**ASTRALARIA: INFORME (10/04/2020)**

1. Se ha probado a transformar la variable CONSTRUCTIONYEAR a otra ANTIQUITY, de año de construcción a antigüedad (ANTIQUITY = 2020 – CONSTRUCTIONYEAR). Las métricas son prácticamente las mismas para ambos modelos (*XGBoost* y *Random Forest*) que las de la entrega de la Fase I.
2. En la variable MAXBUILDINGFLOOR, cuando era un valor nulo, se codificaba a -1 y ahora a 0. Las métricas son prácticamente las mismas para ambos modelos (*XGBoost* y *Random Forest*) que las de la entrega de la Fase I.

**CONCLUSIÓN [A]:** Si los cambios **[1]** y **[2]** mejoran o empeoran la precisión real del modelo, no es posible detectarlo a priori.

1. Se ha probado a transformar la variable CADASTRALQUALITY de tipo categórica a un conjunto de tipo one-hot. Para ello se ha eliminado esta y añadido las 13 one-hot correspondientes (CADASTRALQUALITY\_XX).

**CONCLUSIÓN [B**]: Se ha generado un fichero respuesta con 100 iteraciones con los cambios **[1],** **[2]** y **[3].** Dicho fichero es muy similar al de la entrega de la fase I, aunque contiene menos muestras predichas a la clase RESIDENTIAL.

1. Se ha entrenado un modelo *Support Vector Machine* con los cambios **[1], [2]** y **[3].** La *accuracy* se reduce al 23% y el *recall* al 14%.

**CONLUSIÓN [C]:** Un modelo SVC no parece el idóneo para este problema.

1. Se ha probado a hacer la técnica de *Auto Machine Learning* con la libreria *MLBox* pero no produce ningún resultado.
2. Se ha probado el modelo *Extra Trees* con un número variable de estimadores que oscilan entre el 100 y el 600 y con los cambios **[1], [2]** y **[3].** Las métricas son ligeramente peores para ambos modelos (*XGBoost* y *Random Forest*) que las de la entrega de la Fase I.
3. Se han implementado diversas técnicas de *oversampling* y *undersampling* incluidas combinaciones de estas como *SMOTE + ENN* o *SMOTE + Tomek Links*.
4. Se ha entrenado el modelo original con los cambios **[1], [2]** y **[3]** y con diversas técnicas del **[7].** Utilizando *SMOTE + Tomek Links* se consigue una *accuracy* del 94%, valor altamente sospechoso.

**CONLUSIÓN [D]:** El entrenamiento de este modelo parece haber sufrido *overfitting*. El fichero resultante con este modelo contiene aprox. 1000 muestras más predichas a RESIDENTIAL.

1. Se ha realizado *Grid Search,* de valores más generales a más concretos, para intentar buscar los mejores hiperparametros tanto para *Random Forest* como para XGBoost. Nos encontramos con varios problemas:

* Si al lanzar la búsqueda utilizamos todos los datos originales, obtendremos unos hiperparametros optimizados para la clase RESIDENTIAL
* Si utilizamos el balanceo de la entrega de la Fase I (añadir las muestras de todas las clases excepto la RESIDENTIAL, que solo se añaden entre 5000 y 6000 muestras) al ser las muestras de RESIDENTIAL una fracción aleatoria del original (5000 muestras de los 90000 originales), al realizar una búsqueda los hiperparametros obtenidos no serán los idóneos para el sistema real. Hemos lanzado diversas búsquedas con este método y los hiperparametros cambian y mucho.
* Si realizamos la técnica *SMOTE + Tomek* **[7]** a los datos y luego realizamos la búsqueda los hiperparametros obtenidos no serán realmente los idóneos para mejorar nuestro sistema de la Fase I.

**CONCLUSIÓN [E]:** No parece posible encontrar los mejores hiperparametros para nuestro modelo original, para ello habría que lanzar una búsqueda en cada iteración y teniendo en cuenta que realiza 100 el tiempo de entrenamiento se dispararía exponencialmente.

1. [Paper OAA-DB](https://www.researchgate.net/publication/261086899_Enhancing_classification_performance_of_multi-class_imbalanced_data_using_the_OAA-DB_algorithm): Estamos empezando a implementar este sistema y pensamos que puede ser prometedor. Lo dividiremos en las siguientes fases:
   1. Obtener los datasets:
      1. X, y originales
      2. X\_train\_random, y\_train\_random y los tests random, utilizando en principio un 20% de las muestras para test
      3. X\_train\_notrandom, y\_train\_notrandom y los tests NO random. Nos será más representativo, pues intentamos que la clase RESIDENTIAL sea mucho mayor y así podemos observar cómo va prediciendo el resto de las clases minoritarias
   2. Primer clasificador XGBClassifier
   3. Primer OneVS(…)
      1. Rest
      2. One *(según el paper, más favorable para datasets grandes)*
   4. Balanceo de datos con técnicas:
      1. Oversampling + undersampling:
         1. SMOTE + ENN
         2. SMOTE + Tomek Links
      2. SMOTE + CMTNN → En el paper habla de dos redes neuronales complementarias feedforward con backpropagation. Una TruthNN y otra FalsityNN. No sabemos bien cómo o con qué librerías podríamos realizar lo que en el paper se muestra
   5. OneVS(…) con el nuevo dataset balanceado
   6. Resultado final obtenido

**CONCLUSIÓN FINAL:** Falta investigar en las técnicas de reducción de dimensionalidad y la realización del *autoencoder.* Para la evaluación de la entrega intermedia, la cual admite dos ficheros respuesta, habíamos pensado enviar el fichero del sistema original modificado **CONCLUSIÓN [B**] y o el fichero del modelo de **CONLUSIÓN [D]** o el del **[10]** si logramos implementarlo para la fecha.