PREDICCIÓN DE EMISIONES DE CO_2 EN RUANDA

POR:
Samuel García Bonilla
Jhon Fredy Hoyos Cardenas
Juan José Bustamante Betancur

MATERIA:
Introducción a la Inteligencia Artificial para las Ciencias e Ingenierías

PROFESOR:
Raul Ramos Pollan

UNIVERSIDAD DE ANTIOQUIA
FACULTAD DE INGENIERÍA
Medellín 2023

Contenido

Introducción

La generación de modelos predictivos es esencial en el campo de la inteligencia artificial (IA), implicando la creación de algoritmos y sistemas capaces de analizar datos históricos para prever eventos futuros. Estos modelos, que emplean técnicas como aprendizaje automático y redes neuronales, identifican patrones en los datos. Su aplicación abarca diversos campos, desde pronósticos meteorológicos hasta diagnóstico médico. Los modelos predictivos son fundamentales para la toma de decisiones basadas en datos, permitiendo anticipar tendencias y optimizar operaciones en organizaciones. En este proyecto, se utiliza la herramienta Colab para entrenar modelos predictivos con un conjunto de datos de Kaggle. La competencia seleccionada aborda la predicción de emisiones de CO2 en Ruanda, con datos de múltiples ubicaciones y áreas. La tarea consiste en predecir las emisiones desde 2022 hasta noviembre, utilizando diversas bibliotecas de Python para lograr la mejor predicción.

Para esta última entrega se muestra el proceso de las iteraciones realizadas. La primera iteración corresponde a la segunda entrega del proyecto en donde se entrena el primer modelo y se obtiene una respuesta con un error bastante elevado. Por ello, se procede a realizar la segunda iteración, en donde se incluyen demás factores como la correlación de variables y la prueba de diferentes modelos de predicción para elegir el que mejor rendimiento tenga. Se obtiene además la curva aprendizaje y se realizan los análisis y las conclusiones respectivas.

1. Planteamiento del problema

Los niveles de emisión de carbono son muy importantes a la hora del estudio del calentamiento global, por lo tanto, la predicción de estas emisiones es muy importante. La precisión en estas lecturas permite a las entidades encargadas conocer diferentes tipos de información que puede ser muy relevante a la hora de realizar el análisis respectivo. Mientras que Europa y Norteamérica cuentan con amplios sistemas de seguimiento de las emisiones de carbono sobre el terreno, en África se dispone de muy pocos.

El objetivo es crear un modelo automatizado sobre los datos de emisiones de CO_2 procedentes de observaciones hechas por el satélite Sentinel-5P para predecir futuras emisiones de carbono. Estas soluciones pueden ayudar a los gobiernos y a otros actores a estimar los niveles de emisiones de carbono en África, incluso en lugares donde no es posible el seguimiento sobre el terreno. Para este problema, se utilizarán datos específicamente de Ruanda.

1.1. Dataset

Se hizo una búsqueda de un dataset, en la plataforma Kaggle, que reuniera los requerimientos necesarios para poder llevar a cabo el proyecto (aquí se puede encontrar la página de la competencia).

Para el dataset se seleccionaron aproximadamente 497 ubicaciones únicas de múltiples zonas de Ruanda, con una distribución en torno a tierras de cultivo, ciudades y centrales eléctricas. Los datos para esta competición se dividen por tiempo; los años 2019 - 2021 se incluyen en los datos de entrenamiento (train.csv), y el objetivo es predecir los datos de emisiones de CO_2 de 2022 a noviembre.

Se extrajeron siete características principales semanalmente del Sentinel-5P desde enero de 2019 hasta noviembre de 2022. Cada característica (dióxido de azufre, monóxido de carbono, etc.) contiene subcaracterísticas como column_number_density, qué es la densidad de columna vertical a nivel del suelo, calculada mediante la técnica DOAS. Se dan los valores de estas características en el conjunto de pruebas y se busca predecir las emisiones de CO_2utilizando información temporal, así como estas características.

- Dióxido de azufre
- Monóxido de carbono
- Dióxido de nitrógeno
- Formaldehído

- Ozono
- Nube

Resumiendo, se cuenta con tres archivos, uno de entrenamiento (train.csv), otro de prueba (test.csv) y un último archivo de comparación (sample_submission.csv) con el cual se podrá conocer la exactitud de las predicciones. Ambos archivos de entrenamiento y de testeo (train.csv y test.csv) contienen la siguiente información:

- **ID_LAT_LON_YEAR_WEEK -** El id, la latitud y la longitud del lugar específico, el año y la semana en la que se presenta la emisión.
- **latitude -** Latitud del lugar de estudio.
- **longitude -** Longitud del lugar de estudio.
- **year -** Año en el que se realizó el estudio.
- **week_no -** Número de la semana
- **SulphurDioxide -** Variables con características de interés para el dióxido de sulfuro.
- **CarbonMonoxide -** Variables con características de interés para el monóxido de Carbono.
- **NitrogenDioxide -** Variables con características de interés para el Dióxido de Nitrógeno.
- **Formaldehyde -** Variables con características de interés para el Formaldehído.
- **Ozone -** Variables con características de interés para el Ozone.
- **Cloud -** Variables con características de interés para el Cloud.
- **emission -** Es el resultado de la emisión predicha por el modelo. (solo en el archivo train.csv)
- **Quartile -** Define el cuartil al cual pertenece la época del año.
- **Month -** Mes estimado de acuerdo al número de semanas.

El archivo final es el archivo sample_submission.csv el cual tendrá la siguiente información:

ID_LAT_LON_YEAR_WEEK - El id, la latitud y la longitud del lugar específico, el año y la semana en la que se presenta la emisión que identifica las características en el train.csv.
emission - Es el resultado de la emisión predicha por el modelo.

1.2. Métrica

La métrica de evaluación es el error medio cuadrático o root mean squared error (RMSE) y está definido como:

$$RSME = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}$$

donde N es el número total de observaciones, $\hat{y}_i$ es el valor predecido y y_i es el valor original para cada instancia i.

El error medio cuadrático (RMSE) es una métrica que se utiliza para medir la diferencia promedio entre las predicciones de un modelo y los valores reales. Se calcula tomando la raíz cuadrada de la media de los errores al cuadrado. Cuanto menor sea el RMSE, mejor se ajusta el modelo a los datos, ya que indica que las predicciones son cercanas a los valores reales.

Se obtuvo un resultado de 21.99 en la métrica lo que nos quiere decir que, en promedio, las predicciones del modelo están a 21.99 unidades de los valores reales.

1.3. Variable objetivo: "emission"

Formalizando lo mencionado anteriormente, la variable objetivo que se desea predecir en "emission", la cual nos da como resultado la predicción de las emisiones de CO_2 a partir de los diferentes gases o moléculas (variables) que se tienen en la información reportada en el dataset.

2. Tratamiento de datos

2.1. Reemplazo de datos faltantes

Dentro del Dataset se pudieron encontrar distintos valores faltantes en varias columnas de las variables independientes. Para su reemplazo se optó por darle un valor igual al promedio del dato anterior y del dato siguiente.

A forma de visualización se pueden observar en la Tabla 1 el porcentaje de valores nulos por variable. No se ponen todas las tablas debido a la gran extensión de variables.

Tabla 1. Porcentaje de datos faltantes por variable

index	Total	Percent
NitrogenDioxide_solar_zenith_angle	18320	23.183123900636524
NitrogenDioxide_NO2_column_number_density	18320	23.183123900636524
NitrogenDioxide_solar_azimuth_angle	18320	23.183123900636524
NitrogenDioxide_sensor_zenith_angle	18320	23.183123900636524
NitrogenDioxide_sensor_azimuth_angle	18320	23.183123900636524
NitrogenDioxide_cloud_fraction	18320	23.183123900636524
NitrogenDioxide_absorbing_aerosol_index	18320	23.183123900636524
NitrogenDioxide_tropopause_pressure	18320	23.183123900636524
NitrogenDioxide_NO2_slant_column_number_density	18320	23.183123900636524
NitrogenDioxide_stratospheric_NO2_column_number_density	18320	23.183123900636524
NitrogenDioxide_tropospheric_NO2_column_number_density	18320	23.183123900636524
NitrogenDioxide_sensor_altitude	18320	23.183123900636524
SulphurDioxide_sensor_zenith_angle	14609	18.48702276552396
SulphurDioxide_SO2_column_number_density	14609	18.48702276552396
SulphurDioxide_solar_zenith_angle	14609	18.48702276552396
SulphurDioxide_SO2_column_number_density_15km	14609	18.48702276552396
SulphurDioxide_solar_azimuth_angle	14609	18.48702276552396
SulphurDioxide_sensor_azimuth_angle	14609	18.48702276552396
SulphurDioxide_cloud_fraction	14609	18.48702276552396
SulphurDioxide_SO2_slant_column_number_density	14609	18.48702276552396
SulphurDioxide_SO2_column_number_density_amf	14609	18.48702276552396
Formaldehyde_sensor_azimuth_angle	7277	9.208711387823797
Formaldehyde_sensor_zenith_angle	7277	9.208711387823797
Formaldehyde_solar_azimuth_angle	7277	9.208711387823797
Formaldehyde_solar_zenith_angle	7277	9.208711387823797

2.2. Correlación de variables

La correlación de variables en la construcción de un modelo predictivo determina la relación entre las variables independientes y la variable dependiente. Para este caso se obtuvo la Figura 1, en donde se muestra una matriz de correlación de variables. En ella se observa, dependiendo de la oscuridad del color, qué tanto están correlacionadas las diferentes variables.

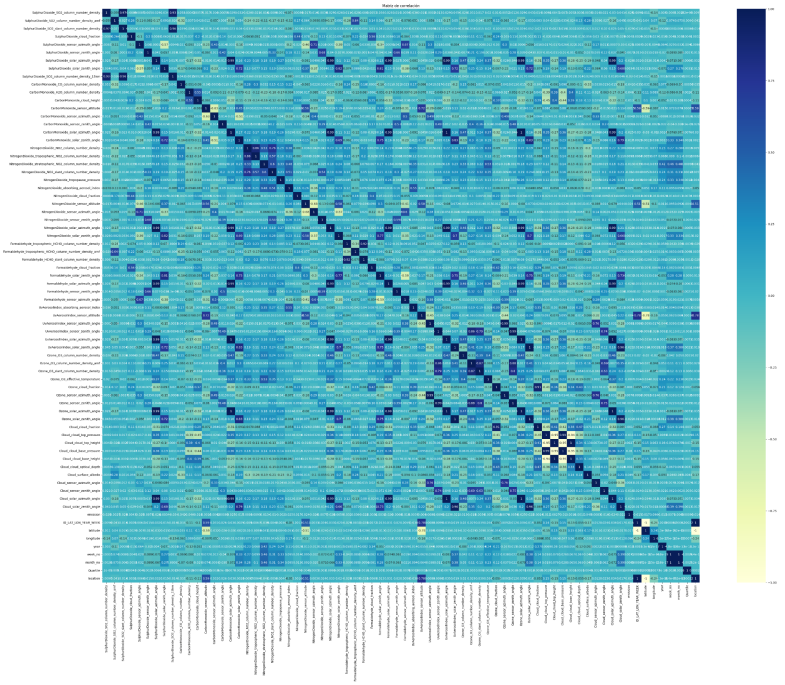

Figura 1. Matriz de correlación de variables.

Dado que el gráfico de la Figura 1 puede ser difícil de visualizar se realiza la Tabla 2, en la cual se permite observar mejor el porcentaje de correlación de las variables independientes con la variable dependiente. En dicha Tabla no se muestran todas las variables debido a que son muchas, se recomienda visualizar desde el código adjunto.

Tabla 2. Porcentajes de correlación de variables con variable objetivo.

index	emission
emission	1.0
longitude	0.09572135911578021
Cloud_surface_albedo	0.048503494478725276
Formaldehyde_tropospheric_HCHO_column_number_density_amf	0.03618731388903605
NitrogenDioxide_absorbing_aerosol_index	0.024679628417427116
NitrogenDioxide_sensor_altitude	0.023798900972664027
NitrogenDioxide_cloud_fraction	0.017427367678887522
latitude	0.01685628719928515
Ozone_O3_column_number_density	0.013278860773711884
NitrogenDioxide_tropospheric_NO2_column_number_density	0.00927976914220908
Cloud_sensor_azimuth_angle	0.008266069320650512
UvAerosolIndex_sensor_azimuth_angle	0.008088405821998403
Ozone_sensor_azimuth_angle	0.0075899634677565634
week_no	0.007085970277214695
month_no	0.006919671881572816
NitrogenDioxide_solar_zenith_angle	0.005906546084612615
SulphurDioxide_cloud_fraction	0.0042941439899283563
UvAerosolIndex_absorbing_aerosol_index	0.0038761574643065284
SulphurDioxide_solar_zenith_angle	0.0033899941457129178
Cloud_cloud_top_height	0.0027491510698715153
Cloud_cloud_base_height	0.0015286345945888684
CarbonMonoxide_sensor_zenith_angle	0.0008344124196995547
NitrogenDioxide_NO2_column_number_density	0.00040343790113245287
Quartile	-0.0009313455651124963
Cloud_cloud_optical_depth	-0.0011200374537392553

De la Tabla 2, se puede observar que casi todas las variables tienen una correlación muy baja con la variable objetivo, en su gran mayoría se acercan mucho a cero.

2.3. Análisis de la variable objetivo

Para este proyecto, la variable objetivo es “emission”, por lo que se toma esta variable del dataset de entrenamiento y se grafica su comportamiento para realizar un análisis de la misma. La gráfica de esta es la Figura 2.

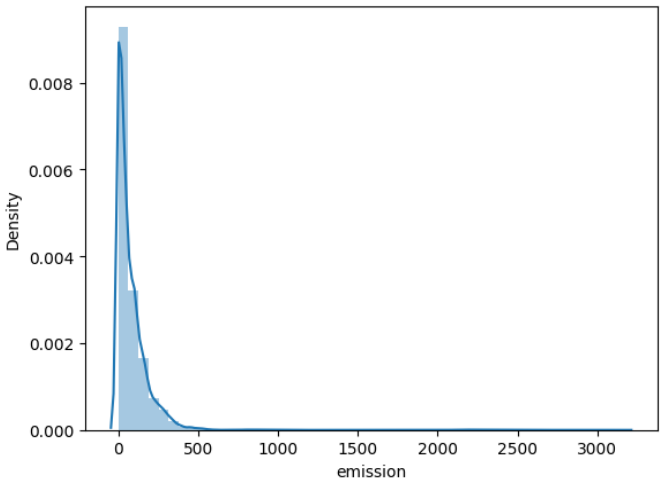

Figura 2. Comportamiento de la variable objetivo “emission”.

De la Figura 2, se puede ver que la variable objetivo se encuentra sesgada hacia la derecha, así que presenta muchas lecturas muy cercanas a cero o que son cero. Esto puede dañar al modelo pues son datos atípicos, así que se aplica una transformación logarítmica a la variable objetivo, obteniendo la gráfica que se muestra en la Figura 3.

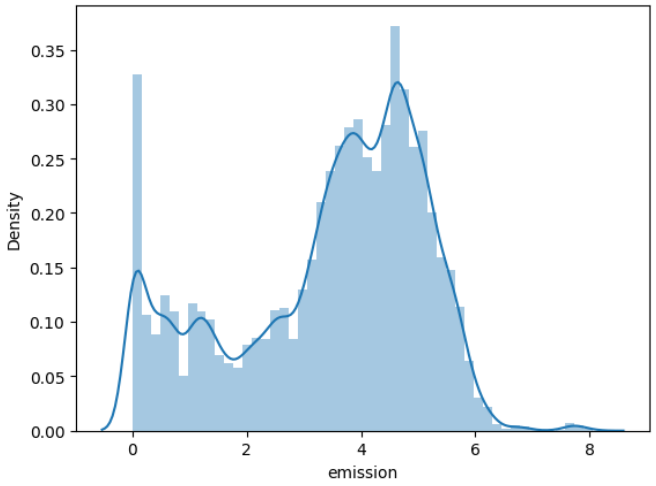

Figura 3. Comportamiento de la variable objetivo con transformación logarítmica.

2.4. Distribución de variables numéricas

En la Figura 4, se puede observar las distintas distribuciones que tiene cada una de las variables del Dataset.

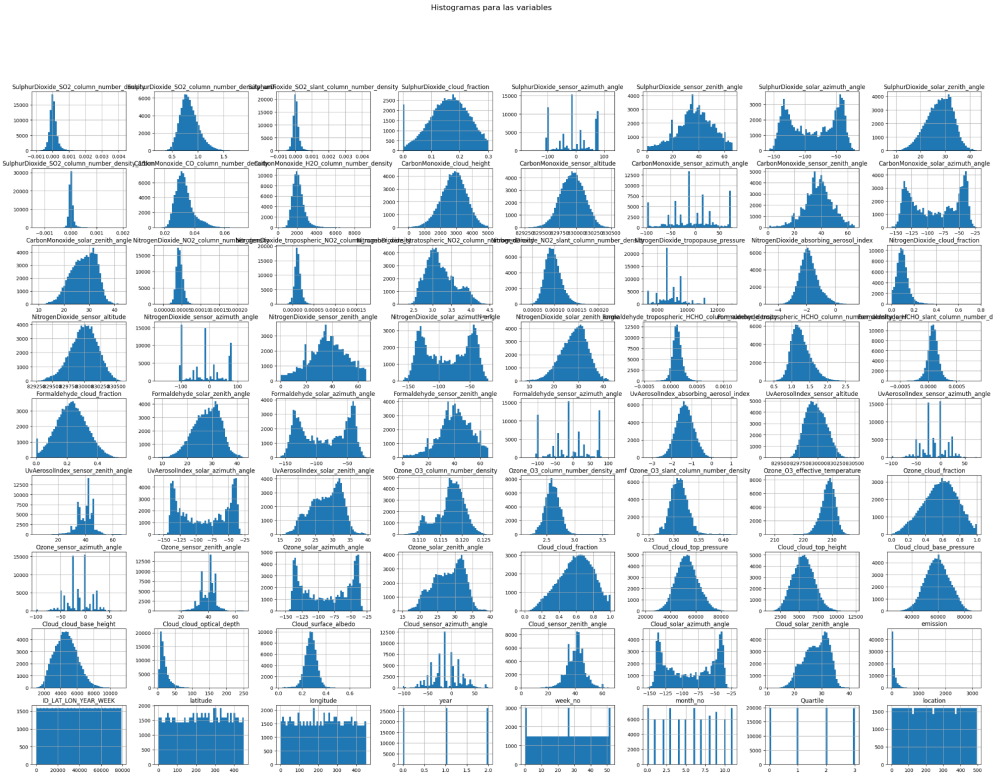

Figura 4. Distribución de las variables numéricas.

La mayoría de las variables tiene una distribución normal como se puede observar. Sin embargo, como se menciona en la sección anterior, la variable objetivo tiene una distribución sesgada hacia a la derecha lo cual implica que hay muchos datos nulos que pueden afectar el modelo, para ello se aplicó la transformación logarítmica.

2.5. Remoción de lecturas cero en la variable objetivo

Luego de realizar el análisis de la variable objetivo en la Figura 3, se puede observar que aún quedan lecturas cero, se requiere eliminar los registros cero faltantes de la misma. Esto debido a que no tiene sentido lecturas tener lecturas de emisiones de CO_2 que sean cero debido a que pueden ser fallos de lectura en los sensores, así que se eliminan estas lecturas.

3. Generación de modelos

3.1. Selección de modelos

3.1.1. Primera iteración

Para la primera iteración, se debe tener en cuenta que solo se realizó el numeral 2.1. correspondiente al tratamiento de datos, es decir, solo se rellenaron los datos faltantes y luego se generó el modelo con la herramienta Random Forest Regressor con 30% de datos para testeo y el resto para entrenamiento, el resultado obtenido aplicando la métrica del problema fue un RMSE de 21.000887.

Lo que quiere decir que la predicción del modelo en promedio está 21 unidades por encima del valor real, por lo que no es un resultado tan satisfactorio y da espacio para mejoras. La comparación entre los valores reales y los predichos se pueden visualizar en la Figura 4.

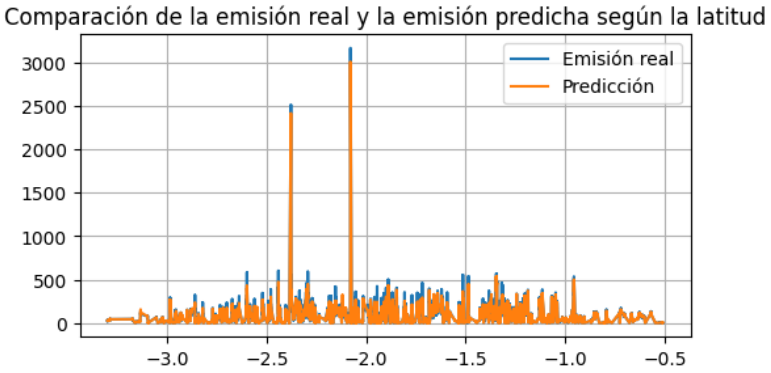

Figura 4. Distribución de las variables numéricas.

3.1.2. Segunda iteración

Efectividad de métodos supervisados

Se busca aplicar métodos supervisados al modelo que permitan definir cuál es el modelo más recomendable a utilizar. Para esto se crea una función de puntaje que permita imprimir las puntuaciones del error cuadrático medio logarítmico (RMSLE) y se define también la métrica para el cálculo. Es importante tener en cuenta que se usa la métrica de RMSLE pues se le aplicó una transformación logarítmica a la variable objetivo.

Se eliminan variables con menor correlación con la variable objetivo

Para continuar con la generación del modelo, se eliminan las variables que tienen una correlación muy pequeña respecto a la variable objetivo.

Partición de los datos

Posteriormente, se reparten los datos de testeo y de entrenamiento en una proporción de 10% / 90% respectivamente. Esto teóricamente permite entrenar el modelo con más datos y obtener un mejor resultado.

Modelos predictivos a evaluar

Se seleccionan los modelos de regresión lineal, árboles de decisión, y RandomForest para la evaluación de su métrica. En la Figura 5, se puede visualizar el código utilizado para esta etapa.

```
# Selección de modelos basados en su métrica

zscores = []
estimadores = [estimador1, estimador2, estimador3]
for estimator in estimadores:
    print("---------")
    z = cross_validate(estimator, Xtv, ytv, return_train_score=True,
                       return_estimator=False,
                       scoring="neg_mean_squared_error",
                       cv=ShuffleSplit(n_splits=10, test_size=tam_val))
    Formato_puntaje(z)
    zscores.append(np.mean(np.sqrt(z['test_score']*(-1))))

best = np.argmin(zscores)
print ("Seleccionado: ", best+1)
best_estimator = estimadores[best]
print ("\n Mejor modelo: ")
print (best_estimator)
```

Figura 5. Código de selección del mejor modelo.

En la Figura 6, se puede observar el resultado. Este informa que el mejor modelo es el RandomForest pues como se observa tiene un menor error que los demás modelos aunque la diferencia no es tan grande.

```
---------
RMSLE prueba:  1.49334 (± 0.08635707 )
RMSLE entrenamiento:  1.45229 (± 0.00110447 )
---------
RMSLE prueba:  1.14304 (± 0.03310625 )
RMSLE entrenamiento:  1.14221 (± 0.02588166 )
---------
RMSLE prueba:  1.12348 (± 0.03285831 )
RMSLE entrenamiento:  1.12479 (± 0.03076983 )
Seleccionado:  3

 Mejor modelo:
RandomForestRegressor(max_depth=5, n_estimators=2)
```

Figura 6. Resultados de rendimiento de los modelos.

Mejora de hiperparametros del modelo seleccionado

La mejora de hiperparámetros sirve para encontrar la configuración de hiperparámetros que produzca el mejor rendimiento para un modelo dado. Los hiperparámetros son parámetros que controlan el proceso de entrenamiento de un modelo de aprendizaje automático. En la Figura 7, se observa el código que permite la mejora de estos hiperparametros junto con los valores. El modelo RandomForest utiliza los hiperparametros son "*n_estimators*" y "*max_depth*". En la Figura 8, se puede visualizar el resultado obtenido con las mejoras de los hiperparametros.

```
# Se busca obtener los mejores parámetros para el modelo elejido

parametros = { 'n_estimators': [5,10,15],
               'max_depth':[5,7,9]}

forest_reg = GridSearchCV(estimator = estimador3,
                          param_grid = parametros,
                          cv = ShuffleSplit(n_splits= 5, test_size=tam_val),
                          scoring = 'neg_mean_squared_error',
                          verbose = 1,
                          return_train_score = True,
                          n_jobs = -1)
forest_reg.fit(Xtv, ytv)
```

```
Fitting 5 folds for each of 9 candidates, totalling 45 fits
```

Figura 7. Código para la mejora de hiperparametros.

```
Mejor modelo Random Forest:  RandomForestRegressor(max_depth=9, n_estimators=15)
Mejores parámetros para el modelo Random Forest:  {'max_depth': 9, 'n_estimators': 15}
```

Figura 8. Resultado de la mejora de los hiperparametros.

El puntaje del modelo RandomForest seleccionado para los datos de entrenamiento fue de 0.60484 y para los datos de testeo fue de 0.62121. De los errores, se puede decir que son pequeños y que pueden hablar de una buena estimación del modelo.

4. Curva de aprendizaje

4.1. Modelo seleccionado

Luego de seleccionar el modelo y de obtener los mejores parámetros del mismo para la disminución del error, se desea conocer el rendimiento del modelo y saber si es realmente un buen modelo para este proyecto, así que se utiliza el concepto de curva de aprendizaje para este fin, la curva correspondiente se visualiza en la Figura 9.

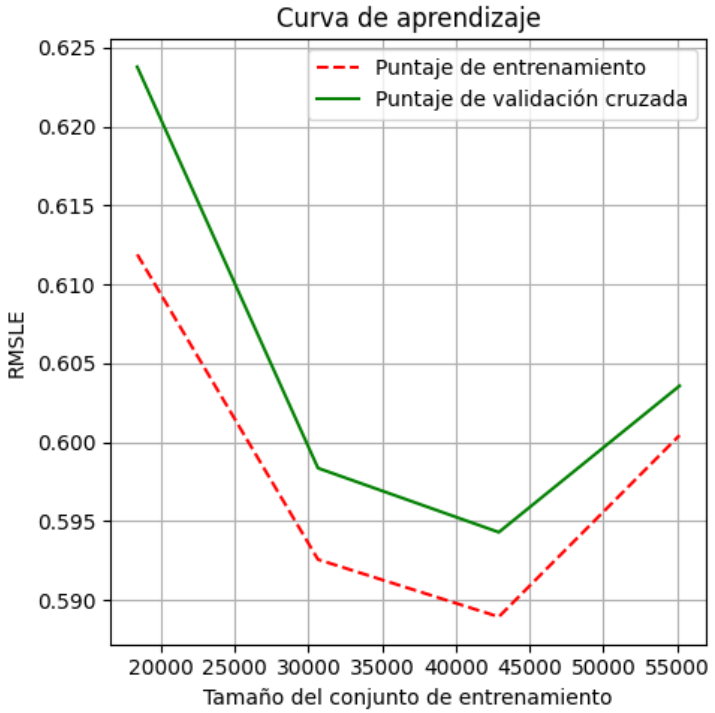

Figura 9. Curva de aprendizaje para el modelo seleccionado

Al observar la Figura 9 es fácil ver que el modelo no es un buen modelo por completo, ya que este presenta un buen comportamiento a medida que se aumentan los datos, pero esto solo ocurre aproximadamente hasta las 43 mil datos de entrenamiento, posterior a eso el modelo empieza a empeorar su rendimiento, lo cual es malo para este modelo y no lo hace recomendable cuando se tienen muchos más datos para el entrenamiento del mismo.

5. Retos y condiciones de despliegue del modelo

- Encontrar un modelo que tenga un rendimiento mucho mejor, que mejore a medida que los datos disponibles crezcan, esto debido a que la emisión de dióxido de carbono es una medición de gran interés hoy en día que debe actualizarse con el tiempo, por lo que la toma de datos regular debería alimentar el modelo y de ser así mejorar las futuras predicciones.
- Este modelo es aplicable para una región específica de Ruanda, por lo que es probable que no sea aplicable para regiones distintas ya que el tratamiento de los datos puede ser muy distinto al que se tiene en este proyecto.
- Los datos usados para este proyecto son de entre 2019 y 2021, no hay datos recientes que puedan denotar la situación actual del país de Ruanda en materia de emisiones de dióxido de carbono.
- Mejora de sensores para la captación de datos pues muchos de los datos eran nulos, lo que puede implicar que los sensores estaban sucios y no midieron bien los componentes presentes.

6. Conclusiones

- El primer modelo realizado tenía un procesamiento de datos esencial que no tomaba mucho detalle en las distribuciones de las variables ni su relación con la variable objetivo, por lo tanto, era un modelo muy sencillo que no tenía una gran precisión a la hora de realizar las predicciones.
- De los métodos utilizados, se eligió el RFR. Sin embargo, al obtener la curva de aprendizaje de este, se observó que el modelo presenta varios problemas con este caso específico, esto debido a que en la curva de aprendizaje se puede observar que el modelo no está aprendiendo lo suficiente con los datos de entrenamiento, debido a que ni los datos entrenamiento ni los datos de validación presentan un rendimiento aceptable, esto puede deberse a que el modelo es demasiado simple, los datos tomados no son representativos en tamaño o están muy mezclados.
- Para solucionar los problemas presentados por este modelo, se podrían utilizar modelos más complejos, como árboles de decisión o redes neuronales. Además, se puede ver que el modelo se comporta de mejor manera aumentando los datos hasta más o menos 42 mil datos, pero de ahí en adelante, el modelo no se comporta correctamente.

Bibliografía

- Predict CO2 emissions in Rwanda | Kaggle. (s. f.). https://www.kaggle.com/competitions/playground-series-s3e20
- Curso de Inteligencia Artificial 2023. https://rramosp.github.io/ai4eng.v1/content/M00_intro_udea.html