**Reporte de selección y parametrización de modelos**

Mariana Arias

Alejandro Murcia Pinilla

Juan Camilo Pérez

Johan Santacruz

**Mayo 2023**

**Contenido**

1. Implementación de los modelos
   1. Tratamiento previo de los datos y verificación de supuestos
   2. Análisis Univariado y Bivariado
   3. selección de conjuntos de datos y métricas de desempeño
   4. Validación y ajuste del proceso experimental
2. Ajustes de Hiperparámetros
3. Evaluación de completitud de la solución
   1. Selección de modelo (variables, métricas y parámetros)
4. Plan de implementación del prototipo entregable
   1. Trabajo Validado y listo para el prototipo
      1. Modelo de datos y primera versión de la ETL
      2. Pruebas sobre el modelo de datos
      3. Primer borrador del tablero
   2. Pendientes por implementar
      1. Alimentación de los resultados del pronóstico al modelo en estrella
      2. Elaboración de encuestas y recolección de resultados.
      3. Construcción de presentación final y pitch
5. Bibliografía

Todos los procesos y resultados obtenidos de los puntos 1, 2 y 3 pueden ser consultados a detalle usando un notebook del [repositorio.](https://github.com/jsantacruzl/ProyectoDeGradoMIAD_o3b/blob/main/ETL/Proyecto%20Final%20-%20Modelos%20de%20ML.ipynb) No es necesario revisar el código para la comprensión de los resultados obtenidos, pero es recomendable de realizar si se desean obtener mayores detalles técnicos.

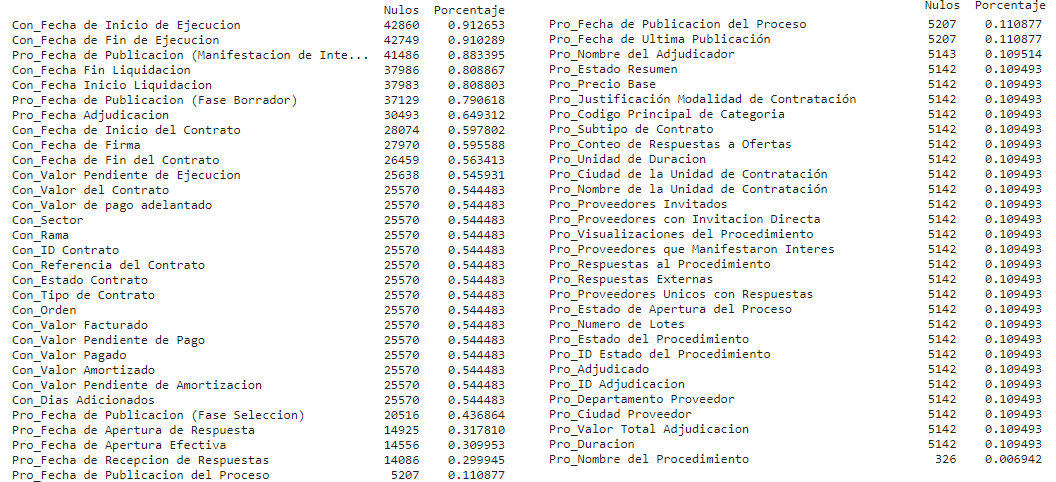
**1. Implementación de los modelos**

1. Tratamiento previo de datos y verificación de supuestos

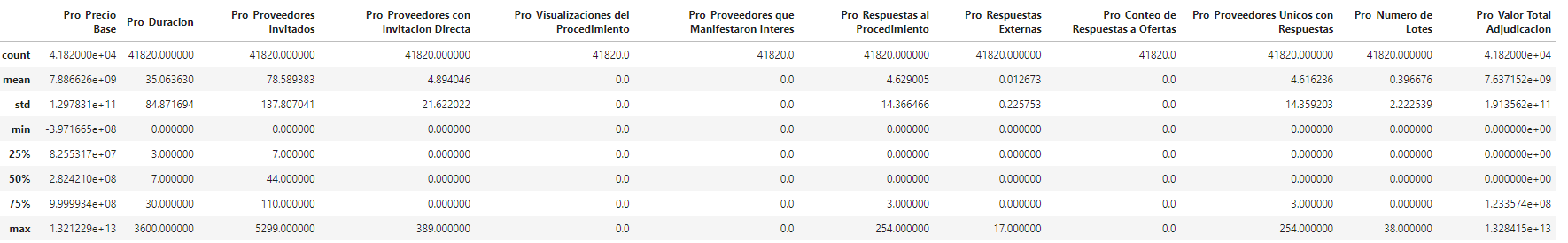
Todo el trabajo inicia desde la sabana de datos creada por el equipo, en la cual se juntó la información de los contratos públicos de obras para Colombia, incluyendo una marcación para aquellos proyectos que recibieron una multa. Todos estos datos fueron almacenados dentro de la base de datos diseñada y recreada por el equipo a partir de la información disponible en la página del SECOP.

Para realizar el modelo de clasificación de la base de datos del SECOP se definió como la variable dependiente a aquellos contratos que resultaron en una multa, lo cual derivó de una serie de análisis previos para el entendimiento de los datos.

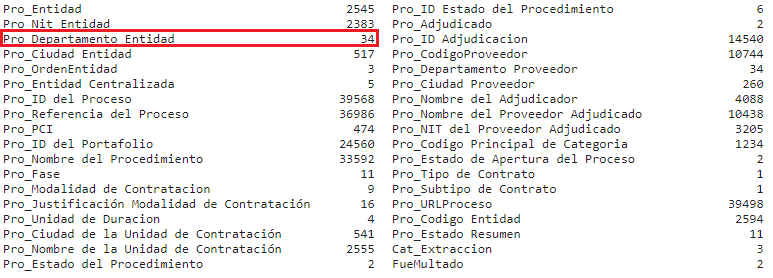
En primer lugar, se llevó a cabo un análisis de los valores nulos presentes en los datos, con el objetivo de quedarnos únicamente con las variables que tenían un nivel de completitud mayor al 88%. De las 88 columnas iniciales nos quedamos con 50 que cumplían el porcentaje de completitud deseado. (El origen de los datos es una tabla llamada *SabanaAnalitica* que combina las 3 fuentes de datos usadas en el proyecto y su construcción será detallada más adelante).

***Fig. 1.*** *Análisis de completitud de datos. La tolerancia aceptada fue del 12%.*

A continuación, se realizó un análisis descriptivo y estadístico de las variables numéricas, observando que algunas de ellas presentaban valores máximos que superaban el percentil 75%, lo cual podría indicar la presencia de outliers. Más adelante se abordará este tema con más detalle.

***Fig. 2.*** *Análisis estadístico de las variables. En este se puede ver que variables como Precio Base puede tener la presencia de atípicos, ya que el valor máximo se aleja bastante del percentil 75%.*

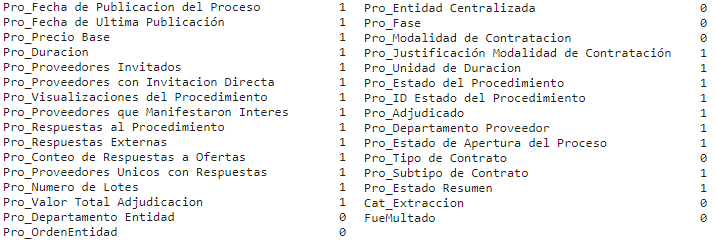
En cuanto al análisis de las variables categóricas, se estableció un límite de 34 categorías para seleccionar las variables. Esta cantidad corresponde al número de departamentos que hay en la base. De las 88 variables disponibles, aplicando estas condiciones, quedaron 31 variables candidatas para usar en el modelo.



***Fig. 3.*** *Análisis variables Categóricas. Se tomó como referencia del número máximo de categorías, las de la variable departamento.*

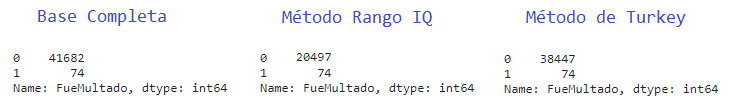
Dado que la variable FueMultado es nuestra variable de respuesta, se analizaron las categorías que tenía internamente, dejándola como una variable numérica con valores 0 y 1, donde 0 indicaba que el contrato no tenía multas y 1 que sí las tenía.

Se llevó a cabo un tratamiento para los registros nulos, donde se eliminaron aquellos registros nulos que no contenían la categoría 1 en la variable de respuesta (pues es la que indica la multa).



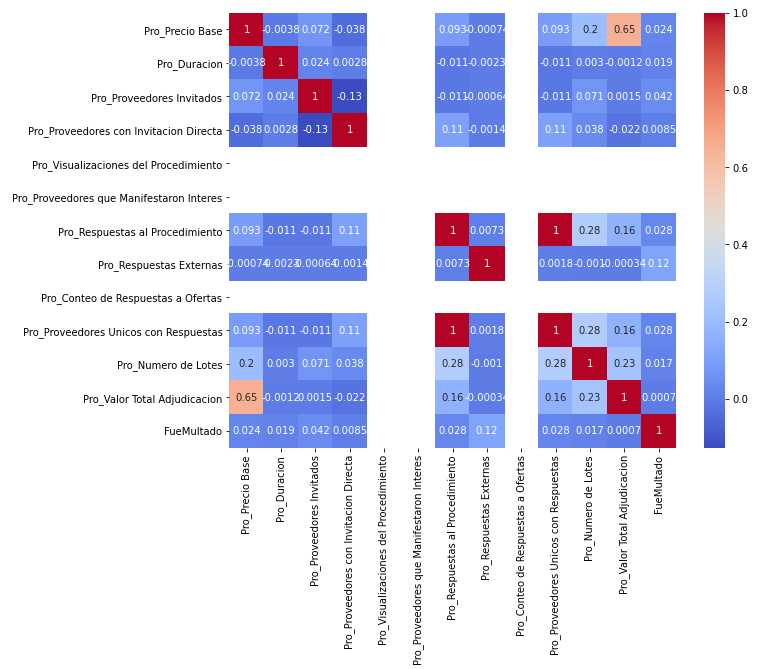
***Fig. 4.*** *Análisis de nulos de las variables seleccionadas luego de haber hecho la limpieza.*

También se trabajó en la detección y tratamiento de valores atípicos o outliers, probando dos métodos diferentes: el método del rango intercuartílico (IQR) y el método de los límites de Tukey. Finalmente, se seleccionó el método de los límites de Tukey ya que eliminaba menos registros del total de la base.



***Fig. 5.*** *Eliminación de* *Outliers. EL método de* *Turkey elimina menos registros que el método del Rango IQR.*

Por último, se realizó un análisis de correlación entre las variables numéricas seleccionadas, concluyendo que presentaban una baja correlación entre ellas, por lo cual se decidió dejar todas las variables numéricas para utilizar en el modelo.

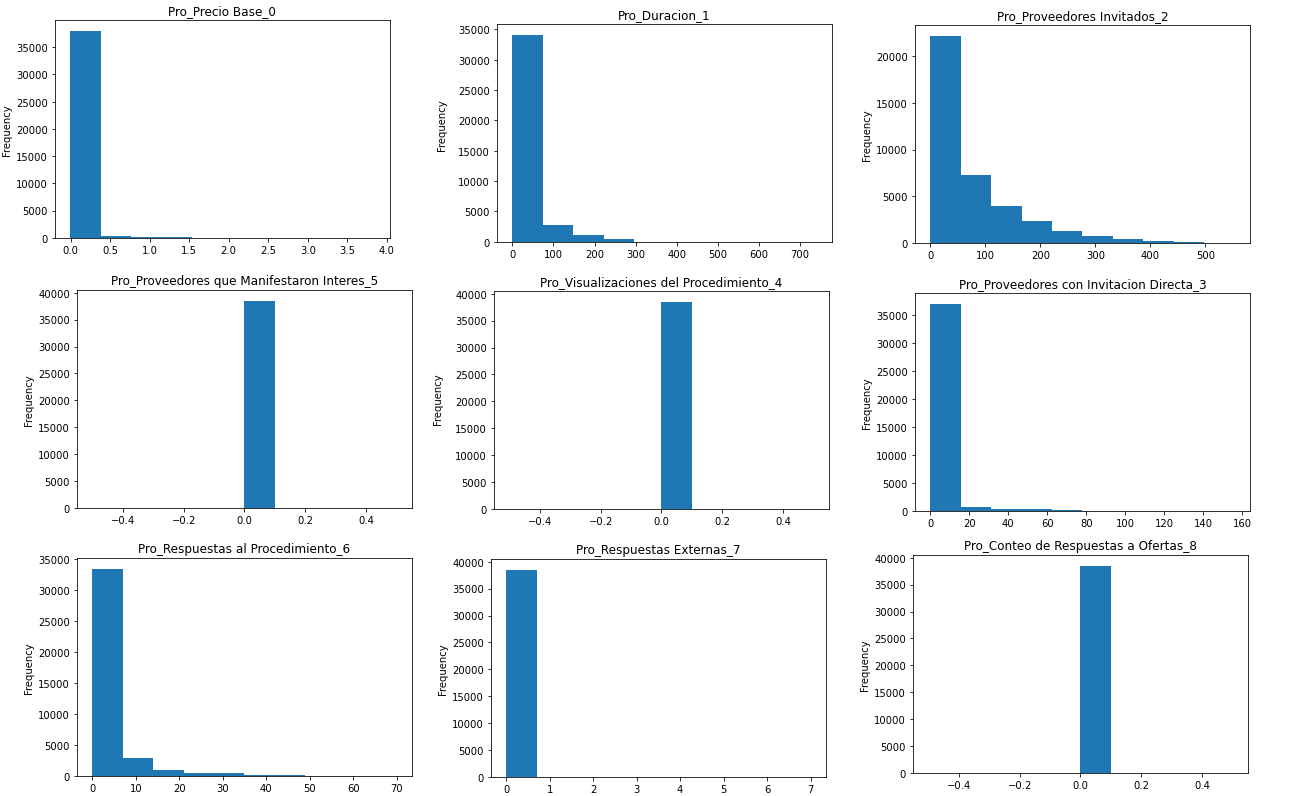


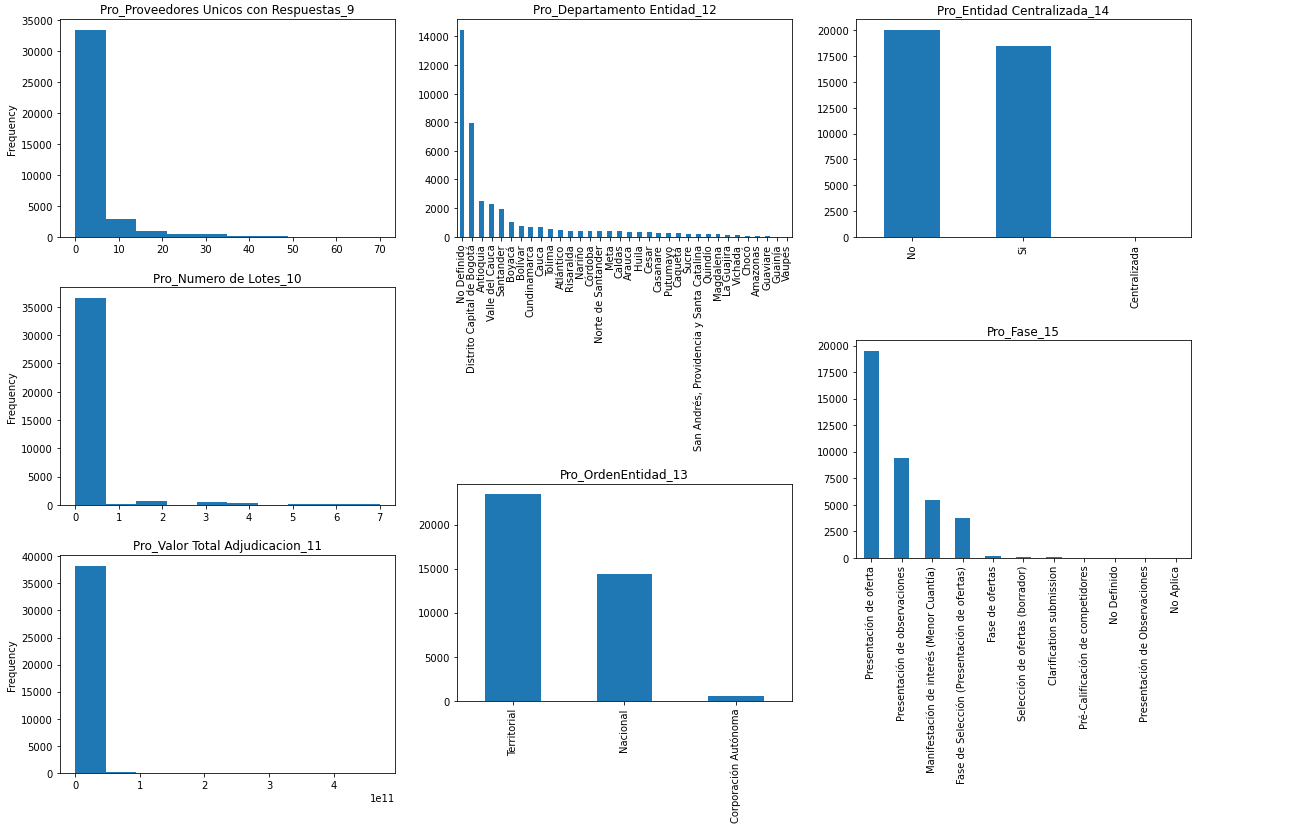
***Fig. 6.*** *Análisis de correlaciones de variables numéricas*

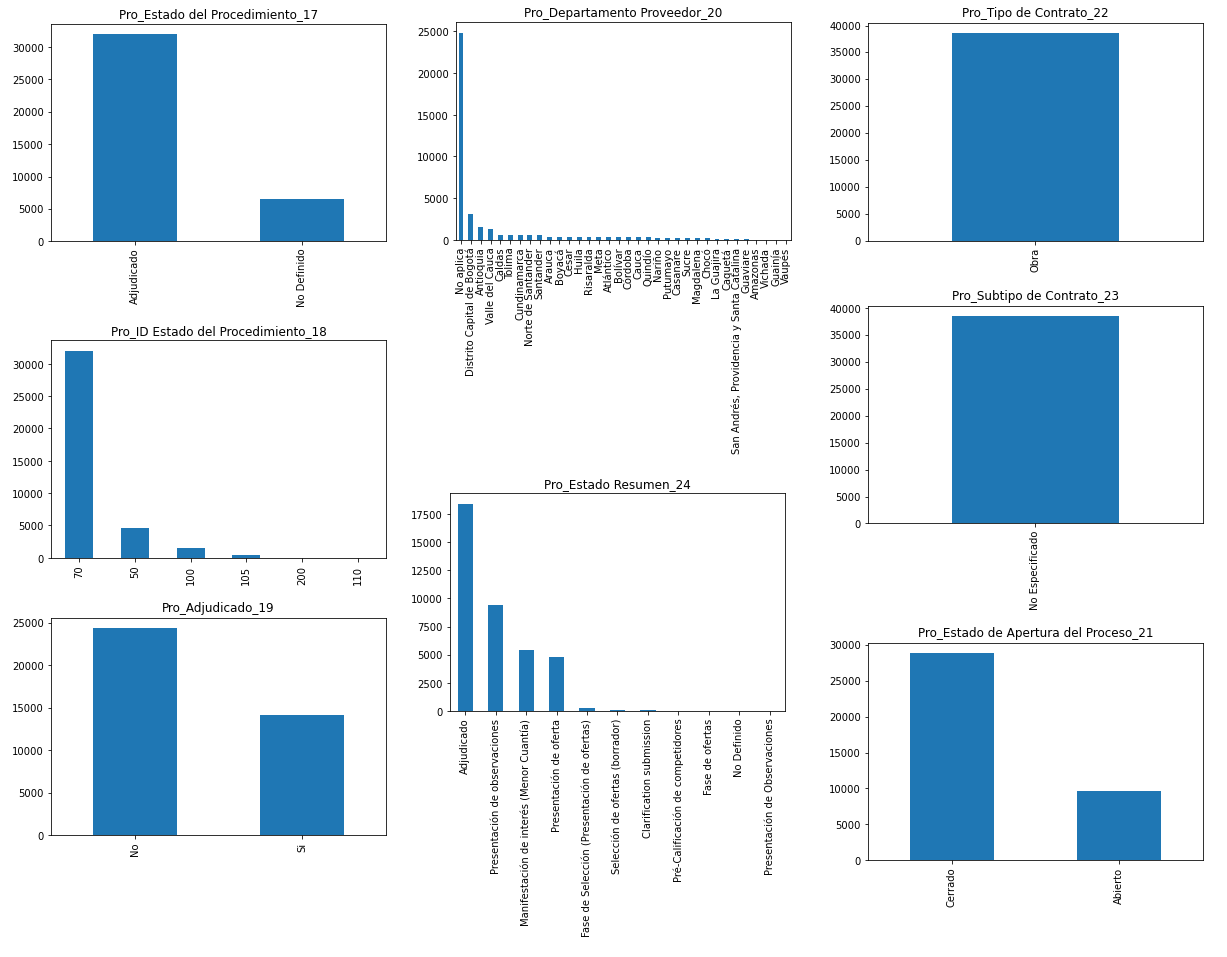
1. **Análisis Univariado y Bivariado**

**Análisis Univariado**

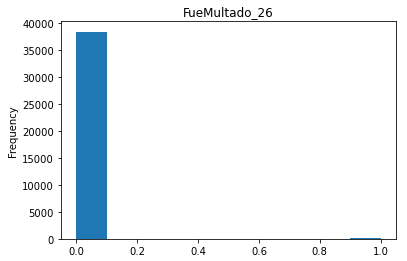
***Variables independientes***





***Fig. 7.*** *Distribución de variables independientes dentro del conjunto de datos.*

**Variable Respuesta**



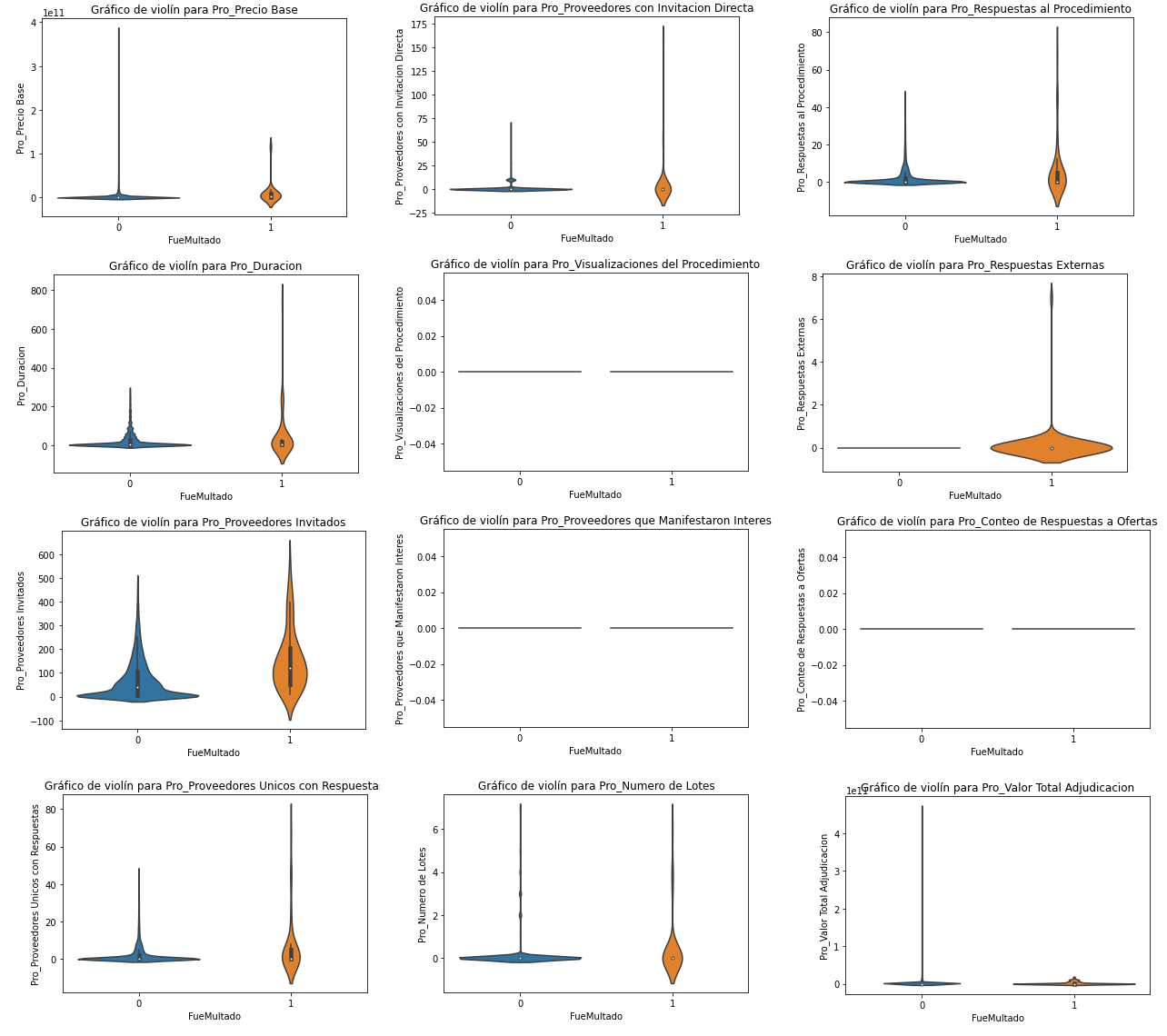
***Fig. 8.*** *Distribución de la variable de respuesta, con evidente desbalanceo de clases.*

Hay bastantes cosas importantes para resaltar dentro del análisis univariado, entre los cuales, lo primero a mencionar es el evidente desbalanceo que existe entre las clases, ya que contamos con un total aproximado de 40.000 proyectos sin multar, comparado contra un total de 75 proyectos multados, lo cual nos generó en su momento la duda con respecto a si se debería aplicar alguna técnica para rebalanceo de clases o no.

Adicionalmente, es posible identificar que hay variables categóricas que solamente representan una categoría, como en este caso sería el tipo y el subtipo de contrato, por lo cual aplicar esta variable dentro de un modelo no tendría mayor relevancia a la hora de explicar los proyectos multados.

Finalmente, nos permitió identificar que existen variables numéricas dentro de los datos que tienen comportamientos que se asemejan a una distribución logarítmica, razón por la cual requieren de la aplicación de ciertos tratamientos y transformaciones especiales si quisiéramos hacer uso de estas en modelos lineales.

**Análisis Bivariado**



***Fig. 9.*** *Análisis bivariados de las variables independientes.*

Dentro del análisis bivariado encontramos cosas bastante interesantes y que tal vez no eran tan deducibles antes de hacer el ejercicio, entre las cuales podeos observar que la variable de proveedores invitados pareciera tener una incidencia bastante marcada con la segmentación entre los proyectos multados y aquellos que no. Esto se puede evidenciar en que la distribución de los proyectos multados se extiende un poco más hacia los proyectos con mayor cantidad de proveedores invitados, mientras que los no multados tienen la mayoría de su población dentro de niveles bajos para dicha variable.

Adicionalmente, podemos observar que, el valor total adjudicado no siempre se traduce en mayor probabilidad de que el proyecto sea multado, ya que puede ser una intuición previa al entendimiento de los datos el llegar a pensar que entre más costoso es el proyecto corre mayor riesgo de presentar fallas que conlleven a una multa; por el contrario, vemos entonces que esta segmentación no es tan clara al hablar únicamente de la relación entre estas dos variables.

En pocas palabras, este análisis nos permitió darle un entendimiento previo a que aún sin la combinación entre las variables, si existe algún tipo de relación entre las variables dependientes y la variable dependiente.

1. **Selección de modelos y métricas de desempeño**

**Modelos**

Se probaron diferentes algoritmos con los datos existentes, de los cuales destacaron dos por los resultados obtenidos, así como por su flexibilidad para poder ser calibrados y obtener información relevante extra para el negocio, los cuales fueron:

***Random Forest***

Es ideal para este problema de negocio ya que en el análisis exploratorio de los datos es evidente la no linealidad de los mismos respecto a la variable de interés. Por ello, un model-based como RF, además de contar con un mejor performance y escalabilidad comparado a otros modelos de ensamble como XGBoost, lo hicieron un candidato muy importante para ser tenido en cuenta en el contexto del presente proyecto.

Otra característica importante es que, debido a que ofrece interpretación de las variables más relevantes del modelo construido, dicha información podría ser también utilizada para ayudarnos a identificar campos importantes que se deben incluir tanto en el modelo en estrella como en el tablero (ya sea al usar dichos campos como filtros o para soportar la explicación de los pronósticos realizados de una manera gráfica, como las categorías de un pie chart).

***Redes Neuronales***

A pesar de no contar con un dataset de datos muy grande, decidimos utilizar RN ya que suelen ofrecer resultados muy buenos en conjuntos de datos donde no hay linealidad y al buen balance en la dicotomía sesgo-varianza usualmente alcanzado con este tipo de modelos.

**Métricas**

Existen varias métricas importantes para evaluar la calidad de un modelo de clasificación que se generan a partir de la matriz de confusión. Esta es una tabla que muestra la cantidad de verdaderos positivos, falsos positivos, verdaderos negativos y falsos negativos. Con esta matriz se pueden calcular diversas métricas, como la precisión (accuracy), sensibilidad (recall), especificidad (specificity) y el F1-Score. Estas métricas son importantes porque permiten evaluar el rendimiento del modelo en términos de su capacidad para identificar correctamente los casos positivos y negativos.

La **precisión** (accuracy) mide la proporción de observaciones clasificadas correctamente como positivas entre todas las observaciones clasificadas como verdaderos positivas y falsos negativos. Para este caso, donde lo que se quiere mirar la posibilidad que un contrato resulte en multa, la precisión es importante porque mide la proporción de detecciones de multas que son correctas. Si el modelo tiene una precisión alta, entonces la mayoría de las detecciones de multas que hace son correctas.

Por su parte, la **sensibilidad** (recall) mide la proporción de observaciones positivas clasificadas correctamente como positivas sobre el total de verdaderos positivos más falsos negativos. Esta es importante porque mide la proporción de casos de multas que son detectados por el modelo. Si el modelo tiene una sensibilidad alta, entonces detecta la mayoría de los casos de multas.

El **F1 Score** combina la precisión y la sensibilidad en una sola medida. Es para este tipo de casos donde el modelo esta desbalanceado porque tiene en cuenta tanto los verdaderos positivos como los falsos positivos. El valor F1 proporciona una medida única de la calidad general del modelo mediante la siguiente relación F1 = 2 \* (precisión \* recall) / (precisión + recall).

En cuanto a la **curva ROC** y el área bajo la curva (**AUC**), son útiles para evaluar el rendimiento general del modelo en diferentes puntos de corte de probabilidad. La curva ROC traza la tasa de verdaderos positivos (sensibilidad) en función de la tasa de falsos positivos (1 - especificidad). El AUC mide la probabilidad de que el modelo clasifique una observación positiva elegida al azar más alta que una observación negativa elegida al azar.

En este punto es importante mencionar que, debido a las necesidades del negocio propuestas en el desarrollo del proyecto, nuestra intención era conseguir un modelo que pudiera ofrecer una alta sensibilidad, es decir, que lograr marcar como multados a la mayoría de los proyectos que realmente habían sido multados. De esta forma, se hizo uso de la precisión como segunda métrica de selección del modelo, debido a la importancia de que aquello que se clasificara como multado realmente fuera multado.

1. **Validación y ajuste del proceso experimental**

Antes de empezar a probar modelos Se realizaron algunos ajustes a los datos para que estos se pudieran trabajar con los modelos. Entre los ajustes realizados se destacan:

* Convertir las variables categóricas en dummies y eliminar una columna para evitar problemas de multicolinealidad.
* Dividir el conjunto de datos en conjuntos de entrenamiento y prueba.
* Realizar un rebalanceo de la variable de respuesta para solucionar el problema de desbalanceo. Se probaron técnicas de sobre muestreo (over sampling) y de sub muestreo (under sampling).
* Realizar una normalización de las variables numéricas, principalmente para aquellas en las cuales son campos de dinero, los cuales puedes estar expresados en millones de pesos. En general se aplicó a todas las variables numéricas.
* Realizar una reducción de dimensionalidad mediante el análisis de componentes principales PCA.
* Seleccionar las características más importantes mediante un modelo de selección de características (por ejemplo, un modelo de regresión logística con regularización L1)
* Entrenamiento de distintos modelos de clasificación (Random Forest y Redes neuronales con y sin RandomGridSearchCV.

**2. Ajustes de Hiperparámetros**

Como se comentó, entre los modelos que se probaron están los de Random Forest y Redes Neuronales, los cuales se combinaron con técnicas de sobremuestreo o submuestreo y con calibración de hiperparámetros con Random Grid Search CV.

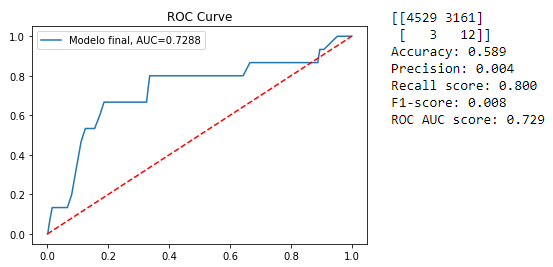
Dicho esto, se recrearon 15 modelos aleatorios con una validación cruzada de 3KFolds, haciendo una búsqueda aleatoria entre diferentes hiperparámetros que según la teoría son bastante influyentes en los resultados de los modelos que se presentan a continuación;

* Random Forest Classifier con datos luego de aplicar Over Sampling.
* Random Forest Classifier con datos luego de aplicar Under Sampling.
* Redes Neuronales aplicando pesos diferentes a las clases como método de balanceo.

De esta forma, al final de las corridas nos quedamos con un total de 45 modelos posibles, para los cuales en cada uno se contaba con el promedio y la desviación de los 3 KFolds en cada una de las métricas especificadas en el documento (incluyendo el ROC, la sensibilidad y la precisión). Todos estos datos se encuentran en el archivo de Excel adjunto, el cual permite obtener los resultados individuales y la comparación.

Uno de los ejemplos de lo que podíamos esperar al final del ejercicio se presenta a continuación;

* **Random Forest Classifier & Under Sampling**



***Fig. 10.*** *Resultados de RF con Undersampling.*

**3. Selección de modelo final (variables, métricas y parámetros)**

Ya con todos los resultados de los 45 modelos, lo que se hizo fue ordenarlos haciendo uso de la métrica de sensibilidad, dejando entonces un top 5 de modelos tal y como se muestra en la siguiente tabla;

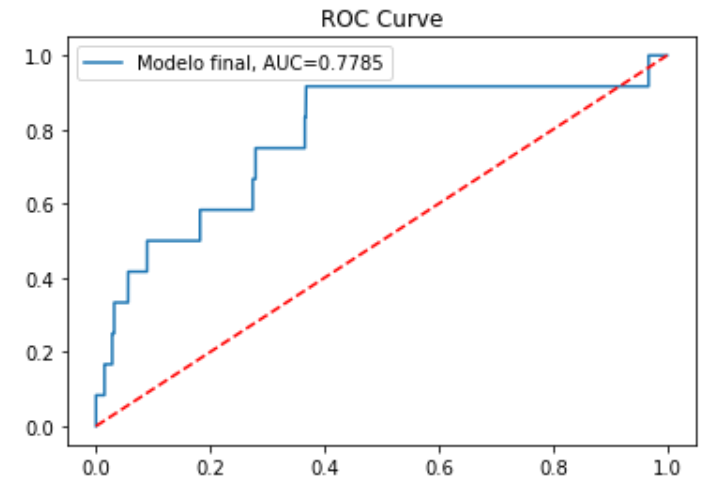
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Indice Modelo** | **mean\_test\_accuracy** | **std\_test\_accuracy** | **mean\_test\_precision** | **std\_test\_precision** | **mean\_test\_recall** | **std\_test\_recall** |
| RFC\_OS\_8 | 0.894 | 0.065 | 0.654 | 0.458 | 0.848 | 0.143 |
| RFC\_OS\_6 | 0.992 | 0.008 | 0.684 | 0.442 | 0.843 | 0.221 |
| RFC\_OS\_12 | 0.992 | 0.008 | 0.684 | 0.443 | 0.843 | 0.221 |
| RFC\_OS\_3 | 0.889 | 0.066 | 0.654 | 0.458 | 0.843 | 0.141 |
| RFC\_OS\_5 | 0.889 | 0.066 | 0.654 | 0.458 | 0.843 | 0.141 |

De esta forma, tal y como habíamos venido mencionando desde la sección de métricas, la idea es entonces seleccionar el modelo que mejor sensibilidad presente, ya que este nos permitirá capturar la mayor cantidad posible de proyectos multados dentro del ejercicio.

Por lo tanto, se llega a la conclusión de que el mejor modelo dentro de los 45 entrenados para nuestro contexto específico es un Random Forest con técnica de Over Sampling para rebalanceo de clases y presentando los siguientes parámetros;

* Numero de estimadores: 100
* Mínimo de proyectos para un split: 2
* Máxima profundidad en cada estimador: 5

Entrenando este modelo y haciendo la proyección sobre los datos de testeo, encontramos la siguiente curva ROC como resultado;



***Fig. 11.*** *Curva ROC modelo seleccionado.*

En conclusión, encontramos un modelo con una curva ROC de 0.78, para el cual implementamos una respuesta a modo de semáforo, ya que se identificaron dos puntos de corte en el threshold que nos permiten sacar conclusiones muy interesantes.

Lo anterior significa que para aquellos proyectos que el modelo genere una calificación por encima de 0.5 serán considerados como proyectos Rojos, y representan aquellos que probablemente sean multados. Por otro lado, cuando el puntaje está dentro de 0.2 hasta 0.5, la clasificación del proyecto será de Amarillo, lo cual representa que probablemente este modelo tenga una probabilidad moderada de ser multado. Finalmente, todos aquellos con puntaje menor a 0.2 son considerados como verdes y representan los que se espera que no sean multados.

La clasificación final que se obtiene al hacer uso de esta clasificación es la que se muestra en la siguiente matriz;

|  |  |  |
| --- | --- | --- |
|  | **Multado** | **Sin multar** |
| **Verde** | 5 | 4,581 |
| **Amarillo** | 5 | 1,017 |
| **Rojo** | 4 | 241 |

Con esto concluimos que el modelo se encuentra listo para ser desplegado y utilizado para la proyección de cuando los proyectos nuevos en licitación pueden ser o no multados.

**4. Plan de Implementación del prototipo entregable**

**a. Trabajo Validado y listo para el Prototipo**

i. Modelo de datos y primera versión de la ETL

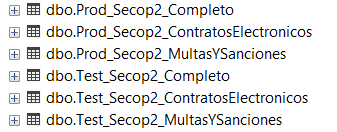
Luego de identificar las fuentes de datos necesarias para alcanzar los objetivos del proyecto, se crearon 3 notebooks de Python que sirvieron para la extracción de los datos desde la fuente, usando Socrata.

|  |  |
| --- | --- |
| **ETL** | **Links** |
| Contratos Electrónicos | [Link a repositorio](https://github.com/jsantacruzl/ProyectoDeGradoMIAD_o3b/blob/main/ETL/GP_Proyecto_ETL%20-%20Contratos%20Electronicos.ipynb) |
| Procesos de Contratación | [Link a repositorio](https://github.com/jsantacruzl/ProyectoDeGradoMIAD_o3b/blob/main/ETL/GP_Proyecto_ETL%20-%20Secop2_Completo.ipynb) |
| Multas y Sanciones | [Link a repositorio](https://github.com/jsantacruzl/ProyectoDeGradoMIAD_o3b/blob/main/ETL/GP_Proyecto_ETL%20-%20Secop2_MultasYSanciones.ipynb) |

En resumen, los pasos ejecutados por las ETL anteriores son:

* Se importan librerías base, entre ellas Socrata
* Se realiza la conexión a la base de datos del proyecto, creada en Azure.
* Se realiza la conexión al servicio del gobierno para extraer los datos.
* Se trunca la tabla de staging (aquellas que empiezan con *Test\_*) (son 3, una por cada fuente)
* Se cargan los datos desde el API a la tabla de staging (se filtra por *Obra* para Contratos Electrónicos y Procesos de Contratación. Se carga todos los datos de Multas y Sanciones)
* Se cierran las conexiones.

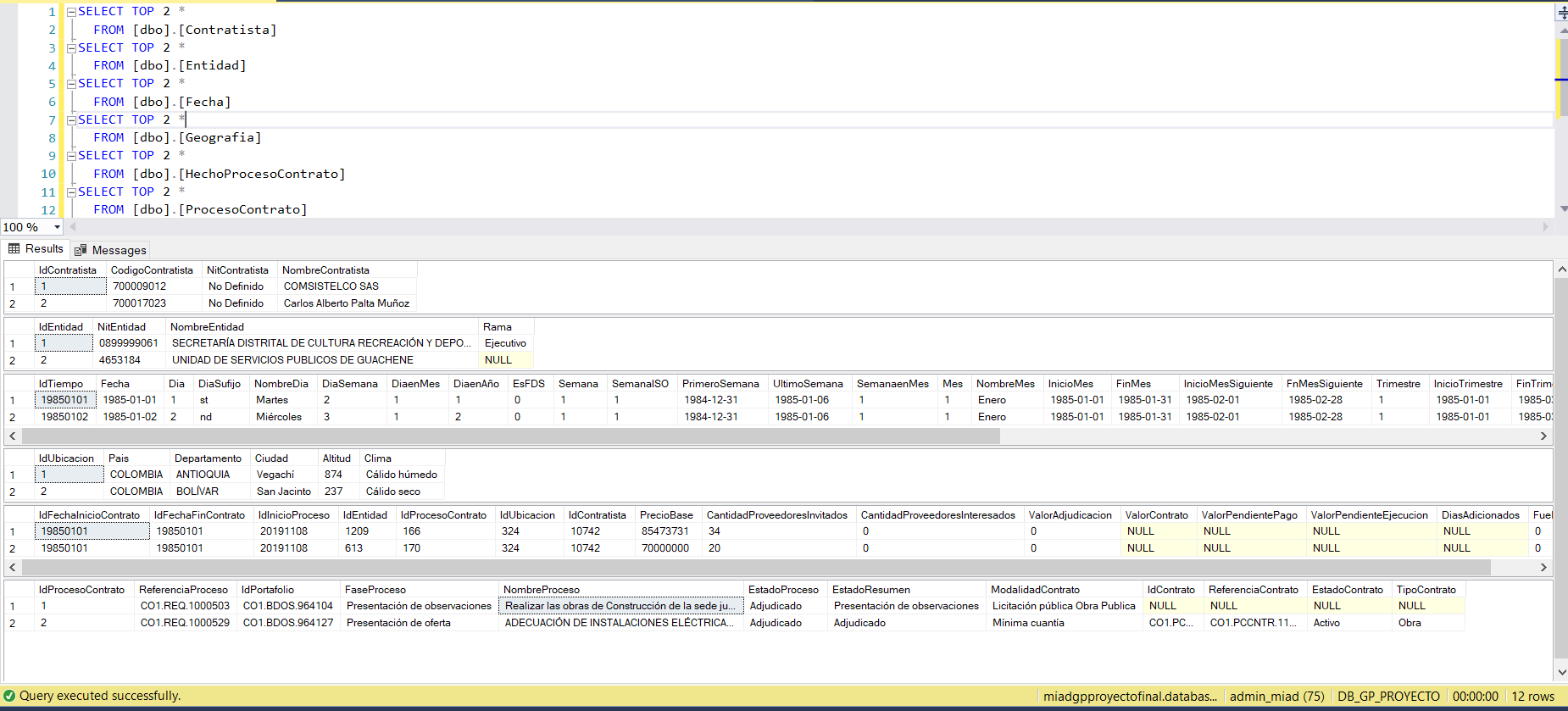
Luego, usando Scripts en SQL, creamos 3 tablas *Prod\_*, que en la dinámica del proyecto son la transformación de las tablas de staging a tablas de negocio propiamente dichas. Este paso es necesario ya que las ETL en Python cargaron la totalidad de los datos como cadenas de caracteres y es claro que muchos de los campos tienen otros tipos, como enteros o fechas. También es necesario reemplazar los “none” y “nan” por NULL en todos los campos.



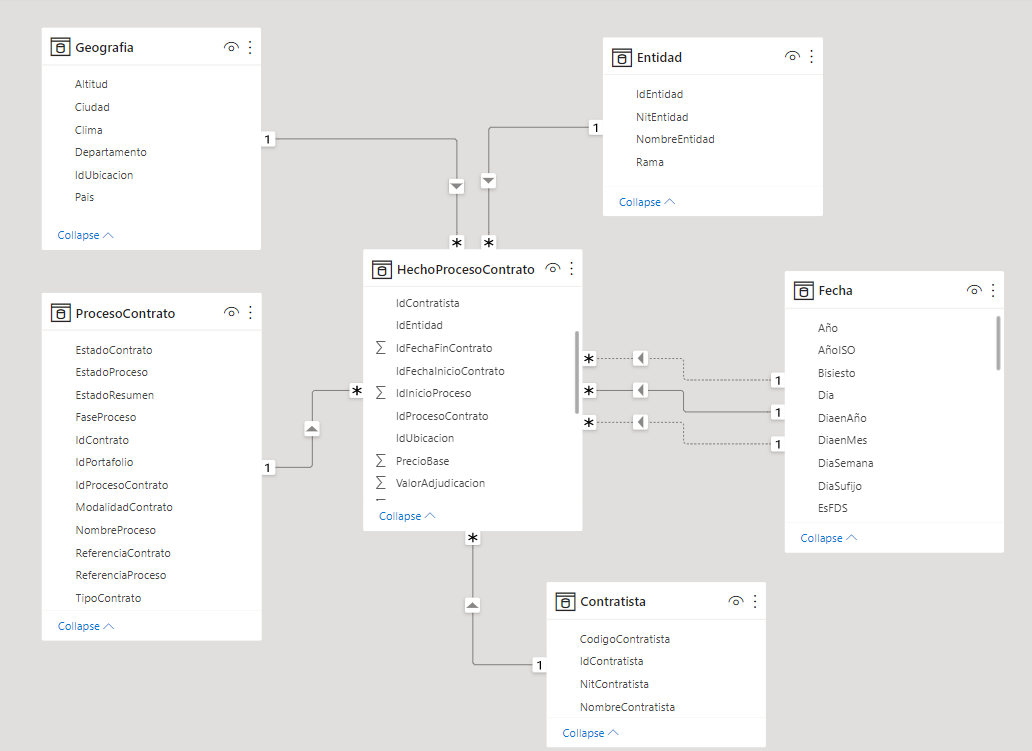
***Fig. 12.*** *Tablas finales creadas para Staging y Tablas de negocio limpias.*

Finalmente, utilizando StoredProcedures y Vistas (Los objetos se encuentran [acá)](https://github.com/jsantacruzl/ProyectoDeGradoMIAD_o3b/tree/main/SQL) se procede a la creación y carga de las tablas del modelo en estrella. Las dimensiones de Fecha y Geografía se cargaron de manera independiente (Cargando las fechas diarias desde 1985 hasta 2057 y la totalidad de los municipios de Colombia). Todas estas tablas cuentan con llaves primarias generadas al momento de la carga, a excepción de la tabla de Hechos que no cuenta con llave primaria, pero si hace referencia a llaves surrogadas de todas las dimensiones. Para las dimensiones de Entidad, Contratista, ProcesoContrato y HechoProcesoContrato, su carga es derivada directamente de las tablas *Prod\_*.

Se adjunta un ejemplo de los datos cargados al modelo en estrella, así como su representación gráfica final.

***Fig. 13.*** *Ejemplos de datos cargados en el modelo en estrella.*

|  |  |  |
| --- | --- | --- |
| **Nombre Dimensión/Hechos** | **Granularidad** | **Fuente de datos y tipo de llaves** |
| Geografia | Ciudad (únicamente ciudades colombianas) | Fuente de datos externa (Inserts con los 1104 municipios). Llave primaria surrogada. |
| Fecha | Fecha | Fuente de datos externa (Script generador de fechas). Llave primaria es la fecha en formato yyyymmdd |
| Entidad | Entidad (Usando NIT como identificador de Negocio) | Contratos Electrónicos  Procesos de Contratación Llave primaria surrogada. |
| Contratista | Proveedor/Contratista (Usando Código de Contratista como identificador de Negocio) | Contratos Electrónicos  Procesos de Contratación Llave primaria surrogada. |
| ProcesoContrato | Proceso y/o Contrato  (En la parte siguiente de pruebas se explica esto con más detalle) | Contratos Electrónicos  Procesos de Contratación Llave primaria surrogada. |
| HechoProcesoContrato | Proceso y/o Contrato | Contratos Electrónicos  Procesos de Contratación  Multas y Sanciones  Llaves foráneas a todas las dimensiones |

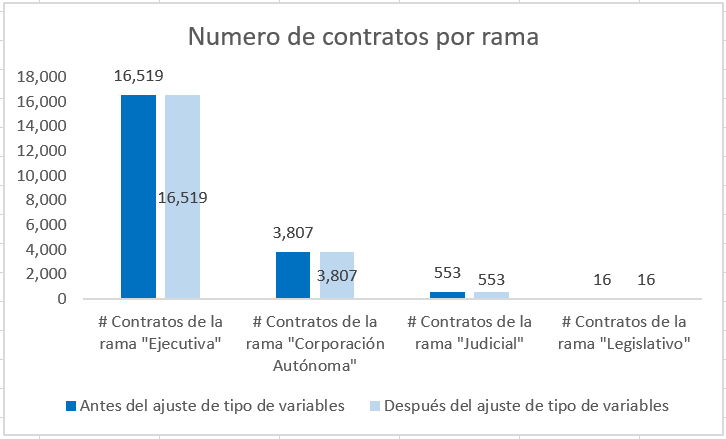
***Fig. 14.*** *Modelo en estrella final con los atributos seleccionados por dimensión y sus relaciones con la tabla de Hechos. La relación triple de Fecha contra la tabla de Hechos permite filtrar por Fecha de Publicación de Proceso, Fecha Inicio de Contrato o Fecha Fin de Contrato*

ii. Pruebas sobre el modelo de datos

1. Contratos electrónicos:Al descargar los datos desde la página del SECOP, se pudo observar que las variables no tenían el tipo de variable correcto. Todas las variables eran de tipo texto a pesar de que algunas variables eran numéricas, fechas, etc. Por lo tanto, el primer paso fue convertir cada variable a su tipo de datos respectivo sin perder información. Para esto, se hicieron las siguientes validaciones:

|  |  |  |
| --- | --- | --- |
| **Criterio** | **Antes de la conversión de las variables a su tipo de datos respectivo** | **Después de la conversión de las variables a su tipo de datos respectivo** |
| # de filas | 20.895 | 20.895 |
| Valores únicos de la variable “Rama” | Ejecutivo, corporación autónoma, judicial, legislativo | Ejecutivo, corporación autónoma, judicial, legislativo |
| Cantidad de valores nulos | 63.419 | 63.419 |

Adicionalmente, se contaron la cantidad de contratos que pertenecían a cada rama. En el siguiente gráfico se pueden ver los resultados:



***Fig. 15.*** *Comparativo de datos entre las tablas de staging y las tablas finales.*

2. Multas y sanciones**:** Al descargar los datos desde la página del SECOP, se pudo observar que las variables no tenían el tipo de variable correcto. Todas las variables tenían el formato de texto a pesar de tener variables numéricas y fechas. Por lo tanto, el primer paso fue convertir cada variable a su tipo de datos respectivo sin perder información. Para esto, se hicieron las siguientes validaciones:

|  |  |  |
| --- | --- | --- |
| **Criterio** | **Antes de la conversión de las variables a su tipo de datos respectivo** | **Después de la conversión de las variables a su tipo de datos respectivo** |
| # de filas | 486 | 486 |
| Valores únicos de la variable “Versión" | 1,2,3,4,5,6,7,8,9,10,11,14,25 | 1,2,3,4,5,6,7,8,9,10,11,14,25 |
| Cantidad de valores nulos | 734 | 746 |

La cantidad de valores nulos es diferente ya que la variable “Aplico Garantías” tenía las opciones de “True”, “False” o “None”. Sin embargo, al ajustar los tipos de variables se agruparon los “None” y los “False”. Por lo tanto, antes del ajuste de los tipos de variables habían 10 nulos adicionales. Sin embargo, es importante mencionar que esta variable “Aplico Garantías” no se usa en nuestros análisis.

3. Procesos de contratación**:** Al descargar los datos desde la página del SECOP, se pudo observar que las variables no tenían el tipo de variable correcto. Todas las variables tenían el formato de texto a pesar de tener variables numéricas y fechas. Por lo tanto, el primer paso fue convertir cada variable a su tipo de datos respectivo sin perder información. Es importante mencionar, que nuestros análisis serán relacionados únicamente con los contratos de tipo obra, por lo que se debía verificar que solo teníamos contratos tipo “Obra”. Para esto, se hicieron las siguientes validaciones:

|  |  |  |
| --- | --- | --- |
| **Criterio** | **Antes de la conversión de las variables a su tipo de datos respectivo** | **Después de la conversión de las variables a su tipo de datos respectivo** |
| # de filas | 40.917 | 40.917 |
| Valores únicos de la variable “Tipo Contrato” | Obra | Obra |
| Valores únicos de la variable “Orden Entidad” | Nacional, Territorial, Corporación Autónoma | Nacional, Territorial, Corporación Autónoma |
| Cantidad de valores nulos | 135.811 | 135.811 |

Después de tener todos los datos con el tipo de dato correcto, se construyeron 2 tablas fundamentales para nuestros análisis posteriores. La primera tabla es “MultaProcesoContrato” en donde se encuentran todos los contratos y procesos de tipo Obra que fueron multados. Por lo tanto, se unieron las 3 fuentes de datos mencionadas anteriormente. Para validar que no se haya perdido información en la unión de las diferentes fuentes de datos se tomaron los siguientes pasos:

1. Se encontraron los contratos que fueron multados (unión entre “Multas y sanciones” y “Contratos Electrónicos” a través de la variable “ID Contrato”)
2. Se encontraron los procesos que fueron multados (unión entre “Multas y sanciones” y “Procesos” a través de la variable “Referencia del proceso”)
3. Se encontraron los valores únicos de los contratos y procesos que fueron multados en los pasos anteriores.
4. Se verifica que la tabla “MultaProcesoContrato” tenga la misma cantidad de procesos y contratos únicos multados encontrados en el paso 3.

|  |  |  |  |
| --- | --- | --- | --- |
| **Criterio** | **Fuente de datos Contratos Electrónicos** | **Fuente de datos Proceso** | **Nueva tabla llamada “MultaProcesoContrato”** |
| # Contratos multados | 27 | N/A | 27 |
| # Contratos multados | N/A | 39 | 39 |

La segunda tabla fundamental para nuestros análisis es “ProcesoContrato” que es la unión entre las fuentes de datos “Procesos” y “Contratos electrónicos”. Para la construcción de esta tabla se tomaron todos los procesos de la fuente de datos “Procesos” y se buscó su respectivo contrato en la fuente de “Contratos electrónicos”. Esta tabla contiene incluso los procesos que no tienen un contrato respectivo o contratos que no tienen un proceso asociado. Por lo tanto, el número de procesos encontrados en la fuente de datos “Procesos” debe ser igual al número de procesos de la nueva tabla “ProcesoContrato”

|  |  |  |
| --- | --- | --- |
| **Criterio** | **Fuente de datos “Procesos”** | **Nueva tabla llamada “ProcesoContrato”** |
| # Procesos únicos | 24.559 | 24.559 |

Finalmente, se construyó una sábana de datos uniendo las últimas dos tablas mencionadas anteriormente (MultaProcesoContrato y ProcesoContrato). Es decir que se construyó una tabla final llamada “SabanaAnalitica” que sería la unión de todas las fuentes de datos. El principal objetivo de esta tabla final es servir como entrada para el análisis de ML realizado con el fin de pronosticar futuros proyectos que potencialmente podrían ser sancionados, así como servir de fuente para las dimensiones *ProcesoContrato* y *HechoProcesoContrato* descritos anteriormente. Se hicieron las siguientes validaciones:

|  |  |  |
| --- | --- | --- |
| **Criterio** | **Tabla “ProcesoContrato”** | **Nueva tabla llamada “SabanaAnalitica”** |
| # Contratos únicos | 20.892 | 20.892 |
| # Procesos únicos | 36.986 | 36.986 |

Los scripts de SQL utilizados para la construcción de los objetos se pueden encontrar en este [link.](https://github.com/jsantacruzl/ProyectoDeGradoMIAD_o3b/tree/main/SQL) El notebook usado para las pruebas se puede encontrar [acá.](https://github.com/jsantacruzl/ProyectoDeGradoMIAD_o3b/blob/main/ETL/Pruebas%20Sobre%20Modelo%20de%20Datos%20e%20Integraciones.ipynb)

iii. Primer Borrador del Tablero

Se cuenta con una primera versión del tablero, que por restricciones con el sitio powerbi.com y con la licencia que se cuenta, no permiten exponerlo al público general. Sin embargo, en el [repositorio](https://github.com/jsantacruzl/ProyectoDeGradoMIAD_o3b/tree/main/Dashboard) contamos con dos archivos: El archivo PBI propiamente dicho y un archivo en PDF que muestra el avance parcial obtenido. De este tablero vale la pena mencionar que faltan detalles de narrativa, UI/UX que serán completados para la entrega final. Lo adjuntamos en esta sección ya que es la muestra concreta del uso de los datos extraídos en los procesos anteriores.

**b. Pendientes por implementar**

i. Alimentación de los resultados del pronóstico al modelo en estrella.

Una vez seleccionado el modelo, se debe almacenar las predicciones realizadas por el mismo para proyectos con fechas superiores a 2023. Para esto, se debe realizar un script en Python que cambie a 1 el valor de la columna *FueMultado* de la tabla *HechoProcesoContrato* a aquellos *ProcesoContrato* que aplique. Con esto y con el manejo de filtros ya definidos en el tablero, se mostrará al usuario únicamente aquellos proyectos que son sensibles a ser afectados por una multa o sanción en un futuro. Este script de Python debe tener en cuenta las llaves surrogadas que son las que identifican cada registro en la tabla de Hechos, así como también tener en cuenta que algunos procesos o contratos pueden tener uno o más proveedores asociados en la tabla de Hechos.

ii. Elaboración de encuestas y recolección de resultados

Una vez finalizado el desarrollo del tablero, elaboraremos la encuesta que servirá para medir el nivel de comprensión inicial (antes de usar el tablero) y final (luego de usarlo) de algunas personas del público general. La idea es poder medir el cambio en su conocimiento sobre los procesos de contratación estatales. Por cuestiones de tiempo y recursos, tenemos como meta realizar la encuesta entre 10 y 20 personas y luego de obtener sus resultados, consolidaremos los resultados para la presentación final.

iii. Construcción de presentación final y pitch

Elaboraremos una presentación corta en la cual consolidaremos las principales conclusiones y logros obtenidos a lo largo del proyecto. También listaremos potenciales pasos futuros para continuar con la evolución de este prototipo.

**5. Bibliografía**

* Accuracy y Recall - <https://developers.google.com/machine-learning/crash-course/classification/precision-and-recall?hl=es-419>
* F1 Score - <https://towardsdatascience.com/accuracy-precision-recall-or-f1-331fb37c5cb9>
* ROC y AUC - <https://scikit-learn.org/stable/auto_examples/model_selection/plot_roc.html>