Estimación de Duración de Proyectos de Ciencia, Tecnología e Innovación

Generación y Evaluación de Múltiples Modelos de Regresión

Gastón E. Rimbano
Project Manager en BOGE Ibérica
Ingeniero Mecánico

Abstract

El objetivo de este trabajo es estimar la duración en días de proyectos de ciencia, tecnología e innovación

Keywords

Proyectos, Ciencia, Tecnología, Innovación, Duración,
Regresión, Lineal, Ridge, Lasso, ElasticNet, KNN, SVR, Random Forest, MLP

1 INTRODUCCIÓN

Este trabajo se basa en un análisis de información asociada a diferentes proyectos de ciencia, tecnología e innovación, buscando así encontrar el mejor modelo de regresión que permita estimar la duración en días de los mismos.

Para esto, se toman como datos de partida información como su área del conocimiento, cantidad de miembros y montos adjudicados, entre otras variables.

El objetivo de esta estimación es mejorar el seguimiento realizado sobre este tipo de proyectos. Se cree que, partiendo de la fecha de inicio y la duración en días, sería relevante establecer fechas límites para la revisión del trabajo realizado por los equipos que conforman los diferentes proyectos.

2 EXTRACCIÓN Y PROCESAMIENTO DE DATOS

2.1 Origen de los Datos

Los datos con los que se ha trabajado han sido obtenidos del portal abierto de datos llamado "Datos Argentina". Pueden descargar los sets de datos utilizados accediendo al siguiente enlace:

https://datos.gob.ar/dataset/mincyt-proyectos-ciencia-tecnologia-e-innovacion

El data set en su totalidad se compone de unidades estadísticas que incluyen los proyectos de I+D y/o de innovación (conjunto de actividades que se llevan a cabo para crear resultados CyT y/o innovativos en un tiempo determinado). Su carga y actualización está a cargo del Sistema de Información de Ciencia y Tecnología Argentino (SICYTAR).

2.2 Estructura del Data Set

El repertorio de datos cuenta con un total de 20 ficheros de tipo .csv, de los cuales 12 corresponden al registro de proyectos generados entre los años 2008 y 2019. Esto se debe a que se tiene un fichero por año. El resto de los ficheros son datos complementarios, entre los que se encuentran la disciplina del proyecto, el estado del proyecto, la moneda de los montos, entre otra información.

Los ficheros utilizados en el presente trabajo fueron 17, valiéndose de la información detallada en el párrafo anterior. Los ficheros con información de cada proyecto contienen 19 columnas, de las cuales 2 son de tipo 'datetime64[ns]', 7 son de tipo 'float64', 3 son de tipo 'int64', y 7 son de tipo 'object'. En el total de ficheros de cada año se reúnen unos 19.266 registros.

2.3 Análisis Exploratorio de Datos (EDA)

El pipeline utilizado en esta etapa fue el que se muestra en la **Figura 1**:

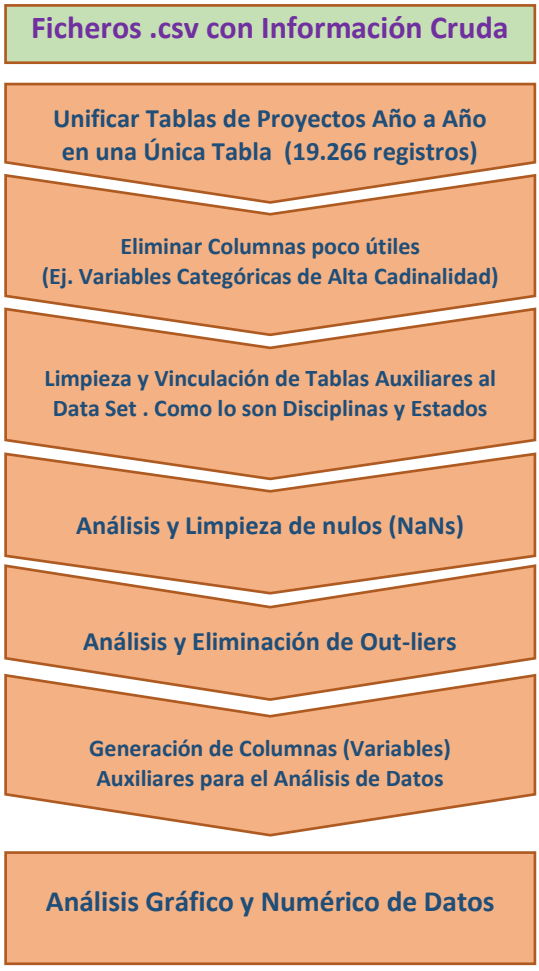

Figura 1 – Pipeline del EDA

A lo largo de todo el pipeline mostrado se resolvieron las siguientes complicaciones:

- Necesidad de generar un data set unificado que centralice los proyectos de todos los años.
- Existencia de columnas que no aportan información para el análisis, como el caso de variables categóricas de alta cardinalidad, como 'codigo_identificacion', 'titulo', 'resumen', 'palabras_claves'; y columnas que presentaban un valor único, como 'moneda_id'. Estas columnas fueron eliminadas del data set.
- Siguiendo con el descarte de columnas poco informativas, también se tuvo que eliminar la columna 'tipo_proyecto_id'. Si bien inicialmente esta columna se tuvo en cuenta a pesar de que casi un 10% de sus valores eran nulos, ya que permitía discriminar entre proyectos de 'ciencia' y de 'tecnología e innovación', al continuar con el pre-procesamiento de datos (limpieza de nulos y out-liers) se encontró que esta columna quedaba con un valor único ('ciencia'). Es por esto que, para no perder un 10% de registros del data set al mantener una columna que se volvía poco informativa, se decidió eliminar la misma al comienzo del pre-procesamiento, consiguiendo así aumentar la cantidad final de datos limpios.
- Las etiquetas de ciertas columnas que tenían datos indexados se encontraban en tablas independientes. Se anexó el campo correspondiente de estas tablas auxiliares, reemplazando los campos indexados. En el caso de 'estado', como se trataban de pocas etiquetas, se realizó un map sobre la columna.
- Existencia de columnas con casi un 50% de registros nulos. Se tuvo que realizar un análisis detallado de que era conveniente hacer en cada caso, determinando que lo óptimo era eliminar la columna 'fondo_anpcyt', por su alta tasa de registros nulos; para otros casos, en los que se tenía menor cantidad de registros nulos, se procedió a eliminar dichos registros.
- Siguiendo con el tratamiento de nulos, gracias a un análisis utilizando un heatmap para visualizar la matriz de correlación entre las variables referidas a montos (**Figura 2**), y un scatter plot de las mismas (**Figura 3**), se consiguió determinar que se podían eliminar las columnas de "montos solicitados", ya que estas contenían registros nulos, pero mostraban una información similar a la de "montos adjudicados".

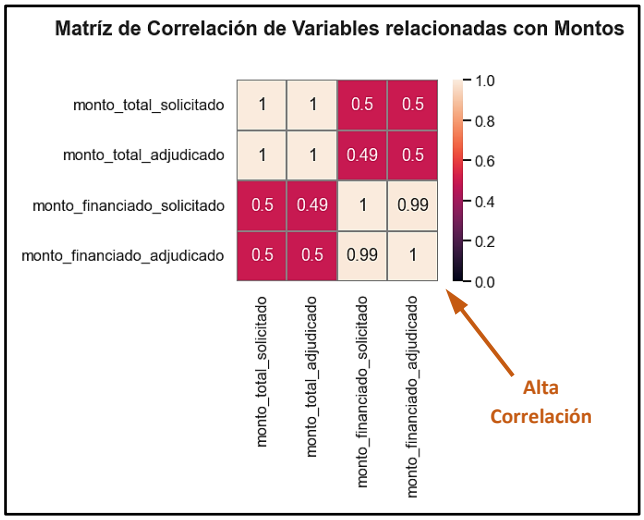

Figura 2 – Heat Map de Matriz de Correlación de Variables de Montos

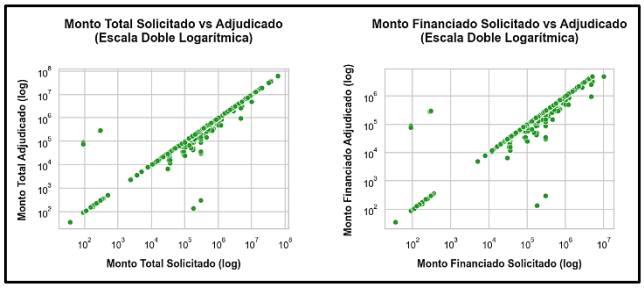

Figura 3 – Verificación Gráfica de Correlación Lineal de Montos

- Mediante un análisis gráfico y numérico se determinaron los valores out-liers de las variables numéricas 'cantidad_miembros_F' y 'cantidad_miembros_M'. En el gráfico de la **Figura 4** podemos observar la distribución de estas variables, y el umbral de outliers, calculado con el *criterio del percentil 99* [1] de cada conjunto de datos. Se eliminaron registros cuya cantidad total de miembros superaba el umbral establecido.

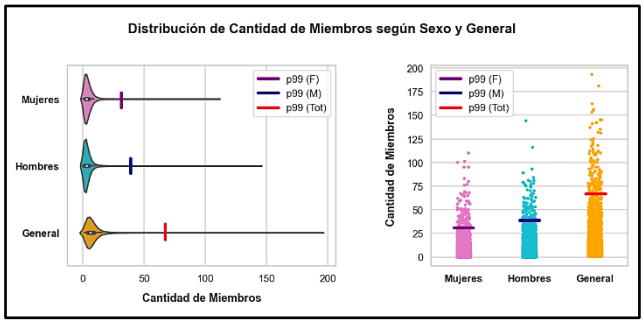

Figura 4 – Análisis Gráfico de Distribución de Cantidad de Miembros

- Para las variables 'monto_total_adjudicado' y 'monto_financiado_adjudicado' inicialmente se recurrió a un rápido análisis gráfico para ver la existencia de out-liers (**Figura 5**). Como este análisis mostró valores anómalos, se decidió proseguir con un análisis numérico, que arrojo que se tenía tanto out-liers superiores (tomando como umbral el percentil 99), como así también algunos registros con monto cero, lo que no hace ningún sentido. Estos registros fueron eliminados del data set, y se volvió a visualizar la distribución de datos de ambas variables (**Figura 6**).

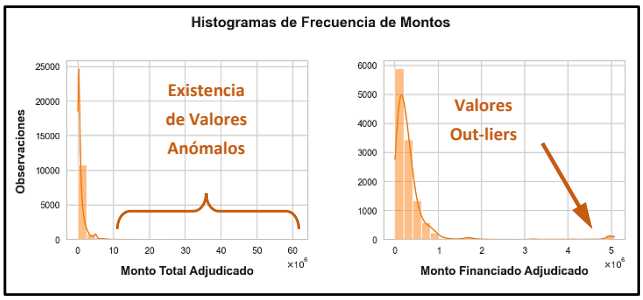

Figura 5 – Análisis Gráfico de Distribución de Montos Total y Financiado Adjudicados

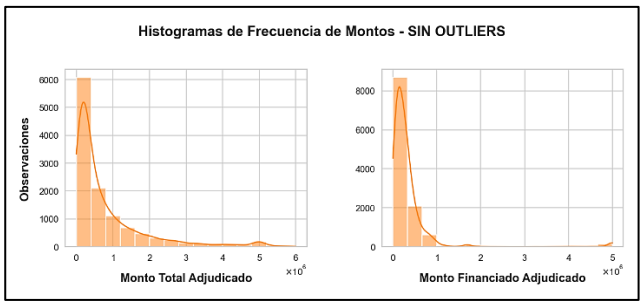

Figura 6 –Gráfico de Distribución de Montos Total y Financiado Adjudicados sin Outliers

- El data set no contaba con la variable a predecir, es decir, la duración en días del proyecto. Es por esto que se agregó la columna 'duracion_dias', calculada a partir de la diferencia entre las columnas 'año_inicio' y 'año_finalizacion'. También se generó la columna 'año_inicio', la cual resultó muy útil para el análisis gráfico del data set.
- Durante el análisis gráfico del data set, se identificó que la variable 'sexo_director' mostraba un valor poco aprovechable, que era 'S/D', y que, como puede verse en la **Figura** 7, se presentaba en muy pocos registros. Es por esto que se procedió a eliminar los mismos.

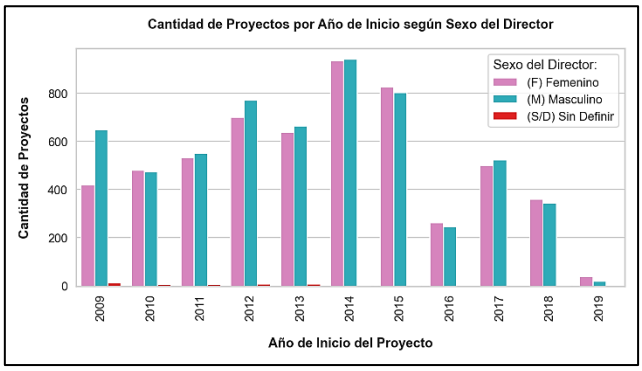

Figura 7 –Gráfico de Distribución de Proyectos por Año según el Sexo del Director

Al finalizar el pre-procesamiento de datos el data set cuenta con 13 columnas y 11.672 registros.

2.4 Análisis Gráfico y Conclusiones del mismo

Se inició este análisis visualizando la matríz de correlación de todas las variables numéricas del data set pre-procesado utilizando un heat map (**Figura 8**). En este heat map puede observarse, como podía intuirse, que las variables referidas a montos parecen correlacionar mejor entre sí que con el resto de las variables. Lo mismo ocurre con las variables de cantidad de miembro parecen. También podemos ver una leve correlación entre la variable 'monto_financiado_adjudicado' y 'cantidad_miembros_total'. Es interesante ver que 'año_inicio' tiene una pequeña correlación con las variables de montos. Esto era de esperarse, ya que los montos de los proyectos están expresados en pesos, y seguramente parte de los recursos destinados a los proyectos tienen su costo en dólares.

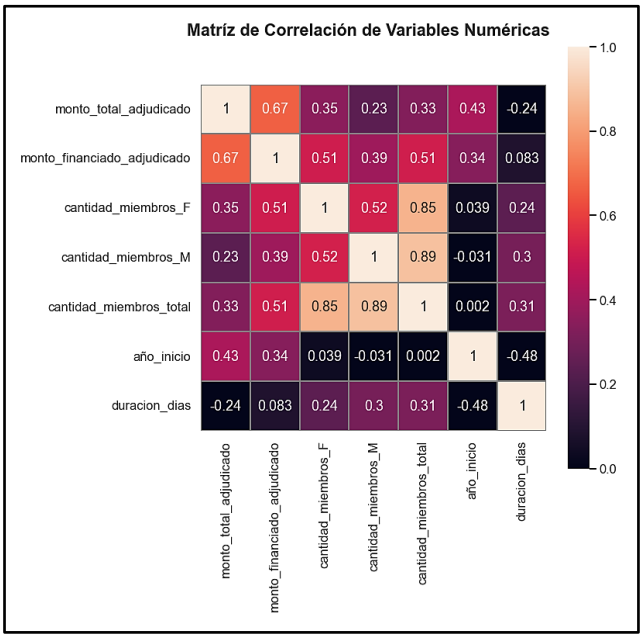

Figura 8 – Heat Map de Matriz de Correlación de Variables Numéricas

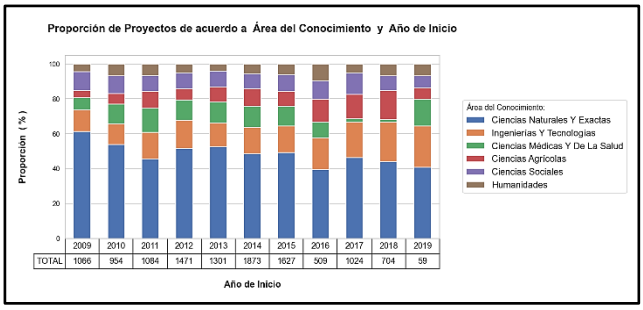

Figura 9 – Distribución de Proyectos Año a Año según su Área del Conocimiento

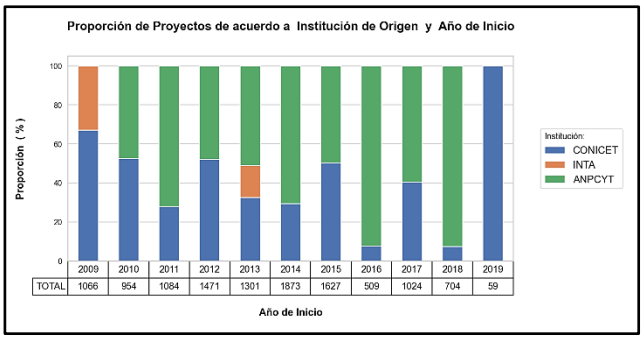

Figura 10 – Distribución de Proyectos Año a Año según su Institución de Origen

También se realizó un análisis de cómo se distribuían (en proporciones) los proyectos año a año en función de variables categóricas consideradas relevantes, como 'area_conocimiento' (**Figura 9**) e 'institucion_origen' (**Figura 10**). En ambos casos de facilitó una tabla con la cantidad total de proyectos de cada año, para poder tener una referencia de las proporciones brindadas.

En la **Figura 9** puede observarse que los proyectos de *Ciencias Naturales y Exactas* son lo que todos los años se llevan la mayor proporción. En segundo lugar, se encuentran los proyectos de *Ingenierías y Tecnologías*. También puede observarse que las áreas *Ciencias Agrícolas, Ciencias Sociales* y *Humanidades* parecen mantener la proporción (respecto del total) año a año.

Por otro lado, en la **Figura 10** puede observarse la mayor proporción año a año la presenta ANPCYT (Agencia Nacional de Promoción de la Investigación, el Desarrollo Tecnológico y la Innovación)y, en segundo lugar, se encuentra al CONICET. Esta observación acompaña la lógica de lo que se ha observado en la **Figura 9**, ya que estas instituciones son las principales impulsoras de proyectos de Ciencias Naturales y Exactas y de Ingenierías y Tecnologías, áreas la que ocupaban la mayor proporción en la figura anterior. Otro aspecto a rescatar de este gráfico es que el INTA solo muestra proyectos en los años 2.009 y 2.013. Seguramente esta institución tiene proyectos en el resto de los años, pero, o bien no ha proporcionado los datos, o no lo ha hecho de manera completa, y se han perdido durante la limpieza del data set.

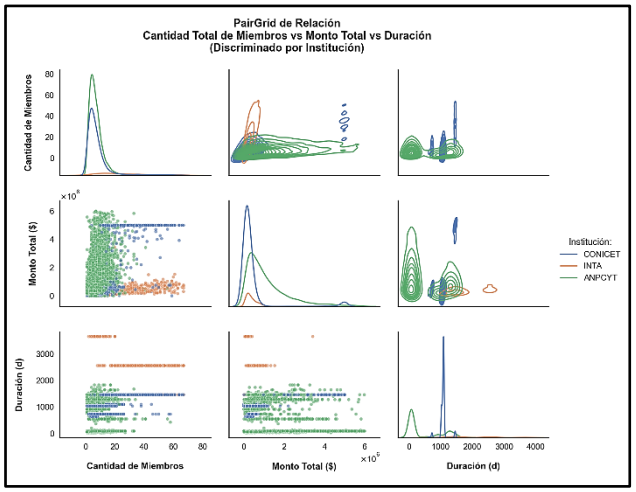

Figura 11 – Pair Grid de Principales Variables Numéricas

Esto puede ser un problema, ya que puede ser que la cantidad de registros de esta institución no sean suficientes como para que nuestro modelo de Machine Learning pueda identificar los patrones en ellos.

Luego de analizar variables categóricas, se hizo foco en las variables numéricas, considerando como principales a 'cantidad_miembros_total', 'monto_total_adjudicado' y 'duracion_dias'. Para esto se confeccionó el Pair Grid de la **Figura** 11, en el cual se pueden visualizar KDE Plots (Kernel Density Estimate), como Scatter Plots, con el objetivo de tratar de mejorar la comprensión sobre cómo se distribuyen, y posiblemente agrupan, los datos del data set en función de las variables antes dichas.

De acuerdo a lo comentado respecto a la **Figura 9**, en los gráficos de distribución no llegan a apreciarse las curvas correspondientes al INTA, ya que esta institución tiene una cantidad muy reducida de registros, en comparación a las otras dos.

Del gráfico de Monto Total vs Cantidad de Miembros puede observarse que muchos proyectos de CONICET tienen un mismo importe de monto total, a pesar de que presentan una cantidad de miembros que se distribuye a lo largo de todo el rango. Sería interesante poder averiguar un poco más al respecto, ya que esto no parece ser un fenómeno normal. Por otro lado, puede observarse que los proyectos del INTA presentan una variedad de monto total acotada a un rango bien definido, el cual es bajo, mientras que, por el contrario, la cantidad de miembros de estos proyectos se encuentra por encima del rango que abarca al grueso de los datos. Por último, puede comentarse que los proyectos del ANPCYT presentan un rango acotado de cantidad de miembros, pero un amplio rango de monto total en sus proyectos.

En lo que respecta a la duración, puede concluirse que los proyectos del INTA son lo que mayor duración presentan, mientras que una parte de los de ANPCYT son los de menor duración.

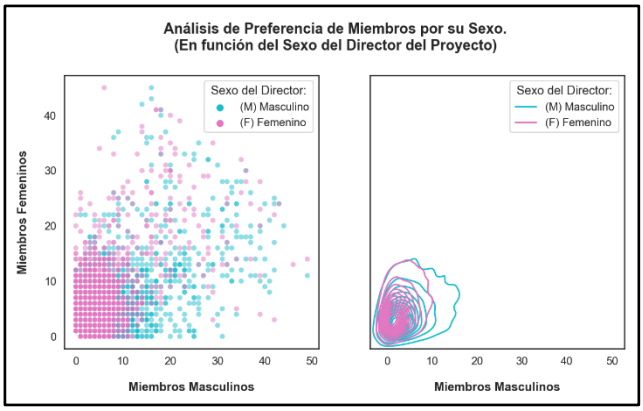

Figura 12 – Preferencia de Miembros según Sexo del Director

Por último, se realizó un gráfico para tratar de comprender si el sexo del director del proyecto influía en la cantidad de miembros de cada sexo (**Figura 12**). Es decir, lo que se buscó fue determinar si los directores de los proyectos mostraban cierto favoritismo por un sexo o el otro. En rasgos generales, la cantidad de miembros masculinos y femeninos parece estar bien distribuida. Cómo único comentario puede decirse que aquellos proyectos cuyo director es de sexo femenino parecen tener una tendencia de mantener una menor proporción de miembros masculinos respecto de los femeninos; mientras que en aquellos proyectos en que su director es de sexo masculino, para una cantidad de miembros total similar, parecen mantener una proporción más homogénea de miembros de ambos sexos.

3 MODELOS DE MACHINE LEARNING

El Pipe Line desarrollado en esta etapa puede verse en la **Figura 13**. En el mismo se detallan todos los pasos que se siguieron, partiendo del data set pre-procesado, obtenido como resultado del EDA, hasta las métricas de performance de los distintos modelos empleados, obtenidas en la fase de train y de test.

El primer paso se trató de descomponer el data set pre-procesado en dos datasets independientes, a los que se llamó *Features* y *Targets*. El primero contaba con todas las variables que utilizarían los modelos para predecir, y el segundo se trata de la variable objetivo, es decir, el resultado que se quiere obtener.

Debido a que el data set contaba con múltiples variables categóricas, fue necesario generar dummies. Es decir, se generó una columna por cada valor único que pueden tomar estas variables, y se aplicó a cada registro el valor de *1* si pertenecía a esa variable, o de *0* si no es así.

Una vez que se cuenta con los data sets a utilizar, se debe convertir los mismos a X (features) e Y (target), quedando entonces únicamente los valores de los registros.

Luego, tanto X como Y deben ser fraccionados en sets de entrenamiento y de prueba. Para esto se utiliza la función '*train_test_split*', obteniendo así 'x_train', 'x_test', 'y_train' e 'y_test'. Se utilizó un '*test_size*' de 0.25.

Debido a que las variables numéricas presentaban una gran diferencia entre sus valores (**Figura 14**), fue necesario escalar los mismos. Para esto se utilizó un Standard Scaler, consiguiendo que el promedio de los valores de cada variable sea 0, y su desvío standard sea 1. La manera de lograr esto es la siguiente:

$$x_i' = \frac{x_i - \mu}{\sigma}$$

Lo que se hace es, a cada registro, restarle la media de todos los registros de dicha variable, y dividir por el desvío estándar. Es importante aclarar que la media y el desvío estándar a utilizar son los del set de entrenamiento.

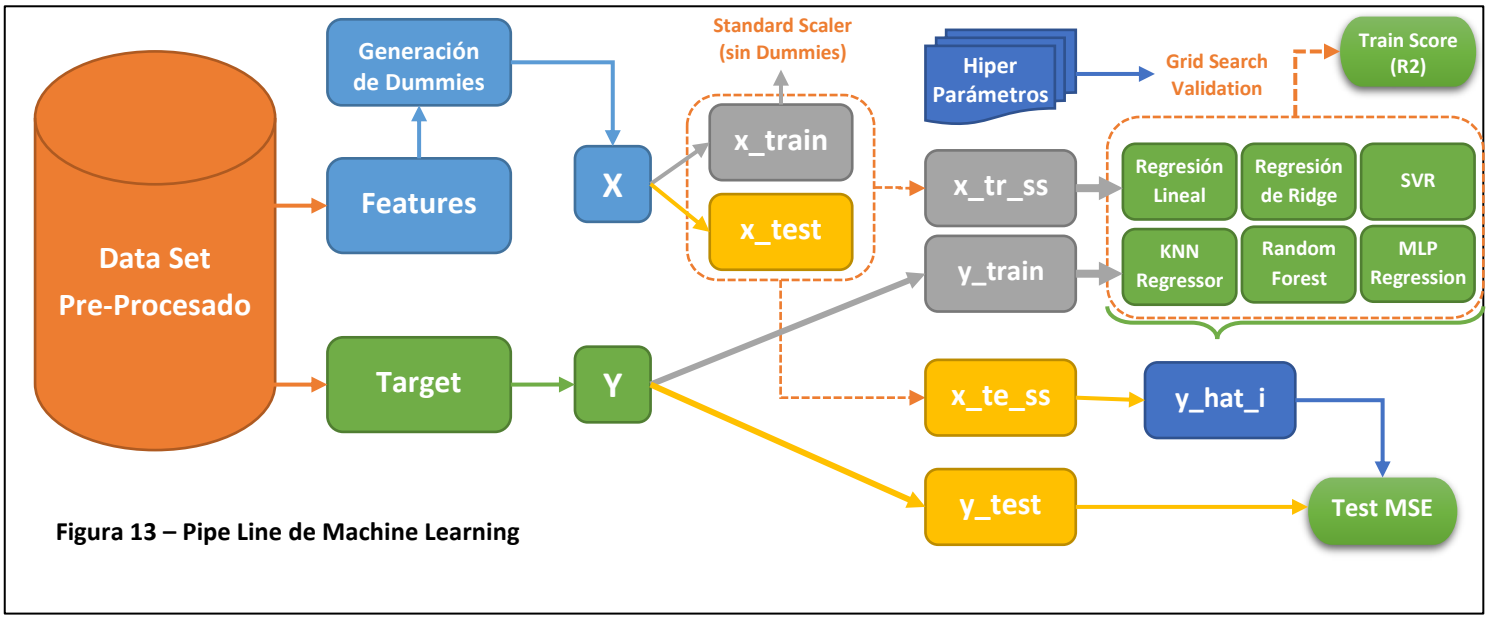

Figura 13 – Pipe Line de Machine Learning

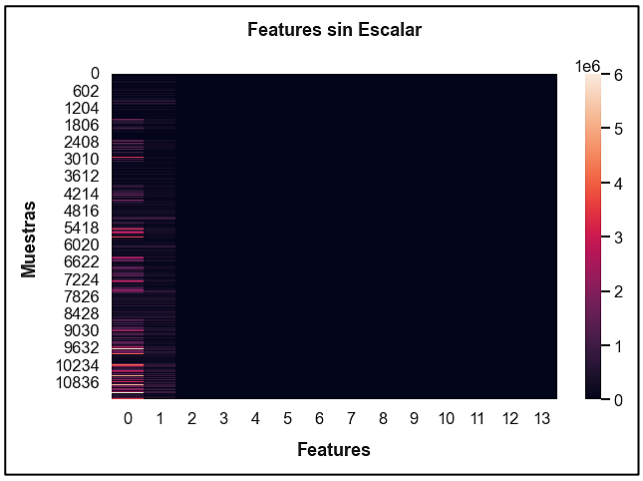

Figura 14 –Heat Map de Visualización de Features sin Escalar

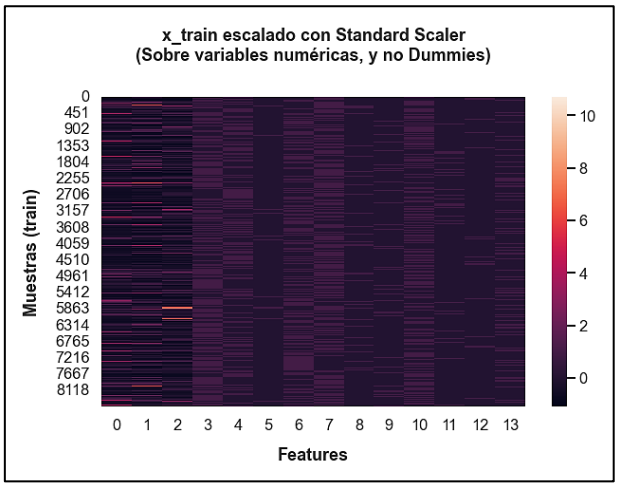

Figura 15 – Heat Map de Visualización de Features Escaladas con Standard Scaler

Esta transformación, con la media y el desvío estándar del set de entrenamiento, se aplica tanto al set de entrenamiento como al de prueba, y se los renombró como 'x_tr_ss' y 'x_te_ss', respectivamente. El resultado del escalado de datos puede verse en la **Figura 15**.

Finalmente, se trabaja con los distintos modelos de regresión propuestos, los cuales son explicados a continuación.

3.1 Modelos de Regresión

Como primera aproximación a la problemática a resolver se trabajó con *modelos lineales*.

Dentro de los *modelos lineales* planteados se encuentran:

Regresión Líneal Simple[2]: Se trata de un modelo lineal con coeficientes $w = (w_1, \dots, w_p)$, para minimizar la suma de los residuos al cuadrado. Recordemos que se entiende por *residuos* a la diferencia entre los valores de target reales y los predichos por el modelo. Los parámetros validados en Grid Search fueron 'fil_intercept' y 'copy_X'.

Regresión de Ridge[3]: Se trata de una Regresión Lineal de Mínimos Cuadrados con una Regularización L2. Este modelo resuelve un modelo de regresión donde la función de pérdida es la función de mínimos cuadrados lineales y la regularización viene dada por la norma L2. También es conocida como Regularización de Tikhonov. Los parámetros validados en Grid Search fueron 'alpha', 'fit_intercept', 'copy_X', 'solver' y 'positive'.

Regresión de Lasso[3]: Se trata de un modelo lineal entrenado con prior L1 como regularizador. Técnicamente, el modelo Lasso está optimizando la misma función objetivo que Elastic Net cuando se utiliza sin penalización L2. Los parámetros validados en Grid Search fueron 'alpha', 'fit_intercept', 'copy_X' y 'positive'.

Regresión Lineal con Regularización Elastic Net: Regresión Lineal con priors L1 y L2 combinados como Regularizadores. Los parámetros validados en Grid Search fueron 'alpha', 'fit_intercept', 'copy_X' y 'positive'.

Con esto se completa el listado de los *modelos lineales* utilizados. A continuación, se comenta el resto de los modelos utilizados.

KNN Regressor: Se trata de una regresión basada en k-vecinos más cercanos. La variable objetivo se predice mediante la interpolación local de los targets asociados a los vecinos más cercanos en el conjunto de entrenamiento. Los parámetros validados en Grid Search fueron 'n_neighbors', 'weights', 'algorithm' y 'p'.

Suport Vector Regression (SVR): A Se trata de una regresión vectorial con soporte de épsilon. Los parámetros libres en el modelo son C y épsilon. La complejidad del tiempo de ajuste es más que cuadrática con el número de muestras, lo que dificulta escalar a conjuntos de datos con más de 10.000 muestras. Los parámetros validados en Grid Search fueron 'kernel', 'C' y 'gamma'.

Random Forest Regression: Es un meta-estimador que se ajusta a una serie de árboles de decisión de clasificación en varias sub-muestras del conjunto de datos y utiliza promedios para mejorar la precisión predictiva y controlar el sobreajuste. Los parámetros validados en Grid Search fueron 'n_estimators', 'criterion', 'max_features' y 'min_samples_leaf'.

Multi-Layer Perceptron Regressor (MLP)[4]: Este modelo optimiza el error al cuadráticoo usando LBFGS (método de optimización quasi-Newton de funciones con un gran número de parámetros o de una gran complejidad) o descenso de gradiente estocástico. El Multi-Layer Perceptron Regressor entrena iterativamente ya que en el paso de cada época se calculan las derivadas parciales de la función de pérdida con respecto a los parámetros

del modelo para actualizar los parámetros. También se puede agregar un término de regularización a la función de pérdida que reduce los parámetros del modelo para evitar el sobreajuste. Los parámetros validados en Grid Search fueron 'hidden_layer_sizes', 'activation', 'solver', 'batch_size', 'learning_rate_init' y 'early_stopping'.

Es importante comentar que en todos los casos se utilizó Grid Search Cross Validation para la selección de Hyper-Parámetros durante el entrenamiento. Luego haber terminado dicha etapa, se obtenían las métricas correspondientes a los arreglos de Hiper-Parámetros ganadores. La métrica de train utilizada por Scikit-Learn es R2.

3.2 Resultados de Métricas

Una vez entrenados los modelos, y obtenidas las métricas de entrenamiento (R2), se procedió a estimar los valores de la variable objetivo (y) con los registros de test. Así es como se obtiene lo que se llamó 'yte_hat_(modelo)'.

Al comparar entre el valor real de la variable objetivo y el valor estimado por cada modelo, se consigue calcular el MSE (Mean Square Error) sobre el set de prueba.

Finalmente, lo que se hizo fue graficar, para cada modelo, el valor real de y frente al valor estimado por el modelo. Esto puede observarse en la **Figura 16**, y la línea roja representa el caso ideal, en que el valor estimado se corresponde con el valor real.

4 CONCLUSIONES

Para complementar los gráficos de la **Figura 16**, se generó la siguiente tabla:

	Train Score (R2)	Test MSE
Linear Regression	0.461494	167040.0
Ridge Regression	0.461409	236711.0
Lasso Regression	0.461444	248336.0
Elastic Regression	0.461396	297722.0
KNN Regression	0.541587	144424.0
SVR Regression	0.387586	243420.0
Random Forest Regression	0.637317	381204.0
Multi-Layer Percep. Regression	0.558808	227833.0

Figura 17 – Tabla Comparativa de Métricas de Rendimiento de Modelos de Regresión

Como puede verse en la tabla, el modelo que presenta el mejor score en train es Random Forest, pero es también es el que presenta el mayor valor de MSE durante el test. Esto lo podemos interpretar como que el modelo está sobre-ajustando, por lo que habría que probar con otra parametrización para tratar de regularizar.

Por el contrario, el modelo que presenta el mejor MSE en test es el KNN Regressor. Este modelo también presenta un score de los mejores durante el train. Puede decirse que, si bien el modelo dista de ser bueno ya que presenta un MSE considerable, parece generalizar lo bastante bien.

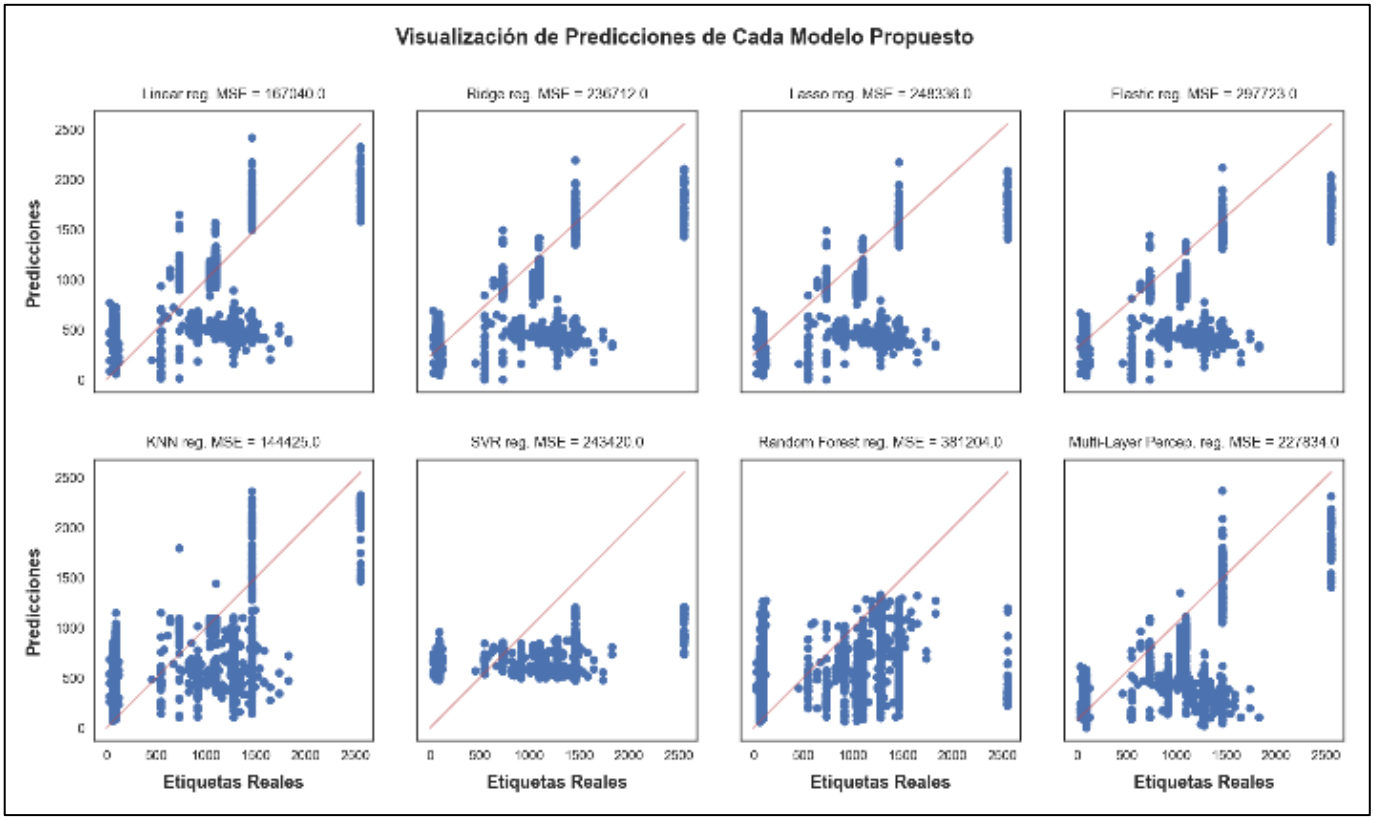

Figura 16 – Visualización de Predicciones de los Diferentes Modelos

Por último, es interesante comentar los resultados del modelo de Regresión Lineal, ya que resultó ser el mejor modelo dentro de los lineales en la fase de train, y también consiguió un MSE de los mejores en la fase de test. Puede decirse que se trata de un modelo simple, con un costo computacional relativamente bajo, que presenta resultados aceptables para nuestro caso.

Como es sabido, los modelos de regresión no suelen dar tan buenos resultados como los de clasificación. Tal y como lo muestran las ***Figuras 16*** *y* ***17****, el caso en estudio no es la excepción. Ninguno de los modelos ha conseguido generar una predicción que esté alineada con la recta de correspondencia con los valores reales.*

Esto se puede deber, o bien a que el caso en estudio es demasiado complejo para los modelos propuestos, o no contamos con suficientes variables / datos informativos para conseguir un mejor resultado.

De tener que optar por uno de los modelos propuestos más arriba, nos quedaremos con KNN Regressor, pero lo ideal sería continuar trabajando con los datos y proponer un modelo que pueda abordar mejor el problema, como puede serlo quizás una Red Neuronal con mayor configuración de capas, neuronas, funciones activadoras, entre tantos otros parámetros a configurar.

5 REFERENCIAS

[1] Krishna, M., 2007, "Selecting the Appropriate Outlier Treatment for Common Industry Applications", pp. 1-2, Inductis Inc.

[2] VanderPlas, J, 2017, "Python Data Science Handbook", pp. 347-350, O' Reilly.

[3] Tibshirani, R.; James, G.; Witten, D.; Hastie, T., 2013, "An Introduction to Statistical Learning", pp. 215-228, Springer.

[4] Kingma, D.; Ba, J., 2014, "Adam: A method for stochastic optimization.", arXiv preprint.