Pronóstico de erupciones volcánicas

Óscar Anuar Alvarado Morán
OscarAlvarado@ciencias.unam.mx

Dante Bermúdez Marbán
bermudezmarbandante@comunidad.unam.mx

Resumen

En este trabajo se presenta la implementación del modelo de aprendizaje máquina LGBM (Light Gradiente Boosting Machine) para el pronóstico temprano de erupciones volcánicas dado un conjunto de entrenamiento con 4,431 archivos, cada uno con medidas de 10 sensores de la velocidad de tierra en la superficie del volcán. Se prueba el modelo con la plataforma de evaluación en línea de *kaggle*. Los resultados obtenidos son: 695, 174.9 cs, que es aproximadamente una hora, 55 minutos y 52 segundos. y $2,034,299.5\,cs$, que es aproximadamente 5 horas, 39 minutos y 3 segundos. El modelo resulta ser más confiable respecto a los datos de prueba en un intervalo entre los 100,000 y los 230,000 segundos, que es entre un día + 4 horas y 2 días + 16 horas. El mejor modelo finalmente se puso en producción.

Keywords: Erupcion, Volcán, Forecasting, Predicción, Pronóstico, Ciencia de Datos.

1. Introducción

Alrededor del mundo hay varios tipos de desastres naturales que afectan desde las comunidades más marginadas hasta las más prósperas, por lo que uno de los desafíos más grandes para los científicos ha sido intentar predecir este tipo de eventos y mas que poder evitarlos, tomar las precauciones pertinentes. Este desafío abarca desde el modelado del clima a nivel mundial mediante modelados numéricos [1], el estudio de medios deformables como la tierra para comprender de mejor manera los terremotos, y hasta la predicción de erupciones volcánicas [2][3][4][5][6][7][8][9], etc. Este último tema es de sumo interés ya que se encuentran varios volcanes activos en muchas partes del planeta en el que vivimos, de los cuáles muchos se encuentran en México [10], país junto con Japón por ser conocidos de los que tienen más actividad sísmica y volcánica. Desde hace años se estudia el comportamiento de los volcanes mediante sensores que están recolectando en tiempo real lo que pasa dentro y fuera de estos. Recientemente en América Latina se tuvo la erupción del volcán de Fuego en Guatemala donde se tuvieron más de cien fallecidos y más de 1.7 millones de personas afectadas [11], ya sean de daños colaterales o directos. Además, México es uno de los países con más volcanes activos en el mundo, contenido a su vez en América, que junto con Asia son los continentes que reinan esta parte en cuanto a desastres.

Europa es uno de los continentes que también se puede ver muy afectado por este tipo de desastres, y el *National Institute of Geophysics and Volcanology* está consiente de ello, por lo que ha estado desempeñando un papel importante en cuanto a su liberación de datos respecto a sensores. Esta iniciativa podría ser de sumo interés para otros países con notable actividad volcánica, ya que los modelos que implementa este instituto podrían servir en teoría para cualquier volcán.

En este trabajo se hace uso de técnicas de aprendizaje máquina y minería de datos para el procesamiento de señales con el fin de detectar o predecir erupciones volcánicas en el futuro respecto a los datos que se tienen. Debido a que la variable a pronosticar es el tiempo de la próxima erupción, las técnicas de aprendizaje máquina a utilizar son aquellas que resuelvan un problema de regresión.

Los datos[12] del proyecto se encuentran en la página de Kaggle. Estos son proporcionados por el *Instituto Nazionale Di Geofisica e vulcanologia* [13] para un concurso en la página ya mencionada que se lleva a cabo a finales del año 2020. Los datos contienen 10 minutos de lecturas de 10 sensores localizados sobre el volcán Etna, ubicado en la costa este de Sicilia, Italia. Los datos están normalizados (todos juntos) de tal manera que se pudieran almacenar en enteros de 16 bits ($[-32768, 32767]$) y son proporcionales a velocidades de la tierra. Todos los datos aluden al mismo volcán. Hay conjuntos de datos tanto de prueba como de entrenamiento que en conjunto pesan un poco más de 30 GB.

Para el presente trabajo se implementa el modelo conocido como LGBM[14] (Light Gradient Boosting Machine por sus siglas en inglés) debido a su enfoque en resultados precisos y manejo fácil de grandes cantidades de datos, además de que ofrece una alternativa más rápida que su predecesor, el GBM. Actualmente se ha popularizado este modelo de aprendizaje máquina en la comunidad de ciencia de datos debido al

gran crecimiento en las bases de datos y el gran potencial que muestra este modelo para lidiar con esta gran cantidad de datos, además de que es muy fácil de implementar, que puede no serlo a la hora de implementar una red neuronal multicapa o algo por el estilo. Otra ventaja que tiene LGBM es que tiene soporte para aprendizaje con GPU, por lo que si ya es rápido por sí mismo, se puede recortar aún más su tiempo de entrenamiento. Se recomienda usar con una gran cantidad de datos, ya que es un modelo que es muy sensible a tener sobre ajuste.

FIGURA 1: Crecimiento de la mayoría de los algoritmos basados en árboles.

La mayoría de los algoritmos basados en árboles hacen el crecimiento de árbol horizontalmente 1, mientras que LGBM lo hace verticalmente 2, lo que significa que el crecimiento lo hace por hoja y los otros (como XGBoost) lo hacen más bien por nivel. Debido a esto, LGBM tiene más libertad en cuanto a su crecimiento al poder escoger la hoja con el máximo cambio de pérdida.

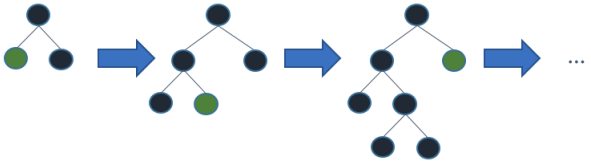

FIGURA 2: Crecimiento del modelo LGBM.

Dada la explicación para su velocidad, ahora hablemos de la parte del *Gradient Boosting*: La potenciación de gradiente [15] (o *Boosting Gradient*) es una técnica de aprendizaje máquina utilizada para problemas de regresión o clasificación que consiste en un conjunto de modelos de predicción débiles que normalmente son árboles de decisión, los construye en forma escalonada y los generaliza, con lo que obtiene una optimización de una función de pérdida diferenciable.

El LGBM tiene tres técnicas de potenciación del gradiente, *GBDT*, *DART*, y *GOSS*. Por el momento sólo nos interesa el primero, que se llama así por sus siglas en inglés de Gradient Boosting Decision Trees. Este método de potenciación del gradiente es justamente como ya lo habíamos planteado, pero más específicamente lo que se tienen son muchos árboles de decisión que se construyen secuencialmente, el primero aprenderá cómo ajustarse a la etiqueta de los datos, el segundo aprenderá cómo ajustarse al residuo entre la predicción del primero y el valor verdadero de los datos, el tercero aprenderá cómo ajustar los residuos del segundo árbol y así sucesivamente. Gráficamente se puede razonar como se observa en la Figura 3. Todos estos árboles son entrenados propagando los gradientes de los errores a través del sistema.

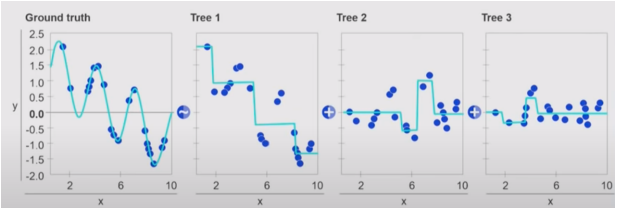

FIGURA 3: Arboles de decisión aprendiendo cómo ajustarse. El primer recuadro muestra la distribución real de los datos, los siguientes recuadros representan a cada árbol que se va agregando escalonadamente. Obtenido de [16].

Uno de los retos con este modelo de aprendizaje máquina son sus hiperparámetros ya que son muchos, hay que conocerlos bien y saber para qué sirven de modo que se tenga un modelo adecuado para el conjunto de datos que se esté utilizando. Es importante recalcar el hecho de que sean muchos hiperparámetros, ya que a la hora de ajustarlos puede ser tardado con alguna técnica como *grid search*, más si no se tiene idea en qué rango podrían estar los mejores. Los principales hiperparámetros se muestran a continuación, además de una breve explicación de estos:

- ***learning_rate***: La tasa de aprendizaje determina el impacto de cada árbol sobre el resultado final. Como sabemos, cuando se habla del gradiente, la tasa de aprendizaje es el tamaño del paso con el que se va a mover la búsqueda de mínimos.
- ***boosting_type***: El tipo de potenciamiento se mencionó anteriormente.
- ***objective***: Como objetivo se utiliza la regresión para el tipo de problema a abordar en este trabajo.
- ***metric***: La métrica servirá para medir el desempeño y tratar de obtener un modelo con menores pérdidas, puede utilizarse mse (Mean Squared Error) y mae (Mean Absolute Error) para un problema de regresión.
- ***sub_feature***: El subconjunto de características se refiere a que el modelo se puede tomar una fracción de los vectores de características para entrenar cada árbol. Puede ser usado para lidiar con el sobreajuste.
- ***num_leaves***: Define el número máximo de hojas que tendrá cada árbol. Con esto se controla la complejidad del modelo, se pretende tener un número de hojas menor a 2^{max_depth} para evitar el sobreajuste.
- ***min_data***: El número mínimo de registros que puede tener una hoja. Se suele usar para lidiar con el sobreajuste.

- ***max_depth***: La máxima profundidad de los árboles, es por esto que tiene tanta relación con el número de hojas del árbol.
- ***num_iterations***: El número de iteraciones del *boosting*, se relaciona directamente con el número de árboles que se están creando.
- ***early_stopping_rounds***: Las rondas de frenado pronto hacen que el modelo no se sobreajuste al alcanzar todas las iteraciones mencionadas arriba, esto viendo que el modelo tenga mejor rendimiento respecto a la métrica que se le pase a cada paso de este parámetro.

Se espera obtener un modelo lo suficientemente bueno que permita pronosticar futuras erupciones de dicho volcán con datos desconocidos únicamente sabiendo las mediciones de las velocidades de tierra obtenidos por 10 sensores diferentes, sin saber su ubicación sobre el volcán, altura u otro agregado. Actualmente se pueden predecir erupciones volcánicas con minutos de anticipación, se espera tener un buen modelo que pueda hacer pronóstico con más tiempo de anticipación.

Con un modelo implementado, es importante poner producción para que haya cierta facilidad de uso de parte de cualquiera que esté interesado en saber los resultados para el modelo y los pronósticos asociados.

2. Desarrollo

2.1. Mapeo del sistema

Para obtener información del interior del planeta es importante monitorear los cambios temporales asociados a las subestructuras superficiales ya que las velocidades sísmicas son sensibles a las propiedades de los minerales y a los esfuerzos [17]. Recientemente la interferometría sismica, que se encarga de examinar el fenómeno de interferencia en general entre un par de señales para así poder obtener información acerca de un fenómeno sismológico, ha ido evolucionando de tal modo que existen métodos para obtener la función de Green entre dos señales de sensores y así poder hacer mejores modelos [18][19]. Las funciones de green son muy utilizadas en la física, en especial cuando se tienen funciones de correlación, que en este caso nos referimos a las funciones de autocorrelación para ambientes sísmicos ruidosos, por lo que de este modo las señales que se obtienen de mediciones de sensores en la superficie volcánica son modeladas con varias suposiciones válidas mediante estas funciones. En la referencia [20] se reporta una reducción en la velocidad antes de una erupción en la isla *La Réunion*, por lo que es válido tomar en cuenta la forma de las señales antes de una erupción. De este modo, el problema a resolver consiste en extraer características de las señales para estimar en cuánto tiempo el volcán entrará nuevamente en erupción.

Sensor	count	mean	std	min	max
sensor_1	264318629.0	-0.528839	1655.230584	-32767.0	32767.0
sensor_2	215294802.0	1.367050	2272.633675	-32767.0	32767.0
sensor_3	239365260.0	-1.317182	1640.209644	-32767.0	32767.0
sensor_4	265858221.0	-0.115193	1544.979419	-32767.0	32767.0
sensor_5	216622925.0	3.485285	711.074217	-32767.0	32767.0
sensor_6	265775527.0	0.139576	1451.132223	-32767.0	32767.0
sensor_7	263317804.0	-0.153654	974.698587	-32767.0	32767.0
sensor_8	239398859.0	-0.007276	990.390372	-32767.0	32767.0
sensor_9	255344318.0	-0.105223	1770.637589	-32767.0	32767.0
sensor_10	263968458.0	-0.330716	2109.052451	-32767.0	32767.0

TABLA 1: Datos generales de los sensores.

2.2. Definición de métricas adecuadas

Para medir el desempeño del modelo, se tomaron en cuenta dos métricas. La primera, que se usó en el entrenamiento y validación, es el error medio absoluto

$$\text{MAE} = \frac{1}{n}\sum_{i=1}^{n}|\hat{y}_i - y_i| \tag{1}$$

También se utilizó la raíz del error medio cuadrático, conocido por sus siglas en inglés como RMSE

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2} \tag{2}$$

2.3. Exploración de datos

Se cuenta con un conjunto de lecturas de 10 sensores en archivos con extensión `.csv`. Cada archivo contiene 60,000 lecturas de cada sensor realizadas en un intervalo de 10 minutos, por lo que cada lectura corresponde a un centisegundo medido respecto a la lectura anterior, es decir, el fragmento que corresponde de dividir un segundo en 100 partes. Dichos archivos se encuentran divididos en dos partes: entrenamiento y prueba. Los archivos de entrenamiento constan de 4, 431 archivos `.csv`, mientras que los de prueba son 4, 520.

Como se mencionó al principio, los datos de los sensores corresponden a velocidades de la tierra, normalizados a 16 bits. En la tabla 1 podemos ver algunas estadísticas por sensor y en la figura 4 podemos ver la visualización de uno de los archivos.

Adicionalmente, se cuenta con un archivo que contiene el id del archivo y el tiempo de la próxima erupción, el cual se mide a partir de la última lectura del intervalo de los 10 minutos y se indica que la unidad es "número de lecturas", es decir, en centisegundos. En otras palabras, primero se toman lecturas por 10 minutos, y luego transcurre el tiempo de erupción anotado.

Como se puede ver en la figura 5, se podría pensar que pareciera uniforme excepto por el límite superior (hasta la

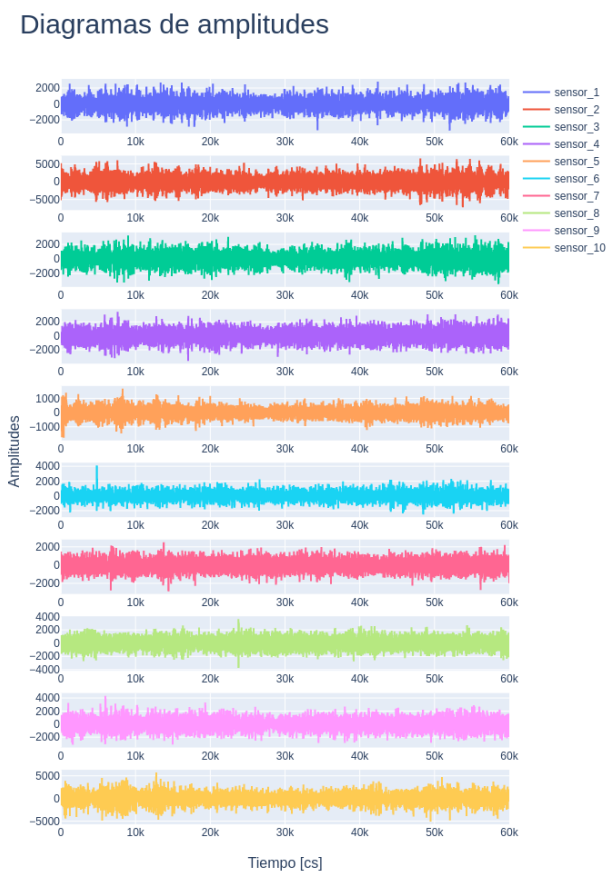

FIGURA 4: Visualización archivo de las lecturas de los 10 sensores.

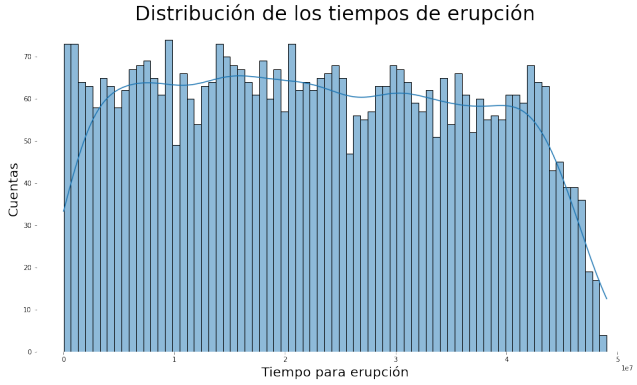

FIGURA 5: Distribución de los tiempos de erupción

derecha) que desciende abruptamente. Significa que hay menos archivos con tiempos de erupción pequeños.

Se hizo un procedimiento para ver los nulos por archivo y si en algún archivo se tenía alguna señal que fuera constante. Los nulos parecían no tener nada de extraordinario en ninguno de los dos conjuntos de datos, sin embargo, se detectó que el sensor 10 se tenían bastantes archivos (814) con señales planas no nulas pero únicamente los datos de prueba, los de entrenamiento no tenían ninguna señal de este tipo.

2.4. Preparación de datos

2.4.1. Completando datos faltantes

Cada archivo podía tener varios datos faltantes de distintos sensores. La señal de un sensor podía tener varios intervalos de datos faltantes o hasta los 10 minutos de lectura. Este problema de los datos faltantes se solventó con dos opciones: la primera fue rellenando con ceros, y la segunda se describe a continuación.

Se contempló una estrategia algo compleja para rellenar los datos faltantes. Si la medición del sensor carece de todos los valores, rellenar con la media. De lo contrario:

- Si hay valores nulos en los primeras observaciones, rellenar con la media.
- Identificar índice del primer dato que no es nulo en la serie. Lo mismo para el último. Si hay nulos de por medio, rellenar con una interpolación por splines.
- Para el resto de nulos, rellenarlos por media de un modelo de Holt - Winters.

El problema con dicha estrategia es que se tardaba bastante por la parte de la interpolación cuando el número de datos faltantes era pequeño. Esto se debe a que es costoso computacionalmente ajustar splines a muchos datos.

Visto lo anterior, se descartó la interpolación, quedando el método para rellenar de la siguiente manera:

Se debe localizar el índice del último dato no nulo, luego, se utiliza la media para rellenar aquellos datos faltantes que existan desde el inicio hasta dicho índice. Después, se realiza el mismo procedimiento de Holt - Winters descrito anteriormente. Esta metodología era más veloz.

Para el modelo de Holt - Winters, se deben revisar dos cosas:

- Tendencia: Se refiere a si la media va cambiando con el tiempo, es decir, si la señal en general va subiendo o bajando. En este caso es cero, porque la media no cambia con el tiempo (o sea no va subiendo y/o bajando)
- Periodicidad: Se refiere a patrones que se repitan cada determinado pasos de tiempo. Se puede encontrar con una gráfica de autocorrelación (como el de la figura 6). Consiste en calcular la correlación respecto los datos y estos mismo pero desplazados una cierta cantidad de unidades en el tiempo. Correlaciones altas indican posibles periodos.

  Correlaciones altas indican posibles periodicidades

En la figura 7 se ve un ejemplo de como se vería la señal, completada con la estrategia descrita.

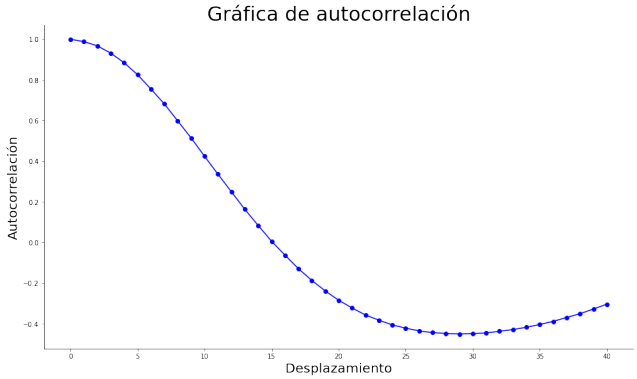

FIGURA 6: Autocorrelación

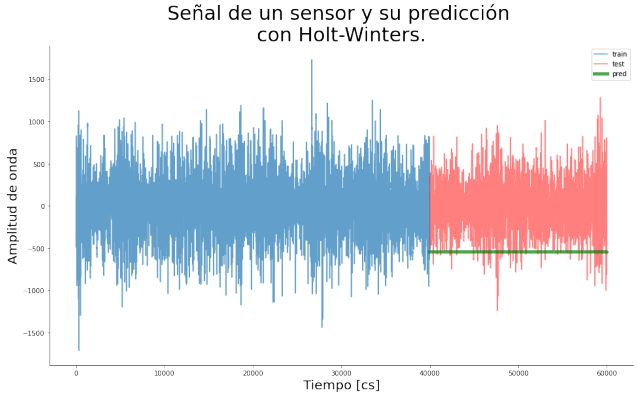

FIGURA 7: Holt-winters como posible estrategia de imputación

2.4.2. Estandarización

A los datos se les aplicó una estandarización con la finalidad de centrarlos. Esto se hace restando la media y dividiendo sobre la desviación estándar:

$$Z = \frac{X - \bar{x}}{s} \tag{3}$$

2.4.3. Extracción de características

Para formar los vectores de características que se le pasarían al modelo se eligieron las siguientes características que son muy específicas para señales:

- *zero crossing rate*: es la proporción de qué tanto cambia de signo la señal, es decir, de positivo a negativo o viceversa
- Número de picos: Se buscan los picos de la señal comparando con valores cercanos.
- Prominencias en los picos (media y máximo): La prominencia de un pico mide que tanto sobresale de la señal y se define como la distancia vertical entre el pico y la linea de contorno más baja.

  La manera de calcularlo es de la siguiente manera:

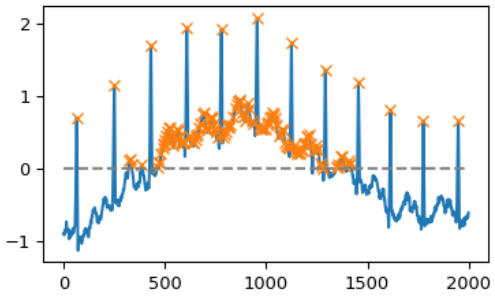

FIGURA 8: Ejemplo de picos.

1. Situado en el pico, se dibuja una línea horizontal que se extiende por ambos lados. Se extiende hasta que se llega al fin (o principio) de la señal ó hasta encontrarse con alguna pendiente (en palabras mundanas, hasta chocar).
2. Encontrar el mínimo por la izquierda y por la derecha del pico.
3. Se elige el máximo de los mínimos encontrados, y la diferencia entre la altura del pico[1] y el máximo es la prominencia.

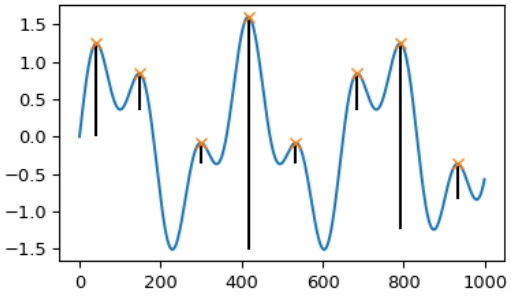

FIGURA 9: Ejemplo de prominencias

- Ancho de los picos (media y máximo): Se calcula como

$$\text{ancho} = h_{\text{pico}} - PR \tag{4}$$

  Donde P es la prominencia y R es un valor entre 0 y 1 que indica a qué altura relativa se mide la anchura como porcentaje de la prominencia. Por ejemplo, con $R = 0.5$ se computa el valor tomando la mitad de la altura de la prominencia.
- *periodogram* (media y máximo): La raíz del máximo es un aproximación de la raíz de la media al cuadrado, o por sus siglas, Root Mean Square (RMS).

[1] La altura se mide a partir del cero.

2.5. Métodos y modelos

Una vez hecho el preprocesamiento de los datos y el análisis de señales, se procedió a implementar el modelo planteado, para esto se utilizaron variantes de los vectores de características dependiendo del preprocesamiento de los datos. Primero se entrenó un modelo con los datos tal cual eran obtenidos, también se entrenó un modelo únicamente con la estandarización y transformando los datos faltantes a puros ceros, después de haber pasado por el proceso de señales. Finalmente se utilizó el proceso de imputación visto en la sección 2.4.1 y posteriormente una estandarización para después obtener las características de señales mencionadas al final de la sección anterior. Como primer modelo se utilizó el LGBM así sencillo, más concretamente se utilizó la *trainig API* de la biblioteca `lightgbm` de `python` [21] con unos parámetros aleatorios encontrados en la referencia [22]. El modelo se entrenaba partiendo a los datos de entrenamiento dados en el concurso en datos de entrenamiento y datos de validación.

2.6. Evaluación de modelos

Como no se tenía idea acerca de los hiperparámetros a utilizarse y ni siquiera algún intervalo para cada uno de estos, se optó por utilizar una búsqueda aleatoria más que una búsqueda por cuadrícula. Para la búsqueda aleatoria se implementó código para fijar los parámetros que se mencionaron al final de la sección 1 tal como se muestra a continuación:

- ***learning_rate***: $\sim U(0,1)$
- ***boosting_type***: Se escogió entre *GBDT*, *DART* y *GOSS*.
- ***objective***: Este se quedó fijo en *regression*.
- ***metric***: Este se quedó fijo en *MAE*.
- ***sub_feature***: $\sim U(0,1)$.
- ***num_leaves***: $\sim U(20,300)$ de enteros.
- ***min_data***: $\sim U(1,100)$ de enteros.
- ***max_depth***: $\sim U(5,200)$ de enteros.
- ***num_iterations***: Este se quedó fijo en 10,000, ya que se pensaba en un número y que tendría mejor rendimiento.

Al hacer un primer análisis de los modelos que resultaban de esta búsqueda aleatoria se tuvieron que cambiar la selección aleatoria de algunos hiperparámetros: Se mostró que la tasa de aprendizaje resultó en mejores resultados mientras más pequeña fuese (véase Figura 10), por lo que se cambió a $\sim U(0,0.2)$ además de que se eliminó por completo la estrategia de *GOSS* de los tipos de potenciamiento a elegir, ya que en general mostraba peores resultados.

Posteriormente, como se notó que existía un sobreajuste se optó primero por agregar a los hiperparámetros la variable *early_stopping_rounds* y que así el modelo no hiciera tantas iteraciones, de modo que no se sobreajustaba tanto a los

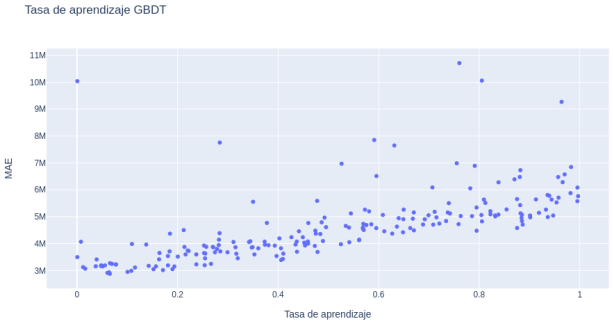

FIGURA 10: Se muestra cómo influye la tasa de aprendizaje en el error del modelo.

datos de entrenamiento. Una limitación de esta estrategia es que no se permite ser utilizada con la metodología de potenciamiento *DART*, por lo que la única opción viable ahora era *GBDT*.

Con estos hiperparámetros para elegir y con el problema de sobreajuste, se optó por una estrategia de validación cruzada para intentar generalizar más el modelo de aprendizaje. Se utilizó la misma API de entrenamiento pero usando el método de `.cv()` que hacía automáticamente la validación cruzada tomando la métrica que le diéramos (MAE en este caso) y recibiendo todo el conjunto de entrenamiento junto con sus etiquetas, que en este caso eran el tiempo para la erupción.

3. Resultados y análisis

Los mejores resultados que se obtuvieron con las distintas técnicas de preprocesamiento se muestran en la tabla 2.

Es importante notar que los mejores modelos respecto al entrenamiento y las pruebas se muestran en colores en la Tabla 2, se observa que hay dos diferentes modelos que toman el primero lugar dependiendo si es el conjunto de pruebas o de entrenamiento, sin embargo, vemos que el segundo lugar lo ocupa otro modelo que tiene buenos resultados tanto en entrenamiento como en pruebas.

Se obtuvo como error medio absoluto MAE $= 695,174.9cs$, que es aproximadamente 1 hora, 55 minutos y 52 segundos. En la figura 11 se puede ver un gráfico de violín para este error

Para el error cuadrático, se obtuvo que RMSE $= 2034299.51cs$, que es aproximadamente 5 horas, 39 minutos y 3 segundos. En la figura 12 se puede ver un gráfico de violín para este error

En la figura 13 se observan tres histogramas: uno es la distribución de los datos de entrenamiento (rojo), otra es la distribución aprendida por el modelo (azul) y la última es la distribución de las predicciones en el conjunto de prueba (naranja).

	Sin CV ni ESR, datos estandarización.	Con CV y ESR, datos sin estandarizar.	Con CV y ESR, datos estandarizados.	Con CV y ESR, datos imputados y estandarizados.	Con CV y ESR, datos sin estandarizar 0.75 % de los Datos
boosting_type	*DART*	*GBDT*	*GBDT*	*GBDT*	*GBDT*
learning_rate	0.1412	0.0048	0.0224	0.0029	0.0289
sub_feature	0.5487	0.73	0.4677	0.464	0.541
num_leaves	43	186	149	272	126
min_data	23	8	6	13	9
max_depth	68	112	13	125	64
Iteraciones	10,000	4193	2181	4643	781
MAE_train	2,547,251.82	2,182,416.74	2,545,143.14	2,555,633.41	2,857,459.40
MAE_test	4,915,508	5,069,405	5,066,116	5,139,018	5,255,184

TABLA 2: Hiperparámetros con los que se obtuvieron mejores resultados. La última fila muestra los resultados que se tuvieron al comparar con los tiempos de erupción reales, estos resultados los arrojaba la misma página de *kaggle*.

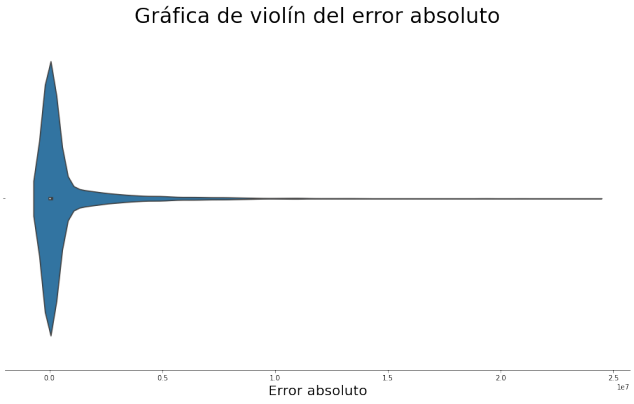

FIGURA 11: Error absoluto

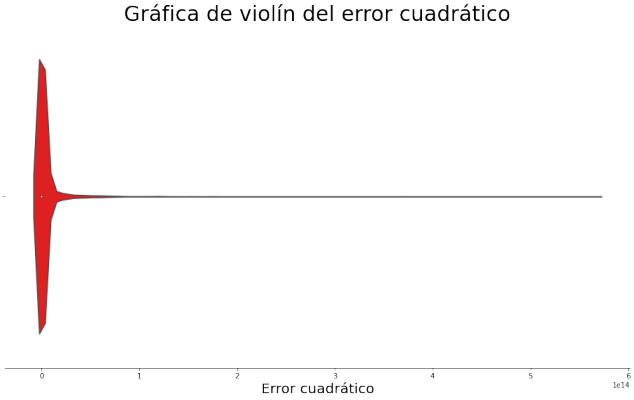

FIGURA 12: Error cuadrático

Lo principal que hay que notar es que las distribuciones que se tienen para los conjuntos de entrenamiento y de prueba son muy diferentes. La de entrenamiento es casi uniforme exceptuando por la última parte mientras que la segunda presenta más la forma de campana pero sesgada negativamente,

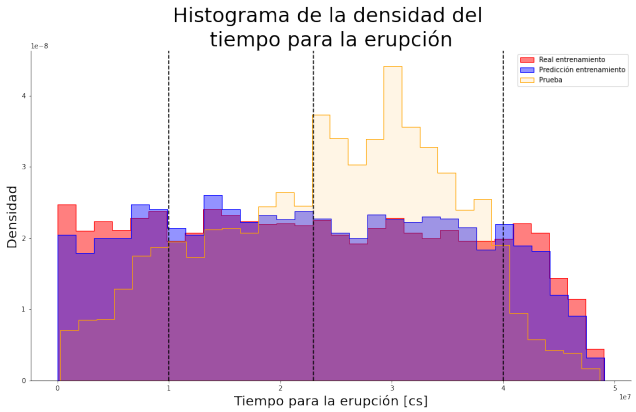

FIGURA 13: Resultados

es decir, a la izquierda.

Por un lado, se ve que el modelo aprendió la distribución de los datos de entrenamiento. Por otro lado, se observa que la distribución de las predicciones no corresponde a la distribución de los datos de entrenamiento. Esto lleva a pensar que los datos fueron alterados de alguna manera cuya naturaleza no es del todo aleatoria o no se refiere al mismo fenómeno.

Tendría que analizarse el fenómeno para conocer cual de las dos distribuciones es más adecuada. Dado que se tienen las características y los tiempos de erupción para el entrenamiento, es razonable pensar que la distribución aprendida es la que caracteriza al fenómeno.

4. Implementación

Para poner el modelo en producción, se eligió streamlit como plataforma, ya que permite montar una pequeña aplicación fácil de usar.

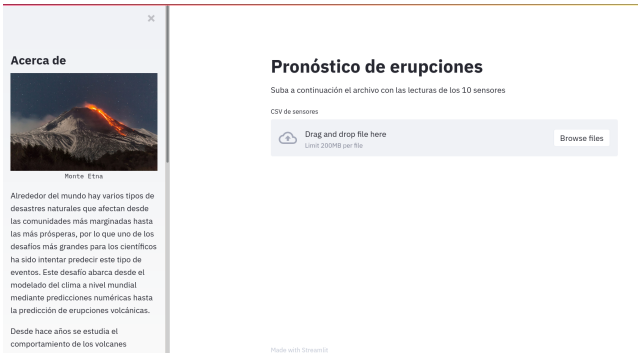

FIGURA 14: Aplicación

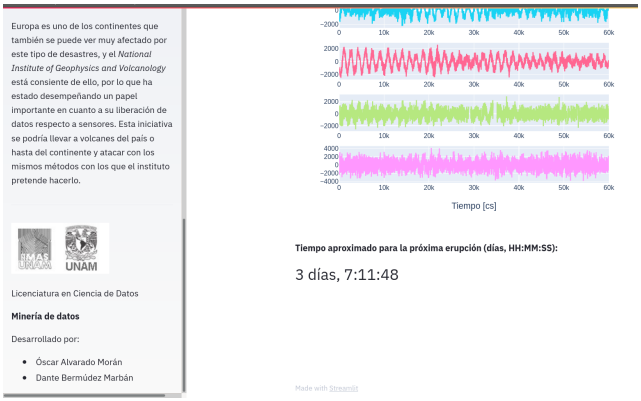

FIGURA 15: Aplicación tras subir csv

La implementación consistió de poner el modelo en *batch*, es decir, el usuario sube el archivo con las lecturas de los 100 sensores. Tal y como se ve en la figura 14.

Al subir un archivo válido, el usuario obtiene una tabla con estadísticas, junto con una visualización de las señales como la de la figura 4 y la predicción en un formato de días, horas, minutos y segundos, tal como se ve en la figura 15.

La aplicación cuenta con una barra lateral con información del problema que se resolvió.

5. Conclusiones

Dicho lo anterior, se plantea que parte del error puede ser causado por una mala división de entrenamiento y prueba. Respaldado por el proceso que se hizo de análisis de nulos y de distribuciones de las señales, indicando que en los datos de prueba había señales planas, mientras que en los de entrenamiento no.

Respecto a las predicciones en datos no vistos, el modelo es más confiable con predicciones entre 100,000 y 230,000 segundos (1 y 2.3 en la gráfica de la figura 13), lo cual se traduce entre 27 y 64 horas, aproximadamente.

Por otro lado, predicciones antes de 100,000 segundos y después de 230,000 ya no son tan confiables. En el caso de los tiempos anteriores a 100,000 segundos, dado que hay menos de los que debería, se debe entender que es probable que la erupción ocurra más tarde, así como cierto intervalo de los tiempos de erupción que se tienen después de 400,000 segundos. Los tiempos de erupción que se obtienen de 230,000 y 400,000 segundos, dado que hay más de los que debería, se entiende que es probable que la erupción ocurra más temprano. Se recomienda explorar el uso de *DART* como *boosting_type* debido a que presenta mejor exactitud en los resultados aunque no se pueda usar *early_stopping_rounds* con este método, quizás se podría tratar de eliminar el sobreajuste mediante otras técnicas, aunque en sí es el potenciamiento que menor sobreajuste presenta según la teoría. Sería importante también tomar en cuenta que el modelo funciona mejor con más datos debido a su sensibilidad a sobreajustarse, por lo que se recomieda su uso con más que los datos utilzados en este trabajo para el entrenamiento (4,431).

Referencias

[1] The Royal Society. *People of Science with Brian Cox - Professor Joanna Haigh on Lewis Fry Richardson*. https://www.youtube.com/watch?v=0MdUDpOvVr4. 2020 (Recuperado el 14-12-2020).

[2] L. Wilson y J. Head. "A comparison of volcanic eruption processes on Earth, Moon, Mars, Io and Venus". En: *Nature* 302 (1983), págs. 663-669. DOI: https://doi.org/10.1038/302663a0.

[3] B. Voight. "A method for prediction of volcanic eruptions. Nature 332, 125-130". En: *Nature* 332 (feb. de 1988), págs. 125-130. DOI: 10.1038/332125a0.

[4] K Nagamine y col. "Method of probing inner-structure of geophysical substance with the horizontal cosmic-ray muons and possible application to volcanic eruption prediction". En: *Nuclear Instruments and Methods in Physics Research Section A: Accelerators, Spectrometers, Detectors and Associated Equipment* 356.2 (1995), págs. 585-595. ISSN: 0168-9002. DOI: https://doi.org/10.1016/0168-9002(94)01169-9. URL: http://www.sciencedirect.com/science/article/pii/0168900294011699.

[5] Glyn Williams-Jones y Hazel Rymer. "Detecting volcanic eruption precursors: a new method using gravity and deformation measurements". En: *Journal of Volcanology and Geothermal Research* 113.3 (2002), págs. 379-389. ISSN: 0377-0273. DOI: https://doi.org/10.1016/S0377-0273(01)00272-4. URL: http://www.sciencedirect.com/science/article/pii/S0377027301002724.

[6] Hazel Rymer y Glyn Williams-Jones. "Volcanic eruption prediction: Magma chamber physics from gravity and deformation measurements". En: *Geophysical Research Letters* 27 (ago. de 2000). DOI: 10.1029/1999GL011293.

[7] R.S.J. Sparks. "Forecasting volcanic eruptions". En: *Earth and Planetary Science Letters* 210.1 (2003), págs. 1-15. ISSN: 0012-821X. DOI: https://doi.org/10.1016/S0012-821X(03)00124-9. URL: http://www.sciencedirect.com/science/article/pii/S0012821X03001249.

[8] Andrew Bell y col. "Forecasting volcanic eruptions and other material failure phenomena: An evaluation of the failure forecast method". En: *GEOPHYSICAL RESEARCH LETTERS* 38 (ago. de 2011). DOI: 10.1029/2011GL048155.

[9] F. B. Wadsworth y col. "Explosive-effusive volcanic eruption transitions caused by sintering". En: *Science Advances* 6 (2020).

[10] Volcano Discovery. *Interactive map of currently active volcanoes*. https://www.volcanodiscovery.com/volcano-map.html. 2020 (Recuperado el 14-12-2020).

[11] BBC News Mundo. *Volcán de Fuego: 10 de los volcanes más peligrosos de América Latina*. https://www.bbc.com/mundo/noticias-america-latina-44357073. 2018 (Recuperado el 14-12-2020).

[12] *National Institute of Geophysics and Volcanology*. *INGV - Volcanic Eruption Prediction*. https://www.kaggle.com/c/predict-volcanic-eruptions-ingv-oe. Recuperado el 14-12-2020.

[13] Instituto Nazionale Di Geofisica E Vulcanologia. *Instituto Nazionale Di Geofisica E Vulcanologia*. https://www.bbc.com/mundo/noticias-america-latina-44357073. (Recuperado el 14-12-2020).

[14] Guolin Ke y col. "LightGBM: A Highly Efficient Gradient Boosting Decision Tree". En: *31st Conference on Neural Information Processing System* 30 (2017). Ed. por I. Guyon y col., págs. 3146-3154. URL: https://proceedings.neurips.cc/paper/2017/file/6449f44a102fde848669bdd9eb6b76fa-Paper.pdf.

[15] Jerome H. Friedman. "Greedy function approximation: A gradient boosting machine." En: *Ann. Statist.* 29.5 (oct. de 2001), págs. 1189-1232. DOI: 10.1214/aos/1013203451. URL: https://doi.org/10.1214/aos/1013203451.

[16] Mateusz Susik. *Can one do better than XGBoost?* https://www.youtube.com/watch?v=5CWwwtEM2TA&feature=emb_logo. 13 de Noviembre de 2017 (Recuperado el 14-01-2021).

[17] Y. Yukutake, Tomotake Ueno y Kazuki Miyaoka. "Determination of temporal changes in seismic velocity caused by volcanic activity in and around Hakone volcano, central Japan, using ambient seismic noise records". En: *Progress in Earth and Planetary Science* 3 (dic. de 2016), pág. 29. DOI: 10.1186/s40645-016-0106-5.

[18] Michel Campillo y Anne Paul. "Long-Range Correlations in the Diffuse Seismic Coda". En: *Science (New York, N.Y.)* 299 (feb. de 2003), págs. 547-9. DOI: 10.1126/science.1078551.

[19] Nikolai Shapiro y col. "High-Resolution Surface-Wave Tomography from Ambient Seismic Noise". En: *Science (New York, N.Y.)* 307 (abr. de 2005), págs. 1615-8. DOI: 10.1126/science.1108339.

[20] Florent Brenguier y col. "Toward forecasting volcanic eruption using seismic noise". En: *Nature geoscience* 1 (ene. de 2008). DOI: 10.1038/ngeo104.

[21] Microsoft Corporation. *LightGBM*. https://lightgbm.readthedocs.io/en/latest/Python-Intro.html. Copyright 2021, Microsoft Corporation Revision f997a069. (Recuperado el 14-01-2021).

[22] Pushkar Mandot. *What is LightGBM, How to implement it? How to fine tune the parameters?* https://cutt.ly/pjEQDiL. 17 de Agosto de 2017 (Recuperado el 14-01-2021).