Universidad de Buenos Aires
Facultad de Ciencias Exactas y Naturales
Departamento de Matemática e Instituto de Cálculo

Trabajo Final Integrador

Martín Gabriel Cargnel

Profesor: Dr. Ricardo Maronna

Febrero, 2024

1 Introducción

Para determinar la composición de un conjunto de vasijas de vidrio de un yacimiento arqueológico se puede optar por dos alternativas: llevar a cabo un análisis químico ó calibrar un modelo que use los datos de una espectometría. Siendo el primero más caro que el segundo, es de interés encontrar un modelo que nos permita reemplazar el análisis químico. Por lo cual, en este trabajo se compara la *performance* predictiva de distintos modelos para encontrar el que mejor prediga el contenido de uno de los compuestos químicos de las vasijas.

Los datos con los que se realiza este análisis son una muestra de 180 vasijas a las que se les realizó un tanto una espectometría de rayos X sobre 1920 frecuencias, como un análisis de laboratorio para determinar el contenido de 13 compuestos químicos:

$$Na_2O,\ MgO,\ Al_2O_3,\ SiO_2,\ P_2O_5,\ SO_3,\ Cl,\ K_2O,\ CaO,\ MnO,\ Fe_2O_3,\ BaO\ y\ PbO$$

Para este estudio solo se tendrán en cuenta las frecuencias 100 a 400 (por ser las únicas que no tienen valores casi nulos) y se buscará predecir el compuesto SO_3. Cabe aclarar que cada covariable del modelo será la energía correspondiente a cada frecuencia.

Este trabajo fue realizado en su totalidad con el *software* estadístico R y se organiza de la siguiente manera: en la Sección 2 se lleva a cabo un análisis exploratorio de los datos con los que vamos a trabajar. Luego se presentan e implementan distintos modelos en la Sección 3 para después comparar su poder predictivo en términos del error de predicción en la Sección 4. Una vez identificado el modelo que tenga el error de predicción más bajo se procede a estudiar en profundidad las predicciones que realiza en la Sección 5. Finalmente, el trabajo concluye en la Sección 6. En el apéndice se pueden ver en detalle los resultados de cada modelo.

2 Análisis Exploratorio

En primer lugar se realizó un análisis exploratorio de los datos con los que se cuenta. Por lo que se graficó la media de cada espectro de frecuencia y la varianza, asi como también la covarianza que cada frecuencia tiene con el compuesto SO_3 que se quiere predecir. Se puede observar en las Figura 1 , Figura 2 y Figura 3 que no todas las frecuencias son igual de relevantes *a priori*, ya que no todas tienen ni mucha media o varianza, ni tampoco correlacionan mucho con la variable de respuesta. Hay rangos de frecuencias, como los que están alrededor de la frecuencia 70, 220 o 260, que condensan la mayor cantidad de "*información*" del compuesto. Más adelante se valida si dichas frecuencias son las que el modelo que mejor prediga toma como más relevante en sus coeficientes.

Para entender cómo se distribuye la que será nuestra variable dependiente podemos utilizar la Figura 4. En la misma, mediante un histograma de densidad del compuesto SO_3, se puede ver que pareciera haber una leve asimetría hacia la derecha y que se trata de una distribución unimodal.

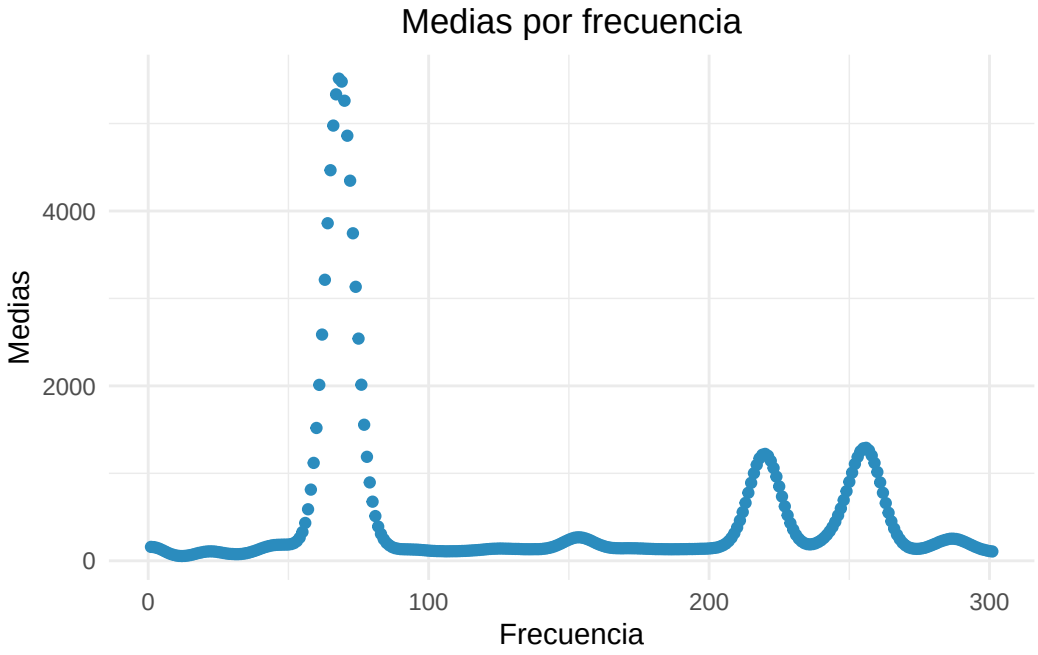

Figura 1: Media de la enegía del espectro correspondiente a las frecuencias.

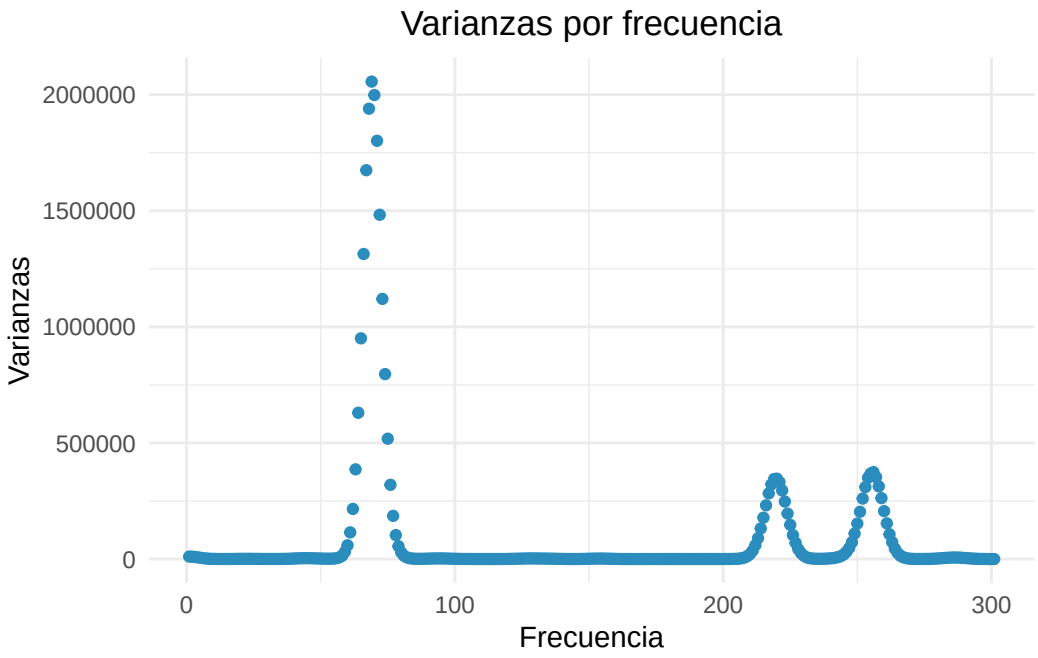

Figura 2: Varianza de la energía del espectro correspondiente a las frecuencias.

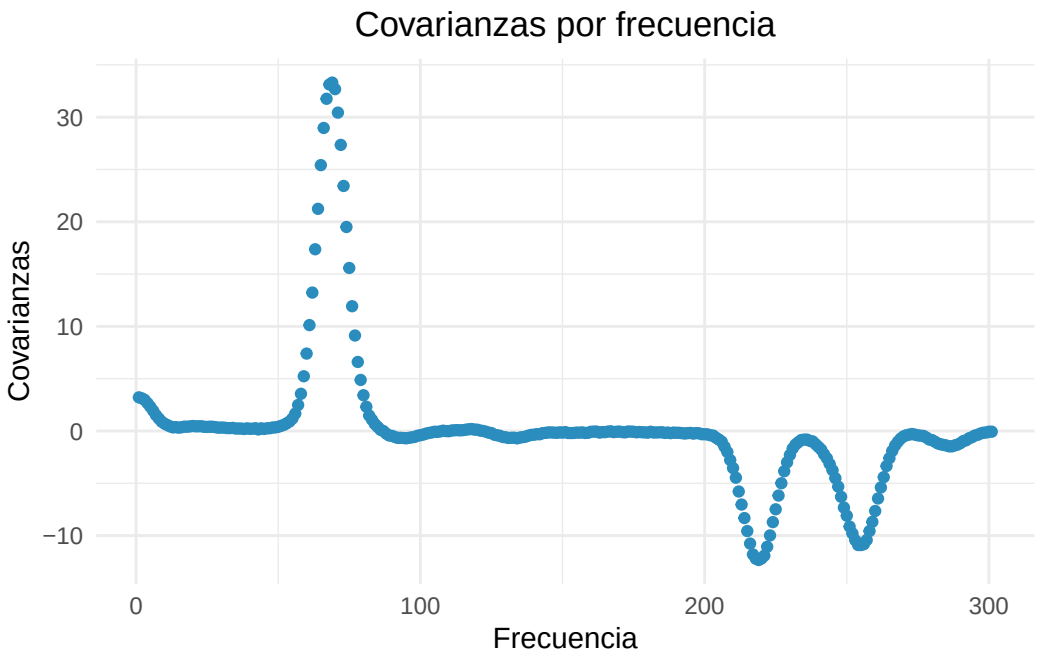

Figura 3: Covarianza de la energía del espectro correspondiente a las frecuencias y el Compuesto 6

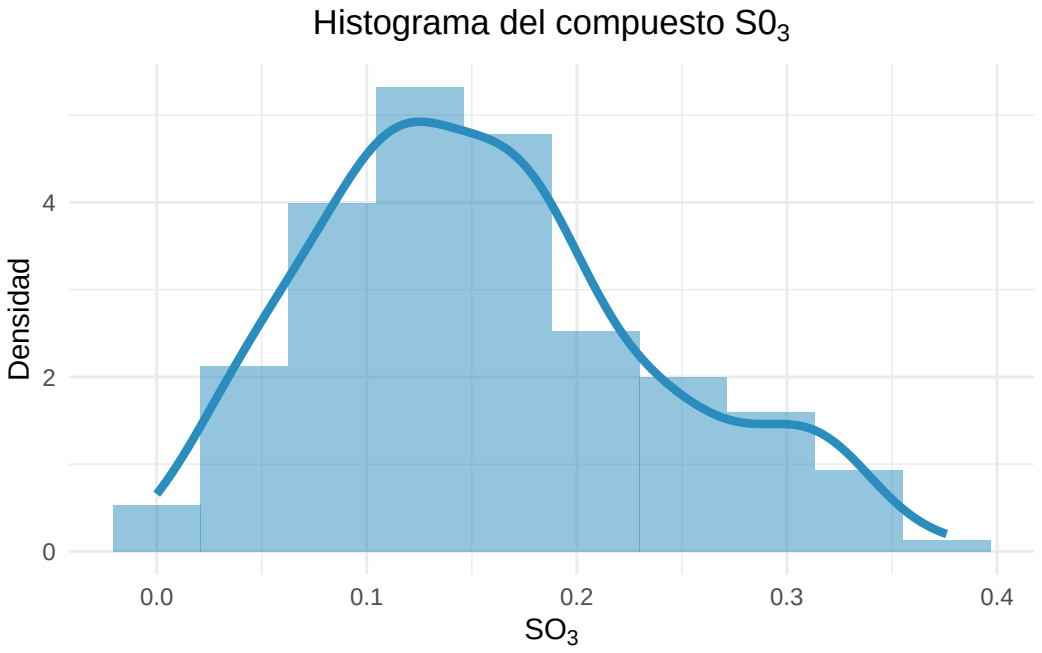

Figura 4: Histograma y estimación de densidad del compuesto SO_3.

3 Métodos

Dado que el objetivo de este estudio es encontrar un modelo que nos permita predecir el componente SO_3 de vasijas de vidrio de un yacimiento arqueológico, se evaluará primero qué tipo de modelo tiene la mejor capacidad predictiva para este problema en particular.

Para ello, primero se divide nuestro set inicial de datos en *train* y *test*, en esta partición inicial el 80% de los datos se asignó a la muestra de entrenamiento y el 20% restante a la muestra de testeo utilizando el paquete *rsample*. Luego se evalúa la *performance* predictiva de los modelos ajustados usando las siguientes metodologías: Regresión Regularizada (*Ridge*, *LASSO* y *Elastic Net*), *Principal Component Regression* (PCR) y *Partial Least Square Regression* (PLS), Árbol de regresión, *Random Forest*, *Boosting* y K-vecinos más cercanos exclusivamente con la muestra de entrenamiento, para luego reportar el error final en la muestra de testeo con el modelo seleccionado.

Cabe mencionar que los datos provienen de una espectrometría y cada una de las 180 observaciones es compuesta por 301 covariables, representando cada una de ellas la medición de una frecuencia. Por lo tanto, este problema ya parte del hecho de que el número de covariables a usar es mayor al número de observaciones y, por lo tanto, es de esperar que los modelos con mejor capacidad predictiva sean los que abordan de una u otra manera la reducción del número de covariables que se utilizarán en el modelo. Podemos dividir las metodologías mencionadas anteriormente en dos grupos: En primer lugar se encuentran las que incluyen la regularización de las covariables como es el caso de las regresiones *Ridge*, *LASSO* y *Elastic Net* y en segundo se encuentran las que reducen la cantidad de covariables en un menor número de componentes como PCR y PLS. De todas formas decidimos probar modelos que no incluyan esta propiedad.

También hay que notar que las técnicas mencionadas incluyen hiper-parámetros. Por lo tanto, se utiliza la validación cruzada para obtener el mejor modelo (es decir, la elección del hiperparámetro), que será el que minimice el error cuadrático medio en la muestra de entrenamiento. La validación cruzada consta de dividir aleatoriamente la muestra de entrenamiento en K sub-muestras de aproximadamente igual tamaño y hacer K ajustes. En cada uno de los ajustes se usan los datos pertenecientes a las K-1 sub-muestras y la sub-muestra de validación se utiliza para estimar el error de dicho modelo ajustado. Como este procedimiento se realiza K veces, el error estimado es el promedio de las K estimaciones del error. Se decidió usar un K=5 para que la partición de validación cruzada no quede demasiado pequeña, entendiendo que contamos con únicamente 144 datos en la muestra de entrenamiento. Por último, se comparan los modelos ajustados con la mejor elección de sus respectivos hiper-parámetros utilizando el error estimado por validación cruzada.

3.1 Reducción de la dimensión: PCR y PLS

Estos métodos tienen como estrategia primero calcular las componentes principales[1] de las covariables y luego quedarse con un número menor de estas para usarse como predictoras en el modelo de regresión.

3.1.1 PCR

Para el caso de PCR, la construcción de las componentes principales se hace buscando explicar la mayor proporción de la variabilidad entre las covariables, por lo que no toma en cuenta la variabilidad que existe entre las covariables y la variable de respuesta. Esto último puede generar que no se obtengan los mejores resultados para el problema de predicción que buscamos resolver.

En nuestro caso, se implementó PCR usando el paquete *pls* y usando validación cruzada para la estimación del error de predicción. Como se puede ver en la Figura 5 el mejor modelo se construye con 120 componentes principales como predictoras, el detalle de los 10 resultados con mejor *performance* se puede ver en la Tabla 2.

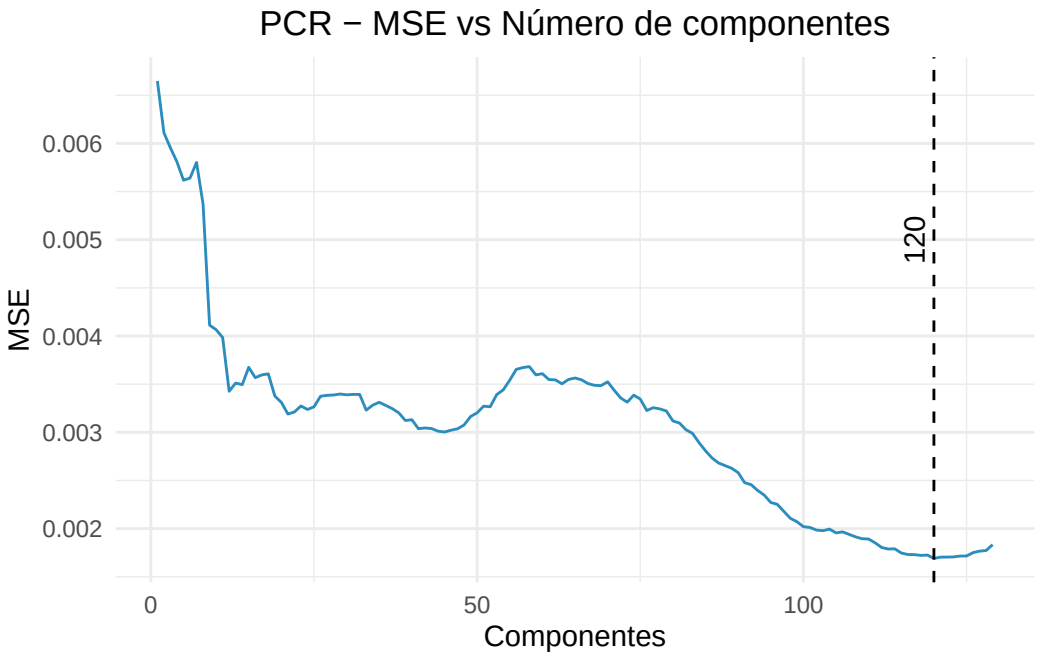

Figura 5: MSE en funcón del número de componentes en PCR, estimado por CV.

[1]La técnica de Componentes Principales es utilizada para reducir la dimensión de los datos de forma de preservar la mayor proporción de variabilidad de los mismos.

3.1.2 PLS

Como se vio anteriormente la metodología de PCR no tiene en cuenta la variabilidad que cada covariable tiene con respecto a la variable a predecir en la construcción de las componentes principales, por lo que también se utilizó la metodología de PLS que construye las componentes principales de una forma en que sí intenta maximizar la proporción de variabilidad de la variable a predecir.

Así, al hacer la implementación mediante el paquete *pls*, se observa en la Figura 6 (y en detalle en la Tabla 3) que el mejor modelo usando esta metodología (obtenido a través de validación cruzada) se logra con 21 componentes principales como predictoras.

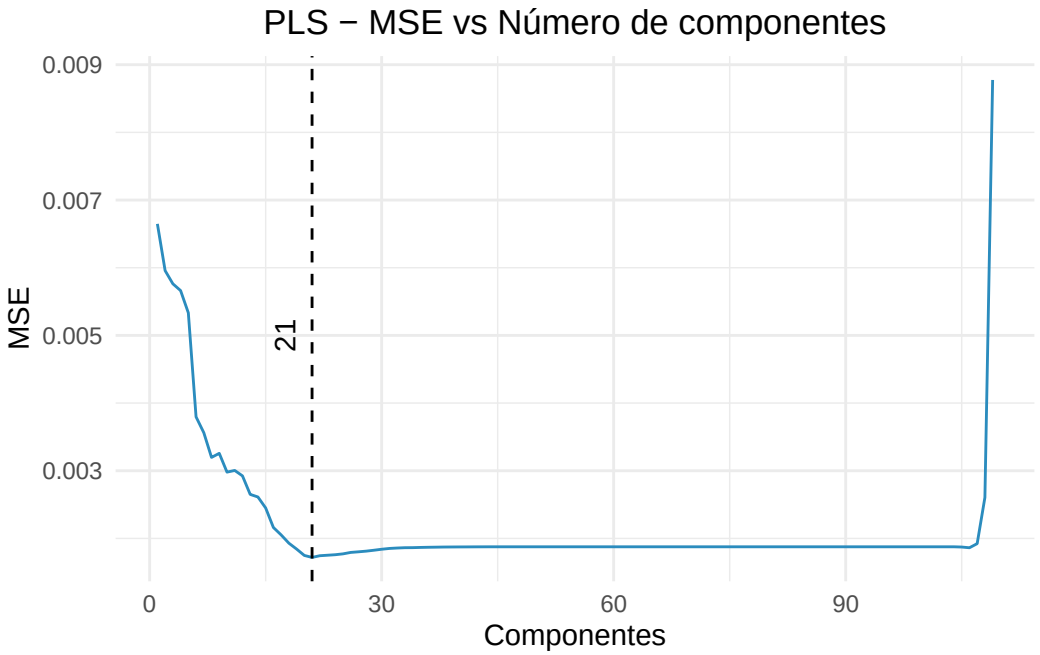

Figura 6: MSE en funcón del número de componentes en PLS, estimado por CV.

3.2 Regularización: Ridge, LASSO y Elastic Net

Las técnicas de regularización empleadas en esta sección agregan una penalización al tamaño de los coeficientes en la estimación de los parámetros de la regresión lineal. Sin embargo, esto introduce hiper-parámetros que serán elegidos por validación cruzada con el objetivo de obtener la combinación de los mismos que minimice el error de predicción.

3.2.1 Ridge

La regularización *Ridge* tiene como principal característica que λ- el hiper-parámetro que regula cuán estricta tiene que ser la penalización incluida- comprime los coeficientes pero no permite que ninguno llegue a ser 0. Se puede ver en la Figura 7 que el $log(\lambda)$ que minimiza el error de predicción es -1.161. Para ver los resultados de

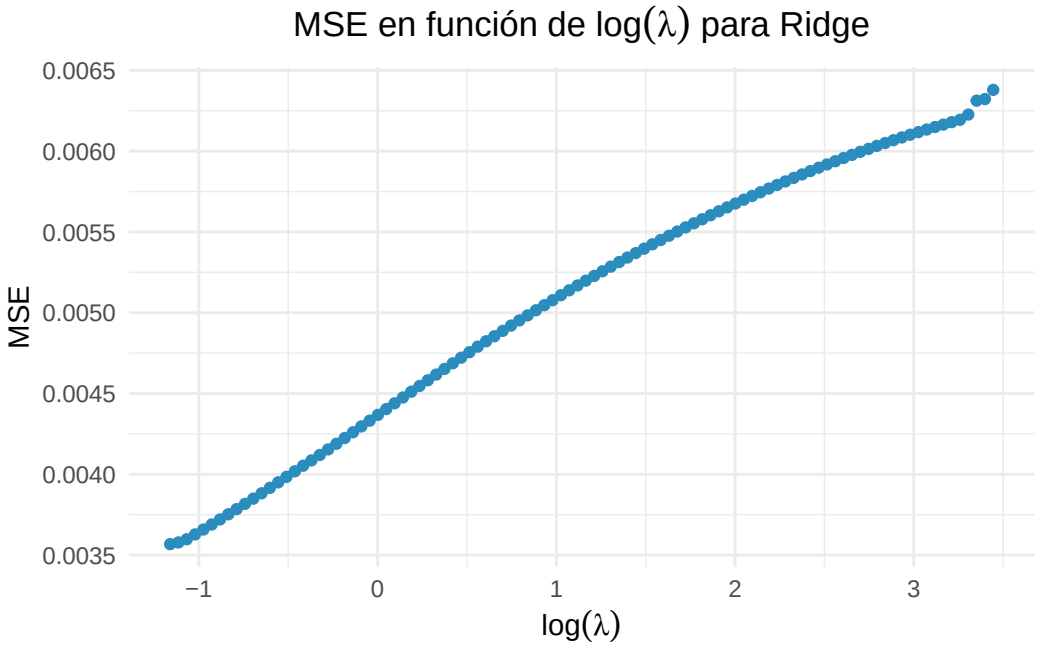

Figura 7: Estimación del error cuadrático medio por validación cruzada en la muestra de entrenamiento para distintos valores de log(λ) en el modelo Ridge.

3.2.2 LASSO

A diferencia de *Ridge*, en *LASSO* los coeficientes si pueden truncarse hasta el 0 por lo que el término de *shrinkage* del primer modelo puede considerarse más suave. Para nuestra aplicación podemos ver en la Figura 8 que el $log(\lambda)$ que minimiza el MSE es -8.069.

3.2.3 Elastic Net

El modelo de *Elastic Net* es una combinación lineal de las penalizaciones utilizadas por *Ridge* y *LASSO*, por lo que se le suma un hiperparámetro adicional (α) que pondera cuanto pesa más una penalización por sobre la otra. Así, para este modelo también se utilizó la técnica de validación cruzada para hallar el mejor modelo, ahora eligiendo la combinación de

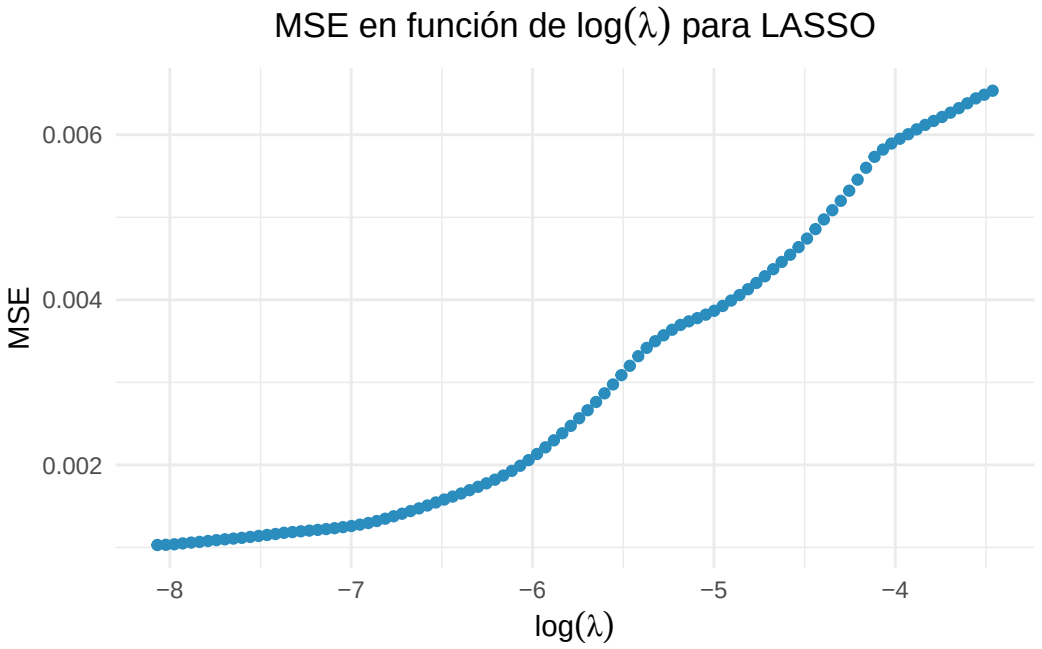

Figura 8: Estimación del error cuadrático medio por validación cruzada en la muestra de entrenamiento para distintos valores de log(λ) en el modelo LASSO.

hiperparámetros α y λ que generan un modelo con el menor error de predicción en la muestra de entrenamiento.

En la Figura 9 se puede ver que el $log(\lambda)$ que minimiza el error de predicción es -7.9632 para un α de 0.9.

3.3 Árbol de regresión

El método del árbol de regresión consiste en hacer particiones recursivas en el espacio de covariables de forma tal de obtener subconjuntos terminales disjuntos (las hojas o nodos terminales del árbol) lo más homogéneos posibles. Por lo tanto, el árbol se construye para definir regiones en las cuales las observaciones que caigan en cada región sean lo más parecidas entre ellas.

En nuestro caso se decidió hacer una implementación de este modelo usando el paquete *caret*, dado que permite obtener el árbol con menor error de predicción por medio de validación cruzada, comparando entre árboles con distinto número de nodos terminales. Siendo este último el hiper-parámetro para el cual exploraremos valores que van desde el 1 al 15.

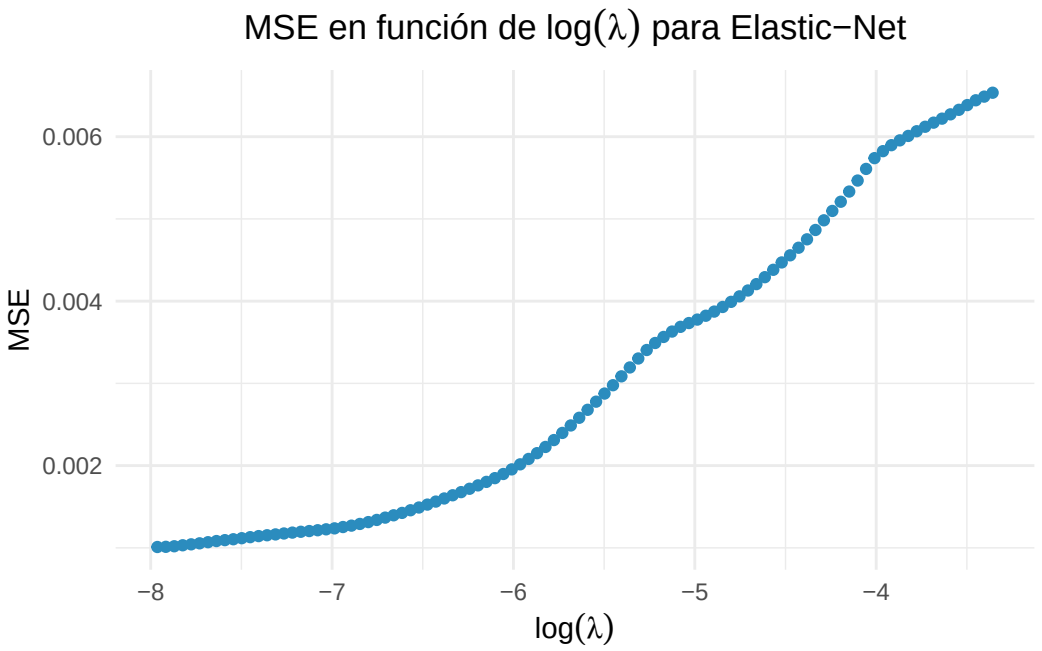

Figura 9: Estimación del error cuadrático medio por validación cruzada en la muestra de entrenamiento para distintos valores de $\log(\lambda)$ en el modelo Elastic net.

En la Figura 10 vemos que el árbol con menor MSE es el que tiene 3 nodos terminales[2], por lo qué podemos referirnos a la Figura 11 para ver que solamente se usaron V243 y V122 para para predecir. Si bien este modelo es simple e interpretable, no tuvo una gran capacidad predictiva con respecto al resto de los modelos evaluados. Por lo que en la próxima subsección se utilizarán dos métodos de ensambles que se espera mejoren la *performance*.

3.4 Ensambles de árboles

Los ensambles se basan en combinar muchos árboles (generalmente sencillos) para obtener un único modelo. Esto se busca porque, pese a que los árboles de decisión tienen la ventaja de ser fácilmente interpretables, generalmente su poder de predicción es bajo. Por este motivo se implementó *Random Forest* y *Boosting* usando el paquete *caret*.

[2]Este resultado fue muy sencible a distintas semillas, pero las estimaciones del MSE se mantuvieron en rangos similares (por arriba del resto de los modelos), como se estudia en las conclusiones.

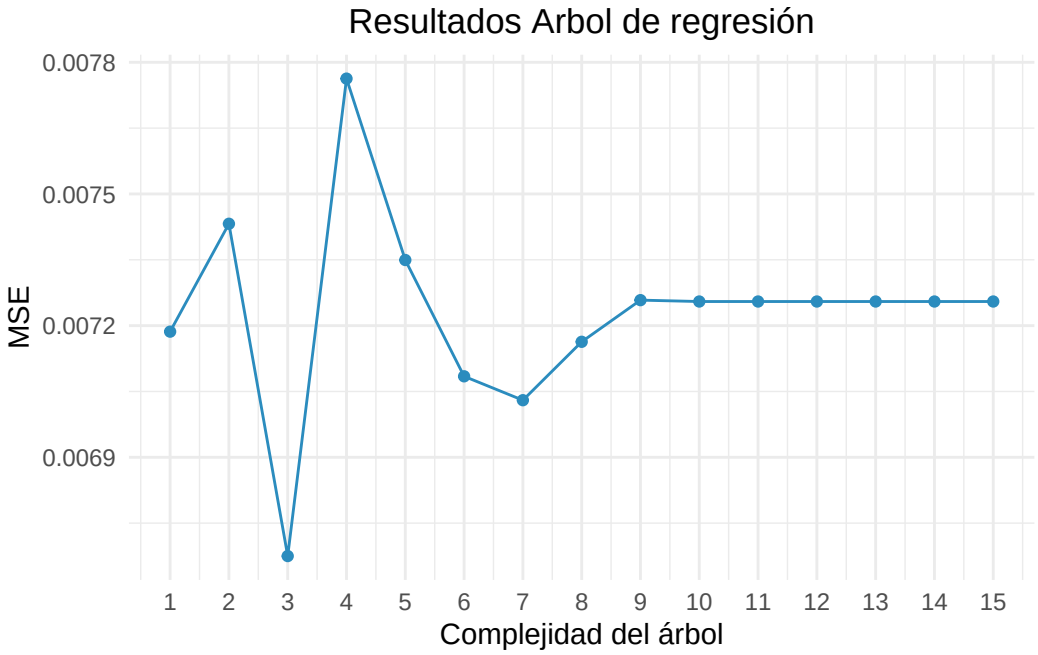

Figura 10: MSE de arboles con distintos nodos terminales

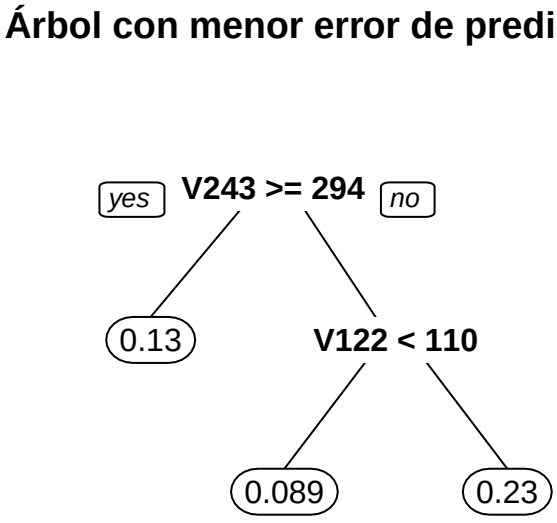

Figura 11: Arbol podado a 3 nodos terminales.

3.4.1 Random Forest

Random Forest construye una cierta cantidad de árboles y reporta el promedio de las predicciones. Para crear los árboles usa un sub-conjunto de las covariables, de esta forma se logra que sean menos dependientes entre sí y reducir la varianza. Tanto la cantidad de árboles a generar como el subconjunto de covariables son considerados hiperparámetros en este modelo, por lo que se decidió probar distintas combinaciones de estos para encontrar la que tenga menor MSE por validación cruzada. En relación con la cantidad de árboles, se exploraron valores en el rango de 50 a 500, con incrementos de 50. Mientras que en cuanto a los subconjuntos de covariables, se optó por evaluar modelos que utilicen 2, 6 y 10 covariables para crear los árboles.

Cómo se puede ver en la Figura 12 los resultados obtenidos fueron 250 arboles para un sub-conjunto de 6 covariables. También podemos estudiar en la Figura 13 que las covariables más importantes para predecir el SO_3 fueron V246 y V248.

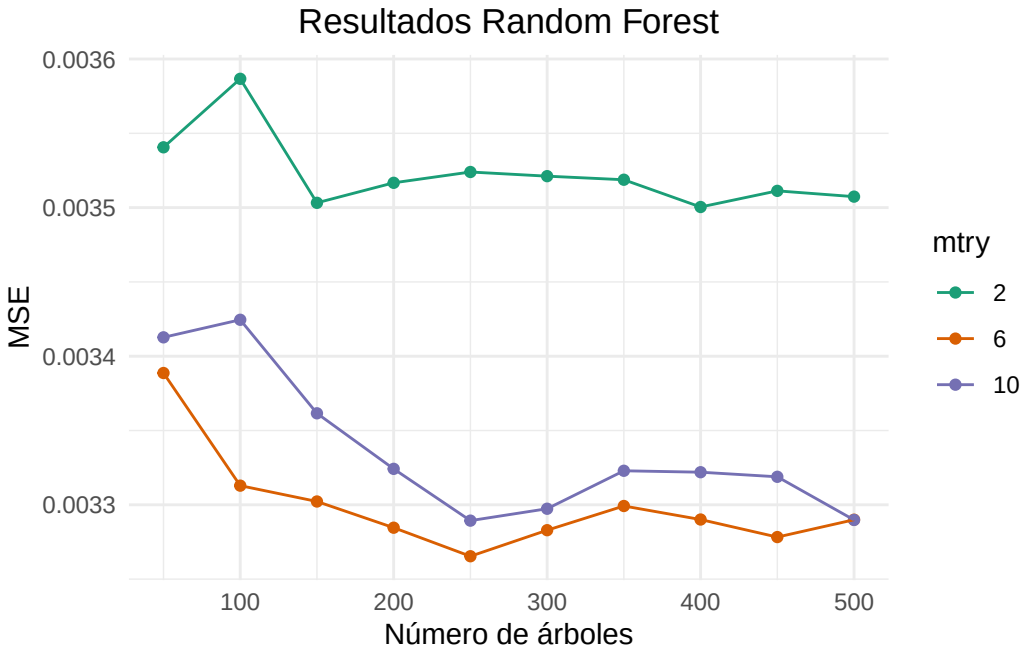

Figura 12: MSE del modelo GBM para distintos valores de mtry y ntree.

3.4.2 Boosting

En cambio en *Boosting* los árboles se crean secuencialmente utilizando información sobre los anteriores. Es decir que cada árbol se crea sobre los residuos del árbol anterior. De esta forma se pueden mejorar las predicciones de forma paulatina y así evitar el *overfitting*. En este estudio

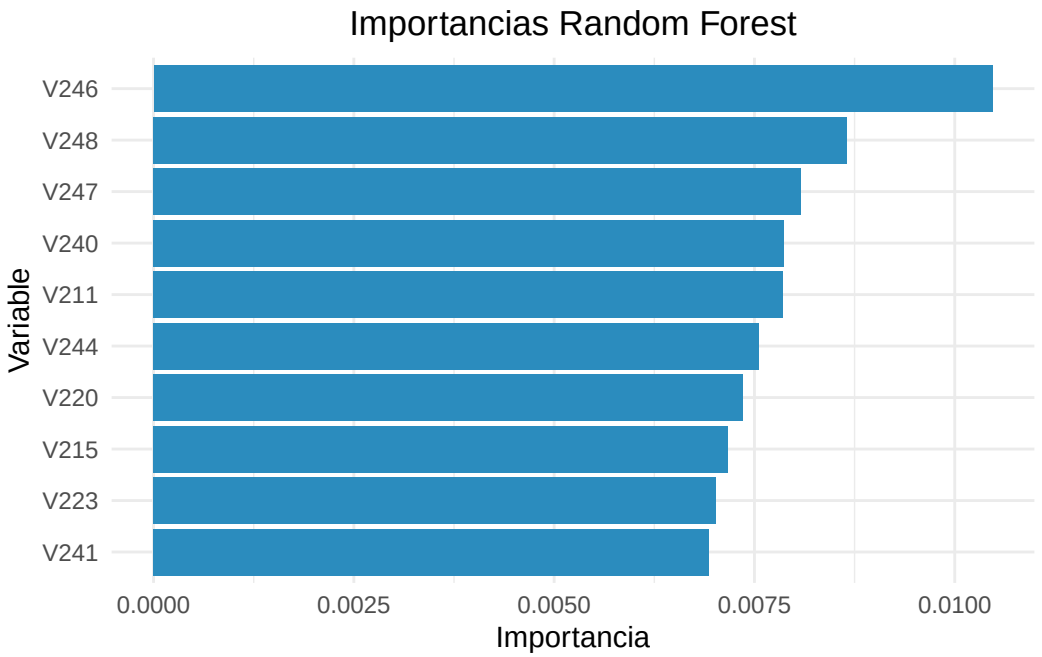

Figura 13: Importancia de las covariables Random Forest ordenadas de mayor a menor.

los hiper-parámetros evaluados fueron la cantidad de árboles y el *shrinkage* ó *learninig rate* que se usa para regular cuanto se utiliza la información de los arboles anteriores. La búsqueda de los hiper-parámetros se realizó para valores de árboles que van desde 100 hasta 2000 en incrementos de a 200 para los siguientes valores de *shrinkage*: 0.001, 0.01 y 0.01.

Al calcular el MSE para las distintas combinaciones de hiper-parámetros encontramos que el mínimo se alcanza en 1900 para *Boosting* con un *learning rate* de 0.01. Para entender un poco más qué covariables utilizó el modelo de *Boosting* con menor error de predicción podemos ver la Figura 15 donde vemos que, V120 fue la covariable más importante en predecir el compuesto SO_3.

3.5 K vecinos cercanos

Este modelo se basa en promediar los valores de los k vecinos más cercanos a x en el espacio definido por las covariables. El hiper-parámetro con el que cuenta es k que refiere al número de vecinos que se tendrá en cuenta para calcular el promedio.

En este trabajo se utilizó el paquete *caret* y se evaluaron valores de k que van desde 1 hasta 10. Se puede ver en la Figura 16 que el número de vecinos que minimiza el MSE es 4. Cabe destacar que este modelo es sensible a cambios de escala, por lo que estandarizamos las covariables restando la media y dividiendo por el desvío.

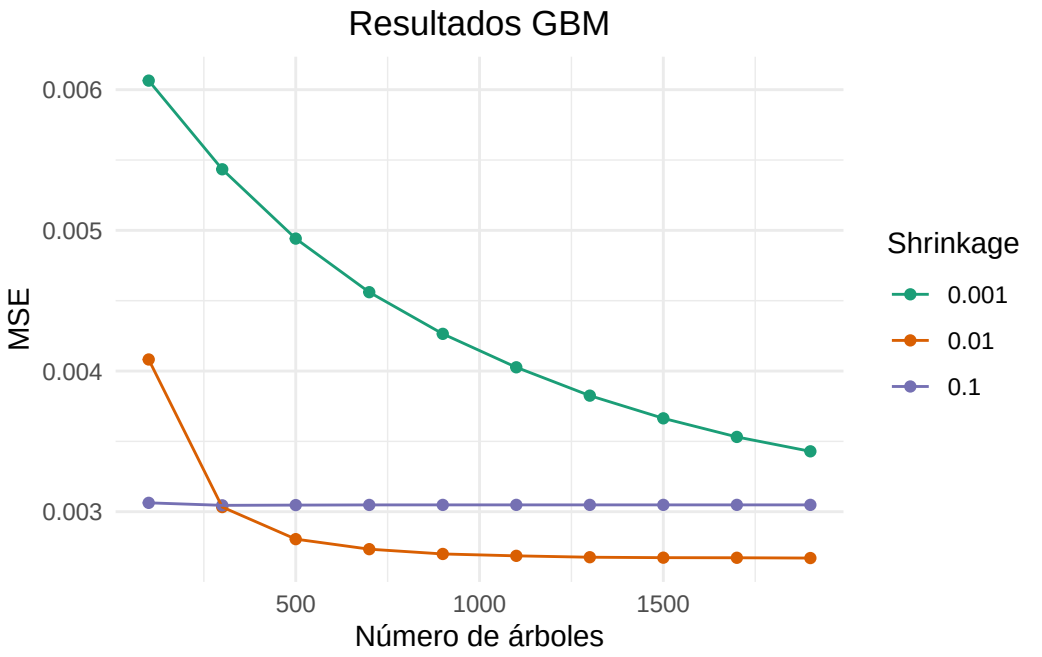

Figura 14: MSE del modelo GBM para distintos valores de skrinkage y ntree.

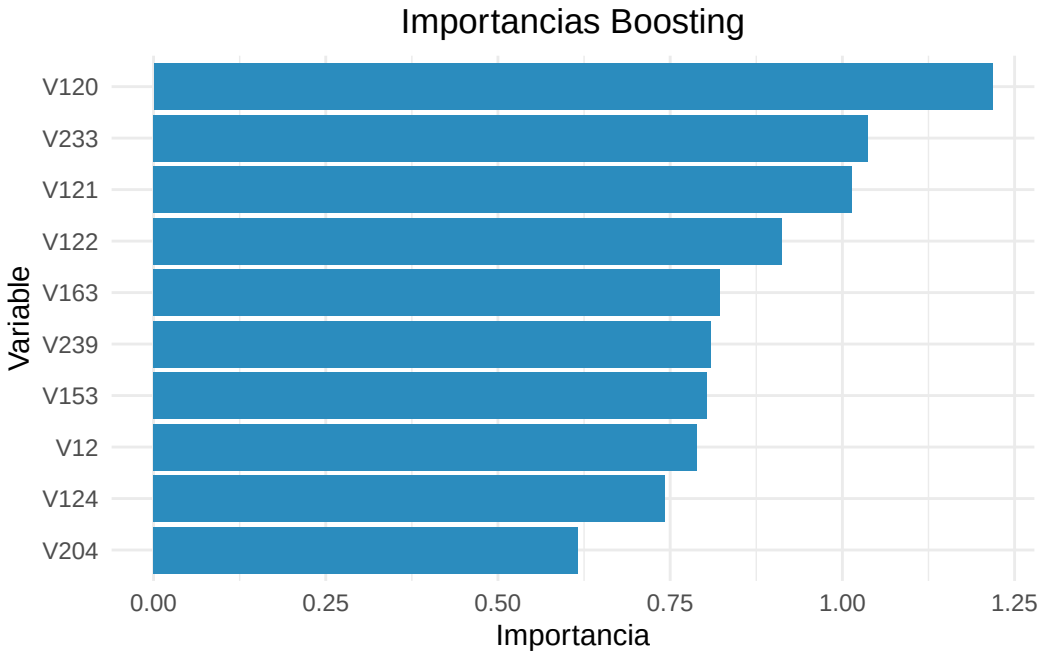

Figura 15: Importancia de las covariables Boosting ordenadas de mayor a menor.

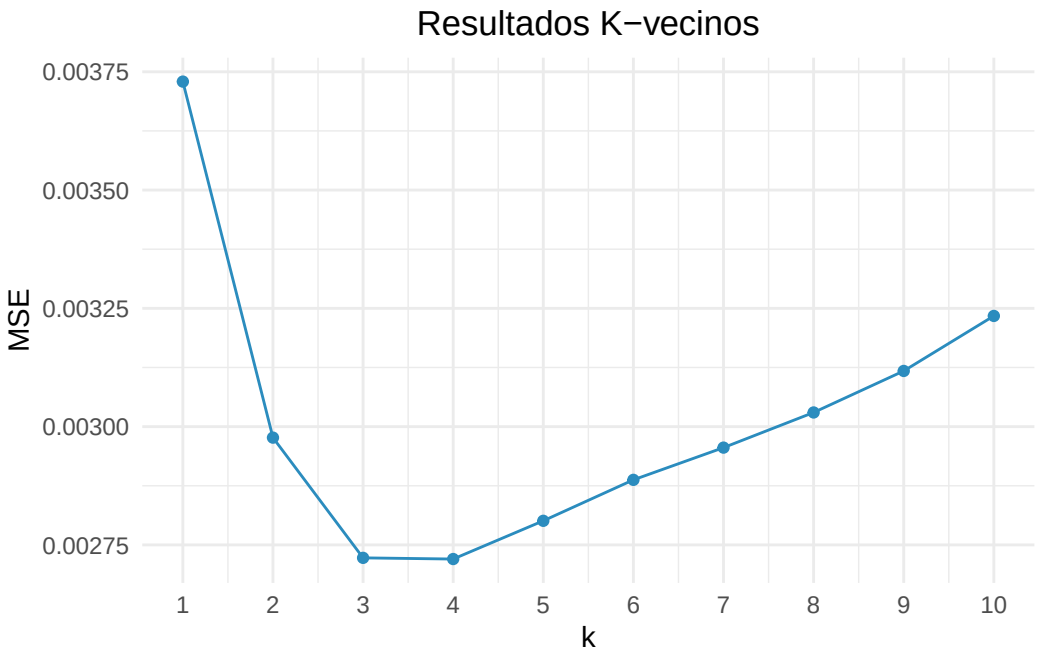

Figura 16: Error cuadrático medio del modelo KNN para distintos valores de k.

4 Comparación de resultados

Con lo expuesto previamente se ajustaron cada uno de los modelos mencionados de forma de obtener el "mejor" de cada uno de ellos -es decir, el que tiene el menor error de predicción estimado por validación cruzada en la muestra de entrenamiento. Así, obtuvimos los siguientes errores con cada modelo, mostrados en la Tabla 1.

Tabla 1: Comparación de modeloes usando las estimaciones del error de predicción con validación cruzada en los datos de entrenamiento.

Modelo	MSE
Elastic Net	1.01e-03
LASSO	1.03e-03
PCR	1.69e-03
PLS	1.72e-03
Boosting	2.67e-03
Random Forest	3.27e-03
Ridge	3.57e-03
K-vecinos	3.73e-03
Arbol de Regresión	6.67e-03

Se puede ver que el modelo con mejor performance predictiva fue *Elastic Net*, seguido muy de cerca por *LASSO*. También observamos que casi todos los modelos que de alguna forma reducen el número de covariables[3] tienen una mejor *performance* que los demás (a excepción de *Ridge*), esto era de esperarse dado que los datos con lo que contamos tienen más covariables que observaciones. Finalmente, notamos que tanto *Boosting* como *Random Forest* tuvieron un mejor desempeño, en términos del error cuadrático medio, que el Árbol de regresión. Esto último también podía preverse dado que los ensambles de árboles se crearon justamente para mejorar el poder predictivo de los árboles individuales.

5 Supuestos

Una vez elegido el modelo con mejor capacidad predictiva en nuestros datos de entrenamiento, el cual es *Elastic Net*, se procede a estudiarlo con mayor profundidad. El modelo ajustado de *Elastic Net* tiene los siguientes coeficientes para cada frecuencia (covariable), que se pueden ver en la Figura 17. De forma contra intuitiva, los coeficientes distintos de cero (y más grandes en valor absoluto) -los cuales son 75- no coinciden con las frecuencias que se habían comentado en la sección exploratoria que eran las más "informativas", en el sentido que tenían mayor varianza en sí mismas y covarianza con la variable de respuesta. Sin embargo, no se encontró una razón "intuitiva" de este hecho, y se lo atribuye a la heurísitca de cómo el procedimiento de *Elastic Net* aplicando las penalizaciones L1 y L2.

Por otro lado, para corroborar los supuestos de regresión lineal se realizaron los siguientes dos gráficos: en el primero se ven los residuos versus los valores ajustados (Figura 18), y en el segundo un QQ-plot de los residuos (Figura 19). Como se puede notar de los mismos no se percibe una estructura en los residuos que pueda indicar un problema con el modelo, y además los residuos parecen cumplir el supuesto de normalidad (no se ven colas pesadas que nos alertasen de posibles *outliers*). Por lo tanto, concluimos que por el análisis gráfico, pareciera que el modelo propuesto cumple los supuestos de regresión lineal.

6 Conclusión

El modelo que mejor capacidad predictiva tiene para nuestro objetivo de predecir el componente SO_3 presente en las vasijas, y sujeto a los datos con los que contamos, es el modelo ajustado *Elastic Net*. El error de predicción (MSE) estimado en la muestra de testeo es 0.02697. Cabe mencionar que esta estimación final del error fue hecha con datos que el modelo no utilizó para su ajuste ni para la comparación entre los otros modelos probados, ya que esas fueron hechas con los datos de entrenamiento y la técnica de validación cruzada para las estimaciones de los errores de cada modelo en dichos datos.

[3]Los modelos son *Elastic Net*, *LASSO*, textitRidge, *PCR* y *PLS*

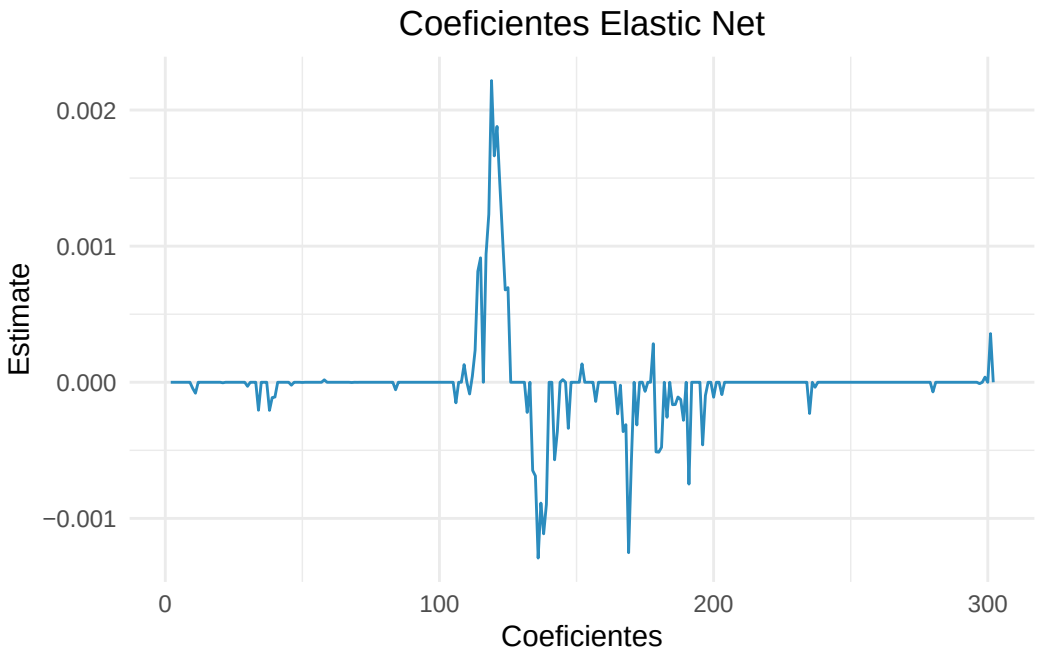

Figura 17: Coeficientes del modelo ajustado Elastic net para cada frecuencia

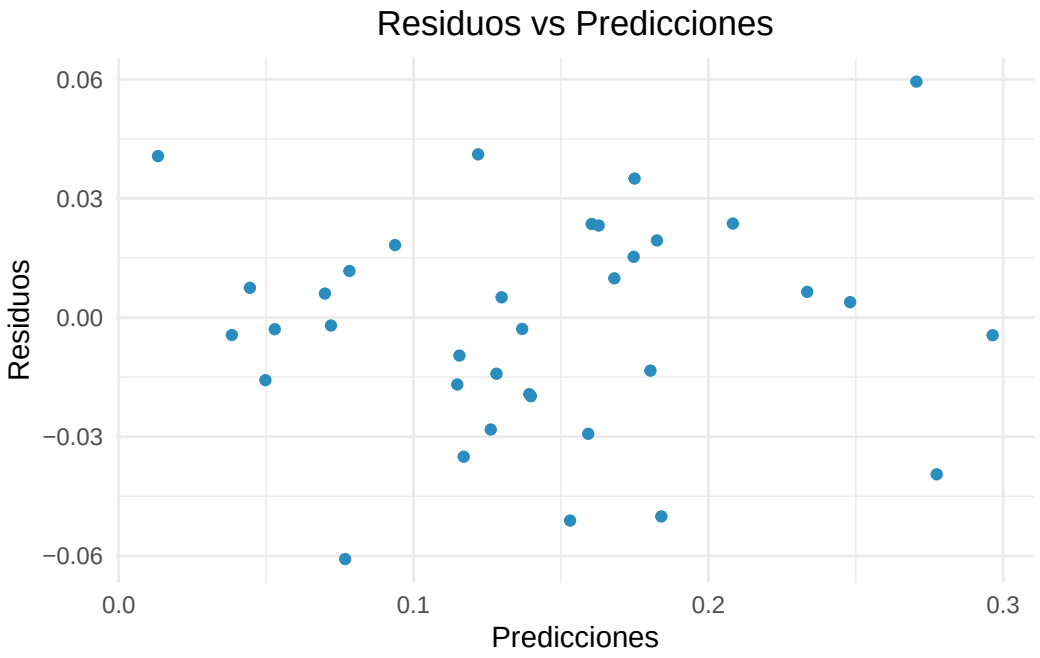

Figura 18: Valores ajustados versus resudiuos en muestra de validación

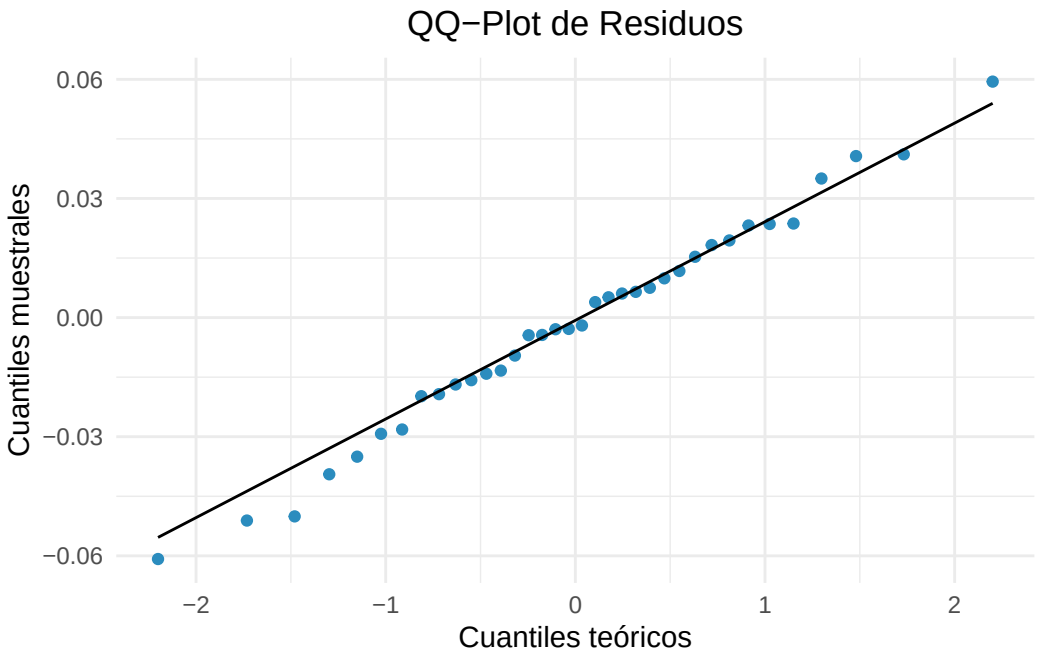

Figura 19: QQ-plot de los residuos del modelo Elastic Net.

Apéndice

Tabla 2: MSE para PCA con distintos componentes

Componentes	MSE
120	1.69e-03
121	1.70e-03
122	1.70e-03
123	1.71e-03
124	1.72e-03
125	1.72e-03
118	1.72e-03
119	1.73e-03
117	1.73e-03
116	1.73e-03

Tabla 3: MSE para PLS con distintos componentes

Componentes	MSE
21	1.72e-03
22	1.74e-03
20	1.75e-03
23	1.75e-03
24	1.76e-03
25	1.77e-03
26	1.79e-03
27	1.80e-03
28	1.81e-03
29	1.83e-03

Tabla 4: MSE para distintos hiperparámetros en Elastic-net, Ridge y LSASO.

Alpha	Log Lambda	MSE
0.9	-7.963	1.01e-03
0.8	-7.845	1.02e-03
0.7	-7.712	1.02e-03
0.6	-7.558	1.03e-03
1.0	-8.069	1.03e-03
0.5	-7.375	1.05e-03
0.4	-7.152	1.07e-03
0.3	-6.865	1.12e-03
0.2	-6.459	1.21e-03
0.1	-5.766	1.40e-03
0.0	-1.161	3.57e-03

Tabla 5: MSE para distintos hiperpárametros en Random Forest.

Número de árboles	mtry	MSE
250	6	3.27e-03
450	6	3.28e-03
300	6	3.28e-03
200	6	3.28e-03
250	10	3.29e-03
500	10	3.29e-03
500	6	3.29e-03

Número de árboles	mtry	MSE
400	6	3.29e-03
300	10	3.30e-03
350	6	3.30e-03

Tabla 6: MSE para distintos hiperpárametros en Boosting.

Número de árboles	shrinkage	MSE
1900	0.010	2.67e-03
1700	0.010	2.67e-03
1500	0.010	2.67e-03
1300	0.010	2.68e-03
1100	0.010	2.69e-03
900	0.010	2.70e-03
700	0.010	2.73e-03
500	0.010	2.81e-03
300	0.010	3.03e-03
300	0.100	3.05e-03
500	0.100	3.05e-03
700	0.100	3.05e-03
900	0.100	3.05e-03
1100	0.100	3.05e-03
1300	0.100	3.05e-03
1500	0.100	3.05e-03
1700	0.100	3.05e-03
1900	0.100	3.05e-03
100	0.100	3.06e-03
1900	0.001	3.43e-03
1700	0.001	3.53e-03
1500	0.001	3.66e-03
1300	0.001	3.83e-03
1100	0.001	4.03e-03
100	0.010	4.08e-03
900	0.001	4.26e-03
700	0.001	4.56e-03
500	0.001	4.94e-03
300	0.001	5.43e-03
100	0.001	6.06e-03

Tabla 7: MSE para distintos arboles con distinta cantidad de nodos terminales

Nodos terminales	MSE
3	6.7e-03
7	7.0e-03
6	7.1e-03
8	7.2e-03
1	7.2e-03
10	7.3e-03
11	7.3e-03
12	7.3e-03
13	7.3e-03
14	7.3e-03
15	7.3e-03
9	7.3e-03
5	7.3e-03
2	7.4e-03
4	7.8e-03

Tabla 8: Error cuadrático medio para distintos valores de k en K-vecinos

k	MSE
4	2.720e-03
3	2.723e-03
5	2.801e-03
6	2.888e-03
7	2.956e-03
2	2.977e-03
8	3.030e-03
9	3.118e-03
10	3.234e-03
1	3.729e-03

Bibliografía

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). An introduction to statistical learning: with applications in R. Corrected edition. New York, Springer.

Hastie, T., Tibshirani, R., & Friedman, J. H. (2009).The elements of statistical learning: data mining, inference, and prediction. 2nd ed. New York, Springer.

R Core Team (2020). R: A language and environment for statistical computing. R Foundation for Statistical Computing, Vienna, Austria.