ESCUELA POLITÉCNICA NACIONAL

FACULTAD DE INGENIERÍA DE SISTEMAS

DESARROLLO DE DOS MODELOS DE CLASIFICACIÓN USANDO BOOSTING Y REDES NEURONALES PARA CODIFICAR LAS ACTIVIDADES ECONÓMICAS Y OCUPACIONES DE INVESTIGACIONES SOCIODEMOGRÁFICAS DEL INEC

PROYECTO PREVIO A LA OBTENCIÓN DEL TÍTULO DE MAGÍSTER EN SISTEMAS DE INFORMACIÓN
MENCIÓN INTELIGENCIA DE NEGOCIOS Y ANALÍTICA DE DATOS MASIVOS

DIANA CAROLINA MÉNDEZ MORENO
diana.mendez@epn.edu.ec

DIRECTOR: MARCO E. BENALCÁZAR, PhD.
marco.benalcazar@epn.edu.ec

Quito, septiembre 2023

APROBACIÓN DEL DIRECTOR

Como director del trabajo de titulación “Desarrollo de dos modelos de clasificación usando boosting y redes neuronales para codificar las actividades económicas y ocupaciones de investigaciones sociodemográficas del INEC” desarrollado por Diana Carolina Méndez Moreno, estudiante de la Maestría en Sistemas de Información, habiendo supervisado la realización de este trabajo y realizado las correcciones correspondientes, doy por aprobada la redacción final del documento escrito para que prosiga con los trámites correspondientes a la sustentación de la defensa oral.

Firmado electrónicamente por:
MARCO ENRIQUE
BENALCAZAR PALACIOS

Marco Enrique Benalcázar Palacios

DIRECTOR

DECLARACIÓN DE AUTORÍA

Yo, Diana Carolina Méndez Moreno, declaro bajo juramento que el trabajo aquí descrito es de mi autoría; que no ha sido previamente presentada para ningún grado o calificación profesional; y, que he consultado las referencias bibliográficas que se incluyen en este documento.

La Escuela Politécnica Nacional puede hacer uso de los derechos correspondientes a este trabajo, según lo establecido por la Ley de Propiedad Intelectual, por su Reglamento y por la normatividad institucional vigente.

Diana Carolina Méndez Moreno

DEDICATORIA

A mis padres y hermanos por su amor incondicional y apoyo. A mis amigos por sus buenas platicas, sabios consejos y ocurrencias. A Eddy por compartir su vida y anhelos conmigo, por motivarme, por cuidarme y quererme. A todos aquellos que han sido parte de este aprendizaje.

AGRADECIMIENTO

A mi director de tesis y amigo PhD. Marco E. Benalcázar, por brindarme su valioso tiempo y por haberme guiado de manera acertada en el desarrollo de este proyecto.

A los profesores de la maestría en Sistemas de Información, por plantar en mí la semilla de la duda respecto a varios temas relacionados a los datos. Gracias por motivarme a aprender cosas nuevas.

AI Instituto Nacional de Estadística y Censos INEC por ser mi espacio de aprendizaje en el cual pude comprender la enorme responsabilidad que conlleva la producción de estadísticas oficiales y la necesidad de incorporar herramientas tecnológicas innovadoras en la recolección, procesamiento, análisis y visualización de datos.

ÍNDICE DE CONTENIDO

LISTA DE FIGURAS

LISTA DE TABLAS

LISTA DE ANEXOS

RESUMEN

En las oficinas estadísticas e instituciones que recopilan datos de las características laborales de las personas, se realiza la codificación de las variables textuales actividad económica y ocupación. Dicha actividad sirve para facilitar el procesamiento de los datos y generar indicadores relevantes para la planificación gubernamental. En el Instituto Nacional de Estadística y Censos de Ecuador (INEC), la codificación es realizada por personas entrenadas para esta actividad. Este tipo de codificación denominada manual requiere de un gran número de personas y puede durar tiempos extensos dependiendo de la cantidad de datos. Por ejemplo, en el Censo de Población y Vivienda del 2010 la codificación fue realizada por 310 personas durante 5 meses. Por lo mencionado, en esta tesis de maestría se desarrolló dos modelos de clasificación para codificar automáticamente las actividades económicas y ocupaciones de investigaciones sociodemográficas del INEC. Para el desarrollo de los modelos se utilizó los algoritmos Xgboost y Redes Neuronales Artificiales de tipo Feedforward y LSTM. Los modelos con mejor rendimiento se obtuvieron usando las Redes Neuronales Feedforward, con una exactitud de 95.18% para actividad económica y 86.85% para ocupaciones. En comparación con la codificación manual, la implementación de los modelos para codificar automáticamente alrededor de 15.000 actividades económicas y ocupaciones permitió reducir el tiempo de días a minutos. Además, considerando un enfoque combinado (automático y manual), en el cual la tasa de error de los modelos fue menor al 5%, el tiempo se redujo a la cuarta parte y la cantidad de personal a la mitad respecto a la codificación manual.

Palabras clave: codificación automática, boosting, redes neuronales, clasificación de texto

ABSTRACT

In statistical offices and institutions that collect data on the labor characteristics of individuals, the textual variables economic activity and occupation are coded. This activity serves to facilitate data processing and generate relevant indicators for government planning. At the National Institute of Statistics and Census of Ecuador (INEC), coding is carried out by people trained for this activity. This type of coding, called manual coding, requires a large number of people and can take a long time depending on the amount of data. For example, in the 2010 Population and Housing Census, coding was performed by 310 people lasting 5 months. Because of this, in this master's thesis two classification models were developed to automatically code the economic activities and occupations of sociodemographic research of INEC. For the development of the models, we used Xgboost and Artificial Neural Networks of Feedforward and LSTM type algorithms. The best performing models were obtained using Feedforward Neural Networks, with an accuracy of 95.18% for economic activity and 86.85% for occupations. Compared to manual coding, the implementation of these models to automatically code around 15,000 economic activities and occupations, allowed the reduction time from days to minutes. Furthermore, considering a combined approach (automatic and manual), in which the error rate of the models was less than 5%, the time was reduced to a quarter regarding manual coding and the number of personnel to half.

Keywords: automated coding, boosting, neural networks, text classification

1. INTRODUCCIÓN

1.1 Planteamiento del problema

En las oficinas estadísticas se usan clasificadores para codificar diferentes variables de texto abierto de encuestas, censos, registros administrativos, etc. La codificación es una actividad que permite estandarizar respuestas textuales abiertas y convertirlas en un código numérico. Dicha actividad facilita el procesamiento de los datos y la elaboración de tabulados e indicadores consistentes y comparables en el tiempo y con la información de otros países [1]. De acuerdo con investigaciones relacionadas a la codificación de actividades económicas y ocupaciones, existen tres tipos de codificación, que son: codificación manual, codificación asistida por computador y codificación automática [2].

En Ecuador, el organismo público encargado de generar estadísticas oficiales es el Instituto Nacional de Estadística y Censos (INEC). Existe un 36% de investigaciones sociales y demográficas en el inventario de operaciones estadísticas del Sistema Estadístico Nacional (SEN) [3]. Además, en investigaciones sociodemográficas se recopilan datos de las características laborales de las personas, por ejemplo, actividad económica y ocupación. En el modelo de producción estadística del INEC se define la fase de procesamiento en la cual se realizan las siguientes actividades: a) criticar e integrar la base de datos, b) clasificar y codificar, c) validar e imputar, d) derivar nuevas variables y unidades, e) ajustar los factores de expansión, f) tabular y generar indicadores, g) finalizar los archivos de datos [4].

En la mayor parte de investigaciones estadísticas del INEC, la asignación de códigos es realizada de forma manual, por personas entrenadas para esta actividad, por lo que es susceptible a errores. De acuerdo con estudios relacionados a la codificación manual de actividades económicas y ocupaciones, la tasa de error aceptable es del 5% [5]. Además, la codificación manual es un proceso que requiere de periodos largos de tiempo para su ejecución y supervisión, debido a la variedad de respuestas textuales que se obtienen, por ejemplo, en el Censo de Población y Vivienda del 2010 se contrataron 310 personas durante 5 meses para dicho proceso [6]. Se estima que una persona puede codificar alrededor de 250 observaciones de actividad económica y ocupación por día.

La Encuesta Nacional de Empleo Desempleo y Subempleo (ENEMDU) que realiza el INEC, es una investigación sociodemográfica continua, que requiere la asignación de recursos permanentes para la codificación. Además, en la ENEMDU se recopila alrededor de 15.000 observaciones mensuales de actividad económica y ocupación. También, el Censo de Población y Vivienda, que es la investigación estadística más importante del país, requiere de gran cantidad de personal especializado y entrenado para la codificación de las variables antes mencionadas, debido a la cantidad de datos que se generan. Según la ENEMDU II trimestre 2022, en Ecuador existen 8.184.509 personas con empleo [7].

Adicionalmente, dada la relevancia que ha tenido el aprendizaje de máquina para resolver problemas de clasificación de manera exitosa, algunas oficinas estadísticas e investigadores han realizado estudios para automatizar la codificación de actividades económicas y ocupaciones usando algoritmos de aprendizaje automático. Con estos antecedentes, y una vez realizada la revisión de estudios relacionados, se propone desarrollar modelos de clasificación de actividades económicas y ocupaciones basados en algoritmos de aprendizaje supervisado, específicamente Xgboost y Redes Neuronales Artificiales (Feedforward y LSTM).

En el entrenamiento y evaluación de los modelos se usará datos de la ENEMDU mensual de 12 periodos, desde septiembre 2021 hasta agosto 2022. Además, se realizará un análisis comparativo de los resultados de esta investigación con la literatura científica. La finalidad de este proyecto es automatizar la codificación para reducir los tiempos y/o la cantidad de personal especializado necesario en la asignación de códigos de investigaciones sociodemográficas.

1.2 Objetivo general

- Desarrollar dos modelos de clasificación usando boosting y redes neuronales artificiales para codificar automáticamente las actividades económicas y ocupaciones de investigaciones sociodemográficas del INEC.

1.3 Objetivos específicos

- Realizar una revisión de la literatura científica acerca de los trabajos relacionados con la codificación automática de actividades económicas y ocupaciones usando algoritmos de aprendizaje supervisado.
- Diseñar un modelo de clasificación para codificar automáticamente las actividades económicas de las investigaciones sociodemográficas del INEC usando algoritmos de boosting y redes neuronales artificiales.
- Diseñar un modelo de clasificación para codificar automáticamente las ocupaciones de las investigaciones sociodemográficas del INEC usando algoritmos de boosting y redes neuronales artificiales.
- Evaluar la exactitud, la precisión y la sensibilidad de los modelos de clasificación desarrollados, con datos no utilizados para el diseño de los modelos.
- Analizar y comparar la exactitud de los modelos de clasificación de actividades económicas y ocupaciones desarrollados con la exactitud de los modelos de otras investigaciones similares.

1.4 Alcance

Desarrollar dos modelos de clasificación, uno para actividades económicas y otro para ocupaciones usando boosting y redes neuronales artificiales para automatizar el proceso de codificación de investigaciones sociodemográficas del INEC. Para la construcción y evaluación de los modelos se usará datos de la encuesta ENEMDU mensual. Automatizar la codificación permitirá reducir los tiempos de procesamiento de la información estadística, disminuir personal y reducir los errores de la codificación manual, contribuyendo de esta manera a la visión institucional del INEC que es ser un referente a nivel nacional e internacional por la calidad, oportunidad e innovación en la producción de información estadística[1]. También se evaluarán los resultados de los modelos desarrollados y se realizará una comparación con los de otras investigaciones relacionadas. Finalmente se desarrollará aplicativos web como herramienta para comunicar los resultados del proyecto.

[1] https://www.ecuadorencifras.gob.ec/mision-vision-valores/

1.5 Marco Teórico

1.5.1 Clasificaciones Estadísticas

Las clasificaciones estadísticas son un conjunto de categorías discretas que pueden ser asignadas a variables de investigaciones estadísticas o registros administrativos, y son usadas para la producción y difusión de estadísticas. Pueden tener una estructura plana o jerárquica. Una clasificación plana es un listado de categorías con un solo nivel de agregación. Una clasificación jerárquica tiene varios niveles de agregación, en el nivel más alto las categorías son amplias y agrupan varias categorías de los niveles más bajos, en los niveles más bajos las categorías son más detalladas. Las categorías en los diferentes niveles son mutuamente excluyentes y exhaustivas [1].

Las clasificaciones usadas por el INEC para codificar las actividades económicas y ocupaciones de sus operaciones estadísticas son instrumentos normativos internacionales adaptados a la realidad nacional, que permiten estandarizar la información para su comparabilidad nacional, internacional y a lo largo del tiempo [4]. Además, las clasificaciones utilizadas tienen una estructura jerárquica.

Clasificación Industrial Internacional Uniforme de todas las Actividades Económicas CIIU 4.0

La CIIU 4.0 es una estructura de clasificación que permite organizar los datos y presentar estadísticas de las actividades económicas para su divulgación y análisis, así como para la toma de decisiones y la elaboración de políticas. También es utilizada para la estandarización de los datos de recaudación fiscal y de la emisión de licencias comerciales. La División de Estadística de las Naciones Unidas es la encargada de elaborar y actualizar la clasificación [8]. La CIIU 4.0 tiene 4 niveles de agregación, el primer nivel tiene 21 secciones, el segundo nivel 88 divisiones, el tercer nivel 238 grupos y el cuarto nivel 419 clases [9]. En el Anexo I se indican las secciones del nivel más agregado de la CIIU 4.0.

Clasificación Internacional Uniforme de Ocupaciones CIUO Rev. 08

La CIUO Rev. 08 es un sistema de clasificación y agregación de ocupaciones que se utiliza para recopilar y procesar estadísticas de empleo, para el análisis de mercado laboral, ocupaciones sanitarias y de seguridad, para el análisis de salarios, para la planificación educativa y de recursos humanos, entre otras [10,11]. El custodio encargado de la actualización y difusión de la clasificación es la Organización Internacional del Trabajo

(OIT). La CIUO Rev. 08 tiene 4 niveles de agregación. El primer nivel tiene 10 grandes grupos, el segundo nivel 43 subgrupos principales, el tercer nivel 130 subgrupos y el cuarto nivel 438 grupos primarios [12]. En el Anexo II se indican los grandes grupos del nivel más agregado de la CIUO Rev. 08.

1.5.2 Encuesta Nacional de Empleo, Desempleo y Subempleo ENEMDU

La ENEMDU es una operación estadística continua y especializada, mediante la cual se obtienen datos relacionados a la situación laboral, actividades económicas e ingresos de la población ecuatoriana, tomando en cuenta sus características sociodemográficas. Con los datos de esta operación estadística, se construyen indicadores que permiten crear y establecer políticas públicas de empleo [13]

1.5.3 Codificación de variables estadísticas y sus tipos

En el Modelo de Producción Estadística (MPE) del INEC se establecen ocho fases para la generación de estadísticas oficiales, una de ellas es el procesamiento, en la cual se depuran los datos y se preparan para su análisis y difusión. En el procesamiento de los datos se realiza la actividad de clasificar y codificar que consiste en estandarizar las variables textuales abiertas mediante códigos numéricos previamente establecidos en las clasificaciones y nomenclaturas estipuladas en una Norma Técnica publicada en el Registro Oficial[2][4].

Existen trabajos previos relacionados a la codificación de ocupaciones, los cuales se refieren a tres tipos de codificación: manual, asistida y automática [2]. En algunas oficinas estadísticas de diferentes países, las actividades económicas también se codifican usando estos tres tipos de métodos.

Codificación manual: Es realizada por personas entrenadas para esta actividad. El proceso manual es costoso e involucra el uso de un gran número de personas y tiempos largos, especialmente cuando se tiene grandes cantidades de datos [2]. A pesar de las desventajas, algunas instituciones mantienen la codificación manual debido a que se obtienen tasas de error bajas.

[2] Registro Oficial N.º 230 del 22 de abril del 2014

Codificación asistida: En la codificación asistida por computador un software sugiere un número limitado de códigos para que una persona elija la mejor opción. La codificación asistida reduce costos y tiempo respecto a la codificación manual [2].

Codificación automática: En la codificación automática se asigna los códigos sin intervención humana mediante un programa informático. La codificación automática reduce considerablemente los costos y el tiempo, también evita los errores humanos. La codificación automática de ocupaciones ha sido investigada desde tres enfoques, los cuales son: basada en reglas, usando aprendizaje automático y soluciones híbridas.

Una de las desventajas de la codificación automática basada en reglas es la necesidad de grandes cantidades de datos y la creación de reglas manualmente [2]. El Instituto Nacional de Estadística y Geografía de México INEGI no ha superado el 75% de codificación de actividades económicas y ocupaciones usando el enfoque basado reglas. También logró exactitudes de clasificación del 89,21% en actividades económicas y del 85,05% en ocupaciones usando una combinación de algoritmos de aprendizaje automático [14]. Las soluciones híbridas combinan la codificación basada en reglas con algoritmos de aprendizaje automático [2].

1.5.4 Métricas de evaluación de la codificación automática

Para evaluar el rendimiento de la codificación automática en cualquiera de los tres enfoques mencionados en 1.5.3, se utilizan las siguientes métricas: a) Tasa de producción, que es el porcentaje de casos codificados automáticamente. b) Tasa de aciertos o tasa de acuerdo fijo, que es el porcentaje de casos codificados correctamente respecto a los casos codificados automáticamente. c) Tasa de error, que es el porcentaje de casos codificados incorrectamente respecto a los casos codificados automáticamente [15].

Para garantizar la calidad de la información codificada usando modelos de aprendizaje automático, la tasa de error que se obtiene debe ser menor o por lo menos igual que la tasa de error de la codificación manual [15].

1.5.5 Clasificación de texto

En los últimos años el crecimiento exponencial de los datos de texto en documentos, redes sociales, la web, etc., ha conducido a investigaciones de aprendizaje automático que han dado resultados sorprendentes en la clasificación de texto y en el procesamiento del

lenguaje natural (PLN) [16]. La clasificación de texto es una tarea de aprendizaje supervisado, mediante la cual, datos textuales previamente etiquetados con una categoría o clase son usados para entrenar un modelo de clasificación que predice categorías o clases para nuevos datos de texto [17].

Por otra parte, la clasificación de texto puede ser a nivel de documentos, párrafos, oraciones o partes de oraciones [16]. Dependiendo del número de clases etiquetadas existentes, la clasificación puede ser binaria o multiclase. En la clasificación binaria existen dos clases etiquetadas, mientras que en la clasificación multiclase existen más de dos clases [18]. La mayoría de los sistemas de clasificación de texto, incluyen las cuatro fases que se ilustran en la Figura 1 [16].

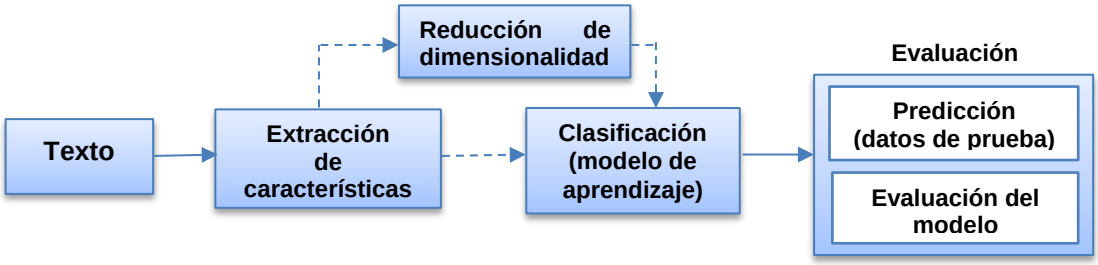


Figura 1 – Fases del proceso de clasificación de texto. Adaptado de [16]

En [19] se concluye de forma experimental que el óptimo rendimiento de un clasificador de texto depende de la adecuada combinación de algunos factores como: la selección del algoritmo de clasificación, los métodos de selección y extracción de características, la reducción de la dimensionalidad, etc.

Extracción de características

Generalmente, los conjuntos de datos de texto contienen palabras innecesarias, errores ortográficos, caracteres especiales, etc., los cuales producen ruido y afectan a los modelos de aprendizaje automático. Para solventar este inconveniente es necesario utilizar métodos de limpieza y preprocesamiento de texto antes de la extracción de características [16].

En la clasificación de texto, comúnmente se usan cuatro pasos para el preprocesamiento, estos son: la *tokenización*, la eliminación de *stopwords*, la conversión a minúsculas y el *stemming* [20]. También existen otros pasos considerados en el preprocesamiento y

limpieza de texto, como: el tratamiento de modismos, coloquialismos, jerga o abreviaturas, la eliminación de signos de puntuación y caracteres especiales, la corrección ortográfica y la lematización [16].

La *tokenización* consiste en la segmentación de textos en palabras, frases u otras partes significativas denominadas tokens. Las *stopwords* son palabras encontradas comúnmente y consideradas irrelevantes dentro de un texto, por ejemplo, conjunciones, preposiciones, artículos, etc. El *stemming* consiste en obtener la raíz (stem) de una palabra derivada, ya que existen palabras derivadas que tienen el mismo significado semántico en su forma raíz [20]. En la *lematización* se sustituye o elimina el sufijo de una palabra para obtener la forma base (lemma) de la palabra [16].

La extracción de características consiste en representar el texto (documento, oración, etc.) en un espacio vectorial numérico. Cada texto se representa como uno punto en el espacio de N dimensiones y cada dimensión representa una característica del texto [21]. Las técnicas frecuentemente usadas para la extracción de características pueden ser las de ponderación de palabras y las *word embeddings* [16].

La forma básica de la ponderación de palabras es *Term Frecuency* (TF), en la cual las palabras son representadas por su número de ocurrencia en el conjunto de datos de texto, denominado corpus. Entre las técnicas de ponderación de palabras se encuentran: *Bag of Words* (BoW) y *Term Frequency-Inverse Document Frequency* (TF-IDF). Las *word embeddings* representan una palabra o frase del conjunto de datos con un vector de dimensión N [16], y capturan la semántica y la sintaxis de las palabras, además ayudan a reducir la dimensionalidad de las características [19]. Existen *word embeddings* que pueden ser entrenadas con un conjunto de datos específico y las *word embeddings* preentrenadas como Word2Vec, GloVe, and FastText que han sido exitosamente usadas en la clasificación de texto [16].

Reducción de la dimensionalidad

La extracción de características de textos produce vectores de alta dimensionalidad debido a que generalmente los textos tienen una gran variedad de palabras, esto podría incluso provocar sobreajuste en un modelo de clasificación [21]. La reducción de dimensionalidad

es un paso opcional en la clasificación de texto [16]. Sin embargo, tener vectores de alta dimensionalidad hace costoso el procesamiento.

Para reducir la alta dimensionalidad existen algoritmos que preservan las características y relaciones de los datos originales, por ejemplo, existen enfoques basados en métodos estadísticos, palabras sinónimas y word embeddings [19]. Algunos métodos habitualmente usados son: *Ganancia de la información, Chi-Square, Principal Component Analysis* (PCA), *Linear Discriminant Analysis* (LDA), *T-distributed Stochastic Neighbor Embedding* (t-SNE), *Latent Semantic Indexing* (LSI), entre otros [16,17].

Estudios experimentales demuestran que en algunos casos la reducción de dimensionalidad mejora el desempeño de los modelos predictivos [19,22] y en otros casos el mejor desempeño se obtiene con los datos originales [21,22].

Selección de clasificadores

En [16] se considera que el paso más importante dentro de la categorización de documentos es seleccionar el mejor algoritmo de clasificación. En esta fase los algoritmos se aplican a los datos preprocesados. Existe una gran variedad de algoritmos de clasificación, entre estos tenemos: clasificadores probabilísticos, clasificadores basados en reglas, en regresión, en proximidad, en arboles de decisión, en redes neuronales, entre otros. Cada algoritmo tiene sus ventajas y desventajas y se usan dependiendo del problema que se quiere resolver y de los datos disponibles [17].

Generalmente los algoritmos de clasificación de texto con los que se obtiene una alta exactitud son complejos de explicar y tienen un alto costo computacional, por ejemplo, boosting, bagging y redes neuronales profundas. Los algoritmos más sencillos de explicar en su mayoría no requieren de un alto costo computacional y su precisión depende del conjunto de datos disponible, por ejemplo, regresión logística, arboles de decisión, naive bayes, entre otros [16,17].

Evaluación

Una vez que el modelo ha sido entrenado y ajustado se realiza las predicciones para evaluar el modelo usando datos de prueba, que son datos nuevos que no han sido usados en el entrenamiento. La evaluación de los modelos consiste en medir la capacidad de

clasificar correctamente [16]. Existen varias métricas usadas para medir el desempeño de los modelos de aprendizaje automático. En [23] se mencionan cuatro métricas útiles para evaluar modelos de clasificación, estas son: la exactitud, la precisión, la sensibilidad y la puntuación F1. Para el cálculo de estas métricas es necesario entender la matriz de confusión y sus elementos: verdadero positivo (VP), verdadero negativo (VN), falso positivo (FP) y falso negativo (FN).

En la matriz de confusión VP y VN representan el número de observaciones positivas y negativas respectivamente que fueron clasificadas correctamente mientras que FN y FP representan el número de observaciones positivas y negativas respectivamente que fueron clasificadas incorrectamente [24]. Las filas y las columnas de la matriz de confusión representan las clases actuales y las clases de las predicciones. En la Figura 2 se muestra los elementos de la matriz de confusión.

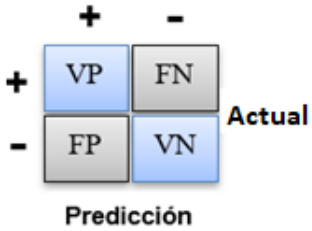


Figura 2 – Matriz de confusión para clasificación binaria. Adaptado de [16].

La *exactitud* mide la proporción predicciones correctas (VP+VN) en relación con el total de observaciones (VP+FP+VN+FN), es decir la proporción de aciertos del clasificador. La *precisión* mide la relación entre las observaciones positivas predichas correctamente (VP) respecto a todas las predicciones positivas (VP+FP), es decir la proporción de observaciones predichas en una determinada clase que en realidad pertenecen a dicha clase. La *sensibilidad* mide la relación de las predicciones positivas clasificadas correctamente (VP) respecto el total de observaciones positivas (VP+FN), es decir que tan bien un clasificador predice determinada clase. La *puntuación F1 o F1-score* representa la media armónica entre la precisión y la sensibilidad [24].

Existe un equilibrio (*trade-off*) entre la precisión y la sensibilidad, a mayor precisión menor sensibilidad y viceversa [2]. En la clasificación multiclase se calculan las métricas para cada clase (una clase vs el resto) y luego se obtiene el promedio (macro - promedio) que asigna

pesos iguales a todas las clases o el promedio ponderado (micro - promedio) que asigna pesos iguales a todas las observaciones [16].

Adicionalmente, para comparar la distribución de los datos reales y de las predicciones se puede utilizar la divergencia de Kullback-Leibler (KLD) que permite medir cuanto difiere una distribución de otra distribución de referencia. KLD indica cuanta información se pierde en bits al elegir una aproximación de la distribución. Mientras el valor de KLD es menor, la distribución aproximada será más similar a la distribución de referencia. Si p(y) es una distribución de probabilidad de referencia y p(x) una distribución de probabilidad aproximada de una variable aleatoria discreta, KLD se define como [25]:

$$KLD[p(y)||p(x)] = \sum_{i}^{N} p(y_i)\log\left(\frac{p(y_i)}{p(x_i)}\right)$$

1.5.6 Term Frequency-Inverse Document Frequency (TF-IDF)

TF-IDF es un algoritmo estadístico muy utilizado en la minería de texto, el cual permite representar un documento como un vector de características con base en la importancia de las palabras que componen el documento. TF representa la importancia de una palabra en un documento. La idea de IDF surge a partir de considerar que una palabra encontrada en el menor número documentos dentro de un corpus (conjunto de documentos) es más importante para discriminar un documento de otro [26]. TF-IDF de una palabra o término *t* en un documento *d* es el producto de TF x IDF y se calcula de la siguiente manera:

$$TF = \frac{tf}{n}$$

$$IDF = \log\left(\frac{N}{df(t)}\right)$$

$$TF\text{-}IDF\ (d, t) = \frac{tf}{n} \times \log\left(\frac{N}{df(t)}\right)$$

Donde,

tf es el número de palabras *t* encontradas en un documento *d*.

n es el número total de palabras en un documento *d.*

N es el número de documentos del corpus.

$df(t)$ es el número de documentos que contienen la palabra *t* en el corpus.

La extracción de características a partir de TF-IDF suele generar vectores de alta dimensionalidad, específicamente el número de palabras únicas en el corpus, el cual podría ser incluso en el orden de los millones [16]. Por esta razón eliminar las palabras con menor ocurrencia en el corpus es una técnica que se aplica para la reducción de dimensionalidad. Una de las desventajas de TF-IDF es que no considera el orden de las palabras ni la semántica del documento [17].

1.5.7 Algoritmos de boosting

Se ha demostrado en estudios experimentales que los algoritmos basados en boosting tienen gran impacto en problemas de clasificación [27]. Los algoritmos de boosting fueron creados para potenciar el rendimiento de algoritmos de aprendizaje débiles (*weak learners*) [16]. El algoritmo inicial de boosting denominado *AdaBoost* fue modificado algunas veces hasta obtener el impulso de gradiente (*gradient boosting*) que se fundamenta en un modelo de optimización numérica cuyo objetivo es minimizar la función de perdida con un proceso similar al del gradiente descendiente. El impulso de gradiente es un modelo aditivo por etapas, en el cual el aprendiz débil es un árbol de decisión. Los residuos del árbol de decisión inicial son considerados para la construcción del siguiente aprendiz débil y así sucesivamente se añaden aprendices débiles hasta cuando se alcanza un valor óptimo para la función de pérdida [28].

Extreme Gradient Boosting (Xgboost) es un algoritmo de impulso de gradiente mejorado, en el cual los aprendices débiles no son añadidos uno después del otro, en su lugar Xgboost usa un enfoque de paralelización para añadir los aprendices débiles y de esta manera aprovechar los núcleos del procesador para obtener una mayor velocidad y rendimiento. Además, Xgboost maneja adecuadamente los datos dispersos y valores perdidos [28]. En la Figura 3 se ilustra el funcionamiento de Xgboost, donde $\propto_i$ son los parámetros de regularización calculados, r_i son los residuos calculados, h_i es una función que es entrenada para predecir residuos, para calcular $\propto_i$ se usa r_i [29].

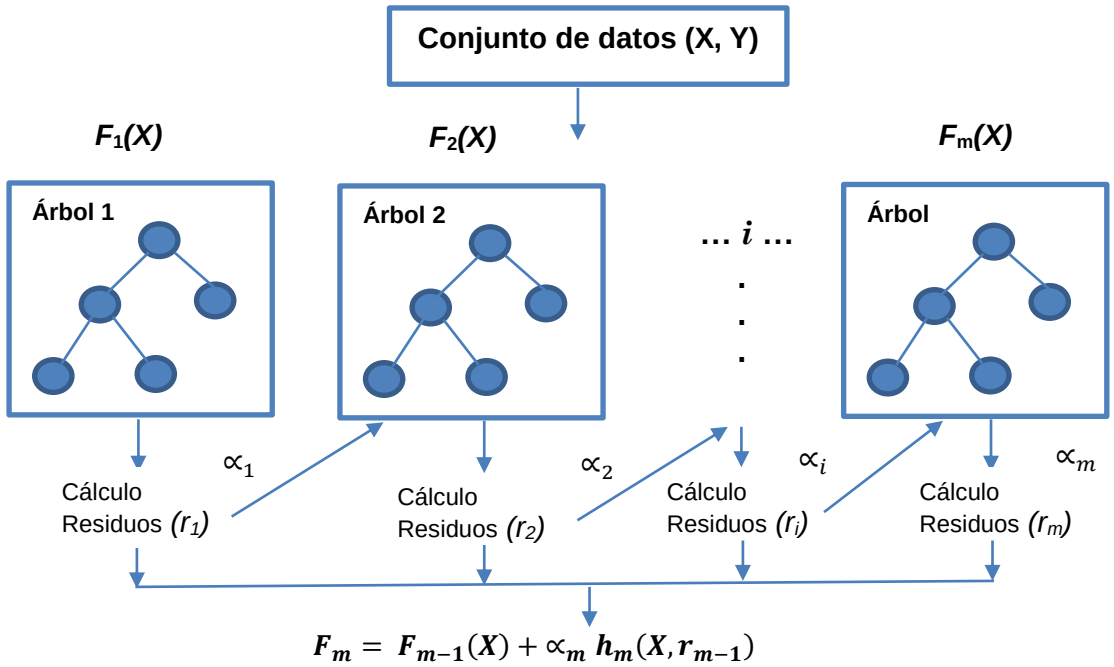


Figura 3 – Funcionamiento de Xgboost. Adaptado de [29]

1.5.8 Redes Neuronales Artificiales

Una Red Neuronal Artificial (ANN) es un modelo matemático compuesto por neuronas artificiales o funciones, las neuronas artificiales están agrupadas en tres tipos de capas que son: entrada, salida y oculta. En la entrada de una neurona artificial los datos de entrada se ponderan con pesos, luego se suma todas las entradas ponderadas más el *bias* y este resultado pasa por una función de activación en la salida de la neurona artificial [30]. En los últimos años las redes neuronales han mostrado resultados impresionantes en la comprensión del lenguaje natural [31]. Por lo cual se utilizó Redes Neuronales Artificiales Feedforward y LSTM para el desarrollo de los modelos de actividad económica y ocupación.

Redes Neuronales Feedforward

La *Red Neuronal Feedforward* (FNN) es uno de uno de los modelos más sencillos del aprendizaje profundo y ha logrado gran precisión en tareas de clasificación de texto [32]. En las FNNs las salidas no retroalimentan a las entradas de las neuronas, es decir la información va en una sola dirección. Generalmente una FNN se entrena con el algoritmo de aprendizaje supervisado de retropropagación, mediante el cual se actualizan los pesos de la red neuronal hacia atrás.

Inicialmente una FNN puede tener pesos aleatorios para calcular el error del modelo, con este valor los pesos de la red se actualizan de desde la capa de salida hacia la capa de entrada pasando por todas las capas que conforman la rede neuronal, el proceso se repite hasta encontrar los pesos óptimos para minimizar el error [33]. En la Figura 4 se ilustra una Red Neuronal Feedforward.

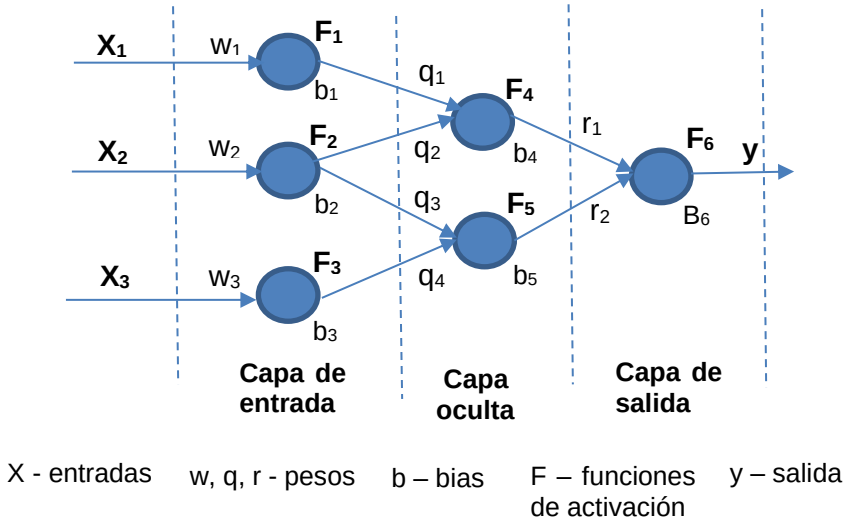


Figura 4 – Red Neuronal Feedforward. Adaptado de [30]

Redes Neuronales Recurrentes

Las Redes Neuronales Recurrentes (RNN) son muy eficaces para la clasificación de texto y datos secuenciales, ya que intentan capturar la dependencia de palabras y las estructuras de texto [32]. En las RNNs existen bucles de retroalimentación entre las salidas y las entradas de las neuronas [33], esta retroalimentación o recurrencia permite a la red recordar la información de entradas anteriores en una secuencia. A menudo las RNNs clásicas tienen menor rendimiento que las FNNs debido al desvanecimiento o explosión del gradiente [32], que se presenta generalmente en secuencias largas.

La explosión del gradiente se refiere al acelerado incremento del gradiente durante el entrenamiento por retropropagación a través del tiempo, mientras que en el desvanecimiento sucede lo opuesto, el gradiente disminuye su valor exponencialmente hasta casi cero, por lo cual el modelo no aprende correlaciones con las entradas temporalmente distantes, ya que los pesos actualizados de las entradas iniciales de una secuencia tienen un valor que tiende a cero [34].

Red Neuronal Long Short-Term Memory

La *Red Neuronal Long Short-Term Memory* (LSTM) es una RNN diseñada para solventar el problema del desvanecimiento o explosión de gradiente, ya que posee una celda de estado o memoria que le permite recordar la información en el tiempo para secuencias de gran tamaño. Además, tiene tres puertas para controlar el flujo de información hacia y desde la celda de memoria [32]. En la Figura 5 se indica una celda LSTM, conformada por la celda de estado, la puerta de olvido, la puerta de entrada y la puerta de salida.

La puerta de olvido determina que información de pasos anteriores se mantiene o se olvida en la celda de estado. Luego de pasar por una función de activación sigmoide, si la salida de la puerta de olvido es 1 la información se mantiene caso contrario si es 0 se ignora [35]. La puerta de entrada decide qué información relevante se mantiene del paso de tiempo actual y la puerta de salida determina el estado oculto para el siguiente paso de tiempo. A continuación, se explica el funcionamiento de una celda LSTM mediante ecuaciones [16], donde W representan los pesos y b los bias.

Puerta de entrada: $i_t = \sigma(W_i[X_t, h_{t-1}] + b_i)$

Valores candidatos: $\tilde{C}_t = tanh(W_c[X_t, h_{t-1}] + b_c)$

Puerta de olvido: $f_t = \sigma(W_f[X_t, h_{t-1}] + b_f)$

Celda de Estado: $C_t = i_t * \tilde{C}_t + f_t * C_{t-1}$

Puerta de Salida: $o_t = \sigma(W_o[X_t, h_{t-1}] + b_o)$

Salida LSTM: $h_t = o_t * tanh(C_t)$

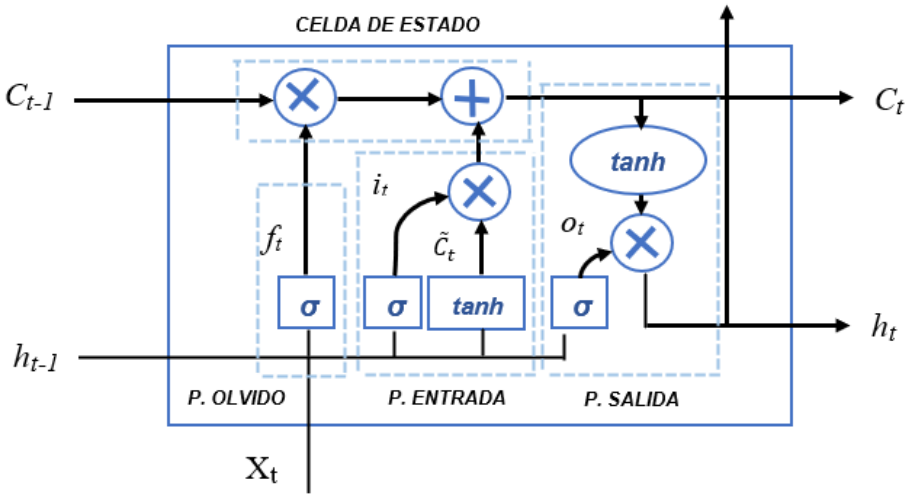


Figura 5 – Celda LSTM. Adaptado de [16]

2. METODOLOGÍA

Inicialmente se planteó el uso de *Design Science Research Methodology* (DSRM) para el desarrollo de este proyecto. La DSRM tiene un enfoque basado en un ciclo continuo de diseño, construcción y evaluación de artefactos de valor prácticos, los cuales permiten contribuir a la eficiencia de los sistemas de información en las organizaciones [36]. La DSRM incluye seis etapas: identificación del problema y motivación, definición de objetivos para la solución, diseño y desarrollo, demostración, evaluación y comunicación [37]. También se consideró la metodología *Cross Industry Standard Process for Data Mining* (CRISP-DM), la cual es ampliamente usada en proyectos de minería de datos. CRISP-DM es un modelo de procesos que es independiente de la industria o de la tecnología utilizada. Además, la metodología tiene como objetivo que los proyectos de minería de datos sean más fiables, menos costosos, replicables y más rápidos. CRISP-DM contempla 6 fases que son: comprensión del problema, comprensión de los datos, preparación de los datos, modelado, evaluación y despliegue [38]. Dado que las etapas de las metodologías mencionadas anteriormente son comparables y que el resultado del proyecto son modelos de datos, finalmente se consideró utilizar CRISP-DM.

2.1 Comprensión del problema

En esta fase se evalúa el problema desde una perspectiva del negocio, por lo tanto, es necesario entender las razones que motivaron el desarrollo de modelos de aprendizaje automático para codificar las actividades económicas y ocupaciones de investigaciones sociodemográficas realizadas en el INEC. Inicialmente es importante señalar que un proceso de codificación automático permite reducir el personal y tiempos requeridos en comparación con la codificación manual. También permite reducir los errores asociados a factores inherentes a la codificación manual, por ejemplo, cansancio de las personas y falta de comprensión de las reglas utilizadas en el proceso. Por consiguiente, la codificación automática aporta a la obtención de información estadística oportuna y de calidad que permite la definición de políticas públicas adecuadas.

Para la codificación de actividades económicas y ocupaciones de operaciones estadísticas se utilizan metodologías de clasificación estándar. Cada oficina estadística establece sus propias reglas para el uso de las Clasificaciones Internacionales CIIU y CIUO, esto sucede debido a que la situación económica y de empleo es diferente en cada país. En 1.5.1 se explica brevemente los clasificadores CIIU y CIUO. El INEC dispone de una metodología

para asignar los códigos de actividad económica CIIU y de ocupación CIUO, en la cual se establecen reglas para la codificación usando la descripción de variables de texto abierto. Además, se usan otras variables denominadas **variables de apoyo** que permiten agrupar dentro de clases homogéneas y excluyentes las actividades económicas y ocupaciones [39]. Se recomienda codificar la actividad económica y ocupación en forma conjunta, debido a que se refieren a las características ocupacionales de una misma persona.

En la ENEMDU se realiza la codificación de actividades económicas y ocupaciones a 4 dígitos. La codificación se realiza de forma manual, por lo que se requiere personal capacitado y algunos días de trabajo. Tomando en cuenta que una persona puede codificar en promedio 250 observaciones diarias y que la ENEMDU mensual tiene aproximadamente 15.000 observaciones, si se asignan 10 personas, la actividad durará 6 días.

Para migrar a un proceso de codificación automática se desarrolló un sistema de codificación basado en reglas, mediante el cual se ha codificado aproximadamente el 40% de información de actividades económicas y ocupaciones de investigaciones relacionadas al Censo de Población y Vivienda. Los datos no codificados se envían a un sistema de codificación asistida para que un usuario capacitado asigne el código. Los esfuerzos por incrementar la tasa de producción (explicada en 1.5.4) de la codificación automática no han superado el 2% debido a la complejidad en la definición de reglas para el universo de actividades económicas y ocupaciones.

Dado el crecimiento exponencial de los datos y los recursos computacionales existentes, la clasificación de texto usando aprendizaje automático ha tenido excelentes resultados en entornos donde se gestionan grandes cantidades de datos [40]. Existen algunas investigaciones relacionadas a la clasificación automática de actividades económicas y ocupaciones usando este enfoque.

2.1.1 Estudios Relacionados

Los inconvenientes y limitaciones de la codificación manual han llevado a los investigadores al desarrollo de métodos de codificación automática [2]. En la Tabla 1 se presentan los resultados de los mejores modelos de clasificación de actividades económicas y ocupaciones desarrollados por diferentes autores.

Tabla 1 – Resultados de investigaciones de codificación automática de actividades y ocupaciones usando aprendizaje automático.

PAIS	ALGORITMOS	ACTIVIDAD ECONÓMICA		OCUPACIÓN	
		T. PRODUCCION	T. ACIERTOS	T. PRODUCCION	T. ACIERTOS
Corea del Sur, 2008 [41]		Censo 2 000 000 registros			
	Modelo de Máxima Entropía **(M1)**	74.40%	99.23%	66.31%	99.53%
	Máxima entropía + Reglas **(M2)**	83.70%	98.75%	72.90%	98.41%
Estados Unidos, 2012 [5]		Encuesta sobre la comunidad estadounidense 1 500 000 registros			
	Regresión Logística **(M3)**	60% 56% 50% 40% 30%	94.61% 95.47% 96.04% 97.88% 98.99%	60% 50% 43% 40% 30%	86.57% 91.65% 94.14% 94.14% 95.51%
Alemania, 2014 [15]		Estudio de panel educativo 307 500			
	Naive Bayes **(M4)**	NA	NA	≈100% 50% 3.1%	73% 90% 95%
	Bayes Multinomial **(M5)**	NA	NA	≈100% 50% 45.8%	68% 94% 95%
Alemania, 2014 [42]		Encuesta ALWA 32 882 registros			
	Boosting **(M6)**	NA	NA	100% 43.25%	63.64% 95%
	Naive Bayes **(M7)**	NA	NA	100% 7.96%	63.23% 95%
	Bayes Multinomial **(M8)**	NA	NA	100% 45.36%	59.06% 95%
Alemania, 2017 [43]		Encuesta Social General Alemana 9137 registros			
	Vecinos Cercanos Modificado **(M9)**	NA	NA	100% 80% ≈55%	65% 81% 95%
Canadá, 2020 [44]		Varias encuestas: ≈ 600 000 registros			
	FastText, Xgboost **(M10)**	100% 46.5%	80.5% ≈95%	100% 34.2%	64.4% ≈ 95%
México,2020 [14]	TF-IDF (6 letras y 2 palabras), SVM, Regresión Logística, Random Forest, Redes Neuronales, Xgboost, KNN, ensamblados con pesos diferentes.	Encuesta de Ingresos y gastos 158 568 registros **(M11)**			
		100% 64.99%	89.21% 95%	100% 64.99%	85.05% 95%
		Censo 1 000 000 registros **(M12)**			
		86.01% 78.56% 64.08%	96% 97.29% 98.45%	82.39% 73.44% 55.14%	96,13% 97,44% 98.71%
Inglaterra, 2021 [45]		Censo 500 000 registros			
	TF-IF, regresión logística **(M13)**	100% 76%	81% 92%	100% 73%	84% 95%

La tasa de producción y la tasa de acierto son métricas que se han utilizado en diferentes investigaciones para evaluar los resultados de la codificación automática de actividades económicas y ocupaciones (ver 1.5.4). Algunos autores establecen una tasa de error aceptable del 5%, de acuerdo con lo permitido en el control de calidad de la *American Community Survey* para la codificación manual [5]. En definitiva, no se pretende codificar automáticamente el 100% de casos disminuyendo la calidad del proceso, lo que se intenta es establecer un equilibrio entre la tasa de producción y la tasa de acierto.

2.2 Comprensión de los datos

En esta fase es primordial explorar y verificar la calidad de los datos para definir las técnicas que se usaran en la preparación de los mismos antes del modelado. Para el desarrollo de los modelos de clasificación se utilizó los datos de la encuesta mensual de empleo ENEMDU, de los periodos septiembre 2020 hasta agosto 2021. Se escogió los datos de los últimos periodos disponibles, ya que a lo largo del tiempo las directrices de asignación de códigos han sido modificadas. Por tal razón, existen actividades económicas y ocupaciones con diferentes códigos dependiendo del periodo de la encuesta. Además, en estos datos existen casos de actividades económicas y ocupaciones que surgieron en la pandemia de COVID-19.

La asignación de códigos de actividad económica y ocupación se realiza con base en reglas establecidas en un documento metodológico denominado Manual de Codificación. De esta manera se evalúan variables textuales, numéricas y categóricas relacionadas a las condiciones laborales, sociales y demográficas de las personas, para clasificar en los estándares CIIU y CIUO las actividades económicas y ocupaciones. Por esta razón, se seleccionaron las variables predictoras para los modelos, en función a las reglas de codificación. Las variables predictoras y las variables objetivo se indican en la Tabla 2. Algunas variables se utilizan como predictoras en los dos modelos. Las variables objetivo son: el código CIIU a 4 dígitos (p40) y el código CIUO a 4 dígitos (p41).

Tabla 2 – Variables predictoras y objetivo de los modelos de clasificación de actividades económicas y ocupaciones.

Variable	Descripción	Tipo	Actividad Económica / Ocupación
provincia	Provincia de residencia	Categórica	Actividad Económica
drama	Descripción de la actividad económica	Texto	Actividad Económica y Ocupación
p42	Categoría Ocupacional	Categórica	
p46	Sitio de trabajo	Categórica	
vi07a	Cuartos exclusivos para negocio	Numérica	
p40	Código CIIU de Actividad Económica	Categórica	Actividad Económica
dgrupo	Descripción de la actividad ocupación	Texto	Ocupación
p03	Edad	Numérica	
p05a	Seguro	Categórica	
p10a	Nivel de instrucción	Categórica	
p10b	Años aprobados	Numérica	
p47a	Tamaño del establecimiento	Categórica	
p47b	Número de trabajadores	Numérica	
p63	Ingreso por venta	Numérica	
p64b	Ingreso por autoconsumo	Numérica	
p41	Código CIUO de Ocupación	Categórica	

Los datos fueron proporcionados en 12 archivos CSV, uno por mes, los cuales se consolidaron posteriormente en un solo archivo. Inicialmente se tenía 38 variables y 212.145 observaciones sin duplicados que incluían características sociodemográficas, laborales y económicas de las personas. Los pasos que se detallan a continuación se realizaron para construir dos conjuntos de datos, uno para el modelo de actividades económicas y otro para el modelo de ocupaciones. Inicialmente se seleccionó las variables predictoras y la variable objetivo, además se eliminó algunos casos con valores perdidos. Las variables con un alto porcentaje de valores perdidos se imputaron de acuerdo con directrices de la encuesta. También se eliminó 83 clases de la variable objetivo, código CIUO, del conjunto de datos de ocupaciones que tenían menos de 20 registros.

2.2.1 Análisis descriptivo

Se realizó representaciones gráficas de cada variable para entender su comportamiento individual. En el Anexo III se puede observar nubes de palabras de las variables textuales, en las cuales se evidenció que las principales ocupaciones están relacionadas con las principales actividades económicas que corresponden a los sectores económicos de la agricultura y cría de animales, comercio y construcción.

Respecto a las variables categóricas, se pudo observar que, a pesar del tratamiento inicial de los datos, el comportamiento en la distribución de las categorías se mantuvo con respeto a periodos anteriores de la encuesta. Además, se pudo identificar variables que tenían categorías con poca representación, dichas categorías podrían ser un problema en la validación de los modelos. Se puede observar las representaciones gráficas de las variables categóricas en el Anexo IV. De las variables numéricas se graficó únicamente la edad, obteniendo una distribución que corresponde al rango de edad de la población económicamente activa, el histograma de la variable se puede ver en el Anexo V. No se graficó el resto de las variables numéricas debido a que en la siguiente etapa serán usadas para el cálculo de otras variables.

En las variables a predecir existen 321 clases para el modelo de actividades económicas y 344 clases para el modelo de ocupaciones**.** Se identificó que las clases de los dos conjuntos de datos estaban desbalanceadas, es decir existen clases con gran número de datos y clases con pocos datos, como se puede observar en el Anexo VI. En la clasificación multiclase los datos desbalanceados pueden afectar el desempeño del modelo, produciendo un bajo rendimiento para las clases minoritarias, esto supone un gran inconveniente cuando la correcta predicción de las clases minoritarias es el objetivo [46]. Existen algunos métodos para el balanceo de datos, sin embargo, se debe analizar la afectación de las clases desbalanceadas en un problema particular.

2.3 Preparación de los datos

En esta fase se construyó características derivadas de las variables existentes, también se realizó la división y preprocesamiento de los datos. La preparación de los datos se realizó en un notebook Jupyter con Python, se usó las librerías numpy y scipy para manejar las estructuras de datos y para realizar operaciones numéricas y vectoriales, pandas para manipulación y análisis de datos, matplotlib y seaborn para generar gráficos, scikit-learn para extraer características y dividir el conjunto de datos, nltk para el preprocesamiento de las variables textuales e imblearn para balancear las clases.

Se construyó nuevas variables predictoras combinando variables categóricas y numéricas en base a reglas de codificación. De modo que, se combinó el nivel de instrucción y los años aprobados para generar la variable categórica **nivel de estudios** (nivel), el tamaño del establecimiento y el número de trabajadores para obtener la variable categórica **tipo**

de empresa (tipo_emp), el ingreso por venta menos el ingreso por autoconsumo para generar la variable categórica **autoconsumo** (autoc), la variable cuartos exclusivos para negocio se convirtió en la variable categórica **tenencia de cuarto exclusivo** (cneg). La representación gráfica de las variables construidas se puede observar en el Anexo VII

La evaluación y ajuste de hiperparámetros de los modelos se realizó con el método de validación *holdout* que se indica en la Figura 6. Por tal razón, se dividió el conjunto de datos de la siguiente manera: entrenamiento 80%, validación 10% y prueba 10%, ya que se tiene un conjunto relativamente grande de datos [47]. Los datos de validación se usaron para evaluar los modelos mientras se ajustan y optimizan los hiperparámetros y los datos de prueba se usaron para evaluar los modelos de mejor desempeño [48]. Debido al desbalance de las clases se estratificó la división de los datos, de esta manera en los tres subconjuntos se mantuvo la proporción de las clases del conjunto de datos original [47].

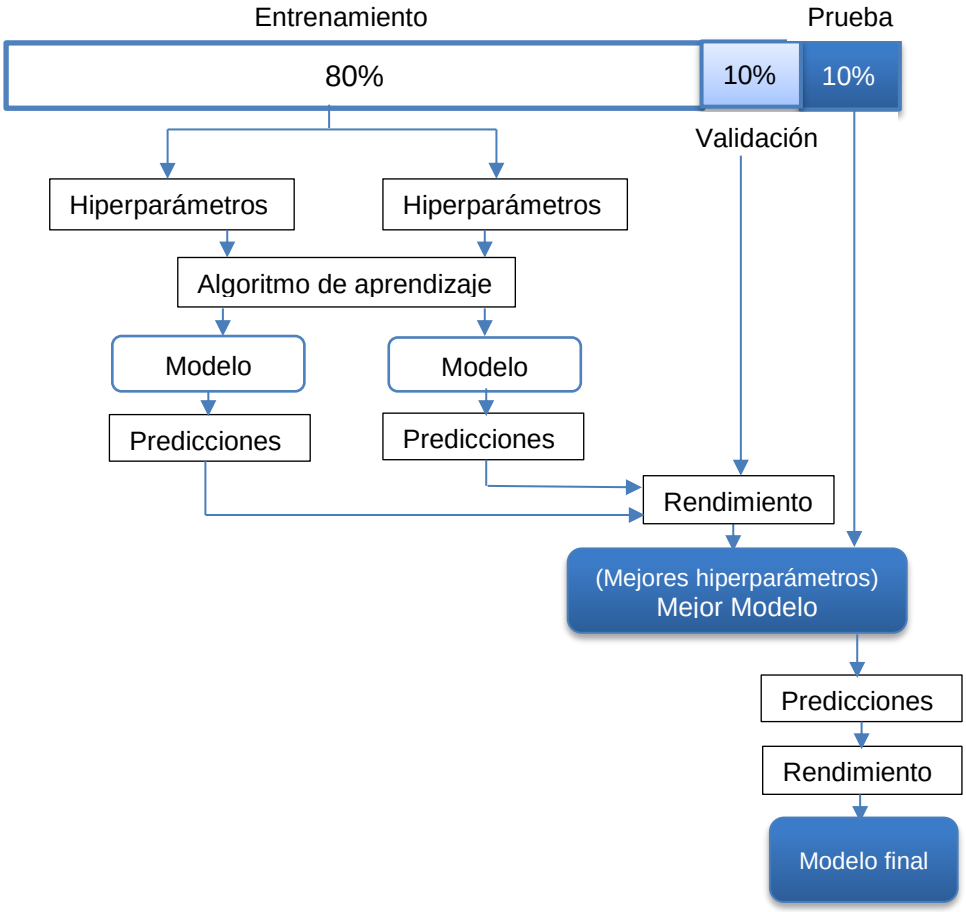


Figura 6 – Método Holdout para ajuste de hiperparámetros y evaluación de modelos. Adaptado de [47]

2.3.1 Preparación de variables textuales

Las variables más importantes para la asignación de un código CIIU o CIUO son las descripciones de actividad económica y ocupación respectivamente, estas variables de texto requieren ser preprocesadas antes de la extracción de características para eliminar información no relevante. El preprocesamiento de texto se realizó en los siguientes pasos: a) convertir todos los caracteres alfabéticos a minúsculas, b) eliminar los signos de puntuación, c) eliminar los números, d) eliminar los espacios en blanco dobles y los del inicio y final del texto, e) eliminar tildes, f) eliminar palabras con 2 o menos caracteres, g) eliminar *stopwords*, h) extraer la raíz de las palabras (*stemming*). (Ver 1.5.5)

Se definió el idioma español para las *stopwords* y para el *stemmer*. En la definición de las *stopwords* se eliminó las palabras **con, sin y para** debido a que estas palabras son necesarias para identificar ciertas actividades económicas, como: Servicios de salud **con** hospitalización, servicios de salud **sin** hospitalización, elaboración de productos **para** la venta, entre otros.

Extracción de características

Para convertir el texto en un vector, en el modelo de boosting se utilizó el método TF-IDF (ver 1.5.6). Mediante *TfidfVectorizer ()* de scikit-learn se generó vectores de características con las descripciones de actividades económicas y ocupaciones, el número de características es igual al número de palabras únicas encontradas en cada variable.

En la extracción de características para el modelo de redes neuronales artificiales, se convirtió el texto en secuencias de números enteros usando la clase *Tokenizer* de la librería Keras, cada palabra fue representada por un número entero. Dado que cada una de las observaciones tiene un número de palabras variable, la dimensión de los vectores numéricos generados también es variable. Por tal razón, se completó con 0s todas las observaciones para tener vectores de 25 elementos para actividad económica y 30 elementos para ocupación.

2.3.2 Preparación de variables categóricas y numéricas

Se utilizó *OneHotEncoder()* y *StandardScaler()* de sckit-learn para el preprocesamiento de las variables categóricas y numéricas. Las variables predictoras provincia, categoría ocupacional, sitio de trabajo, seguro, nivel de estudios, tipo de empresa, autoconsumo y tenencia de cuarto exclusivo tienen categorías representadas por números, dichas

categorías no representan un orden sino una etiqueta de clase. En el entrenamiento de los modelos los valores de las categorías se pueden interpretar como un número entero, lo cual es inadecuado. Por esta razón, se utilizó el método *One-Hot Encoding* que crea una columna por cada categoría de una variable y que se representa con 1 cuando la columna corresponde a la categoría de la observación y con 0 para el resto de las categorías. *One-Hot Encoding* permite que cada categoría se represente como un concepto independiente [49]. Algunos algoritmos de aprendizaje automático funcionan mejor cuando las variables numéricas son estandarizadas, la estandarización consiste en modificar la distribución de una variable de tal manera que su media sea 0 y su deviación estándar 1 [50]. Se estandarizó la variable edad para el modelo de ocupaciones.

Se utilizó *LabelEncoder()* de sckit-learn para convertir los códigos CIIU y CIUO en entradas adecuadas para el algoritmo de boosting. *Label Encoding* codifica las etiquetas de una variable categórica con valores entre 0 y el número de clases menos 1 [51]. En la variable objetivo de las redes neuronales artificiales cada clase debe estar representada por una columna, por lo cual luego de *label encoding* se utilizó la codificación *One-Hot*. Además, se usaron los métodos *fit()* y *transform()* de scikit-learn para ajustar y transformar los datos. Los objetos del preprocesamiento fueron ajustados con los datos de entrenamiento y posteriormente se realizó las transformaciones en los datos de entrenamiento y validación [52]. Estos objetos ajustados se guardaron usando *joblib.dump ()*

2.3.3 Balanceo de clases

Antes de balancear las clases de las variables objetivo es necesario concatenar las variables predictoras. Para el modelo de actividades económicas se concatenó la representación vectorial de la variable drama y las variables categóricas provincia, p42, p46 y cneg representadas con *One-Hot Encoding*. Para el modelo de ocupaciones se concatenó la representación vectorial de las variables dgrupo y drama, las variables categóricas p42, p46, p05a, nivel, tipo_emp, autoc y cneg representadas con *One-Hot Encoding* y la variable numérica estandarizada p03. En la concatenación se obtuvo una matriz densa la cual se convirtió en una matriz dispersa usando *csr_matrix ()* de scipy, que permite productos matriciales más rápidos.

Se realizó submuestreo y sobremuestreo en el conjunto de datos de entrenamiento de los modelos de actividades económicas y ocupaciones para equilibrar las observaciones de las clases desbalanceadas. *SMOTE* es una técnica de sobremuestreo que permite

incrementar las observaciones de las clases minoritarias, genera muestras sintéticas usando los vecinos cercanos de las observaciones de las clases minoritarias. *Near-Miss* es una técnica de submuestreo en la cual se evalúa la distancia promedio entre las observaciones de las clases mayoritarias y minoritarias para eliminar observaciones de la clase mayoritaria [53]. Se utilizó *SMOTE()* y *NearMiss()* de imblearn para balancear las clases de las variables objetivo.

En los datos de entrenamiento del modelo de actividad económica se realizó submuestreo de 5 clases y sobremuestreo de 316 clases, mientras que en los datos de entrenamiento del modelo de ocupación se realizó submuestreo de 7 clases y sobremuestreo de 337 clases. Para los dos conjuntos de datos de entrenamiento se obtuvo 5.000 registros por clase.

2.4 Modelado

Para desarrollar los modelos de clasificación de actividades económicas y ocupaciones se utilizó los algoritmos Xgboost y Redes Neuronales Artificiales (Feedforward y LSTM). Los algoritmos fueron implementados en Python usando las librerías *Xgboost* con su interfaz de scikit-learn para Xgboost y *Keras* para las Redes Neuronales Artificiales. Antes del balanceo de clases, el conjunto de datos para el modelo de actividades económicas estaba formado por 201.190 registros y el conjunto de datos para el modelo de ocupaciones por 209.243 registros. En la Tabla 3 se indica el número de características por tipo de variable que se obtuvo posterior a la preparación de los datos.

Tabla 3 – Número de características por tipo de variable de los modelos de actividades económicas y ocupaciones.

Algoritmo	Modelo	Número de características variables de texto	Número de características variables categóricas	Número de características variables numéricas	Número total de características
Xgboost	Actividad Económica	8.537	48		8.585
Xgboost	Ocupación	16.113	45	1	16.159
Redes Neuronales Artificiales	Actividad Económica	25	48		73
Redes Neuronales Artificiales	Ocupación	55	45	1	101

Una actividad importante en el diseño de modelos de aprendizaje automático es el ajuste u optimización de hiperparámetros, ya que la definición de hiperparámetros impacta en la exactitud y tiempo de entrenamiento de los modelos. Los hiperparámetros son parámetros del algoritmo que no se actualizan durante el entrenamiento y se ajustan previamente para optimizar el rendimiento del modelo.

Se realizó el ajuste manual de hiperparámetros ya que técnicas populares de optimización de hiperparámetros como *grid search* o *random search* requieren suficiente experiencia para definir un rango de valores óptimo para conformar el espacio de búsqueda. Además, dependiendo del número de hiperparámetros seleccionados el costo computacional de *grid search* suele ser alto [31]. A pesar de que puede resultar tedioso, el ajuste manual permite tener más control sobre el proceso y evaluar el impacto de cada hiperparámetro en el rendimiento del modelo [54].

2.4.1 Modelado usando el algoritmo Xgboost

Para el modelado de los datos de actividad económica y ocupación se utilizó el algoritmo Xgboost (ver ítem 1.5.7). Además, dado que se tiene un problema de clasificación multiclase se utilizó *XGBClassifier* del API scikit-learn. Xgboost es un algoritmo que posee varios hiperparámetros que deben ser configurados antes del entrenamiento. El hiperparámetro general *booster* fue configurado en la opción por defecto *gbtree* que usa como aprendices modelos basados en árboles de decisión. En los hiperparámetros de aprendizaje se configuró la función objetivo *objective* con la opción *multi:softprob,* que corresponde a la función softmax y genera como resultado una matriz de probabilidades de cada clase para cada una de las observaciones [55]. Entre los hiperparámetros que más impacto producen en el rendimiento del algoritmo se encuentran *learning_rate*, *max_depth* y *n_estimators,* estos hiperparámetros fueron ajustados para los modelos de actividad económica y ocupaciones antes del entrenamiento [56].

Learning_rate o shrinkage es la tasa de aprendizaje con la cual se controla que tan rápido aprende el modelo, tasas de aprendizaje pequeñas reducen la influencia de cada árbol permitiendo añadir nuevos árboles para mejorar el modelo y de esta manera evitar el sobreajuste [27]. *Max_depth* determina la máxima profundidad o el tamaño del árbol de decisión, incrementar este valor hace modelos más complejos, lo que puede provocar sobreajuste [55]. *N_estimators* define el número máximo de iteraciones del modelo [57], además, el número de árboles que generan los modelos Xgboost es igual al número de

clases del modelo multiplicado por *n_estimators*. En el Anexo VIII y IX se puede observar árboles de decisión entrenados con Xgboost para los modelos de mejor rendimiento de actividades económicas y ocupaciones que se seleccionaron en función de la puntuación F1 macro.

Inicialmente para el entrenamiento de los modelos se utilizó los hiperparámetros por defecto *learning_rate* = 0.3, *max_depth* = 6 y *n_estimators* =100. También se estableció una semilla aleatoria con *random_state=42* para que los resultados de los modelos sean reproducibles. Para el modelo de ocupaciones se configuró los parámetros de regularización *reg_alpha o L1,* mediante el cual se mejora la velocidad de procesamiento y *reg_lambda* o L2, que permite controlar el sobreajuste [58]. Luego*, s*e entrenó los modelos usando datos desbalanceados y balanceados, también diferentes combinaciones de los hiperparámetros *learning_rate, max_depth y n_estimators*. Se ajustó uno por uno estos hiperparámetros para conocer la influencia de cada uno en los modelos. Debido al costo computacional, se entrenó el algoritmo con datos balanceados únicamente usando los hiperparámetros de los modelos con datos desbalanceados de mejor rendimiento.

Usando el conjunto de datos de validación se evalúo con que combinación de hiperparámetros los modelos alcanzaron el mejor rendimiento. Se usó el conjunto de datos de validación y la métrica *mlogloss* que calcula el valor de la función de perdida Log Loss de una clasificación multiclase [55], para medir el rendimiento del modelo en cada iteración. El mejor modelo de actividad económica obtuvo una puntuación F1 de 81.59% y una exactitud de 91.92%, mientras el mejor modelo de ocupación tuvo una puntuación F1 de 56.13% y una exactitud de 81.01%. En los Anexos X y XI se indica los resultados del entrenamiento de los modelos de actividad económica y ocupación con diferentes hiperparámetros.

2.4.2 Modelado usando Redes Neuronales Artificiales

Para la implementación de las Redes Neuronales Artificiales (Feedforward y LSTM) mencionadas en el ítem 1.5.8, se utilizó la API Keras que permite crear redes neuronales construyendo una pila de módulos desde la capa de entra da hasta la capa de salida de la red neuronal [57]. Específicamente se usó la API funcional de Keras que permite construir una topología no lineal con múltiples entradas [59]. En el modelado de los datos de actividades económicas se incluyó dos entradas, la representación vectorial de la variable textual **drama** y el conjunto de variables categóricas. Mientras que para el modelo de

ocupaciones se incluyó tres entradas, las representaciones vectoriales de las variables textuales **dgrupo** y **drama** y el conjunto de **variables no textuales***.* En cada entrada de texto se añadió una capa de incrustación de palabras (*embeddings*) de Keras (ver 1.5.5), la cual se entrena al mismo tiempo que la red neuronal. Con la capa de incrustación cada palabra es representada por un vector de un tamaño específico, comúnmente de 50 o 100 dimensiones [60]. En la capa de incrustación se definieron los hiperparámetros *input_dim* que representa el número de palabras únicas existentes en el conjunto de datos de entrenamiento*, output_dim* que representa la longitud del vector de incrustación de palabras e *input_length* que representa la longitud máxima de la secuencia de palabras [61].

En las Redes Neuronales Feedforward se añadió a la capa de incrustación una capa de aplanamiento *Flatten* mediante la cual se aplana o reduce las dimensiones de una entrada multidimensional. Adicionalmente se añadió una capa de *Dropout* por medio de la cual en el entrenamiento se desactiva aleatoriamente un porcentaje de neuronas de una capa oculta con el objetivo de reducir el sobreajuste [62]. Se concatenó a la red neuronal la entrada de las variables no textuales y se añadió una capa *Densa,* que es una capa oculta de una red neuronal en la cual se define el número de neuronas y la función de activación que se encarga de transformar las entradas de una neurona de acuerdo con una función específica para generar la salida. En la capa densa se usó la función de activación Relu, en la cual $f(x) = x$, si $x > 0$ y $f(x) = 0$, si $x < 0$ [63].

En las redes neuronales LSTM luego de la capa incrustación se incluyó una capa LSTM en la cual se estableció el hiperparámetros *units,* que representa la dimensión de las salidas LSTM del estado oculto y la celda de estado [64]. También se añadió una capa de *Dropout* y luego se concatenó a la red neuronal la entrada de las variables no textuales. Además, se utilizaron las funciones de activación por defecto de Keras, *activation*="tanh" y *recurrent_activation*="sigmoid" [65]. Además, en la salida de los dos tipos de redes neuronales se utilizó una capa Densa con la función de activación softmax y el número de neuronas igual al número de clases del modelo respectivo. En las Figuras 7, 8, 9 y 10 se indican las arquitecturas de las Redes Neuronales Feedforward y LSTM de los modelos de mejor rendimiento de actividades económicas y ocupaciones.

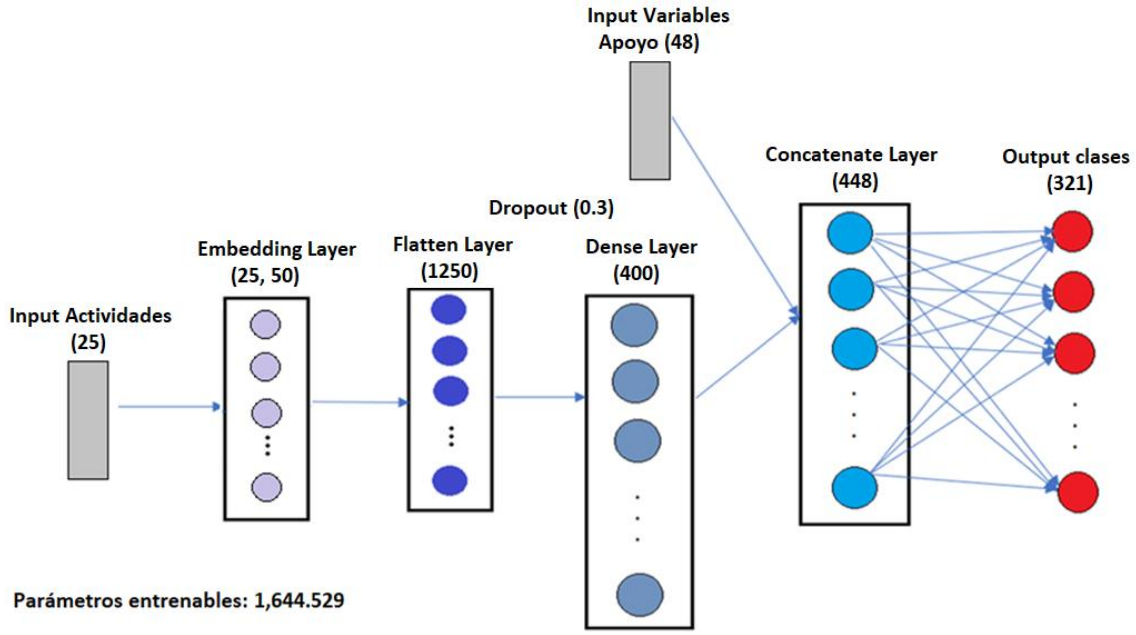


Figura 7 – Arquitectura del modelo de clasificación de actividades económicas usando Redes Neuronales Feedforward.

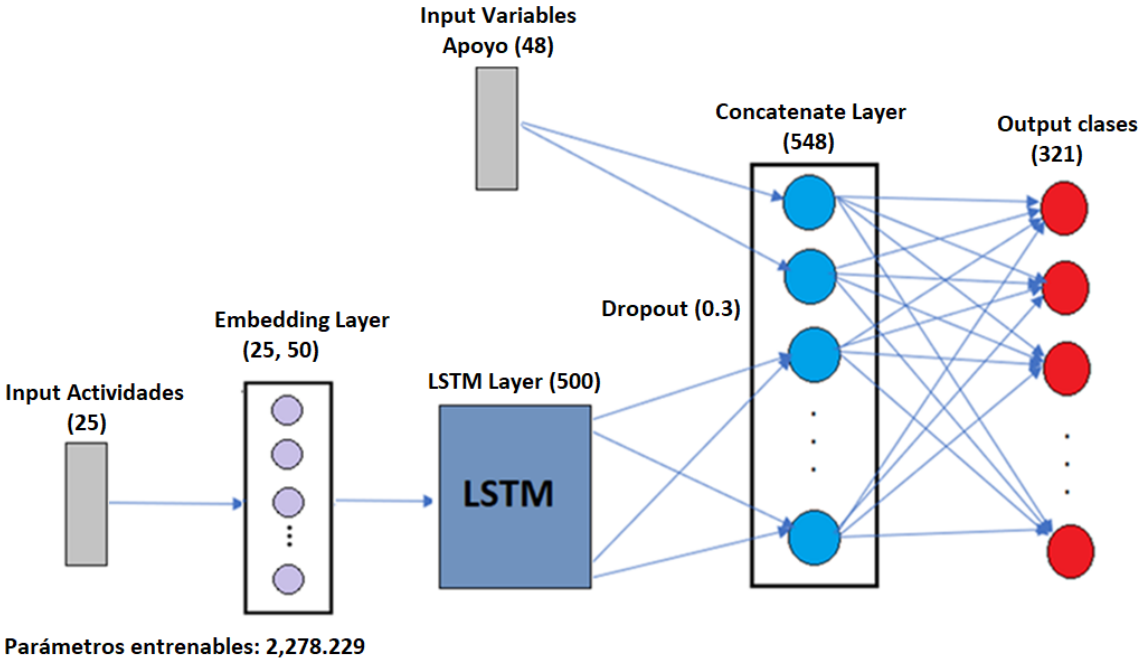


Figura 8 – Arquitectura del modelo de clasificación de actividades económicas usando Redes Neuronales LSTM.

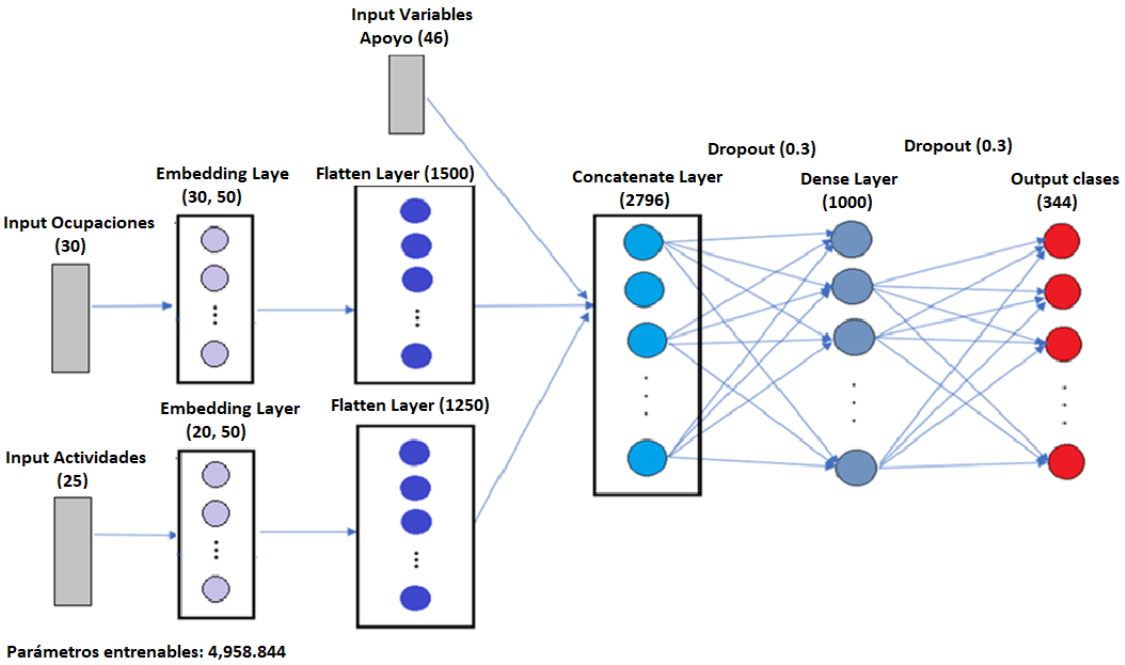


Figura 9 – Arquitectura del modelo de clasificación de ocupaciones usando Redes Neuronales Feedforward.

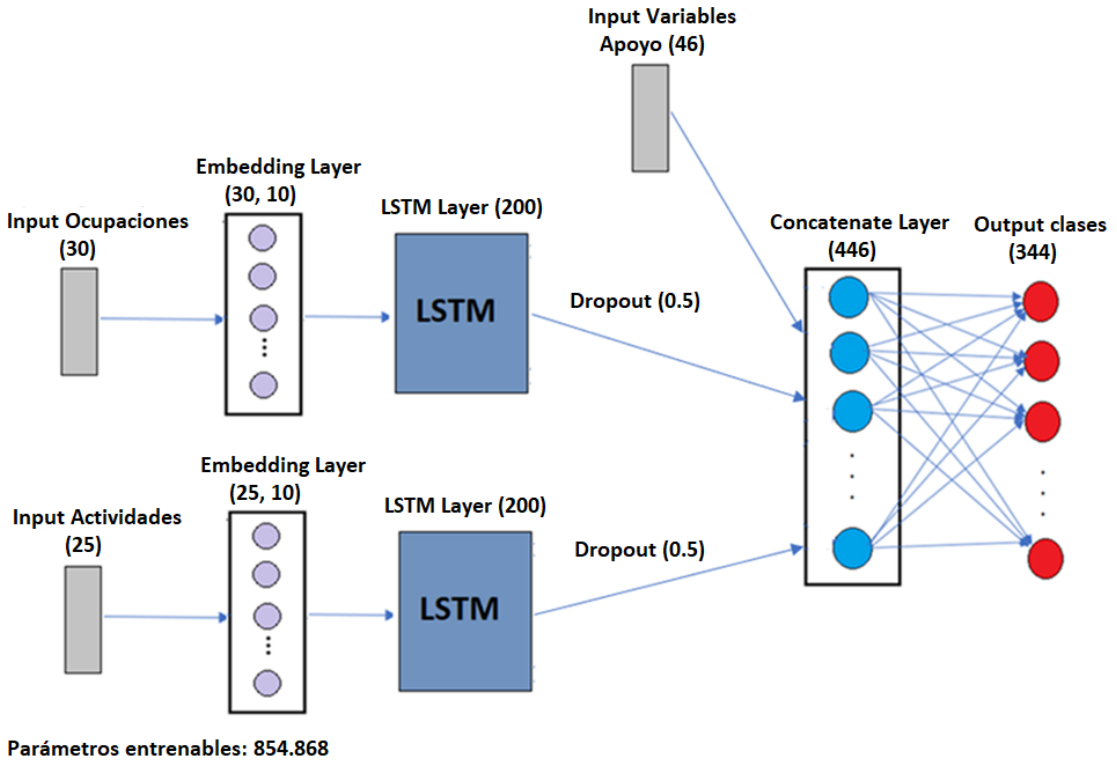


Figura 10 – Arquitectura del modelo de clasificación de ocupaciones usando Redes Neuronales LSTM.

Se ajustó los hiperparámetros manualmente utilizando el conjunto de datos de validación para encontrar la combinación de hiperparámetros con la cual los modelos alcanzaron el mejor desempeño. Se ajustó el *batch size* que representa el número de observaciones que constituyen un lote y se procesan en el entrenamiento antes que se actualicen los pesos del modelo, las *epochs* que es el número de veces que el conjunto de datos completo es procesado en el entrenamiento [66], el *número de neuronas* de las capas Densas y las *unidades* LSTM.

Además, se utilizó la función de optimización *Adam* que es una extensión del gradiente descendiente estocástico y es muy utilizada en la actualidad en redes neuronales ya que es computacionalmente eficiente [67], y se ajustó el hiperparámetro *learning rate* del optimizador, que por defecto es 0.001. También se configuró la función de pérdida *categorical crossentropy* que se utiliza para clasificación multiclase. En el entrenamiento se utilizó la técnica *earlystopping* que permite monitorizar en cada época el error de validación, para identificar cuando comienza a incrementar y de esta manera evitar el sobreajuste deteniendo el entrenamiento. Se entrenó los modelos utilizando los conjuntos de datos desbalanceados y balanceados.

Al igual que en Xgboost, debido al costo computacional se entrenó las redes neuronales con datos balanceados únicamente usando los hiperparámetros de los modelos con datos desbalanceados de mejor rendimiento. El mejor modelo de clasificación de actividades económicas desarrollado con Redes Neuronales Feedforward obtuvo una puntuación F1 de 86.60% y una exactitud de 94.90%, mientras el mejor modelo de ocupación tuvo una puntuación F1 de 61.81% y una exactitud de 87.44%. Respecto a las Redes Neuronales LSTM, el mejor modelo de clasificación de actividades económicas obtuvo una puntuación F1 de 79.40% y una exactitud de 93.20%, mientras el mejor modelo de ocupación tuvo una puntuación F1 de 50.66% y una exactitud de 83.16%. En los Anexos XII, XIII, XIV y XV se indican los resultados del entrenamiento de los modelos con diferentes hiperparámetros.

2.5 Evaluación

Las métricas de evaluación mencionadas en el ítem 1.5.5 se usaron en dos etapas, en el entrenamiento para optimizar el algoritmo y seleccionar la combinación de hiperparámetros que produzcan los mejores resultados y en la etapa de pruebas para evaluar los resultados de los modelos con datos no utilizados en el entrenamiento [24]. Cuando se evalúan

modelos con clases desbalanceadas, en las cuales existen clases con una gran cantidad de datos y clases con pocos datos, la medida de exactitud no es útil para seleccionar un modelo ya que se tiende a realizar predicciones correctas en mayor proporción para las clases mayoritarias. En este caso, la precisión, la sensibilidad y la puntuación F1 son las métricas que se deben considerar [23]. Dado que los conjuntos de datos disponibles tienen clases desbalanceadas se eligió la puntuación F1 macro para seleccionar los modelos de mejor rendimiento, debido a que proporciona una métrica compuesta de la precisión y la sensibilidad, y da la misma importancia a todas las clases. La exactitud se utilizó para realizar un análisis comparativo con los resultados de otros estudios relacionados y para evaluar los modelos con datos de la ENEMDU mensual de los periodos septiembre 2021 a mayo 2022.

Adicionalmente, como se mencionó en 1.5.1 la clasificación de actividades económicas CIIU 4.0 tiene 21 secciones y la clasificación de ocupaciones 10 grandes grupos en el primer nivel de agregación. Generalmente, estas secciones y grandes grupos se usan para presentar resultados agregados de las operaciones estadísticas. Por tal motivo se realizó la comparación a nivel de secciones y grandes grupos de las distribuciones de frecuencias de los conjuntos de datos de prueba con las distribuciones de frecuencias de las predicciones obtenidas con los modelos de mejor desempeño. Para cuantificar la diferencia de las dos distribuciones de probabilidad discretas se utilizó la entropía relativa o divergencia de *Kullback-Leibler* referida en 1.5.5.

2.6 Despliegue

Posterior a la evaluación y selección de los modelos con mejor rendimiento, se creó prototipos de aplicativos web que fueron probados en un ambiente de desarrollo con la finalidad de presentar los resultados. Se utilizó el *framework* de desarrollo web Flask para la creación de las aplicaciones basadas en Python, mediante las cuales se generan las predicciones de actividades económicas y ocupaciones, mientras que para la interfaz gráfica se usó HTML, que permite el ingreso de los datos a través de un formulario. En los Anexos XVI y XVII se pueden observar las interfaces gráficas en donde se ingresan los datos de las variables predictoras y se realiza la solicitud para obtener la predicción del código CIIU y CIUO respectivamente. Como respuesta se visualiza el código, la descripción del código de acuerdo con los clasificadores estándares y la probabilidad de predicción.

Flask es un *microframework* diseñado para crear una aplicación web de manera rápida y es adecuado para proyectos a pequeña escala [68]. Para crear el aplicativo web inicialmente se importó Flask a la aplicación desarrollada en Python y se creó un objeto tipo Flask, también se estableció la ruta para ejecutar la función *predict()* junto con el método POST y luego se definió la función correspondiente que va a dar una respuesta JSON con las predicciones obtenidas. Flask provee un comando *run* para ejecutar la aplicación con un servidor de desarrollo que proporciona un depurador interactivo cuando el código es actualizado.

En el despliegue para el ambiente de producción es necesario utilizar una plataforma de hosting o el servidor *Web Server Gateway Interface* (WSGI) que provee una interfaz entre el servidor web y la aplicación web basada en Python. El ciclo para realizar una petición y recibir una respuesta se puede observar en la Figura 11, inicialmente una solicitud es enviada desde un navegador al servidor web y éste direcciona la solicitud al servidor WSGI, el cual se comunica con la aplicación Flask para procesar la solicitud y generar una respuesta, que posteriormente regresa al servidor WSGI, luego al servidor web y finalmente al navegador web [69].

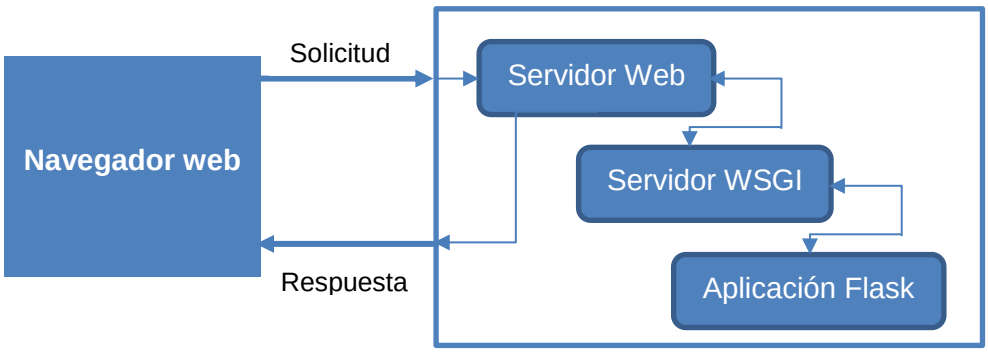


Figura 11 – Ciclo de petición - respuesta a un servidor web usando una aplicación en Flask. Adaptado de [69].

Para la puesta en producción en encuestas o censos del INEC, los aplicativos se deben configurar para acceder a bases de datos PostgreSQL de manera que los datos recopilados en formularios físicos o dispositivos electrónicos sean codificados por lotes o en tiempo real. Además, los aplicativos desarrollados se deben incorporar al Sistema Integrado de Producción Estadística (SIPE) existente. Esta implementación no se incluye en el alcance del presente proyecto.

3. RESULTADOS Y DISCUSIÓN

En esta sección se presentan los resultados del entrenamiento, validación, selección y evaluación de los mejores modelos para la clasificación de actividades económicas y ocupaciones. También se presenta un análisis de los resultados enfocado en garantizar la calidad de la codificación cuando los modelos sean implementados. Finalmente se realiza un análisis comparativo de los resultados obtenidos con la literatura científica.

3.1 Modelo de clasificación de actividades económicas

Para evaluar y seleccionar los modelos entrenados usando los algoritmos Xgboost, Redes Neuronales Feedforward y Redes Neuronales LSTM se calculó la exactitud y las métricas macro de precisión, sensibilidad y puntuación F1. Los datos de validación se usaron en el ajuste de hiperparámetros para seleccionar el mejor modelo de cada algoritmo en función de la puntuación F1 macro, ver anexos X, XII y XIII. En las tablas 4 y 5 se indican los resultados de entrenamiento y validación de los mejores modelos.

Tabla 4 – Resultados de los modelos de clasificación de actividades económicas con datos de entrenamiento

modelo	***exactitud %***	**macro – promedio**		
		precisión %	***sensibilidad %***	***puntuación - F1 %***
Xgboost	96.27	96.73	95.29	95.94
Redes Neuronales Feedforward	98.40	97.21	95.94	96.45
Redes Neuronales LSTM	97.12	93.07	90.83	91.39

Tabla 5 – Resultados de los modelos de clasificación de actividades económicas con datos de validación

modelo	***exactitud %***	**macro – promedio**		
		precisión %	***sensibilidad %***	***puntuación - F1 %***
Xgboost	91.92	85.48	80.21	81.59
Redes Neuronales Feedforward	94.90	88.39	86.44	86.60
Redes Neuronales LSTM	93.20	80.62	80.74	79.40

Los resultados de las métricas de evaluación del conjunto de datos de prueba para seleccionar el mejor modelo de actividades económicas se indican la Tabla 6. La selección del modelo de mejor rendimiento se realizó con base en la puntuación F1 macro que es adecuada para clases desbalanceadas. El modelo de Redes Neuronales Feedforward tiene el mayor valor de puntuación F1 macro, que es 87.68%. Este valor representa el promedio entre la precisión y la sensibilidad, es decir que tan confiable es el modelo para predecir determinada clase y que tan bien predice una clase.

Tabla 6 – Resultados de los modelos de clasificación de actividades económicas con datos de prueba.

modelo	exactitud			macro – promedio		
	entrenamiento %	*validación %*	*prueba %*	*precisión %*	*sensibilidad %*	*puntuación - F1 %*
Xgboost	96.27	91.92	91.91	85.09	80.38	81.64
Redes Neuronales Feedforward	98.40	94.90	**95.18**	89.46	87.52	**87.68**
Redes Neuronales LSTM	97.12	93.20	93.45	83.26	80.93	80.83

En la Tabla 6 también se puede observar que el modelo con menor rendimiento es el de Redes Neuronales LSTM con una puntuación F1 macro de 80.83%, y el modelo que ocupa el segundo lugar es Xgboost con una puntuación F1 macro de 81.64%. Además, se puede notar que el modelo de Redes Neuronales Feedforward tiene los mejores valores de exactitud, precisión y sensibilidad. La exactitud del 95.18% representa la tasa de aciertos, también nos indica una tasa de error menor al 5%, lo cual es aceptable de acuerdo con lo establecido por algunos autores respecto a la codificación manual, ver 2.1.1. Adicionalmente, se puede evidenciar que el modelo está generalizando el aprendizaje porque el valor de la exactitud de prueba (95.18%) no difiere en gran porcentaje de la exactitud de entrenamiento (98.40%). Con respecto al costo computacional, el entrenamiento de Xgboost duró aproximadamente 45 minutos, el de Redes Neuronales LSTM 2 horas, mientras que el de redes Neuronales Feedforward duró 10 minutos.

En la Figura 12 se indica las curvas de aprendizaje del modelo de Redes Neuronales Feedforward, mediante las cuales se puede evaluar el rendimiento del modelo con los datos de entrenamiento y validación. Como se puede apreciar, la exactitud y la perdida

varían de acuerdo con el número de épocas. También se puede observar que el entrenamiento se detiene en la época 28 para evitar el sobreajuste cuando la pérdida de validación comienza a incrementar mientras que la pérdida de entrenamiento continúa disminuyendo.

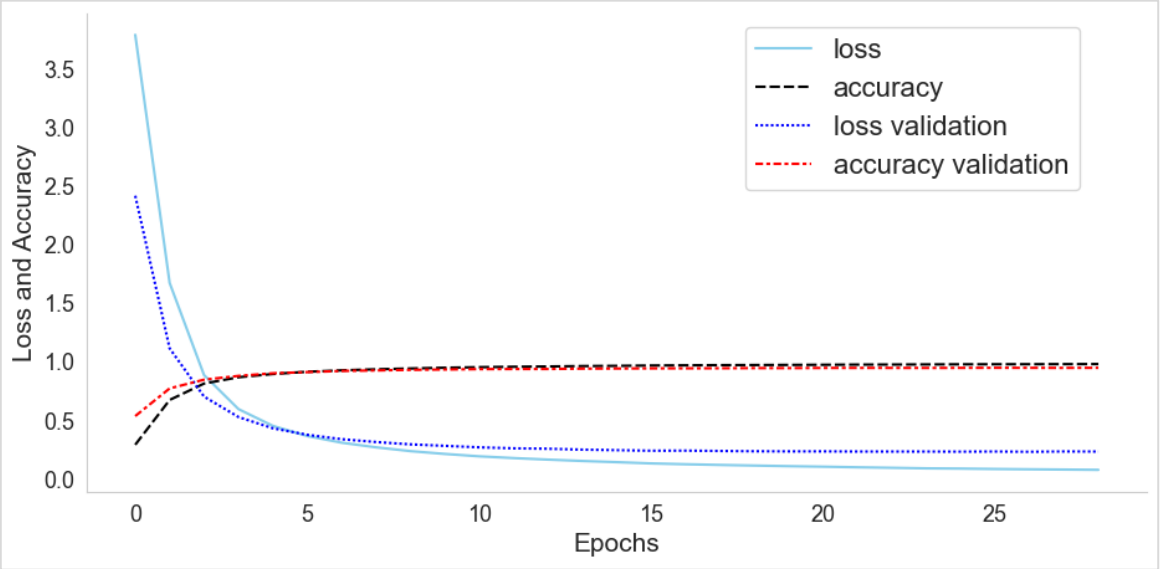


Figura 12 – Curvas de aprendizaje del modelo de actividades económicas usando Redes Neuronales Feedforward.

El resultado del modelo de clasificación de actividades económicas es un código CIIU 4.0 y una probabilidad de predicción asociada a dicho código. En la Tabla 7 se indica la tasa de aciertos y errores en función de diferentes rangos de probabilidad de predicción, en la cual se puede notar que el 87,66% de aciertos se concentran en un rango de probabilidad del 90 al 100%. También se observa que en los rangos de probabilidad inferiores existen pocos aciertos. Tomando en cuenta esta distribución de aciertos y errores se puede reducir el error al disminuir la tasa de producción mencionada en 1.5.4. Para ello, se debe definir un umbral de probabilidad que permita excluir las observaciones que se encuentran por debajo de dicho umbral, para que puedan ser codificadas por especialistas y no predichas por el modelo.

Tabla 7 – Tasa de aciertos y errores respecto al rango de probabilidad de predicción del modelo de actividades económicas.

RANGO DE PROBABILIDADES	TASA DE ACIERTOS	TASA DE ERRORES
Rango (90-100) %	**87.66%**	1.12%
Rango (80-90) %	3.16%	0.62%
Rango (70-80) %	1.74%	0.48%
Rango (60-70) %	1.00%	0.70%
Rango (50-60) %	0.82%	0.68%
Rango (40-50) %	0.46%	0.54%
Rango (30-40) %	0.24%	0.33%
Rango (20-30) %	0.06%	0.22%
Rango (10-20) %	0.04%	0.09%
Rango (0-10) %		0.03%
TOTAL	95.18%	4.82%

Como se observa en la Figura 13 el error varía en función de la tasa de producción para un determinado umbral de probabilidad. Por ejemplo, para un umbral del 90% con una tasa de producción del 88.78%, el error del modelo se reduce a 1.27%. En la práctica, el umbral debe ser definido con la finalidad de lograr un equilibrio entre la tasa de producción y el error deseados.

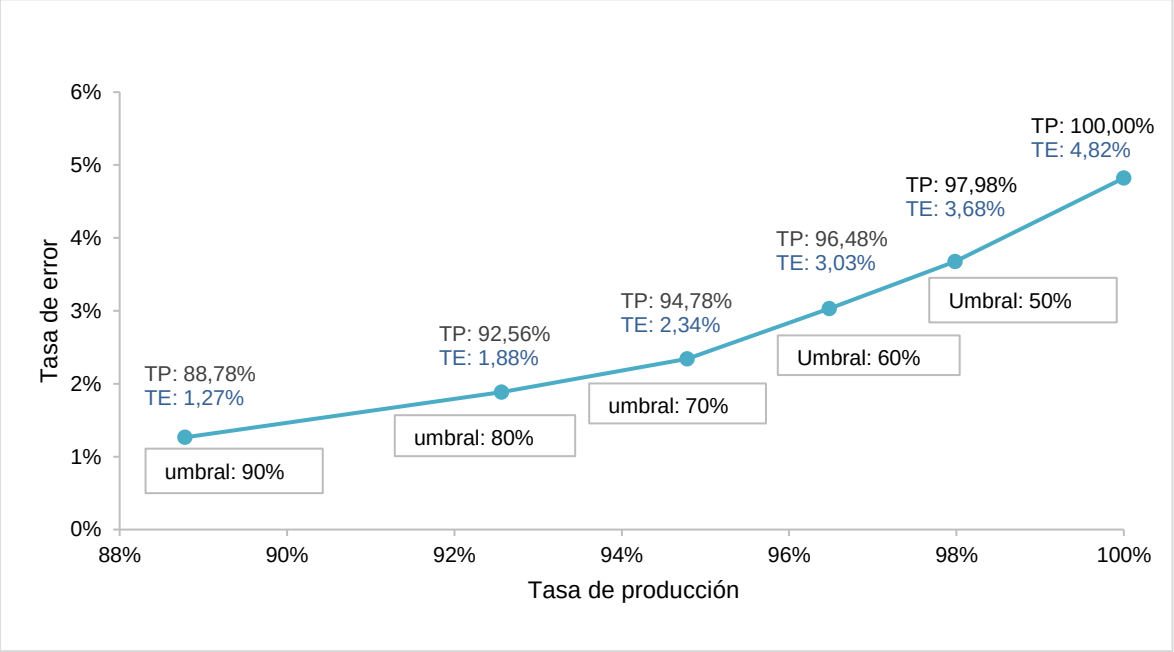


Figura 13 – Tasa de error y tasa de producción para diferentes umbrales de probabilidad de predicción del modelo de actividades económicas.

Como se mencionó en 1.5.1, la clasificación de actividades económicas CIIU 4.0 es jerárquica y tiene *secciones* en el primer nivel de agregación. Generalmente, estas secciones se utilizan para presentar resultados de las operaciones estadísticas. Además, dado que la representación gráfica de los resultados de un modelo de clasificación con un gran número de clases es poco entendible y práctica, se agrupó las 321 clases del modelo de actividades económicas dentro de las 21 secciones de la CIIU 4.0.

En las Figuras 14 y 15 se indican las distribuciones de frecuencias de las actividades económicas a nivel de sección del conjunto de datos de prueba y de sus predicciones, obtenidas con el modelo de Redes Neuronales Feedforward. Se puede observar que la mayor diferencia porcentual entre distribuciones es 0.05% para las secciones F y O. También se calculó la divergencia de Kullback Leibler usando el software estadístico R, el resultado fue $D_{KL}(P||Q) = 0.0000422378$, este valor es muy pequeño e indica que la distribución de las predicciones es bastante similar a la distribución de los datos reales.

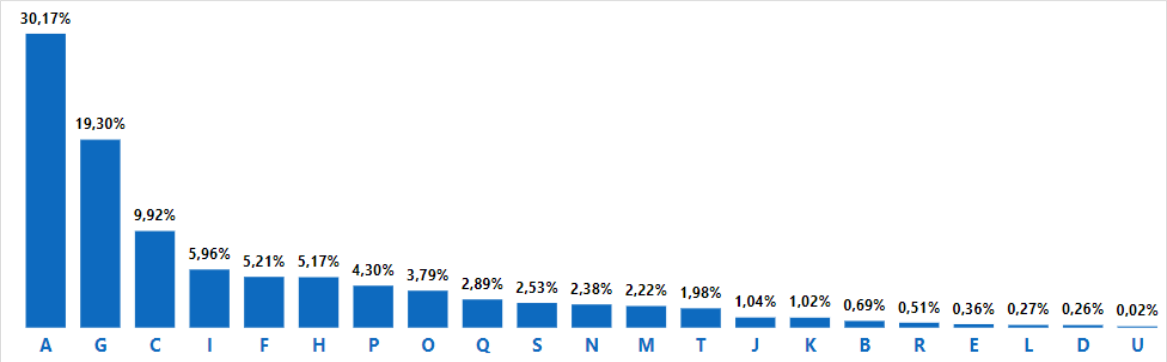


Figura 14 – Distribución de frecuencias del conjunto de datos de prueba, a nivel de secciones de la CIIU 4.0**.**

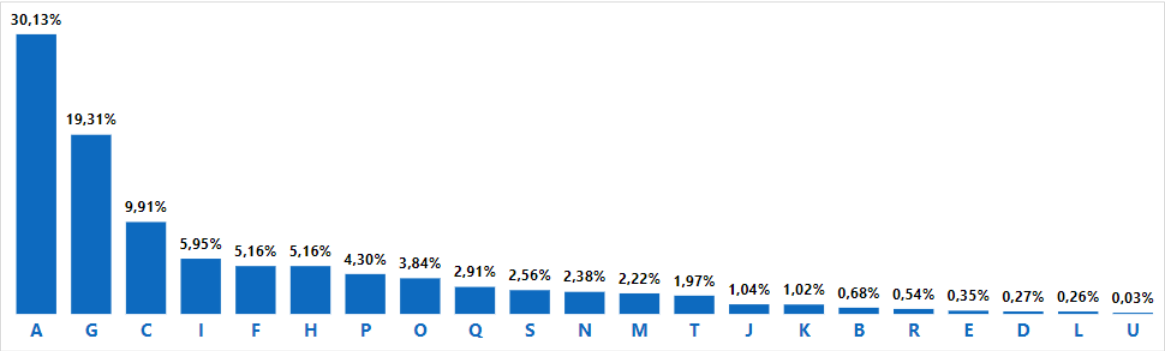


Figura 15 – Distribución de frecuencias de las predicciones del conjunto de datos de prueba, a nivel de secciones de la CIIU 4.0.

En las Figuras 16, 17 y 18 se presentan las matrices de confusión de entrenamiento, validación y prueba, a nivel de las 21 secciones de la clasificación de actividades económicas.

Predicciones

SECCION	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	Total
A	48517		22			2	8						1	1	1							48552
B		1095	8																			1103
C	10	2	15873	1		6	20	1	24	2			4	1			1	1	4			15950
D			1	424																		425
E					567		4								3							574
F			9	3		8338	5	6		4			13	10			1		6			8395
G	3		23			2	30958	4	17	3				1			2	1	2			31016
H					2	5	2	8320		1	1			2	6							8339
I			23			1	19		9551								2	2	1			9599
J			1			4	3	2		1633			10	2					1			1656
K										1	1641	1	3	2								1648
L			1				2					433	1	2	2							441
M			5			8	2		2	3			3536	6	1		1	1				3565
N			1			9	2	5		1		1	16	3794	1	2	5	4		13		3854
O	1		1		5	2		5		4	1		13		6070	1	7		2			6112
P								1			1		2		1	6918	6	3	3			6935
Q			4						1				3	4	7	5	4635		4	3		4666
R			2			1	2		1	3			4	3	3	11		790	3			823
S	1	1	15			5	13	9	1	1		2	4	4	6	2	18	5	3989		1	4077
T														11			11			3159		3181
U																			2		39	41
Total	48532	1098	15989	428	574	8383	31040	8353	9597	1656	1644	437	3610	3843	6101	6939	4689	807	4017	3175	40	160952

Figura 16 – Matriz de confusión del conjunto de datos de entrenamiento a nivel de secciones de la CIIU 4.0.

Predicciones

SECCION	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	Total
A	6058		2			1	1		1					2	2		1	1				6069
B		138																				138
C	7	2	1939	2		2	13	1	10	2	1	1	2	3	2			2	4			1993
D				51		1									1							53
E					68		1								3							72
F	1		6	3		1017		3		3			10	3	2				2			1050
G	2	1	13		3	1	3838	3	5		1		1	3			2	2	2			3877
H						2	4	1029					1	6			1					1043
I	2		13				6	1	1171			1	1					3	2			1200
J			2			3	5			188			3	1	1		1	1				205
K											199		3	1	1		1					205
L											1	53		1								55
M		3	7			5							418	4	5		3		1			446
N	1	1	1			5	4	3	1	2	2		6	444			4	1	2	6		483
O	2							1					2	2	753	1	1		1			763
P															2	866						868
Q			1									1	1		2	1	575		3			584
R										1			1		1			102				105
S			4					1									2		501			508
T									1					1			3			392		397
U																			1		4	5
Total	6073	145	1988	56	71	1037	3872	1042	1189	196	204	56	449	471	775	868	594	112	519	398	4	20119

Figura 17 – Matriz de confusión del conjunto de datos de validación a nivel de secciones de la CIIU 4.0.

En la diagonal se observa el número de observaciones por sección que fueron predichas correctamente. Además, fuera de la diagonal se puede ver en que secciones se produce la mayor cantidad de errores, lo cual es útil para los analistas de codificación. También, en la Figura 18 se puede notar que existen 11 observaciones de la sección G “Comercio” que fueron predichas en la sección C “Manufactura”. Revisando cada una de estas observaciones, se encontró que en las descripciones existen palabras como elaboración y venta que está asociadas a las secciones C y G respectivamente.

Predicciones

SECCION	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	Total
A	6052		5			2	6		2					2								6069
B		134	3											1								138
C	5	1	1950	2	1	1	10		6	2			3	6	2			1	5			1995
D			1	51		1																53
E					69									2					1			72
F	1		6	2		1019		2		4			6	6	1				1			1048
G	2	1	11			1	3852	1	4	3				2	1			1	3			3882
H						1	1	1028		1				6	3	1						1041
I			6				9		1183					1					1			1200
J			1			1	3			196			5		1			2				209
K								1			202		1		2							206
L							1					50		2	2							55
M			4			4				2	1	1	424	3	4	2			1			446
N	1					8	1	3	1	1		1	5	445	1	2	3	3	2	2		479
O	1				1			2			1				750	1	6		1			763
P							1				1		1	1	1	858		2	1			866
Q									1						2		575	1	2		1	582
R													1		1			99	1			102
S			6					1	1			1		1	2	1			497			510
T														1			2			395		398
U																					5	5
Total	6062	136	1993	55	71	1038	3884	1038	1198	209	205	53	446	479	773	865	586	109	516	397	6	20119

Figura 18 – Matriz de confusión del conjunto de datos de prueba a nivel de secciones de la CIIU 4.0.

3.2 Modelo de clasificación de ocupaciones

La selección y evaluación del modelo de ocupaciones se realizó con las mismas métricas y consideraciones utilizadas para el modelo de actividades económicas. Los datos de validación se usaron en el ajuste de hiperparámetros para seleccionar el mejor modelo de cada algoritmo en función de la puntuación F1 macro, ver anexos XI, XIV y XV. En las tablas 8 y 9 se indican los resultados de entrenamiento y validación de los mejores modelos.

Tabla 8 – Resultados de los modelos de clasificación de ocupaciones con datos de entrenamiento

modelo	*exactitud %*	macro – promedio		
		precisión %	*sensibilidad %*	*puntuación - F1 %*
Xgboost	85.36	75.87	79.93	77.16
Redes Neuronales Feedforward	94.19	90.28	83.47	85.45
Redes Neuronales LSTM	90.63	81.01	73.96	75.79

Tabla 9 – Resultados de los modelos de clasificación de ocupaciones con datos de validación

modelo	*exactitud %*	macro – promedio		
		precisión %	*sensibilidad %*	*puntuación - F1 %*
Xgboost	81.01	57.13	57.51	56.13
Redes Neuronales Feedforward	87.44	66.33	60.50	61.81
Redes Neuronales LSTM	83.16	53.66	50.29	50.66

Los resultados de las medidas de evaluación del conjunto de datos de prueba para seleccionar el mejor modelo de ocupaciones, se indican la Tabla 10, en la cual se puede observar que el mayor valor de la puntuación F1 macro es 60.02% que corresponde al modelo de Redes Neuronales Feedforward. Dicha tasa representa que tan confiable es el modelo para predecir determinada clase y que tan bien predice una clase.

Tabla 10 – Resultados de los modelos de clasificación de ocupaciones con datos de prueba.

modelo	exactitud			macro – promedio		
	entrenamiento %	*validación %*	*prueba %*	*precisión %*	*sensibilidad %*	*puntuación - F1 %*
Xgboost	85.36	81.01	80.50	58.18	58.17	56.67
Redes Neuronales Feedforward	94.19	87.44	**86.85**	63.84	59.30	**60.02**
Redes Neuronales LSTM	90.63	83.16	82.89	54.16	50.22	50.82

En la Tabla 10 también se puede observar que el modelo con menor rendimiento es el de Redes Neuronales LSTM con una puntuación F1 macro de 50.82% y el modelo que ocupa el segundo lugar es Xgboost con una puntuación F1 macro de 56.67%. Además, dado el valor de la exactitud del modelo 86.85%, se calculó el error del 13.15% que es un valor alto y no aceptable de acuerdo con lo establecido por algunos autores respecto a la codificación manual. Sin embargo, es un error esperado en comparación con otros estudios relacionados que obtienen errores similares. Una estrategia para reducir el error de codificación que se produce con el modelo es definir una tasa de producción menor al 100%. En cuanto al tiempo de entrenamiento, Xgboost tardo aproximadamente 1 hora, las Redes Neuronales LSTM 2.5 horas mientras que las Redes Neuronales Feedforward 15 minutos.

En la Figura 19 que indica las curvas de aprendizaje de la exactitud y la pérdida del modelo de Redes Neuronales Feedforward, se observa que el modelo tiende a sobreajustarse rápidamente. De hecho, el entrenamiento se detiene en la época 10 para evitar el sobreajuste cuando la pérdida de validación comienza a incrementar mientras que la pérdida de entrenamiento continúa disminuyendo.

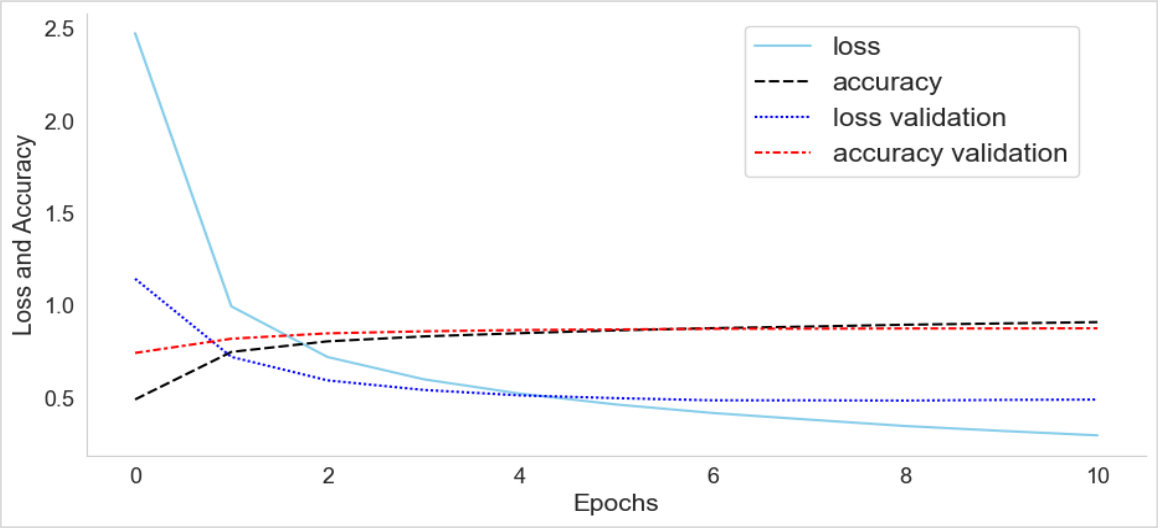


Figura 19 – Curvas de aprendizaje del modelo de ocupaciones usando Redes Neuronales Feedforward.

En la Tabla 11 se indica la tasa de aciertos y errores en función de diferentes rangos de probabilidad de predicción, en la cual se puede notar que el 67,19% de aciertos se concentran en un rango de probabilidad del 90 al 100%. También se observa que en los rangos de probabilidad inferiores existen pocos aciertos. Tomando en cuenta esta

distribución de aciertos y errores, al definir un umbral de probabilidad se puede reducir el error disminuyendo la tasa de producción, de modo que las observaciones que se encuentran por debajo de dicho umbral puedan ser codificadas por especialistas y no predichas por el modelo.

Tabla 11 – Tasa de aciertos y errores respecto al rango de probabilidad de predicción del modelo de ocupaciones.

RANGO DE PROBABILIDADES	TASA DE ACIERTOS	TASA DE ERRORES
Rango (90-100) %	**67.19%**	2.16%
Rango (80-90) %	7.93%	1.38%
Rango (70-80) %	4.18%	1.46%
Rango (60-70) %	3.00%	1.57%
Rango (50-60) %	2.41%	1.98%
Rango (40-50) %	1.05%	1.65%
Rango (30-40) %	0.68%	1.43%
Rango (20-30) %	0.30%	1.02%
Rango (10-20) %	0.11%	0.49%
Rango (0-10) %		0.02%
TOTAL	86.85%	13.15%

Como se observa en la Figura 18 el error varía en función de la tasa de producción para un determinado umbral de probabilidad. Por ejemplo, para un umbral del 80% con una tasa de producción del 78.65%, el error del modelo se reduce a 4.49%, que es menor al error aceptable del 5%. En la práctica, el umbral debe ser definido con la finalidad de lograr un equilibrio entre la tasa de producción y el error deseados.

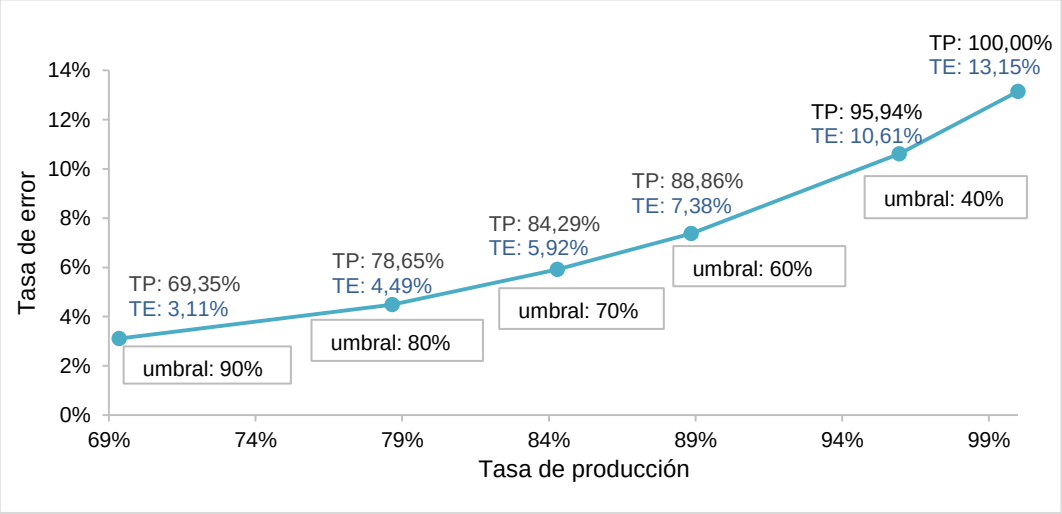


Figura 20 – Tasa de error y tasa de producción para diferentes umbrales de probabilidad de predicción del modelo de ocupaciones.

Al igual que la clasificación de actividades económicas CIIU 4.0, la clasificación de ocupaciones CIUO 08 es jerárquica y tiene *grandes grupos* en el primer nivel de agregación. Por tal motivo, para la presentación de resultados también se agrupó las 344 clases del modelo de ocupaciones dentro de los 10 grandes grupos de la CIUO 08.

En las Figuras 21 y 22 se indican las distribuciones de frecuencias del conjunto de datos de prueba de las ocupaciones y sus predicciones a nivel de grandes grupos, obtenidas con el modelo de Redes Neuronales Feedforward. Se puede observar que la mayor diferencia porcentual entre distribuciones es 0.54% en el gran grupo 6. También se calculó la divergencia de Kullback Leibler, el resultado fue $D_{KL}(P||Q) = 0.0005566866$, este valor es muy pequeño e indica que la distribución de las predicciones es bastante similar a la distribución de los datos reales.

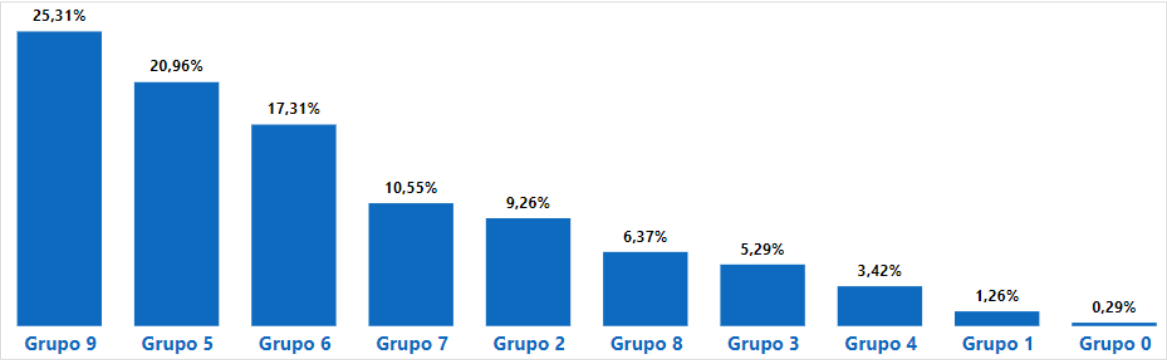


Figura 21 – Distribución de frecuencias del conjunto de datos de prueba, a nivel de grandes grupos de la CIUO 08.

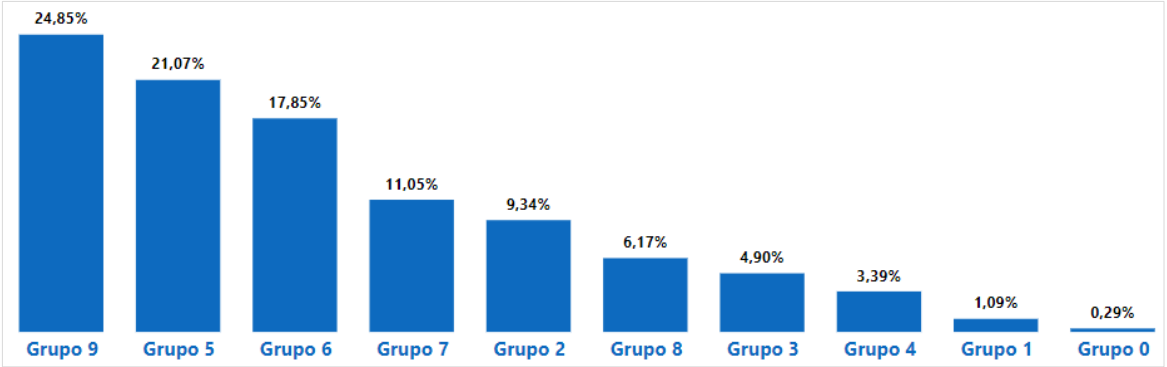


Figura 22 – Distribución de frecuencias de las predicciones del conjunto de datos de prueba, a nivel de grandes grupos de la CIUO 08.

En las Figuras 23, 24 y 25 se presentan las matrices de confusión de los datos de entrenamiento, validación y prueba respectivamente, a nivel de grandes grupos de la clasificación de ocupaciones.

Predicciones

GRAN GRUPO	Grupo 0	Grupo 1	Grupo 2	Grupo 3	Grupo 4	Grupo 5	Grupo 6	Grupo 7	Grupo 8	Grupo 9	Otros	Total
Grupo 9		1	3	12	23	249	710	293	66	40965		**42322**
Grupo 8		2	3	16	5	16	6	117	10439	72		**10676**
Grupo 7		11	33	62	4	84	18	17275	44	125		**17656**
Grupo 6		3		1	3	4	28791	6	4	154		**28966**
Grupo 5		32	31	113	44	34350	7	92	10	375		**35054**
Grupo 4		29	63	261	5209	101	12	5	7	30		**5717**
Grupo 3	1	54	395	7870	193	147	22	154	11	20	1	**8868**
Grupo 2		77	14909	274	56	30	28	106	2	3	54	**15539**
Grupo 1		1868	59	46	21	66	25	18	2		13	**2118**
Grupo 0	477		1									**478**
Total	**478**	**2077**	**15497**	**8655**	**5558**	**35047**	**29619**	**18066**	**10585**	**41744**	**68**	**167394**

Figura 23 – Matriz de confusión del conjunto de datos de entrenamiento a nivel de grandes grupos de la CIUO 08.

Predicciones

GRAN GRUPO	Grupo 0	Grupo 1	Grupo 2	Grupo 3	Grupo 4	Grupo 5	Grupo 6	Grupo 7	Grupo 8	Grupo 9	Total
Grupo 0	60										**60**
Grupo 1		191	28	12	9	20	6	1			**267**
Grupo 2		19	1745	107	22	14	7	32	1	1	**1948**
Grupo 3		14	117	802	74	48	2	40	3	13	**1113**
Grupo 4		4	24	77	557	23	4	6	5	11	**711**
Grupo 5		8	11	46	23	4194	4	17	6	69	**4378**
Grupo 6		2		3		3	3573	3	2	32	**3618**
Grupo 7		5	4	30	2	27	8	2074	20	39	**2209**
Grupo 8				3	6	8		32	1248	38	**1335**
Grupo 9			1	9	3	59	111	65	23	5014	**5285**
Total	**60**	**243**	**1930**	**1089**	**696**	**4396**	**3715**	**2270**	**1308**	**5217**	**20924**

Figura 24 – Matriz de confusión del conjunto de datos de validación a nivel de grandes grupos de la CIUO 08.

En la diagonal se observa el número de observaciones por gran grupo que fueron predichas correctamente. Además, fuera de la diagonal se puede ver en que grandes grupos se produce la mayor cantidad de errores. También, en la Figura 25 se puede notar que existen 119 observaciones del gran grupo 9 “Ocupaciones Elementales” que fueron predichas en el gran grupo 6 “Agricultores y Trabajadores Calificados Agropecuarios, Forestales y Pesqueros”. Revisando cada una de las observaciones se encontró que las descripciones textuales de ocupación en los dos grandes grupos son similares, sin embargo, tienen

diferentes variables de apoyo, es decir al parecer hay ciertas reglas que el modelo no aprendió. Esto podría ser debido a que existen muy pocas observaciones de dichas reglas.

Predicciones

GRAN GRUPO	Grupo 0	Grupo 1	Grupo 2	Grupo 3	Grupo 4	Grupo 5	Grupo 6	Grupo 7	Grupo 8	Grupo 9	Total
Grupo 0	60										**60**
Grupo 1		185	27	13	10	15	8	4	1		**263**
Grupo 2	1	18	1757	96	24	14	3	25			**1938**
Grupo 3		14	112	764	85	64	6	48	3	10	**1106**
Grupo 4		5	26	75	549	42	4	5	1	8	**715**
Grupo 5		6	18	49	24	4176	4	27	9	72	**4385**
Grupo 6			2	2	2	2	3587	1	3	23	**3622**
Grupo 7		1	10	18	2	22	3	2086	14	52	**2208**
Grupo 8			1	4	2	10	1	38	1241	35	**1332**
Grupo 9			2	4	11	63	119	78	19	5000	**5296**
Total	**61**	**229**	**1955**	**1025**	**709**	**4408**	**3735**	**2312**	**1291**	**5200**	**20925**

Figura 25 – Matriz de confusión del conjunto de datos de prueba a nivel de grandes grupos de la CIUO 08.

3.3 Evaluación de los modelos con datos de la ENEMDU mensual.

Se evaluó el desempeño de los modelos de clasificación de actividad económica y ocupación utilizando datos de la encuesta mensual ENEMDU, de los periodos septiembre 2021 hasta mayo 2022. Debido a que el objetivo principal de la encuesta en el proceso de codificación es minimizar los errores de asignación de códigos, la métrica de evaluación fue la exactitud. En la Figura 26 se puede observar que el desempeño de los modelos con diferentes conjuntos de datos es estable en el tiempo. También se puede notar en los periodos evaluados que la exactitud se reduce ligeramente en comparación con la exactitud de los modelos evaluados con los datos de prueba. Dado que la exactitud del modelo de actividades económicas es 95.18%, en octubre 2021 se redujo en 2.58%, mientras que la exactitud del modelo de ocupaciones, 86.85%, se redujo en 2.72% en noviembre 2021.

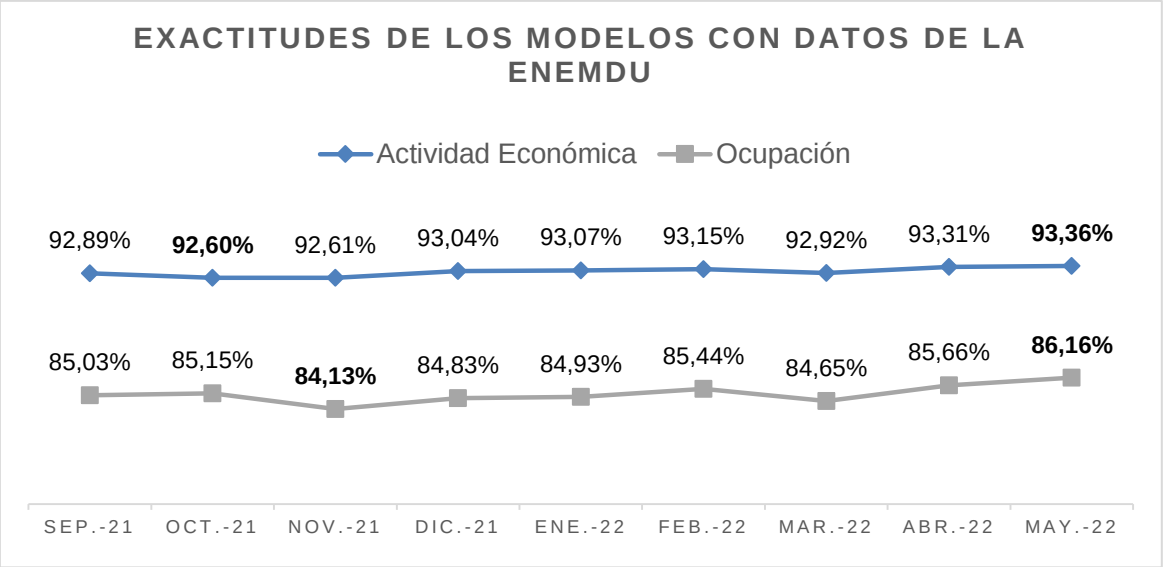


Figura 26 – Exactitud de los modelos de clasificación con datos de la encuesta mensual ENEMDU septiembre 2021- mayo 2022.

El promedio de observaciones que se codifican mensualmente en la encuesta es 15.000, que es una muestra de menor tamaño en comparación con el tamaño de las muestras con las que se evaluaron los modelos, 20.119 para actividades económicas y 20.925 para ocupaciones. Debido a esta situación es posible que las distribuciones sean distintas, de hecho, es posible que ciertas clases no existan en algunos periodos. Esto puede justificar el menor rendimiento de los modelos. Además que, existen cambios periódicos en la metodología de asignación de códigos.

Adicionalmente, el tiempo en que se realizaron las predicciones de actividad económica y ocupación de la ENEMDU mensual utilizando los modelos de clasificación desarrollados fue alrededor de 3 minutos. Por lo tanto, los modelos de clasificación permiten reducir el tiempo y el número de personas asignadas al proceso de codificación. Por ejemplo, en la ENEMDU, 10 codificadores resuelven 15.000 casos en 6 días, ya que en promedio una persona codifica 250 casos de actividad económica y 250 casos de ocupación diariamente. Considerando una tasa de error de los modelos de 5% o menos, se debe establecer una tasa de producción del 100% para actividades económicas y del 78.65% para ocupaciones. Dicha información sería codificada automáticamente en minutos, mientras que el 23.35% de ocupaciones restantes, serían codificadas por 5 personas en aproximadamente un día y medio.

3.4 Análisis y comparación de los resultados con estudios relacionados

En la Tabla 1 que se encuentra en 2.1.1 se recopiló información de estudios relacionados, en los cuales se desarrolló modelos de aprendizaje automático para la codificación de actividades económicas y ocupaciones de países como Corea del Sur **(M1, M2)**, Estados Unidos **(M3)**, Alemania **(M4, M5, M6, M7, M8, M9)**, Canadá **(M10)**, México **(M11, M12)** e Inglaterra **(M13)**. Los estudios fueron publicados desde el 2008 hasta 2021, por lo que se pudo evidenciar cómo evolucionó el uso de diferentes técnicas y algoritmos, y como éstos impactaron en los resultados. Se usaron algoritmos como Máxima Entropía, Naive Bayes, Bayes Multinomial, Regresión Logística, Boosting, K-Vecinos Cercanos, Máquinas de Soporte Vectorial, Random Forest, Redes Neuronales y modelos en los que se combinan varios algoritmos. Para vectorizar el texto se usó n-gramas, TF-IDF y las incrustaciones preentrenadas FastText. Además, el tamaño del conjunto de datos de entrenamiento varía desde los miles cuando proviene de encuestas hasta los millones en el caso de censos.

En general, se puede notar que los mejores resultados se obtuvieron cuando se utilizaron más datos para el entrenamiento. Y los modelos con menor desempeño fueron aquellos donde se utilizó pocos datos y/o algoritmos como Naive Bayes, Bayes Multinomial y Regresión Logística. También se puede observar que los modelos de actividades económicas tienen mejor desempeño en comparación a los modelos de ocupaciones, excepto los modelos desarrollados en Inglaterra **(M13)**. Al comparar los resultados hay que tomar en cuenta que las clasificaciones usadas en los distintos países son adaptadas a la realidad nacional por lo que tienen diferente número de clases y son escritas en diferentes idiomas.

En la Tabla 12 se puede observar que las exactitudes de los modelos desarrollados en otras investigaciones son menores a las encontradas en esta investigación con 95.18% y 86.85%. El segundo lugar con 89.21% y 85.05% es para los modelos desarrollados en México **(M11)** usando TF-IDF (6 letras y 2 palabras) y varios algoritmos combinados como SVM, Regresión Logística, Random Forest, Redes Neuronales, Xgboost y KNN ensamblados con pesos diferentes.

Tabla 12 – Exactitudes de los modelos desarrollados en diferentes investigaciones.

Modelo	M4	M5	M6	M7	M8	M9	M10	M11	M13	Modelos de esta investigación
Actividad Económica	NA	NA	NA	NA	NA	NA	80.50	89.21	81.00	**95.18**
Ocupación	73.00	68.00	63.64	63.23	59.06	65.00	64.40	85.05	84.00	**86.85**

En la Tabla 13 se indica las tasas de producción con las cuales se obtiene un error aproximado de 5%, el máximo aceptado en codificación manual. Se puede notar que las mayores tasas de producción se obtienen con los modelos desarrollados en la presente investigación 100% y 78.65%. Esto quiere decir que, para mantener el error aceptable al implementar los modelos en la codificación en investigaciones sociodemográficas se debería enviar a revisión de expertos el 21.35% de las ocupaciones.

Tabla 13 – Tasas de producción cuando la tasa de error es aproximadamente 5%.

Modelo	M3	M5	M6	M8	M9	M10	M11	M13	Modelos de esta investigación
Actividad Económica	56.00	NA	NA	NA	NA	46.50	64.99	NA	**100**
Ocupación	30.00	45.80	43.25	45.36	55.00	34.20	64.99	73.00	**78.65**

En la Tabla 14 se puede observar que el modelo de actividades económicas desarrollado tiene tasas de producción sobre el 88.78% para errores del 1.27%, 1.88% y 2.34%. Estos resultados superan a los de otras investigaciones **(M2, M3, M12)** en las cuales, para obtener tasas de error similares las tasas de producción están entre 30% y 83.70%. Además, el modelo desarrollado en Corea del Sur **(M1)** usando el algoritmo de máxima entropía tiene la menor tasa de error 0.77% asociada a una tasa de producción del 74.40%, mientras que el modelo de esta investigación con una tasa de error de 1.27% tiene una tasa de producción del 88.78%.

Respecto al modelo de ocupaciones, en la Tabla 14 se indica que para tasas de error menores al 4%, el modelo desarrollado en México **(M12)** tiene tasas de producción mayores en comparación con el modelo desarrollado en este proyecto. Además, para una tasa de error menor al 2% el modelo de Corea del Sur **(M2)** tiene la mayor tasa de

producción, 72.90%. La menor tasa de error 0.47% tiene el modelo **(M1)** con una tasa de producción del 66.31%.

Tabla 14 – Tasa de producción cuando la tasa de error es menor al 5%.

MODELO	ACTIVIDAD ECONÓMICA		OCUPACIÓN	
	T. PRODUCCION	**T. ERROR**	**T. PRODUCCION**	**T. ERROR**
M1	74.40%	0.77%	66.31%	0.47%
M2	83.70%	1.25%	72.90%	1.59%
M3	50% 40% 30%	3.96% 2.12% 1.01%	NA	NA
M12	78.56% 64.08%	2.71% 1.55%	82.39% 73.44% 55.14%	3.87% 2.56% 1.29%
Modelos de esta investigación	94.78% 92.56% 88.78%	2.34% 1.88% 1.27%	73.83% 67.93% 50.72%	3.64% 2.95% 1.43%

4. CONCLUSIONES Y RECOMENDACIONES

4.1 CONCLUSIONES

- En este trabajo se desarrolló dos modelos de clasificación para codificar automáticamente las actividades económicas y ocupaciones de investigaciones sociodemográficas. Para lo cual, se diseñaron y evaluaron modelos usando los algoritmos Xgboost y Redes Neuronales Artificiales de tipo Feedforward y LSTM. Los mejores resultados de exactitud se obtuvieron con las Redes Neuronales Feedforward, 95.18% para actividad económica y 86.85% para ocupaciones.

- La exactitud del modelo de Redes Neuronales Feedforward, 95.18% para actividades económicas fue mejor en comparación con los resultados encontrados en la literatura científica (Canadá 80.5 %, México 89.21% e Inglaterra 81%). Del mismo modo, la exactitud del 86.85% del modelo de Redes Neuronales Feedforward para ocupaciones, fue mejor en comparación con estudios realizados en otros países (Alemania 73%, Canadá 64.4%, México 85.05% e Inglaterra 84%).

- Los modelos de actividad económica y ocupación entrenados con Redes Neuronales LSTM tuvieron una exactitud de 93.45% y 82.89%, respectivamente. Dichos resultados son más bajos en comparación con las Redes Neuronales Feedforward (95.18% y 86.85%). Teniendo en cuenta que el 87.46% de las descripciones de actividades económicas y el 89.13% de ocupaciones tienen una longitud no mayor a 7 palabras, se considera que aquello influyó en el menor desempeño de las Redes Neuronales LSTM, debido a que las secuencias muy cortas no tienen un contexto amplio que aprender.

- En la evaluación de los modelos de actividades económicas y ocupaciones, se encontró que las exactitudes de las Redes Neuronales LSTM (93.45% y 82.89%) fueron mayores en comparación con las exactitudes de Xgboost (91.91% y 82.47%). En sentido opuesto, las puntuaciones F1 macro de Xgboost (81.64% y 55.82%) fueron mayores en comparación con las puntuaciones F1 macro de Redes Neuronales LSTM (80.83% y 50.82%). Por lo cual, se dedujo que las clases desbalanceadas influyeron en la menor puntuación F1 macro de las Redes Neuronales LSTM.

- Se realizó el balanceo de las 321 clases de actividades económicas y las 344 clases de ocupaciones con técnicas de sobremuestreo y submuestreo. Con el balanceo el rendimiento de los modelos no mejoró, incluso cuando se usó Xgboost el tiempo de entrenamiento incrementó de horas a días. También se identificó que, al realizar el submuestreo de las clases mayoritarias, la sensibilidad y precisión disminuyó, debido a que se perdió información importante para clasificar correctamente dichas clases.

- Los modelos de clasificación desarrollados en este trabajo permitieron reducir el tiempo y el número de personas destinadas al proceso de codificación, en comparación con el proceso 100% manual. Con el proceso automático se codificó alrededor de 15.000 observaciones de la ENEMDU mensual en 3 minutos, mientras que aproximadamente 10 personas en 6 días codifican la misma cantidad de observaciones.

- En la literatura científica se encontró que el error aceptable de codificación en un proceso manual era máximo 5%. Por lo tanto, el proceso automático debería al menos garantizar el mismo límite de la tasa de error de codificación. Para disminuir la tasa de error del modelo de ocupaciones de 13.15% a 5% o menos, se tuvo que reducir la tasa de producción o codificación automática. Para lo cual, se utilizó la probabilidad asociada a las predicciones y se definió un umbral sobre el cual las observaciones predichas fueron consideradas como parte de la codificación automática. Para una tasa de producción de 78.65% y un umbral del 80%, la tasa de error del modelo de ocupaciones disminuyó a 4.49%.

- Para una tasa de codificación automática de actividades económicas del 100% y de ocupaciones del 78.65%, las tasas de error de los modelos son 4.82% y 4.49%, respectivamente. Si el 23.35% de ocupaciones que no fueron codificadas automáticamente son enviadas a un proceso manual, para codificar 15.000 observaciones de la ENEMDU se requieren 5 personas y aproximadamente un día y medio. Por lo tanto, en comparación con la codificación manual en la cual se requieren 10 personas y 6 días, en un proceso combinado (automático y manual) el personal se reduce a la mitad y el tiempo a la cuarta parte.

4.2 RECOMENDACIONES

- Dado que existen cambios habituales en las reglas de asignación de los códigos de actividad económica y ocupación, además debido al dinamismo del mercado laboral, se debe elaborar de manera periódica un documento de las actualizaciones realizadas. De esta manera, se puede planificar la recodificación de las etiquetas de las clases y el posterior reentrenamiento de los modelos. Si esta actividad no se realiza el rendimiento de los clasificadores automáticos disminuirá.

- Para garantizar la calidad de la codificación automática de actividades económicas y ocupaciones, inicialmente es importante realizar la revisión de una muestra de datos cuyas predicciones incluso superen el umbral de probabilidad establecido. Mediante dicha revisión se debe verificar que los resultados de los modelos se mantengan en el tiempo para diferentes conjuntos de datos de investigaciones sociodemográficas.

- Obtener más datos para las clases minoritarias. Se puede realizar un levantamiento de información direccionado en lugares donde se desarrollen determinadas actividades económicas u ocupaciones. También se puede utilizar datos de otras investigaciones sociodemográficas de mayor alcance, por ejemplo, el Censo de Población y Vivienda. Además, se puede solicitar datos de otras instituciones pertenecientes al Sistema Estadístico Nacional.

- Desarrollar un aplicativo informático que permita automatizar la codificación usando los modelos desarrollados. Mediante el aplicativo se debe obtener las predicciones, las probabilidades de predicción y se debe permitir ingresar un umbral de probabilidad para seleccionar el conjunto de datos que será codificado por expertos. Además, se debe incluir un módulo para el reentrenamiento automatizado de los modelos.

REFERENCIAS BIBLIOGRÁFICAS

[1] A. Hancock, “Lineamientos sobre mejores prácticas para desarrollar clasificaciones estadísticas internacionales.” *Grupo de Expertos en Clasificaciones Estadísticas Internacionales*, pp. 1-24, Nov. 06, 2013. [En línea]. Disponible en: https://tinyurl.com/25mhzzcw.

[2] N. Nahoomi, “Automatically Coding Occupation Titles to a Standard Occupation Classification.” Ontario, Canadá, Sep. 2018. [En línea]. Disponible en: http://hdl.handle.net/10214/14251.

[3] Instituto Nacional de Estadística y Censos. “Programa Nacional de Estadística del Ecuador noviembre 2014.” INEC. Nov. 2014. [En línea]. Disponible en: https://tinyurl.com/3fk6skpp.

[4] K. Valdivieso, I. Benítez, D. Muñoz, and M. Lastra. “Modelo de Producción Estadística del Ecuador 2016.” INEC. 2016. [En línea]. Disponible en: https://tinyurl.com/yc2wsb3t.

[5] M. Thompson, M. Kornbau, and J. Vesely, “Creating an Automated Industry and Occupation Coding Process for the American Community Survey.” *U.S. Census Bereau,* 2012. [En línea]. Disponible en: https://tinyurl.com/u33jhte4. (Acceso: 23 de septiembre de 2022).

[6] Instituto Nacional de Estadística y Censos, "VII Censo de Población y VI de Vivienda etapa de procesamiento", Memoria Técnica, 2011.

[7] Instituto Nacional de Estadística y Censos. “Encuesta Nacional de Empleo, Desempleo y Subempleo - ENEMDU Indicadores laborales.” INEC. Jun. 2022. [En línea]. Disponible en: https://tinyurl.com/2s3am99e.

[8] División de Estadística, “Clasificación Industrial Internacional Uniforme de todas las actividades económicas (CIIU).” vol. 4. Departamento de Asuntos Económicos y Sociales, New York, USA, 2009. [En línea]. Disponible en: https://tinyurl.com/2cdhyeuc.

[9] Instituto Nacional de Estadística y Censos. “Clasificación Nacional de Actividades Económicas (CIIU Rev. 4.0).” INEC. Jun. 2012. [En línea]. Disponible en: https://tinyurl.com/3hc4dvpk.

[10] International Labour Office. “International Standard Classification of Occupations 08 (ISCO-08).” ILO. 2012. [En línea]. Disponible en: https://tinyurl.com/evdyjry6.

[11] International Labour Office. “Resolución sobre la actualización de la Clasificación Internacional Uniforme de Ocupaciones.” ILO. 2008. [En línea]. Disponible en: https://tinyurl.com/3rbm6tn6.

[12] Instituto Nacional de Estadística y Censos. “Clasificación Nacional de Ocupaciones (CIUO 08).” INEC. Jun. 2012. [En línea]. Disponible en: https://tinyurl.com/yu2pwxhh.

[13] D. Rivadeneira and W. Villavicencio. “Metodología de la Encuesta Nacional de Empleo, Desempleo y Subempleo ENEMDU 2021 - 2024.” INEC. Apr. 2022. [En línea]. Disponible en: https://tinyurl.com/2758mtxf.

[14] J. Ruiz and J. Perez, “Natural language processing for the variables of Occupation and Economic activity.” *Instituto Nacional de Estadística y Geografía*, Nov. 2020. https://tinyurl.com/3zcs7bfe (acceso: 17 de septiembre de 2022).

[15] A. Bethmann, M. Schierholz, K. Wenzig, and M. Nester, “Automatic Coding of Occupations.” *Beyond traditional survey taking: adapting to a changing world, 2014 International Methodology Symposium, Statistics Canada*, vol. 2014, Oct. 2014. [En línea]. Disponible en: http://dx.doi.org/10.13140/RG.2.1.3321.1287. (Acceso: 26 de septiembre de 2022).

[16] K. Kowsari, K. J. Meimandi, M. Heidarysafa, S. Mendu, L. Barnes, and D. Brown, “Text Classification Algorithms: A Survey.” *Information*, vol. 10, no. 4, p. 150, 2019, doi: 10.3390/info10040150.

[17] V. K. Vijayan, K. R. Bindu, and L. Parameswaran, “A comprehensive study of text classification algorithms.” *2017 International Conference on Advances in Computing, Communications, and Informatics (ICACCI)*, pp. 1109-1113, 2017, doi: 10.1109/icacci.2017.8125990.

[18] M. Muntean and F. D. Militaru, “Design Science Research Framework for Performance Analysis Using Machine Learning Techniques.” *Electronics*, vol. 11, no. 16, p. 2504, 2022, doi: 10.3390/electronics11162504.

[19] K. N. Singh, S. D. Devi, H. M. Devi, and A. K. Mahanta, “A novel approach for dimension reduction using word embedding: An enhanced text classification

approach." *International Journal of Information Management Data Insights*, vol. 2, no. 1, pp. 10-61, Apr. 2022, doi: 10.1016/j.jjimei.2022.100061.

[20] A. K. Uysal and S. Gunal, "The impact of preprocessing on text classification." *Information Processing & Management*, vol. 50, no. 1, pp. 104-112, 2014, doi: 10.1016/j.ipm.2013.08.006.

[21] R. Dzisevic and D. Sesok, "Text Classification using Different Feature Extraction Approaches." *2019 Open Conference of Electrical, Electronic and Information Sciences (eStream)*, pp. 1-4, 2019, doi: 10.1109/estream.2019.8732167.

[22] Z. Wan, Y. Xu, and B. Šavija, "On the Use of Machine Learning Models for Prediction of Compressive Strength of Concrete: Influence of Dimensionality Reduction on the Model Performance." *Materials*, vol. 14, no. 4, p. 713, 2021, doi: 10.3390/ma14040713.

[23] P. S. Parmar, P. K. Biju, M. Shankar, and N. Kadiresan, "Multiclass Text Classification and Analytics for Improving Customer Support Response through different Classifiers." *2018 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, pp. 538-542, 2018, doi: 10.1109/icacci.2018.8554881.

[24] M. Hossin and M. Sulaiman, "A Review on Evaluation Metrics for Data Classification Evaluations." *International Journal of Data Mining & Knowledge Management Process*, vol. 5, no. 2, pp. 01-11, 2015, doi: 10.5121/ijdkp.2015.5201.

[25] "KLD: Kullback-Leibler Divergence (KLD)", RDocumentacion. https://www.rdocumentation.org/packages/LaplacesDemon/versions/16.1.6/topics/KLD (acceso: 10 de diciembre de 2022).

[26] C.-Z. Liu, Y.-X. Sheng, Z.-Q. Wei, and Y.-Q. Yang, "Research of Text Classification Based on Improved TF-IDF Algorithm." *2018 IEEE International Conference of Intelligent Robotic and Control Engineering (IRCE)*, pp. 218-222, Aug. 2018, doi: 10.1109/irce.2018.8492945.

[27] T. Chen and C. Guestrin, "XGBoost: A Scalable Tree Boosting System." *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pp. 785-794, Aug. 2016, doi: 10.1145/2939672.2939785.

[28] R. Santhanam, N. Uzir, S. Raman, and S. Banerjee, "Experimenting XGBoost Algorithm for Prediction and Classification of Different Datasets." *International*

Journal of Control Theory and Applications, vol. 9, no. 40, pp. 651-662, 2016. [En línea]. Disponible en: https://tinyurl.com/4ys7dypm. (Acceso: 5 de septiembre de 2022).

[29] Amazon Web Services, Inc. "How XGBoost Works - Amazon SageMaker." AWS. 2022. https://tinyurl.com/4mrffnxw (acceso: 5 de septiembre de 2022).

[30] A. Krenker, J. Bester, and A. Kos, "Introduction to the Artificial Neural Networks." *Artificial Neural Networks - Methodological Advances and Biomedical Applications*, pp. 1-18, Apr. 2011, doi: 10.5772/15751.

[31] Y. Tong and Z. Hong, "Hyper-Parameter Optimization: A Review of Algorithms and Applications." *Cornell University*, Mar. 12, 2020. [En línea]. Disponible en: https://doi.org/10.48550/arXiv.2003.05689. (Acceso: 3 de octubre de 2022).

[32] S. Minaee, N. Kalchbrenner, E. Cambria, N. Nikzad, M. Chenaghlu, and J. Gao, "Deep Learning-based Text Classification." *ACM Computing Surveys*, vol. 54, no. 3, pp. 1-40, Apr. 2021, doi: 10.1145/3439726.

[33] M. H. Sazli, "A brief review of feed-forward neural networks." *Communications, Faculty Of Science, University of Ankara*, vol. 50, no. 1, pp. 11-17, Jan. 2006, doi: 10.1501/0003168.

[34] R. Pascanu, T. Mikolov, and Y. Bengio, "On the difficulty of training recurrent neural networks." *Proceedings of the 30th International Conference on Machine Learning, PMLR*, vol. 28, no. 3, pp. 1310-1318, 2013. [En línea]. Disponible en: https://tinyurl.com/24c259h6. (Acceso: 4 de noviembre de 2022).

[35] Y. Yu, X. Si, C. Hu, and J. Zhang, "A Review of Recurrent Neural Networks: LSTM Cells and Network Architectures." *Neural Computation*, vol. 31, no. 7, pp. 1235-1270, 2019, doi: 10.1162/neco_a_01199.

[36] K. Peffers, T. Tuunanen and B. Niehaves, "Design science research genres: introduction to the special issue on exemplars and criteria for applicable design science research". *European Journal of Information Systems*, 27(2), 129-139, 2018. [En línea]. Disponible en: https://doi.org/10.1080/0960085X.2018.1458066. (Acceso: 26 de septiembre de 2022).

[37] K. Peffers et al., "Design Science Research Process: A Model for Producing and Presenting Information Systems Research." *Proceedings of the First International Conference on Design Science Research in Information Systems and Technology*, pp.

16-83, 2006. [En línea]. Disponible en: https://doi.org/10.48550/arXiv.2006.02763. (Acceso: 26 de septiembre de 2022).

[38] C. Schröer, F. Kruse, and J. M. Gómez, "A Systematic Literature Review on Applying CRISP-DM Process Model." *Procedia Computer Science*, vol. 181, no. 2021, pp. 526-534, 2021, doi: 10.1016/j.procs.2021.01.199.

[39] D. Rivadeneira. "Encuesta Nacional de Empleo Manual del Encuestador." INEC. Abr. 2022. [En línea]. Disponible en: https://tinyurl.com/mwuydmfe.

[40] S. Gonzalez-Carvajal and E. Garrido-Merchan, "Comparing BERT against traditional machine learning text classification." May 26, 2020. [En línea]. Disponible en: https://doi.org/10.48550/arXiv.2005.13012. (Acceso: 28 de septiembre de 2022).

[41] Y. Jung, J. Yoo, S.-H. Myaeng, and D.-C. Han, "A Web-Based Automated System for Industry and Occupation Coding." *Lecture Notes in Computer Science*, pp. 443-457, Sep. 2008, doi: 10.1007/978-3-540-85481-4_33.

[42] M. Schierholz, "Automating Survey Coding for Occupation." FDZ-Methodenreporte, Germany, Oct. 2014. [En línea]. Disponible en: https://doku.iab.de/fdz/reporte/2014/MR_10-14_EN.pdf. (Acceso: 24 de septiembre de 2022).

[43] H. Gweon, M. Schonlau, L. Kaczmirek, M. Blohm, and S. Steiner, "Three Methods for Occupation Coding Based on Statistical Learning." *Journal of Official Statistics*, vol. 33, no. 1, pp. 101-122, 2017, doi: 10.1515/jos-2017-0006.

[44] "HLG-MOS ML Project Pilot Study" Statistics Canada, Ontario, Canada, Apr. 2020. https://tinyurl.com/vk5dt5rp (acceso: 17 de septiembre de 2022).

[45] T. Anthopoulos. "Automated coding of Standard Industrial and Occupational Classifications (SIC/SOC)." Data Science Campus. Nov. 18, 2021. https://tinyurl.com/yx6dzftk (acceso: 24 de septiembre de 2022).

[46] J. Tanha, Y. Abdi, N. Samadi, N. Razzaghi, and M. Asadpour, "Boosting methods for multi-class imbalanced data classification: an experimental review." *Journal of Big Data*, vol. 7, no. 1, 2020, doi: 10.1186/s40537-020-00349-y.

[47] S. Raschka, "Model Evaluation, Model Selection, and Algorithm Selection in Machine Learning." Cornell University, Nov. 2018. doi: https://doi.org/10.48550/arXiv.1811.12808.

[48] L. Martinez. “Machine Learning Q&A: All About Model Validation.” MathWorks. https://tinyurl.com/583kzu7e (acceso: 4 de julio de 2022).

[49] C. Seger, “An investigation of categorical variable encoding techniques in machine learning: binary versus one-hot and feature hashing.” Stockholm, Sweden, 2018. [En línea]. Disponible en: https://tinyurl.com/4czj5wzm.

[50] J. Brownlee. “How to Use StandardScaler and MinMaxScaler Transforms in Python.”, Machine Learning Mastery. Jun. 10, 2020. https://tinyurl.com/mrd5spcc (acceso: 4 de octubre de 2022).

[51] F. Pedregosa. “sklearn.preprocessing.LabelEncoder — scikit-learn 1.1.3.”, scitkit learn. 2022. https://tinyurl.com/bdhf77ta (acceso: 4 de julio de 2022).

[52] A. Chand. “Difference between fit() , transform() and fit_transform() method in Scikit-learn.”, Nerd For Tech. Jun. 02, 2021. https://tinyurl.com/5n74mkh4 (acceso: 4 de julio de 2022).

[53] J. Joloudari, A. Marefat, M. Nematollahi, S. Oyelere, and S. Hussain, “Effective Class-Imbalance learning based on SMOTE and Convolutional Neural Networks.” *Cornell University*, Sep. 01, 2022. [En línea]. Disponible en: https://doi.org/10.48550/arXiv.2209.00653. (Acceso: 3 de octubre de 2022).

[54] A. Shahul. “Hyperparameter Tuning in Python: a Complete Guide - neptune.ai.”, MLOps Blog. Nov. 14, 2022. https://neptune.ai/blog/hyperparameter-tuning-in-python-complete-guide (acceso: 4 de junio de 2022).

[55] Xgboost developers. “XGBoost Parameters — xgboost 1.7.1 documentation.” dmlc XGBoost. 2022. https://xgboost.readthedocs.io/en/stable/parameter.html (acceso: 4 de junio de 2022).

[56] O. E. Ørebæk and M. Geitle. “Exploring the Hyperparameters of XGBoost Through 3D Visualizations.” Mar. 2021. [En línea]. Disponible en: https://ceur-ws.org/Vol-2846/paper22.pdf. (Acceso: 5 de mayo de 2022).

[57] M. Bloice and A. Holzinger, “A Tutorial on Machine Learning and Data Science Tools with Python.” *Lecture Notes in Computer Science*, pp. 435-480, 2016, doi: 10.1007/978-3-319-50478-0_22.

[58] B. Sun, T. Sun, and P. Jiao, “Spatio-temporal segmented traffic flow prediction with ANPRS data based on improved XGBoost,” *Journal of Advanced Transportation,* vol. 2021, pp. 1–24, 2021, doi: 10.1155/2021/5559562.

[59] A. Gulli, S. Pal, and A. Kapoor, *Deep Learning with TensorFlow 2 and Keras* - Second Edition. Packt Publishing Ltd., 2019. [En línea]. Disponible en: https://tinyurl.com/bdcpxafj.

[60] F. Bogale Gereme and W. Zhu, "Fighting Fake News Using Deep Learning." *2020 The 3rd International Conference on Computational Intelligence and Intelligent Systems*, pp. 24-29, 2020, doi: 10.1145/3440840.3440847.

[61] "Keras documentation: Embedding layer", Keras.io. https://keras.io/api/layers/core_layers/embedding/ (acceso: 20 de noviembre de 2022).

[62] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A Simple Way to Prevent Neural Networks from Overfitting," *Journal of Machine Learning Research*, vol. 15, no. 2014, pp. 1929–1958, Jun. 2014. [En línea]. Disponible en: https://tinyurl.com/yvnjce9x. (Acceso: 20 de noviembre de 2022).

[63] J. Feng and S. Lu, "Performance Analysis of Various Activation Functions in Artificial Neural Networks," *Journal of Physics: Conference Series*, vol. 1237, p. 022030, Jun. 2019, doi: 10.1088/1742-6596/1237/2/022030.

[64] "LSTM: Understanding the number of parameters", Kaggle.com. https://tinyurl.com/484tktd2 (acceso: 20 de noviembre de 2022).

[65] "Keras documentation: LSTM layer", Keras.io. https://keras.io/api/layers/recurrent_layers/lstm/ (acceso: 20 de noviembre de 2022).

[66] Brownlee, J. "Difference Between a Batch and an Epoch in a Neural Network," *Machine Learning Mastery*. Deep Learning, August 10, 2022. https://tinyurl.com/3shb4cd2 (acceso: 20 de noviembre de 2022).

[67] "Keras documentation: Adam", Keras.io. https://keras.io/api/optimizers/adam/ (acceso: 20 de noviembre de 2022).

[68] D. Ghimire, "Comparative study on python web frameworks: Flask and Django", Bachelor of Engineering, Software Engineering, Metropolia University of Applied Sciences, 2020. [En línea]. Disponible en: https://urn.fi/URN:NBN:fi:amk-2020052513398.

[69] P. Kennedy, "Deep dive into Flask's application and request contexts*"*, Testdriven.io. https://testdriven.io/blog/flask-contexts-advanced/ (acceso: 5 de diciembre de 2022).

ANEXOS

Anexo I – Estructura esquemática de la CIIU 4.0 por Secciones.

SECCIÓN	DESCRIPCIÓN
A	Agricultura, ganadería, silvicultura y pesca
B	Explotación de minas y canteras
C	Industrias manufactureras
D	Suministro de electricidad, gas, vapor y aire acondicionado
E	Suministro de agua; evacuación de aguas residuales, gestión de desechos y descontaminación
F	Construcción
G	Comercio al por mayor y al por menor; reparación de vehículos automotores y motocicletas
H	Transporte y almacenamiento
I	Actividades de alojamiento y de servicio de comidas
J	Información y comunicaciones
K	Actividades financieras y de seguros
L	Actividades inmobiliarias
M	Actividades profesionales, científicas y técnicas
N	Actividades de servicios administrativos y de apoyo
O	Administración pública y defensa; planes de seguridad social de afiliación obligatoria
P	Enseñanza
Q	Actividades de atención de la salud humana y de asistencia social
R	Actividades artísticas, de entretenimiento y recreativas
S	Otras actividades de servicios
T	Actividades de los hogares como empleadores; actividades no diferenciadas de los hogares como productores de bienes y servicios para uso propio
U	Actividades de organizaciones y órganos extraterritoriales

Anexo II – Estructura esquemática de la CIUO Rev. 08 por Grandes Grupos.

GRAN GRUPO	DESCRIPCIÓN
0	Ocupaciones Militares
1	Directores y Gerentes
2	Profesionales, Científicos e Intelectuales
3	Técnicos y Profesionales de Nivel Medio
4	Personal de Apoyo Administrativo
5	Trabajadores de los Servicios y Vendedores de Comercios y Mercados
6	Agricultores y Trabajadores Calificados Agropecuarios, Forestales y Pesqueros
7	Oficiales, Operarios y Artesanos de Artes Mecánicas y de otros Oficios
8	Operadores de Instalaciones y Máquinas y Ensambladores
9	Ocupaciones Elementales

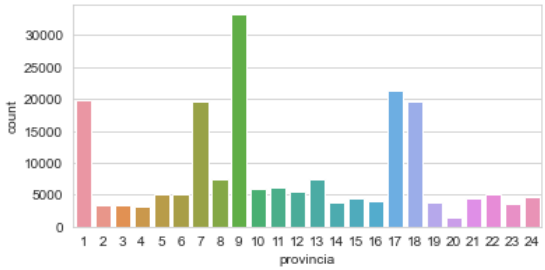
Anexo III – Representación gráfica de las variables textuales de la ENEMDU usadas para la construcción de los modelos de clasificación.

Figura obtenida del conjunto de datos del modelo de actividad económica. Descripción de la actividad económica (drama)

Figura obtenida del conjunto de datos del modelo de ocupaciones. Descripción de la ocupación (dgrupo)

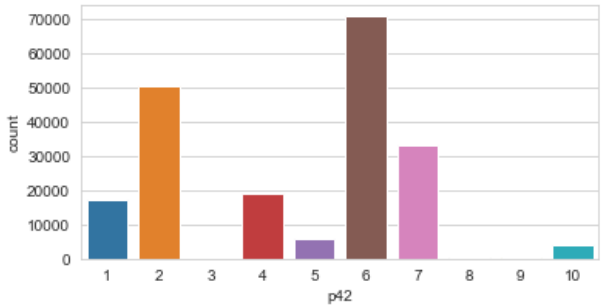
Anexo IV – Representación gráfica de las variables categóricas de la ENEMDU usadas para la construcción de los modelos de clasificación.

Provincia (provincia)



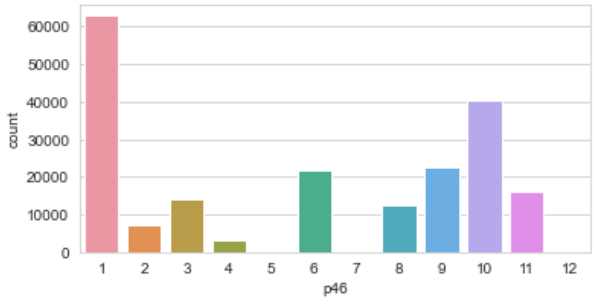
PROVINCIA			
Azuay	1	Manabí	13
Bolívar	2	Morona Santiago	14
Cañar	3	Napo	15
Carchi	4	Pastaza	16
Cotopaxi	5	Pichincha	17
Chimborazo	6	Tungurahua	18
El Oro	7	Zamora Chinchipe	19
Esmeraldas	8	Galápagos	20
Guayas	9	Sucumbíos	21
Imbabura	10	Orellana	22
Loja	11	Santo domingo de los Tsáchilas	23
Los Ríos	12	Santa Elena	24

Categoría Ocupacional (p42)



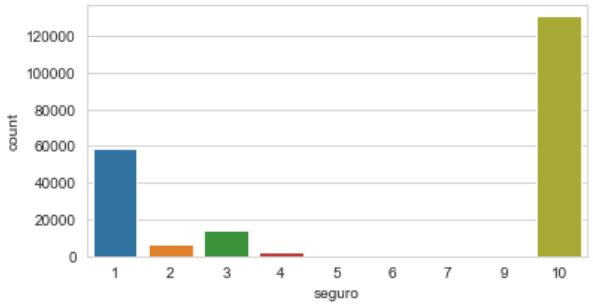
CATEGORIA OCUPACIONAL	
Empleado/Obrero de Gobierno/Estado	1
Empleado/Obrero Privado	2
Empleado/Obrero Tercerizado	3
Jornalero o Peón	4
Patrono	5
Cuenta Propia	6
Trabajador del Hogar no Remunerado	7
Trabajador no Remunerado de otro Hogar	8
Ayudante no Remunerado de Asalariado/ Jornalero	9
Empleado(a) Doméstico(a)	10

Sitio de trabajo (p46)



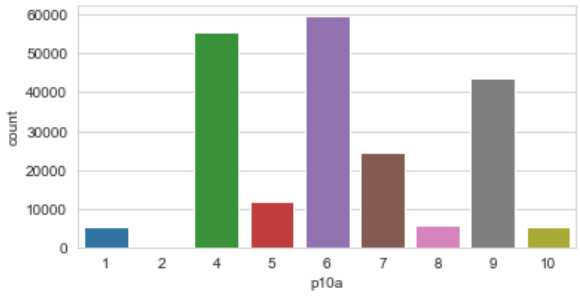
SITIO DE TRABAJO	
Local de una empresa o del patrono	1
Una obra en construcción	2
Se desplaza	3
Al descubierto en la calle	4
Kiosco en la calle	5
Local propio o arrendado	6
Local de cooperativa o asociación	7
Vivienda distinta a la suya	8
Su vivienda	9
Su finca o terreno	10
Finca o terreno ajeno	11
Finca, terreno o establecimiento comunal	12

Seguro (p05a)



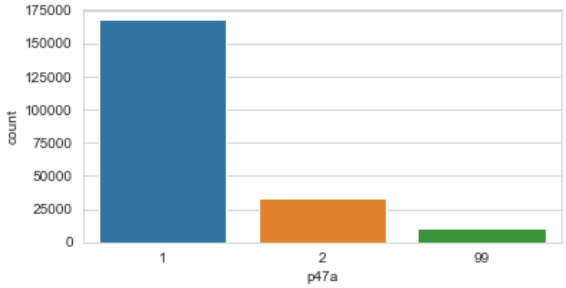
SEGURO	
IESS, Seguro General	1
IESS, Seguro Voluntario	2
Seguro campesino	3
Seguro del ISSFA o ISSPOL	4
Seguro de salud privado con hospitalización	5
Seguro de salud privado sin hospitalización	6
AUS	7
Seguros municipales y de consejos provinciales	8
Seguro M. S. P	9
Ninguno	10

Nivel de Instrucción (p10a)



NIVEL DE INSTRUCCIÓN	
Ninguno	1
Centro de Alfabetización	2
Jardín de Infantes	3
Primaria	4
Educación Básica	5
Secundaria	6
Educación Media/ Bachillerato	7
Superior no Universitario	8
Superior Universitario	9
Postgrado	10

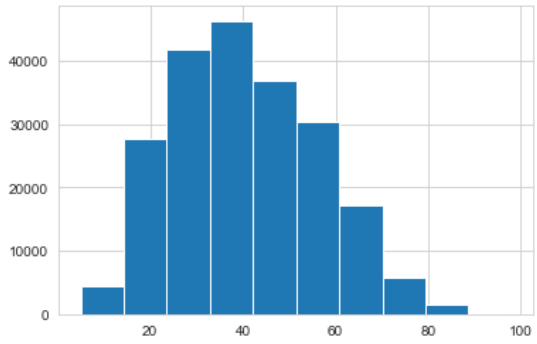
Tamaño del establecimiento (p47a)



TAMAÑO DEL ESTABLECIMIENTO	
Menos de 100	1
100 y más	2
No sabe/No responde	99

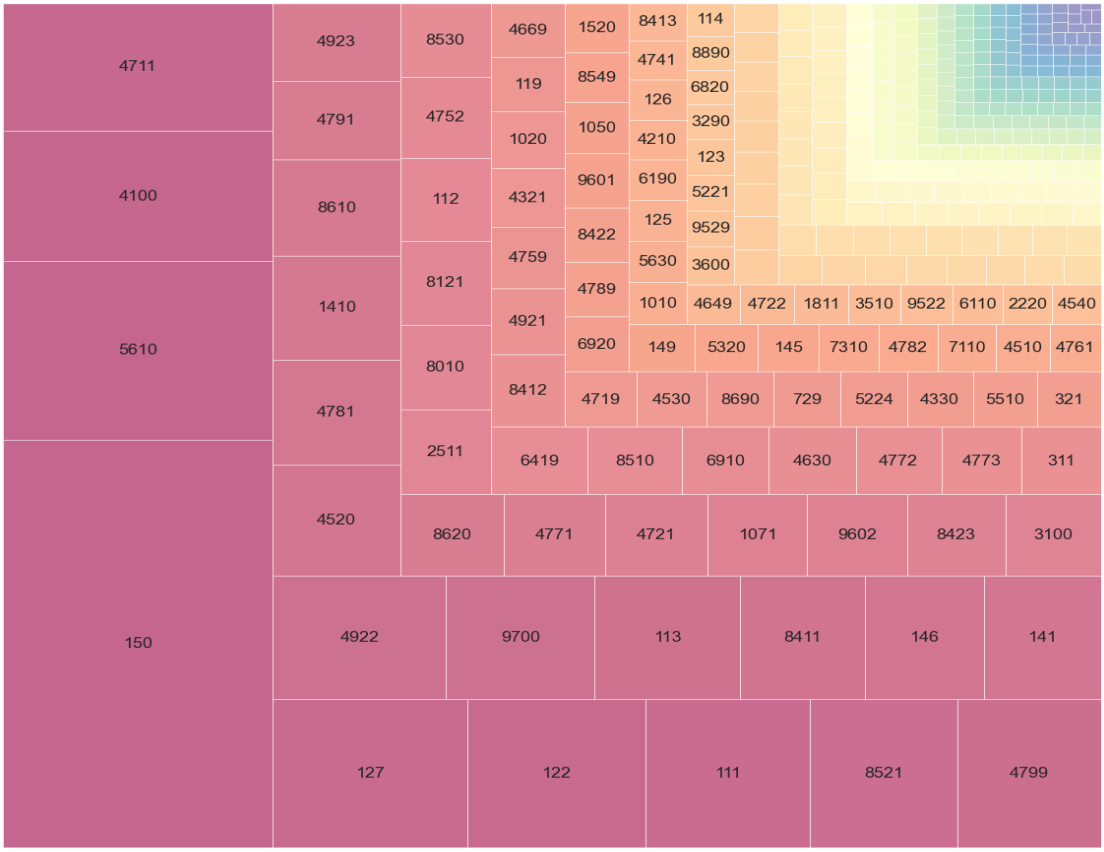
Anexo V – Histograma de la variable edad usada para la construcción del modelo de clasificación de ocupaciones.

Edad (p03)



Anexo VI – Representación gráfica de las variables objetivo de la ENEMDU usadas para la construcción de los modelos de clasificación.

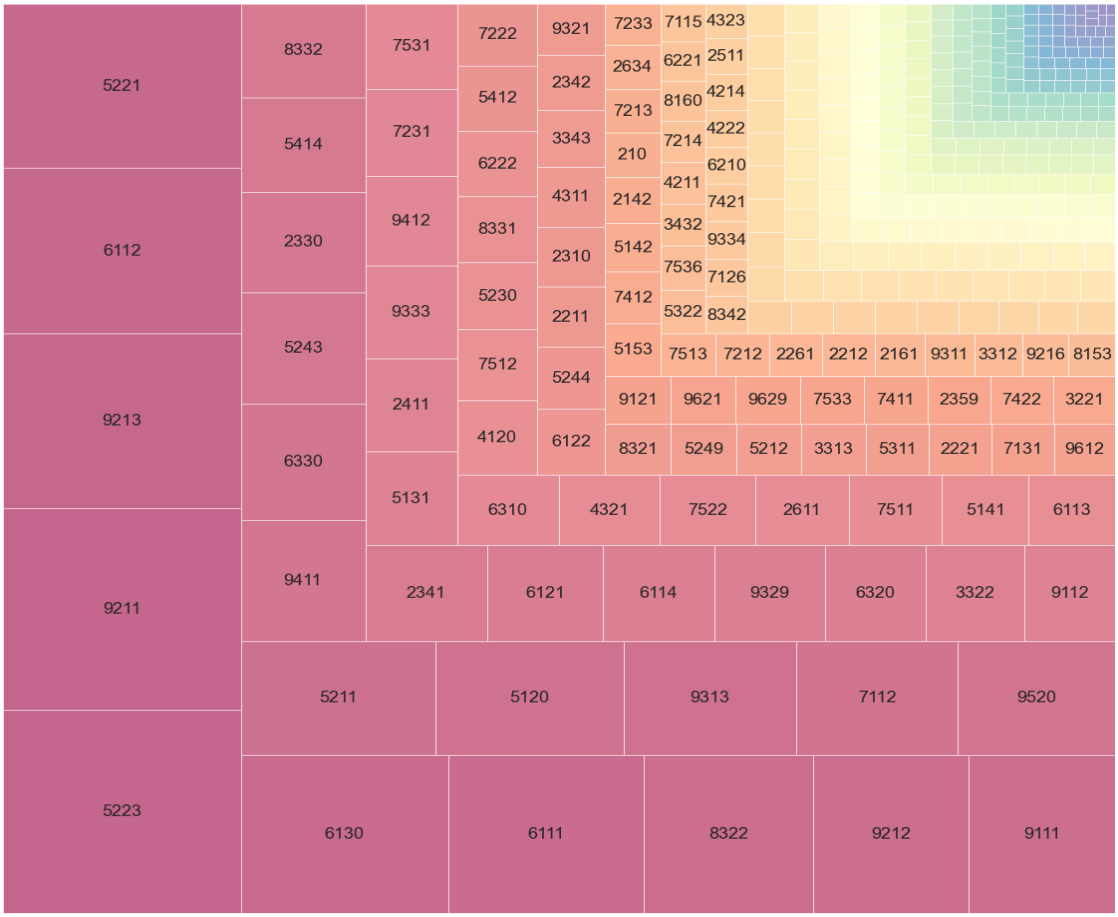
Código CIIU de Actividad Económica (p40)



Top 5 de las clases mayoritarias de actividades económicas

CÓDIGO DE ACTIVIDAD ECONÓMICA	DESCRIPCIÓN CIIU	NÚMERO DE CASOS
150	Cultivo de productos agrícolas en combinación con la cría de animales	23889
5610	Actividades de restaurantes y de servicio móvil de comidas	10448
4100	Construcción de edificios	7638
4711	Venta al por menor en comercios no especializados con predominio de la venta de alimentos, bebidas o tabacos	7476
127	Cultivo de plantas con las que se preparan bebidas	6253

Código CIUO de Ocupación (p41)

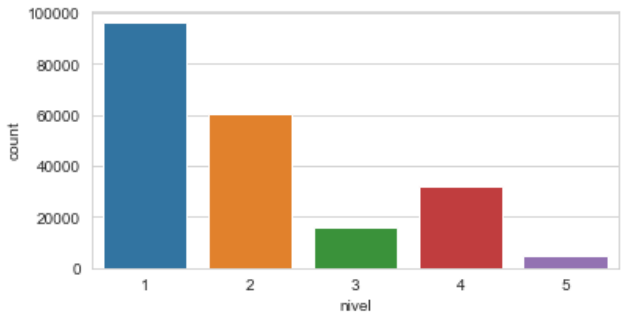


Top 5 de las clases mayoritarias de ocupaciones

CÓDIGO DE OCUPACIÓN	DESCRIPCIÓN CIUO	NÚMERO DE CASOS
5223	Asistentes de venta de tiendas y almacenes	9968
9211	Peones de explotaciones agrícolas	9906
9213	Peones de explotación de cultivos mixtos y ganaderos	8619
6112	Agricultores y trabajadores calificados de plantaciones de árboles y arbustos	8112
5221	Comerciantes de tiendas	8086

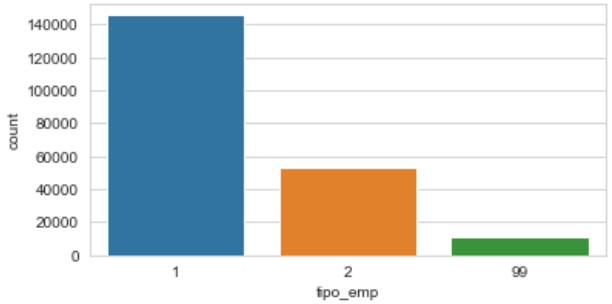
Anexo VII – Representación gráfica de las variables construidas para los modelos de clasificación.

Nivel de estudios (nivel)



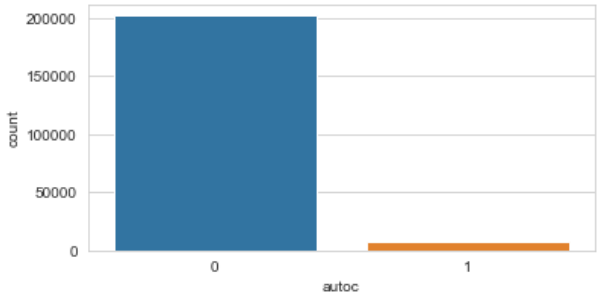
NIVEL DE ESTUDIOS	
Secundaria incompleta	1
Secundaria	2
Técnico o Tecnología	3
Superior Universitario	4
Posgrado	5

Tipo de empresa (tipo_emp)



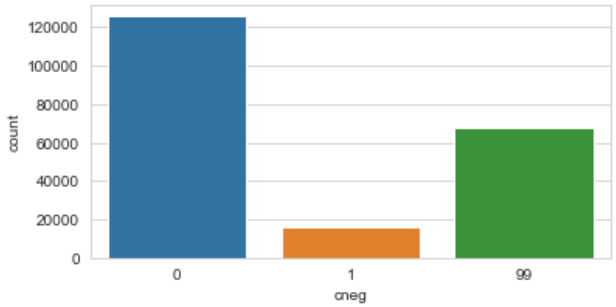
TIPO DE EMPRESA	
Empresa hasta 5 empleados	1
Empresa más de 5 empleados	2
No sabe/No responde	99

Autoconsumo (autoc)



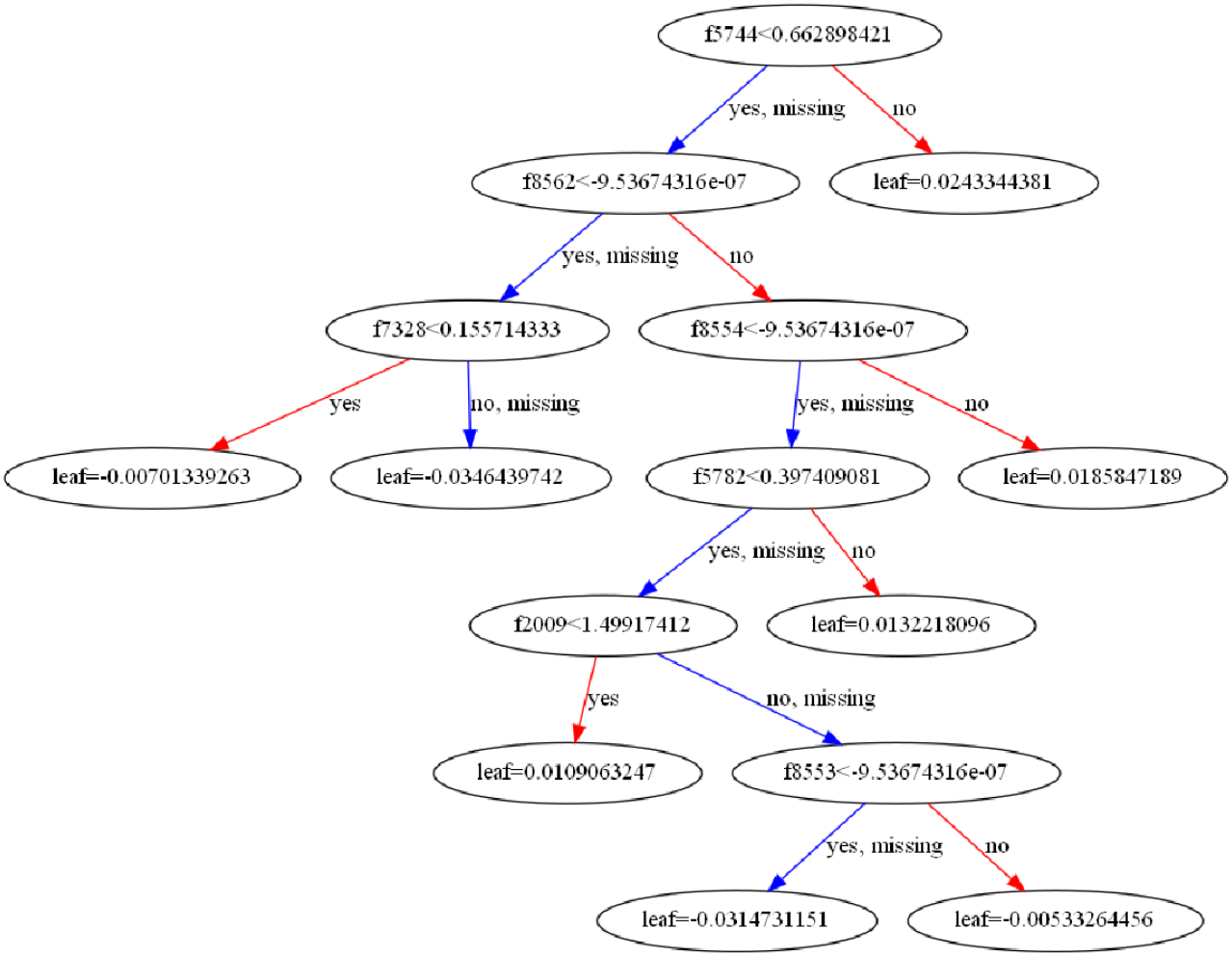
AUTOCONSUMO	
Producción para venta	0
Producción para autoconsumo	1

Tenencia de cuarto exclusivo (cneg)

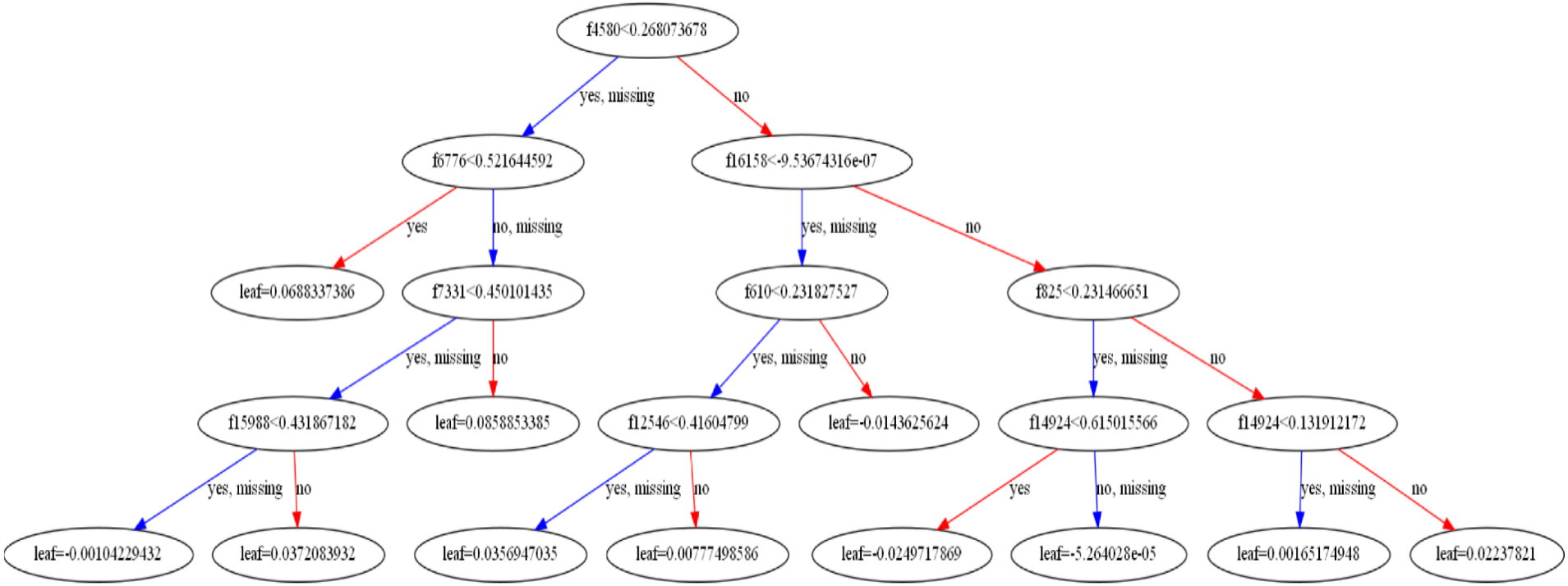


TENENCIA CUARTO EXCLUSIVO	
No tiene cuarto exclusivo	1
Tiene cuarto exclusivo	2
No sabe/No responde	99

Anexo VIII – Árbol de decisión 32100 de Xgboost para el modelo de actividad económica.



Anexo IX – Árbol de decisión 34400 de Xgboost para el modelo de ocupaciones.



Anexo X – Tabla de resultados del entrenamiento del algoritmo Xgboost para el modelo de actividades económicas.

N	balanceo de clases	tasa aprendizaje	profundidad del árbol	número de iteraciones	exactitud entrenamiento %	exactitud validación %	macro - promedio		
							precisión %	sensibilidad %	puntuación-F1 %
1	No	0.3	6	100	57.28	56.59	6.57	10.28	7.58
2	No	0.2	6	100	96.10	90.29	83.02	78.71	79.96
3	No	0.1	6	100	96.27	91.92	85.48	80.21	**81.59**
4	No	0.08	6	100	95.84	91.71	84.92	79.54	81.12
5	No	0.06	6	100	95.01	91.41	84.76	78.25	80.17
6*	No	0.1	10	100	97.86	92.31	85.51	80.52	81.83
7	No	0.1	20	100	98.75	92.21	84.82	80.48	81.45
8*	No	0.1	10	200	98.63	92.67	86.12	81.45	82.64
9	5000/ clase	0.1	6	100	95.27	90.99	82.82	82.40	81.56

* Evaluando la exactitud de entrenamiento y validación se determinó que el modelo tiene sobreajuste.

Anexo XI – Tabla de resultados del entrenamiento del algoritmo Xgboost para el modelo de ocupaciones.

N	balanceo de clases	tasa aprendizaje	profundidad del árbol	número de iteraciones	exactitud entrenamiento %	exactitud validación %	macro - promedio		
							precisión %	sensibilidad %	puntuación-F1 %
1	No	0.3	6	100	94.42	79.50	56.17	50.49	52.04
2*	No	0.1	6	100	94.83	83.58	60.93	54.42	56.25
3*	No	0.08	6	100	94.85	84.61	60.60	54.74	56.30
4*	No	0.06	6	100	93.62	84.56	60.83	54.88	56.46
5	No	0.04	6	100	91.42	84.21	60.11	54.39	55.93
6	No	0.01	6	100	85.38	81.98	55.44	48.29	49.78
7	No	0.01	10	100	87.83	83.35	55.42	48.92	50.37
8	No	0.01	10	200	91.69	84.49	58.34	52.14	53.53
9	No	0.04	4	100	87.53	82.84	60.72	53.76	55.42
10	No	0.04	6	100	89.28	83.87	59.47	52.71	54.16
11	5000/ clase	0.04	4	100	85.36	81.01	57.13	57.51	**56.13**

El modelo 10 tiene regularización alpha=1 lambda =2

* Evaluando la exactitud de entrenamiento y validación se determinó que el modelo tiene sobreajuste.

Anexo XII – Tabla de resultados del entrenamiento de las Redes Neuronales Feedforward para el modelo de actividades económicas.

Dimensión de incrustaciones = 50

N	tasa aprendizaje	épocas	dimensión lote	número de neuronas capa oculta	abandono (dropout)	exactitud entrenamiento %	exactitud validación %	macro - promedio		
								precisión %	sensibilidad %	puntuación-F1 %
1	0.001	50	1000	400	0.3 - Dense	98.51	94.65	87.30	84.96	85.12
2	0.001	50	500	400	0.3 - Dense	98.27	94.67	87.55	84.67	84.97
3	0.001	50	2000	400	0.3 - Dense	98.41	94.49	87.34	84.92	85.26
4	0.01	50	1000	400	0.3 - Dense	98.50	94.70	87.98	85.42	85.75
5	0.01	50	500	400	0.3 - Dense	98.75	94.75	87.09	85.16	85.01
6	0.01	50	2000	400	0.3 - Dense	98.40	94.90	88.39	86.44	**86.60**
7	0.0001	50	1000	400	0.3 - Dense	98.44	94.62	86.66	85.47	85.25
8	0.0001	50	500	400	0.3 - Dense	98.48	94.67	87.66	85.3	85.64
9	0.0001	50	2000	400	0.3 - Dense	98.46	94.66	87.75	85.65	85.91
10*	0.01	50	2000	400	0.3 - Dense	89.86	85.55	64.27	76.66	67.82

*Modelo con datos balanceados

Anexo XIII – Tabla de resultados del entrenamiento de las Redes Neuronales LSTM para el modelo de actividades económicas.

Dimensión incrustaciones = 50

N	tasa aprendizaje	épocas	dimensión lote	número de unidades LSTM	abandono (dropout)	exactitud entrenamiento %	exactitud validación %	macro - promedio		
								precisión %	sensibilidad %	puntuación-F1 %
1	0.01	50	2000	400	LSTM - 0.3	95.30	90.97	74.30	69.78	70.46
2	0.01	50	500	100	No	98.05	92.53	78.49	76.98	76.72
3	0.01	50	500	800	LSTM - 0.3	96.27	92.02	76.09	74.37	73.70
4	0.1	70	500	100	LSTM - 0.3	97.06	92.19	76.40	74.29	73.82
5	0.001	50	1000	400	LSTM - 0.3	97.01	92.50	79.33	77.67	77.16
6	0.001	50	1000	500	LSTM - 0.3	97.12	93.20	80.62	80.74	**79.40**
7*	0.001	50	1000	500	LSTM - 0.3	89.69	83.77	58.15	70.33	61.65

Al modelo 3 se añadió una capa oculta (Dense 4000) luego de concatenar todas las variables de entrada.

*Modelo con datos balanceados

Anexo XIV – Tabla de resultados del entrenamiento de las Redes Neuronales Feedforward para el modelo de ocupaciones.

Épocas =50

N	tasa aprendizaje	dimensión incrustación	dimensión lote	número de neuronas capa oculta	abandono (dropout)	exactitud entrenamiento %	exactitud validación %	macro - promedio		
								precisión %	sensibilidad %	puntuación-F1 %
1	0.0001	50	2000	800	0.3 - Dense	94.2	87.13	64.51	58.83	59.8
2	0.0001	50	1000	800	0.3 - Dense	93.93	87.1	64.64	59.41	60.2
3	0.0001	50	1000	1000	0.3 - Dense	94.57	87.35	64.67	60.21	60.77
4	0.0001	50	1000	1000	0.3 - Dense - 0.3 - Softmax	94.23	87.34	65.98	60.39	61.29
5	0.001	50	1000	800	0.3 - Dense	94.91	87.11	65.89	60.41	61.43
6	0.001	50	2000	800	0.3 - Dense	93.96	86.84	64.95	59.95	60.94
7	0.001	50	1000	1000	0.3 - Dense - 0.3 - Softmax	93.97	87.43	66.15	60.42	61.49
8	0.01	50	1000	1000	0.3 - Dense - 0.3 - Softmax	94.19	87.44	66.33	60.5	**61.81**
9	0.01	50	1000	1000	0.5 - Dense - 0.5 - Softmax	93.5	86.86	65.2	60.99	61.52
10	0.01	25	1000	1000	0.3 - Dense - 0.3 - Softmax	93.92	87.27	65.53	59.83	60.9
11	0.1	50	1000	1000	0.3 - Dense - 0.3 - Softmax	94.33	87.39	65.09	59.36	60.31
12*	0.01	50	1000	1000	0.3 - Dense - 0.3 - Softmax	82.16	76.84	44.73	49.71	45.09

*Modelo con datos balanceados

Anexo XV – Tabla de resultados del entrenamiento de las Redes Neuronales LSTM para el modelo de ocupaciones.

Épocas =50

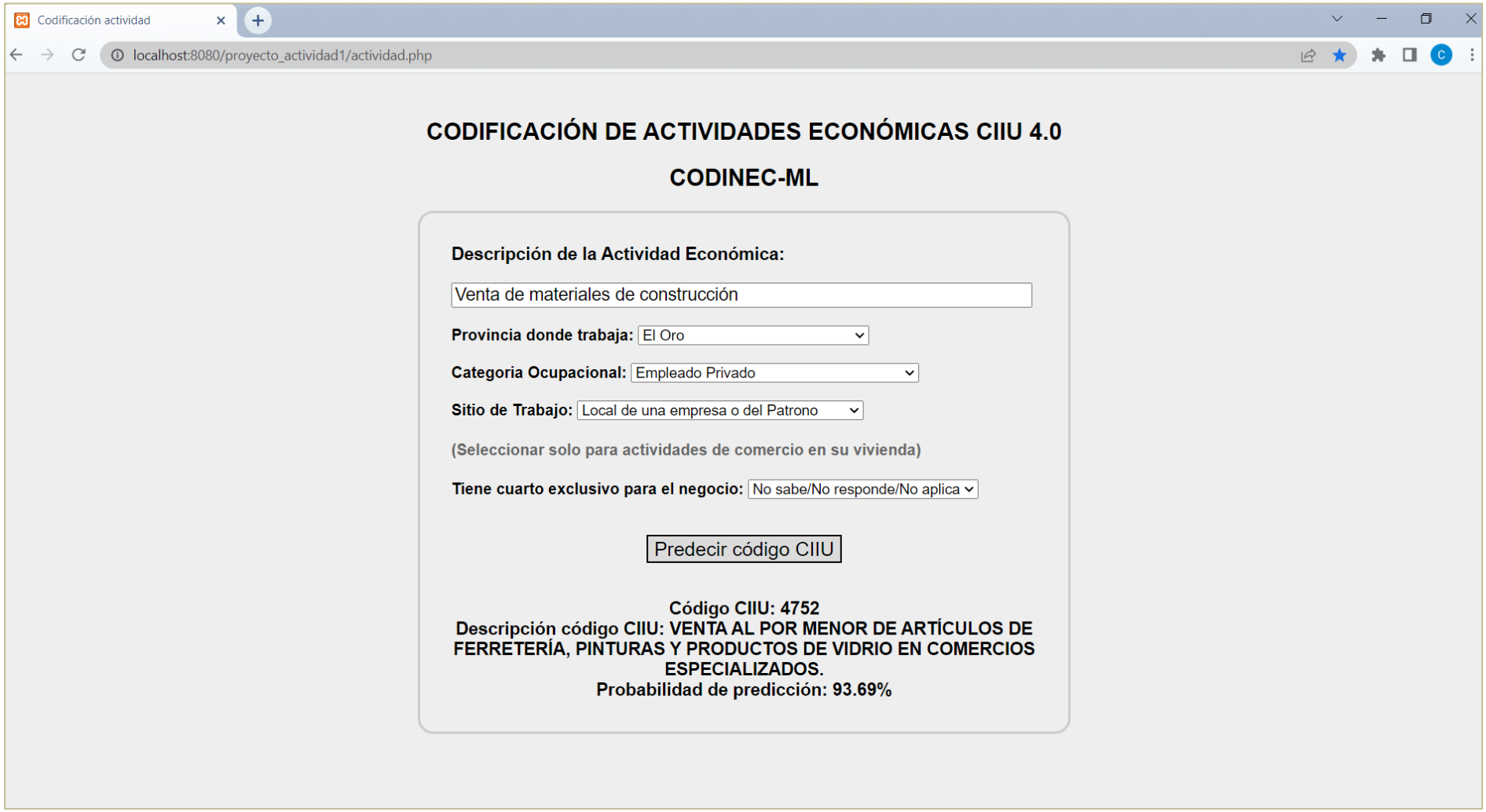
N	tasa aprendizaje	dimensión incrustación	dimensión lote	número de unidades LSTM	abandono (dropout)	exactitud entrenamiento %	exactitud validación %	macro - promedio		
								precisión %	sensibilidad %	puntuación-F1 %
1	0.01	50	1000	1000 y 800	(LSTM - 0.3) y (LSTM - 0.3)	93.05	82.78	52.48	49.72	49.88
2*	0.01	50	1000	1000 y 800	0.3 - Dense - 0.3 - Softmax	94.66	84.33	56.11	52.66	53.07
3	0.01	50	1000	100 y 100	(LSTM - 0.3) y (LSTM - 0.3)	88.23	80.31	46.64	43.98	44.28
4	0.01	10	1000	200 y 200	(LSTM - 0.3) y (LSTM - 0.3)	91.45	82.96	51.64	47.93	48.74
5	0.01	10	1000	200 y 200	(LSTM - 0.5) y (LSTM - 0.5)	90.13	83.16	53.66	50.29	**50.66**
6	0.01	10	1000	200 y 200	concatenate - 0.3	87.29	80.58	48.43	45.8	46.02
7**	0.01	10	1000	200 y 200	(LSTM - 0.5) y (LSTM - 0.5)	90.14	81.76	51.29	47.89	48.19

En el modelo 2 se añadió una capa oculta Dense 1000 luego de concatenar todas las entradas.

*Evaluando la exactitud de entrenamiento y validación se determinó que el modelo tiene sobreajuste.

**Modelo con datos balanceados

Anexo XVI – Interfaz gráfica del aplicativo web para clasificar las actividades económicas.



Anexo XVII – Interfaz gráfica del aplicativo web para clasificar las ocupaciones.