



---

# INTELIGENCIA ARTIFICIAL CENTRADA EN LOS HUMANOS PARA LA TOMA DE DECISIONES FINANCIERAS (HUMAN-CENTRIC ARTIFICIAL INTELLIGENCE FOR SUPPORTING FINANCE DECISIONS)

---

Trabajo de Fin de Grado del Grado en Ingeniería Informática, Facultad de Informática,  
Universidad Complutense de Madrid



CURSO ACADÉMICO 2019-20  
MIGUEL SIERRA ESTESO  
Director: IVÁN GARCÍA-MAGARIÑO GARCÍA

# Índice

Resumen.....	5
Abstract.....	5
Introducción.....	6
Introduction.....	7
<b>CAPÍTULO 1. Estado del Arte.....</b>	<b>8</b>
1.1.- ¿Qué es una hipoteca?.....	8
1.2.- Machine Learning y la industria hipotecaria.....	8
1.3.- Inteligencia Artificial centrada en los Humanos.....	8
1.4.- Inteligencia Artificial explicable.....	9
1.5.- Revisión de artículos científicos similares.....	10
<b>CAPÍTULO 2. Especificación de Requisitos Software.....</b>	<b>12</b>
2.1.- Introducción.....	12
2.1.1.- Propósito.....	12
2.1.2.- Alcance.....	12
2.1.3.- Definiciones, acrónimos y abreviaturas.....	12
2.2.- Descripción general.....	13
2.2.1.- Perspectiva del producto.....	13
2.2.2.- Funcionalidades del producto.....	13
2.2.3.- Características del usuario.....	14
2.2.4.- Restricciones.....	14
2.2.5.- Supuestos y dependencias.....	14
2.2.6.- Requisitos futuros.....	14
2.3.- Requisitos específicos.....	15
2.3.1.- Interfaces externos.....	15
2.3.2.- Funcionalidades.....	15
2.3.3.- Requisitos de rendimiento.....	17
2.3.4.- Restricciones de diseño.....	17
2.3.5.- Atributos del sistema software.....	17
<b>CAPÍTULO 3. Planificación.....</b>	<b>19</b>
3.1.- Modelo de proceso.....	19
3.2.- Estimaciones del proyecto.....	19
3.2.1.- Datos históricos.....	19
3.2.2.- Técnicas de estimación.....	19

3.2.3.- Estimaciones de esfuerzo .....	19
3.3.- Planificación temporal.....	20
3.3.1.- Estructura de descomposición del trabajo .....	20
3.3.2.- Gráfico Gantt .....	20
3.3.3.- Red de tareas .....	21
<b>CAPÍTULO 4. Diagramas UML.....</b>	<b>23</b>
4.1.- Diagrama de caso de uso.....	23
4.2.- Diagrama de clases.....	23
4.3.- Diagrama de secuencias.....	26
<b>CAPÍTULO 5. Implementación del sistema.....</b>	<b>28</b>
5.1.- Dataset .....	28
5.1.1.- Exploración de los datos .....	28
5.1.2.- Datos ausentes .....	30
5.1.3.- Visualización de los datos .....	30
5.1.4.- Partición de los datos.....	32
5.1.5.- Almacenamiento del dataset de entrenamiento en una base de datos.....	32
5.1.6.- Normalización de los datos .....	32
5.2.- Algoritmos de clasificación.....	33
5.2.1.- Redes neuronales (Neural Networks).....	33
5.2.2.- Máquinas Soporte Vector (SVM).....	33
5.2.3.- K-means .....	34
5.3.- Generación de explicaciones.....	34
5.4.- Ejemplo de uso.....	35
<b>Conclusión.....</b>	<b>40</b>
<b>Conclusion.....</b>	<b>41</b>
<b>Bibliografía.....</b>	<b>42</b>

# Índice de Figuras

Figura 1: Grupos interesados en la Inteligencia Artificial explicable .....	10
Figura 2: Gráfico de Gantt (elaborado con Microsoft Project).....	21
Figura 3: Red de tareas .....	22
Figura 4: Diagrama de caso de uso (elaborado con BoUML).....	23
Figura 5: Diagrama de clases (elaborado con BoUML).....	26
Figura 6: Diagrama de secuencias (elaborado con BoUML).....	27
Figura 7: Cantidad de muestras de cada clase en el dataset.....	31
Figura 8: Distribución de los atributos en el dataset, separados por clases (I) .....	31
Figura 9: Distribución de los atributos en el dataset, separados por clases (II).....	32
Figura 10: Exactitud de la red neuronal de scikit-learn según el término de regularización y el número de neuronas en la capa oculta.....	33
Figura 11: Exactitud de SVM con kernel gaussiano de scikit-learn según el valor de sigma ( $\sigma$ ) para distintos valores del parámetro de penalización (C).....	34
Figura 12: Algoritmo para encontrar "Most Weighted Path" en la red neuronal.....	35
Figura 13: Interfaz gráfica.....	36
Figura 14: Ejemplo de uso .....	37
Figura 15: Gráfica asociada a la red neuronal.....	38
Figura 16: Gráfica asociada al algoritmo K-means.....	38

# Índice de Tablas

<b>Tabla 1: Estimación del esfuerzo en horas de trabajo .....</b>	<b>19</b>
<b>Tabla 2: Estructura de descomposición del trabajo .....</b>	<b>20</b>

## Resumen

Este Trabajo de Fin de Grado se ha centrado en el desarrollo de una inteligencia artificial centrada en los humanos para la toma de decisiones financieras, concretamente, la concesión de una hipoteca. Para el desarrollo de este proyecto, nos hemos adentrado en el ámbito de la Inteligencia Artificial centrada en los Humanos para acompañar la decisión tomada por los algoritmos de Inteligencia Artificial empleados para llevar a cabo dicha decisión financiera con una explicación autogenerada acerca del conocimiento aprendido por los mismos y que justifique dicha decisión financiera, proporcionando a los posibles usuarios de la inteligencia artificial una explicación fácil de entender acerca de las decisiones tomadas. En esta memoria, se describe con detalle el desarrollo del proyecto.

**Palabras clave:** Inteligencia Artificial, Redes Neuronales, Inteligencia Artificial centrada en los Humanos, Explicaciones Autogeneradas, Finanzas, Decisiones Financieras, Hipotecas.

## Abstract

This Final Degree Project has focused on the development of a Human-centric Artificial Intelligence approach for financial decision-making, specifically, the granting of a mortgage. For the development of this project, we have entered the field of Human-centric Artificial Intelligence to accompany the decision made by the Artificial Intelligence algorithms used to carry out said financial decision with a self-generated explanation about the knowledge learned by themselves and that justifies such financial decision, providing potential users of this Artificial Intelligence approach with an easy-to-understand explanation of the decisions made. In this report, the development of the project is described in detail.

**Key words:** Artificial Intelligence, Neural Networks, Human-centric Artificial Intelligence, Self-generated Explanations, Finances, Financial Decisions, Mortgages.

# Introducción

La mayoría de las técnicas de Inteligencia Artificial aplicadas en la actualidad funcionan como una caja negra: por ejemplo, una red neuronal es entrenada con una serie de datos y, gracias a ello, el sistema es capaz de proponer decisiones para nuevos casos propuestos. Sin embargo, los humanos generalmente quieren entender las decisiones tomadas para fiarse de las mismas, echando de menos una explicación acerca de dichas decisiones.

En el mundo financiero, si se tiene que tomar una decisión, por ejemplo, acerca de la concesión de una hipoteca a un cierto individuo o grupo de individuos, aquellos solicitantes de dicho préstamo desearán obtener una explicación que justifique la decisión tomada y que resulte comprensible.

El objetivo de este Trabajo de Fin de Grado consiste en el desarrollo de una inteligencia artificial centrada en los humanos encargada de tomar la decisión financiera consistente en la concesión de una hipoteca, proporcionando además una explicación que justifique dicha decisión y cuya comprensión resulte sencilla para el usuario del sistema.

Esta memoria incluye una descripción detallada de todo el proceso que se ha seguido para llevar a cabo el desarrollo de este proyecto, organizándose el contenido de la misma de la siguiente manera:

- En primer lugar, se presenta una descripción del estado del arte en la cual se explica el contexto en el cual nos situamos para identificar el problema que nuestra inteligencia artificial trata de resolver, y se presentan algunos artículos científicos con características similares que han sido consultados previamente al desarrollo de este proyecto.
- Luego, se presenta una Especificación de Requisitos Software (SRS), realizada para definir los requisitos previos al desarrollo de la inteligencia artificial.
- Posteriormente, se presenta una planificación seguida durante el desarrollo de este Trabajo de Fin de Grado.
- Después, se presentan una serie de diagramas UML (*Unified Modeling Language*) que capturan la descripción del comportamiento de la inteligencia artificial.
- A continuación, se presentan detalles acerca de la implementación de la inteligencia artificial en cuestión: *dataset* utilizado para el entrenamiento del sistema, algoritmos encargados de la toma de decisiones, cómo se generan las explicaciones que justifican dichas decisiones..., además de un ejemplo de uso de la misma.
- Por último, se analizan los resultados obtenidos y se elabora una conclusión tras la realización de este Trabajo de Fin de Grado.

# Introduction

Most of the Artificial Intelligence techniques applied today work like a black box: for example, a neural network is trained with a series of data and, thanks to this, the system is able to propose decisions for new proposed cases. However, humans generally want to understand the decisions made to trust them, missing an explanation for those decisions.

In the financial world, if a decision has to be made, for example, about granting a mortgage to a certain individual or group of individuals, those applicants for such loan will want to get an explanation that justifies the decision made and that is understandable.

The objective of this Final Degree Project is to develop a Human-centered Artificial Intelligence approach in charge of making the financial decision consisting of the granting of a mortgage, also providing an explanation that justifies said decision, and whose understanding is easy for the system user.

This report includes a detailed description of the entire process that has been followed to carry out the development of this Project, the content of which is organized as follows:

- First, a description of the state of the art is presented, in which the context in which we are located is explained to identify the problem our Artificial Intelligence approach tries to solve, and some scientific articles with similar characteristics that have been consulted prior to the development of this project are presented.
- Then, a Software Requirements Specification (SRS) is presented, carried out to define the prerequisites for the development of this Artificial Intelligence approach.
- Subsequently, a planning followed during the development of this Final Degree Project is presented.
- Then, a series of UML (Unified Modeling Language) diagrams are presented, that capture the description of the behaviour of this Artificial Intelligence approach.
- Next, details about the implementation of the Artificial Intelligence approach in question are presented: dataset used for training the system, algorithms responsible for making decisions, how the explanations that justify these decisions are generated..., plus an example of using it.
- Finally, the results obtained are analyzed and a conclusion is drawn up after completing this Final Degree Project.



# CAPÍTULO 1

## Estado del Arte

El objetivo de este primer capítulo es realizar una descripción del contexto en el cual nos encontramos para identificar el problema que nuestra inteligencia artificial trata de resolver, además de presentar artículos científicos que han sido revisados previamente al desarrollo e implementación de la misma.

### 1.1.- ¿Qué es una hipoteca?

Una hipoteca es un derecho real de garantía que se constituye para asegurar el cumplimiento de una obligación (normalmente, de pago de un crédito o préstamo), que otorga a su titular un derecho de realización de valor de un bien (generalmente, inmueble), el cual, aunque gravado, permanece en poder de su propietario, pudiendo el acreedor hipotecario, en caso de que la deuda garantizada no sea satisfecha en el plazo pactado, promover la venta forzosa del bien gravado con la hipoteca, cualquiera que sea su titular en ese momento (concepto de *reipersecutoriedad*), para, con su importe, hacerse el pago del crédito debido, hasta donde alcance el importe obtenido con la venta forzosa promovida para la realización de los bienes hipotecados [1].

### 1.2.- Machine Learning y la industria hipotecaria

La Inteligencia Artificial es considerada como un avance revolucionario en cada aspecto de nuestras vidas, y una de las industrias más beneficiadas por la llegada de la Inteligencia Artificial (y más concretamente, el campo conocido como *Machine Learning*) ha sido la industria hipotecaria.

Durante años, los prestadores de hipotecas se han encontrado estancados en procedimientos manuales que les han impedido alcanzar una mayor calidad de préstamo y una mejor experiencia del consumidor por un coste razonable.

La entrada de la Inteligencia Artificial y las técnicas de *Machine Learning* en el sector hipotecario ha permitido entender de un mejor modo cómo se originan las hipotecas, cómo se garantizan y cómo son vendidas en el mercado secundario, favoreciendo el análisis de datos, el reconocimiento de patrones y el aprendizaje de los sistemas a medida que una mayor cantidad de información sale a la luz.

En el año 2018, Fannie Mae (FNMA, *Federal National Mortgage Association*) encontró que la Inteligencia Artificial y el *Machine Learning* se adentraban más en este sector: “el 63% de los prestadores estaban familiarizados con la Inteligencia Artificial y la tecnología de *Machine Learning*, y el 27% de los prestadores ya estaban desplegándolas” [2].

### 1.3.- Inteligencia Artificial centrada en los Humanos

La Inteligencia Artificial centrada en los Humanos se define por sistemas que se encuentran en un proceso de mejora continua debido a la colaboración y la aportación humanas, proporcionando una experiencia efectiva entre humanos y robots. Al desarrollar inteligencia artificial con el objetivo de comprender el lenguaje, las emociones y el comportamiento humanos, este tipo de Inteligencia Artificial logra superar los límites de las soluciones previamente limitadas para cerrar la brecha que separa la máquina y el ser humano [3].

Entre los beneficios de la Inteligencia Artificial centrada en los Humanos, podemos mencionar los siguientes [3]:

- **Toma de decisiones informadas:** el objetivo de la Inteligencia Artificial centrada en los Humanos no es reemplazar por completo a los humanos, sino mejorar nuestras capacidades a través de tecnología inteligente e informada por los humanos. Mediante la combinación de la precisión del aprendizaje automático con las aportaciones y los valores humanos, permite tomar decisiones más informadas y desarrollar estrategias y soluciones más claras para los desafíos.
- **Confiabilidad y escalabilidad:** la Inteligencia Artificial centrada en los Humanos toma nuestras habilidades como pensadores humanos y permite que nuestras ideas escalen para satisfacer necesidades de datos mucho más grandes. Su propósito es ayudar a los humanos gracias a su aportación y comprensión. Adoptar este enfoque pone parte del trabajo pesado computacional sobre los hombros de la tecnología y, al mismo tiempo, aprovecha las aportaciones emocionales y cognitivas de los seres humanos, permitiendo la expansión de procesos e información a una mayor escala sin comprometer la integridad de los datos o aumentar el gasto en recursos humanos.
- **Software y creación de productos más exitosos:** al aplicar los principios de la ciencia del comportamiento a la tecnología a través de la Inteligencia Artificial centrada en los Humanos, los desarrolladores y diseñadores de productos pueden aprovechar el comportamiento del usuario y los patrones subconscientes para crear productos y servicios más satisfactorios, informados y enriquecedores.

#### 1.4.- Inteligencia Artificial explicable

Como hemos mencionado anteriormente en la introducción, la demanda de una explicación acerca de las decisiones tomadas por un sistema de Inteligencia Artificial resulta cada vez mayor entre los usuarios de este tipo de sistemas.

Pueden distinguirse cuatro grupos de individuos interesados en explicaciones por parte de un sistema de Inteligencia Artificial (como puede verse en la Figura 1), cada uno con diferentes motivaciones [4]:

- **Creadores de sistemas de inteligencia artificial:** los individuos técnicos (científicos de datos y desarrolladores) que implementan un sistema de Inteligencia Artificial desean conocer si su sistema está trabajando según lo previsto, cómo diagnosticar y mejorarlo, y posiblemente, adquirir una percepción acerca de sus decisiones.
- **Tomadores de decisiones del usuario final:** las personas que utilizan las recomendaciones proporcionadas por un sistema de Inteligencia Artificial desean explicaciones que puedan aumentar su confianza en las recomendaciones del sistema y posiblemente proporcionar una visión adicional para mejorar sus futuras decisiones y su comprensión del fenómeno analizado.
- **Cuerpos reguladores:** las agencias gubernamentales, encargadas de proteger los derechos de sus ciudadanos, quieren asegurar que las decisiones se toman de manera segura y justa, de forma que la sociedad no se vea afectada negativamente por las decisiones.
- **Consumidores finales:** las personas impactadas por las recomendaciones realizadas por un sistema de Inteligencia Artificial desean explicaciones que puedan ayudar a comprender si fueron tratados de una manera justa, y qué factores podrían modificarse para obtener un resultado diferente.

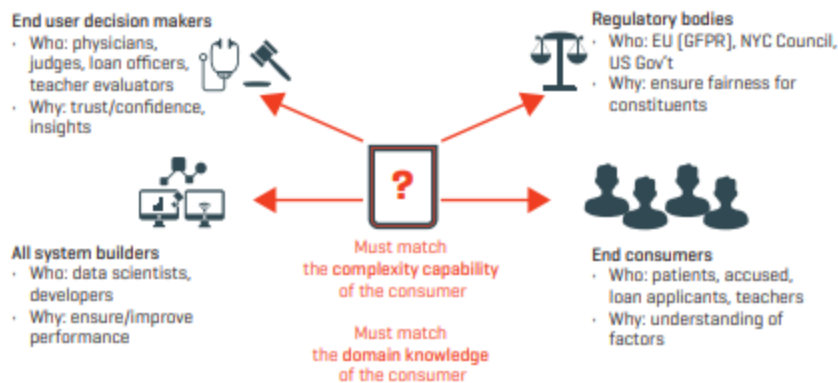


Figura 1: Grupos interesados en la Inteligencia Artificial explicable

Nota. Figura adaptada de “Explaining explainable AI” (p. 18), por M. Hind, 2019, *XRDS: Crossroads, The ACM Magazine for Students*, 25(3).

La Inteligencia Artificial explicable podría clasificarse de diferentes maneras [4]:

- Según la interpretación, podemos distinguir entre IA directamente interpretable (lo suficientemente simple para el entendimiento por parte de los usuarios) o con una interpretación “post hoc” (comienza con un modelo de caja negra examinado con un modelo acompañante para crear interpretaciones, de manera que el modelo de caja negra proporciona la predicción mientras que la interpretación mejora las interacciones humanas).
- Según la explicación, podemos distinguir entre IA global (proporciona al usuario el modelo predictivo completo para contribuir en su entendimiento) o local (proporciona las explicaciones asociadas a predicciones individuales).
- Según el nivel de interactividad, podemos distinguir entre IA estática (la explicación es simplemente presentada al usuario) o interactiva (el usuario puede interactuar con la explicación mediante la analítica visual o agentes de conversación).

La propuesta de sistema de Inteligencia Artificial que queremos desarrollar e implementar puede ser considerada local, según la explicación proporcionada al usuario del sistema, interactiva, según el nivel de interactividad, y con una interpretación “post hoc”, como se verá más adelante.

## 1.5.- Revisión de artículos científicos similares

Previamente al desarrollo e implementación de nuestra inteligencia artificial, se ha realizado la revisión de diferentes artículos científicos con características similares al problema que se trata de resolver.

En primer lugar, se realizó la revisión de artículos científicos centrados en la predicción de concesiones de hipotecas. Sin embargo, la mayoría de artículos encontrados buscaban realizar una predicción acerca de la viabilidad de una hipoteca ya concedida al solicitante del préstamo, tomando como ejemplo un estudio en el cual se utilizaban diferentes algoritmos de clasificación para realizar una predicción acerca de si una hipoteca ya concedida será abonada en su totalidad o resultará defectuosa [5]. La intención de nuestra inteligencia artificial será efectuar una predicción acerca de la concesión o no de una hipoteca una vez el solicitante de la misma ha realizado una solicitud, pudiendo predecir acciones como, por ejemplo, si la institución que ha recibido dicha solicitud terminará vendiendo la hipoteca a otra institución, o si el solicitante terminará retirando su solicitud, aparte de generar las explicaciones pertinentes acerca de dicha decisión, tal y como se explicará más adelante.

En segundo lugar, se realizó la revisión de artículos científicos en los cuales se proporcionan propuestas de Inteligencia Artificial explicable, tomando como ejemplo un artículo en el cual se propone la aplicación de Inteligencia Artificial explicable para la detección de personas con depresión en función del consumo de determinados alimentos en sus comidas mediante el uso de redes neuronales [6]. Entre las técnicas propuestas para la generación de explicaciones acerca de las decisiones tomadas, se encuentra el recorrido de la estructura de una red neuronal para encontrar la entrada más relevante a la hora de realizar la predicción, o el cálculo de la diferencia de frecuencias, dado un determinado valor para una determinada entrada, entre el número de ocasiones en las cuales se ha originado un valor de salida u otro en el modelo aprendido por la red neuronal. La intención de nuestra inteligencia artificial será utilizar diferentes algoritmos de clasificación a partir de los cuales se podrán generar explicaciones acerca de las predicciones realizadas [incluyendo, por ejemplo, máquinas soporte vector (SVM, *Support Vector Machines*) o el algoritmo K-means], además de generar una serie de gráficas que acompañen dichas explicaciones, como se explicará más adelante.

# CAPÍTULO 2

## Especificación de Requisitos Software

En este segundo capítulo, se presenta una Especificación de Requisitos Software (cuyas siglas en inglés serían SRS, *Software Requirements Specification*) que fue realizada en primer lugar durante el desarrollo de este proyecto con el objetivo de definir los prerrequisitos a tener en cuenta a la hora de llevar a cabo la implementación del proyecto. Más adelante, se indicarán más detalles acerca de la implementación del mismo.

### 2.1.- Introducción

#### 2.1.1.- Propósito

El propósito de este capítulo es presentar una descripción general y una explicación de la funcionalidad de una inteligencia artificial centrada en los humanos para la toma de decisiones financieras.

El capítulo está destinado para los clientes del sistema, así como los desarrolladores y gestores del mismo.

#### 2.1.2.- Alcance

El objetivo de esta inteligencia artificial es tomar la decisión financiera consistente en la concesión de una hipoteca.

Para la toma de dicha decisión, este sistema partirá de un conjunto de datos o *dataset*, que será almacenado en una base de datos y será utilizado como un conjunto de datos de entrenamiento con la finalidad de entrenar la inteligencia artificial en cuestión. Los datos pertenecientes al *dataset* se encuentran divididos en una serie de campos que definen las características en las cuales se basará la inteligencia artificial para la concesión o no de una hipoteca.

El sistema dispondrá de una interfaz gráfica que permitirá al usuario introducir una serie de datos, los cuales circularán a través de un conjunto de algoritmos que determinarán la concesión o no de una hipoteca, mostrando una serie de gráficas que reflejarán el peso de cada dato introducido a la hora de tomar la decisión financiera, al igual que una explicación autogenerada que justificará dicha decisión.

#### 2.1.3.- Definiciones, acrónimos y abreviaturas

- **Cliente:** Persona(s) que solicita y financia el desarrollo del producto y, por lo general, decide los requisitos del sistema. En el contexto de este proyecto, el cliente y el desarrollador serán la misma entidad.
- **CSV:** Comma-Separated Values (Valores Separados por Comas).
- **Desarrollador:** Persona(s) que desarrolla un producto software para un cliente. En el contexto de este proyecto, el cliente y el desarrollador serán la misma entidad.

- **Gestor:** Persona(s) que utilizará el sistema para trabajar sobre él en la aplicación que sea conveniente. En el contexto de este proyecto, el gestor será la misma entidad que el cliente y los desarrolladores.
- **Python:** Lenguaje de programación utilizado para desarrollar el producto.
- **Usuario:** Persona(s) que interactuará con el producto. En el contexto de este proyecto, el usuario será la misma entidad que el cliente.

## 2.2.- Descripción general

### 2.2.1.- Perspectiva del producto

Será posible acceder al producto a través de una interfaz gráfica destinada a interactuar con el usuario. Únicamente se podrá acceder a la inteligencia artificial en forma de usuario, que en el contexto de este proyecto será el individuo solicitante de una hipoteca. Con la finalidad de averiguar la concesión o no de dicha hipoteca, el usuario podrá interactuar con la interfaz gráfica introduciendo una serie de datos que influirán en mayor o menor medida en la toma de la decisión financiera.

Como el uso de esta inteligencia artificial es concreto y enfocado a un único usuario, no resulta necesaria la implementación de un sistema de usuarios con identificación y contraseña.

El objetivo es que el usuario sea capaz de introducir un conjunto de información cuyo formato coincide con los campos en los cuales se divide el *dataset* utilizado para entrenar la inteligencia artificial encargada de decidir la concesión de una hipoteca.

El *dataset* será almacenado en una base de datos tras haber sido divididos en datos de entrenamiento (para entrenar la inteligencia artificial) y datos de validación (para ajustar los hiperparámetros de los diferentes algoritmos que integran la inteligencia artificial), almacenando los datos utilizados para entrenar la inteligencia artificial. Por tanto, se puede intuir que los datos introducidos por el usuario serán utilizados como datos de test para comparar los resultados proporcionados por los distintos algoritmos. El formato de la base de datos coincidirá con el formato del *dataset* anteriormente mencionado. El usuario nunca llamará a una función de carga o almacenamiento, se encargará el sistema única y exclusivamente cuando el programador lo considere adecuado.

Como se ha mencionado con anterioridad, el usuario podrá visualizar el peso o relevancia de cada dato introducido para tomar la decisión financiera a través de una serie de gráficas que analizarán los resultados de los distintos algoritmos que componen la inteligencia artificial, al igual que una explicación autogenerada que justificará dicha decisión.

### 2.2.2.- Funcionalidades del producto

El usuario podrá hacer uso de la inteligencia artificial mediante las siguientes funcionalidades. Otras funciones, como las de carga y almacenamiento, no son ejecutables por el mismo, sino por el propio sistema.

#### Introducción de datos de test

- **Descripción:** El usuario podrá introducir una serie de datos de test, los cuales serán utilizados como datos de entrada para los diferentes algoritmos que integran la inteligencia artificial, encargados de decidir la concesión o no de una hipoteca.
- **Prerrequisitos:** Los datos introducidos serán similares con respecto a la información almacenada previamente en la base de datos, situados dentro de un rango de valores.

### **Toma de decisión financiera: Concesión de una hipoteca**

- **Descripción:** Los datos introducidos previamente por el usuario circularán a través de una serie de algoritmos que tendrán la finalidad de tomar la decisión financiera consistente en la concesión de una hipoteca.
- **Prerrequisitos:** El usuario ha de haber introducido cierta información como entrada para la inteligencia artificial.

### **Representación de los resultados**

- **Descripción:** La inteligencia artificial mostrará al usuario los resultados del procesamiento realizado por los algoritmos que la integran.

### **2.2.3.- Características del usuario**

Como se ha mencionado anteriormente, el usuario que tenga acceso al sistema lo hará en forma de individuo solicitante de una hipoteca, al cual se le darán ciertas indicaciones acerca de cómo utilizar la inteligencia artificial. Tendrá la posibilidad de ejecutar las funciones detalladas en el apartado anterior (apartado 2.2.2).

### **2.2.4.- Restricciones**

- Existirá un límite de registros almacenados en la base de datos que contiene el *dataset* utilizado para entrenar la inteligencia artificial encargada de decidir la concesión de una hipoteca.
- El número máximo de datos introducidos por el usuario coincide con el número de campos en los cuales se dividen los datos pertenecientes al *dataset* anteriormente mencionado.

### **2.2.5.- Supuestos y dependencias**

En caso de que resulte inviable realizar el proyecto con una base de datos, se utilizarán ficheros con formato CSV que contendrán los mismos registros incluidos en la base de datos.

Para poder realizar una correcta implementación de esta funcionalidad, el sistema dispondrá de la siguiente función:

#### **Carga de *dataset* de entrenamiento**

- **Descripción:** La inteligencia artificial cargará desde un fichero con formato CSV el *dataset* utilizado para entrenar los algoritmos que integran dicha inteligencia artificial, encargada de decidir la concesión de una hipoteca.
- **Prerrequisitos:** El sistema ha de localizar el fichero CSV en el cual se encuentra almacenado el *dataset* en cuestión.

### **2.2.6.- Requisitos futuros**

Se considerará la posibilidad de aumentar el número de registros de entrenamiento y de validación almacenados en la base de datos, lo cual permitiría dotar a la inteligencia artificial de una mayor precisión a la hora de tomar la decisión financiera. Sin embargo, en la codificación

actual, estos límites no pueden verse modificados al haber sido definidos como constantes del sistema.

## 2.3.- Requisitos específicos

### 2.3.1.- Interfaces externos

El usuario interactuará con la inteligencia artificial a través de una interfaz gráfica. Un usuario accederá al producto desde un ordenador que tenga instalado un entorno de programación en lenguaje Python y un programa que sea capaz de lanzar aplicaciones de este tipo (por ejemplo, haciendo uso del entorno Jupyter Notebook, proporcionado por Anaconda).

### 2.3.2.- Funcionalidades

Para toda función, el usuario es el único que podrá utilizarla (o el propio sistema, en el caso de tratarse de la carga o el almacenamiento del *dataset* utilizado por la inteligencia artificial).

#### Inicio de la aplicación

- **Resumen:** Lanzamiento de la interfaz gráfica de la inteligencia artificial.
- **Precondición:** El usuario ha de tener acceso a la aplicación y la carga del *dataset* utilizado por la inteligencia artificial ha de haber resultado satisfactoria.
- **Descripción:** A través de dicha interfaz gráfica, el usuario interactuará con la inteligencia artificial introduciendo una serie de datos que circularán a través de los diferentes algoritmos que la integran con la finalidad de determinar la concesión o no de una hipoteca.
- **Excepciones:** Si se produce un error en la carga del *dataset* anteriormente mencionado, el sistema se encargará de detectar el error y reportarlo.
- **Resultado:** El usuario podrá interactuar con la interfaz gráfica y realizar una acción sobre la inteligencia artificial.

#### Carga de *dataset* de entrenamiento

- **Resumen:** Carga del *dataset* utilizado por la inteligencia artificial desde un fichero con formato CSV.
- **Precondición:** El fichero CSV ha de encontrarse en la dirección indicada en el programa.
- **Descripción:** El sistema cargará desde el fichero con formato CSV un *dataset* que incluirá una serie de datos empleados para entrenar la inteligencia artificial (datos de entrenamiento) y ajustar los hiperparámetros de los diferentes algoritmos existentes dentro de la misma (datos de validación).
- **Excepciones:** El fichero CSV puede ser localizado, pero su formato puede ser erróneo. En tal caso, el sistema detectará y reportará el error.
- **Resultado:** Quedará registrado en el sistema el *dataset* anteriormente mencionado.

#### Almacenamiento de *dataset* de entrenamiento

- **Resumen:** Almacenamiento del *dataset* utilizado por la inteligencia artificial en una base de datos.



- **Precondición:** El *dataset* en cuestión ha de haber quedado previamente registrado en el sistema.
- **Descripción:** El sistema almacenará en la base de datos un *dataset* que incluirá una serie de datos empleados para entrenar la inteligencia artificial (datos de entrenamiento).
- **Excepciones:** Si se produce un error durante el proceso de almacenamiento, el sistema se encargará de detectar el error y reportarlo.
- **Resultado:** Quedará registrado en la base de datos utilizada por el sistema el *dataset* anteriormente mencionado.

### Introducción de datos de test

- **Resumen:** Introducción de datos de test en la inteligencia artificial por parte del usuario.
- **Precondición:** El usuario ha de introducir una serie de datos situados dentro de determinados rangos de valores.
- **Descripción:** Los datos introducidos por el usuario serán utilizados como datos de entrada para los diferentes algoritmos que integran la inteligencia artificial, con el objetivo de determinar la concesión o no de una hipoteca.
- **Excepciones:** Si algún dato introducido por el usuario resulta ser considerado no válido, el sistema se encargará de detectar el error y reportarlo.
- **Resultado:** Los nuevos datos de test serán utilizados por la inteligencia artificial para decidir la concesión de una hipoteca.

### Toma de decisión financiera: Concesión de una hipoteca

- **Resumen:** Toma de decisión acerca de la concesión de una hipoteca por parte de la inteligencia artificial.
- **Precondición:** El usuario de la inteligencia artificial ha de haber introducido previamente cierta información.
- **Descripción:** La inteligencia artificial tomará la decisión financiera consistente en la concesión de una hipoteca a partir de una serie de datos de test introducidos previamente por el usuario mediante una interfaz gráfica.
- **Excepciones:** Si se produce un error durante el procesamiento de la información introducida por el usuario, el sistema reportará el error detectado.
- **Resultado:** La inteligencia artificial mostrará los resultados del procesamiento realizado por medio de una serie de gráficas que ilustrarán la relevancia de la información introducida por el usuario a la hora de tomar la decisión financiera, al igual que una explicación autogenerada que justificará dicha decisión.

### Representación de los resultados

- **Resumen:** Visualización de los resultados del procesamiento realizado por la inteligencia artificial.
- **Precondición:** El usuario ha de haber realizado una consulta a la inteligencia artificial.
- **Descripción:** La inteligencia artificial explicará la decisión adoptada acerca de la concesión de una hipoteca a través de una serie de gráficas que reflejarán el peso de cada dato introducido previamente por el usuario a la hora de tomar la decisión financiera,

junto con una explicación autogenerada que justifique la decisión tomada por cada uno de los algoritmos que integran la inteligencia artificial.

- **Excepciones:** Ninguna, la inteligencia artificial se encargará de que toda la información esté completa.
- **Resultado:** El usuario visualizará los resultados de la consulta realizada a la inteligencia artificial.

### 2.3.3.- Requisitos de rendimiento

La carga del *dataset* utilizado por la inteligencia artificial ha de haber sido satisfactoria para una buena funcionalidad de la misma. En caso de que se hayan producido errores durante el proceso, la aplicación no podrá ejecutarse.

A continuación, se definirán las constantes del sistema:

- **Tamaño de la base de datos:** 100 registros.
- **Tamaño del *dataset* (datos de entrenamiento):** 73 registros.

Según se estipula en el apartado 2.2.6, estas constantes podrán ser modificadas en un futuro, pero no en esta versión del sistema.

### 2.3.4.- Restricciones de diseño

El sistema será ejecutado desde la consola proporcionada por el entorno de programación en lenguaje Python, así como la codificación también se ha realizado desde el mismo.

El funcionamiento de la base de datos se describirá más adelante en esta memoria.

### 2.3.5.- Atributos del sistema software

#### Fiabilidad

La inteligencia artificial está codificada de tal manera que será capaz de trabajar con casi cualquier entrada que el usuario introduzca de manera errónea, y emitirá mensajes de error cada vez que ocurra un hecho de tal naturaleza.

A su vez, se ha trabajado para que el sistema no caiga si se quiere acceder a una base de datos completa y demás posibles errores en tiempo de ejecución.

#### Disponibilidad

En caso de hacer uso de una base de datos, si llegara a caer, siempre se podrá acceder al *backup* más actualizado.

En caso de hacer uso de un fichero con formato CSV, éste será incluido siempre en cada entrega del producto.

#### Seguridad

Dado que el producto no funciona con un sistema de usuarios, toda persona con acceso a la inteligencia artificial tendrá la posibilidad de interactuar con la totalidad de la funcionalidad del mismo.

## **Mantenibilidad**

La base de datos será mantenida por los administradores, así como el sistema entero.

## **Portabilidad**

El sistema está desarrollado con la finalidad de que pueda ser ejecutado en cualquier equipo que permita la ejecución de programas Python, obviando su sistema operativo.

Para un correcto funcionamiento, toda vez que sea ejecutada, la inteligencia artificial ha de estar acompañada por la base de datos o el fichero con formato CSV que contenga la información necesaria para la funcionalidad de la misma. En caso contrario, la funcionalidad del sistema será nula.

# CAPÍTULO 3

## Planificación

En este tercer capítulo, se presentan detalles acerca de la planificación seguida durante el desarrollo de este proyecto.

### 3.1.- Modelo de proceso

El modelo de proceso utilizado es el modelo clásico en cascada.

En primer lugar, se realizó una conceptualización del sistema, diferenciando las partes más visibles. Se realizó una Especificación de Requisitos Software (SRS) para determinar los requisitos funcionales y de rendimiento, realizando con ello un diseño claro y conciso del sistema.

Posteriormente, se procedió a la implementación de dicho diseño y, una vez finalizada dicha fase, se realizaron las pruebas pertinentes para comprobar su funcionamiento.

### 3.2.- Estimaciones del proyecto

#### 3.2.1.- Datos históricos

El esfuerzo será calculado en horas de trabajo.

#### 3.2.2.- Técnicas de estimación

La técnica de estimación utilizada es la técnica por descomposición.

#### 3.2.3.- Estimaciones de esfuerzo

En la Tabla 1, se recogen las estimaciones de esfuerzo de este proyecto medidas en horas de trabajo.

Función	Diseño	Codificación		Prueba
		Análisis	Implementación	
Introducción de datos de test	1.0 3 horas	1.1 3 horas	1.2 16 horas	1.3 4 horas
Toma de decisión financiera: Concesión de una hipoteca	2.0 4 horas	2.1 4 horas	2.2 24 horas	2.3 4 horas
Representación de los resultados	3.0 4 horas	3.1 4 horas	3.2 24 horas	3.3 4 horas

Tabla 1: Estimación del esfuerzo en horas de trabajo

**Total:** 98 horas de trabajo.

### 3.3.- Planificación temporal

#### 3.3.1.- Estructura de descomposición del trabajo

En la Tabla 2, se recoge la estructura de descomposición del trabajo asociado a este proyecto.

Función	Diseño	Codificación		Prueba
		Análisis	Implementación	
<b>Introducción de datos de test</b>	1.0 I: 9-4-2020 F: 9-4-2020 R: Miguel Sierra Esteso T: Diseño "Introducción de datos de test"	1.1 I: 13-4-2020 F: 15-4-2020 R: Miguel Sierra Esteso T: Definición del estilo de programación	1.2 I: 24-4-2020 F: 29-4-2020 R: Miguel Sierra Esteso T: Codificación "Introducción de datos de test"	1.3 I: 12-5-2020 F: 13-5-2020 R: Miguel Sierra Esteso T: Prueba "Introducción de datos de test"
<b>Toma de decisión financiera: Concesión de una hipoteca</b>	2.0 I: 8-4-2020 F: 8-4-2020 R: Miguel Sierra Esteso T: Diseño "Toma de decisión financiera: Concesión de una hipoteca"	2.1 I: 13-4-2020 F: 15-4-2020 R: Miguel Sierra Esteso T: Definición del estilo de programación	2.2 I: 16-4-2020 F: 23-4-2020 R: Miguel Sierra Esteso T: Codificación "Toma de decisión financiera: Concesión de una hipoteca"	2.3 I: 8-5-2020 F: 11-5-2020 R: Miguel Sierra Esteso T: Prueba "Toma de decisión financiera: Concesión de una hipoteca"
<b>Representación de los resultados</b>	3.0 I: 10-4-2020 F: 10-4-2020 R: Miguel Sierra Esteso T: Diseño "Representación de los resultados"	3.1 I: 13-4-2020 F: 15-4-2020 R: Miguel Sierra Esteso T: Definición del estilo de programación	3.2 I: 30-4-2020 F: 7-5-2020 R: Miguel Sierra Esteso T: Codificación "Representación de los resultados"	3.3 I: 14-5-2020 F: 15-5-2020 R: Miguel Sierra Esteso T: Prueba "Representación de los resultados"

Tabla 2: Estructura de descomposición del trabajo

- **I:** Fecha de inicio.
- **F:** Fecha de finalización.
- **R:** Recursos (en este caso, el desarrollador).
- **T:** Tarea a realizar.

#### 3.3.2.- Gráfico Gantt

En la Figura 2, se puede observar el gráfico de Gantt (elaborado con Microsoft Project) que refleja la planificación seguida durante el desarrollo de este proyecto.

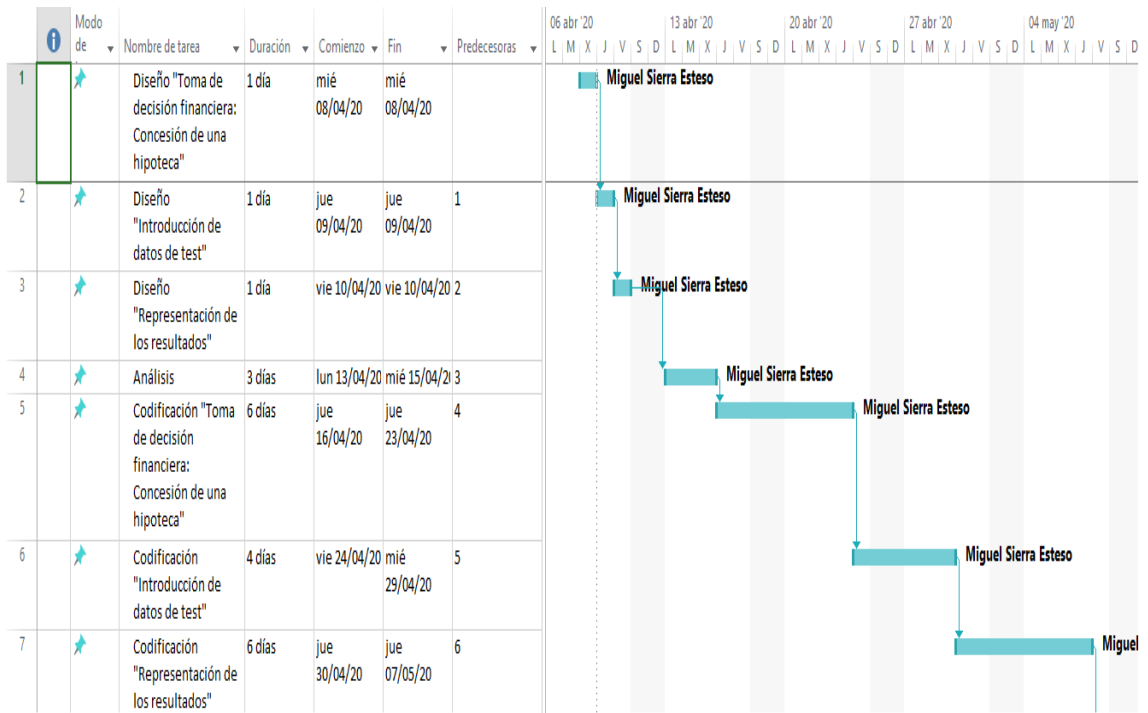


Figura 2: Gráfico de Gantt (elaborado con Microsoft Project)

### 3.3.3.- Red de tareas

En la Figura 3, se puede observar la red de tareas realizadas durante el desarrollo de este proyecto.












	 Modo de	Nombre de tarea	Trabajo	Duración	Comienzo	Fin
1		▲ Diseño "Toma de dec <i>Miguel Sierra Este</i>	8 horas	1 día	mié 08/04/20	mié 08/04/20
2		▲ Diseño "Introducción <i>Miguel Sierra Este</i>	8 horas	1 día	jue 09/04/20	jue 09/04/20
3		▲ Diseño "Representac <i>Miguel Sierra Este</i>	8 horas	1 día	vie 10/04/20	vie 10/04/20
4		▲ Análisis <i>Miguel Sierra Este</i>	24 horas	3 días	lun 13/04/20	mié 15/04/20
5		▲ Codificación "Toma d <i>Miguel Sierra Este</i>	48 horas	6 días	jue 16/04/20	jue 23/04/20
6		▲ Codificación "Introdu <i>Miguel Sierra Este</i>	32 horas	4 días	vie 24/04/20	mié 29/04/20
7		▲ Codificación "Repres <i>Miguel Sierra Este</i>	48 horas	6 días	jue 30/04/20	jue 07/05/20
8		▲ Prueba "Toma de dec <i>Miguel Sierra Este</i>	16 horas	2 días	vie 08/05/20	lun 11/05/20
9		▲ Prueba "Introducciór <i>Miguel Sierra Este</i>	16 horas	2 días	mar 12/05/20	mié 13/05/20
10		▲ Prueba "Representac <i>Miguel Sierra Este</i>	16 horas	2 días	jue 14/05/20	vie 15/05/20

Figura 3: Red de tareas

# CAPÍTULO 4

## Diagramas UML

En este cuarto capítulo, se muestran una serie de diagramas UML (*Unified Modeling Language*) elaborados antes de realizar la implementación final de la inteligencia artificial con la herramienta BoUML, que permiten capturar una descripción del comportamiento de la misma.

### 4.1.- Diagrama de caso de uso

En primer lugar, observamos un diagrama de casos de uso, descripciones de un conjunto de secuencias de acciones que ejecuta el sistema para producir un resultado observable de valor para el usuario. Como se puede apreciar en el diagrama de la Figura 4, el usuario de la inteligencia artificial introducirá una serie de datos al sistema que serán utilizados para que un conjunto de algoritmos de clasificación realicen una predicción acerca de la decisión financiera consistente en la concesión de una hipoteca, mostrando posteriormente una explicación autogenerada por el sistema que justifique la decisión tomada por los algoritmos junto con una serie de gráficas que mostrarán los pesos de los datos introducidos por el usuario a la hora de tomar la decisión financiera.

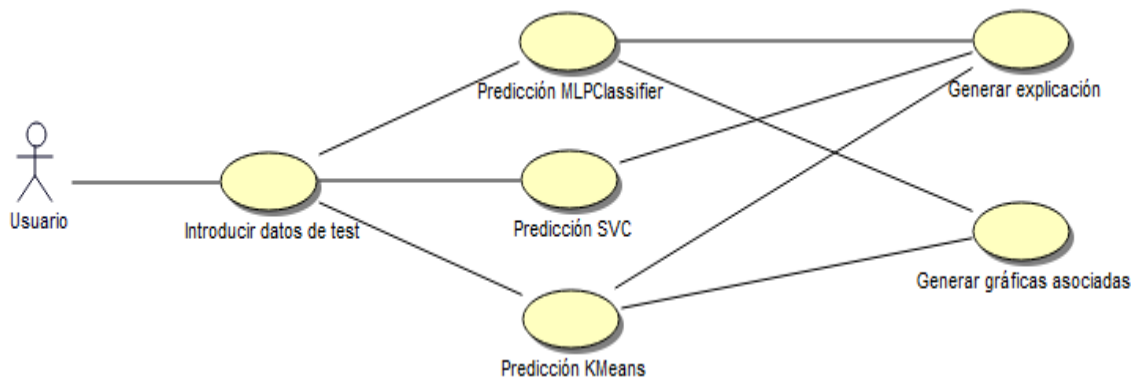


Figura 4: Diagrama de caso de uso (elaborado con BoUML)

### 4.2.- Diagrama de clases

En segundo lugar, observamos un diagrama de clases que muestra la estructura de una clase llamada **App**, la cual hemos utilizado para incluir toda la funcionalidad necesaria para un correcto funcionamiento de la inteligencia artificial [7], tal y como podemos apreciar en la Figura 5.

La estructura de la clase **App** incluye los siguientes atributos y métodos:

- **Atributos:**
  - **Master:** un objeto de la clase **Tk**, proporcionada por la librería **tkinter** de Python, que representa la ventana que incluirá todos los elementos pertenecientes a la interfaz gráfica de usuario de la inteligencia artificial.
  - **Features:** una lista con los nombres de los atributos del *dataset* utilizado para entrenar los algoritmos de clasificación que integran la inteligencia artificial.
  - **SensibleFeatures:** una lista con los nombres de los atributos del *dataset* anteriormente mencionado que tendrán asociada una explicación diferente con



respecto al resto de atributos en el caso de que resulten ser el atributo más relevante a la hora de tomar la decisión financiera consistente en la concesión o no de una hipoteca (en este caso, el origen étnico, la raza y el sexo del solicitante o co-solicitante de dicha hipoteca).

- **ActionsTaken:** una lista con los nombres de las acciones asociadas a cada posible predicción realizada por los algoritmos que integran la inteligencia.
- **UserActions:** una lista utilizada a la hora de indicar al usuario de la inteligencia artificial el atributo más relevante para decidir la concesión o no de una hipoteca.
- **Values:** un diccionario que incluye el significado asociado a todos los posibles valores de los atributos del *dataset* utilizado para entrenar la inteligencia artificial (excepto aquellos atributos expresados en miles de dólares, como se explicará más adelante).
- **Mlp:** un objeto de la clase **MLPClassifier** [8], proporcionada por la librería **sklearn.neural\_network** de Python, que representa una red neuronal utilizada para realizar predicciones acerca de la concesión o no de una hipoteca.
- **Svc:** un objeto de la clase **SVC** [9], proporcionada por la librería **sklearn.svm** de Python, que representa un algoritmo SVM (*Support Vector Machine*) utilizado para realizar predicciones acerca de la concesión o no de una hipoteca.
- **Kmeans:** un objeto de la clase **KMeans** [10], proporcionada por la librería **sklearn.cluster** de Python, que representa un algoritmo K-means utilizado para realizar predicciones acerca de la concesión o no de una hipoteca.
- **Cursor:** un objeto de la clase **Cursor**, perteneciente a la librería **sqlite3** de Python, utilizado para realizar operaciones sobre una base de datos SQLite.
- Objetos que representan *widgets* asociados a cada atributo introducido por el usuario para realizar una predicción acerca de la concesión o no de una hipoteca.
- Un objeto de la clase **ScrolledText** que representa un área de texto en la cual aparecerán las explicaciones asociadas a las predicciones efectuadas por los algoritmos que integran la inteligencia artificial.
- Dos objetos de la clase **Tk** que representan las ventanas que incluirán las gráficas que complementarán las explicaciones acerca de las decisiones realizadas por los algoritmos que integran la inteligencia artificial.
- Dos objetos de la clase **FigureCanvasTkAgg** que representan las gráficas anteriormente mencionadas.
- **Métodos:**
  - **\_\_init\_\_():** método constructor invocado en el momento de la creación de nuevos objetos asociados a la clase.
  - **\_reset():** método invocado cuando el usuario de la inteligencia artificial desea resetear la aplicación para realizar una nueva predicción.
  - **\_quit():** método invocado cuando el usuario de la inteligencia artificial desea salir de la aplicación.
  - **makeDecision():** método invocado cuando el usuario de la inteligencia artificial ha terminado de introducir los datos solicitados por la misma para realizar una predicción acerca de la concesión o no de una hipoteca, generando las explicaciones pertinentes sobre dicha decisión.
  - **generateExplanation():** método utilizado para generar la explicación asociada a la decisión tomada por los diferentes algoritmos que integran la inteligencia artificial, acompañándola con una serie de gráficas que complementan dicha explicación.
  - **mlpExplanation():** método utilizado para generar la explicación asociada a la predicción realizada por la red neuronal anteriormente mencionada.

- **svcExplanation()**: método utilizado para generar la explicación asociada a la predicción realizada por el algoritmo SVM anteriormente mencionado.
- **kmeansExplanation()**: método utilizado para generar la explicación asociada a la predicción realizada por el algoritmo K-means anteriormente mencionado.
- **mostWeightedPathExplanation()**: método utilizado para generar una explicación acerca de una predicción realizada por la inteligencia artificial en función de la entrada del usuario que ha resultado más relevante a la hora de efectuar dicha predicción.
- **frequencyExplanation()**: método utilizado para generar una explicación acerca de una predicción realizada por la inteligencia artificial tras haber realizado una serie de consultas en una base de datos SQLite para calcular la diferencia de frecuencia, dado el valor de un determinado atributo, entre el número de veces que la salida resultante es igual a la salida proporcionada por la inteligencia artificial con ese valor de atributo, y el número de veces que la salida resultante es diferente.
- **mostWeightedPath()**: método utilizado para encontrar la entrada más relevante en la predicción realizada por la red neuronal anteriormente mencionada, recorriendo la estructura de dicha red neuronal desde la neurona correspondiente al valor de salida generado y buscando el camino más relevante hasta encontrar el atributo buscado.
- **mostWeightedPathFirstDendrite()**: método utilizado para encontrar la primera dendrita del camino más relevante dentro de la estructura de la red neuronal anteriormente mencionada.
- **percentage()**: método utilizado para calcular el porcentaje de registros con un valor de salida asociado que tienen un determinado valor para un determinado atributo.
- **diffPerccen()**: método utilizado para calcular la diferencia de frecuencia, dado el valor de un determinado atributo, entre el número de veces que la salida resultante es igual a la salida proporcionada por la inteligencia artificial con ese valor de atributo, y el número de veces que la salida resultante es diferente.
- **completeExplanation()**: método utilizado para completar una explicación acerca de la decisión tomada por los algoritmos que integran la inteligencia artificial con el valor del atributo considerado más relevante a la hora de tomar dicha decisión.
- **createPlots()**: método utilizado para crear las gráficas que complementarán las explicaciones acerca de las decisiones tomadas por los algoritmos que integran la inteligencia artificial.
- **mlpPlot()**: método utilizado para crear una gráfica acerca de la predicción realizada por la red neuronal anteriormente mencionada, mostrando los pesos de cada atributo introducido por el usuario a la hora de decidir la concesión o no de una hipoteca.
- **kmeansPlot()**: método utilizado para crear una gráfica acerca de la predicción realizada por el algoritmo K-means anteriormente mencionado, mostrando las coordenadas del centro del *cluster* asociado al valor de salida generado por dicho algoritmo.

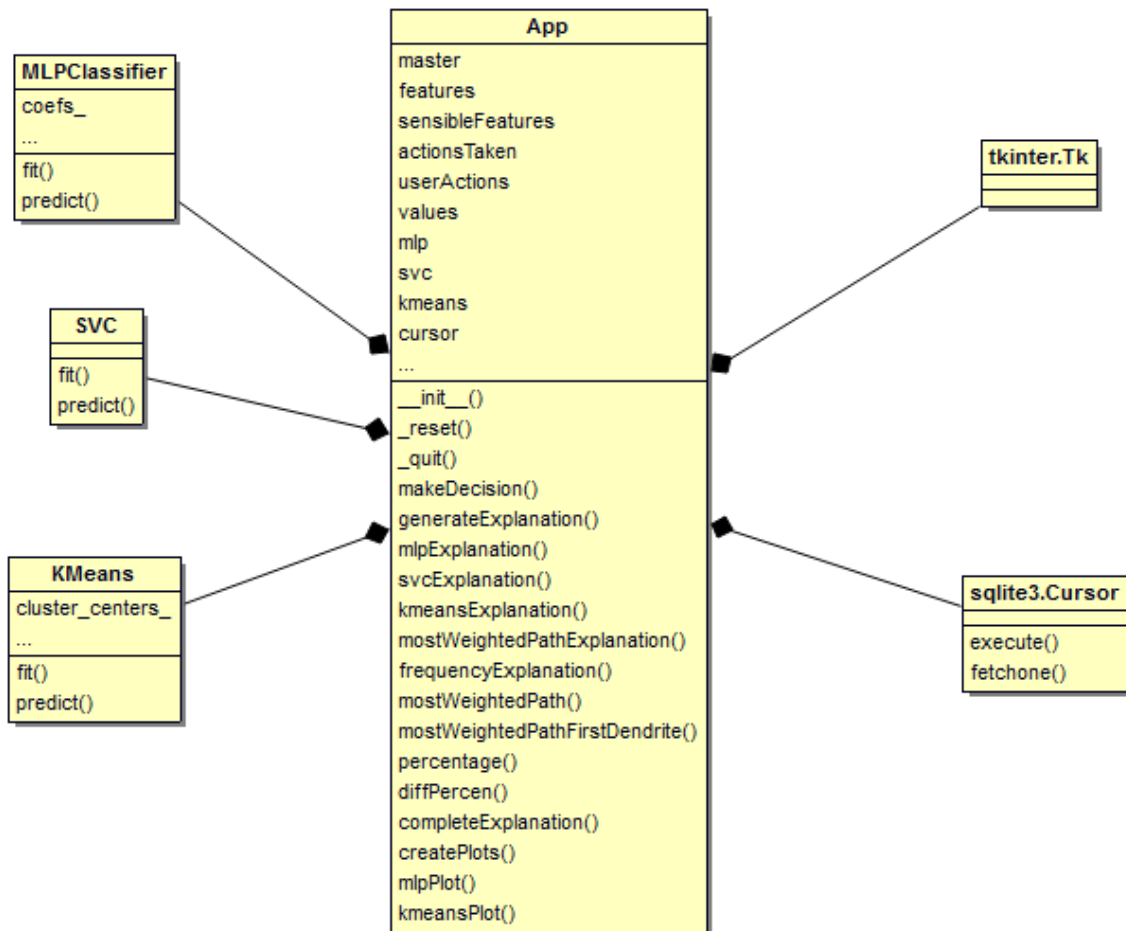


Figura 5: Diagrama de clases (elaborado con BoUML)

### 4.3.- Diagrama de secuencias

En tercer lugar, observamos un diagrama de secuencia, que muestra las interacciones entre objetos organizadas en una secuencia temporal. En particular, podemos apreciar en la Figura 6 que el usuario de la inteligencia artificial interactuará con un objeto correspondiente a la clase **App** anteriormente mencionada, que incluirá toda la funcionalidad requerida para procesar los datos introducidos por el usuario para tomar la decisión financiera correspondiente. Dichos datos pasarán por cada uno de los algoritmos que integran la inteligencia artificial para realizar la predicción solicitada y generar las explicaciones pertinentes acerca de dicha decisión.

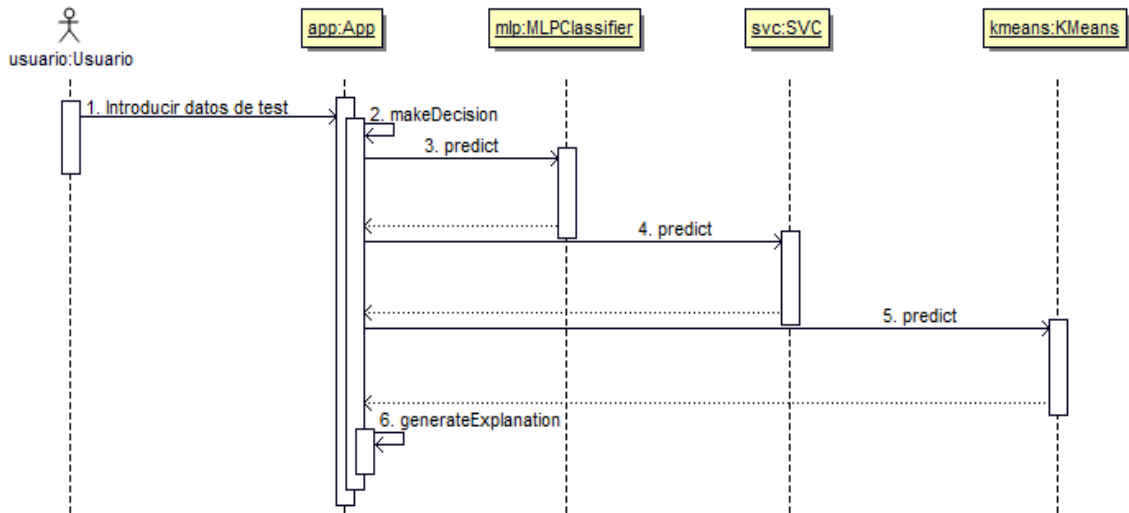


Figura 6: Diagrama de secuencias (elaborado con BoUML)

# CAPÍTULO 5

## Implementación del sistema

En este quinto capítulo, se realiza una descripción detallada de los aspectos más importantes a conocer acerca de la implementación final de la inteligencia artificial, y se incluye un ejemplo de uso de la misma.

### 5.1.- Dataset

#### 5.1.1.- Exploración de los datos

El conjunto de datos (o *dataset*) empleado para el desarrollo de esta inteligencia artificial recibe el nombre de “Home Mortgage Disclosure Act Data for the years 2007-2014” [11]. Home Mortgage Disclosure Act (HMDA) es una ley aprobada en Estados Unidos en el año 1975 que requiere a ciertas instituciones financieras proveer al público datos sobre hipotecas que pueden ayudar a determinar si las instituciones financieras están sirviendo las necesidades de vivienda de sus comunidades, ayudar a funcionarios públicos en la distribución de inversiones del sector público para atraer inversión privada a las áreas donde se necesita, y ayudar a identificar posibles patrones de préstamos discriminatorios [12]. Podemos examinar el *dataset* utilizando el método `head()` de la librería `pandas`. Este *dataset* contiene un total de 100 muestras, de las cuales hemos seleccionado los atributos que consideramos más importantes para nuestra inteligencia artificial [13] [14]:

- **Applicant\_ethnicity**: Origen étnico del solicitante de una hipoteca. Puede tener asignados valores enteros situados entre 1 y 4 que representan las siguientes posibles entradas:
  - “*Hispanic or Latino*”: origen étnico hispano o latino.
  - “*Not Hispanic or Latino*”: origen étnico no hispano o latino.
  - “*Information not provided by applicant in mail, Internet, or telephone application*”: información no proporcionada por el solicitante de la hipoteca.
  - “*Not applicable*”: no aplicable.
- **Applicant\_income\_000s**: Ingresos anuales del solicitante de una hipoteca (en miles de dólares). Puede tener asignados valores enteros.
- **Applicant\_race\_1**: Raza del solicitante de una hipoteca. Puede tener asignados valores enteros situados entre 1 y 7 que representan las siguientes posibles entradas:
  - “*American Indian or Alaskan Native*”: india americana o nativo de Alaska.
  - “*Asian*”: asiática.
  - “*Black or African American*”: negra o afroamericana.
  - “*Native Hawaiian or Other Pacific Islander*”: nativo de Hawaii u otra isla del Pacífico.
  - “*White*”: blanca.
  - “*Information not provided by applicant in mail, Internet, or telephone application*”: información no proporcionada por el solicitante de la hipoteca.
  - “*Not applicable*”: no aplicable.
- **Applicant\_sex**: Sexo del solicitante de una hipoteca. Puede tener asignados valores enteros situados entre 1 y 4 que representan las siguientes posibles entradas:
  - “*Male*”: masculino.
  - “*Female*”: femenino.

- “*Information not provided by applicant in mail, Internet, or telephone application*”: información no proporcionada por el solicitante de la hipoteca.
  - “*Not applicable*”: no aplicable.
- **Co\_applicant\_ethnicity**: Origen étnico del co-solicitante de una hipoteca. Puede tener asignados valores enteros situados entre 1 y 5 (similar al caso del atributo “*applicant\_ethnicity*”, añadiendo como opción 5 la entrada “*No co-applicant*”, indicando que no existe un co-solicitante).
- **Co\_applicant\_race\_1**: Raza del co-solicitante de una hipoteca. Puede tener asignados valores enteros situados entre 1 y 8 (similar al caso del atributo “*applicant\_race\_1*”, añadiendo como opción 8 la entrada “*No co-applicant*”, indicando que no existe un co-solicitante).
- **Co\_applicant\_sex**: Sexo del solicitante de una hipoteca. Puede tener asignados valores enteros situados entre 1 y 5 (similar al caso del atributo “*applicant\_sex*”, añadiendo como opción 5 la entrada “*No co-applicant*”, indicando que no existe un co-solicitante).
- **Lien\_status**: Estado del derecho de retención (reportado por los prestamistas para la recopilación de datos). Puede tener asignados valores enteros situados entre 1 y 4 que representan las siguientes posibles entradas:
  - “*Secured by a first lien*”: asegurado por un primer gravamen.
  - “*Secured by a subordinate lien*”: asegurado por un gravamen subordinado.
  - “*Not secured by a lien*”: no asegurado por un gravamen.
  - “*Not applicable*”: no aplicable.
- **Loan\_purpose**: Propósito del préstamo solicitado. Puede tener asignados valores enteros situados entre 1 y 3 que representan las siguientes posibles entradas:
  - “*Home purchase*”: compra de una vivienda.
  - “*Home improvement*”: mejora de una vivienda.
  - “*Refinancing*”: refinanciación.
- **Loan\_type**: Tipo de préstamo (convencional, garantizado por el gobierno, o asegurado por el gobierno). Puede tener asignados valores enteros situados entre 1 y 4 que representan las siguientes posibles entradas:
  - “*Conventional*”: convencional.
  - “*FHA-insured*”: asegurado por la FHA (*Federal Housing Administration*).
  - “*VA-guaranteed*”: garantizado por la VA (*Veterans Administration*).
  - “*FSA/RHS-guaranteed*”: garantizado por la FSA (*Farm Service Agency*) o el RHS (*Rural Housing Service*).
- **Owner\_occupancy**: Indica si la propiedad a la cual se relaciona el préstamo solicitado será la vivienda principal del propietario. Puede tener asignados valores enteros situados entre 1 y 3 que representan las siguientes posibles entradas:
  - “*Owner-occupied as a principal dwelling*”: propiedad ocupada por el propietario como vivienda principal.
  - “*Not owner-occupied*”: propiedad no ocupada por el propietario.
  - “*Not applicable*”: no aplicable.
- **Preapproval**: Indica si se solicita una aprobación previa. Puede tener asignados valores enteros situados entre 1 y 3 que representan las siguientes posibles entradas:
  - “*Preapproval was requested*”: se ha solicitado una aprobación previa.
  - “*Preapproval was not requested*”: no se ha solicitado una aprobación previa.
  - “*Not applicable*”: no aplicable.
- **Property\_type**: Tipo de la propiedad a la cual se relaciona el préstamo solicitado. Puede tener asignados valores enteros situados entre 1 y 3 que representan las siguientes posibles entradas:
  - “*One-to-four family dwelling (other than manufactured housing)*”: vivienda para una familia de 1-4 miembros.
  - “*Manufactured housing*”: vivienda manufacturada.
  - “*Multifamily*”: vivienda multifamiliar.

- **Purchaser\_type**: Tipo de un posible comprador de un préstamo. Puede tener asignados valores enteros situados entre 0 y 9 que representan las siguientes posibles entradas:
  - “*Loan was not originated or was not sold in calendar year covered by register*”: el préstamo no fue originado ni vendido en el año cubierto por el registro.
  - “*Fannie Mae (FNMA)*”: *Federal National Mortgage Association*.
  - “*Ginnie Mae (GNMA)*”: *Government National Mortgage Association*.
  - “*Freddie Mac (FHLMC)*”: *Federal Home Loan Mortgage Corporation*.
  - “*Farmer Mac (FAMC)*”: *Federal Agricultural Mortgage Corporation*.
  - “*Private securization*”: titulación privada.
  - “*Commercial bank, savings bank or savings association*”: banco comercial, caja de ahorros o asociación de ahorro.
  - “*Life insurance Company, credit unión, mortgage bank, or finance company*”: compañía de seguros de vida, cooperativa de crédito, banco hipotecario o compañía financiera.
  - “*Affiliate institution*”: institución afiliada.
  - “*Other type of purchaser*”: otro tipo de comprador.
- **Loan\_amount\_000s**: Cantidad del préstamo solicitado (en miles de dólares). Puede tener asignados valores enteros.

Como se puede apreciar, todos los atributos están indicados con valores enteros. Las muestras del *dataset* se encuentran clasificadas en 6 clases diferentes, indicando la acción llevada a cabo (atributo “*action\_taken*”) [13] [14]:

1. **Loan originated**: Préstamo originado.
2. **Application approved but not accepted**: Solicitud aprobada, pero no aceptada.
3. **Application denied by financial institution**: Solicitud denegada por la institución financiera.
4. **Application withdrawn by applicant**: Solicitud retirada por el solicitante.
5. **File closed for incompleteness**: Archivo cerrado por incompletitud.
6. **Loan purchased by your institution**: Préstamo comprado por institución.

### 5.1.2.- Datos ausentes

Antes de comenzar con el pre-procesado de los datos, es conveniente que todos los datos estén disponibles. El *dataset* contiene un total de 3 muestras a las que les falta el valor del campo “*applicant\_income\_000s*” (es decir, hay un valor nulo, **NaN** en el lenguaje de programación Python). Una posible solución a este problema sería sustituir los valores ausentes por la media de este atributo. Además, debemos asegurarnos de que todos los atributos utilizados tienen el mismo tipo de dato asociado (en Python, un atributo con valores nulos asume el tipo de dato “*float*” o “número real”).

### 5.1.3.- Visualización de los datos

Tras la anterior “limpieza” de los datos, observamos la distribución de las muestras del *dataset* en la Figura 7.

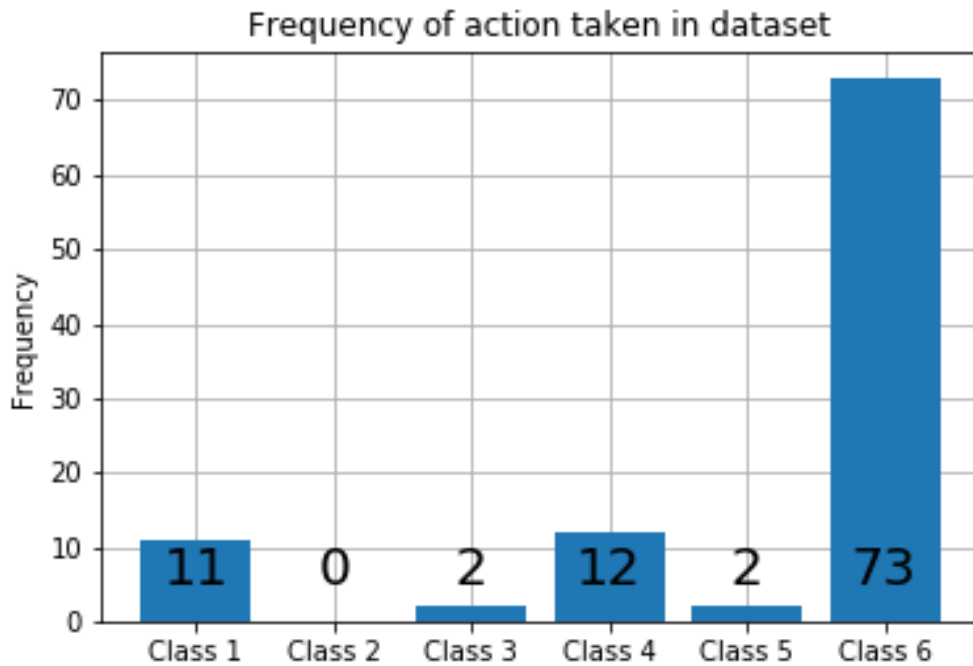


Figura 7: Cantidad de muestras de cada clase en el dataset

Como se puede observar, las clases no están en absoluto equilibradas, lo cual habrá que tener en cuenta para la posterior división de datos en los conjuntos de entrenamiento y validación.

Para obtener una visión global de los atributos que contiene el *dataset*, visualizaremos la distribución/frecuencia de cada uno de los atributos. En la Figura 8 y la Figura 9 se muestran dichos histogramas, diferenciando la clase a la cual pertenecen.

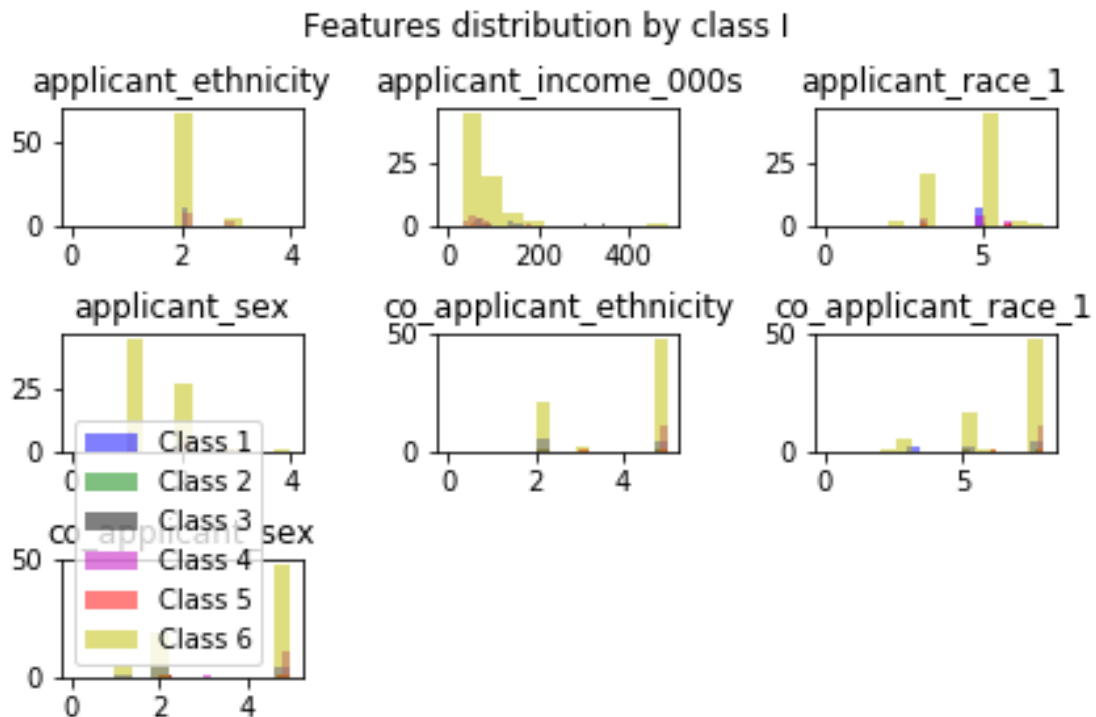


Figura 8: Distribución de los atributos en el dataset, separados por clases (I)



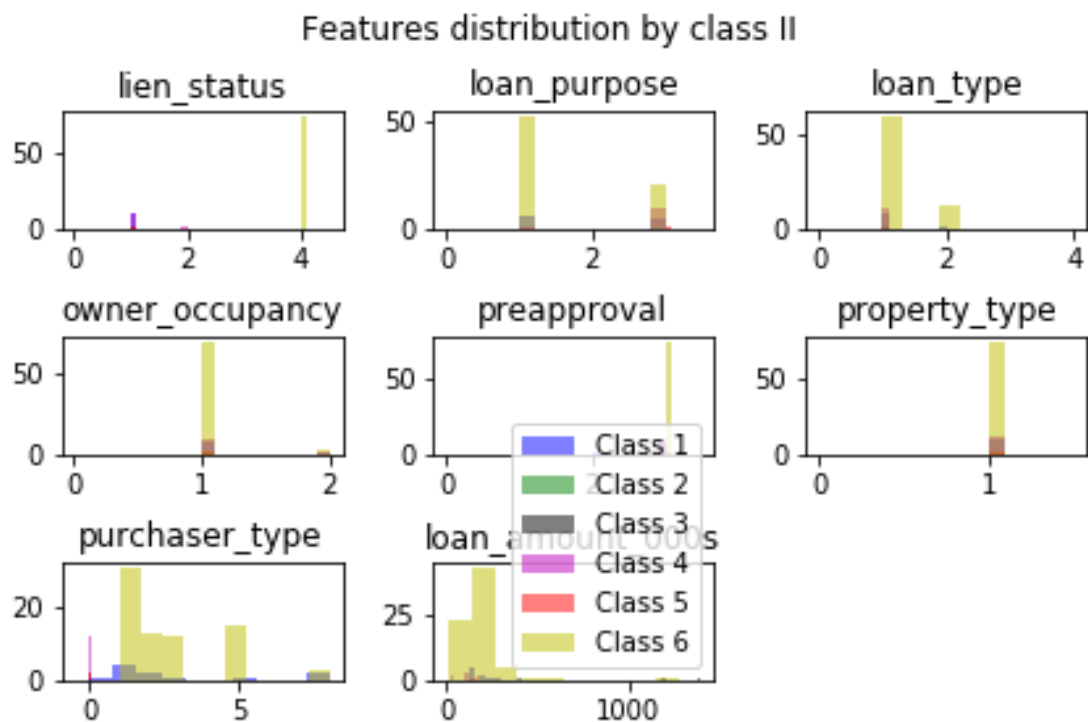


Figura 9: Distribución de los atributos en el dataset, separados por clases (II)

#### 5.1.4.- Partición de los datos

Para un correcto análisis de los datos, debemos dividir los datos en dos conjuntos: entrenamiento (para entrenar los algoritmos de clasificación que integran la inteligencia artificial) y validación (para ajustar los hiperparámetros de dichos clasificadores). Debido a que la distribución de las clases no es homogénea, es importante mantener estas distribuciones a la hora de realizar dichas particiones. El los datos del *dataset* en cuestión han sido divididos en 75 % (*training*) y 25 % (*validation*).

#### 5.1.5.- Almacenamiento del dataset de entrenamiento en una base de datos

Después de realizar la división de los datos que constituyen nuestro *dataset*, el conjunto resultante destinado a entrenar los clasificadores que integran la inteligencia artificial ha sido almacenado en una base de datos SQLite (haciendo uso del módulo `sqlite3` de Python). La estructura de la tabla que almacena estos datos incluye un identificador (que actúa como clave primaria), todos los atributos de las muestras almacenadas, y la clase asociada a cada muestra (atributo "*action\_taken*"), por lo que todas las columnas de la tabla almacenarán números enteros. El uso de una base de datos se justificará más adelante.

#### 5.1.6.- Normalización de los datos

Para que los algoritmos de clasificación que integran la inteligencia artificial puedan converger más rápidamente, se ha procedido a la normalización de los datos del *dataset*, utilizando `StandardScaler()` del paquete `sklearn.preprocessing`. Eliminamos la media ( $\mu = 0$ ) del conjunto de entrenamiento y escalamos los datos a varianza unitaria ( $\sigma = 1$ ). Cabe destacar que debe realizarse exactamente la misma transformación a todos los datos, no solamente a los datos de entrenamiento.

Una vez hemos normalizado todos los datos, concluye el proceso de pre-procesamiento de los datos y podemos proceder a analizarlos.

## 5.2.- Algoritmos de clasificación

A continuación, utilizamos los distintos modelos/clasificadores disponibles en la inteligencia artificial. Todos los clasificadores utilizados son proporcionados por la librería *scikit-learn*.

En todos los casos, realizamos distintos entrenamientos variando el valor de los hiperparámetros de cada modelo con el conjunto de datos de entrenamiento, y para cada uno de esos modelos, tras entrenarlos, computamos su coste/exactitud con el conjunto de datos de validación. Finalmente, escogemos la mejor configuración de hiperparámetros para entrenar el modelo final, con el cual se realizará una predicción con los datos introducidos por el usuario de la inteligencia artificial.

### 5.2.1.- Redes neuronales (Neural Networks)

Hemos utilizado la versión `MLPClassifier()` proporcionada por *scikit-learn* [8]. La estructura de redes neuronales utilizada incluye tantas neuronas de entrada como número de atributos, 1 capa oculta y tantas neuronas en la capa de salida como número de clases. En este caso, los parámetros a ajustar son el número de neuronas de la capa oculta y el término de regularización. Los resultados de entrenar distintas configuraciones de redes neuronales se muestran en la Figura 10. El mejor resultado se ha obtenido para 3 neuronas en la capa oculta con  $\lambda = 0,05$ . El entrenamiento del modelo se ha realizado en 0,015625 segundos.

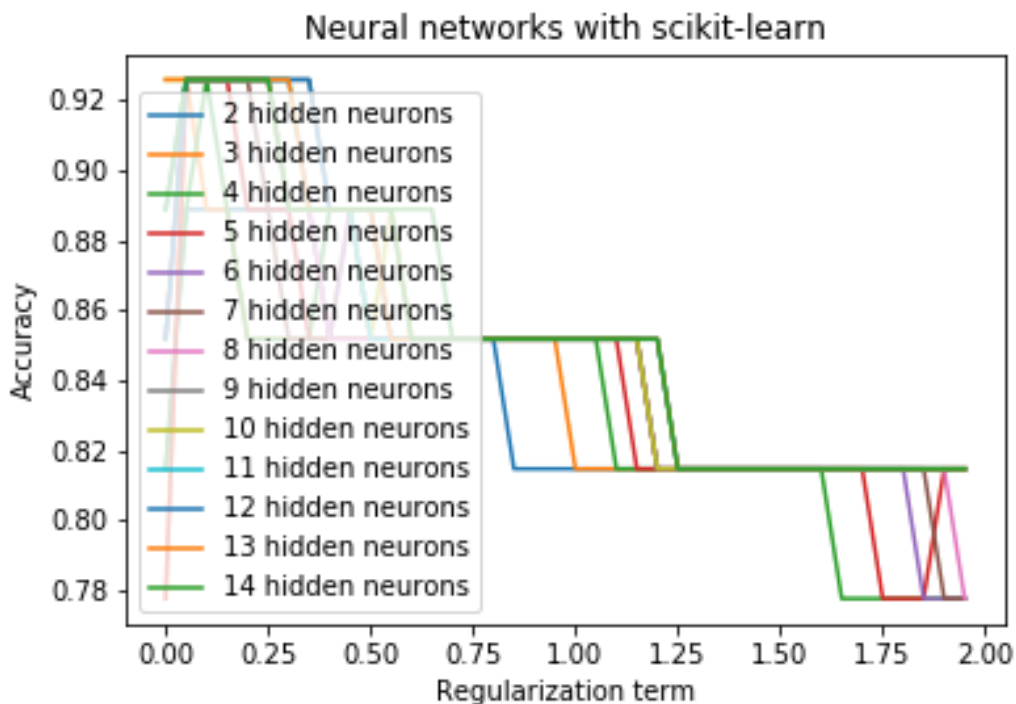


Figura 10: Exactitud de la red neuronal de scikit-learn según el término de regularización y el número de neuronas en la capa oculta

### 5.2.2.- Máquinas Soporte Vector (SVM)

Hemos utilizado el clasificador `SVC()` proporcionado por *scikit-learn* [9] con una función de kernel gaussiano (o su correspondiente aproximación en esta librería). Variamos los valores del parámetro de penalización ( $C$ ) y de la variable  $\sigma$  utilizada en el cálculo del coeficiente asociado al kernel, para encontrar el mejor modelo. Tras comparar las distintas precisiones obtenidas al

predecir el conjunto de validación (véase la Figura 11), el mejor resultado se ha obtenido con unos valores de  $C = 10$  y de  $\sigma = 10$ , siendo despreciable el tiempo de entrenamiento del modelo.

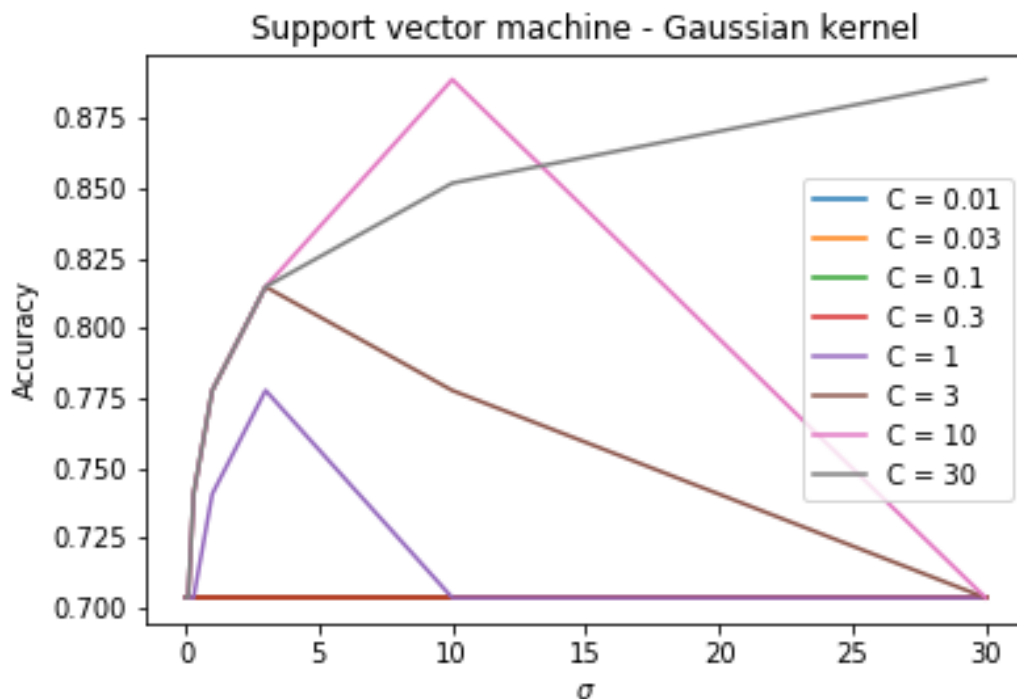


Figura 11: Exactitud de SVM con kernel gaussiano de scikit-learn según el valor de sigma ( $\sigma$ ) para distintos valores del parámetro de penalización ( $C$ )

### 5.2.3.- K-means

Pese a no ser un método de aprendizaje supervisado, se ha decidido contrastar los clasificadores anteriores con un algoritmo de aprendizaje no supervisado, K-means (utilizando la versión proporcionada por la librería *scikit-learn* [10]), configurado con un número de *clusters* igual al número de clases existentes en el conjunto de datos de entrenamiento. El tiempo de entrenamiento consumido ha sido de 0,03125 segundos.

### 5.3.- Generación de explicaciones

Uno de los aspectos más destacables de esta inteligencia artificial es la capacidad de generar explicaciones que el usuario de la misma pueda entender acerca de las predicciones realizadas por los algoritmos que la integran.

En primer lugar, en el caso de la predicción realizada por nuestra red neuronal, logramos generar explicaciones tras encontrar el camino más relevante (*Most Weighted Path*) dentro de la red neuronal [6]. Para ello, recorremos la red neuronal gracias al atributo *coefs\_* de la clase **MLPClassifier**, situándonos en primer lugar en la neurona correspondiente al valor de salida de la predicción efectuada por el algoritmo, y encontrando la dendrita entre capa y capa de neuronas con el mayor peso hasta alcanzar la neurona correspondiente al valor de entrada más relevante a la hora de realizar la predicción (véase la Figura 12). Sin embargo, si el atributo resultante es relativo al origen étnico, la raza o el sexo del solicitante o del co-solicitante, se ha optado por generar una explicación en función de la diferencia de frecuencias cuando se predice un valor de salida u otro distinto dado el valor del atributo introducido por el usuario [6], tras realizar determinadas consultas en la base de datos SQLite anteriormente mencionada, la cual contiene el conjunto de datos de entrenamiento utilizado por la inteligencia artificial.

```

max_weight = -10
current_layer = len(self.mlp.coefs_) - 1
dendrite = -1
while current_layer >= 0:
    for i in range(len(self.mlp.coefs_[current_layer])):
        if self.mlp.coefs_[current_layer][i][current] > max_weight:
            max_weight = self.mlp.coefs_[current_layer][i][current]
            dendrite = i

    max_weight = -10
    current = dendrite
    current_layer = current_layer - 1

inputFeature = current
return inputFeature

```

Figura 12: Algoritmo para encontrar "Most Weighted Path" en la red neuronal

En el caso del algoritmo SVM, al haber utilizado un kernel gaussiano, no es posible utilizar el atributo *coef\_* que proporciona la clase **SVC** para encontrar los pesos de los atributos de entrada a la hora de realizar la predicción.

Por último, en el caso del algoritmo K-means, podemos encontrar el atributo más relevante al realizar la predicción consultando las coordenadas del centro del *cluster* correspondiente al valor de salida generado (consultando el atributo *cluster\_centers\_* de la clase **KMeans**), incluyéndolo en la explicación correspondiente a este algoritmo.

## 5.4.- Ejemplo de uso

El usuario de la inteligencia artificial tendrá a su disposición una interfaz gráfica como la mostrada en la Figura 13, que podrá utilizar para introducir datos de test a partir de los cuales los diferentes algoritmos realizarán la predicción correspondiente. En la interfaz gráfica, implementada gracias a la librería **tkinter** de Python [15], existirá un *widget* para cada atributo introducido por el usuario (el usuario debe de haber introducido un valor para cada atributo antes de realizar la predicción): un conjunto de elementos **Combobox** y dos elementos **Spinbox** correspondientes a los atributos medidos en miles de dólares. Además, dispondrá de un área de texto donde se insertarán y podrán leer las explicaciones generadas a partir de la predicción.

Trabajo de Fin de Grado 2019-20 (Grado en Ingeniería Informática)

Welcome to this Human-Centric Artificial Intelligence to decide the granting of a mortgage.

INSTRUCTIONS:  
 In order to use this application correctly, fill ALL the fields below, and once you are done, click the 'Done!' button to see the results of data processing.  
 In order to reset the application, click the 'Reset' button.  
 In order to exit the application, click the 'Quit' button.

<p>Applicant ethnicity Information not provid <input type="text"/></p> <p>Applicant income (000s \$) <input type="text"/></p> <p>Applicant race Information not provid <input type="text"/></p> <p>Applicant sex Information not provid <input type="text"/></p> <p>Co-applicant ethnicity No co-applicant <input type="text"/></p> <p>Co-applicant race No co-applicant <input type="text"/></p> <p>Co-applicant sex No co-applicant <input type="text"/></p> <p>Lien status Not secured by a lien <input type="text"/></p>	<p>Loan purpose Home purchase <input type="text"/></p> <p>Loan type Conventional <input type="text"/></p> <p>Owner occupancy Owner-occupied as a p <input type="text"/></p> <p>Preapproval Preapproval was not re <input type="text"/></p> <p>Property type One-to-four family dw <input type="text"/></p> <p>Purchaser type Loan was not originate <input type="text"/></p> <p>Loan amount (000s \$) <input type="text"/></p>
--	--

Figura 13: Interfaz gráfica

Una vez el usuario ha introducido todos los datos solicitados, tras hacer click en el botón “Done!”, se podrán visualizar los resultados del procesamiento de los datos. El área de texto anteriormente mencionado contendrá la explicación generada por la inteligencia artificial (como se puede observar en la Figura 14), y además, el usuario podrá visualizar dos gráficas adicionales, cada una en una pequeña ventana complementaria.

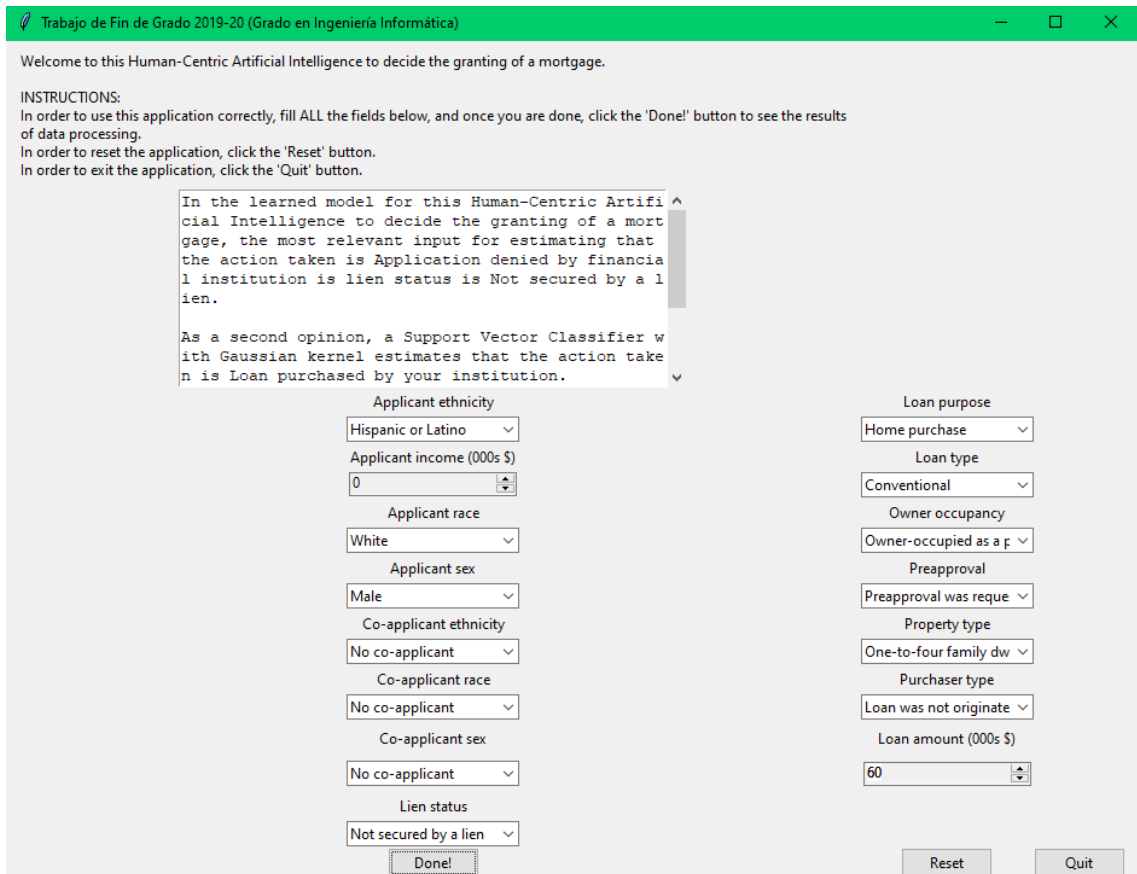


Figura 14: Ejemplo de uso

En la primera gráfica (mostrada en la Figura 15), se pueden apreciar los pesos de los atributos introducidos por el usuario al buscar la primera dendrita del camino más relevante dentro de la red neuronal que ha realizado la predicción (el orden de los atributos es el mismo que el utilizado para describirlos al principio del capítulo).

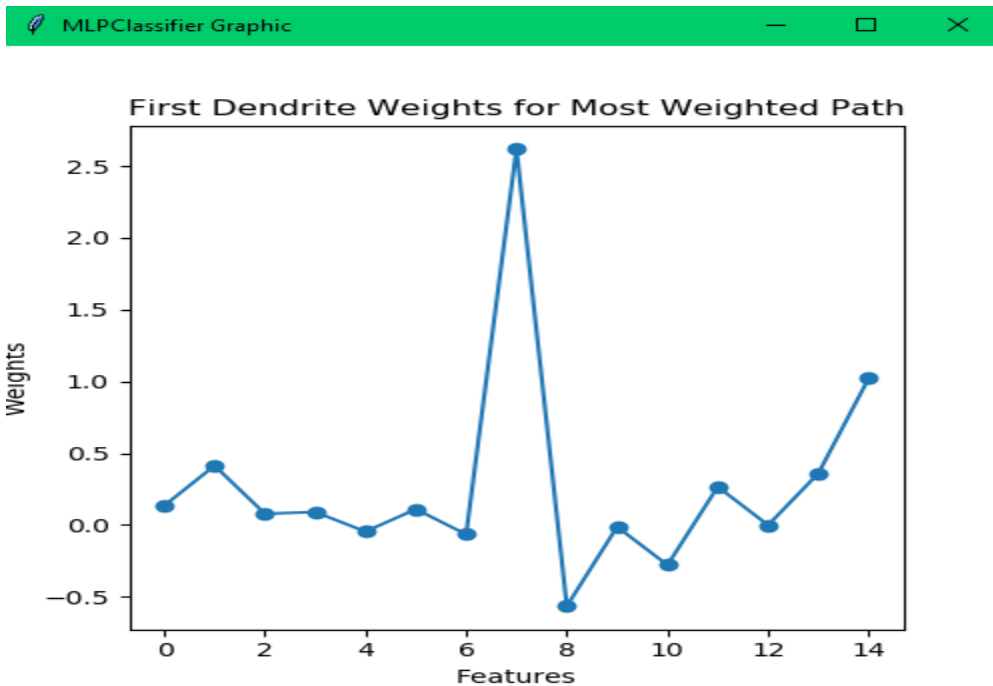


Figura 15: Gráfica asociada a la red neuronal

Por otro lado, la segunda gráfica (mostrada en la Figura 16) refleja las coordenadas del centro del *cluster* correspondiente al valor de salida obtenido del algoritmo K-means.

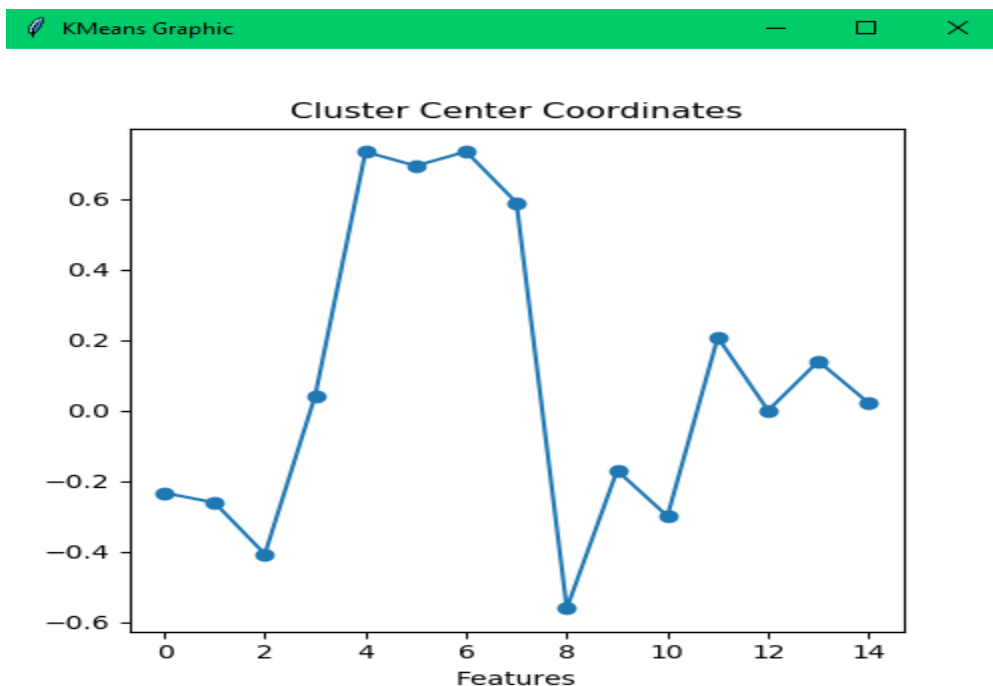


Figura 16: Gráfica asociada al algoritmo K-means

Cuando el usuario pulsa el botón “Reset”, cada uno de los *widgets* existentes en la interfaz de usuario tendrá asociado de nuevo su valor por defecto, y se vaciará el área de texto que contiene las explicaciones generadas, preparando la interfaz gráfica para una nueva predicción. Cuando el usuario desee finalizar su interacción con la inteligencia artificial, pulsará el botón “Quit” y se destruirán las ventanas existentes.





## Conclusión

Tras haber completado el desarrollo de este proyecto, hemos logrado una inteligencia artificial que permite predecir la acción tomada acerca de la concesión o no de una hipoteca a partir de una cierta cantidad de datos proporcionada por un usuario, acompañando el resultado de la predicción con una explicación autogenerada acerca de la decisión tomada que sea fácil de comprender por el usuario.

Hemos comprobado que, en términos de precisión, el algoritmo más preciso a la hora de tomar la decisión financiera ha sido una red neuronal, al igual que ha sido el algoritmo que ha permitido en mayor medida generar la mejor explicación posible acerca de sus predicciones.

Sin embargo, el mundo financiero cambia constantemente, y como consecuencia de ello, como se ha mencionado anteriormente, puede resultar recomendable ampliar la cantidad de datos utilizados para entrenar nuestra inteligencia artificial, con la finalidad de lograr en el futuro la máxima precisión posible en las predicciones realizadas y en sus correspondientes explicaciones.

# Conclusion

After having completed the development of this project, we have achieved a Human-centric Artificial Intelligence approach that allows us to predict the action taken regarding the granting or not of a mortgage from a certain amount of data provided by an user, accompanying the result of the prediction with a self-generated explanation of the decision made that is easy for the user to understand.

We have verified that, in terms of accuracy, the most accurate algorithm when making the financial decision has been a neural network, just as it has been the algorithm that has allowed us to generate the best possible explanation about its predictions.

However, the financial world is constantly changing, and as a consequence, as mentioned above, it may be advisable to expand the amount of data used to train our Artificial Intelligence approach, in order to achieve in the future the highest possible precision in the predictions made and in their corresponding explanations.

# Bibliografía

1. Definición de Hipoteca:  
URL <https://es.wikipedia.org/wiki/Hipoteca>
2. Clem, Brenda. "The Capture 2.0 Revolution." *MReport* Jan. 2020: 32-35. Print.
3. Inteligencia Artificial centrada en los Humanos:  
URL <https://www.cognizant.com/glossary/human-centered-ai>
4. Hind, M. (2019). Explaining explainable AI. *XRDS: Crossroads, The ACM Magazine for Students*, 25(3), 16-19.
5. Zurada, J., & Zurada, M. (2002). How Secure Are Good Loans: Validating Loan-Granting Decisions And Predicting Default Rates On Consumer Loans. *Review of Business Information Systems (RBIS)*, 6(3), 65-84.
6. García-Magariño, I., Muttukrishnam, R., & Lloret, J. (2019). Human-centric AI for trustworthy IoT systems with explainable multilayer perceptrons. *IEEE Access*, 7(1), 125562-125574.
7. Introducción a la programación de GUI con TKinter:  
[https://python-textbok.readthedocs.io/en/1.0/Introduction to GUI Programming.html](https://python-textbok.readthedocs.io/en/1.0/Introduction%20to%20GUI%20Programming.html)
8. Página de manual de MLPClassifier:  
[https://scikit-learn.org/stable/modules/generated/sklearn.neural\\_network.MLPClassifier.html](https://scikit-learn.org/stable/modules/generated/sklearn.neural_network.MLPClassifier.html)
9. Página de manual de SVC:  
URL <https://scikit-learn.org/stable/modules/generated/sklearn.svm.SVC.html>
10. Página de manual de KMeans:  
URL <https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html>
11. Datos sobre hipotecas:  
URL <https://data.world/datasets/mortgage>
12. FFIEC Home Mortgage Disclosure Act:  
URL <https://www.ffiec.gov/hmda/>
13. Loan/Application Register Code Sheet:  
URL <https://www.ffiec.gov/hmda/pdf/code.pdf>
14. 2015 HMDA Code Sheet:  
URL <https://www.ffiec.gov/hmdarawdata/FORMATS/2015HMDACodeSheet.pdf>
15. Tutorial de TKinter:  
URL <https://likegeeks.com/es/ejemplos-de-la-gui-de-python/>