UNIVERSIDAD GALILEO

FACULTAD DE INGENIERÍA DE SISTEMAS, INFORMÁTICA Y CIENCIAS DE LA COMPUTACIÓN



**Reporte de Competencia de Kaggle**

**“House Prices”**

RODRIGO ALBERTO CORDERO ALVAREZ 17001922

MARCELO ALFREDO DEL ÁGUILA MORAGA 17001380

KEVIN JOSÉ HERNÁNDEZ MARROQUÍN 17001095

INVESTIGACIÓN DE OPERACIONES I

SÉPTIMO SEMESTRE

GUATEMALA, 22 DE ARBIL DE 2020

**Índice**

[Objetivo General: 3](#_Toc38134872)

[Objetivos Específicos: 3](#_Toc38134873)

[ Correlación: 4](#_Toc38134874)

[ Feature Engineering: 4](#_Toc38134875)

[ Integer Encoding: 4](#_Toc38134876)

[ Métricas: 4](#_Toc38134877)

[o RMS (Root Mean Squared): 4](#_Toc38134878)

[o Mean Absolute Error: 4](#_Toc38134879)

[ Normalización: 5](#_Toc38134880)

[ One Hot Encoding 5](#_Toc38134881)

[ R: 5](#_Toc38134882)

[ Redes Neuronales: 5](#_Toc38134883)

[ Regresión: 5](#_Toc38134884)

[ Regresión Lineal: 5](#_Toc38134885)

[ Tensorflow: 5](#_Toc38134886)

[Situación por Enfrentarse: 6](#_Toc38134887)

[Problemas Potenciales 6](#_Toc38134888)

[Regresiones Lineales 7](#_Toc38134889)

[Regresiones Lineales con XGBoost 8](#_Toc38134890)

[Redes Neuronales 8](#_Toc38134891)

[Fase de Análisis de Datos 10](#_Toc38134892)

[Fase de Creación y Evaluación del Modelo 11](#_Toc38134893)

[Fase de Resultados y Retroalimentación 12](#_Toc38134894)

[Fase de Análisis de Datos 13](#_Toc38134895)

[Resultados de todos los modelos que realizamos 15](#_Toc38134896)

[Resultados del mejor modelo que realizamos 15](#_Toc38134897)

Introducción

Con la tecnología de hoy en día, ha sido posible realizar modelos que tengan la capacidad de predecir valores o situaciones.

En todo el mundo, siempre existirán problemas de predicción. Esto se hace notar considerablemente en las empresas ya que siempre buscan tomar la mejor decisión para evitar caer en situaciones no deseables (como en bancarrota). Escenarios como estos, demuestran la importancia de la capacidad de poder predecir en base a ciertos parámetros importantes.

Un problema muy común es el de poder predecir un número dado ciertos parámetros. Este se conoce como un problema de regresión.

Un problema de regresión que se hace destacar es el problema de predecir el costo de casas en base a ciertas características.

Este problema se le conoce como “House Prices” y ha sido publicado en la página de Kaggle.com como un tipo de competencia para determinar quién logra hacer el mejor modelo de predicción.

Nuestra tarea como grupo, es de construir un modelo muy bueno dedicado a la predicción de los costos de casas.

Objetivos

# Objetivo General:

Construir un modelo robusto que sea capaz de predecir los costos de casas (en base a ciertos parámetros) y que nos permita colocarnos en una muy buena posición en la competencia de Kaggle.

# Objetivos Específicos:

* Aplicar los conocimientos adquiridos en el curso de Investigación de Operaciones I y de Seminario Profesional I.
* Realizar análisis a los Datasets proporcionados para obtener una idea de la naturaleza de los datos.
* Aplicar distintos modelos para alcanzar el mejor resultado.
* Aprender nuevas técnicas para el manejo de datos y aplicarlos a nuestro problema.

Marco Teórico

* Correlación:La Correlación en la estadística determina la relación o dependencia entre dos variables, en otras palabras, determina si los cambios de una de las variables influyen en la otra variable, de ser este el caso se dice que las variables están correlacionadas.(SuperProf, n.d.)
* Feature Engineering:Es un proceso que se encarga de usar “domain knowledge” de la data para crear cualidades que son necesarios para que el algoritmo del machine learning funcione. Si esas cualidades son hechas de manera correcta, incrementa la certeza de las predicciones al crear nuevas cualidades desde raw data que facilita el proceso de aprendizaje del algoritmo. (Shekhar, n.d.)
* Integer Encoding:Al igual que Integral Encoding este proceso se encarga de tomar columnas de cualitativas y convertirlas en números para poder utilizarlas en un modelo matemático, la diferencia es que este tipo de proceso se aplica a columnas cullas cualidades tengan una relación entre ellas, por ejemplo si se tiene cualidades que van de muy malo a muy bueno se le puede asignar un numero a cada cualidad por ejemplo muy malo es 0 y muy bueno es 5 lo cual nos permite mantener el mismo nivel de cualidad pero con valores numéricos.
* Métricas:en este proyecto destacaron 2 métricas que dictaron lo bueno de nuestros modelos:
  + RMS (Root Mean Squared): Es un algoritmo que se encarga de dar un promedio de un valor donde no se necesita o no es importante el signo de los valores. Por ejemplo, si se quiere obtener el promedio de los errores de un modelo de una manera convencional empleando la ecuación de la media no nos daría un valor útil debido a que los errores pueden ser negativos o positivos. El RMS nos permite obtener el promedio del error del modelo, lo cual nos permite tener una métrica para calificar nuestro modelo. (M., 2019)
  + Mean Absolute Error: Al igual que el valor RMS el MAE nos permite obtener el promedio de las diferencias entre los valores obtenidos por el modelo y los valores reales, la diferencia entre ambas métricas es su forma de calcularla. (Urquhart, 2013)
* Normalización:La normalización es un proceso en el cual se toma una columna con valores enteros o reales y se restringen a valores entre 0 y 1, lo cual nos permite reducir el procesamiento a la hora de entenar el modelo debido a que es más fácil para la computadora trabajar con ese tipo de valores.
* One Hot Encoding**:** Es un proceso diseñado para tomar columnas cualitativas que sus ítems no sigan un patrón de escala (peor a mejor o viceversa) sino que son cualidades separadas y convertirlas en columnas con números enteros.

La manera en la que funciona es, se observa las distintas opciones que la columna puede tener y se crea una nueva columna por cada opción. Los datos que en la columna original su valor coincida con alguna de las nuevas columnas, en esa columna se pone un “1” y en las restantes 0.

* R:Es un lenguaje y un ambiente de programación dedicado a estadística computacional y gráficos. R comparte muchas similitudes con otro leguaje de programación llamado S; R es un proyecto de GNU y S es un lenguaje desarrollado por AT&T. R proporciona una amplia gama de funciones estadísticas como herramientas gráficas y es un ambiente Open Source. (Project, n.d.)
* Redes Neuronales:Es una clase de inteligencia artificial que intenta imitar la forma en la que el cerebro humano funciona. Una red neuronal funciona creando conexiones entre “elementos de procesamiento”, que es el equivalente computacional de una neurona. Se le asigna un peso a cada conexión entre las neuronas y este peso cambia conforme la maquina aprende la naturaleza de los datos. (webopedia, 2020)
* Regresión: Es una clase de algoritmos en estadística que se utilizan para predecir valores o resultados en dependencia de una cantidad n de variables, las regresiones pueden ser: lineales, polinómicas, logísticas, logarítmicas, entre otras.
* Regresión Lineal:una regresión lineal es un algoritmo de predicción que se encarda de generar una recta de predicción que se acople de la mejor manera a los valores que se tengan, una de sus características mas importantes es que la distancia entre los valores reales a la recta de predicción sumados da 0, eso nos permite obtener una sola recta de interés.
* Tensorflow: Es una librería open source y compatible con Python que facilita el desarrollo de Machine learning. (Yegulalp, 2019)

Problema

# Situación por Enfrentarse:

El problema que nos enfrentamos, según la página oficial de Kaggle, es la predicción del valor de una casa en base a ciertos parámetros. Quieren demostrar que el precio de una casa depende más que solo el material con el que está hecho, el número de cuartos, etc.

Nosotros podemos deducir que este problema es de tipo de regresión.

En la página oficial de Kaggle, menciona una breve introducción al problema y nos proporciona ciertos archivos que nos ayudarán a resolver este problema.

Entre los archivos que nos proporciona podemos encontrar:

* *train.csv*: Archivo que contiene el dataset para el entrenamiento del modelo.
* *test.csv*: Archivo que contiene el dataset para que el modelo prediga los precios de las casas.
* *data\_description.txt*: Archivo que contiene una descripción detallada de todos los features que se encuentran en los datasets.
* *sample\_submission.csv*: Archivo que muestra un template de cómo se debe subir las predicciones a la competencia.

La descripción que nos muestra Kaggle sobre los datasets es que hay un total de 79 features y en base a ellas se quiere predecir ‘SalePrice’ (nuestra variable objetivo), que es el costo de la casa; este es el objetivo del modelo a construir.

# Problemas Potenciales

* Poco conocimiento y poca práctica en el área de Ciencias de Datos.
* Muchas variables explicativas; hay tantas variables categóricas como cuantitativas.
* Disponemos de 1460 datos para entrenamiento, consideramos que es poco.
* No conocemos demasiadas herramientas para armar un modelo muy robusto.
* Es una competencia abierta para muchos participantes para todos los niveles, entonces será difícil conseguir una posición muy alta.

Approaches

Los modelos utilizados son los siguientes:

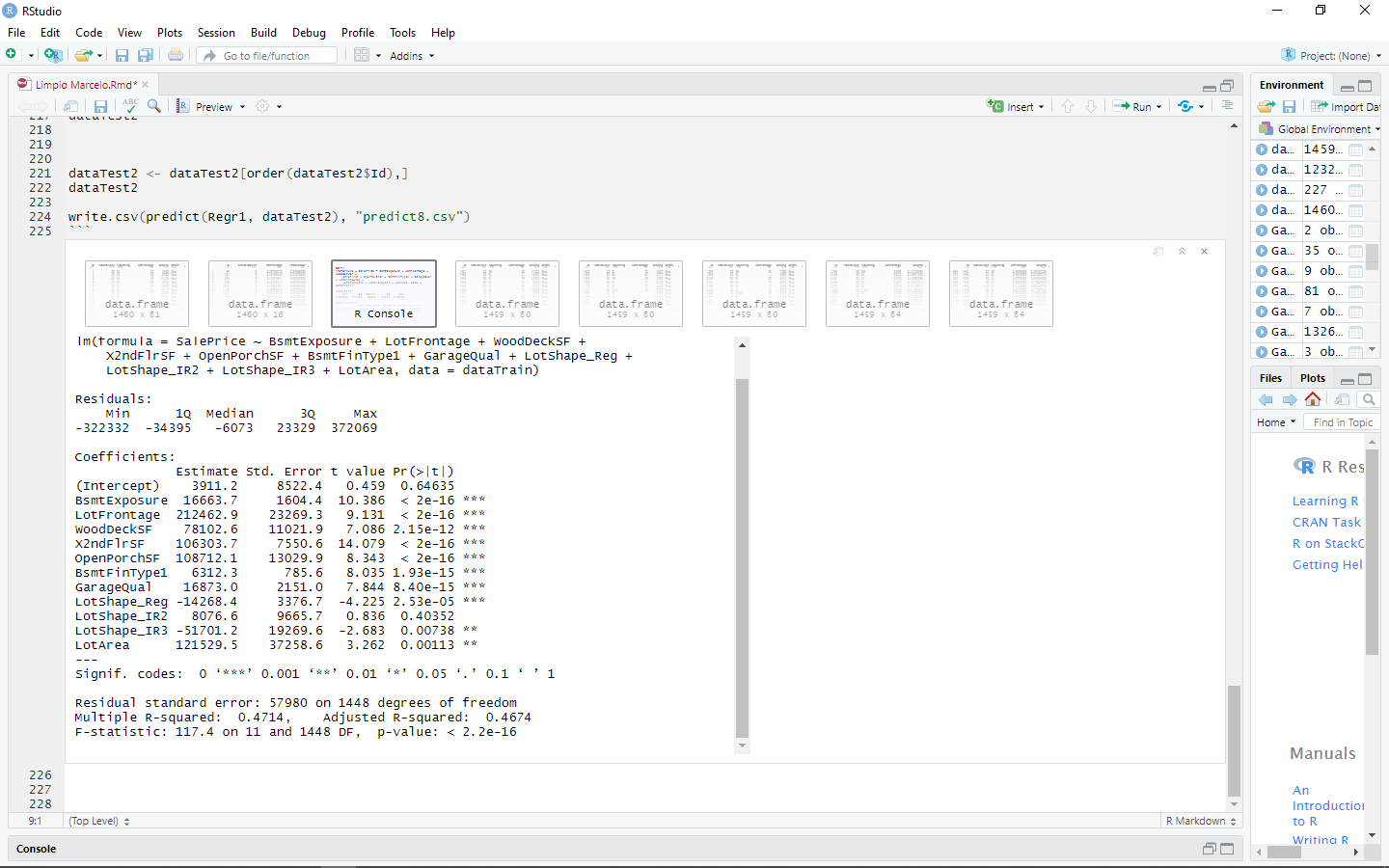
* Regresiones Lineales
* Redes Neuronales
* Regresiones Lineales con XGBoost

# Regresiones Lineales

Las regresiones lineales pueden ser el tipo de regresiones mas sencillo y de los menos exactos en todos los casos, pero nosotros decidimos utilizar este modelo de regresiones debidos a que teníamos una cantidad muy alta de variables que nos permitiría hacer una hiper recta de +80 dimensiones para poder obtener por medio de la función de summary() las variables mas importantes o con más peso.

Después de limpiar toda la data y de darla un formato, normalizándola y aplicándoles One Hot Encoding e Integral Encoding, generamos una regresión lineal de +80 variables y fuimos descartando de a poco las variables con menos peso hasta que obtuvimos 33 variables con tres estrellas, y nos decidimos a hacer 3 modelos de 11 variables y observar cuales variables nos daban el mejor resultado y unir los 3 modelos y obtener nuestro modelo final.

Luego de obtener los modelos logramos tener un modelo que nos daba el mejor resultado posible con ese modelo, que no era malo, pero no era el mejor.



Como podemos observar en la imagen anterior el modelo no solo consta de variables de 3 estrellas y so se debe a que un modelo solo con 3 estrellas no nos daba un r2

mayor a 0.30 lo cual es muy bajo, pero con el modelo actual a pesar de solo tener un r2 = 0.4674 nos permitió tener un valor RMSE de 0.338 el cual no es malo pero para los estándares de la competencia no era suficiente.

# Regresiones Lineales con XGBoost

# Redes Neuronales

Las redes neuronales pueden ser modelos muy eficaces tanto para los problemas de regresión como los problemas de clasificación.

Se decidió utilizar redes neuronales debido a que un integrante del grupo tiene experiencia moderada con dichos modelos.

El reto de las redes neuronales es encontrar la mejor arquitectura que pueda realizar muy buenas predicciones.

El approach que se tomó es armar una arquitectura cuyas capas ocultas están compuestas por una cantidad de neuronas en potencias de 2 y decrecientes e ir agregando o quitando en dependencia de los resultados que proporcione la página de Kaggle.

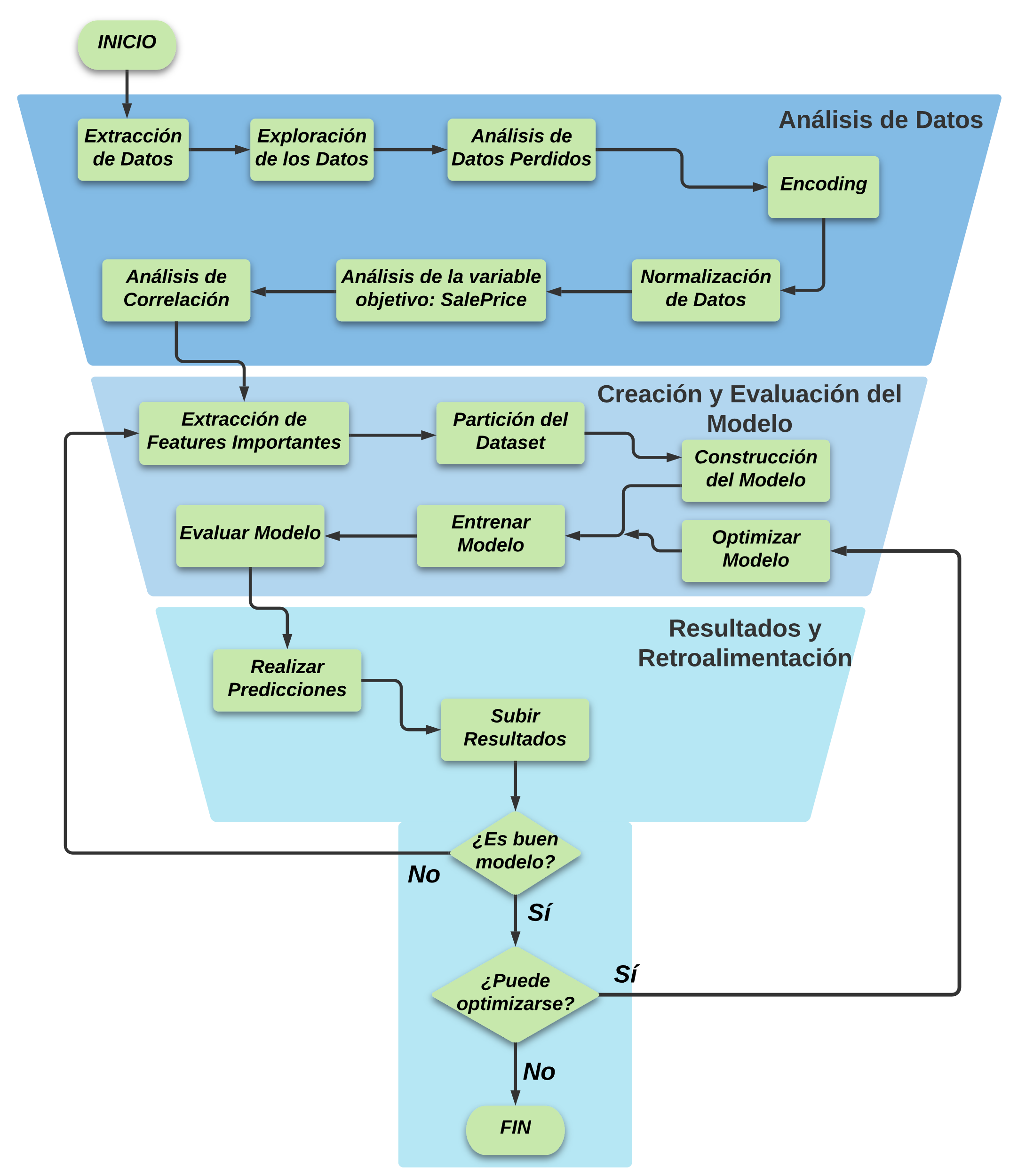
Para las redes neuronales, hay una cantidad muy grande de parámetros que se pueden modificar, sin embargo, para reducir dicha cantidad, solo se enfocará en modificar:

* El número de capas ocultas de la arquitectura.
* El número de neuronas por capa oculta.
* Inicialización de pesos de las neuronas.
* El optimizador de la red.
* La métrica de la red.
* Batch Size y Validation Split.
* Número de Epochs.
* Función de Activación

Para evaluar qué tan buena es la red, se utiliza la métrica de Loss.

Metodología

La metodología utilizada para la realización de este proyecto es el siguiente:



Detalles de la Metodología

# Fase de Análisis de Datos

1. Extracción de Datos:

Consiste en extraer los datos contenidos en los archivos *train.csv* y *test.csv* y construir los datasets de train y test, respectivamente.

1. Exploración de los Datos:

Consiste en conocer ambos datasets (train y test) y observar qué features tiene cada uno, el tipo de dato de cada feature (si es categórico o cuantitativo) y cuántos datos tiene cada dataset. Se decidió combinar ambos datasets en uno solo para analizar los datos.

En este paso se utilizó mucho el archivo *data\_description.txt* para ir analizando qué significa cada feature y los posibles valores que podía tener.

1. Análisis de Datos Perdidos:

Consiste en conocer qué features tienen “Missing Values”, valores Nulos o valores de tipo NaN. Se consultó muy frecuentemente el archivo *data\_description.txt*, ya que especificaba que algunos features que poseen Missing Values o NaN sí representaban un valor como tal.

Después de haber hecho el análisis descrito anteriormente, se procede a aplicar técnicas[1] para el reemplazo y rellenado de Missing Values o NaNs.

1. Encoding:

Consiste en aplicar las técnicas de Integer Encoding o One Hot Encoding a cada feature de tipo categórico, según sea necesario.

1. Normalización de Datos:

Consiste en normalizar los datos de cada feature para disminuir el tiempo de entrenamiento del modelo y mejorar su capacidad para predecir.

Una vez terminada la normalización de datos, se separa el dataset combinado en los datatsets train y test.

1. Análisis de la variable objetivo ‘*SalePrice*’:

Consiste en realizar un análisis estadístico a la variable SalePrice:

* Analizar su distribución
* El sesgo de la gráfica de distribución
* Media
* Varianza

1. Análisis de Correlación:

Consiste en analizar la correlación de todos los features con la variable objetivo: SalePrice.

Además, se realiza un mapa de calor de correlación para facilitar la visualización de las correlaciones entre los distintos features.

# Fase de Creación y Evaluación del Modelo

1. Extracción de Features Importantes:

En base al análisis de correlación, se extraen los features que cumplen con el criterio de “Threshold”, es decir, si la correlación de un determinado feature con SalePrice es igual o mayor a un Threshold establecido, entonces es considerado como un feature importante.

Por último, se crea un nuevo dataset ‘*train’* compuesto únicamente por el conjunto de features extraídos anteriormente.

1. Partición del Dataset:

Consiste en hacer un shuffle al nuevo dataset y particionarlo en 2 nuevos datasets: *newTrain* y *evaluation*. El propósito del dataset *newTrain* es ser utilizado para el entrenamiento del modelo, y el propósito del dataset *evaluation* es para evaluar qué tan bueno es nuestro modelo. Se utiliza un criterio de 80% de los datos para newTrain y 20% para *evaluation* y

1. Construcción del Modelo:

Consiste en crear el modelo ya sea usando una regresión lineal, regresión XGBoost o una red neuronal.

1. Entrenar Modelo:

Consiste en utilizar el dataset *newTrain* para entrenar el modelo construido anteriormente.

1. Evaluar Modelo:

Consiste en utilizar el dataset *evaluation* para evaluar el desempeño del modelo.

Para los modelos de regresión lineal o regresión XGBoost se utilizó la métrica RMSE para evaluar el desempeño.

Para los modelos de tipo red neuronal, se utilizó la métrica Mae y Loss, y en algunos casos RMS y Loss.

# Fase de Resultados y Retroalimentación

1. Realizar Predicciones:

Consiste en utilizar el modelo para predecir la variable objetivo (SalePrice) para los distintos datos contenidos en el dataset *test*.

1. Subir Resultados:

Consiste en crear el archivo con el formato especificado en Kaggle (usando las predicciones obtenidas anteriormente) para luego subir nuestras predicciones y así obtener un score y avanzar en la tabla de Leaderboard.

1. (Paso Extra) Optimizar Modelo:

Luego de obtener el respectivo punteo en Kaggle, se decide si vale la pena optimizar el modelo cambiando ciertos parámetros según sea el modelo.

Hallazgos

# Fase de Análisis de Datos

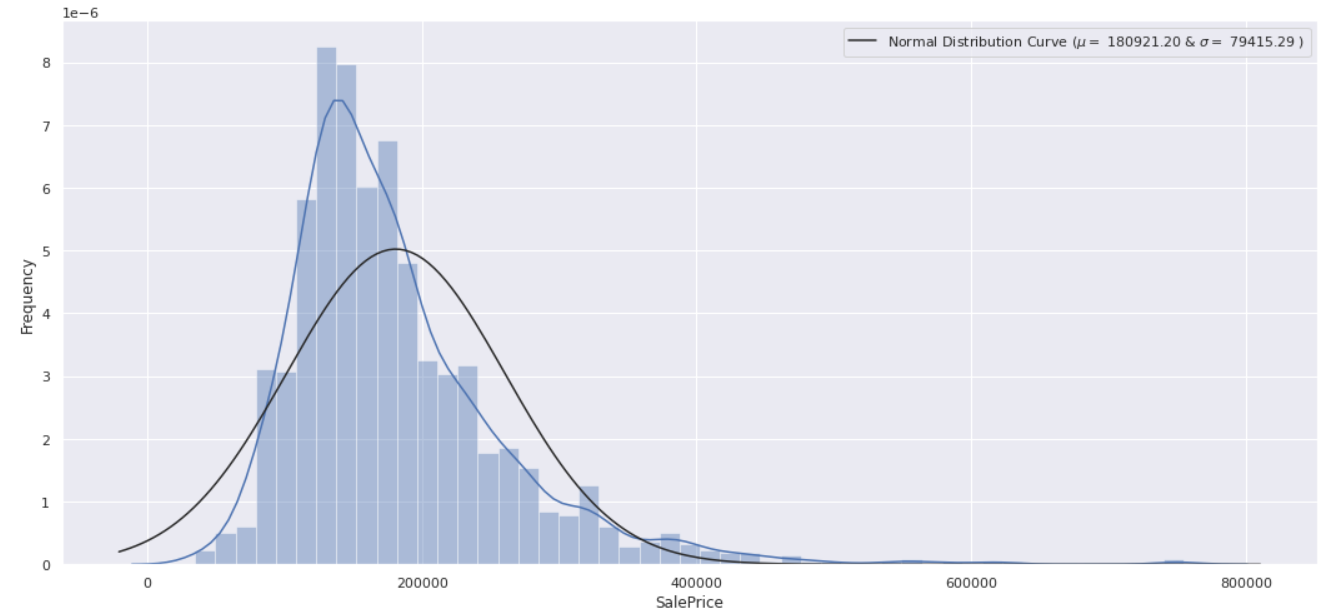
* Análisis de Datos Perdidos:

Se encontraron varios features que poseen Missing Values o NaNs. Sin embargo, el archivo no describía que dichos features tenían la posibilidad tener un valor NaN o un “Missing Value”.

Para las técnicas de reemplazo y rellenado de Missing Values o Nans, se decidió que:

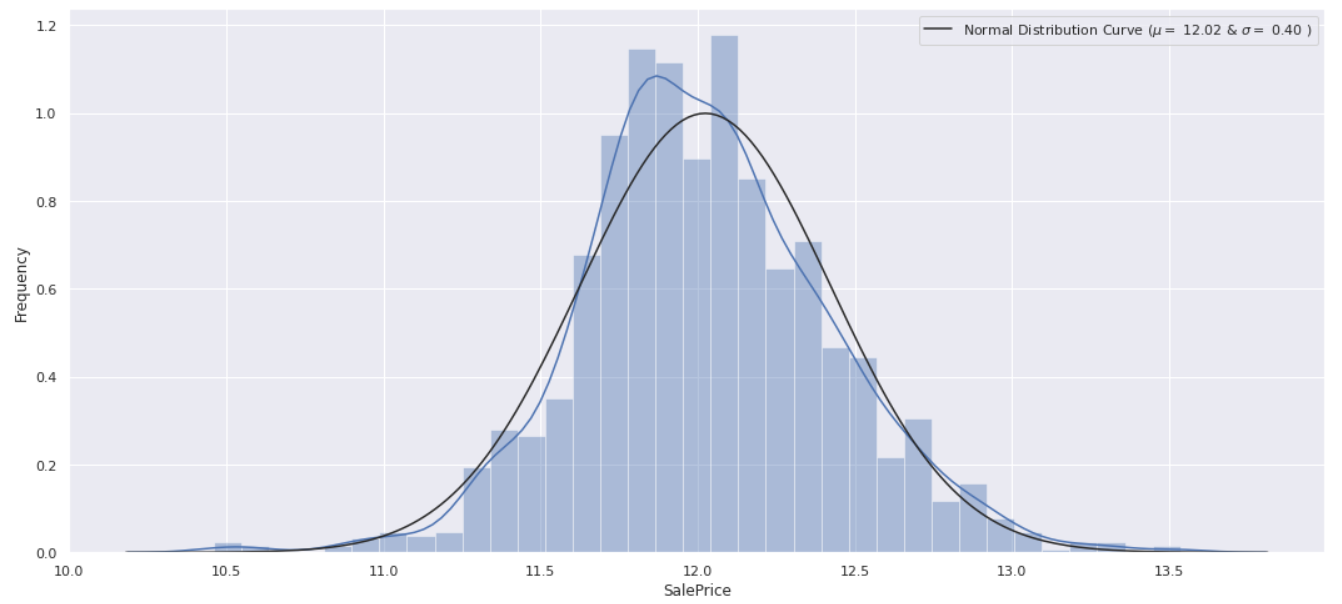
* + Para el feature *‘LotFrontage’,* los valores NaN son reemplazados por la mediana del feature*.*
  + Para los Features *‘GarageyrBlt’*, *‘GarageArea’*, *‘GarageCars’*, *‘BsmtFinSF1’*, *‘BsmtFinSF2’*, *‘BsmtUnfSF’*, *‘TotalBsmSF’*, *‘BsmtFullBath’*, *‘BsmtHalfBath’*, *‘BsmtFinSF1’* y *‘MasVnrArea’*, los valores NaN son reemplazados por un 0.
  + Para el feature *‘MSZoning’*, los valores NaN son reemplazados por la moda del feature.
* Análisis de la variable objetivo ‘*SalePrice*’:

Se descubre que la gráfica de distribución de frecuencias de la variable SalePrice se encuentra sesgada hacia la izquierda, además se descubre que dicha gráfica tiene una curtosis de tipo Leptocúrtica.



***Gráfica de Distribución de Frecuencias de SalePrice***

A la variable objetivo se le aplica una transformación de tipo Log(p+1) con el fin de normalizar la data.



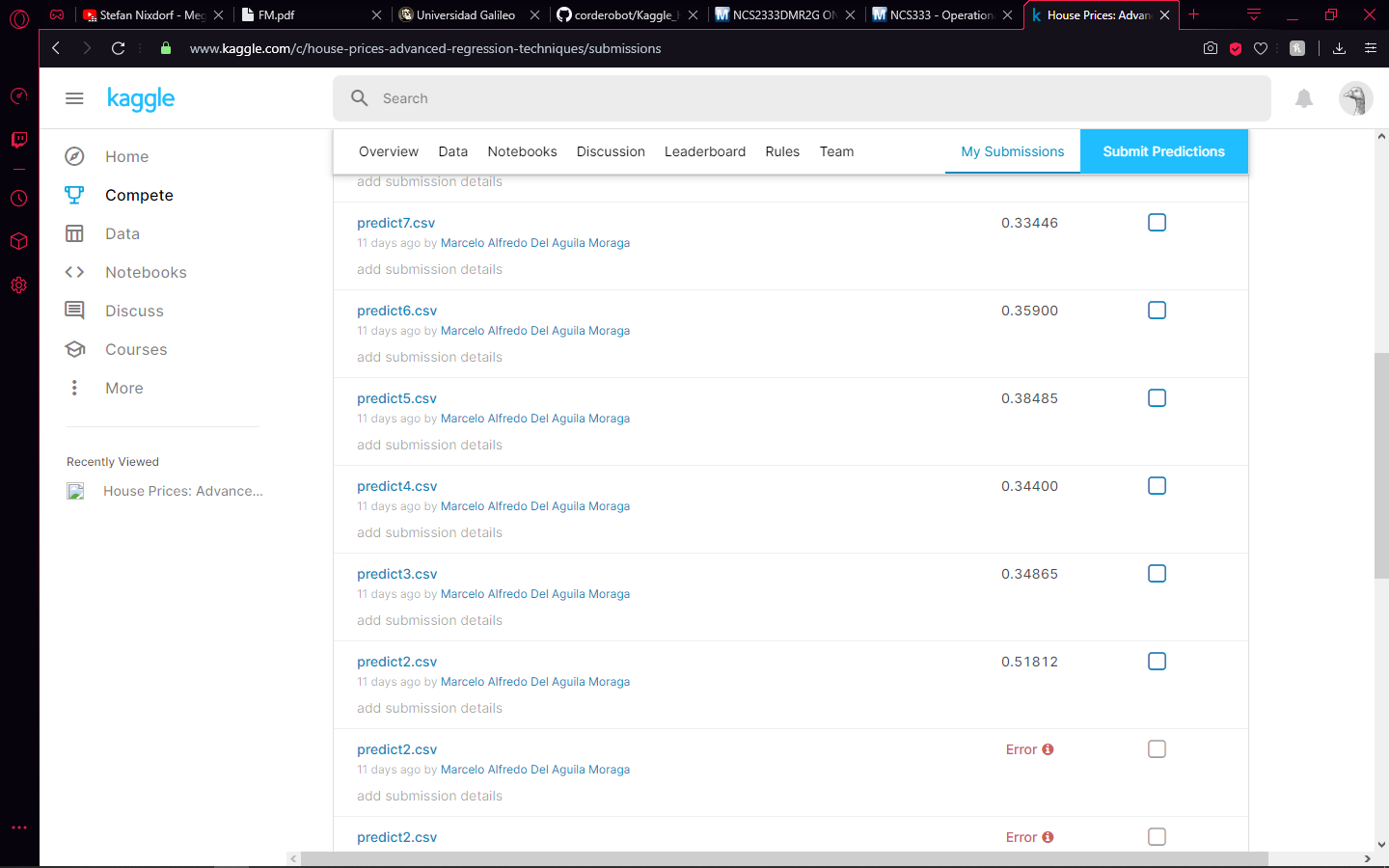
***Gráfica de Distribución de Frecuencias de SalePrice luego al aplicarle la transformación Log(p+1)***

Resultados

# Resultados de todos los modelos que realizamos

De los modelos realizados con regresiones lineales nuestros modelos fueron variando mucho, pero ninguno nos llevo a la meta esperada.

De parte de Marcelo del Aguila, los mejores resultados obtenidos con una regresión lineal de 15 variables fueron los siguientes:



A pesar de la mejora no nos llevo mas lejos que el P30.

**Resultados: Redes Neuronales**

Las redes neuronales probaron ser modelos eficaces para la solución de este problema. Se hicieron más de 50 modelos de redes neuronales, sin embargo, solo se mantuvo registro de aquellos modelos que mejoraron el puntaje en la competencia de Kaggle.

Las redes neuronales demostraron ser un arma de doble filo: son muy eficaces pero muy complejas de optimizar. El hecho de haber creado más de 50 modelos demuestra esto.

Fue realmente muy difícil de optimizar o mejorar los modelos, ya que un cambio en la arquitectura no significaba una mejora directa, si no que podría perjudicar al modelo o simplemente no afectaba ni positiva ni negativamente.

Las métricas utilizadas para medir el desempeño de la red son: Loss y Mae.

De igual forma, una mejora en el desempeño de la red no significaba una mejora directa en el punteo de Kaggle, si no se puede obtener hasta peores resultados.

Se intentaron muchas combinaciones a la hora de crear el modelo, es decir, se utilizaba distintos criterios de Threshold, distintas arquitecturas internas y distintos tiempos de entrenamiento.

Cabe mencionar que cada Red Neuronal tomaba bastante tiempo en entrenar, y este tiempo aumentaba conforme se incrementaba la complejidad de la arquitectura.

Los modelos más notables son los siguientes:

**Modelo 3:**

* Hidden Layers: 5
* Número de neuronas por capa:
  + Primera Capa: 128 Función de Activación: Relu
  + Segunda Capa: 128 Función de Activación: Relu
  + Tercera Capa: 128 Función de Activación: Relu
  + Cuarta Capa: 128 Función de Activación: Relu
  + Quinta Capa: 128 Función de Activación: Linear
  + Sexta Capa: 1
* Threshold: 0.5
* Epochs: 500
* Batch Size: 64
* Optimizer: Adam
* Metricas: RMS
* Loss: No hay registro
* Kaggle Score: 0.18878

**Modelo 4:**

* Hidden Layers: 5
* Número de neuronas por capa:
  + Primera Capa: 128 Función de Activación: Relu
  + Segunda Capa: 128 Función de Activación: Relu
  + Tercera Capa: 128 Función de Activación: Relu
  + Cuarta Capa: 128 Función de Activación: Relu
  + Quinta Capa: 128 Función de Activación: Linear
  + Sexta Capa: 1
* Threshold: 0.35
* Epochs: 500
* Batch Size: 64
* Optimizer: Adam
* Metricas: RMS
* Loss: 17771.123046875
* Kaggle Score: 0.15551

**Modelo 57:**

* Hidden Layers: 5
* Número de neuronas por capa:
  + Primera Capa: 512 Función de Activación: Relu
  + Segunda Capa: 256 Función de Activación: Relu
  + Tercera Capa: 256 Función de Activación: Relu
  + Cuarta Capa: 256 Función de Activación: Relu
  + Quinta Capa: 256 Función de Activación: Relu
  + Sexta Capa: 128 Función de Activación: Linear
  + Séptima Capa: 1
* Threshold: 0.1
* Epochs: 475
* Batch Size: 32
* Optimizer: Adamax
* Metricas: Mae
* Loss: 0.08765507489442825
* Kaggle Score: 0.15383

**Modelo 74:**

* Hidden Layers: 5
* Número de neuronas por capa:
  + Primera Capa: 512 Función de Activación: Sigmoid
  + Segunda Capa: 256 Función de Activación: Sigmoid
  + Tercera Capa: 256 Función de Activación: Sigmoid
  + Cuarta Capa: 256 Función de Activación: Sigmoid
  + Quinta Capa: 256 Función de Activación: Sigmoid
  + Sexta Capa: 128 Función de Activación: Linear
  + Séptima Capa: 1
* Threshold: 0.1
* Epochs: 475
* Batch Size: 32
* Optimizer: Adam
* Metricas: Mae
* Loss: 0.07929326593875885
* Kaggle Score: 0.13637

Interpretación de los Resultados

Conclusiones

* El paso fundamental es hacer un análisis del dataset previo a construir los modelos.
* Es muy importante realizar un análisis a todas las variables involucradas para obtener una idea de la naturaleza de los datos.
* Es necesario aplicar técnicas de ‘Feature Engineering’ y de limpieza de datos para mejorar el desempeño de los modelos a construir.
* Es difícil encontrar cómo ‘mejorar’ un modelo y asegurar que esa ‘mejora’ sea algo positivo y no sea perjudicial.
* El desempeño de las redes neuronales decrece conforme se aumenta la complejidad del mismo.

Observaciones

* Al principio nos dispusimos en resolver el problema con únicamente regresiones lineales, pero debido a que las mismas no nos llevaron muy lejos en el ranking, entonces decidimos que uno de nosotros se fuera por otro camino, con Redes neuronales el cual fue más productivo
* Cuando se empezó a limpiar la data y a seleccionar las variables que más efecto tenían en el modelo, hubiera sido mejor haber tomado grupos de 10 en 10 e ir obteniéndolas variables más impactantes de esa manera porque al obtener un modelo con 80 variables la función summary pierde fiabilidad.
* Constantemente nos estancábamos y había largos períodos de tiempo (2 a 3 días) que no se lograba avanzar en el Leaderboard.
* Nos percatamos que era sumamente importante conocer muchas técnicas de Feature Engineering y de procesamiento de la data.
* El proyecto obliga a investigar mucho sobre qué se debe hacer para tener un puesto relativamente bueno.

**Recomendaciones**

* Es recomendable siempre limpiar su data antes de analizarlo y leer las instrucciones para ver si algunos NaN o Null tienen significado útil.
* Se recomienda el uso de Hot One Encoding e Integer Encoding debido a que hay muchas columnas que son cualitativas que llegan a ser muy importantes para los modelos, ya sea Regresión o Red neuronal.
* Siempre al realizar un modelo, es necesario conocer sus parámetros y tener una idea cómo está estructurado. Es muy recomendable conocer cómo está estructurado el modelo luego de haberlo entrenado.
* Se recomiendo realizar una extensa investigación y trabajos previamente hechos para tener una idea clara de qué proceso se debe seguir para lograr el objetivo de hacer un modelo robusto.

Anexos

* **Repositorio de GitHub:**

<https://github.com/corderobot/Kaggle_HousePrices>

Referencias Bibliográficas

# 

M., M. (24 de Octube de 2019). *Media cuadrática*. Obtenido de Wikipedia: https://es.wikipedia.org/wiki/Media\_cuadrática#Media\_cuadrática\_de\_la\_velocidad\_de\_un\_gas

Project, R. (s.f.). *What is R?* Obtenido de r-project: https://www.r-project.org/about.html

Shekhar, A. (s.f.). *LetsLearnAI: What Is Feature Engineering for Machine Learning?* Obtenido de Medium: https://medium.com/mindorks/what-is-feature-engineering-for-machine-learning-d8ba3158d97a

SuperProf. (s.f.). *Qué significa correlación estadística en Matemáticas*. Obtenido de SuperProf Diccionario: https://www.superprof.es/diccionario/matematicas/estadistica/correlacion-estadistica.html

Urquhart, B. (2013). *Mean Absolute Error*. Obtenido de ScienceDirect: https://www.sciencedirect.com/topics/engineering/mean-absolute-error

webopedia. (2020). *neural network*. Obtenido de webopedia: https://www.webopedia.com/TERM/N/neural\_network.html

Yegulalp, S. (18 de Junio de 2019). *What is TensorFlow? The machine learning library explained*. Obtenido de Info World: https://www.infoworld.com/article/3278008/what-is-tensorflow-the-machine-learning-library-explained.html