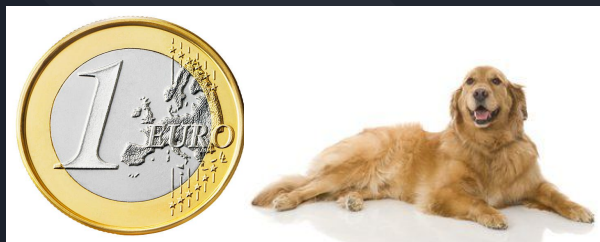


# Repte Eurecat



Equip EureDog

Alejandro Clavera, Joel Aumedes, Marc  
Cervera, Moises Bernaus



# Introducció

- Hem escollit el repte Eurecat atès que la nostra menció en els estudis d'informàtica es la de computació, per tant, la menció es centra principalment en la IA.
- L'hem trobat molt interessant.
- Hem vist que el podem solucionar mitjançant eines vistes en altres tasques que hem realitzat en el passat.

# Primer pas

- El primer pas de tots ha estat plantejar en un full les possibles maneres de resoldre el problema i repartir les diferents tasques a realitzar entre l'equip.
- Hem repartit les diferents tasques entre els membres de l'equip, així doncs cadascú es centraria en una part específica del programa.
- Finalment ens hem ficat a programar (tot i les adversitats: internet, sang del nas... :) ).



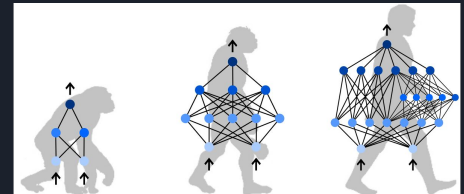
# Tractament de les dades

- Hem hagut d'entendre com estan organitzats els datasets i posteriorment fer un merge de tots ells realitzant un *full-join* en la columna de 'row\_date'.
- Gràcies a la funció *dropna()* s'han eliminat els registres per els quals no es disposava de valor.
- Un cop fet el merge, s'han introduït en una llista els inputs (els diferents valors) i en una altra els outputs (les variables booleanes).
- Les variables booleanes es tractaran com un tipus enter binari (0 o 1) mitjançant la funció *astype(int)*



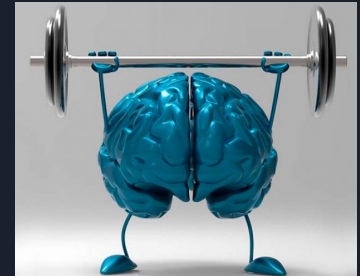
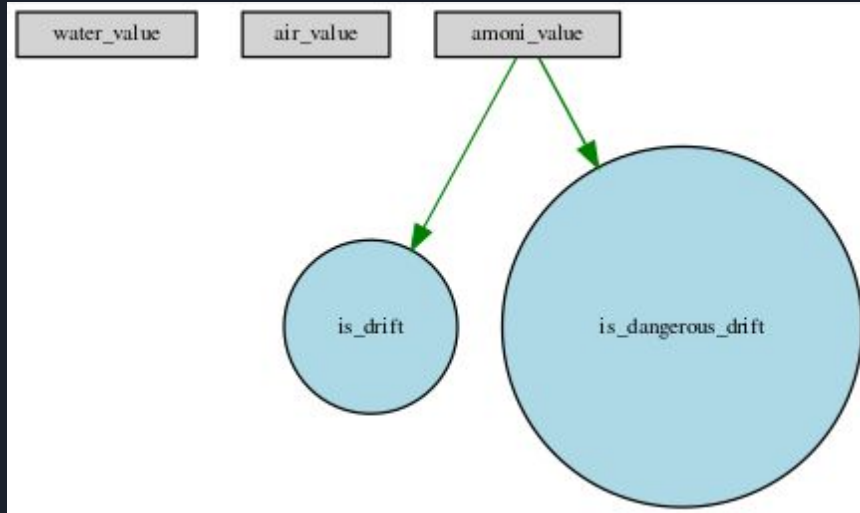
# Algoritme utilitzat

- L'algorisme que s'ha usat per resoldre el repte ha estat: *NeuroEvolution of augmenting topologies* (NEAT).
- En termes generals, proposa un algorisme genètic per a evolucionar poblacions de xarxes neuronals per arribar així a una solució òptima.
- L'objectiu principal és usar un algorisme genètic per buscar una solució òptima d'una xarxa neuronal modificant tant els seus pesos com la topologia i té un total de 3 propietats:
  - Existeix una representació genètica de la xarxa que permet la combinació de les estructures d'una manera significativa.
  - Protegeix la innovació topològica que necessita evolucionar algunes generacions per a ésser optimitzada i així no desaparegui del pool de genomes prematurament.
  - Minimitzar les topologies mitjançant entrenament i funcions de penalització



# Resultats primer algoritme

- Originalment volíem fer més d'una versió, amb nivells de complexitat incrementals, però sol hem pogut implementar la més simple.
- Aquesta primera generació tracta cada fila de les dades com a un registre aïllat.
- Al cap de poques generacions de training del primer model, l'*accuracy* es quedava encallada més o menys al 70%, i generava una xarxa neuronal amb una forma curiosa.



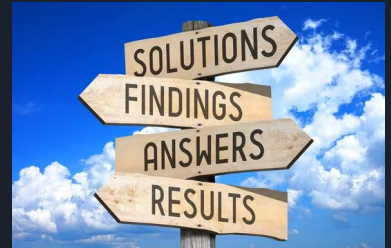
# Altres intents de solució

- Vam provar RandomForest i Regressions Lineals amb els 3 inputs, sense èxit.
- Una xarxa simple creada amb TensorFlow amb els mateixos inputs i outputs que l'algoritme NEAT, va donar també resultats similars, sense ser capaç de superar 70% d'*accuracy*.
- Vam intentar solucionar el problema utilitzant sol les dades d'aroni.



# Conclusions i resultats

Ens ho hem passat molt bé fent el repte, tot i haver realitzat bastant progrés al principi però haver-nos encallat i no haver pogut implementar tot el que teníem pensat originalment.





# Diapositiva extra 4K link full mega no fake no virus marcianito captat en camara

