



UNIVERSIDAD DE SANTIAGO DE CHILE

Predicción de Calidad en Proceso de Flotación Inversa de Concentrado de Hierro.

Proyecto de título, Diplomado Data Science.

ALUMNO: Matías Obaid González.

SANTIAGO DE CHILE.

2023.

ENTREGA 1

Introducción:

El mineral extraído de la mina, si bien contiene los elementos valiosos que justifican económicamente su extracción, se encuentra inicialmente en un estado “impuro”. Es necesario reducir la presencia de la “ganga” (material de nulo o negativo interés económico) con la que se encuentra asociado, mientras que simultáneamente se busca obtener concentraciones mayores del elemento de valor.

Para esto es que existen las plantas de procesamiento de mineral, donde hay una gran cantidad de métodos y diseños de procesamiento, principalmente dependiendo de la composición química del mineral a tratar, así mismo como el ritmo de producción deseado, entre muchas otras variables.

En el caso de este proyecto, se cuenta con un dataset que se refiere específicamente al proceso de flotación inversa dentro una planta concentradora de hierro en Brasil, teniendo como variables objetivo o “Target Labels”, el porcentaje de Hierro (%) y el porcentaje de Sílice/ganga (%) presente en el concentrado posterior al proceso de flotación.

El diagrama de flujo resumido a gran escala que ocurre en un proceso minero de este tipo es el siguiente:

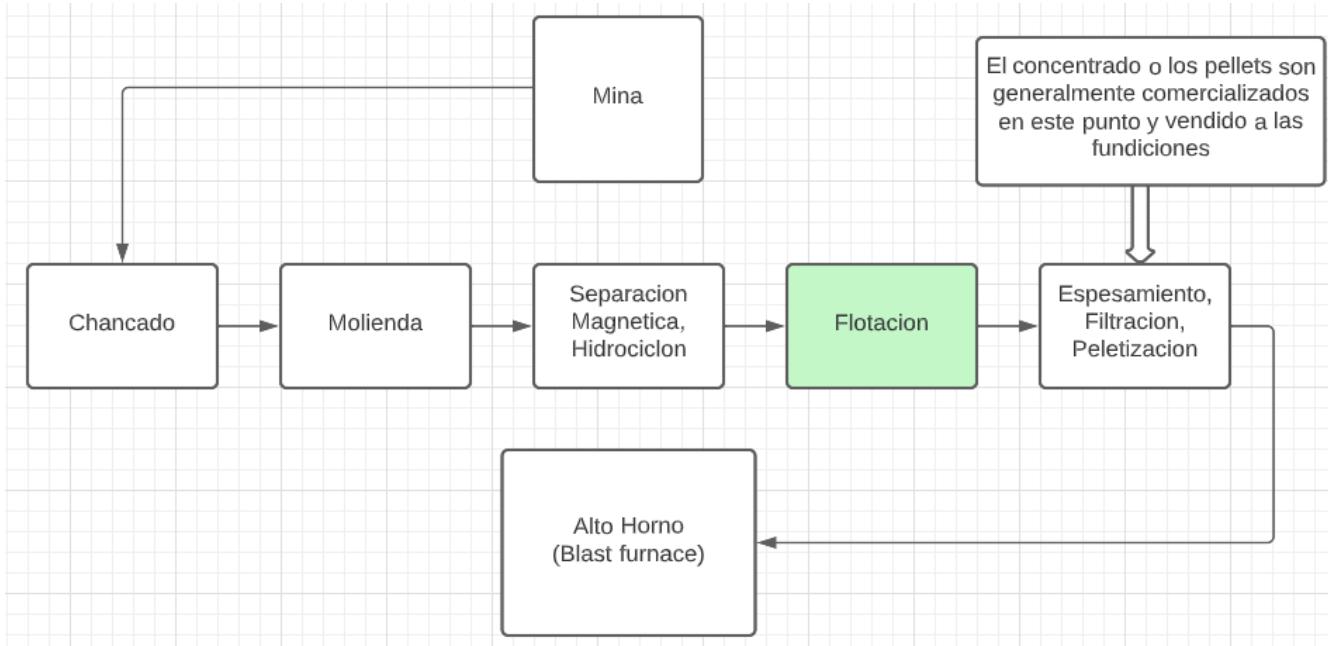


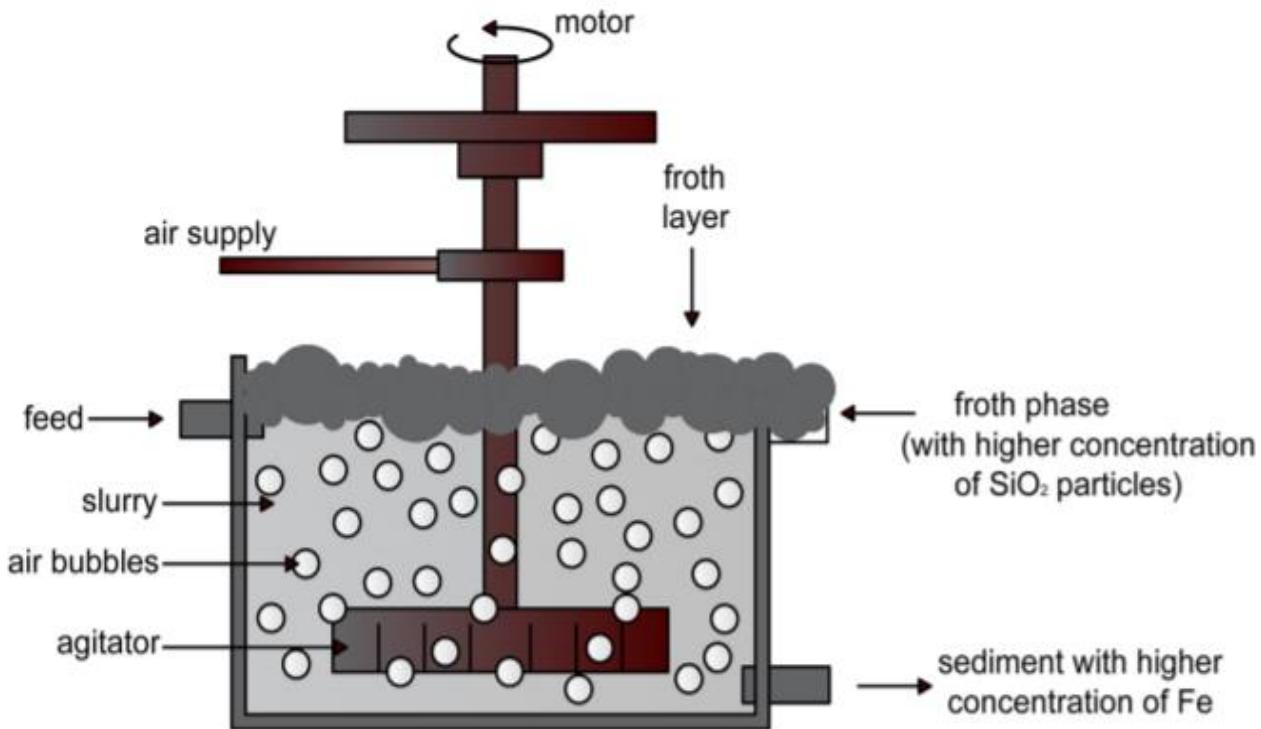
Figura 1: Diagrama de flujo, Fuente: Elaboración propia basado en información de 911metallurgist.com.

Proceso de flotación:

La separación de minerales, aprovechándose de una diferencia inducida de propiedades superficiales mediante reactivos (hidrofobicidad en este caso) se denomina flotación. La flotación inversa se usa comúnmente para separar el hierro de la ganga. Al ajustar la 'química' de la pulpa agregando varios reactivos químicos, los minerales de hierro permanecen en el agua y crean sedimentos con una alta concentración de hierro (minerales valiosos). Al mismo tiempo, las partículas de sílice (ganga) se adhieren a las burbujas de aire y flotan hacia la superficie.

No confundir con el proceso de flotación directa comúnmente conocido en Chile en el ámbito del procesamiento de sulfuros de cobre, en la flotación directa, el mineral de interés (Cobre) es el que "flota" con las burbujas de aire, mientras que la ganga precipita.

El proceso de flotación inversa como ya se mencionó, es lo opuesto, donde los minerales de interés precipitan, mientras que la ganga es la que queda adherida a las burbujas y emerge a la parte superior de la celda de flotación.



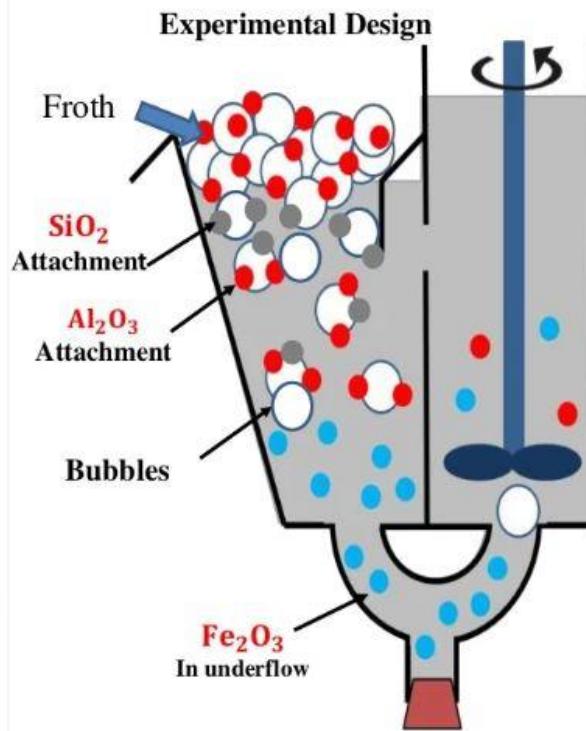
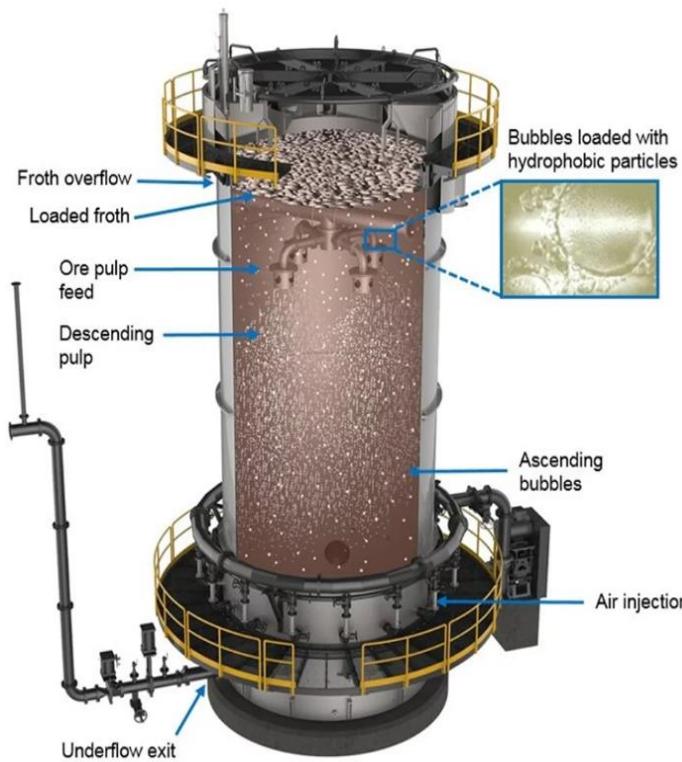


Figura 2,3,4: Ilustraciones de celdas de flotación inversa, Fuentes: Diversos sitios web adjuntados en bibliografía.

Justificación y preguntas/objetivos de valor:

El concentrado de hierro o los pellets son comercializados lógicamente en base a su contenido de hierro. Sin embargo, existen “penalizaciones” o “premios” a el precio de venta en base al porcentaje de impurezas que este tiene, en el caso de los pellets o concentrados de hierro una de las impurezas más comunes es la sílice.

Esta impureza entorpece los procesos de alto horno y refinación posteriores. Estos son los motivos económicos para poder controlar el porcentaje de sílice en el proceso de flotación.

También existe el dilema de que, en la etapa de flotación, es imposible saber de inmediato tanto el % de hierro como el % de sílice presente en el concentrado posterior al proceso, es necesario tomar muestras y realizar análisis de laboratorio para determinar las concentraciones de estos, cuyos resultados son conocidos tiempo más tarde.

Teniendo esto en cuenta surgen las siguientes **preguntas u objetivos de valor:**

- Construir un modelo de aprendizaje supervisado para predecir el porcentaje de sílice presente en el concentrado posterior al proceso de flotación, esto permitiría a los Ingenieros Metalurgistas o de Procesos, actuar de forma temprana para tomar medidas correctivas/preventivas ya que no tendrían que estar esperando los resultados de laboratorio para saber exactamente el % de Sílice presente en el concentrado, un modelo de machine learning les podría dar de forma rápida una buena aproximación de este valor, lo anterior ayudaría a incrementar el valor del negocio, reduciendo las penalizaciones en el precio de venta por presencia de impurezas de sílice en el concentrado.
- Identificar las variables más relevantes que intervienen a la hora de obtener un concentrado con mayor o menor porcentaje de impureza de sílice, de esta forma los ingenieros involucrados en el proceso tendrán mayor claridad acerca de que variables explicativas intervenir en el proceso de flotación para modificar el resultado de % de sílice presente en el Concentrado, afectando como es mencionado anteriormente el valor del negocio ya que se podrían reducir las penalizaciones al precio de venta causados por esta impureza.

ENTREGA 2

Información de variables ORIGINALES del dataset:

El dataset es relativamente masivo, cuenta con 737.453 filas, 23 columnas y pesa alrededor de 179mb como CSV.

Existen problemas temporales de frecuencia de medición entre las variables, las variables objetivo son actualizadas con una frecuencia de 1 hora, mientras que la mayoría de las variables explicativas son actualizadas con una frecuencia de 20 segundos con la excepción de % Iron Feed y % Silica Feed, que son actualizadas cada 8 horas.

El dataset fue adquirido a través del sitio web Kaggle, publicado por el sr Eduardo Magalhaes Oliveira. Se adjunta enlace del dataset en la bibliografía.

Variables Explicativas:

- **Date:** Fecha de la medición. (2017-03-10 1:00:00 a 2017-09-09 23:00:00) (DateTime64ns)
- **% Iron Feed:** Porcentaje de hierro de la pulpa que está siendo alimentada a las celdas de flotación (0-100%). (Min 42.74%, max 65.78%) (Float64)
- **% Silica Feed:** Porcentaje de sílice de la pulpa que está siendo alimentada a las celdas de flotación. (0-100%). (Min 1.31%, max 33.4%) (Float64)
- **Starch Flow:** Flujo de Almidón (reactivo) medido en m3/h. (min 0.002026 m3/h, Max 6300.23 m3/h) (Float64)
- **Amine Flow:** Flujo de Amina (reactivo) medido en m3/h. (min 241.669 m3/h, Max 739.538 m3/h) (Float64)
- **Ore Pulp Flow:** Flujo de alimentación de pulpa medido en t/h. (min 376.249 t/h, Max 418.641 m3/h) (Float64)
- **Ore Pulp pH:** pH de la pulpa, escala de 0 a 14. (Min 8.7533 ph, max 10.808ph) (Float64)
- **Ore Pulp Density:** Densidad de la pulpa medida en kg/cm³. (Min 1.519 kg/cm3, max 1.853 kg/cm3) (Float64)

- **Flotation Column Air Flow (1):** Flujo de aire que está entrando en celda de flotación 1, medido en Nm³/h. (min 175.510 Nm³/h, Max 373.871 Nm³/h) (Float64)
- **Flotation Column Air Flow (2):** Flujo de aire que está entrando en celda de flotación 2, medido en Nm³/h. (min 175.156 Nm³/h, Max 375.992 Nm³/h) (Float64)
- **Flotation Column Air Flow (3):** Flujo de aire que está entrando en celda de flotación 3, medido en Nm³/h. (min 176.469 Nm³/h, Max 364.346 Nm³/h) (Float64)
- **Flotation Column Air Flow (4):** Flujo de aire que está entrando en celda de flotación 4, medido en Nm³/h. (min 292.195 Nm³/h, Max 305.871 Nm³/h) (Float64)
- **Flotation Column Air Flow (5):** Flujo de aire que está entrando en celda de flotación 5, medido en Nm³/h. (min 286.295 Nm³/h, Max 310.27 Nm³/h) (Float64)
- **Flotation Column Air Flow (6):** Flujo de aire que está entrando en celda de flotación 6, medido en Nm³/h. (min 189.928 Nm³/h, Max 370.91 Nm³/h) (Float64)
- **Flotation Column Air Flow (7):** Flujo de aire que está entrando en celda de flotación 7, medido en Nm³/h. (min 185.962 Nm³/h, Max 371.593 Nm³/h) (Float64)
- **Flotation Column Level (1):** altura de la capa de burbujas en la parte superior de la celda de flotación 1, medido en mm. (min 149.2 mm, Max 862.2 mm) (Float64)
- **Flotation Column Level (2):** altura de la capa de burbujas en la parte superior de la celda de flotación 2, medido en mm. (min 210.7 mm, Max 828.9 mm) (Float64)
- **Flotation Column Level (3):** altura de la capa de burbujas en la parte superior de la celda de flotación 3, medido en mm. (min 126.2 mm, Max 886.8 mm) (Float64)
- **Flotation Column Level (4):** altura de la capa de burbujas en la parte superior de la celda de flotación 4, medido en mm. (min 162.2 mm, Max 680.3 mm) (Float64)
- **Flotation Column Level (5):** altura de la capa de burbujas en la parte superior de la celda de flotación 5, medido en mm. (min 166.9 mm, Max 675.6 mm) (Float64)
- **Flotation Column Level (6):** altura de la capa de burbujas en la parte superior de la celda de flotación 6, medido en mm. (min 155.8 mm, Max 698.8 mm) (Float64)

- **Flotation Column Level (7):** altura de la capa de burbujas en la parte superior de la celda de flotación 7, medido en mm. (min 175.3 mm, Max 659.9 mm) (Float64)

VARIABLES OBJETIVO:

- **% Iron Concentrate:** Porcentaje de hierro en el concentrado al final del proceso de flotación (%), obtenido con análisis de laboratorio posterior. (min 62.05%, max 68.01%) (Float64)
- **% Silica Concentrate:** Porcentaje de sílice en el concentrado al final del proceso de flotación (%), obtenido con análisis de laboratorio posterior. (min 0.6%, max 5.63%) (Float64)

Preprocesamiento:

El preprocesamiento de este dataset es particularmente complejo y único (se recomienda referirse al Jupyter Notebook en caso de confusión, duda o mayor detalle), las frecuencias de medición de las variables son distintas, existe un tramo de discontinuidad temporal, existen intervalos donde la variable objetivo se actualiza muy por encima de su frecuencia habitual y tramos donde no se actualiza en lo absoluto. A continuación, se detallarán los análisis y acciones más significativas realizadas.

Valores Perdidos:

El Dataset no cuenta con valores faltantes.

	Total	Percent
% Iron Feed	0	0.0
% Silica Feed	0	0.0
Starch Flow	0	0.0
Amina Flow	0	0.0
Ore Pulp Flow	0	0.0
Ore Pulp pH	0	0.0
Ore Pulp Density	0	0.0
Flotation Column 01 Air Flow	0	0.0
Flotation Column 02 Air Flow	0	0.0
Flotation Column 03 Air Flow	0	0.0
Flotation Column 04 Air Flow	0	0.0
Flotation Column 05 Air Flow	0	0.0
Flotation Column 06 Air Flow	0	0.0
Flotation Column 07 Air Flow	0	0.0
Flotation Column 01 Level	0	0.0
Flotation Column 02 Level	0	0.0
Flotation Column 03 Level	0	0.0
Flotation Column 04 Level	0	0.0
Flotation Column 05 Level	0	0.0
Flotation Column 06 Level	0	0.0
Flotation Column 07 Level	0	0.0
% Iron Concentrate	0	0.0
% Silica Concentrate	0	0.0

Figura 5: Tabla de Valores Perdidos.

Filas Duplicadas:

El Dataset inicialmente cuenta con 1171 filas duplicadas.

Continuidad temporal:

Si bien se observa que no existen valores faltantes, es de vital importancia analizar el factor tiempo y garantizar que exista una continuidad temporal en la data. Sobre todo, si uno pretende crear modelos de series de tiempo.

Se verifico que existe una inconsistencia temporal, hay horas faltantes en el dataset entre el 2017-03-16 06:00:00 y 2017-03-29 11:00:00, es posible que quizás la planta entro en mantenimiento entre estas fechas.

Se procedió a eliminar las filas previas al "2017-03-29 12:00:00".

El dataset posterior a esta eliminación conservo el 96.36% de las filas, 710.639 en total.

Continuando con el Preprocesamiento:

Observando los 2 gráficos a continuación (Para más detalle ver notebook) observamos algunas anomalías, las variables objetivo (**% Silica Concentrate y % Iron Concentrate**), son actualizadas en promedio 15 y 11 veces por hora respectivamente, esto es contrario a lo que señala el autor del dataset que especifica que estas columnas son actualizadas 1 vez por hora, realizando un poco de manejo del dataset identificamos que en la mayoría de los casos el valor se actualiza efectivamente cada 1 hora, sin embargo existen algunas horas donde estos valores son actualizados cada 20 segundos, es decir 180 veces por hora, lo que explica porque el promedio está dando 15 y 11 para estas columnas. Esto nos hace pensar que el dataset en estas horas fue **intervenido/modificado por alguien y se intentó realizar una especie de interpolación** para estas 2 variables en algunas horas en concreto.

Se observa también que las columnas que hablan de la ley de la alimentación del material (**% Silica Feed y % Iron Feed**), son actualizadas en promedio 3 veces por día, es decir 1 vez cada 8 horas.

El resto de las variables '**Starch Flow**', '**Amina Flow**', '**Ore Pulp Flow**', '**Ore Pulp pH**', '**Ore Pulp Density**', '**Flotation Column 01-07 Air Flow**', '**Flotation Column 01-07 Level**', son actualizadas cada 20 segundos, es decir 180 veces por hora, si bien en el gráfico de frecuencia horaria más abajo se observa que los valores son un poco inferiores a 180 para estas variables, esto es perfectamente entendible/plausible ya que pueden existir valores repetidos entre una medición y otra dentro de la misma hora considerando un intervalo tan corto de medición como 20 segundos.

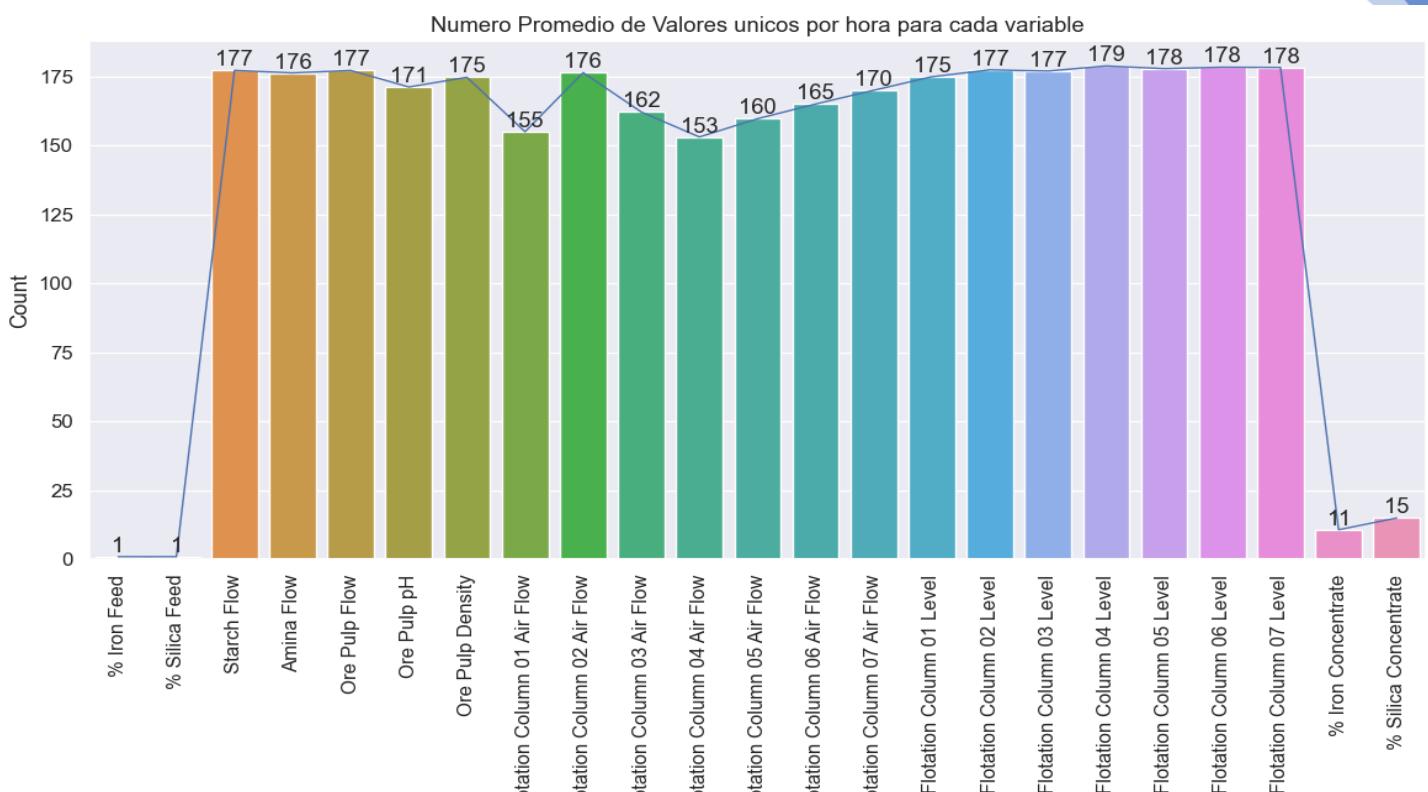


Figura 6: Número Promedio de Valores únicos por hora para cada variable.

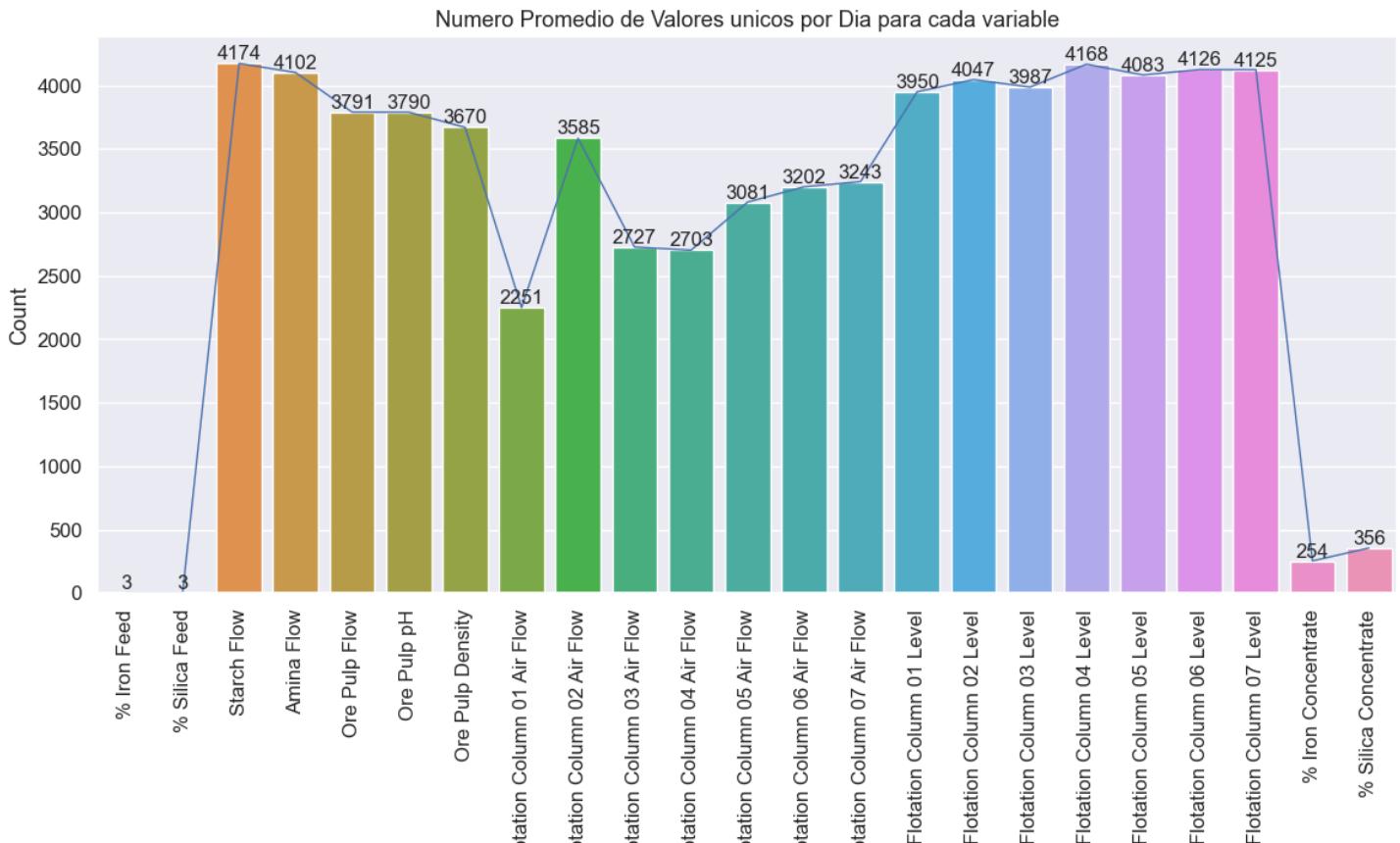


Figura 7: Número Promedio de Valores únicos por día para cada variable.

Teniendo en cuenta la información expuesta anteriormente y considerando principalmente que mi variable objetivo de interés (**% Silica Concentrate**), se actualiza 1 vez por hora. (sin tener en cuenta la anomalía detectada anteriormente mencionada), Es que se ha decidido **RESAMPLEAR EL DATASET CON UNA FRECUENCIA DE 1 HORA**, Tomando solo el primer valor de determinada hora. No tiene sentido tener casi 700.000 filas si es que mi variable objetivo no está experimentando variaciones ya que su frecuencia de medición es pequeña, mientras que gran parte de las variables explicativas tienen una frecuencia de medición/actualización mucho mayor.

A continuación, en el siguiente gráfico (posterior al resampleo), observamos que nuestra variable objetivo (**% Silica Concentrate**) se actualiza en promedio con una frecuencia Diaria (un poco menor) al resto de las variables explicativas posterior al resampleado, con excepción de nuestras variables de alimentación **% Iron feed y % Silica Feed** que sabemos que tienen una frecuencia de actualización de 3 veces por día o una vez cada 8 horas.

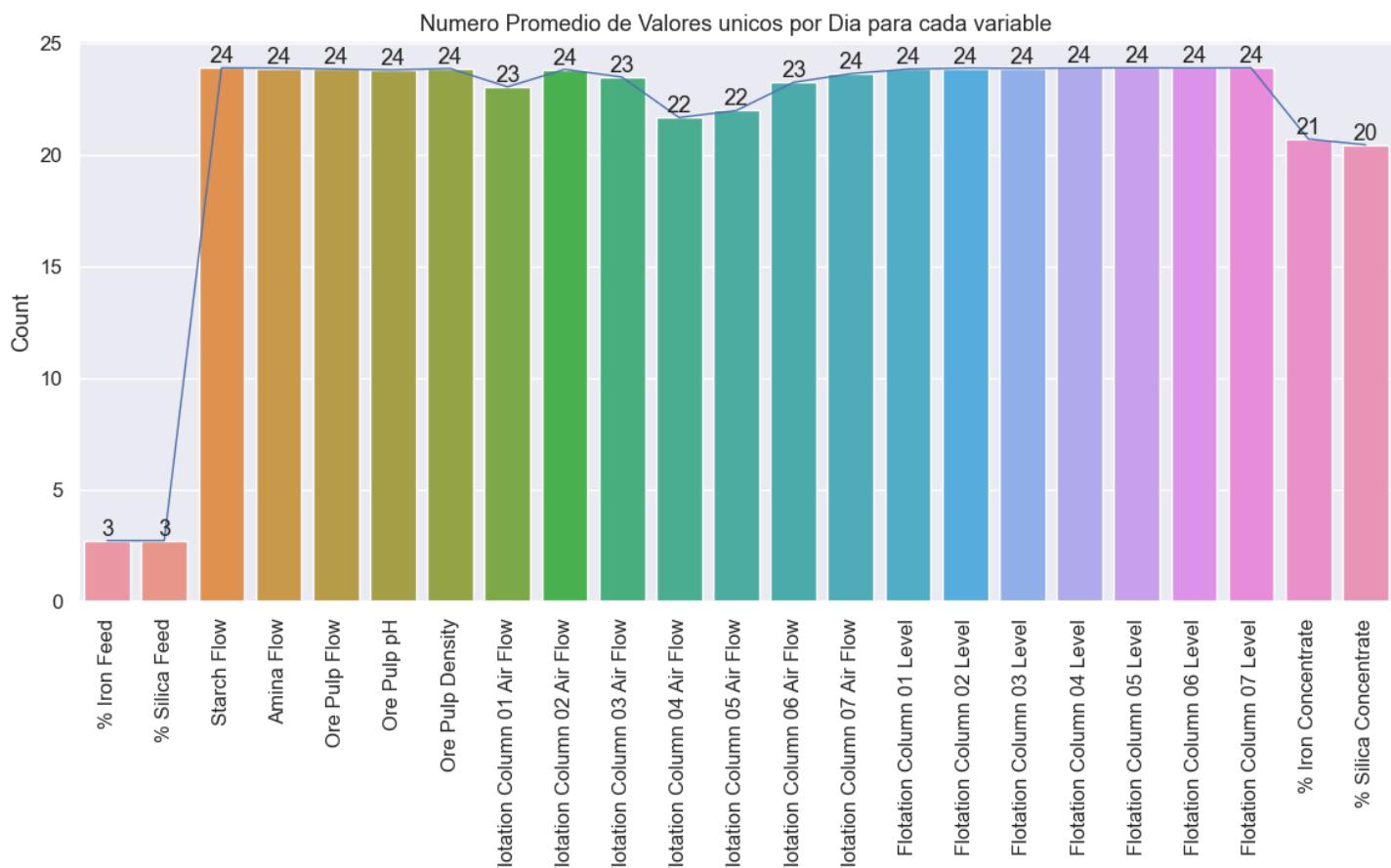


Figura 8: Numero Promedio de Valores únicos por día para cada variable posterior al resampleado.

Indagando el motivo por el cual nuestra variable objetivo % Silica Concentrate se está actualizando en promedio 20 veces por día en lugar de 24, **descubrimos que hay extensiones de tiempo específicos, donde esta variable no se actualiza**, esto es se visualiza y entiende de mejor forma con el siguiente gráfico.

Creare un lineplot y señalare visualmente con líneas rojas verticales, los intervalos de tiempo (más evidentes, pero hay más) que tienen estos problemas de "no actualización" de la variable objetivo respecto al registro de la hora anterior para que quede más claro.

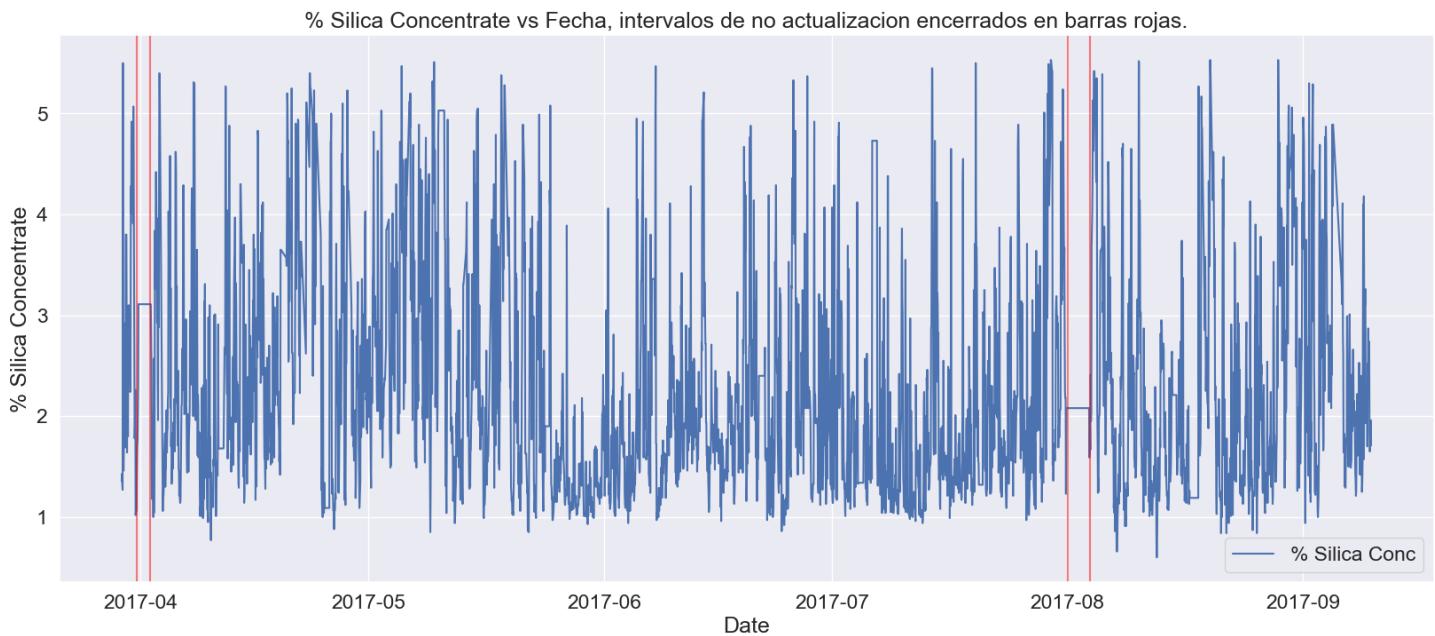


Figura 9: % Silica Concentrate vs Fecha, señalando intervalos de no actualización.

Esto sigue siendo un problema que merece ser analizado (idealmente la frecuencia de actualización de mi variable objetivo debería ser la misma que mis variables explicativas). No considero que la interpolación de la variable objetivo sea una opción válida para remediar este problema entre estas fechas, tampoco me gusta la opción de eliminar estos intervalos ya que tendría discontinuidad temporal. Desde la data que tengo disponible no puedo hacer nada para solucionar este problema en específico. Sería ideal conversar con las personas que construyeron el dataset y operaron la planta de flotación en estas fechas para poder entender las causas de esta anomalía y ver alguna posible solución. De todas formas, en gran parte del dataset vemos que la variable objetivo se va actualizando correctamente con el correr del tiempo.

Feature Engineering:

Creare un par de columnas en relación con el tiempo, una que me indique la hora y otra que me indique el día de la semana de la medición (fila). También creare otro par de columnas con transformaciones para cada una de estas dos variables recién creadas, con la finalidad de capturar la "ciclicidad" tanto del día de la semana, como de la hora. Ya que no puedo alimentar directamente a los algoritmos de ML el valor numérico de la "hora" por ejemplo (0,1, 2, ..., 23), si bien la hora es ordinal, esto no se cumple para el caso de extremo de que 0 viene después que 23. Necesito encontrar alguna forma de decirle a mi algoritmo la hora y día de la semana en un formato cíclico.

Diversos artículos señalan que una solución a esta problemática es aplicar 2 transformaciones basadas en el seno y el coseno, creando dos dimensiones (columnas) en base a la variable temporal ordinal no cíclica original.

Ej.: para el caso de tener la variable ordinal numérica de la hora (0,1, 2, ..., 23):

Debo crear 2 columnas con las siguientes transformaciones (adjunto código de Python):

$$df["sin_hora] = np.sin(2 * np.pi * df[hora]/24)$$

$$df["cos_hora] = np.cos(2 * np.pi * df[hora]/24)$$

Estas dos columnas cumplen con la finalidad de representar mi variable hora en un formato cíclico que puede ser alimentado a los algoritmos de ML.

Esto también me ahorraría el problema de crear 24 variables dummies para representar la hora en formato categórico. con este método solo debo crear dos columnas nuevas.

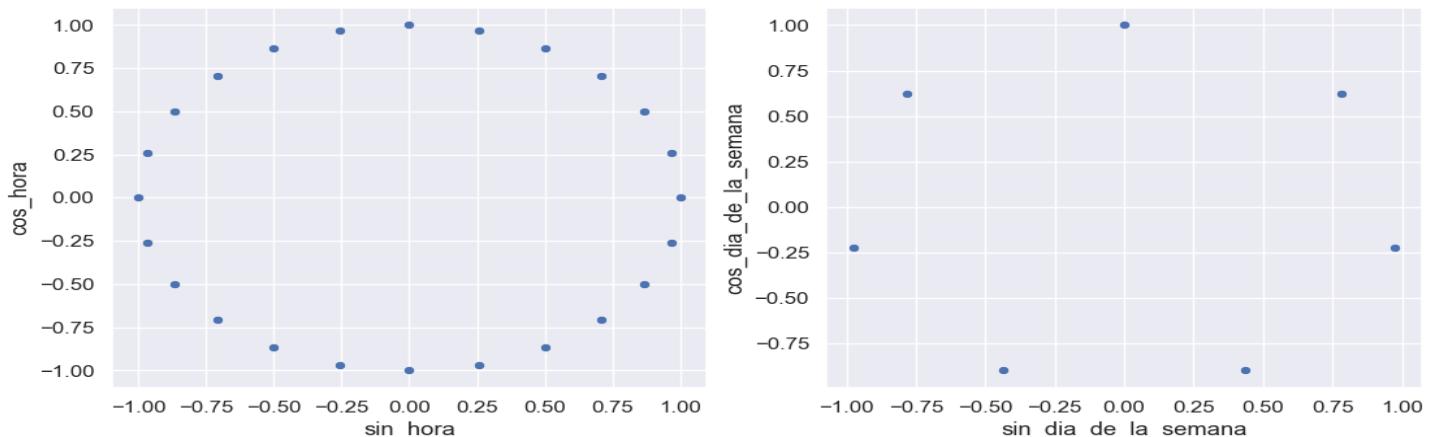


Figura 10: Gráficos que demuestran la ciclicidad obtenida mediante las transformaciones de seno y coseno.

También teóricamente se podría crear una nueva variable útil, pero no se dispone del feedback necesario para crearla, en un ambiente de trabajo real no tendría ningún problema para crear esta variable, pero ya que no tengo interacción con los creadores del dataset y/o la planta de flotación en cuestión me es imposible, esta nueva variable correspondería al **sistema de turnos del personal** que opera la planta.

Por dar un ejemplo:

Si la hora esta entre 1 - 9 = "Turno 1".

Si la hora esta entre 10 -17 = "Turno 2"

Si la hora esta entre 18 – 0 = "Turno 3".

Como medida final para concluir el preprocesamiento eliminare la variable (**% Iron Concentrate**), ya que esta no puede ser utilizada como variable explicativa de mi variable objetivo (**% Silica Concentrate**), ya que ambas son conocidas de forma simultánea tiempo después del proceso de flotación mediante exámenes de laboratorio, por lo que incluirla como variable explicativa supondría un "Data Leakage".

Volveré a adjuntar un diccionario de mis variables posterior al preprocesamiento y Feature Engineering, se confirmó también que la data **no cuenta con valores faltantes y filas duplicadas posterior al preprocesamiento**.

Información de variables después del preprocesamiento del dataset:

El dataset al final del Preprocesamiento cuenta con 3948 filas, 26 columnas y se extiende temporalmente entre el 2017-03-29 12:00:00 al 2017-09-09 23:00:00.

Variables Explicativas:

- **Date:** Fecha de la medición. (2017-03-29 12:00:00 al 2017-09-09 23:00:00) (DateTime64ns)
- **% Iron Feed:** Porcentaje de hierro de la pulpa que está siendo alimentada a las celdas de flotación (%). (Min 42.74%, Max 65.78%) (Float64)

- **% Silica Feed:** Porcentaje de sílice de la pulpa que está siendo alimentada a las celdas de flotación. (%). (Min 1.31%, Max 33.4%) (Float64)
- **Starch Flow:** Flujo de Almidón (reactivo) medido en m3/h. (min 0.561 m3/h, Max 6288.9 m3/h) (Float64)
- **Amine Flow:** Flujo de Amina (reactivo) medido en m3/h. (min 241.7 m3/h, Max 739.3 m3/h) (Float64)
- **Ore Pulp Flow:** Flujo de alimentación de pulpa medido en t/h. (min 376.2 t/h, Max 418.6 m3/h) (Float64)
- **Ore Pulp pH:** pH de la pulpa, escala de 0 a 14. (Min 8.75 ph., max 10.807 ph.) (Float64)
- **Ore Pulp Density:** Densidad de la pulpa medida en kg/cm³. (Min 1.519 kg/cm³, max 1.852 kg/cm³) (Float64)
- **Flotation Column Air Flow (1):** Flujo de aire que está entrando en celda de flotación 1, medido en Nm³/h. (min 175.84 Nm³/h, Max 372.44 Nm³/h) (Float64)
- **Flotation Column Air Flow (2):** Flujo de aire que está entrando en celda de flotación 2, medido en Nm³/h. (min 177.56 Nm³/h, Max 367.25 Nm³/h) (Float64)
- **Flotation Column Air Flow (3):** Flujo de aire que está entrando en celda de flotación 3, medido en Nm³/h. (min 176.94 Nm³/h, Max 304.54 Nm³/h) (Float64)
- **Flotation Column Air Flow (4):** Flujo de aire que está entrando en celda de flotación 4, medido en Nm³/h. (min 293.30 Nm³/h, Max 305.71 Nm³/h) (Float64)
- **Flotation Column Air Flow (5):** Flujo de aire que está entrando en celda de flotación 5, medido en Nm³/h. (min 286.54 Nm³/h, Max 307.52 Nm³/h) (Float64)
- **Flotation Column Air Flow (6):** Flujo de aire que está entrando en celda de flotación 6, medido en Nm³/h. (min 191.89 Nm³/h, Max 370.32 Nm³/h) (Float64)
- **Flotation Column Air Flow (7):** Flujo de aire que está entrando en celda de flotación 7, medido en Nm³/h. (min 195.026 Nm³/h, Max 371.24 Nm³/h) (Float64)

- **Flotation Column Level (1):** altura de la capa de burbujas en la parte superior de la celda de flotación 1, medido en mm. (min 152.34 mm, Max 861.6 mm) (Float64)
- **Flotation Column Level (2):** altura de la capa de burbujas en la parte superior de la celda de flotación 2, medido en mm. (min 211.33 mm, Max 828.5 mm) (Float64)
- **Flotation Column Level (3):** altura de la capa de burbujas en la parte superior de la celda de flotación 3, medido en mm. (min 127.1 mm, Max 886.7 mm) (Float64)
- **Flotation Column Level (4):** altura de la capa de burbujas en la parte superior de la celda de flotación 4, medido en mm. (min 162.7 mm, Max 678.5 mm) (Float64)
- **Flotation Column Level (5):** altura de la capa de burbujas en la parte superior de la celda de flotación 5, medido en mm. (min 167.2 mm, Max 674.0 mm) (Float64)
- **Flotation Column Level (6):** altura de la capa de burbujas en la parte superior de la celda de flotación 6, medido en mm. (min 159.87 mm, Max 698.57 mm) (Float64)
- **Flotation Column Level (7):** altura de la capa de burbujas en la parte superior de la celda de flotación 7, medido en mm. (min 177.47 mm, Max 656.7 mm) (Float64)
- **sin hora:** transformación trigonométrica ocupando el seno de la variable creada "hora". (min -1, Max 1) (Float64)
- **cos hora:** transformación trigonométrica ocupando el coseno de la variable creada "hora". (min -1, Max 1) (Float64)
- **sin_dia_de_la_semana:** transformación trigonométrica ocupando el seno de la variable creada "dia_de_la_semana". (min -0.974, max 0.974) (Float64)
- **cos_dia_de_la_semana:** transformación trigonométrica ocupando el coseno de la variable creada "dia_de_la_semana". (min -0.9, max 0.9) (Float64)

Variable Objetivo:

- **% Silica Concentrate:** Porcentaje de sílice en el concentrado al final del proceso de flotación (%), obtenido con análisis de laboratorio posterior. (min 0.6%, Max 5.53%) (Float64)

EDA:

Comenzare el EDA realizando Histogramas para entender/visualizar las distribuciones de mis variables (todas son variables continuas)

Se observa que varias tienen una distribución “más o menos” normal (ninguna tiene una distribución normal muy perfecta), Se identifica también que varias variables de “Air Flow” tienen distribuciones bastante feas, también se observa que las distribuciones de alimentación (**% Iron Feed y % Silica Feed**) tienen un comportamiento casi de espejo inverso.

Este comportamiento inverso tiene sentido (ya que mientras más % de hierro este presente en la alimentación, habrá menos volumen disponible para otros elementos como impurezas de sílice., y viceversa.)

Mi variable objetivo **% Silica Concentrate** pareciera tener una distribución parecida a una gamma, donde la mayoría de las observaciones son de bajo % de Sílice.

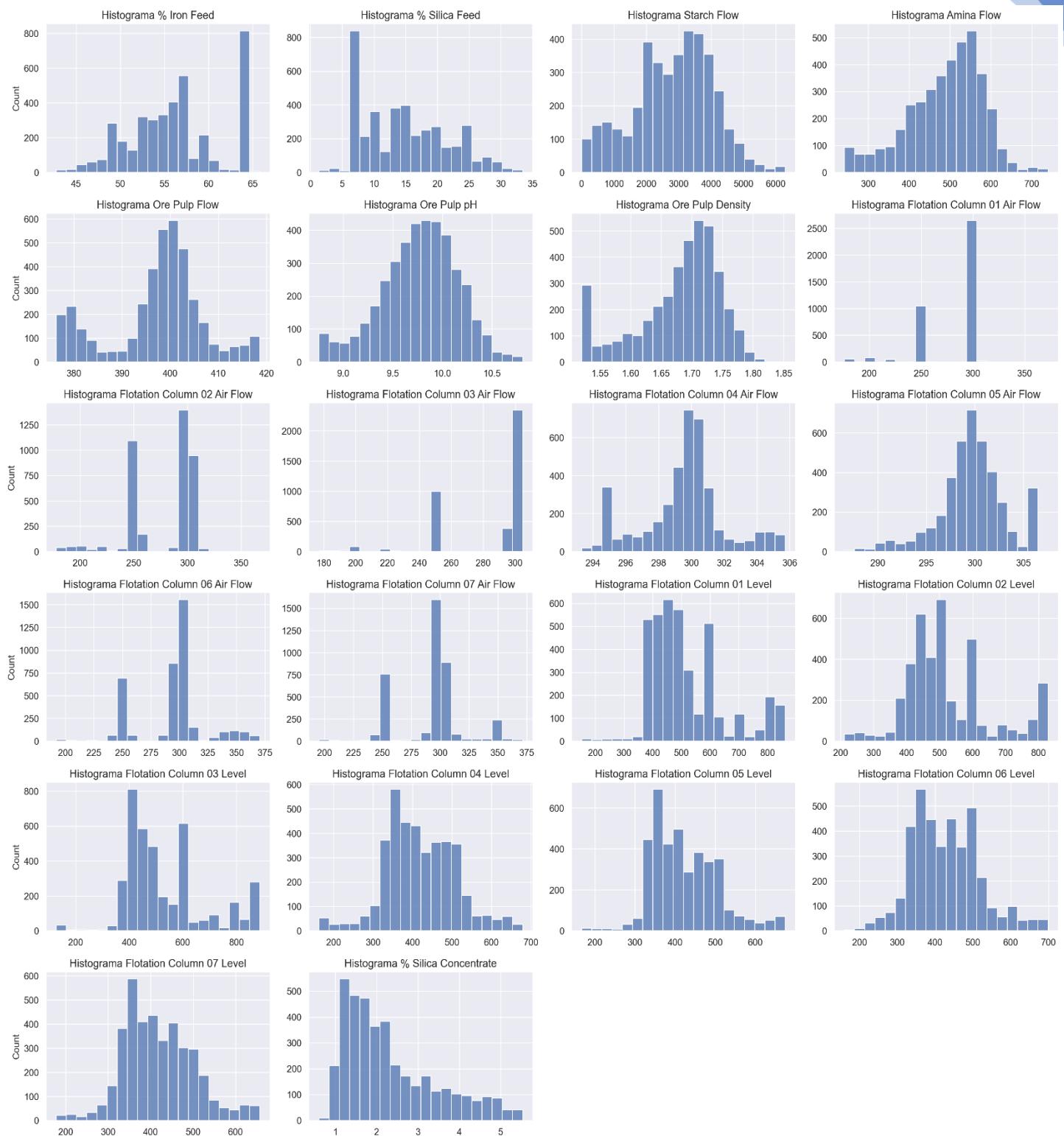


Figura 11: Histogramas de Variables Continuas.

En el siguiente gráfico, se observa visualmente a través de boxplots univariados escalados (usando StandardScaler) que la mayoría de mis variables no tiene outliers o tiene muy pocos, con la excepción de algunas columnas de **Flotation Columns Air Flow** y un poco la variable **Ore Pulp Flow**.

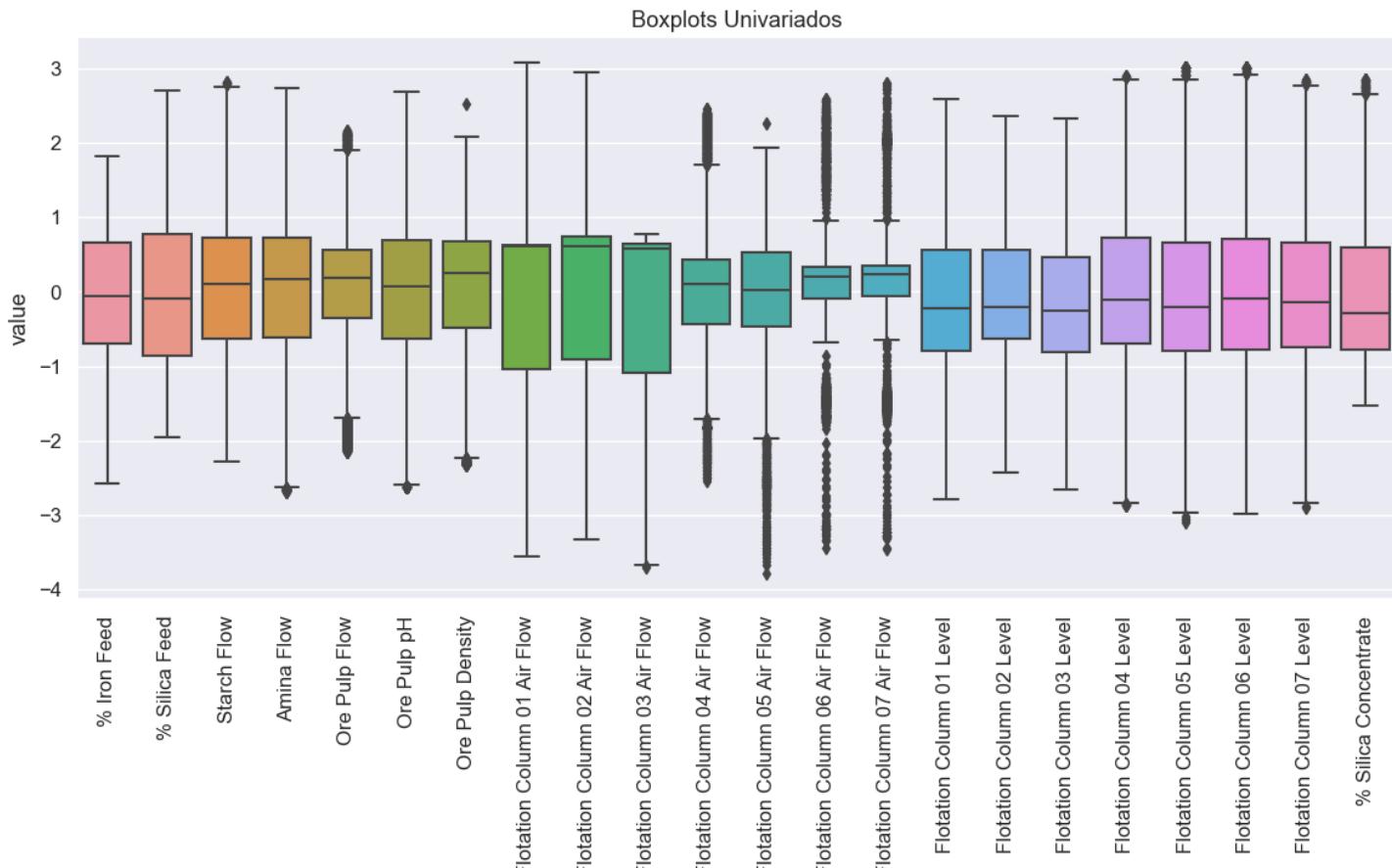


Figura 12: Boxplots univariados.

A continuación, se realizan Scatterplots entre mis variables de proceso (principalmente reactivos y características de la alimentación hacia mis celdas de flotación) y estos son coloreados según la variable objetivo **% Silica Concentrate**, se excluirán las variables "**Flotation Column Air Flow (1-7)** y **Flotation Column Level (1-7)**". Principalmente porque quedaría un PairPlot gigante y difícil de ver.

El único patrón claro y distinguible es qué **% Silica Feed** y **% Iron feed** tienen una relación lineal Inversa Fuerte

No se observa tampoco de forma evidente que estas combinaciones de variables (análisis bivariado) sean muy claras a la hora de separar el concentrado resultante con mayor o menor presencia de sílice.

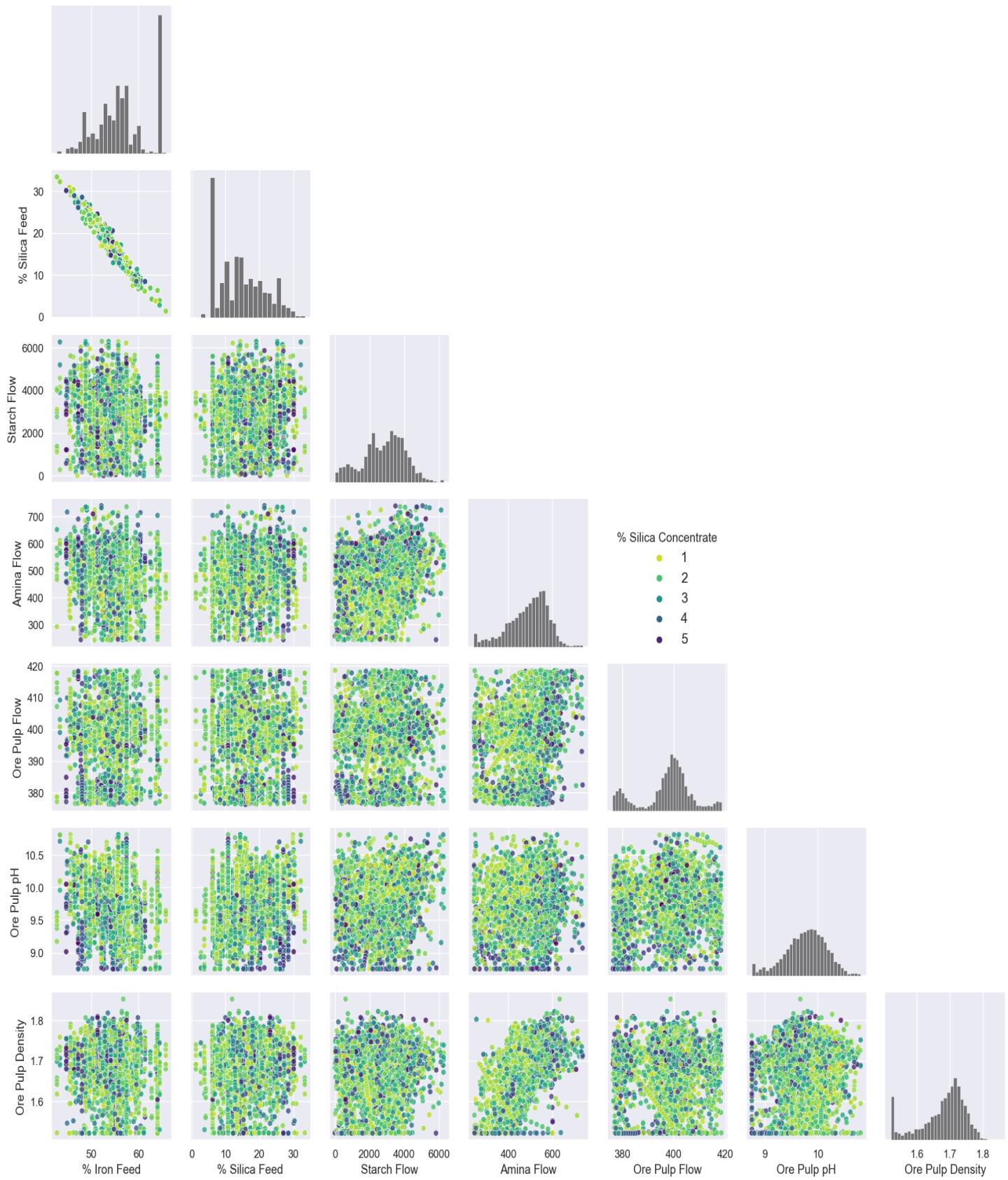


Figura 13: Scatterplots de variables de proceso, coloreados por la variable objetivo.

En el siguiente Grafico, observamos el comportamiento del % de sílice presente en la alimentación y el % de sílice presente en el Concentrado posterior al proceso de flotación (variable objetivo) a través del tiempo, y podemos comprobar la eficacia en general del proceso de flotación para reducir la impureza de sílice.

En promedio y porcentualmente se calculó que el proceso de flotación reduce en un 84.26% la impureza de sílice.

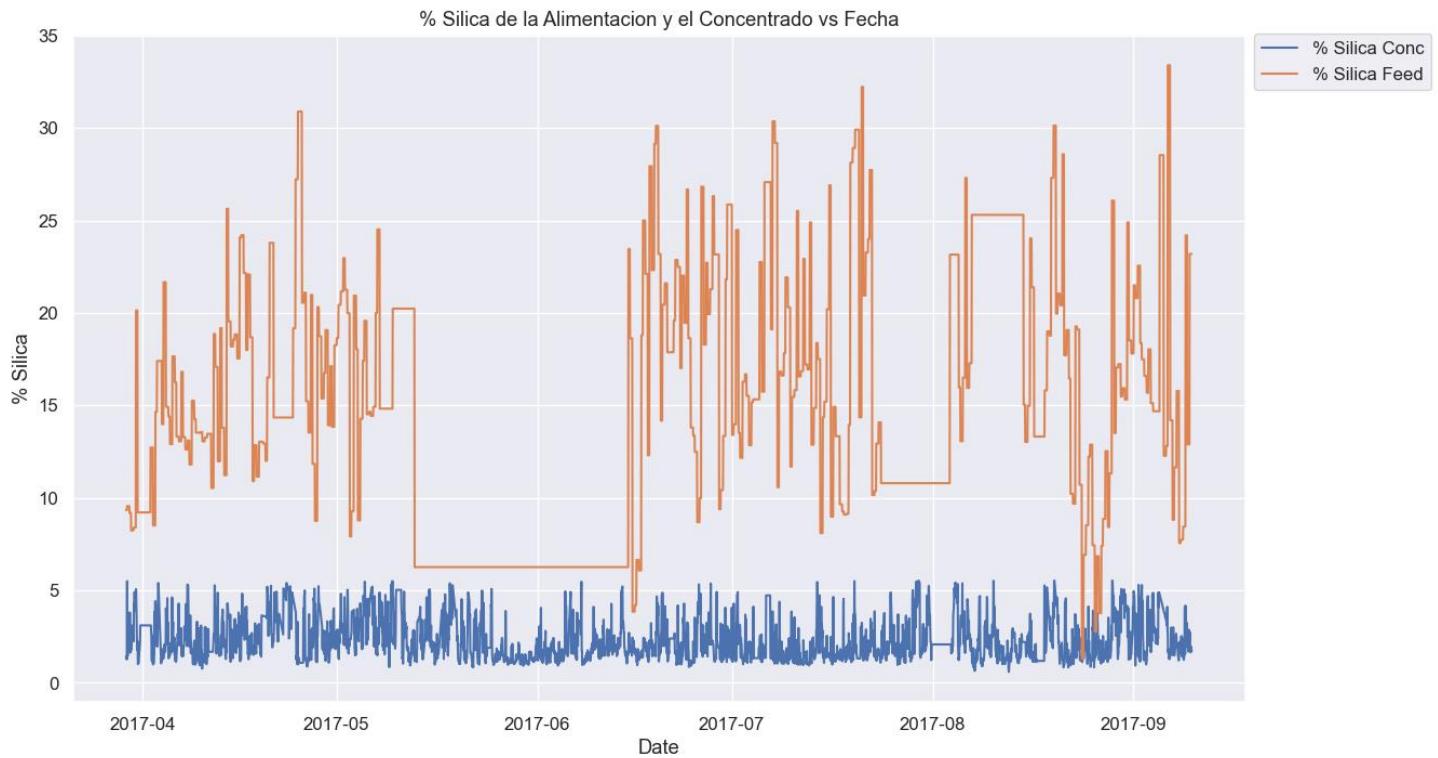


Figura 14: % Silica de la alimentación y el concentrado resultante vs fecha.

A continuación, se realiza un heatmap de correlaciones lineales.

No se observa ninguna correlación lineal significativa fuerte entre las variables explicativas y mi variable objetivo **% Silica Concentrate**, Ninguna pasa de +-0.25 esto no quiere decir que estas variables no tengan capacidad explicativa, pueden tener relaciones no lineales, relaciones con delay temporal o relaciones que en conjunto con otras si son relevantes.

La mayoría de las correlaciones lineales también son débiles entre las mismas las variables explicativas.

Se realiza otro heatmap donde se intenta distinguir si existe algún patrón visualmente reconocible entre los porcentajes de sílice del concentrado (mi variable objetivo) vs la hora y el día de la semana en la que se realizaron las mediciones. Visualmente no se puede apreciar ningún patrón significativo, pero da la pequeña impresión no concluyente de que en promedio la planta estaría logrando menores % de sílice en el concentrado, en la madrugada o en la tarde/noche, en comparación a medio día.

Visualmente tampoco se puede distinguir nada claro con los días de la semana.

Se utilizo la variable objetivo como valor dentro del heatmap y se estandarizo con la finalidad de ocupar un cmap divergente.

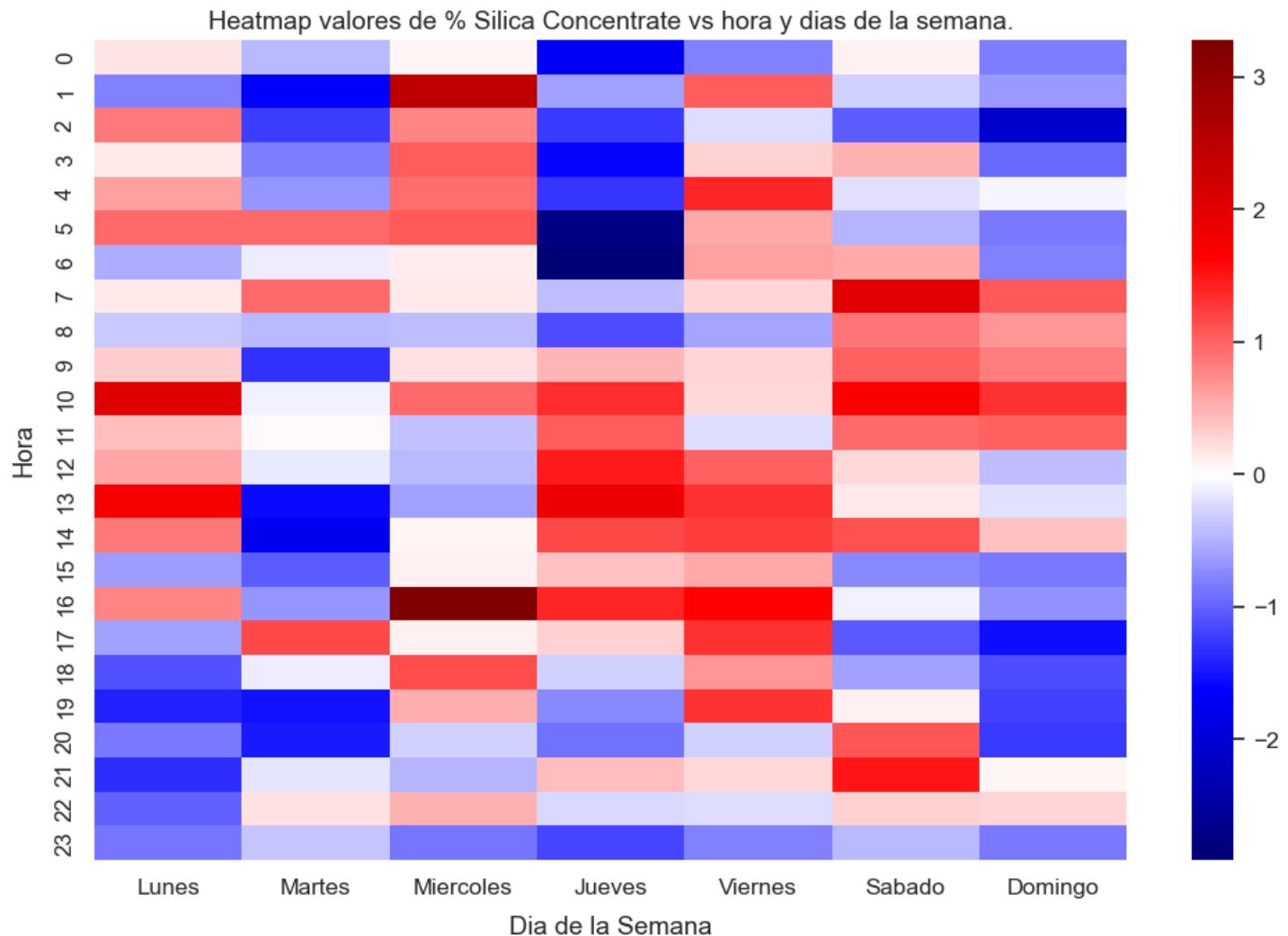


Figura 16: Heatmap variable objetivo vs hora y días de la semana.

Ahora Utilizare **DBSCAN** como algoritmo no supervisado de clustering y detección de Outliers, DBSCAN es un algoritmo de agrupamiento basado en densidad (density-based clustering). Si bien no pretendo eliminar outliers ya que eso rompería con la continuidad temporal de mi data y precisamente me interesa que mi modelo conozca esas observaciones, me parece relevante hacer un análisis visual y ver qué puntos el algoritmo identifica como outliers.

Es sabido que algoritmos basados en distancia & densidad sufren de la "maldición de la dimensionalidad" (Curse of dimensionality). Que implica en términos simples que métricas de distancia como por ejemplo la distancia euclíadiana, funcionan mal en datasets que tienen un alto número de features (columnas) y pocas samples en comparación (filas).

De acuerdo con Haohan Wang (Ph.D in Computer Science), la definición de "alta dimensionalidad" tiene una definición muy rigurosa, significa que en un dataset existen p (features) $>$ n (muestras), sin importar el tamaño de p o n . Teniendo en cuenta esto el dataset de este proyecto no tendría ese problema.

Posterior a correr DBSCAN en la data, ocupare el algoritmo de reducción de dimensionalidad no lineal **T-SNE**, usado principalmente por sus capacidades de visualización capaces de retener la estructura local de la data en lugar de la global, también usare PCA al final para comparar.

Como ya se mencionó, T-SNE tiene como objetivo preservar la **estructura local**. Esto implica que se preocupa más por preservar observaciones similares o cercanas entre ellas.

En T-SNE cuando se analiza la estructura local, hay que tener en cuenta que las distancias entre los distintos clusters formados por el algoritmo podrían no significar nada y que los "bounding boxes" o contornos de estos clusters tampoco.

En T-SNE una conclusión que se puede sacar es la siguiente. Los puntos dentro de los distintos clusters en el espacio de dimensionalidad reducida creado por T-SNE deben ser interpretados como puntos que son cercanos **localmente** en el espacio de dimensionalidad superior original del dataset, siempre y cuando no se exagere el hyperparametro "perplexity".

PCA trata de preservar la **estructura global** del dataset. El principal objetivo de PCA no consiste en preservar la distancia relativa entre los puntos, sino en preservar la varianza general a lo largo de los ejes. Se caracteriza por preservar observaciones muy distintas entre una y otra.

Hay que entender que en algoritmos de reducción de dimensionalidad siempre existe un “trade-off” entre estructura global y local.

Se seguirá el siguiente plan.

- Correr DBSCAN, sin eliminar a la variable objetivo. Ya que deseo identificar outliers y para eso debo considerarla también. (DBSCAN siempre se ejecutó en el espacio de dimensionalidad original, no en las versiones reducidas por T-SNE o PCA).
- Encontrar Hyperparametros óptimos para la detección de outliers en DBSCAN.
- Utilizar T-SNE, eliminando la variable objetivo antes de correrlo y realizar distintas visualizaciones con los embeddings creados. Se realizarán también mas gráficos con relación a outliers detectados.
- Experimentar volviendo a utilizar T-SNE pero con un valor del hyperparametro perplexity mucho más alto, algunos artículos señalan que incrementando este parámetro (de forma exagerada), causa que T-SNE empiece a capturar mayor estructura global y menos estructura local, un artículo se refiere a esto como que se “degrada en PCA”. Cabe destacar que T-SNE es una herramienta meramente exploratoria y que sus embeddings no pueden ser usados como inputs en un modelo de ML supervisado posterior, como si pudieran hacerlo los componentes principales de PCA.
- Utilizar PCA, a modo de comparación y ver su usabilidad en este data set en específico.

En estos dos gráficos, trato de encontrar un punto razonable de corte para el hyperparametro épsilon de DBSCAN, usando el método de “elbow”, observamos que un valor de 4 para este hyperparametro es una alternativa razonable.

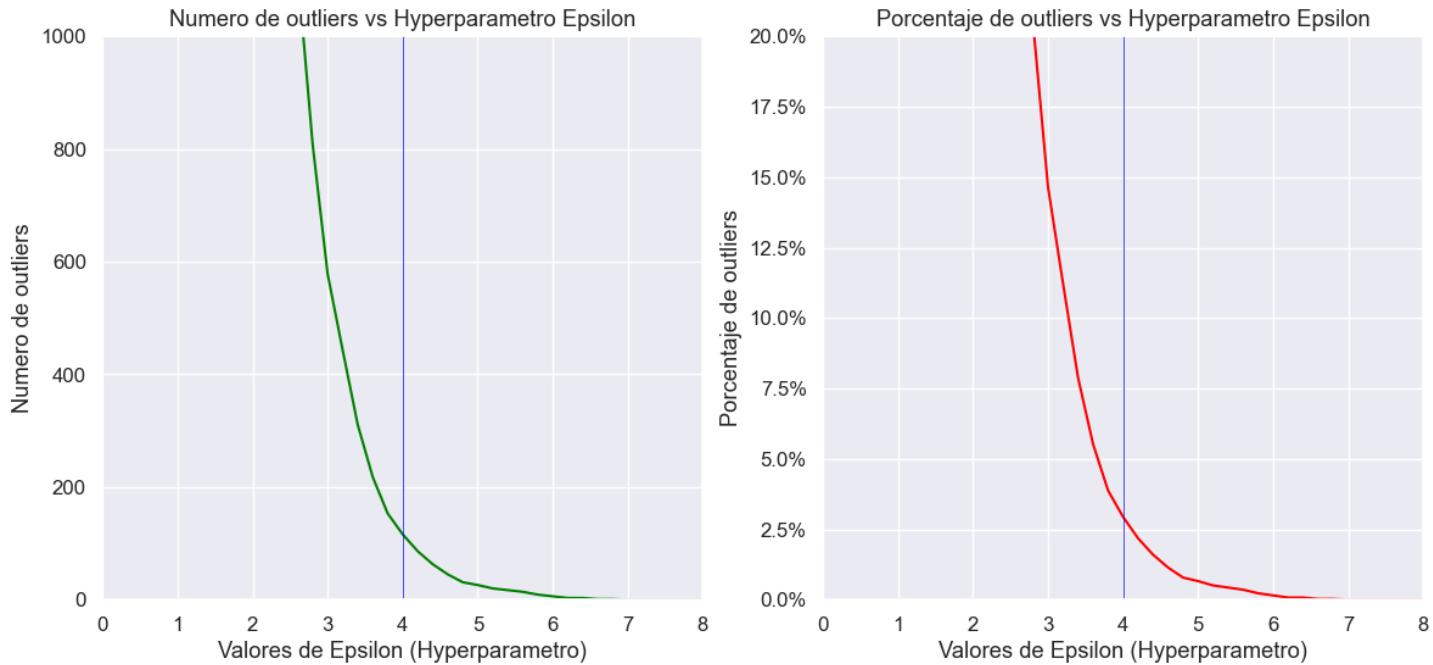


Figura 17: Gráficos de Elbow para obtener hyperparametro épsilon de DBSCAN.

En los siguientes gráficos se ejecuta T-SNE con un hyperparametro perplexity de 100, recordando que T-SNE con valores relativamente bajos de perplexity se enfoca en capturar **la estructura local**.

Los embeddings de T-SNE visualizados con gráficos a continuación no son particularmente fáciles de interpretar (y la interpretación puede ser totalmente erronea), se observan 3 clusters principales y varios mini clusters, los outliers identificados previamente por DBSCAN se encuentran principalmente rodeando o pegados a los diversos clusters en los contornos, cuando coloreamos el grafico con la variable objetivo se observa de forma parcial que existen zonas agrupadas más claras dentro de los mismos clusters (menores porcentajes de sílice en el concentrado) y zonas oscuras (mayores porcentajes de sílice en el concentrado). Esto podría ser una señal prometedora (o no), recordemos que T-SNE se ejecutó sin ver la variable objetivo.

En el gráfico de más abajo, observamos a la variable objetivo a través del tiempo y los outliers detectados por DBSCAN con marcadores rojos, vemos que hay una buena cantidad de outliers en

los distintos picos (valores altos) de la variable objetivo y también muchos en las zonas más bajas, recordar que DBSCAN si vio a la variable objetivo al ejecutarse.

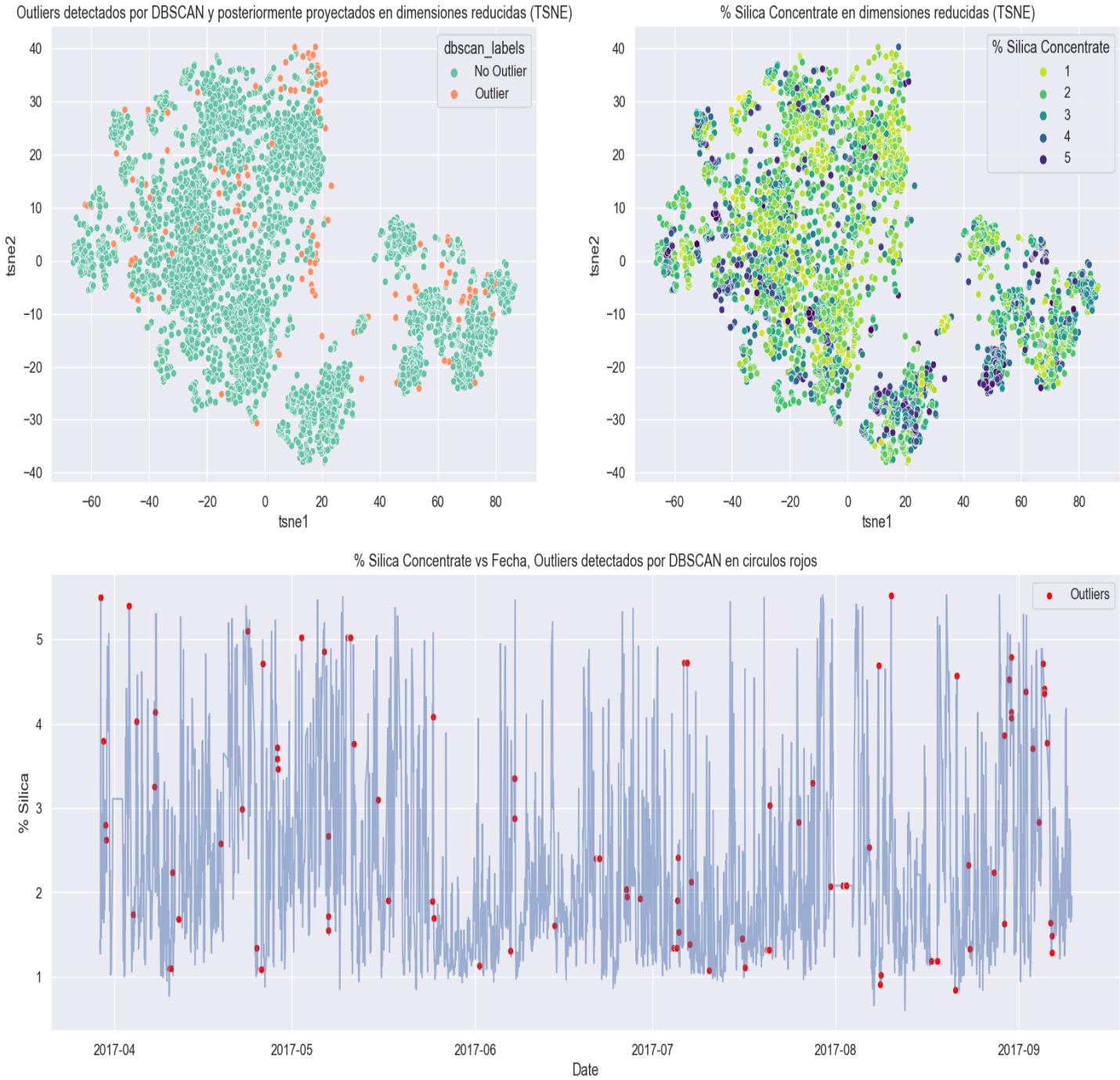


Figura 18: Gráficos de dimensionalidad reducida por T-SNE con un perplexity de 100, coloreados por outliers detectados por DBSCAN y variable objetivo (% Silica Concentrate), Grafico inferior representa variable objetivo a través del tiempo, con outliers detectados por DBSCAN en rojo.

Los histogramas a continuación confirman y manifiestan que los outliers detectados se encuentran mucho más cargados hacia ambos extremos de mi variable objetivo, en comparación a la distribución de los valores que no son outliers.

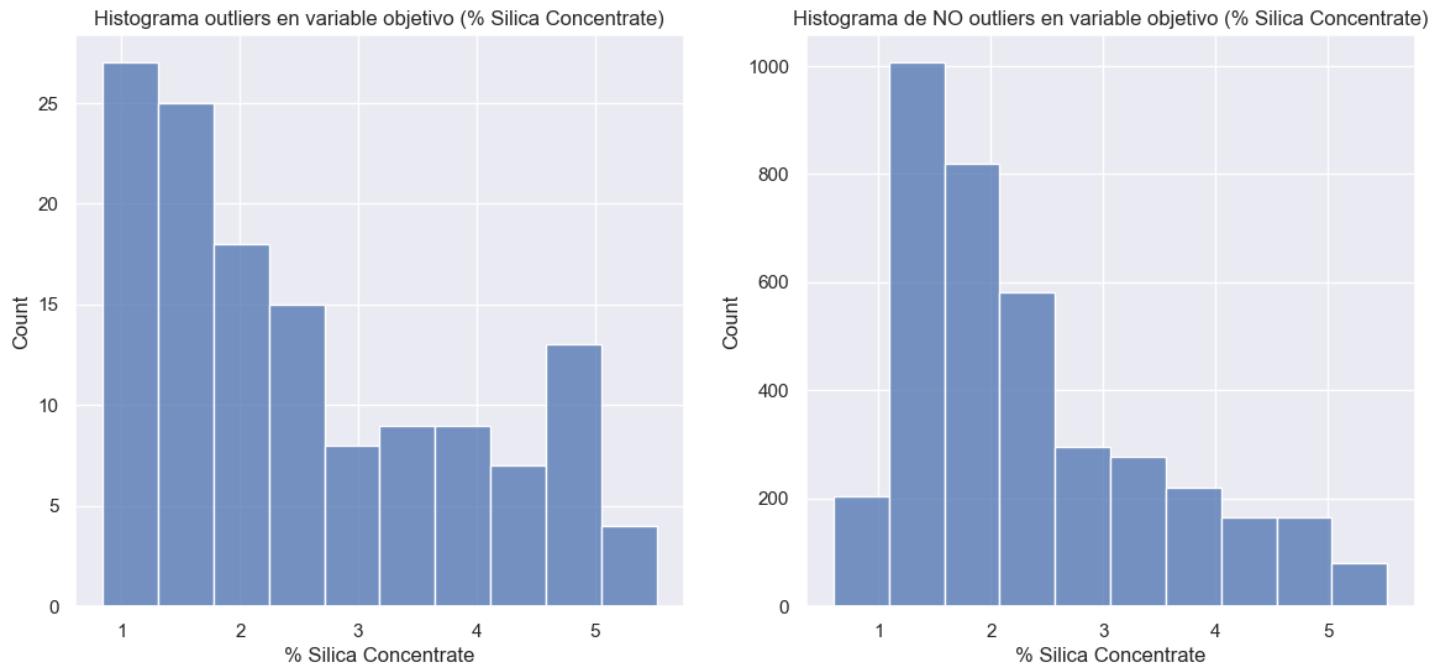
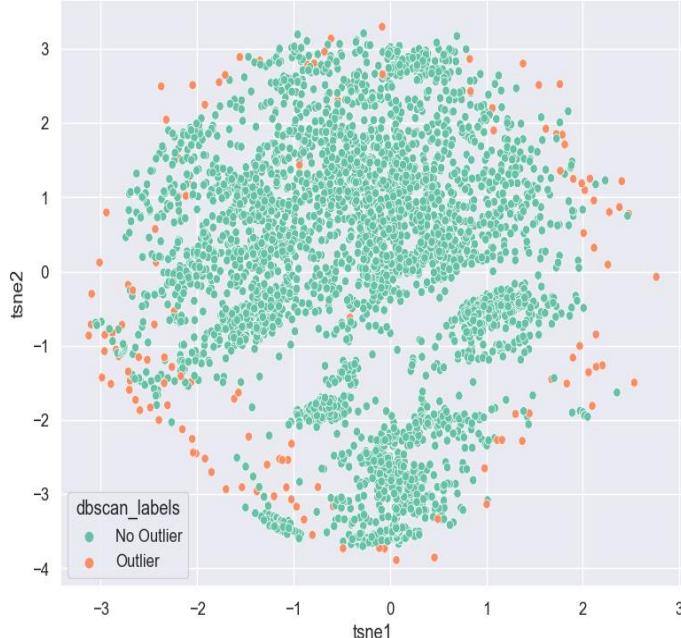


Figura 19: Gráficos de Elbow para obtener hyperparametro épsilon de DBSCAN.

A continuación, se vuelve a ejecutar T-SNE con un valor de perplexity de 2.000, esto es con la finalidad de que ahora el algoritmo se enfoque más en la **estructura global de la data**.

Se observa que los outliers aparecen mucho más distanciados del resto de las observaciones, prácticamente se encuentran rodeando el círculo central que representa la mayoría de la data, recordemos que T-SNE tampoco vio a la variable objetivo en esta ocasión, se sigue observando cierta separación parcial entre zonas claras y oscuras en la data, vemos también que ya no existen clusters tan bien definidos en comparación a cuando T-SNE se ejecutó con un valor de perplexity bajo.

Outliers detectados por DBSCAN y posteriormente proyectados en dimensiones reducidas (TSNE)



% Silica Concentrate en dimensiones reducidas (TSNE)

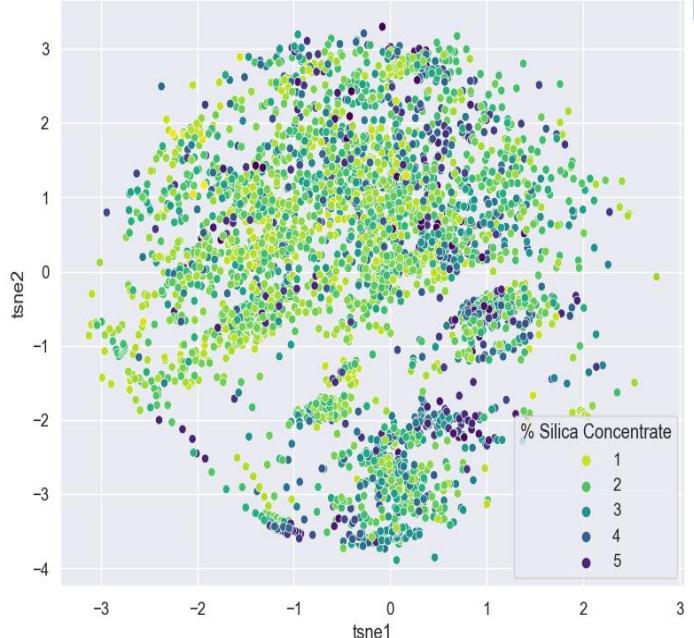


Figura 20: Gráficos de dimensionalidad reducida por T-SNE con un perplexity de 2000, coloreados por outliers detectados por DBSCAN y variable objetivo (% Silica Concentrate).

Para finalizar se ejecutó PCA, eliminando primero la variable objetivo al igual que en T-SNE y vemos que tiene un desempeño bastante malo en términos de varianza explicada (recordemos que PCA se enfoca en la **estructura global**), hacen falta 11 componentes principales para recién poder explicar un 80% de la varianza total del dataset, (tengo 25 columnas explicativas y 1 objetivo), esto puede ser atribuible a que las variables de mi dataset no tienen comportamientos muy lineales entre ellas y no hay muchas variables explicativas fuertemente correlacionadas que PCA pueda simplificar fácilmente en pocos componentes principales, también observamos que la visualización de los 2 primeros componentes principales no es la mejor ya que entre ambos solo logran un 34% de la varianza explicada, los outliers se encuentran mucho menos claros que en T-SNE con perplexity 2.000, PCA sin embargo sigue teniendo la ventaja de que los componentes principales pueden ser alimentados a los algoritmos supervisados de ML, mientras que los de T-SNE no pueden ya que siempre existe cierta distorsión en los embeddings.

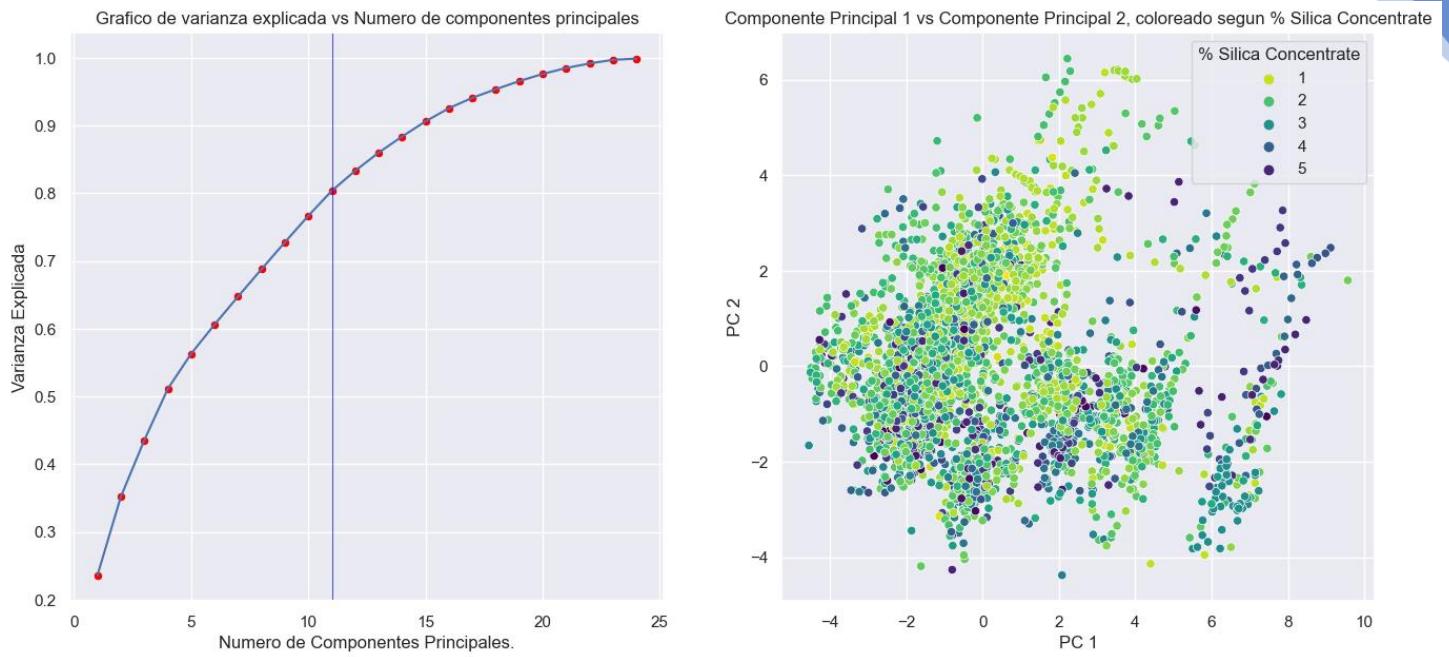


Figura 21: Gráfico de Componentes principales de PCA vs varianza explicada y Scatterplot de los 2 primeros componentes principales coloreado según la variable objetivo.

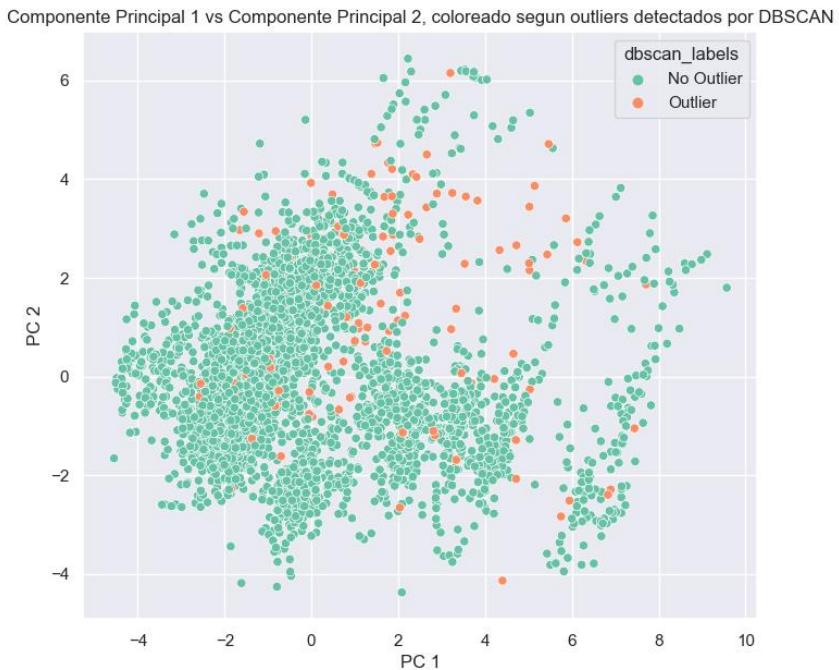


Figura 22: Scatterplot de los 2 primeros componentes principales coloreado según la outliers detectados por DBSCAN.

Conclusiones hasta el momento:

- El preprocessamiento fue particularmente desafiante y distinto a lo que me he encontrado hasta ahora con la poca experiencia que tengo, sin embargo, creo que lo aborde de una forma correcta y coherente dentro de lo posible, se recuerda que se descubrió cierta “corrupción” en el dataset que no puede ser solucionada.
- El EDA no entrega información clara a la hora de identificar que variables explicativas podrían ser más relevantes a la hora de obtener concentrados de mayor o menor % de sílice, el heatmap de correlaciones lineales muestra que todas las variables explicativas tienen una correlación baja con mi variable objetivo y los scatterplots entre las distintas variables explicativas coloreados según la variable objetivo tampoco son claros. Se observa que hay algunas variables explicativas que tienen distribuciones no muy gaussianas que podrían ser candidatas a alguna transformación más adelante si es que ocupo algún modelo supervisado que tenga algún supuesto fuerte de distribución normal en las variables explicativas de entrada. El EDA quizás se ve un poco pobre ya que no tengo variables categóricas, pero realice todos los análisis para variables continuas vistos en el módulo de visualización de datos (y más).
- Hay algunas pocas variables explicativas que podrían ser candidatas a eliminación ya que están fuertemente correlacionadas linealmente como lo son % Iron Feed y % Silica Feed. También hay correlaciones altas en algunas de Air Flow y Column level. Sin embargo, prefiero esperar hasta la parte de aprendizaje supervisado para ocupar técnicas de selección de variables como RFE (Recursive Feature Elimination) o tomar la decisión de eliminar alguna variable posterior a tener un modelo supervisado ejecutado, viendo información como por ejemplo la “feature importance” de un modelo de árboles y/o utilizando la librería de explicabilidad de modelos “SHAP”.
- El Análisis de aprendizaje no supervisado fue bastante interesante, creo que el punto más relevante fue la detección, visualización y comprensión de la distribución de los outliers detectados por DBSCAN.

ENTREGA 3:

Antes comenzar esta sección, es necesario recordar que durante el preprocesamiento se detectaron inconsistencias en el dataset que no pueden ser solucionadas, se reitera que los puntos más preocupantes son la evidencia de interpolación en algunos tramos de la variable objetivo (demostrada por los registros de actualización de esta muy por encima de su frecuencia habitual de 1 hora antes del resampleo), también se evidencio otro problema donde registros de la variable objetivo no se actualizan respecto a la medición temporalmente anterior (1 hora atrás o mas).

Esto puede implicar que nos enfrentamos a un dataset “corrupto” y puede ser difícil que nuestras variables explicativas logren “explicar” de buena forma la variable objetivo ya que posiblemente los distintos modelos se encuentren con señales o patrones contradictorios a la hora del entrenamiento.

Esperamos que nuestro modelo ingenuo sea el “sanity check” a la hora de enfrentarnos a estos problemas mencionados, si nos encontramos con un dataset corrupto es probable que el modelo ingenuo no pueda ser vencido o que este sea vencido por muy poco.

Modelamiento:

Se realizarán una serie de experimentos de modelamiento para afrontar la problemática de predecir/estimar la variable objetivo (**% Silica Concentrate**).

Se realizará un train/test split del 90% de la data, recordar que para problemáticas de series de tiempo el split debe respetar la temporalidad (no aleatorio).

Se usará el TimeSeriesSplit Cross-Validator de SkLearn en la data de entrenamiento (Train) a modo de validar los distintos modelos y encontrar hyperparametros. (excepto para red neuronal).

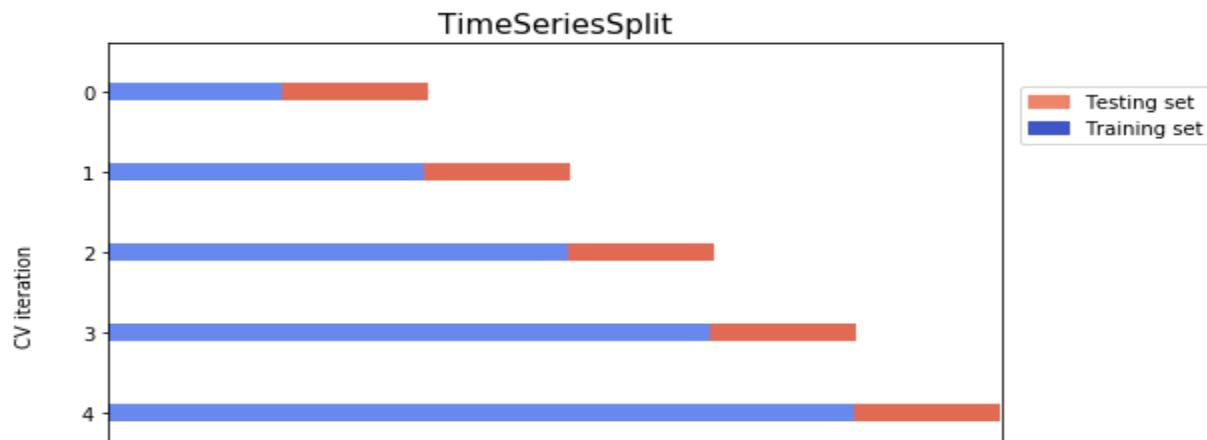


Figura 23: Grafico que detalla cómo funciona el TimeSeriesSplit Cross-validator de SkLearn.

Las métricas consideradas de mayor relevancia para evaluar los experimentos serán el **MASE** y el **RMSE**:

Un mean absolute scaled error (**MASE**), mayor a 1, implica que las predicciones del modelo no son mejores que las que estaría realizando un modelo ingenuo de series de tiempo (se define mas adelante cual es el modelo ingenuo), mientras que para valores menores a 1, estaría realizando mejores predicciones en comparación al modelo ingenuo.

$$MASE = \frac{MAE}{MAE(naive)}$$

Se identifica también como métrica de desempeño relevante la raíz de los errores cuadráticos medios (**RMSE**), ya que esta sobre penaliza los errores grandes por su componente cuadrático, la naturaleza del problema en cuestión implica la necesidad de reducir errores de magnitudes grandes para nuestra variable objetivo (% Silica Concentrate).

$$RMSE = \sqrt{\frac{\sum_{i=1}^N (Actual - Predicted)^2}{N}}$$

El horizonte de predicción (forecast horizon) de todos mis modelos es de 1.

No se utilizarán modelos estadísticos clásicos de series de tiempo como holtz-winters o arima/arimax, debido a que no soportan variables explicativas adicionales a la misma variable objetivo y/o ofrecen nula o casi nula interpretabilidad.

Se utilizará Optuna (búsqueda bayesiana), para encontrar Hyperparametros en los distintos experimentos (excluyendo Red Neuronales).

Las métricas de desempeño de cada uno de los modelos (experimentos) serán adjuntadas al final en una tabla.

Los experimentos serán los siguientes:

- **Experimento 0:** Modelo Ingenuo, el modelo ingenuo en problemáticas de series de tiempo (sin estacionalidad) consiste en seleccionar el ultimo valor real de la serie (observación) y usar a esta misma como el forecast del siguiente valor.

$$\text{Forecast}(t + 1) = \text{Observation}(t)$$

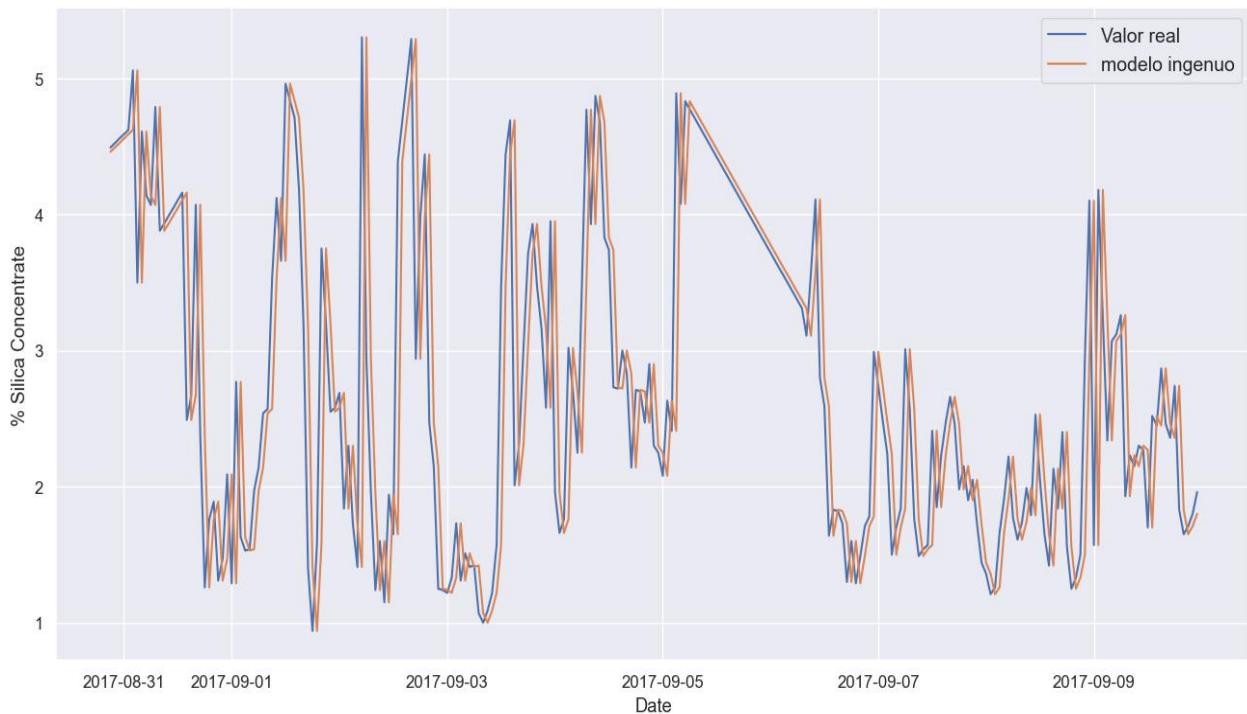


Figura 24: Grafico predicciones modelo ingenuo vs test set.

- **Experimento 1:** LightGBM regressor, como variables explicativas se utilizará solo la variable objetivo lageada temporalmente, con un tamaño de ventana de 3 (**Window size**), no se usarán otras variables explicativas.

Por ende, en este experimento las variables explicativas serían 3. (variable objetivo retrasada 1 hora, 2 horas y 3 horas respectivamente).

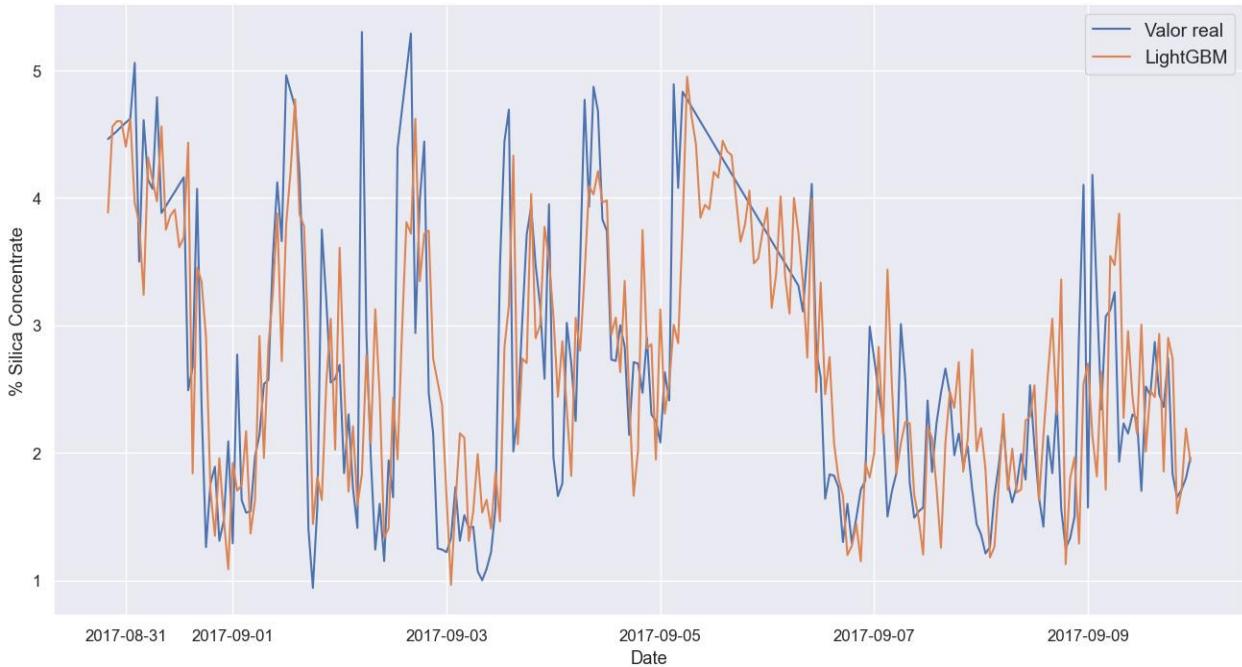


Figura 25: Grafico predicciones experimento 1 vs test set.

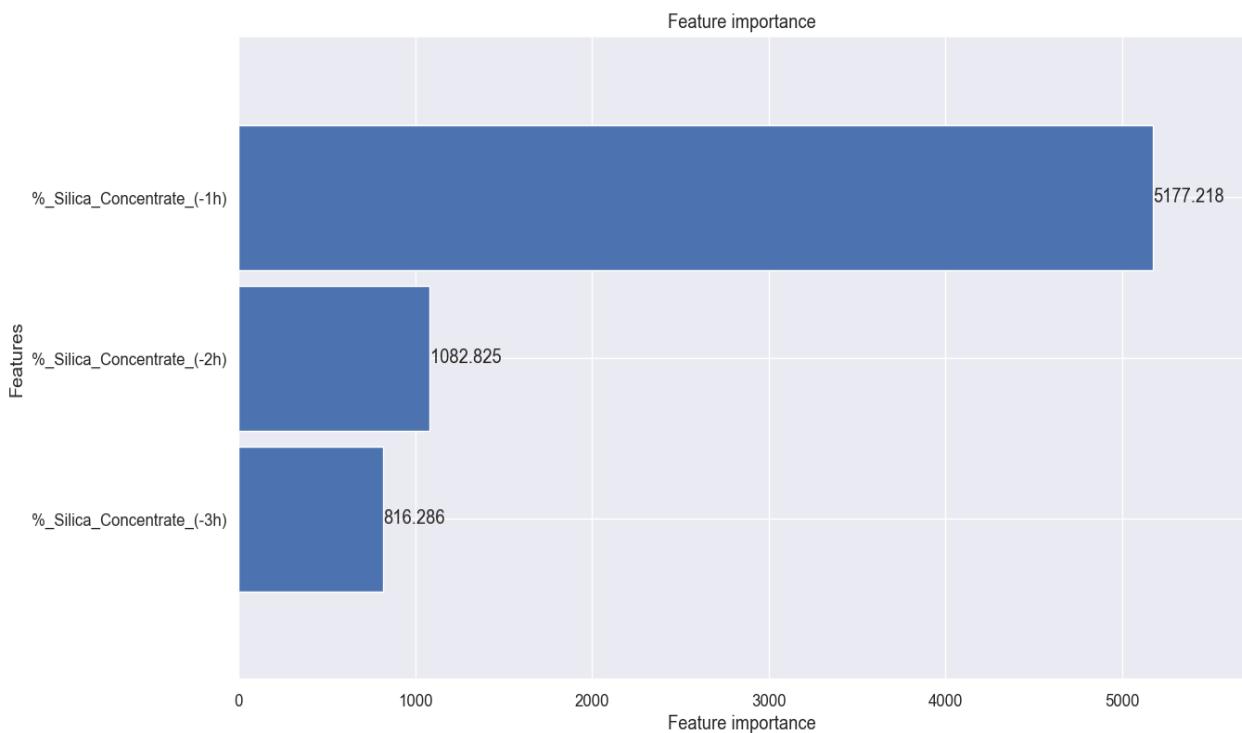


Figura 26: Tabla de importancia de variables explicativas experimento 1

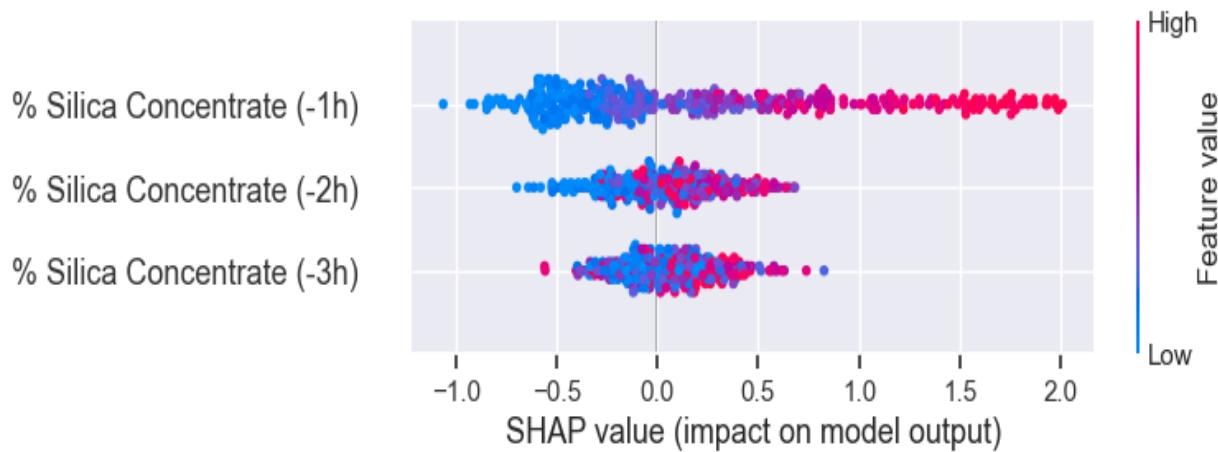


Figura 27: Grafico de valores SHAP, que identifican variables explicativas con mayor impacto, Experimento 1.

- **Experimento 2:** LSTM recurrent neural Network, utilizare una red neuronal recurrente con una layer "Long short term memory" (**LSTM**), son conocidas por ser bastante buenas en problemáticas de series de tiempo **univariadas y multivariadas** donde existen patrones complejos. Se utilizarán las mismas variables explicativas que el experimento 1.

Por ende, en este experimento las variables explicativas serian 3.



Figura 28: Grafico predicciones experimento 2 vs test set.

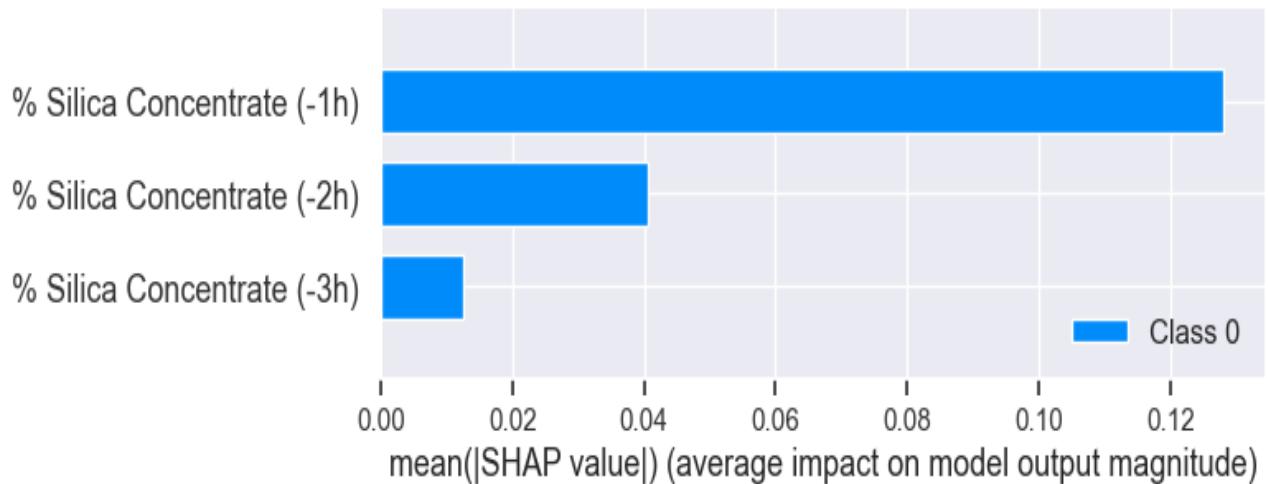


Figura 29: Grafico de valores SHAP, que identifican en promedio el impacto de cada una de las variables explicativas, Experimento 2. (Similar a feature importance).

- **Experimento 3:** Lasso regression, la regresión lineal múltiple es bastante útil debido a sus bondades de explicabilidad, a esta tambien se le pueden agregar distintas técnicas de regularización, que ayudan a combatir el overfitting, conseguir un mejor ajuste o incluso seleccionar las variables explicativas con comportamientos lineales más relevantes y/o significativos mientras que los coeficientes de las variables irrelevantes se hacen 0, este es el caso de la **regresión lineal múltiple con regularización Lasso**. Se ocuparán todas las variables explicativas del dataset para este experimento, incluyendo la variable objetivo lageada con un window size de 3, también usando un poco de manipulación del dataset pre-resampleo, podemos agregarle a la data mediciones de variables explicativas retrasadas en algunos minutos (Se incluirán variables explicativas retrasadas 15min, 30min, y 45min). Recordemos que esto es factible ya que muchas variables explicativas tenían una frecuencia de actualización mucho mayor en comparación a la variable objetivo.

Considerando lo anterior la cantidad de variables explicativas de este experimento es de 91.



Figura 30: Grafico predicciones experimento 3 vs test set.

Observamos que de 91 variables explicativas, Lasso descarto la gran mayoría, dejando solo 5 con coeficientes distintos a 0, las variables explicativas fueron escaladas antes de ejecutar la regresión por lo que los coeficientes representan la importancia relativa de cada variable con respecto a la predicción de la variable objetivo, (más importantes mientras más distintas de 0).

Se observa que los valores lageados de la variable objetivo son los variables explicativas de mayor importancia por lejos.

	Valores
% Silica Concentrate (-1h)	0.625229
% Silica Concentrate (-2h)	0.145222
% Silica Concentrate (-3h)	0.062978
Flotation Column 05 Level (-15mins)	-0.002351
Flotation Column 01 Air Flow	-0.031177

Figura 31: Coeficientes (betas), experimento 3 (Lasso).

Visualmente en el siguiente grafico de predicciones vs residuos, se observa heterocedasticidad, violándose a mi parecer supuesto de homocedasticidad de la regresión lineal.

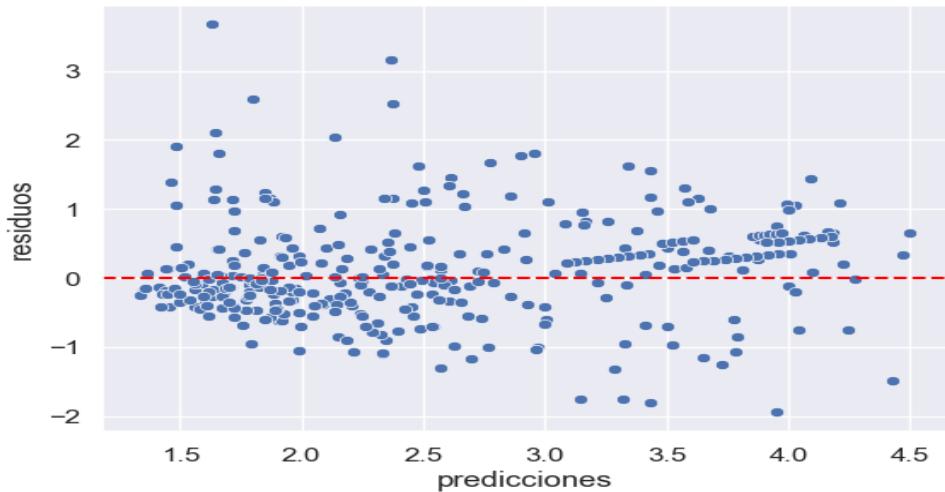


Figura 32: Grafico predicciones vs residuos, Lasso Regression (experimento 3).

- **Experimento 4:** LightGBM regressor. se utilizan las mismas variables explicativas del experimento 3.

Considerando lo anterior la cantidad de variables explicativas de este experimento es de 91.

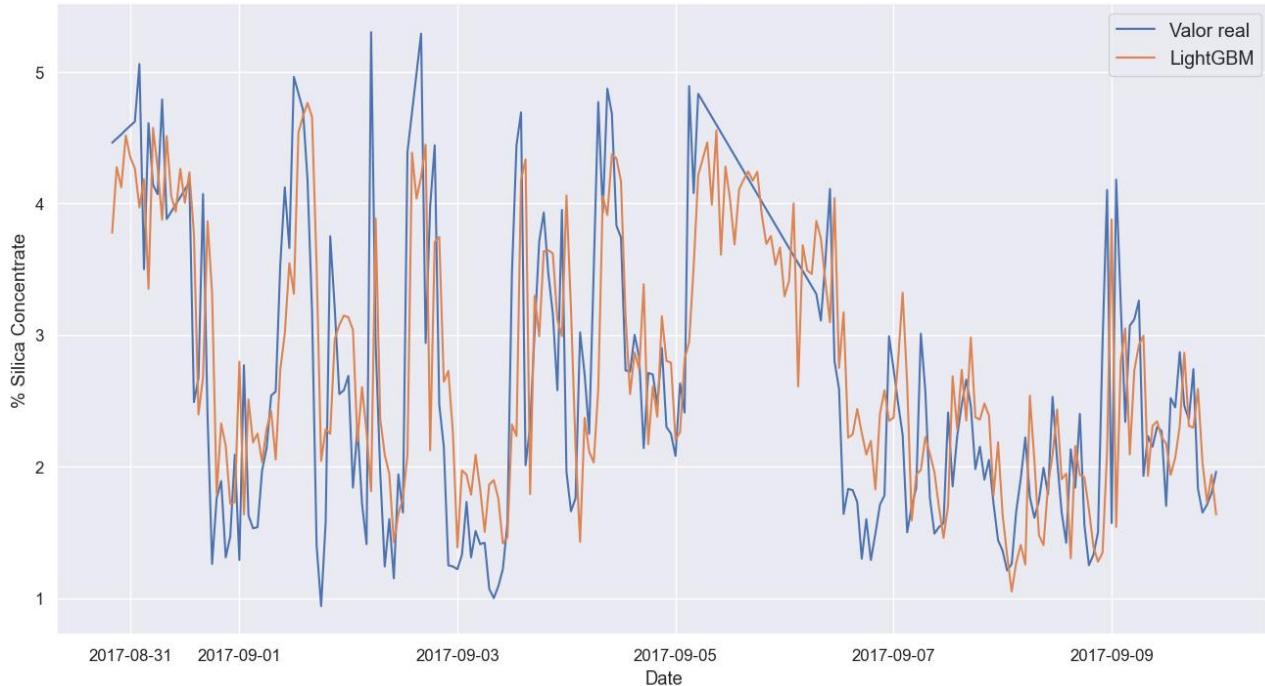


Figura 33: Grafico predicciones experimento 4 vs test set.

Se observa que el modelo de Gradient boosting (LightGBM) da muy poca importancia a prácticamente todas las variables explicativas y se concentra en los valores lageados de la variable objetivo.

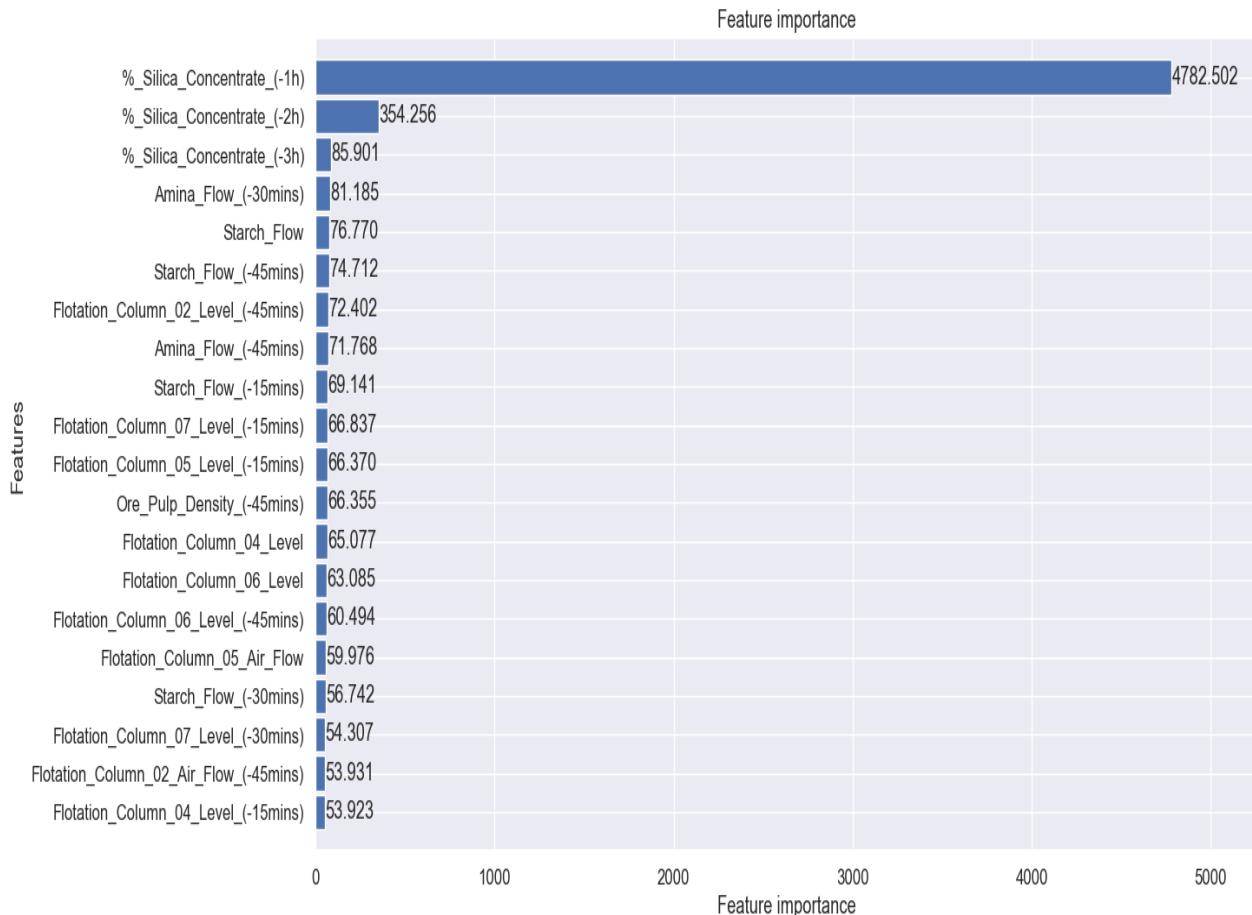


Figura 34: Tabla de importancia de variables explicativas experimento 4

- **Experimento 5:** LSTM recurrent neural network, a modo de experimentación se utilizarán las variables cuyos coeficientes el experimento 3 (Lasso Multiple linear regression) no transformo en 0.

Considerando lo anterior la cantidad de variables explicativas de este experimento es de 5.

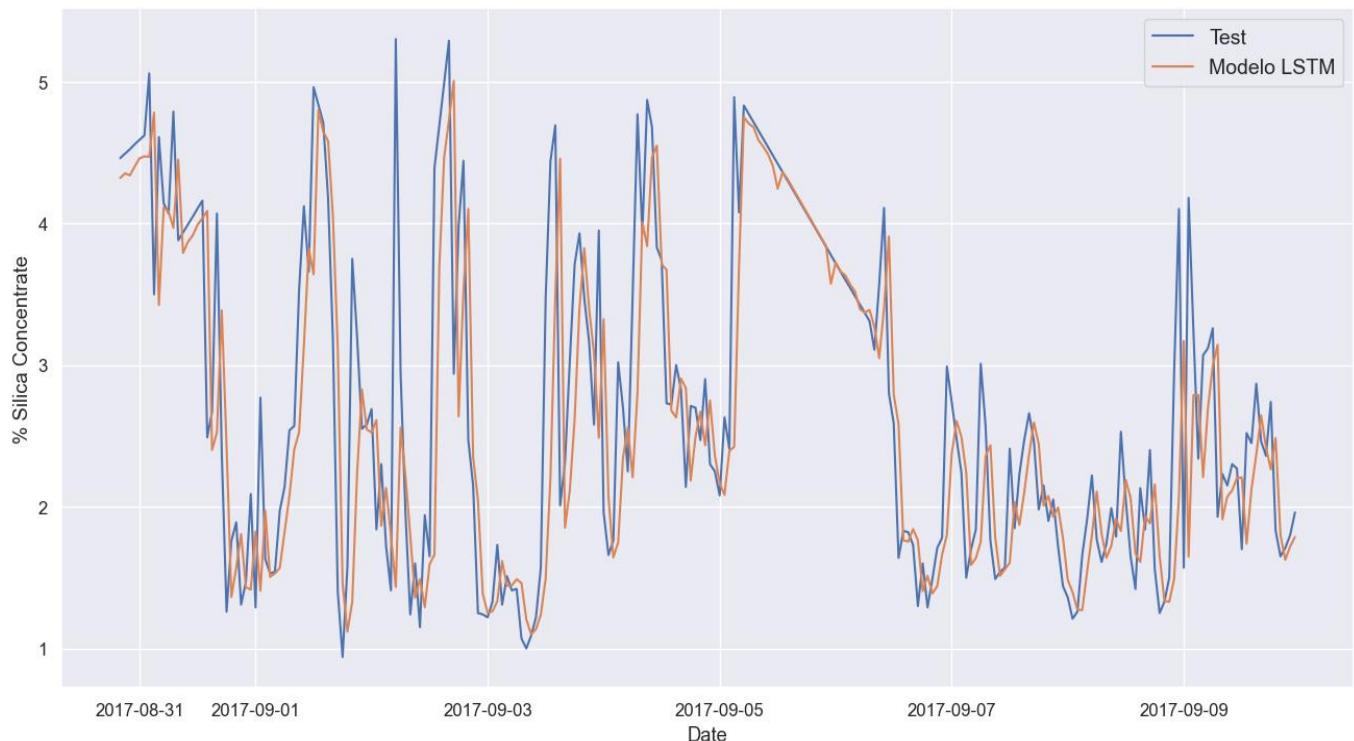


Figura 35: Grafico predicciones experimento 5 vs test set.

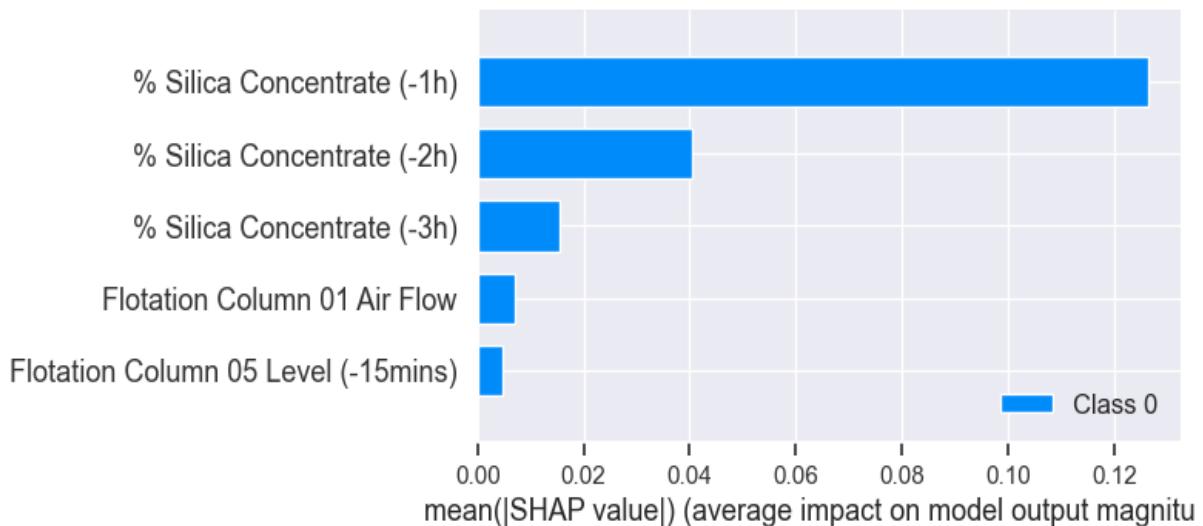


Figura 36: Grafico de valores SHAP, que identifican en promedio el impacto de cada una de las variables explicativas, Experimento 5. (Similar a feature importance).

- **Experimento 6:** LSTM recurrent neural network. Se utilizarán todas las variables explicativas. Considerando lo anterior la cantidad de variables explicativas de este experimento es de 91.

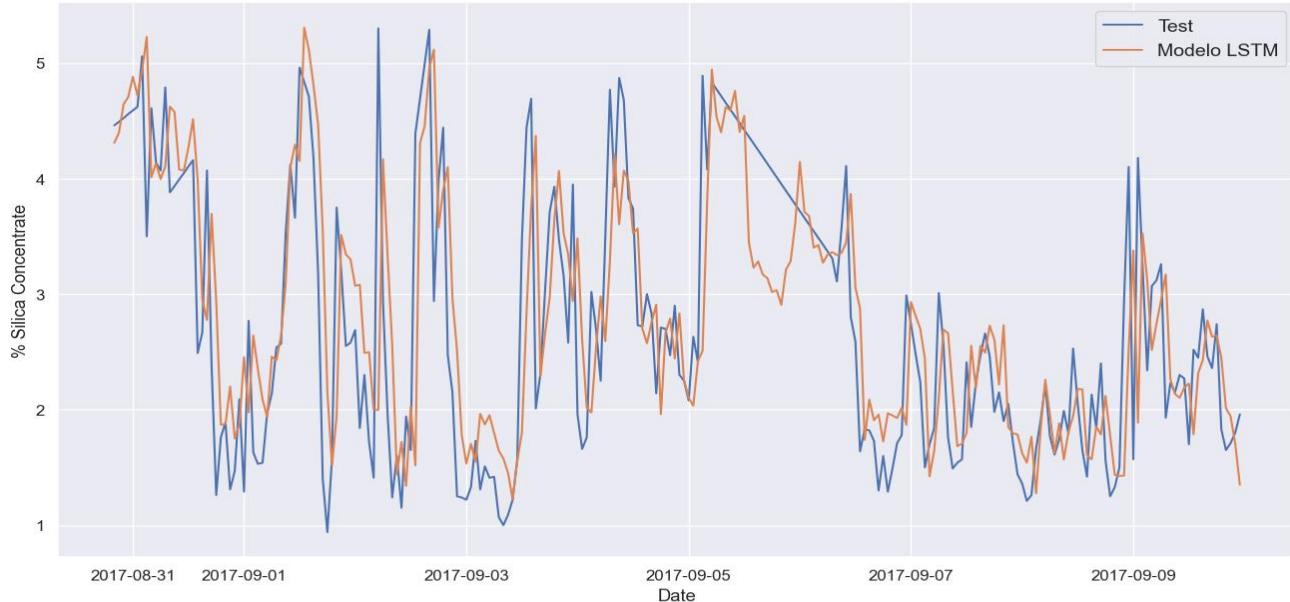


Figura 37: Grafico predicciones experimento 5 vs test set.

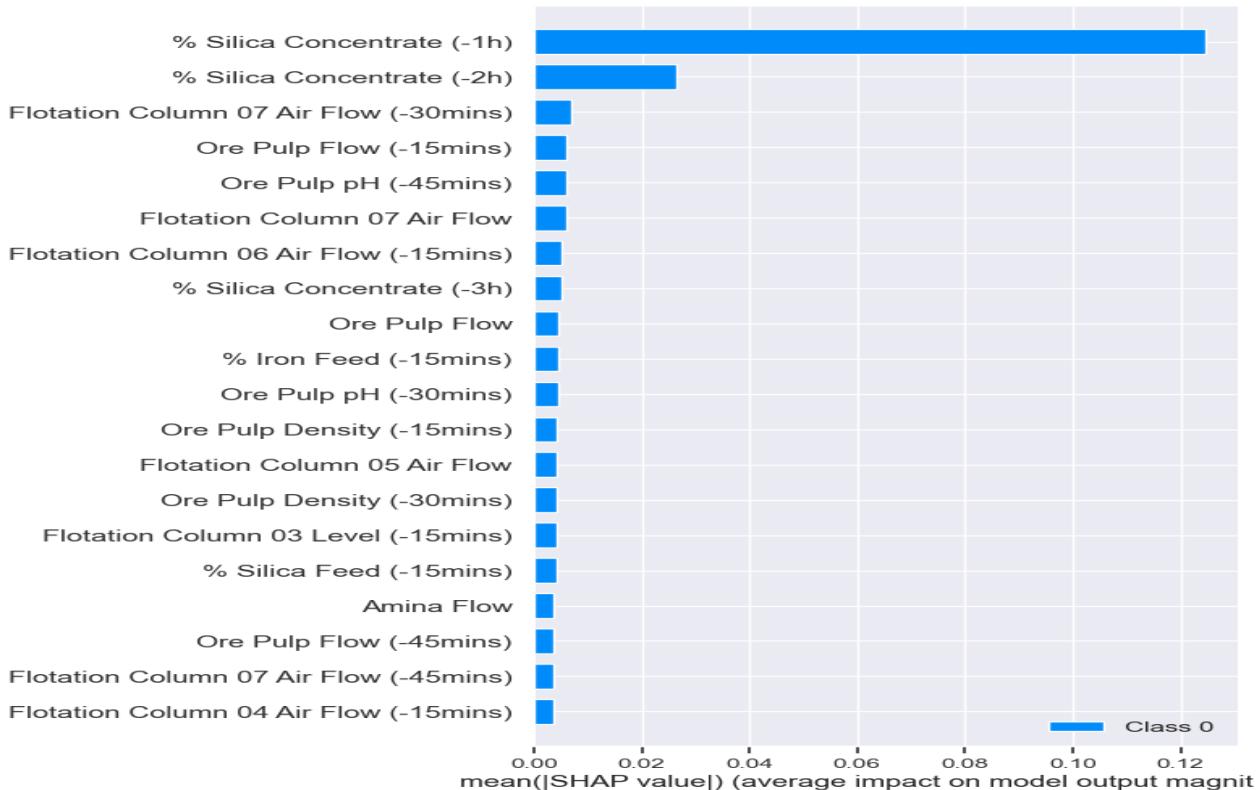


Figura 38: Grafico de valores SHAP, que identifican en promedio el impacto de cada una de las variables explicativas, Experimento 6. (Similar a feature importance).

Métricas de los experimentos y comentarios:

Observamos que los únicos experimentos que vencen al modelo ingenuo en prácticamente todas las métricas son las redes neuronales recurrentes con layer LSTM (experimento 2 y 5) incluyendo el **MASE**, la regresión Lasso (experimento 3) fue la que mejor **RMSE** y **R2** obtuvo, aunque pierde en las demás métricas frente a los experimentos 2, 5 y el modelo ingenuo. Se observa que los experimentos vencen al modelo ingenuo por muy poco.

Los modelos no dan importancia a casi TODAS las variables explicativas y solo se enfocan en la variable objetivo ladeada como variable explicativa más relevante, como se evidencia en los distintos gráficos de interpretabilidad. De hecho, los modelos que no incluyen o descartan el resto de las variables como los experimentos 2, 3 o 5, son a los que mejor les ha ido. Ya que probablemente suprimen ruido.

Esto confirma a mi parecer que el dataset se encuentra corrupto y no se puede obtener información útil de las variables explicativas para predecir la variable objetivo.

Creemos (según la evidencia expuesta en el preprocessamiento) que la variable objetivo en sí misma es la fuente principal de la corrupción y del fracaso de las variables explicativas en aportar valor predictivo a los distintos modelos para poder predecirla.

	model_name	MAE	RMSE	MASE	R2	MAPE
0	Modelo Ingenuo	0.5299	0.8021	1.0000	0.5396	0.2216
1	Experimento 1, LightGBM	0.6194	0.8355	1.1689	0.4991	0.2652
2	Experimento 2, LSTM (pocas variables)	0.5050	0.7683	0.9530	0.5764	0.2006
3	Experimento 3, Lasso_lin_reg	0.5482	0.7532	1.0346	0.5929	0.2243
4	Experimento 4, LightGBM	0.6320	0.8404	1.1928	0.4932	0.2775
5	Experimento 5, LSTM	0.5041	0.7732	0.9514	0.5710	0.1988
6	Experimento 6, LSTM	0.5776	0.7900	1.0901	0.5522	0.2547

Figura 39: Metricas de desempeño de los distintos experimentos.

Conclusiones y próximas etapas:

Es un poco desilusionante que esta sea la conclusión de este trabajo, pero a mi parecer es la conclusión es correcta, **el dataset se encuentra corrupto y no se pueden realizar análisis sobre este para responder a las distintas preguntas de valor.**

Es curioso que en Kaggle nadie se haya dado cuenta de esto, aunque sea un dataset muy poco conocido/utilizado, a nadie se lo ocurrió utilizar un modelo ingenuo para comparar los modelos más complejos que elaboraron, la mayoría ni siquiera considero el factor temporal para realizar su análisis, también nadie o casi nadie detecto los distintos problemas encontrados en la etapa de preprocesamiento de este trabajo.

La misma persona que posteó el dataset comentó ingenua y equivocadamente que un MAE de 1.0 +- 0.2 sería un resultado satisfactorio, pero como hemos demostrado un simple modelo ingenuo es capaz de vencer eso y por mucho.

Los únicos que han llegado a una conclusión más o menos similar (aunque distinta) a la mía sobre la baja calidad de este dataset son las personas que hicieron el siguiente artículo.

<https://techlabs-aachen.medium.com/quality-prediction-in-a-mining-process-1a2b70b51303>

Las próximas etapas serían subir el notebook del proyecto a Kaggle, traducir y agregar comentarios en inglés y sugerir eliminar el dataset porque este se encuentra corrupto o mantenerlo como ejemplo de un dataset corrupto. También sería interesante ver el Feedback de las demás personas.

Bibliografía:

LINKS TEORIA Y JUSTIFICACION FLOTACION:

<https://www.911metallurgist.com/metalurgia/beneficio-mineral-hierro/>

<https://tools.thermofisher.com/content/sfs/posters/Infographic-Penalty-Elements-In-Iron.pdf>

<https://www.spglobal.com/commodityinsights/en/market-insights/latest-news/metals/062317-analysis-iron-ore-impurities-penalties-surge-on-resurgence-of-varied-supply>

[https://www.researchgate.net/figure/Flowsheet-of-iron-ore-flotation-process-with-zoom-on-single-flotation-cell fig1 339239261](https://www.researchgate.net/figure/Flowsheet-of-iron-ore-flotation-process-with-zoom-on-single-flotation-cell_fig1_339239261)

<https://www.britannica.com/technology/iron-processing/Iron-making>

<https://www.mogroup.com/insights/blog/mining-and-metals/flotation-columns-getting-the-most-from-fine-ores/>

<https://en.wikipedia.org/wiki/Beneficiation>

<https://www.intechopen.com/chapters/58868>

<https://www.linkedin.com/pulse/seven-factors-influence-effect-froth-flotation-xinhai/>

LINKS DATA SCIENCE:

<https://ianlondon.github.io/blog/encoding-cyclical-features-24hour-time/>

<https://stats.stackexchange.com/questions/126230/optimal-construction-of-day-feature-in-neural-networks/>

<https://towardsdatascience.com/why-you-should-not-rely-on-t-sne-umap-or-trimap-f8f5dc333e59>

<https://www.quora.com/What-do-we-mean-by-high-dimensional-data>

<https://www.statology.org/high-dimensional-data/>

https://en.wikipedia.org/wiki/Curse_of_dimensionality

<https://towardsdatascience.com/the-curse-of-dimensionality-5673118fe6d2>

<https://www.geeksforgeeks.org/difference-between-pca-vs-t-sne/>

<https://medium.com/analytics-vidhya/t-sne-intuition-7d373819088c>

<https://www.thekerneltrip.com/statistics/tsne-vs-pca/>

<https://datascience.stackexchange.com/questions/109276/what-is-the-meaning-of-preserving-local-or-global-structure-of-the-data>

<https://datascience.stackexchange.com/questions/36889/what-does-it-mean-by-t-sne-retains-the-structure-of-the-data>

<https://distill.pub/2016/misread-tsne/>

<https://opentsne.readthedocs.io/en/latest/parameters.html#perplexity>

<https://towardsdatascience.com/tsne-degrades-to-pca-d4abf9ef51d3>

<https://pair-code.github.io/understanding-umap/>

Artículos y códigos de otras personas que hacen referencia al Dataset usado en este proyecto:

<https://techlabs-aachen.medium.com/quality-prediction-in-a-mining-process-1a2b70b51303>

https://github.com/nishp763/Quality_Prediction_ML/blob/master/src/Approach%202/Final_Project_Mo del2.ipynb

<https://www.kaggle.com/datasets/edumagalhaes/quality-prediction-in-a-mining-process/code>

FUENTE DEL DATASET:

<https://www.kaggle.com/datasets/edumagalhaes/quality-prediction-in-a-mining-process>