ENTREGA FINAL: PROYECTO DE ANALÍTICA DE DATOS
PREDICT CO2 EMISSIONS IN RWANDA.

INTEGRANTES:

LAURA CRISTINA DIAZ OSORIO.

C.C: 1018351214

JUAN FELIPE ESCOBAR RENDÓN.

C.C:1001416321

CURSO:

INTRODUCCION A LA INTELIGENCIA ARTIFICIAL

TUTOR:

RAUL RAMOS POLLAN

UNIVERSIDAD DE ANTIOQUIA
FACULTAD DE INGENIERÍA
AMALFI-ANTIOQUIA
2023

Contenido

PREDICT CO2 EMISSIONS IN RWANDA.

1. Problema Predictivo:

El objetivo de este desafío es utilizar datos de emisiones de CO2 de observaciones satelitales de Sentinel-5P para crear modelos de aprendizaje automático que puedan predecir las emisiones futuras de carbono.
CO2: "El CO2, dióxido de carbono o, simplificando mucho, 'carbono', es el principal gas de efecto invernadero de origen humano. Significa que contribuye al calentamiento global, cuyas consecuencias notamos a diario. Es incoloro y carece de olor. Está presente de forma natural en la atmósfera y, sí, forma parte esencial de nuestro organismo."

1.1. Dataset Utilizado:

El conjunto de datos utilizado en este proyecto proviene de kaggle competitions y se compone de observaciones satelitales de CO2 de Sentinel-5P. para la recolección de la información se seleccionaron aproximadamente 497 ubicaciones únicas de varias áreas en Rwanda, distribuidas alrededor de tierras de cultivo, ciudades y plantas de energía.
El dataset es un archivo en formato zip el cual está compuesto por los siguientes archivos csv:

- traind.csv: En este archivo se encuentra el porcentaje de datos que fueron seleccionados para entrenar el modelo.
- test.csv: En este archivo se encuentra el porcentaje que será usado para observar la precisión del modelo a la hora de predecir los niveles de emisión de CO2.
- sample_submission: En este archivo muestra el formato correcto en el que se deben presentar las predicciones del modelo.

Los datos contienen siete características principales que se extrajeron semanalmente desde enero de 2019 hasta noviembre de 2022. Estas características incluyen:

- Dióxido de Azufre (Sulphur Dioxide)
- Monóxido de Carbono (Carbon Monoxide)
- Dióxido de Nitrógeno (Nitrogen Dioxide)
- Formaldehído (Formaldehyde)
- Índice de Aerosol UV (UV Aerosol Index)
- Ozono (Ozone)

- Nubosidad (Cloud).

Algunas de las variables del dataset son: ubicación geoespacial (latitud y longitud), año y semana, junto con mediciones de emisiones de CO2.El conjunto de datos contiene información histórica de emisiones de CO2 hasta el año 2021:

- latitude y longitude: Representan las coordenadas geográficas de las ubicaciones para las cuales se han recopilado los datos. La latitud representa la posición norte-sur y la longitud la posición este-oeste en la superficie terrestre.
- year: Indica el año al que corresponden los datos.
- week_no: Esta columna indica el número de semana del año al que corresponden los datos.
- SulphurDioxide_SO2_column_number_density: Esta característica representa la densidad de columnas de dióxido de azufre (SO2) en la atmósfera.
- CarbonMonoxide_CO_column_number_density: Densidad de columnas de monóxido de carbono (CO) en la atmósfera. proporciona información sobre la concentración de CO en la atmósfera en una ubicación y momento dados.
- NitrogenDioxide_NO2_column_number_density: densidad de columnas de dióxido de nitrógeno (NO2) en la atmósfera. Refleja la concentración de NO2 en la atmósfera en un lugar y tiempo específicos.
- Formaldehyde_tropospheric_HCHO_column_number_density: Densidad de columnas de formaldehído (HCHO) en la troposfera. El formaldehído es una sustancia química que puede tener implicaciones en la calidad del aire y la salud humana.
- UvAerosolIndex_absorbing_aerosol_index: Índice de aerosol absorbente en el ultravioleta (UV). Los aerosoles pueden afectar la calidad del aire y la visibilidad, y el índice de aerosol absorbente proporciona información sobre las partículas absorbentes en la atmósfera.
- Ozone_O3_column_number_density: la densidad de las columnas de ozono (O3) en la atmósfera. El ozono en la atmósfera desempeña un papel importante en la protección contra la radiación ultravioleta del sol.
- Cloud_cloud_fraction: Fracción de nubes en la atmósfera. Puede indicar el grado de cobertura de nubes en una ubicación y momento específicos
- emission: indica los niveles emitidos de CO2 correspondientes a cada semana en las que se tomaron los datos.

1.2. Métricas de Desempeño:

Root Mean Squared Error (RMSE):

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

Esta métrica mide la raíz cuadrada del error cuadrático medio entre las emisiones de CO2 predichas y los valores reales. Cuanto menor sea el valor de RMSE, mejor será el rendimiento del modelo en términos de precisión de las predicciones, la intención es predecir las emisiones de CO2 para el año 2022 a través de las observaciones satelitales.

1.3. Criterio de Desempeño Deseable:

Un criterio deseable de desempeño es que el modelo de predicción de emisiones de CO2 arroje un RMSE lo más pequeño posible, con la finalidad de que las predicciones sean precisas y esto ayude a los gobiernos con la toma de decisiones respecto al cambio climático y las futuras acciones a tomar para tratar de revertir o tratar que el proceso del cambio climático sea más lento.

2. Exploración de datos.

Iniciamos la visualización y exploración de los datos obtenidos, realizando, histogramas, gráficos de correlación e interacciones entre las variables para tratar de identificar patrones y distribuciones que estos tengan.

2.1. Variable predictora.

Procedemos con la exploración de la variable a predecir, con esto queremos observar, la distribución que sigue la variable 'emission', la presencia de valores atípicos los cuales podrían afectar la precisión en las predicciones futuras.

2.1.1. Observación de datos atípicos en la variable respuesta.

En el histograma se puede observar que la variable a predecir no sigue una distribución normal y si bien la mayoría de los datos se encuentran a la izquierda del histograma, es claro que estos tienen un sesgo bastante amplio a la derecha, indicando que se presentan valores muy grandes lo que puede generar un problema a la hora de observar la precisión en las predicciones; sin embargo, para esta primera versión de limpieza del dataset, se optó por no realizar ninguna transformación y conservar todos los datos atípicos de la variable respuesta.

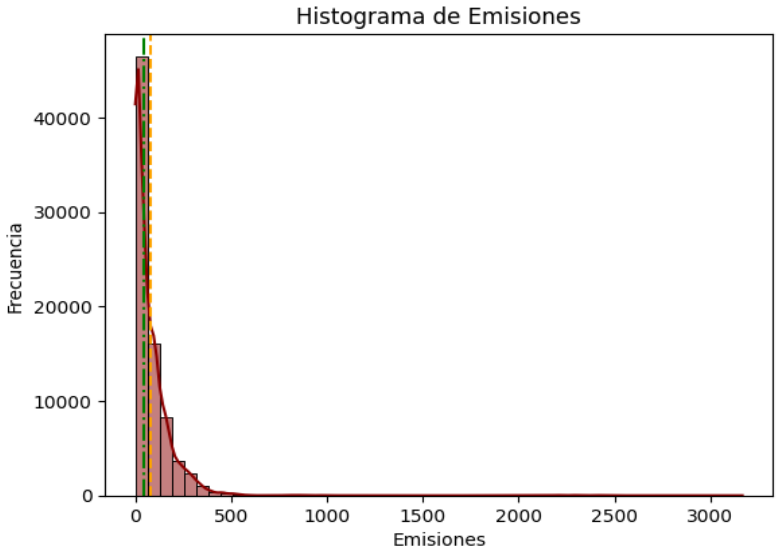

Fig. 1

2.1.2. Gráfica de serie de tiempo.

Se grafica la variable emisión como una serie de tiempo para observar patrones temporales que esta posee en los datos, lo que puede ser beneficioso al momento de escoger y descartar posibles modelos predictivos, para este caso se identifican algunos datos atípicos en la varianza de los datos, lo que nos indicaría la posibilidad de usar algún tipo de transformación o diferenciación para los datos en caso de realizar un modelo de serie de tiempo.

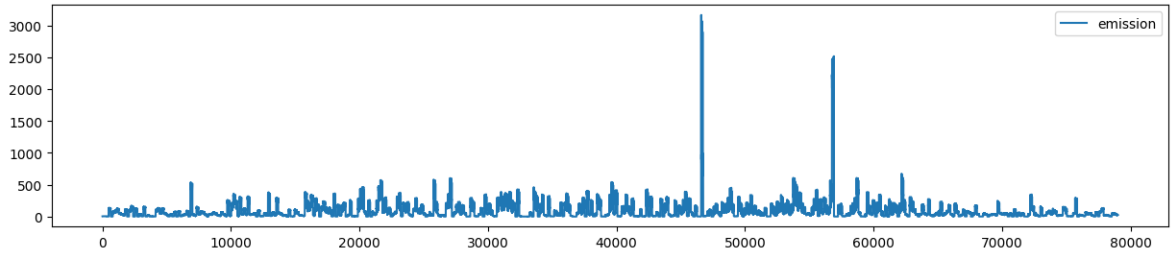

Fig. 2

2.2. Observación general.

Ahora procedemos a visualizar el comportamiento de las otras variables que conforman la base de datos en los que podremos observar correlaciones que las variables tienen entre ellas y con la variable predictora.

	Variable	Correlación
0	emission	1.000000
1	longitude	0.102746
2	UvAerosolLayerHeight_aerosol_height	0.069008
3	Cloud_surface_albedo	0.046587
4	Formaldehyde_tropospheric_HCHO_column_number_d...	0.040263
...	...	...
70	Formaldehyde_tropospheric_HCHO_column_number_d...	-0.033333
71	NitrogenDioxide_solar_azimuth_angle	-0.033417
72	CarbonMonoxide_CO_column_number_density	-0.041328
73	CarbonMonoxide_H2O_column_number_density	-0.043217
74	UvAerosolLayerHeight_aerosol_pressure	-0.068138

75 rows × 2 columns

Fig. 3.

Se puede visualizar que la correlación lineal general que tiene la variable a predecir con las otras es bastante baja por lo que se podría decir que un modelo de regresión lineal no sería óptimo ya que las variables se pueden estar teniendo interacciones más complejas entre ellas.

3. Preprocesado y limpieza de datos

3.1. Manejo de datos faltantes.

Para este punto encontramos que el dataset cumple con el requisito de al menos 5% de datos faltantes en al menos tres columnas, igualmente se observó que en algunas de ellas el porcentaje de datos faltantes llegaba hasta más del 50% por lo que optamos por eliminar las columnas que superan este valor y se procedió con la imputación de datos. El método usado para la imputación de datos fue el de crear una muestra con distribución normal a partir la media y desviación de cada columna de para así evitar sesgos en la estimación de la media y varianza de estos.

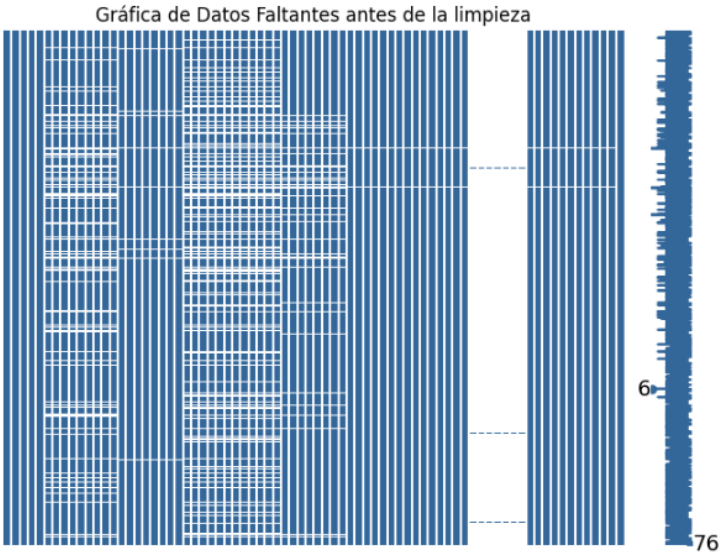

Fig. 4

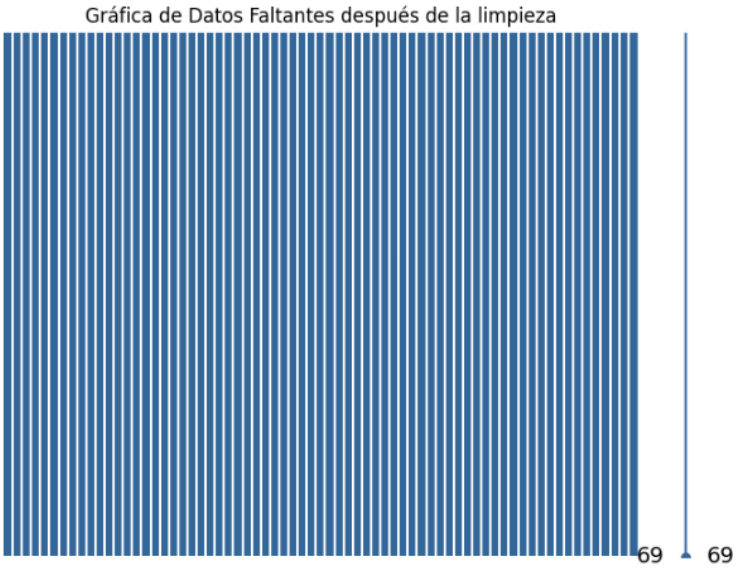

Fig. 5

3.2. Columnas categóricas.

Como el dataset no cumple con el requisito del 10% de columnas categóricas mínimas, por ende, para cumplir con este requisito escogimos las columnas con menor correlación lineal, para proceder a discretizar estas, en intervalos y en cada intervalo categorizar con un número entero.

Comparación gráfica de columnas originales vs categorizadas

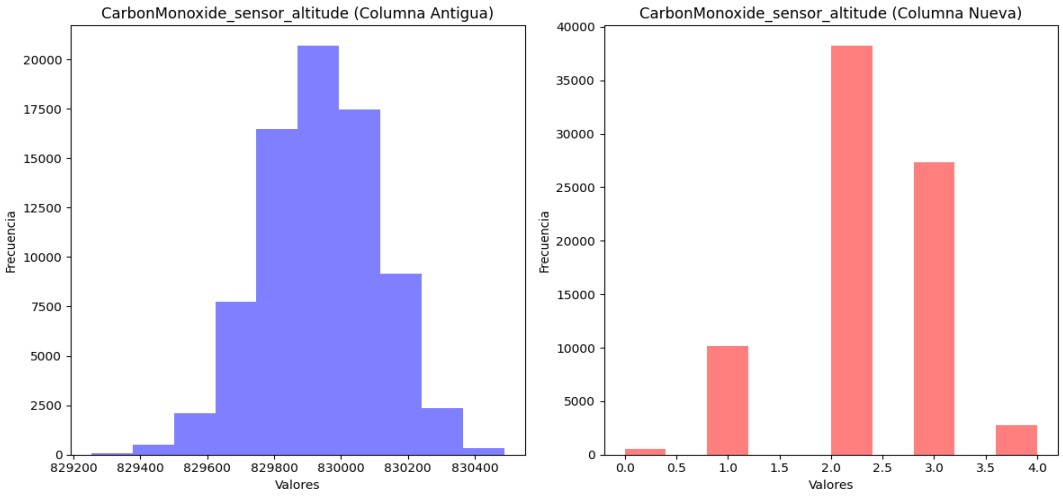

Fig. 6

4. Algoritmos predictivos supervisados.

4.1. División de datos.

Se procede a dividir el dataset en dos conjuntos de datos uno para realizar el entrenamiento del modelo y otro para testear el rendimiento del modelo entrenado. La proporción de datos para cada conjunto que se utilizará es 70% para entrenamiento y 30% para testeo.

4.2. Selección de Algoritmos predictivos.

Teniendo en cuenta que al realizar la visualización de datos y el procesamiento de estos se encontró que las correlaciones lineales son muy bajas aplicar una regresión lineal per0 se decide no descartar este modelo así que se se probaron los siguientes modelos con sus respectivos RMSE:

- Regresión polinómica.

```
El error (RMSE) de test es: 137.89944983901452
```

- Bosques Aleatorios para Regresión (Random Forest Regressor)

```
El error (RMSE) de test es: 23.668992519508265
```

- Gradient Boosting

```
El error (RMSE) de test es: 66.46885942266577
```

- XG Boost

```
RMSE: 28.995346397121075
```

Y se escogieron los dos modelos con menor RMSE que fueron el Random forest regressor y XGboost.

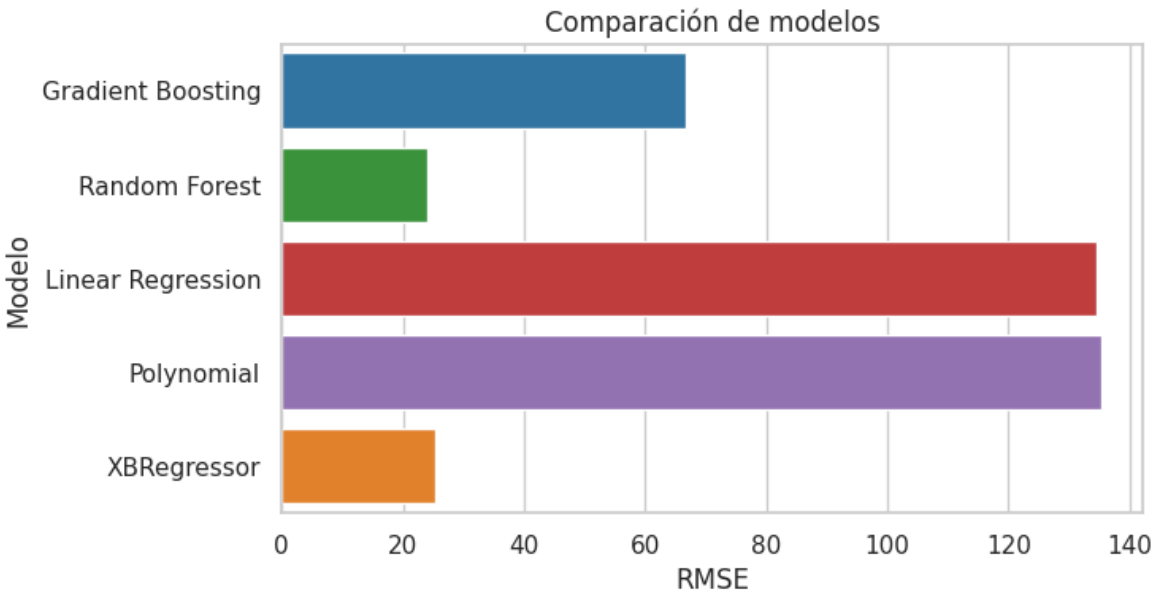

Fig.7 comparación de RMSE de los modelos

4.2.1. Hiper Parámetros.

Se utilizó la búsqueda aleatoria en cuadrícula Grid Search Cv en los dos modelos para hallar los hiperparametros que optimizan el RMSE, obteniendo así los siguientes resultados.

XGboost:

```
Fitting 2 folds for each of 288 candidates, totalling 576 fits
```

```
{'colsample_bytree': 0.7, 'learning_rate': 0.1, 'max_depth': 10,
'min_child_weight': 5, 'n_estimators': 500, 'objective':
'reg:squarederror', 'subsample': 0.7}
```

```
model = XGBRegressor(subsample= 0.7,
                     objective= 'reg:squarederror',
                     n_estimators= 500,
                     min_child_weight= 5,
                     max_depth= 10,
                     learning_rate= 0.1,
                     colsample_bytree= 0.7)
```

```
RMSE: 26.620338086921606
```

Curva de aprendizaje:XGBregressor

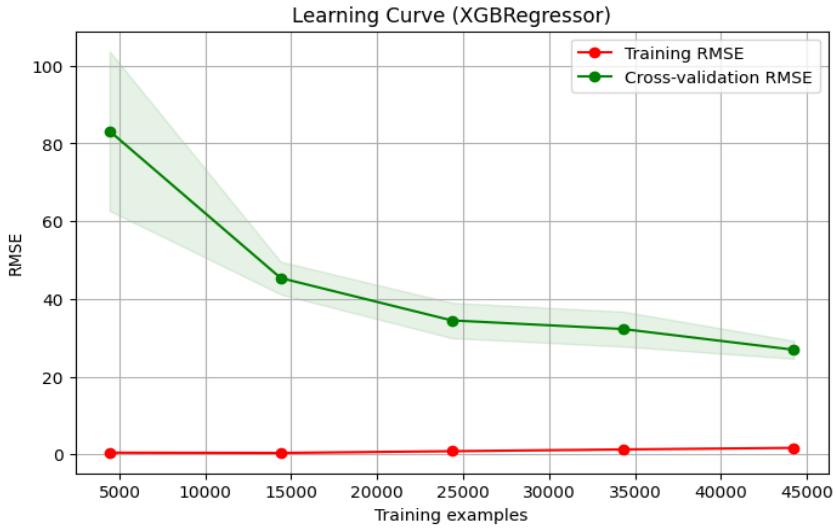

Fig 8. Curva de aprendizaje XGBregressor

Random forest Regressor:

```
Fitting 2 folds for each of 81 candidates, totalling 162 fits
Mejores hiperparámetros encontrados:
{'max_depth': None, 'min_samples_leaf': 2, 'min_samples_split':
2, 'n_estimators': 50}
RandomForestRegressor(min_samples_leaf=2, n_estimators=50,
random_state=42)
```

```
model_rf = RandomForestRegressor(max_depth= None,
                                 min_samples_leaf= 2,
                                 min_samples_split= 2,
                                 n_estimators= 50)
```

```
RMSE: 23.42571799640674
```

Curva de aprendizaje: Random Forest

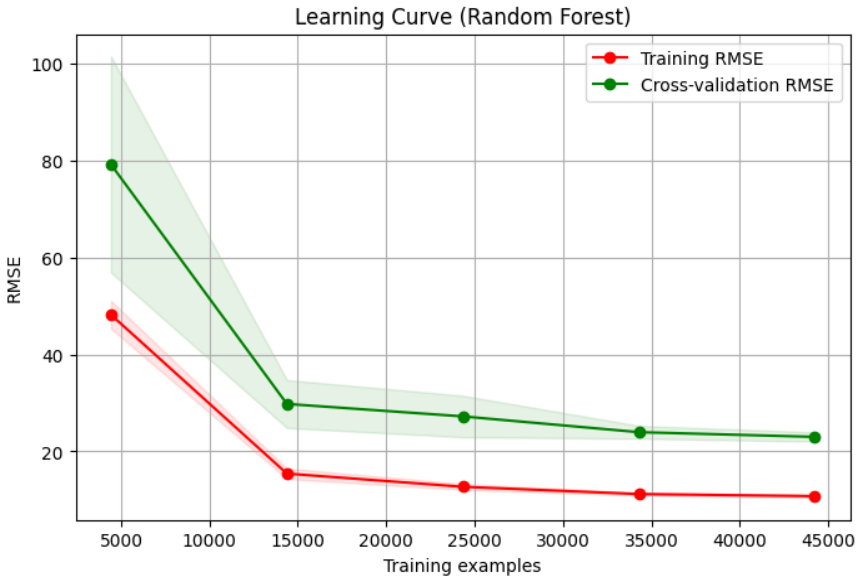

Fig.9 Curva de aprendizaje Random Forest.

En la gráfica podemos observar que el modelo, puede no estar generalizando bien los datos y está presentando un caso de sobreajuste y por esto presenta valores de RMSE más altos al momento de realizar predicciones, igualmente cabe resaltar que el modelo presenta algunas tendencias de bias que van mejorando a medida que se va aumentando la cantidad de datos una buena manera de mejorar el desempeño del modelo sería tomar un poco más de datos a la hora de entrenar el modelo, otra buena opción es tratar de limitar la profundidad máxima del árbol para así controlar la complejidad del modelo o tratar de aumentar la cantidad de árboles para tratar de suavizar las predicciones.

5. Algoritmos no supervisados.

Para este caso se tomó la técnica PCA para escoger los componentes principales y así tratar de reducir la dimensionalidad de los datos ya que al hacer la limpieza no encontramos argumentos suficientes para discernir de elementos que hacen ruido y que pueden no aportar información válida o importante a los modelos

5.1. Hiperparámetros.

XBGboost + PCA

```
Mejores hiperparámetros para PCA encontrados:
{'pca__n_components': 2, 'pca__random_state': 42}
Pipeline(steps=[('scaler', StandardScaler()),
                ('pca', PCA(n_components=2, random_state=42)),
                ('model',
```

```
                  XGBRegressor(base_score=None, booster=None,
callbacks=None,
                               colsample_bylevel=None,
colsample_bynode=None,
                               colsample_bytree=0.7, device=None,
                               early_stopping_rounds=None,
                               enable_categorical=False,
eval_metric=None,
                               feature_types=None, gamma=None,
grow_policy=None,
                               importance_type=None,
                               interaction_constraints=None,
learning_rate=0.1,
                               max_bin=None,
max_cat_threshold=None,
                               max_cat_to_onehot=None,
max_delta_step=None,
                               max_depth=10, max_leaves=None,
min_child_weight=5,
                               missing=nan,
monotone_constraints=None,
                               multi_strategy=None,
n_estimators=500,
                               n_jobs=None,
num_parallel_tree=None,
                               random_state=None, ...))])
```

```
RMSE: 137.25271886199053
```

Curva de aprendizaje: XGBRegressor + PCA

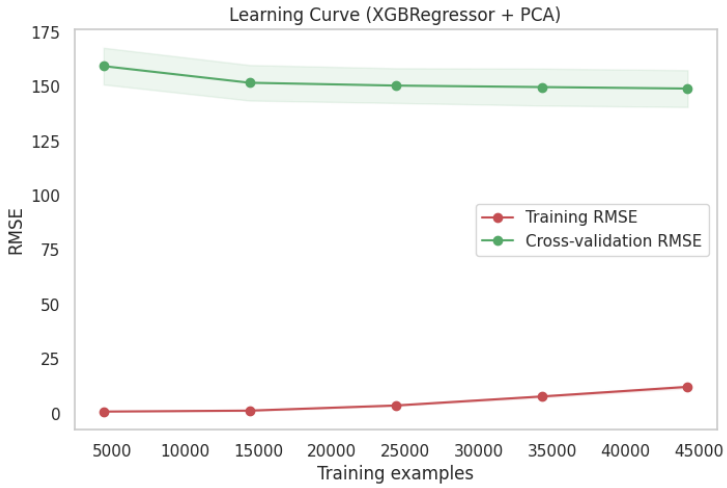

Fig 10. Curva de aprendizaje XGBRegressor+ PCA

Random Forest Regressor + PCA

```
output
Mejores hiperparámetros para PCA encontrados:
{'pca__n_components': 40, 'pca__random_state': 42}
Pipeline(steps=[('scaler', StandardScaler()),
                ('pca', PCA(n_components=40, random_state=42)),
                ('model',
                 RandomForestRegressor(min_samples_leaf=2,
n_estimators=50,
                                       random_state=42))])
```

```
RMSE: 136.74269881673362
```

Curva de aprendizaje: Random Forest + PCA

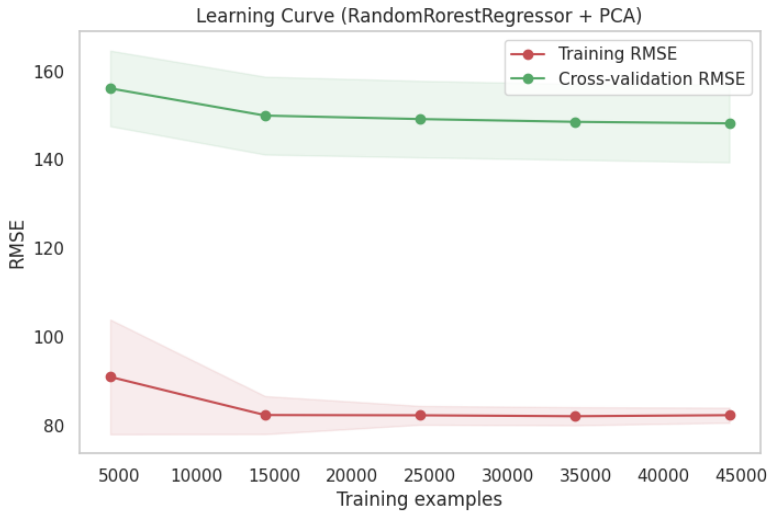

Fig 11. curva de aprendizaje Randomforest + PCA

Con el gridsearch encontramos que el mejor modelo que este arrojaba era el que menos componentes de los datos elimina y al aplicar la combinación de algoritmos al modelo se presentó un problema de aumento en el RMSE. tal vez se debió a que al aplicar el PCA se eliminaron las características importantes para predecir ya que los modelos (Random Forest y XBGboost) son sensibles a la dimensionalidad de las bases de datos, pero se sigue presentando un sobreajuste del modelo esta vez mucho más marcado y tendencias bias con los primeros datos, por lo que esto corrobora que el problema de este esta en la elección de hiperparametros o del modelo debido a su complejidad. Por esto se recomienda tal vez la aplicación de un modelo menos complejo como el lineal o polinomial, pero para este caso se

puede considerar el uso de métodos como Bagging o Boosting que ayudan a reducir el overfitting del modelo o métodos de validación cruzada más robustos.

6. Restos y condiciones para desplegar el modelo.

El reto principal para desplegar estos modelos es la capacidad limitada de características con las que se pudo realizar la búsqueda de hiper parámetros debido que se tornaba muy pesada la búsqueda y en el caso del random forest el tiempo de espera fue de varias horas. La limpieza de datos se complica un poco ya que las correlaciones lineales entre de la variable a predecir con las otras es demasiado baja lo que complica la búsqueda de tal vez elementos innecesarios para los modelos y que pueden distorsionar las predicciones y por ende la elección del mejor modelo.

7. Conclusiones.

- Modelos como Random Forest y XGBoost son sensibles a la dimensionalidad de las bases de datos, y la aplicación de PCA puede eliminar características importantes para la predicción.
- Se propone la exploración de métodos como Bagging o Boosting y métodos de validación cruzada más robustos para mejorar la generalización del modelo.
- La gráfica indica que el modelo puede no estar generalizando bien los datos y muestra signos de sobreajuste, evidenciado por los valores altos de RMSE en las predicciones.
- La capacidad limitada de características utilizadas en la búsqueda de hiperparámetros presenta un desafío, ya que la búsqueda se vuelve pesada.
- La complejidad en la limpieza de datos, debido a bajas correlaciones lineales, complica la identificación de elementos innecesarios que podrían afectar las predicciones y la elección del mejor modelo.

Bibliografía.

- Newtral, CO2: el aire que exhalamos y que está matando el planeta. https://www.newtral.es/que-es-co2-peligros/20190725/
- Kaggle,Playground Series - Season 3, Episode 20, Predict CO2 Emissions in Rwanda. https://www.kaggle.com/competitions/playground-series-s3e20/overview
- Evaluando el error en los modelos de regresión, 2018. https://aprendeia.com/evaluando-el-error-en-los-modelos-de-regresion/
- Aprendizaje no supervisado. https://aprendeia.com/aprendizaje-no-supervisado-machine-learning/