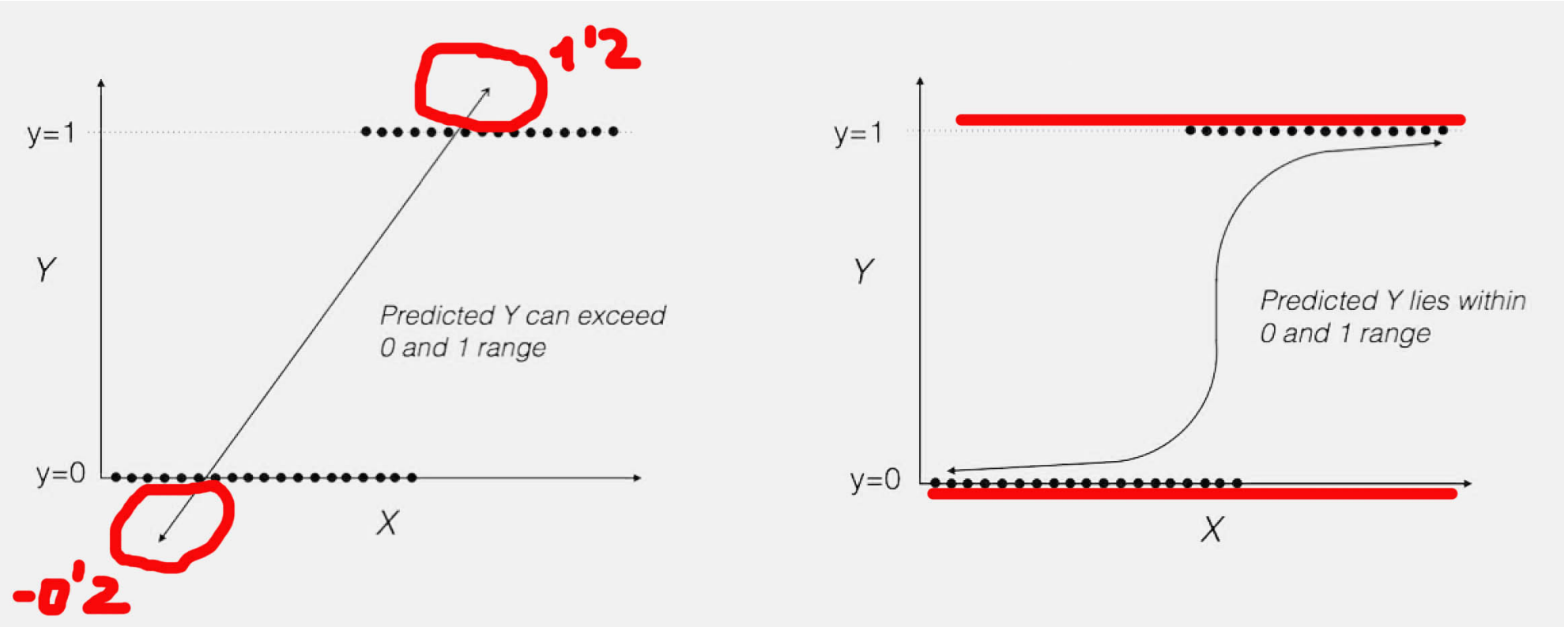
123 category

Logistic regression is used for **classification problems** when the dependant/ target variable is **binary.** That is, its values are **true or false**.
Logistic regression is one of the most popular and widely used algorithms in practice

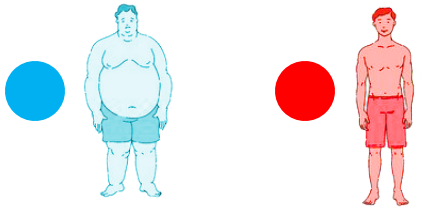
Logistic regression is an algorithm that is used in **solving classification problems.**
It is a predictive analysis that describes data and **explains the relationship between variables.**
Logistic regression is applied to an input variable (X) where the output variable (y) is a discrete value which ranges between **1 (yes) and 0 (no).**

Logistic Regression: why?

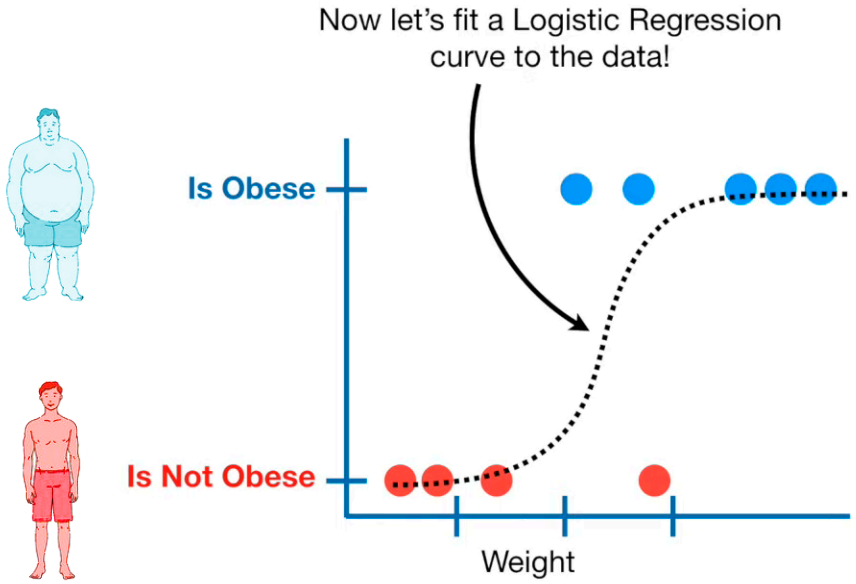
123 category

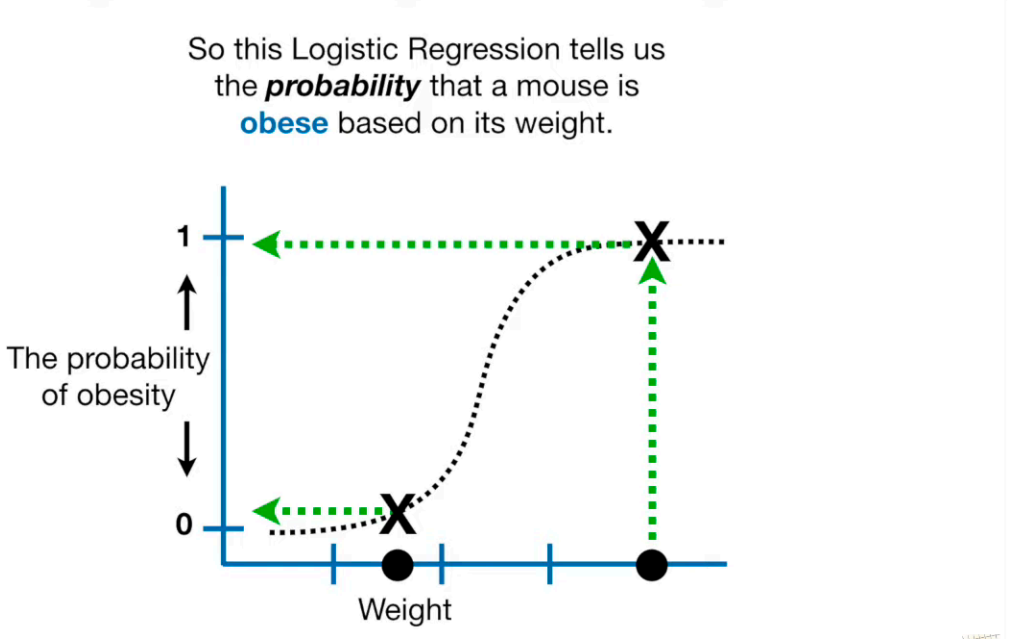


The response target variable **y** of the **Linear regression** model is not restricted within the **[0,1] interval.**

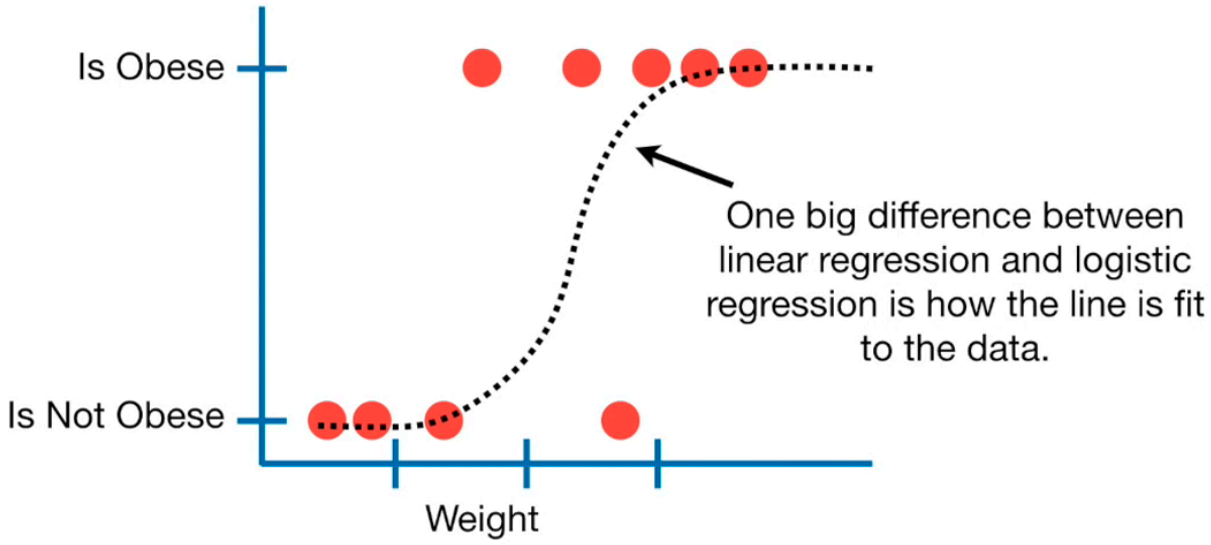


Logistic regression: train & test

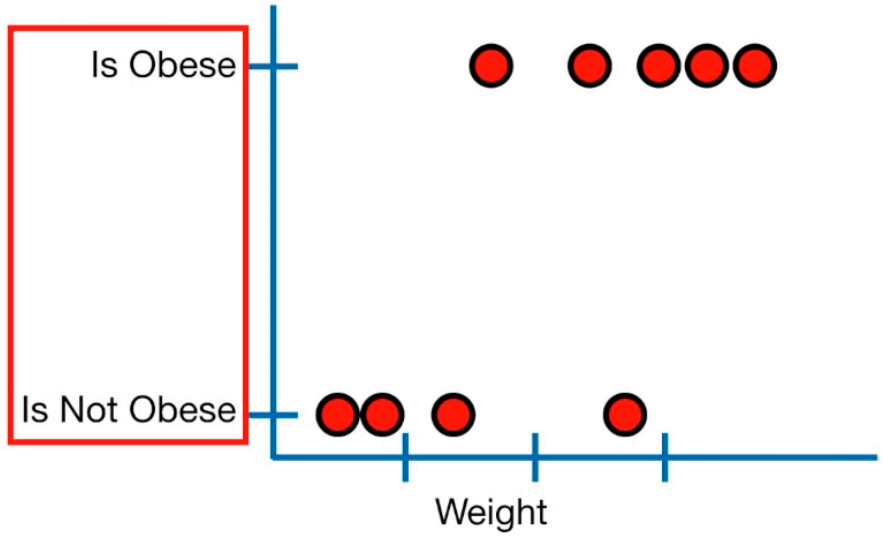




Logistic regression

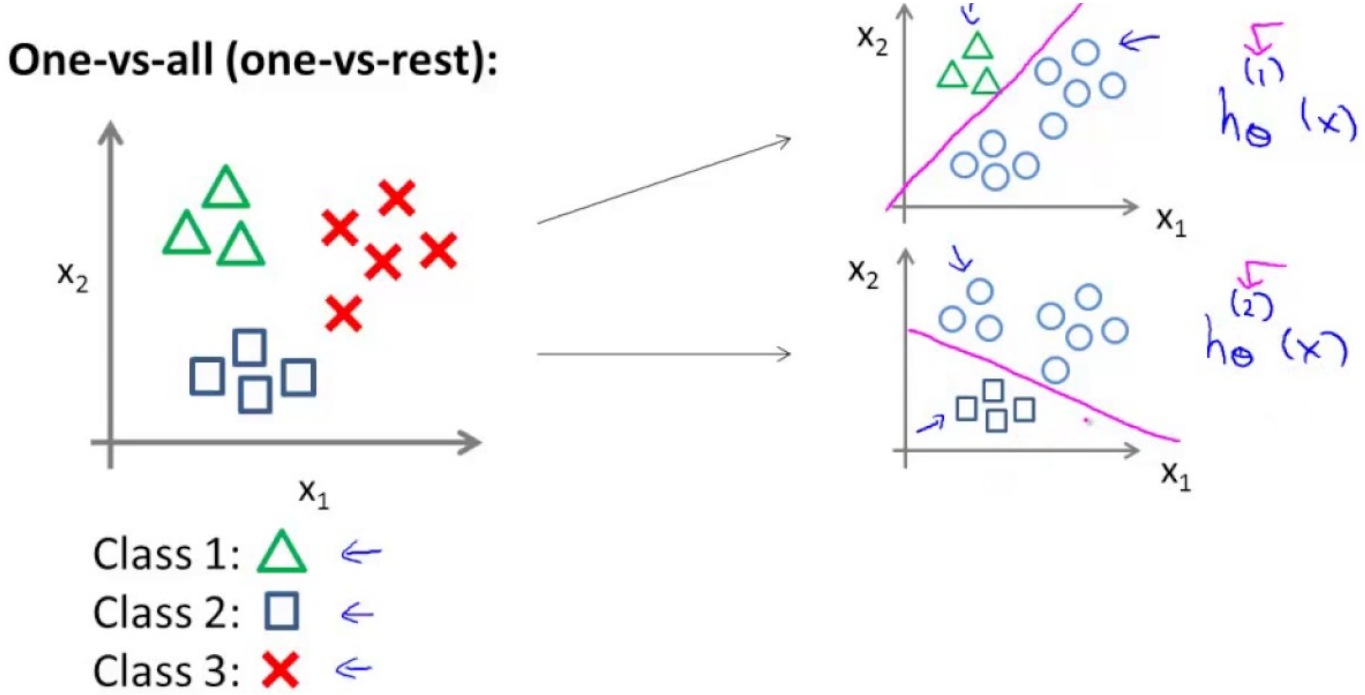


Logistic regression predicts whether something is ***True*** or ***False***, instead of predicting something continuous like ***size***.

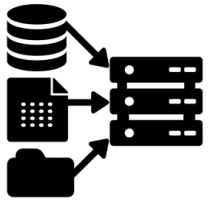


Logistic regression

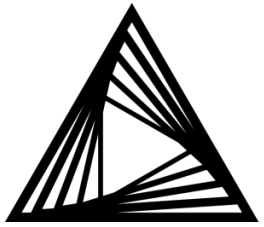
Also could be used as Multiclass (one-vs-all)



ML // SUPERVISED // classification

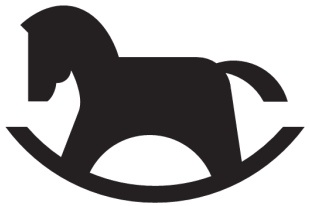


Wine.csv



Wine logistic regression

Use case Logistic Regression



Playground Logistic Regression

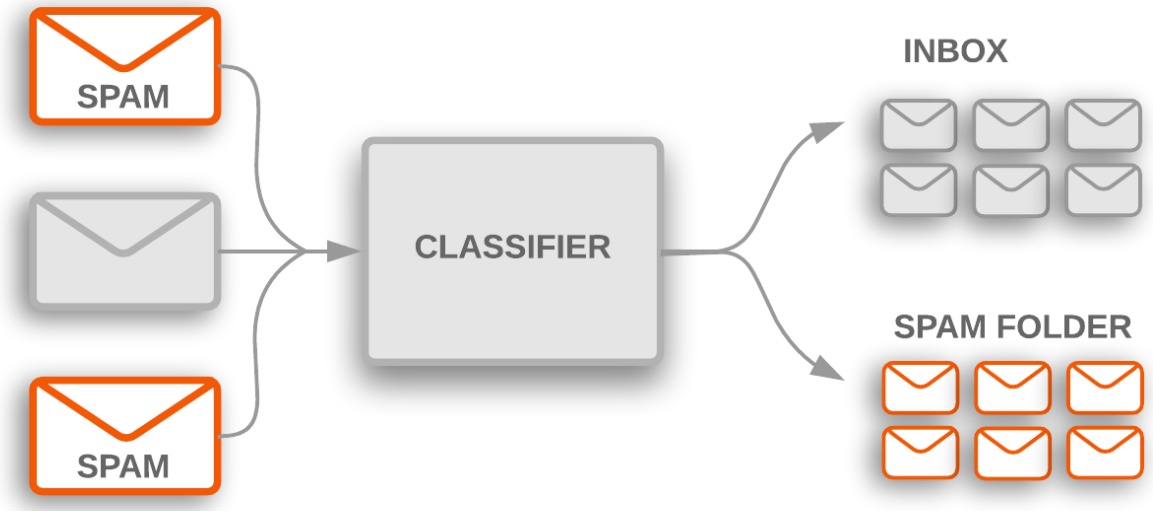
Use case Logistic Regression

Logistic Regression Example: Spam Detection

Spam detection is a binary classification problem where we are given an email and we need to classify whether or not it is spam. If the email is spam, we label it 1; if it is not spam, we label it 0. In order to apply Logistic Regression to the spam detection problem, **the following features of the email are extracted:**

- Sender of the email
- Number of typos in the email
- Occurrence of words/phrases like “offer”, “prize”, “free gift”, etc.

The resulting feature vector is then used to train a Logistic classifier which emits a score in the range 0 to 1. If the score is more than 0.5, we label the email as spam. Otherwise, we don’t label it as spam.

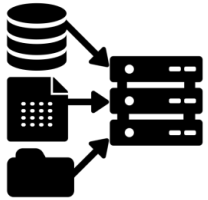


Logistic Regression: **fraud detection**

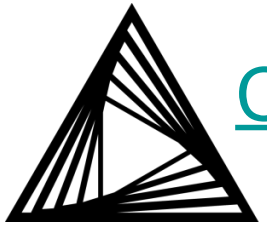
Logistic Regression: **credit card transactions**

This dataset presents **transactions that occurred in two days**, where we have 492 frauds out of 284,807 transactions. The dataset is highly unbalanced, the positive class (frauds) account for 0.172% of all transactions.
It contains only **numerical input variables which are the result of a PCA transformation**. Unfortunately, due to confidentiality issues, we cannot provide the original features and more background information about the data.
Features V1, V2, ... V28 are the principal components obtained with PCA, the only features which have not been transformed with PCA are 'Time' and 'Amount'.
Feature 'Time' contains the seconds elapsed between each transaction and the first transaction in the dataset.
The feature 'Amount' is the transaction Amount, this feature can be used for example-dependant cost-senstive learning.
Feature 'Class' is the response variable and it takes value 1 in case of fraud and 0 otherwise.

Logistic Regression: **credit card transactions**



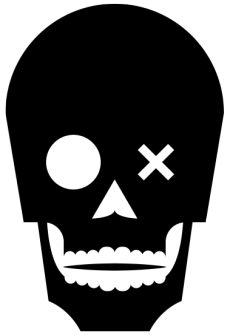
Creditcard.csv

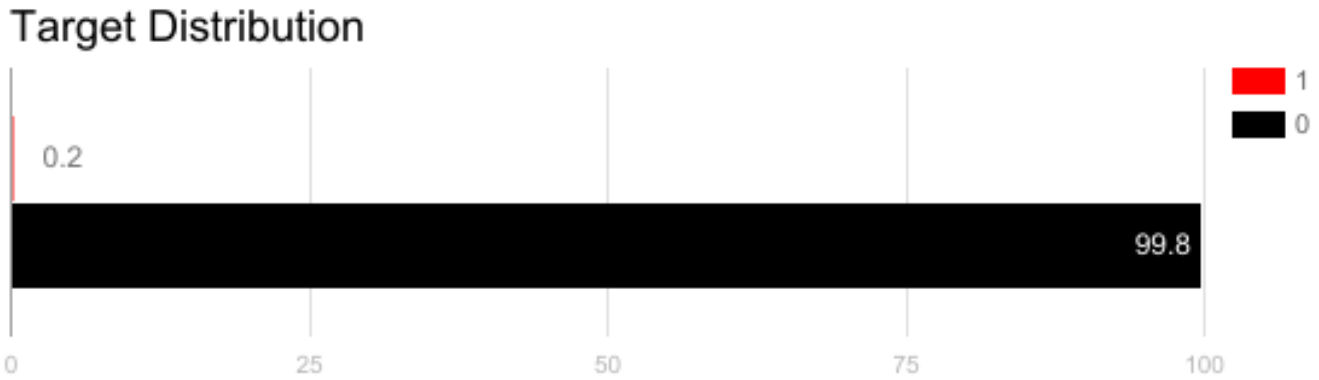


Credit Card Logistic regression

Logistic Regression: **Unbalanced dataset**

Most machine learning algorithms do not work very well with imbalanced datasets. The following seven techniques can help you, to train a classifier to detect the abnormal class.





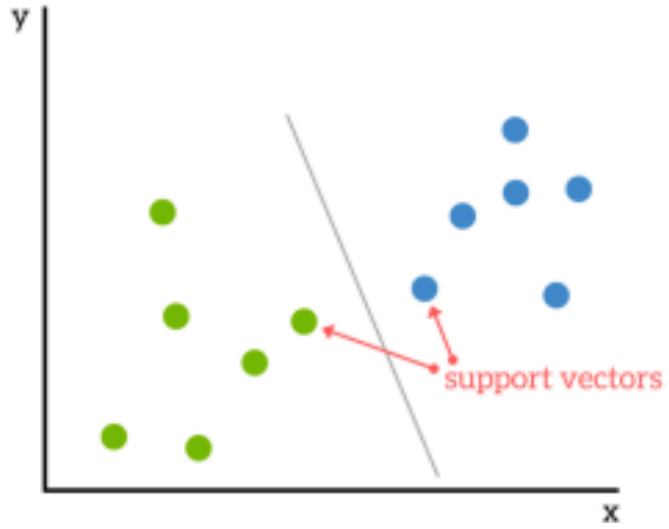
1. Use the right evaluation metrics: AUC , ROC, Confusion matrix
2. Resample the training set: **under-sampling and over-sampling**
3. **Use K-fold Cross-Validation** in the right way
4. Ensemble different resampled **datasets**.

Support Vector Machines (SVM)

Support Vector Machines (SVM) 123

SVM´s are based on the idea of finding a hyperplane that best divides a dataset **into two classes.**

Support vectors are the data points nearest to the hyperplane, the points of a data set that, if removed, would alter the position of the dividing hyperplane. Because of this, **they can be considered the critical elements of a data set.**



Support Vector Machines (SVM) 123

The support vector machine is a model used for both classification and regression problems though it is mostly used to solve classification problems.

The algorithm creates a **hyperplane or line(decision boundary) which separates data into classes**. It uses the kernel trick to find the best line separator (decision boundary that has same distance from the boundary point of both classes).

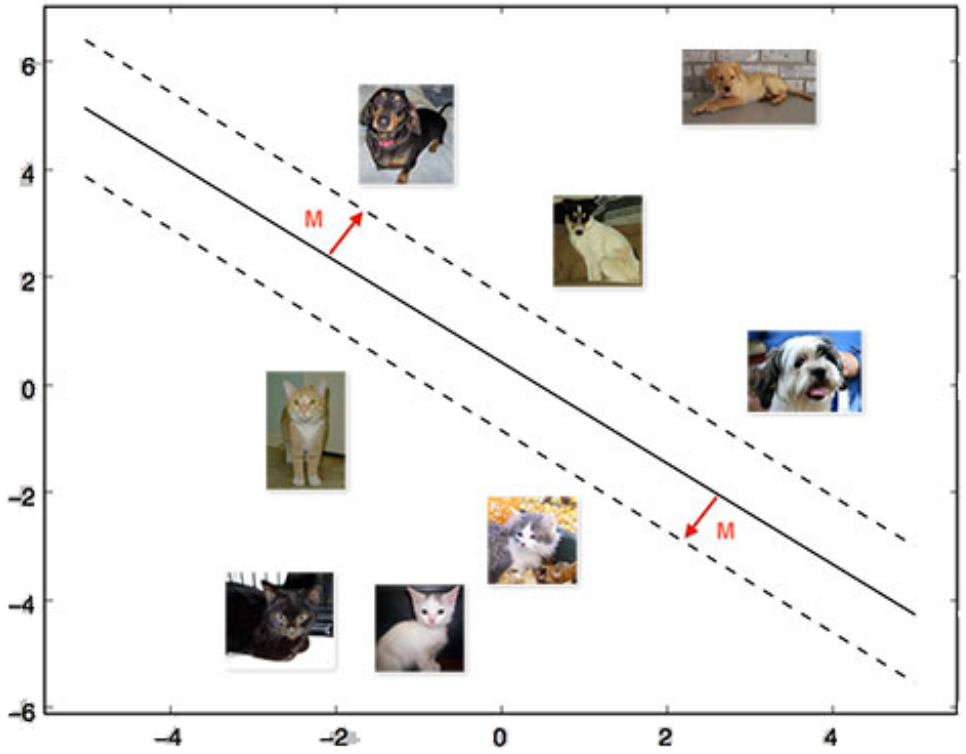
It is a clear and more powerful way of learning complex **non linear functions.**

The aim of using SVM is to correctly classify unseen data. SVMs have a number of applications in several fields.

Some common applications of SVM are:
-Binary Classification
-Multi-class Classification
-Regression

Support Vector Machines (SVM)

123



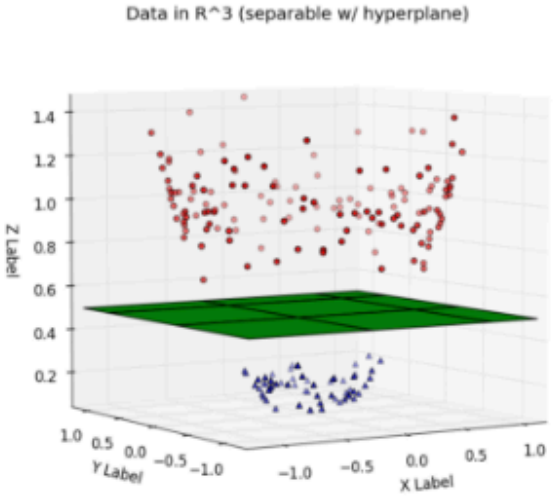
Support Vector Machines (SVM)

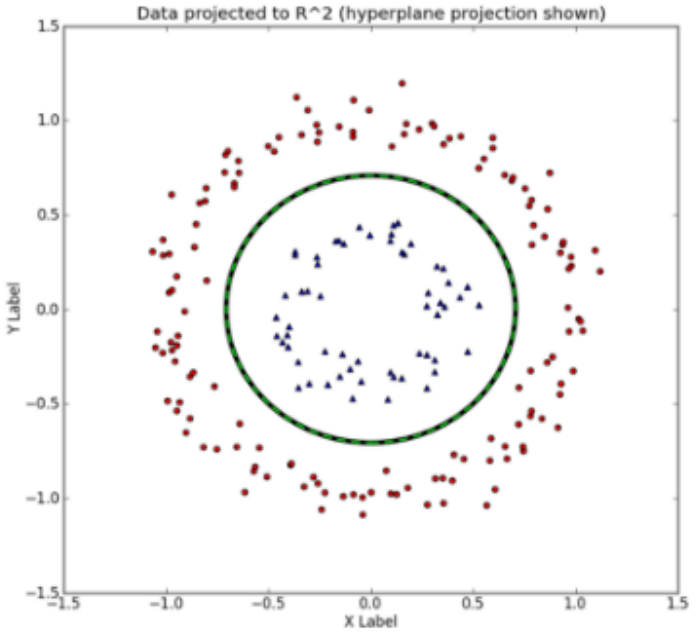
123

Data Points

Support Vector Machines (SVM)

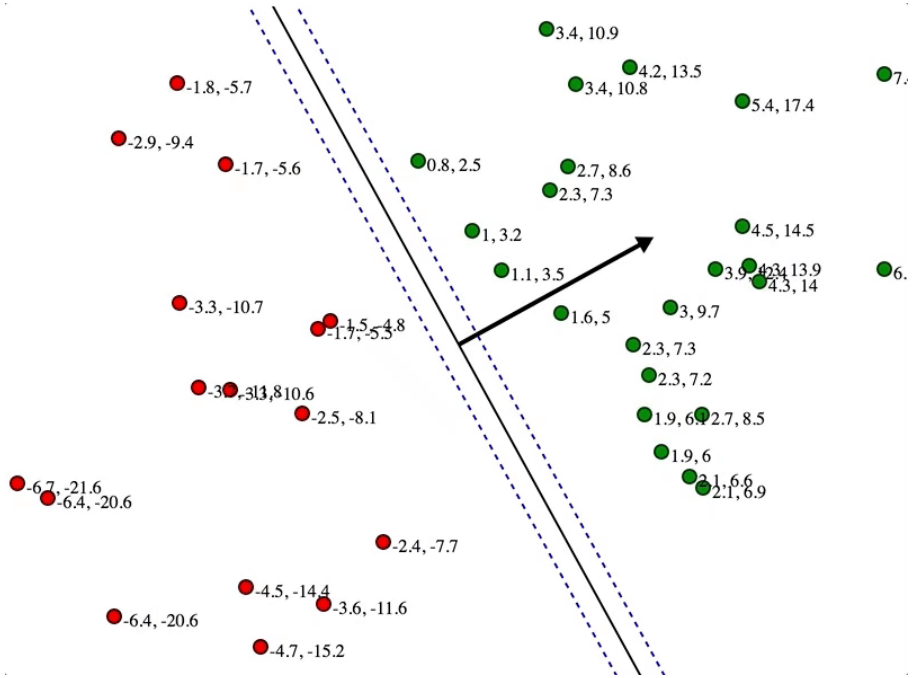
123

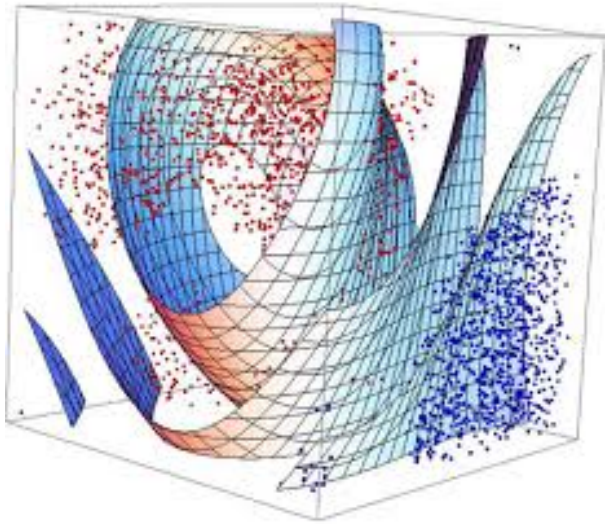




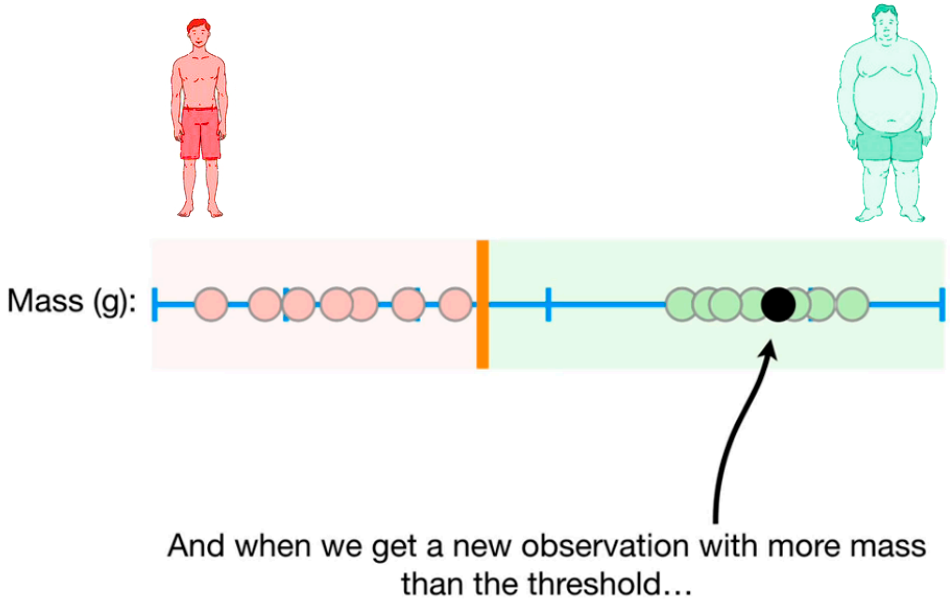
Support Vector Machines (SVM)

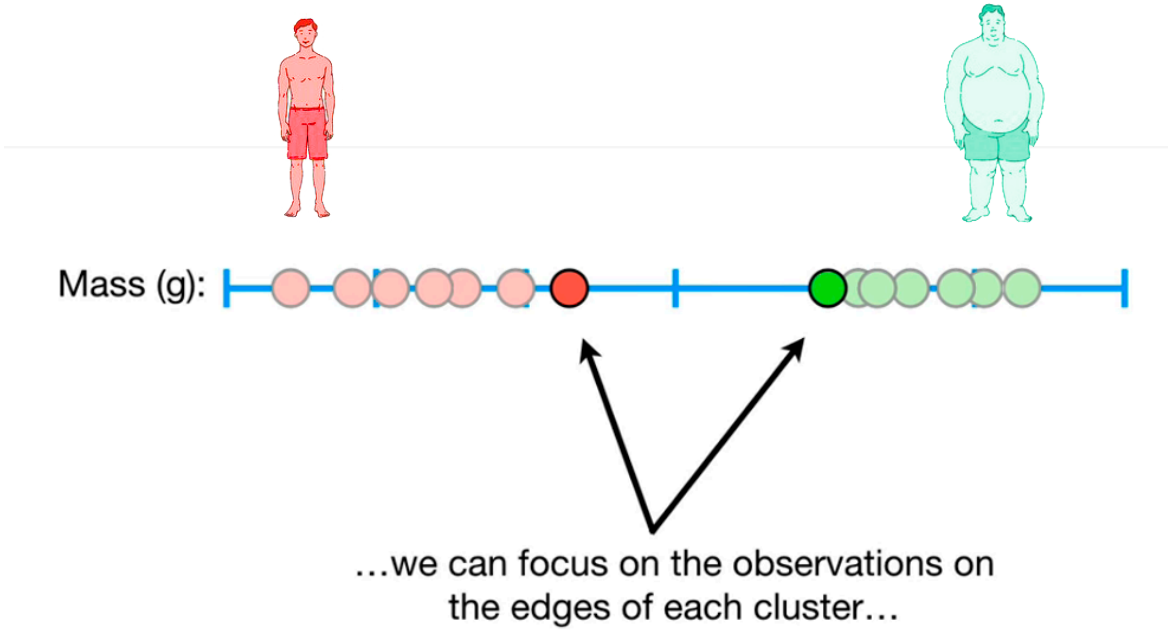
123



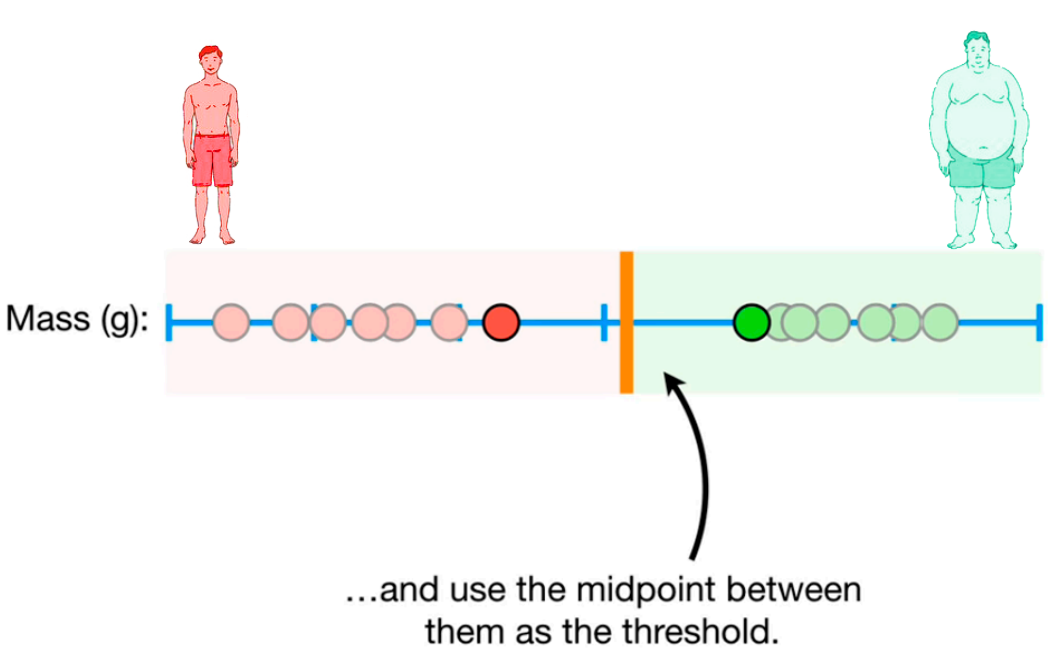


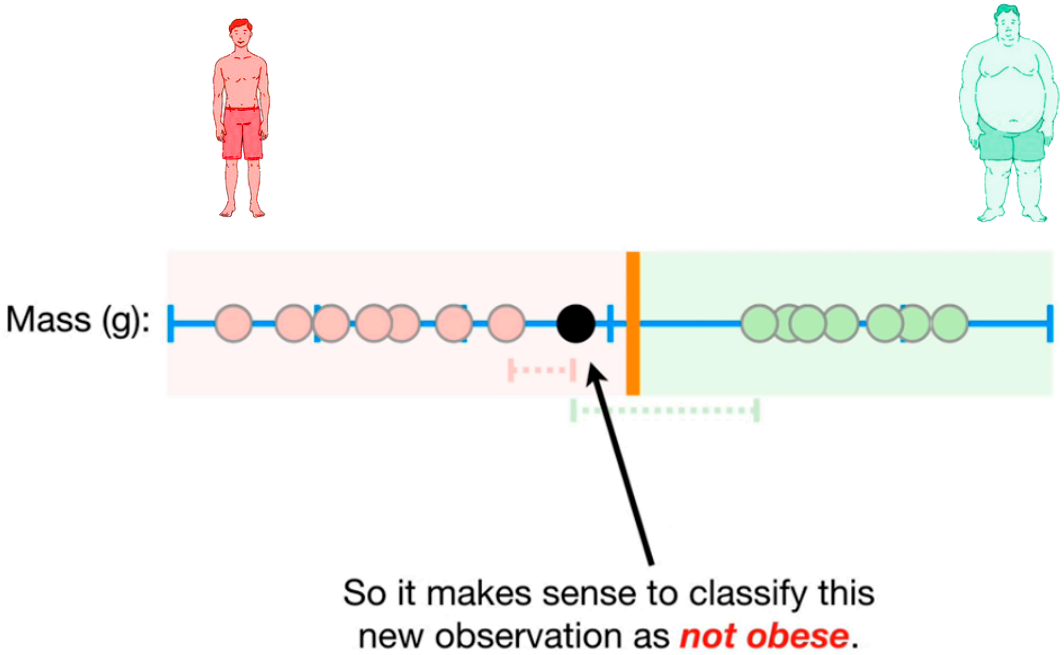
Support Vector Machines (SVM)



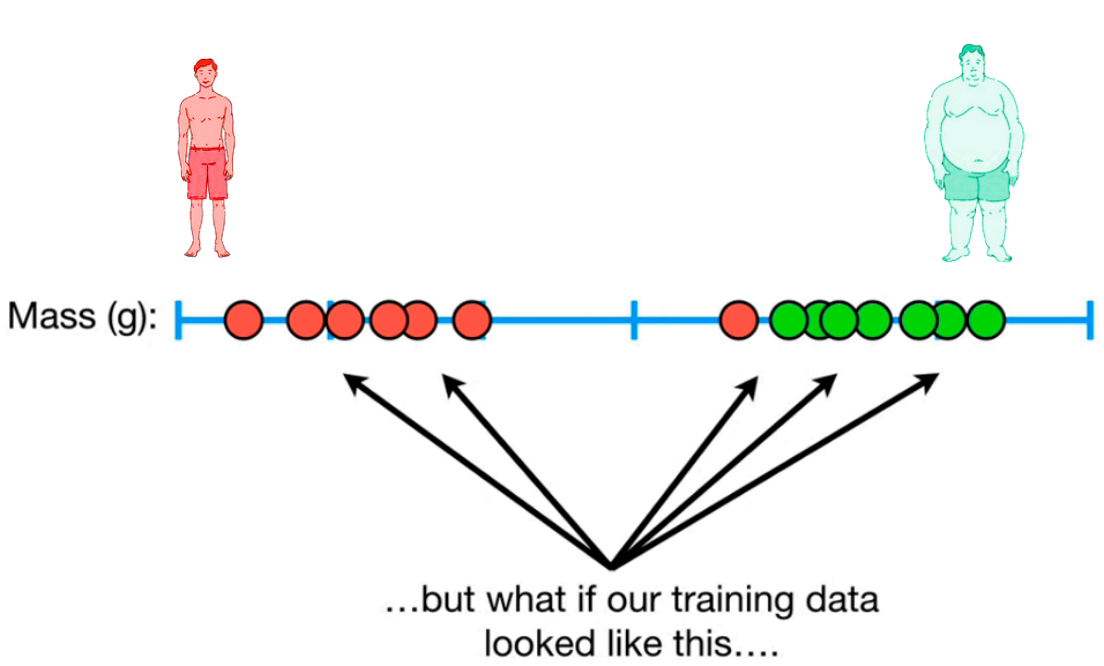


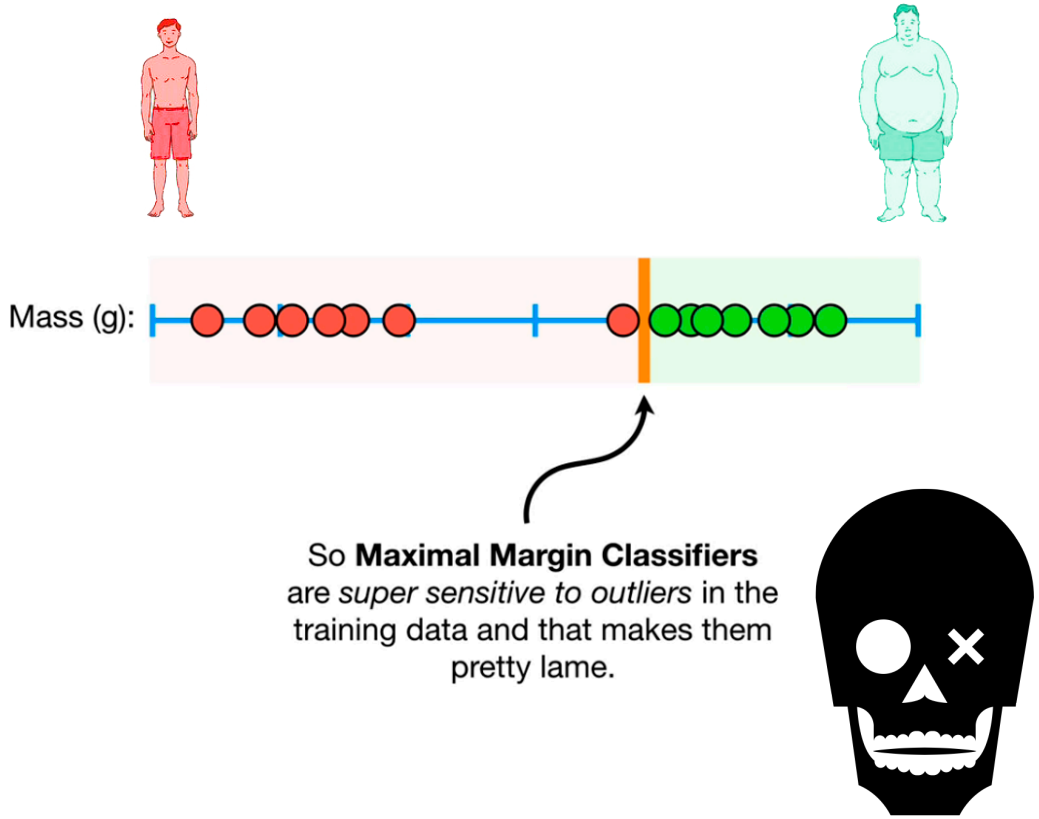
Support Vector Machines (SVM)



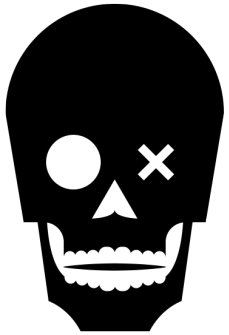


Support Vector Machines (SVM)



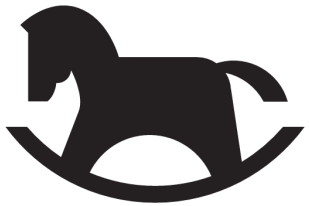


Support Vector Machines (SVM) 123



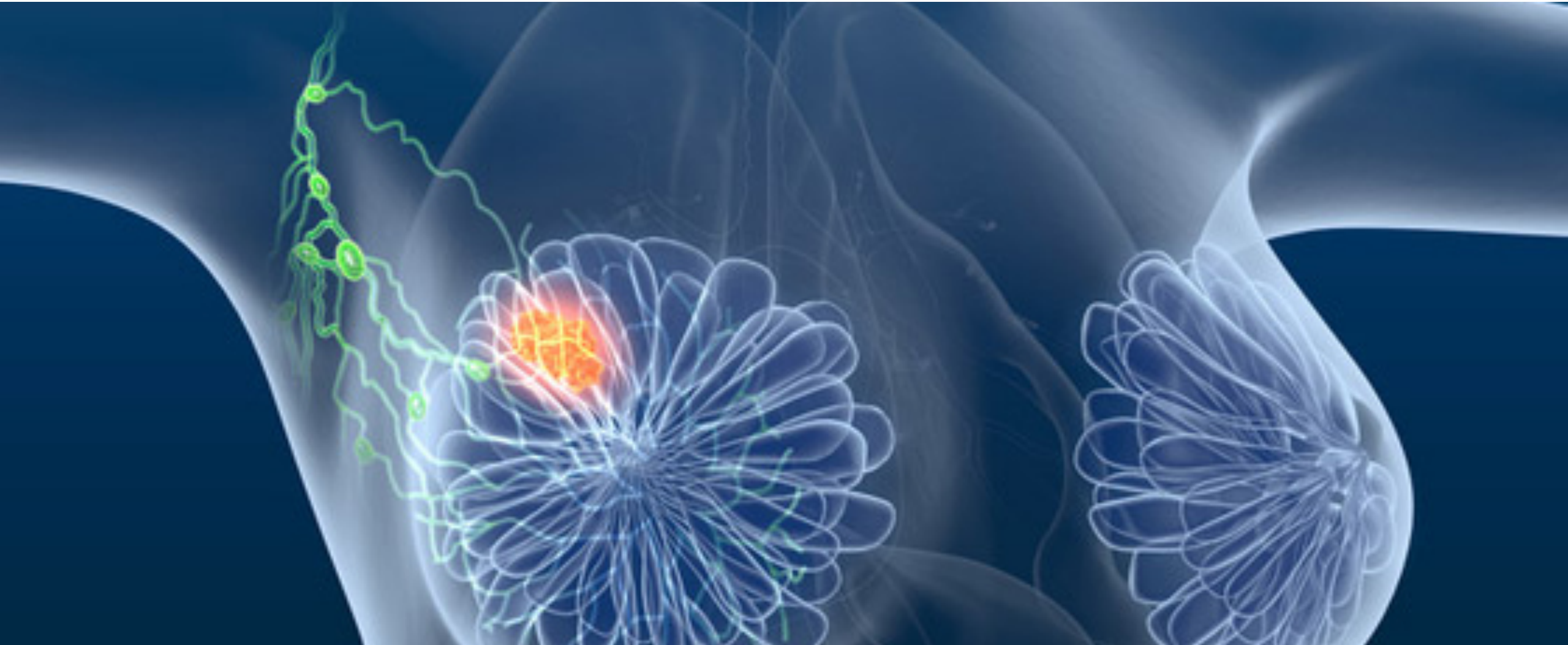
-SVM will **only accept numerical inputs**, for which it will use to train its model.
-**The numerical data must be normalized**, since numerical data with a wider range will be given higher importance than data with a narrower range.
-Use first **Cross Validation**
-**Missing values need to be filled**, in order for the SVM classifier to work.
-SVM **can also forecast** (like linear regression)
-Remove the **outliers**

Use case SVM



Playground SVM

SVM: **breast cancer**



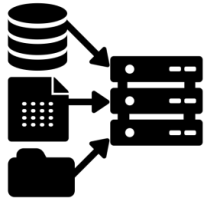
SVM: **breast cancer**

1) ID number
2) **Diagnosis (M = malignant, B = benign)** 3-32)

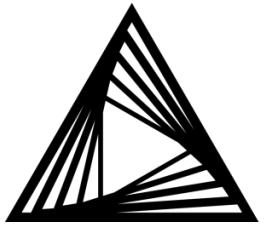
Ten real-valued features are computed for each cell nucleus:

a) **radius** (mean of distances from center to points on the perimeter)
b) **texture** (standard deviation of gray-scale values)
c) **perimeter**
d) **area**
e) **smoothness** (local variation in radius lengths)
f) **compactness** (perimeter^2 / area - 1.0)
g) **concavity** (severity of concave portions of the contour)
h) **concave points (**number of concave portions of the contour)
i) **symmetry**
j) **fractal dimension.**

SVM: **breast cancer**



breastcancer



SVM Breast Cancer

ML // SUPERVISED // classification

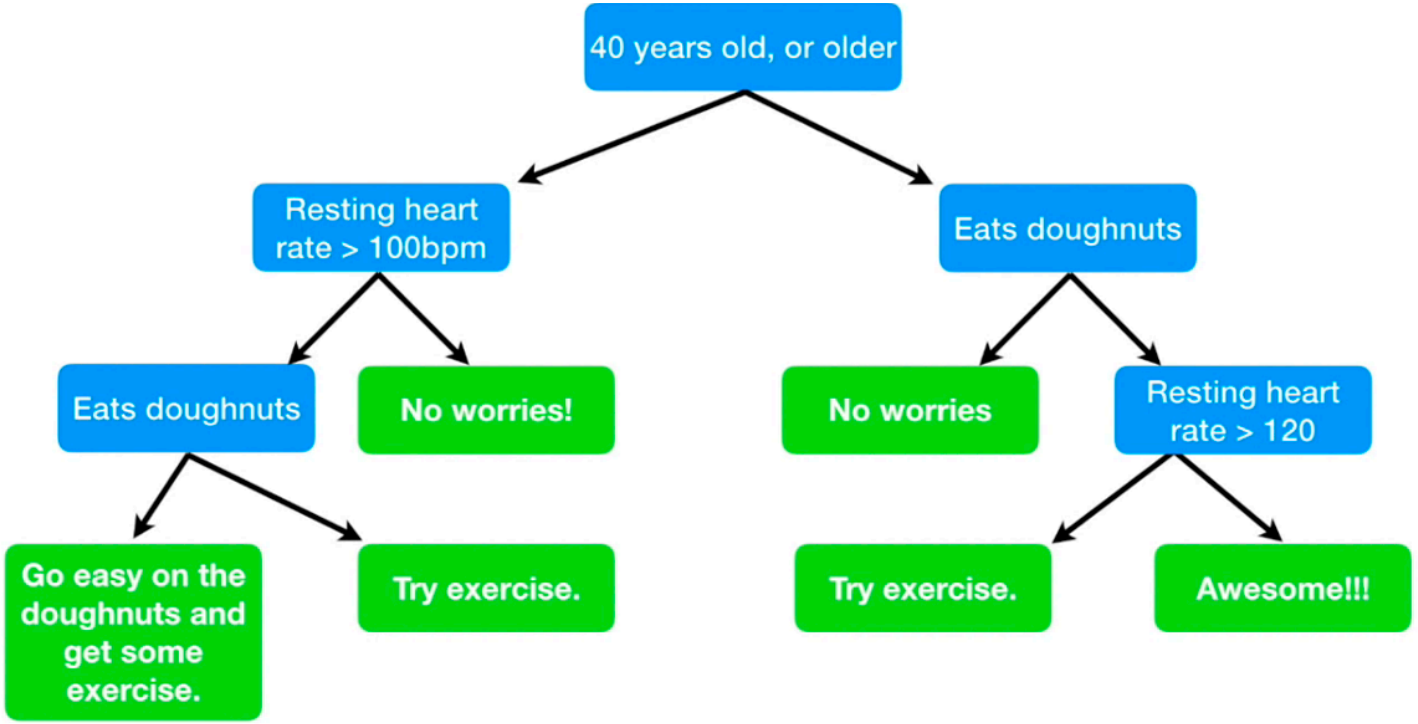
Decision trees

123 category

Decision trees

123 category

"*A simple way to visualize a decision*"



Decision trees

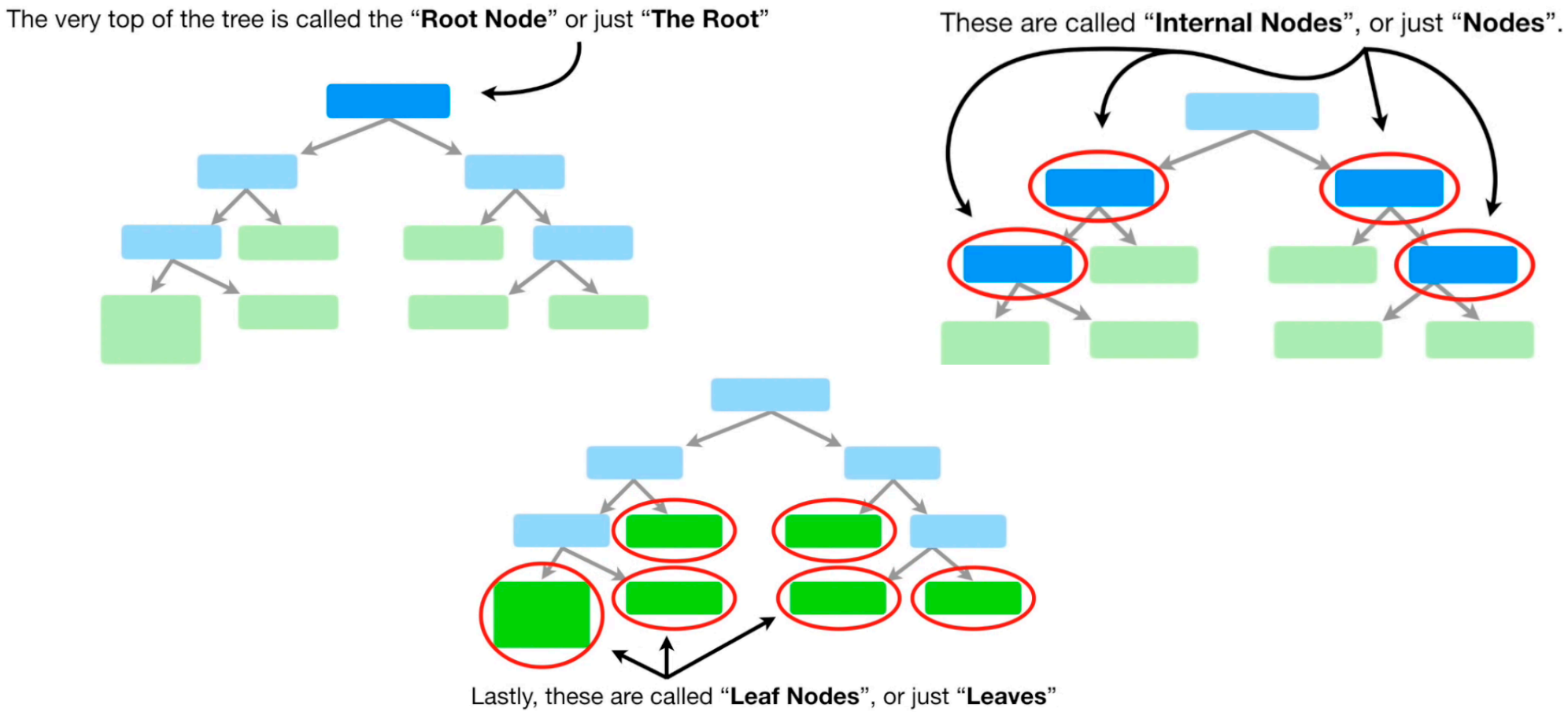
123 category

Types of Decision Trees
Types of decision tree is based on the type of target variable we have. It can be of two types:

Categorical Variable Decision Tree: Decision Tree which has categorical target variable then it called as categorical variable decision tree. E.g.:- In above scenario of student problem, where the target variable was “Student will play cricket or not” i.e. ***YES or NO.***
Continuous Variable Decision Tree: Decision Tree has **continuous target variable** then it is called as Continuous Variable Decision Tree.

E.g.:- Let’s say we have a problem to predict whether a customer will pay his renewal premium with an insurance company (yes/ no). Here we know that income of customer is a significant variable but insurance company does not have income details for all customers. Now, as we know this is an important variable, then we can build a decision tree to predict customer income based on occupation, product and various other variables. In this case, we are predicting values for continuous variable.

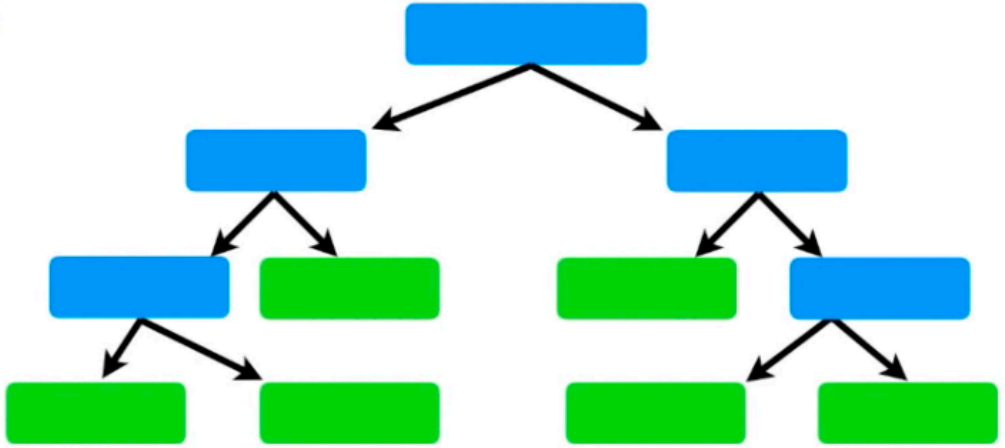
Decision trees



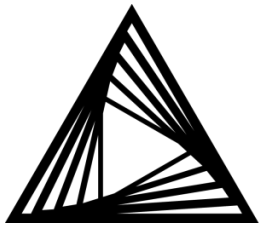
Decision trees

In this example, we want to create a tree that uses **chest pain**, **good blood circulation** and **blocked artery status** to predict…

Chest Pain	Good Blood Circulation	Blocked Arteries	Heart Disease
No	No	No	No
Yes	Yes	Yes	Yes
Yes	Yes	No	No
Yes	No	???	Yes
etc…	etc…	etc…	etc…



Kinme: Decision trees



Basic examples_Customer intelligence_
Churn prediction_building a churn prediction model

ML // SUPERVISED // classification

Decision trees: **adult income**

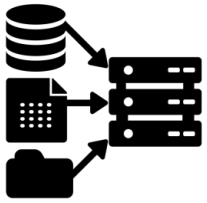
Decision trees: **adult income**

The Us Adult income dataset was extracted by Barry Becker from the 1994 US Census Database.
The data set consists of anonymous information such as occupation, age, native country, race, capital gain, capital loss, education, work class and more.
Each row is labelled as either having a salary greater than ">50K" or "<=50K".

The goal here is to train a binary classifier on the training dataset to predict the column **income** which has two possible values **">50K" and "<=50K"** and evaluate the accuracy of the classifier with the test dataset.

Variables:
- age
- workclass
- fnlwgt
- education
- education.num
- marital.status
- occupation
- relationship
- race
- sex
- capital.gain
- capital.loss
- hours.per.week
- native.country
- income

Decision trees: **adult income**



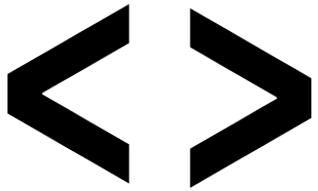
Adult.csv

KNWF TO IPYNB

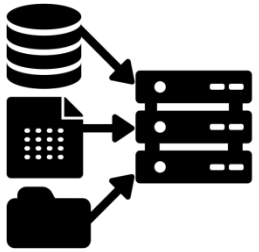
Knime-Python translation

KNWF TO IPYNB

Knime-Python translation



Titanic_translation.pynb



Train

Test