Signal Maritime

Interview info

SQL query that required to join multiple tables and select some columns of an sample database. No emphasis was given on syntax details.

The interview was not easy and included a short statistical test. After the test, there was a mini discussion about the results of the test and then a typical HR interview. The HR interview had a strict format. Generally speaking the whole process was not too friendly. Last but not least the company told me that I would hear from them in two weeks, but they never called me back.

Interview

I was interviewed from two managers. Each interview last around half an hour. One of them was really friendly but the second one had a weird attitude. They wanted to know about my business and technical skills, background of education and experience on data analysis, statistical techniques used previously.

Interview Questions

How will you rate your skill on R and SQL ?

Interview Questions

Ideal working enviroment, strengths and weaknesses, previous working experience

A rather demanding interview, needless to say don't go unprepared.

It started with a 30' quiz which contained rather complex questions on algorithmic methods, data analysis models, "IQ" riddles and statistics-probabilities.

The technical interview part followed, where the job was explained and some questions were asked, regarding my academic and working background.

The last part was a typical HR interview, where I was asked to provide examples of my ability to deal with…

Show More

Interview Questions

Tricky questions on optimal stopping theory, binary search algorithm, classification models etc.

Test in sql and general math questions including probabilities and clock hands degrees, some algorithms, some graph explanations etc. It was a typical test but needed focus in order to perform.

Interview Questions

SQL query that required to join multiple tables and select some columns of an sample database. No emphasis was given on syntax details.

A quiz including logical questions, probabilities, algorithms and sql queries. Most of the questions quite irrelevant to the related position. The interviewer was not that friendly and he seemed sarcastic at some times.

Interview Questions

What algorithm would you use in order to search for a name in a sorted list.

I remember a few questions like in comments below moreover Q's in sql and general math questions including probabilities/ statistics and some graph/table explanationsnumerical test (WITHOUT A CALCULATOR) etc. It was a OK test but needed focus in order to perform. You had a choice between exercises in order to have "full marks"

The second stage after 3 weeks was with one manager (however I was informed that I had also with the HR representatives but they were absent). He came with my test on hand and demanded to answer all questions I didn't answer in the test. He insisted on the hand clock question which seemed very odd and weird compared to the Statistics Skills required for this Job position. The interviewer was not that friendly and I felt that he lacked empathy with the interviewee . I showed some sample work templates also and asked a few things about the position also. Got a rejection letter two weeks after.

What is the ankle between the hour 3 and minutes 15?

A bank has a 10% probability of default this year. What is the probability of default of the bank in two years?

A Function generates random numbers distributed as a N(0, sigma). How would you modify the function to generate numbers distributed as a N(a, sigma) ?

Step 0: Understanding Data

Exploratory Data Analysis (EDA): Understanding the data we are using:
Assess limitations
assess underlying assumptions in the data structure

EDA is used for:

1. Detection of mistakes
2. Checking of assumptions
3. Preliminary selection of appropriate model and tools
4. Determining relationships of explanatory variables (independent)

i.e. renaming variables
encoding data
Correlation Matrix

The main causes of error in learning are due to **noise, bias and variance**

Step 1: Cleaning Data

A. Format the data in a “readable” form.

B. Take care of missing values (either replacing with mean value, median or deleting them)

C. Take care of anomalous values that deviate significantly (outliers) (replacing them with max or min value or deleting them). **We can easily detected them via Scatter-Plot. Πχ σε μια Normal distribution διωχνουμε τις τιμες που αντιστοιχούν 3σ ή χρησιμοποιουμε quantiles για να δουμε τί θα διώξουμε**

1. If missing or extreme values are a few we inspect them one by one.

2. If missing or extreme values are a lot they might hold clues to new insights. We try to find patterns, or we suspect selection bias or bad selection sampling.

Imputation: If we deal with missing values in a column we can

1. Remove the observation from the dataset.
2. Calculate a value for the NULL (imputate) [use of mean , median or multiple regression formula].

Step 2: Choosing a model to train

Supervised learning (Categories of data is known)

Continuous

Categorical

1. Regression (linear/polynomial)
2. Linear Discrimination Analysis (LDA)
3. Decision trees
4. Random Forest
5. Gaussian trees
6. MLP neural network
7. K nearest neighbors
8. Logistic Regression
9. Naive Bayes
10. SVM (Support Vector Machine)

Un-Supervised learning (Learning process attempts to find categories for data)

1. Clustering & Dimensionality Reduction (SVD,PCA, K-means)
2. Association Analysis (Apriori , FP-growth)
3. Hidden Markov Model

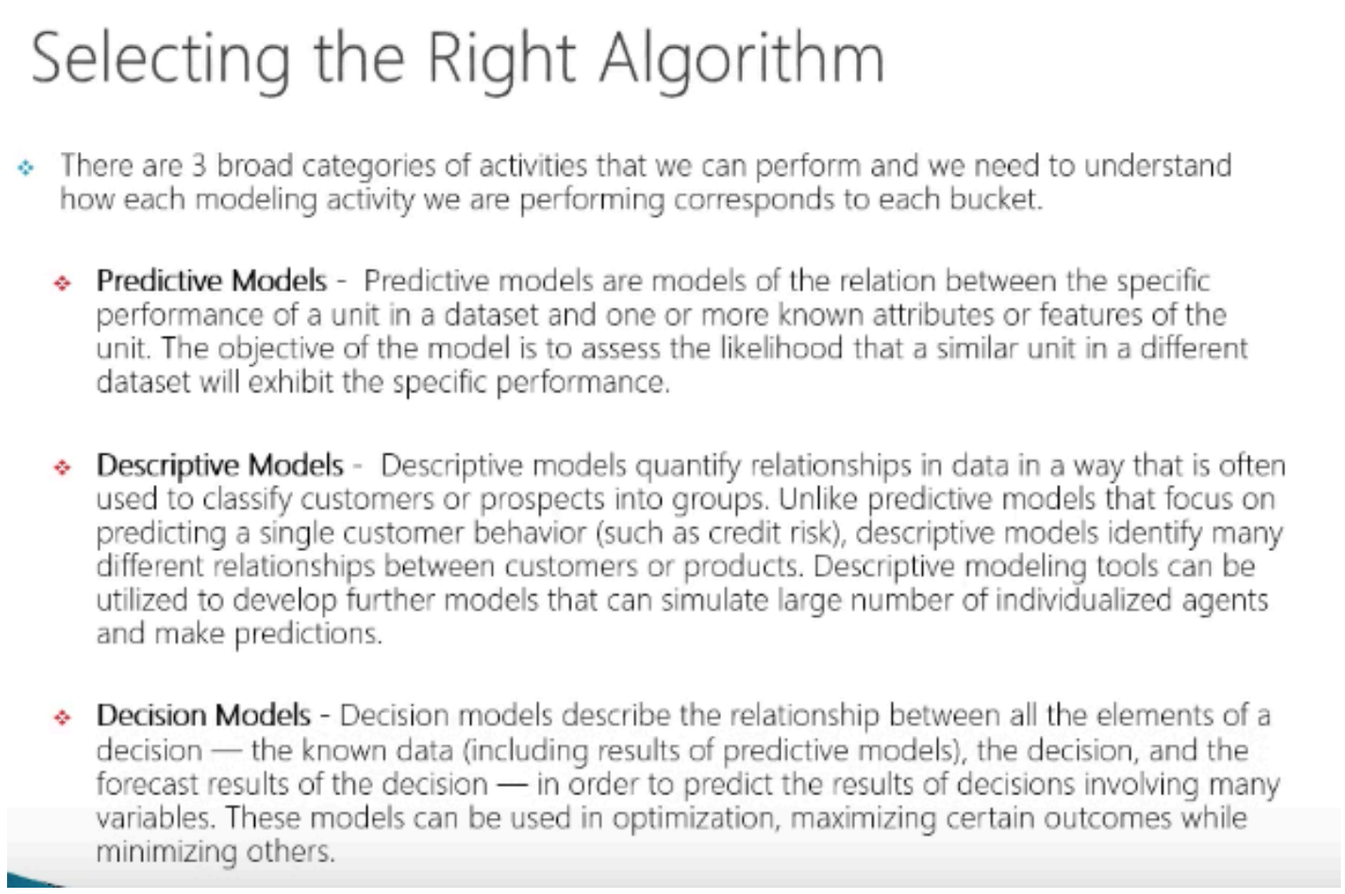
Clustering, like regression, describes the class of problem and the class of methods.

Association rule learning methods extract rules that best explain observed relationships between variables in data.

Πως επιλέγω μοντέλο.

A. Supervised - unsupervised
B. Binary ή multi class
C. εαν ειναι supervised classification ή regression - εαν ειναι unsupervised clustering ή dimensionality reduction
D. Linear ή non linear
E. Έξαρτάται απο στοχους high precision, speed , computational power

KNN πολυ αργό.



2. Linear Discriminant Analysis

Logistic Regression is a classification algorithm traditionally limited to only two-class classification problems. If you have more than two classes then the Linear Discriminant Analysis algorithm is the preferred linear classification technique.

The representation of LDA is pretty straight forward. It consists of statistical properties of your data, calculated for each class. For a single input variable this includes:

. The mean value for each class.

. The variance calculated across all classes.

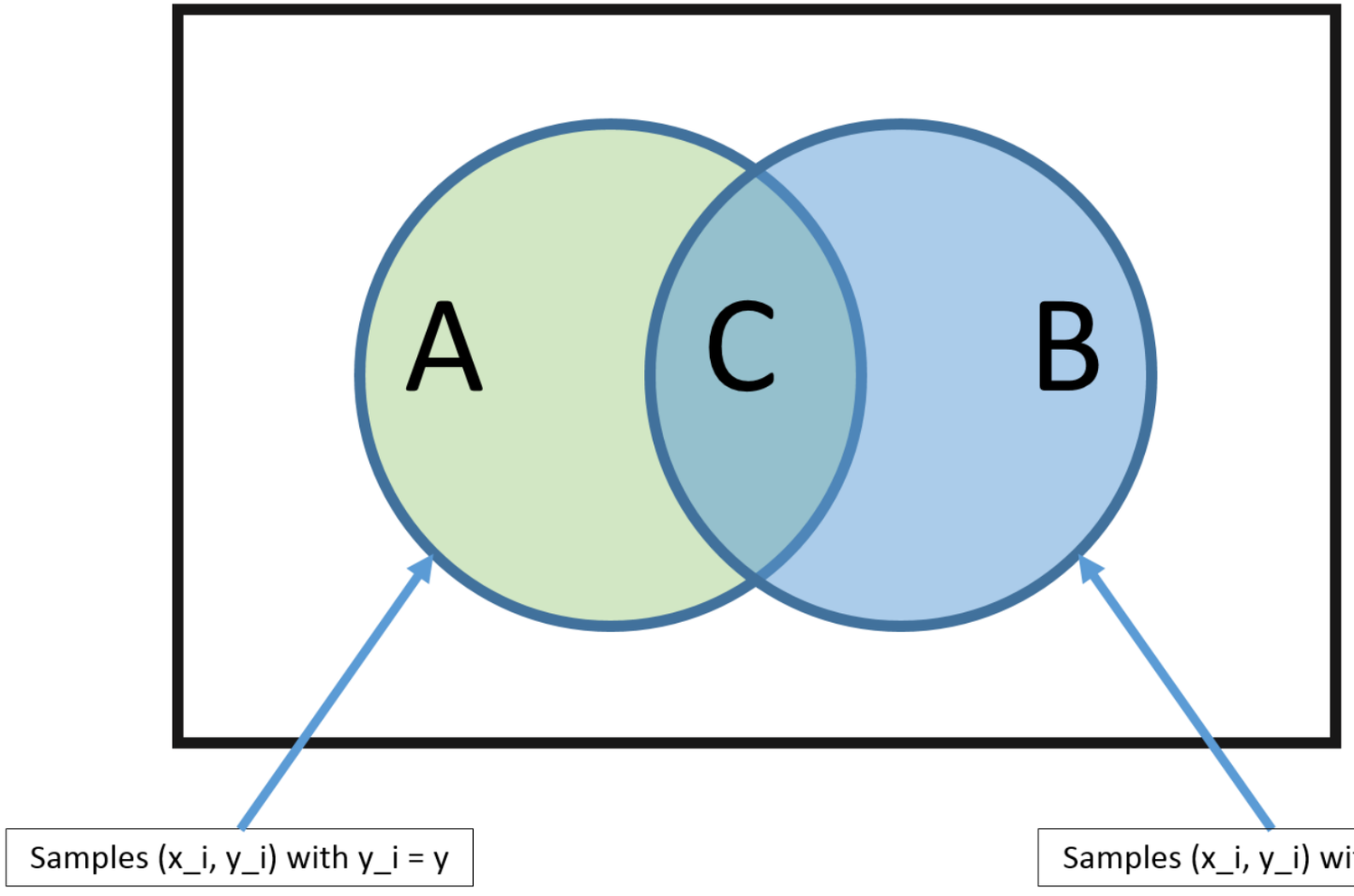
Predictions are made by calculating a discriminate value for each class and making a prediction for the class with the largest value. The technique assumes that the data has a Gaussian distribution (bell curve), so it is a good idea to remove outliers from your data before hand. It's a simple and powerful method for classification predictive modeling problems.

7. Naive Bayes

Naive Bayes is a simple but surprisingly powerful algorithm for predictive modeling.

The model is comprised of two types of probabilities that can be calculated directly from your training data: 1) The probability of each class; and 2) The conditional probability for each class given each x value. Once calculated, the probability model can be used to make predictions for new data using Bayes Theorem. When your data is real-valued it is common to assume a Gaussian distribution (bell curve) so that you can easily estimate these probabilities.

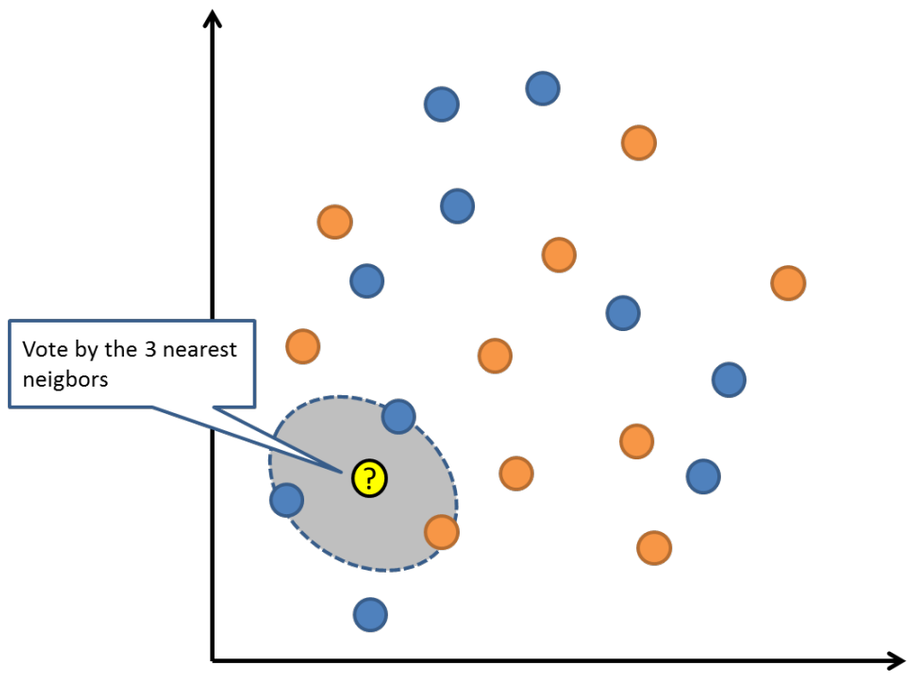
Naive Bayes is called naive because it assumes that each input variable is independent. This is a strong assumption and unrealistic for real data, nevertheless, the technique is very effective on a large range of complex problems.



6. K-Nearest Neighbors

The KNN algorithm is very simple and very effective. The model representation for KNN is the entire training dataset. Predictions are made for a new data point by searching through the entire training set for the K most similar instances (the neighbors) and summarizing the output variable for those K instances. For regression problems, this might be the mean output variable, for classification problems this might be the mode (or most common) class value.
The trick is in how to determine the similarity between the data instances. The simplest technique if your attributes are all of the same scale (all in inches for example) is to use the Euclidean distance, a number you can calculate directly based on the differences between each input variable.

KNN can require a lot of memory or space to store all of the data, but only performs a calculation (or learn) when a prediction is needed, just in time. You can also update and curate your training instances over time to keep predictions accurate.
The idea of distance or closeness can break down in very high dimensions (lots of input variables) which can negatively affect the performance of the algorithm on your problem. This is called the curse of dimensionality. It suggests you only use those input variables that are most relevant to predicting the output variable.



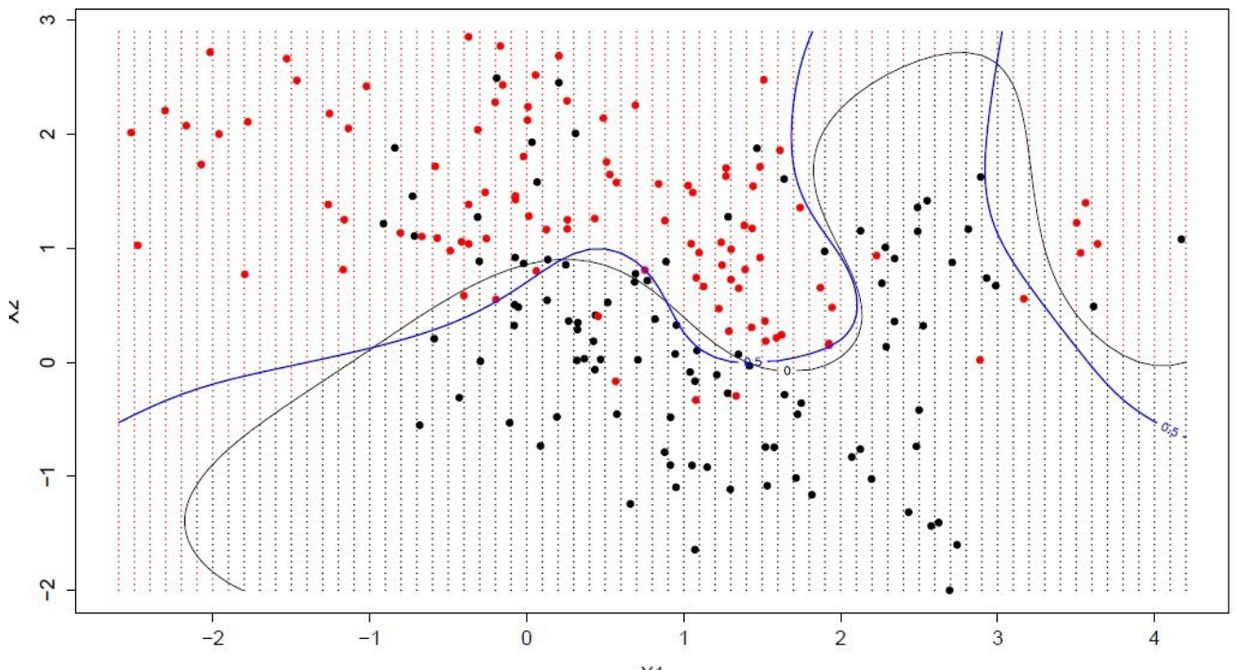
8. Support Vector Machines (SVD)

Support Vector Machines are perhaps one of the most popular and talked about machine learning algorithms.

A hyperplane is a line that splits the input variable space. In SVM, a hyperplane is selected to best separate the points in the input variable space by their class, either class 0 or class 1. In two-dimensions, you can visualize this as a line and let's assume that all of our input points can be completely separated by this line. The SVM learning algorithm finds the coefficients that results in the best separation of the classes by the hyperplane.

The distance between the hyperplane and the closest data points is referred to as the margin. The best or optimal hyperplane that can separate the two classes is the line that has the largest margin. Only these point are relevant in defining the hyperplane and in the construction of the classifier. These points are called the support vectors. They support or define the hyperplane. In practice, an optimization algorithm is used to fin the values for the coefficients that maximizes the margin.

SVM might be one of the most powerful out-of-the-box classifiers and worth trying on your dataset.

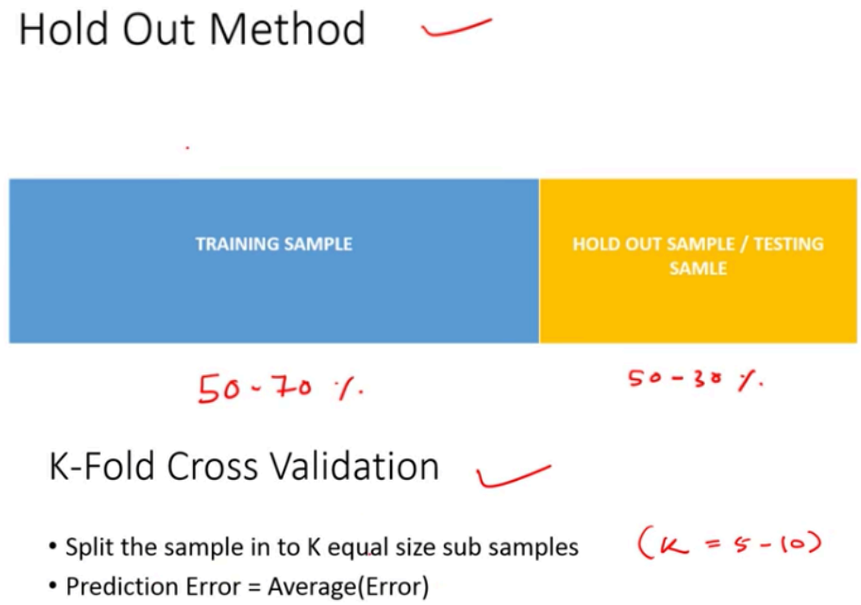


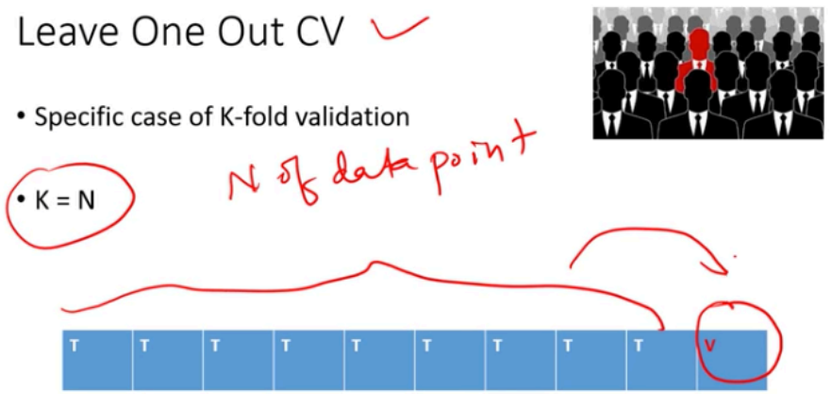
Step 3: Cross-Validation (CV)

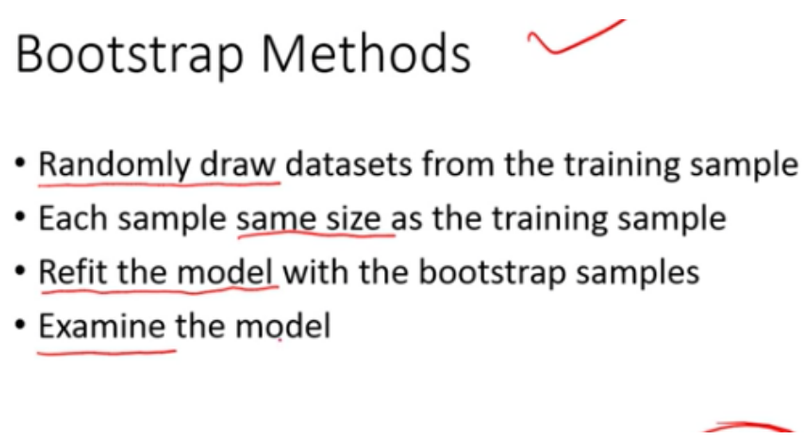
Cross-Validation allows us to compare different machine-learning methods (1. select the best fit model 2. ensures model is not over-fit)

CV METHODS:

1.Hold out method
2.K-Fold CV
+It matters less how data are divided
+Selection bias not present
3.Leave one out CV
+Good validation
-Long computational time
4. Bootstrap methods

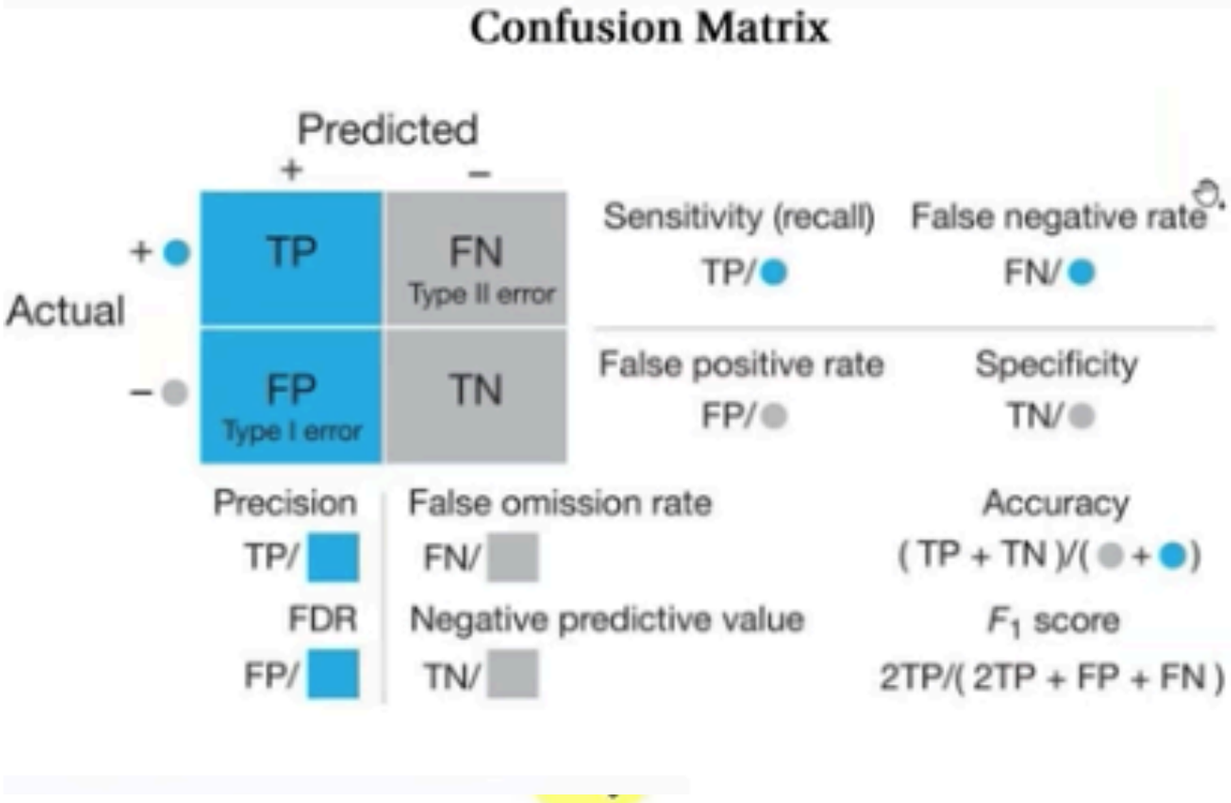


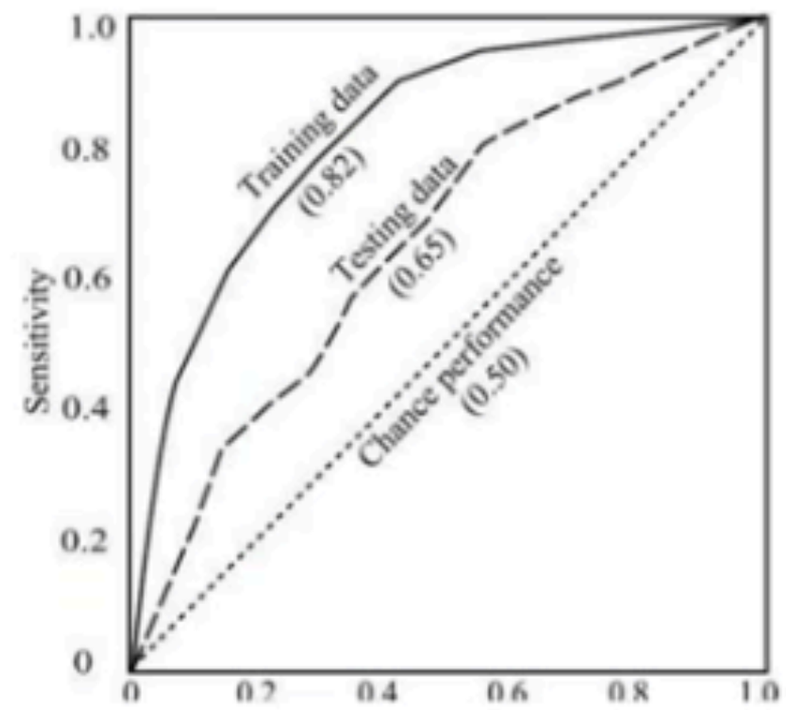




In Cross Validation all data is used both in training and testing and all methods are compared

Step 4: Testing





Key Metrics:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

Correct classification / Total number of cases

$$\text{Recall} = \frac{TP}{TP + FN}$$

TP / Actual Positives

$$\text{Precision} = \frac{TP}{TP + FP}$$

TP / Predicted Positives

Step : Bagging(bootstrapping, jack knife) vs Boosting

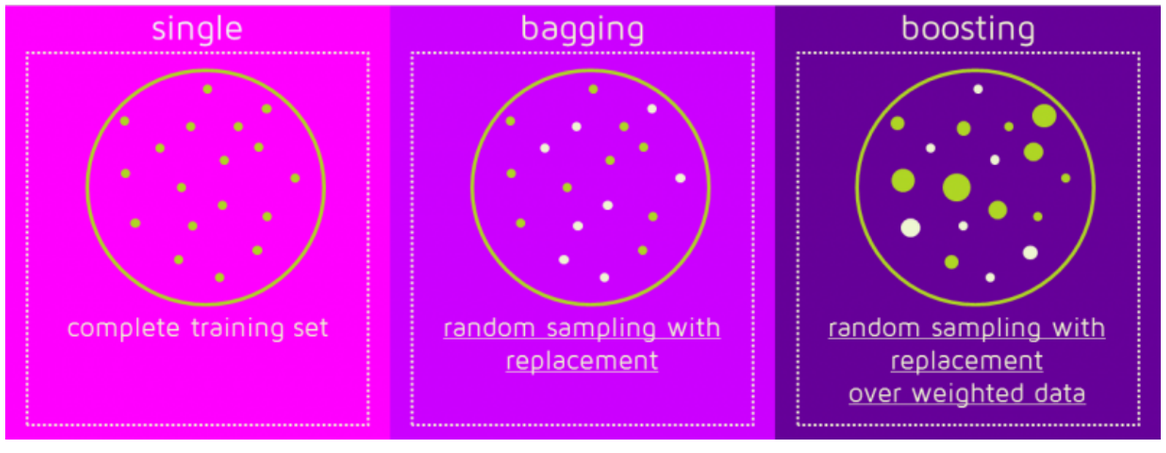
https://quantdare.com/what-is-the-difference-between-bagging-and-boosting/

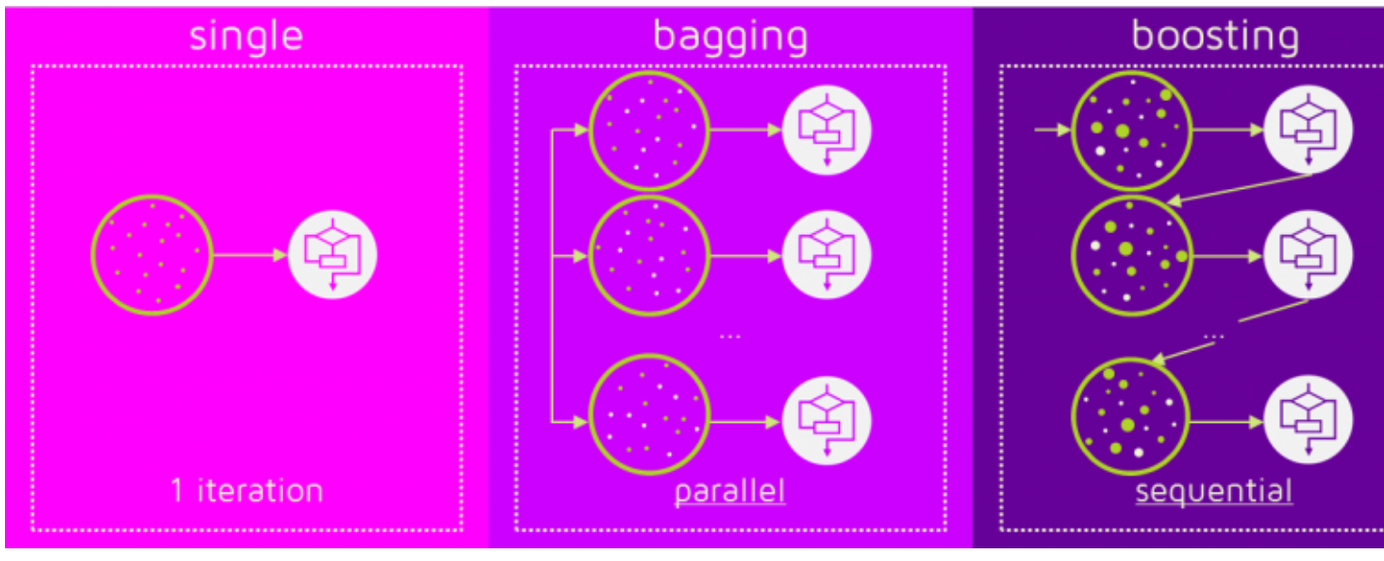
Ensemble is a Machine Learning concept in which the idea is to train **multiple models** using the same learning algorithm. The ensembles take part in a bigger group of methods, called **multiclassifiers,** where a set of hundreds or thousands of learners with a common objective are fused together to solve the problem.

Bagging and Boosting are both ensemble methods in Machine Learning, but what's the key behind them?
Bagging and Boosting are similar in that they are both **ensemble techniques**, where **a set of weak learners are combined t create a strong learner** that obtains better performance than a single one.

N new training data sets are produced by **random sampling with replacement** from the original set. By sampling with replacement some observations may be repeated in each new training data set.

In the case of Bagging, any element has the same probability to appear in a new data set. However, for Boosting the observations are weighted and therefore some of them will take part in the new sets more often:





Parralel **Sequential**

Step : Overfitting

Το σύστημα μαθαίνει απεξω τα train data, κανει capture τον θόρυβο

1. **Το overfitting δίνει πλασματικο high precision (many FP):**

Α.αποφευγεται με το cross validation και ensemble techniques (Bagging, boosting, random forest).

Β. Οταν εχεις πολλα parameters πιθανο το overfitting αρα παμε σε simple model (reducing features)

Πως κανουμε reduce τα features:
Α. αφαιρουμε correlated parameters (regression ή PCA, principal component analysis , unsupervised learning)

Β. random forest h boosting σου δειχνουν ποσο σημαντικο ειναι καθε feature

C. regularization (LASO,READS) τιμωρει τα features που πανε να γινουν overfit.

Clock angle problems

Καθε ακεραιη ωρα αντιστοιχει σε 360/12=30 μοίρες

A method to solve such problems is to consider the rate of change of the angle in degrees per minute. The hour hand of a normal 12-hour analogue clock turns 360° in 12 hours (720 minutes) or 0.5° per minute. The minute hand rotates through 360° in 60 minutes or 6° per minute.[1]

Equation for the angle of the hour hand

Equation for the angle of the hour hand [edit]

$$\theta_{\text{hr}} = 0.5^\circ \times M_\Sigma = 0.5^\circ \times (60 \times H + M)$$

where:

- θ is the angle in degrees of the hand measured clockwise from the 12
- H is the hour.
- M is the minutes past the hour.
- M_Σ is the number of minutes since 12 o'clock. $M_\Sigma = (60 \times H + M)$

Equation for the angle of the minute hand [edit]

$$\theta_{\text{min.}} = 6^\circ \times M$$

where:

- θ is the angle in degrees of the hand measured clockwise from the 12 o'clock pos
- M is the minute.

Example [edit]

The time is 5:24. The angle in degrees of the hour hand is:

$$\theta_{\text{hr}} = 0.5^\circ \times (60 \times 5 + 24) = 162^\circ$$

The angle in degrees of the minute hand is:

$$\theta_{\text{min.}} = 6^\circ \times 24 = 144^\circ$$

Equation for the angle between the hands [edit]

The angle between the hands can be found using the following formula:

$$\begin{aligned}\Delta\theta &= |\theta_{\text{hr}} - \theta_{\text{min.}}| \\ &= |0.5^\circ \times (60 \times H + M) - 6^\circ \times M| \\ &= |0.5^\circ \times (60 \times H + M) - 0.5^\circ \times 12 \times M| \\ &= |0.5^\circ \times (60 \times H - 11 \times M)|\end{aligned}$$

machine learning

1. overfitting (πλασματικο high precision)

A.αποφευγεται με το cross validation και ensemble techniques (Bagging, boosting, random forest).

Β. Οταν εχεις πολλα parameters πιθανο το overfitting αρα παμε σε simple model

1. πως κανουμε reduce τα features.

A. αφαιρουμε correlated parameters (regression ή PCA, principal component analysis , unsupervised learning)

B. random forest h boosting σου δειχνουν ποσο σημαντικο ειναι καθε feature

C. regularization (LASO,READS) τιμωρει τα features που πανε να γινουν overfit.

BINOMIAL

$$\binom{n}{x} = \frac{n!}{x!(n-x)!}$$

- $P(\text{Success}) = p$, and this stays constant from trial to trial.
- $P(\text{Failure}) = 1 - p$.
- X represents the number of successes in n trials.

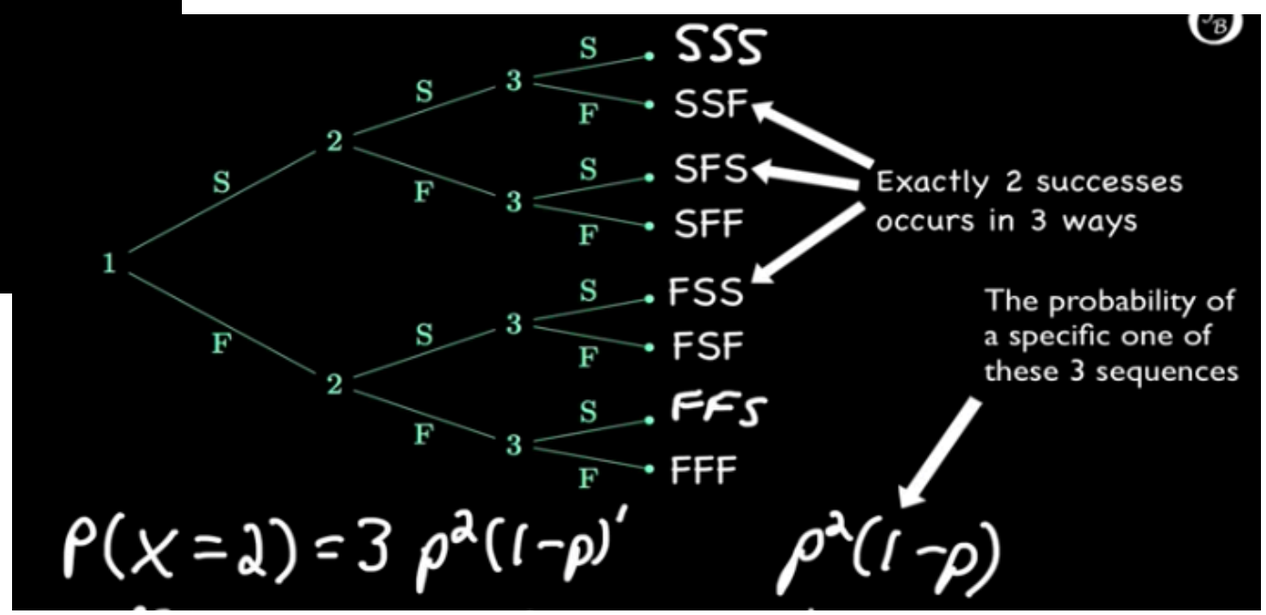
n= number of trials

Then X has a binomial distribution:

$$P(X = x) = \binom{n}{x} p^x (1-p)^{n-x}$$

for $x = 0, 1, 2, \ldots, n$.

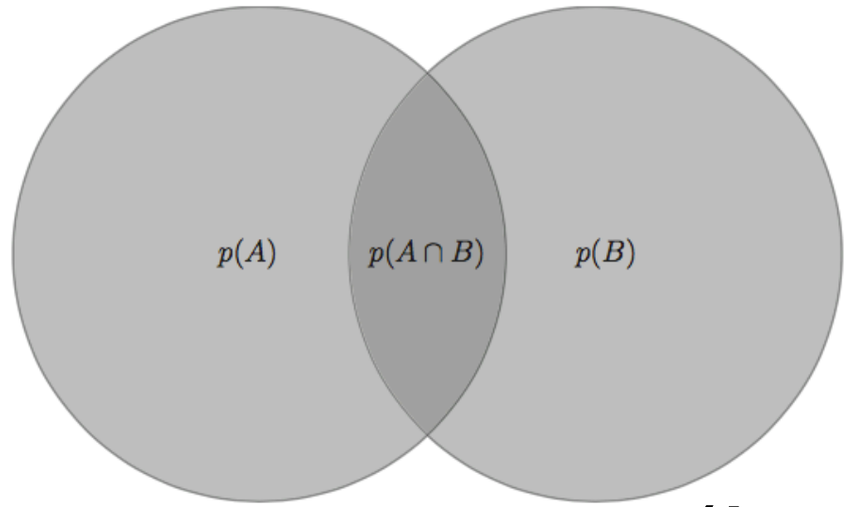
$\mu = E(X) = np$



Bayes

Both A and B (multiply)

$$P(A|B) = \frac{P(A \cap B)}{P(B)}$$



$p(A \cup B) = p(A) + p(B) - p(A \cap B)$ **(A or B)**

(Add)

$$\Pr(\mathrm{A}|\mathrm{X}) = \frac{\Pr(\mathrm{X}|\mathrm{A})\Pr(\mathrm{A})}{\Pr(\mathrm{X}|\mathrm{A})\Pr(A) + \Pr(\mathrm{X}|\sim \mathrm{A})\Pr(\sim A)}$$

+ result

cancer

Q. Given the following statistics, what is the probability that a woman has cancer if she has a positive mammogram result?

1. One percent of women over 50 have breast cancer.
2. Ninety percent of women who have breast cancer test positive on mammograms
3. Eight percent of women will have false positives

$P(A) = .01 \quad P(\sim A) = .99$

$P(X|A) = .9 \quad P(X|\sim A) = 0.08$

$$\frac{.9 * .01}{(.9 * .01) + (.08 * .99)} = 0.1$$

Bayes' theorem is stated mathematically as the following equation:[2]

$$P(A \mid B) = \frac{P(B \mid A)\, P(A)}{P(B)},$$

where A and B are events and $P(B) \neq 0$.

- $P(A \mid B)$ is a conditional probability: the likelihood of event A occurring given that B is true.
- $P(B \mid A)$ is also a conditional probability: the likelihood of event B occurring given that A is true.
- $P(A)$ and $P(B)$ are the probabilities of observing A and B independently of each other; this is known as the marginal probability.

Clock angle problems

Καθε ακεραιη ωρα αντιστοιχει σε 360/12=30 μοίρες

A method to solve such problems is to consider the rate of change of the angle in degrees per minute. The hour hand of a normal 12-hour analogue clock turns 360° in 12 hours (720 minutes) or 0.5° per minute. The minute hand rotates through 360° in 60 minutes or 6° per minute.[1]

Equation for the angle of the hour hand

Equation for the angle of the hour hand [edit]

$$\theta_{\text{hr}} = 0.5^\circ \times M_\Sigma = 0.5^\circ \times (60 \times H + M)$$

where:

- θ is the angle in degrees of the hand measured clockwise from the 12
- H is the hour.
- M is the minutes past the hour.
- M_Σ is the number of minutes since 12 o'clock. $M_\Sigma = (60 \times H + M)$

Equation for the angle of the minute hand [edit]

$$\theta_{\text{min.}} = 6^\circ \times M$$

where:

- θ is the angle in degrees of the hand measured clockwise from the 12 o'clock pos
- M is the minute.

Example [edit]

The time is 5:24. The angle in degrees of the hour hand is:

$$\theta_{\text{hr}} = 0.5^\circ \times (60 \times 5 + 24) = 162^\circ$$

The angle in degrees of the minute hand is:

$$\theta_{\text{min.}} = 6^\circ \times 24 = 144^\circ$$

Equation for the angle between the hands [edit]

The angle between the hands can be found using the following formula:

$$\begin{aligned}\Delta\theta &= |\theta_{\text{hr}} - \theta_{\text{min.}}| \\ &= |0.5^\circ \times (60 \times H + M) - 6^\circ \times M| \\ &= |0.5^\circ \times (60 \times H + M) - 0.5^\circ \times 12 \times M| \\ &= |0.5^\circ \times (60 \times H - 11 \times M)|\end{aligned}$$

- **Abstraction.** Abstraction means using simple things to represent complexity. We all know how to turn the TV on, but we don't need to know how it works in order to enjoy it. In Java, abstraction means simple things like **objects**, **classes**, and **variables** represent more complex underlying code and data. This is important because it lets avoid repeating the same work multiple times.

- **Encapsulation.** This is the practice of keeping fields within a class private, then providing access to them via public methods. It's a protective barrier that keeps the data and code safe within the class itself This way, we can re-use objects like code components or variables without allowing open access to the data system-wide.

- **Inheritance.** This is a special feature of Object Oriented Programming in Java. It lets programmers create new classes that share some of the attributes of existing classes. This lets us build on previous wo without reinventing the wheel.

- **Polymorphism.** This Java OOP concept lets programmers use the same word to mean different things in different contexts. One form of polymorphism in Java is **method overloading**. That's when different meanings are implied by the code itself. The other form is **method overriding**. That's when the different meanings are implied by the values of the supplied variables. See more on this below.

There are 2 types of polymorphism which are commonly mentioned.

. Dynamic Polymorphism

. Static Polymorphism

Fibonacci Recursive

Python

https://www.youtube.com/watch?v=Qk0zUZW-U_M

Simple

```
def fibonacci(n):
  if n == 1:
    return 1
  elif n == 2:
    return 1
  elif n > 2:
    return fibonacci(n-1) + fibonacci(n-2)

for n in range(1, 101):
  print(n, ":", fibonacci(n))
```

Simple- Memoization

```
fibonacci_cache = {}

def fibonacci(n):
  # If we have cached the value, then return it
  if n in fibonacci_cache:
    return fibonacci_cache[n]

  # Compute the Nth term
  if n == 1:
    value = 1
  elif n == 2:
    value = 1
  elif n > 2:
    value = fibonacci(n-1) + fibonacci(n-2)

  # Cache the value and return it
  fibonacci_cache[n] = value
  return value

for n in range(1, 101):
  print(n, )
```

Memoization tool

```
from functools import lru_cache

@lru_cache(maxsize = 1000)
def fibonacci(n):
  if n == 1:
    return 1
  elif n == 2:
    return 1
  elif n > 2:
    return fibonacci(n-1) + fibonacci(n-2)
```

Memoization+errors

```
from functools import lru_cache

@lru_cache(maxsize = 1000)
def fibonacci(n):
  # Check that the input is a positive integer
  if type(n) != int:
    raise TypeError("n must be a positive int")
  if n < 1:
    raise ValueError("n must be a positive int")

  # Compute the Nth term
  if n == 1:
    return 1
  elif n == 2:
    return 1
  elif n > 2:
    return fibonacci(n-1) + fibonacci(n-2)
```

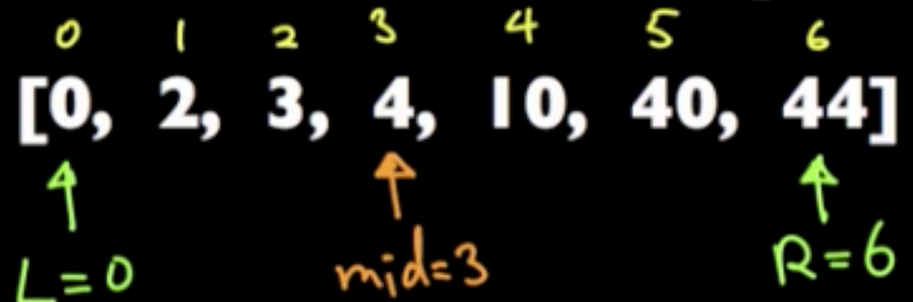
Insertion Sort Recursive

Python

https://www.youtube.com/watch?v=IEA31vHiry4

```python
def insertion_sort(list):
    for index in range(1,len(list)):
        value = list[index]
        i = index - 1
        while i>=0:
            if value < list[i]:
                list[i+1] = list[i] #shift number in slot i right to slot i+1
                list[i] = value # shift value left into slot i
                i = i - 1
            else:
                break
```

Recursive Binary Search (Code)




```
def BS (arr, L, R, x):

    if R >= L:
        mid = L + (R - L) / 2
        if arr[mid] == x:
            return mid
        else if arr[mid] > x:
            return BS(arr, L, mid-1, x)
        else:
            return BS(arr, mid+1, R, x)

    else:
        return -1
```

SQL

```
SQL> CREATE TABLE CUSTOMERS(
   ID   INT              NOT NULL,
   NAME VARCHAR (20)     NOT NULL,
   AGE  INT              NOT NULL,
   ADDRESS  CHAR (25) ,
   SALARY   DECIMAL (18, 2),
   PRIMARY KEY (ID)
);
```

```
INSERT INTO TABLE_NAME (column1, column2, column3,...columnN)
VALUES (value1, value2, value3,...valueN);
```

```
SELECT ...
FROM ...
(WHERE) ...
GROUP BY ...
(HAVING) ...
```

10 highest salary

```
SELECT TOP (1) Salary FROM
(
    SELECT DISTINCT TOP (10) Salary FROM Employee ORDER BY Salary DESC
) AS Emp ORDER BY Salary
```

```
SELECT salary from Employee order by salary DESC
LIMIT N-1,1
```

Accordingly we can find out Nth salary using `LIMIT (N-1),1`.

```
/*return employee name, highest salary and department*/
select e.first_name, e.last_name, e.salary, d.department_name
from Employee e Inner Join Department d ON (e.department_id = d.departm
where salary IN (select Max(salary) from employee)
```

To select all the **even** number records from a table:

```
Select * from table where id % 2 = 0
```

To select all the **odd** number records from a table:

```
Select * from table where id % 2 != 0
```

duplicate records with one field

```
SELECT name, COUNT(email)

FROM users
GROUP BY email
HAVING COUNT(email) > 1
```

duplicate records with more than one field

```
SELECT name, email, COUNT(*)

FROM users
GROUP BY name, email
HAVING COUNT(*) > 1
```

There are different types of joins available in SQL –

- INNER JOIN – returns rows when there is a match in both tables.
- LEFT JOIN – returns all rows from the left table, even if there are no matches in the right table.
- RIGHT JOIN – returns all rows from the right table, even if there are no matches in the left table.
- FULL JOIN – returns rows when there is a match in one of the tables.
- SELF JOIN – is used to join a table to itself as if the table were two tables, temporarily renaming at least one table in the SQL statement.
- CARTESIAN JOIN – returns the Cartesian product of the sets of records from the two or more joined tables.

1NF (First Normal Form) Rules

- Each table cell should contain a single value.
- Each record needs to be unique.

2NF (Second Normal Form) Rules

- Rule 1- Be in 1NF
- Rule 2- Single Column Primary Key

3NF (Third Normal Form) Rules

- Rule 1- Be in 2NF
- Rule 2- Has no transitive functional dependencies