UNIVERSIDAD NACIONAL DE INGENIERÍA
RECINTO UNIVERSITARIO SIMON BOLIVAR
FACTULAD DE ELECTROTECNIA Y COMPUTACIÓN

Trabajo Monográfico para optar al Título de

Ingeniero en Computación

"Proyecto de Estimación de Niveles de Co Ubicación Comercial para Estaciones Bases de Telefonía Móvil en Tecnología LTE Basado en Algoritmos de Machine Learning"

- **Elaborado por:**

  - Wilmer Antonio García Ramos. **Carnet: 2016-0206S**

  - María Gloria Navas Alfaro **Carnet: 2016-0222S**

- **Tutor: Msc. Cedrick Dalla Torre**

Managua, Nicaragua, noviembre 2022

Dedicatoria

A nuestros padres por habernos apoyado en todo momento, por sus consejos y motivaciones constantes, por todo el sacrificio que han dado para que pueda culminar mis estudios.

A nuestra familia, quienes han sido un gran apoyo durante este tiempo, y han logrado
contribuir con este éxito.

A nuestros profesores que nos han enseñado todo lo que necesitamos para nuestro futuro profesional, que nos brindaron su conocimiento y su apoyo en toda nuestra carrera.

Resumen

El presente trabajo monográfico propone una técnica basada en un algoritmo de machine Learning, que determine de una manera supervisada, la estimación para la selección de ubicaciones para la implementación de infraestructura de telefonía móvil.

Se aplicarán algoritmos de Machine Learning para el aprovechamiento de extraer información de mejor calidad para incrementar la productividad de organizaciones o empresas en el sector de las telecomunicaciones, que tengan un plan de mejoramiento que implique la reducción de costos y generar valor para la toma de decisiones.

Se valoraron varios métodos de aprendizaje supervisado para su debida aplicación. Sin embargo, el algoritmo K-NN mostro mejor consistencia que los demás.

Se realizó un código en el lenguaje de programación Python, donde se muestran la utilización de los distintos tipos de aprendizaje supervisado. Logrando demostrar la consistencia que tiene el algoritmo K-NN para una base de datos que en formato .csv, donde se reúne información importante para determinar la viabilidad mediante las variables: factor de cobertura, factor de zona y factor de distancia, para determinar cuatros distintos niveles que podría tener un sitio donde se encuentre una estación base de telefonía móvil.

1 Contenido

Dedicatoria 2

Resumen 3

1. INTRODUCCIÓN 5

2. ANTECEDENTES 7

3. JUSTIFICACIÓN 12

4. OBJETIVOS 13

a. Objetivo General: 13

b. Objetivos específicos: 13

5. MARCO TEORICO 14

Aplicaciones del Machine Learning [13] 31

Aprendizaje Supervisado 31

Categorizaciones de algoritmos de Machine Learning 51

Los desafíos del Machine Learning 52

6. ESTUDIO DE MEDICIONES DE LOS NIVELES DE RSRP PARA TECNOLOGÍA LTE EN UN OPERADOR DE TELECOMUNICACIONES 53

7. MACHINE LEARNING EN PYTHON. 61

8. DESARROLLO DEL CODIGO BASADO EN APRENDIZAJE SUPERVISADO. 64

9. CONCLUSIONES 75

10. Recomendaciones 76

11. BIBLIOGRAFÍA 77

1. INTRODUCCIÓN

El presente trabajo monográfico propone una técnica basada en un algoritmo de machine Learning, que determine de una manera supervisada, la estimación para la selección de ubicaciones para la implementación de infraestructura de telefonía móvil.

La importancia de mejorar los niveles de potencia de la tecnología LTE-A en zonas rurales de Nicaragua es significativa. Lo anterior, tendrá un impacto en la mejora de la calidad de servicio de cobertura celular, basado en variables independientes, tales como distancia de altura, potencia transmitida, tilt eléctrico, tilt mecánico y azimut, y como variables dependientes el nivel de RSRP o de potencia recibida en el móvil.

Se aplicarán algoritmos de Machine Learning para el aprovechamiento de extraer información de mejor calidad para incrementar la productividad de organizaciones o empresas en el sector de las telecomunicaciones, que tengan un plan de mejoramiento que implique la reducción de costos y generar valor para la toma de decisiones.

En esta propuesta de investigación, se pretende la utilización de un algoritmo de machine Learning, conocido como aprendizaje supervisado que normalmente empieza con un conjunto de datos y un análisis de la clasificación que tienen dichos datos entre sí.

El aprendizaje supervisado tiene la intención de encontrar patrones en datos que se pueden aplicar a un proceso analítico que aporte significativamente en la toma de decisiones.

Estos datos tienen características etiquetadas que definen el significado de los datos. Se creará una aplicación de machine Learning que puede crear con base a valores las variables independientes antes mencionadas.

2. ANTECEDENTES

Dentro del contexto educativo superior nacional e internacional, en lo que nos atañe con la presente propuesta monográfica, se realizó un proceso de búsqueda y recopilación de datos orientados a verificar cuantos trabajos de este nivel, con temática relacionada a la propuesta presente.

A nivel nacional, en la Universidad Nacional de Ingeniería se hizo la disertación del trabajo de tesis ***"Método para la detección de cáncer de mama en mamografías usando Convolutional Neural Networks (CNN)"*** [1]. Se logró Desarrollar un algoritmo de procesamiento basado en una arquitectura de CNN llamada ResNet que genere mapas de calor a partir de las características obtenidas en el preprocesamiento. Además, se implementó un clasificador binario de imágenes usando Convolutional Neural Networks para obtener el nivel de predicción del modelo propuesto y se validó el modelo propuesto mediante la utilización de bases de datos de mamografías de acceso libre.

En el siguiente trabajo de investigación con título ***"Optimización de Redes UMTS soportada en Machine Learning"*** [2], se obtuvo un análisis, la validación y la implementación de un algoritmo tomado del amplio mundo del Machine Learning, para generar uno o varios modelos que puedan apoyar en los procesos de optimización de las redes UMTS (Universal Mobile Telecommunications System - Sistema universal de telecomunicaciones móviles), y más específicamente en la fase de diagnóstico de las celdas. Se presentan algunos de los contadores y KPIs(Key Performance Indicators – Indicadores claves de desempeño) estadísticos de las celdas UMTS que normalmente son utilizados por los ingenieros de optimización para diagnosticar el estado de las celdas, y que servirán para armar un data set que pueda ser utilizado por las técnicas de Machine Learning como información de entrada para entrenar los modelos. Para esto se realiza el análisis de varias técnicas de Machine Learning y dependiendo de las características y condiciones del data set disponible, se escogen dos de las técnicas, las cuales serán entrenadas y evaluadas para finalmente escoger la

técnica de mejor desempeño y sobre la cual se realizará la implementación en un lenguaje de programación que permita realizar diagnósticos futuros de las celdas UMTS.

En el Perú, se desarrollo la investigación ***"Aplicación de Machine Learning en las Empresas del Sector de las Telecomunicaciones del Perú"*** [3], tuvo un enfoque cualitativo con el objetivo general de describir la aplicación de machine learning en las empresas del sector telecomunicaciones del Perú, se emplearon técnicas de entrevista a profundidad, observación participante y análisis documental con sus respectivos instrumentos guía de preguntas semi estructuradas, ficha de observación y ficha de análisis documental, las entrevistas y observación se realizaron mediante las herramientas meet de Google y teams de Microsoft, para el análisis documental se utilizó documentos públicos, el escenario de estudio fueron las áreas de análisis de datos de las empresas Movistar, Claro y Entel y participaron expertos en análisis de datos, el tipo de investigación fue tecnológica y de diseño investigación acción. Se concluyó que la aplicación de machine learning en estas empresas es fundamental para aumentar la eficiencia de sus procesos y mejorar la satisfacción de sus clientes reduciendo las tasas de abandono, por lo cual están empezando a incorporarlo en sus estrategias de transformación digital, para su aplicación es importante comprender la conceptualización de esta tecnología, los beneficios actuales para las empresas del sector, las barreras que se presentan en la industria, los casos que influyen en su adopción y las tendencias tecnológicas que potenciarán los beneficios y masificarán su uso. Se hace necesario la incorporación de herramientas que faciliten su aplicación, el uso de algoritmos que aumenten la precisión de los resultados, desarrollar soluciones de lenguaje natural para una atención personalizada en cualquier momento, difundir los beneficios a nivel de toda la compañía, extender el ámbito de influencia más allá de la propia compañía y trabajar en políticas que garanticen la ética en el uso de datos y la responsabilidad en las decisiones producto de su uso.

Otro artículo, con título ***"Cómo Telefónica usa la inteligencia artificial y el machine Learning para conectar a los no conectados"*** [4], Telefónica utiliza la ciencia de los datos para identificar y localizar de manera sistemática a los no conectados, incluyéndoles en sus redes y operaciones para llevar, de forma sostenible, conectividad a la mayor parte de América Latina. El proyecto Internet para Todos es el buque insignia de Telefónica para conectar a los no conectados en LATAM. Hoy son más de 100 millones de personas las que viven sin contar con una conexión a Internet segura en el área de Telefónica. Las razones son múltiples, abarcando desde la geografía, la densidad de población y las condiciones socioeconómicas. Históricamente, las redes fijas y móviles han sido diseñadas para alcanzar la máxima eficiencia en entornos urbanos densos. La implementación de estas tecnologías en áreas rurales remotas y de baja densidad es posible pero ineficiente, lo que desafía la sostenibilidad financiera del modelo

En Nicaragua, en la Universidad Nacional de Ingeniería, en el año 2020, se inscribió un protocolo de investigación con el tema: ***"Optimización del tiempo de procesamiento y eficiencia del algoritmo para solucionar el problema inverso generalizado de Voronoi utilizando lenguaje de programación visual C++ y técnicas del tipo divide y vencerás aplicado a imágenes reales"*** [5]. En dicha propuesta se pretende a tener como entrada imágenes con formato BMP o JPG se deben descomponen (para cada caso) en un conjunto de componentes obtenidos mediante una segmentación de la misma para su representación en una forma intermedia. A partir de la misma, se buscará convertirla a una representación (posiblemente con pérdida) basada en diagramas de Voronoi. Una meta de importancia es disminuir el tiempo del proceso de segmentación y conversión de la imagen a una representación de la misma basada en diagramas de Voronoi, tratando de mantener la mayor fidelidad posible con respecto a la imagen original.

Se encontró una tesis doctoral con el título "***Importancia del infiltrado inflamatorio y la neovascularización asociada al melanoma y su correlación con el desarrollo me metástisis: Estudio inmunohistoquímico de 81 casos. Árboles de decisión basados en Machine Learning"*** [6].

También, se encontró a nivel internacional, un artículo científico ***"Machine Learning Algorithms – a Review"*** [7]. Se obtuvo como resultado, que el aprendizaje automático (ML) es el estudio científico de algoritmos y modelos estadísticos que utilizan los sistemas informáticos para realizar una tarea específica sin estar explícitamente programado. Algoritmos de aprendizaje en muchas aplicaciones que utilizamos a diario. Cada vez que un motor de búsqueda web como Google se utiliza para buscar en Internet, una de las razones por las que funciona tan bien es porque un algoritmo de aprendizaje que ha aprendido a clasificar páginas web. Estos algoritmos se utilizan para diversos fines, como minería de datos, procesamiento de imágenes, predicción análisis, etc., por nombrar algunos. La principal ventaja de utilizar el aprendizaje automático es que, una vez que un algoritmo aprende qué hacer con los datos, puede hacer su trabajo automáticamente. En este artículo, se presentó una breve revisión y perspectiva futura de las vastas aplicaciones de los algoritmos de aprendizaje automático.

Además, se revisó el artículo con tema***: "Machine Learning approaches and databases for prediction of drug-target interaction: a survey paper"*** [8]. La tarea de predecir las interacciones entre fármacos y dianas juega un papel clave en el proceso de descubrimiento de fármacos. Existe la necesidad de desarrollar

enfoques de predicción novedosos y eficientes para evitar experimentos costosos y laboriosos, aunque no siempre deterministas, para determinar las interacciones fármaco-objetivo solo mediante experimentos. Estos enfoques deben ser capaces de identificar potenciales de manera oportuna. En este artículo, se describió los datos necesarios para la tarea de predicción, seguidos de un catálogo completo que
consta de métodos y bases de datos de aprendizaje automático, que se han propuesto y utilizado para predecir. También se analizan brevemente las ventajas y desventajas de cada conjunto de métodos. Por último, se destacan los desafíos que uno puede enfrentar en la predicción utilizando enfoques de aprendizaje automático y se concluye arrojando algunas luces sobre importantes direcciones de investigación futuras.

3. JUSTIFICACIÓN

La iniciativa del presente trabajo surge dentro del sector de empresas que ofrecen infraestructura para servicios de telecomunicaciones móviles, para mejorar la rentabilidad de las estructuras, ya que, si se tiene más de un solo operador, el retorno del capital será más acelerado. Además, este trabajo podrá servir de referencia a que los estudiantes que necesiten más conocimientos sobre todo lo que respecta Algoritmos de Machine Learning para resolver problemas ingenieriles a nivel nacional o internacional.

El producto de esta investigación, será un documento alineado al método científico que servirá como material bibliográfico de consulta para los estudiantes de la Facultad de Electrotecnia y Computación de la Universidad nacional de Ingeniería.

Además, que esta temática juega un papel importancia en lo que respecta a la disciplina de ciencias de datos, donde los estudiantes de ingeniería en computación podrían aplicar dichos algoritmos para procesar datos y generar valor en diferentes instituciones, organizaciones y empresas que requieran la realización de predicciones para establecer planes estratégicos en función de la optimización de recursos, y de esa manera lograr mejorar la rentabilidad

4. OBJETIVOS

a. Objetivo General:

- Aplicar algoritmo de Machine Learning para la mejora de precisión del nivel comercial de Co ubicación de estaciones bases para el servicio de telefonía móvil.

b. Objetivos específicos:

- Identificar el algoritmo de machine Learning más idóneo para su aplicación en la selección de sitios de co ubicación de estaciones bases para el servicio de telefonía móvil.

- Procesar datos mediante el uso de algoritmos de Machine Learning que permita la viabilidad de la co ubicación de las estaciones básica, considerando el nivel de cobertura del servicio de telefonía móvil.

- Implementar un código en Python que considere mediante un algoritmo de machine Learning las posiciones de coordenadas más idóneas para garantizar un buen nivel de potencia para las estaciones bases que se pretenden co ubicar.

5. MARCO TEORICO

Introducción al Machine Learning

En los primeros días de las aplicaciones "inteligentes", muchos sistemas usaban reglas codificadas a mano de decisiones "if" y "else" para procesar datos o ajustarse a la entrada del usuario. Piensa en un filtro de spam cuyo trabajo es mover los mensajes de correo electrónico entrantes apropiados a una carpeta de spam. Se podría crear una lista negra de palabras que daría como resultado que un correo electrónico se marque como correo no deseado. Este sería un ejemplo del uso de un sistema de reglas diseñado por expertos para diseñar un Aplicación “inteligente”. La elaboración manual de reglas de decisión es factible para algunas aplicaciones.

Particularmente aquellas en las que los humanos tienen una buena comprensión del proceso modelar. Sin embargo, el uso de reglas codificadas a mano para tomar decisiones tiene dos desventajas principales ventajas:

La lógica requerida para tomar una decisión es específica para un solo dominio y tarea. Cambiar la tarea, aunque sea ligeramente, puede requerir una reescritura de todo el sistema.
El diseño de reglas requiere una comprensión profunda de cómo se debe tomar una decisión por un experto humano.

Un ejemplo de dónde fallará este enfoque codificado a mano es en la detección de rostros en imágenes hoy en día, todos los teléfonos inteligentes pueden detectar una cara en una imagen. Sin embargo, la detección de rostros era un problema sin resolver hasta tan recientemente como 2001. El problema principal es que el forma en que los píxeles (que componen una imagen en una computadora) son

"percibidos" por la computadora es muy diferente de cómo los humanos perciben una cara. Esta diferencia en la representación sentación hace que sea básicamente imposible para un ser humano llegar a un buen conjunto de reglas para describir lo que constituye un rostro en una imagen digital.

Problemas que el aprendizaje automático puede resolver

Los tipos más exitosos de algoritmos de aprendizaje automático son aquellos que automatizan procesos de toma de decisiones generalizando a partir de ejemplos conocidos. En esta configuración, lo que se conoce como aprendizaje supervisado, el usuario proporciona al algoritmo pares de entradas y salidas deseadas, y el algoritmo encuentra una manera de producir la salida deseada.
poner dada una entrada. En particular, el algoritmo puede crear una salida para una entrada nunca ha visto antes sin la ayuda de un ser humano. Volviendo a nuestro ejemplo de clasificación de spam, utilizando el aprendizaje automático, el usuario proporciona al algoritmo una gran cantidad de correos electrónicos (que son la entrada), junto con información sobre si alguno de estos correos electrónicos es spam (que es el resultado deseado). Dado un nuevo correo electrónico, el algoritmo producirá una predicción sobre si el nuevo correo electrónico es spam.

Los algoritmos de aprendizaje automático que aprenden de pares de entrada/salida se denominan supervisados. algoritmos de aprendizaje porque un "maestro" proporciona supervisión a los algoritmos en la forma de los resultados deseados para cada ejemplo del que aprenden. Mientras creaba un conjunto de datos de entradas y salidas es a menudo un proceso manual laborioso, aprendizaje supervisado los algoritmos se entienden bien y su rendimiento es fácil de medir. Si la aplicación se puede formular como un problema de aprendizaje supervisado, y se podría crear un conjunto de datos que incluya el resultado deseado, el aprendizaje automático probablemente será capaz de resolver su problema.

Los ejemplos de tareas de aprendizaje automático supervisado incluyen:
Identificar el código postal a partir de dígitos escritos a mano en un sobre
Aquí la entrada es un escaneo de la escritura a mano, y la salida deseada es el real
dígitos en el código postal. Para crear un conjunto de datos para construir un modelo de aprendizaje automático, necesitas recoger muchos sobres. Entonces puedes leer los códigos postales tú mismo y almacene los dígitos como sus resultados deseados.

Otro ejemplo. Determinar si un tumor es benigno en base a una imagen médica
Aquí la entrada es la imagen y la salida es si el tumor es benigno. Para
crear un conjunto de datos para construir un modelo, necesita una base de datos de imágenes médicas. Tú también necesita la opinión de un experto, por lo que un médico debe observar todas las imágenes y decidir qué tumores son benignos y cuáles no. Incluso podría ser necesario hacer un diagnóstico adicional más allá del contenido de la imagen para determinar si el tumor en la imagen es canceroso o no.

Otro ejemplo. Detección de actividad fraudulenta en transacciones con tarjetas de crédito. Aquí la entrada es un registro de la transacción de la tarjeta de crédito y la salida es si es probable que sea fraudulento o no. Asumiendo que usted es la entidad de tributar las tarjetas de crédito, recopilar un conjunto de datos significa almacenar todas las transacciones y registro si un usuario reporta alguna transacción como fraudulenta.

Una cosa interesante a tener en cuenta sobre estos ejemplos es que, aunque las entradas y salidas parece bastante sencillo, el proceso de recopilación de datos para estas tres tareas es muy diferente, si bien leer sobres es laborioso, es fácil y económico.

La obtención de diagnósticos e imágenes médicas, por otro lado, requiere no sólo costosos maquinaria, sino también conocimientos especializados raros y caros, por no hablar de la ética preocupaciones y problemas de privacidad. En el ejemplo de detección de fraude con tarjeta de crédito, la recopilación de datos
lección es mucho más simple. Sus clientes le proporcionarán el resultado deseado, como denunciarán el fraude. Todo lo que tiene que hacer para obtener los pares de entrada/salida de fraude la actividad prestada y no fraudulenta está a la espera.

Los algoritmos no supervisados son el otro tipo de algoritmo que cubriremos en este libro. En el aprendizaje no supervisado, solo se conocen los datos de entrada y no se conoce la salida. Los datos se le dan al algoritmo. Si bien hay muchas aplicaciones exitosas de esto métodos, por lo general son más difíciles de entender y evaluar. [8]

Ejemplos de aprendizaje no supervisado incluyen:

Identificar temas en un conjunto de publicaciones de blog si tiene una gran colección de datos de texto, es posible que desee resumirlos y encontrar temas predominantes en él. Es posible que no sepa de antemano cuáles son estos temas son, o cuántos temas puede haber. Por lo tanto, no hay salidas conocidas.

Segmentación de clientes en grupos con preferencias similares dado un conjunto de registros de clientes, es posible que desee identificar qué clientes están similar, y si hay grupos de clientes con preferencias similares. Para un sitio de compras, estos pueden ser "padres", "ratones de biblioteca" o "jugadores". Porque tú no sé de antemano cuáles podrían ser estos grupos, o incluso cuántos hay, no tiene salidas conocidas.

Detección de patrones de acceso anormales a un sitio web Para identificar abusos o errores, a menudo es útil encontrar patrones de acceso que sean diferentes.

Cada patrón anormal puede ser muy diferente y usted podría no tener ninguna instancia registrada de comportamiento anormal. porque en esto ejemplo, solo observa el tráfico y no sabe lo que constituye normal y comportamiento anormal, este es un problema no supervisado.

K-Nearest-Neighbor es un algoritmo basado en instancia de tipo supervisado de Machine Learning. Puede usarse para clasificar nuevas muestras (valores discretos) o para predecir (regresión, valores continuos). Al ser un método sencillo, es ideal para introducirse en el mundo del Aprendizaje Automático. Sirve esencialmente para clasificar valores buscando los puntos de datos "más similares" (por cercanía) aprendidos en la etapa de entrenamiento

A diferencia de K-means, que es un algoritmo no supervisado y donde la "K" significa la cantidad de "grupos" (clusters) que deseamos clasificar, en K-Nearest Neighbor la "K" significa la cantidad de "puntos vecinos" que tenemos en cuenta en las cercanías para clasificar los "n" grupos -que ya se conocen de antemano, pues es un algoritmo supervisado.

K-Nearest Neighbor

Es un método que simplemente busca en las observaciones más cercanas a la que se está tratando de predecir y clasifica el punto de interés basado en la mayoría de datos que le rodean. Como se dijo con antelación, es un algoritmo:

Supervisado: esto -brevemente- quiere decir que tenemos etiquetado nuestro conjunto de datos de entrenamiento, con la clase o resultado esperado dada "una fila" de datos.

Basado en Instancia: Esto quiere decir que nuestro algoritmo no aprende explícitamente un modelo (como por ejemplo en Regresión Logística o árboles de decisión). En cambio, memoriza las instancias de entrenamiento que son usadas como "base de conocimiento" para la fase de predicción.

¿Dónde se aplica k-Nearest Neighbor?

Aunque, sencillo, se utiliza en la resolución de multitud de problemas, como en sistemas de recomendación, búsqueda semántica y detección de anomalías.

Pros y contras

Como pros tiene sobre todo que es sencillo de aprender e implementar. Tiene como contras que *utiliza todo el dataset* para entrenar "cada punto" y por eso requiere de uso de mucha memoria y recursos de procesamiento (CPU). Por estas razones kNN tiende a funcionar mejor en datasets pequeños y sin una cantidad enorme de features (las columnas).

¿Cómo funciona kNN?

1. Calcular la distancia entre el item a clasificar y el resto de items del dataset de entrenamiento.
2. Seleccionar los "k" elementos más cercanos (con menor distancia, según la función que se use)
3. Realizar una "votación de mayoría" entre los k puntos: los de una clase/etiqueta que <<dominen>> decidirán su clasificación final.

Teniendo en cuenta el punto 3, veremos que para decidir la clase de un punto es muy importante el valor de k, pues este terminará casi por definir a qué grupo pertenecerán los puntos, sobre todo en las "fronteras" entre grupos. Por ejemplo -y a priori- yo elegiría valores impares de k para desempatar (si las features que utilizamos son pares). No será lo mismo tomar para decidir 3 valores que 13. Esto no quiere decir que necesariamente tomar más puntos implique mejorar la precisión. Lo que es seguro es que cuantos más "puntos k", más tardará nuestro algoritmo en procesar y darnos respuesta.

Las formas más populares de “medir la cercanía” entre puntos son la distancia Euclidiana (la “de siempre”) o la Cosine Similarity (mide el ángulo de los vectores, cuanto menores, serán similares). Recordemos que este algoritmo -y prácticamente todos en ML- funcionan mejor con varias características de las que tomemos datos (las columnas de nuestro dataset). Lo que entendemos como “distancia” en la vida real, quedará abstracto a muchas dimensiones que no podemos “visualizar” fácilmente (como por ejemplo en un mapa). [9]

El proyecto de ML. [10]

Primero definamos en grandes rasgos las diversas etapas que conforman el desarrollo de un proyecto de Machine Learning.

1. Análisis de Negocio
2. Infraestructura de IA
3. Ingeniería de Datos
4. Modelado
5. Implementación / Despliegue

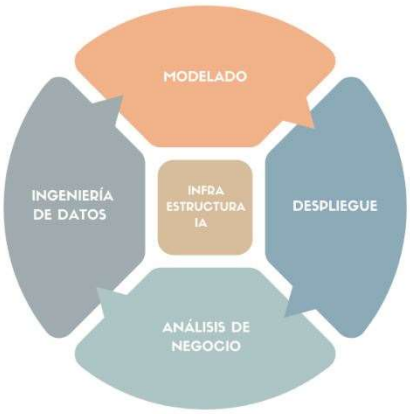


Figura 1. Ciclo de vida de un proyecto de machine learning. [10]

- **Siete pasos de machine learning para construir tu máquina.**

Paso 1: Colectar Datos

Dada la problemática que deseas resolver, deberás investigar y obtener datos que utilizaras para *alimentar a tu máquina*. Importa mucho la calidad y cantidad de información que consigas ya que **impactará directamente en lo bien o mal** que luego funcione nuestro modelo. Puede que tengas la información en una base de datos ya existente o que la debas crear desde cero. Si es un pequeño proyecto puedes crear una planilla de cálculos que luego se exportará fácilmente como archivo csv. También es frecuente utilizar la técnica de web scraping para recopilar información de manera automática de diversas fuentes (y/o servicios rest/ APIs).

Paso 2: Preparar los datos

Es importante mezclar "las cartas" que obtengas ya que el orden en que se procesen los datos dentro de tu máquina no debe de ser determinante. También es un buen momento para hacer visualizaciones de nuestros datos y revisar si hay correlaciones entre las distintas características ("features", suelen ser las columnas de nuestra base datos o archivo) que obtuvimos. Habrá que hacer Selección de Características, pues las que elijamos impactarán directamente en los tiempos de ejecución y en los resultados, también podremos hacer reducción de dimensiones aplicando PCA si fuera necesario. Deberemos tener balanceada la cantidad de datos que tenemos para cada resultado(clase), para que sea representativo, ya que si no, el aprendizaje podrá ser tendencioso hacia un tipo de respuesta y cuando nuestro modelo intente generalizar el conocimiento fallará.

También deberemos separar los datos en en dos grupos: uno para entrenamiento y otro para evaluación del modelo. Podemos fraccionar aproximadamente en una proporción de 80/20 pero puede variar según el caso y el volumen de datos que tengamos.

En esta etapa también podemos preprocesar nuestros datos normalizando, eliminar duplicados y hacer corrección de errores.

Paso 3: Elegir el modelo

Existen diversos modelos que podemos elegir de acuerdo al objetivo que tengamos: utilizaremos algoritmos de clasificación, predicción, regresión lineal, clustering (ejemplo k-means ó k-nearest neighbor), Deep Learning (ej: red neuronal), bayesiano, etc y podrá haber variantes si lo que vamos a procesar son imágenes, sonido, texto, valores numéricos.

Paso 4: Entrenar nuestra máquina

Utilizaremos el set de datos de entrenamiento para ejecutar nuestra máquina y deberemos de ver una mejora incremental (para la predicción). Recordar inicializar los “pesos” de nuestro modelo aleatoriamente, los pesos son los valores que multiplican o afectan a las relaciones entre las entradas y las salidas, se irán ajustando automáticamente por el algoritmo seleccionado cuanto más se entrena. Revisar los resultados obtenidos y corregir (por ej. inclinación de la pendiente) y volver a iterar.

Paso 5: Evaluación

Deberemos comprobar la máquina creada contra nuestro set de datos de Evaluación que *contiene entradas que el modelo desconoce* y verificar la precisión de nuestro modelo ya entrenado. Si la exactitud es menor o igual al 50% ese modelo no será útil ya que sería como lanzar una moneda al aire para tomar

decisiones. Si alcanzamos un 90% o más podremos tener una buena confianza en los resultados que nos otorga el modelo.

Paso 6: Parameter Tuning (configuración de parámetros)

Si durante la evaluación no obtuvimos buenas predicciones y nuestra precisión no es la mínima deseada es posible que tengamos problemas de overfitting (ó underfitting) y deberemos retornar al paso de entrenamiento (4) haciendo antes *una nueva configuración de parámetros de nuestro modelo*. Podemos incrementar la cantidad de veces que iteramos nuestros datos de entrenamiento (EPOCHs). Otro parámetro importante es el conocido como “Learning Rate” (taza de aprendizaje) que suele ser un valor que multiplica al gradiente para acercarlo poco a poco al mínimo global (o local) para minimizar el coste de la función. No es lo mismo incrementar nuestros valores en 0,1 unidades que de 0,001 esto puede afectar significativamente el tiempo de ejecución del modelo.

También se puede indicar el máximo error permitido de nuestro modelo. Podemos pasar de tardar unos minutos a horas (y días) en entrenar nuestra máquina. A estos parámetros muchas veces se les llama Hiperparámetros. Este “tuneo” sigue siendo más un arte que una ciencia y se ira mejorando a medida que experimentamos. Suele haber muchos parámetros para ir ajustando y al combinarlos se pueden disparar todas nuestras opciones.

Cada algoritmo tiene sus propios parámetros a ajustar. Por nombrar alguno más, en las Redes Neuronales Artificiales deberemos definir en su arquitectura la cantidad de hidden layers que tendrá e ir probando con más o con menos y con cuantas neuronas cada capa. Este será un trabajo de gran esfuerzo y paciencia para dar con buenos resultados.

Paso 7: Predicción o Inferencia

Cuando se tiene listo el para la utilización del modelo de Aprendizaje Automático con nueva información, se podrá hacer la realización de predicción o inferir resultados.

Introducción a PCA (Reducción de Dimensiones). [11]

Imaginemos que queremos predecir los precios de alquiler de vivienda del mercado. Al recopilar información de diversas fuentes tendremos en cuenta variables como tipo de vivienda, tamaño de vivienda, antigüedad, servicios, habitaciones, con/sin jardín, con/sin piscina, con/sin muebles pero también podemos tener en cuenta la distancia al centro, si hay colegio en las cercanías, o supermercados, si es un entorno ruidoso, si tiene autopistas en las cercanías, la "seguridad del barrio", si se aceptan mascotas, tiene wifi, tiene garaje, trastero… y seguir y seguir sumando variables.

Es posible que *cuanta más (y mejor) información, obtengamos una predicción más acertada*. Pero también empezaremos a notar que la ejecución de nuestro algoritmo seleccionado (regresión lineal, redes neuronales, etc.) empezará a tomar más y más tiempo y recursos. Es posible que algunas de las variables sean menos importantes y no aporten demasiado valor a la predicción. También podríamos acercarnos peligrosamente a causar overfitting al modelo.

Al quitar variables estaríamos haciendo Reducción de Dimensiones. Al hacer Reducción de Dimensiones (las características) tendremos menos relaciones entre variables a considerar. Para reducir las dimensiones podemos hacer dos cosas:

- Eliminar por completo dimensiones
- Extracción de Características

Eliminar por completo algunas dimensiones no estaría mal, pero *deberemos tener certeza* en que estamos quitando dimensiones poco importantes. Por ejemplo, para nuestro ejemplo, podemos suponer que el precio de alquiler no cambiará mucho si el dueño acepta mascotas en la vivienda. Podría ser un acierto o podríamos estar perdiendo información importante.
En la Extracción de Características si tenemos 10 características crearemos otras 10 características nuevas independientes en donde cada una de esas "nuevas" características es una combinación de las 10 características "viejas". Al crear estas nuevas variables independientes lo haremos de una manera específica y las pondremos en un orden de "mejor a peor" sean para predecir a la variable dependiente.

¿Y la reducción de dimensiones?

Como tenemos las variables ordenadas de "mejor a peores predictoras" ya sabemos cuáles serán las más y menos valiosas. A diferencia de la eliminación directa de una característica "vieja", nuestras nuevas variables son combinaciones de todas las variables originales, aunque eliminemos algunas, estaremos manteniendo la información útil de todas las variables iniciales.

¿Qué es Principal Component Analysis?

Entonces Principal Component Analysis es una técnica de Extracción de Características donde combinamos las entradas de una manera específica y podemos eliminar algunas de las variables "menos importantes" manteniendo la parte más importante todas las variables. Como valor añadido, luego de aplicar PCA conseguiremos que todas las nuevas variables sean independientes una de otra.

¿Cómo funciona PCA?

En resumen, lo que hace el algoritmo es:

- Estandarizar los datos de entrada (ó Normalización de las Variables).
- Obtener los autovectores y autovalores de la matriz de covarianza.
- Ordenar los autovalores de mayor a menor y elegir los “k”. autovectores que se correspondan con los autovectores “k” más. grandes (donde “k” es el número de dimensiones del nuevo subespacio de características).
- Construir la matriz de proyección W con los “k” autovectores seleccionados.
- Transformamos el dataset original “X estandarizado” vía W para obtener las nuevas características k-dimensionales.

Todo esto ya lo hace solito scikit-learn (u otros paquetes Python). Ahora que tenemos las nuevas dimensiones, deberemos seleccionar con cuales nos quedamos.

Selección de los Componentes Principales

Típicamente utilizamos PCA para reducir dimensiones del espacio de características original (aunque PCA tiene más aplicaciones). Hemos rankeado las nuevas dimensiones de “mejor a peor reteniendo información”. Pero ¿cuantas elegir para obtener buenas predicciones, sin perder información valiosa? Podemos seguir 3 métodos:

Método 1:
Elegimos arbitrariamente “las primeras n dimensiones” (las más importantes). Por ejemplo, si lo que queremos es poder graficar en 2 dimensiones, podríamos tomar las 2 características nuevas y usarlas como los ejes X e Y.

Método 2:
calcular la *"proporción de* variación explicada*"* de cada característica e ir tomando dimensiones hasta alcanzar un mínimo que nos propongamos, por ejemplo hasta alcanzar a explicar el 85% de la variabilidad total.

Método 3:
Crear una gráfica especial llamada scree plot -a partir del Método 2- y seleccionar cuántas dimensiones usaremos por el método "del codo" en donde identificamos visualmente el punto en donde se produce una caída significativa en la variación explicada relativa a la característica anterior.

¿Por qué funciona PCA?

Suponiendo nuestras características de entrada estandarizadas como la matriz Z y ZT su transpuesta, cuando creamos la matriz de covarianza ZTZ es una matriz que contiene estimados de cómo cada variable de Z se relaciona con cada otra variable de Z. Comprender como una variable es asociada con otra es importante.

Los autovectores representan dirección. Los autovalores representan magnitud. A mayores autovalores, se correlacionan direcciones más importantes.

Por último, asumimos que a más variabilidad en una dirección particular se correlaciona con explicar mejor el comportamiento de una variable dependiente. Mucha variabilidad usualmente indica "Información" mientras que poca variabilidad indica "Ruido".

Con PCA obtenemos:

1. Una medida de como cada variable se asocia con las otras (matriz de covarianza).

2. La dirección en las que nuestros datos están dispersos (autovectores).

3. La relativa importancia de esas distintas direcciones (autovalores).

PCA combina nuestros predictores y nos permite deshacernos de los autovectores de menor importancia relativa.

Definiendo Machine Learning. [12]

El Machine Learning -traducido al español como “Aprendizaje Automático”- es un subcampo de la Inteligencia Artificial que busca resolver el “cómo construir programas de computadora que mejoran automáticamente adquiriendo experiencia”.
Esta definición indica que el programa que se crea con ML no necesita que el programador indique explícitamente las reglas que debe seguir para lograr su tarea si no que esta mejora automáticamente.

Grandes volúmenes de datos están surgiendo de diversas fuentes en los últimos años y el Aprendizaje Automático relacionado al campo estadístico consiste en extraer y reconocer patrones y tendencias para comprender qué nos dicen los datos. Para ello, se vale de algoritmos que pueden procesar Gygas y/o Terabytes y obtener información útil.

Drew Conway creó un simpático diagrama de Venn en el que inerrelaciona diversos campos.

Figura 2. Interrelación con diferentes campos. [12]

Aproximación para programadores

Los programadores sabemos que diversos algoritmos de búsqueda pueden tomar mucho tiempo en resolverse y que cuanto mayor sea el campo de búsqueda crecerán potencialmente las posibilidades de combinación de una respuesta óptima, haciendo que los tiempos de respuesta tiendan al infinito o que tomen más tiempo de lo que un ser humano tolerar (por quedarse sin vida o por impaciencia).

Para poder resolver este tipo de problemas surgen soluciones de tipo heurísticas que intentan dar “intuición” al camino correcto a tomar para resolver un problema. Estos pueden obtener buenos resultados en tiempos menores de procesamiento, pero muchas veces su intuición es arbitraria y pueden llegar a fallar.

Los algoritmos de ML intentan utilizar menos recursos para “entrenar” grandes volúmenes de datos e ir aprendiendo por sí mismos. Podemos subdividir el ML en 2 grandes categorías: Aprendizaje Supervisado o Aprendizaje No Supervisado.

Entre los Algoritmos más utilizados en Inteligencia Artificial encontramos:

- Arboles de Decisión.
- Regresión Lineal.
- Regresión Logística .
- k Nearest Neighbor.
- PCA / Principal Component Analysis.
- Gaussian Naive Bayes.
- K-Means.
- Redes Neuronales Artificiales.
- Aprendizaje Profundo ó Deep Learning.
- Clasificación de imágenes.

Una mención distintiva merece las RNAs ya que son algoritmos que utilizan un comportamiento similar a las neuronas humanas y su capacidad de sinopsis para la obtención de resultados, interrelacionándose diversas capas de neuronas para darle mayor poder. Aunque estos códigos existen desde hace más de 70 años, en la última década han evolucionado notoriamente –Breve Historia de las RNA– (en paralelo a la mayor capacidad tecnológica de procesamiento, memoria RAM y disco, la nube, etc) y están logrando impresionantes resultados para analizar textos y síntesis de voz, traducción de idiomas, procesamiento de lenguaje natural, visión artificial, análisis de riesgo, clasificación y predicción y la creación de motores de recomendación

El Machine Learning es una nueva herramienta clave que posibilitará el desarrollo de un futuro mejor para el hombre brindando inteligencia a robots, coches y casas. Las Smart Cities, el IOT ya se están volviendo una realidad y también las aplicaciones de Machine Learning en Asistentes como Siri, las recomendaciones de Netflix o Sistemas de Navegación en Drones. Para los ingenieros o informáticos es una disciplina fundamental para ayudar a crear y transitar este nuevo futuro.

Aplicaciones del Machine Learning [13]

Aprendizaje Supervisado

En el Aprendizaje Supervisado los datos para el entrenamiento incluyen la solución deseada, llamada “etiquetas” (labels). Un claro ejemplo es al clasificar correo entrante entre Spam o no. Entre las diversas características que queremos entrenar deberemos incluir si es correo basura o no con un 1 o un 0. Otro ejemplo son al predecir valores numéricos por ejemplo precio de vivienda a partir de sus características (metros cuadrados, nº de habitaciones, incluye calefacción, distancia del centro, etc.) y deberemos incluir el precio que averiguamos en nuestro set de datos.

Por tanto, El aprendizaje supervisado es una rama de Machine Learning, un método de análisis de datos que utiliza algoritmos que aprenden iterativamente de los datos para permitir que los ordenadores encuentren información escondida sin tener que programar de manera explícita dónde buscar. El aprendizaje supervisado es uno de los tres métodos de la forma en que las máquinas "aprenden": supervisado, no supervisado y optimización. [14]

Wl aprendizaje supervisado resuelve problemas conocidos y utiliza un conjunto de datos etiquetados para entrenar un algoritmo para realizar tareas específicas. Utiliza modelos para predecir resultados conocidos como "¿Cuál es el color de la imagen?" "¿Cuántas personas hay en la imagen?" "¿Cuáles son los factores determinantes para el fraude o los defectos del producto?" etc.

Por ejemplo, un proceso de aprendizaje supervisado podría consistir en clasificar vehículos de dos y cuatro ruedas a partir de sus imágenes. Los datos de entrenamiento tendrían que estar correctamente etiquetados para identificar si un vehículo es de dos o cuatro ruedas.

El aprendizaje supervisado permite que los algoritmos 'aprendan' de datos históricos/de entrenamiento y los apliquen a entradas desconocidas para obtener la salida correcta. Para funcionar, el aprendizaje supervisado utiliza árboles de decisión, bosques aleatorios y Gradient Boosting Machine.

Por el contrario, el aprendizaje no supervisado es un tipo de Machine Learning que se utiliza para identificar nuevos patrones y detectar anomalías. Los datos que se introducen en los algoritmos de aprendizaje no supervisados no están etiquetados.

El algoritmo (o modelos) intentan dar sentido a los datos por sí mismos mediante la búsqueda de características y patrones. Una pregunta de muestra que Machine Learning sin supervisión podría responder es "¿Están surgiendo nuevos clústeres de fraude o patrones de compra o modos de falla?" El aprendizaje no supervisado utiliza el agrupamiento, los componentes principales, las redes neuronales y las máquinas de vectores de soporte.

La optimización, el tercer tipo de Machine Learning, encuentra la mejor solución incluso cuando existen restricciones complejas. Por ejemplo, la optimización podría responder a la pregunta "¿Cuál es la ruta óptima a seguir o la asignación de recursos o el programa de mantenimiento del equipo?" La optimización utiliza algoritmos genéticos, que se basan en la teoría de la evolución de Darwin.

¿Qué es la clasificación en el aprendizaje supervisado?

Existen dos tipos principales de aprendizaje supervisado; clasificación y regresión. La clasificación es el lugar donde se entrena a un algoritmo para clasificar los datos de entrada en variables discretas. Durante el entrenamiento, los algoritmos reciben datos de entrada de entrenamiento con una etiqueta de 'clasificación'. Por ejemplo, los datos de entrenamiento pueden consistir en las últimas facturas de

tarjetas de crédito de un conjunto de clientes, con la etiqueta de si realizaron una compra futura o no fue así. Cuando el saldo de la tarjeta de un nuevo cliente se presenta al algoritmo, este clasificará al cliente en el grupo de "comprará" o "no comprará".

¿Qué es la regresión en el aprendizaje supervisado?
A diferencia de la clasificación, la regresión es un método de aprendizaje supervisado en el que se entrena a un algoritmo para predecir una salida a partir de un rango continuo de valores posibles. Por ejemplo, los datos de entrenamiento inmobiliario tomarán nota de la ubicación, el área y otros parámetros relevantes, la salida será el precio de un inmueble específico.

En la regresión, un algoritmo necesita identificar una relación funcional entre los parámetros de entrada y salida. El valor de salida no es discreto como en la clasificación, sino que es una función de los parámetros de entrada. La exactitud de un algoritmo de regresión se calcula en función de la desviación entre la salida precisa y la salida prevista.

Aplicaciones prácticas de la clasificación

Clasificación binaria
Este algoritmo clasifica los datos de entrada en uno de dos grupos posibles. A menudo, una de las clases indica un estado "normal/deseado" y la otra indica un estado "anormal/no deseado". Las aplicaciones prácticas de la clasificación binaria incluyen:

Detección de spam
El algoritmo recibe ejemplos de correos electrónicos que están etiquetados como "spam" o "no spam" durante la fase de aprendizaje supervisado. Posteriormente, cuando el algoritmo recibe una nueva entrada de correo electrónico, predice si el correo corresponde a un "spam" o "no spam".

Predicción de migración de clientes

El algoritmo utiliza un conjunto de datos de entrenamiento de clientes que previamente cancelaron la suscripción de un servicio. Según el entrenamiento, el algoritmo predice si un nuevo cliente finalizará la suscripción o no en función de los parámetros de entrada.

Predicción de conversión

El algoritmo se entrena con los datos del comprador y si compró el artículo o no. Luego, basándose en esta capacitación, el algoritmo predice si un nuevo cliente realizará una compra o no.

Los principales algoritmos utilizados para la clasificación binaria incluyen la regresión logística y las máquinas de vectores de soporte.

Clasificación multiclase

En la clasificación multiclase, el conjunto de datos de entrenamiento se etiqueta con una de las múltiples clases posibles. A diferencia de la clasificación binaria, un algoritmo multiclase se entrena con datos que se pueden clasificar en una de las muchas clases posibles.

Las aplicaciones para la clasificación multiclase incluyen:

Clasificación de rostros: según los datos de entrenamiento, un modelo categoriza una foto y la asigna a una persona específica. Un detalle a tener en cuenta aquí es que podría haber una gran cantidad de etiquetas de clase. En este caso, miles de personas.

Clasificación de correo electrónico: la clasificación multiclase se utiliza para segregar los correos electrónicos en varias categorías: social, educación, trabajo y familia.

Los principales algoritmos utilizados para la clasificación multiclase son Bosques Aleatorios, Naive Bayes, árbol de decisiones, K-vecinos más cercanos y Gradient Boosting.

Clasificación de etiquetas múltiples

A diferencia de la clasificación binaria y multiclase donde el resultado tiene solo una clase posible, la salida de etiquetas múltiples pertenece a una o más clases, lo cual significa que los mismos datos de entrada podrían clasificarse en diferentes compartimentos. Las aplicaciones de la clasificación de etiquetas múltiples incluyen:

Detección de fotos: en los casos en que las fotos tienen varios objetos, como un vehículo, un animal y personas, la foto podría caer en varias etiquetas.

Clasificación de audio/video: las canciones y los videos pueden encajar en varios géneros y estados de ánimo. Se puede utilizar la clasificación de etiquetas múltiples para asignar estas etiquetas múltiples.

Clasificación de documentos: es posible clasificar artículos en función de su contenido.

Clasificación con datos desbalanceados:

Este es un caso especial de clasificación binaria, donde existe un desbalance de clases en el conjunto de datos de entrenamiento. La mayoría de los ejemplos de los datos de entrenamiento pertenecen a un conjunto y una pequeña parte pertenece al segundo conjunto.

Desafortunadamente, la mayoría de los algoritmos de Machine Learning funcionan mejor cuando existe una distribución equitativa entre las clases. Por ejemplo, en sus datos de entrenamiento, usted tiene 10.000 transacciones de clientes genuinos y solo 100 son fraudulentas.

Para igualar la precisión, se necesitan técnicas especializadas debido al desbalance en los datos. Las aplicaciones de la clasificación con datos desbalanceados podrían ser:
Detección de fraude: en el conjunto de datos etiquetados que se utilizan para el entrenamiento, solo una pequeña cantidad de entradas se etiquetan como fraude.
Diagnósticos médicos: en una gran cantidad de muestras, las que tienen un caso positivo de una enfermedad podrían ser mucho menos.

Se utilizan técnicas especializadas como enfoques basados en costos y enfoques basados en muestreo para ayudar a lidiar con casos de clasificación con datos desbalanceados.

Aplicaciones prácticas de la regresión

Regresión lineal:
La regresión lineal en el aprendizaje supervisado entrena a un algoritmo para encontrar una relación lineal entre los datos de entrada y salida. Es el modelo más simple utilizado donde las salidas representan una combinación linealmente ponderada de las salidas.

La regresión lineal se puede utilizar para predecir valores dentro de un rango continuo (por ejemplo, ventas, pronóstico de precios) o clasificarlos en categorías (por ejemplo, gato, perro - regresión logística). En los datos de entrenamiento para la regresión lineal, se proporcionan una variable de entrada (independiente) y una respectiva variable de salida (la variable dependiente).

A partir de los datos proporcionados de entrada que son etiquetados, el algoritmo de regresión calcula la intersección y el coeficiente x en la función lineal. Las aplicaciones de la regresión lineal pueden incluir:

Pronóstico: una de las aplicaciones más importantes de la regresión lineal es el pronóstico. El pronóstico puede ser de diferentes naturalezas. Las empresas utilizan la regresión lineal para pronosticar las ventas o los comportamientos de compra de sus clientes.

También se utiliza para predecir el crecimiento económico, las ventas de bienes raíces y los precios de productos básicos como el petróleo. La regresión lineal también se utiliza para estimar el salario óptimo para un nuevo empleado, basándose en los datos históricos de los salarios.

Regresión logística:

Se utiliza para determinar la probabilidad de que ocurra un evento. Los datos de entrenamiento tendrán una variable independiente, y el resultado deseado será un valor entre 0 y 1. Una vez que el algoritmo se entrena con la regresión logística, podrá predecir el valor de una variable dependiente (entre 0 y 1) en función del valor de la variable independiente (entrada).

La regresión logística utiliza la función sigmoidea clásica en forma de S. En la regresión logística en el contexto de aprendizaje supervisado, un algoritmo calcula los valores del coeficiente beta b0 y b1 a partir de los datos de entrenamiento proporcionados.

probabilidad = e^(b0 + b1 * X)

Las aplicaciones de la regresión logística incluyen:

Determinación de la probabilidad: Una de las principales aplicaciones de la regresión logística es determinar la probabilidad de un evento. La probabilidad de cualquier evento se encuentra entre 0 y 1, y ese es el resultado de una función logística.

Los algoritmos de regresión logística en Machine Learning se pueden utilizar para predecir los resultados de las elecciones, las probabilidades de un desastre natural y otros eventos similares.

Clasificación: aunque la regresión logística utiliza una función continua, algunas de sus aplicaciones están en la clasificación. Se puede utilizar para la segregación de imágenes y problemas de clasificación relacionados.

Regresión polinomial:

La regresión polinomial se utiliza para un conjunto de datos más complejo que no encajaría perfectamente en una regresión lineal. Un algoritmo se entrena con un conjunto de datos complejos y etiquetados que podrían no encajar adecuadamente en una regresión en línea recta. Si dichos datos de entrenamiento se utilizan con regresión lineal, podría causar un ajuste insuficiente, donde el algoritmo no capturará las tendencias verdaderas de los datos.

Las regresiones polinomiales permiten una mayor curvatura en la línea de regresión y, por lo tanto, una mejor aproximación de la relación entre la variable dependiente y la independiente.

El sesgo y la desviación son dos términos principales asociados con la regresión polinomial. El sesgo es el error en el modelado que se produce al simplificar la función de ajuste. La desviación también se refiere a un error causado por el uso de una función demasiado compleja para ajustar los datos.

Pasos básicos del aprendizaje supervisado

Para ejecutar y resolver un problema mediante Machine Learning supervisado, se deberá:
Seleccionar el tipo de datos de entrenamiento: el primer paso en el aprendizaje supervisado es determinar cuál es la naturaleza de los datos que se utilizarán para el entrenamiento. Por ejemplo, en el caso del análisis de escritura a mano, esto podría ser una sola letra, una palabra o una oración.

Recopilar y limpiar los datos de entrenamiento: en este paso, los datos de entrenamiento se recopilan de varias fuentes y se someten a una limpieza rigurosa de datos.

Elegir un modelo utilizando un algoritmo de aprendizaje supervisado: según la naturaleza de los datos de entrada y el uso deseado, elija un algoritmo de clasificación o de regresión. Pueden ser árboles de decisión, SVM, Naïve Bayes o bosques aleatorios. La consideración principal al seleccionar un algoritmo es la velocidad de entrenamiento, el uso de la memoria, la precisión de la predicción de nuevos datos y la transparencia/interpretación del algoritmo.

Entrenar el modelo: la función de ajuste se perfecciona a través de múltiples iteraciones de datos de entrenamiento para mejorar la precisión y la velocidad de predicción.

Realizar predicciones y evaluar el modelo: una vez que la función de ajuste sea satisfactoria, se podrán proporcionar nuevos conjuntos de datos al algoritmo para realizar nuevas predicciones.

Optimizar y volver a entrenar el modelo: la degradación de datos es una parte natural de Machine Learning. Por lo tanto, los modelos se deberán volver a entrenar periódicamente con datos actualizados para garantizar la precisión.

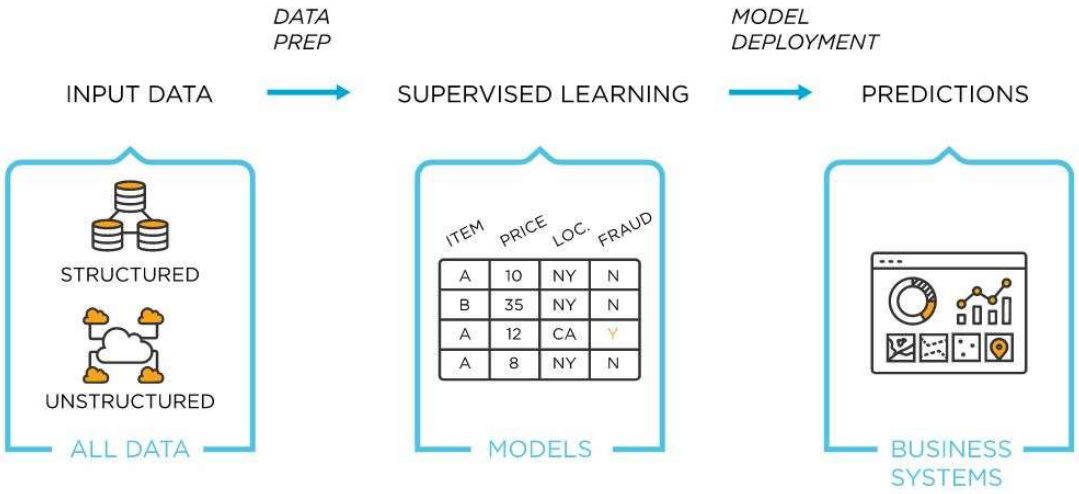


Figura 3. Estructura del Aprendizaje Supervisado. [14]

Aprendizaje No Supervisado

En el aprendizaje No Supervisado los datos de entrenamiento no incluyen Etiquetas y el algoritmo intentará clasificar o descifrar la información por sí solo. Un ejemplo en el que se usa es para agrupar la información recolectada sobre usuarios en una Web o en una app y que nuestra Inteligencia detecte diversas características que tienen en común.

También, el aprendizaje no supervisado:

es una de las formas en que Machine Learning (ML) "aprende" los datos. El aprendizaje no supervisado tiene datos sin etiquetar que el algoritmo tiene que intentar entender por sí mismo. El aprendizaje supervisado es en el que se etiquetan los conjuntos de datos para que haya una clave de respuestas con la que la máquina pueda medir su precisión. Si Machine Learning fuera un niño que

aprendiera a andar en bicicleta, el aprendizaje supervisado es el padre que corre detrás de la bicicleta y la sostiene en posición vertical. El aprendizaje no supervisado consiste en entregar la bicicleta, darle palmaditas en la cabeza al niño y decirle "buena suerte". [15]

El objetivo es simplemente dejar que la máquina aprenda sin ayuda o indicaciones de los científicos de datos. En el camino, también deberá aprender a ajustar los resultados y agrupaciones cuando haya resultados más adecuados, permitiendo que la máquina comprenda los datos y los procese como mejor le parezca.
El aprendizaje no supervisado se utiliza para explorar datos desconocidos. Puede revelar patrones que podrían haberse pasado por alto o examinar grandes conjuntos de datos que serían demasiado para que los abordara una sola persona.
Para comprender el aprendizaje no supervisado, antes, tendremos que comprender el aprendizaje supervisado. Si una computadora estuviera aprendiendo a identificar frutas en un entorno de aprendizaje supervisado, se le darían imágenes de ejemplo de frutas etiquetadas, a esto se le llama datos de entrada.

Por ejemplo, las etiquetas dirían que los plátanos son largos, curvos y amarillos, que las manzanas son redondas y rojas, mientras que una naranja es esférica, de aspecto ceroso y anaranjada.

Después de un tiempo conveniente, la máquina debería poder identificar con seguridad qué fruta es cuál, basándose en esos descriptores. Si se le presenta una manzana, por ejemplo, podría decir con seguridad que no es de color naranja, por lo tanto, no es una naranja, pero también que no es amarilla y larga, por lo tanto, no es un plátano.

Entonces, será una manzana debido a que es redonda y roja. Por el contrario, el aprendizaje no supervisado es cuando no existe ninguna categorización o

etiquetado de los datos. La máquina no tendrá idea del concepto de fruta, por lo que no podrá etiquetar los objetos.

Sin embargo, podrá agruparlos según sus colores, tamaños, formas y diferencias. La máquina agrupará las cosas de acuerdo con las similitudes, encontrando estructuras y patrones ocultos en datos sin etiquetar. No existe un camino correcto o incorrecto, ni tampoco un maestro. No existen resultados, solo un análisis puro de los datos.

El aprendizaje no supervisado utiliza una variedad de algoritmos para ajustar los datos en grupos amplios, clústeres y asociaciones.

Algoritmos de agrupación en clústeres en el aprendizaje no supervisado

La agrupación en clústeres se presenta cuando los objetos se agrupan en subconjuntos llamados clústeres. Esta es una de las mejores formas de obtener una descripción general de la estructura de sus datos. Habrá algunas características similares en estos grupos. Este método está diseñado para tener grupos con las mismas características y luego asignarlos a los respectivos clústeres.

Agrupación en clústeres jerárquica

Esto ocurre cuando la máquina agrupa las cosas que van juntas en un árbol de clústeres. Todos los datos son un grupo, luego se dividen en grupos cada vez más pequeños. Los datos pertenecerán a un conjunto en cascada de clústeres, desde los más genéricos hasta los más específicos y estrechamente agrupados. Entonces, el resultado final será que usted verá cómo los diferentes subgrupos se relacionan entre sí o qué tan separados están.

Agrupación en clústeres k-medias

Este algoritmo separa los datos en distintos clústeres que no se etiquetaron en los datos. Según la precisión de la asociación los datos están más o menos cerca del centro.

Los puntos de datos pueden pertenecer a un solo clúster. Una k más amplia significa que el grupo es más pequeño y con más granularidad. A cada clúster se le asigna una etiqueta de punto de datos.

Modelos de mezcla gaussiana

Sobre la base de una distribución de curva de campana normal, los clústeres de grupos se distribuyen a lo largo de las densidades normales previstas, mostrando subpoblaciones en los datos generales.

Agrupación en clústeres difusa

Estos clústeres pueden superponerse, por lo que cada punto de datos puede pertenecer a tantos clústeres como sea relevante en contraposición a los clústeres no difusos donde los puntos de datos solo pueden pertenecer a un clúster. Este es el diagrama de Venn del mundo del aprendizaje no supervisado.

La agrupación en clústeres asume relaciones entre grupos, por lo que no siempre es la mejor manera de segmentar los clientes. Este algoritmo no trata los puntos de datos como individuos. Usted necesitará aplicar más métodos estadísticos para analizar los datos en más detalle.

Asociación en el aprendizaje no supervisado

En Machine Learning, el algoritmo crea reglas que detectan asociaciones entre puntos de datos. Se detectan las relaciones entre variables y se identifican elementos que tienden a ocurrir juntos. Por ejemplo, si se analizan canastas en el supermercado se identifican los artículos que la gente tiende a comprar al mismo tiempo, por ejemplo: sopa y panecillos. O, cuando la gente compra una casa

nueva, ¿qué otras cosas nuevas compra probablemente? Este algoritmo es excelente para identificar oportunidades de marketing.

Modelos de variables latentes en el aprendizaje no supervisado

Un modelo de variables latentes muestra la relación entre las variables observables (o variables manifiestas) y las que están ocultas o no se pueden observar (variables latentes). Los modelos de variables latentes se utilizan principalmente en el procesamiento previo o limpieza de datos, para reducir las características de un conjunto de datos o dividir el conjunto de datos en varios componentes.

Debido a que la máquina no sabe que existe una respuesta 'correcta', permitir que las decisiones se tomen sobre los datos basándose únicamente en la información (es decir, sin sesgos por parte del científico) les permite a los científicos de datos aprender más sobre estos. Los algoritmos pueden encontrar estructuras interesantes u ocultas en los datos que antes no eran visibles para los científicos de datos. Estas estructuras ocultas se denominan vectores de características.

Normalmente los datos están sin etiquetar, por lo que el aprendizaje no supervisado evita que un científico de datos tenga que etiquetar todo, lo que podría ser una tarea que requiere mucho tiempo y, frecuentemente, imposible de completar. Los algoritmos de aprendizaje no supervisados también permiten tareas de procesamiento más complejas. Una vez más, la ausencia de etiquetado significa que se pueden mapear relaciones complicadas y grupos de datos. La ausencia de etiquetado en los datos significa que no existen ideas preconcebidas ni sesgos.

El mejor momento para utilizar el aprendizaje no supervisado es cuando no existen datos preexistentes sobre los resultados preferidos. El aprendizaje no supervisado puede identificar características que pueden resultar útiles en la categorización de conjuntos de datos desconocidos. Por ejemplo, cuando una empresa necesita determinar el mercado objetivo de un producto nuevo.

El aprendizaje no supervisado utiliza una técnica llamada reducción de dimensionalidad, lo cual ocurre cuando la máquina asume que muchos datos son redundantes y elimina dimensiones o combina algunas partes de datos según corresponda. La compresión de datos da como resultado ahorro de tiempo y ahorro en potencia de procesamiento.

Los modelos generadores son otro punto fuerte del aprendizaje no supervisado. Los modelos generadores muestran la distribución en los datos. Aquí es cuando se revisan los datos y se pueden crear nuevas muestras a partir de ellos. Por ejemplo, a un modelo generador se le puede dar un conjunto de imágenes y crear un conjunto de imágenes fabricadas a partir de ellas.

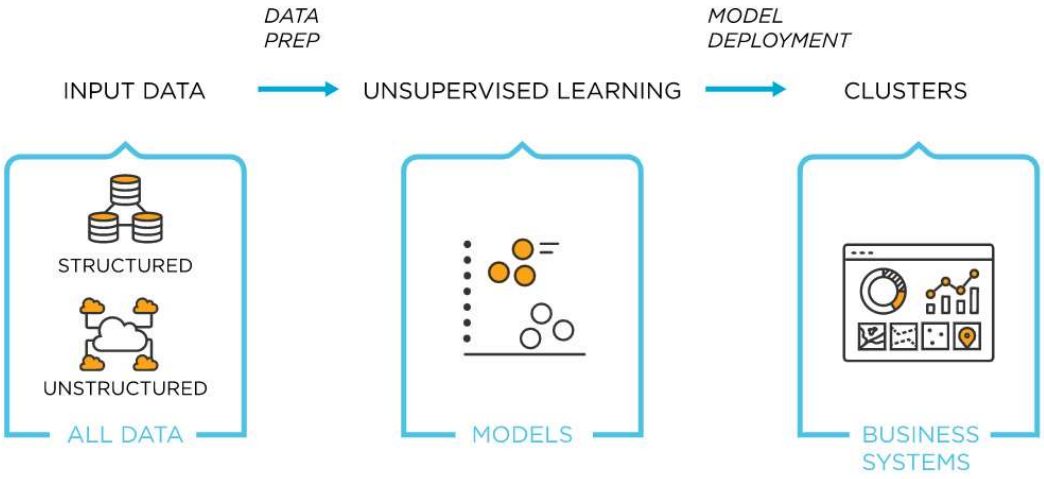


Figura 4. Estructura Aprendizaje no Supervisado. [15]

Machine Learning es una Aplicación de Inteligencia Artificial. [16]

Por tanto, Machine Learning (ML) es una aplicación de inteligencia artificial en la que los programas informáticos utilizan algoritmos para encontrar patrones en los datos. Pueden hacerlo sin estar programados específicamente para ello, sin depender de un ser humano. En el mundo actual, los algoritmos de Machine

Learning están detrás de casi todos los avances tecnológicos y aplicaciones de inteligencia artificial (AI) que existen en el mercado.

Figura 5. Sistemas de inteligencia artificial. [16]

Los sistemas de AI generalmente tienen la capacidad de planificar, aprender, razonar, resolver problemas, percibir, desplazarse e incluso manipular. Machine Learning es uno de los muchos enfoques que se utilizan en los sistemas de inteligencia artificial. Otros incluyen computación evolutiva y sistemas expertos.

Machine Learning es parte de muchas cosas que hacemos todos los días. Considere algunas circunstancias donde los sistemas de Machine Learning podrían influir en su vida:

Los sistemas de recomendación en sus servicios de transmisión favoritos como Netflix o Spotify se ejecutan mediante Machine Learning.

Los motores de búsqueda utilizan Machine Learning para resolver y optimizar sus resultados de búsqueda.

Los canales de redes sociales sugieren amigos, grupos y también videos.

Si tiene un refrigerador moderno, a menudo estos aparatos aprenden cuándo más los utiliza y lo tienen listo antes de la hora de la comida.

El GPS anticipa qué partes de su ruta tendrán mucho tráfico y lo redireccionará utilizando algoritmos de Machine Learning.

Los asistentes de voz como Alexa y Siri utilizan Machine Learning para operar.

Cada una de estas plataformas acumula datos de las elecciones diarias que usted realizará, aprenden de usted y, a partir de la información obtenida, hacen predicciones sobre lo que verá a continuación, a qué hora preparará la cena o adónde viajará o comprará.

Todos estos datos accionan los algoritmos de Machine Learning, que luego ayudarán a una marca a anticipar lo que usted hará o comprará posteriormente. No solo eso, sino que sus gustos y preferencias se combinan con otros puntos de datos de millones de personas, lo que permitirá a las empresas crear listas de sugerencias precisas y altamente efectivas.

La AI está preparada para responder a nuevos desafíos utilizando las aplicaciones de Machine Learning.

Aplicaciones de Machine Learning

Las aplicaciones de Machine Learning son amplias. A continuación, se muestra cómo se está utilizando en áreas principales que son parte integral de la vida humana cotidiana.

Machine Learning en la educación

Cuando se aplica en el campo de la educación, Machine Learning puede ayudar a los maestros a examinar el tipo de lecciones que los estudiantes podrán tomar. También, podrán evaluar cómo se manejan los estudiantes con las lecciones enseñadas: cuánto son capaces de comprender, cuáles son los temas comunes con los que los estudiantes tienden a tener dificultades y cuáles son los más sencillos, lo cual ayudará a los maestros a planificar mejor las lecciones e identificar a los estudiantes que podrían estar rezagados, lo que permitirá interacciones e intervenciones mucho más efectivas.

Machine Learning en motores de búsqueda

Cuando escribe un término de búsqueda en Google, es frustrante cuando los resultados que aparecen no son los que se busca. Machine Learning ha sido una parte integral de la optimización de motores de búsqueda durante mucho tiempo. Ayuda constantemente a los motores de búsqueda a mostrar resultados más relevantes para las búsquedas. También ha ayudado a potenciar los servicios de búsqueda con asistentes de voz, búsquedas de imágenes y varias otras funciones relacionadas con la búsqueda.

Machine Learning en marketing digital

La personalización es la clave para las modernas campañas de marketing digital y Machine Learning ha sido fundamental para lograrlo. Con datos basados en interacciones con los consumidores, Machine Learning ayudó a las empresas a personalizar sus enfoques hacia los clientes potenciales, enfocando los mensajes correctos justo en el momento adecuado. Desde correos electrónicos personalizados hasta ventas cruzadas basadas en compras recientes, Machine Learning ayudó a las empresas a aprovechar sus datos sobre el comportamiento del consumidor.

Machine Learning en el cuidado de la salud

Machine Learning tuvo una amplia aplicación en el campo médico. El diagnóstico mediante imágenes médicas es un caso importante en el que Machine Learning funciona con herramientas de diagnóstico. Machine Learning visualiza las imágenes médicas, identifica áreas que son inusuales o anormales, sin ningún tipo de sesgos que podrían ser el caso de un profesional médico.

Machine Learning también se está utilizando para ayudar a los médicos a tratar casos únicos de enfermedades específicas brindándoles sugerencias sobre protocolos de tratamiento basados en información recopilada de otros casos. Por ejemplo, un estudio de macrófagos se podría rastrear en algunas horas mediante

aparatos que identifican fagos probablemente eficaces para tratar cepas de bacterias resistentes a los antibióticos.

La aplicación también está experimentando la forma de convertir los datos agrupados de los consumidores recopilados a partir de dispositivos personales para ofrecer a los profesionales médicos sugerencias y opciones de tratamiento. Esta es, por supuesto, un área en constante evolución.

Las aplicaciones para Machine Learning son diversas y se pueden encontrar en casi cualquier campo o tipo de actividad comercial. Los beneficios son enormes para las empresas comerciales, gubernamentales y sociales.

Ventajas de Machine Learning

Machine Learning tiene ventajas increíblemente amplias en casi todas las facetas de la vida. Estas son solo algunas de las ventajas universales de Machine Learning:

Predecir el comportamiento del cliente

Los análisis de los patrones de compra de los consumidores ayudan a las empresas a comprender el camino a seguir para las líneas de productos y servicios. Estos patrones pueden ser tan precisos como las razones por las que un cliente podría optar por un producto sobre otro, las influencias del precio, la temporada, la lealtad a la marca y otras más en estas decisiones. Estos hallazgos orientados a los datos se obtienen mucho más rápido con Machine Learning y la velocidad es la clave para una toma de decisiones más inteligente.

La más aburrida de las tareas humanas está relacionada con la entrada de datos. Las posibilidades de error son elevadas con tareas muy repetitivas. Estos errores podrían resultar costosos para una empresa en varios niveles. Machine Learning garantiza que la entrada de datos se complete rápidamente, con precisión, sin dejar lugar a errores. También elimina las tareas rutinarias de los empleados, lo

que les permite concentrarse en trabajos más desafiantes y beneficiosos para la empresa.

Identificar clientes potenciales en las experiencias del usuario

Cada empresa crece sobre la base de nuevos clientes potenciales que se convierten en compradores reales. Ser capaces de llevar la delantera tiene que ver con evolucionar para satisfacer las necesidades del cliente. Machine Learning ayuda a las empresas a sumergirse en los viajes de los clientes y proporcionar información sobre las tendencias y anticipar sus necesidades. Los estudios demuestran que Machine Learning marcó una diferencia en la trayectoria de crecimiento ascendente de las empresas al ayudarlas a predecir el comportamiento de sus clientes, encontrar las deficiencias, etc.

Mantener una ventaja competitiva

Las empresas pueden crecer junto con el mercado cuando tienen una buena inteligencia de negocio a la que recurrir. Machine Learning cumple un papel importante al proporcionar a las empresas información sobre sus puntos de venta únicos y sus aspectos positivos en comparación con las marcas de la competencia. Cualquier enfoque nuevo podrá formularse rápidamente como una hipótesis y evaluarse en función de los datos disponibles ayudando a las empresas a crear rápidamente un plan de comercialización.

Potenciar asistentes virtuales

Los lugares de trabajo, grandes o pequeños, se disponen a aumentar la eficiencia y hacer un uso inteligente de las horas de trabajo. Cuando se aplica Machine Learning al reconocimiento automático de voz, ayuda a crear asistentes virtuales más inteligentes y eficientes, que pueden tomar notas, desarrollar actas de reuniones y mantener mejores registros. Todo esto reduce la documentación rutinaria que es esencial pero agotador. Con mejores asistentes virtuales, se garantiza la precisión y se cumplen bien las normas de privacidad.

Categorizaciones de algoritmos de Machine Learning

Los algoritmos forman la base de toda la estructura de Machine Learning y su desarrollo. Estos algoritmos se pueden dividir en cuatro categorías principales:

Algoritmos supervisados de Machine Learning

Aquí, las lecciones aprendidas anteriormente se pueden aplicar a nuevos datos con la ayuda de ejemplos etiquetados para predecir resultados futuros . Esto comienza con el análisis de conjuntos de datos conocidos de aprendizaje. El algoritmo de aprendizaje crea una función inferida que hará predicciones de posibles resultados. Con la cantidad necesaria de aprendizaje, todas las nuevas entradas de datos se proporcionarán con objetivos.

Algoritmos no supervisados de Machine Learning

Estos están en contraste con los algoritmos supervisados y entran en juego cuando la información de aprendizaje no está etiquetada o clasificada absolutamente. El aprendizaje no supervisado no proporciona resultados "correctos" para los nuevos datos. En su lugar, estos algoritmos exploran los datos, extraen inferencias a partir de los conjuntos de datos y revelan cualquier estructura oculta que pueda estar en los datos sin etiquetar.

Algoritmos semi-supervisados de Machine Learning

Estos algoritmos siguen la línea media entre los dos primeros tipos, debido al uso de datos etiquetados y no etiquetados para el aprendizaje. Normalmente, la cantidad de datos sin etiquetar es mayor que la cantidad de datos etiquetados y el algoritmo utiliza los datos etiquetados para conocer los datos sin etiquetar. Los sistemas sobre esa base mejoran constantemente el nivel de precisión del aprendizaje.

Algoritmos reforzados de Machine Learning

Este es un método de aprendizaje donde la interacción con el entorno produce acciones y descubre errores y recompensas. Con este enfoque, los aparatos y todos los agentes de software pueden determinar el comportamiento adecuado dentro de un contexto específico para obtener el mejor rendimiento posible.

Los desafíos del Machine Learning

A pesar de todos los avances tecnológicos, todavía hay una serie de desafíos que Machine Learning deberá superar.

Las redes todavía necesitan grandes cantidades de memoria de trabajo para almacenar y procesar datos. Si bien algunas técnicas de aprendizaje no supervisado eliminan datos innecesarios, todavía existe la necesidad de una enorme potencia de procesamiento, lo cual resolverá parcialmente, con algoritmos de aprendizaje no supervisado que eliminarán los datos innecesarios, lo que reducirá la potencia de procesamiento necesaria. Sin embargo, esto no es suficiente para todos los escenarios.

El procesamiento del lenguaje natural está todavía muy lejos de ser una traducción natural y precisa. La jerga, los acentos y la comprensión del idioma siguen siendo grandes desafíos para Machine Learning. Si bien la máquina tiene constantemente nuevos datos para escuchar y aprender, todavía necesita mucha capacitación para resolver aquellos acentos poco claros.

AI washing ocurre cuando la tecnología se etiqueta como inteligencia artificial (o un ordenador inteligente), cuando en realidad es solo Machine Learning o los mismos algoritmos antiguos que se utilizaron desde siempre. Para muchas personas, esta diferencia no es importante, pero sobrepasa las expectativas tecnológicas, socava la confianza en la tecnología y ambos campos crean una reacción fuerte. Se necesita la educación del público en general y una mayor comprensión de la inteligencia artificial y Machine Learning.

La falta de capacitación en vídeo está frenando a la industria. En lugar de depender de imágenes estáticas y un mundo 2D, el video proporciona conjuntos de datos mucho más enriquecedores. Nuestro mundo es dinámico y nuestras máquinas necesitan aprender de eso. Este es un campo de estudio emergente.

Las máquinas no piensan como los seres humanos. La gente utiliza la heurística para tomar decisiones rápidas y utilizan un amplio campo de atención para integrar una comprensión holística de una escena. Pero Machine Learning todavía sigue siendo una variedad de datos, lo que limita las formas actuales en que se pueden utilizar de manera efectiva. A medida que las máquinas aprendan más, esto se resolverá, pero es incierto si alguna vez pensarán realmente como seres humanos o se volverán "artificialmente inteligentes".

6. ESTUDIO DE MEDICIONES DE LOS NIVELES DE RSRP PARA TECNOLOGÍA LTE EN UN OPERADOR DE TELECOMUNICACIONES

A continuación, se muestra, en la siguiente figura, los sitios en los cuáles se realizará el estudio de co ubicación a partir de los niveles de potencia, factor de zona y distancia entre las estaciones base.

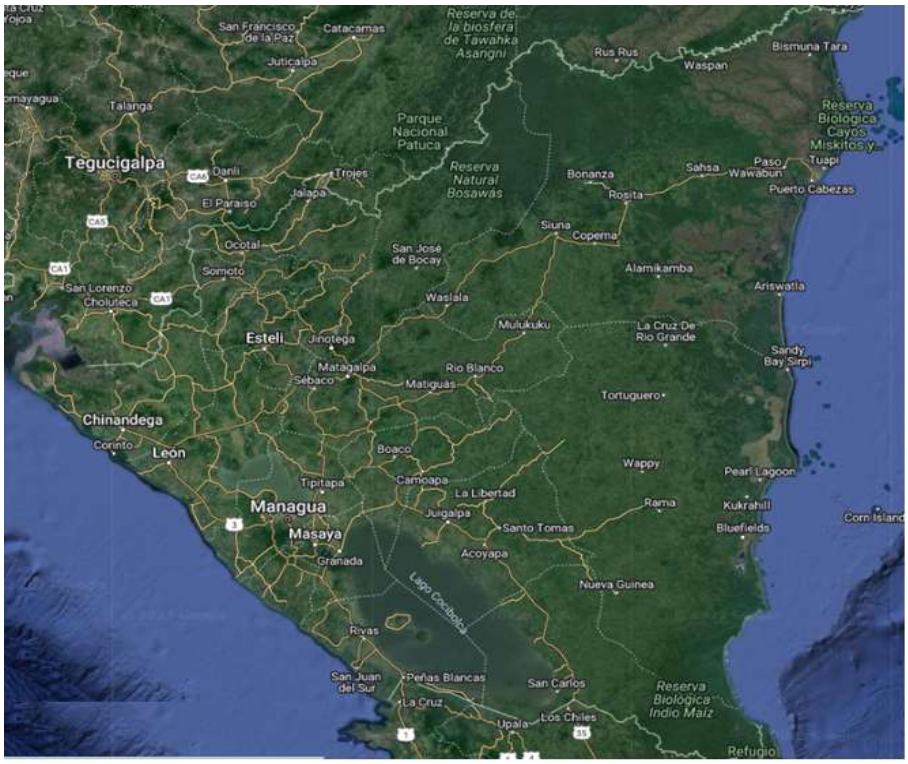


Figura 2. Distribución de estaciones bases para la realización de mediciones

La importancia de la realización de Bench Marking para conocer el estado del arte de las operadoras en muy significativo ya que permite conocer las trayectorias y áreas que no están siendo cubiertas. Mediante el Drive Test se podrá hacer un análisis "Land Banking" que permitirá proponer estructuras que permitan la instalación de equipos RF para satisfacer las zonas de no cobertura. En este Planning Plan se pretende describir la dirección y gestión del proyecto Drive Test para la medición de niveles de potencia en 4G en los sitios de interés.

Por tanto, se realizó un Drive Test en los sitios de interés que permita la recolección de los factores de cobertura que tienen los sitios existentes de los operadores de servicios de telefonía móvil, y así comparar de potencia estimados por predicciones y los obtenidos en Drive Test para determinar el factor de Co Ubicación del portafolio de sitios que se estudiará en el presente estudio.

- **Requisitos del Proyecto**

a. Requisitos Técnicos

- **Descripción del equipo (Pilot RCU 2.0)**

Pilot RCU 2.0 es una sonda autónoma para la recolección de datos y medición que se utiliza principalmente para el control de la red para potencia, calidad, voz y verificación de servicio de datos, y realiza una evaluación comparativa. Se compone de una variedad de módulos de prueba que apoyan múltiples estándares de redes (2G, 3G & 4G). Pilot RCU 2.0 ofrece un sistema incorporado en la plataforma de procesamiento de datos, un diseño con un alto rendimiento y la estabilidad, y bajo consumo de energía.

- **Requisitos de funcionamiento del equipo (Pilot RCU 2.0)**

  - Para el servidor Fleet se requiere una IP pública, estática, con acceso a Internet. En este caso no es recomendable el uso de redes privadas o corporativas donde se haga uso de aplicaciones VPN para la configuración del servidor Fleet, ya que esto requeriría de solicitar a los operadores de red acrear APNs para aplicaciones especiales.

  - Para las Pilo RCU 2.0 se requiere SIM cards con lo saldo para hacer pruebas de voz y de datos.

  - Se requieren también tener protocolos de pruebas definidos para voz y datos.

- **Proceso de medición, recolección de información, post process & realización de informe por el ingeniero RF.**

  - Pilot RCU 2.0 recolecta la información en función de la configuración que se requiera, posteriormente transfiere esa información al servidor Fleet. El Sistema Pilot Fleet es capaz de gestionar de forma remota el examen de manejo del proceso de recolección de datos de todos los equipos DINGLI y la gestión de datos centralizada.

  - Posteriormente se configura el Fleet para que envié la información al servidor donde está Gladiator para realizar el Post Process.

  - Esa información es enviada nuevamente al ingeniero RF para que realice el Post Process de los Logs.

  - Realizar el informe de rutas conteniendo los niveles de potencia en 3G y 4G.

  - El Project Manager realizará el Reporte de estatus semanalmente.

- El PM y el Ingeniero RF realizará la comparación de los niveles de potencia medido mediante Drive Test y los estimados por Predicciones. Después se elaborará un informe con ambos resultados para el cálculo de los niveles de Co Location de los sitios de interés.

b. Requisitos de personal

- **Perfil del Ingeniero RF que se requiere para el manejo del Pilot RCU 2.0**

El Ingeniero RF debe tener un título superior en ingeniería en Electrónica, Computación, Sistemas, Telemática o Telecomunicaciones. Posee conocimientos en tecnologías 2G, 3G & 4G. Sirve como apoyo para el despliegue de los operadores telefonía móvil.

Realiza predicciones para determinar la cobertura, sobre propagación, relación señal ruido e interferencia, que sirven para el determinar el comportamiento de la propuesta del diseño del plan nominal y candidatos con nuevos emplazamientos. Propone las características de los parámetros estaciones base (azimuth, tilt eléctrico, tilt mecánico, etc).

Hace análisis de drive-test. Propone opciones como candidatos para presentarse al cliente con un alto grado de colaboración e integración de la infraestructura a implementar en el proceso de construcción. Apoya al operador coordinando tareas con propuestas que satisfagan al cliente e intercambiar información con el fin de lograr sinergia y cumplir con los objetivos, metas y plan estratégico de nuestra organización y de la empresa operadora.

- **Estructura de desglose de trabajo**

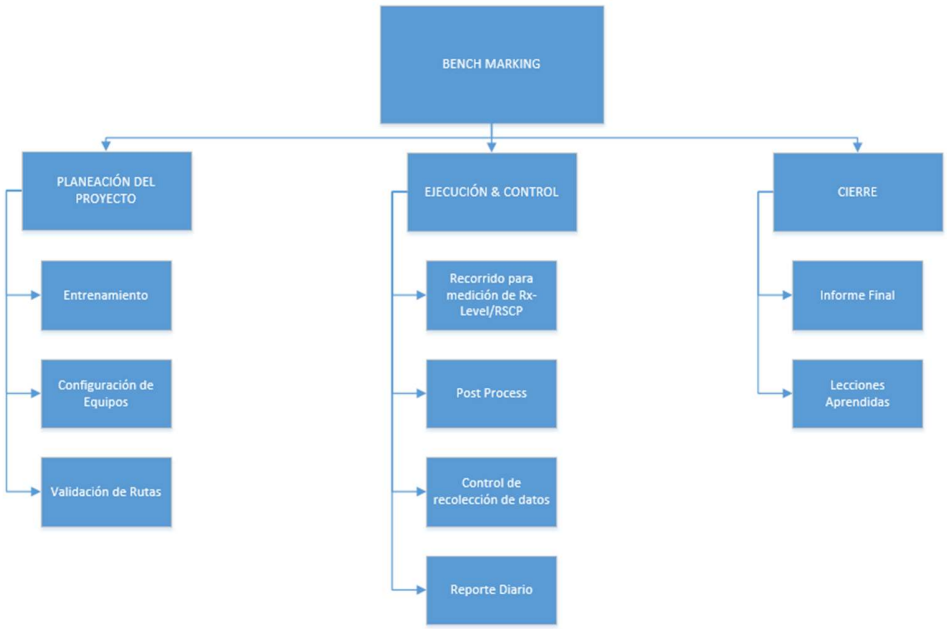


Figura 3. Estructura de desglose de trabajo

Alcance del Proyecto:

Se realizó pruebas de potencia en 4G para los sitios, posteriormente se realizó el Post Procesamiento para elaboración de informe de resultados de los niveles de potencia obtenidos en Drive Test para hacer la comparación con los niveles de potencia estimados en las predicciones para determinar los niveles de Co ubicación del portafolio de sitios.

- **Gestión del Tiempo**

La unidad de medida de tiempo se realizará en semanas. El tiempo total del de las mediciones fue de 14 semanas, el entrenamiento y la configuración de los equipos para hacer el proceso de Drive Test.

- **Cronograma de actividades**

Tabla 1. Cronograma de actividades del proceso de medición

Actividad	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10	S11	S12	S13	S14
1.														
2.														
3.														
4.														
5.														

Descripción de actividades

Actividad 1: Entrenamiento, Configuración de los equipos, Planeación y Validación de Rutas.

Actividad 2: Recorridos para medición

Actividad 3: Post Process

Actividad 4: Entregable (Informe de recorrido & Status de Proyecto)

Actividad 5: Elaboración de informe final y lecciones aprendidas.

- **Flujo de Proceso de las Actividades del Proyecto**

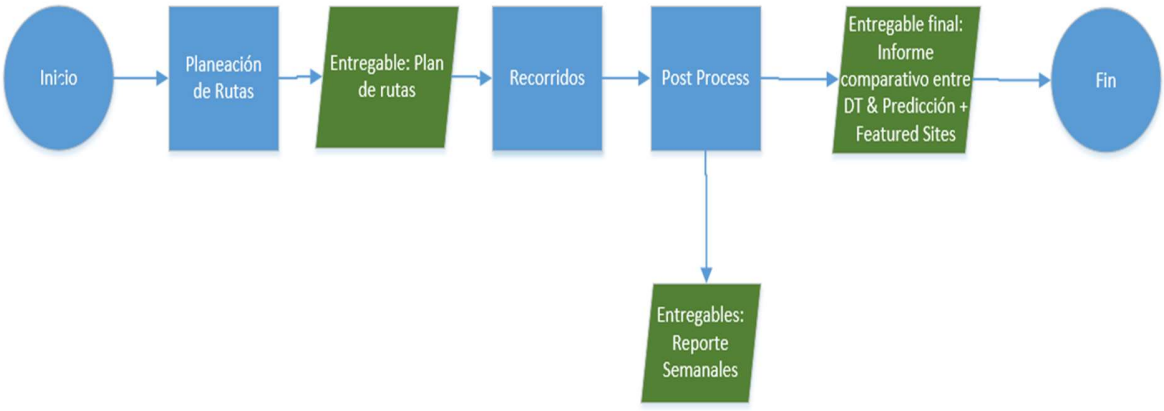


Figura 4. Flujo de proceso

- **Entregables:**

Reporte Semanales de los niveles de potencia

Informe comparativo entre DT y análisis en herramienta de predicción + anillos de búsqueda para featured sites (land banking)

- **Costos**

Se utilizará la moneda dólar para determinar los costos de las actividades. Para estimar los costos se hará uso de la herramienta “Estimación Ascendente” y juicio de expertos.

- Línea base de costos para la realización de las mediciones.

Concepto	Mes 1	Mes 2	Mes 3	Total
1.Salario	U$ 1,200	U$ 1,200	U$ 1,200	U$3,600
2.Vehículo	U$ 900	U$ 900	U$ 900	U$2,700
3.Viáticos (Incluyendo hospedaje)	U$ 1,140	U$ 1,140	U$ 1,140	U$3,420
4.Combustible	U$ 975	U$ 975	U$ 975	U$2,925
Total	U$4,215	U$4,215	U$4,215	**U$12,645**

- **Control de costos**

El control de los costos se realizará a partir de los entregables definidos, estos entregables serán los informes semanales que desarrollará el ingeniero RF con las rutas planeadas. La herramienta que se utilizará para determinar el estado del proyecto en costos y tiempo será “Valor Ganado” (EV).

7. MACHINE LEARNING EN PYTHON.

El machine learning en Python es la forma más popular de desarrollar estos modelos. Desde su lanzamiento en 1991 el lenguaje de programación Python ha tenido un crecimiento asombroso, convirtiéndose en uno de los preferidos por los programadores y con el auge de la inteligencia artificial tomo el liderato al ser el lenguaje de programación más utilizado para el desarrollo de este tipo de proyectos. [17]

A continuación, se determinarán algunas de las razones por las cuales Python es el lenguaje favorito de los desarrolladores de machine learning.

Python es un lenguaje de programación interpretado que busca desarrollar una sintaxis que priorice la legibilidad del código. Este lenguaje de programación es conocido como multiparadigma ya que soporta diferentes orientaciones. En Python podrás orientar el código a objetos, a programación imperativa y funcional.

Además, con algunas extensiones pueden soportarse paradigmas de programación adicionales, esto favorece la utilización de diversos estilos. Esta libertad es una de las características que ha logrado que miles de programadores se sumen a desarrollar en Python. Conozcamos algunas otras características que posee este lenguaje que favorece el desarrollo de proyectos de machine learning.

La legibilidad del código desarrollado en Python es sencillo, elegante y busca ser consistente. Esto permite que la estructura del lenguaje se asemeje a la estructura que implementamos los seres humanos y a lo que conocemos como lenguaje matemático permitiendo que este sea leído como un pseudocódigo.

Su facilidad de implementación ha ayudado a que Python sea el lenguaje con mayor crecimiento en la actualidad.

Adicionalmente a esto, Python cuenta con iteraciones rápidas de datos que favorecen la concentración en los datos y en el desarrollo de los algoritmos. Python se caracteriza por funcionar como un lenguaje puente entre el mundo científico y el mundo empresarial.

Cumple con una función de ser la pieza de rompecabezas perfecta para conectar ambos ecosistemas ya que facilita la creación de códigos entendible de rápido aprendizaje como los que son necesarios en proyectos de machine learning. [17]

Las librerías de Python son amplias. Existen miles de librerías de data science y matemáticas, pero el sistema de empaquetamiento de este lenguaje permite construir librerías nuevas sobre las ya existentes para que estas sean más amplias y potentes. Por último, es importante destacar que la capacidad de combinar librerías como NumPy y ScyPi, permite que Python sea uno de los lenguajes con mejor rendimiento para realizar proyectos de machine learning.

Para desarrollar proyectos de cualquiera de los tipos de machine learning en Python necesitamos la creación de un entorno virtual en el que podamos desarrollar con comodidad el código. Para esto existen distribuciones como Anaconda en la que podemos activar entornos estables y en ellos instalar las librerías que necesitamos. Conozcamos algunas que nos serán de mucha utilidad.

Scikit:

Es una biblioteca para aprendizaje automatizado construida en software libre especialmente para programar en Python. En ella podemos encontrar algoritmos de clasificación de regresión lineal y análisis de grupos. Es perfecta para operar en conjunto te las librerías numéricas y científicas NumPy y SciPy.

Open CV:

Es una librería de inteligencia artificial que en sus inicios fue desarrollada por el gigante tecnológico Intel. Ha sido implementada en diferentes tipos de proyectos como sistemas de detección de movimiento o hasta reconocimiento visual de objetos. Es una librería multiplataforma que tiene versiones estables en GNU/Linux, MacOs X, Windows y Android. Contiene más de 500 funciones que facilitan el desarrollo de proyectos de calibración de cámaras, visión robótica y reconocimiento de objetos.

Matplotlib:

La libreria Matplotlib está diseñada para la generación de gráficos a partir de conjuntos de datos que deben estar contenidos en listas o en arrays de programación Python. Esta librería cuenta con una API diseñada de forma similar a la que contiene MATLAB.

TensorFlow:

Esta librería de código abierto es desarrollada por Google y es utilizada para construir y entrenar redes neuronales en la detección y descifrado de patrones y correlaciones. Es actualmente utilizada en múltiples proyectos desde que se publicara una versión de código abierto Apache 2.0 en el año 2015.

Ambientes en la nube para desarrollar proyectos de Machine Learning en la nube
En la actualidad la nube es nuestro segundo hogar. Están quedando en el pasado las cuantiosas inversiones de dinero en hardware para albergar desarrollos tecnológicos.

Para trabajar machine learning en Python existen en la actualidad diversas herramientas o entornos de programación en la nube con los que podemos contar. En estos entornos contamos con todas las herramientas y librerías necesarias para iniciar el desarrollo de nuestros códigos en los Jupyter Notebooks.

Entre los espacios más destacados para desarrollar proyectos de machine learning en Python en la nube contamos con Google Colaboratory y con Microsoft Azure Notebooks. Ambos espacios permiten acceder a los Jupyter Notebooks en una interfaz en línea muy cómoda y funcional. [17]

8. DESARROLLO DEL CODIGO BASADO EN APRENDIZAJE SUPERVISADO.

Una vez que se tienen las mediciones, cálculo de distancias y determinar el factor de zona, se desarrolla un documento en formato .csv, el cuál será utilizado para el desarrollo del algoritmo de machine learning en aprendizaje supervisado.

Se utilizó Spyder, para la programación en Python. Teniendo como resultado lo siguiente:

```
import pandas as pd
import matplotlib.pyplot as plt
data = pd.read_table('Collo.csv',sep=";")
```

Se importa la librería pandas y matplotlib.pylot, además se define como data, el archivo llamado “Collo.csv”, que es precisamente el que contiene los datos obtenidos de las mediciones, distancia y factor de zonas de los sitios de interés.

```
print(data)
```

Con ese comando, se puede imprimir la información del archivo. Mostrándose de la siguiente manera:

	ID	FD CLARO	FZ	FC CLARO	prob
0	T0102	0	12	0	No Potencial
1	T0103	25	8	25	Medium Potencial
2	T0104	10	16	0	No Potencial
3	T0106	15	8	0	No Potencial
4	T0107	25	8	12	Medium Potencial
..	...	...	..	...	...
372	TNMB16033	25	8	18	Medium Potencial
373	TNMB15019	20	12	12	Medium Potencial
374	TNMB03017	25	8	6	Low Potencial
375	TNMB04027	25	8	12	Medium Potencial
376	TNMB02049	15	12	12	Low Potencial

Se puede la información contenida en el archivo “Collo.csv”.

Posteriormente, podemos hacer un análisis descriptivo estadístico, de la siguiente manera:

```
print(data.describe())
```

Obteniendo lo siguiente:

	FD CLARO	FZ	FC CLARO
count	377.000000	377.00000	377.000000
mean	20.291777	9.31565	12.604775
std	7.503174	3.20382	11.093427
min	0.000000	4.00000	0.000000
25%	15.000000	8.00000	0.000000
50%	25.000000	8.00000	12.000000
75%	25.000000	12.00000	25.000000
max	25.000000	20.00000	25.000000

Se puede apreciar la media, desviación estándar, valores mínimos y máximos, etc., que son variables estadísticas que son importantes considerar.

Si únicamente se quisieran los primeros datos. Por ejemplo, las primeras 8 filas, entonces podemos lograrlo, mediante el siguiente comando:

```
print(data.head(8))
```

Obteniendo lo siguiente:

```
      ID  FD CLARO  FZ  FC CLARO             prob
0  T0102         0  12         0      No Potencial
1  T0103        25   8        25  Medium Potencial
2  T0104        10  16         0      No Potencial
3  T0106        15   8         0      No Potencial
4  T0107        25   8        12  Medium Potencial
5  T0108         0  12         0      No Potencial
6  T0109        10  12         0      No Potencial
7  T0111        10  12         6      No Potencial
```

Si quisiéramos verificar que el archivo esté correcto, respecto al número de filas y columnas, podemos aplicar la siguiente instrucción:

```
print(data.shape)
```

Obteniendo lo siguiente:

```
(377, 5)
```

De esta manera, se constata que está completa la información del archivo, ya que tiene 377 filas y 5 columnas.

Para lograr cuáles son los valores de etiquetas que tiene una determinada columna, se puede lograr mediante la siguiente instrucción.

```
print(data['prob'].unique())
```

Por tanto, se logra conocer cuáles son las etiquetas que pueden estar contenida en la columna “prob”.

```
['No Potencial' 'Medium Potencial' 'Low Potencial' 'High Potencial']
```

En el caso que se trabajó, se muestra los 4 posibles resultados, que son:

- No Potencial
- Medium Potencial
- Low Potencial
- High Potencial

Estos criterios, estarán en función de las variables que son conformadas por los factores de cobertura, distancia y zona.

Sin embargo, no se muestra la distribución en función del número de sitios del portafolio.

Para conocer, dicha distribución se puede utilizar la siguiente instrucción.

```
print(data.groupby('prob').size())
```

Obteniendo lo siguiente:

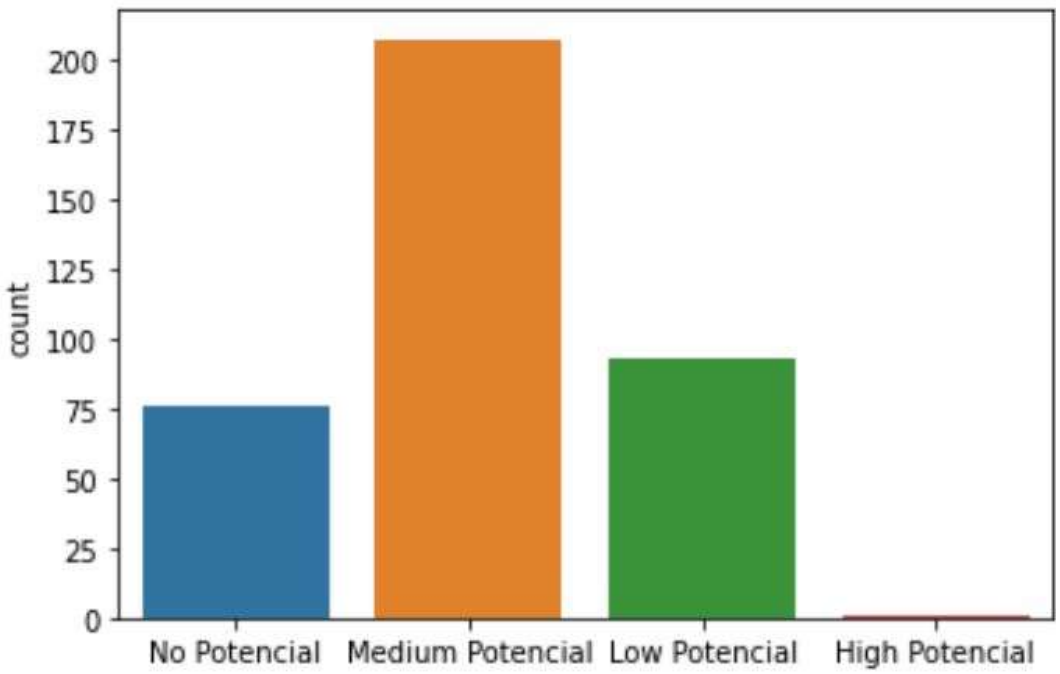
```
prob
High Potencial        1
Low Potencial        93
Medium Potencial    207
No Potencial         76
```

Se puede ver, que hay únicamente un sitio, con un valor de “High Potencial”, 207 sitios con un “Medium Potencial”, 93 sitios con un “Low Potencial” y 76 sitios con un “No Potencial”.

Mediante las siguientes instrucciones, podemos graficar el resultado anterior.

```
import seaborn as sns
sns.countplot(data['prob'],label="Count")
plt.show()
```

A continuación, se muestra la siguiente gráfica:

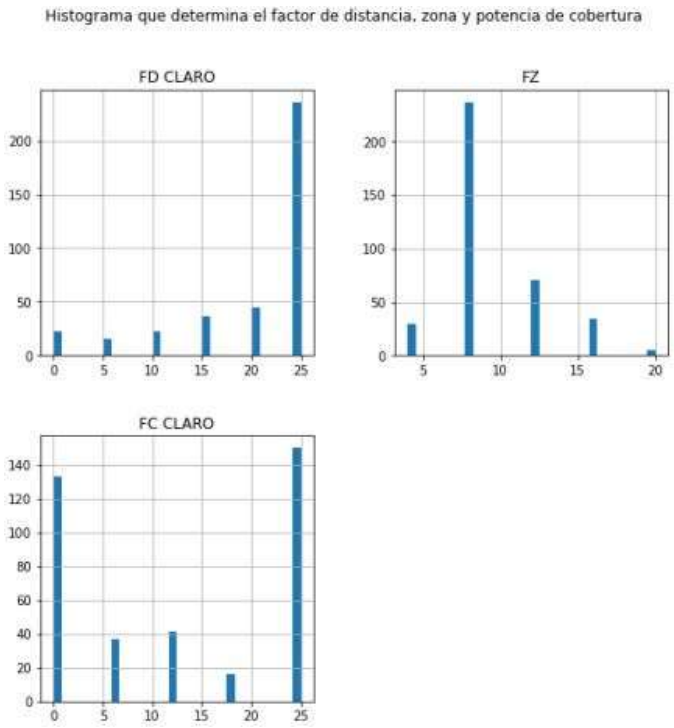


Se puede apreciar en un diagrama de barras la distribución de los 4 valores para los sitios de interés.

También, se puede generar un histograma con dicha información, con la utilización de los siguientes comandos.

```
import pylab as pl
data.drop('prob' ,axis=1).hist(bins=30, figsize=(9,9))
pl.suptitle("Histograma que determina el factor de distancia, zona y potencia de cobertura")
plt.savefig('prob_hist')
plt.show()
```

Obteniendo el siguiente comportamiento:

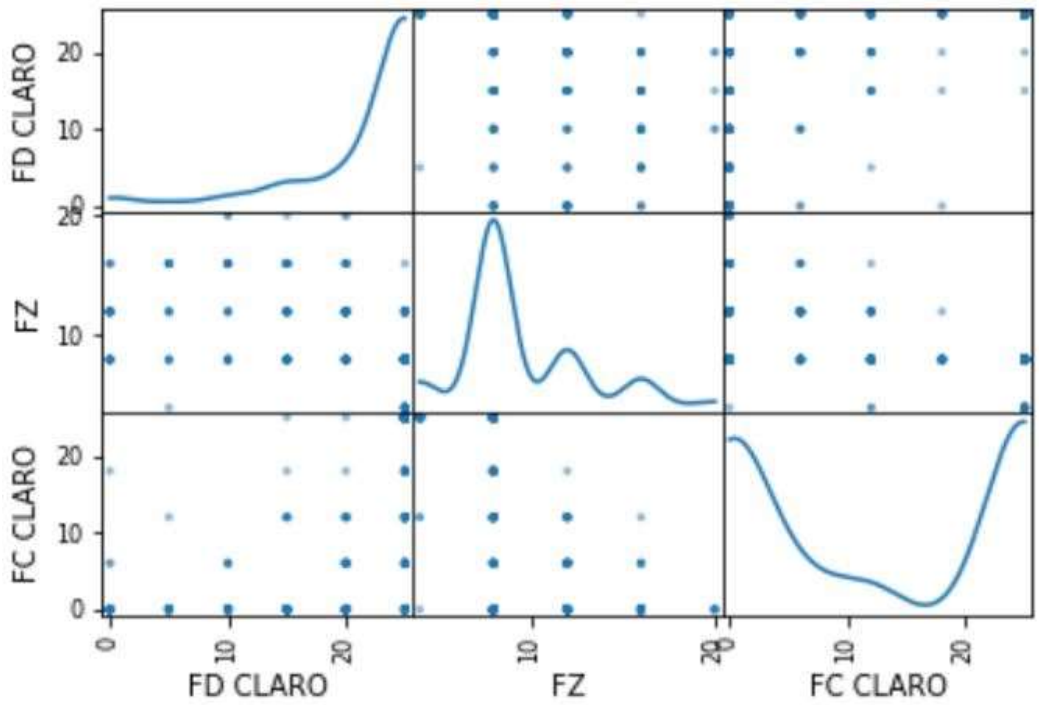


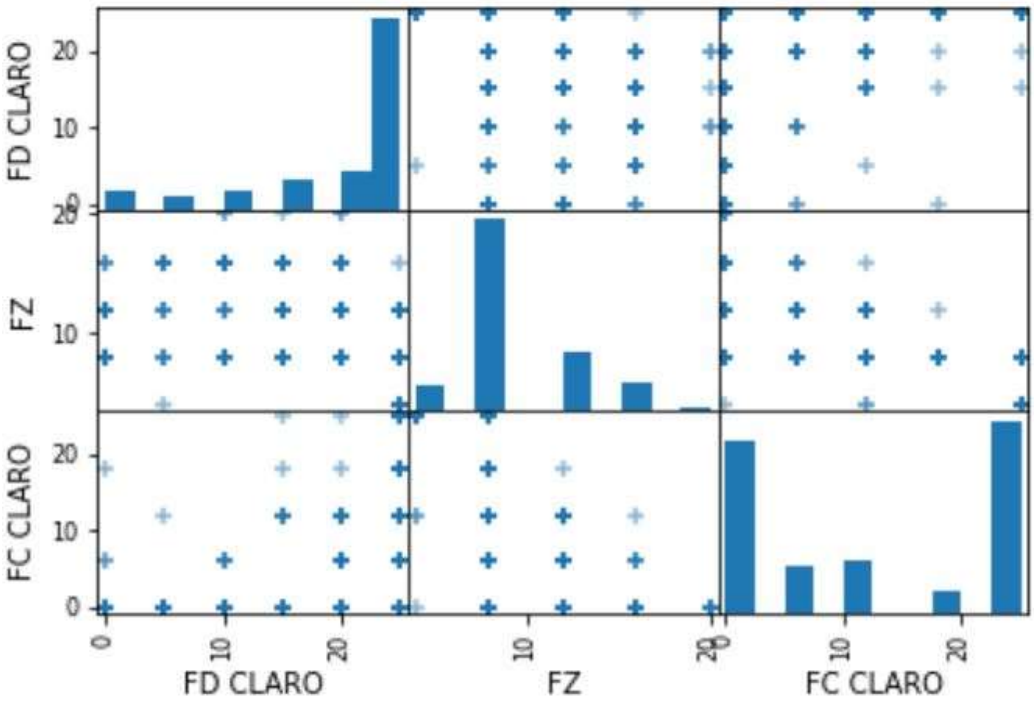
De esta manera se puede analizar de una mejor manera los datos que están contenidos en el archivo que contiene los sitios de interés, a partir de las 3 variables que se consideran para determinar el nivel de viabilidad de co ubicación que puede tener un sitio.

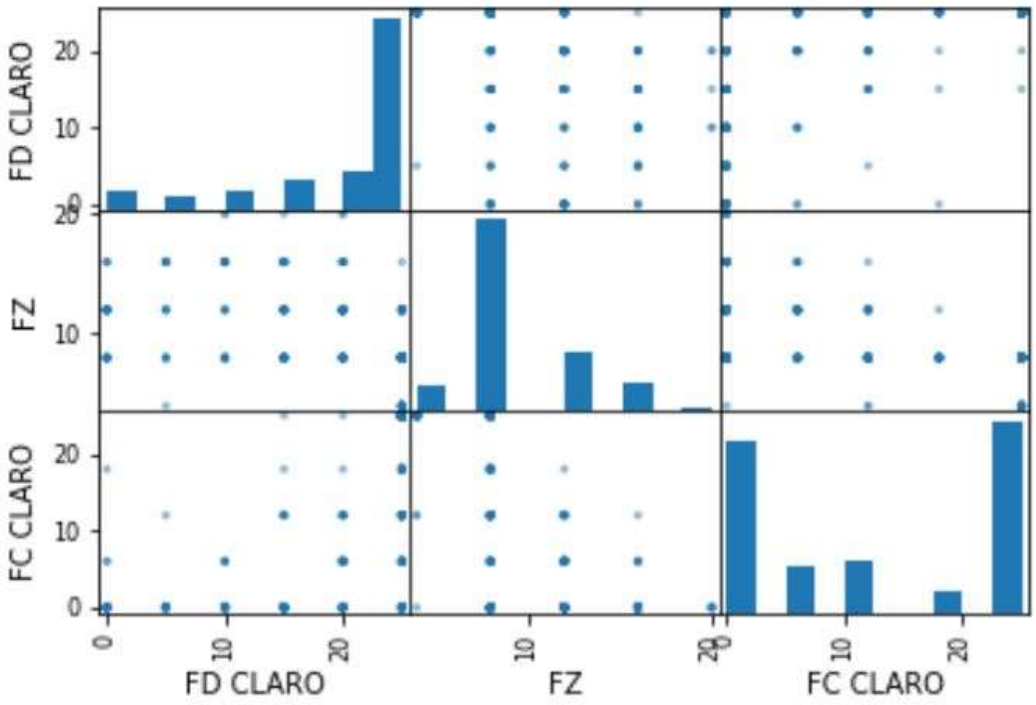
Además, se pueden ver los datos en modo de dispersión con las siguientes instrucciones:

```
pd.plotting.scatter_matrix(data, diagonal='kde')
pd.plotting.scatter_matrix(data, marker='+')
pd.plotting.scatter_matrix(data)
```

Se logró obtener lo siguiente:







Para determinar la confianza para cada una de los algoritmos de machine learning basado en aprendizaje supervisado, se hace de la siguiente manera:

```
from sklearn.model_selection import train_test_split
feature_names = ['FD CLARO', 'FZ', 'FC CLARO']
X = data[feature_names]
y = data['prob']


X_train, X_test, y_train, y_test = train_test_split(X, y, random_state=0)
from sklearn.preprocessing import MinMaxScaler
scaler = MinMaxScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)

from sklearn.neighbors import KNeighborsClassifier
knn = KNeighborsClassifier()
knn.fit(X_train, y_train)
print('Accuracy of K-NN classifier on training set: {:.2f}'
     .format(knn.score(X_train, y_train)))
print('Accuracy of K-NN classifier on test set: {:.2f}'
     .format(knn.score(X_test, y_test)))


from sklearn.linear_model import LogisticRegression
logreg = LogisticRegression()
logreg.fit(X_train, y_train)
print('Accuracy of Logistic regression classifier on training set: {:.2f}'
     .format(logreg.score(X_train, y_train)))
print('Accuracy of Logistic regression classifier on test set: {:.2f}'
     .format(logreg.score(X_test, y_test)))

from sklearn.tree import DecisionTreeClassifier
clf = DecisionTreeClassifier().fit(X_train, y_train)
print('Accuracy of Decision Tree classifier on training set: {:.2f}'
     .format(clf.score(X_train, y_train)))
print('Accuracy of Decision Tree classifier on test set: {:.2f}'
     .format(clf.score(X_test, y_test)))

from sklearn.discriminant_analysis import LinearDiscriminantAnalysis
lda = LinearDiscriminantAnalysis()
lda.fit(X_train, y_train)
print('Accuracy of LDA classifier on training set: {:.2f}'
     .format(lda.score(X_train, y_train)))
print('Accuracy of LDA classifier on test set: {:.2f}'
     .format(lda.score(X_test, y_test)))
```

Obteniendo el siguiente resultado:

```
Accuracy of K-NN classifier on training set: 0.91
Accuracy of K-NN classifier on test set: 0.93
Accuracy of Logistic regression classifier on training set: 0.88
Accuracy of Logistic regression classifier on test set: 0.89
Accuracy of Decision Tree classifier on training set: 0.94
Accuracy of Decision Tree classifier on test set: 0.91
Accuracy of LDA classifier on training set: 0.90
Accuracy of LDA classifier on test set: 0.92
Accuracy of GNB classifier on training set: 0.85
Accuracy of GNB classifier on test set: 0.85
Accuracy of SVM classifier on training set: 0.90
Accuracy of SVM classifier on test set: 0.91
```

Se logra observar, que la precisión del algoritmo K-NN es mayor que las demás.

Por ello, será el algoritmo a aplicar en el presente estudio.

```
from sklearn.metrics import classification_report
from sklearn.metrics import confusion_matrix
pred = knn.predict(X_test)
print(confusion_matrix(y_test, pred))
print(classification_report(y_test, pred))
```

Al aplicar las instrucciones anteriores se obtiene lo siguiente matriz para la predicción:

```
[[21  2  1]
 [ 0 47  0]
 [ 4  0 20]]
```

Y finalmente:

	precision	recall	f1-score	support
Low Potencial	0.84	0.88	0.86	24
Medium Potencial	0.96	1.00	0.98	47
No Potencial	0.95	0.83	0.89	24
accuracy			0.93	95
macro avg	0.92	0.90	0.91	95
weighted avg	0.93	0.93	0.93	95

```
k_range = range(1,10)
scores = []
for k in k_range:
    knn = KNeighborsClassifier(n_neighbors = k)
    knn.fit(X_train, y_train)
    scores.append(knn.score(X_test, y_test))
plt.figure()
plt.xlabel('k')
plt.ylabel('accuracy')
plt.scatter(k_range, scores)
plt.xticks([0,2,4,6,8,10])
```

```
n_neighbors = 2

knn = KNeighborsClassifier(n_neighbors)
knn.fit(X_train, y_train)
print('Accuracy of K-NN classifier on training set: {:.2f}'
     .format(knn.score(X_train, y_train)))
print('Accuracy of K-NN classifier on test set: {:.2f}'
     .format(knn.score(X_test, y_test)))

print(clf.predict([[25,8,18]]))
```

9. CONCLUSIONES

En este mundo moderno, donde hay una constante demanda de procesamiento de datos, para convertirla en información, los algoritmos de Machine Learning juegan un papel significativo, ya que permite a las computadoras aprender por sí mismas, sin necesidad de ser programadas por una persona previamente. Lo que tiene mucha relevancia muchos campos, en el caso de la presente investigación, fue en el de las telecomunicaciones.

Con el aprendizaje automático supervisado, se puede implementar para que aprendan de manera constante y no solo para realizar una tarea en particular. Machine Learning tiene mucho alcance en un mundo digital. Lo que permite crear modelos analíticos que permiten predecir o estimar valores en función de las características de las variables que se consideren.

En valoraron varios métodos de aprendizaje supervisado para su debida aplicación. Sin embargo, el algoritmo K-NN mostro mejor consistencia que los demás.

Se realizó un código en el lenguaje de programación Python, donde se muestran la utilización de los distintos tipos de aprendizaje supervisado. Logrando demostrar la consistencia que tiene el algoritmo K-NN para una base de datos que en formato .csv, donde se reúne información importante para determinar la viabilidad mediante las variables: factor de cobertura, factor de zona y factor de distancia, para determinar cuatros distintos niveles que podría tener un sitio donde se encuentre una estación base de telefonía móvil.

10. Recomendaciones

Se recomienda que se sigan desarrollando temas aplicados en el área de Machine Learning, con el objetivo de resolver problemas en entornos sociales y empresariales. Los algoritmos supervisados y no supervisados se aplican para hacer predicciones a partir de datos obtenidos, dichas predicciones son de mucha utilidad para la toma de decisiones, que podría ayudar a optimizar recursos, y de esa manera la reducción de costos, y por ende una mayor utilidad.

11. BIBLIOGRAFÍA

[1] Montenegro Christian, Morales Luis. "'Método para la detección de cáncer de mama en mamografías usando Convolutional Neural Networks (CNN)". Universidad Nacional de Ingeniería. Managua, Nicaragua. Agosto del 2021.

[2] Ordoñez Marco. "Optimización de Redes Soportada en Machine Learning". Universidad Distrital Francisco José de Caldas. Bogotá, Colombia. Julio del 2021.

[3] González Julio. "Aplicación de Machine Learning en las Empresas del Sector de Telecomunicaciones del Perú". Universidad César Vallejo. Perú. 2020.

[4] López Patrick. "Cómo Telefónica usa la Inteligencia Artificial y el Machine Learning para Conectar a los no Conectados". 2018. Disponible en: https://blogthinkbig.com/internet-para-todos

[5] Pérez Jarliev. *"Optimización del tiempo de procesamiento y eficiencia del algoritmo para solucionar el problema inverso generalizado de Voronoi utilizando lenguaje de programación visual C++ y técnicas del tipo divide y vencerás aplicado a imágenes reales".* Protocolo de Investigación. Universidad Nacional de Ingeniería. Managua, Nicaragua. 2020.

[6] Salgüero Irene. "*Importancia del infiltrado inflamatorio y la neovascularización asociada al melanoma y su correlación con el desarrollo me metástisis: Estudio inmunohistoquímico de 81 casos. Árboles de decisión basados en Machine Learning".* Madrid, España. 2020.

[7] Mahesh Batta.*"Machine Learning Algorithms – a Review".* International Journal of Science and Research (IJSR). ISSN: 2319-7064. ResearchGate Impact Factor (2018): 0.28.

[8] Bagherian Maryam, Sabeti Elyas, Wang Kai, Sartor Maureen, Nikojovska-Coleska Zaneta, Najarian Kayvan. *"Machine Learning approaches and databases for prediction of drug-target interaction: a survey paper"*. Enero del 2021.

[9] K-Nearest Neighbor. Disponible en:

https://www.aprendemachinelearning.com/clasificar-con-k-nearest-neighbor-ejemplo-en-python/

[10] Aprendizaje por refuerzo. Disponible en:

https://www.aprendemachinelearning.com/

[11] Reducción de dimensiones. Disponible en:

https://www.aprendemachinelearning.com/comprende-principal-component-analysis/

[12] Definiendo Machine Leaning. Disponible en:

https://www.aprendemachinelearning.com/que-es-machine-learning/

[13] Aplicaciones del Machine Learning. Disponible en.

https://www.aprendemachinelearning.com/aplicaciones-del-machine-learning/#no_supervisado

[14] ¿Qué es el lenguaje supervisado? Disponible en:

https://www.tibco.com/es/reference-center/what-is-supervised-learning

[15] Lenguaje No Supervisado. Disponible en:

https://www.tibco.com/es/reference-center/what-is-unsupervised-learning

[16] ¿Qué es Machine Learning?. Disponible en:

https://www.tibco.com/es/reference-center/what-is-machine-learning