Reto Minsait Land Classification Astralaria

Asier Serrano Aramburu y Mario Campos Mocholí

Universitat Politècnica de València

asserar@inf.upv.es, macammoc@inf.upv.es

1 Introducción

La Organización nos reta a desarrollar un sistema de clasificación automática multiclase de suelos en base a imágenes de satélites. La métrica a maximizar del sistema es el *accuracy* [1], definida por:

$$accuracy = \frac{registros\ \ bien\ \ clasificados}{total\ \ registros} \quad (1)$$

Para lograr este fin, se proporcionan dos conjuntos de muestras:

- **Modelar**: 103230 muestras etiquetadas, utilizado para entrenar y evaluar el modelo por el grupo. En adelante, conjunto modelar.
- **Estimar**: 5618 muestras sin etiquetar, utilizado para evaluar el sistema por la Organización. En adelante, conjunto estimar.

La dificultad de este problema recae en el gran desbalanceamiento de muestras de cada clase, el gran solapamiento que existe entre las clases y la diferente distribución de muestras entre los dos conjuntos.

El *script* que genera el fichero respuesta es *Astralaria.py*.

2 Dominio del problema

Antes de realizar un análisis exploratorio de las variables, se ha llevado a cabo una labor de búsqueda y documentación del reto para comprender mejor su naturaleza. Las muestras proporcionadas están compuestas por variables que provienen de dos fuentes: la Dirección General del Catastro (DGC) [2] y los satélites de la misión Sentinel II de la Agencia Espacial Europea (ESA) [3].

2.1 El catastro

La Dirección General del Catastro, o El Catastro, es el órgano directivo de la Secretaría de Estado de Hacienda del Ministerio de Hacienda encargado del Catastro, el registro administrativo donde se describen los bienes inmuebles rústicos, urbanos y de características especiales.

De este organismo proceden las variables referentes al identificador, el área, varias medidas geométricas condensadas, el año de construcción, la altura máxima construida en los registros colindantes y el valor catastral.

2.2 Copernicus Sentinel II

La misión Copernicus Sentinel II está compuesta por una constelación de dos satélites situados en una órbita heliosíncrona y desfasados 180º. La función de estos es monitorizar los cambios que se producen en la superficie terrestre siendo capaces de volver a analizar una misma localización cada 2-5 días, dependiendo de la latitud.

Cada satélite está equipado con un instrumento de medición multiespectral del que se obtienen trece bandas [4]. Los datos producidos son de acceso libre. De los datos de la misión se han obtenido variables referentes a las banda 2, 3, 4 y 8 que corresponden al canal azul, verde, rojo e infrarrojos respectivamente.

2.3 Naturaleza de las variables

Cada muestra está formada por 55 variables, 56 en el caso del conjunto modelar. Cada una representa:

- **Identificador**: Variable que permite identificar cada muestra. El valor está ofuscado por lo que no se puede utilizar para buscar información en El Catastro. Corresponde a la variable ID.
- **Coordenadas**: Dos variables que hacen referencia a la longitud y latitud del terreno. Sus valores también están ofuscados pero manteniendo la relación de la distribución real. Corresponden a las variables X e Y.
- **Variables de medición**: Un total de 44 variables que representan la densidad de color de la banda azul, verde y roja, y el valor del canal infrarrojo en cada decil. Hace referencia a las variables Q_*b*_*c*_*n*_*m* siendo *b* el nombre del canal: R, G, B o NIR, *c* el número del canal: 2, 3, 4 u 8 y *n*_*m* el número del decil en el rango `0_0 a 1_0`.
- **Área**: Variable que representa la extensión en m2 del registro. Corresponde a la variable AREA.

- **Variables geométricas**: Cuatro variables que hacen referencia a propiedades geométricas inespecíficas. Corresponde a GEOM_Rn, siendo n el número de la variable en el rango del 1 al 4.
- **Año de construcción**: Variable que representa el año del que data el registro. Corresponde a la variable CONSTRUCTIONYEAR.
- **Altura máxima de construcción**: Variable que representa la altura máxima construida de los registros que rodean al terreno. Corresponde a MAXBUILDINGFLOOR.
- **Identificador catastral**: Variable que representa la clasificación de calidad que otorga El Catastro [5]. Corresponde a la variable CADASTRALQUALITYID.
- **Clase**: Variable que representa la clase a la que pertenece la muestra y que solo se haya en las muestras del primer conjunto. Corresponde a la variable CLASE.

3 Análisis exploratorio de los datos

En esta sección se realiza el análisis y representación de los datos para entenderlos mejor, descubriendo así que realmente las clases están desbalanceadas y solapadas.

3.1 Tipo de variables

El conjunto de variables es íntegramente numérico a excepción de la variable CADASTRALQUALITYID, que es de tipo categórico. Dicha variable toma los valores [A, B, C, 1, 2, 3, 4, 5, 6, 7, 8, 9] y, atendiendo a la naturaleza de la variable, el valor A es la máxima puntuación y el 9 la mínima.

En el conjunto modelar existen 20 muestras con algún valor nulo y 7 en el conjunto estimar. Los valores nulos se encuentran en las variables MAXBUILDINGFLOOR y CADASTRALQUALITYID.

3.2 Distribución de las clases

Los datos del conjunto modelar siguen la distribución de la Tabla 1. En la Figura 1 se puede observar la desproporción entre la clase RESIDENTIAL y el resto.

Tabla 1. Distribución de las clases

Clase	Nº muestras	Total %
RESIDENTIAL	90173	87.28%
INDUSTRIAL	4490	4.36%
PUBLIC	2976	2.88%
OFFICE	1828	1.77%
OTHER	1332	1.29%
RETAIL	2093	2.03%
AGRICULTURE	338	0.33%

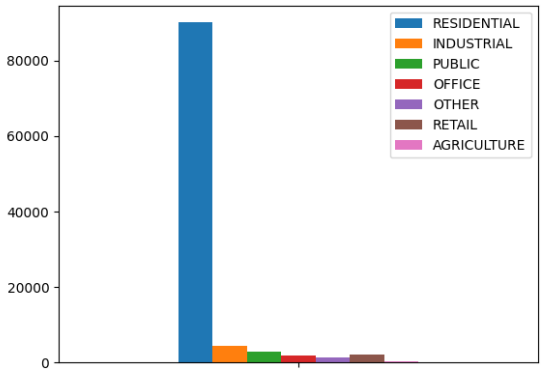

Figura 1. Histograma de las clases.

3.3 Distribución entre ambos conjuntos

Se ha explorado en detalle la distribución de las variables en ambos conjuntos encontrando una dato significante.

En el conjunto de modelar, el valor medio de la variable AREA en la clase RESIDENTIAL es de $281m^2$ mientras que el valor media del resto de clases es superior a los $1000m^2$, siendo la media total de $441m^2$.

Al calcular la media de la misma variable en el conjunto estimar se ha descubierto que el valor es de $967m^2$.

Este hallazgo lleva a suponer que el conjunto de estimar tendrá una distribución más equitativa de las clases.

3.4 Visualización

Se han aplicado diversas técnicas para visualizar la distribución de las muestras en el espacio dimensional y así determinar que técnicas de variación de la dimensionalidad se deben aplicar.

3.4.1 PCA

Se ha aplicado la técnica de *Principal Component Analysis* [6] en diferentes dimensionalidades para descubrir algún patrón, más concretamente, a dos dimensiones en la Figura 2 y a tres dimensiones en la Figura 3. Se puede observar en ambos casos que el solapamiento es claro.

3.4.2 t-SNE

Se ha aplicado la técnica de *t-Distributed Stochastic Neighbor Embedding* [7], adecuada para descubrir patrones en grandes conjuntos de datos.

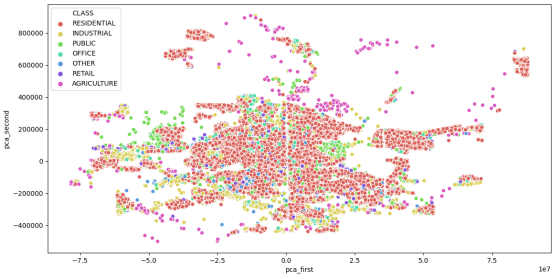

Figura 2. Representación de los datos en el espacio bidimensional mediante PCA.

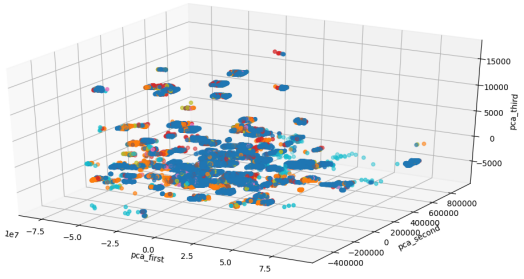

Figura 3. Representación de los datos en el espacio tridimensional mediante PCA.

Tal y como se observa en la Figura 4, la clase RESIDENTIAL solapa todas las clases. Asimismo, en la Figura 5 también se contempla que existe solapamiento entre el resto de clases.

3.5 Conclusión del análisis exploratorio

Observando las representaciones se confirma que existe solape entre las clases y que no presentan patrones evidentes que permitan su separación.

Por lo tanto, aplicar técnicas de reducción de dimensionalidad no dará buenos resultados y se deben buscar

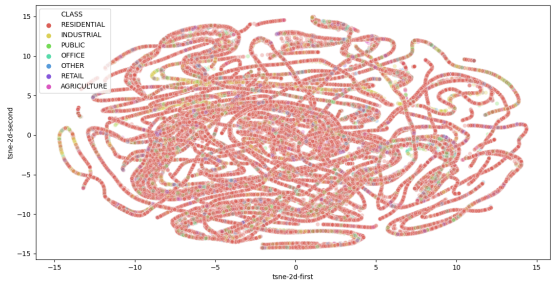

Figura 4. Representación de los datos en el espacio bidimensional al 50 de perplejidad mediante t-SNE.

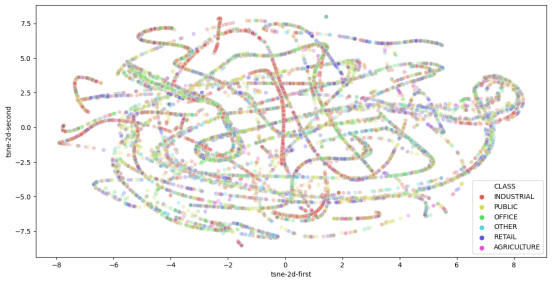

Figura 5. Representación de los datos, sin la clase RESIDENTIAL, en el espacio bidimensional al 50 de perplejidad mediante t-SNE.

técnicas alternativas, como *feature engineering*.

4 Manipulación de las variables

Para solucionar los problemas descritos anteriormente será necesario manipular los datos y adaptarlos para su correcto funcionamiento en los sistemas y obtener mejores métricas. A continuación, se describen todas las transformaciones realizadas.

4.1 Pretratamiento

Paso previo a aplicar técnicas más complejas, se ha realizado un preprocesamiento de los datos para adaptar las variables a la mayoría de sistemas así como manipular los valores nulos para no descartar las muestras que lo contienen, ya que pueden ser representativas de la clase:

- **CONSTRUCTIONYEAR**: Se ha modificado la variable para representar la antigüedad de los terrenos colindantes y no el año.
- **MAXBUILDINGFLOOR**: Se ha tratado la variable para poder utilizar las muestras con valores nulos. Se ha optado por añadir el valor 0 a estas.
- **CADASTRALQUALITYID**: Se ha transformado de una variable categórica a una variable one-hot. Además, también se ha tratado para poder utilizar las muestras con valores nulos. Por lo tanto, cada muestra aumenta su dimensionalidad en 12.

4.2 Reducción de la dimensionalidad

Análisis previos indicaban que las técnicas de reducción de dimensionalidad no iban a aportar resultados y se ha comprobado empíricamente que así es. Algunas de las transformaciones realizadas son:

- **PCA**: Se ha aplicado para cambiar, tanto aumentar como reducir, la dimensión de las variables pero,

como ya se ha observado, el solapamiento entre clases aumenta.

- **Banda RGB**: Análisis realizados en diferentes modelos determinan que algunos deciles son muy poco significativos por lo que se ha probado a eliminarlos. También se ha probado a crear una variable categórica que condense la densidad de los deciles y ordenen los tres colores según importancia. Ambas transformaciones han empeorado las métricas globales.

- **Banda NIR**: Se ha probado a condensar los valores de los deciles y transformar la variable a una categórica según el umbral de densidad. La transformación empeora las métricas.

- **Geometría**: Se ha probado a relacionar estas variables con la variable AREA sin exito alguno.

4.3 Feature engineering avanzado

Las técnicas del apartado anterior no han dado buenos resultados y se ha optado por aplicar técnicas que aumenten la dimensionalidad.

- ***Deep Feature Synthesis***: Se ha probado una técnica de *feature engineering* automática basada en *Deep Feature Synthesis* [8] mediante la librería FeatureTools [9]. Las variables añadidas no aportaban valor a las muestras por lo que se ha descartado este cambio.

- ***K-D Tree***: Las coordenadas X e Y mantienen la relación entre los terrenos. Ello lleva a pensar que hay combinaciones de vecinos menos probables que otras, por ejemplo, una muestra de la clase OFFICE será más difícil que colinde con otra de AGRICULTURE. Como consecuencia de esta observación, se ha implementado un *k-d tree* que determine los K-1 vecinos más próximos, o la probabilidad de que cada clase lo sea, según si empleamos el conjunto modelar o estimar, respectivamente. Dicha técnica aumenta en 7 la dimensionalidad, una por clase.

5 Implementación del modelo

Tras el anterior análisis de los datos podemos afirmar, tal como indicaba la Organización, que los grandes problemas del reto son los expuestos en la introducción de este documento Por ello, el sistema a desarrollar deberá contar con la flexibilidad y robustez necesaria para clasificar correctamente pese a estos inconvenientes.

Tabla 2. *Accuracy* de las técnicas

Técnica	***Accuracy* %**
SVM	23%
K-Neighbors	26%
Naive Bayes	16%
Extra Decision Trees	55%
GBoosting	60%
Random Forests	59%
K-Means	17%
Spectral Clustering	21%
OAA-DB	56%

5.1 Antecedentes

El modelo final es consecuencia de la experimentación con diversos modelos y algoritmos del ámbito del machine learning. Respecto al aprendizaje supervisado se han probado *Support Vector Machines*, algoritmo de *K-Neighbors*, clasificadores *Naive Bayes*, clasificadores basados en el algoritmo de *Decision Trees*, algoritmo de *Gradient Boosting* y clasificadores basados en *Random Forests*.

Pese que a la naturaleza del problema no es la indicada para utilizar *clustering*,o aprendizaje no supervisado, se ha probado el algoritmo de *K-Means* y técnicas de *Spectral Clustering*.

Las técnicas de regresión lineal no se adaptan al problema por lo que se ha optado por obviarlas.

Finalmente, se han probado redes neuronales como *autoencoders*, para descubrir patrones en las muestras, y algoritmos más complejos como OAA-DB [10].

Los valores de *accuracy* aproximados obtenidos, sin realizar ninguna técnica de balanceo ni tuneado de hiperparámetros, son los que se observan en la Tabla 2.

5.2 Modelo elegido

De entre todas las técnicas analizadas destacan dos de ellas: *Random Forests* y *Gradient Boosting*. Son las elecciones finales para la realización del sistema de predicción. Más concretamente se ha utilizado el clasificador basado en *Random Forest*s de Sklearn [11] y el clasificador *Gradient Boosting* de XGBoost [12] con *booster* basado en *Random Forests* también.

Se han utilizado estos algoritmos ya que son los que mejores resultados aportan al reto y, con una posterior optimización de hiperparámetros, se han conseguido mejores métricas.

5.3 Balanceo de datos

Paso previo a la implementación del sistema es necesario resolver el problema del desbalanceo entre clases para no generar imparcialidad, favoreciendo a la clase mayoritaria.

Se han realizado diversas técnicas [13], todas ellas implementadas mediante la librería Imblearn [14]. Más concretamente se ha experimentado con:

- Oversampling
  - *Random oversampling*
  - SMOTE [15]
  - ADASYN [16]
- Undersampling
  - *Random undersampling*
  - ENN [17]
  - *Tomek Links* [18]
- Combinadas
  - SMOTE + ENN
  - SMOTE + *Tomek Links*

Las técnicas de *oversampling* aumentan el ruido de las muestras obteniéndo así peores métricas mientras que las de *undersampling* eliminaban instancias significativas de la clase RESIDENTIAL.

Finalmente se ha optado por un método de *random undersampling* que más adelante se detallará.

5.4 Clasificación por consenso

El modelo final sigue el diagrama de la Figura 6. En él, los dos conjuntos de datos son pretratados y se les aplica técnicas de *feature engineering*.

A partir de este punto se inicia un proceso iterativo de entrenamiento. Los datos tratados del conjunto modelar son sometidos a un proceso de *random undersampling* en el cual se mantienen todas las muestras de cada clase excepto de la de RESIDENTIAL, que se obtienen 6000 aleatorias. Por tanto, el conjunto de modelar pasa a tener 19057 muestras.

Se realiza un *train-test split* del conjunto y se entrena, por separado, el clasificador basado en *Random Forests* y el clasificador *Gradient Boosting*, guardando las métricas de cada uno, por separado, para una posterior selección. Los mejores hiperpárametros para cada modelo se han obtenido mediante *Grid Search*.

Como el proceso de *random undersampling* puede eliminar instancias significativas de la clase RESIDENTIAL, el proceso de balanceo, *train-test split* y entrenamiento se repite un número determinado de veces que es 100 por defecto.

Al terminar las iteraciones se cuenta con una lista de modelos, de ambos tipos, ordenada por su métrica F_1. Se selecciona un porcentaje, que por defecto es del 40%, y a estos modelos se les pasa el conjunto de estimar para predecir.

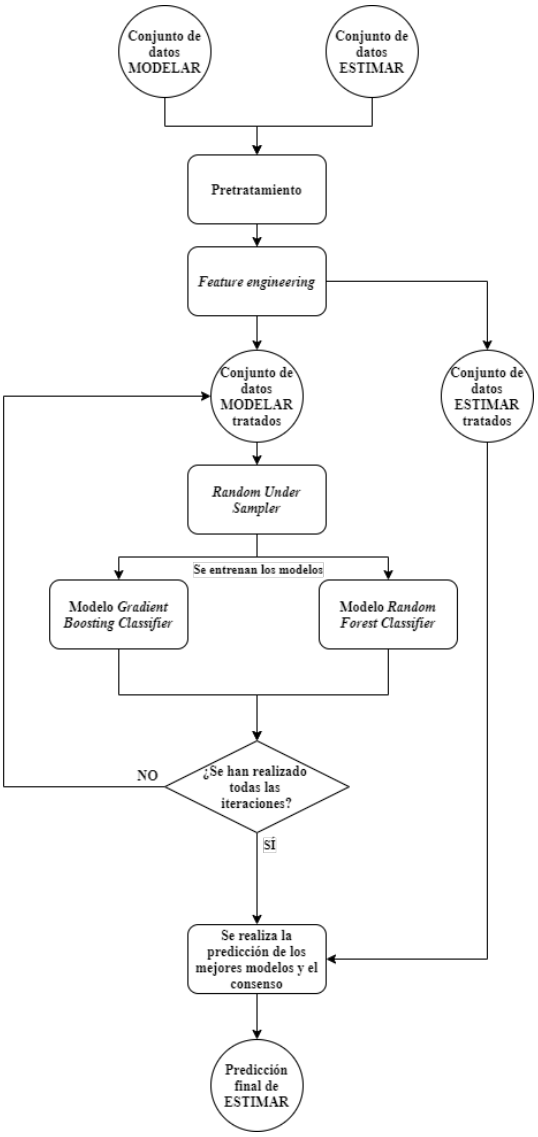

Figura 6. Diagrama de flujo del modelo.

Se crea un diccionario siendo las claves de este los ID del conjunto estimar y como valor una lista para cada entrada. Cada modelo añadirá una predicción independiente a cada lista.

Finalmente, se recorre el diccionario realizado una predicción por consenso, es decir, la clase que más se haya obtenido para un ID será la clase que se predecirá. El resultado de este proceso se vuelca a un fichero.

Tabla 3. Métricas del sistema

Métrica	Sin tratamiento%	Pretr. + FE %
accuracy	63.21%	64.13%
precision	61.41%	62.45%
recall	56.04%	57.43%
F1	57.81%	59.1%

5.5 Evaluación del modelo

Pese que la métrica a optimizar es el *accuracy*, hemos buscado la máximización del *F1* [1] ya que, como hemos determinado en el análisis exploratorio, el conjunto estimar está más balanceado. Esta métrica se define como:

$$F_1 = 2 \cdot \frac{precision \cdot recall}{precision + recall} \quad (2)$$

Las métricas obtenidas para las diferentes versiones del sistema son las que se muestran en la Tabla 3 y tal como se observa los procesamientos mejoran ligeramente todas las métricas. Se ha realizado *cross validation* para reducir el riesgo de *overfitting*.

6 Conclusiones

El *accuracy* obtenido dista mucho de ser el óptimo de un sistema cuya labor sea de este calibre. Para intentar mejorar las métricas se recomendaría a la Organización añadir variables más significativas.

Si a su vez se dispone de las imágenes satelitales, se aconseja combinar las técnicas anteriores con técnicas de *computer vision* para resolver este problema.

Bibliografía

[1] Precision and recall. `https://en.wikipedia.org/wiki/Precision_and_recall`, Accedido: 22/04/2020.

[2] Dirección general del catastro. `https://www.sedecatastro.gob.es/`, Accedido: 22/04/2020.

[3] Misión sentinel 2. `https://sentinel.esa.int/web/sentinel/missions/sentinel-2`, Accedido: 22/04/2020.

[4] Misión sentinel 2: Tipo de datos medidos. `https://en.wikipedia.org/wiki/Sentinel-2`, Accedido: 22/04/2020.

[5] Clasificación registros de el catastro. *http://www.catastro.meh.es/documentos/publicaciones/ct/ct36/ct36_5.pdf*, year = Accedido: 22/04/2020,.

[6] Jonathon Shlens. *A Tutorial on Principal Component Analysis* 51. (2014).

[7] Laurens van der Maaten y Geoffrey H. *Viualizing data using t-SNE* 9. (2008).

[8] James M. K. y Kalyan V. *Deep feature synthesis: Towards automating data science endeavors* pp. 1-10. (2015).

[9] Feature tools library. `https://www.featuretools.com`, Accedido: 22/04/2020.

[10] Piyasak J. y Kok Wai W. *Enhancing classification performance of multi-class imbalanced data using the OAA-DB algorithm* 1-8. (2012).

[11] Scikit-learn library. `https://scikit-learn.org/stable/`, Accedido: 22/04/2020.

[12] Xgboost library. `https://xgboost.readthedocs.io/en/latest/index.html`, Accedido: 22/04/2020.

[13] Guillaume L. y Fernando N. y Christos K. A. *Imbalanced-learn: A Python Toolbox to Tackle the Curse of Imbalanced Datasets in Machine Learning*. (2017).

[14] Imbalanced-learn. `https://imbalanced-learn.readthedocs.io/en/stable/api.html`, Accedido: 22/04/2020.

[15] Nitesh V. y Kevin W. y Lawrence O. y W. Philip. *SMOTE: Synthetic Minority Over-sampling Technique*. 16. (2002).

[16] Haibo H. y Yang B. Edwardo A. y Shutao L. *ADASYN: Adaptive Synthetic Sampling Approach for Imbalanced Learning*- 1322 - 1328. (2008).

[17] Haibo H. y Yang B. Edwardo A. y Shutao L. *Resampling techniques for improving classification performance in unbalanced datasets*. (2016).

[18] Haibo H. y Yang B. Edwardo A. y Shutao L. *Mitigating the effects of class imbalance using smote and tomek link undersampling*. (2018).