Universidad Cenfotec

Maestría en Tecnología de Bases de Datos

Documento Final de Proyecto de Investigación Aplicada 2

Elaboración de una aplicación de minería de datos para clasificar e identificar monedas romanas del siglo IV en Egipto

Caldas Donato, Ana Cristina

Mayo 2018

Agradecimientos

Le agradezco a mi tutor, Luis Naranjo, por su ayuda durante todo el proceso de elaboración de esta tesis.

A los profesores y a la Universidad Cenfotec, que me han impulsado a crecer profesionalmente.

A mis amigos, y en especial a Irene Soto, quien me permitió colaborar en su investigación de doctorado.

A mi familia, por todo su apoyo en todas las áreas de mi vida, especialmente durante esta maestría.

Sobre todo, le agradezco a Dios por permitirme llegar hasta acá. Sin Él, nada de esto sería posible.

Tabla de Contenido

Lista de Figuras

Abstract

El presente trabajo tiene como objetivo la elaboración de una aplicación de minería de datos para clasificar e identificar monedas romanas del Siglo IV en Egipto.

Se utilizaron las características conocidas de monedas de este periodo para crear modelos de minería de datos que pudieran ser utilizados para encontrar características desconocidas de las monedas y para clasificarlas cuando no ha sido posible hacerlo utilizando los métodos existentes.

Al concluir la investigación, se esperaba seleccionar los modelos con los mejores resultados y crear una aplicación que utilizara solamente estos modelos.

Al no encontrar modelos útiles con los datos obtenidos durante la investigación, se elaboró una solución que le permite al usuario crear sus propios modelos predictivos con datos nuevos utilizando árboles de clasificación, evaluar los modelos, y utilizarlos para clasificar las monedas.

Se espera que esta solución permita que investigadores en New York University descubran vínculos económicos entre sociedades antiguas al encontrar patrones que no han sido detectados utilizando los procesos arqueológicos tradicionales, en el contexto de una tesis doctoral sobre el tema.

Palabras Clave: arqueología, minería de datos, mundo antiguo, comercio, patrones, monedas antiguas, clasificación, predicción, Egipto.

Capítulo 1. Introducción

1.1 Generalidades

Esta investigación se realizó con datos públicos que se han tomado de otras publicaciones y no fue necesario ningún tipo de acuerdo de confidencialidad. La investigación se efectuó en paralelo a una tesis doctoral del Institute for the Study of the Ancient World, realizada por Irene Soto, pero se espera que la investigación sea útil para al menos tres instituciones: Institute for the Study of the Ancient World (New York University), American Numismatic Society, y Oxford Roman Economy Project (University of Oxford, Faculty of Classics).

El propósito es que ambas investigaciones se logren complementar; los datos e información contextual para esta investigación fueron provistas por la persona que realizó la tesis doctoral, y el resultado de esta investigación fue utilizado para completar aquella.

Se ha incluido la carta de aval como Apéndice 1 y el cronograma del proyecto como Apéndice 2.

1.2 Antecedentes del Problema

Se han recopilado grandes cantidades de datos y se han analizado de manera manual a través de los años, pero se ha hecho muy poco en estas organizaciones, especialmente en el campo de la numismática, por identificar los patrones o relaciones de manera automatizada. Esto ha limitado mucho los resultados obtenidos hasta el momento y las posibilidades de encontrar vínculos entre los datos. Las personas afiliadas al instituto no tienen el conocimiento necesario para realizar un proyecto de

este tipo, y hasta el momento no se ha buscado contratar a nadie externo. En otras palabras, el problema no se ha intentado resolver utilizando minería de datos antes, y los resultados obtenidos hasta el momento a través de otras investigaciones más tradicionales, las cuales usan métodos manuales, no resuelven el problema.

1.3 Definición y Descripción del Problema

Desde principios del último siglo hasta ahora, se han encontrado miles de monedas en diferentes excavaciones y expediciones arqueológicas. Utilizando las características de cada moneda, los expertos han logrado identificar y clasificar gran parte de ellas, pero no todas. Al trabajar con una cantidad tan alta de monedas, es difícil para los arqueólogos comparar cada moneda con todas las demás monedas que existen.

La economía del Imperio Romano en el cuarto siglo después de Cristo es un tema de interés que la arqueología estudia actualmente. Aunque existen muchos objetos arqueológicos, es un periodo particularmente difícil para el análisis histórico. Consecuentemente, no se han realizado tantos descubrimientos de la economía egipcia en el cuarto siglo.

El reinado del emperador Diocleciano trajo mucha confusión política, económica, y numismática. Dos de los cambios económicos que implementó Diocleciano a finales del tercer siglo fue la abolición de la zona monetaria aislada de Egipto, y la introducción de nuevas monedas. Esta era la primera vez que los egipcios podían utilizar monedas producidas fuera de su territorio, como en Antioquía, Roma, Nicomedia, y Esmirna, ya que, antes de la reforma de Diocleciano, la mayoría habían sido producidas en Alejandría. Se han encontrado grandes cantidades de estas monedas en Egipto, que

parece indicar una alta producción local de monedas y la existencia de muchas monedas importadas. Entre las monedas encontradas, un porcentaje significativo se compone de imitaciones, creadas mediante moldes. Todas estas características han dificultado la clasificación e identificación de monedas y de las relaciones de comercio entre Egipto y el resto del imperio.

Por las complicaciones que presenta el cuarto siglo, los datos de los estudios no se encuentran centralizados, y en lugar de ello se han tomado de diversas fuentes. De cualquier manera, las monedas que sí se encuentran en bases de datos centralizadas pueden ser analizadas y agrupadas en algunos casos, pero no se llega a encontrar patrones entre estas monedas que permita clasificarlas. Para realizar un análisis más completo de las monedas, se necesita un modelo de minería de datos que tome en cuenta todas las características registradas de una moneda y que pueda clasificarlas y encontrar de manera confiable a cuál grupo de monedas pertenece.

1.4 Justificación

Por medio de esta investigación, se espera mejorar el proceso de identificación y clasificación de monedas antiguas, especialmente las monedas romanas encontradas en Egipto y producidas alrededor del cuarto siglo. Además, se pretende disminuir el tiempo y trabajo requerido de los arqueólogos al no tener que depender completamente de sus habilidades de reconocimiento. También se quieren encontrar relaciones que existen entre las monedas, el lugar donde fueron producidas y dónde fueron encontradas.

Originalmente, se esperaba que la utilización de diferentes algoritmos de clasificación, asociación, segmentación y regresión produjera modelos que fueran útiles

para investigaciones acerca de Egipto en el Imperio Romano y posiblemente de otras civilizaciones antiguas.

Usando los algoritmos mencionados anteriormente, se trataron de crear modelos que pudieran:

- Identificar el año aproximado en el que se produjo una moneda
- Identificar dónde se produjo una moneda
- Identificar grupos de monedas similares
- Identificar monedas, ya clasificadas, que difieran significativamente del resto del grupo.

Al presentarle a los institutos interesados una propuesta de uno o más modelos de minería de datos que clasificaran sus monedas de manera confiable, estos podrían utilizarlos para descubrir información previamente desconocida de las monedas que ya han sido encontradas y para clasificar más rápidamente y con más certeza monedas recién descubiertas, facilitando nuevos descubrimientos históricos en el mundo arqueológico.

Ya que estos algoritmos no presentaron buenos resultados al utilizarlos con los datos recopilados hasta el momento, se creó una aplicación que permite que los mismos institutos elaboren fácilmente sus propios modelos predictivos y utilicen el método que dio mejores resultados durante la investigación. Esta aplicación llegó a ser más útil para los institutos que la establecida originalmente, ya que se pueden crear modelos con nuevos atributos y nuevos datos.

1.5 Viabilidad

Lo más importante para llevar a cabo la investigación fue la obtención de los datos de las monedas que fueron analizadas. Al momento de realizar el análisis de los datos, existía una base de datos de más de treinta mil (30,000) monedas con sus respectivas características.

Los datos han sido proporcionados por Irene Soto, doctoranda de NYU, y la investigación se está realizando con la aprobación del Institute for the Study of the Ancient World.

Los datos se encuentran en formato Excel y contienen:

- El número de RIC (Roman Imperial Coinage) o Cohen, el cual indica el catálogo en el que se encuentra descrita la moneda.
- La ceca de la moneda, es decir, el lugar donde se fabricó.
- La denominación de la moneda.
- El año mínimo en el que pudo haber sido fabricada la moneda, inferido por las características de la moneda, como su denominación (cambia según la época y el emperador).
- El año máximo en el que pudo haber sido fabricada la moneda.
- El metal que se utilizó.
- Marca de la ceca.
- Latitud y longitud aproximada de donde fue fabricada.

Este conjunto de datos fue suficiente para realizar la investigación y buscar patrones que fueran útiles para la clasificación de los objetos.

En general, se concluyó que el proyecto era viable.

1.5.1 Punto de Vista Técnico.

Los principales conocimientos técnicos necesarios para realizar esta investigación fueron análisis de datos, minería de datos y programación. La investigadora ha adquirido estos conocimientos a través de los años, en sus cursos universitarios de bachillerato y maestría.

1.5.2 Punto de Vista Operativo.

La investigación no requirió alterar el funcionamiento normal del Institute for the Study of the Ancient World, ni de ninguna otra organización interesada. Los datos de las monedas existen en catálogos y libros de numismática romana y se recopilaron para la investigación doctoral acerca de la economía de Egipto en el Imperio Romano durante el cuarto siglo después de Cristo. Aparte de estos datos, solamente se requirieron reuniones esporádicas con la persona que trabajó en la investigación doctoral (mencionada arriba) y como los resultados de este proyecto son útiles para dicha investigación, ha sido tratado como una colaboración de ambas partes.

1.5.3 Punto de Vista Económico.

Para el análisis y modelado de datos, se requirió un computador, que fue provisto por la investigadora. La mayor parte del costo del proyecto provino de las horas laboradas por la investigadora durante la realización del proyecto. Este costo puede ser considerado como un costo teórico, ya que la investigadora los asumirá y el instituto no asumirá ningún costo. Si se considera que la investigadora trabajó en el

proyecto por siete meses, dedicándole aproximadamente dos horas diarias y valorando su trabajo en $32 por hora, el costo teórico fue de $14,784.

La aplicación se puede instalar localmente en cualquier computador que utilice Windows 7 64 bit en adelante. El costo total de propiedad correspondería al costo de operación del computador personal de quien decida instalar la aplicación.

1.6 Objetivos

Se ha seleccionado la taxonomía de Bloom de 1956 para la definición de objetivos, ya que es ampliamente aceptada en el ámbito académico y la que mejor sustenta los objetivos planteados para este trabajo de investigación.

1.6.1 Objetivo General.

Elaborar una aplicación de minería de datos para clasificar e identificar monedas romanas del Siglo IV en Egipto.

1.6.2 Objetivos Específicos.

1. Identificar los datos disponibles de monedas romanas producidas en el Siglo IV después de Cristo que hayan sido encontradas en Egipto
2. Descubrir los datos que podrían ser útiles en la búsqueda de patrones
3. Elaborar diferentes modelos de minería de datos para la clasificación de monedas
4. Analizar los resultados de los modelos de minería de datos y seleccionar los mejor modelos.

1.7 Alcances y Limitaciones

1.7.1 Alcances.

Al completar la investigación, se entregó:

- El documento con los resultados de la investigación y la evaluación de los diferentes modelos de minería de datos.
- Una aplicación de escritorio que les permita a la doctoranda y a sus colegas en ISAW crear sus propios modelos de minería de datos, evaluarlos con conjuntos de datos de prueba, y luego producir sus propios resultados de clasificación utilizando los modelos de minería de datos.
- Una guía para la utilización de la aplicación.

1.7.2 Limitaciones.

Los modelos creados con la aplicación utilizarán árboles de clasificación, ya que fueron los que presentaron mejores resultados a lo largo de la investigación. Por lo tanto, los modelos creados por el usuario solamente podrán predecir variables categóricas y no variables continuas.

1.8 Marco de Referencia Organizacional y Socioeconómico

El Institute for the Study of the Ancient World, mejor conocido como ISAW, es un centro de investigación académica avanzada y de educación universitaria de la Universidad de Nueva York (New York University, o NYU por sus siglas en inglés). Se enfoca en el estudio de las conexiones económicas, religiosas, políticas y culturales

entre civilizaciones antiguas, y cuenta con programas de doctorado y posdoctorado. Actualmente dirigen varias excavaciones arqueológicas en Europa y Asia (Institute for the Study of the Ancient World, 2017).

1.8.1 Historia.

ISAW fue fundado en el año 2006 por la Fundación Leon Levy (Leon Levy Foundation). Leon Levy, inversionista exitoso y filántropo, siempre estuvo interesado en la historia del mundo antiguo. En sus últimos años de vida, junto con su esposa Shelby White, planeó la creación de un instituto donde expertos en arqueología e historia pudieran explorar los vínculos culturales y comerciales entre civilizaciones antiguas. Levy murió en el 2003, y una de las primeras iniciativas de su fundación fue la realización de este plan (The Leon Levy Foundation – The Legacy of Leon Levy, s.f.).

1.8.2 Tipo de Negocio y Mercado Meta.

ISAW no pretende ser un negocio ni lucrar por medio de sus investigaciones. Es un instituto de aprendizaje que tiene como fin la preparación de expertos y expandir el conocimiento actual acerca del mundo antiguo, para así educar al público general y compartir los descubrimientos realizados con ellos.

Se podría decir que el mercado meta del instituto son aquellos interesados en el mundo antiguo, pero esto no es del todo cierto. Este sí se podría considerar como su mercado meta, principal pero, realmente lo que se busca es educar al público general, por medio de exhibiciones, publicaciones, charlas, y recursos digitales. Con esto, se espera crear interés en él y que más personas puedan tener un mayor conocimiento de

pueblos antiguos, de su cultura, economía, religión, política, de sus similitudes y diferencias (Institute for the Study of the Ancient World, 2017).

1.8.3 Misión, Visión y Valores.

ISAW no ha compartido su misión y visión de manera pública, pero en términos generales, la misión de ISAW es cultivar investigaciones vinculadas comparativas del mundo antiguo, desde el oeste del Mediterráneo hasta China y comunicar la información encontrada acerca de la antigüedad al público. El estudio de estos patrones y conexiones históricas son el centro de su misión (The Leon Levy Foundation – The Legacy of Leon Levy, s.f.).

Entre los ideales del instituto están la unión de disciplinas y de estudios de pueblos antiguos. ISAW busca cruzar las fronteras entre diferentes campos académicos y promover metodologías que permitan la integración de diferentes métodos de análisis en la antropología, arqueología, geografía, historia, economía, sociología, historia del arte, e historia de ciencia y tecnología (Institute for the Study of the Ancient World, 2017).

1.8.4 Políticas Institucionales.

La interacción con el instituto durante la elaboración de este proyecto fue mínima; por lo tanto, no se encontró ninguna política de la organización que incida de manera directa en la investigación. Toda comunicación se realizó por medio de la doctoranda Irene Soto (ver Apéndice 1), con quien la investigadora se comunicó al menos dos veces al mes durante el transcurso de la investigación.

1.9 Estado de la Cuestión.

1.9.1 Repositorios numismáticos de monedas antiguas.

A través de los años, se han utilizado diferentes métodos para el estudio de historia y arqueología. Un método importante ha sido la investigación numismática. Utilizando diferentes características de las monedas que han sido excavadas y encontradas del siglo pasado, se pueden llegar a muchas conclusiones. Un análisis simple acerca de la cantidad de monedas encontradas por región, o la cantidad de monedas que fueron producidas en oro o en plata, puede ayudar a un historiador a realizar descubrimientos nuevos.

Como se mencionó anteriormente, las monedas son especialmente útiles en estudios del mundo antiguo. Se pueden utilizar como evidencia de relaciones de comercio entre diferentes pueblos y permite que los historiadores tengan un conocimiento más amplio de los lazos económicos que existían en ese momento.

Esto no es nada nuevo para los interesados en la arqueología o la numismática. En el 2011, el Instituto para el Estudio del Mundo Antiguo (ISAW) y la Sociedad Numismática Americana (ANS) comenzaron juntos un proyecto para recopilar todos los datos de monedas romanas antiguas llamado OCRE, Online Coins of the Roman Empire (Meadows & Gruber, 2014). Estas colecciones se enlazan utilizando los objetos individuales, los grupos de objetos a los que pertenecen (llamados “hoards”), y las investigaciones realizadas. OCRE ha llegado a ser, básicamente, una base de datos consolidada que toma datos de diferentes bases de datos alrededor del mundo en diferentes idiomas y los une en un mismo lugar. Permite realizar búsquedas y hacer análisis comparativo entre monedas romanas antiguas. Por ejemplo, la Figura 1,

generada por la herramienta, muestra la cantidad de monedas en la base de datos que fueron producidas con cada tipo de material.

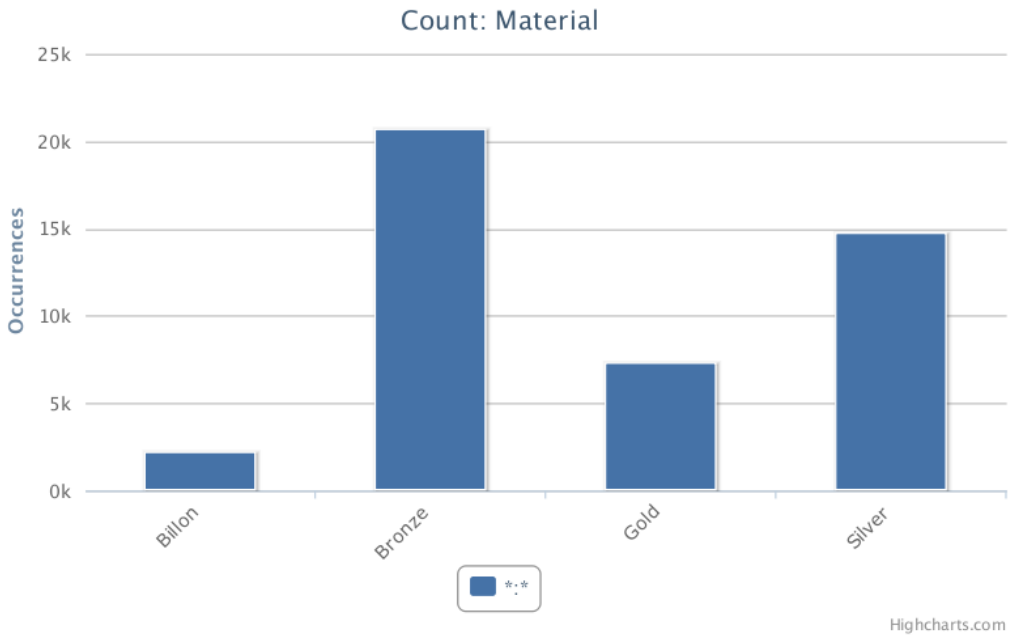


Figura 1: Gráfico producido por OCRE (elaboración propia)

Esta información puede ser relativamente útil para investigaciones arqueológicas o numismáticas, pero si se consideran todas las posibilidades de análisis que se podrían realizar con este conjunto de datos, se puede apreciar que realmente se está haciendo apenas una fracción de lo que se podría hacer con los datos.

Existe otra desventaja al utilizar las herramientas de análisis de OCRE. Si una moneda no se ha logrado agrupar con otras monedas romanas porque no se conoce una de sus características, es excluida de las consultas. Esto quiere decir que aquellas monedas que no tienen características definidas no pueden ser analizadas del todo por medio de OCRE.

En síntesis, existe mucho potencial para minería de datos, especialmente para el estudio de monedas romanas antiguas, pero no está siendo explotado.

1.9.2 Técnicas de minería de datos para estudios numismáticos.

La minería de datos se ha utilizado para proyectos arqueológicos en muchas ocasiones para la identificación y clasificación de objetos. En el año 2010, se realizó un estudio acerca de la minería de datos aplicada a los petroglifos (Zhu, Wang, Keogh & Lee, 2010), en el cual se creó un algoritmo que lograba detectar con considerable precisión los tipos de petroglifos en las piedras.

En el campo de la numismática, también se han realizado investigaciones para crear sistemas de clasificación de monedas (Zambanini, Kampel & Schlapke, 2008), pero, así como el estudio de los petroglifos, se centran en el reconocimiento de monedas por medio de imágenes y de reconocimiento óptico de caracteres (OCR).

Esta tecnología, aunque ha sido bastante exitosa para encontrar monedas que se asemejan, actualmente, no llega a ser de mucha utilidad en investigaciones arqueológicas. En la actualidad, las monedas antiguas, así como cualquier otro objeto antiguo con valor histórico encontrado, pasan por manos de arqueólogos, quienes son expertos en la materia. Ellos no solo identifican qué tipo de moneda es, pero también recopilan todas las características de la moneda que puedan, basándose en su propia observación y su conocimiento adquirido a través de los años.

En estos casos, el problema no es la identificación de monedas visualmente similares, sino es la identificación de sus características. Hay atributos que no son difíciles de descifrar, como el color, el tipo de material, la zona geográfica donde se encontró, entre otros, pero hay algunos que no son tan fáciles de reconocer.

Este es uno de los problemas que se quiere resolver. Si se crean modelos de minería de datos con los datos obtenidos, se puede intentar encontrar el valor más

probable para las características desconocidas. Además, utilizando modelos de minería de datos, también se pueden buscar relaciones y patrones entre las monedas y esto podría llevar a nuevos descubrimientos acerca de las relaciones de comercio entre Egipto y el resto del Imperio Romano durante el Siglo IV.

La minería de datos podría ser muy útil para la arqueología, pero hasta el momento se ha enfocado en lograr el reconocimiento visual de objetos. Se podrían encontrar relaciones mucho más interesantes si el enfoque fuera la búsqueda de patrones basada en las características de los objetos antiguos, particularmente de las monedas.

Capítulo 2. Marco Conceptual

Actualmente, se desconoce qué tan fuerte fue la relación económica entre Egipto y el resto del Imperio Romano en el Siglo IV. Esta relación se ha intentado estudiar utilizando métodos de análisis comunes, pero al ser una época tan confusa política y económicamente, no se ha logrado profundizar en el tema.

Por otro lado, se tienen grandes cantidades de objetos arqueológicos de la época que han sido analizados y documentados. Estos objetos, especialmente las monedas antiguas, podrían ser de gran utilidad en estos estudios, ya que por medio de ellas se puede demostrar el movimiento y comercio entre diferentes pueblos (Soto, I., comunicación personal, 17 de marzo, 2017).

Es aquí donde se propuso hacer uso de la minería de datos. Utilizando técnicas de minería de datos, se investigaron las conexiones económicas de Egipto con el resto del Imperio Romano en el Siglo IV. Se esperaba que las diferentes técnicas ayudaran a encontrar patrones y relaciones entre las monedas, el lugar donde se encontraron y

el lugar donde fueron producidas. Luego, se utilizarían los mejores modelos de minería de datos y se podría intentar encontrar los valores faltantes. Toda esta información se recolectaría con el fin de entender el papel económico de Egipto en el Imperio Romano (ver Figura 2).

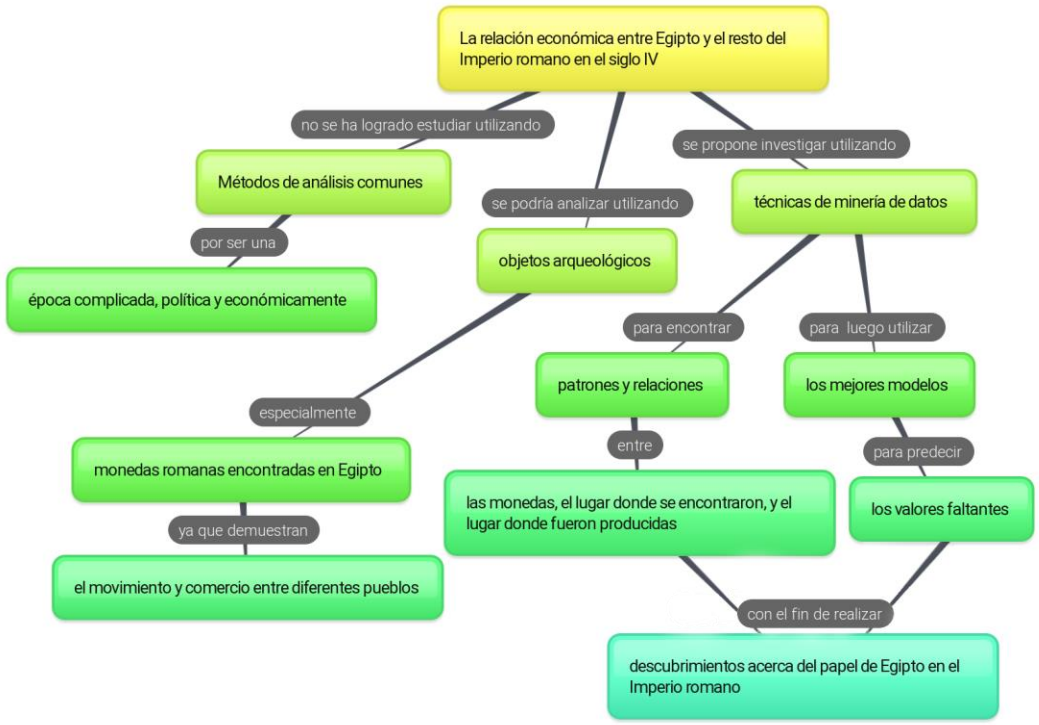


Figura 2: ¿Por qué estudiar la relación económica entre Egipto y el Imperio Romano en el Siglo IV? (elaboración propia)

La minería de datos puede referirse a diferentes procesos o grupos de procesos, pero en este caso se puede definir como el descubrimiento de conocimiento a partir de grandes conjuntos de datos. Algunos consideran más adecuado llamar a la minería de datos “minería de conocimiento a partir de datos” (Han, Kamber, & Pei, 2012), ya que el objetivo final es la obtención de conocimiento y no de datos.

Existen diferentes tipos de variables que se pueden utilizar en la minería de datos. Los atributos categóricos se identifican por nombre y sus valores representan categorías, también llamadas factores o clases, sin ningún orden en particular. Los atributos ordinales se pueden considerar como parte de los atributos categóricos, pero sí siguen un orden específico (Han, Kamber, & Pei, 2012).

El conjunto de datos que se recibió de la doctoranda Soto se debió dividir en dos: el conjunto de datos de entrenamiento y el conjunto de datos de pruebas. El conjunto de datos de entrenamiento, generalmente es el mayor de los dos y es con el cual se crean los modelos de minería de datos. El conjunto de datos de prueba se utiliza para evaluar el rendimiento de los modelos creados con los datos de entrenamiento (Shmueli, Bruce, & Patel, 2010).

Se le llama observación a cada instancia encontrada en el conjunto de datos. Es decir, cada fila del conjunto de datos, que contiene una columna por cada atributo, corresponde a una observación. En el caso de esta investigación, cada observación representa a una moneda (Diez, Barr, & Çetinkaya-Rundel, 2016).

Una etapa importante del análisis de datos es la visualización de los datos y las relaciones entre ellos. Para esto, se utilizan diferentes tipos de gráficos, según las variables que estén siendo comparadas.

Los gráficos de barras se utilizan para ver la distribución de observaciones en relación con un atributo categórico específico. Una variación de esto es la inclusión de un segundo atributo categórico, que puede ser representado por segmentos de varios colores en cada barra. De esta manera, el tamaño de la barra corresponde a la cantidad total de observaciones por cada factor del primer atributo y el tamaño de cada sección de la barra, separada en diferentes colores, corresponde a la proporción de

observaciones por cada factor del segundo atributo (Diez, Barr, & Çetinkaya-Rundel, 2016).

Los diagramas de cajas, o box plots, son gráficos que se utilizan para ver la relación entre un atributo categórico y un atributo numérico. Utiliza cuartiles para mostrar la distribución de los datos por cada atributo categórico, según el atributo numérico. También facilitan la visualización de valores atípicos, o outliers (Diez, Barr, & Çetinkaya-Rundel, 2016).

Los gráficos de dispersión son utilizados para encontrar la relación entre dos atributos numéricos. Cada punto en el gráfico corresponde a una observación donde el eje x representa el valor de un atributo numérico y el eje y representa el otro (Diez, Barr, & Çetinkaya-Rundel, 2016).

La clasificación y la agrupación por clústeres son diferentes técnicas que se pueden utilizar en la minería de datos. La clasificación es el proceso por medio del cual se encuentra un modelo que distingue y describe clases de datos o conceptos. El análisis de clústeres, a su vez, agrupa objetos similares y puede ser usado para crear nuevas clasificaciones (Han, Kamber, & Pei, 2012). Existen diferentes algoritmos de cada tipo que se pueden utilizar para realizar el análisis de los datos (ver figura 3).

El algoritmo utilizado para la creación de modelos de clasificación se selecciona según el tipo de variable. La regresión logística es un modelo generalizado que calcula la probabilidad de que una variable categórica presente uno de dos valores (Diez, Barr, & Çetinkaya-Rundel, 2016). De ser necesario, el algoritmo también se puede adaptar para seleccionar el valor más probable entre tres o más clases, llamado, regresión logística multi clase (Bishop, 2006).

Los modelos de redes neuronales se pueden describir como una serie de transformaciones funcionales. Consiste en nodos que procesan la información para buscar los valores óptimos para el modelo, y luego le pasan esta información a nodos en la siguiente capa, quienes realizan lo mismo, hasta llegar a la capa final con el resultado. Al igual que la regresión logística, se puede utilizar para la categorización de dos o más clases, pero requiere mucho más poder computacional. (Bishop, 2006).

En los árboles de decisión, se construye un modelo utilizando una estructura de árbol, donde los nodos internos representan condiciones de prueba que separan los registros que tengan características diferentes, y donde cada hoja, o nodo final, tiene una clase. Para predecir cuál es el valor del atributo, se comienza a bajar por el árbol de decisión, sometiendo a cada registro a las condiciones de prueba hasta llegar a la clase que se predice (Tan, Steinbach & Kumar, 2005).

Los bosques aleatorios crean varios árboles de decisión separados, donde cada árbol utiliza un subconjunto de predictores. Este método evita la alta correlación entre los árboles y brinda un nivel de confianza mayor al de los árboles de decisión. Los bosques aleatorios requieren más recursos que utilizar solamente un árbol de decisión, pero pueden dar mejores resultados (James, Witten, Hastie & Tibshirani, 2015).

Existen otros métodos híbridos que se pueden utilizar, donde cada modelo procesa los datos por separado, y al final el resultado se decide por un sistema de votación. Son modelos costosos, pero si ningún modelo presenta buenos resultados por sí solo, se puede evaluar el uso de ellos (Bauer & Kohavi, 1999).

En el caso de la agrupación, el análisis de clústeres divide datos en grupos que son significativos o útiles, basándose en sus características. A diferencia de los métodos de clasificación mencionados anteriormente, no se conocen los posibles

grupos o clases a los que pueden pertenecer los datos. K-means es un método de agrupación que establece K centros, tomando la media como centro, y forma K clústeres asignando cada punto a su centro más cercano, hasta que los valores de los centros se mantengan. Las variables numéricas son más apropiadas para los métodos de agrupación que las categóricas, ya que las categóricas no tienen media definida (Tan, Steinbach & Kumar, 2005).

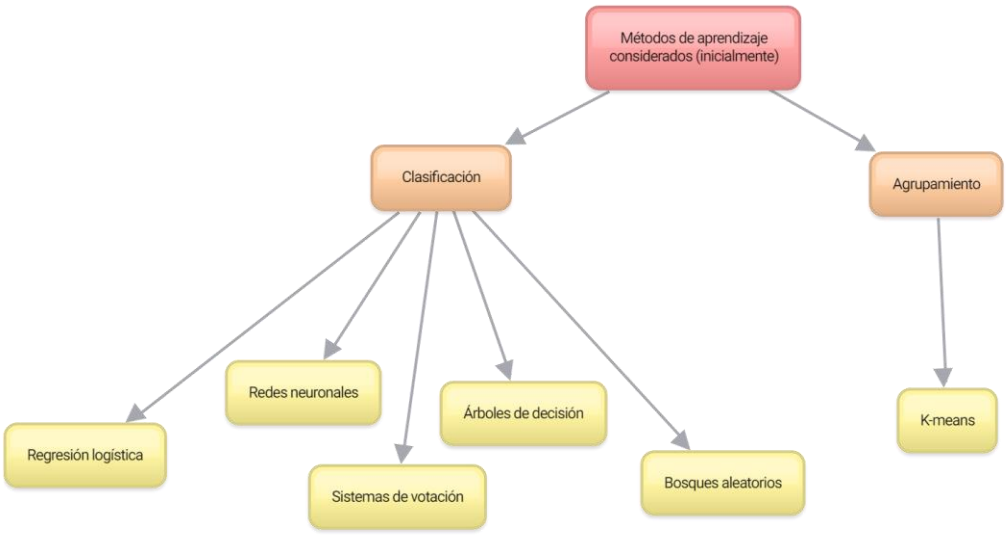


Figura 3: Métodos de aprendizaje (elaboración propia)

Existen diferentes métodos para evaluar los resultados de un modelo predictivo. La matriz de confusión se puede utilizar cuando se está intentando clasificar un atributo categórico. Una matriz de confusión es una tabla que contiene una fila por cada factor del atributo que se encuentra en el conjunto de datos de prueba, y una columna por cada factor del atributo que predijo el modelo. El número en cada celda es la suma de las observaciones donde el factor real en el conjunto de datos de prueba corresponde a la fila y el factor que se predijo corresponde a la columna. Si el factor de la columna y el factor de la fila son iguales, el número en la celda representa la cantidad de

observaciones que se predijeron correctamente para ese factor. El resto de las celdas en esa fila representan la cantidad de observaciones para ese factor que no se predijeron correctamente (Sammut & Webb, 2011).

Existen diferentes valores que pueden tomarse de las matrices de confusión. En casos binarios, donde los únicos factores son “sí” y “no”, los verdaderos positivos son la cantidad de casos donde se predijo “sí”, y el valor real era “sí”. Los verdaderos negativos son la cantidad de casos donde se predijo “no”, y el valor real era “no”. Ambos corresponden a predicciones correctas.

En el caso de los falsos positivos y los falsos negativos, el primero es la cantidad de casos donde se predijo “sí” y el valor real era “no”, el segundo es la cantidad de casos donde se predijo “no” y el valor real era “sí”. Ambos corresponden a predicciones incorrectas.

Con estos valores, se pueden calcular varias métricas, como exactitud (predicciones correctas divididas entre el total), especificidad (falsos positivos divididos entre cantidad real de “no”), sensibilidad (verdaderos positivos divididos entre cantidad real de “sí”), entre otros.

Estas métricas son útiles cuando se trata de clasificación binaria, pero en el caso de clasificación multi-clase, la más común durante esta investigación, las métricas deben ser adaptadas (Kelleher, Mac Namee & D’Arcy, 2015).

La métrica de exactitud global en la clasificación multiclase es similar a la de la clasificación binaria, donde se suman las predicciones correctas de todas las clases y se dividen por el total de observaciones. Las métricas de especificidad y sensibilidad se adaptan para convertirse en la exactitud por clase, donde, por cada clase, se toma

la cantidad de predicciones correctas y se divide por la cantidad de observaciones para esa clase específica.

Las métricas permiten evaluar el modelo desde diferentes puntos de vista y con estos datos se puede decidir de manera informada si el modelo es útil o no.

Si se necesitan evaluar un modelo predictivo donde el atributo que se predice es numérico, existen otras métricas que pueden ser utilizadas.

RMSE es la raíz cuadrada de la varianza de los residuos. Indica la distancia entre los valores reales de las observaciones y los valores de la predicción. Entre más bajo sea el valor de RMSE, mejor es el modelo (Kuhn & Johnson, 2016).

MAE, el error medio absoluto, es más simple que RMSE y más fácil de interpretar. Es el promedio absoluto de la diferencia entre los valores reales de las observaciones y los valores de la predicción. Al igual que el RMSE, entre más bajo sea el valor, mejor es el modelo. Los gráficos de dispersión pueden ser utilizados para visualizar esta relación, donde el eje X del gráfico corresponde al valor real de la observación y el eje Y corresponde al valor de la predicción. Entre más cerca estén los puntos de la línea X=Y, mejor es el modelo (Kelleher, Mac Namee & D'Arcy, 2015).

Capítulo 3. Marco Metodológico

3.1 Tipo de Investigación

El tipo de investigación utilizado en este proyecto fue investigación aplicada. Se aplicaron técnicas de minería de datos a los datos tomados de diferentes fuentes por la doctoranda Irene Soto, con el propósito de complementar su investigación acerca de las relaciones económicas entre Egipto y el resto del Imperio Romano en el Siglo IV.

3.2 Alcance Investigativo

Como no se había realizado este tipo de análisis con estas monedas anteriormente, la investigación debió partir de un estudio exploratorio. Durante esta etapa exploratoria, la investigadora buscó patrones o relaciones significativas entre las diferentes monedas que fueron estudiadas.

Si durante esta primera etapa se descubría que sí parecía existir cierta relación entre las monedas romanas encontradas en Egipto y las cecas donde fueron producidas, se esperaba continuar con estudios descriptivos y explicativos que profundizaran más en estas relaciones.

3.3 Enfoque

Para este estudio se utilizó un enfoque alternativo; no hay por qué separar lo cualitativo y lo cuantitativo. Esto se llevó a cabo al utilizar las siguientes dimensiones:

- Dimensión ontológica: Se han encontrado miles de monedas romanas del Siglo IV en Egipto y varias de sus características han sido documentadas en diferentes libros: su denominación, la colección a la que pertenecen (grupo de artefactos encontrados juntos), la ubicación geográfica de la colección, el rango de años entre los cuales se produjo la moneda, la marca de la ceca, el metal utilizado, la ubicación geográfica de la ceca, entre otros (ver Figura 4). El objetivo fue usar estos atributos para crear diferentes tipos de modelos de minería de datos, ya que son categóricos, con pocos valores posibles y numéricos. Una vez creados los modelos, se esperaba seleccionar los mejores, y crear una solución que presentara los resultados con el mejor grado de

confianza.

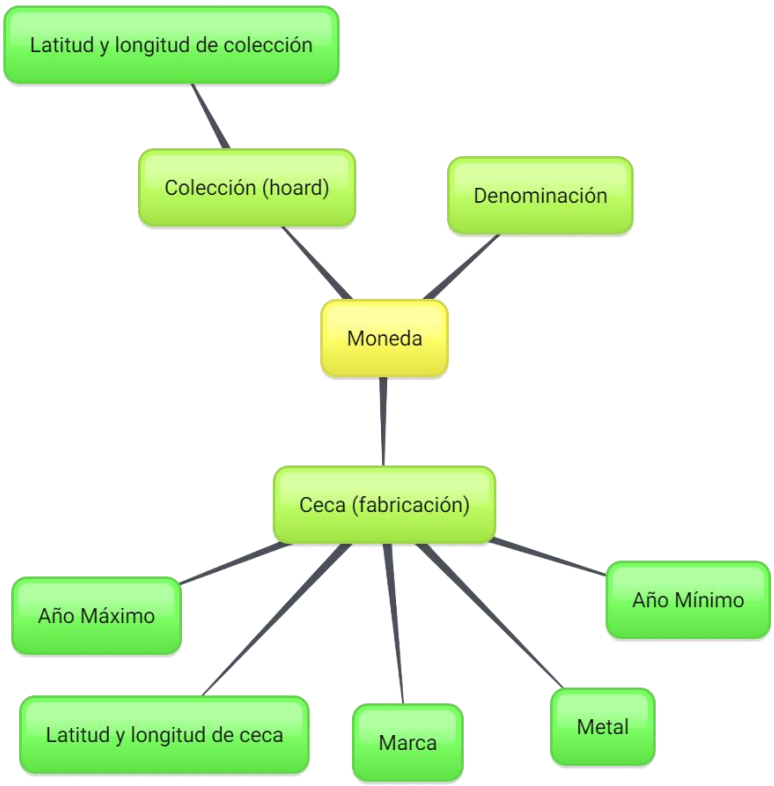


Figura 4: Atributos de una moneda (elaboración propia)

- Dimensión epistemológica: La posición de la investigadora fue de involucrada, ya que tuvo acceso a datos de miles de monedas romanas del Siglo IV encontradas en Egipto y con ellos realizó el análisis y elaboró la solución.
- Dimensión axiológica: La evaluación de los modelos fue esencialmente comparativa. Para realizar esta comparación, se utilizó la escala de valores mostrada en la Tabla 1. Los modelos que presentaran la puntuación más alta para cada uno de los atributos serían seleccionados como los mejores y se utilizarían en la solución final. A su vez, estos grados de confianza son evidencia de la existencia o inexistencia de patrones y relaciones entre los

datos. Al no existir estándares de evaluación para modelos de clasificación de monedas antiguas, la investigadora definió, usando su propio criterio, que un modelo debería tener un puntaje de al menos 70% para considerarse bueno.

	Valor	Opciones			
Exactitud total de modelo	60%	Más del 75%: 50%		Más del 80%: 55%	Más de 90%: 60%
Porcentaje mínimo de aciertos por atributo	20%	Más de 60%: 5%		Más de 70%: 10%	Más de 75%: 20%
Complejidad del algoritmo	10%	Red neuronal: 2%	Bosques aleatorios: 3%	Árboles de decisión: 5%	Regresión lineal: 10%
Cantidad de variables independientes	10%	5 o más variables: 5%		3-4 variables: 7%	1-2 variables: 10%

Tabla 1: Criterios de evaluación de los modelos de minería de datos

3.4 Diseño

El diseño utilizado fue el diseño de integración múltiple, ya que no se haría ninguna separación entre los métodos cualitativo y cuantitativo (Hernández, Fernández, & Baptista, 2014). De manera iterativa, se realizó lo siguiente por cada atributo:

1. Se estudió el atributo.
2. Se intentaron encontrar relaciones de otros atributos con este.

3. Se crearon varios modelos, enfocados en este atributo.
4. Se evaluaron los modelos.

Una vez seleccionados los mejores modelos (los que tuvieran un mayor nivel de confianza) para todos los atributos, se crearía la aplicación y se presentarían los resultados del análisis.

3.5 Población y Muestreo

Como ya se ha mencionado anteriormente, la doctoranda Irene Soto ha recolectado y continúa recolectando datos de monedas romanas del Siglo IV encontradas en Egipto. Actualmente, la base de datos contiene más de treinta mil monedas, pero continúa creciendo y se espera que continúe de manera indefinida.

La población que se utilizó para la investigación fue la base de datos de más de treinta mil monedas, documentadas por Soto. Las monedas recolectadas después del fin de la investigación podrán ser utilizadas con la aplicación para crear nuevos modelos.

La división entre los datos utilizados para el entrenamiento de los modelos y los datos utilizados para la prueba de los modelos fue automatizada y tuvo una proporción 70/30. Usando una función aleatoria, se seleccionaron 70% del conjunto de datos total para utilizar como el conjunto de datos de entrenamiento y el 30% restante se utilizó para el conjunto de datos de prueba. Los modelos de minería de datos se crearon con los datos de entrenamiento y se probaron con los datos de prueba.

3.6 Instrumentos de Recolección de Datos

Gran parte de los datos crudos han sido tomados por la doctoranda Irene Soto del libro de Hans-Christoph Noeske, "*Die Muñzfunde des aëgyptischen Pilgerzentrums Abu Mina und die Vergleichsfunde aus den Dioecesen Aegyptus und Oriens vom 4.-8. Jh. n. Chr: Prolegomena zu einer Geschichte des spaťroïmischen Muñzumlaufs in Aëgypten und Syrien*.", publicado en el año 2000, aunque también incluyó datos de otras fuentes. La investigadora utilizó los datos que recibió de Soto, y no recolectó datos por su cuenta.

3.7 Técnicas de Análisis de Información

Los datos se recibieron en formato Excel, separados en varias hojas de cálculo diferentes y con leves diferencias en su estructura. Antes de comenzar el análisis de los datos, se debió realizar una limpieza y estructuración de los datos, para obtener un archivo CSV que contuviera todos los registros estandarizados en una misma tabla.

Cuando los datos ya se encontraron estructurados, se comenzaron a analizar todos los atributos, uno por uno, utilizando técnicas de minería de datos comunes (ver Figura 5).

Figura 5: Acciones realizadas por atributo (elaboración propia)

Capítulo 4. Análisis del Diagnóstico

4.1 Análisis de Atributos

La cantidad de atributos recibidos fue menor de lo esperado, entonces se crearon algunos adicionales basados en los originales. Los atributos originales son los siguientes:

- **ric**: Roman Imperial Coinage. Identificador de catálogo de algunas monedas romanas. No será utilizado.
- **mint**: Ciudad donde fue fabricada la moneda (ceca). Variable cualitativa.
- **min**: Año mínimo en el que se pudo haber fabricado la moneda. Se va a tratar como variable cuantitativa.
- **max**: Año máximo en el que se pudo haber fabricado la moneda. Se va a tratar como variable cuantitativa.
- **denom**: Denominación de la moneda. Variable cualitativa.

- **hoard**: Grupo de monedas encontradas (tesoro o colección) al que pertenece. Variable cualitativa.
- **metal**: Metal con el que se fabricó la moneda. Puede ser de bronce o de oro. Variable cualitativa.

A partir del atributo mint, se crearon los siguientes atributos derivados:

- **lat**: Latitud en la que se encontraba la ciudad de fabricación. Solamente se utiliza en el cálculo de distLin.
- **long**: Longitud en la que se encontraba la ciudad de fabricación. Solamente se utiliza en el cálculo de distLin.
- **distAlex**: Distancia en kilómetros del recorrido entre el lugar de fabricación hasta Alejandría. Variable cuantitativa.
- **tiempoAlex**: Tiempo en días del recorrido entre el lugar de fabricación hasta Alejandría. Variable cuantitativa.
- **distCategory**: Categoría de distancia del lugar de fabricación hasta Alejandría (near, mid-range, far). Variable cualitativa.
- **port**: Define si la ciudad de fabricación es un puerto o no. Puede ser "yes", "no", o "unknown".
- **sea**: El mar al que pertenece el puerto. Tiene seis posibles valores. En caso de que no sea un puerto, el valor es "none".
- **dirRome**: La dirección de la ciudad de fabricación relativa a Roma. Puede ser este (este de Roma), oeste (oeste de Roma) o Roma.

A partir de los atributos min y max, se crearon los siguientes atributos derivados:

- **capital**: Capital del Imperio Romano en el momento de su fabricación. Variable cualitativa, con tres posibles valores: “con","rom", y "rom/con". En el caso de rom/con, el rango de posibles fechas de fabricación cubre ambos periodos.
- **period**: Periodo Reece en el cual se fabricó la moneda. Los periodos Reece son rangos de años creados por el numismático Richard Reece para analizar monedas del imperio Romano. Existen 21 periodos Reece en total. Los posibles valores utilizados en esta investigación son 1 a 6, los cuales corresponden a rangos de años entre 296 y 388. La numeración real de los periodos Reece utilizados sería 15, 16, 17, 18, 19, y 20, pero para simplificar los modelos se usaron los números 1, 2, 3, 4, 5, y 6.

Aunque el número total de monedas es de poco más de 30,000, la doctoranda Irene Soto consideró que algunos datos desconocidos podrían ser problemáticos para el análisis, por lo que se decidió estudiar solamente las monedas que contengan identificador RIC, que provengan de una ceca conocida, que hayan sido fabricadas entre el año 296 y 388, y que pertenezcan a un solo periodo Reece. Esto bajó el número de monedas utilizadas a 7,141.

Se crearon gráficos para visualizar las relaciones entre los diferentes atributos (ver Apéndice 3). El análisis de cada una de estas relaciones se presenta a continuación en la Tabla 2. Si no se mencionan los valores desconocidos del atributo en el análisis, se asume que las observaciones con valores desconocidos no fueron tomadas en cuenta.

Mint	Min/Max	No parece existir un patrón claro entre el atributo mint y los atributos min y max, pero los rangos de años sí parecen diferir bastante entre una ceca y otra. Podría ser una relación útil.
Mint	Denom	Las proporciones de monedas fabricadas de diferentes

		denominaciones sí parecen tener alguna relación con la ceca, ya que cada denominación tiene proporciones de cecas diferentes. La denominación con más observaciones es hce, aunque existe una cantidad aun mayor de monedas de denominación desconocida. Podría ser una relación útil.
Mint	Period	Se puede ver que la cantidad de monedas fabricadas en cada ceca cambia significativamente en cada periodo. Podría ser una relación útil.
Mint	Hoard	La distribución de cecas por grupos de monedas encontradas es interesante, porque definitivamente varía de un grupo a otro. Como fue el caso en las relaciones anteriores, no se distingue ningún patrón claro, pero podría ser útil en un modelo.
Mint	DistAlex	La relación entre las cecas y su distancia de Alejandría muestra que las cecas con mayor cantidad de observaciones son las más cercanas, aunque no parece ser una relación lineal. Antioquía (ant) no está en Egipto, pero tiene más observaciones que otras cecas en Egipto (aegy). Tesalónica se encuentra más cerca que Cízico, pero muchas menos monedas fueron fabricadas ahí. También es importante notar que Roma, aunque no es una ceca cercana, tiene más monedas que cecas más cercanas, incluyendo Constantinopla. El número total de monedas fabricadas fuera de Egipto es mayor que el de las monedas fabricadas dentro de Egipto (ale, aegy). Es una relación interesante pero no sería útil en un modelo, ya que el atributo de distancia se deriva de la ceca.
Mint	TimeAlex	La relación entre Mint y TimeAlex es similar a la relación Mint y DistAlex. Se puede ver que la mayor producción de monedas se realizó en cecas que estaban entre 0 a 10 días de Alejandría, aunque también hay una producción significativa en cecas entre 10 y 20 días de Alejandría. La cantidad de monedas producidas en cecas a más de 20 días de Alejandría es mínima, aunque es importante notar que se han encontrado monedas fabricadas en cecas a 50 días de distancia. Como DistAlex, es una relación interesante pero no sería útil en un modelo, ya que el atributo de distancia se deriva de la ceca.
Mint	Capital	La proporción de monedas fabricadas antes y después de trasladar la capital parece variar según la ceca. Podría ser una relación útil.
Mint	Port	El gráfico muestra que la mayoría de las monedas provienen de cecas que estaban localizadas en puertos. Es una relación interesante pero no sería útil en un modelo, ya que el atributo puerto se deriva de la ceca.
Mint	Sea	Al dividirlo las cecas por mar, se puede apreciar que la mayoría de las monedas fueron fabricadas en ciudades no costeras, seguidas por cecas en el Mediterráneo. Si no se tomara en cuenta Alejandría, la cantidad de monedas

		producidas en el Mediterráneo sería menor que en el mar de Mármara. Es una relación interesante pero no sería útil en un modelo, ya que el atributo de mar se deriva de la ceca.
Mint	DistCategory	El gráfico muestra que entre más cerca está el grupo de cecas, más observaciones tiene. Es una relación interesante, pero no sería útil en un modelo, ya que el atributo de distancia se deriva de la ceca.
Mint	Metal	No hay muchas monedas de oro comparadas a las de bronce, pero es interesante ver que de la ceca con más monedas (Alejandría), no se tiene ninguna de oro. Antioquía parece ser el mayor productor de monedas de oro. Por si sola no podría ser usada en un modelo, pero junto con otras podría ser una relación útil.
Mint	DirRome	La mayoría de las monedas parecen haber sido fabricadas en cecas al este de Roma. No sería útil en un modelo, ya que el atributo de dirección se deriva de la ceca.
Denom	Min/Max	No parece existir un patrón claro entre el atributo denom y los atributos min y max, pero los rangos de años sí parecen diferir bastante entre una denominación y otra. La denominación fol, por ejemplo, parece ser mucho más común a principios del Siglo IV. Podría ser una relación útil.
Denom	Period	La cantidad de monedas por denominación parece variar según el periodo. Podría ser una relación útil.
Denom	Hoard	Las denominaciones encontradas en los diferentes grupos de monedas parecen ser diferentes. Las monedas solidus son pocas, pero pertenecen a grupos de monedas específicos. No se distingue ningún patrón claro, pero podría ser útil en un modelo.
Denom	DistAlex	La relación entre las denominaciones de las monedas y la distancia de Alejandría no parece ser particularmente útil.
Denom	TimeAlex	La relación entre las denominaciones de las monedas y el tiempo de viaje hasta Alejandría no parece ser particularmente útil
Denom	Capital	Por la distribución de los gráficos, se puede apreciar que las denominaciones de monedas fabricadas cuando Roma era la capital eran más que todo fol, o de denominación desconocida. No parece que sea una relación útil.
Denom	Port	A simple vista, las ciudades con puerto parecen fabricar más monedas hce y menos solidus, proporcionalmente, que las ciudades sin puerto. De cualquier manera, no parece ser una relación útil para un modelo.
Denom	Sea	Según el gráfico, el Mar Negro y Mar Mediterráneo fabricaron muy pocas monedas solidus. Se fabricaron más en el Mar de Mármara y el Mar Egeo, pero la mayor producción fue en ciudades sin puerto. Podría ser una relación útil.
Denom	DistCategory	Como también fue comprobado con otras relaciones, las cecas que se encuentran más lejos de Alejandría son las que fabricaban menos hce, pero proporcionalmente más fol. Las monedas de distancia desconocida (eastmint, westmint) tienen un porcentaje mucho más alto de fol, y no tiene tantas

		monedas de denominación desconocidas como otros grupos de distancias. Puede ser una relación útil.
Denom	Metal	Todas las monedas son de bronce menos las pocas monedas de oro que existen. Estas monedas corresponden a todas las monedas de la denominación solidus. Es decir, si una moneda es solidus, es de oro, y viceversa. Por esta razón, esta relación no sería útil en un modelo.
Denom	DirRome	Proporcionalmente, hay muchas más monedas fol fabricadas al oeste de Roma y muchas menos hce. El porcentaje de cen y de solidus también es bajo en el oeste, comparado a Roma y a las ciudades al este de Roma. Podría ser una relación útil.
Hoard	Min/Max	No parece existir un patrón claro entre el atributo hoard y los atributos min y max, pero los rangos de años sí parecen diferir bastante entre un grupo de monedas y otro. Podría ser una relación útil.
Hoard	DistAlex	La relación entre los grupos de monedas y la distancia de su ceca de Alejandría muestra que la distribución no es igual. Podría ser una relación útil.
Hoard	TimeAlex	La relación entre los grupos de monedas y el tiempo de viaje es similar a la relación entre los grupos de monedas y la distancia. Podría ser una relación útil.
Hoard	Period	La mayoría de las monedas del periodo 1 parecen provenir del mismo grupo de monedas, “ae17”. Esta relación puede ser peligrosa en los modelos, ya que sin considerar ese grupo de monedas, la cantidad de monedas del periodo 1 es mínima. Es una relación que vale la pena explorar.
Hoard	Capital	La mayoría de las monedas fabricadas cuando Roma era la capital también provienen del grupo ae17. No parece ser demasiado útil para crear un modelo, y utilizar min/max posiblemente sería más útil, pero es importante tomar en cuenta la cantidad de observaciones que pertenecen al grupo ae17.
Hoard	Port	La distribución por ceca costal sí parece variar según el grupo de monedas. Podría ser una relación útil, pero otros atributos relacionados podrían ser más útiles.
Hoard	Sea	No parece existir ningún patrón entre los dos atributos. Podría ser una relación útil, pero posiblemente se necesitaría incluir otros atributos en el modelo.
Hoard	DistCategory	La mayoría de los grupos de monedas parecen tener monedas de cecas cercanas y lejanas. Hay algunas que tienen monedas de áreas específicas, pero son grupos de pocas monedas. Podría ser útil junto con otros atributos, pero el atributo de distancia cuantitativo podría ser mejor.
Hoard	Metal	Todas las monedas de oro pertenecen a grupos específicos: “karanis 1974”, todos los “au”, y “alex chatby”. El resto son de bronce. Podría ser útil junto con otros atributos.
Hoard	DirRome	Casi todos los grupos de monedas contienen en su mayoría monedas del este de Roma, pero esto es de esperarse, ya que es donde fueron fabricadas la mayoría de las monedas.

		La mayoría de los grupos parecen tener también monedas de Roma y del oeste, solo que en menor cantidad. No parece ser una relación útil.
Capital	Min/Max	La relación no es útil porque el atributo capital se deriva de los atributos min y max.
Capital	DistAlex	Según lo que muestra el gráfico, en promedio, las monedas fabricadas cuando Roma era la capital fueron fabricadas en cecas más lejos de Alejandría que las que fueron fabricadas después de cambiar la capital a Constantinopla. Podría ser una relación útil.
Capital	TimeAlex	La relación entre la distancia en KM y el tiempo en días del viaje según la capital es parecida. Podría ser útil usar una de las dos en un modelo.
Capital	Period	La relación entre capital y periodo no es útil, ya que ambos se derivan de min y max.
Capital	Port	Proporcionalmente, las cecas no costeras parecen haber producido más monedas cuando Roma era la capital que las cecas costeras. Podría ser una relación útil.
Capital	Sea	Proporcionalmente, el Mar Negro y el Mar Egeo tienen la mayor cantidad de monedas producidas cuando Roma era la capital, Podría ser una relación útil.
Capital	DistCategory	Proporcionalmente, las cecas que producían más monedas cuando Roma era la capital eran las que estaban lejos de Alejandría, aunque en total las cecas a distancia media fueron las que tuvieron la mayor producción cuando Roma era la capital. Podría ser una relación útil.
Capital	Metal	No se tiene ninguna moneda de oro que haya sido fabricada cuando Roma era la capital. Es un dato interesante, pero posiblemente no sea útil para un modelo a menos que se utilice junto con otros atributos.
Capital	DirRome	Aunque se fabricaron muchas más monedas al este de Roma, proporcionalmente, hay un mayor porcentaje de monedas entre las que fueron fabricadas al oeste, que fueron producidas cuando la capital era Roma. Podría ser una relación útil.
Port	Min/Max	Las monedas fabricadas en cecas no costeras parecen tender a ser producidas a principios del siglo. Las cecas con puerto parecen haber aumentado su producción un poco más tarde. Podría ser una relación útil.
Port	DistAlex	El gráfico muestra que la mayoría de las cecas cercanas tienen puertos, con la excepción de Antioquía. No es un atributo útil porque puerto y distancia se derivan del mismo atributo, mint.
Port	TimeAlex	El gráfico muestra una relación similar a la de DistAlex, y de igual manera, no es un atributo útil porque ambos se derivan del atributo mint.
Port	Sea	No es un atributo útil porque puerto y mar se derivan del mismo atributo, mint.
Port	Period	La proporción de monedas fabricadas en ciudades costeras y no costeras parece variar según el periodo. Podría ser una

		relación útil.
Port	DistCategory	El gráfico permite ver que la mayoría de las monedas fabricadas en cecas “mid-range” provienen de puertos. La mayoría de las monedas de cecas más distantes no provienen de puertos. Es una relación interesante pero poco útil, ya que ambos atributos se derivan del atributo mint.
Port	Metal	Según el gráfico, la mayoría de las monedas de bronce proviene de cecas no costales. Podría ser una relación útil junto con otros atributos, ya que la cantidad de monedas de oro no parece ser muy significativa.
Port	DirRome	El gráfico no muestra ninguna relación interesante entre los atributos.
Sea	Min/Max	Los años de fabricación de las monedas parecen ser diferentes según el mar donde fueron fabricadas. Podría ser una relación útil.
Sea	DistAlex	La mayoría de los puertos de donde provienen las monedas están cerca de Alejandría, posiblemente entre 1000 y 1200 km, con la excepción de la misma Alejandría en el Mediterráneo. Hay algunas cecas mediterráneas que están un poco más lejos, pero no son muy significantes. Las monedas que pertenecen a cecas ambiguas, donde no se conoce si tienen puerto o no (por ejemplo, eastmint incluye todas las cecas desconocidas del este del Imperio, y no se sabe si son puertos), están relativamente cerca de Alejandría. De cualquier manera, no es una relación útil, ya que ambos se derivan del atributo ceca.
Sea	TimeAlex	La relación entre el mar y el tiempo de viaje es similar a la relación entre el mar y la distancia. No es una relación útil, ya que ambos se derivan del atributo ceca.
Sea	Period	La cantidad de monedas fabricadas por mar parece variar de periodo a periodo. Podría ser una relación interesante.
Sea	DistCategory	La mayoría de las monedas a una distancia media provienen de cecas en el Mar de Mármara. Las monedas que fueron fabricadas lejos tienen la mayor proporción de monedas que provienen de ciudades no costales. De cualquier manera, no es una relación útil, ya que ambos se derivan del atributo ceca.
Sea	Metal	La mayoría de las monedas de oro provienen de ciudades no costales y de cecas en el Mar de Mármara. Algunas pocas fueron fabricadas en el Mar Egeo y el Mar Mediterráneo. Puede ser una relación útil.
Sea	DirRome	No parece haber una relación significativa entre los atributos.
DistCategory	Min/Max	Parece que hay diferencias entre las cecas lejanas y pequeñas variaciones en las demás. Podría ser útil en un modelo.
DistCategory	DistAlex	El atributo cualitativo se deriva del cuantitativo, entonces no sería útil en un modelo.
DistCategory	Period	Parece que sí existe relación entre la distancia de la ceca de Alejandría y el periodo en el que se fabricó la moneda. Podría ser útil en un modelo.

DistCategory	Metal	La mayoría de las monedas de oro fueron producidas en cecas cercanas y a distancia media. Podría ser una relación útil.
Metal	Min/Max	Las monedas de oro parecen haber sido fabricadas en su mayoría entre los años 350 y 380, mientras que las de bronce tienen un rango mucho más amplio. Podría ser una relación útil.
Metal	DistAlex	Las monedas de oro fueron fabricadas en su mayoría en cecas entre 1000 y 2000 km de Alejandría, mientras que las de bronce cubren un rango mucho más amplio. Puede ser una relación útil.
Metal	TimeAlex	La relación es similar a la relación entre Metal y DistAlex. Cualquiera de las dos relaciones podría ser útil en un modelo.
Metal	Period	La mayoría de las monedas de oro fueron fabricadas en los periodos 4 y 5. Podría ser una relación útil.
DirRome	Min/Max	Las cecas del oeste parecen haber fabricado monedas, en promedio, más temprano que las que fueron fabricadas en el este. Podría ser una relación útil.
DirRome	DistAlex	El gráfico muestra que las cecas del este están más cerca de Alejandría que las del oeste. No sería útil porque ambos atributos se derivan del atributo mint.
DirRome	TimeAlex	No sería útil porque ambos atributos de derivan del atributo mint.
DirRome	Period	La cantidad de monedas producidas al oeste de Roma parece ser mayor en los primeros periodos. Podría ser una relación útil.
DirRome	DistCategory	Se llega a la misma conclusión que el atributo DistAlex.
DirRome	Metal	La mayoría de las monedas de oro provienen del este, Podría ser una relación útil si se utilizan otros atributos en el modelo.
Period	DistAlex	La distancia de la ceca de Alejandría parece ser mayor en los primeros dos periodos. Podría ser útil.
Period	TimeAlex	Al igual que la distancia, el tiempo de viaje de la ceca hasta Alejandría también parece ser mayor en los primeros dos periodos. Podría ser una relación útil.
Min/Max	DistAlex	No se puede ver ningún patrón en la relación entre la distancia de la ceca de Alejandría y el año en que se fabricó la moneda, pero podría llegar a ser útil.
Min/Max	TimeAlex	Tampoco se encuentra un patrón en la relación entre el tiempo de viaje de la ceca hasta Alejandría y el año en que se fabricó la moneda, pero de igual manera podría llegar a ser útil.

Tabla 2: Análisis de relación entre atributos

Atributos que podrían ser utilizados para crear modelos para cada atributo, según análisis:

- **mint**: Min, max, denom, period, hoard, capital, metal.

- **min**: Mint, denom, hoard, port, sea, distCategory, metal, dirRome, distAlex, timeAlex.
- **max**: Mint, denom, hoard, port, sea, distCategory, metal, dirRome, distAlex, timeAlex.
- **denom**: Mint, min, max, period, hoard, sea, distCategory, dirRome.
- **hoard**: Mint, min, max, denom, distAlex, timeAlex, period, port, sea, distCategory, metal.
- **distAlex**: Min, max, hoard, capital, metal.
- **timeAlex**: Min, max, hoard, capital, metal.
- **port**: Min, max, period, hoard, capital, metal.
- **sea**: Min, max, denom, hoard, capital, period, metal.
- **distCategory**: Min, max, denom, hoard, capital, metal.
- **dirRome**: Min, max, denom, capital, metal, period.
- **metal**: Mint, min, max, hoard, distAlex, timeAlex, distCategory, port, sea, period, dirRome.
- **capital**: Mint, distAlex, timeAlex, port, sea, distCategory, dirRome.
- **period**: Mint, denom, hoard, capital, sea, metal, dirRome.

Observaciones acerca de valores de atributos:

Mint: El atributo mint no se encuentra balanceado. Hay cecas como mediolanum, ost, y sir que tienen menos de 10 observaciones. Otras cecas tienen más de mil observaciones. Es difícil que las cecas con pocas observaciones den buenos resultados, pero se podría intentar balancear el resto de las cecas para intentar conseguir mejores resultados.

Denom: La gran mayoría de las denominaciones, 74.2%, son desconocidas. Las denominaciones que sí son conocidas están muy desbalanceadas. Por esta razón, se decidió no utilizar la variable denom. Podría utilizarse en el futuro si se obtuviera la denominación de las monedas.

Hoard: Es otro atributo con observaciones muy desbalanceadas. La cantidad de observaciones por hoard va de 1 hasta 4174. También se podría intentar balancear la cantidad de observaciones por hoard, pero probablemente no sea útil.

Capital: Está desbalanceado, pero los posibles valores tienen suficientes observaciones como para crear un conjunto balanceado.

Metal: La cantidad de monedas de oro es mucho menor a las de bronce. Según la doctoranda Soto, las monedas de oro y las monedas de bronce generalmente son estudiadas por separado, porque sus características y sus usos son muy diferentes. Se puede considerar realizar los modelos juntos o por separado, según los resultados.

Port/Sea: Contienen algunos valores desconocidos que provienen de cecas donde no se sabe si eran costeras o no: aegy, eastmint, y westmint. También tiene cierto desbalance, pero se podría balancear para intentar obtener mejores resultados.

DirRome: Sus observaciones también están desbalanceadas, pero se podría crear un conjunto balanceado.

Period: Las observaciones por periodo también están bastante desbalanceadas, pero se puede intentar balancearlas para obtener mejores resultados.

Atributos de distancia (DistCategory, DistAlex, y TimeAlex): Contienen algunos valores desconocidos que provienen de cecas que se encuentran a distancias desconocidas (eastmint y westmint).

Mint

Para predecir el atributo mint, se consideraron las variables min, max, period, capital y metal.

Usando el conjunto de entrenamiento sin ningún tipo de balanceo con los atributos min, max, period y metal, el modelo de regresión logística dio una exactitud de 22.22%. Al balancear la cantidad de observaciones por mint en el conjunto de entrenamiento, el mismo algoritmo dio una exactitud aún más baja, pero la exactitud por clase aumentó en promedio (ver Tabla 3 y Tabla 4).

	aegy	Ale	ant	aqu	are	con	eastmint	her	kyz	lug	mediolanum
aegy	15	34	6	2	3	19	0	0	5	0	0
ale	113	141	37	12	7	64	15	17	20	7	0
ant	21	40	42	38	6	32	20	31	38	19	3
aqu	0	3	10	15	0	3	1	9	6	8	0
are	1	1	0	1	0	1	1	0	0	0	0
con	3	4	2	0	2	10	4	0	2	2	0
eastmint	0	1	1	0	2	2	7	0	0	0	0
her	2	6	13	19	0	0	4	23	9	4	0
kyz	6	15	13	11	2	5	3	8	11	7	0
lug	0	0	2	1	0	0	1	0	1	0	0
mediolanum	0	0	0	0	0	0	0	0	0	0	0
nio	7	4	13	1	4	4	3	2	3	1	0
ost	0	0	0	0	0	0	0	0	0	0	0
rom	4	10	19	20	5	13	3	17	16	8	0
sir	0	0	1	0	0	1	0	0	0	0	0
sis	0	3	3	9	0	0	1	6	3	11	0
the	2	4	9	12	0	2	3	10	5	0	0
tic	0	1	3	3	0	0	2	3	2	1	0
tre	0	0	0	1	0	0	0	3	0	1	0
westmint	0	1	1	0	2	1	13	0	1	0	0
car	0	0	2	7	0	0	1	4	4	5	0
ser	0	2	3	3	0	0	0	3	2	2	0

Tabla 3: Matriz de confusión al utilizar modelo de regresión logística, creado con un conjunto de datos de entrenamiento balanceado, para predecir el atributo mint utilizando min, max, period, y metal (primera parte)

	Nio	ost	rom	sir	Sis	The	tic	tre	westmint	car	ser	Exactitud por clase
aegy	0	0	7	0	0	1	0	1	1	0	0	15.96
ale	36	0	20	5	18	21	3	8	5	6	18	24.61
ant	41	1	32	3	30	28	19	10	1	12	31	8.43

aqu	2	0	9	0	18	11	2	5	2	5	6	13.04
are	1	0	2	0	0	0	0	0	1	0	0	0
con	13	0	1	1	1	2	0	0	3	0	0	20
eastmint	3	0	0	0	0	1	0	0	8	0	1	26.92
her	11	1	12	0	12	13	14	14	0	16	22	11.79
kyz	13	0	7	1	8	15	11	6	2	15	16	6.29
lug	0	0	1	0	4	1	2	2	0	0	0	0
mediolanum	0	0	0	0	0	0	0	0	0	0	0	N/A
nio	17	0	7	1	5	6	1	1	0	2	15	17.53
ost	0	1	0	0	0	0	0	0	0	0	0	100
rom	11	0	17	1	18	11	9	5	3	18	20	7.46
Sir	0	0	0	1	0	0	0	0	0	0	0	33.33
Sis	0	0	2	0	11	5	3	6	0	3	3	15.94
The	3	1	7	0	4	10	14	4	0	6	6	9.8
Tic	1	0	9	0	3	4	19	3	0	16	3	26.03
Tre	0	1	1	0	0	1	1	1	0	2	0	8.33
westmint	0	0	0	0	0	0	0	1	10	0	0	33.33
Car	2	0	5	0	4	7	31	3	0	27	6	25
Ser	4	0	2	0	1	5	2	2	0	6	10	21.28
Exactitud	15.4											

Tabla 4: Matriz de confusión al utilizar modelo de regresión logística, creado con un conjunto de datos de entrenamiento balanceado, para predecir el atributo mint utilizando min, max, period, y metal (segunda parte)

Al utilizar árboles de decisión, los modelos mejoraron un poco. La exactitud del modelo creado utilizando min, max, y metal fue de 51.15%. En promedio, la exactitud por clase fue de 24.48%. Cuando se utilizó un conjunto de entrenamiento con el atributo mint balanceado, la exactitud del modelo fue de 43.17%, pero la exactitud por clase subió a 44.33% (Tabla 5 y Tabla 6).

	Aegy	ale	ant	aqu	are	con	eastmint	her	kyz	lug	mediolanum
aegy	20	60	0	0	0	14	0	0	0	0	0
ale	32	369	15	12	0	39	0	9	0	27	0
ant	4	106	92	0	0	16	20	44	0	19	0
aqu	0	3	8	12	0	3	0	16	0	28	0
are	1	0	0	0	0	4	0	0	0	0	0
con	3	13	1	0	0	28	0	0	0	0	0
eastmint	0	3	0	0	0	0	3	0	0	0	0
her	0	4	0	0	0	1	0	150	0	0	0
kyz	1	31	2	0	0	11	0	22	64	0	0

lug	0	0	0	0	0	0	0	0	0	9	0
mediolanum	0	0	0	0	0	0	0	0	0	0	0
nio	2	12	0	0	0	13	0	0	0	0	0
ost	0	0	0	0	0	0	0	0	0	0	0
rom	5	18	3	0	0	23	9	29	0	0	0
sir	0	0	0	0	0	3	0	0	0	0	0
sis	0	0	0	6	0	0	1	0	0	25	0
the	0	3	0	0	0	6	0	0	0	0	0
tic	0	0	6	0	0	0	6	6	0	0	0
tre	0	0	0	0	0	0	0	2	0	0	0
westmint	0	2	0	0	0	2	0	0	0	0	0
car	0	0	0	0	0	0	0	6	0	7	0
ser	0	0	0	0	0	0	0	0	0	0	0

Tabla 5: Matriz de confusión al utilizar modelo de árbol de decisión, creado con un conjunto de datos de entrenamiento balanceado, para predecir el atributo mint utilizando min, max, y metal (primera parte)

	Nio	ost	rom	sir	Sis	the	tic	tre	westmint	Car	Ser	Exactitud por clase
aegy	0	0	0	0	0	0	0	0	0	0	0	21.28
ale	33	17	0	0	0	9	0	0	8	0	3	64.4
ant	37	4	0	0	30	94	0	0	8	0	24	18.47
aqu	1	0	0	0	9	11	0	0	2	0	22	10.43
are	1	0	0	0	0	0	0	0	3	0	0	0
con	0	0	0	0	0	0	0	0	5	0	0	56
eastmint	0	0	0	0	0	0	0	0	20	0	0	11.54
her	4	2	0	0	0	0	0	0	0	0	34	76.92
kyz	6	6	0	0	0	0	0	0	4	0	28	36.57
lug	2	0	0	0	1	0	0	0	1	0	2	60
mediolanum	0	0	0	0	0	0	0	0	0	0	0	N/A
nio	67	0	0	0	0	0	0	0	0	0	3	69.07
ost	0	1	0	0	0	0	0	0	0	0	0	100
rom	5	0	0	0	24	68	0	0	7	0	37	0
sir	0	0	0	0	0	0	0	0	0	0	0	0
sis	6	3	0	0	23	0	0	0	0	0	5	33.33
the	15	2	0	0	0	76	0	0	0	0	0	74.51
tic	0	8	0	0	0	5	41	0	0	0	1	56.16
tre	0	1	0	0	0	6	0	0	0	0	3	0
westmint	0	0	0	0	0	0	0	0	26	0	0	86.67
car	2	0	0	0	0	9	0	0	0	60	24	55.56
ser	0	0	0	0	0	0	0	0	0	0	47	100

Exactitud	43.17

Tabla 6: Matriz de confusión al utilizar modelo de árbol de decisión, creado con un conjunto de datos de entrenamiento balanceado, para predecir el atributo mint utilizando min, max, y metal (segunda parte)

Los bosques aleatorios dieron resultados peores que los árboles de decisión, ya sea utilizando conjuntos de entrenamiento balanceados o desbalanceados. La exactitud global nunca llegó a ser mayor que la del árbol de decisión y la exactitud por clase promedio llegó hasta 30.03% en el mejor de los casos.

Las redes neuronales no dieron buenos resultados. Aunque el conjunto de prueba contenía monedas de varias cecas, las redes neuronales siempre predecían la misma ceca (Tabla 7). Se utilizaron los atributos min, max, period, capital y metal y una sola capa. Se intentó con diferentes cantidades de nodos, pero no cambió el comportamiento de la red. Posiblemente, con más capas se obtendrían mejores resultados, pero el tiempo requerido para entrenar un modelo con más capas y más nodos sería muy alto y no se tiene evidencia que la red neuronal sea buena prediciendo el atributo mint con las pruebas realizadas hasta el momento.

	Aegy
aegy	94
ale	573
ant	498
aqu	115
are	9
con	50
eastmint	26
her	195
kyz	175
lug	15
mediolanum	0
nio	97
ost	1
rom	228
sir	3
sis	69
the	102
tic	73
tre	12

westmint	30
car	108
ser	47

Tabla 7: Matriz de confusión al utilizar modelo de red neuronal para predecir el atributo mint utilizando min, max, period, capital y metal

Ya que el árbol de decisión tuvo los mejores resultados por un gran margen, no sería útil crear un sistema de votación.

Al usar los criterios de evaluación establecidos, el mejor modelo para predecir mint obtuvo solamente 12 puntos de 100 (Tabla 8).

Exactitud total	0
Porcentaje mínimo de aciertos por atributo	0
Complejidad	5
Cantidad de variables independientes	7
Total	**12**

Tabla 8: Puntaje de modelo de árbol de decisión para predecir el atributo mint

Con las pruebas realizadas hasta el momento, no parece ser posible predecir el atributo mint con las características que se conocen de las monedas.

Sea

Para predecir el atributo sea, se consideraron las variables min, max, period, capital y metal.

Usando el conjunto de entrenamiento sin ningún tipo de balanceo, el modelo de regresión logística con las variables independientes min, max, period y metal, dio una exactitud de 38.82% y una exactitud promedio por clase de 27.05% (Tabla 9).

	aegean	Black	marmara	mediterranean	none
aegean	8	9	19	21	45
black	7	25	20	40	103
marmara	19	22	66	99	116
mediterranean	21	54	107	395	229
none	44	93	137	245	426

Exactitud	38.82

Tabla 9: Matriz de confusión al utilizar modelo de regresión logística para predecir el atributo sea utilizando min, max, period y metal

Al balancear la cantidad de observaciones por sea en el conjunto de entrenamiento, el mismo algoritmo dio una exactitud de 29.45% y la exactitud por clase promedio aumentó a 28.95% (Tabla 10).

	aegean	Black	marmara	mediterranean	none
aegean	28	23	18	14	19
black	42	52	35	22	44
marmara	62	50	97	56	57
mediterranean	122	105	146	295	138
none	198	187	170	164	226

Exactitud	29.45

Tabla 10: Matriz de confusión al utilizar modelo de regresión logística, creado con un conjunto de datos de entrenamiento balanceado, para predecir el atributo sea utilizando min, max, period y metal

Ninguno de los modelos se puede considerar bueno.

Al utilizar árboles de decisión, la exactitud del modelo creado utilizando min, max, y metal fue de 69.37%. En promedio, la exactitud por clase fue de 58.66% (Tabla 11).

	aegean	Black	marmara	mediterranean	none
aegean	24	0	19	4	55
black	0	119	4	37	35
marmara	0	0	188	92	42
mediterranean	10	25	59	625	87
none	11	14	48	184	688

Exactitud	69.37

Tabla 11: Matriz de confusión al utilizar modelo de árbol de decisión para predecir el atributo sea utilizando min, max y metal

Cuando se utilizó un conjunto de entrenamiento con el atributo sea balanceado, la exactitud del modelo fue de 55.32%, pero la exactitud por clase subió a 68.72% (Tabla 12).

	aegean	Black	marmara	mediterranean	none
aegean	91	0	6	4	1
black	0	181	6	4	4
marmara	35	22	208	43	14
mediterranean	86	73	114	496	37
none	195	111	196	108	335

Exactitud	55.32

Tabla 12: Matriz de confusión al utilizar modelo de árbol de decisión, creado con un conjunto de datos de entrenamiento balanceado, para predecir el atributo sea utilizando min, max, y metal

El modelo creado con el conjunto balanceado logró predecir aegean y black bastante bien, mientras que el conjunto de entrenamiento original logró mejores resultados con mediterranean y none.

El bosque aleatorio con el conjunto de entrenamiento original dio mejores resultados prediciendo la clase “none”, pero en general obtuvo peores resultados (ver Tabla 13).

	aegean	Black	marmara	mediterranean	none
aegean	0	0	15	8	79
black	0	90	4	6	95
marmara	0	0	66	95	161
mediterranean	0	25	34	460	287
none	0	10	10	154	771

Exactitud	58.52

Tabla 13: Matriz de confusión al utilizar modelo de bosque aleatorio para predecir el atributo sea utilizando min, max y metal

Las redes neuronales nuevamente solo predijeron un valor para todas las observaciones (Tabla 14).

	Mediterranean
aegean	102
black	195
marmara	322
mediterranean	806

none	945

Exactitud	34.01

Tabla 14: Matriz de confusión al utilizar modelo de red neuronal para predecir el atributo sea utilizando min, max y metal

Nuevamente, el árbol de decisión parece ser el mejor modelo, pero utilizando los criterios de evaluación establecidos, solamente obtuvo 12 puntos de 100 (Tabla 15).

Exactitud total	0
Porcentaje mínimo de aciertos por atributo	0
Complejidad	5
Cantidad de variables independientes	7
Total	12

Tabla 15: Puntaje de modelo de árbol de decisión para predecir el atributo sea

DistCategory

Para predecir el atributo distCategory, se consideraron las variables min, max, period, capital y metal.

El modelo de regresión logística, sin balancear, dio una exactitud de 52.44%, con exactitudes por clase desde 16.67% hasta 62.27% (Tabla 16).

	Far	mid-range	Near	Exactitud por clase
Far	38	105	85	16.67
midrange	94	465	410	47.99
near	92	386	789	62.27

Exactitud	52.44

Tabla 16: Matriz de confusión al utilizar modelo de regresión logística para predecir el atributo distCategory utilizando min, max, period, capital y metal

Un modelo creado usando un conjunto con el atributo distCategory balanceado dio una menor exactitud en general (46.31%), pero exactitud por clase un poco más balanceada (ver Tabla 17).

	Far	mid-range	Near	Exactitud por clase
far	92	82	54	40.35

midrange	323	350	296	36.12
near	241	327	699	55.17

Exactitud	46.31

Tabla 17: Matriz de confusión al utilizar modelo de regresión logística, creado con un conjunto de datos de entrenamiento balanceado, para predecir el atributo distCategory utilizando min, max, period, capital y metal

Al utilizar un modelo de árbol de decisión con los atributos min, max, period, y metal y un CP (complexity parameter, o parámetro de complejidad) de 0.001, sin balancear, dio una exactitud de 77.07%, pero un promedio de exactitud por clase de 65.53%. Pudo predecir correctamente un 91.79% de las observaciones "near", pero solamente un 37.72% de observaciones "far" (Tabla 18).

	Far	mid-range	near	Exactitud por clase
Far	86	102	40	37.72
Midrange	9	650	310	67.08
Near	5	99	1163	91.79

Exactitud	77.07

Tabla 18: Matriz de confusión al utilizar modelo de árbol de decisión con un CP 0.001 para predecir el atributo distCategory utilizando min, max, period y metal

Un modelo entrenado con un conjunto balanceado dio una exactitud promedio por clase de 68.91%, con 85.24% de las observaciones "near" identificadas correctamente, 85.09% de las "far", y 43.76% de las "mid-range". La exactitud global fue de 68.91% (Tabla 19).

	Far	mid-range	near	Exactitud por clase
Far	194	15	19	85.09
Midrange	265	424	280	43.76
Near	103	84	1080	85.24

Exactitud	68.91

Tabla 19: Matriz de confusión al utilizar modelo de árbol de decisión con un CP de 0.001, creado con un conjunto de datos de entrenamiento balanceado, para predecir el atributo distCategory utilizando min, max, period y metal

Aunque este árbol de decisión parece ser mejor que los modelos creados anteriormente, el modelo parece estarse sobre ajustando a los datos ("overfitting"). Si se baja el CP a 0.01 para evitar que esto suceda y se utiliza el conjunto de entrenamiento balanceado, el resultado empeora porque el modelo ya no es tan específico, pero se esperaría que solucione el sobre ajuste (Tabla 20). El cambio no parece haber tenido este efecto, ya que al visualizar el árbol de decisión, se puede notar que la probabilidad de "overfitting" es muy alta (Figura 6).

	Far	mid-range	near	Exactitud por clase
far	201	15	12	88.16
midrange	361	410	198	42.31
near	284	161	822	64.88

Exactitud	58.16

Tabla 20: Matriz de confusión al utilizar modelo de árbol de decisión con un CP de 0.01, creado con un conjunto de datos de entrenamiento balanceado, para predecir el atributo distCategory utilizando min, max, period y metal

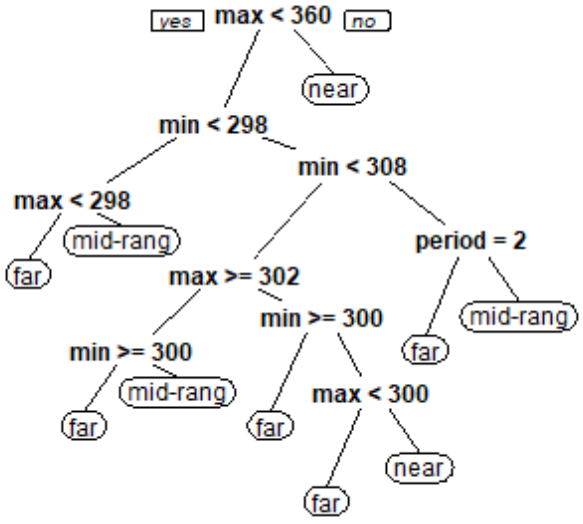


Figura 6: Árbol de decisión para predecir distCategory utilizando un CP de 0.01 (elaboración propia)

En la mayoría de los casos, el modelo está basando sus decisiones en el año mínimo o máximo de fabricación de la moneda. Esto podría indicar fluctuaciones de producción según cada área, pero por la cercanía en años, lo más probable es que no sean patrones reales. El modelo se puede intentar probar con otros datos más adelante para verificar si este es el caso.

Si se utiliza solamente una variable año, ya sea min o max, mejora un poco el problema de overfitting, aunque los resultados con el conjunto de prueba no parecen ser muy buenos (Tabla 21).

	Far	mid-range	near	Exactitud por clase
far	167	55	6	73.25
midrange	338	469	162	48.4
near	327	242	698	55.09

Exactitud	54.14

Tabla 21: Matriz de confusión al utilizar modelo de árbol de decisión con un CP de 0.01, creado con un conjunto de datos de entrenamiento balanceado, para predecir el atributo distCategory utilizando max, period y metal

Sin embargo, todavía se da un poco de overfitting. Este árbol de decisión, por ejemplo, identificaría una observación del año 295 como “far”, del 297 como “mid-range”, y del 300 como “far”. Es difícil pensar que esto realmente sea evidencia de factores económicos de la época al ser rangos de años tan cortos (Figura 7).

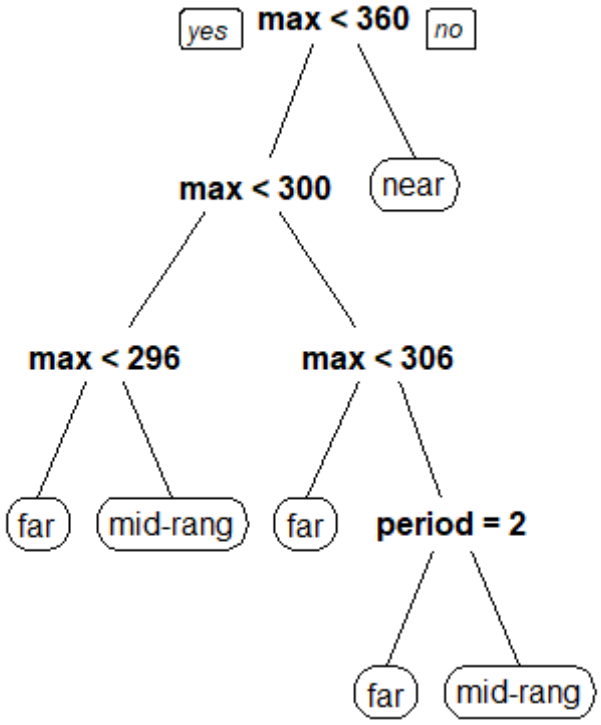


Figura 7: Árbol de decisión para predecir distCategory utilizando un CP de 0.01, sin el atributo min (elaboración propia)

El bosque aleatorio utilizando min, max, period y metal mejoró un poco la predicción de far y mid-range, y empeoró un poco la predicción de near. La exactitud global fue un poco peor que la del árbol de decisión (Tabla 22).

	Far	mid-range	near	Exactitud por clase
far	206	11	11	90.35
midrange	329	451	189	46.54
near	193	111	963	76.01

Exactitud	65.75

Tabla 22: Matriz de confusión al utilizar modelo de bosque aleatorio, creado con un conjunto de datos de entrenamiento balanceado, para predecir el atributo distCategory utilizando min, max, period y metal

La red neuronal nuevamente solo predijo una clase (Tabla 23).

	Far	mid-range	near	Exactitud por clase
far	0	0	228	0
midrange	0	0	969	0
near	0	0	1267	100

Exactitud	51.42

Tabla 23: Matriz de confusión al utilizar modelo de red neuronal, creado con un conjunto de datos de entrenamiento balanceado, para predecir el atributo distCategory utilizando min, max, period y metal

Al usar los criterios de evaluación establecidos, el mejor modelo para predecir distCategory fue el árbol de decisión, con 62 puntos de 100 (Tabla 24). Por los problemas de overfitting, es importante probar los modelos con otro conjunto y ver si los resultados son similares.

Exactitud total	50
Porcentaje mínimo de aciertos por atributo	0
Complejidad	5
Cantidad de variables independientes	7
Total	62

Tabla 24: Puntaje de modelo de árbol de decisión para predecir el atributo distCategory

Port

Para predecir el atributo port, se consideraron las variables min, max, period, capital, y metal.

El mejor modelo de regresión logística se logró utilizando los atributos period y metal y un conjunto de entrenamiento balanceado. La exactitud total fue de 56.84%, con 78.84% de aciertos en monedas fabricadas en ciudades no costales y 42.25% de aciertos en monedas fabricadas en puertos (Tabla 25).

	0	1	Exactitud por clase
0	745	200	78.84
1	823	602	42.25

Exactitud	56.84

Tabla 25: Matriz de confusión al utilizar modelo de regresión logística, creado con un conjunto de datos de entrenamiento balanceado, para predecir el atributo port utilizando period y metal

Un árbol de decisión que utilice solo min y max y el conjunto de datos de entrenamiento balanceado puede predecir correctamente 74.18% de las observaciones (Tabla 26).

	0	1	Exactitud por clase
0	587	358	62.12
1	254	1171	82.18

Exactitud	74.18

Tabla 26: Matriz de confusión al utilizar modelo de árbol de decisión, creado con un conjunto de datos de entrenamiento balanceado, para predecir el atributo port utilizando min y max

Si se le agrega el atributo hoard al modelo, mejoran un poco sus predicciones sobre monedas fabricadas en ciudades no costales (Tabla 27).

	0	1	Exactitud por clase
0	638	307	67.51
1	275	1150	80.7

Exactitud	75.44

Tabla 27: Matriz de confusión al utilizar modelo de árbol de decisión, creado con un conjunto de datos de entrenamiento balanceado, para predecir el atributo port utilizando period, hoard y metal

Los modelos parecen presentar el mismo problema de sobreajuste que los modelos de DistCategory.

El bosque aleatorio dio resultados un poco más balanceados. La exactitud por clase de ambas está alrededor de 75% (Tabla 28).

	0	1	Exactitud por clase
0	715	230	75.66
1	337	1088	76.35

Exactitud	76.08

Tabla 28: Matriz de confusión al utilizar modelo de bosque aleatorio, creado con un conjunto de datos de entrenamiento balanceado, para predecir el atributo port utilizando period, hoard y metal

Una red neuronal que utiliza period y metal para predecir port, con dos capas, una de cinco nodos y otra de dos, no brindó muy buenos resultados (Tabla 29). Aun así, fue el mejor modelo de red neuronal para predecir el atributo port. Es posible que con otra cantidad de nodos y capas podría tener mejores resultados.

	0	1	Exactitud por clase
0	926	19	97.99
1	1378	47	3.3

Exactitud	41.05

Tabla 29: Matriz de confusión al utilizar modelo de red neuronal con una capa de cinco nodos y otra de dos, creado con un conjunto de datos de entrenamiento balanceado, para predecir el atributo port utilizando period y metal

De todos los modelos, el árbol de decisión y el bosque aleatorio que utilizan min, max, y hoard, parecen ser los mejores. Según el criterio de evaluación, el bosque aleatorio tiene un puntaje levemente mayor (Tabla 30) que el del árbol de decisión, ya que cada clase tuvo una exactitud de más de 70%. Sin embargo, por los problemas de overfitting, es importante probar los modelos con otro conjunto y ver si los resultados son similares.

Exactitud total	50
Porcentaje mínimo de aciertos por atributo	10
Complejidad	3
Cantidad de variables independientes	7
Total	**70**

Tabla 30: Puntaje de modelo de bosque aleatorio para predecir el atributo port

DirRome

Para predecir el atributo dirRome, se consideraron las variables min, max, period, capital, y metal.

El modelo de regresión logística, con un conjunto de entrenamiento balanceado y los atributos min, max, metal, capital y period, dio 44.96% de exactitud, con un promedio de 41.88% de exactitud por clase (Tabla 31).

	east	rome	West	Exactitud por clase
East	943	569	532	46.14
Rome	77	82	69	35.96
West	54	86	108	43.55

Exactitud	44.96

Tabla 31: Matriz de confusión al utilizar modelo de regresión logística, creado con un conjunto de datos de entrenamiento balanceado, para predecir el atributo dirRome utilizando min, max, metal, capital y period

El árbol de decisión con los mejores resultados usó los atributos min y max y un CP de 0.001. Aun así, no fue mucho mejor que el modelo de regresión logística. La exactitud fue de 54.92% (Tabla 32). Pudo detectar correctamente casi el 90% de las monedas fabricadas en Roma, pero solo acertó con las monedas producidas al este y al oeste la mitad del tiempo. El modelo también parece estar sobre ajustado, especialmente con un CP de 0.001, pero al podarlo, el modelo mantuvo los mismos nodos.

	east	rome	West	Exactitud por clase
East	1039	955	50	50.83
Rome	17	204	7	89.47
West	5	102	141	56.85

Exactitud	54.92

Tabla 32: Matriz de confusión al utilizar modelo de árbol de decisión con un CP de 0.001, creado con un conjunto de datos de entrenamiento balanceado, para predecir el atributo dirRome utilizando min y max

El bosque aleatorio tuvo resultados levemente mejores, con una exactitud total de 64.21% y una exactitud por clase promedio de 71.39% (Tabla 33).

	east	rome	West	Exactitud por clase
East	1257	613	174	61.5
Rome	25	201	2	88.16
West	14	74	160	64.52

Exactitud	64.21

Tabla 33: Matriz de confusión al utilizar modelo de bosque aleatorio, creado con un conjunto de datos de entrenamiento balanceado, para predecir el atributo dirRome utilizando min y max

La red neuronal, utilizando min, max, capital, period y metal, una capa de cinco nodos y otra capa de dos nodos, tuvo mejores resultados que el modelo de regresión logística, pero peores que el árbol de decisión (Tabla 34). La distribución es similar a la de otros modelos, ya que es relativamente bueno encontrando las monedas fabricadas en Roma, pero no tan bueno con el resto de las monedas.

	east	rome	West	Exactitud por clase
East	1002	972	70	49.02
Rome	41	187	0	82.02
West	38	100	110	44.35

Exactitud	51.55

Tabla 34: Matriz de confusión al utilizar modelo de red neuronal con una capa de cinco nodos y otra de dos, creado con un conjunto de datos de entrenamiento balanceado, para predecir el atributo dirRome utilizando min, max, capital, period y metal

De todos los modelos, se podría decir que el bosque aleatorio que utiliza min y max es el mejor, pero solo alcanzó 18 puntos según los criterios de evaluación (Tabla 35). El resto de los modelos obtuvo puntajes aún más bajos. Ningún modelo parece lograr predecir DirRome correctamente.

Exactitud total	0
Porcentaje mínimo de aciertos por atributo	5

Complejidad	3
Cantidad de variables independientes	10
Total	**18**

Tabla 35: Puntaje de modelo de bosque aleatorio para predecir el atributo dirRome

DistAlex

Para predecir la distancia de Alejandría, se intentó utilizar regresión lineal, el conjunto de entrenamiento balanceado por ceca y los atributos min y metal. Luego, se creó un árbol de decisión usando min, max, period y metal. Utilizando este árbol de decisión, se creó un árbol podado. Por último, se intentó crear un bosque aleatorio con los mismos atributos. También se intentó con varias redes neuronales de hasta 5 capas y hasta 10 nodos por capa, pero el modelo siempre predijo 1841.206 para todas las observaciones del conjunto de prueba.

Se calculó el error de la raíz cuadrada de la media (RMSE) y el error medio absoluto (MAE) para cada uno de los modelos menos la red neuronal (Tabla 36).

Modelo	RMSE	MAE
Ingenuo	1130.03	940.64
Regresión Lineal	986.67	828.96
Árbol	836.43	642
Árbol Podado	847.06	644.63
Bosque Aleatorio	916.64	758.34

Tabla 36: Comparación de RMSE y MAE de modelos creados para predecir el atributo distAlex

En este caso, el árbol podado probablemente sería la mejor opción, pero, al comparar las predicciones con el valor real, se puede apreciar que ningún modelo es realmente bueno prediciendo la distancia en kilómetros de Alejandría (ver Figura 8).

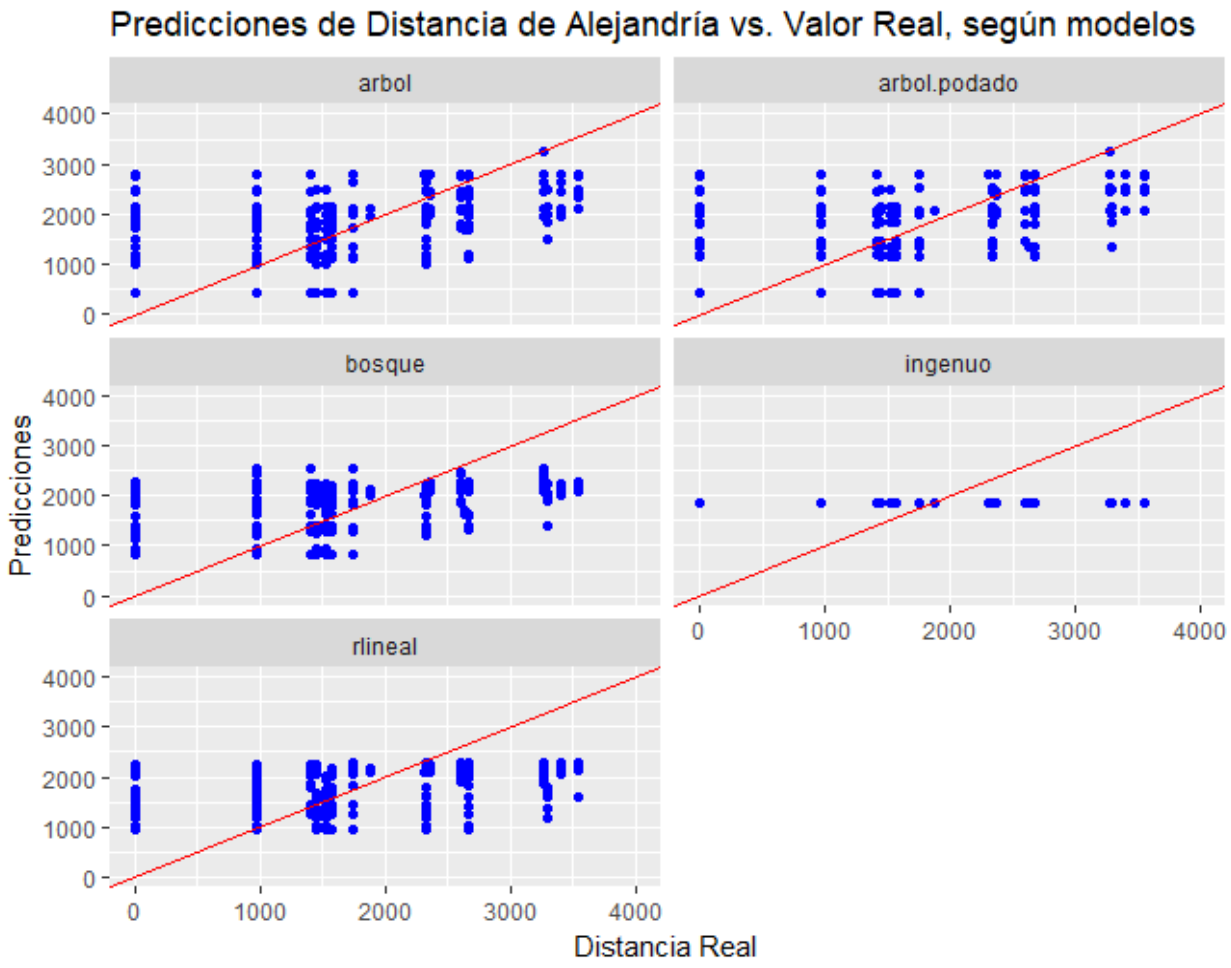


Figura 8: Gráficos de dispersión de cada modelo creado para predecir el atributo distAlex (elaboración propia)

TimeAlex

Para predecir el tiempo de viaje de la ceca hasta Alejandría, se siguió el mismo proceso que se utilizó con DistAlex, y se obtuvieron resultados similares. En este caso, la ceca que está a 50 días cambia un poco los gráficos, pero si ignoramos esa ceca, la relación es muy similar a la de DistAlex. Los modelos tampoco parecen predecir muy bien el tiempo de viaje (ver Tabla 37 y Figura 9).

Modelo	RMSE	MAE
Ingenuo	10.71	8.63
Regresión Lineal	9.51	8.03
Árbol	8.14	6.35

Árbol Podado	8.18	6.37
Bosque Aleatorio	8.93	7.45

Tabla 37: Comparación de RMSE y MAE de modelos creados para predecir el atributo timeAlex

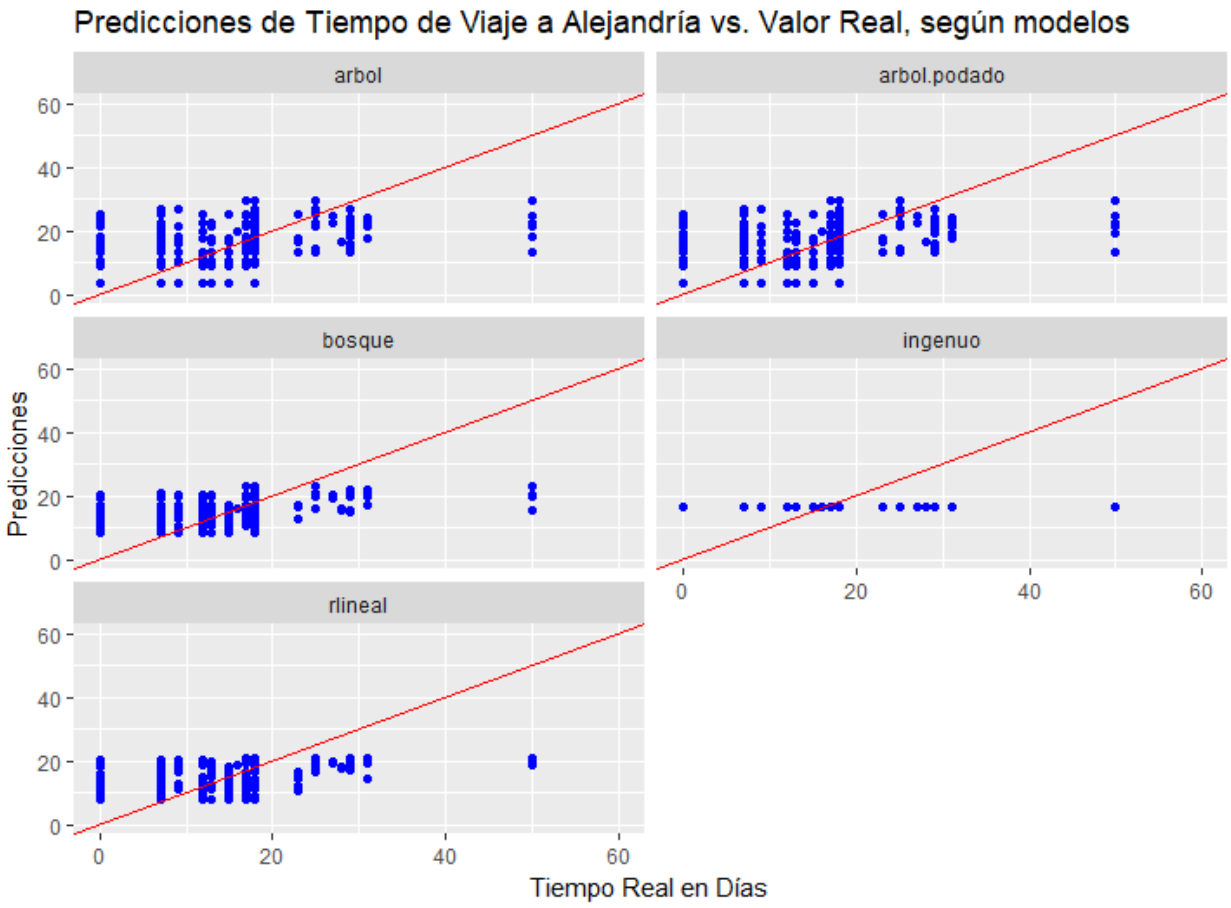


Figura 9: Gráficos de dispersión de cada modelo creado para predecir el atributo timeAlex (elaboración propia)

Min

Para predecir el año mínimo de fabricación de la moneda, se intentó usar regresión lineal, árboles de decisión, bosques aleatorios, y redes neuronales. Para el modelo de regresión lineal, se utilizaron los atributos port, metal y distAlex. Para el árbol de decisión y el bosque aleatorio, se utilizó mint, hoard, port, y distAlex. Nuevamente se creó un árbol podado a partir del árbol original. Para la red neuronal, se usaron los atributos distAlex y port. Este modelo con distAlex y port dio muy malos

resultados, prediciendo 356.50 para todas las observaciones. De todos los modelos, el árbol podado posiblemente sería la mejor opción (ver Tabla 38 y Figura 10). Un error promedio de 5 años parece ser relativamente bueno, pero se tendría que probar contra otro conjunto de prueba para verificar que no hay "overfitting".

Modelo	RMSE	MAE
Ingenuo	42.38	36.54
Regresión Lineal	60.01	48.46
Árbol	7.82	4.97
Árbol Podado	7.85	5.03
Bosque Aleatorio	12.22	9.54

Tabla 38: Comparación de RMSE y MAE de modelos creados para predecir el atributo min

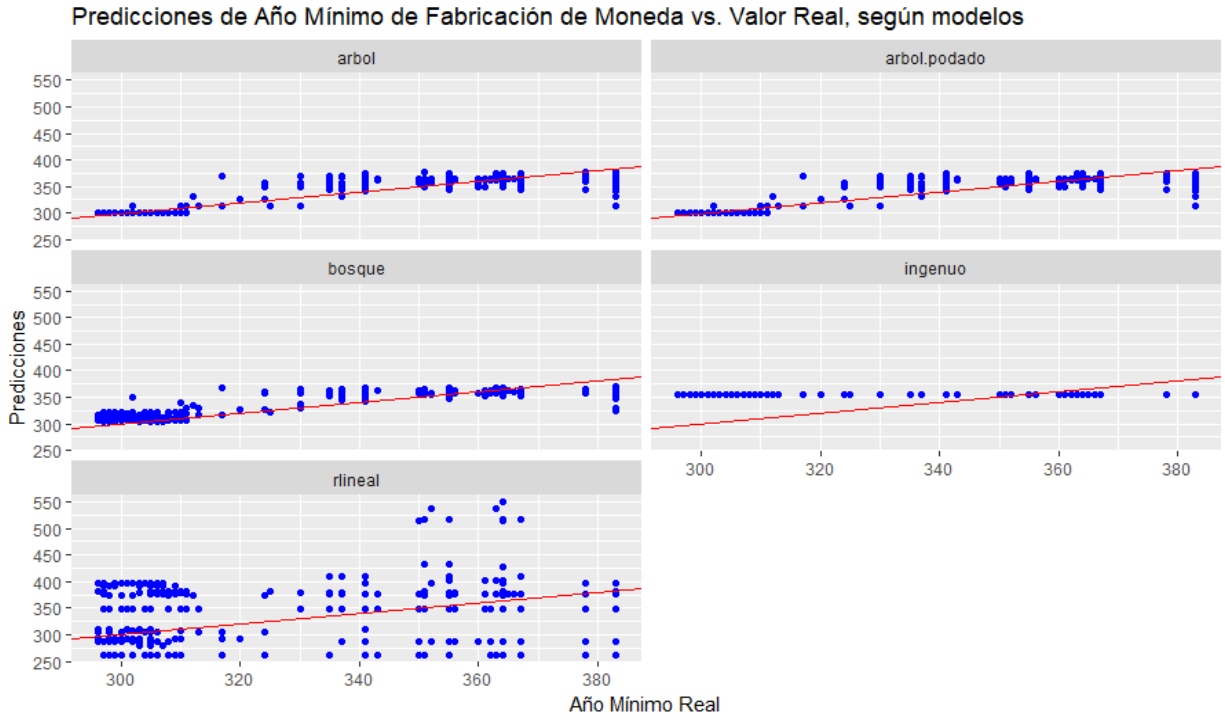


Figura 10: Gráficos de dispersión de cada modelo creado para predecir el atributo min (elaboración propia)

Max

Para predecir el año máximo de fabricación de la moneda, se usaron los mismos algoritmos que se usaron para el año mínimo. No se intentó crear una red neuronal por el mal resultado que dio calcular el año mínimo, ya que ambos atributos parecen

requerir modelos parecidos. Para el modelo de regresión lineal, se utilizaron los atributos metal y distAlex. Para el árbol de decisión y el bosque aleatorio, se utilizó dirRome, distAlex, hoard, port, sea y timeAlex. Nuevamente se creó un árbol podado a partir del árbol original, pero no parece haber podado ninguna rama. De todos los modelos, el árbol posiblemente sería la mejor opción. Nuevamente dio un error promedio de aproximadamente 5 años, lo cual es relativamente bueno (ver Tabla 39 y Figura 11).

Modelo	RMSE	MAE
Ingenuo	44.33	37.82
Regresión Lineal	39.37	33.46
Árbol	7.60	4.83
Árbol Podado	7.60	4.83
Bosque Aleatorio	8.44	5.89

Tabla 39: Comparación de RMSE y MAE de modelos creados para predecir el atributo max

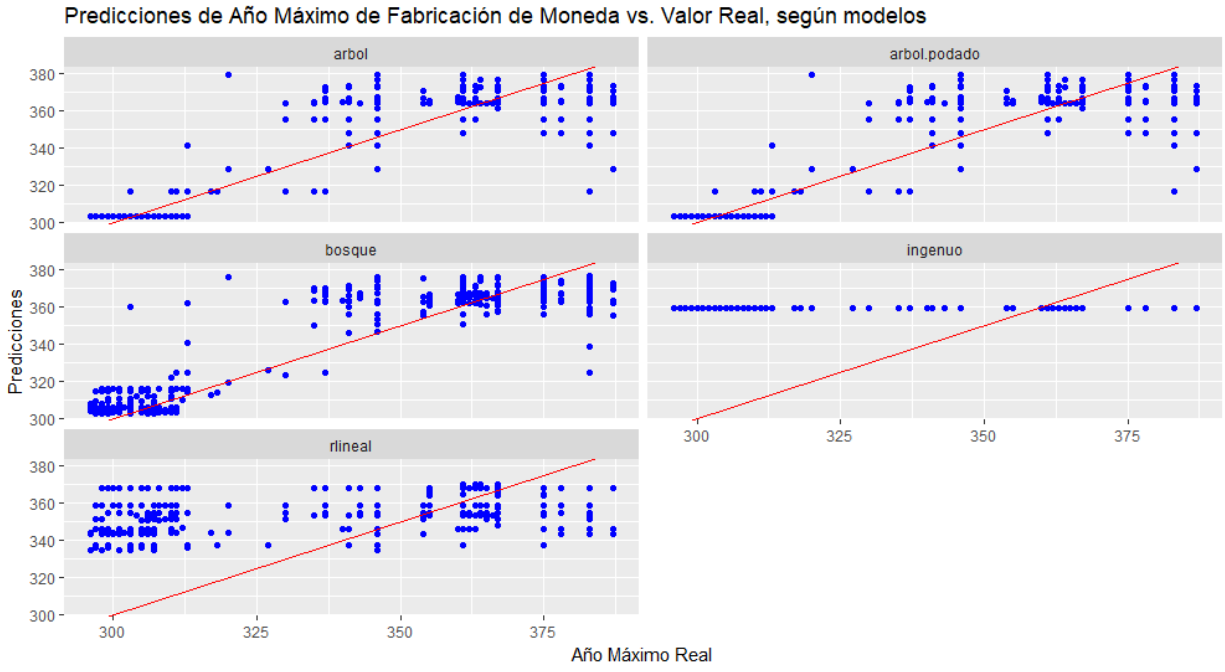


Figura 11: Gráficos de dispersión de cada modelo creado para predecir el atributo max (elaboración propia)

Period

Para predecir el periodo en el que se fabricó la moneda, se realizaron pruebas utilizándolo como atributo categórico y otras como atributo numérico. Con el atributo numérico se creó un modelo de regresión lineal con los atributos metal y distAlex, un árbol de decisión con los atributos hoard, mint y sea, el árbol de decisión podado, un bosque aleatorio con los mismos atributos y una red neuronal (ver Tabla 40 y Figura 12).

Modelo	RMSE	MAE
Ingenuo	2.13	2.02
Regresión Lineal	1.84	1.68
Árbol	0.64	0.30
Árbol Podado	0.64	0.30
Bosque Aleatorio	0.65	0.49

Tabla 40: Comparación de RMSE y MAE de modelos creados para predecir el atributo period

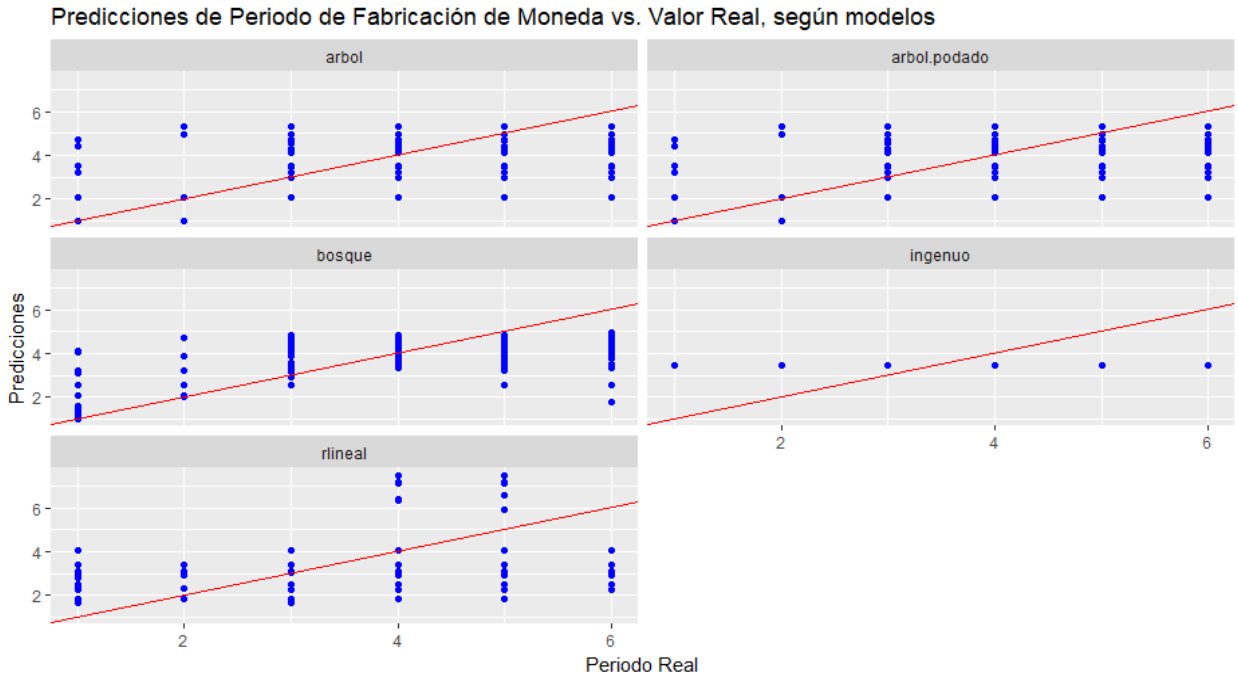


Figura 12: Gráficos de dispersión de cada modelo creado para predecir el atributo period (elaboración propia)

Según el RMSE y MAE, el árbol es el mejor modelo, pero, al tratarlo period como un atributo numérico, los modelos predicen valores continuos. Este valor puede ser útil

de todas formas, pero no está prediciendo un periodo específico. Por ejemplo, aunque el modelo puede predecir “1”, también predice “5.29” y “4.30”. Podemos intentar obtener este valor redondeando las predicciones y volver a calcular el RMSE y MAE (ver Tabla 41 y Figura 13).

Modelo	RMSE	MAE
Ingenuo	2.13	2.02
Regresión Lineal	1.76	1.59
Árbol	0.66	0.27
Árbol Podado	0.66	0.27
Bosque Aleatorio	0.71	0.41

Tabla 41: Comparación de RMSE y MAE de modelos creados para predecir el atributo period al redondear las predicciones al número entero más cercano

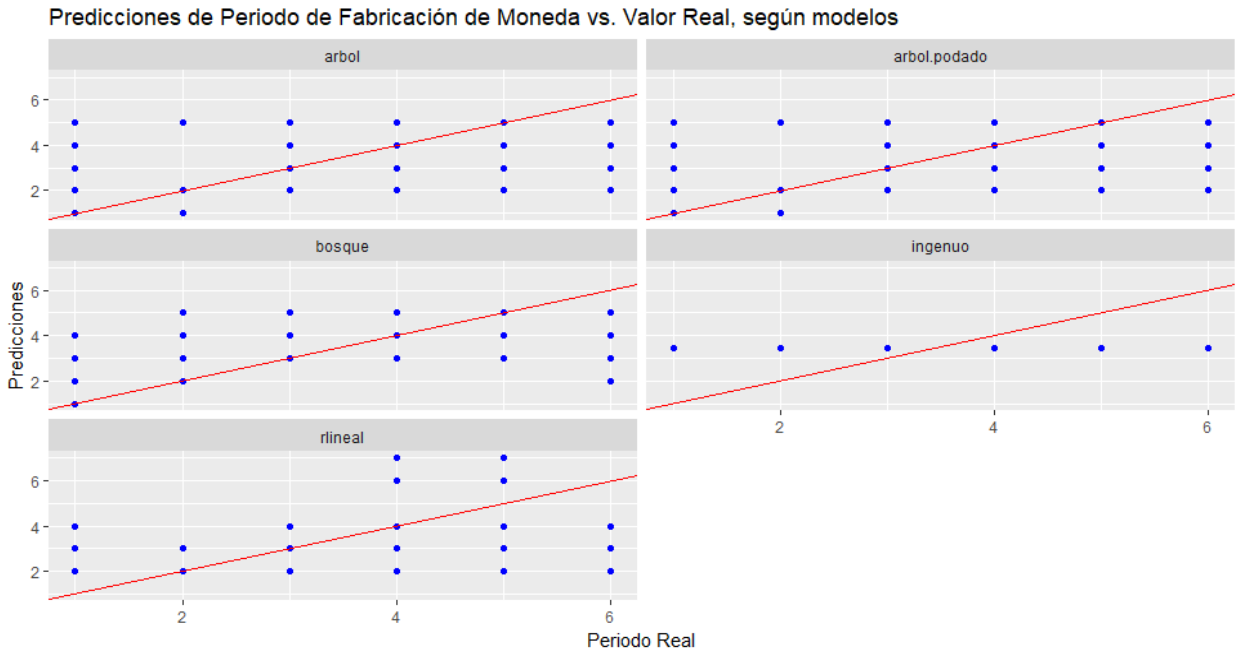


Figura 13: Gráficos de dispersión de cada modelo creado para predecir el atributo period al redondear las predicciones al número entero más cercano (elaboración propia)

El árbol de decisión parece ser el mejor modelo, pero no pudo predecir el periodo 6 (ver Tabla 42).

	1	2	3	4	5	6	Exactitud por Clase
1	1474	1	1	2	2	0	99.59

2	1	10	0	0	2	0	76.92
3	0	23	18	24	60	0	14.4
4	0	17	10	60	99	0	32.26
5	0	6	8	77	437	0	82.77
6	0	3	4	40	141	0	0
Exactitud Global:	**79.33**						

Tabla 42: Matriz de confusión al utilizar modelo de árbol de decisión para predecir el atributo period, redondeando las predicciones al número entero más cercano

Se puede realizar la misma prueba, pero redondeado las predicciones hacia arriba (ver Tabla 43 y Figura 14).

Modelo	RMSE	MAE
Ingenuo	2.13	2.02
Regresión Lineal	2.18	1.85
Árbol	0.63	0.25
Árbol Podado	0.63	0.25
Bosque Aleatorio	0.96	0.83

Tabla 43: Comparación de RMSE y MAE de modelos creados para predecir el atributo period al redondear las predicciones hacia arriba

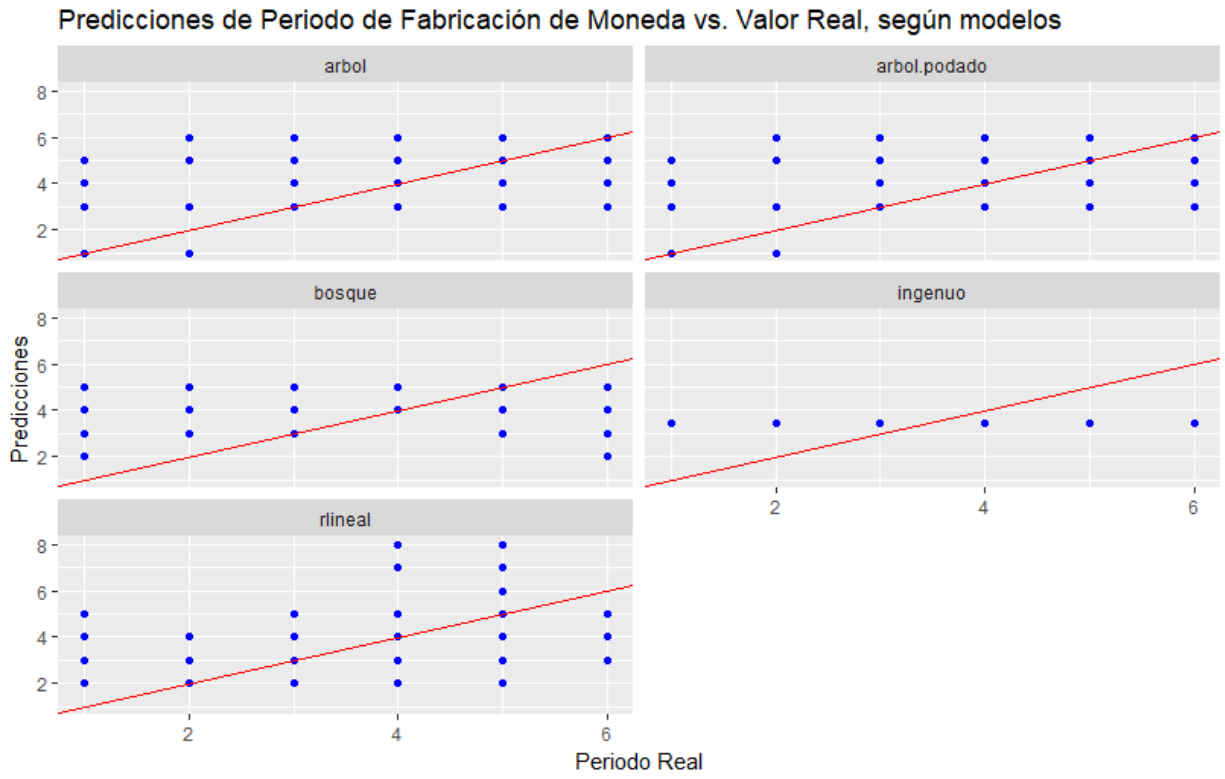


Figura 14: Gráficos de dispersión de cada modelo creado para predecir el atributo period al redondear las predicciones hacia arriba (elaboración propia)

En este caso, el árbol pudo predecir el periodo 6 un poco mejor, pero no pudo predecir el periodo 2 (ver Tabla 44).

	1	2	3	4	5	6	Exactitud por Clase
1	1474	0	1	2	3	0	99.59
2	1	0	10	0	1	1	0
3	0	0	28	15	65	17	22.4
4	0	0	19	13	143	11	6.99
5	0	0	7	9	474	38	89.77
6	0	0	4	5	131	48	25.53
Exactitud Global:	80.83						

Tabla 44: Matriz de confusión al utilizar modelo de árbol de decisión para predecir el atributo period, redondeando las predicciones hacia arriba

Por último, se puede realizar la misma prueba redondeado las predicciones hacia abajo (ver Tabla 45 y Figura 15).

Modelo	RMSE	MAE
Ingenuo	2.13	2.02
Regresión Lineal	1.68	1.47
Árbol	0.78	0.42
Árbol Podado	0.78	0.42
Bosque Aleatorio	0.80	0.41

Tabla 45: Comparación de RMSE y MAE de modelos creados para predecir el atributo period al redondear las predicciones hacia abajo

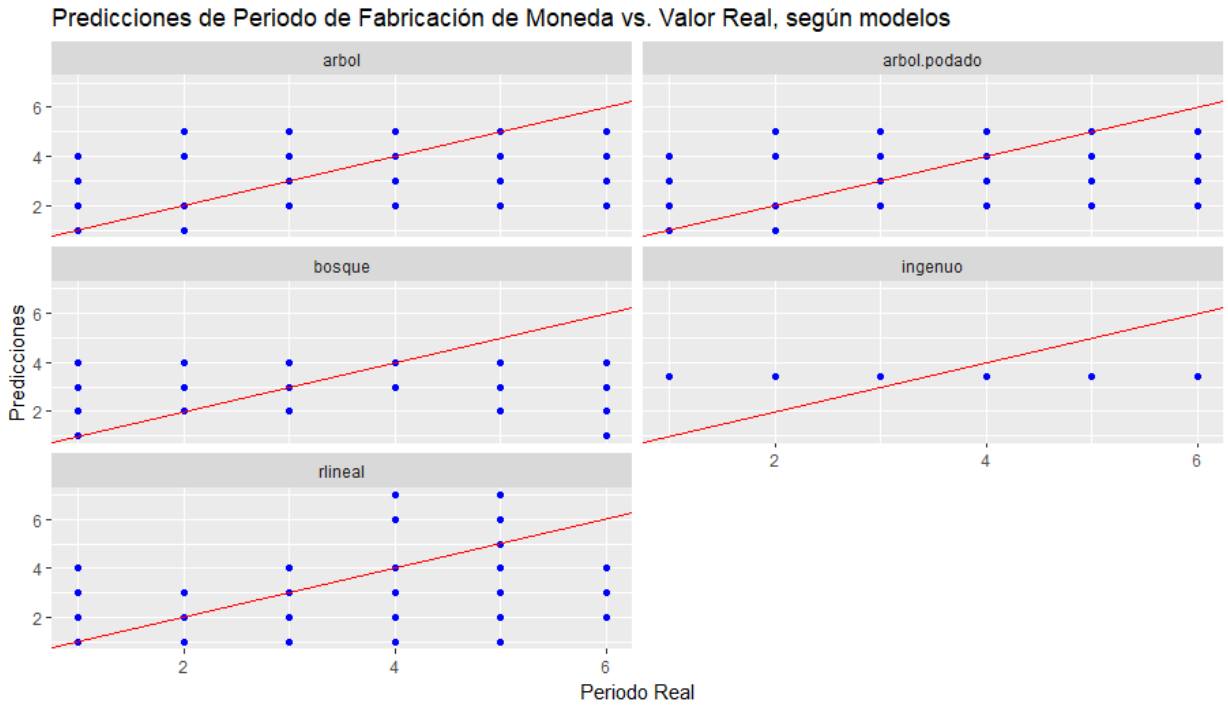


Figura 15: Gráficos de dispersión de cada modelo creado para predecir el atributo period al redondear las predicciones hacia abajo (elaboración propia)

El modelo nuevamente no pudo predecir el periodo 6 y los resultados fueron peores que ambos modelos anteriores (ver Tabla 46).

	1	2	3	4	5	6	Exactitud por Clase
1	1474	1	2	3	0	0	99.59
2	1	10	0	1	1	0	76.92
3	0	23	20	65	17	0	16
4	0	17	15	143	11	0	76.88
5	0	6	10	474	38	0	7.2
6	0	3	6	131	48	0	0
Exactitud Global:	66.87						

Tabla 46: Matriz de confusión al utilizar modelo de árbol de decisión para predecir el atributo period, redondeando las predicciones hacia abajo

Las redes neuronales produjeron malos resultados. Se obtuvo una exactitud de apenas 16.84% y solo predijo dos periodos (ver Tabla 47).

	1	2	3	4	5	6	Exactitud por Clase
1	0	0	1365	0	112	0	0
2	0	0	13	0	0	0	0

3	0	0	47	0	32	0	59.49
4	0	0	100	0	86	0	0
5	0	0	159	0	368	0	69.83
6	0	0	113	0	69	0	0
Exactitud Global:	16.84						

Tabla 47: Matriz de confusión al utilizar modelo de red neuronal para predecir el atributo period

Con el atributo categórico, se crearon modelos de regresión logística, árboles de decisión, y bosques aleatorios.

El modelo de regresión logística, utilizando los atributos mint, hoard, metal y distAlex, dio una exactitud global de 76.63%, pero la exactitud por clase de la mayoría de las clases fue bastante baja (Tabla 48).

	1	2	3	4	5	6	Exactitud por Clase
1	1474	0	2	2	1	1	99.59
2	1	9	2	1	0	0	69.23
3	46	2	29	27	5	16	23.2
4	0	0	38	75	40	33	40.32
5	1	1	52	164	267	43	50.57
6	6	2	49	42	12	77	40.96
Exactitud Global:	76.63						

Tabla 48: Matriz de confusión al utilizar modelo de regresión logística para predecir el atributo period (categórico) utilizando los atributos mint, hoard, metal y distAlex

El modelo de árbol de decisión, utilizando hoard y mint, dio resultados un poco mejores (Tabla 49).

	1	2	3	4	5	6	Exactitud por Clase
1	1474	1	2	1	0	2	99.59
2	1	10	0	1	0	1	76.92
3	0	12	73	11	2	27	58.4
4	0	0	56	56	26	48	30.11
5	0	1	31	74	335	87	63.45
6	0	3	41	9	0	135	71.81
Exactitud Global:	82.66						

Tabla 49: Matriz de confusión al utilizar modelo de árbol de decisión para predecir el atributo period (categórico) utilizando los atributos hoard y mint

El mismo árbol podado dio resultados similares (Tabla 50).

	1	2	3	4	5	6	Exactitud por Clase
1	1474	1	2	0	0	3	99.59
2	1	10	0	0	0	2	76.92
3	0	12	74	6	2	31	59.2
4	0	0	57	41	26	62	22.04
5	0	1	32	71	335	89	63.45
6	0	3	44	2	0	139	73.94
Exactitud Global:	82.26						

Tabla 50: Matriz de confusión al utilizar modelo de árbol de decisión podado para predecir el atributo period (categórico) utilizando los atributos hoard y mint

Los resultados del bosque aleatorio fueron un poco peores que los del árbol de decisión (Tabla 51).

	1	2	3	4	5	6	Exactitud por Clase
1	1474	1	1	1	3	0	99.59
2	1	10	0	1	0	1	76.92
3	0	12	72	9	6	26	57.60
4	0	0	47	57	30	52	30.65
5	4	1	33	107	298	85	56.44
6	2	3	33	9	0	141	75.00
Exactitud Global:	81.43						

Tabla 51: Matriz de confusión al utilizar modelo de bosque aleatorio para predecir el atributo period (categórico) utilizando los atributos hoard y mint

Los mejores modelos parecen ser los que interpretan period como una variable categórica y no numérica. Entre estos modelos, el árbol de decisión tuvo los porcentajes de exactitud más altos, aunque es malo prediciendo algunas clases. En total, recibió 65 puntos (Tabla 52). El periodo 1 también parece influir de manera importante la exactitud total, ya que se predice con un casi 100% de exactitud y tiene muchas más observaciones que los demás periodos. Si no se toma en cuenta el periodo 1, la exactitud baja significativamente.

Exactitud total	50

Porcentaje mínimo de aciertos por atributo	0
Complejidad	5
Cantidad de variables independientes	10
Total	**65**

Tabla 52: Puntaje de modelo de árbol de decisión para predecir el atributo period

Capital

Para predecir cuál era la capital del Imperio Romano cuando se fabricó la moneda, se intentaron usar modelos de regresión logística, árboles de decisión, bosques aleatorios y redes neuronales. La mejor combinación de variables independientes en todos los casos fue mint y metal (Tabla 53). Las predicciones de todos los modelos fueron exactamente las mismas.

	Con	Rom	Exactitud por Clase
con	735	292	71.57
rom	115	1378	92.3
Exactitud Global:	83.85		

Tabla 53: Matriz de confusión al utilizar cualquiera de los modelos para predecir el atributo capital utilizando los atributos mint y metal

Ya que todos dan el mismo resultado, sería mejor usar el modelo de regresión logística, que es el más simple de todos y brinda el mayor puntaje (Tabla 54).

Exactitud total	55
Porcentaje mínimo de aciertos por atributo	10
Complejidad	10
Cantidad de variables independientes	10
Total	**85**

Tabla 54: Puntaje de modelo de regresión logística para predecir el atributo capital

Denom

Por el porcentaje de observaciones en el conjunto de datos con denominación desconocida, no se va a utilizar como variable dependiente ni independiente. Podría ser un atributo útil en estudios futuros si se llegara a determinar la denominación de más monedas.

Metal

Para predecir si la moneda es de oro o de bronce, se creó un modelo de regresión logística utilizando los atributos mint, min y max que dio buenos resultados (ver Tabla 55).

	bronze	Gold	Exactitud por Clase
Bronze	2188	253	89.64
Gold	1	77	98.72
Exactitud Global:	89.92		

Tabla 55: Matriz de confusión al utilizar modelo de regresión logística para predecir el atributo metal utilizando los atributos mint, min y max

Un árbol de decisión, utilizando distAlex y min, dio aún mejores resultados (ver Tabla 56).

	bronze	Gold	Exactitud por Clase
Bronze	2260	182	92.55
Gold	0	78	100
Exactitud Global:	92.78		

Tabla 56: Matriz de confusión al utilizar modelo de árbol de decisión para predecir el atributo metal utilizando los atributos distAlex y min

El árbol podado dio los mismos resultados. El bosque aleatorio con los mismos atributos dio resultados levemente mejores (Tabla 57).

	bronze	Gold	Exactitud por Clase
Bronze	2217	169	92.92
Gold	0	78	100
Exactitud Global:	93.14		

Tabla 57: Matriz de confusión al utilizar modelo de bosque aleatorio para predecir el atributo metal utilizando los atributos distAlex y min

La red neuronal con los atributos mint y distAlex con cuatro capas de 40, 20, 10, y 5 nodos obtuvo resultados relativamente buenos, pero peores que los otros modelos utilizados (Tabla 58).

	bronze	Gold	Exactitud por Clase
Bronze	1687	699	70.7
Gold	6	72	92.31
Exactitud Global:	71.39		

Tabla 58: Matriz de confusión al utilizar modelo de red neuronal para predecir el atributo metal utilizando los atributos mint y distAlex

El árbol de decisión parece ser la mejor opción con el mejor puntaje (ver Tabla 59), pero los resultados de todos los modelos son bastante cercanos. Sería útil probar con otro conjunto de prueba para verificar si el árbol es realmente el mejor modelo.

Exactitud total	60
Porcentaje mínimo de aciertos por atributo	20
Complejidad	5
Cantidad de variables independientes	10
Total	95

Tabla 59: Puntaje de modelo de árbol de decisión para predecir el atributo metal

Hoard

La cantidad de monedas por hoard es muy desbalanceada, ya que la mayoría de las monedas proviene de uno de dos grupos (ver Figura 16).

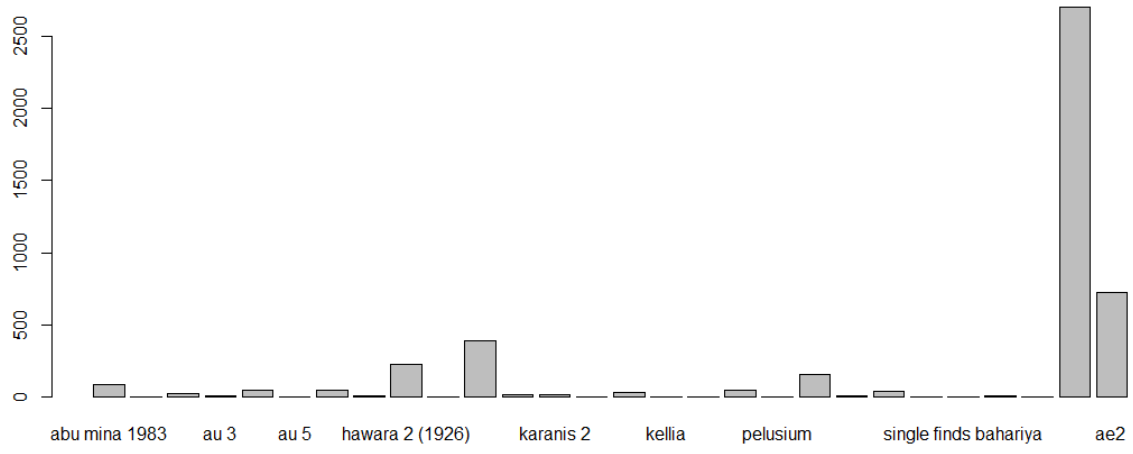


Figura 16: Cantidad de monedas por hoard (elaboración propia)

Si se reduce la cantidad de monedas de los últimos dos grupos, el conjunto de entrenamiento todavía queda muy desbalanceado.

Uno de los problemas es que hay varios hoards en el conjunto de entrenamiento que tienen menos de 10 monedas. Se podrían eliminar estos hoards ya que no parecen tener suficientes observaciones para poder ser predichas, y nivelar la cantidad de observaciones en los otros hoards que tienen más de 10 monedas.

Para predecir el hoard al que pertenece una moneda, se utilizó la regresión logística con los atributos mint, min, max, y metal, árboles de decisión con mint, min, max, distAlex, y metal, y bosques aleatorios con los mismos atributos que los árboles de decisión. No se utilizaron redes neuronales para este atributo, ya que la cantidad de clases por predecir es muy alta y pocos nodos y capas darían malos resultados. Crear un modelo con muchos nodos y capas requeriría de un servidor más poderoso.

Para visualizar los datos más fácilmente, se cambió el nombre de los hoards por su posición en orden alfabética.

En el orden en el que aparecen en los gráficos, las clases en los gráficos con 2 filas corresponden a abu mina 1983, alex chatby, au 2, au 3, au 4, au 5, au 6, fayum (1931), hawara 2 (1926), hawara 5 (1938), hawara 6 (1920), karanis 1974, karanis 2, karanis 2.1, karanis 3, kellia, kom el-ahmar, kom washim-karanis, pelusium, qaw el kebir, single finds abu mina-1906, single finds abu mina 1983, single finds abu mina kaufman, single finds bahariya, single finds clysma, single finds kellia, ae17 y ae2. Estos modelos fueron creados utilizando el conjunto de entrenamiento que tiene un número de observaciones de ae17 y ae2 reducido.

En el orden en el que aparecen en los gráficos, las clases en los gráficos con 15 filas corresponden a abu mina 1983, au 2, au 4, au 6, hawara 2 (1926), hawara 6

(1920), karanis 1974, karanis 2, karanis 3, kom washim-karanis, qaw el kebir, single finds abu mina 1983, single finds clysma, ae17 y ae2. Estos modelos fueron creados utilizando el conjunto de entrenamiento balanceado, del cual también se eliminaron todas las monedas que correspondieran a clases con menos de 10 observaciones.

El modelo de regresión logística con el conjunto de entrenamiento balanceado solamente pudo predecir correctamente un 61.37% de las observaciones del conjunto de prueba, y la exactitud por clase en la mayoría de los casos fue menor que eso. Solo tres de las 15 clases lograron una exactitud de más del 50% (ver Tabla 60).

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	Exactitud por clase
1	16	0	0	0	5	6	0	4	7	5	7	3	1	0	0	29.63
2	0	3	6	3	0	0	5	0	0	0	0	0	0	0	0	17.65
3	0	5	10	4	0	0	2	0	0	0	0	0	0	0	0	47.62
4	0	5	5	8	0	0	6	0	0	0	0	0	0	0	0	33.33
5	17	0	0	0	15	19	0	13	19	10	11	10	0	1	6	12.4
6	21	0	0	0	29	35	0	24	18	30	11	34	0	0	15	16.13
7	0	3	1	2	0	0	3	0	0	0	0	0	0	0	0	33.33
8	2	0	0	0	4	3	0	0	1	1	0	0	0	0	0	0
9	3	0	0	0	2	2	0	2	4	2	1	1	0	0	1	22.22
10	2	0	0	0	1	3	0	2	5	3	3	3	0	0	0	13.64
11	8	0	0	0	7	5	0	1	13	6	25	4	2	0	9	31.25
12	1	0	0	0	2	5	0	1	4	2	3	3	0	0	2	13.04
13	1	0	0	0	0	0	0	0	0	0	0	0	3	0	0	75
14	0	0	0	0	12	2	0	0	0	0	0	0	282	1172	5	79.57
15	21	0	0	0	20	16	0	23	25	16	16	31	0	0	228	57.58

Exactitud	61.37

Tabla 60: Matriz de confusión al utilizar modelo de regresión logística para predecir el atributo hoard utilizando los atributos mint, min, max y metal

El mejor de los árboles de decisión fue en el que se utilizó min, max, mint, distAlex, y metal, con el conjunto de entrenamiento con cantidades reducidas de ae17 y ae2. La exactitud total fue de 87.66%, pero la exactitud por clase aún es muy baja. El

modelo es mejor que el de regresión logística, pero solamente 7 clases lograron más de 50% en exactitud (Tabla 61 y 62).

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
1	17	0	0	0	0	0	0	1	5	0	21	0	0	0	0
2	0	0	0	0	1	0	1	0	0	0	0	0	0	0	0
3	0	0	0	0	9	0	8	0	0	0	0	0	0	0	0
4	0	0	0	0	2	0	2	0	0	0	0	0	0	0	0
5	0	0	1	0	11	0	9	0	0	0	0	0	0	0	0
6	0	0	0	0	0	0	1	0	0	0	0	0	0	0	0
7	0	0	0	0	2	0	22	0	0	0	0	0	0	0	0
8	0	0	0	0	0	0	0	3	0	0	1	0	0	0	0
9	9	0	0	0	0	0	0	0	38	0	58	0	0	0	0
10	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
11	5	0	0	0	0	0	0	0	12	0	190	0	0	0	0
12	0	0	0	0	1	0	8	0	0	0	0	0	0	0	0
13	1	0	0	0	0	0	0	0	4	0	5	0	0	0	0
14	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
16	2	0	0	0	0	0	0	0	2	0	8	0	0	0	1
17	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
18	1	0	0	0	0	0	0	0	3	0	14	0	0	0	0
19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
20	5	0	0	0	0	0	0	0	3	0	13	0	0	0	0
21	1	0	0	0	0	0	0	0	0	0	2	0	0	0	0
22	1	0	0	0	0	0	0	1	3	0	5	0	0	0	0
23	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
24	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
25	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
26	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
27	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
28	1	0	0	0	0	0	0	0	2	0	0	0	0	0	0

Tabla 61: Matriz de confusión al utilizar modelo de árbol de decisión para predecir el atributo hoard utilizando los atributos min, max, mint, distAlex y metal (primera parte)

	16	17	18	19	20	21	22	23	24	25	26	27	28	Exactitud por clase
1	0	0	1	0	8	0	0	0	0	1	0	0	0	31.48
2	0	0	0	0	0	0	0	0	0	0	0	0	0	0
3	0	0	0	0	0	0	0	0	0	0	0	0	0	0
4	0	0	0	0	0	0	0	0	0	0	0	0	0	0
5	0	0	0	0	0	0	0	0	0	0	0	0	0	52.38
6	0	0	0	0	0	0	0	0	0	0	0	0	0	0

7	0	0	0	0	0	0	0	0	0	0	0	0	0	91.67
8	0	0	0	0	0	0	0	0	0	0	0	0	0	75
9	0	0	0	0	14	0	1	0	0	0	0	0	1	31.4
10	0	0	0	0	0	0	0	0	0	0	0	0	0	0
11	0	0	0	0	8	0	0	0	0	0	0	0	2	87.56
12	0	0	0	0	0	0	0	0	0	0	0	0	0	0
13	0	0	1	0	0	0	0	0	0	0	0	0	0	0
14	0	0	0	0	0	0	0	0	0	0	0	0	0	0
15	0	0	0	0	0	0	0	0	0	0	0	0	0	N/A
16	0	0	0	0	5	0	0	0	0	0	0	0	0	0
17	0	0	0	0	2	0	0	0	0	0	0	2	0	0
18	0	0	1	0	3	0	0	0	0	0	0	0	0	4.55
19	0	0	0	0	1	0	1	0	0	0	0	0	0	0
20	0	0	1	0	58	0	0	0	0	0	0	0	0	72.5
21	0	0	0	0	1	0	0	0	0	1	0	0	0	0
22	0	0	1	0	2	0	10	0	0	0	0	0	0	43.48
23	0	0	0	0	1	0	0	0	0	0	0	1	0	0
24	0	0	0	0	1	0	0	0	0	1	0	0	0	0
25	0	0	0	0	3	0	0	0	0	0	0	0	0	0
26	0	0	0	0	0	0	0	0	0	0	0	0	0	0
27	0	0	0	0	5	0	0	0	0	1	0	1467	0	99.59
28	0	0	0	0	1	0	0	0	0	0	0	0	392	98.99

Exactitud	87.66

Tabla 62: Matriz de confusión al utilizar modelo de árbol de decisión para predecir el atributo hoard utilizando los atributos min, max, mint, distAlex y metal (segunda parte)

Por último, se utilizó el conjunto de entrenamiento balanceado y los atributos min, max, mint, distAlex y metal para crear un modelo de bosque aleatorio. Su exactitud es menor que la del árbol de decisión, pero logró predecir 8 clases en total con más de 50% de exactitud, una más que el árbol de decisión (Tabla 63).

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	Exactitud por clase
1	5	0	0	0	2	2	0	10	11	6	1	2	1	0	0	12.5
2	0	3	7	3	0	0	4	0	0	0	0	0	0	0	0	17.65
3	0	2	10	9	0	0	0	0	0	0	0	0	0	0	0	47.62
4	0	0	2	15	0	0	7	0	0	0	0	0	0	0	0	62.5
5	11	0	0	0	7	29	0	21	18	17	2	0	1	0	1	6.54

6	3	0	0	0	9	114	0	44	16	19	6	0	0	0	1	53.77
7	0	0	1	4	0	0	4	0	0	0	0	0	0	0	0	44.44
8	0	0	0	0	0	2	0	7	0	1	0	0	0	0	0	70
9	1	0	0	0	0	2	0	4	6	1	0	0	0	0	0	42.86
10	1	0	0	0	0	5	0	6	2	5	0	0	0	0	0	26.32
11	1	0	0	0	2	13	0	2	6	4	38	2	1	0	0	55.07
12	1	0	0	0	0	2	0	4	0	1	1	13	0	0	0	59.09
13	0	0	0	0	0	0	0	0	1	0	0	0	3	0	0	75
14	0	0	0	0	0	0	0	0	0	0	0	0	96	1374	0	93.47
15	3	0	0	0	5	4	0	23	3	3	1	10	0	0	344	86.87

Exactitud	**80.03**

Tabla 63: Matriz de confusión al utilizar modelo de bosque aleatorio para predecir el atributo hoard utilizando los atributos min, max, mint, distAlex y metal

Ningún modelo parece ser bueno prediciendo el atributo hoard, ya que la exactitud por clase en todos los modelos es muy baja. El modelo con el mayor puntaje según los criterios de evaluación fue el árbol de decisión, con 65 puntos (Tabla 64).

Exactitud total	55
Porcentaje mínimo de aciertos por atributo	0
Complejidad	5
Cantidad de variables independientes	5
Total	**65**

Tabla 64: Puntaje de modelo de árbol de decisión para predecir el atributo hoard

Probando los modelos con otro conjunto de prueba

Las monedas utilizadas en el conjunto de prueba son un 30% aleatorio de las 7000 monedas estudiadas. Quedan aproximadamente 23,000 monedas que se pueden utilizar para probar los modelos y ver si son aplicables a cualquier conjunto de monedas romanas de esta época encontradas en Egipto, o si solo funcionan con las monedas que cumplen las condiciones mencionadas por la doctoranda: tienen RIC,

tienen ceca, y los años min y max no se extienden por más de un periodo (solo pertenecen a un periodo Reece).

El segundo conjunto de prueba que se está utilizando se compone de todas las monedas que cumplen con las mismas condiciones, con la excepción de RIC. Ninguna de las monedas en este conjunto de pruebas tiene RIC.

DistCategory

Al pasar el nuevo conjunto de prueba por el árbol de decisión creado con un conjunto de datos balanceado, la exactitud total pasó de 77% a 46% (Tabla 65). La exactitud por clase también se volvió aún más desbalanceada.

	Far	mid-range	Near	Exactitud por clase
Far	6	160	7	3.47
Midrange	9	2781	130	95.24
Near	19	3547	543	13.21

Exactitud	46.24

Tabla 65: Matriz de confusión al utilizar modelo de árbol de decisión para predecir el atributo distCategory con el nuevo conjunto de prueba

El modelo balanceado tampoco dio buenos resultados (Tabla 66). Bajó de 68% a 44%, y la exactitud por clase quedó desbalanceada de una manera similar al modelo anterior.

	Far	mid-range	Near	Exactitud por clase
Far	13	157	3	7.51
Midrange	13	2850	57	97.60
Near	28	3766	315	7.67

Exactitud	68.91

Tabla 66: Matriz de confusión al utilizar modelo de árbol de decisión, creado con un conjunto de entrenamiento balanceado, para predecir el atributo distCategory con el nuevo conjunto de prueba

El modelo con CP más bajo, que pudo corregir un poco el sobreajuste, tuvo el mismo problema que los anteriores (Tabla 67).

	far	mid-range	Near	Exactitud por clase
far	6	162	5	3.47
midrange	9	2794	117	95.68
Near	37	3634	438	10.66

Exactitud	44.96

Tabla 67: Matriz de confusión al utilizar modelo de árbol de decisión con un CP de 0.01, creado con un conjunto de entrenamiento balanceado, para predecir el atributo distCategory con el nuevo conjunto de prueba

El bosque aleatorio fue el que dio el mejor resultado y el más balanceado, pero, no se puede considerar un buen resultado (Tabla 68).

	far	mid-range	Near	Exactitud por clase
Far	35	132	6	20.23
midrange	150	2640	130	90.41
near	186	3367	556	13.53

Exactitud	44.86

Tabla 68: Matriz de confusión al utilizar modelo de bosque aleatorio para predecir el atributo distCategory con el nuevo conjunto de prueba

La red neuronal nuevamente solo predijo una clase (Tabla 69).

	Far	mid-range	Near	Exactitud por clase
far	0	0	173	0
midrange	0	0	2920	0
near	0	0	4109	100

Exactitud	57.05

Tabla 69: Matriz de confusión al utilizar modelo de red neuronal para predecir el atributo distCategory con el nuevo conjunto de prueba

Si volvemos a usar los criterios de evaluación, se puede ver que ningún modelo llegó a una exactitud total de más de 75% ni una exactitud por clase mínima de 60%, y

se pierden 50% de los puntos (Tabla 70). Esto muestra que el modelo no es útil para predecir las monedas que no tienen RIC, y probablemente está sobreajustado.

Exactitud total	0
Porcentaje mínimo de aciertos por atributo	0
Complejidad	5
Cantidad de variables independientes	7
Total	**12**

Tabla 70: Puntaje de modelo de árbol de decisión para predecir el atributo distCategory, evaluándolo con el nuevo conjunto de datos de prueba

Port

Utilizando el nuevo conjunto de datos, el modelo de regresión logística tuvo una exactitud total de 66.4%, pero una exactitud por clase desbalanceada (Tabla 71).

	0	1	Exactitud por clase
0	314	2083	13.10
1	344	4461	92.84

Exactitud	**66.3**

Tabla 71: Matriz de confusión al utilizar modelo de regresión logística para predecir el atributo port con el nuevo conjunto de prueba

Un árbol de decisión que utilice solo min y max y el conjunto de datos de entrenamiento balanceado pudo predecir la mayoría de las observaciones fabricadas en un puerto, pero solo 0.46 de las observaciones fabricadas en cecas no costales (Tabla 72).

	0	1	Exactitud por clase
0	11	2386	0.46
1	10	4795	99.79

Exactitud	**66.73**

Tabla 72: Matriz de confusión al utilizar modelo de árbol de decisión para predecir el atributo port con el nuevo conjunto de prueba, usando los atributos min y max

Utilizar el atributo hoard como variable predictiva ayudó a mejorar la exactitud cuándo se utilizó el primer conjunto de prueba. Al utilizar el nuevo conjunto de prueba, nos encontramos que el modelo creado con el conjunto de entrenamiento no encuentra algunas clases de hoard en ambos conjuntos. Al dejar solamente las clases que están en ambos conjuntos, solo quedan dos hoards: ae17 y karanis 2.1, cuando el conjunto de prueba tenía 13 clases y el conjunto de entrenamiento tenía 28 clases. Las proporciones también son completamente diferentes. El conjunto de entrenamiento tiene 1 moneda karanis 2.1 y 2183 monedas ae17. El conjunto de prueba tiene 65 monedas karanis 2.1 y 14 monedas ae17. Esto explica porque los conjuntos son tan diferentes. Parece que las monedas del nuevo conjunto de prueba, las monedas sin RIC, son de hoards distintos. Esto también indica la gran diferencia que hay entre todos los hoards, y como cada hoard no es realmente representativo de toda la población.

Si se eliminan las observaciones de hoards que el modelo no conoce, el conjunto de prueba solamente tiene 79 monedas que se pueden usar para modelos que utilicen hoard como variable predictiva. Cualquier árbol de decisión o bosque aleatorio que contenga el atributo hoard, solamente podría intentar predecir el atributo port para estas 79 monedas (ver Tabla 73).

	0	1	Exactitud por clase
0	9	29	23.68
1	0	41	100

Exactitud	63.29

Tabla 73: Matriz de confusión al utilizar modelo de árbol de decisión para predecir el atributo port con el nuevo conjunto de prueba (solamente las observaciones de hoards conocidos por el modelo), usando los atributos min, max, y hoard

La red neuronal predijo que todas las monedas provienen de ciudades costales (ver Tabla 74).

	0	1	Exactitud por clase
0	0	38	0
1	0	41	100

Exactitud	51.9

Tabla 74: Matriz de confusión al utilizar modelo de red neuronal para predecir el atributo port con el nuevo conjunto de prueba

Al igual que los modelos para la variable distCategory, ninguno de los modelos creados para predecir port funcionaron con el nuevo conjunto de prueba, ninguno logró una exactitud de 75%, ni una exactitud mínima por clase de 60%. Esto baja el puntaje de los modelos de la variable port a 17 de 100 (Tabla 75).

Exactitud total	0
Porcentaje mínimo de aciertos por atributo	0
Complejidad	10
Cantidad de variables independientes	7
Total	**17**

Tabla 75: Puntaje de modelo de árbol de decisión para predecir el atributo port, evaluándolo con el nuevo conjunto de datos de prueba

DistAlex

El RMSE y MAE de los modelos utilizados para predecir distAlex aumentó de manera significativa (Tabla 76). Los modelos originales no eran buenos, pero sí se puede notar una diferencia grande, especialmente en el RMSE. Esto quiere decir que muchas predicciones estuvieron bastante alejadas del valor correcto.

Modelo	RMSE	MAE
Regresión Lineal	1017.42	803.73
Árbol	1304.87	1045.98
Árbol Podado	1306.51	1047.78
Bosque Aleatorio	1277.03	1080.19

Tabla 76: Comparación de RMSE y MAE de modelos creados para predecir el atributo distAlex usando el nuevo conjunto de datos de prueba

TimeAlex

Sucede algo similar con las predicciones del tiempo de viaje de la ceca hasta Alejandría (Tabla 77). En este caso, el modelo de regresión lineal parece ser el que se vio menos afectado, ya que tuvo resultados similares a los que se obtuvieron utilizando el conjunto de prueba original. Aun así, nunca fue considerado un buen modelo y tampoco se podría considerar un buen modelo con los nuevos resultados.

Modelo	RMSE	MAE
Regresión Lineal	9.57	7.68
Árbol	10.92	9.20
Árbol Podado	10.92	9.21
Bosque Aleatorio	10.46	8.64

Tabla 77: Comparación de RMSE y MAE de modelos creados para predecir el atributo timeAlex usando el nuevo conjunto de datos de prueba

Min

El modelo de regresión lineal para predecir el año mínimo de fabricación dio mejores resultados con el nuevo conjunto de prueba, pero todos los demás modelos tuvieron malos resultados (Tabla 78). Si se compara el RMSE y el MAE de cada modelo con los que obtuvo al utilizar el conjunto de pruebas original, el error calculado para los árboles es aproximadamente 50% más del original. Aparte de esto, al ver la distribución de las diferencias entre el valor real y el valor predicho, se puede apreciar que el modelo se equivocó por alrededor de 20 años en la mayoría de los casos. Por lo tanto, ninguno de los modelos parece ser bueno.

Modelo	RMSE	MAE
Regresión Lineal	48.65	42.39
Árbol	11.05	7.39
Árbol Podado	11.05	7.45

Bosque Aleatorio	24.23	17.33

Tabla 78: Comparación de RMSE y MAE de modelos creados para predecir el atributo min usando el nuevo conjunto de datos de prueba

Max

En los modelos creados para predecir max, el error incrementó de manera similar a min cuando se utilizó el segundo conjunto de prueba (Tabla 79). Parece que ningún modelo para predecir el atributo max fue exitoso.

Modelo	RMSE	MAE
Regresión Lineal	22.27	19.95
Árbol	12.44	8.55
Árbol Podado	12.44	8.55
Bosque Aleatorio	39.47	34.57

Tabla 79: Comparación de RMSE y MAE de modelos creados para predecir el atributo max usando el nuevo conjunto de datos de prueba

Period

Anteriormente, se crearon diferentes tipos de modelos para el atributo periodo, algunas veces tratándolo como atributo categórico y otras como atributo numérico. Los resultados de los modelos numéricos no fueron muy buenos y al utilizar el conjunto de prueba nuevo, los resultados fueron peores. Luego de probar el mejor modelo numérico (el árbol de decisión) con el conjunto de prueba nuevo, dio una exactitud de 46.84% y una exactitud por clase promedio de 42.08% (Tabla 80).

	1	2	3	4	5	6	Exactitud por Clase
1	14	0	0	0	0	0	100
2	0	0	0	0	0	0	N/A
3	0	0	1	7	1	0	11.11
4	2	0	0	7	4	0	53.85
5	4	0	1	13	15	0	45.45
6	1	0	0	7	2	0	0
Exactitud Global:	**46.84**						

Tabla 80: Matriz de confusión al utilizar modelo de árbol de decisión para predecir el atributo period (numérico) utilizando el nuevo conjunto de prueba

El mejor modelo categórico con el conjunto de prueba original también fue un árbol de decisión, pero al pasar el nuevo conjunto de prueba por los modelos, el que dio los mejores resultados fue el de regresión logística, con una exactitud total de 39.24% y una exactitud por clase promedio de 47.93% (Tabla 81).

	1	2	3	4	5	6	Exactitud por Clase
1	14	0	0	0	0	0	100
2	0	0	0	0	0	0	N/A
3	0	0	4	1	0	4	44.44
4	0	0	4	3	1	5	23.08
5	0	0	10	6	4	13	12.12
6	0	0	2	1	1	6	60.00
Exactitud Global:	39.24						

Tabla 81: Matriz de confusión al utilizar modelo de regresión logística para predecir el atributo period (categórico) utilizando el nuevo conjunto de prueba

Ambos modelos tienen exactitudes muy bajas, de mucho menos de 75%. Por lo tanto, según los criterios de evaluación, el modelo pierde 50 puntos y queda con solamente 15 (Tabla 82).

Exactitud total	0
Porcentaje mínimo de aciertos por atributo	0
Complejidad	5
Cantidad de variables independientes	10
Total	15

Tabla 82: Puntaje de modelo de árbol de decisión para predecir el atributo period, evaluándolo con el nuevo conjunto de datos de prueba

Capital

El modelo creado para predecir el atributo capital dio muy buenos resultados con el conjunto de prueba original, pero la exactitud bajó a 41.81% con el nuevo conjunto de prueba (Tabla 83).

	Con	Rom	Exactitud por Clase

con	2746	3904	41.29
rom	286	265	48.09
Exactitud Global:	41.81		

Tabla 83: Matriz de confusión al utilizar modelo de regresión logística para predecir el atributo capital utilizando el nuevo conjunto de prueba

Esto baja el puntaje total a 20 puntos de 100 (Tabla 84).

Exactitud total	0
Porcentaje mínimo de aciertos por atributo	0
Complejidad	10
Cantidad de variables independientes	10
Total	20

Tabla 84: Puntaje de modelo de regresión logística para predecir el atributo capital, evaluándolo con el nuevo conjunto de datos de prueba

Metal

Los modelos de árbol de decisión y bosque aleatorio creados para predecir el atributo metal sí dieron buenos resultados con el nuevo conjunto de prueba. El bosque aleatorio tuvo resultados levemente mejores (logró identificar con .02% más exactitud las monedas de bronce), pero cualquiera de los dos algoritmos se podría usar y darían resultados similares. La exactitud total del modelo de bosque aleatorio fue de 97.26%, y pudo identificar todas las monedas de oro con una exactitud del 100% y las de bronce con una exactitud de 97.22% (Tabla 85).

	Bronze	Gold	Exactitud por Clase
Bronze	6899	197	97.22
Gold	0	106	100
Exactitud Global:	97.26		

Tabla 85: Matriz de confusión al utilizar modelo de bosque aleatorio para predecir el atributo metal utilizando el nuevo conjunto de prueba

El modelo del atributo metal es el primer modelo que ha tenido buenos resultados con ambos conjuntos de prueba. Los resultados fueron aún mejores al usar el nuevo conjunto de prueba y su puntaje se mantiene en 95 de 100 (Tabla 86).

Exactitud total	60
Porcentaje mínimo de aciertos por atributo	20
Complejidad	5
Cantidad de variables independientes	10
Total	**95**

Tabla 86: Puntaje de modelo de árbol de decisión para predecir el atributo metal, evaluándolo con el nuevo conjunto de datos de prueba

Al visualizar el modelo, se puede notar como es que está clasificando las monedas (ver Figura 17). El modelo clasifica como monedas de oro a todas las que hayan sido fabricadas a más de 484 kilómetros de Alejandría entre el año 346 y el año 372. Estas condiciones parecen cubrir la totalidad de las monedas de oro, porque el modelo ha logrado predecir, con ambos conjuntos de datos de prueba, cuáles monedas son de oro, con una exactitud del 100%. Lo que también se puede notar es que algunas monedas de bronce cumplen estas condiciones y el modelo también predice que son de oro. Proporcionalmente, no es una cantidad de monedas muy grande, especialmente cuando se compara con todas las demás monedas que sí predice que son de oro correctamente.

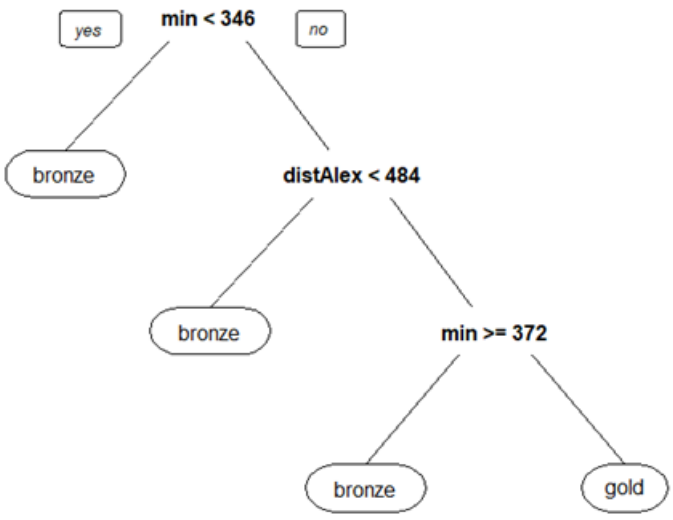


Figura 17: Árbol de decisión para predecir el atributo metal utilizando min y distAlex (elaboración propia)

Hoard

Como el conjunto de monedas con RIC y el conjunto de monedas sin RIC solo comparten dos hoards en común, los modelos para predecir el atributo hoard no son muy útiles. Primero, porque el modelo nunca va a seleccionar una clase que no conozca (es decir, un hoard que exista en el conjunto de datos de prueba y que no exista en el de entrenamiento). Esto significa que la exactitud para casi todas las clases es 0%. Las únicas dos clases que podían ser predichas correctamente por el modelo son "ae17" y "karanis 2.1". El modelo no logró identificar ninguna moneda que perteneciera a karanis 2.1, pero al menos logró identificar 85.71% de las monedas ae17 en el nuevo conjunto de prueba (Tabla 87 y Tabla 88).

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
14	1	0	0	0	0	0	0	0	14	0	41	0	0	0	0
27	0	0	0	0	0	0	0	0	2	0	0	0	0	0	0

Tabla 87: Matriz de confusión al utilizar modelo de árbol de decisión para predecir el atributo hoard utilizando el nuevo conjunto de prueba (primera parte)

	16	17	18	19	20	21	22	23	24	25	26	27	28	Exactitud por clase
14	0	0	1	0	8	0	0	0	0	0	0	0	0	0
27	0	0	0	0	0	0	0	0	0	0	0	12	0	85.71

Tabla 88: Matriz de confusión al utilizar modelo de árbol de decisión para predecir el atributo hoard utilizando el nuevo conjunto de prueba (segunda parte)

De las diez clases de hoards en el conjunto de prueba, nueve no se lograron identificar del todo. El porcentaje de exactitud para la clase ae17 es relativamente bueno, pero el modelo no sería útil si solamente funciona con una clase. Si se reajusta el puntaje del modelo según su rendimiento con el segundo conjunto de datos de prueba, quedaría en 10 puntos (Tabla 89).

Exactitud total	0
Porcentaje mínimo de aciertos por atributo	0
Complejidad	5
Cantidad de variables independientes	5
Total	**10**

Tabla 89: Puntaje de modelo de árbol de decisión para predecir el atributo hoard, evaluándolo con el nuevo conjunto de datos de prueba

Hallazgos

El único modelo que dio buenos resultados de manera consistente fue el modelo creado para el atributo metal que indica si una moneda es de oro o de bronce. Aunque es un modelo efectivo, no es necesario, ya que basta con solo ver una moneda para identificar si es de oro o de bronce. Lo que nos dice el modelo y lo que puede ser útil para fines arqueológicos es que las monedas de oro encontradas fueron fabricadas en cecas fuera de Egipto, en la segunda mitad del Siglo IV.

El problema con los modelos parece ser que las diferencias entre un grupo de monedas y otro son muy grandes y unos cuantos grupos no parecen representar suficientemente bien toda la población. Aunque se crearon modelos que parecían ser buenos con el primer conjunto de datos de prueba, fueron entrenados con monedas de ciertos hoards específicos que también existían en el conjunto de prueba. Al tomar un conjunto de prueba con las mismas características, pero sin RIC, solo el modelo del atributo metal pudo clasificar a las monedas correctamente. Todos los demás fallaron, porque los patrones y relaciones encontradas con el conjunto de entrenamiento no existían en el conjunto de prueba.

Según lo visto hasta el momento, posiblemente es mejor estudiar las monedas por grupo o por periodo e intentar crear modelos específicos para cada uno y no hacerlo por siglo. Lamentablemente, no parecen existir suficientes datos para poder estudiar grupos más pequeños de monedas. Si se consiguieran muchas más monedas, se podría reevaluar la posibilidad de crear modelos predictivos, pero con los datos disponibles actualmente no parece ser posible.

Capítulo 5. Propuesta de Solución

5.1 Creación de aplicación

Como no se lograron crear buenos modelos utilizando los datos originales, se decidió crear una aplicación donde el usuario final pueda crear modelos por medio de una interfaz gráfica fácil de usar (ver Apéndice 4). Esta aplicación le debe decir al usuario si el modelo generado es un buen modelo o no y darle la opción de guardar el modelo, para luego aplicarlo a datos nuevos. De esta forma, el usuario no tiene que depender de modelos estáticos creados con conjuntos de datos de entrenamiento limitados, sino que tienen la oportunidad de crear, probar y utilizar modelos ellos mismos, con datos que reciban en el futuro.

Para hacer la aplicación lo más simple posible para el usuario, se seleccionó el algoritmo con mejores resultados en las pruebas anteriores para problemas de clasificación: el árbol de decisión. El árbol de decisión también tiene la ventaja de ser fácil de visualizar, lo cual le permite al usuario entender cuáles factores está tomando en cuenta el modelo para la clasificación.

La aplicación utiliza Java y R (Figura 18), los cuales se comunican por medio de rJava. La interfaz gráfica fue diseñada en Java y la manipulación de los datos se

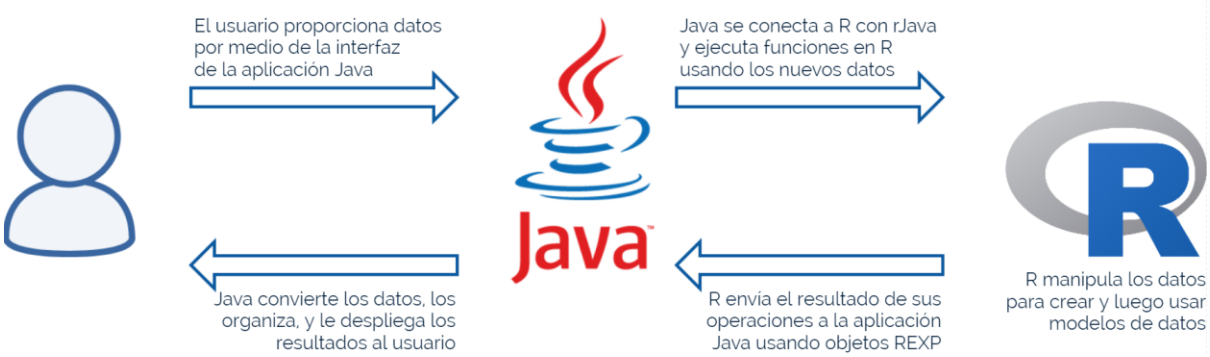


Figura 18: Interacción entre el usuario, Java, y R (elaboración propia)

realiza en R. Se utilizó la librería “rpart” de R para la creación de los árboles de decisión.

Al comenzar la aplicación, se crea una nueva instancia de R y se carga createModelscript.R, el cual contiene todas las funciones de R que podrían necesitarse para la creación y utilización de modelos.

Módulo de creación de modelos

El módulo de creación de modelos requiere que se le proporcione un archivo de datos CSV como conjunto de datos de entrenamiento (Figura 19) y un segundo archivo opcional como conjunto de datos de prueba. Si no se selecciona un segundo archivo, la aplicación divide los datos del primer archivo en dos partes, 70% para el conjunto de entrenamiento y 30% para el conjunto de prueba. La aplicación asume que cada columna corresponde a un atributo, cada fila a una observación y que cualquier celda con el valor ‘?’ corresponde a un valor nulo (NA). La primera fila debe contener el nombre de cada atributo correspondiente.

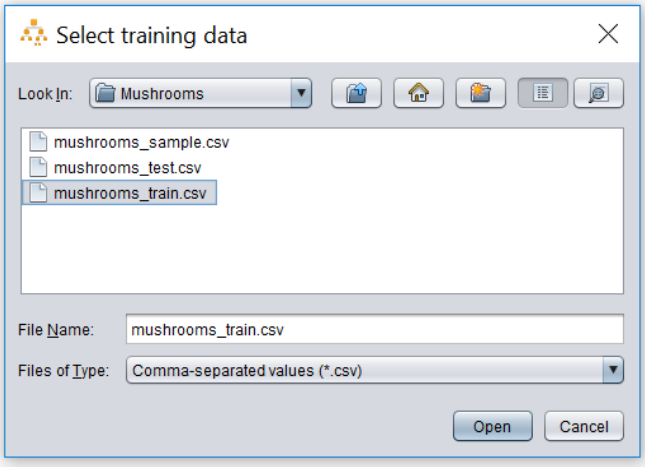


Figura 19: Selección de conjunto de datos de entrenamiento (elaboración propia)

Después de seleccionar los conjuntos de datos, el usuario debe seleccionar la variable dependiente, es decir, el atributo que el modelo intentará predecir. Como la aplicación crea exclusivamente árboles de decisión de clasificación, solo se muestran los atributos que han sido identificados por R como clases (Figura 20).

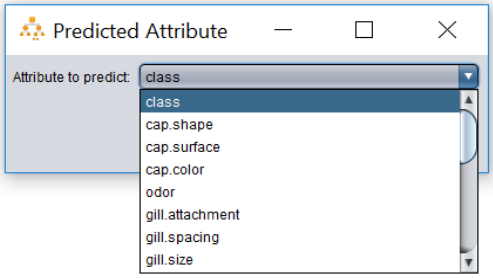


Figura 20: Selección de variable dependiente (elaboración propia)

Luego de seleccionar la variable dependiente, se muestran todos los posibles valores (factores) que puede tener y cuantas ocurrencias hay de cada uno. El usuario puede, de esta manera, entender si hay algún desbalance entre las clases. Si existe un desbalance, el usuario puede continuar con el proceso de creación del modelo o puede utilizar la función “Balance Data Sets” para balancear el conjunto de datos de entrenamiento. Esta función recibe como parámetro la cantidad de observaciones por clase. Si una clase tiene más observaciones que la cantidad especificada, se eliminan observaciones. Si una clase tiene menos observaciones que la cantidad especificada, se duplican observaciones. Las clases no quedan con exactamente la misma cantidad de observaciones, pero quedan con cantidades suficientemente cercanas para considerarse más balanceadas. El usuario puede seleccionar balancear el conjunto de datos de entrenamiento por media, moda, mediana o por un número específico de su elección (Figura 21).

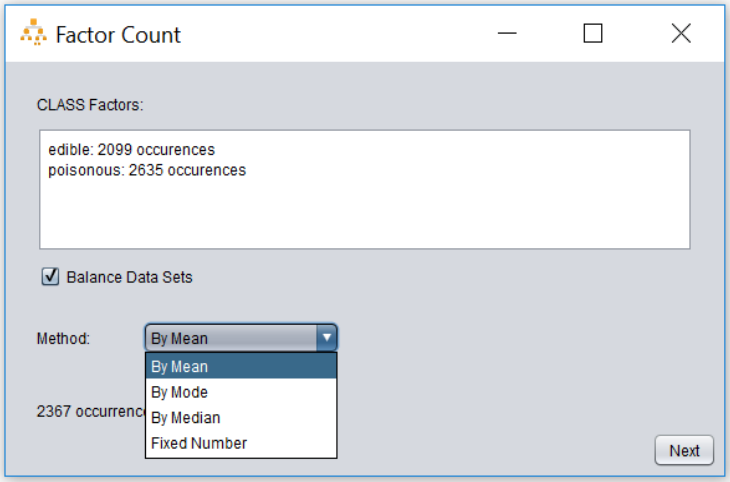


Figura 21: Pantalla para balancear el conjunto de datos de entrenamiento (elaboración propia). En este caso, el conjunto ya está relativamente balanceado.

La siguiente pantalla permite seleccionar las variables independientes que serán utilizadas para predecir la variable dependiente (Figura 22). Junto a cada una de las variables categóricas, se muestra la cantidad de factores que existen. El usuario puede seleccionar desde una hasta todas las variables independientes. Si el usuario selecciona alguna variable con más de 25 factores, se le muestra una advertencia, ya que utilizar tantos factores requiere más tiempo y poder de procesamiento y generalmente da malos resultados.

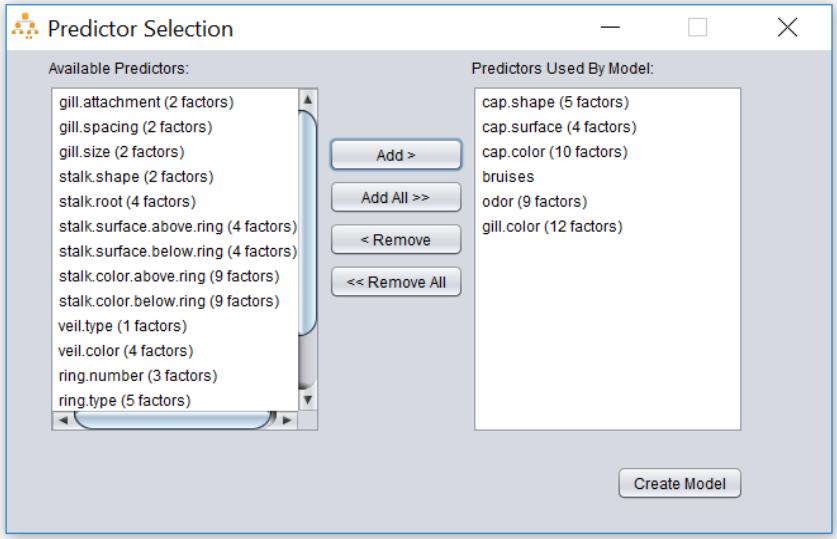


Figura 22: Selección de variables independientes (elaboración propia)

Luego de que el usuario selecciona los predictores, la aplicación busca las mejores variables entre las seleccionadas y crea el modelo. Después, utiliza el conjunto de pruebas para probarlo y muestra los resultados con una matriz de confusión (Figura 23). La interfaz también muestra la exactitud total del modelo, la exactitud por cada clase y la exactitud promedio por clase. La exactitud promedio por clase permite evaluar el modelo dándole el mismo peso a todas las clases, sin tomar en cuenta cuántas observaciones existen por cada clase en el conjunto de prueba.

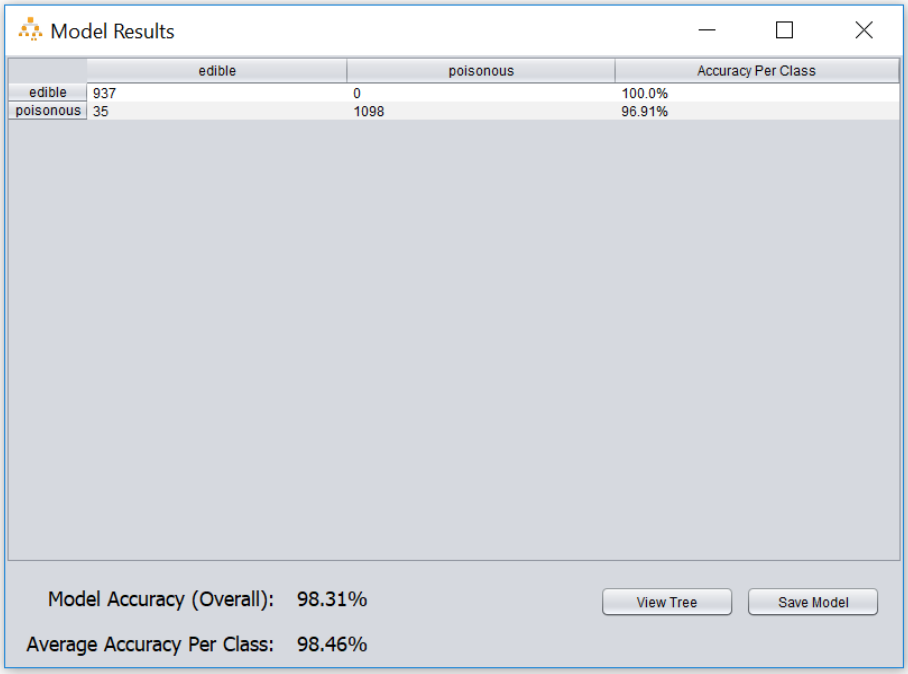


Figura 23: Resultados del modelo generado (elaboración propia)

El botón “View Tree” permite ver, de manera gráfica, el árbol de decisión utilizado por el modelo, y le da la opción al usuario de guardarlo como archivo PNG (Figura 24).

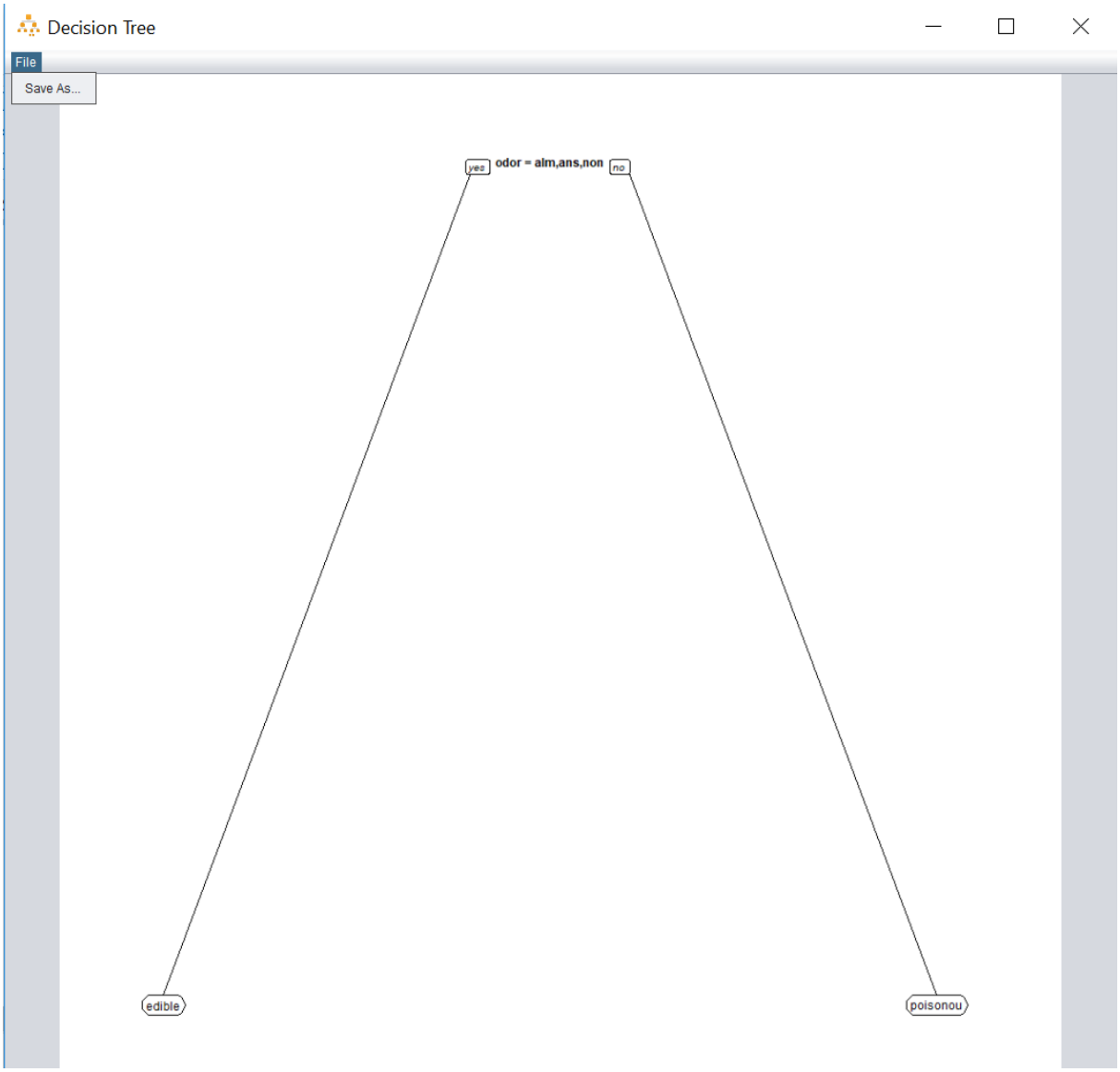


Figura 24: Visualización de árbol de decisión utilizado por el modelo (elaboración propia)

Por último, el usuario puede guardar el modelo para usarlo con otros conjuntos de datos en el futuro o puede cerrar la ventana de resultados para volver a la pantalla principal.

Módulo de utilización de modelos

Cuando ya se haya creado y guardado algún modelo, se puede hacer uso del módulo de utilización de modelos existentes (Figura 25). Desde este módulo, se puede

seleccionar uno de los modelos guardados y realizar varias operaciones. Desde este módulo también se puede visualizar el árbol de decisión y guardar la imagen.

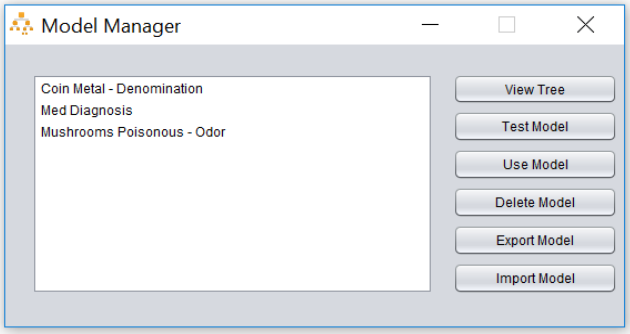
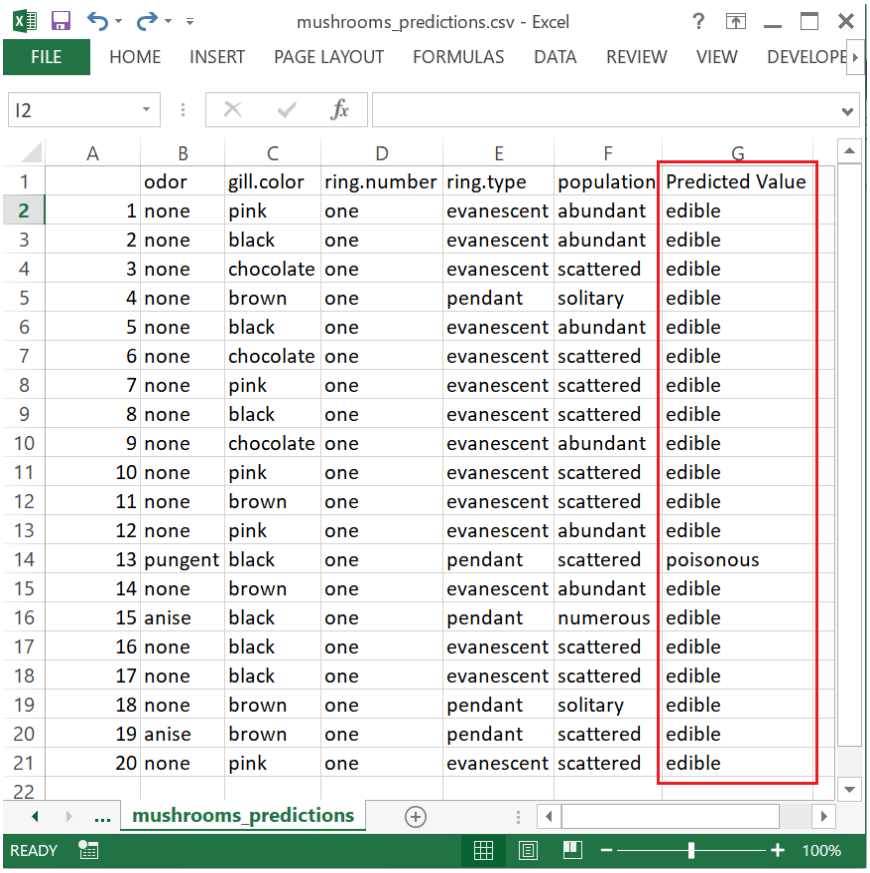


Figura 25: Módulo de administración de modelos (elaboración propia)

Utilizando la opción “Test Model”, se selecciona el modelo y un conjunto de datos de prueba que contenga todas las variables independientes y la variable dependiente. Esto permite que la exactitud del modelo se pueda probar con diferentes conjuntos de prueba. Por ejemplo, si se crea un modelo que predice con una alta exactitud la región donde se fabricó una moneda con el uso de un conjunto de datos exclusivamente de la primera mitad del Siglo IV, se puede probar el modelo con un segundo conjunto de datos, de la segunda mitad del Siglo IV, para verificar si el modelo aun predice correctamente la ceca de las monedas de finales del Siglo IV.

La opción “Use Model” toma el modelo seleccionado y un conjunto de datos que contenga todas las variables independientes, usa el modelo, y agrega una columna con la predicción de la variable dependiente al conjunto de datos. Los datos originales, junto con la nueva columna con predicciones, se guarda en la ubicación especificada por el usuario (Figura 26).



	A	odor	gill.color	ring.number	ring.type	population	Predicted Value
1		odor	gill.color	ring.number	ring.type	population	Predicted Value
2	1	none	pink	one	evanescent	abundant	edible
3	2	none	black	one	evanescent	abundant	edible
4	3	none	chocolate	one	evanescent	scattered	edible
5	4	none	brown	one	pendant	solitary	edible
6	5	none	black	one	evanescent	abundant	edible
7	6	none	chocolate	one	evanescent	scattered	edible
8	7	none	pink	one	evanescent	scattered	edible
9	8	none	black	one	evanescent	scattered	edible
10	9	none	chocolate	one	evanescent	abundant	edible
11	10	none	pink	one	evanescent	scattered	edible
12	11	none	brown	one	evanescent	scattered	edible
13	12	none	pink	one	evanescent	abundant	edible
14	13	pungent	black	one	pendant	scattered	poisonous
15	14	none	brown	one	evanescent	abundant	edible
16	15	anise	black	one	pendant	numerous	edible
17	16	none	black	one	evanescent	scattered	edible
18	17	none	black	one	evanescent	scattered	edible
19	18	none	brown	one	pendant	solitary	edible
20	19	anise	brown	one	pendant	scattered	edible
21	20	none	pink	one	evanescent	scattered	edible
22							

Figura 26: Archivo CSV generado con datos originales y con una columna de predicciones (elaboración propia)

Este módulo también permite administrar los modelos, brindándoles a los usuarios la posibilidad de eliminar modelos, exportar modelos e importar modelos que han sido exportados con el uso de la misma aplicación en otra máquina.

Uso de la aplicación

Aunque ya se realizó la disertación doctoral de la doctoranda utilizando el análisis de datos provisto por la investigadora, Soto seguirá su investigación y recopilación de datos de monedas para la elaboración y eventual publicación de un libro. Se espera que la doctoranda haga uso de esta aplicación para crear sus propios modelos predictivos cuando se logren recopilar más datos de monedas antiguas. También se espera que otros miembros de ISAW y cualquier otra organización

interesada la usen en sus propias investigaciones, ya que la aplicación también puede ser utilizada con otros tipos de conjuntos de datos. Por ejemplo, el instituto podría utilizar la aplicación para crear modelos predictivos para identificar el origen de piezas de cerámica encontradas en excavaciones o para identificar arte de diferentes civilizaciones. Realmente, la aplicación se podría usar para crear árboles de decisión de clasificación con cualquier conjunto de datos que contenga una variable dependiente con dos o más clases y una o más variables independientes.

Capítulo 6. Conclusiones y Recomendaciones

6.1 Conclusiones

Durante la investigación, se obtuvieron datos de treinta y tres mil monedas romanas producidas en el Siglo IV y encontradas en Egipto. De estas monedas, aproximadamente siete mil fueron identificadas como monedas que reunían todas las características necesarias para el análisis, según el criterio de la doctoranda.

Se realizó un análisis exhaustivo de los atributos de las monedas, comparando cada atributo disponible a los demás para encontrar relaciones existentes entre ellos. Al finalizar esta etapa, se descubrieron cuáles eran los datos que podrían ser útiles en la búsqueda de patrones. Si un atributo no parecía tener relación con otro o mostraba una relación que no era útil (por ejemplo, cuando un atributo era derivado de otro), la relación entre ambos atributos no era considerada para crear modelos de datos.

Para cada relación que podría ser considerada útil, se elaboraron varios modelos de minería de datos para la clasificación de las monedas y se utilizaron

diferentes algoritmos según el tipo de dato del atributo que se estaba intentando clasificar. En total, se crearon más de 100 modelos predictivos.

Una vez creados los modelos de minería de datos, se usaron diversos métodos para probar su exactitud, según los tipos de datos de los atributos y el modelo siendo probado, y se analizaron los resultados. De estos resultados, se esperaba seleccionar los mejores modelos, pero ninguno dio buenos resultados según los criterios de evaluación definidos al comienzo de la investigación. Aunque los resultados no fueron buenos, el método que consistentemente daba los puntajes más altos era el árbol de clasificación. Al ser el modelo más fácil de visualizar y de entender, se seleccionó como el mejor método y el que sería utilizado en la aplicación.

Tomando en cuenta que no se encontraron modelos de datos buenos que se pudieran incluir en la aplicación, se tuvo que crear una interfaz que le permitiera a un usuario, posiblemente sin conocimiento de minería de datos, crear sus propios modelos de minería de datos con el uso de conjuntos de datos nuevos.

El resultado fue una aplicación de escritorio que permite crear, evaluar y usar modelos de minería de datos, no solo para la clasificación e identificación de monedas romanas en Egipto producidas en el Siglo IV, sino que también para la clasificación de otros objetos arqueológicos e históricos que se quieran estudiar. Aunque la aplicación planteada originalmente se elaboró principalmente para la investigación de la doctoranda Soto, el producto final podrá ser usado en todo tipo de investigaciones, no solamente en ISAW, pero también en otras instituciones donde Soto considere apropiado compartirlo.

En síntesis, se logró ir más allá del objetivo de la investigación, proporcionando una solución más robusta y versátil que la propuesta original.

6.2 Recomendaciones

Para obtener mejores resultados al usar la aplicación, se presentan las siguientes recomendaciones:

- Aunque un modelo creado por la aplicación parezca ser bueno, se recomienda que el modelo y los resultados sean revisados por un analista de datos antes de utilizarlos en una investigación formal. Hay factores, como el sobreajuste, que pueden hacer que un modelo parezca ser bueno cuando no lo es
- Se recomienda revisar y limpiar los datos antes de utilizarlos con la aplicación. Es importante que cualquier valor desconocido se cambie a “?”, que ningún campo quede en blanco y que las clases de cada atributo se estandaricen para que cada clase sea representada por un solo valor (es decir, que no existan representaciones diferentes de una misma clase como “bronce” y “bronze”)
- Se recomienda estudiar la representación gráfica del árbol de clasificación cada vez que se crea un modelo nuevo. Es una herramienta importante para determinar si el sistema de clasificación del modelo tiene sentido o no
- La aplicación asume que cualquier atributo con valores exclusivamente numéricos se debe tratar en el modelo como un atributo numérico. Si el atributo se quiere tratar como un atributo categórico, se recomienda modificar el conjunto de datos para que uno o más de los posibles valores del atributo contengan una letra. De esta manera, la aplicación reconoce el atributo como un atributo categórico. Por ejemplo, si se tiene un atributo con tres categorías, “1”, “2” y “3”, será interpretado como un atributo numérico. Para que sea interpretado como

un atributo categórico, los valores se pueden cambiar a “uno”, “dos” y “tres”; “A1”, “A2”, y “A3”; o hasta “A1”, “2” y “3”.

Capítulo 7. Reflexiones Finales

La investigación comenzó con el propósito de analizar las relaciones entre atributos de monedas específicos, crear modelos para predecir valores desconocidos en estos atributos y desarrollar una aplicación que utilizara estos modelos. Dadas las restricciones encontradas en el transcurso del trabajo, el producto final ha tenido que cambiar para ser una solución más amplia y flexible, que pueda ser utilizada por la doctoranda en el futuro con nuevos conjuntos de datos y atributos. Sin algunas de estas restricciones, particularmente el tiempo y los recursos disponibles, se pudo haber creado una aplicación aún mejor, que permitiera el uso de más algoritmos de minería de datos, la incorporación de técnicas de agrupamiento y el procesamiento de conjuntos de datos mucho más grandes.

Durante la etapa de análisis, no se encontraron patrones que pudieran ser utilizados para crear buenos modelos predictivos, pero el análisis de los atributos mostró relaciones interesantes entre algunos de los atributos de las monedas utilizadas para la investigación. Este análisis, junto con los gráficos generados en R, fue entregado a la doctoranda Soto para ser utilizado en su propia investigación. Uno de los descubrimientos importantes de la doctoranda, basado en el análisis de datos brindado, fue la relación entre la distancia de una ceca de Alejandría y la cantidad de monedas encontradas de esa ceca en Egipto. En la mayoría de los casos, entre más lejos estaba una ceca de Egipto, menos monedas se han encontrado de dicha ceca. La existencia de cecas atípicas que no cumplen con este patrón, como Roma, parece

implicar que existieron relaciones importantes entre estas ciudades y Egipto, donde se formaron más lazos de comercio de los que se formaron con otras ciudades romanas.

Ya que ninguno de los modelos creados dio buenos resultados con los datos obtenidos hasta el momento, se acordó con la doctoranda crear una aplicación que pudieran utilizar en sus trabajos a futuro. En el transcurso de los próximos años, ella seguirá recopilando datos de monedas romanas del Siglo IV encontradas en Egipto. Con la aplicación, una vez que obtenga nuevos datos, ella podrá crear sus propios modelos, evaluarlos, estudiar los árboles de decisión utilizados por los modelos, y utilizar los modelos para la predicción de atributos desconocidos.

Al darle la libertad al usuario de crear sus propios modelos, la doctoranda y la organización también podrán utilizar la aplicación para investigaciones acerca de otros temas del mundo antiguo, no necesariamente de monedas romanas del Siglo IV encontradas en Egipto.

Se espera que ampliar la solución final de esta manera permita que la aplicación sea utilizada en el futuro por numismáticos, arqueólogos e historiadores en todo tipo de proyectos e investigaciones.

Capítulo 8. Trabajos a Futuro

La aplicación creada como producto de esta investigación puede ser ampliada y mejorada de las siguientes maneras:

- Soporte para más plataformas (macOS y distribuciones de Linux)
- Creación de versión web que permita que la aplicación se utilice de cualquier dispositivo con navegador, sin tener que instalarla

- Incorporación de árboles de regresión para la predicción de valores continuos
- Incorporación de técnicas de minería de datos adicionales como redes neuronales, regresión lineal, agrupamiento, entre otras. Sería interesante agregar un algoritmo para la selección automatizada del mejor modelo utilizando varias de las técnicas.
- Análisis de datos automatizado con inclusión de gráficos.

Ya se han discutido algunos de los puntos más simples, como el soporte para macOS, con la doctoranda. Se estima que el 50% de sus colegas utilizan este sistema operativo, entonces se acordó que se realizará una vez concluida esta investigación. Los temas más complejos, como la incorporación de otros algoritmos y la automatización del análisis de datos, se podrían tomar como punto de partida para realizar otras investigaciones en el futuro.

Referencias

About ISAW. (2017). Recuperado de http://isaw.nyu.edu/about

Bauer, E., & Kohavi, R. (1999). An empirical comparison of voting classification algorithms: Bagging, boosting, and variants. *Machine learning, 36*(1), 105-139.

Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*. New York: Springer.

Diez, D. M., Barr, C. D., & Çetinkaya-Rundel, M. (2016). *OpenIntro Statistics*. OpenIntro.

Han, J., Kamber, M., & Pei, J. (2012). *Data mining: concepts and techniques*. Waltham, MA: Morgan Kaufmann.

Hernández, R., Fernández, C., & Baptista, M. (2014). *Metodología de la investigación.* México: McGraw-Hill.

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2015). *An Introduction to Statistical Learning: with Applications in R*. Springer.

Kelleher, J. D., Mac Namee, B., & D'Arcy, A. (2015). *Fundamentals of Machine Learning for Predictive Data Analytics: Algorithms, Worked Examples, and Case Studies*. Cambridge, MA: MIT Press.

Kuhn, M., & Johnson, K. (2016). *Applied Predictive Modeling*. New York: Springer.

The Leon Levy Foundation. (2015). Recuperado de http://isaw.nyu.edu/about/the-leon-levy-foundation

The Leon Levy Foundation – The Legacy of Leon Levy. (s.f.). Recuperado de http://leonlevyfoundation.org/

Meadows, A. & Gruber. E. (2014). Coinage and Numismatic Methods. A Case Study of Linking a Discipline. *ISAW Papers 7.15.*

Shmueli, G., Bruce, P. C., & Patel, N. R. (2010). *Data Mining For Business Intelligence*. John Wiley & Sons.

Sammut, C., & Webb, G. I. (2011). *Encyclopedia of Machine Learning*. New York: Springer.

Tan, P., Steinbach, M., & Kumar, V. (2005). *Introduction to Data Mining*. Pearson.

Zambanini, S., Kampel, M., & Schlapke, M. (2008). On the Use of Computer Vision for Numismatic Research. *9th International Symposium on Virtual Reality, Archaeology and Cultural Heritage 45(7)*: 17-24.

Zhu, Q., Wang, X., Keogh, E., & Lee, S. (2010). An efficient and effective similarity measure to enable data mining of petroglyphs. *Data Mining and Knowledge Discovery, 23(1)*, 91-127.

Apéndices

Apéndice 1. Carta de Aval

San José, 10 de abril del 2017

Irene Soto Marín
Doctoranda de la Universidad de Nueva York

El motivo de esta carta es para expresar el interés de ayudarle en su investigación acerca de la economía en Egipto y su relación con el resto del Imperio Romano en el siglo IV, y más específicamente, en el análisis de los datos que ha recopilado de monedas de este periodo encontradas en la región. Como ya le había comentado anteriormente, se desean crear y evaluar diferentes modelos de minería de datos utilizando los datos que se tienen de las monedas para poder encontrar patrones y relaciones entre ellos. Se considera muy importante brindarles la siguiente información, con el fin de que todo quede claro y poder demostrar el alto nivel de compromiso con este proyecto.

1. **Objetivo general:** Evaluar modelos de minería de datos para la clasificación e identificación de monedas romanas en Egipto producidas en el siglo IV.
2. **Objetivos específicos:**
   - 2.1. Identificar los datos disponibles de monedas romanas producidas en el siglo IV después de Cristo que hayan sido encontradas en Egipto
   - 2.2. Descubrir los datos que podrían ser útiles en la búsqueda de patrones
   - 2.3. Elaborar diferentes modelos de minería de datos para la clasificación de monedas
   - 2.4. Analizar los resultados de los modelos de minería de datos
   - 2.5. Interpretar los resultados de los modelos de minería de datos.
3. **Alcances**

Al completar la investigación, se entregará:

3.1. El documento con los resultados de la investigación y la evaluación de los diferentes modelos de minería de datos.

3.2. Aplicación que les permita agregar nuevos registros y producir sus propios resultados utilizando los modelos de minería de datos propuestos.

3.3. Guía para la utilización de la aplicación.

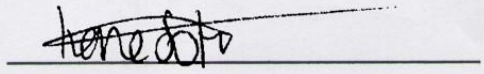
4. Limitaciones

4.1. Los modelos utilizados por la aplicación serán los mejores modelos encontrados durante la investigación utilizando el conjunto de datos proporcionado, y solo se espera que funcione con monedas similares y con los mismos parámetros utilizados en la investigación. Los resultados encontrados por la aplicación usando otros conjuntos de datos pueden contener errores o no brindar el mismo nivel de confianza.

5. Requisitos para este proyecto

5.1. Todos los datos a su disposición de monedas romanas fabricadas en el siglo IV y encontradas en Egipto que quiera tomar en cuenta para el estudio.

5.2. Reuniones por medio de Skype cuando sea necesario recopilar información o aclarar algún punto.



Apéndice 2. Cronograma

ACTIVIDAD	COMIENZO	FIN
Identificar datos disponibles	15/05/2017	19/05/2017
Aceptar última versión del conjunto de datos	15/05/2017	15/05/2017
Limpiar y darle formato a los datos	16/05/2017	19/05/2017
Descubrir datos útiles	20/05/2017	28/05/2017
Analizar datos en R para buscar correlación entre las variables	20/05/2017	27/05/2017
Seleccionar mejores variables para modelos	28/05/2017	28/05/2017
Elaborar modelos	28/05/2017	29/06/2017
Investigar acerca de algoritmos que podrían ser utilizados	28/05/2017	30/05/2017
Seleccionar algoritmos que serán probados	31/05/2017	31/05/2017
Usar algoritmos para crear varios modelos por variable	01/06/2017	29/06/2017
Analizar resultados	29/06/2017	30/07/2017
Ver resultados de los modelos y sus niveles de confianza	29/06/2017	01/07/2017
Seleccionar algoritmos con mejores resultados	02/07/2017	02/07/2017
Crear informe basado en resultados de los modelos, explicando las relaciones encontradas entre las variables y creando hipótesis de su significado junto con la doctoranda Irene Soto	02/07/2017	30/07/2017
Elaborar aplicación	31/07/2017	26/12/2017
Investigar y probar integración de Java y R por medio de rJava	31/07/2017	21/08/2017
Diseñar lógica de creación y utilización de modelos en R	22/08/2017	24/09/2017
Diseñar aplicación en Java y realizar conexiones con R	25/09/2017	28/10/2017
Crear script de instalación que instale todos los componentes necesarios para el funcionamiento de la aplicación	29/10/2017	18/11/2017

Probar el funcionamiento correcto del instalador y de la aplicación	19/11/2017	09/12/2017
Documentar la aplicación y grabar tutorial	10/12/2017	26/12/2017

Apéndice 3. Gráficos de Análisis de Atributos

Los gráficos generados por R Studio y utilizados para analizar la relación entre los atributos de las monedas se pueden encontrar en formato PDF en el siguiente enlace externo: https://drive.google.com/open?id=1vpvz5iB59Mx899zgOHWw9LA3UjR5LByn

Apéndice 4. Tutorial de Aplicación

Un tutorial básico para la utilización de la aplicación se puede encontrar en el siguiente enlace externo: https://youtu.be/VPB_0y5Zqnw

Apéndice 5. Aprobación de Cambios

universidad cenfotec_

Ana Cristina Caldas Donato <acaldasd@ucenfotec.ac.cr>

Confirmación de cambios en el alcance de la investigación

2 mensajes

Ana Cristina Caldas Donato <acaldasd@ucenfotec.ac.cr> 30 de marzo de 2018, 21:30
Para: irene.soto.marin@gmail.com

Hola Irene,

Aunque ya discutimos esto en persona, quería comunicarle por este medio, oficialmente, los cambios que se realizaron al alcance de la investigación. Los cambios se tuvieron que realizar porque los datos recopilados hasta el momento no han logrado crear buenos modelos predictivos. Por esta razón, se creó una aplicación que les permita, una vez que recopilen más datos, crear sus propios modelos, probarlos, y realizar predicciones con ellos. Le agradecería que conteste a este correo confirmando que está de acuerdo con los cambios.

1. **Objetivo general:** Elaborar una aplicación de escritorio que permita crear, evaluar y usar modelos de minería de datos para la clasificación e identificación de monedas romanas en Egipto producidas en el siglo IV.
2. **Objetivos específicos:**
   - 2.1. Identificar los datos disponibles de monedas romanas producidas en el siglo IV después de Cristo que hayan sido encontradas en Egipto
   - 2.2. Descubrir los datos que podrían ser útiles en la búsqueda de patrones
   - 2.3. Elaborar diferentes modelos de minería de datos para la clasificación de monedas
   - 2.4. Analizar los resultados de los modelos de minería de datos y seleccionar los mejor modelos.
3. **Alcances**

Al completar la investigación, se entregará:

- 3.1. El documento con los resultados de la investigación y la evaluación de los diferentes modelos de minería de datos.
- 3.2. Una aplicación de escritorio que les permita a usted y a sus colegas crear sus propios modelos de minería de datos, evaluarlos con conjuntos de datos de prueba, y luego producir sus propias predicciones utilizando los modelos de minería de datos..
- 3.3. Guía para la utilización de la aplicación.

4. **Limitaciones**
   - 4.1. Los modelos creados con la aplicación utilizarán árboles de clasificación, ya que fueron los que presentaron mejores resultados a lo largo de la investigación. Por lo tanto, los modelos creados por el usuario solamente podrán predecir variables categóricas y no variables continuas.

Gracias,

Ana Cristina Caldas

Irene Soto <irene.soto.marin@gmail.com> 31 de marzo de 2018, 1:01
Para: Ana Cristina Caldas Donato <acaldasd@ucenfotec.ac.cr>

Querida Ana,
Gracias por el email. De acuerdo a lo que hemos discutido previamente, estoy de acuerdo y apoyo los cambios que han sido descritos para el proyecto, ya que considero que no son sólo necesarios sino que aumentarán la calidad y efectividad del producto final y el impacto del proyecto.

Cordialmente,
Irene Soto Marín
[El texto citado está oculto]
--
Irene Soto, MPhil
PhD Candidate
Institute for the Study of the Ancient World, New York University
15 East 84th St.
New York, NY 10028

Apéndice 6. Carta de Aprobación

Basilea, Suiza, 4 de abril del 2018

A quien corresponda:

Por medio de la presente le saludo y hago de su conocimiento que he recibido la aplicación DT Model Creator, creada por la estudiante Ana Cristina Caldas, junto con los resultados de su investigación para apoyarme en el análisis de datos de mi proyecto de investigación.

El análisis de datos realizado por Ana Cristina fue de gran ayuda para mi tesis doctoral, y ha sido una agradable sorpresa que la aplicación que creó funcione con objetos arqueológicos de todo tipo, cuando originalmente solo esperábamos aplicarla a monedas del siglo IV.

Agradezco el apoyo recibido y reitero que estoy muy satisfecha con el trabajo realizado. Confiamos que será una herramienta muy útil para nosotros y esperamos poder trabajar con Ana Cristina en el futuro.

Atentamente,

Irene Soto Marín

Doctoranda

Institute for the Study of the Ancient World, New York University

Apéndice 7. Manual de Usuario

DT Model Creator
User Guide

Contents

Minimum System Requirements

- Windows 7 and above, 64-bit versions only
- Intel Core i3 or better
- 4 GB RAM or above
- 500 MB hard drive space

Note: These specifications can be used with small data sets and basic data models. The larger the data sets and the more complex the data models, the more resources they will require. A complex model that is created easily on a server may fail to be created on a personal computer with limited resources.

Installing DT Model Creator

To install DT Model Creator, select a folder where you want to store its files and extract the contents of the zip file there. The folder should now include three folders and one JAR file.

Open the folder "setup". Right-click "install.bat" and select "Run as administrator". This will begin the setup process.

During the setup process, unless a recent version of Java is found on the machine, a new folder named "JRE" will be created and JRE 8 Update 151 will be installed under it. The folder "R" will also be created, and R 3.4.2 will be installed. The installer will install all required R libraries and will add set several environment variables. A shortcut named "DT Model Creator" will be added to the desktop. Once the installation process is complete, a message will be displayed. It is necessary to restart the computer before using DT Model Creator.

Note: If the installation has completed successfully, the folder "setup", under the DT Model Creator installation path, can be deleted to decrease the amount of disk space being used by the application. Only the setup folder should be deleted. The following files and folders in the same path should NOT be deleted and are required by DT Model Creator:

- DTModelCreator.jar
- JRE
- R

- Lib
- Res

Deleting any of these files, folders, or its contents, would prevent DT Model Creator from running.

Creating a New Model

DT Model Creator allows users to create their own classifier data models. To do this, DT Model Creator needs data files. These data files should meet the following requirements:

- The data needs to be saved in a CSV file
- The data must be organized as a table. Each column should correspond to one attribute, and each row should correspond to one record.
- The first row of each column should contain the name of the attribute.
- A minimum of two columns are required. One column should correspond to the attribute that the new model will try to predict. One or more additional columns should correspond to the attributes that will be used to create the prediction.
- Since DT Model Creator uses classifier trees, the predicted attribute must be a categorical variable. This means that it can only take one of a limited number of possible values (e.g., mint, hoard, denomination). A categorical variable's possible values can also usually be considered labels, classes, or factors.
- The predictor attributes can be both categorical and numeric variables. Numeric variables contain continuous values (e.g., size, weight, latitude, longitude).
- Any unknown values in the data file should be replaced with "?". By doing this, DT Model Creator will recognize them as unknowns and will not factor them into the models.
- Columns with only numeric values will be interpreted as numeric attributes. If any value in the column contains a letter or symbol, with the exception of unknown values ("?"), the entire attribute will be interpreted as non-numeric, and therefore a categorical variable.
- DT Model Creator is case-sensitive. If the attribute "metal" contains rows with values "Bronze", "bronze", and "BRONZE", the model will interpret each of them as a distinct value, which may skew the results.

Welcome Screen

To begin the model creation process, open DT Model Creator. On the Welcome window, select "New Model" and click "OK".

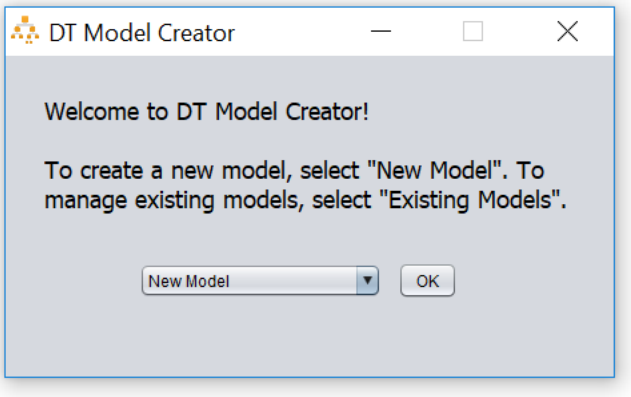


Figure 1: Welcome Screen

Training Dataset

To create a data model, a training data set and a test data set are required. DT Model Creator will first prompt you to select your training data set.

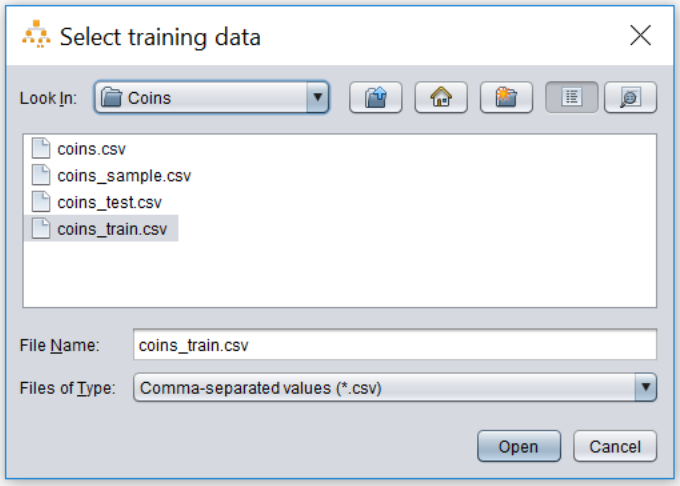


Figure 2: Selecting a training data set

Test Dataset

Once the training data set has been selected, DT Model Creator will ask if you would like to use this data to create the test data set as well. If you select "Yes", 30% of the records in your training data set will be removed and will be used to create your test data set. If you select "No", DT Model Creator will prompt you to select your test data set.

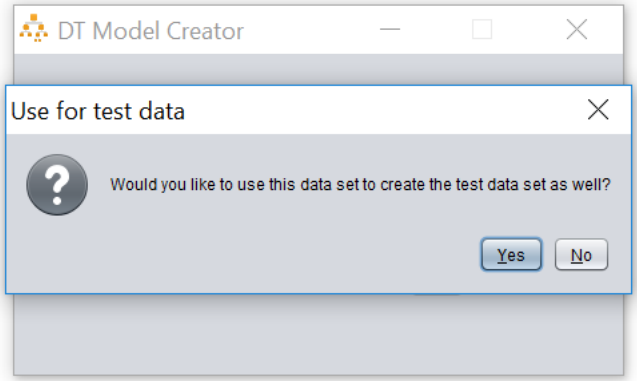


Figure 3: Confirm if test data should be taken from training data or another data file

If you select "No" and choose your own test data set, make sure both data sets contain the columns that you plan on using for your data model.

Predicted Attribute

If the files are loaded successfully, DT Model Creator will ask you which attribute you would like the data model to predict. Select the attribute and click on "Next".

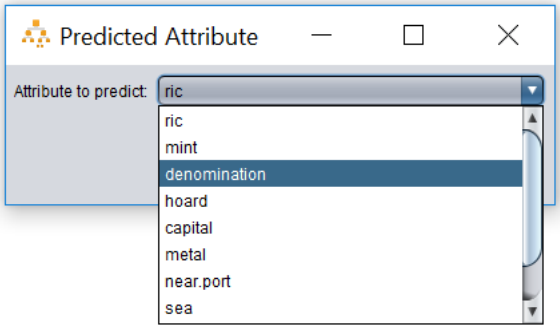


Figure 4: Select the attribute that the model will try to predict

The next screen will display all possible values or factors for the attribute selected and will display how many records (occurrences) there are of each. It is usually best if all factors have a similar number of occurrences. If there are significantly more occurrences of one label than another, the model might be skewed and always tend to predict the factor with more occurrences.

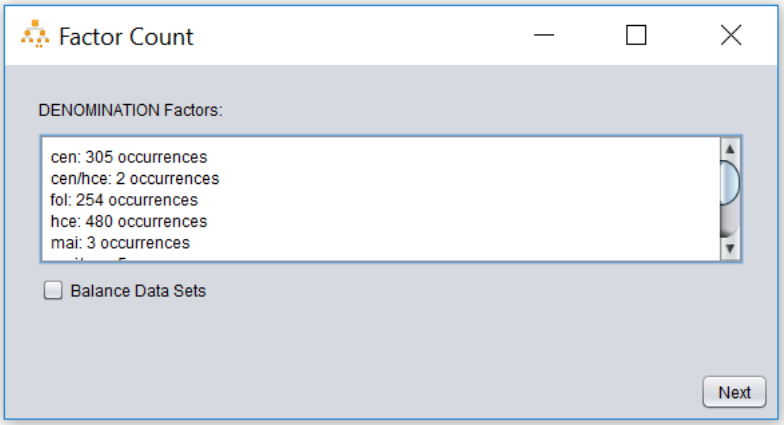


Figure 5: The factor count window displays how many occurrences there are of each factor

Balancing Data Sets

DT Model Creator provides the option to balance data sets.

Leaving "Balance Data Sets" unchecked will leave the training data set unchanged. If the checkbox "Balance Data Sets" is selected, four methods of balancing will be displayed: by mean, by mode, by median or a fixed number provided by the user.

The number of occurrences will be displayed on the bottom left corner, as " [_] occurrences per factor", to show the user the number of occurrences that each factor will end up with. Each factors' number of occurrences will be increased or decreased to get as close as possible to the selected number.

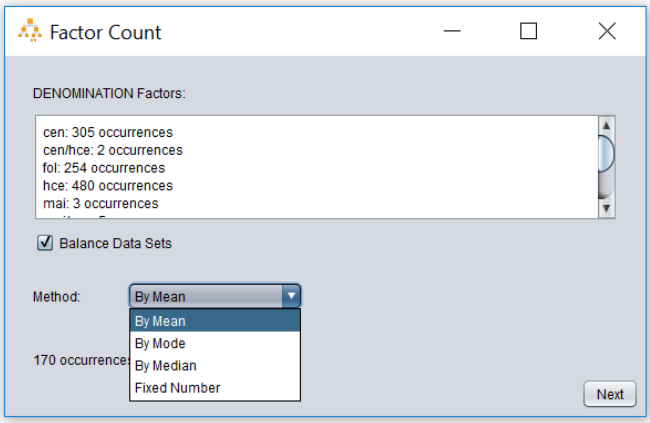


Figure 6: Data sets can be balanced using the mean, mode, and median, or by specified a specific number

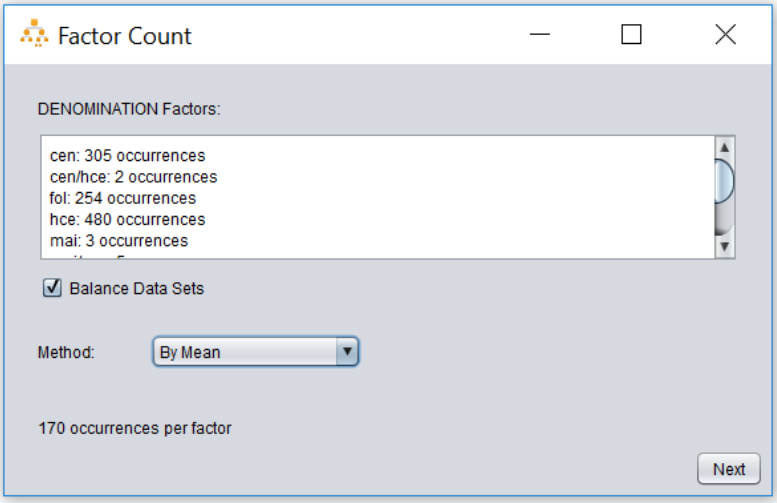


Figure 7: The number of occurrences per factor after balancing appears at the bottom left corner

For example, if my training data set contains 5000 bronze coins and 200 gold coins, I can choose to balance the data set so that they have a similar number of occurrences. If I were to set the number of occurrences at 200, 4800 randomly selected records with bronze coins will be ignored by DT Model Creator, so that each possible value (factor/class) has 200 occurrences. If I set the number of occurrences to 1000, then 4000 randomly selected bronze coin records will be ignored. As for the gold coins, since there are only 200 of them, the records will be copied until they reach 1000, meaning that four duplicates of each of the 200 gold coins will be added to the training data set.

Although this example will give us a balanced data set, this may present its own problems. By ignoring 80% of all bronze coin records, we are dismissing data that may be useful for the model, and by inflating the amount of gold coin records from 200 to 1000, we are giving gold coin records five times more weight in the model than they should actually have.

It is difficult to say when data sets should be balanced and when they should remain unbalanced. Many believe balancing data sets improves classifier performance, while others thinks it makes the models inaccurate. Both approaches can be tested using DT Model Creator, so multiple models can be created and the one with the best results can be selected and used. Usually, artificially balanced data sets provide better average results per class, while unbalanced data sets provide better average results overall.

Once you are ready to continue with the model creation, click "Next".

Selecting Predictors

The next window will give you the option of selecting the predictors you would like to use in your data model. Categorical variables will display the amount of factors (i.e., possible values) they have. Predictors with a low numbers of factor usually have better results and are easier to compute. DT Model Creator will warn you if one of selected predictors contains too many factors. Depending on the computer's specifications, if the data model is too complex and contains too many factors, DT Model Creator may not have enough resources available to create it, and may freeze or close unexpectedly.

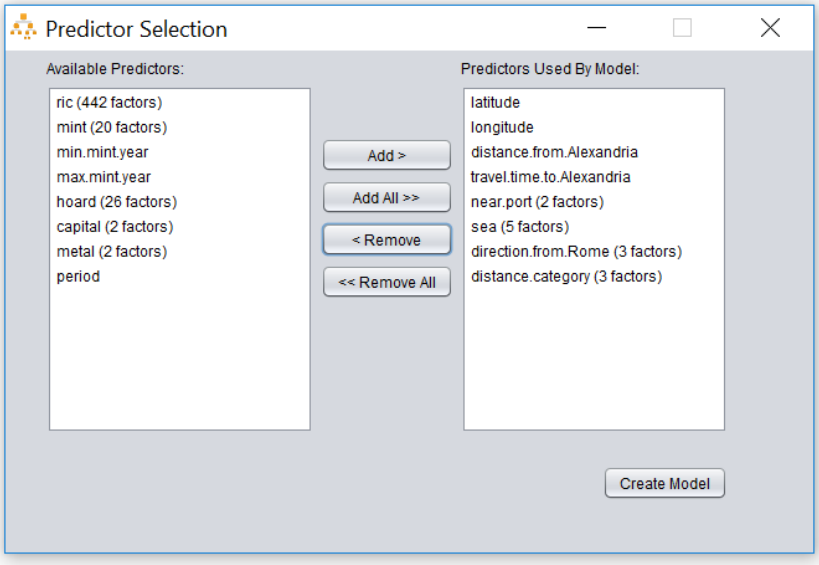


Figure 8: The attributes used for the prediction are specified in the Predictor Selection screen

Once all predictors have been selected, click on "Create Model".

DT Model Creator will try to create a model using the specified predictors. Out of these manually selected predictors, DT Model Creator will choose the best ones, using those for the model, and will discard the rest.

Interpreting Results

When the model has been created, it will be tested using the test data set, and the results will be displayed onscreen using a confusion matrix. Rows represent the actual values and columns represent the predictions. The number in a cell with the same row and column name is the number of correct predictions for that particular class. Any other cells in that row correspond to incorrect predictions.

The last column contains the "Accuracy Per Class". This field contains the percentage of occurrences that were predicted correctly for that particular class. On the bottom left corner, there are two other accuracy percentages to take into account. "Model Accuracy (Overall)" calculates the percentage of overall predictions that are correct, regardless of class. "Average Accuracy Per Class" is an average of all "Accuracy Per Class" percentages.

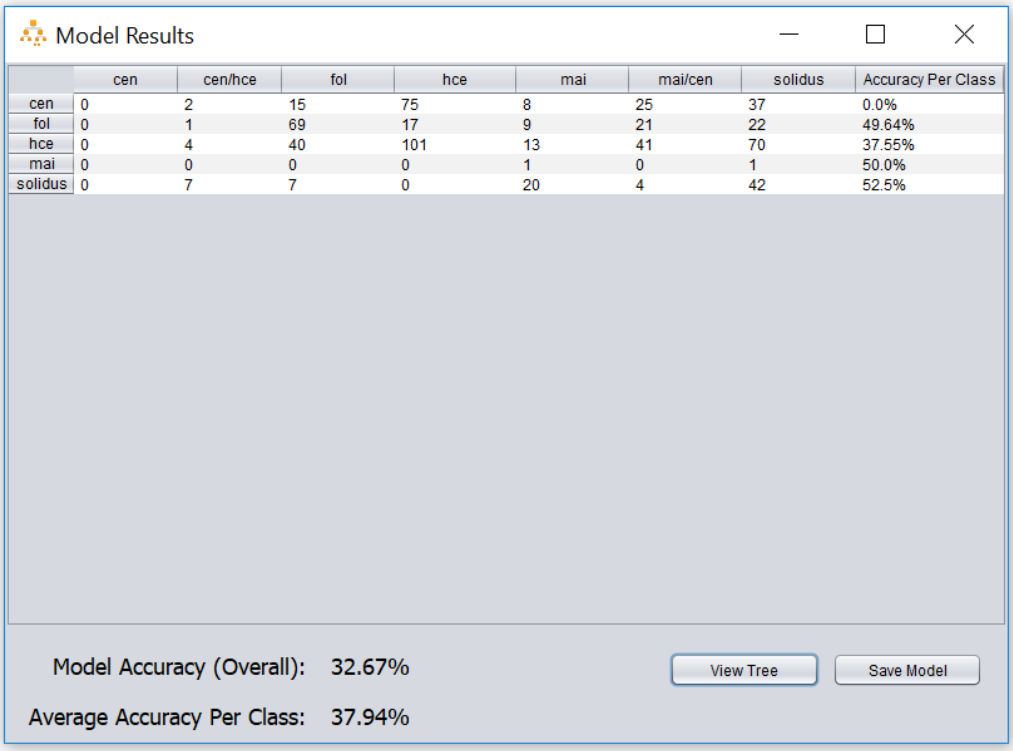


Figure 9: The model's prediction results are displayed in a confusion matrix

It is important to understand what these percentages mean when evaluating a model. For example, let's say a new model's results are:

	A	B	C	Accuracy Per Class
A	9900	0	0	100.0%
B	27	0	0	0.0%
C	73	0	0	0.0%

Model Accuracy (Overall): 99.00%
Average Accuracy Per Class: 33.33%

In this scenario, the new model gives us an overall accuracy of 99%. This may seem promising at first, but the average accuracy per class is only 33%. The "Accuracy Per Class" column tells us that "A" was predicted correctly 100% of the time, but B and C were predicted correctly 0% of the time. If we take a closer look at the results, it looks like the new model predicts "A" for every single record. Since most records in the data set were in fact A, the overall accuracy seems great, but the model did not get any records right for "B" or "C". It is very likely that the model is blindly choosing "A", and will never identify B and C correctly.

Decision Tree

We can see exactly what the model is doing by clicking on "View Tree". This will open a new window that will display the decision tree used by the model.

When a model tries to make a prediction, it checks if a certain attribute (one of the predictors selected by the model during its creation) meets a condition. If it does, it will make a prediction based on that criterion, or will check if other attributes meet other conditions until it has enough data to make a prediction. The image that is displayed onscreen after clicking on "View Tree" is a graphical representation of the decision tree used by the model. The nodes surrounded by circles are the final predictions, and all of the conditions that led to that prediction can be traced by following the path from the node to the top of the tree.

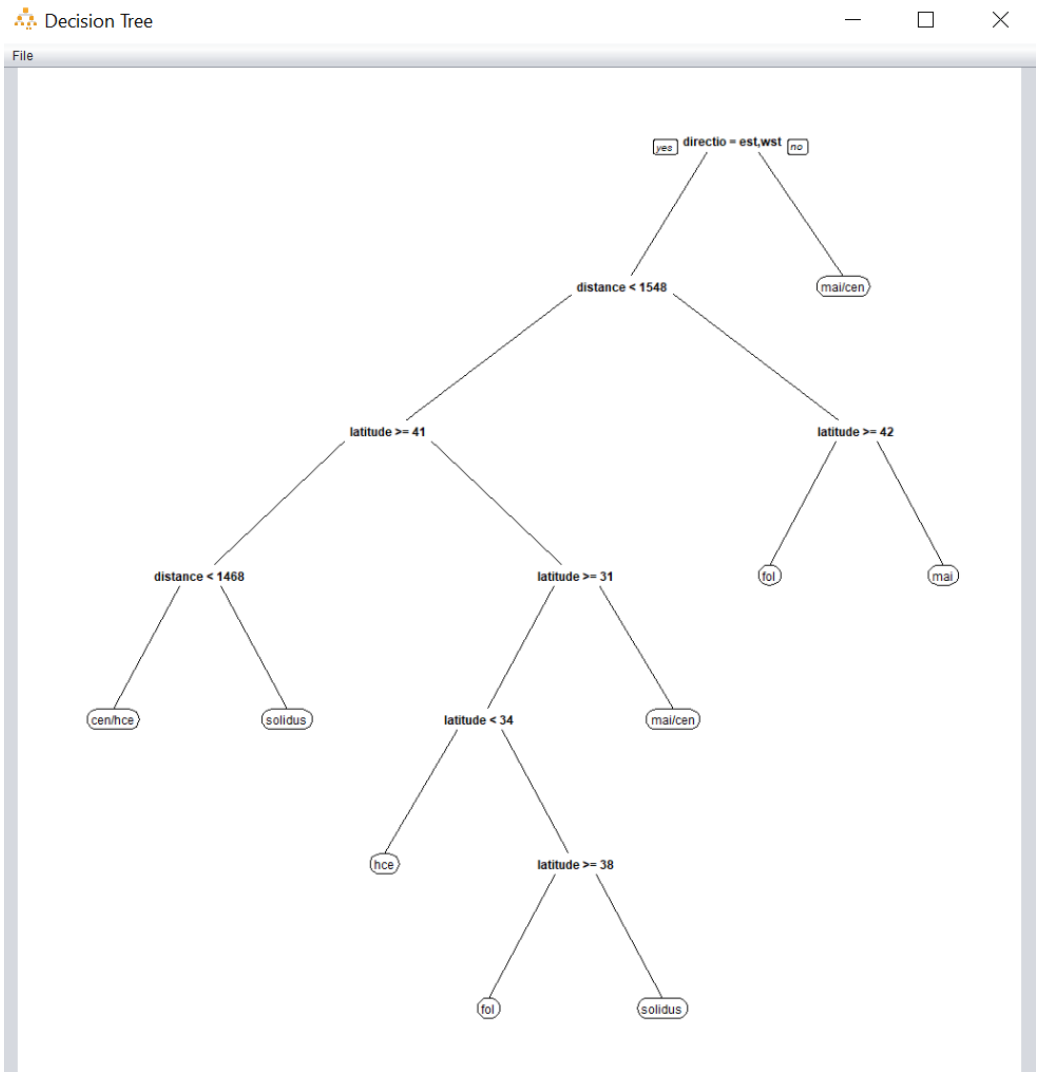


Figure 10: This tree is a visual representation of the predictive model's logic

Viewing a model's decision tree can help users understand how the model works and if the criteria that it is using to make its predictions is valid or not. It may also lead to discoveries of relationships between attributes.

The image can be saved by selecting File – Save As… or can be later viewed if the model is saved.

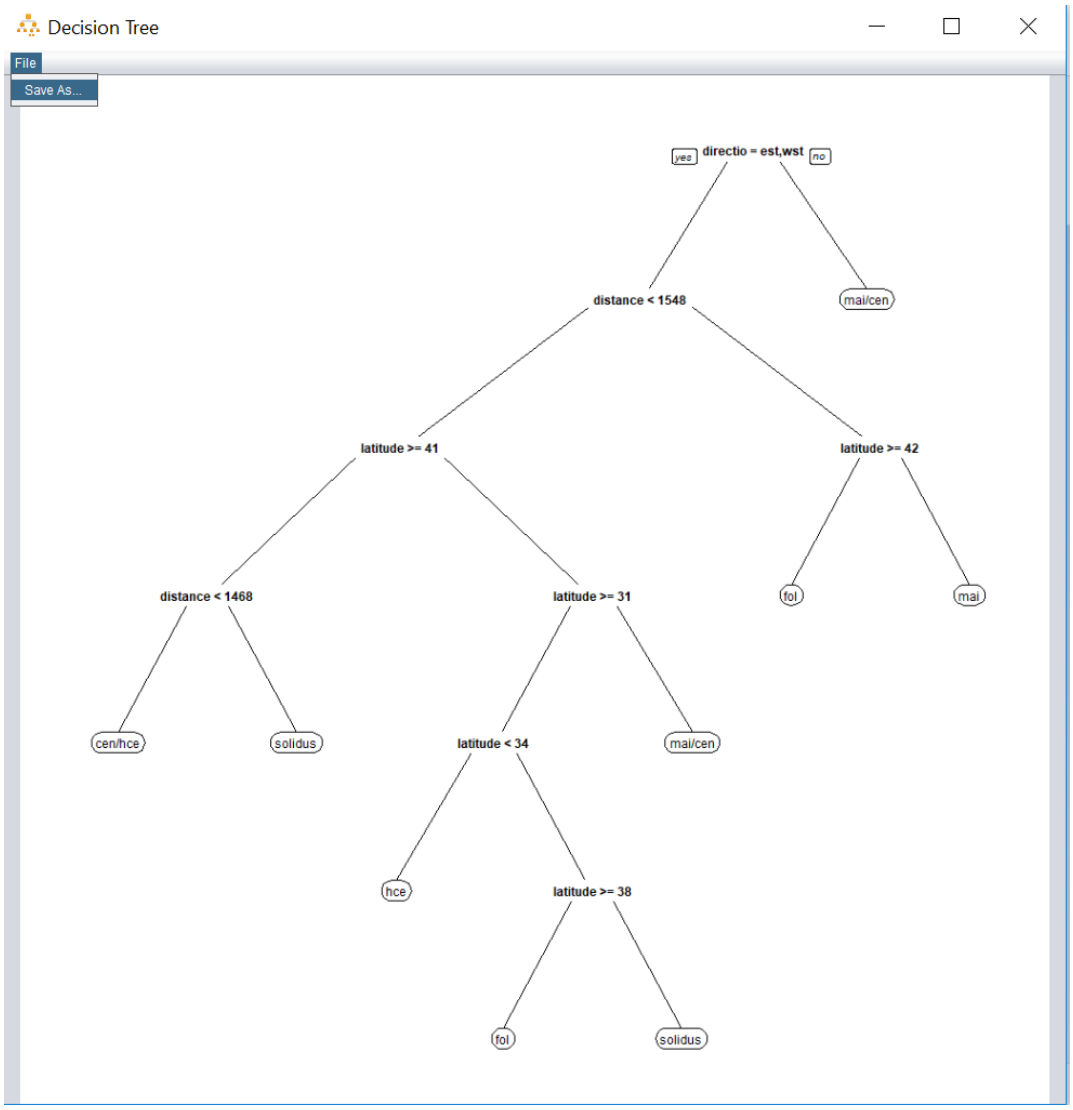


Figure 11: These trees can be saved as PNG files using the Save As... option

Saving the Model

If the model is good and you would like to save it and use it with other data sets in the future, click on "Save Model" and specify a name that you can use to identify it. The model will be saved in DT Model Creator's default location, and can be exported to another location if necessary.

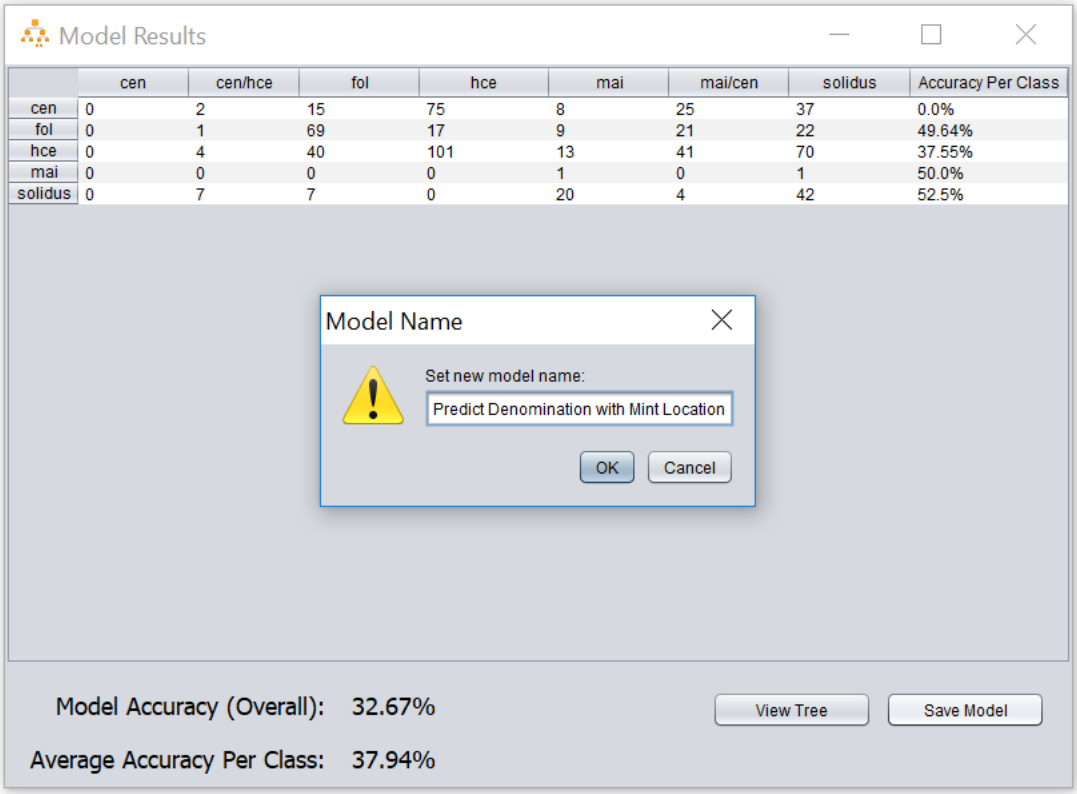


Figure 12: The model's data is stored in a MOD file with the name that is set in this screen. The name should be descriptive, but must also meet the operating system's file naming conventions

Close the results window to create a new model or to manage your existing models.

Managing Existing Models

Once models have been created and saved, they can be accessed by opening DT Model Creator, selecting "Existing Models" and clicking "OK".

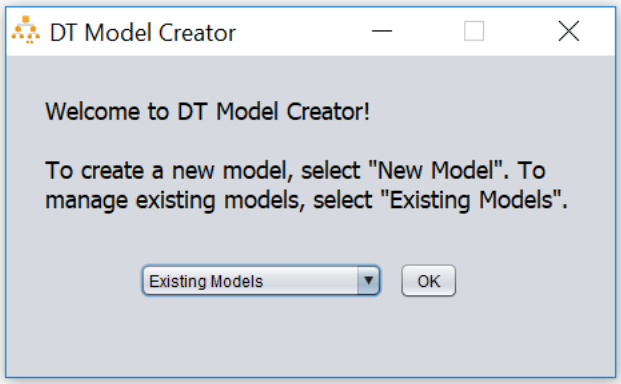


Figure 13: Select "Existing Models" in the Welcome Screen to manage and use models

Main Management Window

All saved models will be displayed in DT Model Creator's management window. When a model is selected, one of the following actions can be performed:

- View Tree
- Test Model
- Use Model
- Delete Model
- Export Model

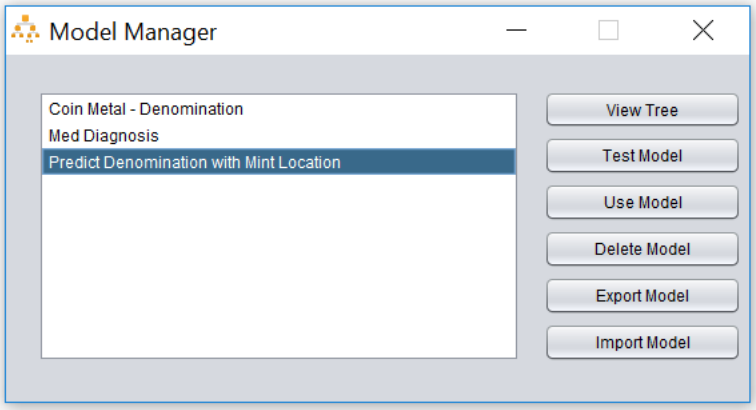


Figure 14: Select the model to perform any of the actions on the right, except for "Import Model"

View Tree

This option will display the graphical representation of the model's decision tree. Please see "Creating a New Model – Decision Tree"

Test Model

Although two data sets may contain similar data, the patterns and relationships between their attributes may not be similar. For example, if a model is created with a data set that contains data about coins from a particular hoard, the same model may not be useful when used on coins from another hoard. If the test data set that was used initially only contained data from the first hoard, we may not be able to know if the model is useful with other hoards.

The "Test Model" option is available to test models again using different test data sets.

If "Test Model" is selected, DT Model Creator will ask the user to select a new test data set. The test data set must contain all the attributes used in the model (you can find these by using the "View Tree" option in the Main Management Window), including the predicted attribute.

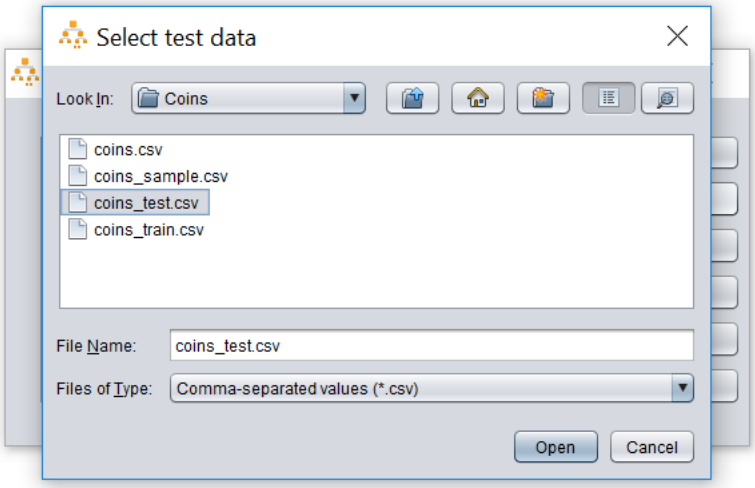


Figure 15: Another data set can be selected and tested against the model

If the test data is valid, the model will make predictions using the new data, and a results window with a confusion matrix and accuracy percentages will be displayed.

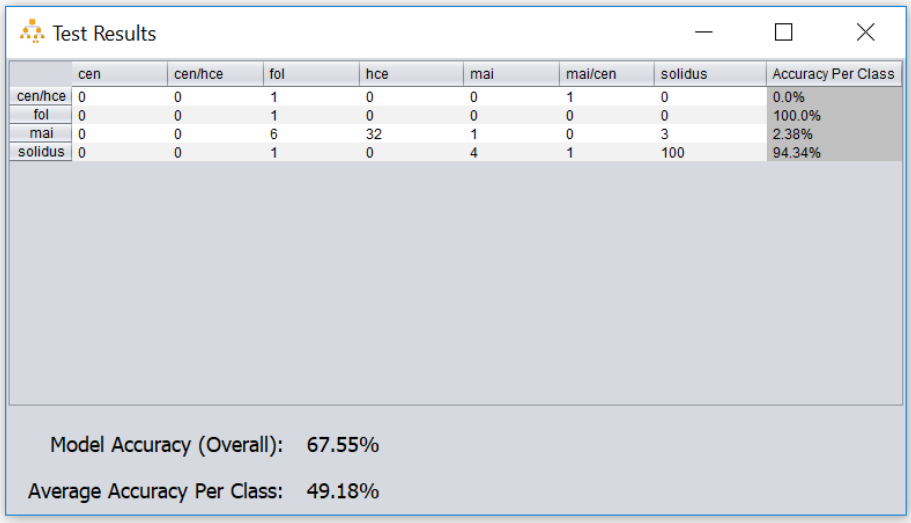


Figure 16: The test results and the model results will display different results if different test data sets are used

Use Model

The main purpose of a DT Model Creator model is to predict an object's unknown attribute by using its known attributes. To do this, select the model and click on "Use Model".

DT Model Creator will display a prompt, asking for the data that will be used to create the prediction. This data file will need to contain all attributes used by the model to make its prediction (i.e., predictors).

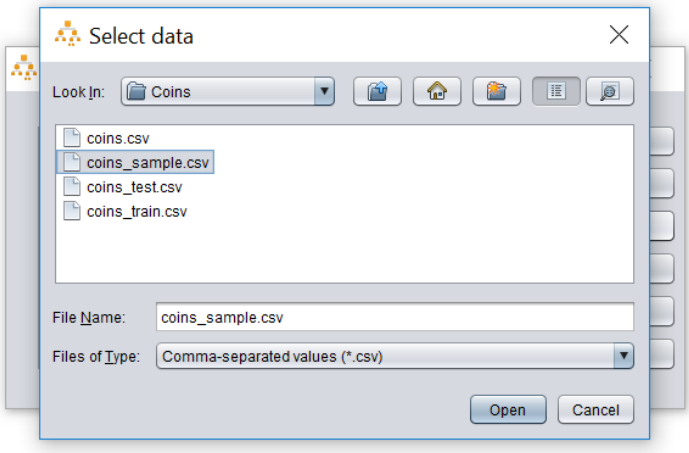


Figure 17: A data set containing all of the predictors required by the model should be selected

If a record in the data file contains a new predictor value that the model does not recognize, the model will ignore this record and will not make a prediction for it. This may occur if the training data set used to create the model did not contain this new value. For example, if a model uses the attribute "metal" as a predictor, and the training data set used to create the model only contained two metal classes, "bronze" and "gold", the model will not know what to do with silver and copper coins. If we later try to use the model with a data set that contains coins labelled as "silver" and "copper", these records will be ignored and will not be displayed in the final results.

After selecting the data used to make the prediction, DT Model Creator will ask for a save location for its predictions. Select the location where this file will be saved.

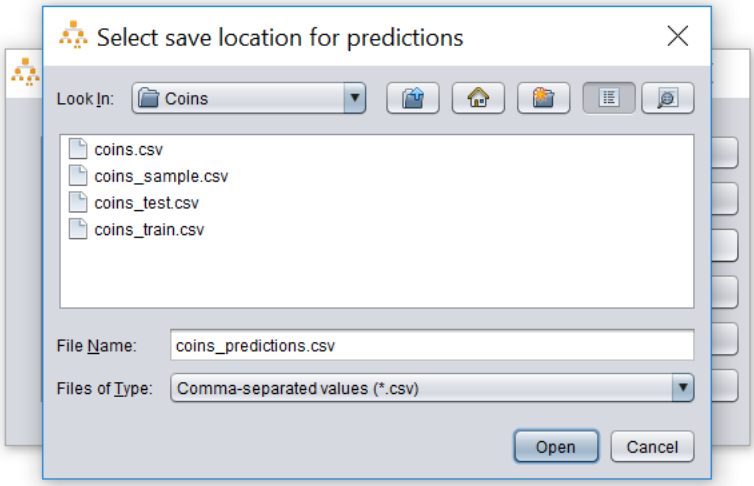
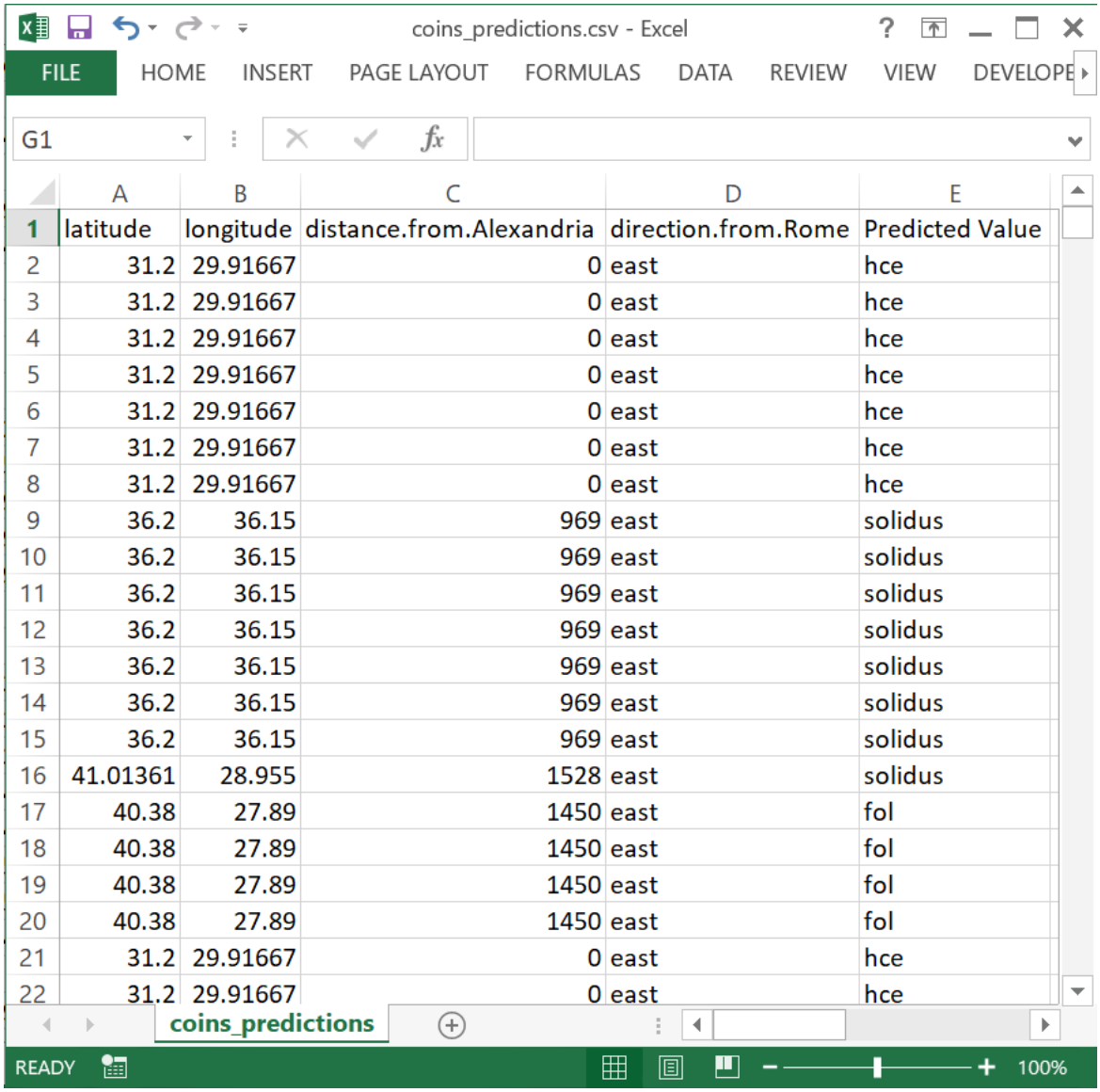


Figure 18: The predictions will be stored in a CSV file, in the location specified by the user

This file is essentially a copy of the data file provided in the previous screen, but the records with unknown attribute values have been removed and a new column named "Predicted Value" has been added. This column contains the model's prediction for each one of the records.



	A	B	C	D	E
1	latitude	longitude	distance.from.Alexandria	direction.from.Rome	Predicted Value
2	31.2	29.91667	0	east	hce
3	31.2	29.91667	0	east	hce
4	31.2	29.91667	0	east	hce
5	31.2	29.91667	0	east	hce
6	31.2	29.91667	0	east	hce
7	31.2	29.91667	0	east	hce
8	31.2	29.91667	0	east	hce
9	36.2	36.15	969	east	solidus
10	36.2	36.15	969	east	solidus
11	36.2	36.15	969	east	solidus
12	36.2	36.15	969	east	solidus
13	36.2	36.15	969	east	solidus
14	36.2	36.15	969	east	solidus
15	36.2	36.15	969	east	solidus
16	41.01361	28.955	1528	east	solidus
17	40.38	27.89	1450	east	fol
18	40.38	27.89	1450	east	fol
19	40.38	27.89	1450	east	fol
20	40.38	27.89	1450	east	fol
21	31.2	29.91667	0	east	hce
22	31.2	29.91667	0	east	hce

Figure 19: The new predictions file will contain all of the columns the data set contained, plus an additional "Predicted Value" column with the prediction

Delete Model

To delete a model, select it and click on “Delete Model”. This will delete it from DT Model Creator’s default location.

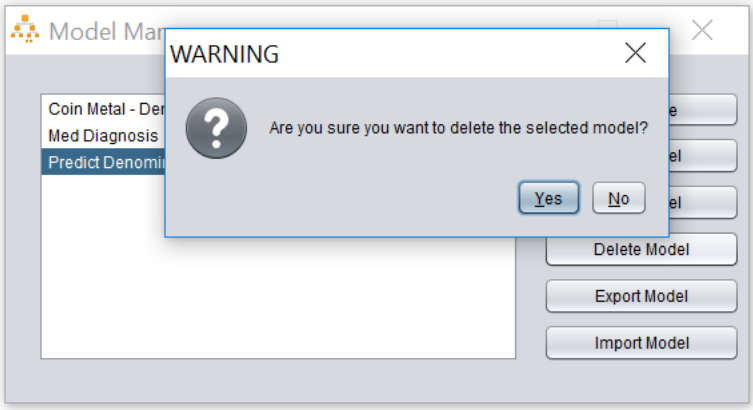


Figure 20: A warning will be displayed before deleting a model

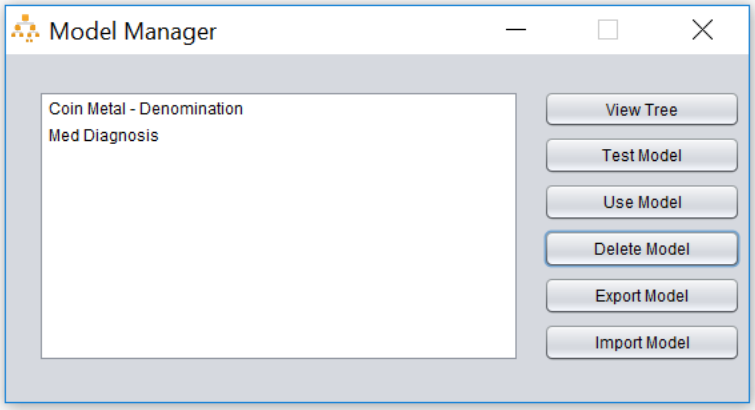


Figure 21: Once a model is deleted, it will be removed from the Model Manager

Export Model

To export a model, select it and click on "Export Model".

DT Model Creator will ask for the location where the model should be saved. A new file with the extension "*.mod" will be generated.

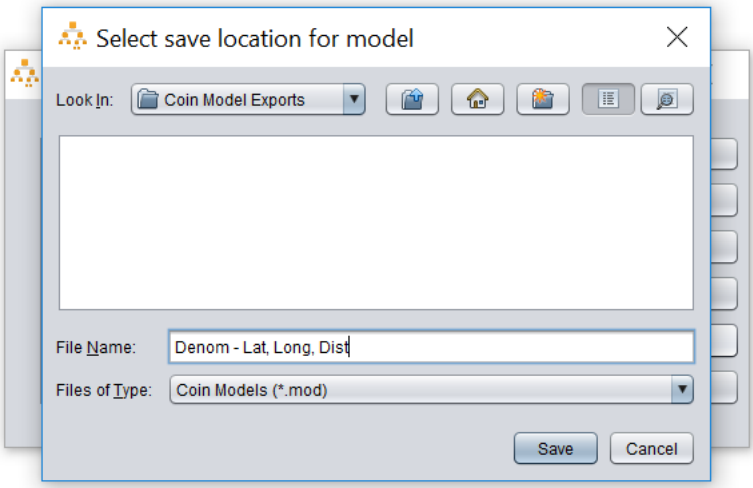


Figure 22: Any model can be exported to a MOD file. It can be imported into DT Model Creator on another computer

Import Model

DT Model Creator can import models that have been exported on other machines. To import a model, click on "Import Model".

DT Model Creator will ask for the location of the model. Select the exported model and click on "Open".

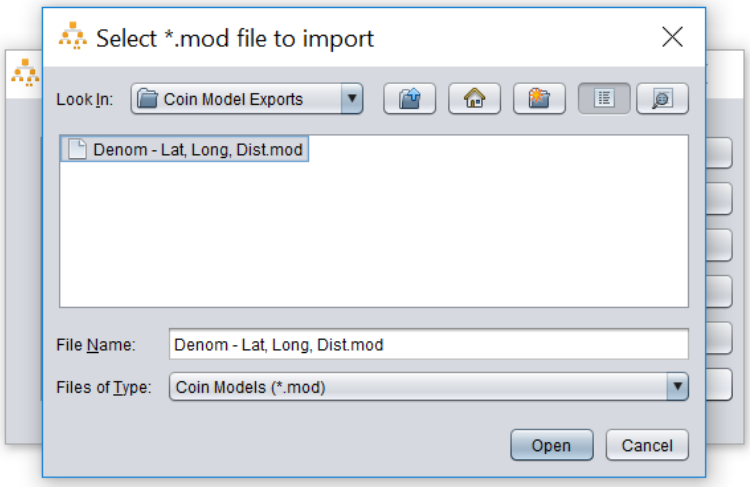


Figure 23: MOD files created with DT Model Creator can be imported. The model's name in Model Manager will be the MOD file's file name.

The model should now show up in DT Model Creator's list of available models.

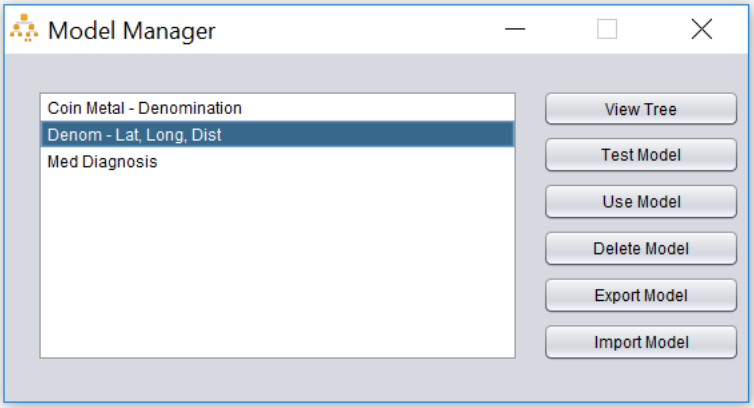


Figure 24: The model will appear in Model Manager after it has been imported