

Utilización de librerías geospaciales libres para la clasificación de datos LiDAR.

Pilar Martínez-Blanco⁽¹⁾, Aitor Bastarrika-Izagirre⁽²⁾ y Javier M^a Sánchez-Espeso⁽³⁾

⁽¹⁾⁽²⁾ Departamento de Ingeniería Minera y Metalúrgica y C. de los Materiales, Escuela Universitaria de Ingeniería de Vitoria-Gasteiz, Universidad del País Vasco / Euskal Herriko Unibertsitatea UPV/EHU, C/ Nieves Cano, 12, 01006 Vitoria-Gasteiz, mp.martinez@ehu.eus⁽¹⁾; aitor.bastarrika@ehu.eus⁽²⁾

⁽³⁾ Departamento Ingeniería Geográfica y Técnicas de Expresión Gráfica, ETSI de Caminos, Canales y Puertos, Universidad de Cantabria, C/ Avda de los Castros s/n, 39005 Santander, javier.sanchez@unican.es.

RESUMEN

En este documento se presentan las herramientas geospaciales libres utilizadas en un trabajo de investigación en el que se ha desarrollado una metodología capaz de clasificar una nube de puntos bruta capturada en un vuelo LiDAR según las especificaciones de la American Society for Photogrammetry and Remote Sensing (ASPRS) en su versión 1,2, en el entorno de la Comunidad Autónoma del País Vasco (CAPV). Se ha trabajado con datos del vuelo LiDAR, obtenidos en el marco del Plan Nacional de Ortofotografía Aérea (PNOA) del año 2008, complementado con el uso de imágenes aéreas en las bandas Roja, Verde, Azul e Infrarrojo de las ortofotografías del mismo año.

La metodología empleada se ha basado en la Minería de Datos utilizando el algoritmo Random Forest bajo el lenguaje de programación de Python en un entorno Linux, a partir del cual se han manejado distintas librerías libres que han permitido gestionar las nubes de puntos como datos matriciales.

Palabras clave: *LiDAR, Random Forest, Python, pandas, numpy, libLAS, GDAL, SAGA, OTB, SPDLib, scikit-learn.*

INTRODUCCIÓN

La tecnología LiDAR (Light Detection and Ranging) permite, en base a la integración de datos capturados por tres sensores desde una aeronave, un láser, un sistema de posicionamiento GNSS (Global Navigation Satellite System) y un sistema inercial (Inertial Navigation System); obtener una nube de puntos tridimensional georeferenciada (figura 1) muy precisa, con una densidad muy elevada.

Uno de los parámetros más característicos de esta nube de puntos se basa en la densidad de los mismos por metro cuadrado, siendo el formato estándar de distribución el LAS (Log ASCII Standard), cuyas especificaciones han sido desarrolladas por la ASPRS (American Society for Photogrammetry and Remote

Sensing) [1]. Según dichas especificaciones, existen diferentes formatos para la grabación de los datos, PDRF (Point Data Record Format), y en función de ellos los valores asociados varían.

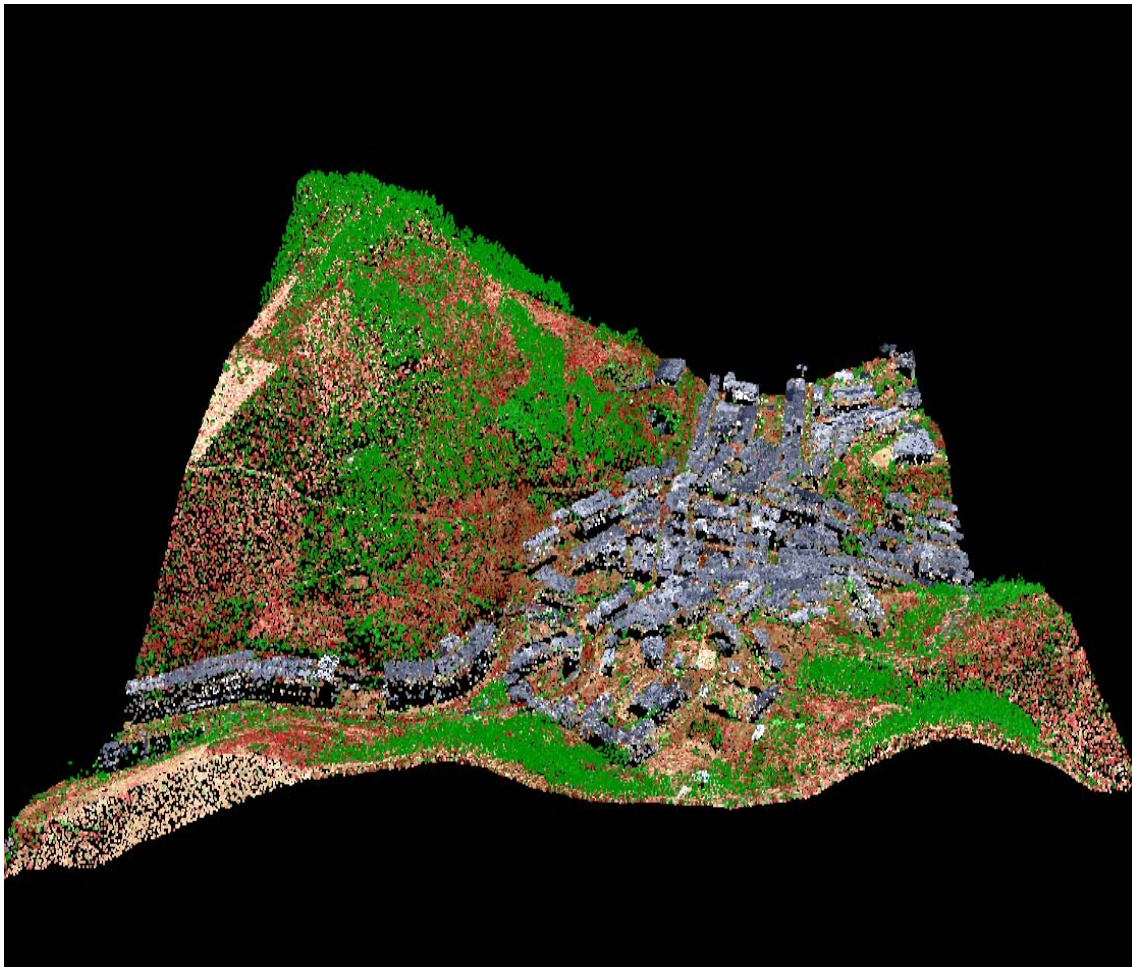


Figura 1: Nube de puntos 3D.

Para llevar a cabo este trabajo se ha utilizado la recopilación de los vuelos realizados en la CAPV desde febrero de 2005 a julio de 2008, constituyendo éste el vuelo LiDAR del 2008 de la CAPV. Estos datos se distribuyen en cuadrículas kilométricas según el PDRF3, que se corresponde con datos georreferenciados, utilizándose el sistema de referencia ETRS89 (European Terrestrial Reference) y el sistema de proyección UTM30N, correspondiente a la proyección UTM, concretamente al huso 30 del hemisferio Norte. Para la determinación de las altitudes obtenidas a partir de las altitudes elipsoidales observadas se hace referencia al modelo de geoide determinado por EGM08_RED NAP.

El formato PDRF3, además de las coordenadas, para cada punto incluye otros valores como son la intensidad, el número de retorno que constituye, el ángulo de escaneo o el valor de la clasificación, entre otros. En la tabla 1 se presentan los distintos campos contemplados con una breve descripción de su contenido. La disponibilidad de esta información posibilita visualizar la nube de puntos por intensidad, clasificación, retornos o en color verdadero, haciendo uso de los valores contemplados en los campos R, G, V. Así por ejemplo, en la figura 1 se visualiza la información combinando los valores de clasificación con los aportados por los canales R, G y V.

Tabla 1: Campos contemplados en el PDRF3

Campo	Descripción
X	Coordenada planimétrica X
Y	Coordenada planimétrica Y
Z	Coordenada altimétrica Z
Intensity (i)	Intensidad del punto laser en el sensor
Return_Number (r)	Número de retorno de ese pulso
Number_of>Returns (n)	Número de retornos detectados para ese pulso
Scan_Direction_flag	Dirección del espejo del escáner
Edge_of_Flight_Line	Borde de línea de vuelo
Classification (c)	Valor de clasificación asignada a ese pulso
Scan_Angle (a)	Ángulo de escaneo
User_Data	Campo a rellenar según necesidad del usuario
Point_Source_ID	Identificador de pasada
GPS_Time	Tiempo GPS
R	Valor asociado al canal Rojo
G	Valor asociado al canal Verde
B	Valor asociado al canal Azul

Los datos LiDAR constituyen una nueva forma de toma de la información distinta a lo utilizado hasta ahora en cartografía, ya que la captura ha pasado de realizarse selectivamente de manera discreta a una captura absolutamente ciega y totalmente masiva, constituyendo cualquier levantamiento un gran volumen de datos de información. En este ámbito, cada uno de los puntos adquiridos con su X, Y, Z necesita de un atributo añadido para reconocer la entidad a la que pertenece y convertirlo en información útil. Esta nueva idea aporta en el ámbito cartográfico una metodología de captura diferente en la que la clasificación constituye la pieza clave. De manera básica, ésta permite distinguir entre puntos pertenecientes al terreno y los puntos que no lo son, posibilitando, en consecuencia, la obtención de manera genérica de modelos digitales de elevación a nivel de suelo, conocidos como Modelos Digitales del Terreno (MDT), constituyendo un producto cartográfico con numerosas aplicaciones en Ingeniería y Ciencias de la Tierra.

El inmenso volumen de los datos que se capturan con este tipo de sistemas justifica la existencia de una gran cantidad de algoritmos de clasificación lo más automáticos posibles. Algunos únicamente centrados en detectar puntos de suelo, otros pensados para la localización de edificios, vías de comunicación o vegetación, pero la mayoría de ellos desarrollados para ser eficientes en un determinado tipo de terreno, ofreciendo limitaciones en condiciones topográficas distintas.

Respecto a los valores de clasificación, la ASPRS, tal y como se puede apreciar en la tabla 2, dependiendo de las versiones del fichero .LAS considerado se contemplan distintas clases para este campo, cuyo número se está incrementando en sucesivas versiones, resultando necesaria, en efecto, la exploración de nuevos algoritmos.

Tabla 2: Clases contempladas para el campo de la clasificación según la ASPRS

Clase	LAS versión 1.4	LAS versión 1.2
0	Nunca clasificado	Nunca clasificado
1	No asignado	No asignado
2	Suelo	Suelo
3	Vegetación baja	Vegetación baja
4	Vegetación media	Vegetación media
5	Vegetación alta	Vegetación alta
6	Edificaciones	Edificaciones
7	Punto bajo (ruido)	Punto bajo (ruido)
8	Reservado	Punto Clave
9	Agua	Agua
10	Vía férrea	Reservado definición ASPRS
11	Superficie pavimentada (carretera)	Reservado definición ASPRS
12	Reservado	Puntos solapados
13	Tendido	13-31 Reservado para la definición de la ASPRS
14	Cable de tendido	
15	Torre transmisora	
16	Conector de tendido	
17	Cubierta de puente	
18	Punto alto (ruido)	
19-63	Reservado	
64-255	Definido por el usuario	

En este contexto, se ha desarrollado una metodología que, a partir del punto de la nube como unidad básica de trabajo, permite clasificar los datos LiDAR con fines cartográficos de zonas de características topográficas y geográficas diversas sin recurrir a los algoritmos clásicos de filtrado y clasificación de nubes de puntos basados, por lo general, en la catalogación de suelo / no suelo y su posterior identificación de los no suelo en otros elementos de la cobertura terrestre, recurriendo a la Minería de Datos. Para llevarlo a cabo se ha hecho uso, además de los datos LiDAR, de otras fuentes de información complementarias que han ayudado a este objetivo. Concretamente, se han utilizado las ortofotografías, próximas a la fecha del vuelo LiDAR, en las bandas roja (R), verde (G), azul (B) e infrarroja (NIR) del 2008 y la Base Topográfica Armonizada (BTA) publicada por el Gobierno Vasco en diciembre del 2011, y que constituye la base cartográfica vectorial con atributos homogeneizada a escala 1/5.000 de la CAPV.

MINERÍA DE DATOS

La Minería o explotación de Datos se puede considerar como una evolución natural de la información tecnológica y la confluencia de varias disciplinas relacionadas como son la inteligencia artificial (artificial intelligence), el aprendizaje automático (machine learning), el reconocimiento de patrones (pattern recognition), el análisis de los datos (data science), la estadística y los sistemas de bases de datos (database management systems) [2]. Es lo que se conoce como Knowledge Discovery in Databases y su objetivo se centra en descubrir patrones dentro de grandes volúmenes de información, pretendiendo aplicarlos a otros datos distintos de manera que se puedan predecir esos mismos comportamientos en otros entornos independientes a los anteriores. Los campos de aplicación de la Minería de Datos son muy variados y heterogéneos, ya que se puede usar desde predicciones médicas hasta temas de seguridad, agricultura, biología, entre otros.

En cualquier entorno que se utilice se distinguen las siguientes fases: exploración de los datos; construcción del modelo que permita la obtención de esos patrones; y, la aplicación y validación del mismo. Hay que señalar que según el tipo de algoritmo que se utiliza se puede abordar el problema de manera supervisada (cuando se usa un conjunto de datos a modo de entrenamiento) o no supervisada (cuando se ajusta el modelo según las observaciones disponibles). A su vez, en cada uno de estos grupos la cantidad de algoritmos de los que se dispone es muy amplia.

En esta investigación se ha utilizado el algoritmo supervisado de Random Forest (RF), por lo que se ha tenido que entrenar parte del conjunto de datos utilizado, considerando 126 hojas (cuadrículas kilométricas) para el entrenamiento (77 %) y 36 para la validación (23 %). El entrenamiento se ha llevado a cabo a partir de la Base Topográfica Armonizada (BTA) a escala 1/5.000, en base a la cual se ha podido determinar si el punto pertenecía a alguna de las siguientes categorías de interés: suelo, edificios, vegetación o vías de comunicación. La muestra se ha seleccionado para que resulte significativa de todo el territorio de la CAPV, pero su número ha quedado restringido por los requerimientos del procesamiento.

Destacar que RF constituye un algoritmo no paramétrico basado en métodos de ensamblado construyendo un modelo predictivo por integración de múltiples modelos, consiguiendo así mejorar el rendimiento de las predicciones independientes. Se trata de una técnica de perturbación-y-combinación diseñada para árboles de decisión, que utiliza alteraciones al azar en el método de aprendizaje para crear los distintos modelos. Normalmente se utilizan dos o más modelos, cada uno con subconjuntos de datos y variables diferentes para combinar las predicciones independiente en un pronóstico único (voting -> el más votado), ofreciendo un mejor resultado que el de cada uno de los subconjuntos de manera individual [3]. En este caso, se han utilizado modelos de 20 árboles.

Para clasificar los puntos LiDAR a través de la Minería de Datos resulta necesario disponer además de un conjunto de variables que posibiliten instaurar las relaciones entre ellas para poder establecer de qué manera cada punto pertenece a una clase determinada. La identificación y obtención de estas variables ha constituido la primera fase del desarrollo expuesto en la implementación. Durante la segunda parte se ha complementado la metodología aplicando el algoritmo elegido, RF, al conjunto de información extraída en el apartado anterior.

IMPLEMENTACIÓN

Para conseguir el objetivo propuesto de clasificar las nubes de puntos LiDAR obviando los métodos más comunes, considerando el punto como unidad de trabajo y haciendo uso de la Minería de Datos, se ha optado por utilizar el lenguaje de programación Python 2.7 [4] en un entorno Linux, por constituir un lenguaje interpretado y multiplataforma con una sintaxis sencilla y poseer licencia de código abierto denominada Python Software Foundation License, compatible con la licencia pública GNU (GNU's Not Unix).

El tratamiento del gran volumen de datos, a modo de puntos, facilitados por la tecnología LiDAR ha sido llevado a cabo a partir de las librerías de Python numpy [5] y pandas (Python Data Analysis Library) [6]. La librería numpy constituye el paquete fundamental para el cálculo científico en el entorno de Python, mientras que pandas permite manejar y analizar fácilmente grandes estructuras de datos, tal y como sucede con la información proporcionada por el tipo de sensores con el que se está trabajando, a modo de matrices.

Además de estas librerías, para la lectura de los datos originales en formato .LAS y su transformación a formato .csv (comma-separated values) se ha considerado el comando las2txt de la librería libLAS [7]. Esta librería ofrece licencia de software libre BSD (Berkeley Software Distribution). Está escrita en C/C++ e incorpora bindings para muchos lenguajes de programación; y, aunque en sus inicios fue desarrollada por Iowa Geological Survey (IGS), actualmente está bajo la protección de OSGEO (Open Source Geospatial Foundation).

Una vez transformados los datos a formato .csv, éstos han sido tratados con utilidades disponibles a partir de QGIS Desktop Wien 2.8.2. [8] para asignar atributos a tablas y unir o recortar unidades cartográficas (ortofotos y cartografía BTA) en función de las cuadrículas kilométricas de los datos LiDAR, para poder así trabajar de manera más rápida.

Para la extracción de información a partir de polígonos o imágenes raster, cálculo de estadísticas y su posterior asignación a los puntos iniciales se ha hecho uso de la librería SAGA, System for Automated Geoscientific Analyses [9], la cual ha sido desarrollada por el Department Physical Geography de Göttingen constituyendo un software libre (Free Open Source Software, FOSS) que dispone de herramientas para análisis científicos digitales de la Tierra [10].

El uso de la librería GDAL/OGR, Geospatial Data Abstraction Library [11], editada en estilo X/MIT (Massachusetts Institute of Technology) bajo licencia Open Source de la Open Source Geospatial Foundation e implementada en lenguajes de programación C++, ha posibilitado la transformación de los datos geoespaciales de formato Esri Shapefile a .csv o de .jpg a .tif, la separación o agregación de la información de las tres bandas contenidas en las ortofotografías RGB, la asignación del sistema de referencia junto con su proyección e incluso la realización de cálculos entre los atributos disponibles, permitiendo, cuando ha sido necesario, la rasterización de imágenes.

Con el objeto de obtener variables con respecto al terreno se ha hecho uso de la librería SPDlib, Sorted Pulse Data Library [12]. Ésta constituye un conjunto de herramientas abiertas para el procesamiento de datos LiDAR, tanto de retorno discreto como de full-waveform, bien capturados desde plataformas aéreas o terrestres. Ha sido desarrollada en C++ por Pete Bunting y colaboradores de la Universidad de Aberystwyth (UK), con bindings para Python e IDL y con licencia GPL (General Public License). Una de las particularidades que presenta es que no utiliza el formato .LAS para procesar los datos y éstos hay que transformarlos al formato .SPD (Sorted Pulse Data) [13]. El flujo de trabajo que se puede desarrollar con esta librería se puede consultar en [14].

Gracias a esta librería se han clasificado los puntos LiDAR en terreno / no terreno según los algoritmos PMF (Progressive Morphological Filter) [15] y MCC (Multiscale Curvature Classification) [16], previa transformación de los datos en formato .LAS a .SPD, eliminación de puntos anómalos y limpieza de la clasificación existente en los datos originales. Finalmente, con cada algoritmo se han generado los modelos digitales tanto a nivel de suelo (MDT) como de superficie (MDS), obteniendo los correspondientes ficheros raster en formato .tif a partir de los cuales se ha obtenido el modelo digital de superficies normalizado (nDSM). Este último modelo se ha realizado a partir de la librería GDAL.

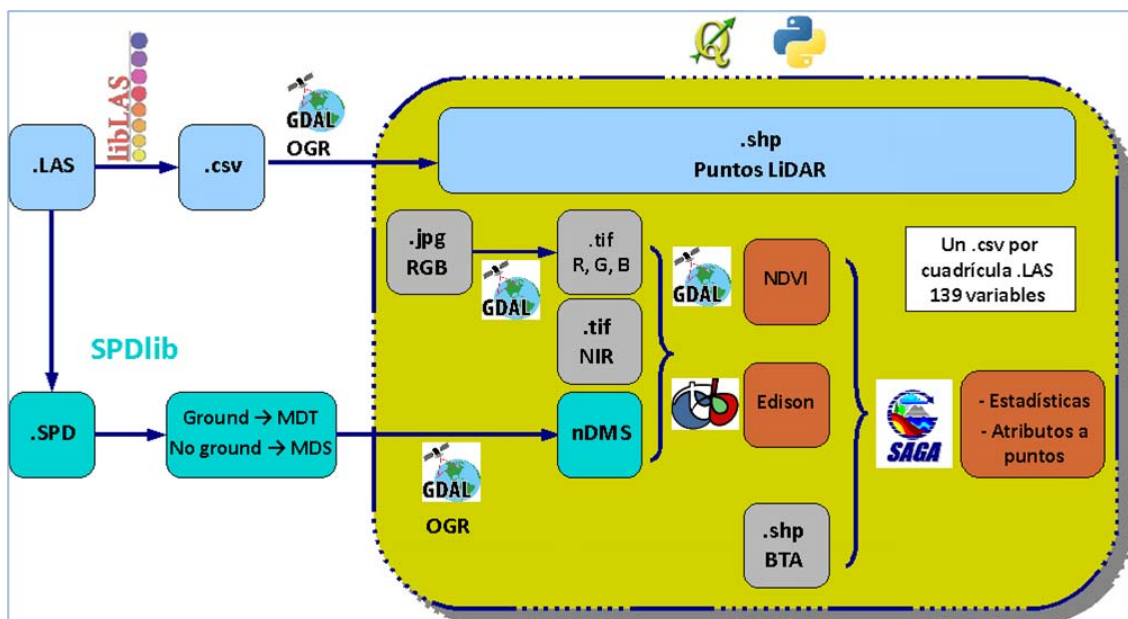


Figura 2: Esquema con las librerías utilizadas.

Para segmentar tanto las ortofotografías originales como los ráster obtenidos a partir de la librería SPDlib se ha considerado del total de opciones disponibles en la librería OTB, Orfeo ToolBox, la que hace referencia a la segmentación de Edison. Esta librería ha sido desarrollada por el Centre National d'Etudes Spatiales (CNES) y es distribuida bajo licencia CeCILL license, constituyendo una agrupación de herramientas para el procesamiento de imágenes de teledetección [17] principalmente, aunque también permite el acceso a datos LiDAR o imágenes aéreas. Está implementada en C++ y dispone de una API (Application Programming Interface) para Python.

A partir del uso de todas estas librerías (figura 2) se ha conseguido obtener un total de 139 variables (figura 3) que han permitido el desarrollo de una metodología para la clasificación de estos datos en el entorno de la Minería de Datos, lo que ha constituido la resolución de la primera fase del procedimiento.

Tipo dato	Variables	Nº variables
LAS	<i>a, i, n, r</i>	4
Ortofotos	<i>R, G, B, NIR</i>	36
	<i>R_min, R_max, R_range, R_mean, R_std, R_Q25, R_Q50, R_Q75</i>	
	<i>G_min, G_max, G_range, G_mean, G_std, G_Q25, G_Q50, G_Q75</i>	
	<i>B_min, B_max, B_range, B_mean, B_std, B_Q25, B_Q50, B_Q75</i>	
	<i>NIR_min, NIR_max, NIR_range, NIR_mean, NIR_std, NIR_Q25, NIR_Q50, NIR_Q75</i>	
Diferencias normalizadas	<i>NIRR, NIRG, NIRB, RG, RB, GB</i>	54
	<i>NIRR_min, NIRR_max, NIRR_range, NIRR_mean, NIRR_std, NIRR_Q25, NIRR_Q50, NIRR_Q75</i>	
	<i>NIRG_min, NIRG_max, NIRG_range, NIRG_mean, NIRG_std, NIRG_Q25, NIRG_Q50, NIRG_Q75</i>	
	<i>NIRB_min, NIRB_max, NIRB_range, NIRB_mean, NIRB_std, NIRB_Q25, NIRB_Q50, NIRB_Q75</i>	
	<i>RG_min, RG_max, RG_range, RG_mean, RG_std, RG_Q25, RG_Q50, RG_Q75</i>	
	<i>RB_min, RB_max, RB_range, RB_mean, RB_std, RB_Q25, RB_Q50, RB_Q75</i>	
	<i>GB_min, GB_max, GB_range, GB_mean, GB_std, GB_Q25, GB_Q50, GB_Q75</i>	
Modelos Digitales	<i>Pmf_t, Pmf_s, Pmf_st, Mcc_t, Mcc_st</i>	45
	<i>Pmf_t_min, Pmf_t_max, Pmf_t_range, Pmf_t_mean, Pmf_t_std, Pmf_t_Q25, Pmf_t_Q50, Pmf_t_Q75</i>	
	<i>Pmf_s_min, Pmf_s_max, Pmf_s_range, Pmf_s_mean, Pmf_s_std, Pmf_s_Q25, Pmf_s_Q50, Pmf_s_Q75</i>	
	<i>Pmf_st_min, Pmf_st_max, Pmf_st_range, Pmf_st_mean, Pmf_st_std, Pmf_st_Q25, Pmf_st_Q50, Pmf_st_Q75</i>	
	<i>Mcc_t_min, Mcc_t_max, Mcc_t_range, Mcc_t_mean, Mcc_t_std, Mcc_t_Q25, Mcc_t_Q50, Mcc_t_Q75</i>	
	<i>Mcc_st_min, Mcc_st_max, Mcc_st_range, Mcc_st_mean, Mcc_st_std, Mcc_st_Q25, Mcc_st_Q50, Mcc_st_Q75</i>	
	TOTAL	

Figura 3: Variables agrupadas según el tipo de dato de procedencia.

Durante la segunda etapa, se han entrenado los datos en base a la información facilitada por la BTA y se ha aplicado el algoritmo RF de Minería de Datos gracias a la librería scikit-learn [18] obteniendo además de la nueva catalogación en base a la predicción derivada del entrenamiento, los indicadores necesarios para su validación. Scikit-learn constituye una librería de código abierto para el aprendizaje automático aplicable en el lenguaje de programación de Python bajo licencia BSD (Berkeley Software Distribution) [19]. Está diseñada para interoperar con las librerías NumPy, SciPy y matplotlib de Python y dispone de algoritmos de clasificación, regresión, clustering o agrupamiento, reducción dimensional, selección de modelos y pre-procesado.

RESULTADOS

Como resultado de la implementación de todas estas librerías bajo el lenguaje de programación Python se ha conseguido clasificar los puntos LiDAR en base a las clases determinadas por la ASPRS para el formato .LAS en su versión 1.2 (tabla 2) obteniendo un fichero .csv en el que además de la información inicial se han guardado las variables calculadas y extraídas de los datos utilizados y los resultados obtenidos de la aplicación del algoritmo RF, fácilmente accesibles a través de QGIS Desktop Wien 2.8.2. para su visualización y comparación.

Como ejemplo del procesamiento descrito en los párrafos anteriores, se ha elegido la cuadrícula 5094793 (figura 4). En ella se puede apreciar a la izquierda el resultado alcanzado tras la aplicación de la metodología desarrollada, visualizando los edificios en color rojo, las carreteras en negro, el suelo en marrón y las áreas de vegetación en verde. A la derecha se muestra la información aportada por la ortofotografía RGB, permitiendo cotejar la solución alcanzada.



Figura 4. Visualización de la cuadrícula 5094793.

CONCLUSIONES

Las conclusiones que aquí se presentan hacen referencia a las herramientas geoespaciales utilizadas, y no a los logros alcanzados en cuanto a la clasificación se refiere, que exceden el objetivo de este documento.

En ese ámbito, hay que señalar que la mayor parte del proyecto se ha realizado con software de dominio público (Open Source Data) y de código abierto bajo Linux, permitiendo de manera eficaz el procesamiento de un gran volumen de datos geoespaciales (algo más de 13 millones de puntos), problema que se ha manifestado como irresoluble con otros softwares y entornos para el tratamiento de Minería de Datos.

Además, cabe remarcar que la utilización del lenguaje de programación Python ha facilitado el acceso a los distintos paquetes y librerías de interés, mencionados en el apartado de la implementación, evitando una programación más ardua. Este lenguaje también ha permitido la integración del conjunto de herramientas en un único entorno evitando la extracción de formatos concretos para procesarlos con otro tipo de programas haciendo posible la consecución del objetivo propuesto de manera más homogénea.

AGRADECIMIENTOS

Este trabajo se ha realizado en su inmensa mayoría con la información geoespacial que el Servicio de Información Territorial, dentro de la Dirección de Planificación Territorial y Urbanismo, del Gobierno Vasco ofrece abiertamente a través de su ftp [20], en la Infraestructura de Datos Espaciales (IDE) de la CAPV.

REFERENCIAS

- [1] ASPRS. (2016). *LASer (LAS) File Format Exchange Activities*, <http://asprs.org/Committee-General/LASer-LAS-File-Format-Exchange-Activities.html>.
- [2] HAN, J., KAMBER, M., PEI, J. (2011). *Data mining: concepts and techniques*. 3rd edition ed. Morgan Kaufmann, USA.
- [3] BREIMAN, L. (2001). *Random forests*. *Machine Learning* 45 (1), 5-32.
- [4] Python 2.7. (2016). <https://www.python.org/download/releases/2.7/>.
- [5] Numpy. (2016). <http://www.numpy.org/>.
- [6] Pandas. (2016). <http://pandas.pydata.org/>.
- [7] libLAS. (2016). <http://www.liblas.org/>.
- [8] QGIS. (2015). <http://qgis.org>.
- [9] SAGA. (2016). <http://www.saga-gis.org/en/index.html>.
- [10] Dept. of Physical Geography, Göttingen. (2015). SAGA, <http://www.saga-gis.org/en/index.html>.
- [11] GDAL. (2016). *Geospatial Data Abstraction Library*. <http://www.gdal.org/>.
- [12] BUNTING, P. (2013). *The Sorted Pulse Data Software Library*.
- [13] BUNTING, P., ARMSTON, J., LUCAS, R.M., CLEWLEY, D. (2013). "Sorted pulse data (SPD) library. Part I: A generic file format for LiDAR data from pulsed laser systems in terrestrial environments". *Computers & Geosciences* 56 (0), pp. 197-206.
- [14] BUNTING, P., ARMSTON, J., CLEWLEY, D., LUCAS, R.M. (2013). "Sorted pulse data (SPD) library—Part II: A processing framework for LiDAR data from pulsed laser systems in terrestrial environments". *Computers & Geosciences* 56 (0), pp. 207-215.
- [15] ZHANG, K., CHEN, S., WHITMAN, D., SHYU, M., YAN, J., ZHANG, C. (2003). "A Progressive Morphological Filter for Removing Nonground Measurements From Airborne LIDAR Data". *IEEE Transactions on Geoscience and Remote Sensing* 41 (4), 872-882.
- [16] EVANS, J.S., HUDAK, A.T. (2007). "A Multiscale Curvature Algorithm for Classifying Discrete Return LiDAR in Forested Environments". *IEEE Transactions on Geoscience and Remote Sensing* 45 (4), 1029-1038.
- [17] CNES. (2016). Orfeo ToolBox, <https://www.orfeo-toolbox.org/>.
- [18] Scikit-learn, (2015). Sklearn.ensemble.RandomForestClassifier, <http://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html>.
- [19] PEDREGOSA, F., VAROQUAUX, G., GRAMFORT, A., MICHEL, V., THIRION, B., GRISEL, O., et al., 2011. "Scikit-learn: Machine learning in Python". *The Journal of Machine Learning Research* 12, pp. 2825-2830.
- [20] GOBIERNO VASCO. "Geoeuskadi. Servicio de descarga FTP". <http://www.geo.euskadi.eus/s69-geodir/es/contenidos/informacion/servicio ftp/es 80/servicio ftp.html>.