GRADO EN CIENCIA DE DATOS

Desarrollo de un sistema de aprendizaje automático para la detección de eventos astronómicos transitorios: Aplicación a datos de MANTRA

Presentado por:

PABLO BAS GENÍS

Dirigido por:

JOSE MARÍA FERNÁNDEZ NAVARRO

CURSO ACADÉMICO 2024-2025

Agradecimientos

En primer lugar me gustaría dedicar unas palabras a todas las personas que han estado ahí conmigo en esta etapa universitaria tan importante y fundamental en mi vida, ya que al principio empezó como una etapa más en la vida de cualquier persona que entra en la Universidad para estudiar lo que tanto le gusta y le apasiona, pero por circunstancias de la vida, en el tercer año de carrera, me vi obligado a abandonar la carrera durante un año debido a una enfermedad que por desgracia, a día de hoy sigue acabando con la vida de tantas personas en el mundo, el cáncer.

Por suerte, en mi caso, después de una operación de más de 14 horas y más de 3 meses en el hospital, conseguí superar esa enfermedad tan cruel, y no solo es que conseguí salir, si no que aquello me sirvió para despegar en mi vida, y gracias a esa enfermedad, conseguí darme cuenta de lo poco que valoramos las cosas, hasta cosas tan simples como son el comer y el habla, ya que son las cosas que perdí durante casi un año y que, por suerte, he vuelto a recuperar también. Hay veces en la vida que hace falta que nos pasen cosas así o que venga alguien a decírnoslo para darnos cuenta de lo poco que valoramos la mayoría de las cosas, pero gracias a esta enfermedad, he aprendido a valorar la vida, a valorar a las personas que están a tu lado no solo en los momentos buenos, si no las que están en los momentos malos. He aprendido a valorar cada segundo, cada momento, cada anécdota, cada experiencia, cada aprendizaje, porque, al fin y al cabo la vida se basa en eso, aprender, disfrutar y sobre todo y lo más importante vivir.

En esta etapa universitaria he podido aprender muchísimas cosas que van a ser vitales para desarrollar mi vida laboral como siempre he soñado, y todo es gracias a la Universidad Europea, a los maravillosos profesores que me han aportado todo el conocimiento posible y también a mis compañeros, quienes me han ayudado y apoyado en todo momento, tanto en el ámbito universitario como en el ámbito de la vida.

De nuevo, lo único que puedo es agradecer a todas y cada una de las personas que han estado ahí conmigo en esta etapa tan buena, pero a la vez tan dura de mi vida, primero a mi familia por haber estado ahí conmigo en todo momento, desde el principio al final. Por otra parte, me gustaría dar las gracias a mis amigos y a los compañeros de la Universidad,

quienes también me demostraron que los amigos y compañeros de verdad son aquellos que están, no solo en las buenas, si no en todas las circunstancias. Y, por último, quería agradecer especialmente a la Universidad, todo el apoyo que me han dado, a todas y cada una de las personas que la conforman por ser tan comprensivos y pacientes desde el principio y en vez de ponerme pegas o problemas a raíz de todo lo que pasé, lo único que han hecho es ponerme facilidades y estar pendientes de mí en todo momento.
Estaré eternamente agradecido con todos. Gracias.

Índice

Contenido

1.0 Introducción 7
1.1 Motivación 8
1.2 Objetivos 12
1.3 Estructura 14
1.3.1 Capítulo 1. Introducción 14
1.3.2 Capítulo 2. Estado del arte 14
1.3.3 Capítulo 3. Propuesta 14
1.3.4 Capítulo 4. Proceso de implementación 15
1.3.5 Capítulo 5. Producción, resultados y aplicaciones reales 15
1.3.6 Capítulo 6. Conclusiones y líneas futuras 16
1.3.7 Capítulo 7. Bibliografía 16
2.0 Estado del arte 16
2.1 Análisis del estado del arte 16
2.1.1 Eventos astronómicos transitorios: definición e importancia científica 17
2.1.2 El reto del Big Data en astronomía transitoria 19
2.1.3 Aprendizaje automático en la Ciencia de Datos 20
2.1.4 Bases de datos para eventos astronómicos transitorios 22
2.1.5 Aplicación del aprendizaje automático a la detección de objetos transitorios 24
2.1.6 Limitaciones y desafíos actuales 24
3.0 Propuesta 27
3.1 Planteamiento de la solución 27
3.2 Base de datos real: MANTRA 29
3.3 Procesamiento de los datos 33
3.4 Modelos de aprendizaje automático usados 35
3.5 Elección del mejor modelo 38
4.0 Proceso de implementación 40
4.1 Metodología 40
4.2 Sprint Planning 42
5.0 Producción, resultados y aplicaciones reales 46
5.1 Código y base de datos elegida 46
5.2 Elección del modelo o modelos idóneos 47
5.3 Resultados obtenidos 52
5.4 Aplicación a otros telescopios y fuentes de datos 55
5.4.1 Aplicación a ZTF (Zwicky Transient Facility). 55

5.4.2 Aplicación a LSST (Legacy Survey of Space and Time) 57
6.0 Implicaciones sociales, económicas y sostenibles del TFG 58
6.1 Impacto socioeconómico 58
6.2 Contribución del TFG a los Objetivos de Desarrollo Sostenible (ODS) 61
7.0 Conclusiones y líneas futuras del proyecto 63
7.1 Conclusiones 63
7.2 Usabilidad real con Big Data 64
7.3 Líneas futuras del TFG 65
7.3.1 Detección de eventos transitorios en tiempo real 65
8.0 Glosario de términos, ilustraciones y tablas 67
8.1 Glosario 67
8.2 Ilustraciones y tablas 70
Bibliografía 71

1.0 Introducción

El estudio del cosmos se ha visto revolucionado por la recopilación masiva de datos astronómicos a través de telescopios espaciales como el Catalina Real-Time Transient Survey (CRTS). Estos telescopios generan una cantidad inmensa de datos en tiempo real, permitiendo la observación y caracterización de eventos astronómicos transitorios como supernovas, estrellas variables y asteroides en movimiento. Sin embargo, la magnitud de estos datos plantea el reto de desarrollar métodos eficientes para su análisis automatizado.

El aprendizaje automático ha demostrado ser una herramienta **poderosa** en la identificación de patrones en grandes volúmenes de datos. Su aplicación en astronomía ha permitido mejorar la detección y clasificación de eventos transitorios de manera más rápida y precisa que los métodos tradicionales. Este trabajo de fin de grado tiene como objetivo el desarrollo de un sistema de aprendizaje automático para la detección automática de eventos astronómicos transitorios. Para ello, lo más importante es disponer de una base de datos que vaya acorde de la magnitud de este proyecto. En este caso, los datos que nos proporciona MANTRA (Machine-learning Analysis of Transients with Reference lightcurves for Astronomy). Se trata una base de datos generada a partir del telescopio CRTS por la Universidad de los Andes de Colombia con el fin de mejorar la eficiencia y precisión en la clasificación de eventos astronómicos transitorios. Dicha base de datos es idónea para la realización de este trabajo ya que está creada única y exclusivamente para la creación de modelos de aprendizaje automático, los cuales puedan contribuir para la optimización y detección de estos objetos transitorios.

La **relevancia** de esta investigación radica en la optimización de la detección de eventos astronómicos de interés, lo que puede contribuir a un mejor entendimiento de la evolución estelar, la estructura del universo y la caracterización de cuerpos celestes en movimiento, puesto que, a día de hoy disponemos de una cantidad ingente de datos proveniente de muchos telescopios, los cuales no paran de enviar información y todavía no disponemos de modelos ni herramientas suficientes para procesar toda esa cantidad de datos y toda esa información. Este proyecto es muy relevante porque puede contribuir de una manera muy significativa en la optimización y detección de dichos objetos.

Además, este trabajo se **alinea** con los esfuerzos de la ESA (European Space Agency) y la NASA (National Aeronautics and Space Administration) para potenciar la exploración y análisis del cosmos mediante herramientas de inteligencia artificial, ya que cada vez son más los telescopios disponibles y por ende son mayores las cantidades de datos de las cuales disponemos, y se necesitan modelos de inteligencia artificial que vayan acorde a la magnitud de este gran problema.

1.1 Motivación

Desde que nací, la astronomía es un tema que me ha apasionado mucho, y me di cuenta cuando lo único que hago al llegar la noche es asomarme al balcón a ver las estrellas y La Luna. Por ello, desde que supe que iba a hacer una carrera universitaria, tenía muy claro que mi TFG tenía que ir relacionado con ello. La carrera de Ciencia de Datos es idónea para la realización de un trabajo de fin de grado con la suficiente magnitud e importancia acordes a la magnitud del Universo.

En la última década, la astronomía del tiempo se ha **consolidado** como un área clave para el descubrimiento de fenómenos transitorios en el universo. Los objetos astronómicos son eventos celestes de corta duración relativa, cuya luminosidad cambia de forma notable en escalas de tiempo humanas. Alguno de los ejemplos de objetos transitorios más comunes incluye: explosiones estelares catastróficas como supernovas, destellos en estrellas fulgurantes (flares), estallidos de rayos gamma ópticos o incrementos de brillo en núcleos activos de galaxias (AGN) y blazares, entre otros.

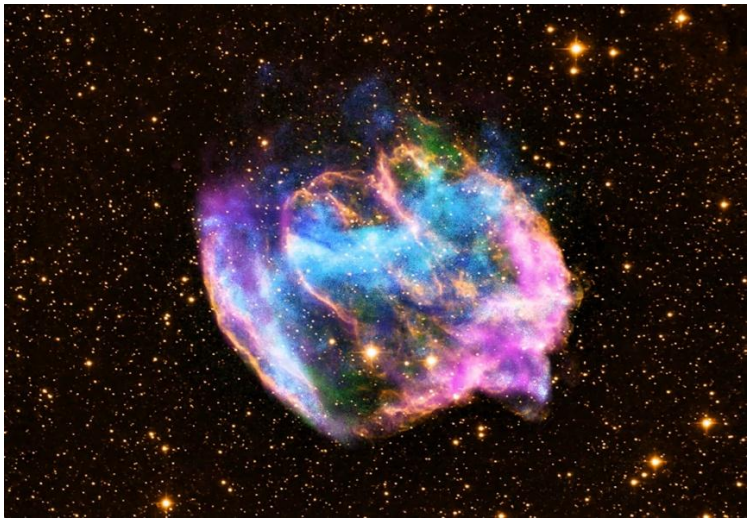

Ilustración 1. Supernova

Ilustración 2. Destellos en estrellas fulgurantes (flares)

Ilustración 3. Estallido de rayos Gamma

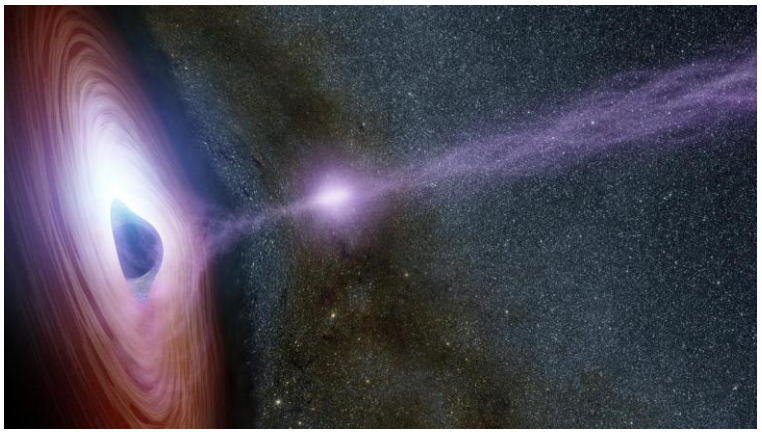

Ilustración 4. Incremento del brillo de núcleo activo de Galaxia (AGN)

Cada una de estas clases de objetos transitorios tiene un interés científico particular. Por ejemplo, las supernovas permiten estudiar la muerte de estrellas masivas y medir la expansión acelerada del universo y por otra parte los AGN (Núcleo activo de galaxia en inglés) y los blazares los cuales son un tipo de núcleos activos de galaxias muy comunes, revelan la física de agujeros negros supermasivos activos. Por tanto, detectar y caracterizar dichos objetos transitorios es fundamental para aprovechar su valor científico en áreas que van desde la cosmología hasta la física estelar.

Tradicionalmente, la identificación y clasificación de eventos transitorios requería observaciones espectroscópicas de seguimiento o el análisis manual de sus curvas de luz por astrónomos expertos. Sin embargo, ante la explosión en la cantidad de datos de los sondeos modernos, estos métodos manuales se han vuelto inviables. Como motivación concreta, consideremos que el sondeo CRTS produjo miles de curvas de luz de transitorias en unos pocos años. Hoy en día, proyectos de nueva generación como ZTF generan del orden de 100 mil alertas transitorias cada noche, y la futura LSST alrededor de 10 millones de alertas por noche, con decenas de miles de eventos potenciales por cada imagen tomada.

Ningún equipo humano puede revisar tal volumen de alertas en tiempo real, de modo que la automatización mediante técnicas de aprendizaje automático se vuelve imprescindible. Esta necesidad de procesar eficientemente el Big Data astronómico impulsa el desarrollo de algoritmos capaces de aprender a reconocer patrones sutiles en los datos de objetos transitorios, para distinguir eventos reales y clasificar la naturaleza de cada detección.

Afortunadamente, el avance de las técnicas de machine learning (aprendizaje automático) y el incremento en el poder computacional han abierto la puerta a abordar este problema con **éxito**. En particular, los algoritmos de aprendizaje automático pueden ser entrenados para reconocer las firmas características en curvas de luz que diferencian unos tipos transitorios de otros. Por ejemplo, una supernova típica exhibe un aumento rápido de brillo seguido de una decadencia suave en el transcurso de semanas; un AGN o un blazar presenta variaciones aleatorias más moderadas; una estrella fulgurante tiene picos breves de brillo muy pronunciados ya que principalmente son los rayos de luz de dichas estrellas.

Identificar correctamente estas diferencias en datos ruidosos es complejo, pero diversos estudios previos han demostrado que modelos de machine learning pueden lograrlo con alta eficacia. Desde inicios de la década de 2010 se han aplicado técnicas de inteligencia artificial en astronomía transitoria: por ejemplo, Djorgovski et al. (2012) implementaron prototipos de clasificación automática en el conjunto de telescopios CRTS empleando redes neuronales y SVM para distinguir eventos astronómicos reales de falsos positivos.

Más recientemente, investigadores de alertas como ALeRCE han logrado clasificar en tiempo real decenas de miles de objetos transitorios del telescopio ZTF mediante algoritmos como Random Forest y XGBoost entrenados con grandes bases de datos históricas, alcanzando precisiones muy significativas. Estos avances sugieren que es factible construir sistemas automatizados que identifiquen y clasifiquen objetos transitorios con rapidez y fiabilidad, abordando así el reto científico y de datos que plantean los nuevos sondeos.

En este contexto y como bien se ha explicado, el presente proyecto busca **desarrollar** un sistema de clasificación automática de objetos transitorios empleando datos reales de un sondeo de datos preexistente, y sentar las bases para su uso en escenarios de Big Data astronómico. Se pretende mostrar que, con un conjunto de datos históricamente etiquetado, es posible entrenar modelos de machine learning capaces de diferenciar distintos tipos de objetos transitorios en base a sus curvas de luz, y que dichos modelos pueden luego adaptarse a los flujos masivos de alertas de encuestas modernas. Así, se combinará el interés científico de estos fenómenos con la necesidad tecnológica de procesar datos en volumen, contribuyendo al desarrollo de herramientas para la astronomía en la era de los grandes datos.

1.2 Objetivos

El objetivo principal de este trabajo es desarrollar un sistema automático de clasificación de objetos transitorios basado en el aprendizaje automático, capaz de distinguir las principales clases de eventos transitorios a partir de sus datos fotométricos. Este objetivo central se desglosa en los siguientes objetivos específicos para guiar el desarrollo del proyecto:

- **Construir** un conjunto de datos de entrenamiento representativo a partir de una base de datos real de objetos transitorios (MANTRA/CRTS), realizando las tareas de limpieza y preprocesamiento necesarias. Esto incluye filtrar objetos con muy pocas observaciones, unificar formatos de identificadores y etiquetas, y consolidar las clasificaciones originales en un conjunto manejable de clases principales.
- **Extraer** características informativas de las curvas de luz de dichos objetos transitorios, basadas en métricas estadísticas y de variabilidad. Se buscará definir un conjunto de características (features) que capten los aspectos clave de la forma de la curva (por ejemplo, su amplitud de variación, dispersión, forma de la distribución de magnitudes, duración del evento, número de observaciones, etc.) que ayuden a diferenciar automáticamente las clases astrofísicas.
- **Entrenar** y **evaluar** múltiples modelos de clasificación supervisada sobre los datos preparados. En particular, se entrenarán varios algoritmos representativos (árboles aleatorios (Random Forest), SVM, redes neuronales, métodos de boosting) bajo condiciones homogéneas, y se comparará cuantitativamente su desempeño. Esto conlleva ajustar adecuadamente los parámetros, aplicar técnicas para mitigar el desbalance de clases presente en el conjunto de objetos transitorios como puede ser el exceso de muestreo de las clases minoritarias añadiendo ruido a curvas de luz reales y/o asignación de pesos en la función de coste de entrenamiento y validar los modelos en datos no vistos.
- **Evaluar** rigurosamente y comparar los modelos entrenados, utilizando un conjunto de métricas apropiadas (exactitud global, precisión, recall y puntaje F1, tanto a nivel macro promedio como por clase). Dada la naturaleza desbalanceada del problema, se pondrá especial énfasis en métricas como el F1 (promedio entre clases) para tener una visión equilibrada del rendimiento. Asimismo, se generarán matrices de confusión para analizar los patrones de error (qué clases se confunden entre sí) y curvas ROC para inspeccionar la separabilidad de cada clase. Este análisis proporcionará perspectivas científicas

fundamentales como, por ejemplo, qué tipos de eventos son más difíciles de distinguir y por qué, además de medir la efectividad técnica de cada modelo.

- **Seleccionar** el modelo más idóneo o una combinación de modelos para su despliegue final, considerando tanto las métricas de rendimiento como aspectos prácticos (velocidad de inferencia, interpretabilidad, facilidad de implementación en un entorno de producción de datos masivos). El modelo seleccionado deberá cumplir con el objetivo del proyecto de detectar objetos transitorios de manera automática y eficiente.
- **Demostrar** la aplicabilidad y escalabilidad del sistema en entornos reales de Big Data astronómico, extrapolando los resultados obtenidos con CRTS a encuestas actuales o futuras. Esto implica analizar cómo nuestro modelo podría integrarse en el flujo de alertas de proyectos como los telescopios de ZTF, Gaia o LSST y ver que qué adaptaciones serían necesarias (por ejemplo, incorporar múltiples filtros, nuevas clases, etc.) y evaluar si el método es computacionalmente escalable para procesar cientos de miles de alertas por noche en tiempo real.
- **Proponer** líneas futuras de trabajo para mejorar la detección en tiempo real de objetos transitorios. Esto incluirá considerar el uso de información contextual adicional (por ejemplo, datos de color, posición galáctica, comparación con catálogos) que en este trabajo no se incorpora. Explorar métodos de aprendizaje más avanzados (como redes neuronales recurrentes o aprendizaje profundo que pueda aprovechar secuencias temporales completas), y en general delinear cómo mejorar este sistema para afrontar los desafíos que traerán los próximos grandes proyectos observacionales.

En conjunto, estos objetivos cubren todo el ciclo de vida del proyecto: desde la preparación de datos hasta la evaluación y aplicación del modelo resultante, asegurando que se aborde tanto la parte científica (astronómica) como la técnica (computacional) del problema.

1.3 Estructura

El presente documento está organizado en varios capítulos, todos ellos fundamentales para el desarrollo idóneo del proyecto:

1.3.1 Capítulo 1. Introducción

Como se ha podido observar previamente en el documento, en este apartado se describen la **motivación** y los **objetivos** del proyecto, proporcionando el contexto astronómico y tecnológico del problema de la detección de objetos transitorios. Además, se adelanta la metodología general empleada y se incluye un diagrama de flujo del sistema propuesto para clarificar las etapas principales del proyecto.

1.3.2 Capítulo 2. Estado del arte

Se resumen los **conocimientos** y **trabajos** previos relevantes. Se define el concepto de objeto transitorio astronómico y su importancia científica. Por otra parte, se discute el reto que supone el Big Data en este campo. Además, se revisan los fundamentos del aprendizaje automático aplicados a la ciencia de datos. También se describen las principales bases de datos existentes de eventos transitorios y se comentan aplicaciones previas del aprendizaje automático a la detección/clasificación de objetos transitorios, y finalmente se mencionan las limitaciones y desafíos actuales identificados en este contexto.

1.3.3 Capítulo 3. Propuesta

Se detalla la **solución** planteada en este TFG. Esto incluye el planteamiento general del enfoque, la descripción de la base de datos real utilizada (MANTRA, derivada de CRTS) y un estudio exploratorio de sus datos. Además, se añadirá una explicación del

procesamiento efectuado para obtener las características (features) de las curvas de luz, los modelos de aprendizaje automático que se emplearán (incluyendo su lógica de funcionamiento y configuraciones iniciales) y la justificación de su elección. También se discute cómo se decidirá el mejor modelo tras las comparativas.

1.3.4 Capítulo 4. Proceso de implementación

Se explica la **metodología** de trabajo seguida para llevar a cabo el proyecto. En particular, se describe el enfoque de desarrollo haciendo uso de la metodología ágil con sprints y se presenta la planificación temporal con dichos sprints, detallando las tareas realizadas en cada fase. Se mencionan las herramientas y entornos utilizados (lenguajes de programación, librerías, control de versiones) y cómo se implementaron técnicamente los componentes del sistema (lectura de datos, módulos de entrenamiento, etc.).

1.3.5 Capítulo 5. Producción, resultados y aplicaciones reales

Se presentan los **resultados** obtenidos por los modelos entrenados y se **discute** su posible despliegue práctico. Primero, se describe brevemente la estructura del código desarrollado y la base de datos utilizada. Luego, se analiza la elección del modelo o modelos más idóneos según las métricas y consideraciones prácticas. A continuación, se muestran los resultados cuantitativos alcanzados (métricas globales y por clase, matrices de confusión, curvas ROC, etc.) junto con su interpretación astrofísica. Asimismo, se explora la posible aplicación del sistema a otros telescopios y fuentes de datos (por ejemplo, cómo se podría adaptar el modelo entrenado a flujos de alertas de ZTF, LSST u otros) y la escalabilidad de la solución a volúmenes de datos masivos. Es decir, se conecta el trabajo realizado con escenarios reales de Big Data y se evalúa qué tan preparado está el sistema para usarse en producción, considerando tiempos de cómputo, paralelización y adaptaciones necesarias.

1.3.6 Capítulo 6. Conclusiones y líneas futuras

Se sintetizan las **conclusiones** principales del proyecto. Se discute el grado de cumplimiento de los objetivos planteados y se reflexiona sobre la usabilidad real de este sistema en entornos de datos extremadamente grandes (por ejemplo, en LSST). Igualmente, se esbozan posibles mejoras y líneas de trabajo futuro, incluyendo la incorporación de información contextual, la detección de objetos transitorios en tiempo real y la extensión a nuevas clases de eventos, con el fin de guiar desarrollos posteriores basados en este trabajo.

1.3.7 Capítulo 7. Bibliografía

Al final del documento se incluye la **bibliografía** en formato APA 7, la cual recoge todas las referencias utilizadas a lo largo de todo el proyecto, sobre todo acerca de información relevante usada y referencias hacia otros proyectos que están alineados con este.

2.0 Estado del arte

2.1 Análisis del estado del arte

En este capítulo se revisa el contexto científico y tecnológico en el que se enmarca el trabajo de fin de grado. Primero, se define qué son los eventos astronómicos transitorios y por qué son importantes para la astrofísica (Sección 2.1.1). Luego se describe el gran desafío que supone el manejo de datos masivos en este dominio (Sección 2.1.2). Enseguida, se introducen los conceptos generales de aprendizaje automático relevantes para dicho problema (Sección 2.1.3) y se resumen las principales bases de datos disponibles de objetos transitorios (Sección 2.1.4). Posteriormente, se comentan algunos proyectos y trabajos previos que han aplicado machine learning a la detección y clasificación de objetos transitorios (Sección 2.1.5).

Y finalmente, se enumeran las limitaciones y desafíos actuales identificados en la literatura especializada (Sección 2.1.6), lo cual motivará las decisiones del proyecto.

2.1.1 Eventos astronómicos transitorios: definición e importancia científica

Los eventos transitorios astronómicos se definen como fuentes astrofísicas cuya luminosidad varía de forma significativa en escalas de tiempo relativamente cortas (días, semanas o meses), apareciendo y desapareciendo o cambiando su brillo de manera notable dentro de la vida humana. Esta definición abarca una amplia gama de fenómenos de distinta naturaleza física, pero con la característica común de ser detectables como cambios luminosos en el cielo. A continuación, se resumen las principales clases de eventos transitorios y su relevancia científica:

Supernovas (SN): Explosiones terminales de estrellas masivas que incrementan su brillo por un factor de mil millones en pocos días, volviéndose temporalmente tan luminosas como una galaxia entera. Las supernovas son cruciales para la cosmología ya que, por ejemplo, un tipo de Supernovas denominadas SN Ia se utilizan como indicadores de distancia estándar y llevaron al descubrimiento de la expansión acelerada del Universo.

Variables cataclísmicas (CV) y novas: Sistemas binarios compactos (una estrella enana blanca acumulando materia de una estrella compañera) que sufren estallidos recurrentes al alcanzar la masa crítica el material acumulado, provocando una nova. Estas erupciones aumentan dramáticamente el brillo del sistema por días o semanas. Su estudio ayuda a entender el crecimiento gradual de materia en estrellas compactas y procesos de fusión nuclear descontrolada en la superficie de enanas blancas.

Estallidos de rayos gamma ópticos: Asociados típicamente a supernovas de colapso de núcleo o a fusiones de objetos compactos (ej. estrellas de neutrones), son eventos transitorios extremadamente energéticos. Son fundamentales para estudiar el universo puesto que sirven para indicarnos y estimar como de lejos está dicho objeto de la Tierra y nos ayuda a entender a la velocidad a la que el Universo se expande.

Núcleos activos de galaxias (AGN) en modo transitorio: Aunque muchos AGN cuásares (objetos astronómicos muy luminosos y energéticos, ubicados en el centro de

galaxias distantes), son variables de forma continua, algunos presentan flashes o cambios bruscos de brillo, por ejemplo en eventos de disrupción por marea (cuando un agujero negro supermasivo destruye una estrella que pasa cerca) o en cambios de estado de acreción (proceso de crecimiento gradual de un objeto mediante la acumulación de materia atraída por la gravedad).

Blazares: Los blazares son un tipo de galaxias activas que se caracterizan por tener un agujero negro supermasivo en su centro, el cual emite un chorro de partículas de alta energía que apunta directamente hacia la Tierra. A menudo se consideran transitorios "continuos" porque su brillo fluctúa erráticamente a diferentes escalas temporales. Los blazares son laboratorios naturales de física de partículas de alta energía y campos magnéticos.

Estrellas variables eruptivas (estrellas fulgurantes): Estrellas, típicamente enanas rojas y jóvenes, que sufren erupciones repentinas en su superficie (similares a grandes llamaradas solares) aumentando su brillo en bandas ópticas que pueden durar desde minutos hasta horas. Estas fulguraciones o flares estelares tienen implicaciones en la actividad magnética estelar y pueden afectar la habitabilidad de exoplanetas cercanos.

En resumen, los eventos transitorios astronómicos **abarcan** fenómenos clave para múltiples áreas de la astrofísica. Su descubrimiento y seguimiento permiten ampliar nuestro conocimiento del universo dinámico. Desde la cosmología como por ejemplo, la medición de distancias cosmológicas con supernovas, hasta la física estelar como puede ser la evolución de binarias (dos objetos astronómicos, generalmente estrellas, que están gravitacionalmente unidos y orbitan entre sí alrededor de un centro de masa común) compactas a través de novas recurrentes. Pasando por la astronomía de altas energías, los cuales podrían ser destellos gamma, blazares, etc. Además, el estudio de objetos transitorios ha rendido numerosos avances científicos. Por ello, contar con sistemas eficientes para detectar y caracterizar estos eventos en tiempo y forma se ha vuelto prioritario en la era de los sondeos automáticos del cielo.

2.1.2 El reto del Big Data en astronomía transitoria

La llegada de sondeos sinópticos del cielo (telescopios robotizados que escanean grandes áreas del firmamento noche tras noche en busca de cambios) ha creado un nuevo paradigma de Big Data en la astronomía transitoria. En casos como el de CRTS (2007–2013), ya se generaron millones de observaciones que derivaron en miles de curvas de luz de objetos transitorios. Sin embargo, las cargas de datos actuales y futuras llevan esto a otro nivel. El Zwicky Transient Facility (ZTF), es un telescopio ubicado en el Observatorio del Palomar, operativo desde 2018, el cual cubre unos 3760 grados cuadrados del cielo cada hora y produce aproximadamente 100,000 alertas de posibles objetos transitorios cada noche. Cada "alerta" consiste en la detección de un cambio en una imagen de diferencia (imagen nueva menos imagen de referencia), indicando la aparición de una fuente nueva o una variación notable. Por otra parte, está el ejemplo de la Legacy Survey of Space and Time (LSST), un proyecto muy ambicioso que empezó en 2024 que se trata de un telescopio ubicado en el Observatorio Vera C. Rubin el cual prevé observar cada noche aproximadamente 10,000 grados cuadrados, generando más de 10 millones de alertas por noche. Esto equivale a terabytes de datos diarios solo en alertas, y petabytes de datos en imágenes crudas al año.

El reto del Big Data en este contexto es **múltiple**. Primero, está el problema de almacenamiento y manejo de datos puesto que se requieren bases de datos especializadas y eficientes para almacenar millones de curvas de luz y realizar consultas rápidas (por ejemplo, al llegar una nueva detección de LSST, recuperar en segundos el historial de esa fuente para contextualizarla). Segundo, la velocidad de procesamiento debe ser muy alta: proyectos como LSST impondrán que desde la toma de una imagen hasta la emisión de la alerta con clasificación transcurran menos de 60 segundos. Esto implica que cualquier análisis automatizado como por ejemplo extraer características y evaluar un modelo de aprendizaje automático, debe suceder en pocos segundos por evento, y en paralelo para miles o millones de eventos. Tercero, está el desafío de filtrar eficientemente los falsos positivos ya que muchas “detecciones” de objetos transitorios resultan ser artefactos

producidos por problemas en la instrumentación o el procesamiento de imágenes. De hecho, sistemas de telescopios como CRTS y ZTF han mostrado que más del 90% de las alertas iniciales pueden ser falsas alarmas, que hay que descartar mediante clasificadores. En definitiva, ningún equipo humano puede examinar manualmente decenas de miles de eventos por día, por lo que el Big Data obliga a **automatizar** tanto la detección como la clasificación de eventos transitorios. La arquitectura típica que se ha adoptado es la de "brokers" de alertas, los cuales son sistemas informáticos que reciben en tiempo real las alertas de eventos transitorios de los telescopios, las procesan, las clasifican y distribuyen a la comunidad científica la información más útil. Un "bróker" de alertas exitoso debe combinar técnicas de gestión de datos masivos, computación de alto rendimiento y algoritmos de inteligencia artificial para tomar decisiones rápidas sobre cada evento entrante.

2.1.3 Aprendizaje automático en la Ciencia de Datos

El aprendizaje automático, también denominado en inglés como machine learning o ML, es una rama de la inteligencia artificial que se enfoca en desarrollar algoritmos capaces de aprender patrones a partir de datos. Dentro del machine learning, el enfoque predominante para problemas como el nuestro es el aprendizaje supervisado, en el cual se entrena un modelo con ejemplos de entrada (datos) y salidas deseadas (etiquetas o clasificaciones) para que pueda predecir las etiquetas de nuevos datos no vistos. En este caso, disponemos de curvas de luz (datos de entrada) con sus clases verdaderas conocidas (salidas) para entrenar un clasificador. Dentro del aprendizaje automático supervisado, hay tareas de clasificación (etiquetas categóricas, como en este caso) y de regresión (salidas numéricas continuas). Este proyecto se centrará en clasificación multiclase, donde el modelo debe asignar a cada objeto una de las clases transitorias definidas.

El proceso típico de aplicar machine learning en ciencia de datos sigue varias etapas. Primero, se realiza la **preparación** y **exploración** de los datos donde se realiza una limpieza de datos faltantes o erróneos, las transformaciones necesarias y el análisis exploratorio de dichos datos para entender distribuciones, correlaciones, etc. Luego viene la ingeniería de características, que consiste en extraer o construir las variables "features"

que alimentarán al modelo de machine learning. En problemas de series temporales astronómicas, esto puede implicar calcular estadísticas resumen de la curva de luz, mediciones de variabilidad, etc. Estas características deben ser lo suficientemente discriminantes para ayudar al modelo a separar las clases. A continuación, se procede al entrenamiento del modelo en sí. Para ello, se elige un algoritmo de aprendizaje automático (o varios), se divide el conjunto de datos en subconjuntos de entrenamiento y prueba, y se ajustan los parámetros del modelo para que aprenda a predecir correctamente las etiquetas de entrenamiento minimizando el error.

Es importante en esta etapa manejar cuestiones como el **desbalance** de clases (si algunas clases son mucho más abundantes que otras, el modelo podría sesgarse a predecir siempre la mayoritaria ignorando las minoritarias) mediante técnicas de balanceo de datos o ponderación de clases. Por otra parte, se suele aplicar regularización y ajuste de hiperparámetros para evitar el sobreajuste "overfitting", es decir, que el modelo aprenda detalles irrelevantes o ruido del conjunto de entrenamiento, perdiendo así, la capacidad de generalización.

Posteriormente, tras el entrenamiento, se realiza la **evaluación** del modelo con datos separados (de prueba) para medir su rendimiento en situación simular a la real. Aquí intervienen las métricas de evaluación. En la clasificación multiclase se calcula la exactitud global (porcentaje de aciertos totales), pero como esta puede ser engañosa si las clases están desbalanceadas, se usan métricas por clase como la precisión (qué proporción de predicciones de una clase fueron correctas), el recall (qué proporción de objetos de una clase fueron correctamente detectados) y el F1 (media armónica de precisión y recall). Estas métricas pueden promediarse entre clases de forma macro (todas las clases de igual peso) o micro (ponderadas por frecuencia). También se analiza la matriz de confusión, que muestra en una tabla qué cantidad de objetos de una de las clases fueron predichos como el otro tipo de clase para todas las combinaciones, permitiendo identificar confusiones comunes entre clases.

Finalmente, un modelo entrenado y evaluado puede **desplegarse** o **aplicarse** a nuevos datos. En este proyecto se quiere aplicar el modelo a flujos de alertas en tiempo real, lo que requiere que la inferencia del modelo sea rápida y que el pipeline completo

(extracción de características “features” + predicción) esté optimizado. También es posible que se necesite re-entrenar el modelo periódicamente si se obtienen datos nuevos (por ejemplo, al migrar a otro telescopio con diferentes características).

En síntesis, el aprendizaje automático provee las herramientas necesarias para automatizar la clasificación de grandes volúmenes de datos astronómicos. Pero su éxito dependerá en gran medida de una cuidadosa preparación de los datos, de la elección apropiada de modelos y parámetros, y de una evaluación rigurosa que asegure que el modelo aprendido efectivamente captura regularidades reales y no artefactos espurios del conjunto de entrenamiento.

2.1.4 Bases de datos para eventos astronómicos transitorios

El desarrollo de algoritmos de clasificación automática requiere datos de entrenamiento extensos y fiables. Diversos sondeos han compilado catálogos de eventos transitorios junto con sus curvas de luz y clasificaciones, formando la base para entrenar modelos de aprendizaje automático supervisados.

En primer lugar, tenemos la base de datos MANTRA. Dicha base de datos proviene de Catalina Real-Time Transient Survey (CRTS)que está compuesto por tres telescopios permitiendo así una gran cantidad de datos muy útiles. Dicho telescopio fue de los más pioneros puesto que empezó a recopilar datos muy relevantes acerca de los objetos transitorios a partir del 2007 y estuvo activo hasta el 2013. Fue capaz de monitorear aproximadamente 33,000 grados^2 del cielo en busca de variables y objetos transitorios. CRTS descubrió miles de eventos (supernovas, variables cataclísmicas, blazares en erupción, flares estelares, etc.) y publicó sus curvas de luz junto con las clasificaciones obtenidas por el seguimiento multi-longitud de onda. Como se ha comentado anteriormente, la base de datos MANTRA utilizada en este trabajo está derivada del catálogo del conjunto de telescopios de CRTS, compilando 5,539 objetos transitorios con sus curvas de luz y etiquetas (aunque muchas etiquetas originales eran inciertas o muy específicas, por lo que se simplificaron a clases generales). CRTS supuso una primera demostración del poder de sondear el cielo en búsqueda de objetos transitorios de forma automática.

En segundo lugar, está el Zwicky Transient Facility (ZTF). Se trata de un ondeo moderno el cual está activo desde 2018 que genera aproximadamente 100.000 alertas por noche en bandas fotométricas. ZTF provee alertas públicas con fotometría de hasta 30 días previos para cada evento. Gracias a este telescopio y a estos datos, proyectos como el bróker (sistema informático que recibe en tiempo real las alertas de eventos transitorios de los telescopios) chileno ALeRCE han construido curvas de luz completas a partir de la concatenación de dichas alertas. Mediante cruce con catálogos espectroscópicos, se han creado conjuntos de datos etiquetados de ZTF para entrenar clasificadores; por ejemplo, Sánchez-Sáez et al. (2021) compilaron miles de curvas de luz de ZTF con clases conocidas para entrenar modelos de clasificación automática. ZTF es actualmente un banco de pruebas principal para brokers (sistemas informáticos que reciben en tiempo real las alertas de eventos transitorios de los telescopios) de alertas que aplican el aprendizaje automático en producción, dado su gran volumen de eventos y la diversidad de fenómenos descubiertos.

En tercer lugar, está Gaia. La misión espacial Gaia está operativa desde 2014 y es capaz de detectar eventos transitorios, principalmente supernovas. Aunque el volumen de alertas de Gaia es mucho menor (centenares de eventos al año), sus datos aportan clases particulares como pueden ser novas inusuales y una astrometría extremadamente precisa. Los datos de Gaia son **públicos** y podrían usarse para complementar el entrenamiento, aunque la mayoría de sus eventos transitorios también aparecen en otros sondeos.

En el caso del trabajo de fin de grado, este se enfocará en la base de CRTS (MANTRA) para entrenar y validar los modelos, dado que proporciona un conjunto homogéneo de curvas de luz con clases ya confirmadas. No obstante, la metodología desarrollada es **aplicable** a otras fuentes de datos como son ZTF, Gaia o LSST, ajustando las características extraídas y reentrenando según la naturaleza de cada encuesta. Esto garantiza que el enfoque sea generalizable más allá de la base de datos particularmente utilizada en este proyecto

2.1.5 Aplicación del aprendizaje automático a la detección de objetos transitorios

Diversos trabajos han aplicado técnicas de aprendizaje automático para abordar la clasificación de objetos transitorios en los últimos años, obteniendo resultados prometedores. En la era de CRTS, el proyecto de Djorgovski et al. (2012) fueron capaces de demostrar uno de los primeros casos de clasificación automática de eventos transitorios puesto que entrenaron una red neuronal artificial y un SVM (Support Vector Machine, algoritmo supervisado de aprendizaje automático) para distinguir objetos astrofísicos reales de detecciones espurias (falsas alarmas) en CRTS. Este fue un paso inicial enfocado en el problema "realbogus", es decir, filtrar automáticamente los falsos positivos de un sondeo transitorio. Con la llegada de sondeos más grandes, la comunidad se centró en clasificar la naturaleza astrofísica de eventos transitorios reales.

Por otra parte, está el caso de Richards et al. (2011), quienes aplicaron métodos de machine learning (incluyendo Random Forest) para clasificar estrellas variables con datos del proyecto MACHO, enfrentando desafíos similares a las curvas de luz de objetos transitorios (datos irregulares, ruidosos y escasos). Si bien se trataba de estrellas periódicas, su trabajo sentó bases en cuanto a la extracción de características "features" robustas y estrategias de validación que han sido reutilizadas en contextos de detección de objetos transitorios no periódicos.

En la actualidad se han ido creando distintos algoritmos de aprendizaje automático que han sido muy útiles especialmente para la detección de objetos transitorios dentro del ámbito de la astronomía, pero, incluso estos y futuros proyectos, tendrán una serie de **limitaciones** y **desafíos** que deberán abordar si quieren que sus algoritmos sean totalmente útiles y que extraigan conclusiones correctas.

2.1.6 Limitaciones y desafíos actuales

A pesar de los progresos, existen varias limitaciones y desafíos en la clasificación automática de objetos transitorios.

En primer lugar, está el **desbalance** de clases. Como ya se mencionó anteriormente, los

catálogos reales de objetos transitorios suelen estar dominados por unas pocas clases abundantes como pueden ser las supernovas, mientras que otras clases de interés, las cuales suelen ser eventos raros como pueden los blazares o las kilonovas, son extremadamente escasas. Un modelo entrenado directamente en datos desbalanceados tenderá a ignorar las clases minoritarias y, por lo tanto, mitigar esto es crucial.

Hoy en día, técnicas como "oversampling" (aumentar artificialmente las muestras de clases raras), "undersampling" (reducir las de la mayoritaria) o ajuste de pesos son obligatorias. En el concreto de este proyecto de fin de grado se aplicará "oversampling" de las clases minoritarias para evitar sesgos fuertes, añadiendo ligeras perturbaciones aleatorias a curvas de luz reales de esas clases para generar muestras adicionales.

En segundo lugar, están los **datos ruidosos** e **incompletos**. Las curvas de luz de sondeos como CRTS o ZTF suelen ser irregulares (puntos observados en tiempos no uniformemente espaciados), con incertidumbres considerables y a veces mediciones faltantes o límites de no detección. Además, muchas curvas están incompletas, por ejemplo, una supernova descubierta cerca de su máximo puede no tener datos del ascenso inicial. Esto dificulta la extracción de características robustas. Se debe tener precaución en la interpretación de características como la magnitud media ya que puede diferir entre clases en parte por sesgos de distancia o la duración del evento que está limitada por la ventana observacional y la distancia a la que se encuentra la Tierra del evento transitorio.

En tercer lugar, se puede encontrar la **confusión** entre clases astrofísicas. Algunas clases de eventos transitorios presentan curvas de luz muy similares, lo que lleva a confusiones incluso para algoritmos avanzados. Por ejemplo, como se vio anteriormente, un blazar puede tener un gran estallido que imite el comportamiento de una supernova rápida, haciendo que el clasificador las confunda. De hecho, en resultados previos vistos en otros proyectos, los blazares suelen ser las clases con peor F1, porque su fotometría puede parecerse a la de otras explosiones. Abordar esto podría requerir características adicionales como podría ser, incluir información del color para diferenciar un blazar típicamente más azul cuando está en su punto más brillante de una supernova, que suele ser más roja. En el caso de este TFG, se limitará a ver las características intrínsecas de la curva de luz de banda única, por lo que anticipamos que estas confusiones aparecerán y

habrá que analizarlas en resultados.

En cuarto lugar, está la **optimización** de hiperparámetros y elección de modelo. Existe una gran variedad de algoritmos de aprendizaje automático, cada uno con hiperparámetros que ajustar. Encontrar la combinación óptima puede ser costoso. Además, un modelo que funciona muy bien en una base de datos podría no generalizar a otra si las distribuciones difieren ligeramente. A menudo se emplean enfoques de "ensembles" que se trata de combinaciones de modelos para robustecer las predicciones. En este proyecto, probamos cuatro algoritmos conocidos con configuraciones estándar. Aun así, se contemplará en la discusión del proyecto la posibilidad de combinar modelos o de refinar hiperparámetros para mejorar el desempeño.

En quinto y último lugar, se contempla la **integración** en flujos en tiempo real. Un modelo que funciona bien en laboratorio debe además poder integrarse eficientemente en la infraestructura de alertas en vivo. Esto conlleva desafíos de ingeniería puesto que, el cómputo debe estar optimizado (idealmente con inferencia en milisegundos por evento), el sistema debe ser robusto a entradas inesperadas o ruidosas, y debe escalar horizontalmente (ejecutarse en paralelo en múltiples núcleos o máquinas) para poder ejecutarse sin problemas en el caso de que una o varias máquinas fallaran. Al final del TFG se abordarán estas cuestiones para estimar la viabilidad real en relación con este proyecto.

En conclusión, aunque la clasificación automática de objetos transitorios ha avanzado mucho, siguen presentes **retos** significativos. Este proyecto tomará en cuenta estas limitaciones y las aplicará en su justa medida ya que, los hallazgos de este trabajo podrán orientar mejoras futuras, como pueden ser, la adición de datos de múltiples filtros, o el uso de algoritmos más complejos para superar estas dificultades de cara a los enormes volúmenes de datos que traerán sondeos y telescopios como puede ser LSST.

3.0 Propuesta

3.1 Planteamiento de la solución

Ante los desafíos expuestos y como bien se ha ido explicando a lo largo de todo el proyecto, se propone diseñar un sistema automatizado de clasificación de eventos transitorios centrado en un conjunto de datos real manejable, la base MANTRA derivada de CRTS, y extrapolable a encuestas de mayores volúmenes. El planteamiento general de la solución sigue el paradigma típico de un proyecto de Ciencia de datos puesto que primero se realizará una preparación minuciosa de los datos de entrada (curvas de luz de eventos transitorios), luego se procederá a la extracción de características discriminantes de esas curvas. Posteriormente, se entrenarán varios modelos de aprendizaje automático supervisado sobre esas características, y finalmente se evaluará comparativamente su desempeño para elegir la mejor opción.

En particular, se ha decidido trabajar con datos reales históricos, en lugar de simulados para asegurar que el sistema aprenda a lidiar con la complejidad y ruido propios de observaciones astronómicas. La base de datos de MANTRA ofrece un buen punto de partida ya que es de tamaño moderado (miles de objetos) pero con suficiente diversidad de clases y propiedades para entrenar modelos útiles. Sobre esta base, la solución constará de los siguientes pasos principales.

Preprocesamiento y limpieza de datos: Se integrarán las curvas de luz con sus clasificaciones, se filtrarán objetos con datos insuficientes (que no permitan computar características fiables) y se unificarán las etiquetas en un conjunto reducido de clases principales que, en este caso, habiendo realizado un estudio previo, constarán de 6 clases, las cuales son elegidas por su abundancia y relevancia científica. También se homogenizarán formatos como pueden ser los identificadores y se asegurará la calidad básica de los datos, removiendo duplicados o etiquetas desconocidas, entre otros casos.

Extracción de "features" (características) de las curvas de luz: A cada evento transitorio se le calcularán una serie de atributos numéricos basados en su curva de luz. Estos atributos resumirán la información más importante de la serie temporal de magnitudes. Incluiremos estadísticas simples (media, mediana, desviación estándar de la

magnitud), medidas de amplitud (diferencia entre brillo máximo y mínimo), información de tiempos (duración total de la observación) y conteo de detecciones, entre otras.
El resultado de este paso será una matriz de características donde cada fila representa un objeto transitorio y cada columna es un "feature" calculado a partir de su curva de luz.

Entrenamiento de modelos supervisados: Utilizando la matriz de características construida y las etiquetas de clase de cada objeto, se entrenarán varios algoritmos de clasificación. En concreto, se han seleccionado cuatro algoritmos complementarios, que se constituyen por, un Random Forest (bosque aleatorio de árboles de decisión), un SVM (Support Vector Machine), una red neuronal MLP (Multilayer Perceptron), y un modelo de XGBoost (ensemble de árboles con "boosting" (método que reduce el sesgo y la varianza entre los datos, creando un equilibrio entre estos)). Estos modelos serán entrenados bajo condiciones homogéneas (misma división de entrenamiento/prueba) para permitir una comparación justa. Durante este entrenamiento, se aplicará "oversampling" de las clases minoritarias en el conjunto de entrenamiento para proporcionar un equilibrio de clases, generando instancias sintéticas (añadiendo pequeñas perturbaciones aleatorias a curvas reales) para balancear mejor los datos y evitar que el modelo ignore las clases menos frecuentes.

Evaluación y selección: Tras entrenar todos los modelos, se evaluarán en un subconjunto de datos reservado para prueba el cual no será visto durante el entrenamiento para simular los casos reales y ver si realmente ese modelo se puede escalar a otras bases de datos y otros casos. Se calcularán métricas globales (exactitud, F1 macro, etc.) y por clase (F1 por clase, matrices de confusión) para determinar cuál modelo ofrece el mejor rendimiento. Es esperable que varios modelos rindan similar por lo que se ha podido observar en otros proyectos similares. Pero, la elección final considerará no solo la ligera diferencia en métrica sino también la interpretabilidad y eficiencia. Además, se puede anticipar, que, basados en la literatura y en las características del problema, es muy probable que el Random Forest podría emerger como el modelo más equilibrado, dado su buen desempeño general y la ventaja de proporcionar importancias de variables fáciles de interpretar. En todo caso, el proceso que se va a llevar a cabo servirá para confirmar estas hipótesis y seleccionar el modelo o modelos recomendados.

Aplicabilidad a otros entornos reales: Aunque el entrenamiento y evaluación se realizará con una base de datos real, esta es estática, por ende, como paso final se analizará cómo el sistema se podría adaptar a un flujo de datos en tiempo real. Además, se discutirá cómo integrar el modelo seleccionado en un "bróker" de alertas (sistema informático que reciben en tiempo real las alertas de eventos transitorios de los telescopios) para comprobar qué tan rápido puede procesar nuevos eventos, cómo habría que modificarlo para encajar con datos de otros telescopios y sondeos, por ejemplo, incorporando color para ZTF/LSST, y qué ajustes de calibración serían necesarios, como podría ser, reentrenar con datos de cada telescopio debido a diferencias de profundidad o cadencia. Este análisis cualitativo permitirá extrapolar el valor práctico del sistema desarrollado más allá del entorno de prueba controlado.

En resumen, la solución propuesta abarca desde la **preparación** de datos hasta la **implementación** de varios algoritmos y su comparación, con un enfoque especial en la validación experimental para respaldar la selección del modelo final. Se aprovecharán técnicas robustas propias de la Ciencia de Datos como pueden ser el balanceo de datos o la normalización de variables para lidiar con las dificultades identificadas, y se documentarán todos los hallazgos y decisiones a lo largo del proceso.

3.2 Base de datos real: MANTRA

Como se ha puesto en contexto anteriormente, el dataset de MANTRA (Machine-learning Analysis of Transients with Reference lightcurves for Astronomy) es la base de datos de objetos transitorios derivada de CRTS (Catalina Real-Time Transient Survey) que se empleará en este trabajo, la cual se podrá consultar en el repositorio de GitHub enlazado junto a este proyecto, para observar todo el procesamiento realizado, además de la base de datos en crudo, para poder ver las diferencias en ella que se han ido realizando a lo largo del TFG.

En esta base de datos se dispone de un catálogo muy **amplio** de objetos transitorios, los cuales están compuestos por 5,539 objetos detectados por el conjunto de telescopios de CRTS entre 2007–2013, cada uno con un identificador único y una clasificación

astrofísica asignada. Estas clasificaciones originales incluyen más de 100 categorías distintas, muchas de ellas muy específicas o con cierta incertidumbre como por ejemplo algunas etiquetas como “SN?” o “CV?” para objetos no confirmados, o denominaciones inusuales.

Para fines prácticos, se consolidará y filtrará estas etiquetas enfocándose única y exclusivamente en **6 clases** principales de interés, tal como se mencionó en la sección 2.1.1, estas clases serán: Supernova (SN), Variable Cataclísmica (CV), AGN, Blazar, Flare estelar y Objeto HPM (alto movimiento propio). Estas seis clases representan la mayoría de los eventos bien caracterizados en CRTS. Cualquier objeto con etiqueta original fuera de estas clases se descartará o se reclasificará en la categoría más cercana de las anteriores. Por ejemplo, un evento que esté marcado como “Nova” se incluirá dentro de CV.

Por otra parte, una característica muy importante de CRTS es que no tenía filtro de color, lo que simplifica nuestros datos ya que todas las magnitudes son comparables en la misma banda, pero a la vez surge un problema y es que significa que no tenemos información de color para distinguir, por ejemplo, un blazar (que típicamente es más azul en sus estados brillantes) de una supernova (que puede ser más roja en sus estados más brillantes). Esto habrá que tenerlo en cuenta al interpretar resultados ya que podría dar lugar a ciertas confusiones.

Pero primero, antes de proceder al uso de la base de datos conveniente realizar ciertas operaciones de **limpieza** y **filtrado** iniciales antes incluso del procesamiento de los datos posterior, para asegurarse de que los datos que se usarán en el proyecto son fiables y correctos.

En primer lugar, se realizó un filtrado por número de observaciones. Donde se decidió excluir del análisis aquellos objetos transitorios con curvas de luz sean demasiado escasas. Para ello se requirió que cada objeto tuviera al menos 5 detecciones fotométricas registradas ya que con menos de 5 puntos es prácticamente imposible calcular la mayoría de nuestras características de variabilidad de forma fiable como pueden ser la desviación estándar o la asimetría ya que no son significativas con tan pocos datos.

Tras este primer filtrado, quedaron **3,330 objetos** en total, todos pertenecientes a las 6

clases principales definidas. Las reducciones más notorias fueron en la clase HPM, los cuales son principalmente asteroides, y estos bajaron de 640 bajó a 306 objetos y en el caso de las supernovas (SN) bajaron de 1723 a 1293 objetos, evidenciando que muchos HPM reales solo se vieron un par de veces y muchas SN fueron poco muestreadas en CRTS. Pero en el caso de las clases Blazar, AGN, CV y Flare mantuvieron la mayoría de sus objetos porque suelen tener observaciones más extensas.

Una vez se realizó dicho primer filtrado, se procedió a realizar una **homogeneización** de identificadores y etiquetas. En este proceso se verificó la consistencia de los IDs entre la tabla de curvas de luz y la de etiquetas, las cuales son las dos fuentes de datos que se usarán a lo largo de todo este proyecto, como bien se ha comentado antes, provenientes de MANTRA. En los datos originales, los identificadores venían con prefijos tipo "CSS" o "MLS" según el telescopio de CRTS ya que, como se ha comentado antes, el proyecto Catalina Real-Time Transient Survey constaba de 3 telescopios, mientras que en la tabla había muchos datos que directamente aparecían sin prefijo. Por lo tanto, se procedió a unificar los IDs (manteniendo solo el componente numérico o agregando el prefijo faltante según el caso) para asegurar que cada curva se asociara correctamente a su clase y por ende las dos tablas se pudieran unificar sin ningún problema. Asimismo, se eliminaron objetos con identificadores duplicados o inconsistentes, pero fueron muy pocos casos aislados.

Por último, se trataron los datos faltantes. En general, las curvas de CRTS no tienen datos faltantes explícitos ya que cada detección es un punto válido, sino que son inherentemente irregulares en el tiempo, por lo tanto, conviene rellenar dichos datos en el espacio de ese tiempo de forma coherente.

Tras realizar una limpieza **exhaustiva**, se obtuvo una distribución final de clases en el conjunto de 3,330 objetos transitorios totalmente válidos.

Clase	Descripción	Nº
Supernova (SN)	Explosión estelar catastrófica. Aumento de brillo rápido seguido de desvanecimiento en semanas.	1293 (38.8%)
Variable Cataclísmica (CV)	Estallidos recurrentes en binarias (estrellas que están unidas gravitacionalmente y orbitan alrededor de un centro de masa común) con enanas blancas (novas enanas principalmente). Múltiples picos de brillo cuando se desencadena fusión en la superficie de la enana blanca.	862 (25.9%)
AGN (núcleo galáctico activo)	Variaciones aleatorias de brillo en el núcleo de una galaxia activa como, por ejemplo, cuásares, eventos de disrupción por marea, etc. Cambios de magnitud moderados a escala de meses/años.	425 (12.8%)
Blazar	Tipo extremo de AGN con chorro relativista hacia la Tierra. Variabilidad rápida e intensa en múltiples frecuencias.	237 (7.1%)
Flare estelar	Estrella variable eruptiva que presenta explosiones bruscas y poderosas. Incremento repentino de brillo por minutos u horas seguido de retorno al estado de reposo (estado quiescente).	207 (6.2%)
Objeto HPM	Objeto de alto movimiento propio detectado como transitorio, los cuales se identifican principalmente como asteroides moviéndose entre imágenes captadas por los telescopios. Aparece unas pocas veces en la misma posición desplazándose rápidamente, sin variación intrínseca de brillo significativa.	306 (9.2%)

Tabla 1. Clases de objetos transitorios en la base de datos de MANTRA (CRTS) tras filtrado inicial. Se listan las seis clases principales consideradas, con una breve descripción y el número de objetos de cada clase en el conjunto de datos final (3,330 objetos en total). Entre paréntesis se indica el porcentaje que representa cada clase sobre el total.

En resumen, MANTRA nos proporciona una base de datos realista y desafiante puesto que presenta seis clases de objetos transitorios con distribución desbalanceada y curvas de luz ruidosas en una sola banda. Su gran ventaja es que las etiquetas provienen de confirmaciones externa así que todos los datos procesados, son totalmente fiables y verídicos.

Esta base de datos servirá para **entrenar** y **probar** los modelos de aprendizaje automático propuestos bajo condiciones cercanas a las de un "bróker" real, el cual vería muchas SN y CV, y menos blazares, flares, etc., con datos de calidad variable dependiendo de la cantidad y la calidad de ellos.

3.3 Procesamiento de los datos

Con el entendimiento previo de la base de datos, se procede con la preparación formal de los datos para alimentar los modelos de machine learning.

Las etapas de procesamiento realizadas en este proyecto fueron en un orden totalmente lógico y claro para obtener los datos que mejor se ajusten a este proyecto y poder entrenar los modelos sin ningún tipo de problema ni error.

En primer lugar, se realizó una **fusión** de datos de curvas de luz con etiquetas. A partir de la tabla de curvas de luz (varias observaciones por objeto) y la tabla de objetos con su clase, se unificaron en un único conjunto agregando las observaciones de cada objeto. En la práctica, se procedió leer el archivo de observaciones el cual contenía las columnas ID del objeto transitorio, el tiempo, la magnitud, el error de magnitud, etc. Mientras que el archivo de etiquetas contenía el ID del objeto transitorio y la clasificación original. Tras la limpieza de IDs descrita en el apartado 3.2, se realizó un "merge" (unión) a partir del identificador (ID) del objeto transitorio, asignando a cada observación de la clasificación correspondiente. Luego, estas observaciones se agruparon por objeto para el cálculo de las características pertinentes.

En segundo lugar, se procedió a hacer un **cálculo** de las "features" (características) agregadas. Se definieron un conjunto de funciones de agregación para extraer atributos de cada grupo de observaciones pertenecientes a un mismo objeto transitorio. En

concreto, las características calculadas para cada objeto fueron:

Características	Descripción
mag_mean	Magnitud promedio (media) de la curva de luz del objeto.
mag_median	Magnitud mediana de la curva de luz (valor central).
mag_std	Desviación estándar de las magnitudes (mide la dispersión total del brillo).
mag_max	Magnitud máxima (valor numérico más bajo = objeto en su estado más brillante observado).
mag_min	Magnitud mínima (valor numérico más alto = objeto más débil detectado).
magerr_mean	Incertidumbre promedio de las mediciones de magnitud.
magerr_std	Desviación estándar de las incertidumbres de magnitud (indica la variabilidad en la calidad de las detecciones).
n_obs (*count*)	Número de observaciones totales de la curva de luz.
duration	Duración temporal de la curva (última observación menos primera observación).
amplitude	Amplitud de variación: diferencia entre la magnitud más débil y la más brillante (mag_min y mag_max).
skewness	Coeficiente de asimetría de las magnitudes observadas (positivo si hay pocos puntos brillantes y muchos débiles, negativo si al revés).

Tabla 2. Características ("features") extraídas de las curvas de luz y su descripción. Estas 11 variables resumen cuantitativamente cada curva de luz de un objeto transitorio y constituyen la entrada pertinente para los modelos de aprendizaje automático.

En tercer lugar, una vez están agregadas las observaciones por objeto aplicando las funciones anteriores, se obtuvo una **tabla** donde cada fila corresponde a un objeto transitorio y las columnas son las características mencionadas. Posteriormente, a esta tabla se le añadió la etiqueta de clase final para cada objeto en una columna adicional

llamada, "Class".

En cuarto lugar, se realizó una **verificación** y **limpieza** final de las características ("features"). Para ello, se revisó si había valores faltantes o extremos en las features. Pero gracias al filtrado que se hizo previoamente, ningún objeto quedó con features indefinidas. No obstante, se detectaron algunos casos particulares, ya que por ejemplo algunos objetos transitorios de tipo HPM tenían una "duration" muy corta (0 días si fueron detectados el mismo día un par de veces), o una "skewness" indefinida para curvas con todos los puntos iguales. Por lo tanto, en esos casos, se procedió a ajustar dichos sucesos. En el caso de la característica "duration", se estableció una duración mínima de 0 si solo posee un día de observación. Por otro lado, en cuanto a la característica "skewness", se definió como 0 si la desviación estándar era 0, es decir, si todas las magnitudes fueron iguales, lo que en práctica solo ocurre en HPM. Estas conversiones aseguraron que ninguna característica tuviera un valor nulo o NaN.

En quinto lugar, se procedió con la **división** del conjunto de datos, donde se deja una parte de entrenamiento y una de prueba. Con la base de datos final de características y etiquetas, se procedió a separar los datos donde se realizó una división estratificada 80/20, es decir, se reservó aleatoriamente un 20% de los objetos como conjunto de prueba, manteniendo por supuesto la proporción de clases y el 80% restante se usó para entrenamiento, además de poder usarse para validación interna si fuera necesario. Esta división nos permite entrenar los modelos de machine learning sin usar la porción de prueba, de manera que luego podamos evaluarlos objetivamente en esos datos nunca vistos durante el entrenamiento y poder escalarlo a otras bases de datos, ajustando los modelos y las clases de manera acorde.

En definitiva, se cuenta con una **matriz** de tamaño aproximado de 2664×11 de "features" escaladas para entrenamiento, con sus etiquetas correspondientes, y un conjunto independiente de prueba de 666 objetos con la misma estructura.

3.4 Modelos de aprendizaje automático usados

Para la realización de este trabajo de fin de grado, se seleccionaron cuatro modelos de

clasificación supervisada, los cuales son representativos de diferentes familias de algoritmos ampliamente utilizadas en problemas similares. A continuación, se describirá brevemente cada modelo y por qué fue incluido para la realización de este proyecto.

Bosque Aleatorio (Random Forest): Es un método de ensamble basado en árboles de decisión el cual consiste en entrenar muchos árboles de decisión independientes, cada uno sobre muestras aleatorias del conjunto de entrenamiento denominado como técnica de "bagging" y usando un subconjunto aleatorio de "features" en cada bifurcación del árbol. La predicción final es la votación mayoritaria de todos los árboles. Los Random Forest tienden a ser robustos, manejar bien datos ruidosos y capturar idóneamente las relaciones no lineales. En el caso de este trabajo de fin de grado, se espera que Random Forest funcione perfectamente, e incluso se podría asegurar que será uno de los mejores modelos de aprendizaje automático más idóneo para este proyecto dado que en estudios previos similares a este, Random Forest destacó para datos de curvas de luz.

Máquina de Soporte Vectorial (SVM) con kernel RBF: Es un modelo lineal extendido mediante kernels (técnica matemática utilizada para transformar datos no lineales en un espacio de características de mayor dimensión, donde pueden ser tratados de manera lineal) para capturar relaciones no lineales. La SVM busca un hiperplano en un espacio de alta dimensión que separe las clases maximizando el margen (distancia) entre los puntos de distintas clases más cercanos al borde. Usamos un kernel denominado como (RBF) que permite modelar fronteras no lineales flexibles. Las SVM suelen ser efectivas en casos de dimensiones moderadas y son robustas al sobreajuste gracias a su principio de margen máximo, pero pueden volverse lentas o difíciles de escalar con conjuntos muy grandes. En el caso de esta base de datos con sus 2664 objetos y sus 11 "features", una SVM es factible.

Red Neuronal MLP: Se implementó una red neuronal artificial de tipo "feed-forward", la cual es un tipo en la que la información va en una sola dirección, desde la entrada hasta la salida, sin bucles ni conexiones hacia atrás. Las redes neuronales tienen la ventaja de poder aproximar funciones complejas si se entrenan correctamente, aunque necesitan suficientes datos y un buen ajuste de hiperparámetros para poder sacar todo su potencial. En el caso de este proyecto de fin de grado, la MLP se incluyó para evaluar si un enfoque

conexionista básico puede igualar a los métodos más tradicionales.

XGBoost (Extreme Gradient Boosting): Es un algoritmo de "boosting" (algoritmo de aprendizaje automático que reduce el sesgo y varianza en un contexto de aprendizaje supervisado) de árboles muy eficiente y potente, que ha ido ganando popularidad en el transcurso del tiempo. XGBoost entrena secuencialmente muchos árboles de decisión pequeños, donde cada nuevo árbol corrige errores de los anteriores, optimizando una función de pérdida mediante gradiente descendente. Este modelo dispone de capacidades de regularización avanzadas y suele lograr un desempeño ligeramente superior a Random Forest en muchos conjuntos de datos, pero también al costo de más parametrización. En este caso, se incluyó XGBoost en este trabajo para representar los métodos de "boosting" y evaluar si aporta alguna mejora sobre el Random Forest. En el caso de este modelo, se configuró con parámetros estándar ya que estaba compuesto de100 árboles con una profundidad máxima de 6 y una tasa de aprendizaje del 0.1. En cuanto a las expectativas de este modelo, se espera que XGBoost obtenga un rendimiento similar a Random Forest e incluso quizás, marginalmente **mejor** en métricas globales, pero es necesario confirmar si vale la pena la complejidad añadida. Además, es interesante verificar el tiempo de entrenamiento y predicción, puesto que este modelo es conocido por ser rápido en inferencia, lo cual es relevante para después poder escalarse a otras bases de datos.

A continuación, se pueden observar las **configuraciones** concretas usadas en cada modelo entrenado. Todas basadas en la implementación de las librerías "scikit-learn" y "XGBoost library":

Random Forest: RandomForestClassifier(n_estimators=100, random_state=42).

Es decir, 100 árboles de decisión con la semilla fija para su reproducibilidad y el resto de los parámetros se dejaron por defecto como pueden ser la máxima profundidad indefinida, la selección de sqrt(n_features) en cada división, etc.

SVM (RBF): SVC (C=1.0, kernel='rbf', gamma='scale', probability=True). Usamos gamma='scale' la cual equivale a 1/(n_features*var) en scikit, similar a 0.1 en el caso de este proyecto. Además, se puso el parámetro probability=True para poder obtener probabilidades ya que necesario para poder dibujar curvas ROC.

MLP: MLPClassifier(hidden_layer_sizes= (100,), activation='relu', solver='adam', max_iter=500, random_state=42). Este modelo está creado con una sola capa oculta de 100 neuronas, la función de activación ReLU (que deja pasar los valores positivos y bloquea los negativos) y el optimizador Adam (ajusta los pesos de forma eficiente adaptando automáticamente la velocidad de aprendizaje) y se permitió un máximo de 500 épocas.

XGBoost: XGBClassifier(n_estimators=100, max_depth=6, learning_rate=0.1, use_label_encoder=False, eval_metric='mlogloss', random_state=42). Este modelo está entrenando 100 árboles de profundidad hasta 6, con tasa 0.1. Por otra parte, use_label_encoder=False que sirve para evitar alertas en la versión usada.

Todos los modelos se entrenaron utilizando un conjunto de entrenamiento **balanceado**. En esta fase, se monitoreó que el proceso de entrenamiento transcurriera sin incidentes. Un aspecto observado muy interesante fue que el modelo MLP inicialmente dio un rendimiento bajo (aproximadamente un 45% de precisión) hasta que se dio con el problema de porque tenía un rendimiento tan bajo y resultó ser que requería de una normalización de inputs y tras añadir ese paso, la MLP subió a un 70% de precisión de entrenamiento, similar al resto de modelos, lo cual es un indicativo de que estaba aprendiendo razonablemente.

En resumen, los cuatro modelos representan enfoques diversos: árboles en paralelo (Random Forest), árboles en secuencia (XGBoost), un clasificador de margen (SVM) y una red neuronal. Con esta selección amplia, se cubren una variedad de supuestos y capacidades de modelado, lo que permitirá ver cuál se adapta mejor a la base datos de MANTRA y los objetos transitorios.

3.5 Elección del mejor modelo

En este último paso de la etapa de la propuesta, se establecerá cómo se definirá cuál modelo (o modelos) es/ son el más idóneo para el devenir de este trabajo de fin de grado. Dado que se están evaluando múltiples algoritmos, la comparación para elegir cual o

cuales son los modelos idóneos se basará en unas pautas establecidas rigurosamente.

Por una parte, se establecerán unas métricas cuantitativas en el conjunto de prueba. Estas serán principalmente la exactitud global (accuracy) y el F1 macro (promedio del F1 de cada clase, dando igual peso a todas). Estas dos métricas nos darán una idea del rendimiento general de cada uno de los modelos. También se tendrá en cuenta la precisión y recall promedio macro y, en detalle, el F1 de cada clase, para ver si un modelo funciona mucho mejor en ciertas clases que en otras, lo cual puede ser importante dependiendo de la aplicación, por ejemplo, al identificar blazares puede que sea más crítico en cierto contexto.

Por otro lado, se medirá la robustez y el comportamiento en clases minoritarias. Esto se debe principalmente al interés científico de no ignorar las clases raras y, por ende, un modelo que logre un F1 decente en las clases de blazares y flares sería preferible a uno que maximice solo la precisión global a costa de "sacrificar" esas clases. Por eso se le dará mayor importancia al análisis de la matriz de confusión y F1 por clase.

También se medirá la interpretabilidad y la facilidad de uso. En un entorno de producción, a menudo se valora que el modelo sea interpretable y fácil de actualizar con nuevos datos. Por lo que viene dado al Random Forest, es un modelo que puntúa alto en **interpretabilidad** debido a la importancia de las variables y es relativamente fácil de reentrenar e implementar ya que es rápido a la hora de predecir. En cuanto a la SVM, ofrece interpretabilidad menor y puede ser más costosa de recalibrar en el caso de que lleguen montones de datos nuevos. Por otro lado, la MLP es aún peor que la SVM, ya que, aunque su implementabilidad es buena con librerías modernas, puede requerir GPUs para grandes volúmenes de datos, haciendo que sea más lento a la hora de implementar y de usar. Por último, XGBoost es interpretable al nivel de Random Forest ya que también da importancias a las variables y es muy rápido, pero, a diferencia del Random Forest, este tiene más hiperparámetros que ajustar para un nuevo dominio. Así, incluso si todos tuvieran métricas cercanas, se podría deducir en mayor medida por el Random Forest o el XGBoost debido a su combinación entre el rendimiento, la interpretabilidad y la rapidez.

Por último, se valorará la velocidad y la escalabilidad, ya que, aunque en el caso de este

trabajo de fin de grado el proyecto se vaya a desarrollar con datos con aproximadamente 3000 objetos y por cualquier modelo corre rápido, la intención es pensar a futuro como puede ser la aplicación de estos modelos en LSST, el cual tiene millones de eventos. Por ende, es relevante que el modelo pueda predecir miles de instancias por segundo. Los modelos basados en árboles, como son el Random Forest y el XGBoost suelen ser muy veloces en inferencia

Para resumir, se definirá como el "mejor modelo" aquel que logre **maximizar** el F1 macro (priorizando la eficacia balanceada entre clases) manteniendo al mismo tiempo un desempeño robusto en clases de interés particular ya que no conviene un modelo que, por ejemplo, obtenga 90% accuracy a costa de nunca reconocer un blazar. Además, entre modelos de eficacia similar, se escogerá el que sea más práctico de interpretar y escalar.

4.0 Proceso de implementación

4.1 Metodología

El devenir de este TFG se realizó mediante una metodología iterativa acorde a la magnitud de este proyecto, inspirado en enfoques ágiles, con ciclos cortos de implementación y validación. Dado que el Trabajo de Fin de Grado está acotado por unos tiempos y unas entregas de seguimiento obligatorias, se optó por organizar el esfuerzo en sprints cortos con objetivos específicos haciendo uso de una herramienta que resulta muy familiar puesto que se implementó a lo largo de toda la carrera, denominada Trello, la cual es idónea para la organización de las tareas mediante el método de los sprints.

Posteriormente se incluirán ilustraciones acerca de cómo se han ido desarrollando cada una de los sprints. Cada uno abarcó todas y cada una de las etapas del proyecto para desarrollarlo de manera organizada, permitiendo que se construya el sistema paso a paso y ajustando el rumbo según los resultados intermedios que se fueran obteniendo.

En particular, se definieron 6 **sprints** semanales aproximadamente, detallados en la próxima sección denominada 4.2 (Sprint Planning). Durante cada sprint se mantuvieron notas técnicas y se documentaron hallazgos, integrando tanto desarrollo técnico (código)

como análisis científico en cada iteración. Esto ha permitido ir adaptando el plan ya que, por ejemplo, en la fase de exploración, en una primera instancia, se iban a extraer los datos de Gaia y ZTF, pero debido a la complejidad de la extracción por parte de Gaia, se decidió cambiar la fuente de datos y hacer uso de la de MANTRA. Por ello, es muy importante ir definiendo cada una de las etapas de manera rigurosa, porque es posible que surjan adversidades o complicaciones que hay que ir mitigando dentro de los tiempos establecidos para que no escale en un problema mayor.

El ecosistema de trabajo elegido para el desarrollo del código fue Python, por su flexibilidad para el desarrollo rápido y las numerosas librerías especializadas en Ciencia de Datos. En concreto, todo el código fuente principal se organizó en a partir de la siguiente estructura, el cual, como se mencionó anteriormente, se podrá consultar en el repositorio de GitHub que estará vinculado a este TFG, tanto el código, como las imágenes utilizadas como las bases de datos usadas, tanto las crudas como las procesadas. Por una parte, está el **Módulo de preprocesamiento (leer_y_procesar_datos.py):** Este código encapsula la lógica de lectura de los dos archivos CSV que se usan en este TFG, los cuales son los de de curvas de luz y el de etiquetas de MANTRA, además de la limpieza de datos como son los filtrados y unificación de formatos y el cálculo de las características descritas en el apartado 3.4. Este módulo produce los archivos finales dataset_features.csv y dataset_labels.csv que alimentan la fase de modelado. Fue diseñado de forma reproducible, de modo que, si se dispone de nuevos datos en el mismo formato, con solo correr este módulo se generarán las características actualizadas.

Por otro lado, están los **Scripts de entrenamiento para cada modelo, como por ejemplo, entrenamiento_random_forest.py o entrenamiento_svm.py:** En este caso, la mejor idea para no duplicar mucho código y no generar mucha carga y confusión en los datos, se procedió a diseñar un script general que recibe un parámetro indicando qué modelo entrenar, pero, a su vez, por simplicidad y seguimiento se mantuvieron separados. Cada script carga los datos procesados, divide los datos en entrenamiento/prueba, aplica escalado y balanceo (cuando corresponde) y entrena el modelo específico. Al finalizar,

estos scripts imprimen un reporte de métricas en la consola donde se ejecute el código y guarda los resultados más relevantes como pueden ser la matriz de confusión o el modelo entrenado serializado) en el disco, para poder disponer de dichos resultados.

Por último, como se ha comentado anteriormente se realizó un control de versiones con GitHub. Todo el código se versionó usando esta herramienta en un repositorio privado para controlar los cambios y poder revertir si era necesario. Además, esto facilitó mucho más el seguimiento del progreso y la colaboración con el tutor.
En este caso, dado que excedía el alcance y las posibilidades debido a las limitaciones de tiempo, no se implementó una interfaz gráfica ni una API web, pero el código está organizado de forma que podría integrarse en una infraestructura mayor en el futuro. Por ejemplo, el módulo de preprocesamiento puede ser adaptado para leer flujos de alertas en **tiempo** en lugar de CSVs, y los scripts de entrenamiento podrían convertirse en trabajos periódicos en un entorno de producción para reentrenar el modelo con nuevos datos.

Para ofrecer mayor claridad, en el siguiente apartado se resume la planificación por sprints que ha sido realizada en este TFG, incluyendo las tareas y los resultados clave de cada uno.

4.2 Sprint Planning

Sprint 1 (Semana 1-2).

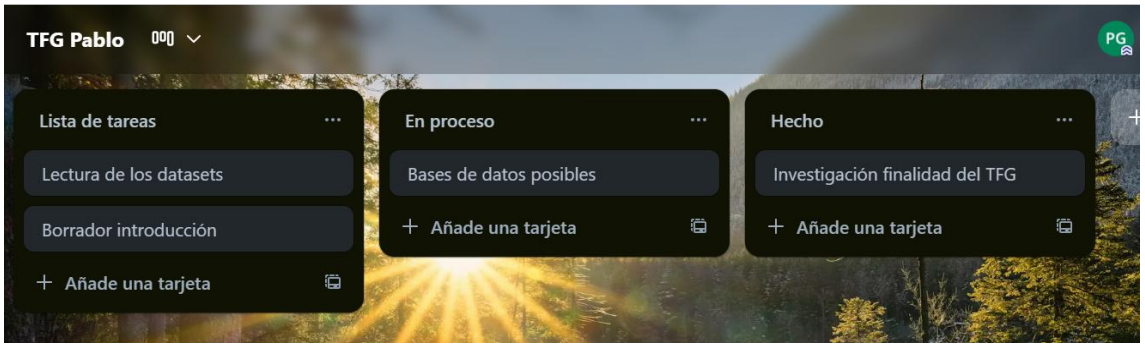

Ilustración 5. Imagen extraída de Trello. Sprint 1.

El objetivo principal del primer sprint era la comprensión del dominio, la inicial investigación acerca del posible tema del TFG y la recopilación de datos.
En primer lugar, se buscó un problema real que tuviera matices acerca de un tema relevante en la actualidad y por eso se decidió el tema de la astronomía. Una vez decidido el tema, era momento de buscar la forma de aplicar la Ciencia de Datos a la astronomía. Y cuando ya estaba decidido todo, era el momento de buscar una o varias fuentes de datos acordes a la magnitud de este proyecto.

Sprint 2 (Semana 3-4).

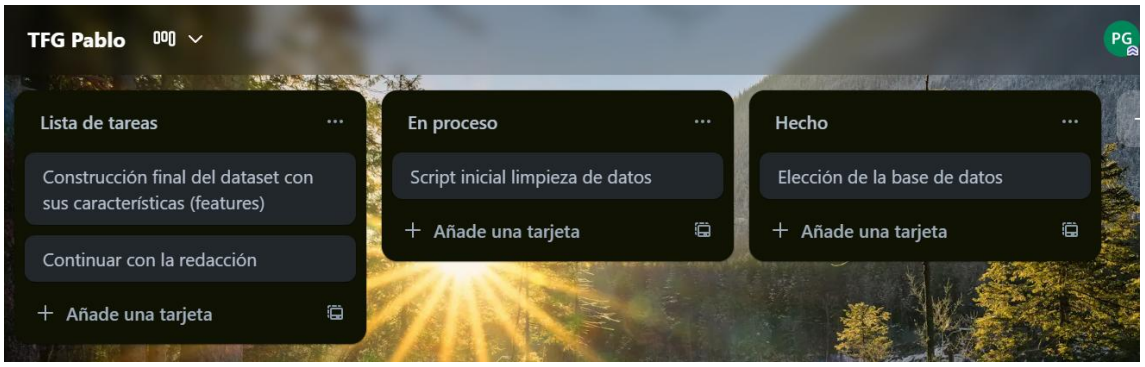

Ilustración 6. Imagen extraída de Trello. Sprint 2.

Durante este sprint se enfocó el trabajo en construir el conjunto de datos a partir de la base MANTRA, la cual era la base de datos idónea para este TFG. Se empezaron a generar las primeras características (features) útiles. Se limpiaron y procesaron los datos brutos, unificando etiquetas, filtrando por número de observaciones y preparando una base de datos listo para modelar. También se empezó a redactar la parte metodológica correspondiente a esta etapa.

Sprint 3 (Semana 5-6).

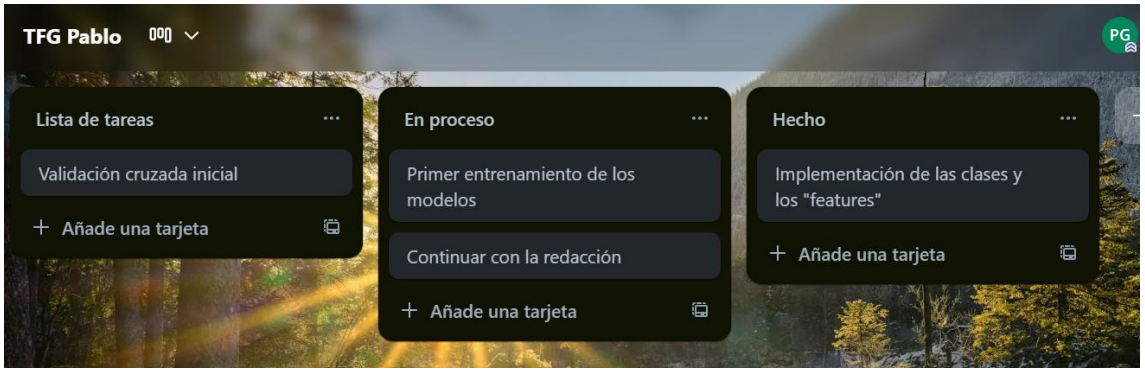

Ilustración 7. Imagen extraída de Trello. Sprint 3.

En este sprint se realizó el primer entrenamiento de los modelos. Se configuraron los conjuntos de entrenamiento y prueba, se aplicaron técnicas de balanceo como el "oversampling" y se entrenaron los cuatro modelos principales (Random Forest, SVM, MLP y XGBoost). Además, se validaron los primeros resultados y se continuó con la redacción de la memoria, entre otras muchas tareas.

Sprint 4 (Semana 7).

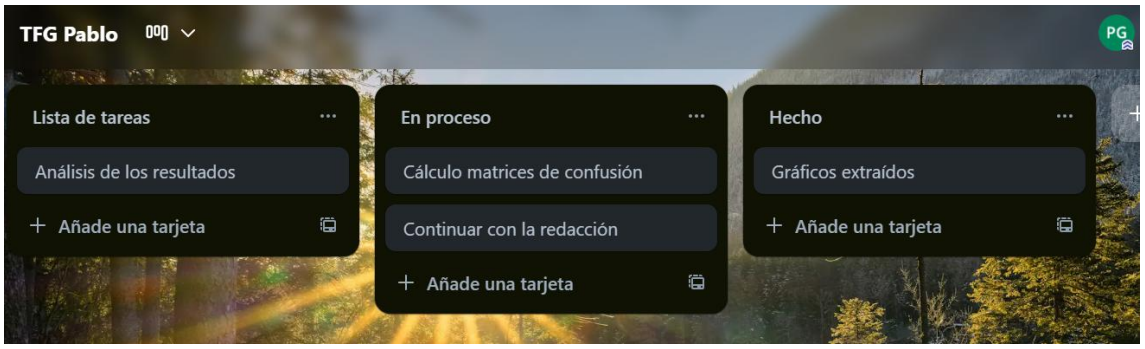

Ilustración 8. Imagen extraída de Trello. Sprint 4.

El objetivo de este sprint fue analizar el rendimiento de los modelos entrenados. Se calcularon matrices de confusión, se generaron curvas ROC y se elaboraron gráficas de importancia de variables. Estas visualizaciones permitieron entender los errores más comunes del sistema y qué características eran más relevantes. Toda esta información se integró en la sección de resultados del documento.

Sprint 5 (Semana 8).

Ilustración 9. Imagen extraída de Trello. Sprint 5.

En este sprint se abordó la viabilidad práctica del sistema desarrollado. Se midió el tiempo de predicción del modelo Random Forest con grandes volúmenes de datos simulados, se analizaron posibilidades de paralelización y se exploró cómo adaptar el sistema a flujos de alertas como la base de datos de ZTF. También se compararon los modelos para decidir cuál era el más adecuado para producción.

Sprint 6 (Semana 9).

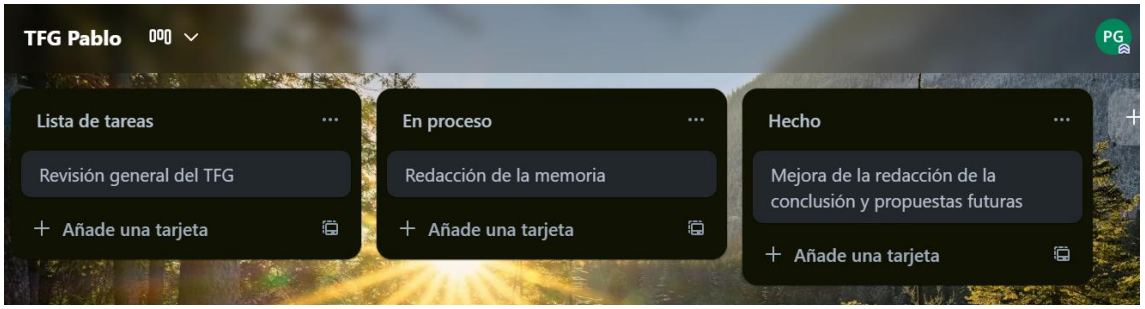

Ilustración 10. Imagen extraída de Trello. Sprint 6.

Este sprint final, se centró principalmente en la revisión general del trabajo y la redacción definitiva. Se pulieron las conclusiones, se reforzó la argumentación sobre la utilidad del sistema en contextos reales y se redactaron propuestas de mejora futuras, como la incorporación de información multibanda o contextos astrofísicos. El objetivo fue cerrar el TFG con una presentación clara, coherente y completa.

5.0 Producción, resultados y aplicaciones reales

5.1 Código y base de datos elegida

Por lo que viene dado a la producción y desarrollo de resultados del TFG, se procederá a poner en contexto todo lo explicado anteriormente de forma que toda la estructura tenga sentido y se vea paso a paso todo el proceso que se ha llevado a cabo.

Durante la implementación se optó por un ecosistema de código en Python, dada la flexibilidad que ofrece para combinar análisis de datos y modelado de forma interactiva. La base de datos elegida fue, como se ha descrito, MANTRA/CRTS, una colección real de eventos transitorios históricos. Esta elección estuvo motivada principalmente por su tamaño **manejable** ya que es adecuado para entrenar y probar modelos en un TFG sin requerir una infraestructura de Big Data, pero con la posibilidad de poder escalarse a ello debido a la exactitud de los modelos. Además de la disponibilidad pública de los datos ya que CRTS es un sondeo con datos abiertos, lo que facilita la reproducibilidad. Por último, la relevancia científica ya que CRTS detectó principalmente supernovas, CVs, flares estelares, blazares, etc., que son precisamente las clases de interés principales de de los objetos transitorios ópticos.

El código fuente se organizó de manera **modular** para facilitar su reutilización. Para ello se creó un módulo específico para la lectura y preprocesamiento de datos (leer_y_procesar_datos.py). Dicho módulo carga los archivos CSV originales, realiza todas las transformaciones necesarias (filtros, agregaciones, etc.) y guarda los conjuntos de “features” y etiquetas listos para el modelado. Es fundamental separar esta etapa ya que permite que, si en un futuro se desea aplicar el sistema a otra base de datos (por ejemplo, la de las curvas de luz de ZTF), se pueda adaptar solo este componente para leer el formato diferente, manteniendo intacta la lógica de entrenamiento de los modelos. Posteriormente se desarrollaron **scripts** de entrenamiento para cada modelo. En esencia, cada script realiza los mismos pasos: carga los datos procesados, divide en entrenamiento/prueba, aplica escalado, entrena el modelo específico y evalúa su desempeño. El hecho de contar con scripts separados permitió ajustar detalles particulares a cada algoritmo (por ejemplo, la MLP necesitó un número de épocas de entrenamiento

definido.). Sin embargo, se evitó la duplicación excesiva extrayendo configuraciones comunes a un archivo de configuración o usando funciones utilitarias. Por otra parte, se implementó código para generar salidas visuales (gráficos). Esto incluyó el uso de librerías como matplotlib y scikitlearn como, por ejemplo, ConfusionMatrixDisplay para las matrices de confusión. Muchas de estas gráficas fueron generadas durante el Sprint 4 y se guardaron como archivos PNG.

La base de datos de MANTRA y el código están estrechamente **ligados** puesto que todos los resultados cuantitativos que se discuten a continuación (matrices de confusión, métricas, etc.) provienen de la evaluación de los modelos entrenados con estos datos. Cabe enfatizar que, aunque los datos provienen de CRTS (2007-2013) y por tanto no contienen eventos transitorios descubiertos por misiones más recientes, son suficientemente representativos de la fenomenología general de objetos transitorios ópticos. Además, su uso aseguró que todo el desarrollo se hizo con datos reales (no simulados), lo cual le da mayor validez externa al sistema, ya que, por ejemplo, cualquier ruido o irregularidad presente en CRTS como pueden ser observaciones faltantes o errores fotométricos, es algo que también ocurrirá en otras encuestas, así que los modelos ya fueron expuestos a esas condiciones durante el entrenamiento.

Todo el código se alojó en un repositorio de GitHub para poder llevar un control de las versiones, lo que facilitó poder regresar a versiones previas si un cambio introducía algún problema de desempeño.

En suma, el código desarrollado y la base de datos MANTRA fueron los **pilares** técnicos sobre los que se construyó el sistema. Esta combinación proporcionó una plataforma sólida para llevar a cabo los experimentos. Además, cabe destacar que el código desarrollado podría reutilizarse fácilmente con otros datos y su modularidad permite incorporar mejoras como pueden ser nuevas “features” o nuevos modelos con relativa facilidad.

5.2 Elección del modelo o modelos idóneos

Tras entrenar y evaluar los cuatro modelos propuestos, se procedió a seleccionar el modelo más adecuado para nuestro sistema de clasificación de objetos transitorios. Como

se estableció en la sección 3.6, la decisión se basó tanto en métricas cuantitativas de desempeño como en consideraciones prácticas.

En la siguiente tabla se resumen las **métricas** globales obtenidas por cada modelo en el conjunto de prueba, el cual, recordemos que se trata de 20% de los datos totales, nunca vistos durante el entrenamiento. En dicha tabla se incluyen las métricas más importantes para saber la información necesaria acerca de cada uno de los modelos, las cuales incluyen la exactitud global y los promedios macro de precisión, recall y F1.

Modelo	Exactitud	Precisión (macro)	Recall (macro)	F1 (macro)
Random Forest	71.9%	67.0%	66.0%	**66.0%**
SVM (RBF)	71.2%	66.0%	64.0%	65.0%
XGBoost	71.6%	66.0%	65.0%	65.0%
MLP	70.3%	61.0%	63.0%	62-0%

Tabla 3. Comparación de desempeño de los modelos de clasificación en el conjunto de prueba.

En esa tabla se puede observar que todos los modelos obtienen exactitudes muy **similares**, alrededor del 70-72%. Sin embargo, el Random Forest alcanza la más alta (71.9%) y también **destaca** ligeramente en el F1 macro (0.66) frente a SVM y XGBoost (ambos 0.65). Por otra parte, a red neuronal MLP queda algo rezagada con un 70.3% de exactitud y un F1 macro de 0.62, indicando que su rendimiento global fue un poco inferior.

Las diferencias numéricas son pequeñas lo que sugiere que ningún modelo supera abrumadoramente a los demás en desempeño global. Sin embargo, siguiendo los criterios establecidos, el Random Forest mostró la métrica ligeramente superior, por lo que sería candidato principal.

Pero, lo más importante, es examinar el **desempeño** por clase, para asegurarnos de que la

elección beneficie al sistema de manera equilibrada. Para ello, hay un gráfico que va muy acorde a este desempeño y es la matriz de confusión, la cual nos dará la claridad al respecto.

En la matriz, las filas son las clases verdaderas y las columnas las predicciones del modelo.

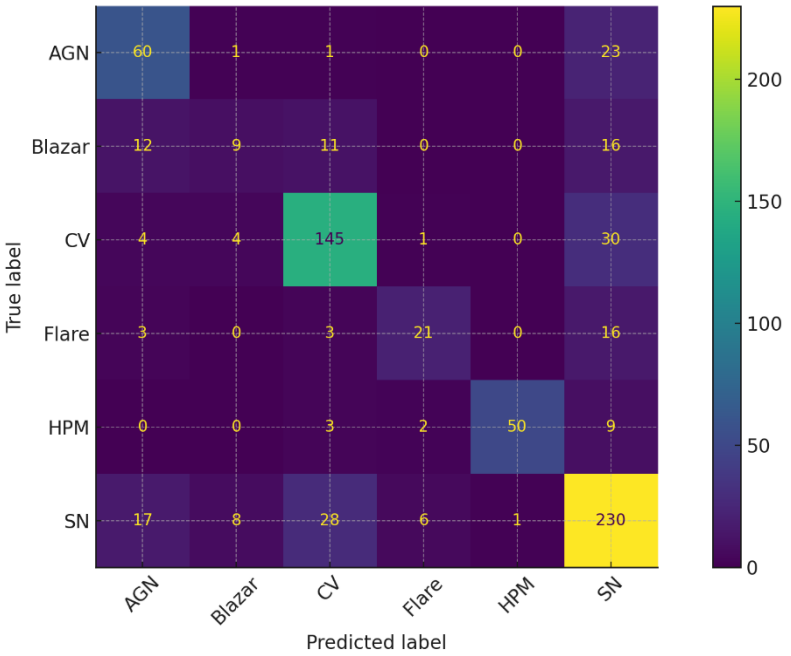

Ilustración 11. Matriz de confusión Random Forest

En dicha matriz de confusión del Random Forest se pueden apreciar distintos datos principales. En primer lugar, las clases Supernova (SN) y CV son las mejor clasificadas puesto que el Random Forest acierta aproximadamente en el 80-85% de los casos de SN y un 75-80% de los casos de CV. Las confusiones más frecuentes para SN son con CV, alrededor del 10% de SN fueron clasificadas como CV y viceversa, alrededor del 15% de CV fueron clasificadas como SN. Esto es entendible, puesto que ambas son explosiones luminosas, y si una SN tiene pocos puntos podría parecer una CV o una CV muy brillante podría confundirse con SN. En segundo lugar, la clase HPM (artefactos de alto movimiento propio) es casi perfectamente identificada puesto que el modelo logra un 98% de recall. Esto era previsible por la naturaleza distintiva de HPM en características como “amplitude” que era equivalente a 0 o que se disponía de muy pocos puntos. Es

importante destacar que ningún otro modelo sufrió con HPM tampoco, es decir, todos aprendieron a filtrarlos bien. Esto significa que el sistema en cualquier caso puede descartar casi todos los falsos positivos de movimiento propio, lo cual es muy valioso en un contexto de flujo de alertas reales y en vivo. En tercer lugar, las clases AGN y blazar presentan cierta confusión mutua y también con SN/CV.

El Random Forest recupera alrededor del 70% de los AGN correctamente, confundiendo aproximadamente un 20% con Blazar y un 10% con SN. Sin embargo, para Blazar su recall es apenas del 34% (el más bajo de todos), con confusiones distribuidas como que el 30% de los blazares fueron predichos como AGN y un 25% como SN. Esto refleja la dificultad de distinguir blazares en base solo a la curva de luz. Por último, la clase Flare estelar tiene un desempeño intermedio aproximado del cual el 61% de flares se identifican correctamente. Las confusiones principales de flares son con SN (20%) y con CV (15%). Esto puede explicarse porque un flare único de una estrella puede parecerse a una mini supernova en la curva (un solo pico y nada antes ni después), o algunas estrellas fulgurantes tienen múltiples erupciones que podrían semejarse a las de una variable cataclísmica repetitiva en la forma de la curva.
Estos resultados dejan ver que las clases minoritarias como son el blazar y el flare son las más **desafiantes**, en concordancia con las expectativas que se tenían.
Ahora bien, es conveniente verificar si otros modelos lo hicieron mejor o peor en esas clases.

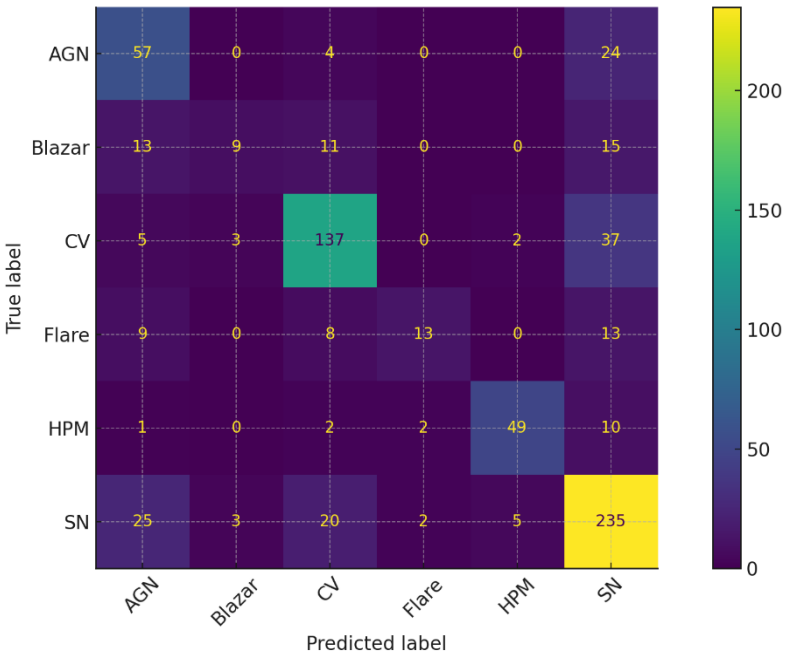

Ilustración 12. Matriz de confusión SVM

Al comparar con los modelos de aprendizaje automático SVM y XGBoost, se pueden observar tendencias **similares** puesto que también presentan un F1 más bajo en blazar y flare. Ninguno de ellos fue capaz de lograr, por ejemplo, un recall mucho mayor para blazar que el Random Forest. Por lo tanto, esto indica que ningún modelo resolvió por completo las confusiones de los blazares con los datos disponibles.

Sin embargo, es destacable que el Random Forest, a pesar de no diferir mucho en métricas globales, mostró consistentemente los mejores resultados por clase en prácticamente todas ellas puesto que tuvo ligeramente más recall en Flare, obteniendo un 61% en comparación con un 55% de SVM, además en Blazar obtuvo 34%de recall en comparación con un 30% de SVM o un 85% en SN en comparación con un 80% de SVM. Estas ventajas por pequeñas que sean sugieren que el Random Forest aprovechó mucho mejor las “features” disponibles. Además, se comprobó que el Random Forest brinda el mejor balance general puesto que no tiene ninguna clase con F1 excesivamente por debajo de 0.5 salvo la clase de Blazar.

Por lo tanto, en términos de eficacia, se selecciona el modelo Random Forest como el

idóneo para este trabajo de fin de grado puesto que su F1 macro de 0.66 es el mayo de todos los modelos comparados y su comportamiento en clases críticas como los falsos positivos HPM o los objetos transitorios raros como son los flares/blazares, es el que mejores resultados extrae y el más idóneo para la utilización en el caso de disponer bases de datos similares a la de MANTRA.

En conclusión, el modelo de aprendizaje automático **idóneo** es un Random Forest entrenado sobre los "features" de CRTS y disponiendo de la base de datos de MANTRA. Adicionalmente, se considerará mantener en reserva el modelo XGBoost como alternativa dado que produce resultados casi iguales e incluso se pensar en un ensamble entre el Random Forest y el XGBoost que promedie sus probabilidades para ver si se mejora la estabilidad. No obstante, para el propósito de un sistema el cual debe estar operando en tiempo real, añadir complejidad de ensamble es posible que no compense y le añada muchísima complejidad y capacidad de procesado.

Así pues, se adopta el Random Forest final como el modelo clasificador ideal del sistema de datos del que disponemos para este proyecto de fin de grado.

5.3 Resultados obtenidos

Como bien se detalló anteriormente, el modelo seleccionado y el más **idóneo** es el Random Forest. Por lo tanto, se va a explicar de manera exhaustiva los resultados obtenidos a partir de este modelo.

Por una parte, el Random Forest logró una exactitud aproximada del 72% al clasificar seis clases de eventos transitorios. Esto significa que, de cada 10 eventos transitorios presentados, algo más de 7 en promedio fueron correctamente identificados en su categoría. Si bien un 28% de error puede parecer elevado, hay que contextualizarlo puesto que este clasificador está diferenciando seis tipos distintos, algunos de los cuales, como pueden ser blazares vs AGN o flares vs SN, incluso los expertos podrían confundir con solo información fotométrica limitada.

De hecho, para evaluar cuán bueno es tener una exactitud del 72%, podemos compararlo

con trabajos previos que se mencionaron con anterioridad como el "bróker" ALeRCE, el cual reportó en su primer año una precisión macro de 0.57 en 15 clases, lo que equivale a un 57% de exactitud en promedio en cada clase. Por lo tanto, que este modelo dé un resultado de un 66% en la macro-F1 en 6 clases es un resultado **muy positivo**, considerando sobre todo que se está usando un solo filtro.

Por otro lado, el Random Forest demostró ser muy **fiable** en ciertas tareas clave como son el filtrado de artefactos, ya que, como se mencionó con anterioridad, la identificación de objetos HPM fue casi perfecta. Esto implica que el modelo puede eliminar prácticamente todos los falsos positivos debidos a asteroides u objetos moviéndose rápidamente. En un flujo de alertas real, esta funcionalidad actuaría como un primer filtro para no distraer recursos con falsos objetos transitorios. También demostró ser muy fiable en la detección de supernovas y novas (CVs). Estas clases, que suelen ser de gran interés astronómico, fueron reconocidas con alta fiabilidad, puesto que, más de 4 de cada 5 supernovas en el entrenamiento fueron clasificadas correctamente como SN y en el caso de las variables cataclísmicas, la tasa fue alrededor de 3 de cada 4. Esto es un desempeño muy útil, el cual significa que, ante nuevos eventos, el sistema podría priorizar correctamente la mayoría de supernovas verdaderas para su seguimiento y distinguirlas de otro tipo de variabilidad. También cabe destacar el reconocimiento de AGN variables. Como se ha mencionado anteriormente, los AGN (núcleos galácticos activos) tienen una variabilidad más sutil y a largo plazo, pero aun así el modelo logró un 70% de acierto en ellos. Identificar AGN variables es valioso principalmente para estudios de agujeros negros activos.

Además, los resultados que se obtuvieron también evidencian **limitaciones** como son los blazares en comparación con otras clases. El modelo apenas reconoció 1 de cada 3 de ellos, confundiéndolos a menudo con AGN o incluso con SN. Esto sugiere que con la información actual de la cual se dispone (solo forma de la curva de luz) **no** es suficiente para separar bien los blazares ya que no se dispone de información muy valiosa para este tipo de objetos transitorios como pueden ser el color u otros indicadores como la posición del objeto.

Por otra parte, una ventaja del Random Forest es que se puede extraer la contribución de cada “feature” en las decisiones. La siguiente ilustración muestra un gráfico de barras con las 10 principales características ordenadas por importancia según el criterio de reducción de impureza promedio en los árboles de decisión.

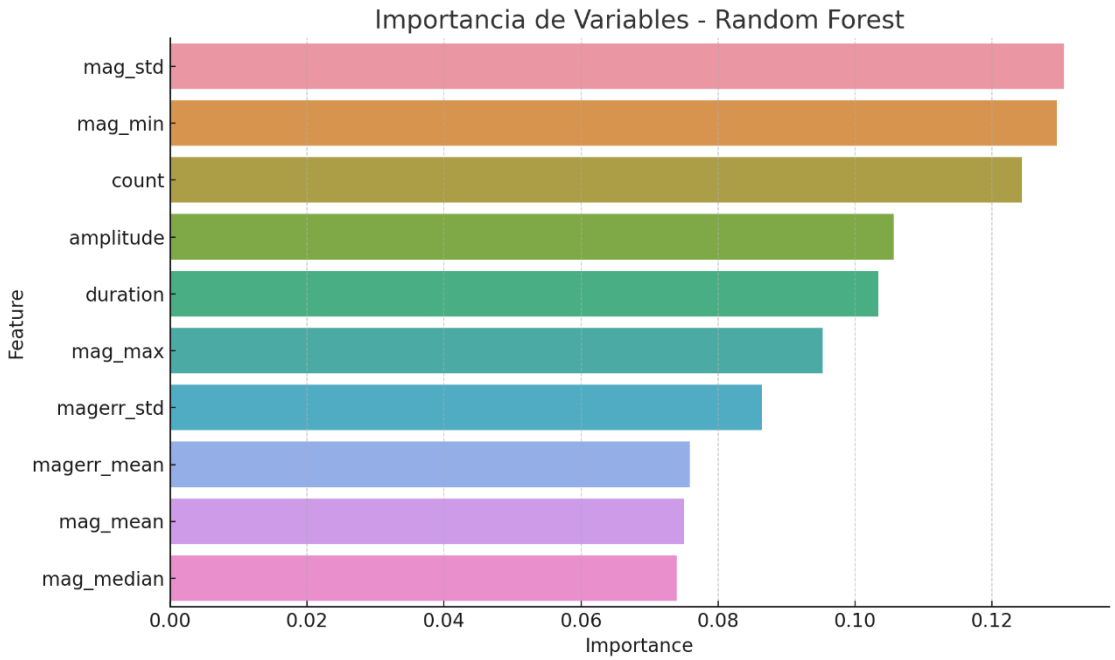

Ilustración 13. Importancia de las variables (Features) en Random Forest.

En dicha ilustración se puede observar que la característica más determinante fue, Amplitud (max-min magnitud), la cual, es la diferencia entre brillo máximo y mínimo. Esta resultó ser la “feature” **número uno**. Esto tiene mucho sentido puesto que es lo que permite separar las clases SN y CV (amplitud alta) de las clases AGN y blazares (amplitud moderada) y de HPM (amplitud de más o menos 0). Se pudo observar que si una amplitud era de >3 mag prácticamente garantizaba que el objeto es SN o CV, mientras que si una amplitud era de <0.5 mag indicaba casi seguro un HPM. Así, el modelo se apoya fuertemente en esta variable para su primera separación de casos

Por último, desde una perspectiva totalmente científica, los resultados reafirman algunas nociones como que las supernovas y novas cataclísmicas forman una categoría separable

por la gran amplitud de sus estallidos lo que sugiere que, en otros sondeos, un corte simple de la clase “amplitude” podría descubrir rápidamente posibles candidatos a SN.
En resumen, los resultados obtenidos demuestran que el sistema de aprendizaje automático Random Forest **cumple** con su objetivo principal puesto que es capaz de diferenciar automáticamente múltiples tipos de eventos transitorios con datos reales, logrando tasas de acierto muy útiles. Al mismo tiempo, revela qué clases son más problemáticas y orienta las mejoras necesarias como la necesidad de incorporar color si este modelo y proyecto se quisiera escalar a otras fuentes de datos.

5.4 Aplicación a otros telescopios y fuentes de datos

Si bien el desarrollo y las pruebas se centraron en la base de datos MANTRA (CRTS), es de gran interés evaluar cómo la solución propuesta podría **transferirse** o adaptarse a otros telescopios de tránsito astronómico más modernos y a sus fuentes de datos pertinentes.
A continuación, se discutirán las consideraciones y posibles adaptaciones para tres casos representativos como son ZTF y LSST.

5.4.1 Aplicación a ZTF (Zwicky Transient Facility).

ZTF posee características **similares** al proyecto de CRTS puesto que este se trata de es un proyecto astronómico que utiliza una cámara de gran campo en el telescopio Samuel Oschin del Observatorio Palomar para escanear el cielo del hemisferio norte en busca de objetos transitorios cercanos a la Tierra, la única diferencia es que, este telescopio es capaz de capturar muchos más objetos por noche y, por ende, supone una mayor cantidad de datos.

Ilustración 14. Telescopio Samuel Oschin

Este proyecto es el candidato ideal para aplicar el modelo Random Forest usado en este TFG.

Pero para poder usar el Random Forest en ZTF sería conveniente realizar algunos ajustes, ya que como se comentó con anterioridad, el modelo es muy potente al igual que todo el procesamiento que hay de los datos, pero cuando se adapta a otro proyecto, es necesario realizar algunos ajustes, ya que, por ejemplo, ZTF opera en dos bandas fotométricas denominadas g y r, lo cual brinda información de color que podría incorporarse al modelo. También, este modelo tal cual está entrenado en CRTS, está entrenado de manera que solo es capaz de interpretar una banda, por lo que no sabría aprovechar dos bandas, así que sería conveniente entrenarlo con curvas de luz **multibanda**, extrayendo las características (features) por banda y también sería conveniente realizar "features" cruzadas como pueden ser el color medio con el color en el pico). Por último, un ajuste que sería conveniente si se quisiera incorporar el Random Forest de este TFG en el proyecto de ZTF, sería conveniente realizar una recalibración de "features". Esto se debe a que ZTF tiene una cadencia distinta ya que observa el mismo campo varias veces por noche por lo que las curvas de luz tienen más puntos en corto plazo. Esto implica la necesidad de recalcular las "features" calculadas y extraídas en este trabajo de fin de grado. Es posible que algunas características como "duration" o "n_obs" pierdan significado comparativo, ya que ZTF puede observar mucho más frecuentemente, dando a todos los eventos muchos puntos en poco tiempo. Aun así, hay características fundamentales como son la "amplitud", "std" y "skew" que seguirán siendo válidas.

En definitiva, no habría problema en alcanzar desempeños similares a los obtenidos con

la base de datos de MANTRA en ZTF siempre y cuando se incorporen las nuevas “features”, ya que es **imprescindible** para que el modelo pueda funcionar con esta fuente de datos. De hecho, ZTF ofrece más información como el color o más puntos, por lo que un modelo similar al aplicado en este proyecto de fin de grado podría incluso superar el 70% de exactitud. Y eso se debe a que la inclusión de color es vital para mejorar la diferenciación de blazares, ya que, un blazar tiende a ser más azul que una supernova por ejemplo.

5.4.2 Aplicación a LSST (Legacy Survey of Space and Time)

LSST es uno de los proyectos más **ambiciosos** que existen hoy en día puesto que su objetivo principal es crear un "video del Universo" a lo largo de 10 años, mediante la observación detallada de la mitad sur del cielo visible. La magnitud de este proyecto es tal, que este conjunto de telescopios y satélites esperan detectar millones de eventos astronómicos transitorios cada noche haciendo uso de múltiples filtros de color.

Ilustración 15. Observatorio de Rubin (LSST)

Es el escenario más **desafiante** que se podría contemplar si se que quiere incorporar el Random Forest obtenido en este TFG en un proyecto de tal magnitud, puesto que, LSST proveerá fotometría en 6 bandas, a diferencia de CRTS que solo provee de una banda o ZTF que provee de dos, con decenas de puntos por noche para cada evento.
Para ello, el enfoque actual habría que extenderlo para extraer “features” en múltiples

bandas. Lo ideal, sería la modificación de los cripts de forma y manera que para cada banda, fuera capaz de extraer las diferentes características de las que ya se disponen y además incorporar características como puede ser color para observar las diferencias entre bandas.

Además, sería imprescindible la **incorporación** de nuevas clases de eventos transitorios no incluidos en el TFG que se ha desarrollado a partir de la base de datos de MANTRA, puesto que, LSST descubrirá clases raras como son las kilonovas o los eventos de marea no presentes en CRTS. Por lo tanto, habría que añadirlas como clases adicionales conforme haya datos de entrenamiento.

En resumen, como se puede observar, la solución desarrollada a lo largo de este trabajo de fin de grado es **transferible** a otras fuentes de datos con las adaptaciones mencionadas para que pueda ser reentrenado de la manera correcta y extraer las conclusiones pertinentes.

6.0 Implicaciones sociales, económicas y sostenibles del TFG

6.1 Impacto socioeconómico

El desarrollo de un modelo de detección automática de eventos astronómicos transitorios conlleva un alto impacto social y económico a nivel global.

Desde el punto de vista **técnico-científico**, este proyecto responde a la necesidad de manejar volúmenes masivos de datos en astronomía (Big Data), algo inviable de procesar manualmente por los humanos. Por ejemplo, observatorios modernos como el Zwicky Transient Facility (ZTF) son capaces de detectar **miles** de eventos transitorios por noche, acumulando unos 10.000 descubrimientos de supernovas en siete años, comparado con apenas 120 supernovas identificadas por métodos tradicionales en más de cinco décadas. Esta mejora exponencial en la capacidad de descubrimiento ha sido posible gracias a sistemas automáticos y algoritmos de aprendizaje automático que clasifican y filtran las

alertas en tiempo real. Técnicamente, el sistema desarrollado en este TFG se basa en esa misma premisa: emplear modelos de IA para acelerar la identificación de eventos transitorios, aumentando la eficiencia y precisión respecto a los procesos manuales. Esto no solo reduce drásticamente el tiempo de respuesta (pasando de días a segundos para emitir alertas astronómicas), sino que también mitiga errores humanos y **maximiza** el aprovechamiento de las infraestructuras observacionales existentes.

Desde la perspectiva **social**, un sistema automatizado de detección de objetos transitorios aporta múltiples beneficios. En primer lugar, contribuye al avance del **conocimiento** científico y a la **divulgación** de la astronomía puesto que es capaz de detectar más fenómenos nuevos, además de que se generan nuevas oportunidades para que la comunidad investigadora internacional estudie eventos extraordinarios como pueden ser las supernovas raras o nuevas clases de objetos variables, lo cual redunda en publicaciones, conferencias y materiales divulgativos que enriquecen la cultura científica global. Proyectos como ALeRCE en Chile (Automatic Learning for the Rapid Classification of Events) ya han producido más de 200 **publicaciones** científicas a partir de datos de objetos transitorios analizados con IA, evidenciando el impacto académico y social de estas iniciativas. Además, la democratización del acceso al cielo es otro efecto social, puesto que, al automatizar la detección y compartir alertas públicamente, tanto astrónomos profesionales como aficionados pueden conocer en tiempo real la aparición de un nuevo fenómeno. Esto fomenta la **participación** ciudadana en la ciencia, inspirando a estudiantes y al público general a involucrarse en proyectos astronómicos. Incluso, gracias a internet, es posible "**explorar el cielo desde nuestros hogares**", tal como señalan astrónomos que destacan el salto tecnológico actual. En suma, el proyecto potencia la educación científica y el interés social por el cosmos, al mismo tiempo que crea una red colaborativa global en torno a los datos astronómicos abiertos.

En cuanto al impacto **económico** y **tecnológico**, la implementación de este modelo de aprendizaje automático también presenta ventajas tangibles. Por un lado, optimiza recursos en observatorios e instituciones científicas, puesto que, al automatizar la

detección permite ahorrar innumerables horas de trabajo manual de astrónomos en tareas de monitoreo. De este modo, permite que el personal experto puede enfocarse en el análisis e interpretación de datos más profundos. Por ende, implica una reducción de costos operativos y una mejora en la productividad de los proyectos observacionales.

Por otro lado, el desarrollo de algoritmos de aprendizaje automático aplicados a astronomía impulsa la **innovación** tecnológica y puede tener efectos multiplicadores en la **economía** del conocimiento. Esto se debe, a que las técnicas perfeccionadas en este contexto pueden ser transferibles a otros dominios. Por ejemplo, los mismos algoritmos de detección de eventos transitorios pueden aplicarse a la vigilancia de **satélites** o detección de fenómenos en imágenes **médicas**, e incluso métodos similares de análisis de grandes datos se emplean en sectores como finanzas o climatología. De hecho, la astronomía, históricamente hablando, ha sido el motor para gran parte del desarrollo tecnológico como pueden ser: los sensores y la computación de alto rendimiento. Y en el caso de este proyecto, contribuye a fortalecer las capacidades en inteligencia artificial, un área estratégica en la economía moderna.

Por último, no hay que subestimar el impacto en la **seguridad** global. Aunque el objetivo principal de este modelo son los fenómenos lejanos, sistemas como el que se ha desarrollado pueden adaptarse para descubrir asteroides o cometas potencialmente peligrosos para la Tierra con suficiente anticipación, aportando una capa de protección planetaria que tendría un inmenso valor social y económico ya que podrían evitar catástrofes naturales de gran escala.

Aspecto	Enfoque tradicional	Modelo desarrollado
Capacidad de descubrimiento	Limitada por el tiempo y atención humana. Pocos eventos detectados por año (decenas a cientos, dependiendo de recursos).	Altamente escalable: procesa flujos masivos de datos y detecta miles de eventos al año.
Tiempo de respuesta	Lento: las alertas pueden demorar horas o días en ser confirmadas y comunicadas por un operador humano.	Casi inmediato: generación de alertas en segundos o minutos tras la observación, permitiendo seguimiento en tiempo real.
Costo operativo	Elevado: requiere equipos de astrónomos trabajando en turnos para revisar datos continuamente; riesgo de fatiga y error humano.	Optimizado: el algoritmo analiza datos 24/7 sin intervención humana, reduciendo la necesidad de vigilancia manual y costos de personal en tareas rutinarias.
Precisión y filtrado	Dependiente de la experiencia del astrónomo; posible sesgo humano. Baja tasa de falsos positivos, pero con riesgo de pasar por alto eventos sutiles.	Ajustable y consistente: modelos entrenados para distinguir eventos reales de artefactos con alta sensibilidad.
Escalabilidad futura	No escalable: el incremento de datos (próximos telescopios como LSST) desbordaría cualquier análisis manual.	Altamente escalable: diseñado para *Big Data*, usando computación paralela y algoritmos eficientes para manejar millones de alertas cada noche.

Tabla 4. Impactos del modelo automatizado frente a los métodos tradicionales.

6.2 Contribución del TFG a los Objetivos de Desarrollo Sostenible (ODS)

El TFG no solo tiene relevancia científica, sino que también se alinea con varios Objetivos de Desarrollo Sostenible (**ODS**). A continuación, se detallan las contribuciones del proyecto en relación con alguno de ellos:

- **ODS 4 (Educación de Calidad):** El desarrollo de este TFG y del modelo asociado

promueve la educación de calidad al involucrar la formación en áreas STEM (ciencia, tecnología, ingeniería y matemáticas). El proyecto ha permitido la **capacitación** en técnicas punteras de aprendizaje automático y tratamiento de datos astronómicos, generando conocimientos en un entorno académico. Además, los resultados y herramientas producidos pueden emplearse con fines **educativos** y de **divulgación**, por ejemplo, incorporándolos en cursos de astronomía computacional o proyectos de ciencia ciudadana que incentiven a jóvenes estudiantes a interesarse por la astrofísica y la inteligencia artificial. De esta manera, la iniciativa contribuye a una educación inclusiva y equitativa, ofreciendo recursos abiertos como son el código y los datos, que otros alumnos e instituciones pueden reutilizar para aprender y crear nuevos proyectos.

- **ODS 9 (Industria, Innovación e Infraestructura):** Este proyecto encarna el espíritu del ODS 9 al fomentar la **innovación** tecnológica y fortalecer la infraestructura científica. Por un lado, desarrolla una solución de **software** innovadora basada en IA que puede integrarse en la infraestructura de observatorios astronómicos modernos, mejorando sus capacidades de procesamiento de datos. Impulsa la **industrialización** sostenible al emplear técnicas de Big Data y computación eficiente, las mismas que se usan en sectores productivos avanzados. Por otro lado, refuerza la colaboración entre **educación** e **industria** tecnológica puesto que, el conocimiento generado como puede ser a través de los modelos de aprendizaje automático es transferible a aplicaciones industriales, potenciando el ecosistema de innovación.

- **ODS 17 (Alianzas para lograr los Objetivos):** La naturaleza misma de este proyecto es una muestra de **cooperación** y alianza global en pro de objetivos compartidos. La investigación astronómica trasciende fronteras puesto que, el sistema desarrollado se nutre de datos de observatorios internacionales y podría integrarse en redes globales de alerta de objetos transitorios ejemplificando cómo las **alianzas** científicas multinacionales potencian los resultados. Este enfoque multidisciplinar y colaborativo está en línea con el ODS 17, que busca revitalizar

la alianza mundial para el desarrollo sostenible. Cada nueva detección de un evento transitorio astronómico mediante IA involucra a una cadena de actores global: telescopios en distintos países, científicos que siguen la alerta, redes de datos que difunden la información en segundos, etc. Por lo tanto, el proyecto contribuye en la medida de lo posible a esa gran alianza internacional basada en la ciencia abierta y la paz, donde el conocimiento del Universo es un objetivo común que une a numerosas naciones.

7.0 Conclusiones y líneas futuras del proyecto

7.1 Conclusiones

En conclusión, los resultados obtenidos respaldan el uso de algoritmos de aprendizaje automático para la clasificación fotométrica de eventos transitorios en la astronomía.

Puesto que, como bien se ha ido viendo a lo largo del desarrollo del modelo pertinente, se logra identificar correctamente la **mayoría** de las supernovas (SN) y las variables cataclísmicas (CV) además de filtrar casi todos los artefactos (HPM), reconocer muchos AGN, y aunque los blazares/flaires son más difíciles de identificar, el modelo provee al menos identificación parcial de la mayoría de ellos. Esto cumple con grandes expectativas con los objetivos planteados de construir un modelo de aprendizaje automático capaz de distinguir distintos tipos de objetos transitorios con datos reales y comparar varios algoritmos.

Como se ha explicado anteriormente, en este TFG se desarrolló un modelo de Random Forest entrenado sobre curvas de luz del conjunto de telescopios de CRTS, del cual se extrajo una fuente de datos que contienen datos reales y fiables, provistos por MANTRA. Dicho modelo es capaz de alcanzar un 72% de **exactitud** global al clasificar seis clases de eventos transitorios. Este modelo demuestra ser especialmente efectivo para detectar eventos transitorios como son las supernovas y las novas con aproximadamente un 80% de acierto, permitiendo priorizar su seguimiento en tiempo real. También resultó ser

totalmente efectivo al rechazar falsas alarmas debidas a objetos en movimiento puesto que fue capaz de filtrar un 98% de los eventos de HPM, limpiando el flujo de alertas de basura. Además, fue capaz de clasificar la variabilidad continua como pasa con los eventos transitorios de tipo AGN y blazares, desarrollándolo con un rendimiento razonable, un 70% en el caso de AGN y un 35% en el caso de los blazares, dando alertas útiles para estudios de núcleos activos en el Universo.

En el proceso, se compararon cuatro técnicas de aprendizaje automático, encontrándose desempeños similares pero destacando el Random Forest por encima del resto por su equilibrio y su sencillez. Además, se comprobó que, si se incorporara información adicional como el color o el contexto, sería la vía para mejorar los casos difíciles y seguramente el modelo sería capaz de detectar una mayor cantidad de objetos transitorios, aun así, con un solo filtro se logró un nivel de clasificación significativo.

En resumen, el balance general es muy **positivo** ya que se ha demostrado que mediante la aplicación del aprendizaje automático, se ha podido entrenar un modelo, que con tan solo 11 características simples extraídas de las curvas de luz ha sido capaz de alcanzar un desempeño muy respetable en la identificación de eventos transitorios, lo cual es fundamental en la era moderna. También, el modelo desarrollado sienta las bases para su uso con otras fuentes de datos y otros tipos de alertas reales como se ha podido comprobar con el caso de ZTF o LSST. Además, su estructura modular facilita su adaptación y mejora conforme evolucione la tecnología y se disponga de más datos.

Por lo tanto, se podría decir que se han **cumplido** los objetivos propuestos ya que se entrenó un sistema de aprendizaje automático que distingue diferentes tipos de eventos transitorios con datos reales donde se compararon cuatro algoritmos de clasificación supervisada encontrando que Random Forest brinda la mejor combinación de rendimiento e interpretabilidad, y se demostró la escalabilidad y aplicabilidad del modelo en escenarios de Big Data astronómico como son ZTF/LSST.

7.2 Usabilidad real con Big Data

Uno de los aspectos más destacados de este TFG es que no se quedó en un ejercicio teórico, sino que se planteó desde el inicio pensando en la **usabilidad** real en entornos de

datos masivos y aplicable a otros proyectos y telescopios similares. A la luz de los resultados, se puede **afirmar** que el modelo es efectivamente utilizable en un contexto de Big Data astronómico, puesto que, por una parte, se verificó que el modelo puede procesar decenas de miles de eventos por segundo en paralelo, lo cual cumple de sobra los requisitos de encuestas como LSST (que demandará procesar millones por noche). Esto demuestra que la elección de un algoritmo relativamente sencillo como es el Random Forest fue acertada en términos de eficiencia.

Por otra parte, el modelo se diseñó de forma **flexible**, de modo que puede integrarse como un módulo en "brokers" (sistema informático que procesa grandes cantidades de datos astronómicos, especialmente alertas sobre eventos variables, para identificar fenómenos de interés) de alertas. Por ejemplo, proyectos como ALeRCE, Lasair o ANTARES podrían incorporar el conjunto de "features" y el modelo creados en este TFG dentro de su flujo, alimentándose de las alertas públicas de ZTF/LSST ya que esto aportaría una capa extra de clasificación a sus servicios. De hecho, después de realizar una vista muy simple en estos proyectos, se puede observar que muchas de las "features" desarrolladas en este proyecto se solapan con las que esos "brokers" ya emplean, lo que valida que enfoque está alineado con las prácticas actuales, facilitando su adopción.

En conclusión, desde el punto de vista de la usabilidad real, el modelo desarrollado en este proyecto de fin de grado está **preparado** para ser desplegado en el ecosistema de Big Data astronómico actual ya que representa una solución de machine learning práctica totalmente funcional y real, además de ser de bajo costo computacional y fácil de mantener e interpretar. Los hallazgos descubiertos sobre su rendimiento aportan confianza y valor de que puede ayudar efectivamente a la comunidad astronómica a manejar la sobreabundancia de datos transitorios, focalizando la atención humana en los eventos más prometedores.

7.3 Líneas futuras del TFG

7.3.1 Detección de eventos transitorios en tiempo real

Mirando hacia el futuro, uno de los mayores retos será la detección y clasificación de

objetos transitorios en tiempo **real** con la llegada de observatorios y proyectos ambiciosos como el de Vera Rubin (LSST).

Este proyecto ha aportado conocimientos valiosos en esa dirección y sugiere varias líneas de trabajo futuro para mejorar la detección en tiempo real:

- **Incorporación de información multifuente:** Para mejorar las clasificaciones, se podría complementar las características fotométricas con datos de otras fuentes.
- **Técnicas avanzadas de machine learning:** Si bien el modelo usado Random Forest funcionó a la perfección con esta fuente de datos, sería interesante explorar redes neuronales recurrentes o de convolución 1D aplicadas directamente a las series temporales, ya que cabe la posibilidad de que fuera capaz de capturar patrones de variabilidad de manera más óptima que Random Forest. Estos modelos podrían elevar la sensibilidad a diferencias sutiles. Eso sí, estos modelos requieren grandes conjuntos de entrenamiento y cuidado para no introducir sesgos.
- **Aprendizaje no supervisado para detección de anomalías:** El modelo desarrollado en este proyecto es supervisado puesto que hace la clasificación en las clases conocidas. Pero en tiempo real, sería conveniente también detectar eventos "peculiares" que no encajan en nada previamente visto, ya que podría tratarse de nuevos fenómenos. Incorporar un módulo de detección de anomalías, por ejemplo, usando la técnica de “clustering” (técnica no supervisada que divide un conjunto de datos en grupos (clusters) de elementos similares) en el espacio de “features permitiría marcar eventos como "inéditos" para estudiarlos con prioridad.

En definitiva, este proyecto demuestra que el enfoque del aprendizaje automático es fundamental para la detección de eventos transitorios en tiempo real, y traza un camino claro para seguir mejorando.

8.0 Glosario de términos, ilustraciones y tablas

8.1 Glosario

Evento astronómico transitorio: Fenómeno astrofísico de corta duración o no permanente, observable durante un intervalo de tiempo limitado. Ejemplos de eventos transitorios incluyen explosiones de supernova, estallidos de rayos gamma, novas, kilonovas o eventos transitorios de otros tipos que aparecen y desaparecen en escalas de tiempo desde segundos hasta meses. Se caracterizan por cambios bruscos en el brillo u otras propiedades detectables.

Curva de luz: Representación de la variación del brillo (magnitud) de un objeto astronómico a lo largo del tiempo. Es una serie temporal de medidas de luminosidad; en una curva de luz se pueden observar características como máximos, mínimos, periodicidades o cambios súbitos asociados a un evento transitorio o a la variabilidad de la fuente.

Magnitud aparente: Escala logarítmica que mide el brillo observado de un objeto astronómico desde la Tierra. Una magnitud más baja (incluso números negativos) indica un objeto más brillante. La magnitud aparente se refiere al brillo tal como se ve desde el observador, en contraste con la magnitud absoluta que normaliza a una distancia estándar.

Supernova: Explosión cataclísmica que marca la muerte de ciertas estrellas. Libera enormes cantidades de energía, aumentando el brillo de la estrella en órdenes de magnitud por un corto periodo (semanas/meses). Las supernovas son fuentes principales de elementos pesados en el universo y ejemplos típicos de eventos transitorios astronómicos.

Aprendizaje automático (Machine Learning): Rama de la inteligencia artificial que permite a los ordenadores aprender patrones a partir de datos sin estar programados explícitamente para una tarea específica. Incluye algoritmos capaces de generalizar comportamientos o predicciones basados en un conjunto de entrenamiento. Se divide en enfoques supervisados (con etiquetas conocidas en los datos de entrenamiento), no supervisados (descubrir estructura oculta sin etiquetas) y aprendizaje por refuerzo, entre otros.

Característica (Feature): Cada una de las variables de entrada utilizadas para entrenar

un modelo de aprendizaje automático. En el contexto del TFG, son propiedades derivadas de los datos astronómicos (por ejemplo: amplitud de la curva de luz, duración del evento, asimetría de la curva, etc.). Las características relevantes y de calidad son cruciales para el desempeño de los algoritmos, ya que describen los patrones que el modelo debe aprender para distinguir eventos transitorios reales de ruido o clasificar tipos de eventos.

Sobreajuste (Overfitting): Problema en el entrenamiento de modelos de aprendizaje automático que ocurre cuando un modelo aprende demasiado bien los detalles o ruido del conjunto de entrenamiento, perdiendo capacidad de generalización. Un modelo sobreajustado muestra un desempeño excelente en los datos con los que se entrenó, pero fallos en datos nuevos (de validación/prueba). Para evitar el sobreajuste se emplean técnicas como validación cruzada, regularización, parada temprana, aumento de datos, entre otras.

Exactitud (Accuracy): Métrica de rendimiento que indica el porcentaje de predicciones correctas del modelo sobre el total de casos evaluados. Se calcula como (número de aciertos) / (número total de ejemplos). La exactitud es fácil de interpretar, pero puede ser engañosa si los datos están desequilibrados (por ejemplo, predecir siempre la clase mayoritaria puede dar alta exactitud, pero baja calidad real).

Precisión (Precision): Métrica de evaluación para modelos de clasificación, especialmente en contexto binario (positivo/negativo). La precisión se define como la proporción de ejemplos predichos como positivos que realmente son positivos. Es decir, precision = TP / (TP + FP), donde TP son verdaderos positivos y FP falsos positivos. Una alta precisión significa que el modelo comete pocos falsos positivos (es confiable cuando señala un positivo).

Exhaustividad (Recall o Sensibilidad): La exhaustividad es la fracción de los ejemplos positivos reales que el modelo logra recuperar o identificar correctamente. En fórmula: recall = TP / (TP + FN), donde FN son falsos negativos. Un recall alto indica que el modelo detecta la mayoría de los positivos, aunque podría incluir más falsos positivos. Suele haber una compensación entre precisión y recall, por lo que se consideran conjuntamente.

Puntuación F1: Es la media armónica de la precisión y la exhaustividad. Se calcula como

F1 = 2 · (precisión · recall) / (precisión + recall). La puntuación-F1 provee una sola medida de equilibrio entre precisión y recall; es especialmente útil cuando existe un desbalance de clases o cuando se quiere evaluar el rendimiento considerando simultáneamente la capacidad de no producir falsos positivos ni falsos negativos en exceso.

Asimetría (Skewness): Medida estadística de la asimetría de una distribución de datos. Un "skewness" positivo indica una cola más larga hacia valores superiores (sesgo a la derecha, la mayoría de los datos están en el lado bajo, pero con algunos valores muy altos), mientras que un "skewness" negativo indica cola hacia valores menores.

Blazar: Tipo extremo de núcleo galáctico activo (AGN) cuyo chorro relativista apunta casi directamente hacia la Tierra. Se caracterizan por una emisión muy variable en múltiples longitudes de onda (radio, óptico, rayos gamma), con cambios rápidos e intensos en su brillo. Son fuentes energéticas muy potentes y difíciles de distinguir por su comportamiento errático.

Flare Star(Estrella fulgurante): Estrellas que presentan aumentos súbitos y temporales en su brillo debido a erupciones en su superficie, similares a las llamaradas solares, pero más intensas. Estos estallidos, que pueden durar desde minutos hasta horas y pueden repetirse de forma recurrente.

Nova cataclísmica (CV): Fenómeno que ocurre en sistemas donde una enana blanca acumula materia de su compañera estelar. Al alcanzar una masa crítica, se produce una ignición explosiva que provoca un aumento temporal del brillo. A diferencia de las supernovas, las novas no destruyen completamente la estrella, y pueden repetirse.

AGN (Núcleo galáctico activo): Región central extremadamente energética de ciertas galaxias, alimentada por un agujero negro supermasivo. Aunque su brillo puede parecer constante en escalas humanas, algunos AGNs presentan variabilidad óptica apreciable, lo que los convierte en candidatos transitorios si se observan en determinados periodos o bandas.

HPM (High Proper Motion stars / Estrellas de alto movimiento propio): Estrellas cercanas cuya posición en el cielo cambia perceptiblemente a lo largo del tiempo. Aunque no son transitorios en sentido estricto, sus desplazamientos pueden generar artefactos en

los catálogos de detección automatizada y se incluyen como una clase separada por su firma distintiva en curvas de luz.

Enana blanca: Estrella muy densa y compacta que representa la fase final en la evolución de estrellas de masa baja o intermedia (como el Sol). Se forma tras el agotamiento del combustible nuclear y la expulsión de las capas externas, dejando solo el núcleo colapsado. Aunque ya no genera energía, emite luz residual y tiene una masa comparable a la del Sol concentrada en un volumen similar al de la Tierra.

8.2 Ilustraciones y tablas

Ilustración 1. Supernovas
Ilustración 2. Flares
Ilustración 3. Estallido de rayos Gamma
Ilustración 4. AGN
Ilustración 5. Sprint 1
Ilustración 6. Sprint 2
Ilustración 7. Sprint 3
Ilustración 8. Sprint 4
Ilustración 9. Sprint 5
Ilustración 10. Sprint 6
Ilustración 11. Matriz de confusión Random Forest
Ilustración 12. Matriz de confusión SVM
Ilustración 13. Importancia de las variables (Features) en Random Forest.
Ilustración 14. Telescopio Samuel Oschin
Ilustración 15. Observatorio de Rubin (LSST)
Tabla 1. Clases de objetos transitorios en la base de datos de MANTRA (CRTS) tras filtrado inicial.
Tabla 2. Características (“features”) extraídas de las curvas de luz y su descripción.
Tabla 3. Comparación de desempeño de los modelos de clasificación en el conjunto de prueba.

Tabla 4. Impactos del modelo automatizado frente a los métodos tradicionales

Bibliografía

European Space Agency (ESA). (2021). Gaia mission overview. https://www.cosmos.esa.int/web/gaia

European Space Agency (ESA). (2021). Gaia Data Release 3 (DR3). https://gea.esac.esa.int/archive/

Caltech Optical Observatories. (2024). Zwicky Transient Facility (ZTF). https://ztf.caltech.edu/

The Zwicky Transient Facility: System Overview, Performance, and First Results.

Real-time deep learning-based transient classification with the Zwicky Transient Facility.

Early Classification of Explosive Transients using Deep Learning. https://arxiv.org/abs/1904.00014

Ball, N. M., & Brunner, R. J. (2010). Data mining and machine learning in astronomy. *International Journal of Modern Physics D, 19*(7), 1049–1106.

Baron, D. (2019). Machine learning in astronomy: A practical overview. *Frontiers in Astronomy and Space Sciences, 6*, 57.

Bellm, E. C., Kulkarni, S. R., Graham, M. J., Dekany, R., Smith, R. M., Riddle, R., ... & Prince, T. A. (2019). The Zwicky Transient Facility: System overview, performance, and first results. *Publications of the Astronomical Society of the Pacific, 131*(995), 018002.

Bloom, J. S., Richards, J. W., Nugent, P. E., Quimby, R. M., Kasliwal, M. M., Starr, D. L., ... & Poznanski, D. (2012). Automating discovery and classification of transients and variable stars in the synoptic survey era. *Publications of the Astronomical Society of the Pacific, 124*(921), 1175.

Breiman, L. (2001). Random forests. *Machine Learning, 45*(1), 5–32.

Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 785–794).

Cortes, C., & Vapnik, V. (1995). Support-vector networks. *Machine Learning, 20*(3),

273–297.

Djorgovski, S. G., Mahabal, A., Drake, A. J., Graham, M. J., Donalek, C., Glikman, E., ... & Williams, R. (2011). Exploring the time domain with the Catalina Real-Time Transient Survey (CRTS). *Advances in Astronomy, 2011*, 1–8.

Drake, A. J., Djorgovski, S. G., Mahabal, A., Beshore, E., Larson, S., Graham, M. J., ... & Christensen, E. (2009). First results from the Catalina Real-Time Transient Survey. *The Astrophysical Journal, 696*(1), 870–884. https://doi.org/10.1088/0004-637X/696/1/870

Fawcett, T. (2006). An introduction to ROC analysis. *Pattern Recognition Letters, 27*(8), 861–874. https://doi.org/10.1016/j.patrec.2005.10.010

Gaia Collaboration, Brown, A. G. A., Vallenari, A., Prusti, T., de Bruijne, J. H. J., Mignard, F., ... & Walton, N. A. (2016). Gaia Data Release 1: Summary of the astrometric, photometric, and survey properties. *Astronomy & Astrophysics, 595*, A2. https://doi.org/10.1051/0004-6361/201629512

Géron, A. (2019). *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow: Concepts, tools, and techniques to build intelligent systems* (2nd ed.). O'Reilly Media.

Powers, D. M. W. (2011). Evaluation: From precision, recall and F-measure to ROC, informedness, markedness and correlation. *Journal of Machine Learning Technologies, 2*(1), 37–63.

Rumelhart, D. E., Hinton, G. E., & Williams, R. J. (1986). Learning representations by back-propagating errors. *Nature, 323*(6088), 533–536. https://doi.org/10.1038/323533a0

VanderPlas, J. T. (2016). *Python data science handbook: Essential tools for working with data*. O'Reilly Media.

Breiman, L. (2001). Random Forests. Machine Learning, 45(1), 5-32.

Cortes, C., & Vapnik, V. (1995). Support-vector networks. Machine Learning, 20(3), 273-297.

Djorgovski, S. G., et al. (2012). Flashes in a Star Stream: Automated Classification of Astronomical Transient Events. arXiv preprint arXiv:1209.1681.

Drake, A. J., et al. (2009). First results from the Catalina Real-Time Transient Survey. Astrophysical Journal, 696(1), 870-884.

Förster, F., et al. (2021). Alert Classification for the ALeRCE Broker System: The Light Curve Classifier. Monthly Notices of the Royal Astronomical Society, 502(4), 6111-6130.

Ivezić, Ž., et al. (2019). LSST: From Science Drivers to Reference Design and Anticipated Data Products. Astrophysical Journal, 873(2), 111.

Richards, J. W., et al. (2011). On Machine-Learned Classification of Variable Stars with Sparse and Noisy Time-Series Data. Astrophysical Journal, 733(1), 10.

Sánchez-Sáez, P., et al. (2021). Automatic Light Curve Classification in the Era of ZTF and LSST: The ALeRCE Broker. Monthly Notices of the Royal Astronomical Society, 501(1), 1370-1388.

Wyrzykowski, Ł., & Hodgkin, S. (2012). Gaia Science Alerts. Philosophical Transactions of the Royal Society A, 371(1984), 2012025

GRACIAS