UNIVERSIDAD NACIONAL DE MOQUEGUA

INGENIERÍA DE SISTEMAS E INFORMÁTICA



PREDICCIÓN DE CONSUMO IRREGULAR DEL SERVICIO DE AGUA POTABLE BASADO EN MÉTODOS DE MACHINE LEARNING EN LA PROVINCIA DE ILO

TESIS

Presentado por:

Alonso Moron Coaquira

Para optar:

Título Profesional de Ingeniero de Sistemas e Informática

Ilo, Febrero del 2021

UNAM
UNIVERSIDAD NACIONAL DE MOQUEGUA

ESCUELA PROFESIONAL DE INGENIERÍA DE SISTEMAS E INFORMÁTICA

Folio **Nº 004**

ACTA DE SUSTENTACIÓN

I. Modalidad:

Tesis : [X]
Trabajo de Suficiencia Profesional : [ ]
Otro: ..... : [ ]

II. Documento de aprobación para la sustentación:

Resolución Comisión Organizadora Nº 684-2021-UNAM

Fecha: 15 de Julio 2021

III. El jurado ha dictaminado la (el) tesis, TITULADO:

"Predición de consumo irregular del servicio de agua potable basado en métodos de machine learning".

Bachiller: Alonso Moron Coaquira

Asesor Principal: M.Sc. Hugo Euler tito Chura

Co-Asesor: .....

IV. Se dictamina:

Como: Aprobado por Unanimidad la sustentación de manera:

Presencial [ ] Virtual [X]

Con nota: Catorce (14) por lo que; Sí [X] No [ ]

Opta el Título Profesional de INGENIERO DE SISTEMAS E INFORMÁTICA .

Siendo las 19:30 horas, del 26 de Julio del año 2021 se procede con la firma en conformidad de lo dictaminado por el Jurado.

Anibal F. Flores Garcia
Presidente

Mgr. Juan C. Clares P.
Primer Miembro

MSc. Mario Gauna Ch.
Segundo Miembro

Alonso Moron Coaquira
Bachiller

Índice general

I. Introducción **10**

1.1. Descripción y formulación del problema 11

1.2. Antecedentes 13

1.3. Objetivos 15

1.3.1. Objetivo general 15

1.3.2. Objetivos específicos 15

1.4. Justificación e importancia 16

1.5. Hipótesis 16

1.5.1. Hipótesis general 16

1.5.1.1. Hipótesis específicas 16

1.5.1.2. Variables 17

II. Marco teórico **18**

2.1. Bases teóricas 18

2.1.1. Inteligencia artificial 18

2.1.1.1. Machine Learning 19

2.1.1.2. Ciencia de datos 20

2.1.2. Metodología 21

2.1.2.1. Comprensión de negocio 22
2.1.2.2. Comprensión de datos 23
2.1.2.3. Preparación de datos 23
2.1.2.4. Modelado 23
2.1.2.5. Evaluación 24
2.1.3. Algoritmos propuestos 24
2.1.3.1. Árbol de decisiones 24
2.1.3.2. Random forest 26
2.1.3.3. Baseline model 27
2.2. Empresa Prestadora de Servicios de Saneamiento Ilo S.A. 29
2.2.1. Descripción de la empresa 29
2.2.2. Organización 29
2.2.3. Principales oficinas encargadas 32
2.3. Definición de términos 33

III.Método **35**

3.1. Tipo y nivel de investigación 35
3.1.1. Tipo de investigación 35
3.1.2. Nivel de investigación 35
3.2. Ámbito temporal y espacial 36
3.3. Población y muestra 36
3.3.1. Población 36
3.3.2. Muestra 36
3.4. Instrumentos 36

3.5. Procedimientos 37

3.6. Análisis de datos 37

IV. Resultados **39**

4.1. Comprensión de negocio 39

4.2. Comprensión de datos 40

4.3. Preparación de datos 45

4.4. Modelado 48

4.4.1. Árbol de decisiones 48

4.4.2. Red neuronal simple 52

4.4.3. Random forest 55

4.5. Evaluación 58

V. Conclusiones **63**

VI. Recomendaciones **64**

Referencias **65**

VIIAnexos **68**

Índice de figuras

II.1. *Relación entre Inteligencia Artificial, Machine Learning y Data Science.* 21

II.2. *Etapas de la metodología CRISP-DM* 22

II.3. *El árbol en formato Graphviz usando el export_graphviz exportador.* 27

II.4. *Modelo de red neuronal de una sola capa oculta.* 28

II.5. *Organigrama de la EPS Ilo S.A.* 30

IV.1. Conjunto de datos de entrenamiento (dataset) 45

IV.2. *Número de registros por clase* 46

IV.3. *Número de registros por clase sin aplicar SMOTE.* 47

IV.4. *Número de registros por clase aplicando SMOTE.* 48

IV.5. *Diagrama del algoritmo Árbol de decisión.* 49

IV.6. *Código del algoritmo de Árbol de decisión en Python.* 50

IV.7. *Matriz de confusión del Árbol de decisión con datos desbalanceados.* 51

IV.8. *Número de registros por clase aplicando SMOTE.* 52

IV.9. *Preparación del dataset para Red Neuronal Simple* 53

IV.10.*Selección de variables de entrada(X) y variables de salida(Y)* 53

IV.11.*Modelo de Red Neuronal Simple* 54

IV.12.*Resultado de predicción de con Red Neuronal Simple* 55

IV.13.*Preparación de dataset para Random Forest* 56

IV.14.*Código de algoritmo Random Rorest en Python* 57

IV.15.*Matriz de confusión de resultados con Random Forest* 58

IV.16.*Ejemplo de Matriz de Confusión* 59

IV.17.*Matriz de confusión de resultado del algoritmo de Árbol de Decisiones* 60

IV.18.*Resultado de predicción de Red Neuronal Simple* 61

IV.19.*Precisión de predicción de clases con Random Forest* 61

Índice de cuadros

IV.1. *Descripción de variables del dataset de la empresa EPS Ilo S.A.* . . . 41

IV.2. *Codificación de la atributos de estados de lectura.* 43

IV.3. *Análisis de los resultados* 62

Resumen

El presente trabajo de investigación tiene el enfoque de experimentación de algoritmos de Machine Learning en el problema de predicción de clases de consumo irregular de agua potable de la Empresa Prestadora de Servicios de Saneamiento, EPS Ilo S.A. La empresa cuenta con el registro histórico de los incidentes o casos de consumo irregular de sus usuarios, los datos registrados son variables que conducen a identificar a que clase o tipo de consumo irregular pertenece. Esta información permitió experimentar diferentes algoritmos de Machine Learning, y finalmente, definir cual de ellos se comporta mejor en este caso especifico.

El algoritmo más eficiente en resolver el problema es Random Forest con 98 % de precisión en predicciones de clases, seguido por Árbol de decisiones con 82 % de precisión y finalmente Red Neuronal Simple con el 70.83 %.

Abstract

This research work focuses on the experimentation of Machine Learning algorithms in the problem of predicting classes of irregular drinking water consumption of the Empresa Prestadora de Servicios de Saneamiento, EPS Ilo S.A. The company has a historical record of incidents or cases of irregular consumption by its users, the data recorded are variables that lead to identifying what kind of irregular consumption it belongs to. This information allowed experimenting with different Machine Learning algorithms, and finally, defining which of them behaves better in this specific case.

The most efficient algorithm to solve the problem is Random Forest with 98 % precision, followed by Decision Tree with 82 % precision and finally, Simple Neural Network with 70.83 % precision.

Dedicatoria

Esta tesis está dedicada:

Principalmente a Dios, por haberme permitido llegar hasta este momento tan importante en mi formación profesional. A mis padres, que a pesar de nuestras diferencias de opiniones, siempre me brindaron su apoyo, paciencia y comprensión. También a todos mis amigos, compañeros de trabajo y todos aquellos quienes me ofrecieron sus buenos consejos y sus experiencias.

Capítulo I

Introducción

En los últimos tiempos las diversas disciplinas de inteligencia artificial han solucionado diferentes problemas de tratamiento de la información. En la actualidad, aún existen muchos problemas sin resolver en la sociedad, tanto en las empresas del sector público como en el privado, sin embargo, bajo este panorama, el presente trabajo de investigación aborda el problema de predicción de consumo irregular de agua potable que administra la Empresa Prestadora de Servicios de Saneamiento, EPS IlO S.A.

La empresa cuenta con un registro histórico de consumos de agua potable a lo largo de su funcionamiento, en el presente, se consideran los datos de los últimos dos años donde existen datos de consumo irregular clasificados manualmente. Por otro lado, el estado de arte de la investigación en línea de inteligencia artificial y en la disciplina Machine Learning, nos propone ciertos algoritmos o modelos para resolver este tipo de problemas.

Entonces, se realizó el estudio de diferentes modelos de Machine Learning para resolver el problema planteado esperándose un resultado favorable, la misma que son descritas y expresadas en los resultados y conclusiones de este informe científico.

El presente escrito trata de un informe de investigación que consta de cuatro partes: Introducción, marco teórico, metodología, resultados y conclusiones.

1.1. Descripción y formulación del problema

En los últimos tiempos, el sector de saneamiento se ha visto afectado por diversos motivos, tal es así que los usuarios ”reclaman por cobros excesivos en la EPS Moquegua S.A"(Uno, 2019), similar caso ocurre en la ciudad de Tacna, ”los reclamos se incrementaron hasta llegar a un 60 % por parte de los pobladores que han buscado alguna manera de evitar pagar más de lo habitual por la misma prestación que tenían"(Correo, 2019).

Por otro lado se menciona que ”vienen quejas de facturación, también llegan por que su medidor está en revisión o porque se les puso un consumo promedio"(Correo, 2019).

En realidad, el origen de los problemas son diversos y cada uno es particular, pero también existen problemas comunes o similares, por ejemplo, cuando una urbanización se encuentra en mantenimiento o cualquier otra actividad que afecte a un barrio en conjunto, también ocurre otro caso cuando el medidor de agua está vandalizado o superó su tiempo de vida, entre otros; a estos problemas se suma el error humano del personal encargado de realizar la toma de lecturas de los medidores de agua potable.

Entonces, existe alguna razón por el cual hay un registro de consumos atípicos o irregulares, los cuales son expresadas en los recibos de agua, por ende, genera malestar en los usuarios y posteriormente los reclamos a la empresa.

Por otro lado, en la Empresa Prestadora de Servicios de Saneamiento EPS Ilo S.A. tiene reunido los datos históricos de consumo de agua de sus usuarios, así mismo, la información del estado de sus medidores, consumos promedios, tipos

de usuarios y registro de anomalías identificadas y clasificadas manualmente. Estos datos por sí solos no ayudan mucho a identificar y anticiparse a futuras quejas de parte de los usuarios.

Los datos recolectados por la empresa se identifican en un total de 24 variables entre las directas e indirectas, influyentes para definir la irregularidad del consumo. En cuanto a los tipos o clases de irregularidades, se tiene a 10 tipos de anomalías.

Desde el punto de vista de la computación, existen diversos métodos de procesamiento de este tipo de datos para anticiparse a los hechos en la vida real, la inteligencia artificial nos presenta a Machine Learning como una disciplina científica, según (Drew Conway, 2012) ”un conjunto de herramientas y métodos que intentan inferir patrones y extraer información de un registro del mundo observable"(Pag. 2).

El problema de poseer múltiples variables y una información considerablemente grande, requiere acudir a algunas áreas o disciplinas de la inteligencia artificial, en la investigación sobre el modelo de detección de fraude en clientes del servicio de agua potable de una empresa sanitaria en la ciudad de Santiago, Chile, se utilizó la metodología KDD y se aplicó tres modelos de aprendizaje supervisado: árbol de decisión, regresión logística binaria y red neuronal, (Espinoza, 2014) ”indica que ante un problema coherente cuyas características hacen que pueda tener una solución confiable en el área de Data Mining"(pag. 2). Se definió como mejor modelo a árbol de decisión con la eficacia mas alta igual al 75% y su eficiencia fue de 81.2%. En la investigación Sistema inteligente basado en Machine Learning para la detección de fraude de facturación de agua potable, según (Rosales, 2019) las pruebas hechas con otros modelos de entrenamiento, se pueden incluir más técnicas de Machine Learning, tales como Redes Bayesianas, Regresión Lineal, entre otras, y finalmente obtener un híbrido que llegue a un mínimo de 97% de detección.

Entre los modelos de predicción se considera a Baseline Model, Segun (Peter A. Whigham, 2015),"puede usarse con datos mixtos numéricos y categóricos y no requiere ajuste de parámetros"(pag. 5). Sin embargo, en la investigación para predecir el consumo de energía HVAC por hora de un hotel en Madrid, España, (Muhammad Waseem Ahmad, 2017) "la incorporación de parámetros sociales, como el número de invitados, aumentó ligeramente la precisión de la predicción en el algoritmo Random Forest y ANN con error cuadrático medio (RMSE) de 4.97 y 6.10 respectivamente"(pag. 3).

En este caso particular, los datos son de múltiples variables de entrada y que a su vez son del tipo categórico y cuantitativo, el problema es predecir una de las 10 clases de consumo irregular identificadas manualmente por la empresa EPS Ilo S.A., por lo tanto, se propone la experimentación con el modelo Random Forest, Baseline Model y entre otras que se encuentren en el estado de arte en el proceso de experimentación.

La EPS Ilo S.A. no cuenta con un modelo de predicción de consumo irregular de sus clientes haciendo el uso de los datos históricos, por lo que el presente trabajo de investigación plantea averiguar el método eficiente de Machine Learning para el caso de predicción de consumo irregular del servicio de agua potable en la Empresa Prestadora de Servicios de Saneamiento Ilo S.A.

1.2. Antecedentes

En el estudio de (Quispe, 2017) sobre el consumo domestico mensual de agua potable en la ciudad de Ilave, se usó la metodología Box-Jenkis y el modelo ARIMA multiplicativo, el mejor modelo univariante que nos permite predecir y pronosticar el comportamiento del consumo mensual de agua potable de EMSA Puno S.A. Según (José Caridad y Ocerin, 2001), "el consumo de agua en una población de tamaño medio, evoluciona a lo largo del tiempo en función de

diversos factores causales, entre estos, cabe citar el número de habitantes y su densidad, la estructura de edad, la situación socio-económica, la industrialización de la zona, el clima, etc."(pag. 49).

En el estudio de modelo de detección de fraude en clientes del servicio de agua potable de una empresa sanitaria en la ciudad de Santiago, Chile, Según (Espinoza, 2014) concluye que ”en un mes se detectó alrededor de 360 clientes irregulares y, al mismo tiempo, se van produciendo nuevas conexiones fraudulentas, se sugiere comenzar a generar una lista de sospechosos con un modelo para cada zona, con la finalidad de tener parámetros más personalizados, debido a que cada zona posee ciertas características marcadas"(pag. 49).

En la investigación que propone ”desarrollar un Sistema Inteligente basado en un modelo híbrido de técnicas de minería de datos que pretende mejorar la tasa de exactitud en detección de un cliente en fraude de facturación de agua potable (...) según los antecedentes expuestos, existen varias empresas a nivel nacional e internacional prestadoras de servicios de agua potable que tienen el mismo problema presentado, por lo tanto, es posible plantearles dicha solución a estas empresas"(Rosales, 2019) (pag. 80).

También se realizó el estudio de influencia del sistema Dialog 3G en la optimización del proceso de toma de lecturas y facturación de SEDAPAL, caso: condominio nuevo cercado del distrito de Lima (Salcedo, 2015), donde se afirma que ”este tipo de sistemas influyen positivamente en el proceso de lectura y facturación"(pag. 57).

En la investigación de (Vera Letamendi, 2018) denominada ”minería de datos aplicada a la detección de patrones de reclamos del servicio de agua potable en la ciudad de Guayaquil", se aplicó la metodología KDD para descubrir un modelo predictivo válido, útil y entendible que especifique patrones de acuerdo a la información presentada; utilizando la herramienta KNIME como plataforma

tecnológica y la ejecución de los algoritmos de clasificación fundamentados en inteligencia artificial como: Árbol de decisión, Naive Bayes y Weka J48; dando como resultados: Árboles de decisión (éxito=98,485 %, error=1,515 %), Naive Bayes (éxito=97,273 %, error=2,727 %) y Weka (éxito=96,97 %, error=3,03 %); entonces se concluye que estos algoritmos son de utilidad en estos tipos de datos para obtener un modelo eficiente.

Existen estudios muy generales relacionados al consumo de agua y facturación, enfocados en la sociedad en conjunto, más otros estudios muestran la necesidad de recolección de información acerca del proceso de toma de lecturas y facturación para futuras predicciones más específicas, así mismo, otros proponen modelos de Machine Learning para predicciones más avanzadas y eficientes.

1.3. Objetivos

1.3.1. Objetivo general

Predecir con métodos de Machine Learning el consumo irregular del servicio de agua potable en la Empresa Prestadora de Servicios de Saneamiento S.A. en la provincia de Ilo.

1.3.2. Objetivos específicos

- Modelar la predicción de consumo irregular de agua potable con diferentes métodos de Machine Learning de acuerdo a la metodología CRISP-DM.
- Estudiar el método con mayor precisión para la predicción de consumo irregular de agua potable en la provincia de Ilo.

1.4. Justificación e importancia

La importancia del presente trabajo de investigación se justifica en el beneficio mutuo entre la sociedad y la empresa en cuanto a consumo irregular de agua potable, al tener al alcance este modelo de predicción la empresa podría prevenir la molestia de sus usuarios, porque con un modelo de predicción de este tipo se anticiparía a cobros excesivos e injustos. Entonces, este método ayudaría a las autoridades a tomar decisiones oportunas sobre los posibles usuarios de consumo irregular que son causados por diferentes factores, como el medidor obsoleto, medidores en mal estado, fuga de agua, conexiones clandestinas o algún otro particular externo que cause el incremento del recibo de agua.

1.5. Hipótesis

1.5.1. Hipótesis general

El uso de métodos de Machine Learning para predecir el consumo irregular del servicio de agua potable en la Empresa Prestadora de Servicios de Saneamiento S.A. en la provincia de Ilo es de mucha utilidad.

1.5.1.1. Hipótesis específicas

- El modelado de predicción de consumo irregular de agua potable con diferentes métodos de Machine Learning de acuerdo a la metodología CRISP-DM es muy ventajoso.
- El modelo con mayor precisión para la predicción de consumo irregular de agua potable en la provincia de Ilo es la red neuronal simple.

1.5.1.2. Variables

- **Variable independiente**: Modelos de Machine Learning.
- **Variable dependiente**: Predicción de consumo irregular del servicio de agua potable.

Capítulo II

Marco teórico

2.1. Bases teóricas

A continuación, se describen resumidamente los conceptos teóricos sobre los métodos y herramientas de Machine Learning aplicadas en el presente trabajo de investigación, también se define y se describe la estructura organizacional de la empresa según el Manual de Organizaciones y Funciones (MOF) de (EPS-Ilo-S.A., 2020b).

2.1.1. Inteligencia artificial

La Inteligencia Artificial (IA) es una ciencia que se utiliza para construir soluciones utilizando hardware y software. ”Está inspirado en la ingeniería inversa, por ejemplo, en la forma en que funcionan las neuronas en el cerebro humano. Nuestro cerebro consta de pequeñas unidades llamadas neuronas, y redes de neuronas llamadas redes neuronales. Más allá de las redes neuronales, hay muchos otros modelos en neurociencia que pueden usarse para resolver problemas del mundo real en inteligencia artificial"(Nagy, 2018).

Según (Joshi, 2020), ”la IA no significa construir una máquina extraordinariamente inteligente que pueda resolver cualquier problema en poco tiempo, sino que significa construir una máquina que sea capaz de comportarse como un ser humano". Además, según (Cruz, 2010) ”los métodos de la inteligencia artificial (IA) son una respuesta al deseo de aproximar el comportamiento y el pensamiento humano a diversos sistemas para la solución de determinadas problemáticas".

2.1.1.1. Machine Learning

Machine Learning se considera como una rama de inteligencia artificial, también tiene relación con los campos de las matemáticas, estadística y probabilidades, también se le considera un sub campo de la computación.

En cierto sentido, Machine Learning trata de la capacidad de convertir la experiencia en conocimientos o detectar patrones significativos en datos sensoriales complejos es una piedra angular de la inteligencia humana y animal (Ertel, 2017) (pag. 7).

La principal pretensión de Machine Learning es que la máquina aprenda de las experiencias y de los datos históricos de algún tema en particular para así ser expresados y utilizados en las técnicas o modelos de esta rama. Entonces, los modelos de aprendizaje automático toman estos datos para reconocer o detectar ciertos patrones de comportamiento para aprender, y este conocimiento aprendido es usado en un futuro problema. Existen tres tipos de aprendizajes, las cuales son:

- Aprendizaje supervisado:

  Se considera cuando existe un apoyo del ser humano en el proceso de etiquetado de la data, es decir, Machine Learning de tipo aprendizaje supervisado muestra un resultado ya calculado a partir de una data de entrada previamente elaborada por una persona, la misma que sabe del contenido

de los campos de la data y que espera un resultado.

- Aprendizaje no supervisado:

  Machine Learning del tipo aprendizaje no supervisado se trata de datos no identificados, es decir, los datos de entrada no están etiquetados y la tarea de Machine Learning no supervisado se encarga de diferenciarlos por si solos para que finalmente muestre las diferencias o grupos de tipos, mediciones, clases, entre otros, según el contexto del problema.

- Aprendizaje reforzado:

  En este tipo de Machine Learning difiere de las dos anteriores, porque se trata de retroalimentación interna, consulta constantemente para mejorar su precisión para descubrir el comportamiento deseado.

- Aprendizaje profundo

  El Deep Learning es un método específico de Machine Learning que incorpora las redes neuronales en capas sucesivas para aprender de los datos de manera iterativa. ”El Deep Learning es especialmente útil cuando se trata de aprender patrones de datos no estructurados. Las redes neuronales complejas de Deep Learning están diseñadas para emular cómo funciona el cerebro humano, así que las computadoras pueden ser entrenadas para lidiar con abstracciones y problemas mal definidos"(Judith Hurwitz, 2018).

2.1.1.2. Ciencia de datos

Ciencia de datos es una área de computación que consta de una variedad de técnicas estadísticas y probabilísticas para procesar grandes cantidades de información. Según (Davy Cielen, 2016) ”La ciencia de datos implica el uso de métodos para analizar cantidades masivas de datos y extraer el conocimiento que contiene".

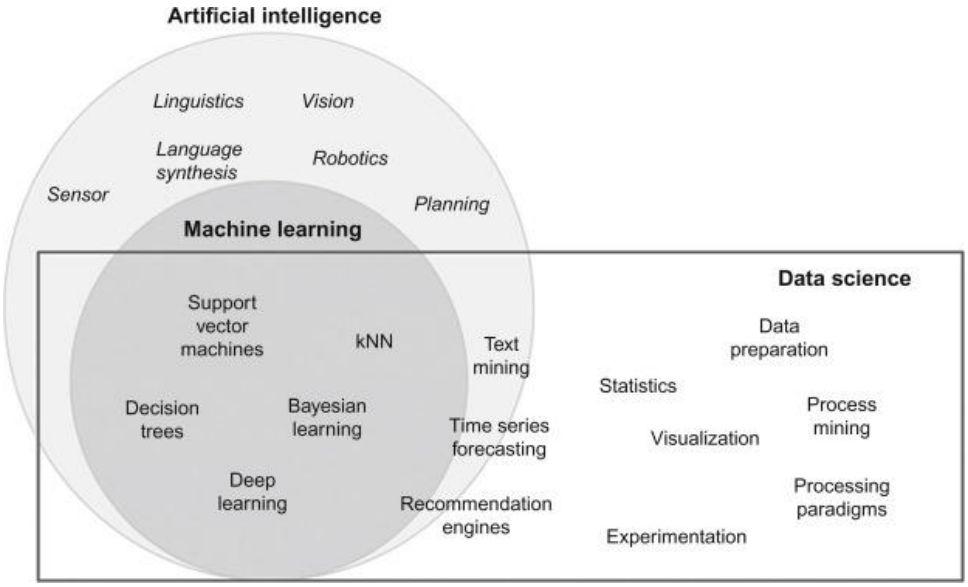
La ciencia de datos es una disciplina amplia, consta de técnicas y algoritmos

para manejar conjuntos de datos no estructurados muy grandes de manera automatizada, incluso sin interacción humana, para realizar transacciones en tiempo real o para hacer predicciones.

La inteligencia artificial, el aprendizaje automático y la ciencia de datos están relacionados entre sí. "Como era de esperar, a menudo se usan indistintamente y se combinan entre sí en los medios populares y la comunicación empresarial. Sin embargo, todos estos tres campos son distintos según el contexto"(Kotu y Deshpande, 2019). La relación entres estas disciplinas se puede apreciar en la figura II.1.

Figura II.1:

Relación entre Inteligencia Artificial, Machine Learning y Data Science.



Fuente: Kotu y Deshpande, 2019

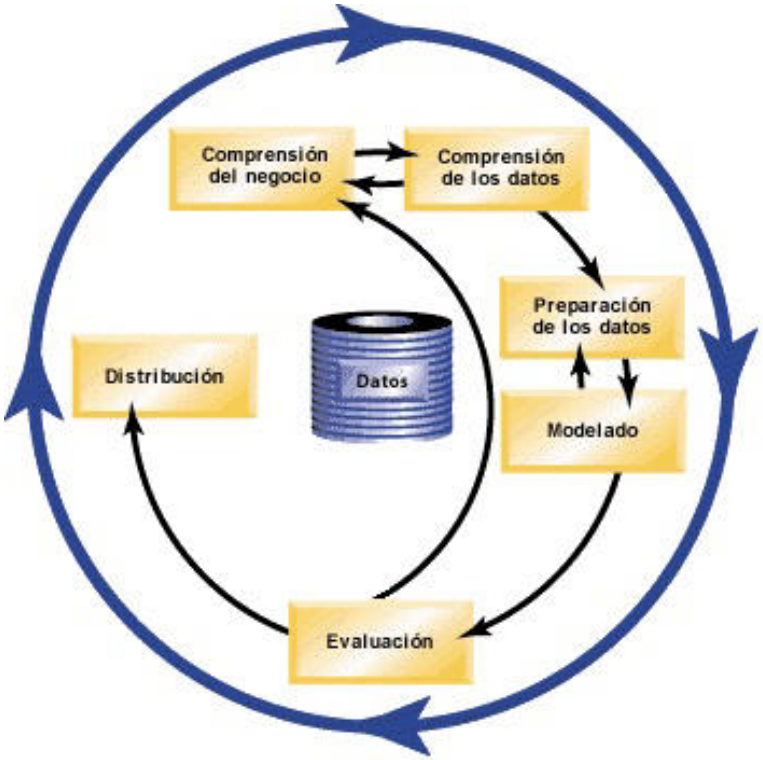
2.1.2. Metodología

Los siguientes procedimientos de la metodología CRISP-DM (Cross-industry standard process for data mining), están basadas en el autor (Rodica Branzei, 2005). Este consta de seis pasos para concretar un proyecto de minería de datos, el cual se detalla más adelante.

En el presente trabajo de investigación se adapta los cinco primeros procedimientos de la metodología, ya que el proyecto contempla hasta la evaluación del modelo, mas no la implementación del mismo, porque esta tapa demandaría una inversión de dinero a la empresa. Sin embargo, puesto que se trata de un trabajo de investigación, lo mas importante es evaluar la eficiencia de los modelos de Machine Learning.

Figura II.2:

Etapas de la metodología CRISP-DM



Fuente: Rodica Branzei, 2005

2.1.2.1. Comprensión de negocio

Debe obtenerse la máxima información posible de los objetivos de la minería de datos. Puede reducir los futuros riesgos clarificando los problemas, objetivos y recursos.

2.1.2.2. Comprensión de datos

Se estudia más de cerca los datos disponibles de minería. Es un paso esencial para evitar problemas inesperados durante la siguiente fase que suele ser la fase más larga del proyecto.

Esta fase implica acceder a los datos y explorarlos con la ayuda de tablas y gráficos. De esta manera se podrá determinar la calidad de los datos y describir los resultados de estos pasos.

2.1.2.3. Preparación de datos

Es uno de los aspectos más importantes y con frecuencia el que más tiempo exige en la minería de datos. La preparación de datos suele implicar las siguientes tareas:

- Fusión de conjuntos y/o registro de datos.
- Selección de una muestra de un subconjunto de datos.
- Agregación de registros.
- Derivación de nuevos atributos.
- Clasificación de los datos para el modelado.
- Eliminación o sustitución de valores en blanco o ausentes.
- División en conjuntos de datos de prueba y entrenamiento.

2.1.2.4. Modelado

Los datos preparados anteriormente se incorporan a las herramientas analíticas y los resultados comienzan a arrojar algo de luz al problema planteado.

El modelado se suele ejecutar en múltiples iteraciones. Normalmente se ejecutan varios modelos utilizando los parámetros por defecto, se ajustan los parámetros o vuelven a la fase de preparación de datos para las manipulaciones necesarias para el modelo.

2.1.2.5. Evaluación

En este punto se habrá completado la mayor parte del proyecto de minería de datos. También se habrá determinado, en la fase de modelado, que los modelos son técnicamente correctos y efectivos.

Sin embargo, antes de continuar, se debe evaluar los resultados de sus esfuerzos utilizando los criterios de rendimiento establecidos en el inicio del proyecto. Es la clave para asegurar que la organización pueda utilizar los resultados que se ha obtenido. La minería de datos produce dos tipos de resultados:

- Los modelos finales seleccionados en la fase anterior de CRISP-DM.
- Las conclusiones o interferencias obtenidas de los modelos y del proceso de minería de datos; reciben el nombre de descubrimientos.

2.1.3. Algoritmos propuestos

2.1.3.1. Árbol de decisiones

El algoritmo Decision Tree pertenece a la familia de algoritmos de aprendizaje supervisado. El algoritmo de árbol de decisiones consta de dos pasos importantes, el primer paso es aprendizaje automático del set de datos y el segundo paso es predecir en base a los patrones identificados en el anterior paso. Se puede utilizar para resolver problemas de regresión y clasificación. Según (Chauhan, 2020) ”El

objetivo de usar un árbol de decisión es crear un modelo de entrenamiento que se pueda usar para predecir la clase o el valor de la variable objetivo aprendiendo reglas de decisión simples inferidas de datos anteriores (datos de entrenamiento)".

- Nodo raíz: representa toda la población o muestra, y esta se divide en dos o más conjuntos homogéneos.
- División: es un proceso de división de un nodo en dos o más subnodos.
- Nodo de decisión: cuando un subnodo se divide en otros subnodos, se denomina nodo de decisión.
- Nodo Hoja / Terminal: Los nodos que no se dividen se denominan nodo Hoja o Terminal.
- Poda: cuando eliminamos subnodos de un nodo de decisión, este proceso se denomina poda. Se puede decir el proceso opuesto a la división.
- Rama / Subárbol: una subsección de todo el árbol se denomina rama o subárbol.
- Nodo padre e hijo: un nodo, que se divide en subnodos, se denomina nodo padre de subnodos, mientras que los subnodos son hijos de un nodo padre.

El método de selección de atributos es mediante el uso de Medias de selección de Atributos, el cual se conoce en matemáticas como una Heurística, este método divide los datos de mejor manera posible.

- Ganancia de la información:
  Es una propiedad estadística que mide la eficiencia de división de atributos en grupo de entrenamiento de acuerdo al objetivo de clasificación. Entonces, acudimos a la entropía para medir la impureza de conjunto de datos de entrada.

La entropía se puede calcular haciendo uso de la siguiente fórmula matemática:

$$Entropia = -plog_2 p - qlog_2 q \tag{II.1}$$

p y q son probabilidades de mayor y menor, mientras sea menor, la entropía es mejor.

- Gini:

  Gini es para saber la homogeneidad de la población, es 0 la máxima igualdad y 1 la máximo desigualdad. Entonces, funciona como valor objetivo de categoría de éxito o fracaso.

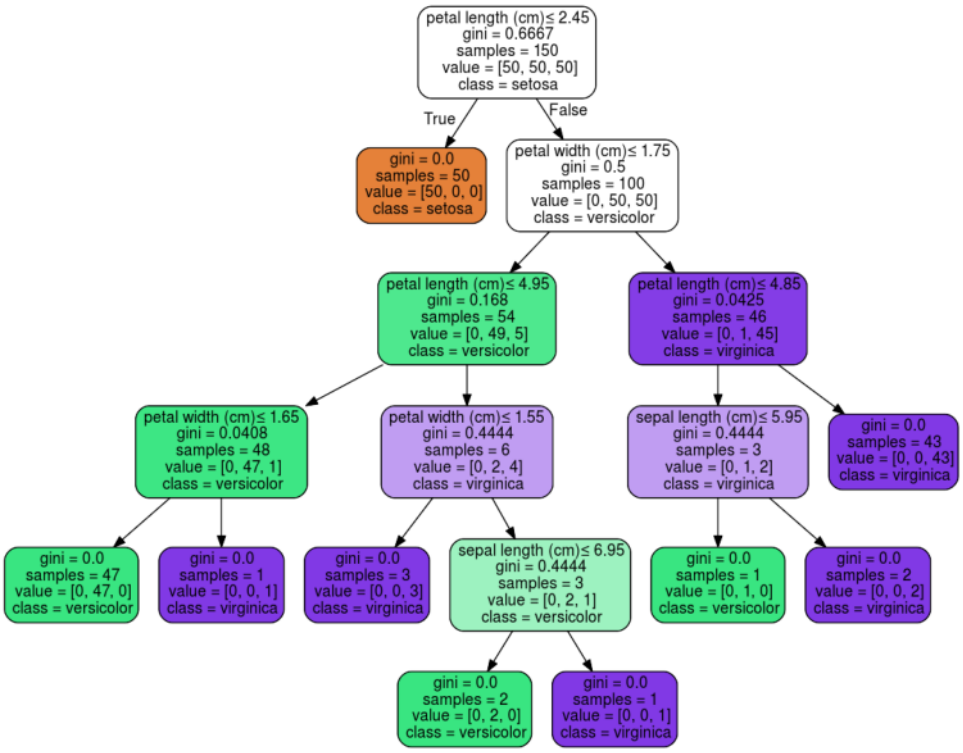
2.1.3.2. Random forest

Random Forest es una de las técnicas de Machine Learning más potentes y totalmente automatizadas. Casi sin experiencia en preparación de datos o modelado, los analistas pueden obtener sin esfuerzo modelos sorprendentemente efectivos. 'Bosques aleatorios' es un componente esencial en el conjunto de herramientas del científico de datos moderno y en esta breve descripción se toca los elementos esenciales de esta metodología innovadora.

Un bosque aleatorio es un clasificador que consiste en una colección de clasificadores estructurados en árbol $\{h(x,\Theta_k), k = 1,...\}$ donde los $\{\Theta_k\}$ son vectores aleatorios independientes distribuidos idénticamente y, cada árbol, emite un voto por unidad para la mayor clase popular en la entrada x (Breiman, 2001). Se trata de un método no perimétrico, robusto y fácil de interpretar (Cánovas-García, Alonso, y Castillo, 2016), la misma que también puede ser visualizada en forma gráfica como se aprecia en la figura II.3.

Figura II.3:

El árbol en formato Graphviz usando el export_graphviz exportador.



Fuente: Breiman, 2001

2.1.3.3. Baseline model

Baseline model es un modelo simple que tiene una probabilidad razonable para proporcionar resultados decentes, en la literatura comentan que es un modelo para probar y un paso para elegir modelos mas óptimas de Machine Learning. La ventaja de este modelo es que es fácil de experimentar sin mucho recurso de cómputo, puesto que los paquetes existen.

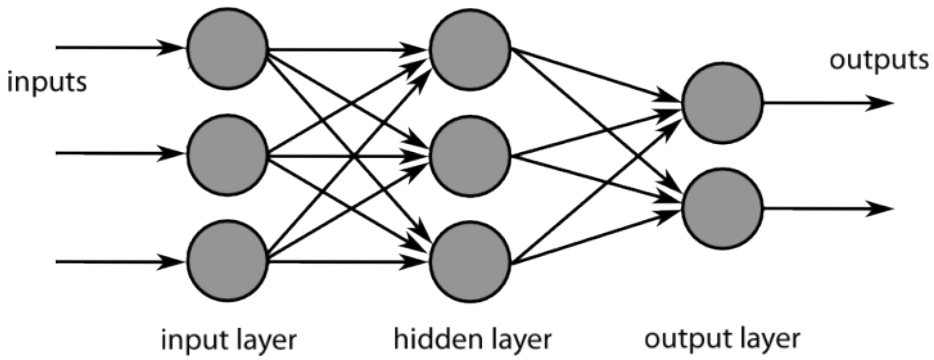
Es un modelo simple que tiene una sola capa oculta completamente conectada con el mismo número de neuronas que los atributos de entrada. La red utiliza buenas prácticas como la función de activación del rectificador para la capa oculta. Se utiliza el eficiente algoritmo de optimización ADAM y se optimiza una función de pérdida de error cuadrático medio(*Machine Learning Mastery*, s.f.).

Una línea de base es un método que utiliza heurística, estadísticas de resumen simples, aleatoriedad o aprendizaje automático para crear predicciones para un conjunto de datos.

La topología de red de esta red neuronal simple de una capa se puede resumir, como se puede apreciar en la figura II.4.

Figura II.4:

Modelo de red neuronal de una sola capa oculta.



Fuente: Machine Learning Mastery, s.f.

Existe una clase KerasClassifier en Keras que se puede usar como Estimador en scikit-learn, el tipo base de modelo en la biblioteca. El KerasClassifier toma el nombre de una función como argumento. Esta función devuelve el modelo de red neuronal construido, listo para el entrenamiento. Este se muestra en una función que creará una Baseline Model.

2.2. Empresa Prestadora de Servicios de Saneamiento Ilo S.A.

2.2.1. Descripción de la empresa

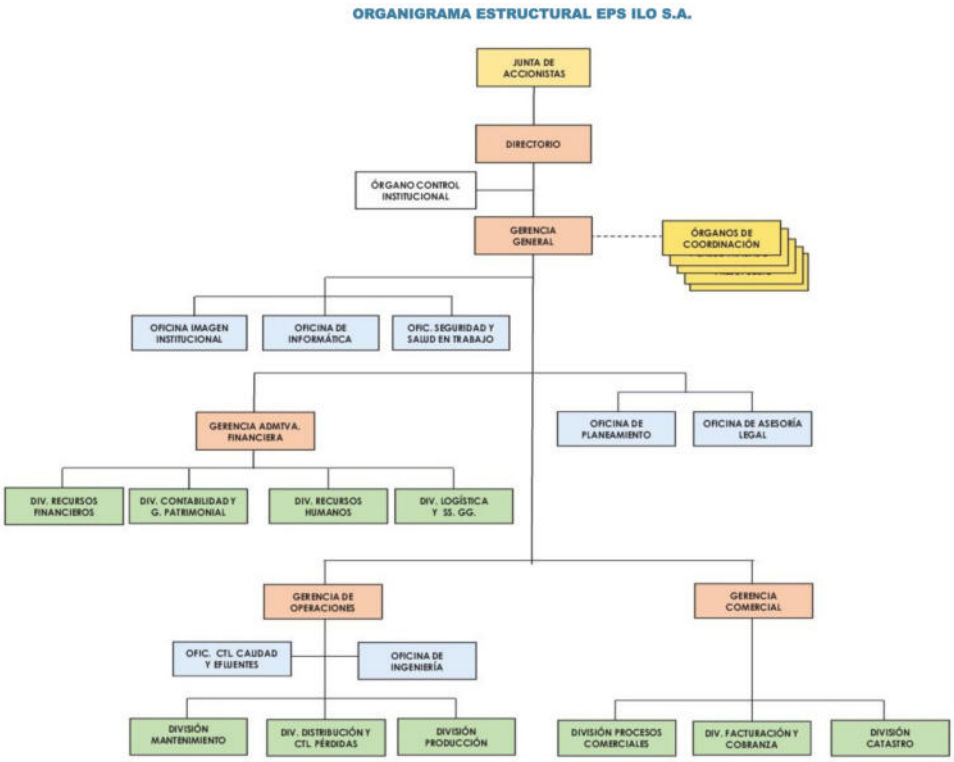
Según (EPS-Ilo-S.A., 2020a) la Empresa Prestadora de Servicios de Saneamiento Ilo S.A. es una entidad pública de derecho privado que actúa con autonomía financiera - administrativa encargándose de la prestación de servicios públicos de agua potable y alcantarillado en el ámbito de la Provincia de Ilo.

2.2.2. Organización

La EPS Ilo S.A. tiene la siguiente estructura organizacional, la cual fue extraída del MOF de la empresa.

Figura II.5:

Organigrama de la EPS Ilo S.A.



Fuente: EPS Ilo S.A. (2020a)

- Junta general de acciones:

  Consta de tres representantes de la Municipalidad Provincia de Ilo, Pacocha y El Algarrobal. Se encargan de examinar los acuerdos, constatar acciones, decidir en el momento en que las acciones se sometan a voto y cuidar que se de cumplimiento a las disposiciones de los Estatutos.

- Directorio:

  Vigila el cumplimiento de los fines y objetivos para los que fue creada la empresa, vela por el cumplimiento de los acuerdos de la Junta General de Accionista y del Directorio, además de coordinar con el Jefe de Órgano de Control Interno, el estricto cumplimiento de las normas de control.

- Gerencia general:

  Supervisa a todos los órganos de linea, de apoyo y asesoría. Comprende el

planeamiento, organización, dirección, control, evaluación, resolución y coordinación general de las actividades operacionales, comerciales y administrativos de la EPS.

- Gerencia administrativa - financiera:

  Esta conformado por el Gerente Administrativo Financiero y un Asistente Administrativo II. Controla la capacidad empresarial y el auto-financiamiento de la EPS, optimizando la utilización de los recursos financieros, humanos, materiales y de bienes de capital.

- Gerencia de operaciones:

  Comprende al Gerente de Operaciones, Asistente Administrativo II y Supervisor de Servicios. Supervisa a la Oficina de Ingeniería, Oficina de Control de Calidad y Efluentes, División de Producción, División de Distribución y Control de Perdidas, División de mantenimiento, demás personal de la Gerencia de Operaciones.

  Planea, organiza, dirige, coordina, controla, evalúa y norma la producción, distribución y control de calidad del agua potable, la recolección, tratamiento y disposición final de desagües, el mantenimiento de la infraestructura de producción y distribución.

- Gerencia comercial:

  Conformado por el Gerente Comercial y Asistente Administrativo II. Supervisa a la División de Procesos Comerciales, División de Facturación y Cobranza, División de Catastro y Asistente Administrativo II. Entre sus actividades generales se tiene planificar, organizar, ejecutar y controlar todas las actividades vinculadas a la comercialización, catastro, medición, facturación y cobranza para la prestación de los servicios de agua potable, alcantarillado y servicios colaterales en el ámbito de la EPS ILO S.A.

2.2.3. Principales oficinas encargadas

A continuación se detalla las funciones de las principales oficinas que dan soporte a los procesos de la gerencia comercial, facturación y catastro, divisiones que se involucran directamente en los problemas que la presente investigación estudia.

- División de procesos comerciales:

  Supervisa las áreas de Asistente de Atención al Cliente, Auxiliar de Contratos y comercialización, Supervisor Operativo Comercial, Supervisor de Banco de Medidores.

  Las actividades del puesto son planificar, organizar, formular, coordinar, ejecutar y controlar las actividades y/o programas de los procesos comerciales y atención al cliente. Así como la supervisión y control de las actividades de mantenimiento preventivo y correctivo de los micro medidores

- División de facturación y cobranza:

  Supervisa el área de Técnico de Cobranza Dirige la programación, ejecución y control del proceso de facturación y cobranza, buscando alta eficiencia de cobranza de los montos facturados por los servicios que presta la empresa. Así mismo, disminuir la cartera pasada por deudas y los índices de morosidad debiendo realizar las acciones correspondientes de conformidad a las normas técnicas legales e internas vigentes.

- División de Catastro:

  Planifica, ejecuta, coordina, supervisa y controla el mantenimiento del registro catastral de los clientes reales y factibles, identificando los clientes potenciales de la EPS ILO S.A.

2.3. Definición de términos

- Machine Learning:

  El aprendizaje de máquina o aprendizaje automático es una forma de inteligencia artificial que permite que un sistema aprenda utilizando una variedad de algoritmos para mejorar, describir datos y predecir resultados.

- CRISP-DM:

  Son las siglas de Cross-Industry Standard Process for Data Mining, es un método probado para crear un modelo y orientar los trabajos de minería de datos. Como metodología, incluye descripciones de las fases normales de un proyecto, las tareas necesarias en cada fase y una explicación de las relaciones entre las tareas; como modelo de proceso, ofrece un resumen del ciclo vital de minería de datos.

- Random Forest:

  Es una técnica de aprendizaje automático, útil para problemas de predicción. aplica la agregación de datos y la selección de características aleatorias a árboles de clasificación o regresión individuales para la predicción.

- Baseline Model:

  Es un algoritmo de Machine Learning que utiliza heurística, estadísticas de resumen simples, aleatoriedad o aprendizaje automático para crear predicciones para un conjunto de datos.

- IA:

  El propósito de la inteligencia artificial es hacer computacional el conocimiento humano por procedimientos simbólicos (programable y basado en el supuesto del conocimiento explicable por procedimientos de manipulación de símbolos) o conexionistas (autoprogramable por aprendizaje y donde el conocimiento viene representado la propia estructura de la red neuronal).

- EPS ILO S.A.:

  Empresa Prestadora de Servicios de Saneamiento S.A.

- MOF:

  Manual de organización y funciones.

Capítulo III

Método

3.1. Tipo y nivel de investigación

3.1.1. Tipo de investigación

El tipo de investigación es aplicativo, puesto que se pone a prueba métodos o técnicas desarrolladas y validadas en la comunidad científica de la computación, inteligencia artificial, específicamente en la disciplina Machine Learning.

3.1.2. Nivel de investigación

EL nivel de investigación es experimental, ya que en el proceso se manipula la variable dependiente e independiente, los 17 atributos de influencia directa a la clase a predecir, y el estado de lectura(EL) respectivamente.

3.2. Ámbito temporal y espacial

La investigación es realizada en el año 2020 con los datos recolectados en los años 2018 y 2019, estos datos corresponden a los usuarios afiliados a la EPS Ilo S.A. que cuentan con un medidor de agua potable, por lo tanto, se puede definir el ámbito espacial a todos los medidores de agua potable de los usuarios de la EPS Ilo S.A. de la provincia de Ilo, región Moquegua.

3.3. Población y muestra

3.3.1. Población

La población de la presente investigación son los medidores de agua potable de los usuarios de la EPS Ilo S.A. de la Provincia de Ilo, región Moquegua.

3.3.2. Muestra

La muestra considerada en la presente investigación son los medidores de agua potable que alguna vez registraron consumo irregular del servicio de agua potable, en total suman 83797 registros de la empresa mencionada.

3.4. Instrumentos

La presente investigación es desarrollada con instrumentos de computación, por ejemplo: una computadora para redacción y procesamiento de datos, internet para la investigación de diferentes técnicas y estado de arte de machine learning, lenguaje de programación Python para automatizar el procesamientos de datos y

visualización de resultados.

Todo el proceso esta basada en la metodología CRISP DM, los procedimientos establecidos por la metodología fueron efectuados para cumplir con los objetivos planteados, que en la siguiente sección se explica a mayor detalle.

3.5. Procedimientos

Los procedimientos en la presente investigación están basadas en la metodología Cross Industry Standard Process for Data Mining (CRISP DM), una metodología bastante utilizada en el ámbito académico y en la industria.

La metodología empleada consta de cinco pasos, los cuales son: Comprensión de negocio, comprensión de datos, preparación de datos, modelado y evaluación. La teoría de la metodología ha sido desarrollada en la parte de marco teórico de la presente investigación.

3.6. Análisis de datos

Se considera muy importante el uso de métodos de inteligencia artificial en resolución de problemas con datos de la Empresa Prestadora de Servicios de Saneamiento (EPS ILO S.A.). Se puede alcanzar excelentes resultados con aplicación de métodos de Machine Learning en este tipo de problemas en las empresas locales de la provincia de Ilo.

Los resultados de la presente investigación ha demostrado que sí se puede predecir diferentes clases de consumo irregular de agua potable de la empresa mencionada.

EL conocimiento generado en el presente trabajo es un aporte importante

para los investigadores locales y autoridades para tomar decisiones sobre los datos tratados con Machine Learning.

Capítulo IV

Resultados

A continuación se detallan los resultados de los procedimientos realizados de acuerdo a las fases de la metodología CRISP DM.

4.1. Comprensión de negocio

El objetivo de Machine Learning que se va a aplicar en esta investigación es el hacer predicciones lo más fiables posible a partir de los datos que ya se disponen de los usuarios de la EPS Ilo S.A. El objetivo es reducir los reclamos de los usuarios para así brindar un mejor servicio.

Contexto:
En cuanto a la situación de la empresa, desde el principio de esta investigación se cuenta con una base de datos de los usuarios, tales como: datos del predio, de la conexión, del medidor, facturaciones, rebajas realizadas por reclamos, etc. Sin embargo no existe ningún estudio en profundidad sobre el comportamiento de lecturas y consumos de los medidores de los que se puedan sacar conclusiones o patrones para hacer predicciones sobre los futuros estado de lectura.

Objetivos del negocio:

Los objetivos de la investigación, como ya se ha mencionado, son la predicción de consumo irregular de los usuarios, de tal manera que se pueda hacer una estimación fiable partiendo de los datos que ya se tiene. Sin embargo, algunos objetivos de la empresa y del área involucrada, es la reducción de reclamos y la identificación de usuarios con consumos elevados (irregulares).

Los resultados de estas predicciones pueden ayudar a aquellos usuarios que presentan problemas en sus consumos y de esta forma intentar averiguar el por qué de estas situaciones.

Criterios de éxito del negocio:

Desde el punto de vista de la empresa se establece como criterio de éxito la posibilidad de realizar predicciones sobre consumos irregulares de agua potable con un elevado porcentaje de precisión y fiabilidad, de tal forma que se pueda anticipar y notificar a los usuarios acerca de su consumo, además de advertir ante posibles fugas de agua en el predio. Otro criterio de éxito de la empresa sería elevar el porcentaje de recaudación y la disminución de reclamos.

4.2. Comprensión de datos

La EPS Ilo S.A. cuenta con una base de datos histórica de consumo de agua potable en la ciudad de Ilo, el cual consta de un total de 24 atributos (campos) que presentan de manera explícita a los detalles de los usuarios y la empresa, estos datos representan el comportamiento de muchos usuarios a lo largo de los dos últimos años, por ejemplo; en el campo IMPF, registra todas las facturaciones efectuadas a cada uno de los usuarios, ente tipo de datos sirven para mostrar gráficamente cuánto o cómo ha evolucionado un determinado usuario en cuanto a su facturación. La descripción de cada uno de los atributos se puede apreciar en la tabla (IV.1).

Cuadro IV.1:

Descripción de variables del dataset de la empresa EPS Ilo S.A.

Numero	Código	Descripción
1	MES	Corresponde al mes facturado.
2	SUC	Sucursal: Puede ser Ilo, El algarrobal o Pacocha.
3	SEC	Sector: Cada sucursal se divide en sectores, y este a la vez se divide en calles. Por ejemplo: Sucursal Ilo se divide en trece (13) sectores, donde el sector 1 contiene las calles Villa el Eden, 2 de marzo, Vista al mar, Jose Olaya, entre otros.
4	URB	Tipo de calle, puede ser Urbanización, Asociación, Jiron, Promuvi, entre otros.
5	CALLE	Nombre de la zona, tales como Miramar, 24 de octubre, Chalaca, Ciudad nueva, etc.
6	ACT	Es la actividad a la que un usuario se dedica, puede ser un restaurante, vivienda, pileta publica, ferretería, colegio, etc.
7	CODIGO	Código de inscripción del usuario.
8	TS	Tipo de servicio que un usuario posee: Agua y desagüe, solamente agua o solamente desagüe.
9	CAT	La categoría tarifaria a la que un usuario pertenece, pudiendo ser domestico, comercial, industrial, estatal o social.
10	MEDIDOR	Número de medidor.

*Continua en la página siguiente.

Numero	Código	Descripción
11	TP	Tipo de promedio: El código 0 significa que se tomó lectura al medidor, 1 significa que la conexión no tiene medidor y se le asigna un consumo fijo, 2 es el consumo promediado de los últimos 6 meses debido a algún tipo de impedimento al tomar la lectura.
12	LECTA	Corresponde a la lectura anterior tomada el mes previo.
13	LECTU	Corresponde a la última lectura del mes actual.
14	CSM	Consumo por diferencia de lecturas.
15	EL	Estado de lectura: Se define de acuerdo a la lectura tomada, pudiendo ser Lectura normal, consumo excesivo, consumo atípico, no registró consumo, invertido, entre otros.
16	CSMF	Consumo final facturado.
17	IMPF	Importe facturado en el mes.
18	IMPR	Importe rebajado por algún tipo de reclamo realizado por el usuario.
19	IMPC	Importe cobrado / pagado por el usuario.
20	IMPD	Importe faltante a pagar.
21	LECTP	Lectura promedio: Es el promedio calculado por las 6 últimas lecturas.
22	FINS	Fecha de instalación del medidor.
23	SM	Situación del medidor: Puede ser que el medidor esté o no reinstalado.
24	EM	Estado del medidor, puede ser: buena condición, luna opaca, paralizado, enterrado, retirado, etc.

Fuente: Elaboración propia

Nota 1. Algunos de los atributos definidos en la tabla anterior son de tipo categoría, por lo tanto estas son definidas por un código que lo identifica.

Nota 2. Los datos de la tabla anterior son de propiedad de la empresa (EPS-Ilo-S.A., 2020a).

Sin embargo, el campo principal es el Estado de lectura (EL) que consta de 10 clases que son definidas por un código que la identifica. Se puede apreciar mas al detalle en la tabla (**??**).

Cuadro IV.2:

Codificación de la atributos de estados de lectura.

Numero	Código	Descripción
1	000	Lectura normal.- Corresponde a una lectura dentro del rango de lo normal.
2	001	Consumo Excesivo.- Solo se aplica a usuarios de categoría comercial, se establece cuando el consumo de agua potable es excesivo de acuerdo a su historial de consumos.
3	002	Lecturas Iguales.- Es cuando la lectura del mes anterior es igual a la lectura del mes actual, es decir, no existe consumo pero se exceptúan las lecturas ceros.
4	003	No Registro Consumo y/o Lectura.- Se establece si la lectura actual es cero o hubo algún tipo de impedimento en la toma de lectura por parte del operador.
5	004	Consumo Negativo.- Usualmente corresponde a medidores invertidos, donde la lectura va disminuyendo mensualmente en lugar de aumentar.

*Continua en la página siguiente.

Numero	Código	Descripción
6	007	Conexión Cerrada con Lectura.- El usuario tiene el servicio cortado pero aún así el medidor registra consumos, lo cual no debería darse.
7	008	Consumo Atípico.- Se aplica a todas las categorías tarifarías excepto a la categoría comercial, y corresponde cuando el consumo es mayor al doble del promedio de los últimos 6 meses.
8	010	Consumo menor al Promedio.- Se aplica si el consumo es menor al promedio histórico.
9	011	Consumo Doble al Promedio.- Se aplica si el consumo es el doble del promedio histórico.
10	999	No registra Lectura.- Similar al ítem 4, con la diferencia de que esta opción se utiliza en versiones anteriores del sistema integro comercial.

Fuente: Elaboración propia

Nota 1. Los datos de la tabla anterior son de propiedad de la empresa (EPS-Ilo-S.A., 2020a).

Los datos definidos en la Tabla **??**, son las clases a predecir en el presente trabajo, ya que representa a los diferentes consumidores irregulares. Sin embargo, la clase del código 000 (lectura normal) no se considera en el presente por no ser considerado como consumo irregular.

Además, también es excluido la clase 009, porque su existencia es de solo un registro, por lo tanto el numero de clases de reduce a 8.

4.3. Preparación de datos

En este apartado se define el conjunto de datos para experimentar en el presente trabajo de investigación. Para lograr los objetivos planteados se ha seleccionado el conjunto de datos que se puede apreciar en la Figura (IV.1). En este conjunto de datos la clase a predecir es el atributo Estado de lectura(EL), por lo tanto, los atributos seleccionados son directamente influyentes a la clase Estado de lectura (EL). Entonces, en resumen se tiene 17 atributos como variable independiente y 1 atributo como variable dependiente que es la clase a predecir.

Figura IV.1:

Conjunto de datos de entrenamiento (dataset)

```
df = pd.read_csv("/content/dic_2020.csv")
df.head(n=5)
```

	SUC	SEC	URB	CALLE	ACT	TS	CAT	TP	LECTA	LECTU	CSM	IMPF	IMPR	IMPC	IMPD	LECTP	EM	EL
0	1	1	2	8253	5	1	113	2	0	0	0	22.1	0.0	22.1	0.0	10	1	3
1	1	1	2	8258	5	1	113	2	308	498	190	37.6	0.0	37.6	0.0	15	1	8
2	1	1	2	8253	5	1	113	0	126	135	9	20.3	0.0	20.3	0.0	3	1	11
3	1	1	2	8256	5	1	113	0	124	126	2	7.4	0.0	7.4	0.0	1	1	11
4	1	6	2	8158	5	1	113	0	154	154	0	3.7	0.0	3.7	0.0	2	1	2

Fuente: Elaboración propia

Los datos de la tabla anterior son de propiedad de la empresa (EPS-Ilo-S.A., 2020a).

En total se tiene 17 atributos independientes y 1 atributo dependiente que es el estado de lectura(EL).

El conjunto de datos seleccionado para la presente experimentación consta de 8 clases, la cantidad de registros existentes por clases se puede a preciar en la Figura (IV.2).

Figura IV.2:

Número de registros por clase

```
1 print(df.shape)
2 print(pd.value_counts(df['EL'], sort = True))
```

```
(83714, 18)
10    26703
2     20140
3     15108
11    12934
4      7095
8      1017
7       404
1       313
Name: EL, dtype: int64
```

Fuente: Elaboración propia

Las clases están representadas por códigos que se detallan en la sección de comprensión de datos.

Se evidencia que los registros por clase no son equilibrados como se puede apreciar en la Figura IV.3, el cual no es bueno para los algoritmos de Machine Learning (Mikel Galar, s.f.) porque los conjuntos de datos que sufren de distribuciones de clases desequilibradas es un problema desafiante en la comunidad de minería de datos. Este problema ocurre cuando el número de ejemplos que representan una clase es mucho menor que el de las otras clases.

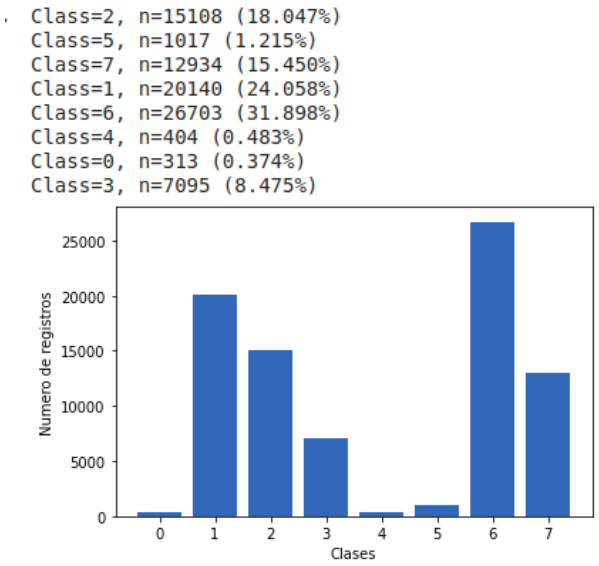
Según (Kantardzic, 2020), una de las técnicas de sobremuestreo más populares es la técnica de sobremuestreo de minorías sintéticas (SMOTE). Su idea principal es formar nuevas muestras de clases minoritarias interpolando entre varias muestras de clases minoritarias que se encuentran muy juntas.

De acuerdo al estado de arte de minería de datos, con respecto a conjunto de datos desbalanceados, recomiendan a la técnica Synthetic Minority Oversampling Technique (SMOTE), un técnica eficiente en sobre muestrear la clases minoritarias para equilibrar conjunto de datos. Por lo tanto, en la presente investigación se considera la aplicación de la técnica mencionada para este objetivo. En la Figura (IV.3) se puede apreciar los datos NO balanceados, sin embargo, en la Figura (IV.4)

se llega apreciar los datos completamente balanceados.

Figura IV.3:

Número de registros por clase sin aplicar SMOTE.

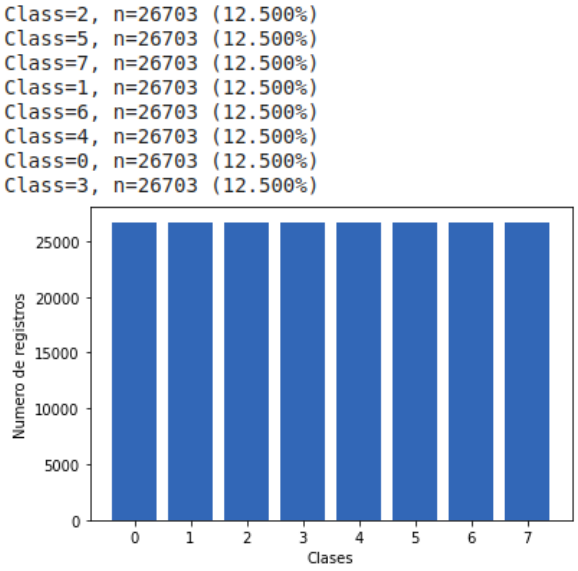


Fuente: Elaboración propia

En esta figura, el código de cada clase es reasignada de acuerdo al orden de la clase en el conjunto de datos.

Figura IV.4:

Número de registros por clase aplicando SMOTE.



Fuente: Elaboración propia

En esta figura, el código de cada clase es reasignada de acuerdo la orden de la clase en el conjunto de datos.

El conjunto de datos usado para el propósito de la presente investigación queda preparado para el procesamiento con los modelos a estudiar.

4.4. Modelado

En esta sección se presentan a mayor detalle el estudio de los modelos de Machine Learning para predecir el consumo irregular del servicio de agua potable basado en el conjunto de datos preparado en la sección anterior.

4.4.1. Árbol de decisiones

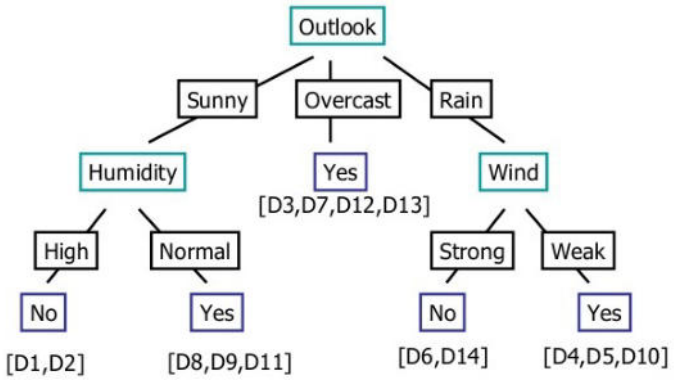
Árbol de decisiones es uno de los algoritmos mas populares para el problema de predicción de clases. Los algoritmos de aprendizaje basados en árboles se

consideran uno de los mejores y más utilizados métodos de aprendizaje supervisado. Los métodos basados en árboles potencian los modelos predictivos con alta precisión, estabilidad y facilidad de interpretación.

Árbol de decisión tiene una estructura muy similar a un diagrama de flujo, como se aprecia en la figura IV.5. Donde el nudo representa a una propiedad (atributo); en el presente estudio, los datos son las características de los usuarios de la empresa. La rama representa a una regla de decisión que permite ascender al siguiente nivel. El procedimiento de decisión es de manera recursiva, esta basado en la condición de la propiedad o atributo. De esta manera el enfoque de arriba hacia abajo alcanza una ramificación óptima posible de acuerdo a la complejidad de los datos, cada nudo actúa como una pregunta o condición para un atributo, dependiendo de la condición del atributo, asciende por uno de las dos ramas y al siguiente nivel de nudo, el cual es una posible respuesta a la pregunta. Todo este proceso se le denomina recursivo.

Figura IV.5:

Diagrama del algoritmo Árbol de decisión.



Fuente: Kantardzic,M. (2020).

El método de selección de atributos es mediante el uso de Medias de selección de Atributos, el cual se conoce en matemáticas como una Heurística, este método divide los datos de mejor manera posible.

- Ganancia de la información:

  Es una propiedad estadística que mide la eficiencia de división de atributos en grupo de entrenamiento de acuerdo al objetivo de clasificación. Entonces, acudimos a la entropía para medir la impureza de conjunto de datos de entrada.

  La entropía se puede calcular haciendo uso de la siguiente fórmula matemática:

$$Entropia = -plog_2 p - qlog_2 q \tag{IV.1}$$

  p y q son probabilidades de mayor y menor, mientras sea menor la entropía es mejor.

- Gini:

  Gini es para saber la homogeneidad de la población, es 0 la máxima igualdad y 1 la máxima desigualdad. Entonces, funciona como valor objetivo de categoría de éxito o fracaso.

A continuación, se presenta el código del algoritmo de Árbol de Decisión realizado para la experimentación del presente trabajo de investigación.

Figura IV.6:

Código del algoritmo de Árbol de decisión en Python.

```
[ ]  1 # Splitting the dataset into the Training set and Test set
     2 from sklearn.model_selection import train_test_split
     3 X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.25, stratify=y, random_state = 42)
```

```
     1 # Fitting Classifier to the Training Set
     2 from sklearn.tree import DecisionTreeClassifier
     3 classifier = DecisionTreeClassifier(criterion='entropy',max_depth=3, random_state=42)
     4 classifier.fit(X_train, y_train)

DecisionTreeClassifier(ccp_alpha=0.0, class_weight=None, criterion='entropy',
                       max_depth=3, max_features=None, max_leaf_nodes=None,
                       min_impurity_decrease=0.0, min_impurity_split=None,
                       min_samples_leaf=1, min_samples_split=2,
                       min_weight_fraction_leaf=0.0, presort='deprecated',
                       random_state=42, splitter='best')
```

Fuente: Elaboración propia

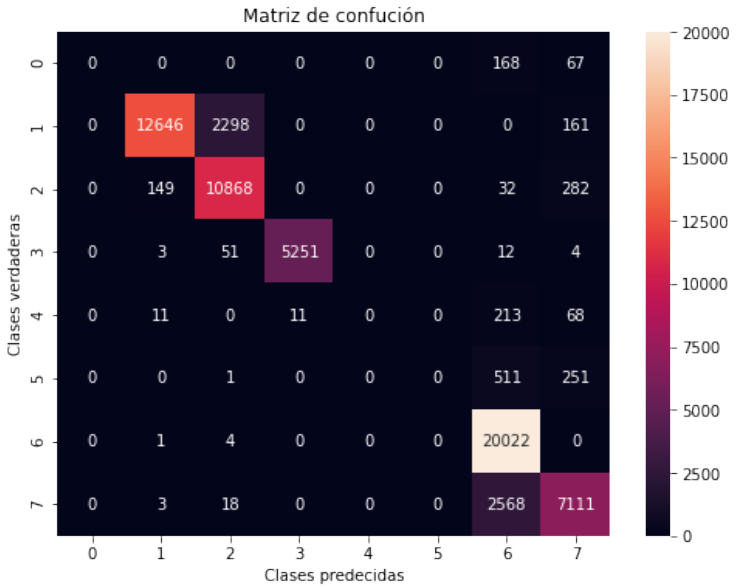
También se presentan los resultados de predicción sin haber preprocesado con el algoritmos SMOTE. Se puede apreciar en la Figura IV.7 que los resultados

no muestran el mayor número de aciertos en la predicción de clases. Sin embargo, la diferencia se nota en la Figura IV.8.

Esta diferencia existe porque el primer entrenamiento es realizado sobre los datos desbalanceados, es decir, existen mucha diferencia entre cantidades de registros por clase.

Figura IV.7:

Matriz de confusión del Árbol de decisión con datos desbalanceados.



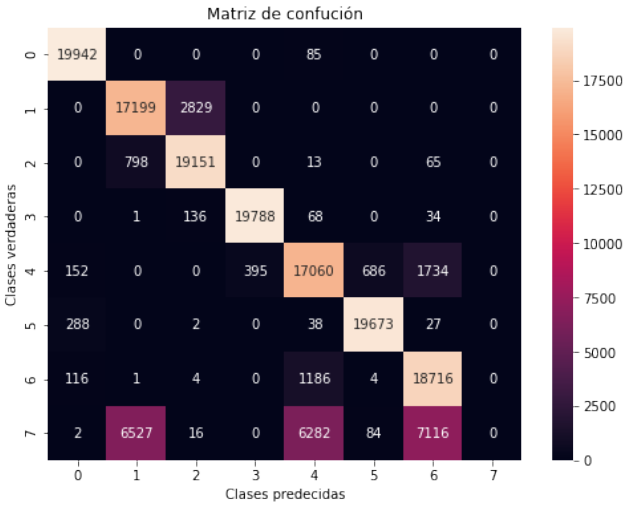
Fuente: Elaboración propia

En esta figura, el código de cada clase es reasignada de acuerdo al orden de la clase en el conjunto de datos.

El resultado de predicción del algoritmo árbol de decisión se muestra en la Figura IV.8, donde se puede apreciar el incremento en el número de aciertos en la predicción de clases. Sin embargo, existe una deficiencia en la predicción de clase 7, aparentemente el algoritmo no logra diferenciar entre la clases número 6 y número 7, ya que se muestra en la figura IV.7 que existen 7111 registros clasificados como clases 7, mientras que en la figura IV.8 muestran 7116 registros clasificados como clase 6.

Figura IV.8:

Número de registros por clase aplicando SMOTE.



Fuente: Elaboración propia

En esta figura el código de cada clase es reasignada de acuerdo al orden de la clases en el conjunto de datos.

4.4.2. Red neuronal simple

En esta sección se presentan los resultados del algoritmo Red Neuronal Simple, un algoritmo muy popular para iniciar proyectos de clasificación con múltiples entradas y salidas.

Es importante tener en cuenta que se usa softmax función de activación en la capa de salida, esto es para asegurar que los valores de salida estén en el rango de 0 y 1, para que así puedan usarse como probabilidades predichas.

Finalmente, la red utiliza el eficiente algoritmo de optimización de descenso de gradiente de Adam con una función de pérdida logarítmica, que se denomina categórica_cruzada en Keras.

Figura IV.9:

Preparación del dataset para Red Neuronal Simple

```
In [3]: dataframe = pd.read_csv("dic_2020_f.csv", header=None, delimiter=",")
        dataset = dataframe.values
        df=pd.DataFrame(dataset)
        df.head()
```

Out[3]:

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
0	1.0	1.0	2.0	8253.0	5.0	1.0	113.0	2.0	0.0	0.0	0.0	22.1	0.0	22.1	0.0	10.0	1.0	3.0
1	1.0	1.0	2.0	8258.0	5.0	1.0	113.0	2.0	308.0	498.0	190.0	37.6	0.0	37.6	0.0	15.0	1.0	8.0
2	1.0	1.0	2.0	8253.0	5.0	1.0	113.0	0.0	126.0	135.0	9.0	20.3	0.0	20.3	0.0	3.0	1.0	11.0
3	1.0	1.0	2.0	8256.0	5.0	1.0	113.0	0.0	124.0	126.0	2.0	7.4	0.0	7.4	0.0	1.0	1.0	11.0
4	1.0	6.0	2.0	8158.0	5.0	1.0	113.0	0.0	154.0	154.0	0.0	3.7	0.0	3.7	0.0	2.0	1.0	2.0

Fuente: Elaboración propia

La separación de variables de entrada y salida se realizan de la siguiente forma: Como variables de entrada para la red neuronal son los 17 primeros campos de la base de datos, y el último campo del conjunto de datos se considera como variables de salida, porque son las clases a predecir.

Figura IV.10:

Selección de variables de entrada(X) y variables de salida(Y)

```
In [4]: X = dataset[:,0:17].astype(str)
        pd.DataFrame(X).head()
```

Out[4]:

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
0	1.0	1.0	2.0	8253.0	5.0	1.0	113.0	2.0	0.0	0.0	0.0	22.1	0.0	22.1	0.0	10.0	1.0
1	1.0	1.0	2.0	8258.0	5.0	1.0	113.0	2.0	308.0	498.0	190.0	37.6	0.0	37.6	0.0	15.0	1.0
2	1.0	1.0	2.0	8253.0	5.0	1.0	113.0	0.0	126.0	135.0	9.0	20.3	0.0	20.3	0.0	3.0	1.0
3	1.0	1.0	2.0	8256.0	5.0	1.0	113.0	0.0	124.0	126.0	2.0	7.4	0.0	7.4	0.0	1.0	1.0
4	1.0	6.0	2.0	8158.0	5.0	1.0	113.0	0.0	154.0	154.0	0.0	3.7	0.0	3.7	0.0	2.0	1.0

```
In [5]: Y = dataset[:,17]
        pd.DataFrame(Y).head()
```

Out[5]:

	0
0	3.0
1	8.0
2	11.0
3	11.0
4	2.0

Fuente: Elaboración propia

Se utilizó los argumentos en la construcción de la clase KerasClassifier que pa-

san por la función fit() para así, internamente, entrenar la red neuronal. Pasamos el número de épocas como 200 y el tamaño de lote como 5 para el entrenamiento. La depuración también se desactiva durante en el proceso, estableciendo verbose en 0.

A continuación en la Figura IV.11 se muestra una función que creará una red neuronal simple para el problema de clasificación. Crea una red simple completamente conectada con una capa oculta que contiene 8 neuronas.

La capa oculta utiliza una función de activación de rectificador, lo cual es una buena práctica. Debido a que usamos una codificación one-hot para nuestro conjunto de datos, la capa de salida debe crear 8 valores, uno para cada clase. El valor de salida más grande se tomará como la clase predicha por el modelo.

Figura IV.11:

Modelo de Red Neuronal Simple

```
In [8]: def baseline_model():
            # create model
            model = Sequential()
            model.add(Dense(8, input_dim=19, activation='relu'))
            model.add(Dense(8, activation='softmax'))
            # Compile model
            model.compile(loss='categorical_crossentropy', optimizer='adam', metrics=['accuracy'])
            return model
```

Fuente: Elaboración propia

Scikit-learn tiene una excelente capacidad para evaluar modelos utilizando un conjunto de técnicas. El estándar para evaluar modelos de aprendizaje automático es la validación cruzada de k-fold, como se puede apreciar en la Figura IV.12, se llega apreciar que alcanza un 78 % de precisión.

Figura IV.12:

Resultado de predicción de con Red Neuronal Simple

```
In [9]: estimator = KerasClassifier(build_fn=baseline_model, epochs=200, batch_size=5, verbose=0)
        kfold = KFold(n_splits=10, shuffle=True)
        results = cross_val_score(estimator, X, Y, cv=kfold)

In [10]: print("Baseline: %.2f%% (%.2f%%)" % (results.mean()*100, results.std()*100))
         Baseline: 70.83% (20.73%)
```

Fuente: Elaboración propia

4.4.3. Random forest

Random Forest es un algoritmo de aprendizaje de máquina, que el procesamiento consta en construir un bosque de árboles de forma aleatoria para la clasificación, regresión y otras tareas que cuya salida se expresa en clases.

El bosque aleatorio consta de una gran cantidad de árboles de decisión individuales que operan como un conjunto. Cada árbol individual en el bosque aleatorio realiza una predicción de clases y la clase con más votos se convierte en la predicción de nuestro modelo.

Por lo tanto, se toma en cuenta en la presente investigación que aborda el problema de clasificar el tipo de consumo irregular de agua potable por parte de sus usuario de le empresa EPS Ilo.

A continuación se presenta los resultados obtenidos del conocido algoritmo Random Forest, en la Figura IV.13 se muestra el conjunto de datos a procesar con el algoritmo, el cual no varia en absoluto a los datos procesados por otro métodos.

Figura IV.13:

Preparación de dataset para Random Forest

```
df = pd.read_csv("/content/dic_2020.csv")
df.head(n=5)
```

	SUC	SEC	URB	CALLE	ACT	TS	CAT	TP	LECTA	LECTU	CSM	IMPF	IMPR	IMPC	IMPD	LECTP	EM	EL
0	1	1	2	8253	5	1	113	2	0	0	0	22.1	0.0	22.1	0.0	10	1	3
1	1	1	2	8258	5	1	113	2	308	498	190	37.6	0.0	37.6	0.0	15	1	8
2	1	1	2	8253	5	1	113	0	126	135	9	20.3	0.0	20.3	0.0	3	1	11
3	1	1	2	8256	5	1	113	0	124	126	2	7.4	0.0	7.4	0.0	1	1	11
4	1	6	2	8158	5	1	113	0	154	154	0	3.7	0.0	3.7	0.0	2	1	2

Fuente: Elaboración propia

A continuación, definimos los parámetros para el entrenamiento de Random Forest de la siguiente manera:

- n_estimators: este es el número de árboles en la clasificación de bosque aleatoria.
- criterion: es la función de pérdida utilizada para medir la calidad de la división. Existe dos opciones disponibles en sklearn: gini y entropy. En este caso se usó la entropía.
- random_state: esta es la semilla utilizada por el generador de estado aleatorio para aleatorizar el conjunto de datos.

En la Figura IV.14 se observa el código de promoción del algoritmo Random Forest.

Figura IV.14:

Código de algoritmo Random Rorest en Python

```
[ ]  1 # Feature Scaling
     2 scaler = StandardScaler()
     3 X_train = scaler.fit_transform(X_train)
     4 X_test = scaler.transform(X_test)
```

```
[ ]  1 # Fitting Random Forest Classification to the Training set
     2 classifier = RandomForestClassifier(n_estimators = 10, criterion = 'entropy', random_state = 42)
     3 classifier.fit(X_train, y_train)
```

```
RandomForestClassifier(bootstrap=True, ccp_alpha=0.0, class_weight=None,
                       criterion='entropy', max_depth=None, max_features='auto',
                       max_leaf_nodes=None, max_samples=None,
                       min_impurity_decrease=0.0, min_impurity_split=None,
                       min_samples_leaf=1, min_samples_split=2,
                       min_weight_fraction_leaf=0.0, n_estimators=10,
                       n_jobs=None, oob_score=False, random_state=42, verbose=0,
                       warm_start=False)
```

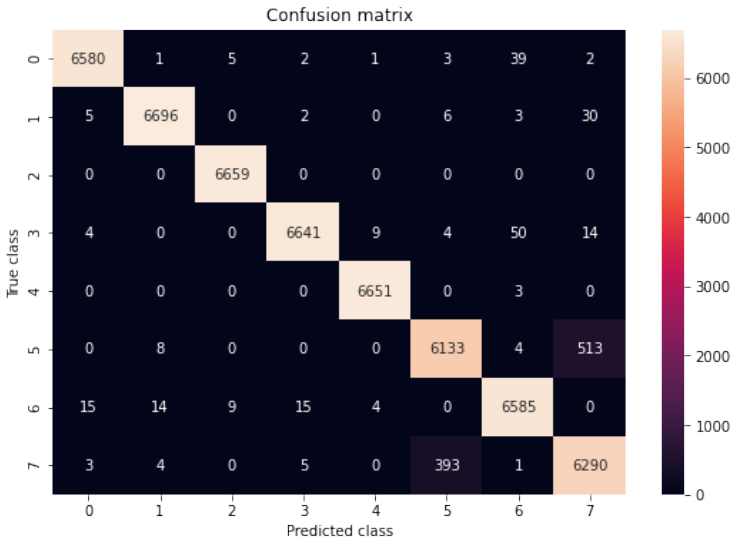
Fuente: Elaboración propia

El resultado de la clasificación del conjunto de datos se puede apreciar en la Figura IV.15, en donde demuestra que la predicción de clases tienen mayor asertividad que otros métodos o algoritmos presentados en la presente investigación.

A comparación con el método Árbol de Decisión, se aprecia que no existe un clase sin clasificar como tal. En el método anteriormente mencionado, existe la predicción nula de la clase 7, sin embargo, aquí se aprecia que no existe clases sin predicciones; también, se puede observar que aún sigue existiendo la mala clasificación de algunos registros pero que es de menor porcentaje.

Figura IV.15:

Matriz de confusión de resultados con Random Forest



Fuente: Elaboración propia

En cuanto a la precisión de predicción de clasificación, se muestra en la Figura IV.15 que existe una precisión muy alta en las 7 primeras clases, sin embargo, se denota un leve falla en los 3 últimos.

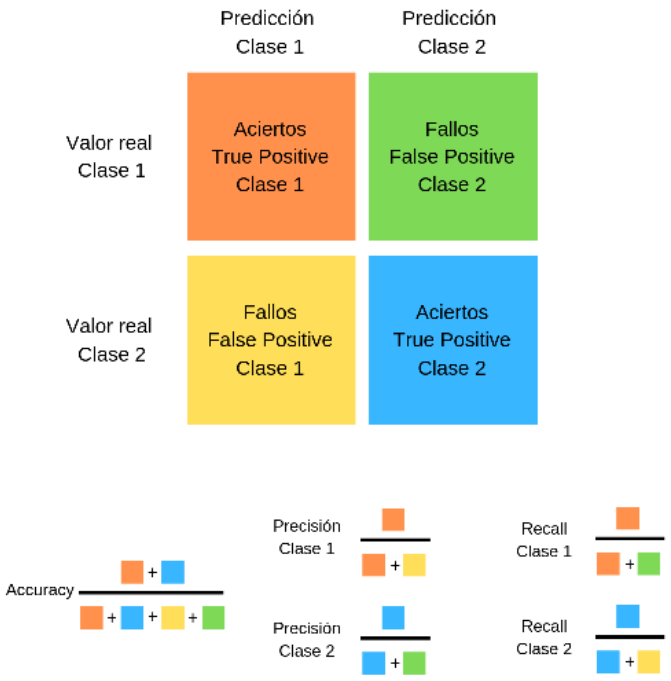
4.5. Evaluación

En esta sección se evalúan lo resultados de predicción de clases de los algoritmos experimentados en la presente investigación.

Para interpretar correctamente los resultados, se utilizó el método Matriz de Confusión que ayuda a comprender las salidas de los algoritmos experimentados. En la Figura IV.16 se observa una ilustración sobre el método de cálculo de la precisión de predicción de las clases (Perros y Gatos).

Figura IV.16:

Ejemplo de Matriz de Confusión



Fuente: Mikel Galar, E. B., Alberto Fernandez. (s.f.)

- Accuracy del modelo representa al número total de predicciones correctas dividido por el número total de predicciones.
- La Precisión de una clase define cuan confiable es un modelo en responder si un registro pertenece a esa clase.
- El Recall de una clase expresa cuan bien puede el modelo detectar a esa clase.
- El F1 Score de una clase es dada por la media armónica de precisión y recall (2 x $precision$ x $recall/(precision + recall)$) que combina precisión y recall en una sola métrica.

En la figura IV.17 se muestran los resultados de predicción de clases procesadas por el algoritmo Árbol de Decisiones, en donde se aprecia una precisión de 82 %, sin

embargo también se detalla la precisión alcanzado por cada clase y su respectiva confianza.

Figura IV.17:

Matriz de confusión de resultado del algoritmo de Árbol de Decisiones

	precision	recall	f1-score	support
0	0.97	1.00	0.98	20027
1	0.70	0.86	0.77	20028
2	0.87	0.96	0.91	20027
3	0.98	0.99	0.98	20027
4	0.69	0.85	0.76	20027
5	0.96	0.98	0.97	20028
6	0.68	0.93	0.78	20027
7	0.00	0.00	0.00	20027
accuracy			0.82	160218
macro avg	0.73	0.82	0.77	160218
weighted avg	0.73	0.82	0.77	160218

Fuente: elaboración propia

A continuación, presentamos los resultados obtenidos al procesar el conjunto de datos por el algoritmo Red Neuronal Simple, en este caso usamos el método de validación denominada Validación Cruzada.

La validación cruzada es un procedimiento de remuestreo que se utiliza para evaluar modelos de aprendizaje automático en una muestra de datos limitada.

El procedimiento tiene un solo parámetro llamado k que se refiere al número de grupos en los que se dividirá una muestra de datos determinada. Como tal, el procedimiento a menudo se denomina validación cruzada de k veces.

En la Figura IV.18 se puede apreciar que la precisión del algoritmo Red Neuronal Simple, llega al 70.83 %. de precisión en la predicción de clases.

Figura IV.18:

Resultado de predicción de Red Neuronal Simple

```
In [9]: estimator = KerasClassifier(build_fn=baseline_model, epochs=200, batch_size=5, verbose=0)
        kfold = KFold(n_splits=10, shuffle=True)
        results = cross_val_score(estimator, X, Y, cv=kfold)

In [10]: print("Baseline: %.2f%% (%.2f%%)" % (results.mean()*100, results.std()*100))
         Baseline: 70.83% (20.73%)
```

Fuente: elaboración propia

Por último, presentamos los resultados del algoritmo Random Forest, en la Figura IV.19 se puede apreciar que tenemos el 98 % de precisión en predicción de clases. Por lo tanto, se determina como el mejor de los algoritmos experimentados en la presente investigación.

Figura IV.19:

Precisión de predicción de clases con Random Forest

	precision	recall	f1-score	support
1	1.00	0.99	0.99	6633
2	1.00	0.99	0.99	6742
3	1.00	1.00	1.00	6659
4	1.00	0.99	0.99	6722
7	1.00	1.00	1.00	6654
8	0.94	0.92	0.93	6658
10	0.99	0.99	0.99	6642
11	0.92	0.94	0.93	6696
accuracy			0.98	53406
macro avg	0.98	0.98	0.98	53406
weighted avg	0.98	0.98	0.98	53406

Fuente: elaboración propia

En esta sección se resume los resultados obtenidos en el procesamiento de datos con los algoritmos mencionados y experimentados en el presente estudio.

En la Tabla IV.3 se muestran los algoritmos con más alta precisión, Random Forest con 98 %, seguido por Árbol de Decisión con 82 % y finalmente se encuentra el algoritmo Red Neuronal Simple con 70.83 %.

También se muestra el Recall, lo que indica que tan bien puede detectar el modelo las clases correspondientes, aquí se tiene a Random Forest con 98 % y

Árbol de Decisión con 82 %.

F1-score combina las medidas de Precisión y Recall en un solo valor para comparar su rendimiento, Random Forest posee el 98 % y Árbol de Decisión el 77 %.

En cuanto a Red Neuronal Simple, no se muestra el Recall ni F1-score, puesto que utiliza otro método de validación denominada Validación Cruzada, el cual es el más indicado para este caso.

Cuadro IV.3:

Análisis de los resultados

Número	Algoritmos	Precisión	Recall	F1-score
1	Random Forest	98 %	98 %	98 %
2	Árbol de Decisión	82 %	82 %	77 %
3	Red Neuronal Simple	70.83 %	-	-

Fuente: elaboración propia

Capítulo V

Conclusiones

Los métodos de Machine Learning han resultado ser muy útiles en la predicción de consumo irregular del servicio de agua potable por los usuarios de la Empresa Prestadora de Servicios de Saneamiento S.A. en la provincia de Ilo.

La metodología CRISP-DM ha resultado ser muy importante para llevar a cabo la experimentación de todas las etapas de machine learning, para alcanzar el objetivo de predicción de consumo irregular de agua potable con diferentes métodos de Machine Learning.

En el presente trabajo se determina a Random Forest como mejor método de predicción para la predicción de consumo irregular del servicio de agua potable, ya que alcanzó una precisión del 98 %.

Capítulo VI

Recomendaciones

Se recomienda a las empresas locales establecer un plan de recopilación de información histórica de los procesos de la empresa encargada, porque esta estrategia será muy importante en un futuro cercano, ya que los algoritmos de Aprendizaje Supervisado necesitan datos históricos en cantidad y estructurados.

Realizar un estudio de Agrupación y Clustering para agrupar usuarios de condiciones o características muy similares para tener identificado qué problemas podrían tener los usuarios de una determinado grupo, que podría ser por la ubicación geográfica, tipos de usuarios, condición socio económica, etc.

Se recomienda plantear un proyecto de investigación orientado al procesamiento de quejas o reclamos escritos por los usuarios para identificar los tipos o clases, este caso podría ser tratado con la disciplina de Procesamiento Lenguaje Natural, porque generalmente las quejas son formulados en el lenguaje natural (español). Este en conjunto con la presente investigación podría ayudar en canalizar los problemas más importantes y urgentes a tomar en cuenta para dar una solución.

Referencias

Breiman, L. (2001). Random forests. *University of California, Berkeley, CA 94720*.

Chauhan, N. S. (2020). *Decision tree algorithm*. https://www.kdnuggets.com/2020/01/.

Correo, D. (2019). Reclamos por facturación de agua subieron en un 60 %.

Cruz, P. P. (2010). *Inteligencia artificial con aplicaciones a la ingeniería* (Vol. 1). Alfaomega Grupo Editor, S.A. de C.V.

Cánovas-García, F., Alonso, F., y Castillo, F. (2016, 07). Modificación del algoritmo random forest para su empleo en clasificación de imágenes de teledetección. En (p. 359-368).

Davy Cielen, M. A., Arno D. B. Meysman. (2016). *Introducing data science*. Manning Publications Co.

Drew Conway, J. M. (2012). *Machine learning for hackers*. Oreilly.

EPS-Ilo-S.A. (2020a). *Empresa prestadora de servicios de saneamiento ilo s.a.* https://www.epsilo.com.pe/.

EPS-Ilo-S.A. (2020b). *Manual de organizaciones y funciones (mof)*.

Ertel, W. (2017). *Introduction to artificial intelligence* (2.ª ed.). Springer International Publishing AG.

Espinoza, V. P. (2014). Modelo de detección de fraude en clientes del servicio de agua potable de una empresa sanitaria. *Universidad de Chile*.

Joshi, A. V. (2020). *Machine learning and artificial intelligence* (1st ed. 2020 ed.). Springer International Publishing. Descargado de http://gen.lib.rus

.ec/book/index.php?md5=ff6757393a91c4bdb249d134a9ed95dc

José Caridad y Ocerin, R. D. P., Genoveva Millán Vázquez de la Torre. (2001). Predicción del consumo de agua en córdoba. *Ingeniería del agua, 08*.

Judith Hurwitz, D. K. (2018). *Machine learning for dummies*. John Wiley & Sons, Inc.

Kantardzic, M. (2020). *Data mining,concepts, models,methods, and algorithms,* (2.ª ed.). simultaneously in Canada.

Kotu, V., y Deshpande, B. (2019). Chapter 1 - introduction. En V. Kotu y B. Deshpande (Eds.), *Data science (second edition)* (Second Edition ed., p. 1 - 18). Morgan Kaufmann. Descargado de http://www.sciencedirect.com/science/article/pii/B9780128147610000010 doi: https://doi.org/10.1016/B978-0-12-814761-0.00001-0

Machine Learning Mastery. (s.f.). https://machinelearningmastery.com/regression-tutorial-keras-deep-learning-library-python/. (Accessed: 2020-08-16)

Mikel Galar, E. B., Alberto Fernandez. (s.f.). *A review on ensembles for the class imbalance problem: Bagging, boosting, and hybrid-based approaches*.

Muhammad Waseem Ahmad, Y. R., Monjur Mourshed. (2017). Trees vs neurons: Comparison between random forest and ann for high-resolution prediction of building energy consumption. *Elsevier-Energy and Buildings is an international journal, 147*.

Nagy, Z. (2018). *Artificial intelligence andmachine learningfundamentals*. Packt Publishing.

Peter A. Whigham, S. G. M., Caitlin A. Owen. (2015). A baseline model for software effort estimation. *ACM Transactions on Software Engineering and Methodology, 24*(20).

Quispe, J. D. L. (2017). Modelo univariante para el consumo doméstico mensual de agua potable en el distrito de ilave - emsa puno, periodo 2002 - 2013. *Universidad Nacional del Altiplano*.

Rodica Branzei, S. T. a., Dinko Dimitrov. (2005). *Models in cooperative game theory: Crisp, fuzzy, and multi-choice games* (1.ª ed.). Springer Berlin Heidelberg. Descargado de http://gen.lib.rus.ec/book/index.php?md5=c6bd44d6ad733e6b03ad37b9a946bac1

Rosales, A. J. C. (2019). Sistema inteligente basado en machine learning para la detección de fraude de facturación de agua potable. *Universidad Nacional Mayor de San Marcos*.

Salcedo, M. L. (2015). Sistema dialog 3g y su influencia del en la optimizacion del proceso de lectura y facturacion de sedapal, caso condominio nuevo cercado del distrito de lima. *Universidad Tecnológica del Perú*.

Uno, R. (2019). Pobladores reclaman por cobros excesivos en eps-moquegua.

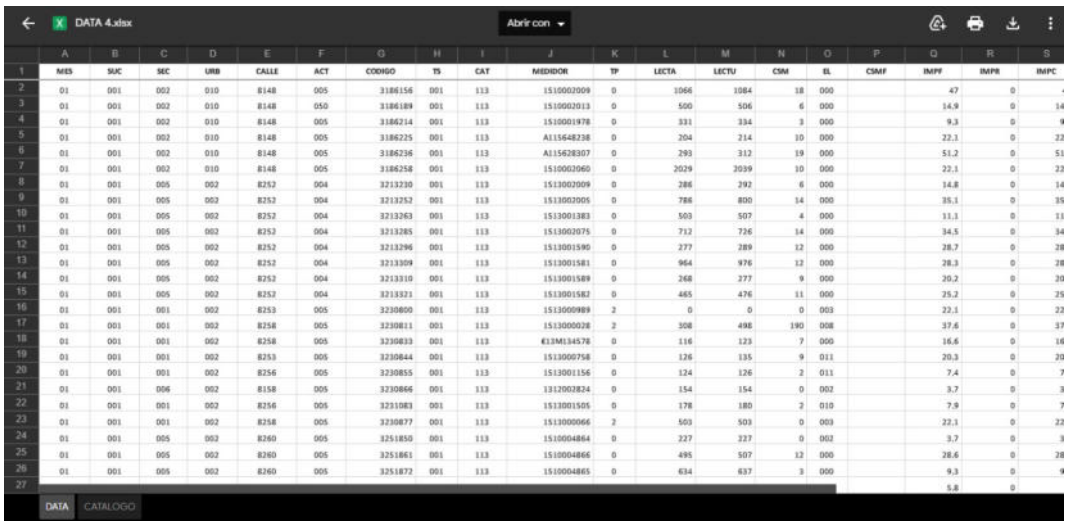
Vera Letamendi, M. M. (2018). Minería de datos aplicada a la detección de patrones de reclamos del servicio de agua potable en la ciudad de guayaquil. *Universidad Católica de Santiago de Guayaquil*.

Capítulo VII

Anexos

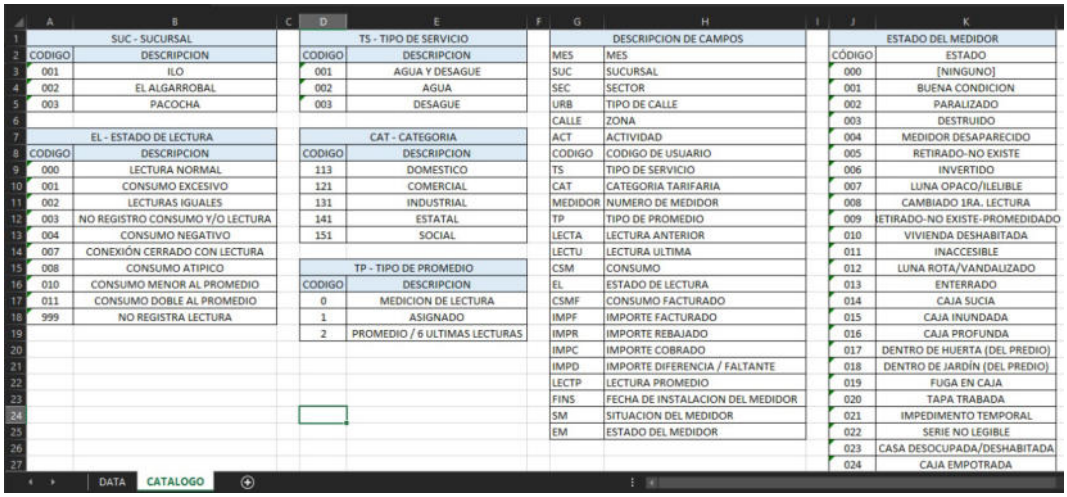
Datos de la investigación

A continuación, se adjunta la base de datos de los usuarios con sus respectivos campos, tales como ubicación, importes, consumos, lecturas, entre otros; cada uno de estos registros contiene un código de identificación. Por motivos de confidencialidad de datos e información de la empresa, no es posible detallar todos los registros existentes.



Enseguida se aprecia el catálogo de datos que describe el significado de cada registro y cada campo en el dataset para tener una mejor comprensión. Esto ha sido parte de la etapa de comprensión de datos en el proceso de desarrollo de

investigación.



SUC - SUCURSAL

CODIGO	DESCRIPCION
001	ILO
002	EL ALGARROBAL
003	PACOCHA

EL - ESTADO DE LECTURA

CODIGO	DESCRIPCION
000	LECTURA NORMAL
001	CONSUMO EXCESIVO
002	LECTURAS IGUALES
003	NO REGISTRO CONSUMO Y/O LECTURA
004	CONSUMO NEGATIVO
007	CONEXIÓN CERRADO CON LECTURA
008	CONSUMO ATIPICO
010	CONSUMO MENOR AL PROMEDIO
011	CONSUMO DOBLE AL PROMEDIO
999	NO REGISTRA LECTURA

TS - TIPO DE SERVICIO

CODIGO	DESCRIPCION
001	AGUA Y DESAGUE
002	AGUA
003	DESAGUE

CAT - CATEGORIA

CODIGO	DESCRIPCION
113	DOMESTICO
121	COMERCIAL
131	INDUSTRIAL
141	ESTATAL
151	SOCIAL

TP - TIPO DE PROMEDIO

CODIGO	DESCRIPCION
0	MEDICION DE LECTURA
1	ASIGNADO
2	PROMEDIO / 6 ULTIMAS LECTURAS

DESCRIPCION DE CAMPOS

MES	MES
SUC	SUCURSAL
SEC	SECTOR
URB	TIPO DE CALLE
CALLE	ZONA
ACT	ACTIVIDAD
CODIGO	CODIGO DE USUARIO
TS	TIPO DE SERVICIO
CAT	CATEGORIA TARIFARIA
MEDIDOR	NUMERO DE MEDIDOR
TP	TIPO DE PROMEDIO
LECTA	LECTURA ANTERIOR
LECTU	LECTURA ULTIMA
CSM	CONSUMO
EL	ESTADO DE LECTURA
CSMF	CONSUMO FACTURADO
IMPF	IMPORTE FACTURADO
IMPR	IMPORTE REBAJADO
IMPC	IMPORTE COBRADO
IMPD	IMPORTE DIFERENCIA / FALTANTE
LECTP	LECTURA PROMEDIO
FINS	FECHA DE INSTALACION DEL MEDIDOR
SM	SITUACION DEL MEDIDOR
EM	ESTADO DEL MEDIDOR

ESTADO DEL MEDIDOR

CÓDIGO	ESTADO
000	[NINGUNO]
001	BUENA CONDICION
002	PARALIZADO
003	DESTRUIDO
004	MEDIDOR DESAPARECIDO
005	RETIRADO-NO EXISTE
006	INVERTIDO
007	LUNA OPACO/ILELIBLE
008	CAMBIADO 1RA. LECTURA
009	ETIRADO-NO EXISTE-PROMEDIDADO
010	VIVIENDA DESHABITADA
011	INACCESIBLE
012	LUNA ROTA/VANDALIZADO
013	ENTERRADO
014	CAJA SUCIA
015	CAJA INUNDADA
016	CAJA PROFUNDA
017	DENTRO DE HUERTA (DEL PREDIO)
018	DENTRO DE JARDÍN (DEL PREDIO)
019	FUGA EN CAJA
020	TAPA TRABADA
021	IMPEDIMENTO TEMPORAL
022	SERIE NO LEGIBLE
023	CASA DESOCUPADA/DESHABITADA
024	CAJA EMPOTRADA

DATA CATALOGO

Se anexa también el código fuente del algoritmo Árbol de decisiones

```
  import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns

from sklearn.metrics import confusion_matrix
from sklearn.metrics import classification_report
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.decomposition import PCA
from sklearn.tree import DecisionTreeClassifier

from pylab import rcParams

from imblearn.under_sampling import NearMiss
from imblearn.over_sampling import RandomOverSampler
from imblearn.combine import SMOTETomek
from imblearn.ensemble import BalancedBaggingClassifier

from collections import Counter
```

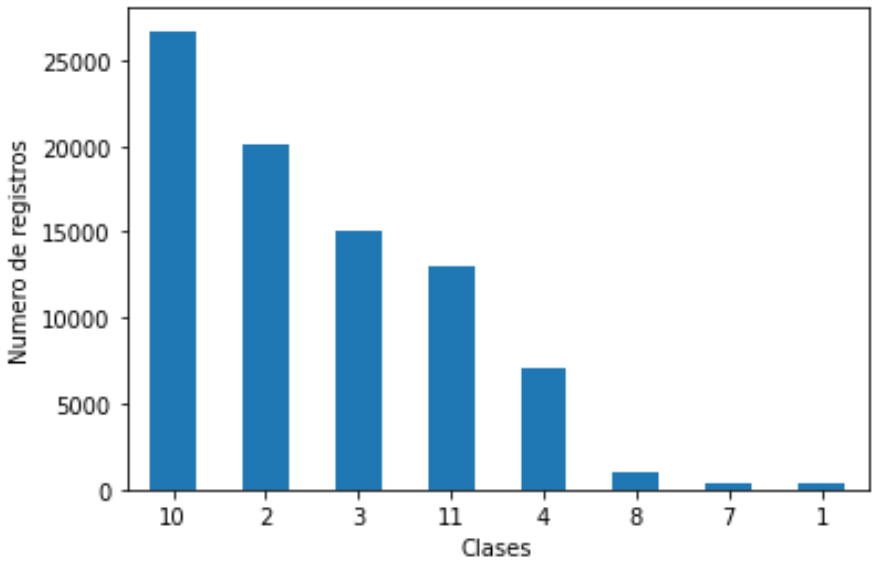
```
df = pd.read_csv("/content/dic_2020.csv")
df.head(n=5)
```

	SUC	SEC	URB	CALLE	ACT	TS	CAT	TP	LECTA	LECTU	CSM	IMPF	IMPR	IMPC	IMPD	LECTP	EM	EL
0	1	1	2	8253	5	1	113	2	0	0	0	22.1	0.0	22.1	0.0	10	1	3
1	1	1	2	8258	5	1	113	2	308	498	190	37.6	0.0	37.6	0.0	15	1	8
2	1	1	2	8253	5	1	113	0	126	135	9	20.3	0.0	20.3	0.0	3	1	11
3	1	1	2	8256	5	1	113	0	124	126	2	7.4	0.0	7.4	0.0	1	1	11
4	1	6	2	8158	5	1	113	0	154	154	0	3.7	0.0	3.7	0.0	2	1	2

```
print(df.shape)
print(pd.value_counts(df['EL'], sort = True))
```

```
(83714, 18)
10    26703
2     20140
3     15108
11    12934
4      7095
8      1017
7       404
1       313
Name: EL, dtype: int64
```

```
count_classes = pd.value_counts(df['EL'], sort = True)
count_classes.plot(kind = 'bar', rot=0)
plt.xticks(range(8))
#plt.title("Numero de registros por clase")
plt.xlabel("Clases")
plt.ylabel("Numero de registros");
```



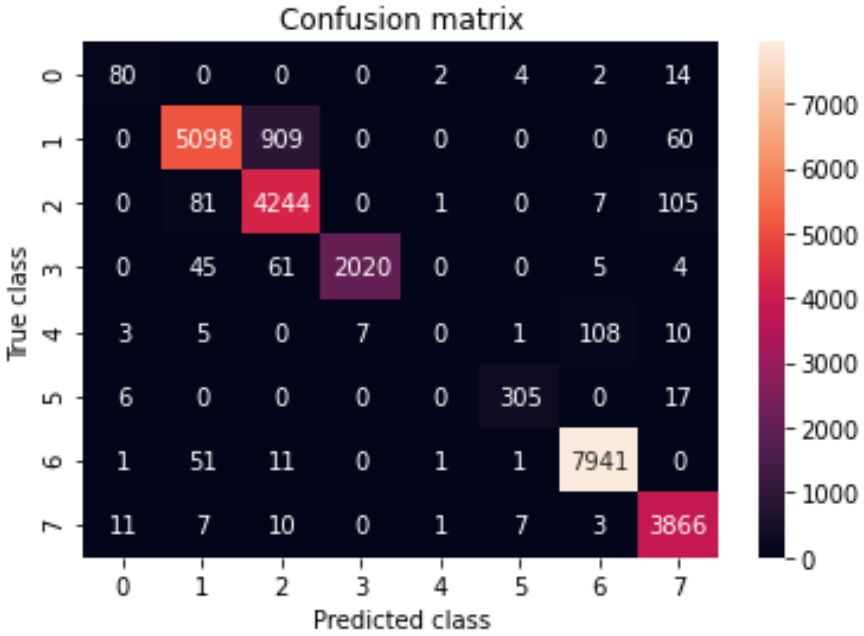
```
[ ]  y = df['EL']
     X = df.drop('EL', axis=1)
     #dividimos en sets de entrenamiento y test
     X_train, X_test, y_train, y_test = train_test_split(X, y, train_size=0.7)
```

```
     #creamos una función que crea el modelo que usaremos cada vez
     def run_model(X_train, X_test, y_train, y_test):
         clf_base = LogisticRegression(C=1.0,penalty='l2',random_state=1,solver="newton-cg")
         clf_base.fit(X_train, y_train)
         return clf_base

     #ejecutamos el modelo "tal cual"
     model = run_model(X_train, X_test, y_train, y_test)

     #definimos funciona para mostrar los resultados
     def mostrar_resultados(y_test, pred_y):
         conf_matrix = confusion_matrix(y_test, pred_y)
         plt.figure(figsize=(6, 4))
         sns.heatmap(conf_matrix, annot=True, fmt="d");
         plt.title("Confusion matrix")
         plt.ylabel('True class')
         plt.xlabel('Predicted class')
         plt.show()
         print (classification_report(y_test, pred_y))

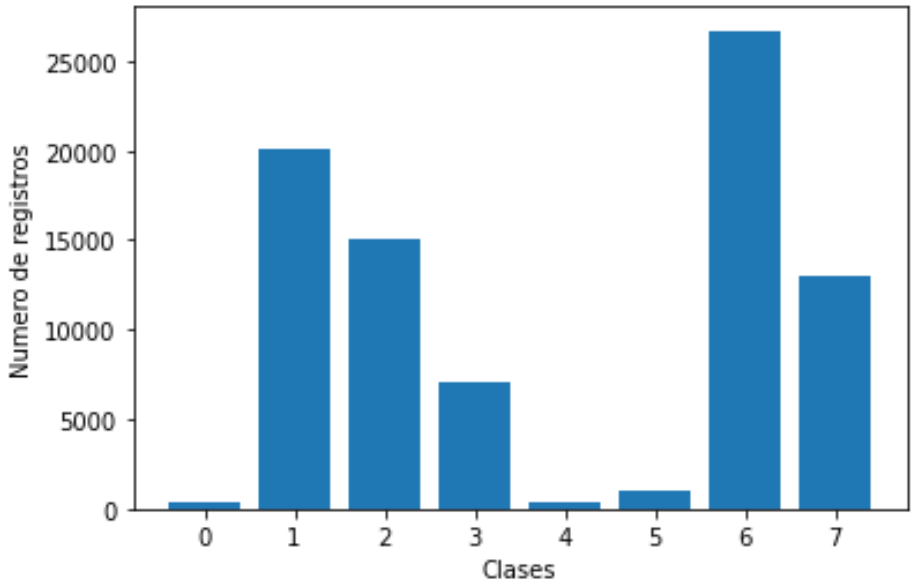
     pred_y = model.predict(X_test)
     mostrar_resultados(y_test, pred_y)
```



	precision	recall	f1-score	support
1	0.79	0.78	0.79	102
2	0.96	0.84	0.90	6067
3	0.81	0.96	0.88	4438
4	1.00	0.95	0.97	2135
7	0.00	0.00	0.00	134
8	0.96	0.93	0.94	328
10	0.98	0.99	0.99	8006
11	0.95	0.99	0.97	3905
accuracy			0.94	25115
macro avg	0.81	0.80	0.80	25115
weighted avg	0.94	0.94	0.94	25115

```
[ ]
    # load and summarize the dataset
    from pandas import read_csv
    from collections import Counter
    from matplotlib import pyplot
    from sklearn.preprocessing import LabelEncoder
    # define the dataset location
    url = '/content/dic_2020_f.csv'
    # load the csv file as a data frame
    df = read_csv(url, header=None)
    data = df.values
    # split into input and output elements
    X, y = data[:, :-1], data[:, -1]
    # label encode the target variable
    y = LabelEncoder().fit_transform(y)
    # summarize distribution
    counter = Counter(y)
    for k,v in counter.items():
      per = v / len(y) * 100
      print('Class=%d, n=%d (%.3f%%)' % (k, v, per))
    # plot the distribution
    pyplot.bar(counter.keys(), counter.values())
    plt.xlabel("Clases")
    plt.ylabel("Numero de registros");
    pyplot.show()
```

```
Class=2, n=15108 (18.047%)
Class=5, n=1017 (1.215%)
Class=7, n=12934 (15.450%)
Class=1, n=20140 (24.058%)
Class=6, n=26703 (31.898%)
Class=4, n=404 (0.483%)
Class=0, n=313 (0.374%)
Class=3, n=7095 (8.475%)
```




```
[ ] # Splitting the dataset into the Training set and Test set
    from sklearn.model_selection import train_test_split
    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.25, stratify=y, random_state = 42)
```

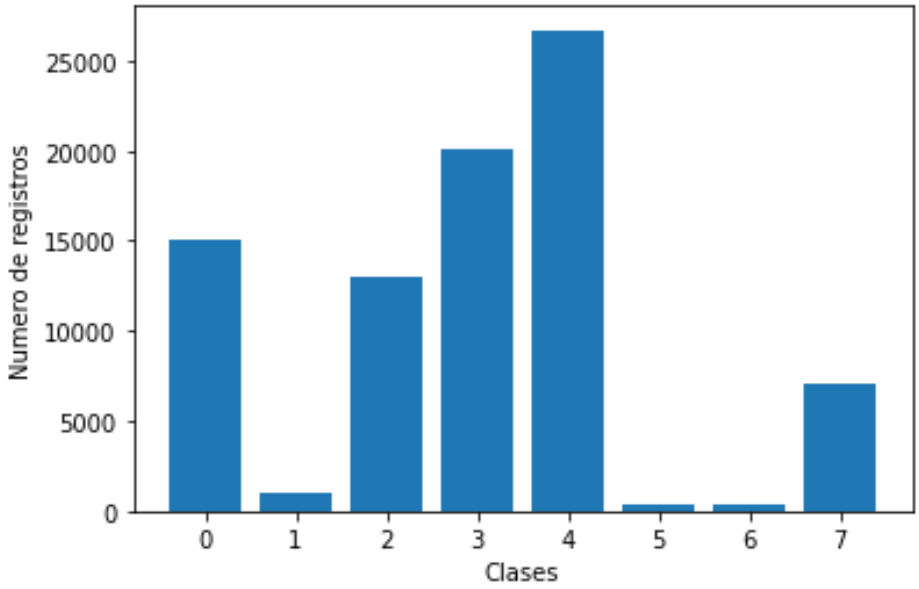
```
[ ] # Fitting Classifier to the Training Set
    from sklearn.tree import DecisionTreeClassifier
    classifier = DecisionTreeClassifier(criterion='entropy',max_depth=3, random_state=42)
    classifier.fit(X_train, y_train)
```

```
DecisionTreeClassifier(ccp_alpha=0.0, class_weight=None, criterion='entropy',
                       max_depth=3, max_features=None, max_leaf_nodes=None,
                       min_impurity_decrease=0.0, min_impurity_split=None,
                       min_samples_leaf=1, min_samples_split=2,
                       min_weight_fraction_leaf=0.0, presort='deprecated',
                       random_state=42, splitter='best')
```

```
[ ] # encoding categorical data e.g. disease outcome as a dummy variable
    y,class_names = pd.factorize(y)
```

```
[ ] counter = Counter(y)
    for k,v in counter.items():
      per = v / len(y) * 100
      print('Class=%d, n=%d (%.3f%%)' % (k, v, per))
    # plot the distribution
    pyplot.bar(counter.keys(), counter.values())
    plt.xlabel("Clases")
    plt.ylabel("Numero de registros");
    pyplot.show()
```

```
Class=0, n=15108 (18.047%)
Class=1, n=1017 (1.215%)
Class=2, n=12934 (15.450%)
Class=3, n=20140 (24.058%)
Class=4, n=26703 (31.898%)
Class=5, n=404 (0.483%)
Class=6, n=313 (0.374%)
Class=7, n=7095 (8.475%)
```




```python
# Model performance on training set
y_pred_train =classifier.predict(X_train)

from sklearn import metrics
from sklearn.metrics import confusion_matrix, classification_report

accuracy = metrics.accuracy_score(y_train, y_pred_train)
print("Accuracy: {:.2f}".format(accuracy))
cm=confusion_matrix(y_train,y_pred_train)
print('Confusion Matrix: \n', cm)
target_names=class_names
print(target_names)
print(classification_report(y_train, y_pred_train))
```

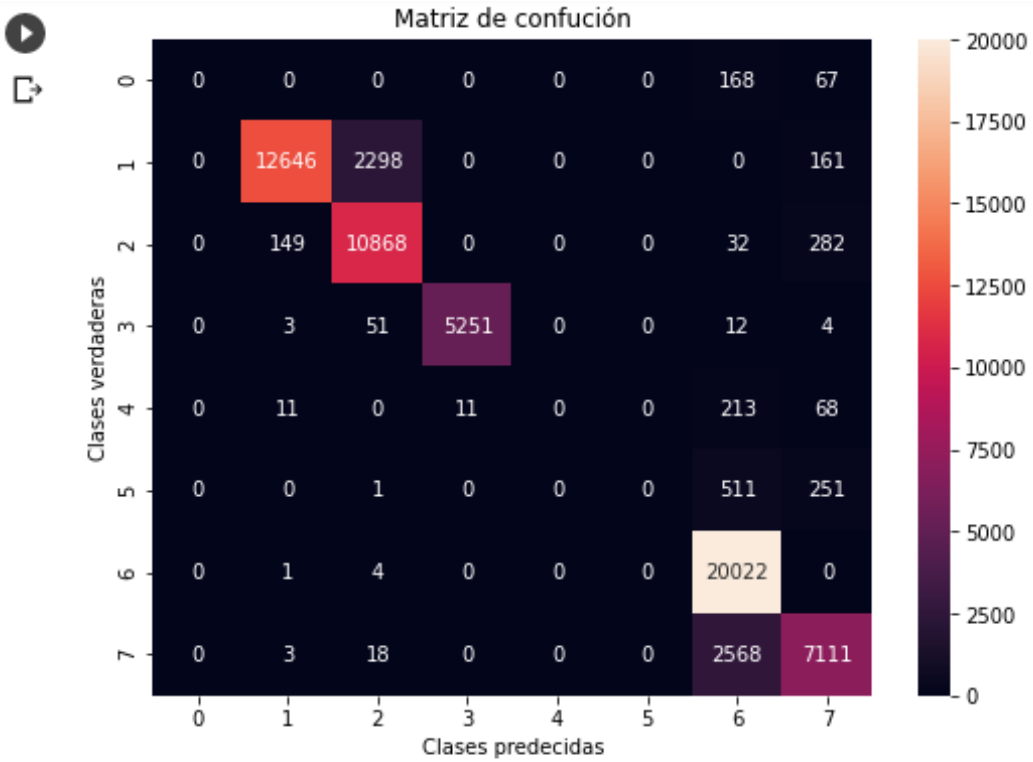
```
Accuracy: 0.89
Confusion Matrix:
 [[    0     0     0     0     0     0   168    67]
 [    0 12646  2298     0     0     0     0   161]
 [    0   149 10868     0     0     0    32   282]
 [    0     3    51  5251     0     0    12     4]
 [    0    11     0    11     0     0   213    68]
 [    0     0     1     0     0     0   511   251]
 [    0     1     4     0     0     0 20022     0]
 [    0     3    18     0     0     0  2568  7111]]
[2 5 7 1 6 4 0 3]
              precision    recall  f1-score   support

           0       0.00      0.00      0.00       235
           1       0.99      0.84      0.91     15105
           2       0.82      0.96      0.88     11331
           3       1.00      0.99      0.99      5321
           4       0.00      0.00      0.00       303
           5       0.00      0.00      0.00       763
           6       0.85      1.00      0.92     20027
           7       0.90      0.73      0.81      9700

    accuracy                           0.89     62785
   macro avg       0.57      0.56      0.56     62785
weighted avg       0.88      0.89      0.88     62785
```

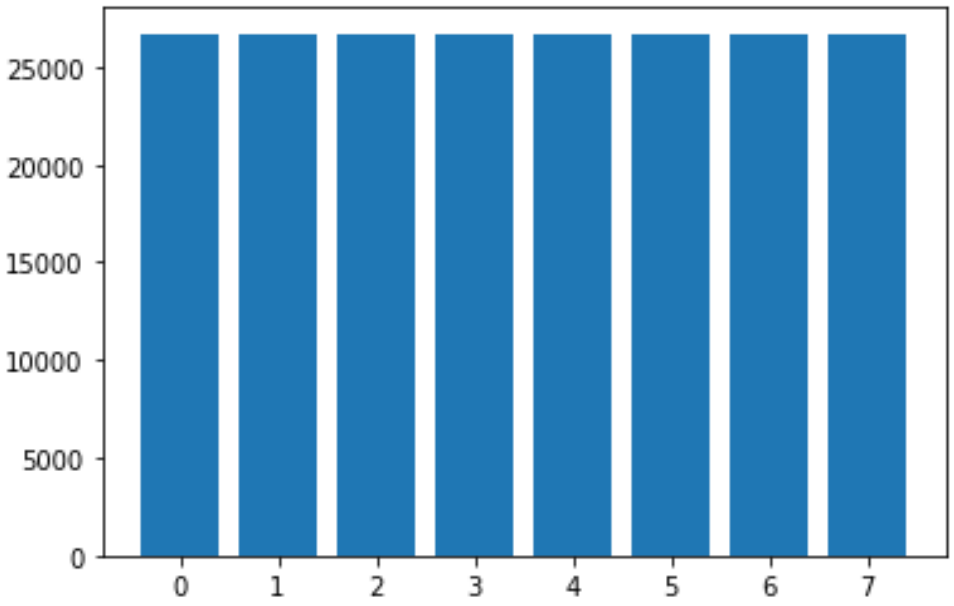
```python
def mostrar_resultados(y_train, y_pred_train):
    conf_matrix = confusion_matrix(y_train, y_pred_train)
    plt.figure(figsize=(8, 6))
    sns.heatmap(conf_matrix, annot=True, fmt="d");
    plt.title("Matriz de confución")
    plt.ylabel('Clases verdaderas')
    plt.xlabel('Clases predecidas')
    plt.show()
    print (classification_report(y_train, y_pred_train))

#pred_y = model.predict(X_test)
mostrar_resultados(y_train, y_pred_train)
```



	precision	recall	f1-score	support
0	0.00	0.00	0.00	235
1	0.99	0.84	0.91	15105
2	0.82	0.96	0.88	11331
3	1.00	0.99	0.99	5321
4	0.00	0.00	0.00	303
5	0.00	0.00	0.00	763
6	0.85	1.00	0.92	20027
7	0.90	0.73	0.81	9700
accuracy			0.89	62785
macro avg	0.57	0.56	0.56	62785
weighted avg	0.88	0.89	0.88	62785

```
# example of oversampling a multi-class classification dataset
from pandas import read_csv
from imblearn.over_sampling import SMOTE
from collections import Counter
from matplotlib import pyplot
from sklearn.preprocessing import LabelEncoder
# define the dataset location
url = '/content/dic_2020_f.csv'
# load the csv file as a data frame
df = read_csv(url, header=None)
data = df.values
# split into input and output elements
A, b = data[:, :-1], data[:, -1]
# label encode the target variable
b = LabelEncoder().fit_transform(b)
# transform the dataset
X=A
y=b
oversample = SMOTE()
X, y = oversample.fit_resample(X, y)
# summarize distribution
counter = Counter(y)
for k,v in counter.items():
  per = v / len(y) * 100
  print('Class=%d, n=%d (%.3f%%)' % (k, v, per))
# plot the distribution
pyplot.bar(counter.keys(), counter.values())
plt.xlabel("Clases")
plt.ylabel("Numero de registros");
pyplot.show()
```



```
[ ] # Splitting the dataset into the Training set and Test set
    from sklearn.model_selection import train_test_split
    X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.25, stratify=y, random_state = 42)
```

```
    # Fitting Classifier to the Training Set
    from sklearn.tree import DecisionTreeClassifier
    classifier = DecisionTreeClassifier(criterion='entropy',max_depth=3, random_state=42)
    classifier.fit(X_train, y_train)
```

```
DecisionTreeClassifier(ccp_alpha=0.0, class_weight=None, criterion='entropy',
                       max_depth=3, max_features=None, max_leaf_nodes=None,
                       min_impurity_decrease=0.0, min_impurity_split=None,
                       min_samples_leaf=1, min_samples_split=2,
                       min_weight_fraction_leaf=0.0, presort='deprecated',
                       random_state=42, splitter='best')
```

```
[ ] # encoding categorical data e.g. disease outcome as a dummy variable
    y,class_names = pd.factorize(y)
```

```
[ ] # Model performance on training set
    y_pred_train =classifier.predict(X_train)

    from sklearn import metrics
    from sklearn.metrics import confusion_matrix, classification_report

    accuracy = metrics.accuracy_score(y_train, y_pred_train)
    print("Accuracy: {:.2f}".format(accuracy))
    cm=confusion_matrix(y_train,y_pred_train)
    print('Confusion Matrix: \n', cm)
    target_names=class_names
    print(target_names)
    print(classification_report(y_train, y_pred_train))
```

```
Accuracy: 0.82
Confusion Matrix:
 [[19937     0     0     0    90     0     0     0]
 [    0 17194  2834     0     0     0     0     0]
 [    0   774 19184     0     9     0    60     0]
 [    0     1   106 19829    56     0    35     0]
 [  143     1     0   357 17203   638  1685     0]
 [  303     0     2     0    39 19643    41     0]
 [  116     1     4     0  1186     4 18716     0]
 [    0  6547    17     0  6236    85  7142     0]]
[2 5 7 1 6 4 0 3]
```

	precision	recall	f1-score	support
0	0.97	1.00	0.98	20027
1	0.70	0.86	0.77	20028
2	0.87	0.96	0.91	20027
3	0.98	0.99	0.99	20027
4	0.69	0.86	0.77	20027
5	0.96	0.98	0.97	20028
6	0.68	0.93	0.78	20027
7	0.00	0.00	0.00	20027
accuracy			0.82	160218
macro avg	0.73	0.82	0.77	160218
weighted avg	0.73	0.82	0.77	160218

Código fuente del algortimo Random Forest

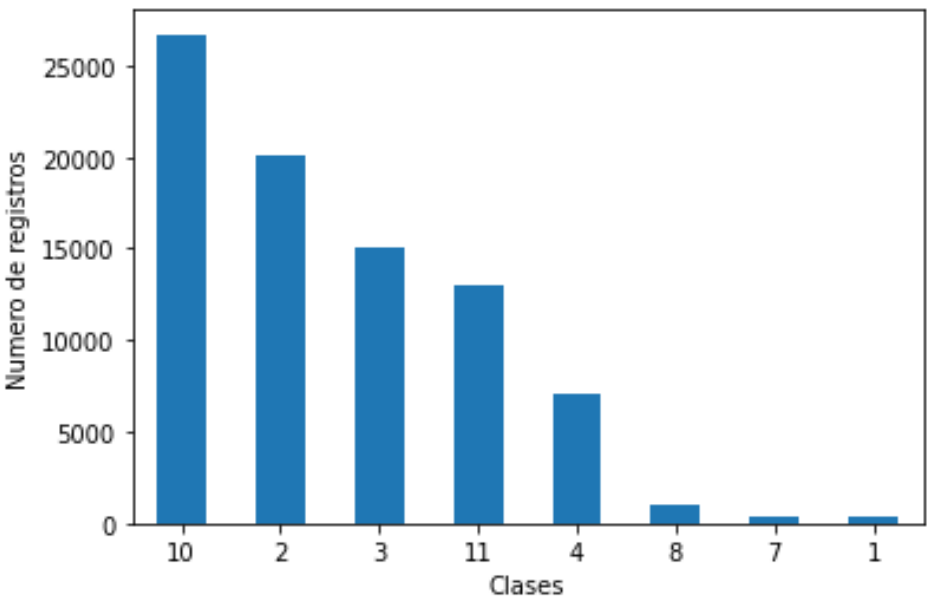
```
[ ] import pandas as pd
    df = pd.read_csv("/content/dic_2020.csv")
    df.head(n=5)
```

	SUC	SEC	URB	CALLE	ACT	TS	CAT	TP	LECTA	LECTU	CSM	IMPF	IMPR	IMPC	IMPD	LECTP	EM	EL
0	1	1	2	8253	5	1	113	2	0	0	0	22.1	0.0	22.1	0.0	10	1	3
1	1	1	2	8258	5	1	113	2	308	498	190	37.6	0.0	37.6	0.0	15	1	8
2	1	1	2	8253	5	1	113	0	126	135	9	20.3	0.0	20.3	0.0	3	1	11
3	1	1	2	8256	5	1	113	0	124	126	2	7.4	0.0	7.4	0.0	1	1	11
4	1	6	2	8158	5	1	113	0	154	154	0	3.7	0.0	3.7	0.0	2	1	2

```
print(df.shape)
print(pd.value_counts(df['EL'], sort = True))
```

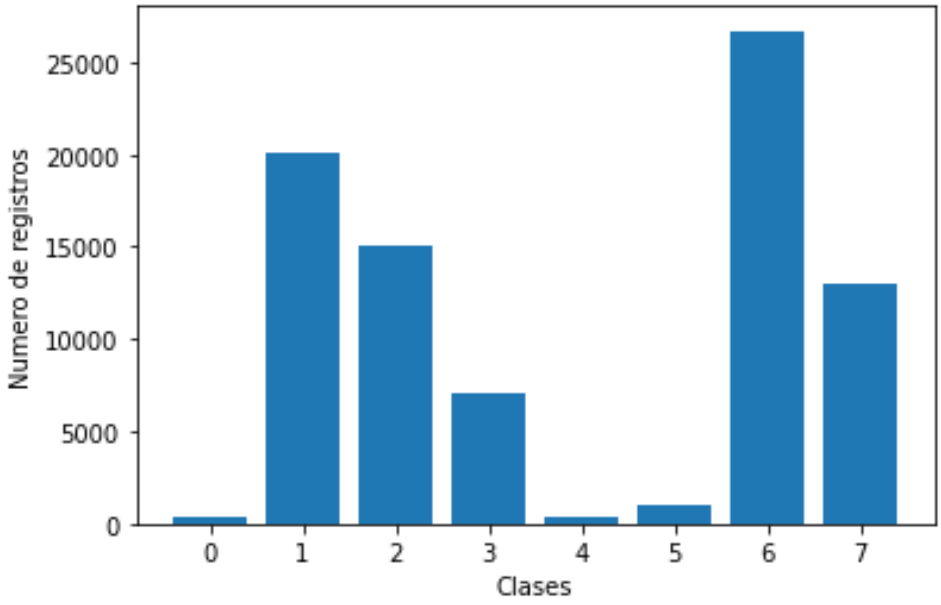
```
(83714, 18)
10    26703
2     20140
3     15108
11    12934
4      7095
8      1017
7       404
1       313
Name: EL, dtype: int64
```

```
[ ] import matplotlib.pyplot as plt
    count_classes = pd.value_counts(df['EL'], sort = True)
    count_classes.plot(kind = 'bar', rot=0)
    #plt.xticks(range(8))
    #plt.title("Numero de registros por clase")
    plt.xlabel("Clases")
    plt.ylabel("Numero de registros");
```


25000
20000
15000
10000
5000
0
Numero de registros
10
2
3
11
4
8
7
1
Clases

```
# load and summarize the dataset
from pandas import read_csv
from collections import Counter
from matplotlib import pyplot
from sklearn.preprocessing import LabelEncoder
# define the dataset location
url = '/content/dic_2020_f.csv'
# load the csv file as a data frame
df = read_csv(url, header=None)
data = df.values
# split into input and output elements
X, y = data[:, :-1], data[:, -1]
# label encode the target variable
y = LabelEncoder().fit_transform(y)
# summarize distribution
counter = Counter(y)
for k,v in counter.items():
  per = v / len(y) * 100
  print('Class=%d, n=%d (%.3f%%)' % (k, v, per))
# plot the distribution
pyplot.bar(counter.keys(), counter.values())
pyplot.xlabel("Clases")
pyplot.ylabel("Numero de registros");
pyplot.show()
```

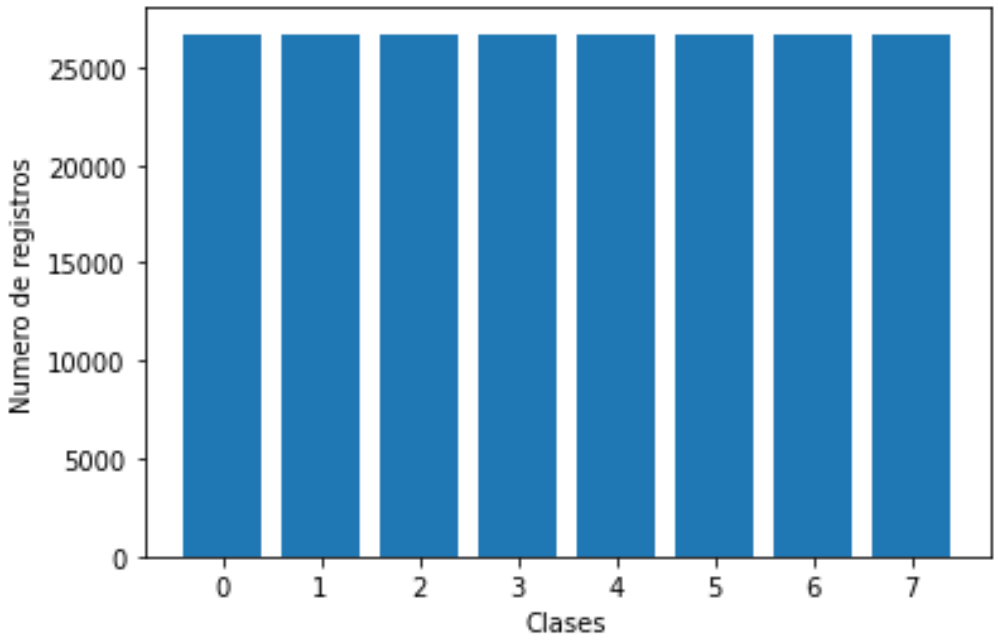
```
Class=2, n=15108 (18.047%)
Class=5, n=1017 (1.215%)
Class=7, n=12934 (15.450%)
Class=1, n=20140 (24.058%)
Class=6, n=26703 (31.898%)
Class=4, n=404 (0.483%)
Class=0, n=313 (0.374%)
Class=3, n=7095 (8.475%)
```




```
[ ]  #Importing Libraries
     import numpy as np
     import pandas as pd
     from sklearn.model_selection import train_test_split
     from sklearn.preprocessing import StandardScaler
     from sklearn.ensemble import RandomForestClassifier
     from sklearn.metrics import confusion_matrix
     from sklearn.externals import joblib
     print('Libraries Imported')
```

```
# example of oversampling a multi-class classification dataset
from pandas import read_csv
from imblearn.over_sampling import SMOTE
from collections import Counter
import matplotlib.pyplot as plt
import matplotlib as plt
from sklearn.preprocessing import LabelEncoder
# define the dataset location
url = '/content/dic_2020_f.csv'
# load the csv file as a data frame
df = read_csv(url, header=None)
data = df.values
# split into input and output elements
A, b = data[:, :-1], data[:, -1]
# label encode the target variable
b = LabelEncoder().fit_transform(b)
# transform the dataset
X=A
y=b
oversample = SMOTE()
X, y = oversample.fit_resample(X, y)
# summarize distribution
counter = Counter(y)
for k,v in counter.items():
  per = v / len(y) * 100
  print('Class=%d, n=%d (%.3f%%)' % (k, v, per))
# plot the distribution
pyplot.bar(counter.keys(), counter.values())
pyplot.xlabel("Clases")
pyplot.ylabel("Numero de registros")
pyplot.show()
```

```
Class=2, n=26703 (12.500%)
Class=5, n=26703 (12.500%)
Class=7, n=26703 (12.500%)
Class=1, n=26703 (12.500%)
Class=6, n=26703 (12.500%)
Class=4, n=26703 (12.500%)
Class=0, n=26703 (12.500%)
Class=3, n=26703 (12.500%)
```




```
[ ]  #Creating the dependent variable class
     import pandas as pd
     factor = pd.factorize(df['EL'])
     df.clases = factor[0]
     definitions = factor[1]
     print(df.EL.head())
     print(definitions)
```

```
[ ]  X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.25, random_state = 21)
```

```
[ ]  # Feature Scaling
     scaler = StandardScaler()
     X_train = scaler.fit_transform(X_train)
     X_test = scaler.transform(X_test)
```

```
[ ]  print(X.shape, y.shape)
```

```
# Fitting Random Forest Classification to the Training set
classifier = RandomForestClassifier(n_estimators = 10, criterion = 'entropy', random_state = 42)
classifier.fit(X_train, y_train)
```

```
RandomForestClassifier(bootstrap=True, ccp_alpha=0.0, class_weight=None,
                       criterion='entropy', max_depth=None, max_features='auto',
                       max_leaf_nodes=None, max_samples=None,
                       min_impurity_decrease=0.0, min_impurity_split=None,
                       min_samples_leaf=1, min_samples_split=2,
                       min_weight_fraction_leaf=0.0, n_estimators=10,
                       n_jobs=None, oob_score=False, random_state=42, verbose=0,
                       warm_start=False)
```

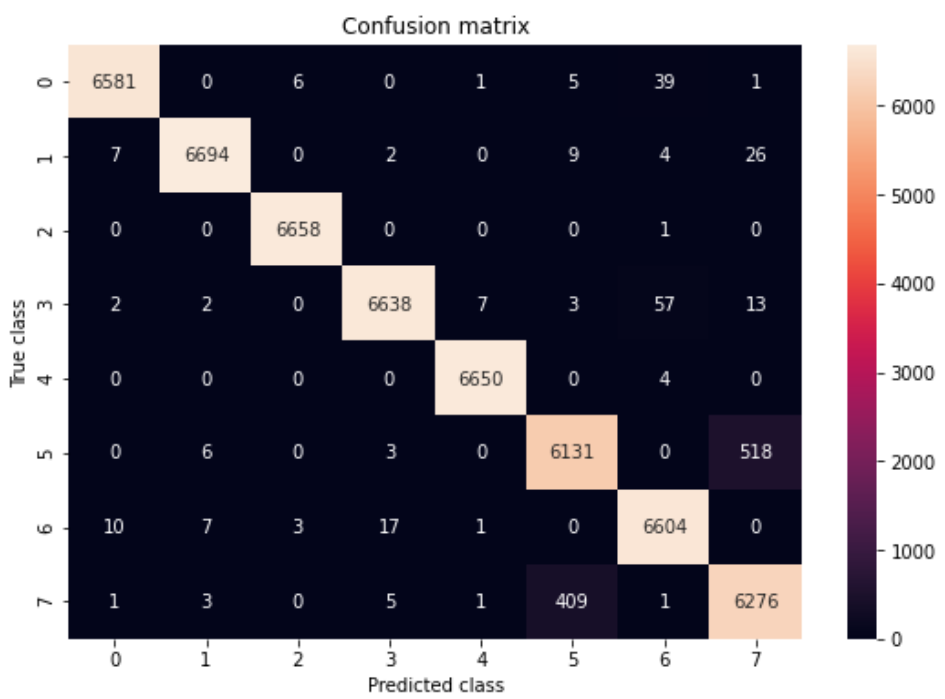
```
[ ]
# Predicting the Test set results
y_pred = classifier.predict(X_test)
#Reverse factorize
reversefactor = dict(zip(range(8),definitions))
y_test = np.vectorize(reversefactor.get)(y_test)
y_pred = np.vectorize(reversefactor.get)(y_pred)
# Making the Confusion Matrix
print(pd.crosstab(y_test, y_pred, rownames=['clases'], colnames=['variables']))
```

variables	1	2	3	4	7	8	10	11
clases								
1	6581	0	6	0	1	5	39	1
2	7	6694	0	2	0	9	4	26
3	0	0	6658	0	0	0	1	0
4	2	2	0	6638	7	3	57	13
7	0	0	0	0	6650	0	4	0
8	0	6	0	3	0	6131	0	518
10	10	7	3	17	1	0	6604	0
11	1	3	0	5	1	409	1	6276

```
from sklearn import metrics
accuracy = metrics.accuracy_score(y_test, y_pred)
print("Accuracy: {:.2f}".format(accuracy))
```

```
Accuracy: 0.98
```

```
[ ] from sklearn.metrics import confusion_matrix, classification_report
from matplotlib import pyplot as plt
import seaborn as sns
conf_matrix = confusion_matrix(y_test, y_pred)
plt.figure(figsize=(9, 6))
sns.heatmap(conf_matrix, annot=True, fmt="d");
plt.title("Confusion matrix")
plt.ylabel('True class')
plt.xlabel('Predicted class')
plt.show()
print(classification_report(y_test, y_pred))
```



	precision	recall	f1-score	support
1	1.00	0.99	0.99	6633
2	1.00	0.99	1.00	6742
3	1.00	1.00	1.00	6659
4	1.00	0.99	0.99	6722
7	1.00	1.00	1.00	6654
8	0.94	0.92	0.93	6658
10	0.98	0.99	0.99	6642
11	0.92	0.94	0.93	6696
accuracy			0.98	53406
macro avg	0.98	0.98	0.98	53406
weighted avg	0.98	0.98	0.98	53406

Código fuente del algoritmo Red neuronal simple

```
[ ]  import pandas as pd
     import numpy as np
     from keras.models import Sequential
     from keras.layers import Dense
     from keras.wrappers.scikit_learn import KerasClassifier
     from keras.utils import np_utils
     from sklearn.model_selection import cross_val_score
     from sklearn.model_selection import KFold
     from sklearn.preprocessing import LabelEncoder
     from sklearn.pipeline import Pipeline
     import pandas as pd
     from sklearn.preprocessing import OneHotEncoder
     from sklearn.compose import ColumnTransformer
     from sklearn.preprocessing import LabelEncoder, OneHotEncoder
```

```
[ ]  dataframe = pd.read_csv("/content/dic_2020_f.csv", header=None, delimiter=",")
     dataset = dataframe.values
     df=pd.DataFrame(dataset)
     df.head()
```

[ ]

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
0	1.0	1.0	2.0	8253.0	5.0	1.0	113.0	2.0	0.0	0.0	0.0	22.1	0.0	22.1	0.0	10.0	1.0	3.0
1	1.0	1.0	2.0	8258.0	5.0	1.0	113.0	2.0	308.0	498.0	190.0	37.6	0.0	37.6	0.0	15.0	1.0	8.0
2	1.0	1.0	2.0	8253.0	5.0	1.0	113.0	0.0	126.0	135.0	9.0	20.3	0.0	20.3	0.0	3.0	1.0	11.0
3	1.0	1.0	2.0	8256.0	5.0	1.0	113.0	0.0	124.0	126.0	2.0	7.4	0.0	7.4	0.0	1.0	1.0	11.0
4	1.0	6.0	2.0	8158.0	5.0	1.0	113.0	0.0	154.0	154.0	0.0	3.7	0.0	3.7	0.0	2.0	1.0	2.0

Separacionde datos X=variables de entrada, Y=Clases de salida

```
X = dataset[:,0:17].astype(str)
pd.DataFrame(X).head()
```

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
0	1.0	1.0	2.0	8253.0	5.0	1.0	113.0	2.0	0.0	0.0	0.0	22.1	0.0	22.1	0.0	10.0	1.0
1	1.0	1.0	2.0	8258.0	5.0	1.0	113.0	2.0	308.0	498.0	190.0	37.6	0.0	37.6	0.0	15.0	1.0
2	1.0	1.0	2.0	8253.0	5.0	1.0	113.0	0.0	126.0	135.0	9.0	20.3	0.0	20.3	0.0	3.0	1.0
3	1.0	1.0	2.0	8256.0	5.0	1.0	113.0	0.0	124.0	126.0	2.0	7.4	0.0	7.4	0.0	1.0	1.0
4	1.0	6.0	2.0	8158.0	5.0	1.0	113.0	0.0	154.0	154.0	0.0	3.7	0.0	3.7	0.0	2.0	1.0

```
Y = dataset[:,17]
pd.DataFrame(Y).head()
```

	0
0	3.0
1	8.0
2	11.0
3	11.0
4	2.0

```
#from sklearn.preprocessing import LabelEncoder
labelencoder = LabelEncoder()
D = labelencoder.fit_transform(Y)

encoded_D = labelencoder.transform(Y)
Y = np_utils.to_categorical(encoded_D)

ct = ColumnTransformer([('encoder', OneHotEncoder(), [0])], remainder='passthrough')
X = np.array(ct.fit_transform(X),dtype=np.float)
```

```
#pd.DataFrame(D).head()
pd.DataFrame(Y).head()
#pd.DataFrame(X).head()
```

	0	1	2	3	4	5	6	7
0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	0.0
1	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0
2	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0
3	0.0	0.0	0.0	0.0	0.0	0.0	0.0	1.0
4	0.0	1.0	0.0	0.0	0.0	0.0	0.0	0.0

```
def baseline_model():
    # create model
    model = Sequential()
    model.add(Dense(8, input_dim=19, activation='relu'))
    model.add(Dense(8, activation='softmax'))
    # Compile model
    model.compile(loss='categorical_crossentropy', optimizer='adam', metrics=['accuracy'])
    return model
```

```
estimator = KerasClassifier(build_fn=baseline_model, epochs=200, batch_size=5, verbose=0)
kfold = KFold(n_splits=10, shuffle=True)
results = cross_val_score(estimator, X, Y, cv=kfold)
```

```
print("Baseline: %.2f%% (%.2f%%)" % (results.mean()*100, results.std()*100))
```

```
print(results)
```

FORMATO DE AUTORIZACIÓN DE PUBLICACIÓN EN EL REPOSITORIO INSTITUCIONAL UNAM

Formulario de autorización para la publicación electrónica en la página web del Repositorio Institucional de la Universidad Nacional de Moquegua, de conformidad con la Ley N° 30035 Ley que regula el Repositorio Nacional Digital de Ciencia, Tecnología e Innovación de Acceso Abierto y Resolución de Comisión Organizadora N° 041-2022-UNAM que aprueba el Reglamento que regula el Repositorio Institucional Digital de la UNAM.

1. DATOS DEL AUTOR

Nombres y Apellidos	: Alonso Moron Coaquira
DNI/Carnet de Extranjería/Pasaporte N°	: 47923713
Correo electrónico	: alonsomc83@gmail.com
Teléfono/Celular	: 971757528

2. GRADO, TITULO O ESPECIALIZACIÓN (Marque con una X)

Bachiller	Título Profesional	Magister	Doctor	Segunda Especialidad
X				

3. TIPO DE PUBLICACIÓN (Marque con una X)

a) Articulo ( )
b) Tesis de pregrado (X)
c) Tesis de Maestría ( )
d) Tesis de Doctorado ( )
e) Monografía ( )
f) Reporte de caso científico o hallazgos ( )
g) Libro ( )
h) Parte de un libro ( )
i) Revisión ( )
j) Conferencia ( )
k) Presentación ( )
l) Artículo preliminar ( )
m) Articulo antes de ser publicado ( )
n) Artículo de prensa ( )
o) Documentos internos ( )
p) Imagen ( )
q) Documentos técnicos ( )
r) Ponencias presentadas en eventos científicos (Jornada, Congreso, Simposio Conferencias, etc.) ( )
s) Informes de suficiencia profesional ( )
t) Patentes producidas por académicos/as ( )

4. TIPO DE TRABAJO DE INVESTIGACIÓN (Marque con una X)

a) Tesis (X)
b) Trabajo de Investigación ( )
c) Trabajo de suficiencia profesional ( )
d) Trabajo académico ( )

5. DATOS DE LA PUBLICACION

a) Título : Predicción de consumo irregular del servicio de agua potable basado en métodos de Machine Learning en la provincia de Ilo.
b) Campo del conocimiento OCDE (Indicar el código de clasificación OCDE de la obra : https://catalogos.concytec.gob.pe/vocabulario/ocde_ford.html) : 1.02.01
c) Fecha de publicación : .....

6. DATOS DEL ASESOR:

a) Nombres y Apellidos : Hugo Euler Tito Chura
b) DNI/Pasaporte/Cédula : 01315850
c) ORCID : https://orcid.org/0000-0001-8269-0655

FIRMA : .....

DOCUMENTO DE IDENTIDAD : 47923713
NOMBRES Y APELLIDOS : Alonso Moron Coaquira

LICENCIAS

Licencia Estándar

Bajo los siguientes términos, autorizo el depósito de mi trabajo de investigación en el Repositorio Institucional Digital de la UNAM.

Con la autorización de depósito da mi trabajo de investigación, otorgo a la Universidad Nacional de Moquegua, una Licencia No Exclusiva para reproducir. distribuir, comunicar al público, transformar (únicamente mediante su traducción a otros idiomas) y poner a disposición del público mi trabajo de investigación (incluido el resumen), en formato físico o digital, en cualquier medio, conocido o por conocerse, a través de los diversos servicios provistos por la Universidad, creados o por crearse, tales como el Repositorio Institucional Digital de la UNAM, Colección de Tesis, entre otros, en el Perú y en el extranjero, por el tiempo y veces que considere necesarias, y libre de remuneraciones.

En virtud de dicha Licencia, la UNIVERSIDAD NACIONAL DE MOQUEGUA, podrá reproducir mi trabajo de investigación en cualquier tipo de soporte y en más de un ejemplar, sin modificar su contenido, solo con los propósitos de seguridad, respaldo y preservación.

Declaro que el trabajo de investigación es una creación de mi autoría y exclusiva titularidad, o coautoría con titularidad compartida, y me encuentro facultado a conceder la presente licencia y, asimismo, garantizo que dicho trabajo de investigación no infringe derechos de autor de terceras personas.

La UNIVERSIDAD NACIONAL DE MOQUEGUA consignara el nombre del/los autor/es del trabajo de investigación, y no le hará ninguna modificación más de la permitida en la presente licencia.

Autorizo el depósito (marque con una X).

(X) **SI autorizo que se deposite inmediatamente.**

FIRMA : .....



DOCUMENTO DE IDENTIDAD : 47923713
NOMBRES Y APELLIDOS : Alonso Moron Coaquira
FECHA : 17 de marzo del 2022