**TO-DO LIST, IDEAS AND USEFUL LINKS FOR THE DATATHON:**

* Mejorar presentación del trabajo:
  + PDF explicativo estilo paper con:
    - Resumen al principio.
    - Tablas y gráficas de la exploración de los datos originales.
    - Explicación y visualización de técnicas.
    - Conclusión.
  + Jupyter notebooks separadas, con buen estilo: https://medium.com/ibm-data-science-experience/markdown-for-jupyter-notebooks-cheatsheet-386c05aeebed
* Tratamiento de datos:
  + ~~Año construcción:~~
    - ~~Cambiar a años que lleva construido~~ y construir histograma.
  + ~~Plantas construidas vecinos:~~
    - ~~Cambiar valor de los nulos de -1 a cero.~~
  + ~~Variables categóricas a one hot vectors, cuyo tamaño sea igual al número de posibilidades consideradas en la variable. Ej: meses del año, vector de 12 vars que pueden ser o uno o cero~~. <https://machinelearningmastery.com/how-to-one-hot-encode-sequence-data-in-python/>
  + Datos faltantes:
    - Imputación
      * [**https://justmachinelearning.com/2019/08/25/data-imputation-techniques-in-machine-learning/**](https://justmachinelearning.com/2019/08/25/data-imputation-techniques-in-machine-learning/)
      * [**https://www.aaai.org/Papers/KDD/1996/KDD96-023.pdf**](https://www.aaai.org/Papers/KDD/1996/KDD96-023.pdf)
      * [**https://towardsdatascience.com/6-different-ways-to-compensate-for-missing-values-data-imputation-with-examples-6022d9ca0779**](https://towardsdatascience.com/6-different-ways-to-compensate-for-missing-values-data-imputation-with-examples-6022d9ca0779) **(importante)**
  + ~~Datos descompensados:~~
    - General: <https://towardsdatascience.com/machine-learning-multiclass-classification-with-imbalanced-data-set-29f6a177c1a>
    - Balanceo del dataset:
      * General: <https://www.datasciencecentral.com/profiles/blogs/handling-imbalanced-data-sets-in-supervised-learning-using-family>
      * Oversampling:
        + SMOTE:

<https://medium.com/@saeedAR/smote-and-near-miss-in-python-machine-learning-in-imbalanced-datasets-b7976d9a7a79>

<https://kite.com/blog/python/smote-python-imbalanced-learn-for-oversampling/>

<https://machinelearningmastery.com/smote-oversampling-for-imbalanced-classification/>

* + - * + ADASYN:
        + Artículo que trata ambos SMOTE y ADASYN: <https://medium.com/coinmonks/smote-and-adasyn-handling-imbalanced-data-set-34f5223e167>
        + ADASYN vs ENN: <https://medium.com/quantyca/oversampling-and-undersampling-adasyn-vs-enn-60828a58db39>
      * Undersampling:
      * Oversampling + undersampling:
        + [**https://machinelearningmastery.com/combine-oversampling-and-undersampling-for-imbalanced-classification/**](https://machinelearningmastery.com/combine-oversampling-and-undersampling-for-imbalanced-classification/)
  + Visualización de datos:
    - General: <https://towardsdatascience.com/the-art-of-effective-visualization-of-multi-dimensional-data-6c7202990c57>
    - PCA vs TSNE: <https://medium.com/analytics-vidhya/pca-vs-t-sne-17bcd882bf3d>
* Pesos de cada clase: <https://scikit-learn.org/stable/modules/generated/sklearn.utils.class_weight.compute_class_weight.html>
* Reducción de la dimensión:
  + General: <https://elitedatascience.com/dimensionality-reduction-algorithms>
  + Autoencoders (para usar K-vecinos próximos):
    - Introducción: <https://elitedatascience.com/dimensionality-reduction-algorithms>
    - Keras: <https://www.datacamp.com/community/tutorials/autoencoder-keras-tutorial>
    - Tutorial simple: <https://towardsdatascience.com/autoencoders-made-simple-6f59e2ab37ef>
* ~~Extra Trees model~~:
  + Vídeo: <https://www.youtube.com/watch?v=Q1qpG7gwix4>
  + Diferencia con Random Forest: <https://stats.stackexchange.com/questions/175523/difference-between-random-forest-and-extremely-randomized-trees>
  + Scikit Learn: <https://scikit-learn.org/stable/modules/ensemble.html>
* ~~GridSearch~~: ‘De lo grande a lo pequeño’. Rango más grande y mayor granularidad.
  + Emplear Google Colab: <https://colab.research.google.com/notebooks/intro.ipynb>
* ~~Support Vector Machines~~ (realizado sin PCA):
  + Tutorial: <https://www.geeksforgeeks.org/classifying-data-using-support-vector-machinessvms-in-python/>
  + Scikit-Learn: <https://scikit-learn.org/stable/modules/svm.html>
  + Explicación: <https://towardsdatascience.com/support-vector-machine-simply-explained-fee28eba5496>
* **Auto machine learning:** 
  + ~~Overview de MLBOX~~: <https://towardsdatascience.com/automl-in-python-an-overview-of-the-mlbox-package-208118a7fe5>
  + Auto-sklearn package: <https://automl.github.io/auto-sklearn/master/>
  + Tutorial: <http://docs.h2o.ai/h2o/latest-stable/h2o-docs/automl.html>

**OAA-DB Algorithm:** <https://ieeexplore.ieee.org/document/6252450>