Universitat Oberta de Catalunya

uoc.edu

Clasificación automática de objetos astronómicos por fotometría en series históricas recogidas por el Large Synoptic Survey Telescope (LSST)

Alumno: **Luis Enrique Arribas Zapater**
Grado Ingeniería Informática
Inteligencia Artificial

Consultor: Joan M Nuñez Do Rio
Profesor/a responsable de la asignatura : Carles Ventura Royo

Fecha Entrega: 12/06/2019

FICHA DEL TRABAJO FINAL

Título del trabajo:	*Clasificación automática de objetos astronómicos por fotometría en series históricas recogidas por el Large Synoptic Survey Telescope (LSST)*
Nombre del autor:	*Luis Enrique Arribas Zapater*
Nombre del consultor/a:	*Joan M Nuñez Do Rio*
Nombre del PRA:	*Carles Ventura Royo*
Fecha de entrega (mm/aaaa):	06/2019
Titulación::	*Grado Ingeniería Informática*
Área del Trabajo Final:	*Inteligencia Artificial*
Idioma del trabajo:	*Castellano*
Palabras clave	*LSST, aprendizaje automático, Random Forest*

Resumen del Trabajo (máximo 250 palabras):

Este proyecto trata un problema de clasificación de objetos astronómicos a partir de los datos registrados por el telescopio LSST, correspondientes a series históricas de flujo. En primer lugar, presentamos las técnicas de aprendizaje computacional y minería de datos utilizadas en el proyecto. A continuación, se describen los conceptos astronómicos necesarios para el trabajo. Analizamos los datos mediante técnicas de minería de datos y agrupamos las muestras en función de dos de sus características. Transformamos los datos, calculando la magnitud y el color de los objetos. Sometemos a los datos a un proceso de reducción de ruido y a una inferencia bayesiana de sus valores de flujo. Convertimos los datos en series temporales a las que realizamos un proceso de extracción de características. Reducimos las características, mediante sucesivas clasificaciones de forma iterativa, hasta encontrar la dimensionalidad óptima y construimos con dichas características tres clasificadores de 1, 2 y 4 bosques aleatorios. Validamos el modelo y discutimos los resultados. Finalmente se proponen lineas de trabajo para el futuro

Abstract (in English, 250 words or less):

In this project we solve an astronomical objects classification problem from a dataset of flux samples taken by the LSST telescope. Firstly, we show the machine learning and data mining techniques that we are going to use afterwards. Then, we describe some astronomical topics, needed to understand the project. We analyze the data by data minning techniques and we make a clustering over the data based on two of it´s features. We transform the data by the calculation of the magnitude and colour of the objects. We remove noisy samples and we estimate flux values with a bayesian inference. Then, we transform the data in time series and we do a feature extraction processing. Afterwards, the most of the features are eliminated in an iterative selection proccess, done with a random forest, untill the best dimensionality is found. Then, we grow three random forest classifiers using that features. Finally we validate each model with a 5 fold cross validation and we discuss the results. Some future research lines are shown at last.

Índice

Lista de figuras

1 INTRODUCCIÓN

1.1 Contexto y justificación del trabajo

El telescopio LSST [14] (Large Synoptic Survey Telescope), que está actualmente en construcción y se instalará en el Cerro Pachón (Chile), entrará en servicio en 2019 y será el mayor telescopio de luz visible construido por la humanidad hasta la fecha. El diseño del telescopio está orientado para llevar a cabo una observación profunda de una extensa región del cielo del hemisferio sur, obteniendo así, cada pocos días, imágenes precisas de dicha región. Estas medidas de luz quedarán registradas en series históricas, con lo que se dispondrá de una información dinámica de la luz emitida por los objetos celestes observados. Esta observación se prolongará durante 10 años y conducirá a la elaboración del mayor catálogo astronómico nunca realizado. Con ella se pretende profundizar en el conocimiento de cuatro grandes áreas de investigación científica: entender la naturaleza de la materia oscura y la energía oscura, estudiar la estructura y formación de nuestra galaxia, catalogar el sistema solar y explorar el cielo en busca de eventos de interés para la comunidad científica.

Para la elaboración de este catálogo de objetos celestes es necesario clasificar los objetos detectados por el telescopio (se espera registrar una colección de 3,5 Millones de objetos aproximadamente), para lo cual el equipo científico del proyecto creó un reto abierto a científicos de datos de todo el mundo; el *Photometric LSST Astronomical Time-series Classification Challenge (*PLAsTiCC*).*

PLAsTiCC es un desafío de datos abiertos presentado en la comunidad científica en linea Kaggle[1] para la clasificación de objetos astronómicos y se

[1] Comunidad científica en línea, dedicada a la ciencia de datos y al aprendizaje computacional. https://en.wikipedia.org/wiki/Kaggle

enmarca en el contexto de la preparación para las observaciones que realizará el LSST.

Los datos abiertos disponibles para los participantes, que son los que se usarán en este trabajo, son dos conjuntos denominados *lcdata* y *metadata*. El primero de estos conjuntos representa un observaciones de flujo luminoso en instantes de tiempo determinados, de forma que de un grupo de observaciones asociadas a un objeto se obtiene el comportamiento en el tiempo del objeto en forma de *curva de luz*. Así, cada una de las clases tiene una *curva de luz* característica para clasificarlo. En m*etadata* se encuentran datos del objeto invariantes en el tiempo, relacionadas con su posición, distancia y la clase a la que pertenece realmente el objeto.

Este proyecto está motivado por la necesidad de dar sentido a la enorme cantidad de datos que recogerá el instrumento una vez entre en servicio. Así, es imprescindible que dichos datos se conviertan en información precisa del universo observable y contribuyan a la consecución de los cuatro grandes objetivos científicos del LSST que se han mencionado anteriormente. Esta tarea se lleva a cabo mediante técnicas de aprendizaje computacional (*machine learnig*) y minería de datos (*Data mining*).

1.2 Objetivos del trabajo

1.2.1 Objetivos generales

1. Resolver un problema de clasificación de objetos astronómicos mediante técnicas de aprendizaje computacional y minería de datos.

2. Evaluar el modelo de clasificación mediante métricas que se ajusten al problema

1.2.2 Objetivos específicos

1. Análisis y preparación del conjunto de datos para su estudio.

2. Transformar los datos en series temporales y extraer nuevos atributos de ellas.

3. Diseñar un modelo de clasificación automática supervisada sobre el conjunto de datos resultante.

4. Racionalizar los recursos de cómputo disponibles para el tratamiento de problemas de inteligencia artificial con grandes volúmenes de datos, atendiendo el compromiso entre precisión y eficiencia.

5. Implementar un *framewok* de aprendizaje computacional que cumpla los objetivos 2 a 4

1.3 Enfoque y método seguido

Los datos, tal y como son proporcionados por los promotores del proyecto, no pueden ser utilizados para la clasificación, ya que primero es necesario extraer de ellos sus series temporales y después extraer los atributos de dichas series. Se desconoce *a priori* la cantidad de atributos que tendrá el conjunto de datos resultante. Por otro lado, el telescopio recoge observaciones en 6 bandas de frecuencia, por lo que cada objeto tendrá seis series temporales.

Ya que los atributos se extraerán de cada serie temporal, y cada objeto tiene seis series, se espera un número elevado de ellos en el conjunto de datos resultante de dicha extracción. Según Gao et al. [2], aplicando los algoritmos basados en bosques aleatorios no es necesario considerar la dimensionalidad de los datos, ya que el algoritmo selecciona los atributos representativos

usados para la clasificación. Esta es una característica muy importante para el presente trabajo, debido al desconocimiento de la dimensionalidad final del conjunto de datos.

Otras ventajas de los bosques aleatorios mencionadas por Gao et al. interesantes para el trabajo son:

- Genera una estimación interna del error de generalización a medida que la construcción de un bosque va progresando.
- Maneja un gran número de variables que permiten ajustar el algoritmo al problema.
- Tiene un rápido aprendizaje.

En base en estas consideraciones, se considera resolver la clasificación implementando un modelo basado en *bosques aleatorios*.

1.4 Planificación del Trabajo

1.4.1 Distribución temporal del trabajo.

	Tarea	Inicio	Fin	Días
1	Definición contenidos	20/2/19	4/3/19	12
1.1	Lectura de fuentes (TFG, TFM, artículos)	20/2/19	27/2/19	7
1.2	Consensuar TFG	27/2/19	2/3/19	3
1.3	Definición	2/3/19	3/3/19	1
1.4	Redactar Propuesta de TFG	3/3/19	4/3/19	1
2	Plan de trabajo	5/3/19	18/3/19	13
2.1	Contexto y justificación del trabajo	5/3/19	7/3/19	2
2.2	Objetivos del trabajo	7/3/19	9/3/19	2
2.3	Enfoque y metodo de trabajo	9/3/19	11/3/19	2
2.4	Planificación del trabajo	11/3/19	12/3/19	1
2.5	Redacción de la memoria capítulo 1	12/3/19	18/3/19	6
3	Fase 1 - Desarrollo	19/3/19	22/4/19	34
3.1	Análisis preliminar de los datos	19/3/19	29/3/19	10
3.1.1	Análisis metadata	19/3/19	22/3/19	3
3.1.2	Estudio del problema Z	22/3/19	29/3/19	7
3.2	Transformación de los datos	30/3/19	22/4/19	23
3.2.1	Corrección de medidas de Flujo	30/3/19	3/4/19	4
3.2.2	Extracción de características de series históricas	3/4/19	8/4/19	5
3.2.3	Análisis de correlación	8/4/19	9/4/19	1
3.2.4	Partición del conjunto	9/4/19	12/4/19	3
3.2.5	Clasificador RF preliminar. Primeros resultados	12/4/19	16/4/19	4
3.2.6	Actualización de la planificación	16/4/19	17/4/19	1
3.2.7	Elaboración informe fase 1	17/4/19	19/4/19	2
3.2.8	Redacción de la memoria capítulo 2 - 3	19/4/19	22/4/19	3
4	Fase 2 - Desarrollo	23/4/19	20/5/19	27
4.1	Evaluación y mejoras del modelo	23/4/19	11/5/19	18
4.2	Reducción de dimensionalidad	11/5/19	13/5/19	2
4.3	Evaluación final y ajuste fino	13/5/19	18/5/19	5
4.4	Elaboración informe fase 2	18/5/19	20/5/19	2
6	Redacción de la memoria capítulo 3 - 6	20/5/19	27/5/19	7
6	Elaboración presentación	21/5/19	12/6/19	22
7	Defensa del trabajo	13/6/19	24/6/19	11

1.4.2 Recursos necesarios

- Anaconda-navigator 1.9.2: distribución abierta para científicos de datos y profesionales de TI de los lenguajes Python y R. Integra las aplicaciones que usaremos (RStudio y Spyder) entre otras orientadas a la ciencia de datos.

- AstroML: módulo ML para Python orientado al trabajo con datos astronómicos en Python.

- Scikit_learn: conjunto de herramientas científicas para Python.

- Pandas: librería para análisis de datos en Python.

- Matplotlib: librería de representación gráfica para Python.

- numpy : paquete de computación científica para Python.

- Cesium: librería para extracción de características de series temporales para Python

- Pandas: estructuras de datos y herramientas de análisis de datos para Python

- Astropy: librería construida sobre pandas para uso en astrofísica. Permite operaciones relacionales con tablas

- feets: librería para extracción de características de series temporales

1.5 Breve sumario de productos obtenidos

Los productos resultantes del proyectos son:

- un algoritmo de clasificación implementado en Python. Este algoritmo realizará, previa a la clasificación, la limpieza y transformación de los datos, así como la extracción de nuevos atributos
- una memoria explicativa del trabajo donde se recojan todas las actividades realizadas y las conclusiones del proyecto
- una presentación multimedia en la aplicación *Presenta* donde se exponga el proyecto

1.6 Breve descripción de los otros capítulos de la memoria

En el segundo capítulo se describen las técnicas de aprendizaje computacional de las que se hará uso más adelante. Primero se estudian los *bosques aleatorios* de Breiman [3], para lo cual, se deben discutir primero los árboles clasificadores. A continuación se trata el agrupamiento con *k-medias* [10], usado durante el tratamiento de los datos previo a la clasificación. Se discute la *validación cruzada* para eliminar dependencias en los datos y evaluar la bondad del modelo y finalmente se describen la *pérdida logarítmica* y *accuracy* como métricas evaluadoras.

El tercer capítulo describe el contexto astronómico del problema de aprendizaje automático. Se discuten conceptos necesarios como el principio cosmológico, y la ley de Hubble-Lamaître [11]. Se describen los fenómenos, representados como atributos, que se encuentran en los datos, como la extinción, la magnitud estelar y la medida de distancias a escalas cósmicas. Finalmente, este capítulo tratará las dos formas de observación de la luz emitida por las estrellas y sus diferencias, con el objeto de mejorar la comprensión de los datos.

En el cuarto capítulo se discute el modelo clasificador a través de todas sus etapas. Se describen primero los datos, relacionándolos en los conceptos

vistos en el capítulo anterior. Se realiza un examen del conjunto de datos mediante técnicas de minería de datos. Seguidamente, se discuten los procesos de preparación a los que son sometidos estos datos, la extracción de nuevos atributos y la clasificación mediante bosques aleatorios. Finalmente, se discuten los resultados del modelo.

El quinto capítulo recoge la conclusiones de este trabajo. En él, se plantea qué objetivos se han alcanzado y cuales no. Se estudian las lecciones aprendidas de todo el proceso de elaboración del proyecto y se analiza la metodología empleada, exponiendo qué ha funcionado, qué no lo ha hecho y cuales son las causas. Se analiza la planificación temporal y grado de cumplimiento del cronograma, identificando los errores y logros de dicha planificación. A este respecto, se proponen una serie de mejoras a tener en cuenta en el futuro. Por último se tratan las cuestiones que no han podido ser objeto del trabajo, bien por quedar fuera de su alcance, o por no haberse cumplido los objetivos temporales, y que son interesantes como líneas de trabajo para el futuro.

2. Algoritmos y técnicas utilizadas

2.1 Árboles de decisión

Un *árbol de decisión* es un algoritmo de clasificación en el que se recorre un grafo de estructura arborescente, partiendo de un nodo raíz, visitando los nodos de decisión y terminando en un nodo hoja, siendo el nodo hoja la etiqueta correspondiente a la clase del elemento clasificado.

Sea *D* un conjunto de datos con *e* elementos y *c* características y *a* un árbol de decisión, tales que

$$a = \left\{ n_0,\ n_{11},\ n_{12} \ldots\ n_{1j},\ n_{21},\ n_{22} \ldots\ n_{j\ldots k} \right\} \text{ y } e = \left\{ c_1,\ c_2 \ldots\ c_n \right\}$$

En cada nodo de decisión *n* se contrasta una única característica c_i con un umbral u_n predeterminado. En función de este contraste, se continúa recorriendo *a* hacia un nodo hijo o hacia otro, tal como se ilustra en la Figura 1. Así, se recorre completamente *a* hasta llegar a un nodo hoja, marcado con la etiqueta que corresponde a la clase que se asigna a e_i.

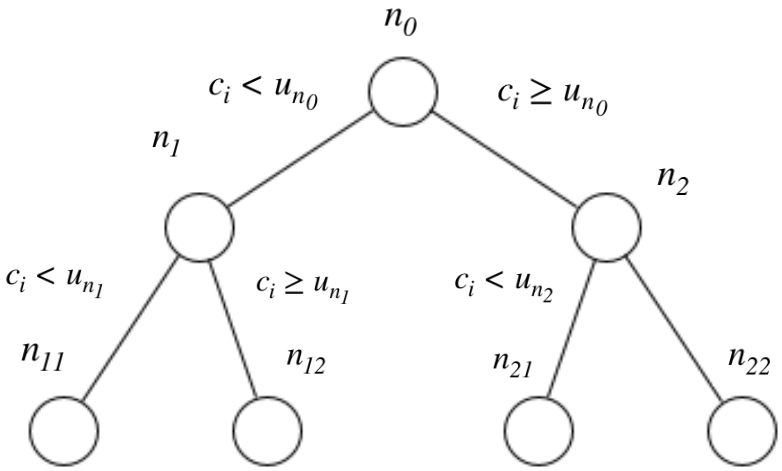


Figura 1. Árbol de decisión

2.1.1 Árbol mínimo

Para la construcción del árbol de decisión de menor profundidad posible, en cada nodo se selecciona c_n de forma que se consiga la mejor partición posible de *D*. Para determinar la bondad de dicha partición se utiliza, en el modelo clasificador presentado en este trabajo, el criterio de *Impureza de Gini:*

$$Impureza_G(p) = 1 - \sum_{1}^{n} p^2$$

donde *p* es la probabilidad de escoger aleatoriamente un elemento con la etiqueta *i* en el subconjunto, tal que:

$$i \in \{ 1, 2, .., n \}$$

Otra métrica para determinar la bondad de la partición de cada nodo durante la clasificación se denomina *ganancia de información y está* basada en la entropía de Shannon:

$$Ganancia\ de\ Información(p) = - \sum_{1}^{n} p_i \log_2 p_{i_i}$$

2.2 Bosques aleatorios

2.2.1 introducción

Los *bosques aleatorios* [3], es una técnica basada en la construcción de un conjunto de *árboles de decisión*, de manera que cada árbol clasifica los elementos del conjunto de datos, y para ello utiliza un conjunto reducido de atributos, siendo estos conjuntos distintos entre los diferentes árboles. Así, terminada la clasificación de todos los árboles, se realiza una votación, de

manera que el resultado más votado es la etiqueta elegida para cada elemento del conjunto.

En palabras del propio Breiman: “Un *bosque aleatorio* es un clasificador compuesto por una colección de clasificadores de estructura arbórea $\left\{ h(x, \Theta_1), k = 1, \dots \right\}$ donde $\{ \Theta \}$ son vectores aleatorios independientes e idénticamente distribuidos y donde cada árbol emite un voto para la clase más popular para una determinada entrada ***x***.” [3]

2.2.2 *Bagging*

El *bagging* consiste en combinar diferentes modelos clasificadores para mejorar la precisión de un clasificador. Por tanto, dado que los *bosques aleatorios* combinan múltiples árboles clasificadores, esta es una técnica basada en *bagging*.

2.2.3 *Bootstraping*

Dado un conjunto de datos *D* con *s* elementos (filas) y *m* atributos (columnas). El *bootstraping* consiste en construir un nuevo conjunto de datos *B,* con *s* elementos, tomando aleatoriamente con reemplazo elementos de *D*. Al realizarse una selección con reemplazo, parte de los elementos de *B* estarán repetidos y parte de los elementos de *D* no se hallarán presentes en B. A modo de ejemplo

$$D = \begin{bmatrix} s_{A1} & s_{B1} & s_{C1} & s_{D1} \\ s_{A2} & s_{B2} & s_{C2} & s_{D2} \\ s_{A3} & s_{B3} & s_{C3} & s_{D3} \\ s_{A4} & s_{B4} & s_{C4} & s_{D4} \end{bmatrix} \quad B_i = \begin{bmatrix} s_{A1} & s_{B1} & s_{C1} & s_{D1} \\ s_{A2} & s_{B2} & s_{C2} & s_{D2} \\ s_{A3} & s_{B3} & s_{C3} & s_{D3} \\ s_{A2} & s_{B2} & s_{C2} & s_{D2} \end{bmatrix}$$

donde S_{VW} representa la característica *V* de la *W-esima* muestra *S*. Nótese que la segunda y cuarta fila de *B* son iguales y que no se ha incluido en B la cuarta fila de *D*. Así, se construyen un número *i* de conjuntos de datos mediante el *bootstrap*. Empíricamente se demuestra que aproximadamente el 30% de las filas de *D* quedan fuera de B_i en cada *i*-esimo *bootstraping*.

2.2.4 Selección aleatoria de características y construcción del bosque

Para cada uno de los conjuntos B_i se seleccionan aleatoriamente un número arbitrario de características. Siguiendo con el ejemplo anterior un posible conjunto

$$C_i = \begin{bmatrix} s_{A1} & s_{C1} \\ s_{A2} & s_{C2} \\ s_{A3} & s_{C3} \\ s_{A2} & s_{C2} \end{bmatrix}$$

se obtendría eliminando la segunda y tercera característica de B_i. A continuación, se construyen *i* árboles *t* tales que cada t_i se entrena con C_i, de forma que

$$t_i = \delta(C_i)$$

donde δ es la función de construcción del árbol de decisión sobre el conjunto de entrenamiento C_i. Este proceso, resulta en una colección de árboles distintos entre sí, pero construidos todos a partir de *D*. Se denomina a esta colección *bosque aleatorio* F, donde $F = \{t_1, t_2, \ldots, t_i\}$

2.2.5 Clasificación con bosques aleatorios

Cada una de las muestras es introducida en cada uno de los árboles del bosque aleatorio y se genera una etiqueta, correspondiente a la clase predicha por el árbol para la muestra. Así para cada árbol t_i y para cada muestra s_j se obtiene la etiqueta l_{ij}. Finalmente, se asigna a cada muestra la etiqueta correspondiente a la moda del conjunto formado por $\left\{ l_{1j},\ l_{2j}, \ldots,\ l_{ij} \right\}$. Este proceso está representado en la Figura 2.

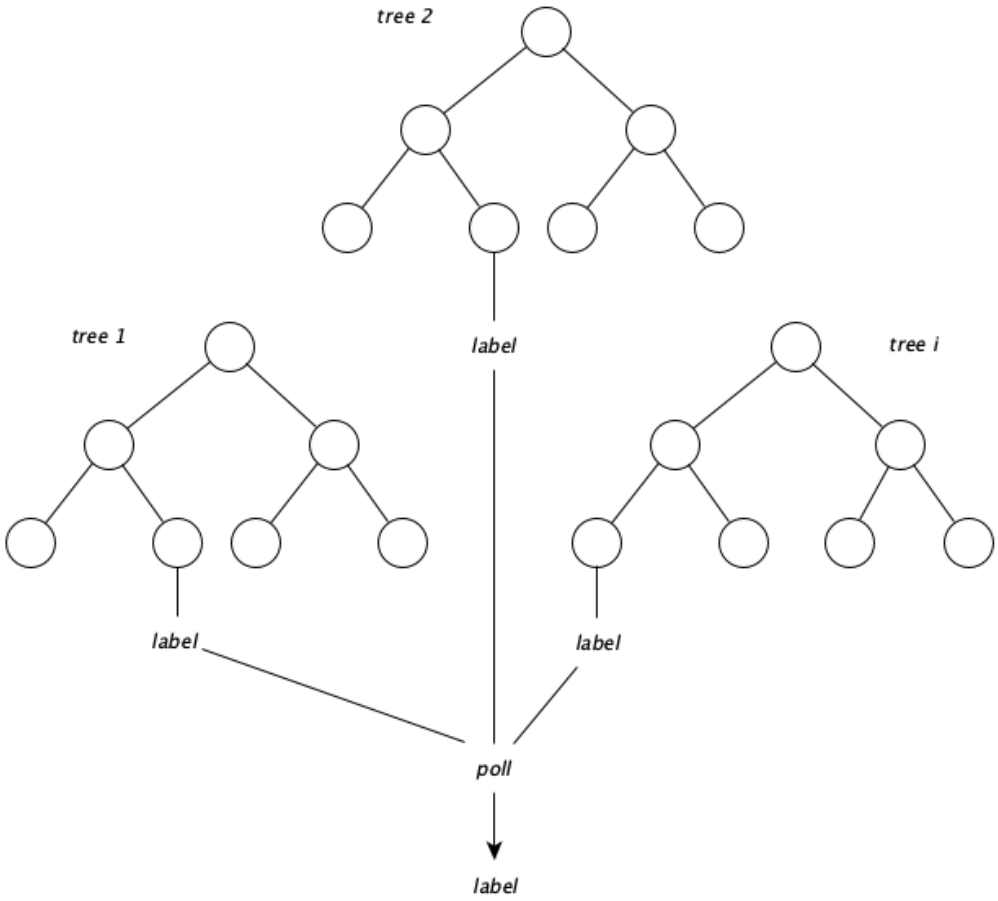


Figura 2. Clasificación con un Bosque Aleatorio. La misma muestra se evalúa en varios árboles y el valor más repetido es asignado como etiqueta.

2.2.6 Reducción de características

El exceso de atributos en los *árboles de decisión* produce un efecto denominado sobreajuste, u *overfitting* en inglés, que tiene lugar cuando un algoritmo clasificador intenta ajustarse en exceso al conjunto de entrenamiento, tomando ciertas singularidades de algunos elementos del conjunto de entrenamiento como generalidades, extrapolables a la totalidad de los datos. En conjuntos de datos donde existe gran número de atributos, conteniendo cada uno una pequeña parte de información, los árboles de decisión se comportan apenas ligeramente mejor que una selección aleatoria [3]

Por lo tanto, para mejorar el rendimiento de un bosque aleatorio, un parámetro fundamental es la cantidad de atributos que se utilizan en cada árbol de decisión. Para esto, típicamente se divide el conjunto de atributos *Q,* tomando para cada árbol t_i un conjunto de atributos q_i tales que:

$$\left|q_i\right| = \left|Q\right|^{\frac{1}{2}} \qquad \text{o bien} \qquad \left|q_i\right| = \log_2\left|Q\right|$$

donde las dos barras verticales representan la cardinalidad del conjunto.

2.2.7 Ventajas y desventajas de los Bosques Aleatorios

Limitando la cantidad de características en cada *árbol* del *bosque* el modelo no sobreajusta si crece el numero de atributos. Además esto permite manejar conjuntos de datos con grandes dimensionalidades, ya que no está basado en el cálculo de distancias. Por otro lado, en conjuntos de datos muy ruidosos, la clasificación pierde precisión rápidamente, debido a la naturaleza de los *árboles de decisión*, que propagan el ruido hacia los nodos inferiores. Así, cierto ruido es mitigado por el bosque al producirse la votación, pero a partir de

cierto umbral, la cantidad de aciertos no llega a imponerse al ruido en la votación.

2.3 K-medias.

El algoritmo de *k-medias* (k-*means*), propuesto por Lloyd en 1957 [10], es una técnica de agrupación (*clustering*) en la que un conjunto de datos se agrupa en un número arbitrario *k* de grupos. La pertenencia de cada elemento del conjunto de datos a uno u otro grupo viene determinada por la distancia más corta de este al centroide (promedio) de cada uno de los grupos.

Sean un conjunto *M* de *k* centroides $\{m_1, m_2, \ldots, m_k\}$, donde $k \in \mathbb{N}$; sea un conjunto de elementos $D = \{d_1, d_2, \ldots, d_n\}$ y sean $\{C_{m_1}, C_{m2}, \ldots, C_{mk}\}$ *k* subconjuntos de D, tales que:

$$C_{m_1} \cup C_{m2} \cup, \ldots, \cup C_{mk} = D \qquad \text{y} \qquad C_{mi} \cap C_{mj} = \varnothing, \ \forall\, i, j$$

El algoritmo se desarrolla en dos pasos:

1- se calcula la distancia de cada d_i a cada m_j, etiquetándose a d_i con el m_j que se encuentre más próximo.

2- Una vez que todas las observaciones d_i están etiquetadas se recalculan los *k* centroides:

$$m_j = \frac{1}{\left| C_{m_j} \right|} \sum d_i \in C_{m_j}$$

donde las barras paralelas representan la cardinalidad del subconjunto. El algoritmo se detiene al producirse la convergencia. Se dice que el algoritmo converge cuando entre una iteración y la siguiente no ha habido ningún cambio en la pertenencia a un C_{mj} de ningún d_i

La selección de los centroides iniciales, puesto que no existe anteriormente a la primera iteración ningún d_i etiquetado, se realiza de forma aleatoria. En la Figura 3 se ilustra el agrupamiento por el método de *k-medias*, con *k*=2

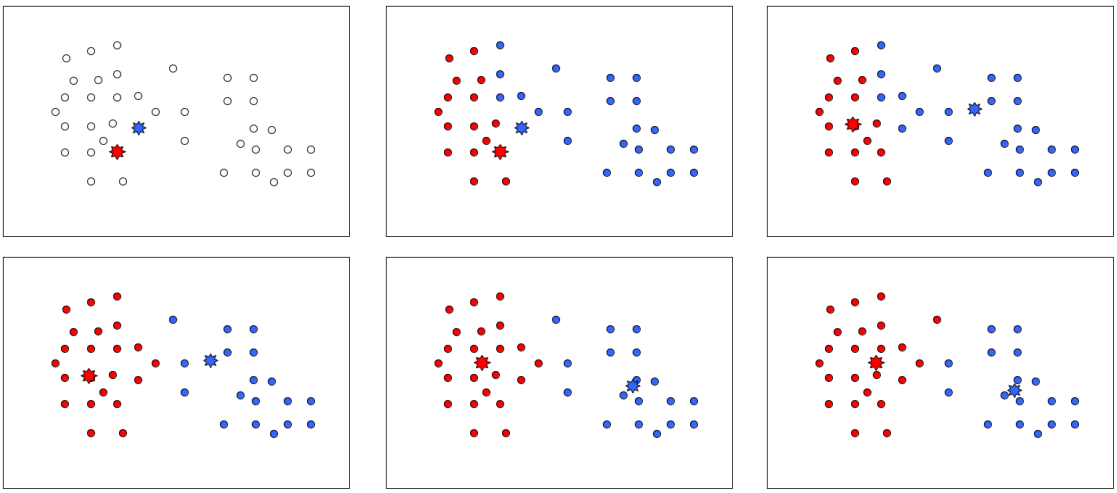


Figura 3. Ejecución paso a paso del algoritmo k-medias. Arriba a la izquierda, como paso inicial, se escogen dos centroides aleatoriamente. Se calculan las pertenencias de cada elemento y se recalculan los centroides iterativamente hasta alcanzar un estado de convergencia, en el que no tienen lugar más cambios de pertenencia a las agrupaciones.

3. Contexto astronómico

3.1 Principio cosmológico

El universo, según el principio cosmológico, es isotrópico y homogéneo. Así, este presenta la misma composición y propiedades físicas en todas las direcciones, siempre que esta observación se realice en órdenes de magnitud suficientemente grandes. Para este trabajo, la asunción de este principio, permite descartar características de los datos relacionadas con la posición de cada objeto.

3.2 Paralaje y distancia

En escalas astronómicas la medida lineal de distancia es el *parsec,* abreviado *pc*, y corresponde al paralaje de 1 segundo de arco. El paralaje es un técnica astronómica de cálculo de distancias fundamentada en el hecho de que la tierra tiene un desplazamiento debido a su órbita alrededor del sol. Mediante la observación de su desplazamiento, realizada desde dos posiciones opuestas de la órbita terrestre puede determinarse la distancia hasta un objeto. Diremos que un objeto se encuentra a *1 pc* de distancia cuando varía su posición en 1" de arco al completar la tierra la mitad de su órbita y se define el *parsec* como:

$$1\ pc = 206265\ ua = 3{,}2616\ años\ luz = 3{,}0857 \times 10^{16}\ m$$

En la Figura 4 se representa la técnica del paralaje de forma esquemática.

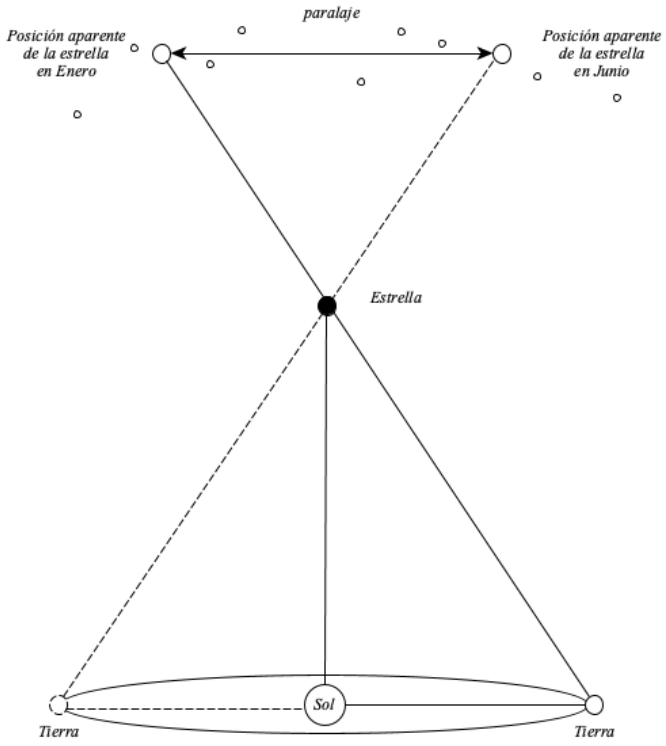


Figura 4. Representación del paralaje

3.3 Luminosidad y flujo

La *luminosidad* de un objeto se define como la cantidad total de energía radiada en todas las direcciones y en todas las frecuencias por unidad de tiempo. Es, por tanto, una medida de potencia (energía por unidad tiempo) y se mide en Julios/segundo. Si se encerrase la fuente de luz con una esfera imaginaria de radio *d*, entonces, en una determinada superficie de la esfera *s* se recibiría en un tiempo *t* una cantidad de energía, *e.* De manera, si se ampliase el radio de dicha esfera, por el principio de conservación de la energía, la misma energía debería repartirse en mayor superficie. Esta disminución de *e* en función de la distancia *d* se define como:

$$e = L\frac{1}{4\pi d^2}t$$

donde *L* es la *luminosidad* del objeto. Así, se define el *flujo, F,* a través de una superficie como la cantidad de energía que la atraviesa por unidad de tiempo

$$\frac{e}{t} = L\frac{1}{4\pi d^2} \qquad \text{o bien} \qquad F = L\frac{1}{4\pi d^2}$$

Los datos con los que se cuenta para este proyecto son simulaciones de medidas de flujo y tiempo registradas por el LSST.

3.4 Ley de Hubble-Lemaître y desplazamiento al rojo

Cuando una fuente de luz se aleja de un observador a una velocidad relativista[2], la luz percibida por el observador experimenta un enrojecimiento, o *desplazamiento al rojo,* tanto mayor, cuanto mayor es la velocidad del objeto. La Ley de Hubble–Lemaître [11] establece que el desplazamiento al rojo de una galaxia es proporcional a la distancia hasta el punto de observación. Así, debido a la naturaleza expansiva del universo, el desplazamiento al rojo, denominado *Z,* es utilizado en astrofísica para estimar la distancia hasta los objetos observados.

3.5 Espectrometría

La luz proyectada sobre un prisma con determinadas propiedades ópticas es descompuesta en los colores fundamentales que la componen. Cada color corresponde a una franja de longitudes de onda y al ser proyectados desde el prisma se presentan de forma adyacente, ordenados de menor a mayor longitud de onda. En dicha proyección, denominada *espectro de absorción*, aparecen lineas oscuras características (Figura 5) que corresponden a la

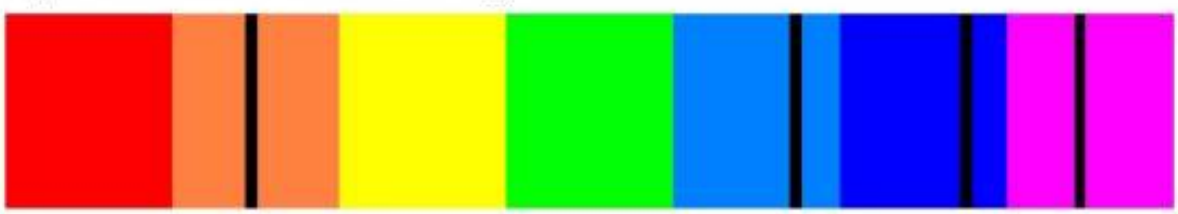


Figura 5. Espectro de absorción del hidrógeno. *Fuente http://www3.gobiernodecanarias.org*

[2] Velocidad que representa un porcentaje significativo de la velocidad de la luz

estructura atómica de la materia atravesada por la luz, de forma que objetos con distintas composiciones presentan lineas distintas.

El espectro de absorción de cada elemento es el mismo en todo el universo, con la particularidad de que, debido al *desplazamiento al rojo*, los objetos que se alejan presentan sus lineas oscuras desplazadas hacia el rojo. Si la composición del universo es la misma en cualquier lugar, el espectro de absorción de un elemento químico conocido y observado en la tierra, debería ser igual que el del mismo elemento observado a grandes distancias. Sin embargo, las líneas espectrales de objetos lejanos presentan un desplazamiento hacia el rojo en el espectro. Este desplazamiento es proporcional a su velocidad y a su distancia, según la ley de Hubble-Lemaître [11]. Así, la espectrometría da una medida precisa (en escala cósmica) de la distancia.

La espectrometría es una técnica que requiere largos tiempos de observación para poder integrar luz suficiente que permita la identificación de las lineas espectrales, sobre todo de objetos muy débiles. Además, los objetos de brillo variable, como *novas* o *supernovas*, presentan grandes dificultades para su análisis por esta técnica, ya que la espectrometría es una técnica que muestra información de forma estática y no tiene en cuenta la dimensión temporal.

3.6 Fotometría

La fotometría trata de estudiar la luz emitida, no por un objeto, sino por miles de ellos simultáneamente. Se realizan observaciones de la bóveda celeste, que registran flujo y tiempo, de forma que se obtiene información dinámica del area observada en forma de series temporales. Mediante esta técnica se genera un mapa del area observada, donde se identifican las fuentes luminosas y sus variaciones de flujo en el tiempo. En lugar de ser descompuesta como en la espectrometría, la luz es filtrada por bandas de frecuencia, correspondientes a colores, para registrar dichas variaciones temporales en cada banda.

La fotometría permite aproximaciones (no siempre precisas) al desplazamiento al rojo de los objetos. Gracias a su capacidad de registrar eventos en la dimensión temporal es una buena herramienta para detección de ciertos tipos de objetos indetectables por espectrometría. En este proyecto los datos usados para la clasificación son registrados por fotometría.

3.7 Curvas de luz

Se denomina comúnmente en astronomía *curva de luz* a la gráfica del flujo registrado de un objeto en el tiempo. Un objeto tienen varias *curvas de luz* para el mismo periodo de tiempo, correspondiendo a las bandas de frecuencia captadas por el telescopio. En el caso del telescopio LSST, estas bandas son seis *(u, g, r, i, z, Y)* y cubren las longitudes de onda desde 4.000Å[3] hasta los 11.000Å. En la Figura 6 se muestra la curva de luz correspondiente a un objeto del conjunto de datos.

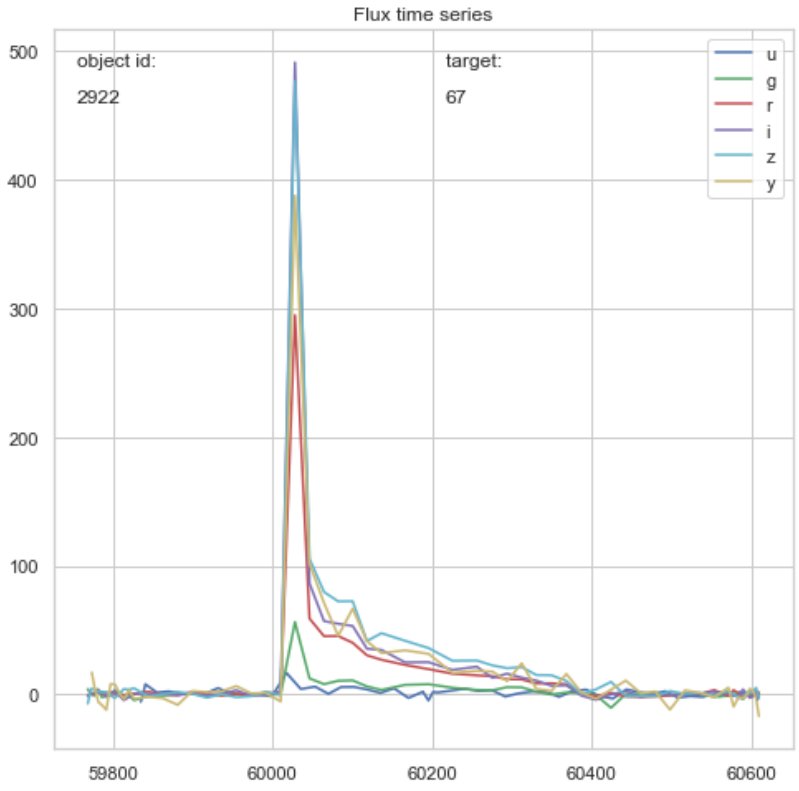


Figura 6. Curva de luz

[3] 1 Å = 10^{-10} m. Léase “Ángstrom”

3.8 Extinción

El interior de las galaxias presenta extensas áreas pobladas de polvo y gas. Desde la tierra, la luz emitida por los objetos celestes sufre una determinada atenuación y enrojecimiento debido al polvo y al gas de la Vía Láctea. Este enrojecimiento no debe confundirse con el *desplazamiento al rojo,* sino que es la atenuación que sufren las longitudes de onda mayores al atravesar el medio interestelar. Esta extinción se encuentra previamente corregida en los datos, por lo que no se requiere ningún preprocesamiento en ese sentido. Baste con saberse que existe y ha sido tenida en cuenta.

3.9 Magnitud aparente, magnitud absoluta y módulo de distancia

El brillo observable por el ojo humano de una estrella se denomina *magnitud*. Por razones históricas las estrellas de primera magnitud son las más brillantes del firmamento, mientras que a mayor grado de magnitud, menor brillo se percibe. William Herschel (1782-1871) calculó que la intensidad luminosa de la primera magnitud es cien veces superior a la sexta, de modo que, para 5 grados de diferencia, el brillo variaba 100 veces, de manera que la razón entre brillo y *magnitud* es $100^{\frac{\Delta m}{5}}$.

Esta *magnitud*, percibida desde la tierra, es denominada *magnitud aparente m,* depende de la distancia del observador al objeto y se define como:

$$m = -2.5\, log_{10} f$$

donde *f* = flujo. La magnitud absoluta *M* de un objeto es la *magnitud aparente* que tendría si se encontrase a 10 pc del punto de observación y se define:

$$M = m + 5\, log_{10} r$$

donde *r* = distancia. El *módulo de distancia (m-M)* es la diferencia entre la *magnitud aparente* y la *magnitud absoluta* de un objeto. Dado que la *magnitud*

es una medida logarítmica, el *módulo de distancia* es una herramienta muy útil para trabajar con distancias a escalas cósmicas, puesto que permite trabajar con distancias de objetos muy alejados entre sí y se define:

$$(m - M) = -5\, log_{10}\frac{r}{10}$$

Calculada la distancia de un objeto luminoso puede predecirse la *magnitud absoluta* de un objeto. Conocido el módulo de distancia, la magnitud absoluta de una estrella se relaciona con el flujo así:

$$M = -2.5 \log_{10} f - (m - M)$$

Los datos se componen de series temporales de flujo, por lo que esta ecuación permitirá, más adelante, calcular la magnitud absoluta de los objetos del conjunto de datos.

3.10 Estrellas variables

Para estudiar la composición de las clases del conjunto de datos es necesario tener en cuenta que el telescopio realiza observaciones por fotometría (ver sección 3.6), registrando curvas de luz (descritas en la Sección 3.7). Así, las estrellas que presentan una variación de su flujo, o que cambian de magnitud con el transcurso del tiempo, son denominadas *estrellas variables*. El origen de esta variabilidad puede ser intrínseco, dadas las propiedades físicas del objeto, o extrínseco (otro cuerpo bloquea parte de la luz). Dentro de estas dos grandes categorías existen otras en función del origen y la naturaleza de la variación.

Las variabilidades de origen pulsante son cíclicas y están provocadas por la expansión y contracción de la estrella debido a procesos físicos en el interior de la misma. También lo son las provocadas por sistemas binarios que se eclipsan cíclicamente, así como las denominadas variables rotantes, que

deben la variación de luminosidad a algún fenómeno relacionado con su propia rotación. No son cíclicas las variaciones de flujo de origen eruptivo, tales como las eyecciones de masa coronal, o las de origen cataclísmico, como supernovas. El objetivo del modelo es clasificar los objetos del conjunto de datos en estas clases variables, cuya taxonomía se muestra en la Figura 7

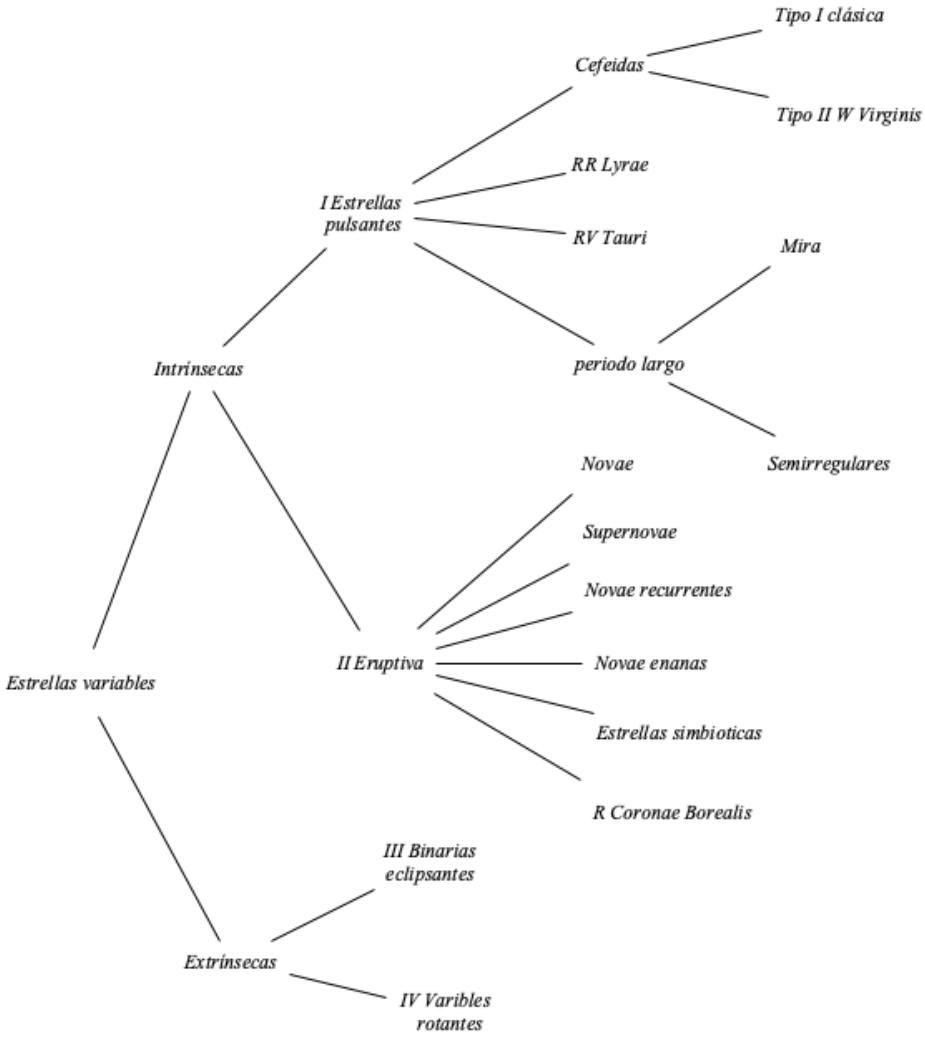


Figura 7. Taxonomía de estrellas variables

3.11 Clases de objetos presentes en el conjunto de datos

Las clases presentes en el conjunto de datos no están identificadas con la nomenclatura de la Figura 7. El equipo científico promotor del reto en Kaggle ha sustituido los nombres de las clases por números enteros aleatorios. Así, a lo largo de este trabajo se hará referencia a las clases *6, 15, 16, 42, 52, 53, 62, 64, 65, 67, 88, 90, 92 y 95* y no a sus nombres astronómicos, que son desconocidos. Este enmascaramiento de las clases ha sido decidido por los promotores del reto para situar en igualdad de condiciones a astrónomos y científicos de datos.

4 Modelo clasificador

4.1 Análisis de Los datos

Para la construcción del modelo clasificador se dispone de dos conjuntos de datos, denominados *data* y *metadata.* El primero es un conjunto compuesto por simulaciones de registros de flujo y tiempo tomadas por el telescopio LSST en cada una de las 6 bandas (colores) que capta el instrumento. El segundo son los metadadatos del objeto, no relacionados con el tiempo. A continuación se describen los atributos de ambos conjuntos en detalle.

4.1.1 Datos de flujo y tiempo (*data*)

- *object_id*: (int32) identificador único para cada objeto.
- *mjd*: (float64) Fecha de la observación en el calendario juliano modificado (MJD), que es el usado en astronomía desde los inicios de la carrera espacial en los años 50 del siglo XX. Cada unidad representa un día.
- *passband*: (int8) cada una de las seis bandas *(u, g , r, i z, Y)* de frecuencia en la que se divide la luz recogida por el instrumento. Cada medida de *flujo* corresponderá, por tanto, a una de las bandas
- *flux*: (float32): *flujo* registrado por el telescopio. Corresponde a la media de un conjunto de medidas *m* en un intervalo de tiempo
- *flux_err*: (float32) estimación del error de la medida de *flujo*. Corresponde a la desviación típica (σ) de *m*
- *detected*: (Boolean flag) un 1 en el flag indica que el brillo registrado en la muestra tiene una significación estadística superior a *3σ*; en caso contrario el flag vale 0.

4.1.2 Datos de posición, extinción estelar, Z y clase (*metadata*)

- *object_id*: (int8) identificador único para cada objeto.

- *ra*: (Float32) ascensión derecha. Longitud del objeto con respecto al punto de observación
- *decl*: (Float32) declinación. Latitud del objeto con respecto al punto de observación
- *gal_l*: (Float32) longitud galáctica. Se calcula con *ra*
- *gal_b*: (Float32) latitud galáctica. Se calcula con *decl*
- *ddf*: (Boolean) identificador para objetos observados en "*deep drill survey area*" o areas de reconocimiento profundo. Los *flujos* registrados en estas areas presentan una mayor precisión en las medidas pero suponen una parte pequeña de las observaciones.
- *hostgal_specz*: (Float32) medida de *desplazamiento al rojo* espectrométrico del objeto. Es importante destacar que esta característica no estará disponible en el conjunto de datos reales que obtendrá el LSST, dado que no se realizarán medidas espectroscópicas desde el telescopio. Esta característica, expuesta en la Sección 3.5, permite calcular de forma precisa (en escala cósmica) el desplazamiento al rojo de una fuente de luz y, por tanto, su velocidad y distancia.
- *hostgal_photoz*: (Float32) Desplazamiento al rojo estimado en función de la observación fotométrica (que sí realizará el LSST). Como se discute en la Sección 3.6, es una estimación mucho más imprecisa que la obtenida por espectrometría y debe considerase como una aproximación a la medida real.
- *hostgal_photoz_err*: (Float32) estimador de incertidumbre de *hostgal_photoz*.
- *distmod*: (Float32): *Módulo de distancia* del objeto.
- *mwebv*: (Float32) MW E(B-V). Extinción provocada por el polvo estelar de nuestra galaxia. Por tanto, es una función de las coordenadas del objeto en el cielo.
- *target*: (int8) La clase del objeto astronómico.

4.1.3 Examen de la composición del conjunto de observaciones

Como primer paso para la construcción del modelo clasificador, se realiza un examen los datos con el fin de extraer conocimiento acerca de ellos que sirva como base para desarrollar una estrategia para la resolución del problema. En primer lugar, si observamos la distribución de las clases, ilustrada por la Figura 8, comprobamos que, algunas clases son más comunes, como las clases 90, 42, 65 y 16 mientras que otras, como la clase 53, 64 y 6 tienen muy pocas observaciones. Además, si se dividen los datos entre objetos detectados en nuestra galaxia y objetos detectados fuera de ella, las distribuciones de los subconjuntos resultantes son completamente distintas, pues ninguna de las clases presentes en la distribución galáctica está presente en la distribución extragaláctica y viceversa.

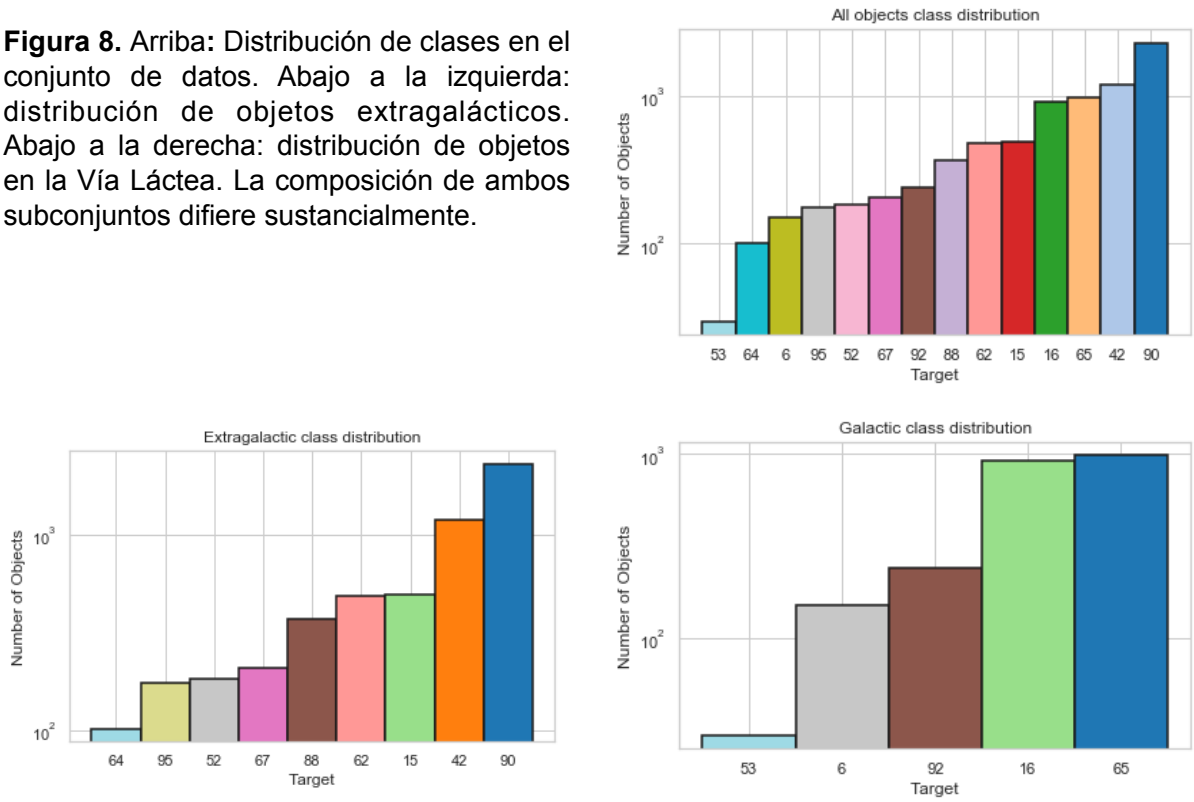


Figura 8. Arriba**:** Distribución de clases en el conjunto de datos. Abajo a la izquierda: distribución de objetos extragalácticos. Abajo a la derecha: distribución de objetos en la Vía Láctea. La composición de ambos subconjuntos difiere sustancialmente.

4.1.4 Examen del desplazamiento al rojo

Los datos de los que se dispone para el diseño del modelo son una simulación de los datos que se esperan obtener con el LSST. Esta simulación incluye

medidas espectrométricas del desplazamiento al rojo del objeto. Dado que el telescopio no realizará espectrometría, este atributo no puede ser evaluado por el clasificador ni formar parte del conjunto de entrenamiento. No obstante, sirve como evaluador de la precisión de la fotometría. En adelante usaremos indistintamente la denominación Z_f para el desplazamiento al rojo fotométrico y Z_s para el espectrométrico.

Dado que en una parte de los datos Z_f es un estimador muy impreciso de Z_s, se desea agrupar los datos en función la precisión de dicho estimador. Comprobamos que la distribución entre ambas medidas es susceptible de ser dividida en tres subconjuntos, de manera que puedan ser tratados como corresponda. Aplicamos k-medias y obtenemos como resultado las agrupaciones de la Figura 9.

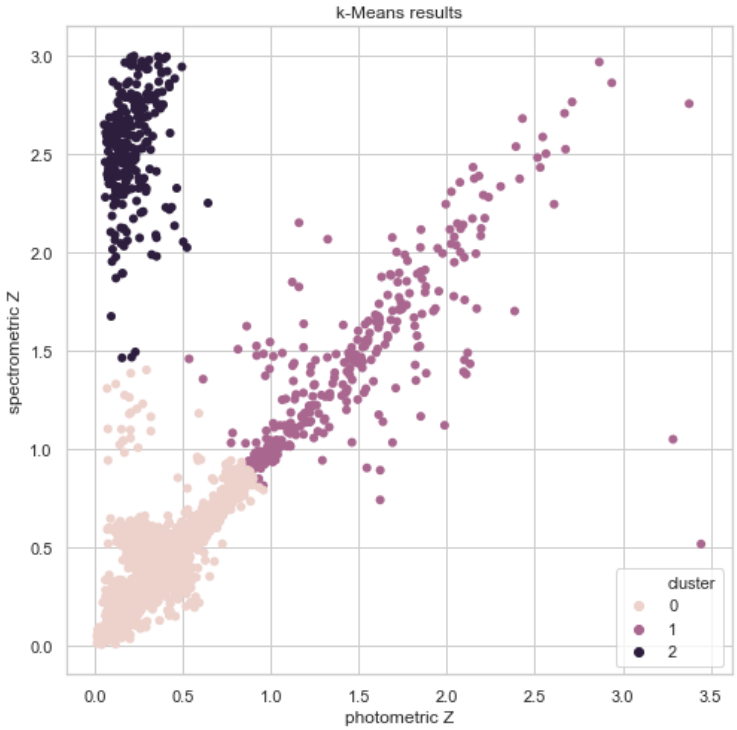


Figura 9. Agrupamientos resultantes de *k-medias*

Dividido el conjunto, se examina la distribución de los tres agrupamientos y se detecta que uno de ellos tienen una distribución notablemente distinta, tal y como muestran los tres histogramas de la Figura 10.

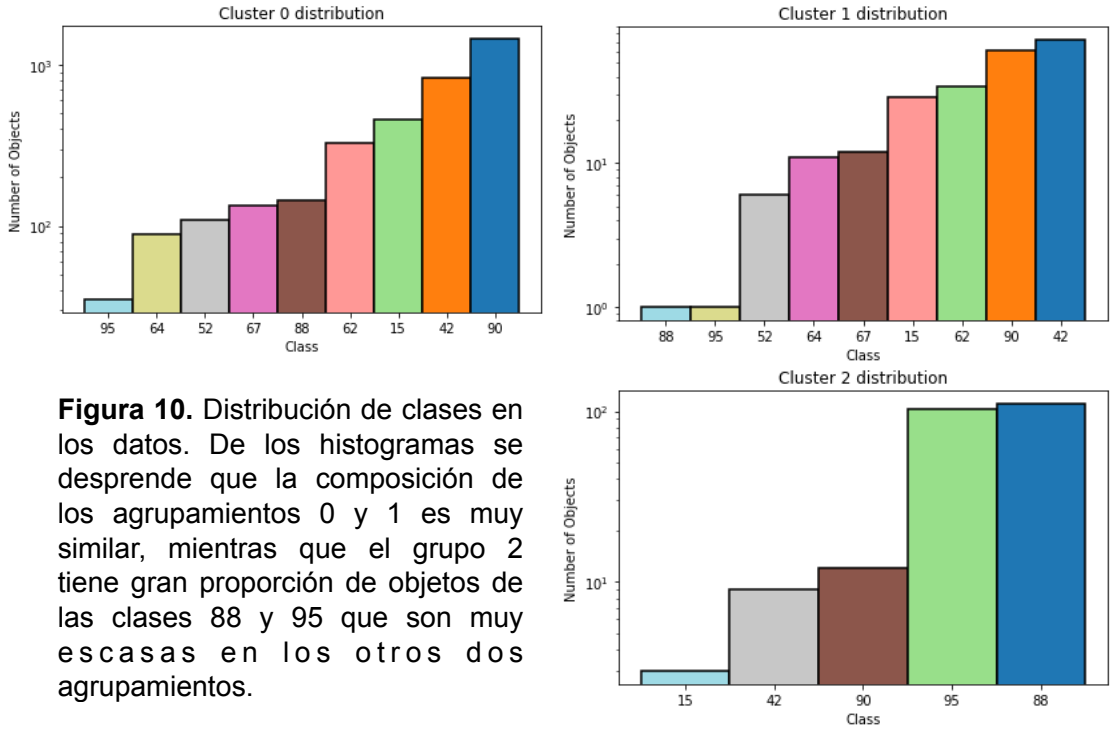


Figura 10. Distribución de clases en los datos. De los histogramas se desprende que la composición de los agrupamientos 0 y 1 es muy similar, mientras que el grupo 2 tiene gran proporción de objetos de las clases 88 y 95 que son muy escasas en los otros dos agrupamientos.

A la vista de estos resultados, de decide usar distintos clasificadores para los diferentes conjuntos, ya que si los subconjuntos son suficientemente distintos, se puede entrenar con ellos distintos clasificadores, de manera que cada clasificador esté ajustado a la distribución de clases de cada grupo. Como ya se ha dicho, no se dispondrá de medidas espectrométricas en futuros datos, de manera que la división de los datos por los agrupamientos de *k-medias* no es posible. No obstante, se puede eliminar el atributo Z_s y determinar el agrupamiento asumiendo cierto grado de error.

4.1.5 División de Z_f en tres bandas

Se divide el dominio de Z_f en tres segmentos del eje o bandas donde se ubican los objetos en función del agrupamiento efectuado por k-medias. Así cada objeto tendrá un nuevo atributo que identificará dicha banda de Z_f. Se discute a continuación el método para determinar los puntos divisorios entre las tres bandas, de manera que se minimice la diferencia entre la pertenencia a un grupo de k-medias y su banda respectiva.

Para estudiar la distribución de valores de Z_f por la correlación se traza un diagrama de caja donde se representa una caja para cada uno de los tres grupos (Figura 11)

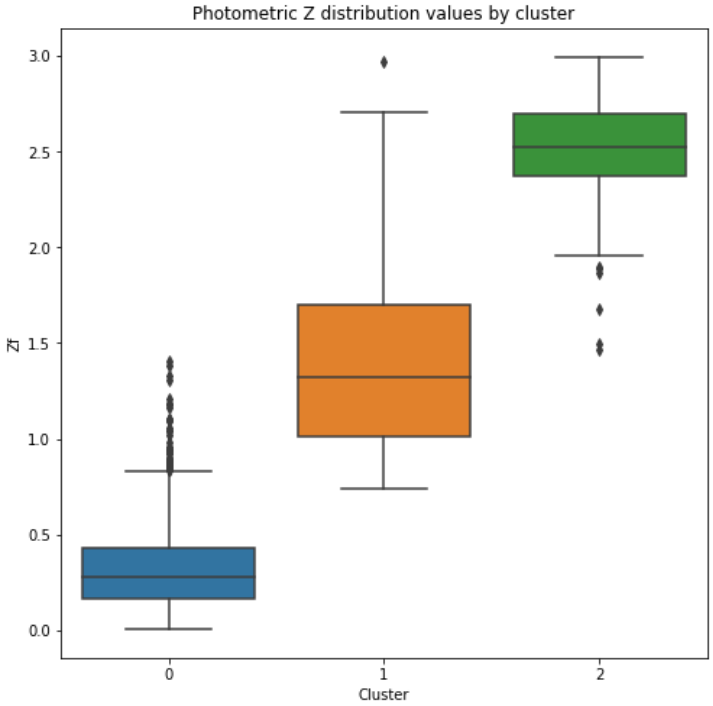


Figura 11. Diagrama de caja de Z_f en cada uno de los tres agrupamientos resultantes de k-medias.

En este diagrama se aprecia que no existe solapamiento entre las cajas y por lo tanto la separación puede realizarse con pocos errores entre agrupamiento y banda de Z_f. Se calculan los tres puntos, p_{ij}, del dominio de Z_f que dividen las bandas, que denominamos coma atributo *zBand*:

$$p_{ij} = Q_{3i} + \frac{\left(Q_{1j} - Q_{3i}\right)}{2}$$

donde: *i, j son* bandas en *hostgal_photo* y Q_{nm}es el cuartil *n* de la banda *m.* Se calculan el centro de la distancia entre Q_3 del primer grupo y Q_2 del segundo

por un lado y el centro de la distancia desde Q_3 del segundo hasta Q_1 del tercero por otro. Se divide el dominio de Z_f en las siguientes bandas:

$$zBand\,1 = \left(0,\ 0.757225\right],\ zBand\,2 = \left(0.757225,\ 2.0419\right],\ zBand\,3 = \left(2.0419,\ 3\right]$$

En la Figura 12 se evalúa mediante diagramas de nube de puntos los grupos de obtenidos por la agrupación y la asignación de zBand a cada elemento de los datos.

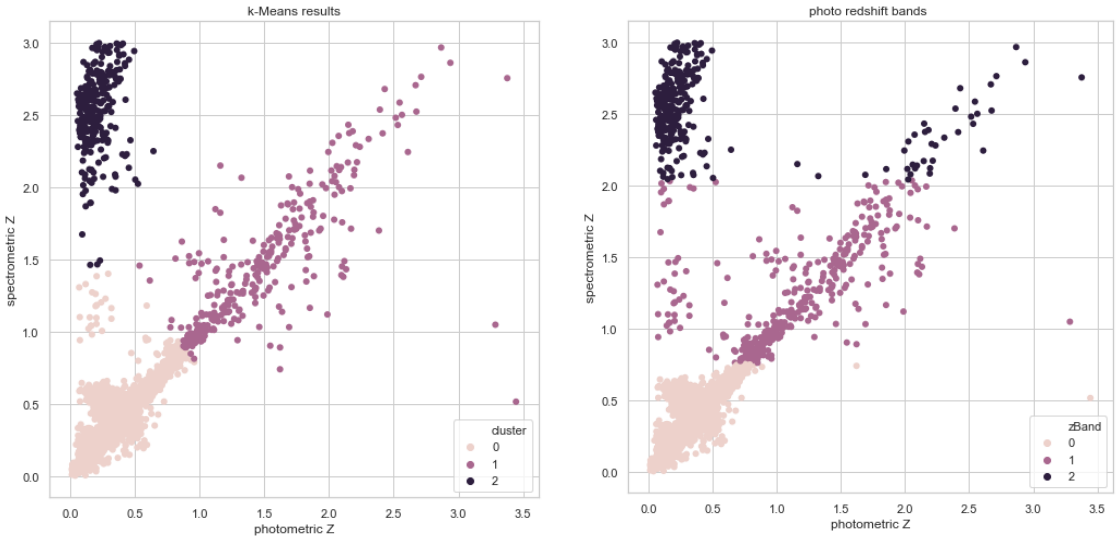


Figura 12. Izquierda: representación de las agrupaciones realizadas por k-medias. Derecha: representación de las muestras en función del atributo zBand asignado.

Para evaluar la pérdida de información producida al eliminar una dimensión, se calcula la proporción sobre el total de objetos cuyo atributo *zBand* es distinto agrupamiento de k-medias, obteniendo un resultado del 4% de muestras mal asignadas.

4.2 Tratamiento del ruido

Las medidas de flujo con las que se construyen las series temporales tienen, en muchos casos, estimaciones de error (atributo *flux_err*) muy grandes. En algunos casos, además, pueden existir muestras tan alejadas de un valor

probable que deben ser consideradas ruido. Así, con el fin de reducir el impacto de este error, se aborda la cuestión con dos diferentes aproximaciones.

4.2.1 Eliminación del ruido

Según la literatura, se considera ruido, en una serie temporal de flujo, a aquellas observaciones *cuya diferencia con la media de las observaciones supere 5σ. Además, se descartan todas las observaciones cuya estimación del error sea superior a tres veces la media del error de la serie temporal. Esto* supone una significación estadística de 99.99994%. La significación de 5σ es utilizada habitualmente en astronomía y, en general, en los procesos donde la existencia de errores sistemáticos (por ejemplo fallos en la calibración del instrumento) pueden desplazar la media de las observaciones significativamente, de manera que con una significación menor, y un cierto sesgo sistemático, podría considerarse como ruido parte de la información. En la Figura 13 se muestra el umbral de 5σ en la *banda i* de un objeto arbitrario.

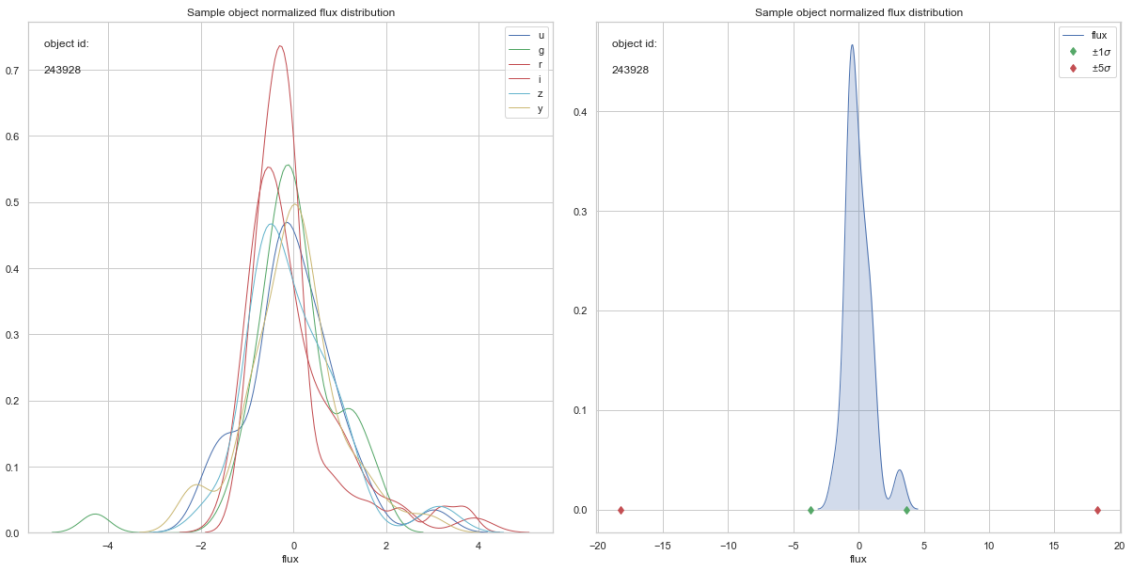


Figura 13. Izquierda: distribución normalizada de flujo por cada banda del objeto *243928*. Derecha: distribución de flujo en la banda *i* (correspondiente al color rojo y parte de infrarrojo). En el eje de abscisas están identificados los puntos correspondientes a *sigma* = 1 y *sigma* = 5. Este último punto es el que arbitrariamente se denomina umbral de ruido. A derecha e izquierda del rango delimitado por estos umbrales se descarta cualquier muestra.

4.2.2 Reducción bayesiana del ruido

Si se observa una serie temporal, como la de la Figura 14, donde el tiempo (atributo *mjd*) está representado en el eje de abscisas y flujo en el eje de ordenadas, se tiene una idea del comportamiento del flujo en el tiempo. Dado que las medidas de flujo tienen una incertidumbre (atributo *flux_err*), se puede representar dicha incertidumbre mediante barras de error, lo que permite hacer una aproximación visual a la significación estadística de las muestras de la *serie temporal*. Así, una serie con grandes varianzas en sus medidas de *flujo* presentará líneas mayores que otra con varianzas pequeñas.

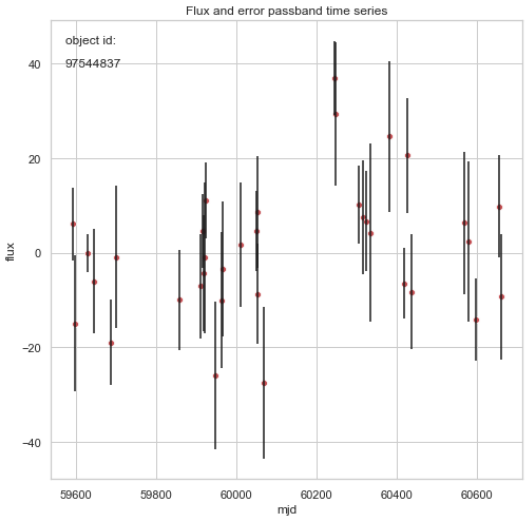


Figura 14. Ejemplo de serie temporal en una única banda. Las lineas negras representan el *error de* flujo, usados como estimador del error.

El estudio de los datos pone de manifiesto que algunos de los datos de *flujo* presentan grandes incertidumbres. Se procede, por tanto, a aproximar los valores de flujo a su valor más probable mediante una inferencia bayesiana. A partir de un distribución de probabilidad observada, denominada *anterior*, se puede estimar el valor más probable en la distribución de probabilidad *posterior* que es desconocida, y que ha generado los resultados de la muestra. Así, a según el teorema de Bayes, puede estimarse la media de la distribución *posterior* o valor más probable de flujo, conocida la varianza y la siendo la semejanza una distribución normal [13].

Se asume que la *anterior* viene dada por observaciones de flujo para un objeto y banda *b*. Se asume que el valor observado, *flux*, es generado por una distribución normal cuya media es el valor real de flujo y tiene una desviación típica *fluexerr.*

Según se describe en en *StatLect-Bayesian inference* [7], la media de la distribución *posterior*, $flux\,estimate$, para una *anterior* normal, con varianza conocida se calcula:

$$flux\,estimate = \left(\frac{n}{{\sigma_b}^2} + \frac{1}{fluxerr^2}\right)^{-1}\left[\frac{n}{{\sigma_b}^2}\left(\frac{1}{n}\sum_1^n flux_b\right) + \frac{1}{fluxerr^2}flux\right]$$

dado que para cada estimador el vector tiene una única observación *(n = 1)* se tiene que

$$flux\,estimate = \frac{\left(\frac{\overline{flux}_b}{{\sigma_b}^2} + \frac{flux}{fluxerr^2}\right)}{\left(\frac{1}{{\sigma_b}^2} + \frac{1}{fluxerr^2}\right)}$$

donde $flux\,estimate$ es el valor de flujo mas probable para cada observación de la serie en la banda *b*. En conclusión, la media *posterior* (el valor estimado) es una media ponderada de dos señales:

- la media de la muestras observadas
- la media *anterior*

De manera que el peso en la ponderación de cada una de las señales varía en función de la incertidumbre. Así, la señal con menor varianza recibe mayor peso. En la Figura 15 se ilustra el proceso.

4.6 Cálculo de magnitud absoluta

Como se ha descrito en la Sección 3.9, la magnitud es el brillo aparente de un objeto situado a 10 pc de distancia y se define

$$M = -2.5\log_{10} f - (m - M)$$

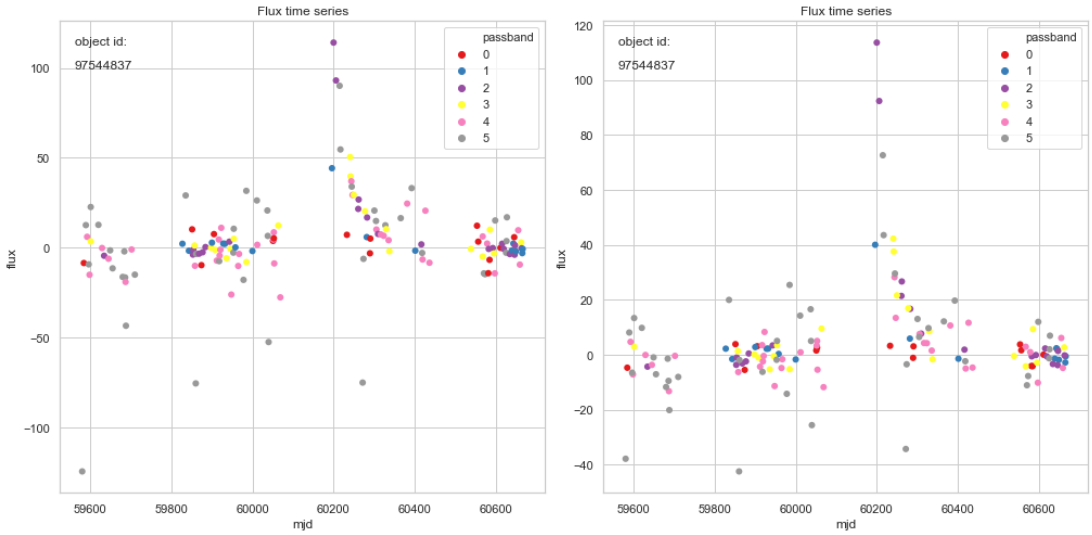


Figura 15. Izquierda: ejemplo de serie temporal de flujo antes del proceso bayesiano de reducción de ruido. Derecha: mismo objeto tras la reducción de ruido.

donde $(m - M) = distmod$ (módulo de distancia) y f es flujo en la *banda* de frecuencia *i,* que es la banda más energética del espectro de una fuente. Por tanto se define una nuevo atributo denominado *magnitud* :

$$Magnitud = -2.5 \log_{10} \sum_{0}^{x} flux_{i_k} - distmod$$

4.7 Cálculo del Color

Una característica de cualquier objeto astronómico es su color, que además es un indicador de la temperatura del objeto. Así, los objetos más azulados son más fríos que los objetos rojizos. En astronomía el color se mide según la escala de Johnson-Morgan [13]. Se entiende el color como la distribución de las distintas longitudes de onda en el espectro visible (3000Å-6800Å), que en los datos se encuentran representadas en las *bandas u, g, r.*

En la escala Johnson-Morgan se calcula la magnitud de cada fuente en cada una de la tres bandas que *U, B, V (Ultraviolet, Blue, Visible),* que corresponden aproximadamente a las bandas *u, g , r* del LSST, y se calculan las magnitudes

para cada banda. El color se incorpora a los datos en dos nuevas características (*U - B* y *B - V*) y se calculan

$$U - B = M_g - M_u \quad , \quad B - V = M_r - M_g$$

donde M_k es la magnitud absoluta calculada (según el método del apartado 4.6) para la *banda k.*

4.7 Extracción de atributos

La extracción de atributos funciona en dos fases. En la primera se construyen las series temporales de cada objeto, de manera que al final del proceso se tiene un conjunto de series de *tiempo (T) flujo (M),* y *error (E)* tales que:

$$S_{ob} = \left(T_{ob},\ M_{ob},\ E_{ob}\right),\ \left|T_{ob}\right| = \left|M_{ob}\right| = \left|E_{ob}\right|$$

donde *S* es una serie temporal, *o* es un objeto del conjunto y *b* es cada una de las seis *bandas*. Además se tiene que

$$o = \left[1, n\right],\ b = \left[0,\ 5\right] : o,\ b \in \mathbb{N}$$

donde *n* es el número de objetos presentes en los datos. En la Figura 16 se muestra el diagrama de bloques del proceso de extracción de atributos.

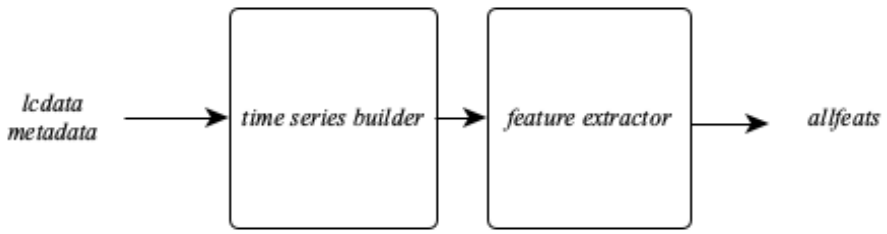


Figura 16. Diagrama de bloques del extractor de atributos

Una vez construidas las series temporales, estas pasan, una a una, como entrada, al extractor de atributos, que recibe una lista de funciones como parámetro, y devuelve, para cada serie, un vector de resultados de dichas funciones aplicadas a S_{ob} y pasadas como parámetros al extractor. Este vector de resultados se añade a un nuevo conjunto de datos denominado *allfeats*

Paralelamente los atributos en *metadata* son filtrados, descartándose aquellos que no aportan información útil al modelo, A continuación se incorporan al vector de resultados obtenido en el bloque descrito anteriormente. El funcionamiento de este segundo bloque se describe la Figura 17**.**

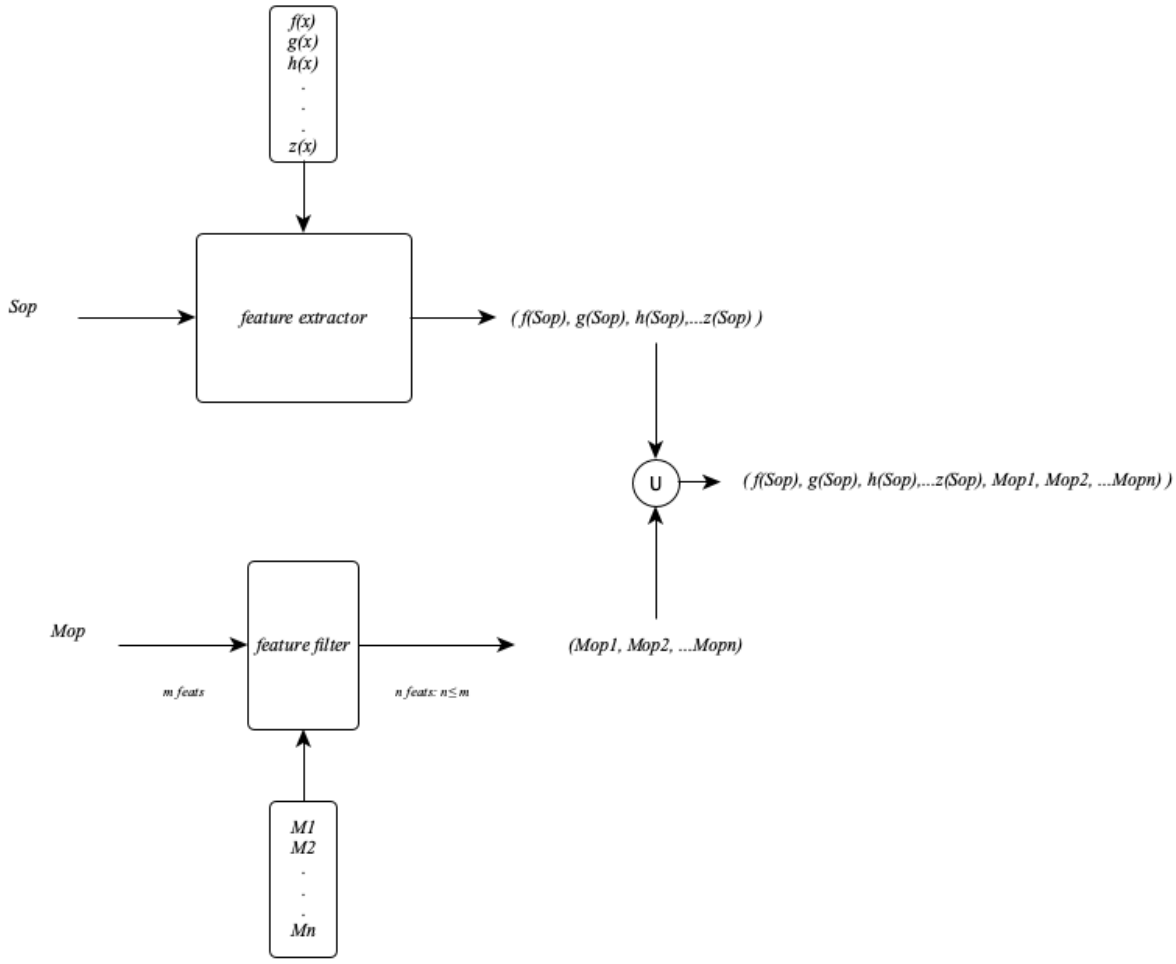


Figura 17. Diagrama de bloques de la extracción de características, donde *S* es la serie temporal de *o-esimo* objeto y la *p-esima banda* y M_n el *n-esimo* metadato.

4.7.2 Funciones de extracción de atributos

Se introducen como parámetros en el extractor 91 funciones y 7 metadatos al filtro, resultando el proceso con la extracción de 553 atributos. Las funciones generadoras de las características están divididas en 3 clases según su origen

- Librería *Cesium-ML* [8]
- Adaptadas de la librería *FEETS* [9]
- Implementadas íntegramente en este trabajo

Los dos últimos grupos funciones están descritas en los comentarios del código, mientras que la descripción de las primeras están disponibles en el API de *Cesium-ML*. Finalizado el proceso de extracción de atributos se tiene un conjunto de datos, denominado *allfeats.*

4.8 Selección de atributos

Tras el proceso de extracción de características se tiene un conjunto numeroso de atributos, en el que cada uno de ellos contiene una parte muy pequeña de información. Se procede a reducir este conjunto, con el fin de reducir la dimensionalidad del problema y evitar la pérdida de rendimiento producida por este exceso de atributos. Así, como se muestra en la Figura 18, se clasifica el conjunto resultante del proceso de extracción de características, en un bosque aleatorio.

Tras la clasificación realizada por el bosque aleatorio se obtienen dos productos:

- *score*: una medida de la precisión de la clasificación realizada

- *ranking*: una lista ordenada por importancia de los atributos usados en su construcción

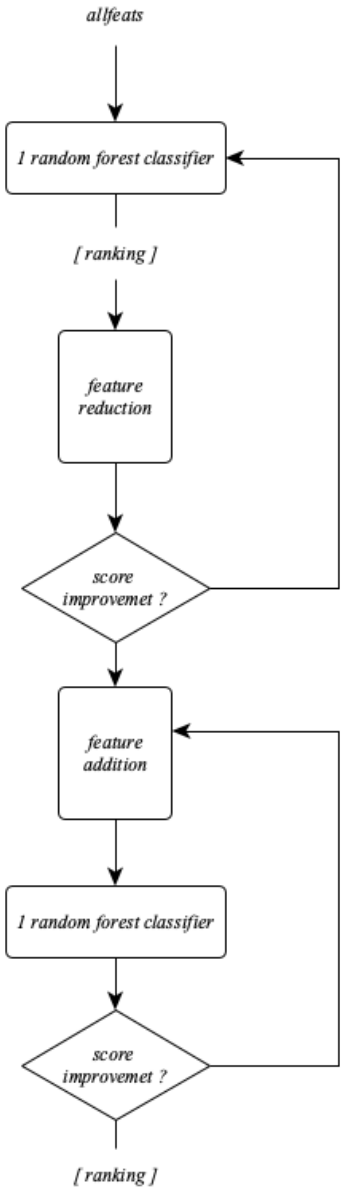


Figura 18. Diagrama de bloques del selector de atributos

Seguidamente, se eliminan de *ranking* los atributos menos representativos. Estos atributos son aquellos cuya fracción de importancia acumulada esté por debajo de una cantidad arbitraria, fijada en 0,8[4]. Así se conservan únicamente los atributos más relevantes usados en la clasificación. Se realiza una nueva clasificación y se contrastan los resultados de la anterior clasificación anterior y

[4] 0.8 Es la fracción que mejores resultados presenta en los diferentes experimentos realizados

de la actual. Si el *score* actual mejora al anterior se repite el proceso de reducción de atributos.

Este proceso continúa hasta que la reducción de atributos provoca en un *score* peor que el obtenido en la anterior iteración. Entonces, se eliminan de uno en uno del *ranking* los atributos tras cada nueva clasificación. Este segundo proceso de eliminación concluye cuando el *score* deja de mejorar. Así, se ajusta el tamaño de *ranking* reduciendo atributos hasta dar con el mejor resultado. A modo de síntesis, este bloque ajusta el número de atributos reduciéndolos hasta encontrar un número aproximado de ellos y después realiza un ajuste fino reduciéndolos de uno en uno. El resultado final de este proceso es un *ranking* de atributos ajustada al mejor *score* posible.

4.9 Clasificación de los datos

Sobre un subconjunto de los datos (aproximadamente el 25%) que no ha intervenido en ninguna de las fases anteriormente descritas, se realizan los procesos de cálculo de magnitud, color, división en zBands, tratamiento del ruido, inferencia bayesiana y extracción de características y se eliminan todos los atributos del conjunto resultante (denominado test allfeats) que no estén presentes en *ranking*. Así, se tiene un subconjunto de los datos con los que entrenar y validar un conjunto de clasificadores que describimos a continuación. En la Figura 19 se da una visión general del *pipeline* de este bloque de clasificadores.

En primer lugar se clasifican los datos en un clasificador de 1 bosque aleatorio y se obtiene un *score*.

El segundo clasificador divide los datos en 2 subconjuntos de objetos galácticos y extragalácticos, dado que, como se trata en la Sección 4.1.3, presentan distribuciones notablemente diferentes. Así, se clasifica independientemente cada subconjunto en un bosque distinto. Finalmente, se

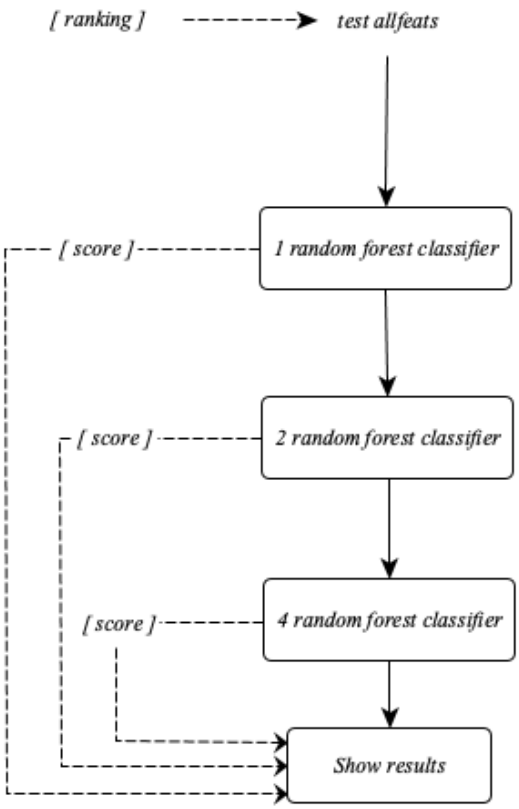


Figura 19. Pipeline del bloque de clasificadores

reconstruye el conjunto y se obtiene el *score* del conjunto completo, de manera que pueda compararse con el resto de clasificadores presentados.

El tercer clasificador divide los datos en 4 subconjuntos: uno para los objetos galácticos y otro más para cada uno de los 3 grupos de que identifica el atributo *zBand* (calculado en la Sección 4.1.6) y, de forma similar al clasificador de 2 bosques, realiza 4 clasificaciones independientes. Finalmente se obtienen los resultados parciales de cada subclasificador y el resultado global.

4.10 Métricas

La evaluación de la bondad de los diferentes clasificadores se efectúa con las métricas descritas a continuación. Por un lado, la precisión de un clasificador (*accuracy*) se define como:

$$accuracy = \frac{1}{n}\sum_{1}^{n} y, y \in \{0, 1\} \mathbb{N}$$

siendo *y* una función de las clasificaciones individuales, que devuelve 1 en caso de ser correcta la clasificación y donde *n* es el número total de elementos clasificados por el modelo. Por tanto, *accuracy,* evalúa qué fracción de las observaciones se clasificó correctamente.

La *pérdida logarítmica* (*logarithmic loss, log loss, binary cross-entropy*) es una método de evaluación de la bondad de un modelo clasificador. Se toma una entrada en forma de probabilidad de pertenencia a un clase *c* para un elemento *e*, $P_{ec} \in [0, 1]$, devuelve un valor, tanto mejor cuanto menor sea este. Así, la *pérdida logarítmica* ideal de un modelo es *0.* Durante el proceso de reducción de características se utiliza la *pérdida logarítmica,* que se define:

$$plog_o = -\sum_{1}^{M} y_{oc} log(P_{oc}), \ M > 2, \ y \in \{0, 1\}$$

donde P_{oc} son las probabilidades de que la observación *o* pertenezca a la clase *c* y *M* el número de clases del conjunto. La *pérdida logarítmica* (*log loss*) de un conjunto de datos con *n* observaciones se calcula mediante el promedio de las pérdidas logarítmicas de cada observación. Esta métrica, al estar basada en probabilidades, permite penalizar de distinta forma a unas clases de otras, en función de las necesidades del modelo. Así una vez obtenidas la probabilidades para cada clase, estas se pueden multiplicar por un vector de pesos, de forma que se incremente la penalización en el error para una o varias clases.

4.11 Validación cruzada

La validación cruzada (*cross-validation*) consiste en repetir la clasificación sobre diferentes particiones *entrenamiento-prueba* del mismo conjunto y calcular la media aritmética obtenida de las medidas de evaluación sobre estas particiones.

Dado un conjunto de datos *D* y un clasificador *c* construido con dicho conjunto, se hace necesario, con en fin de evaluar el rendimiento de *c*, otro conjunto de datos que no hayan intervenido en la construcción del clasificador. Si solo se dispone de los datos *D*, entonces es necesario un método de validación cruzada para realizar dicha evaluación. Así, se divide *D* en dos subconjuntos: prueba (*P*) y entrenamiento (*E*).

Como se deben clasificar todos los elementos de *D*, y dado que una parte de ellos (los que forman *E*) no pueden ser clasificados con *c* porque *este* ha sido entrenado con ellos, se debe entonces dividir *D* en dos nuevos conjuntos de entrenamiento y prueba de forma que los elementos que se han usado para entrenar *c* formen ahora parte del nuevo conjunto de prueba. Así, el proceso de clasificación se repetirá *k* veces con distintas particiones de *D*, y permitirá contrastar las *k* clasificaciones de forma que se obtenga un estimador de la precisión y error de *c*. Para *k* iteraciones se define el *accuracy* de la validación cruzada, o cross validation *accuracy*, como la media del accuracy de las *k* clasificaciones y el error cometido como la desviación estándar de dicho conjunto de clasificaciones. Mediante estos estimadores, (*cross validation accuracy y cross validation error*) se tiene una métrica para la evaluación de los 3 clasificadores de bosques aleatorios usados en el modelo.

4.12 Parametrización

En la fase de pruebas se han ajustado los parámetros del clasificador de 1 *bosque aleatorio* hasta hallar con la configuración que mejores resultados produce. Se describen estos parámetros a continuación.

1. **Máximo número de atributos por árbol**: con el fin de evitar el sobreajuste, limita la cantidad de atributos con los que cada árbol del bosque evalúa cada objeto. Se ha probado con fracciones del conjunto de atributos tales como la raíz cuadrada del numero de atributos o su $\log_2$. Finalmente se ha usado el escalar (27) que daba mejores resultados
2. **Criterio**: como se discute en el apartado 2.1.1, se usa el *criterio de Gini* para determinar el atributo que se evalúa en cada nodo.
3. **Máxima profundidad del árbol**: se limita la profundidad máxima de cada *árbol de decisión* que compone el *bosque aleatorio*, de forma que se limite el *sobreajuste* del clasificador. Se fija en 13 niveles.
4. **Mínimas muestras por partición**: solo se considerará un nodo si al menos deja la cantidad determinada en el parámetro en cada una de sus ramas izquierda y derecha. Experimentalmente, 5 muestras es el parámetro que mejores resultados ofrece.
5. ***Warm start***: los *árboles* del *bosque aleatorio* se inicializan a partir del primer *árbol* del *bosque*, del que se toman ciertas propiedades.

4.13 Resultados

En el proceso de validación cruzada de los clasificadores se obtienen los resultados que se muestran en la tabla 4.1

Tabla 4.1 Resultados del modelo

Test n	ne	bnr	1RF accuracy	2RF accuracy	gal. accuracy	2RFex gal acc	4RF accuracy
1	no	no	0.80 (+/- 0.04)	0.80 (+/- 0.04)	0.98 (+/- 0.04)	0.74 (+/- 0.04)	0.81 (+/- 0.07)
2	no	si	0.80 (+/- 0.03)	0.80 (+/- 0.04)	0.98 (+/- 0.03)	0.74 (+/- 0.05)	0.80 (+/- 0.06)
3	si	no	0.80 (+/- 0.03)	0.80 (+/- 0.03)	0.98 (+/- 0.02)	0.73 (+/- 0.03)	0.80 (+/- 0.06)
4	si	si	0.81 (+/- 0.03)	0.80 (+/- 0.04)	0.98 (+/- 0.02)	0.74 (+/- 0.05)	0.79 (+/- 0.05)

Test n	ne	bnr	4RF gal accuracy	k=0 accuracy	k=1 accuracy	k=2 accuracy	feats
1	no	no	0.98 (+/- 0.04)	0.75 (+/- 0.07)	0.84 (+/- 0.12)	0.52 (+/- 0.18)	53
2	no	si	0.98 (+/- 0.03)	0.75 (+/- 0.06)	0.84 (+/- 0.12)	0.50 (+/- 0.17)	54
3	si	no	0.98 (+/- 0.02)	0.75 (+/- 0.06)	0.83 (+/- 0.10)	0.52 (+/- 0.21)	57
4	si	si	0.98 (+/- 0.02)	0.74 (+/- 0.05)	0.82 (+/- 0.11)	0.45 (+/- 0.14)	100

Interpretación de la tabla de resultados:

- **Test n:** número de prueba realizada. Nótese que las filas se han dividido en dos para ajustarlas al ancho de página, colocando una mitad sobre la otra. Así, columna **Test n** indica la misma prueba en la parte superior e inferior de la tabla.
- **ne**: identifica si la prueba se ha hecho con eliminación del ruido, tal como se describe en la Sección 4.4.1
- **bnr**: identifica si se ha usado la reducción bayesiana de ruido , descrita en la Sección 4.4.2
- **1RF accuracy**: precisión y error en la validación cruzada del clasificador de 1 bosque aleatorio
- **2RF gal accuracy**: precisión y error en la validación cruzada del clasificador de 2 bosques aleatorios en el subconjunto de objetos galácticos
- **2RF ex gal acc**: precisión y error en la validación cruzada del clasificador de 2 bosques aleatorios en el subconjunto de objetos extragalácticos

- **4RF gal accuracy**: precisión y error en la validación cruzada del clasificador de 4 bosques aleatorios en el subconjunto de objetos galácticos
- **4RF k=n accuracy**: precisión y error en la validación cruzada del clasificador de 4 bosques aleatorios en el subconjunto de objetos extragalácticos de la *k*-esima banda de Z_f (atributo *zBand*)
- **feats**: número final de atributos resultantes del proceso de reducción

A continuación se muestran las matrices de confusión de los tres clasificadores.

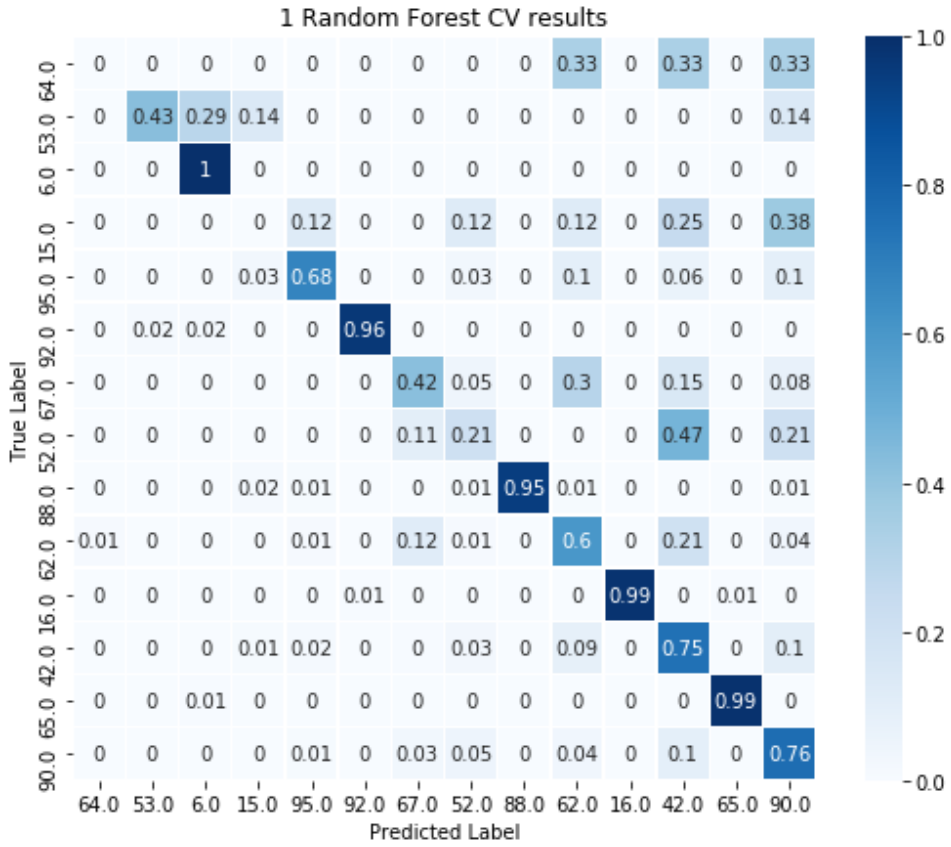
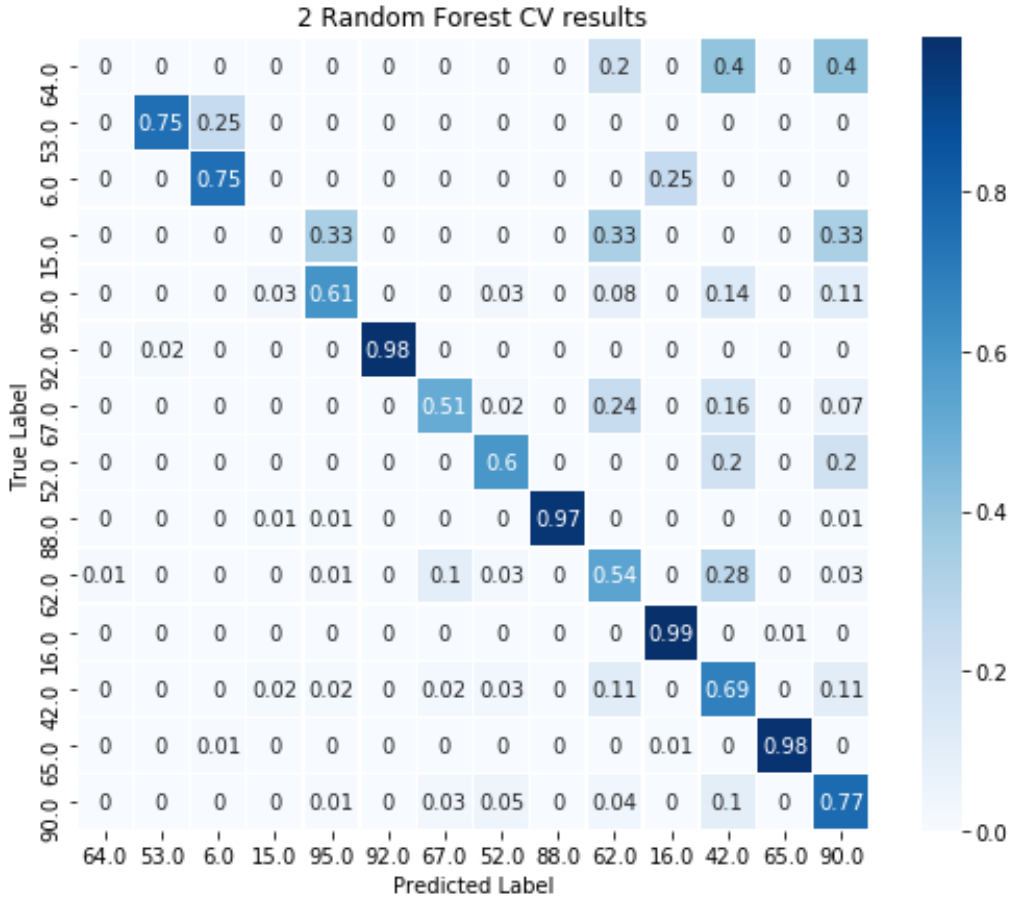


Figura 20. Matriz de confusión del clasificador de 1 bosque aleatorio



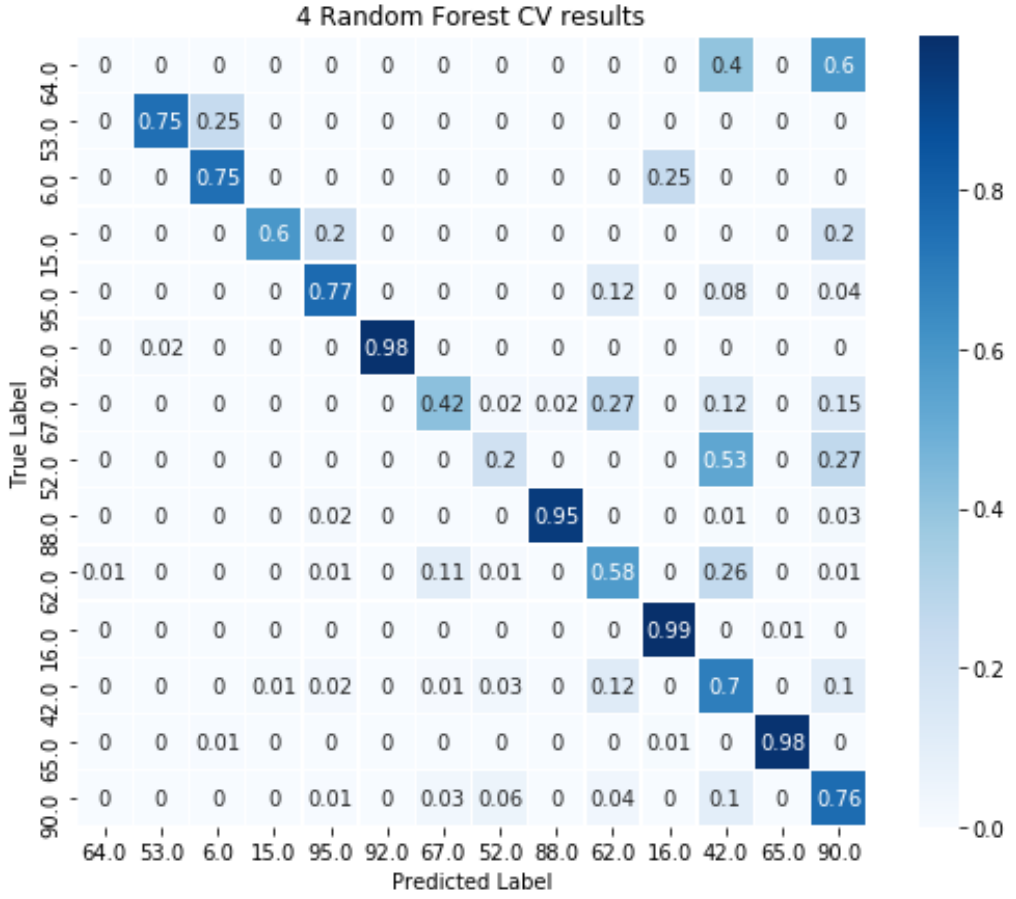


Figura 22. Matriz de confusión del clasificador de 4 bosques aleatorios

4.12 Discusión

Denominaremos en adelante 1RF, 2RF y 4RF a los clasificadores de 1, 2 y 4 bosques aleatorios del modelo. Los resultados indican que el clasificador con mejor resultado en *accuracy* y menor error es RF1 sometido al proceso de eliminación de ruido (que denominaremos en adelante *ne*) y a la reducción bayesiana de ruido (que denominaremos *bnr*). Los resultados muestran que ambas técnicas implementadas, producen variaciones poco significativas del *accuracy* de los diferentes clasificadores pero que reducen el error en torno al 1%.

4.12.1 División en clasificadores independientes

Contra el planteamiento propuesto inicialmente, de que para distintas distribuciones, distintos clasificadores deberían producir una mejor clasificación, los resultados indican que los clasificadores 2RF y 4RF no mejoran el *accuracy* de 1RF. Esto puede deberse a que a los grupos extragalácticos 1 y 2 tienen muy pocas muestras (119 y 80) y por tanto, el bosque de esos subconjuntos se construye con poca información. Por otro lado Z_f se encuentra en el primer lugar del *ranking* de atributos (con un 9,9% de importancia). Recordemos que los grupos extragalácticos 1 y 2 son aquellos en los que Z_f tenía peor comportamiento como estimador del desplazamiento al rojo (ver Sección 4.1.5). Así, la mala calidad del atributo en estos subconjuntos, explicaría también el mal resultado del clasificador en esos grupos. En la figura 23 se ilustran los resultados parciales de 4RF para los subconjuntos 1 y 2.

Se observa que el clasificador del grupo 2 de 4RF tiende a clasificar todas las clases 52, 95, 67 y 15 como clases 42 y 90. Si se observan las curvas de luz en el Anexo I se advierte que se trata de objetos cuyas curvas de luz son bastante parecidas. Por otro lado, el clasificador del grupo 1 de 4RF clasifica todas las observaciones de las clases 62 y 67 como clase 95.

4.12.2 Eliminación de ruido y reducción bayesiana

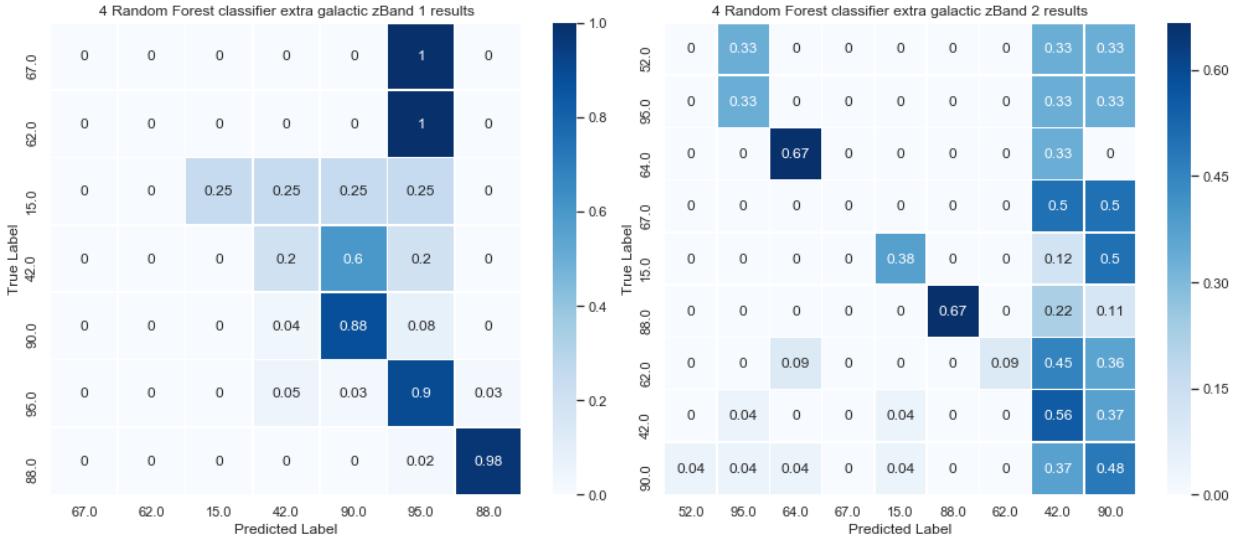


Figura 23. Matrices de confusión de los subconjuntos 1 y 2 de RF4. Nótese que los grupos 15, 62, 52 y 67 son indetectables por los clasificadores y que el clasificador 2 tiende a clasificar a todas las clases como clases 42 y 90

El proceso *ne* tampoco produce mejora perceptible en el *accuracy* de ningún clasificador, si bien causa una mejora del error en torno al 1%. Los resultados de la aplicación de ambas técnicas sobre los datos no parecen ser acumulativos, lo cual es compatible con el hecho de que buena parte de las muestras, que son aproximadas mediante *bnr*, son descartadas por el proceso *ne*.

Aunque la información disponible acerca de los datos no dice nada al respecto, es posible que hayan sido sometidos a un proceso de reducción de ruido previamente (de la misma manera que sí sabemos que previamente se ha calculado la *extinción*). Esto explicaría el escaso impacto en los resultados del proceso *ne.* Además, sabemos que *ne* ha eliminado el 0,42% de las muestras en el proceso, lo que indica unos datos muy poco ruidosos. Esta ausencia de ruido podría provenir también de una generación de los datos que no ha tenido en cuenta esta cuestión (recordemos que son una simulación)

4.12.3 Magnitud absoluta y color

La magnitud absoluta (descrita en la sección 4.6) ha resultado un atributo de gran importancia en la clasificación, ocupando el segundo lugar en el *ranking*, con un 5,9% de importancia en el modelo, solo superado por el desplazamiento al rojo Z_f. En cuanto a los dos atributos de color (atributos *U-B* y *B-V*) descritos en la Sección 4.7, el primero no resulta significativo y queda fuera del *ranking* tras el proceso de reducción de características, mientras que el segundo ocupa el puesto 44 con un 0,68% de importancia.

Esta baja importancia de los atributos de color podría deberse a que no se ha aplicado ningún proceso de corrección del color en función del desplazamiento al rojo [12], tal como proponen Beare et al. Así, consideramos que si no se corrige el color en función del desplazamiento al rojo, los atributos calculados no aportan información relevante al clasificador. Dicho de otro modo, si el color varía con la distancia, entonces pierde validez como atributo clasificador.

4.12.4 Fortalezas del modelo

Determinadas clases, son fácilmente identificables por el modelo y se obtienen valores cercanos a 1 en los 3 clasificadores. Estas clases son *53, 92, 88, 16, 65 y 90*, de las cuales 4 corresponden a objetos galácticos. Además, contrastando estas clases con las curvas de luz de las Figuras ilustradas en el Anexo I se detecta que las todas las clases parecen pertenecer a objetos cuya curva de luz es de naturaleza cíclica, salvo la clase *90,* que parece un objeto de origen cataclísmico. Así, podemos concluir que el modelo detecta mejor objetos de naturaleza periódica. Por ello, en futuras versiones del clasificador será conveniente, al incorporar nuevas funciones al extractor de atributos, buscar aquellas que no analicen la periodicidad.

4.12.5 Correlación entre atributos

Se estudia la *correlación de Pearson* entre los atributos de la tabla 4.1.2, representada en la matriz de la Figura 24. Las zonas coloreadas en tonos

oscuros representan fuertes correlaciones, mientras que las claras correlaciones débiles. La situación ideal es la no correlación entre atributos, de forma que cada uno contenga información única, no compartida por otro atributo. En general la figura muestra buen nivel de independencia de los atributos, aunque hay algunos fuertemente correlacionados. En futuras líneas de trabajo podría explorarse la posibilidad de reducir la dimensionalidad mediante PCA.

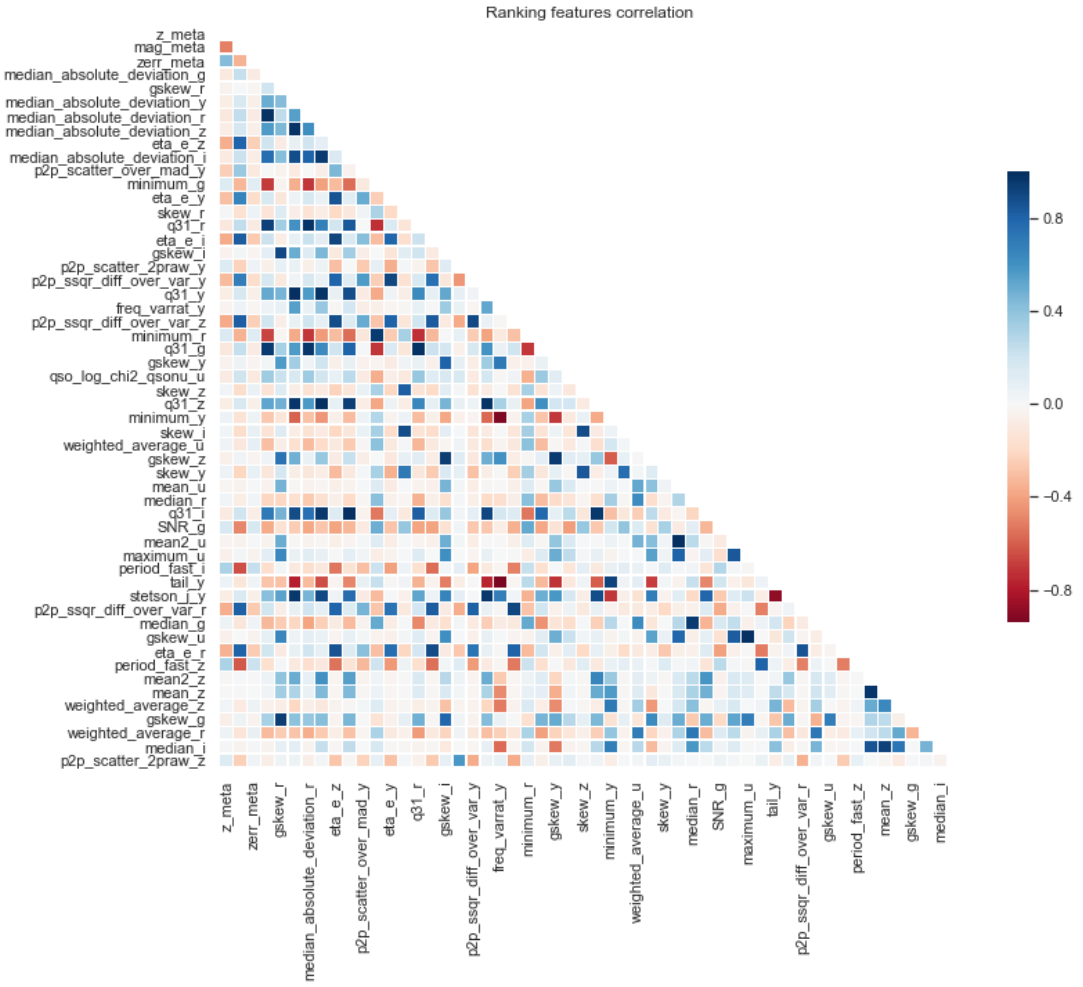


Figura 24. Matriz de correlaciones entre atributos

4.12.6 Otros experimentos

En este apartado se comentan brevemente otros experimentos realizados sobre los datos que no han mejorado el modelo y han sido descartados en fases anteriores a las pruebas finales.

Estimación de flujo en la fuente

Con el fin de normalizar las observaciones de *flujo* en los datos se calcula la distancia hasta el objeto a partir del *desplazamiento al rojo* fotométrico mediante la relatividad general. Después se multiplica el flujo por el cuadrado de dicha distancia. Esta operación no solo no ha mejorado los resultados, sino que en el clasificador de 4RF ha producido un notable empeoramiento. Esto puede deberse a la imprecisión de Z_f. Otra posible explicación es que al manejar distancias en órdenes de magnitud cósmicos, ciertos valores de flujo de los objetos más alejados alcanzan el valor máximo del tipo de dato, de manera que quedan truncados induciéndose así un sesgo importante en los objetos situados a partir de determinado desplazamiento al rojo.

Regresión de Z_f

Mediante *máquinas de soporte de vectores* se ha implementado una regresión para estimar Z_f a través Z_s, de manera que al disponer de los datos reales se aplique la regresión con el fin de corregir la estimación fotométrica. Para ello se ha usado como conjunto de entrenamiento el aquellos objetos de los datos tales que:

$$1 + (1 - T) \geq \frac{Z_f}{Z_s} \leq T,\ T \in (0, 1)$$

donde Z_f y Z_s son los desplazamientos al rojo del objeto y T es un escalar arbitrario. Se ha fijado T en 0,9, de forma que los objetos que tienen un cociente entre ambas variables entre 0.9 y 1,1 sirvan de conjunto de entrenamiento. La idea, por tanto, era entrenar un modelo con las mejores muestras para corregir las peores.

En la aplicación práctica esta regresión no funciona, dado que los objetos con esa relación entre ambas variables tienen valores de Z_f menores de *1,5* mientras que las grandes diferencias de correlación se dan en los objetos con grandes valores de Z_f. Así, la regresión resultaba prácticamente lineal, en lugar de tomar una curva polinómica, como sería deseable

Aproximación bayesiana de Z_f

De manera análoga a la reducción de ruido realizada en el *flujo* (véase apartado 4.4.2) se ha intentado buscar el valor más probable de Z_f. El método asume normalidad en la distribución anterior (que genera la muestra) y estima el valor más probable en dicha distribución. Estudiando la distribución de los valores de Z_f se aprecia que tiene una distribución asimétrica muy particular, y se considera que esta es la razón por la que la estimación empeora los resultados en los clasificadores.

5. Conclusiones

5.2 evaluación del cumplimiento de objetivos

Los objetivos generales se han cumplido. Por una parte, se ha resuelto el problema de clasificación analizando y transformando los datos mediante técnicas de minería de datos para después implementar varios clasificadores mediante técnicas de aprendizaje computacional, comparando finalmente sus rendimientos. Por otro lado, el modelo se ha implementado y validado con métricas que permiten su evaluación en términos de precisión (*accuracy*) y error cometido.

Se discute a continuación en cumplimiento de los objetivos específicos:

1. Se ha analizado la composición de los datos de manera que se ha podido construir un modelo para clasificarlos teniendo en cuenta sus particularidades

2. Se han extraído 553 nuevos atributos a los datos tras la transformación de las series temporales en curvas de luz y además se ha seleccionado los más relevantes

3. Se han implementado tres clasificadores con bosques aleatorios y comparado sus rendimientos

4. No se ha podido tratar como mejorar los recursos de cómputo por razones de tiempo. Así, convendría revisar ciertos procesos para hacer el código más eficiente, principalmente en los procesos ne y bnr, implementando procesamiento paralelo para reducir los tiempos de cómputo.

5.3 evaluación de la planificación

La planificación ha sido deficiente. Se programó un calendario de desarrollo lineal que hubo de ser descartado antes de llegar a la mitad del ciclo de vida del proyecto y replanteado mediante un enfoque más iterativo. Así, la primera versión del clasificador hubiera debido estar planificada para mucho antes y debiera haberse planificado mucho más tiempo para ensayos sobre el modelo y depuración de errores. En la práctica, esto ha supuesto el no poder profundizar en algunos experimentos interesantes por falta de tiempo. Por otro lado, se han cometido errores conceptuales en el modelo que se han detectado cerca de su finalización, por lo que ha sido necesario rehacer buena parte del modelo, no pudiendo seguirse la planificación fijada para la parte final del proyecto.

5.5 Evaluación de la metodología

El trabajo en un entorno de desarrollo en Python, y en particular con las librerías Astropy y Cesium, ha supuesto un dificultad importante en el desarrollo del proyecto. Determinados procesos, como la extracción de atributos o la construcción de clasificadores con múltiples bosques aleatorios, han excedido el tiempo programado. Dado que son ciertos procesos son muy cotosos en términos de tiempo, la depuración de errores ha resultado muy dificultosa, al aparecer algunos de estos errores bien avanzado el proceso.

El manejo de datos con una dimensionalidad tan elevada dificulta su observación, y por tanto la toma correcta de decisiones, por lo que ha sido necesario perfeccionar las competencias para ello. En general ha sido un proceso de lento aprendizaje, sobre todo en la primera mitad del ciclo de vida del proyecto. Se han mostrado muy útiles las competencias adquiridas en las asignaturas de Aprendizaje Computacional, Inteligencia Artificial y estadística.

5.6 Futuras líneas de trabajo

Como se ha visto, Z_f es el atributo de mayor importancia, de manera que entre las futuras líneas de trabajo ha de considerarse la forma de aproximar más dicho atributo al desplazamiento al rojo real del objeto. Por otro lado, como se ha mencionado en la Sección 4.11.3, y tal como proponen Beare et al. [12], la corrección del color en función del desplazamiento al rojo podría suponer una interesante mejora para el clasificador. Otra línea de trabajo interesante sería sustituir o combinar los bosques aleatorios con otros tipos basados en otros paradigmas como *deep learning* o *ensemble learning*.

La extracción de atributos tiene lugar para cada banda de un objeto, de manera que dichos atributos son extraídos a partir únicamente de la relación entre tiempo, flujo y error para una única banda. Sin embargo, si se pudiese extraer atributos basados en la relación entre varia bandas, se podría, por ejemplo, detectar el color de forma dinámica mediante la relación entre sus flujos. En la misma línea, se podrían calcular fases entre las señales de diferentes bandas y convertirlas en atributos.

5. Bibliografía

[1] *Big Universe, Big Data: Machine Learning and Image Analysis for Astronomy - Jan Kremer et al. 2017 IEEE INTELLIGENT SYSTEMS 1541-1672/17 16-22*

[2] *Random forest algorithm for classification of multiwavelength data - Dan Gao et al. 2009 Res. Astron. Astrophys.* ***9*** *220*

[3] *Breiman, L. (2001). Random forests. Machine Learning, 45, 5–32*

[4] *LightGBM: A Highly Efficient Gradient Boosting Decision Tree - Guolin Ke et al. advances in neural information processing systems 30-2017*

[5] *An Empirical Comparison of Supervised Learning Algorithms - Rich Caruana caruana, Alexandru Niculescu-Mizil. ICML '06 Proceedings of the 23rd international conference on Machine learning 161-168*

[6] *Support Vector Regression Machines - Harris Drucker· Chris J.C. Burges, Linda Kaufman, Alex Smola, Vladimir Vapoik. Bell Labs and Monmouth University Department of Electronic Engineering West Long Branch. NJ 07764 BellLabs - A T&T Labs*

[7] https://www.statlect.com/fundamentals-of-statistics/normal-distribution-Bayesian-estimation *visitada el 1/05/2019*

[8] http://cesium-ml.org/docs/feature_table.html *visitada el 20/04/2019*

[9] https://feets.readthedocs.io/en/latest/ visitada el 20/04/2019

[10] *Lloyd, S. P. (1957). «Least square quantization in PCM». Bell Telephone Laboratories Paper.*

[11] *Hubble, E.P., (1937) The observational approach to cosmology. The Clarendon Press, Oxford.*

[12] *An accurate new method of calculating absolute magnitudes and K-corrections applied to the Sloan filter set. RICHARD BEARE , MICHAEL J. I. BROWN, KEVIN PIMBBLET , (October 31, 2014)*

[13] *On the Color-Magnitude Diagram of the Pleiades, H. L. Johnson, W. W. Morgan, ApJ 114, 522 (1951).*

[14] *https://en.wikipedia.org/wiki/Conjugate_prior visitada el 2/06/2019*

[15] https://www.lsst.org *visitada el 12/06/2019*

6. Anexos

i) Curvas de luz y distribución de flujo en las clases

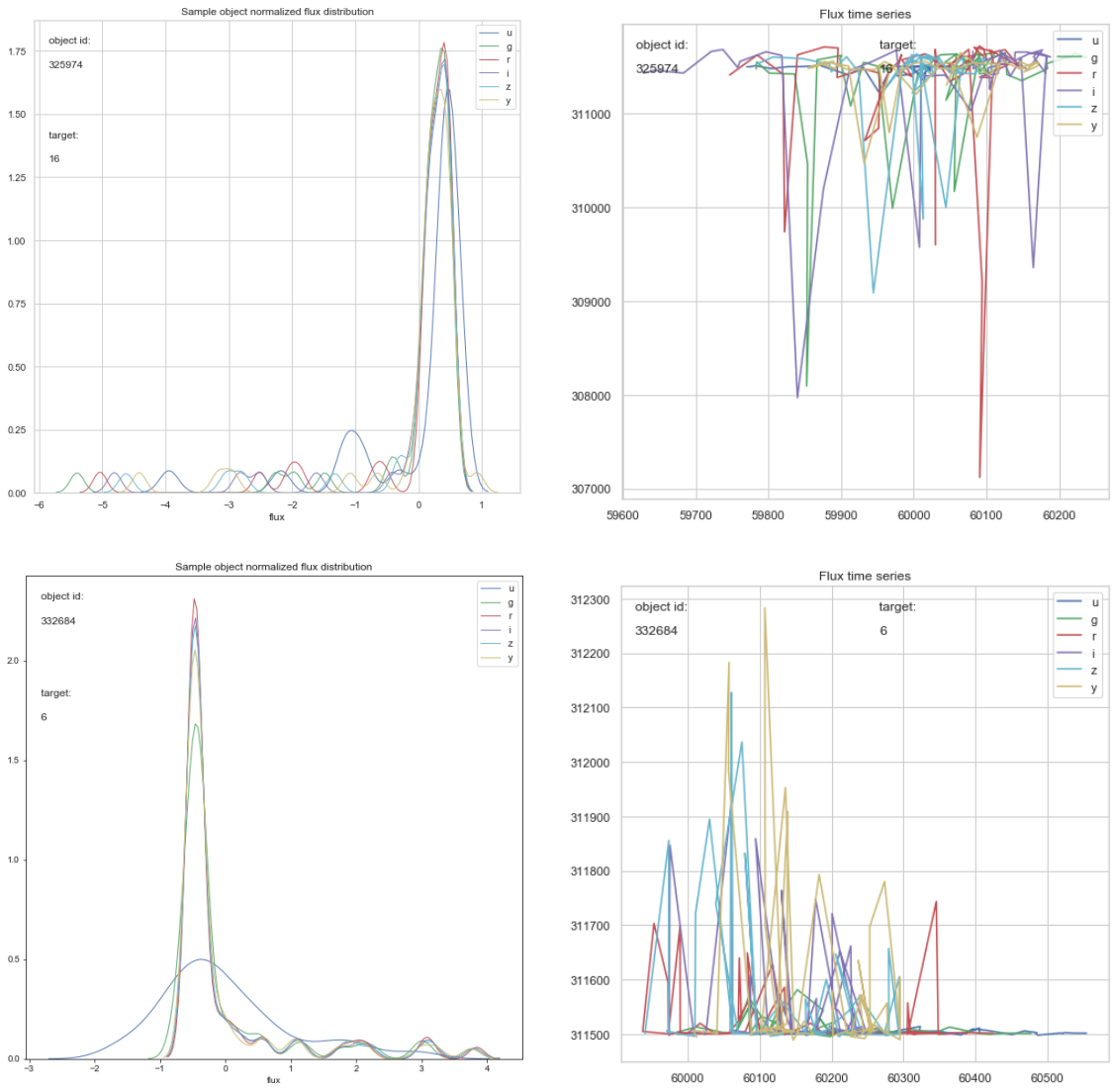


Figura A1. Distribución del flujo (izquierda) y curva de luz (derecha) en objetos galácticos (ii)

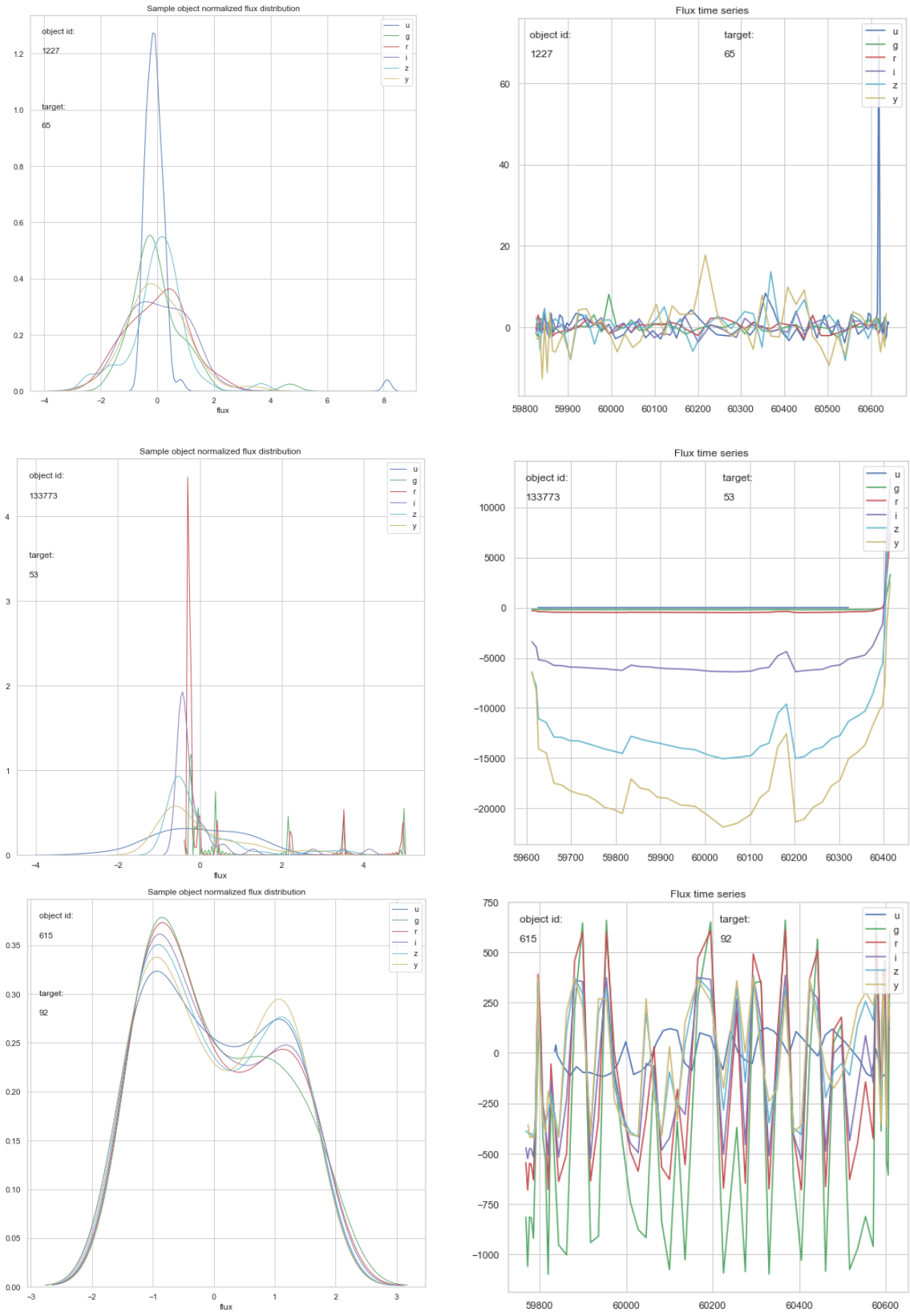


Figura A2. Distribución del flujo (izquierda) y curva de luz (derecha) en objetos galácticos (ii)

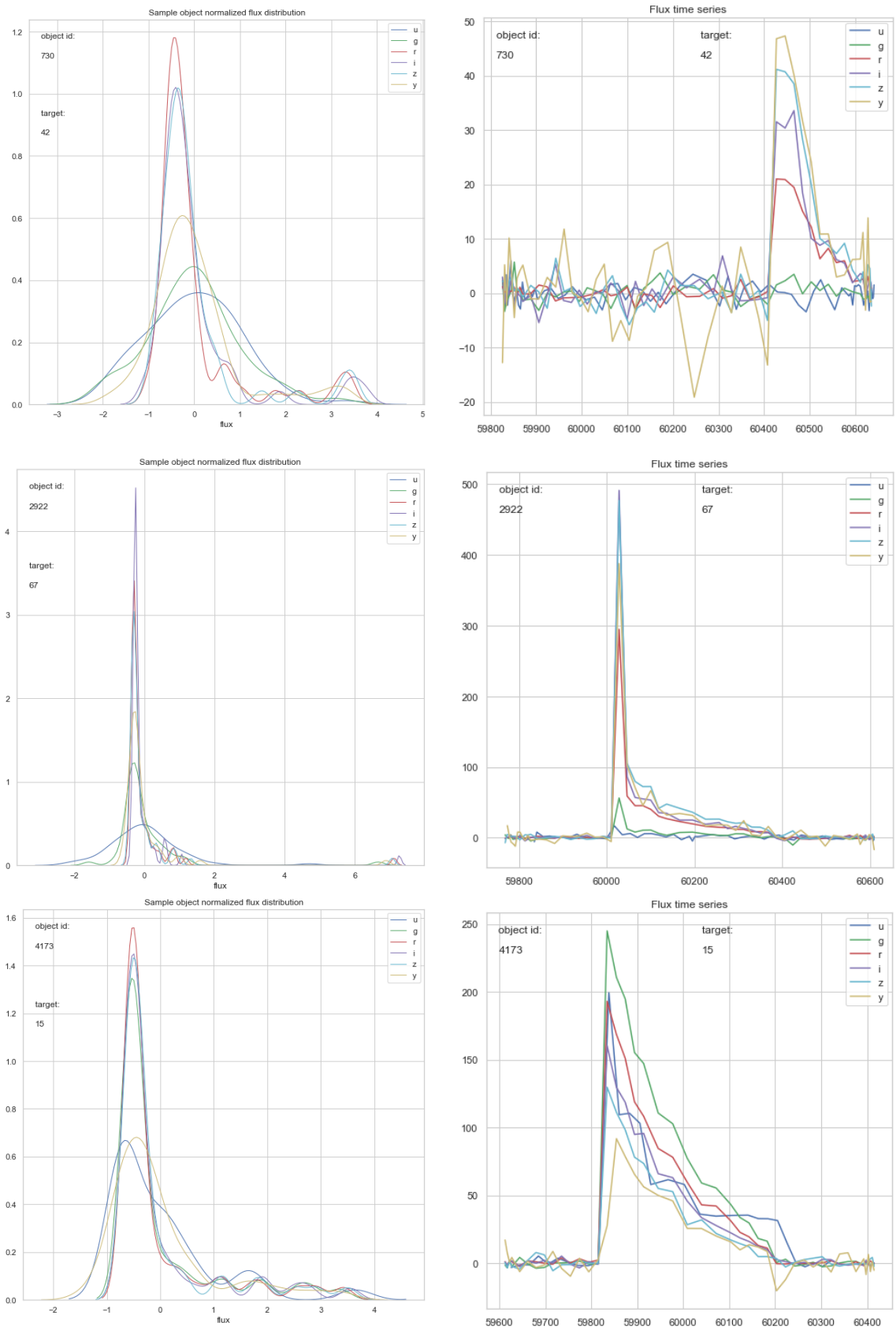


Figura A3. Distribución del flujo (izquierda) y curva de luz (derecha) en objetos extragalácticos (i)

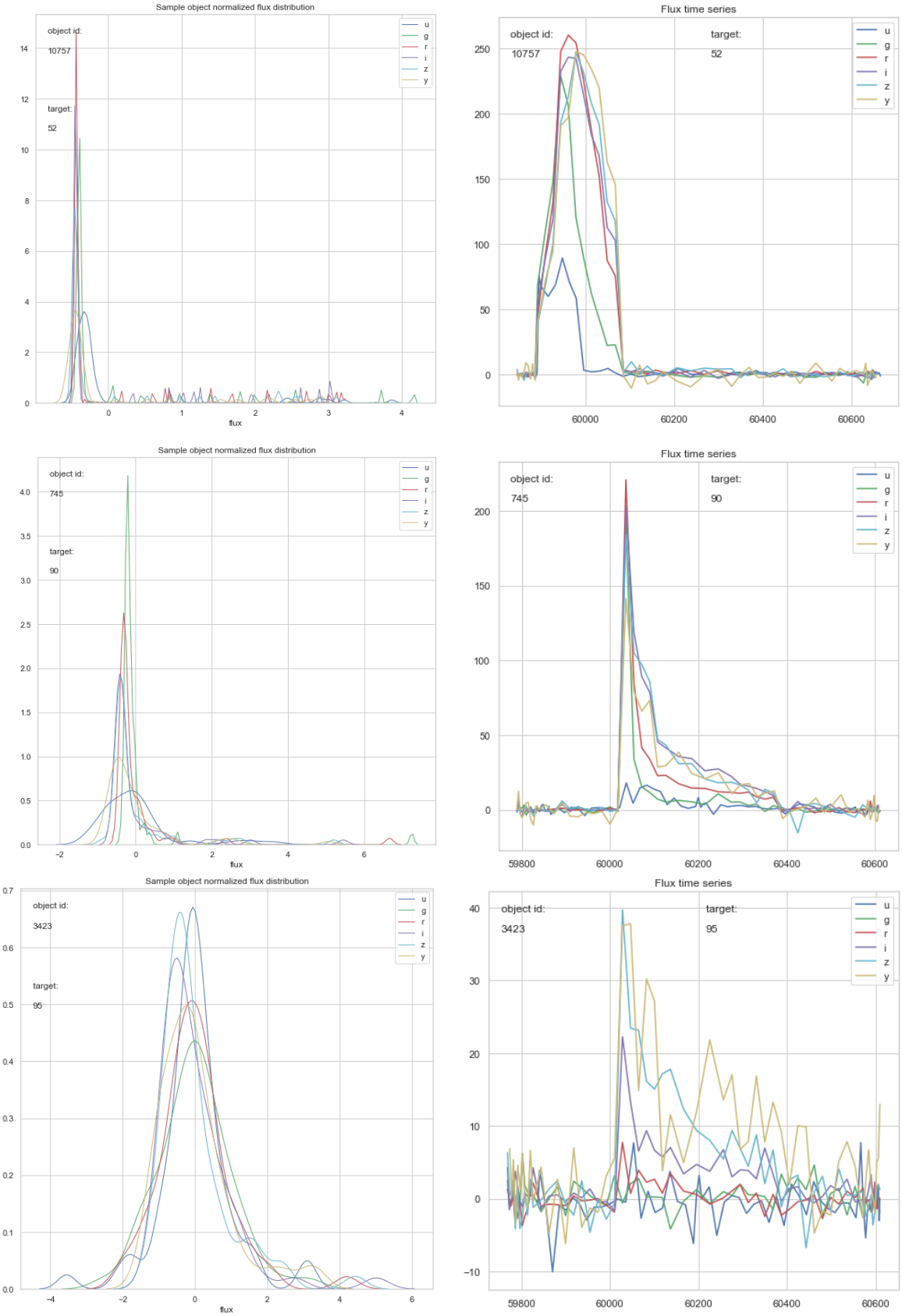


Figura A4. Distribución del flujo (izquierda) y curva de luz (derecha) en objetos extragalácticos (iii)

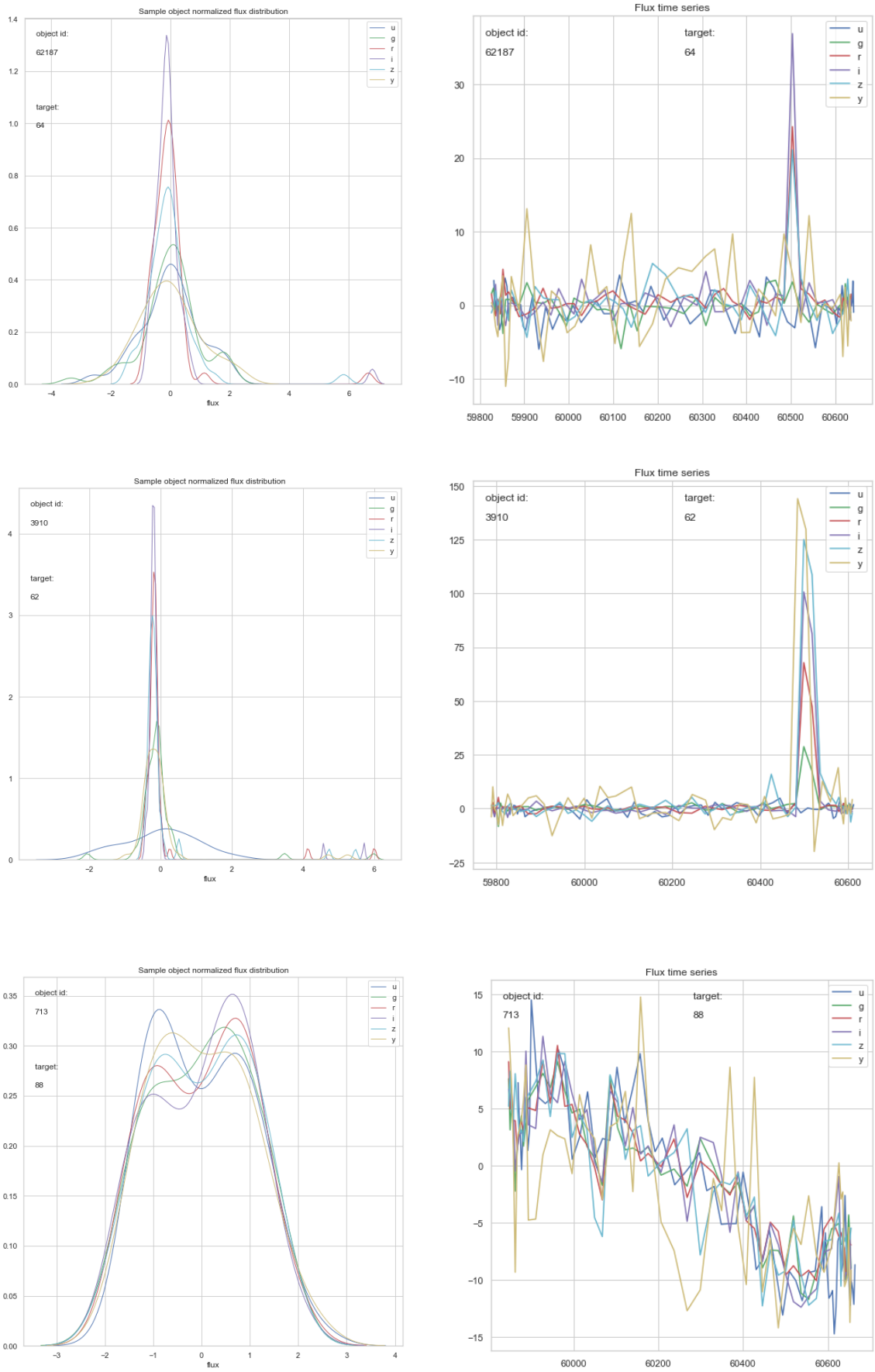


Figura A5. Distribución del flujo (izquierda) y curva de luz (derecha) en objetos extragalácticos (iv)

ii) Ejemplo de ejecución del modelo

```
PLaSTiCC CLASSIFICATION MODEL COSTRUCTION
=========================================

@author: luisarribas UOC 2019


LOADING DATA
============


OK!


FLUX NOISE REMOVAL
==================
Samples 5 sigma far or more from mean are deleted.
Samples with error greater than 3 times flux error are deleted.

NOISE ELIMINATION PROCESS RESULTS
=================================

Deleted samples:  1506
Sample percentage reduction: 0.20%

Saving results to  /Users/luisarribas/plasticc/data/noise_red.csv

OK!


FLUX BAYESIAN NOISE REDUCTION
=============================
Closing distant flux values from Mean value by Bayesian aproach


OK!


COMPUTING MAGNITUDES
```

```
====================

Calculating object´s magnitude as it´s actual
brightness from a standard distance of 10 pc


OK!


COMPUTING COLOR
====================

Calculating object´s color by
Johnson-Morgan scale


OK!


EXTRACTING FEATURES
====================

Building Timeseries....wait
============================

OK!


Computing features....wait
============================
Unexpected error: gskw
Unexpected error: gskw
Unexpected error: gskw
Unexpected error: gskw
Unexpected error: gskw
Unexpected error: gskw
Unexpected error: gskw
Unexpected error: gskw
Unexpected error: gskw
Unexpected error: gskw

OK!
```

```
Extracted features =  553
==========================

Nan Values detected =  0
==========================

CLASSIFYING DATA WITH 553 FEATURES
====================================

ACCURACY: 0.70


CLASSIFYING DATA WITH 197 FEATURES
====================================

ACCURACY: 0.71


CLASSIFYING DATA WITH 101 FEATURES
====================================

ACCURACY: 0.71


CLASSIFYING DATA WITH 56 FEATURES
====================================

ACCURACY: 0.71


CLASSIFYING DATA WITH 100 FEATURES
====================================

ACCURACY: 0.72 (+/- 0.00)


CLASSIFYING DATA WITH 100 FEATURES
====================================

ACCURACY: 0.71 (+/- 0.00)
```

```
FEATURE REDUCTION RESULTS
=========================

1 RANDOM FOREST MODEL LOGLOSS:0.910:

1 RANDOM FOREST MODEL ACCURACY 0.72

 LAPSED  TIME:1237.652687
```

```
FEATURE RANKING
===============
         importance                   name                 cum_import
---------------------- -------------------------- --------------------
  0.07581577542063495                     z_meta 0.07581577542063495
  0.05567914341991905                   mag_meta   0.131494918840554
  0.04342527447826158                  zerr_meta  0.1749201933188156
 0.021767545961025196 median_absolute_deviation_g  0.1966877392798408
  0.01756489473920864 median_absolute_deviation_z 0.21425263401904943
 0.016997271008127174 median_absolute_deviation_r  0.2312499050271766
 0.015108090842210814 median_absolute_deviation_y 0.24635799586938742
 0.014424870801625217                    gskew_r 0.26078286667101264
  0.01369112647388836                     skew_r   0.274473993144901
  0.01260467896687432                  minimum_r 0.28707867211177535
                  ...                        ...
 0.003634309907685965           small_kurtosis_z  0.7601628687292886
0.0035349093142432396                     peak_z  0.7636977780435319
0.0034905997508267407                  minimum_z  0.7671883777943587
0.0034850074332790524                     skew_u  0.7706733852276377
 0.003467969439662431                abs_diffs_r  0.7741413546673002
 0.003445270202915413                   median_i  0.7775866248702156
0.0033848571859534378                     mean_y   0.780971482056169
0.0032999743337404318         freq1_amplitude1_z  0.7842714563899094
 0.003263523662460626                     tail_z    0.78753498005237
0.0031970985434518046           autocor_lenght_z  0.7907320785958218
 0.003162292399813018              period_fast_z  0.7938943709956349
Length = 100 rows
```

```
MODEL TEST WITH AN UBIASED  2000 SAMPLE DATASET
================================================================

FLUX NOISE REMOVAL
==================
Samples 5 sigma far or more from mean are deleted.
Samples with error greater than 3 times flux error are deleted.

NOISE ELIMINATION PROCESS RESULTS
=================================

Deleted samples:  4445
Sample percentage reduction: 0.68%

Saving results to  /Users/luisarribas/plasticc/data/test_noise_red.csv

OK!


FLUX BAYESIAN NOISE REDUCTION
=============================
Closing distant flux values from Mean value by Bayesian aproach


OK!


COMPUTING MAGNITUDES
====================

Calculating object´s magnitude as it´s actual
brightness from a standard distance of 10 pc


OK!


COMPUTING COLOR
====================

Calculating object´s color by
Johnson-Morgan scale
```

```
OK!


EXTRACTING FEATURES
====================

Building Timeseries....wait
============================

OK!


Computing features....wait
============================

OK!


Extracted features =  553
============================

Nan Values detected =  0
============================

1 RANDOM FOREST CLASSIFYER TEST

CROSS VALIDATION ACCURACY: 0.81 (+/- 0.03)


2 RANDOM FOREST CLASSIFYER TEST
================================

CROSS VALIDATION ACCURACY: 0.80 (+/- 0.04)


GALACTIC ACCURACY: 0.98 (+/- 0.02)


EXTRA-GALACTIC ACCURACY: 0.74 (+/- 0.05)
```

```
4 RANDOM FOREST CLASSIFYER TEST
==============================

CROSS VALIDATION ACCURACY: 0.79 (+/- 0.05)


GALACTIC ACCURACY: 0.98 (+/- 0.02)

CLUSTER 0 ACCURACY: 0.74 (+/- 0.05)

CLUSTER 1 ACCURACY: 0.82 (+/- 0.11)

CLUSTER 2 ACCURACY: 0.45 (+/- 0.14)
```