**Universidad ORT Uruguay**

Facultad de Ingeniería

**Machine learning en producción**

**Obligatorio**

**Autores:**

Gonzalo de León - 158545

Martin Ljubicic - 154630

Joaquín Oldan - 159579

**Docentes:**

Federico Zaiter

Matías Sorozabal

**2023**

Tabla de contenido

[I. Introducción 4](#_Toc137312049)

[A. Contexto del estudio 4](#_Toc137312050)

[B. Objetivos 4](#_Toc137312051)

[C. Metodología de investigación 5](#_Toc137312052)

[II. Fundamentos de Machine Learning 6](#_Toc137312053)

[A. Definición de Machine Learning 6](#_Toc137312054)

[B. Tipos de algoritmos de Machine Learning 6](#_Toc137312055)

[C. Importancia del Machine Learning en la producción 7](#_Toc137312056)

[III. Arquitectura de la solución 8](#_Toc137312057)

[A. Recopilación de datos 8](#_Toc137312058)

[B. Tratamiento de datos 8](#_Toc137312059)

[C. Modelo 9](#_Toc137312060)

[D. Producción 9](#_Toc137312061)

[IV. Preparación de los datos 10](#_Toc137312062)

[A. Scrapy 10](#_Toc137312063)

[B. Limpieza y preprocesamiento de datos 10](#_Toc137312064)

[C. Selección de características 10](#_Toc137312065)

[V. Modelado y entrenamiento 12](#_Toc137312066)

[A. Selección de algoritmos de Machine Learning 12](#_Toc137312067)

[B. Adaptar datos al modelo 13](#_Toc137312068)

[C. División de datos 14](#_Toc137312069)

[D. Entrenamiento del modelo 15](#_Toc137312070)

[Arquitectura del modelo: 15](#_Toc137312071)

[Carga de datos: 15](#_Toc137312072)

[Inicialización de pesos y sesgos: 15](#_Toc137312073)

[Definición de la función de pérdida y algoritmo de optimización: 15](#_Toc137312074)

[Ciclo de entrenamiento: 16](#_Toc137312075)

[Evaluación del rendimiento: 16](#_Toc137312076)

[VI. Implementación en producción 17](#_Toc137312077)

[A. Consideraciones de infraestructura 17](#_Toc137312078)

[B. Integración del modelo en un sistema de producción 17](#_Toc137312079)

[C. Monitoreo y mantenimiento del modelo en producción 17](#_Toc137312080)

[VII. Desafíos y recomendaciones 18](#_Toc137312081)

[A. Desafíos comunes en la implementación de Machine Learning en producción 18](#_Toc137312082)

[B. Recomendaciones para superar los desafíos 18](#_Toc137312083)

[VIII. Conclusiones 19](#_Toc137312084)

[A. Resumen de los hallazgos 19](#_Toc137312085)

[B. Contribuciones del estudio 19](#_Toc137312086)

[C. Áreas para futuras investigaciones 19](#_Toc137312087)

# I. Introducción

## A. Contexto del estudio

El Machine Learning surge como una necesidad de estudiar los datos obtenidos de cualquier ámbito para obtener información relevante para problemas planteados o para mejorar sistemas. La toma de decisiones automatizada, la automatización de procesos aporta mucho valor agregado en diferentes ámbitos.

Como se pudo observar en las presentaciones realizadas por el equipo de diversos papers, existen cada vez mas casos de uso exitosos en la materia, en la presentación de los integrantes del grupo sobre Michelangelo, se expuso que no solo es importante los resultados directos del aprendizaje, sino la metodología aplicada para llegar a ellos. Esta herramienta proporciona buenos resultados y ayudo a organizar los grupos de trabajo.

La automatización es vista como como una oportunidad de mejora para las organizaciones y permitirá redirigir sus esfuerzo a trabajos de mejora constante de estas metodologías, logrando mejorar sus procesos.

En este trabajo se buscará incursionar en estas metodologías para la implementación de un producto que pueda ser llevado a producción haciendo hincapié más en el E2E que en sus diferentes partes.

## B. Objetivos

El objetivo general del proyecto consiste en lograr un E2E de un sistema que obtenga datos de una pagina web de venta de propiedades y que sea capaz de aprender a predecir un parámetro de estas relevante, en este caso se apunta a poder distinguir la categoría del inmueble según su valor.

En principio esta variable esta directamente relacionada con el precio, aunque se podría llegar a paralelizar una categoría de valor en función de las necesidades de quien busca adquirir la vivienda. Esto ya es un desafío mas ambicioso y no será el foco del presente trabajo.

Se logran identificar objetivos mas específicos, apuntando a etapas de proceso general.

Obtener un modelo que logre analizar imágenes como datos tabulares de los elementos de estudio, si bien no hay un enfoque sobre obtener los mejores resultados, si se buscara un mínimo desempeño en esta etapa del modelo, que sirva para obtener resultados bastante aproximados del problema.

Análisis de datos, comprensión de cuales son mas relevantes y cuales no, cuales aportan mas al target que se busca aprender para luego inferir con buena precisión en datos de testeo.

Desempeño del sistema, en velocidad y escalabilidad, un buen sistema debe ser capaz de proporcionar una buena respuesta ante múltiples consultas.

Evaluar el impacto de la solución completa en cuanto a su utilidad real y la aplicabilidad a un caso real de estudio, si bien se puede considerar un objetivo secundario, seria bueno tener una referencia a su utilidad, por su valor agregado a algún proceso determinado para un caso de estudio puntual o general.

Comprender que mejoras se pueden proporcionar a futuro en la aplicación para apuntar a un mercado de clientes reales y su posibilidad de venta.

## C. Metodología de investigación

El enfoque del trabajo será con fines académicos, no se analiza un caso de estudio real, aunque luego pueda extrapolarse el trabajo, cambiando la información tratada para atender a un cliente. Se destaca igualmente que si bien no esta enfocado a un cliente, puede ser útil lo realizado en el caso de uso para cierto marco de público objetivo.

El estudio apunta a analizar información de propiedades de la página [www.gallito.com.uy](http://www.gallito.com.uy), para aprender a clasificarlas en clases de interés dependiendo su valor. Se recopilará información, se analizará y se mostraran resultados de forma amigable al usuario.

Los datos obtenidos se analizan para utilizarlos de la mejor manera en su inyección en el modelo, comprender que tan significativo resultara, y realmente si aportara valor. Datos faltantes también serán analizados para comprender la necesidad de completarlos o no, y así no influir en el aprendizaje del modelo y también luego en la inferencia en producción.

La información recopilada en el aplicativo se considera de interés general y acceso público, y no se expondrán datos personales de las cuentas que realizan las publicaciones o datos que puedan ser relacionados a ellas, será necesario revisar el dataset de imágenes y cuidar esta información para completar un trabajo dentro de los marcos éticos adecuados.

El estudio buscara respetar reglas de análisis de datos establecidos, cuidar el data leakage y el training-serving skew que puede repercutir de gran manera en los resultados y desempeños del aplicativo.

Si bien la información de la fuente es obtenida en un momento determinado del mercado de inmuebles, en definitiva, muy relacionada al momento actual del mercado, no se tienen datos históricos, se deberá cuidar las conclusiones a partir de estos datos sesgados en cierta manera.

# II. Fundamentos de Machine Learning

## A. Definición de Machine Learning

El Machine Learning, o aprendizaje automático, es una rama de la inteligencia artificial que se centra en el desarrollo de algoritmos y técnicas que permiten a las computadoras aprender de manera automática a través de la experiencia y los datos, sin ser programadas explícitamente. En lugar de seguir instrucciones específicas, los algoritmos de Machine Learning son capaces de reconocer patrones, identificar correlaciones y tomar decisiones o realizar predicciones basadas en los datos con los que han sido entrenados.

Este enfoque de aprendizaje automatizado ha demostrado ser eficaz en una amplia gama de aplicaciones, desde el reconocimiento de voz y la visión por computadora hasta la detección de fraudes y la personalización de recomendaciones. El objetivo del Machine Learning es permitir que las máquinas adquieran conocimiento y mejoren su rendimiento a medida que se enfrentan a nuevos datos, lo que lo convierte en una herramienta poderosa para abordar problemas complejos y tomar decisiones basadas en información.

## B. Tipos de algoritmos de Machine Learning

Existen diversos tipos de algoritmos de Machine Learning que se utilizan para abordar diferentes tipos de problemas y tareas. A continuación, algunos de los tipos más comunes de algoritmos de Machine Learning:

* Aprendizaje supervisado (Supervised Learning): En este tipo de algoritmo, se entrena al modelo utilizando un conjunto de datos de entrenamiento que incluye ejemplos etiquetados. El objetivo es que el modelo aprenda a predecir o clasificar nuevas instancias en función de las etiquetas conocidas. Los árboles de decisión están en esta categoría, muy explicativos y prácticos.
* Aprendizaje no supervisado (Unsupervised Learning): En los algoritmos de aprendizaje no supervisado, no se proporcionan etiquetas en los datos de entrenamiento. El objetivo es descubrir patrones o estructuras ocultas en los datos. Un ejemplo de estos son el clustering.
* Aprendizaje por refuerzo (Reinforcement Learning): Este tipo de algoritmo se basa en un proceso de toma de decisiones en el que un agente interactúa con un entorno y recibe recompensas o castigos en función de sus acciones. El objetivo es que el agente aprenda a tomar decisiones que maximicen las recompensas a largo plazo. Los ejemplos más característicos se atribuyen a juegos.
* Aprendizaje profundo (Deep Learning): El aprendizaje profundo utiliza redes neuronales artificiales con múltiples capas para aprender características y representaciones complejas de los datos. Estas redes pueden realizar tareas de reconocimiento de imágenes, procesamiento de lenguaje natural y traducción. Las redes CNN para tratamiento de imágenes son un ejemplo.

Este trabajo se enfocará en Deep Learning, realizando modelos de CNN + MLP para el tratamiento de imágenes y datos tabulares provistos de los datos adquiridos.

## C. Importancia del Machine Learning en la producción

Llevar un algoritmo de Machine Learning a producción es de suma importancia debido a que es el paso final para hacer uso efectivo de los modelos desarrollados y obtener beneficios reales en entornos prácticos. Como puntos importantes hay que destacar:

* Utilización efectiva de los modelos: Se puede aprovechar plenamente el valor y el potencial de los modelos desarrollados. Los modelos pueden aplicarse en tiempo real y en escala, lo que permite automatizar tareas, tomar decisiones y mejorar la eficiencia operativa.
* Toma de decisiones automatizada: Se puede tomar decisiones automatizadas basadas en datos y análisis. Esto puede conducir a una toma de decisiones más rápida, precisa y consistente, evitando errores humanos y mejorando la eficacia general de los procesos.
* Optimización de recursos: Se pueden aprovechar eficientemente los recursos disponibles. Los modelos pueden ayudar a optimizar procesos, mejorar la asignación de recursos, reducir el desperdicio y aumentar la productividad.
* Mejora continua: Permite recopilar datos en tiempo real y retroalimentar los resultados a los modelos de Machine Learning. Esto facilita la mejora continua y la capacidad de adaptación a medida que los modelos se enfrentan a nuevos escenarios y cambios en los datos de entrada.

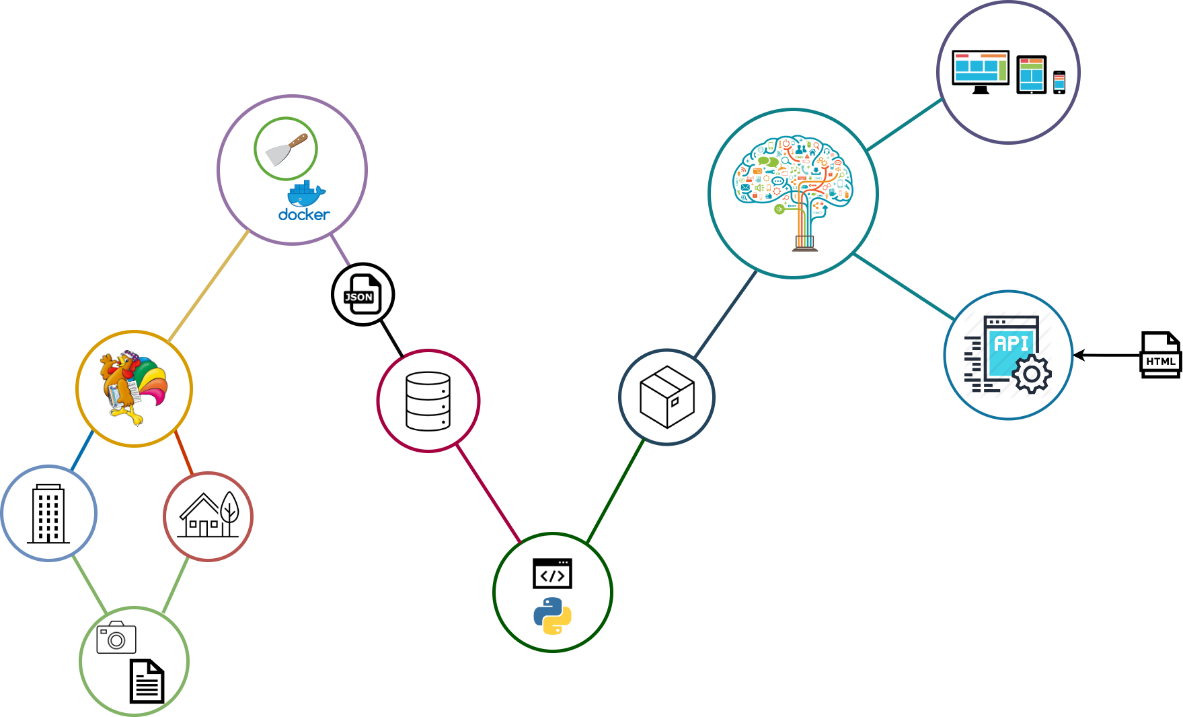
Hay aspectos que hay que cuidar, para lograr un buen sistema!

* Estabilidad y rendimiento: Es esencial garantizar que el algoritmo funcione de manera estable y tenga un rendimiento óptimo en el entorno de producción. Esto implica considerar aspectos como el tiempo de respuesta, la escalabilidad, la eficiencia de los recursos y la capacidad de manejar grandes volúmenes de datos en tiempo real.
* Integración con la infraestructura existente: La integración adecuada del algoritmo en la infraestructura existente es crucial. Esto incluye aspectos como la compatibilidad con los sistemas y tecnologías utilizados, la gestión de datos de entrada y salida, y la seguridad y privacidad estos.
* Monitoreo y mantenimiento: Un algoritmo de Machine Learning en producción debe ser monitoreado de forma continua para asegurar que siga funcionando correctamente y proporcionando resultados precisos. También se debe realizar un mantenimiento regular para actualizar el modelo, reentrenarlo con nuevos datos y abordar posibles desviaciones.
* Evaluación y medición de resultados: Es importante establecer métricas de evaluación para medir el desempeño y el impacto del algoritmo en producción. Esto permitirá evaluar su eficacia, identificar posibles mejoras y demostrar el valor generado a los interesados.

En resumen, la implementación exitosa de un algoritmo de Machine Learning en producción es crucial para aprovechar plenamente los beneficios de los modelos desarrollados y lograr una toma de decisiones automatizada y eficiente.

# III. Arquitectura de la solución

Se implementa una solución de varias etapas como se puede observan en el siguiente diagrama:



Se dará una breve descripción de cada parte y luego en siguientes capítulos se entrará en detalle.

## A. Recopilación de datos

Se implementará una etapa de adquisición de datos utilizando una herramienta de scraper que bajara datos de inmuebles de la pagina de elgallito.com.uy. Esta etapa deberá se implementa para obtener todos los datos que se puedan obtener por ítem, y luego en siguientes secciones nos centraremos en analizar esa data y poder entender cuál será útil y cual no.

## B. Tratamiento de datos

La data obtenida deberá ser analizada, adaptada en formato, y aceptada o descartada en función de su utilidad. Sera necesario ya tener pensado que tipo de datos se necesitaran en el modelo para poder ser input de las etapas de este.

## C. Modelo

El modelo será implementado en un lenguaje de programación universal y en un formato que puede llevarse a varias plataformas para su funcionamiento. Esta etapa se encargará de obtener la clasificación problema planteada en el desafío a partir de su entrenamiento, evaluación, y posterior puesta en producción.

Se utiliza Weights and Biases, que es una plataforma utilizada en el campo del aprendizaje automático y la ciencia de datos para el seguimiento, la visualización y la colaboración en experimentos y modelos. Proporciona herramientas y servicios que permiten a los investigadores y científicos de datos realizar un seguimiento detallado de los experimentos, registrar métricas, visualizar gráficos y compartir resultados con otros miembros del equipo.

Con Weights and Biases, se puede realizar un seguimiento de los hiperparámetros utilizados en los modelos, registrar métricas de rendimiento durante el entrenamiento, visualizar gráficos y tablas para analizar los resultados .

## D. Producción

En esta etapa se realizará la preparación del modelo entrenado, con las herramientas necesarias para poder ser presentado y desplegado en la web. Esta aplicación será capaz de realizar inferencia, a partir de los datos ingresados por el usuario, y utilizará el modelo entrenado para realizar una inferencia y posteriormente será mostrado.

# IV. Preparación de los datos

## A. Scrapy

Para recopilar datos se utiliza Scrapy. COMPLETAR

## B. Limpieza y preprocesamiento de datos

Con los datos obtenidos, se procede a procesarlos para llevarlos a un formato estándar y pensado para ponerlo en un modelo CNN+MLP numérico.

Los formatos serán numéricos para todos los datos que tengan significancia como tal, cantidades de elementos, dimensiones de elementos, y cualquier otra de este estilo. Las variables conceptuales serán trabajadas mas adelante para poderlas llevar a un formato valido para el modelo elegido.

Sera necesario establecer un criterio si existen datos faltantes, al menos mientras tenga significancia ese registro, si ya los faltantes son varios, es mejor descartar ese ítem.

Los criterios tenidos en cuenta en este proyecto son:

COMPLETAR

## C. Selección de características

De los inmuebles se seleccionan determinadas características que luego serán los inputs del modelo.

* Imagen de portada (1024x768)
* Localidad donde se ubica el inmueble (varias, más de 200)
* Cantidad de Habitaciones (0, 1, 2, 3, 4[o más])
* Cantidad de Baños (1,2,3[o más]
* Metros cuadrados (25 a 500)
* Si es casa o apartamento (uno u otro)

La imagen de referencia será la de portada del inmueble, se cree necesario tener un input de este estilo ya que puede aportar información sobre estado, luminosidad, y otros. En una primera aproximación se tomará solamente una imagen, pero podrá establecerse que sea una cantidad mayor. De todas maneras, esa cantidad debería ser fija, para poder luego de limpiar datos, adaptarlos para la etapa de modelado.

La localidad, es un dato importante ya que hay mucha variación en costos debido a este parámetro, hay que trabajar este dato para que logre capturar todas las opciones posibles y luego analizar la mejor forma de inyectarlo en el modelo.

Cantidad de habitaciones, cantidad de baños, Metros cuadrados, son valores numéricos que de cierta forma dan un escalado en los precios de los inmuebles, con una relación que podría considerarse lineal.

Si es casa o apartamento, es un dato no menor, que dará de cierta forma un diferencial entre estas 2 categorías.

# V. Modelado y entrenamiento

## A. Selección de algoritmos de Machine Learning

Se utilizará un modelo de Deep Learning basado en una CNN para analizar las imágenes, y luego será input de una MLP con el adicional de los datos tabulares comentados en la selección de características.

Red Neuronal Convolucional (CNN) es un tipo de red neuronal especializada en el procesamiento de datos estructurados en forma de matrices, como imágenes. Se compone de capas convolucionales que aplican filtros para extraer características relevantes de las imágenes y capas de pooling que reducen la dimensionalidad. Las capas convolucionales y de pooling se combinan con capas completamente conectadas para realizar la clasificación o predicción final. Las CNN son altamente eficientes para el procesamiento de imágenes debido a su capacidad para reconocer patrones espaciales y aprender características jerárquicas.

Un Perceptrón Multicapa (MLP) es una red neuronal artificial compuesta por múltiples capas de neuronas, incluyendo una capa de entrada, una o más capas ocultas y una capa de salida. Cada neurona en una capa está conectada a todas las neuronas de la capa anterior y de la siguiente, formando una red profunda. Las neuronas en cada capa realizan cálculos basados en funciones de activación no lineales para transformar los datos de entrada. Los MLP son utilizados en problemas de clasificación y regresión, y pueden aprender relaciones no lineales entre las variables de entrada y la variable objetivo.

El problema para resolver de clasificación en clases de los inmuebles lleva a la utilización de una métrica de error para este tipo de modelo, la Cross Entropy Loss, esta es una función de costo comúnmente utilizada en problemas de clasificación en el campo del Machine Learning. Es especialmente útil cuando se trabaja con modelos de clasificación que producen probabilidades como salida, como las redes neuronales.

Esta función mide la discrepancia entre la distribución de probabilidades predicha por el modelo y la distribución de probabilidades verdaderas (etiquetas) de los datos. Cuanto más diferentes sean estas dos distribuciones, mayor será el valor de la función de pérdida.

En términos más técnicos, para un problema de clasificación binaria, se calcula como la suma de los productos de las etiquetas verdaderas y el logaritmo de las probabilidades predichas de la clase positiva, junto con los productos de las etiquetas complementarias y el logaritmo de las probabilidades predichas de la clase negativa. En problemas de clasificación multiclase, la función se generaliza para incluir todas las clases.

El objetivo del entrenamiento de un modelo es minimizar este parámetro, ajustando los pesos de las neuronas en la red neuronal para mejorar la concordancia entre las predicciones y las etiquetas verdaderas.

Esta no es la única solución para este tipo de problemas, se puedo haber implementado funciones más visuales de clasificación como los árboles de decisión, estos son un tipo de algoritmo de aprendizaje automático utilizado en problemas de clasificación y regresión. Se basan en la estructura de un árbol en el que cada nodo representa una característica o atributo, las ramas representan las posibles decisiones o reglas basadas en los valores de las características, y las hojas representan las etiquetas de clasificación o los valores de regresión.

En el aprendizaje de árboles de decisión, el algoritmo busca encontrar las divisiones o particiones óptimas en los datos de entrenamiento, con el objetivo de maximizar la pureza o la homogeneidad de las clases dentro de cada subconjunto resultante. Estas divisiones se basan en reglas de decisión que se definen mediante pruebas en los valores de las características.

Durante la construcción del árbol, el algoritmo evalúa diversas métricas de calidad para determinar cuál atributo y qué umbral de separación proporcionan la mejor división en cada nodo. Algunas de las métricas comunes utilizadas son la ganancia de información, el índice Gini y la reducción de error cuadrático.

Una vez construido el árbol, se puede utilizar para hacer inferencia, siguiendo el camino desde la raíz hasta una hoja, tomando decisiones basadas en las pruebas de características en cada nodo. En problemas de clasificación, la etiqueta de la hoja final representa la clase predicha, mientras que en problemas de regresión, el valor de la hoja final es la predicción numérica.

Los árboles de decisión son atractivos debido a su interpretabilidad y facilidad de comprensión. Además, son resistentes al ruido en los datos y pueden manejar características de diferentes tipos. Sin embargo, los árboles de decisión también pueden ser propensos al sobreajuste y pueden tener dificultades para representar relaciones complejas entre variables.

Para mitigar estas limitaciones, existen variantes de árboles de decisión, random forests y gradient boosting trees, que combinan múltiples árboles para mejorar la precisión y generalización del modelo.

## B. Adaptar datos al modelo

Se trabajaran determinados datos para ser utilizados en el modelo, como son el dato de la localidad y de si es casa o apartamento. En una red MLP los datos no numéricos deben ser transformados o codificados de manera adecuada para poder ser utilizados como entradas en la red. Esto se debe a que las neuronas en una red MLP trabajan con valores numéricos y realizan operaciones matemáticas en ellos.

Existen diferentes técnicas para manejar datos no numéricos en una red MLP, algunas de las cuales son las siguientes:

* One-hot encoding: Esta técnica se utiliza cuando los datos no numéricos son variables categóricas, es decir, tienen un conjunto limitado de categorías distintas. Consiste en convertir cada categoría en un vector binario de longitud igual al número de categorías posibles y se e crea una columna binaria separada para cada categoría, y solo una de las columnas contendrá un valor de 1, mientras que las demás contendrán valores de 0.
* Codificación ordinal: Si los datos no numéricos tienen un orden inherente, como clasificaciones ordinales (por ejemplo, "bajo", "medio" y "alto"), se puede asignar un valor numérico a cada categoría basado en su orden. Por ejemplo, "bajo" se podría codificar como 0, "medio" como 1 y "alto" como 2. De esta manera, se preserva el orden y la red puede interpretar estos valores numéricos en relación con la variable original.
* Embedding: Es una técnica más avanzada que asigna vectores de valores numéricos de baja dimensión a cada categoría. Esta técnica es particularmente útil cuando los datos no numéricos tienen una gran cantidad de categorías. Los embeddings son aprendidos durante el proceso de entrenamiento de la red, lo que permite que la red capture relaciones complejas entre las categorías y los resultados.

Es importante tener en cuenta que la elección de la técnica de codificación dependerá del tipo de datos no numéricos y del problema específico que se esté abordando. Cada enfoque tiene sus ventajas y consideraciones, y es necesario evaluar cuál es el más adecuado para el caso en particular.

Se trabajara estas 2 características de la forma:

* Casa o Apartamento: Utilizando One-Hot encoding, utilizando 2 variables, para indicar de que categoría es el dato.
* Localidad: Debido a la cantidad de clases aquí, mas de 200, se opta por incluir una capa de Embedding en el modelo, que se ira entrenando a la par de la CNN y MLP.

## C. División de datos

La división de conjuntos de datos es una etapa fundamental en el desarrollo de modelos de Machine Learning. Consiste en separar el conjunto de datos en diferentes subconjuntos, con el propósito de utilizarlos de manera adecuada durante las diferentes fases del proceso, como entrenamiento, validación y prueba.

La división típica implica tres conjuntos de datos principales:

* Conjunto de entrenamiento (Training set): Este conjunto de datos se utiliza para entrenar el modelo. Contiene ejemplos etiquetados que el modelo utilizará para aprender los patrones y relaciones entre las características y las etiquetas objetivo. Se utiliza generalmente alrededor del 70-80% del conjunto de datos total.
* Conjunto de validación (Validation set): Después de entrenar el modelo, es esencial evaluar su rendimiento en datos no vistos para ajustar los hiperparámetros y realizar ajustes adicionales. El conjunto de validación se utiliza para este propósito. Permite medir la precisión y el rendimiento del modelo en datos desconocidos y realizar comparaciones entre diferentes configuraciones de modelo. Suele ser del 10-15% del conjunto de datos total.
* Conjunto de prueba (Test set): El conjunto de prueba es utilizado para evaluar el rendimiento final y la generalización del modelo después de que se han realizado todas las etapas de entrenamiento y ajuste. Es importante que el conjunto de prueba sea independiente y no se utilice en ninguna etapa previa de desarrollo del modelo, ya que proporciona una evaluación objetiva y realista de su capacidad para generalizar a datos no vistos. Suele ser del 10-15% del conjunto de datos total.

Es importante destacar que la división de conjuntos de datos debe realizarse de manera aleatoria y estratificada en problemas de clasificación, para garantizar que las clases estén representadas de manera equilibrada en cada subconjunto.

En esta implementación se utilizaran para el entrenamiento una división de 80/20 entre datos de entrenamiento y validación. Y se ira entrenando el modelo y observando las métricas calculadas sobre ambos dataset separados para evaluar cuando es un entrenamiento suficiente.

## D. Entrenamiento del modelo

### Arquitectura del modelo:

El modelo esta formado por una primer capa CNN, seguida de una capa MLP que no solo toma los output de la primer capa sino que los concatena con los datos tabulares de la información de cada ítem.

La capa CNN esta conformada por 4 capas con las siguientes características:

* Completar

La capa MLP esta conformada por 4 capas con las siguientes características:

* Completar

### Carga de datos:

Los datos procesados serán cargados para poder entrenar y validar el modelo.

El formato de datos a cargar tendrá la siguiente forma:

FIGURA

Luego se crearan data loaders que agruparan los datos de manera aleatoria en batchs de tamaño solicitado para ser luego ser inyectados en el entrenamiento y validación del modelo.

### Inicialización de pesos y sesgos:

Se utiliza un criterio de inicialización aleatoria de pesos y sesgos para el modelo.

### Definición de la función de pérdida y algoritmo de optimización:

Se utiliza Adam como optimizador y Cross Entropy Loss como función de error. De esta ultima se habló en capítulos anteriores, ahora veremos alguna característica del optimizador.

Adam es un algoritmo de optimización popularmente utilizado en el campo del aprendizaje automático para ajustar los parámetros de los modelos durante el proceso de entrenamiento.

Este algoritmo combina elementos del descenso de gradiente estocástico (SGD) con técnicas de estimación de momentos adaptativos. A diferencia del SGD tradicional, Adam mantiene un conjunto de promedios móviles de los gradientes y de las segundas derivadas de los parámetros del modelo. Estos promedios móviles adaptativos se utilizan para ajustar las tasas de aprendizaje de cada parámetro individualmente.

Las principales ventajas de Adam incluyen:

* Eficiencia: Adam combina las ventajas de los algoritmos de descenso de gradiente estocástico y los métodos adaptativos, lo que lo hace eficiente en términos de tiempo de convergencia y requisitos de memoria.
* Adaptabilidad: Adam adapta las tasas de aprendizaje de manera individualizada para cada parámetro, lo que permite un ajuste más preciso y rápido.
* Robustez ante gradientes dispersos: Adam tiene un buen rendimiento en problemas con datos dispersos o con gradientes de baja variabilidad.

A pesar de sus ventajas, Adam también tiene algunas consideraciones como la Sensibilidad a hiperparámetros(tiene muchos para ajustar) o el Tamaño del batch(muy chico o muy grande).

En general, Adam es una opción popular para el entrenamiento de modelos de aprendizaje automático debido a su eficiencia y adaptabilidad.

### Ciclo de entrenamiento:

Se utiliza la herramienta W&B para realizar varios entrenamiento variando parámetros como será la cantidad de epochs, el Learning rate, y otros.

Abarcar varias configuraciones de hiperparámetros, evita caer en problemas en el algoritmo de aprendizaje como puede ser problemas de gradiente y aprendizaje lento.

La tasa de aprendizaje es un parámetro crítico en los algoritmos de optimización utilizados en el aprendizaje automático. Define la velocidad a la que un modelo de ML ajusta sus parámetros durante el proceso de entrenamiento.

En términos sencillos, la tasa de aprendizaje determina cuánto deben actualizarse los parámetros del modelo en respuesta a los gradientes calculados durante el entrenamiento. Una tasa de aprendizaje alta permite actualizaciones más grandes en cada iteración, lo que puede llevar a una convergencia más rápida. Por otro lado, una tasa de aprendizaje baja realiza actualizaciones más pequeñas y gradualmente se acerca al óptimo, pero puede llevar más tiempo converger.

Seleccionar una tasa de aprendizaje adecuada es crucial, ya que puede influir en el rendimiento y la eficiencia del modelo. Si la tasa de aprendizaje es demasiado alta, el modelo puede oscilar y no alcanzar un óptimo, o incluso divergir. Por otro lado, si la tasa de aprendizaje es demasiado baja, el modelo puede tardar mucho tiempo en converger o quedar atrapado en un mínimo local subóptimo.

### Evaluación del rendimiento:

Al final de cada época o después de completar el entrenamiento, se evalúa el rendimiento del modelo en el conjunto de validación. Se estudian parámetros como son la Loss y la accuracy.

Se utilizara la aplicación W&B para el seguimiento de todos los entrenamientos realizados y poder sobre el final realizar un chequeo de resultados de buena forma.

# VI. Implementación en producción

## A. Consideraciones de infraestructura

## B. Integración del modelo en un sistema de producción

## C. Monitoreo y mantenimiento del modelo en producción

# VII. Desafíos y recomendaciones

## A. Desafíos comunes en la implementación de Machine Learning en producción

## B. Recomendaciones para superar los desafíos

# VIII. Conclusiones

## A. Resumen de los hallazgos

## B. Contribuciones del estudio

## C. Áreas para futuras investigaciones

FIN DEL DOCUMENTO