Kaggle : Titanic

#THE MATHS:

binary classification: 0 or 1
=> y and t are real numbers:

- t=0 if dead, 1 if survived
- y > 0.5 if survived, <0.5 otherwise ~~y>0 and y<0~~.

Observations:

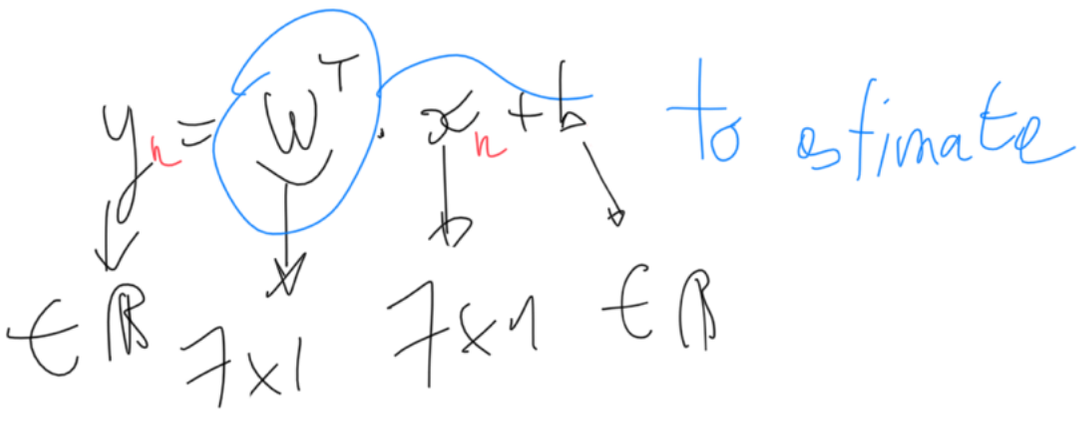
- Some fields are irrelevant for the feature, especially the fields with unique values that are not a qty : Passenger, Name, Ticket, Cabin,

=> We have an input x of 7 features

- Some fields must be converted into sort of indices: sex (0 for male, 1 for female), embarked (0 for Q, 1 for S, 2 for C) = a preprocessing

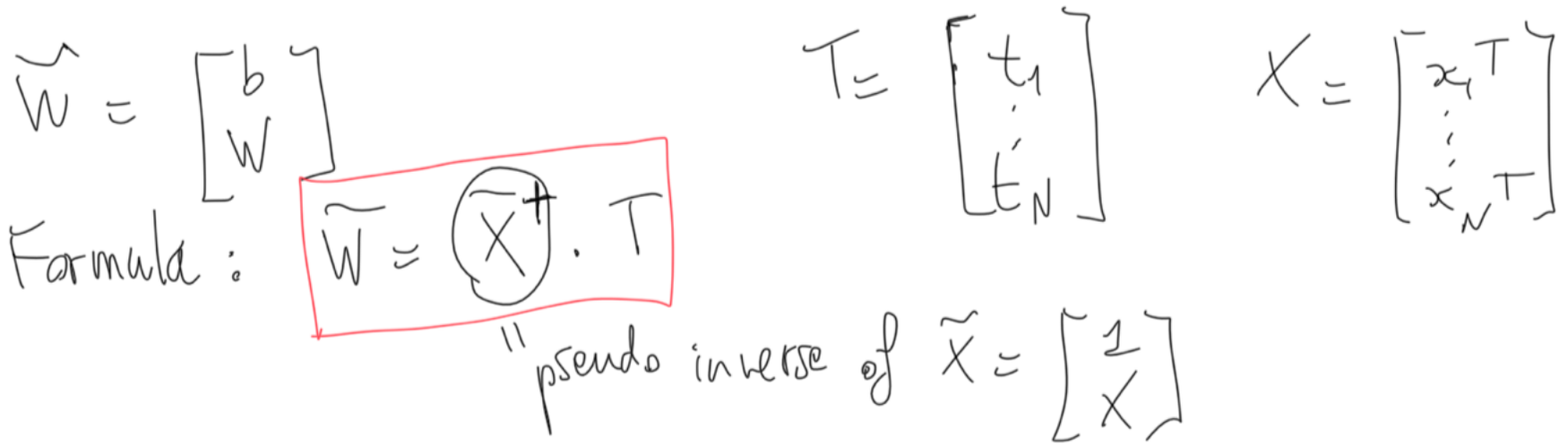
Chosen model

We can use a non generative model, a linear classifier



Estimating W

I will take the most naive method: least square classification (which is very sensitive to outliers). Formula:



#IMPLEMENTATION

Steps

- training
  - data extraction:
    - load X: load, preprocess (ignore useless fields and index some fields)
    - load T
  - estimate W tilde
- testing
  - data extraction:
    - load X
    - no T to load
  - use classifier on test data => collect the yns'.
  - no eval
- output csv : 2 cols: id and surived or not.

Training

###Data extraction

- Initialize data: X a Nx7 matrix and T a N vector.
- => We must know N => count the (used) rows.
- Fill data. I can read line by line, naively!! :
  - convert each line into an array of strings
  - T(n) = I(1) => Load T
  - X(n): completed by extracting the right fields

####Estimating W tilde

- X tilde
- Compute W tilde

Testing

###Data extraction

Load X like before

###Classification

- Compute the yns. Can be done naively in a loop or with repmat. (naively first)
- Classify

Output

Output in a file since it's long. We can output line by line, the first line being the header.

#PROBLEMS

Against missing data: For now, I just ignore the rows with missing data for the useful fields.

=> The code must be adapted since less data than expected..

But for the testing phase, how do we deal with that ?? For now (not good either): I consider that if I can't know the missing info, he/she's dead.

The classes are 0 and 1, not -1 and 1 => Will have to adapt the code: e.g. by converting into -1 and 1 or transforming the

decision boundary into 0.5??

~~By the way: Pandas~~
~~Pandas that I have never used before can be used to simplify loading and writing into files. But for that, I will have to know how to use it of course..~~