Ils veulent qu'on se serve des yearLastFailure comme d'une feature: pourquoi pas les utiliser comme label. Seul problème: les outputs sont ternaire (distinction de 2014 et 2015), et on sait pas si les features1-4 sont time sensitive

Les valeurs dans X des pipe ayant eu des failures sont trop petites, il faut trouver un autre moyen pour distinguer les failures des non failures

Common techniques used for unbalanced problems:

- Over-sampling: add copies of the under-represented class, OR use synthetically generated data, using the SMOTE algorithm (https://github.com/scikit-learn-contrib/imbalanced-learn)
- Under-sampling: delete data from the over-represented class *(note: peut-être qu'au lieu de delete sauvageme, on pourra faire une PCA sur les pipe qui n'ont jamais fail et prendre comme data les n-premiers vecteurs)*

- Decision trees are told to perform better on unbalanced data sets: try those algorithm: C4.5, C5.0, CART, and Random Forest. (*note: j'ai jamais eu de cours sur les decision tree, mais ça parait etre une bonne occasion de comprendre comme ça marche*)

- Try penalized models (models that penalize more errors on the under-represented class. Ex: penalized-SVM and penalized-LDA) (*note: j'y connais rien en LDA)*

- Divide the over-represented class in smaller different classes (*genre par date ou un truc comme ça*)

Papier la dessus:
- http://sci2s.ugr.es/keel/pdf/algorithm/congreso/kubat97addressing.pdf