APRENDIZAJE AVANZADO

Luis Fernando Salgado Durango
Maestría en ciencia de los datos y analítica
lfsalgadod@eafit.edu.co
Universidad EAFIT
Andrés Felipe Mejía Flórez
Maestría en ciencia de los datos y analítica
afmejiaf@eafit.edu.co
Universidad EAFIT
Carlos Andrés Jaramillo Pineda
Maestría en ciencia de los datos y analítica
cajaramilp@eafit.edu.co
Universidad EAFIT
Daniela Lopera Pai
Maestría en ciencia de los datos y analítica
ddloperap@eafit.edu.co
Universidad EAFIT

RESUMEN

El siguiente informe busca mostrar el proceso realizado para la implementación de un algoritmo genético que permita la selección se los mejores hiper parámetros de algoritmos de machine learning, puntualmente en este ejercicio, regresión logística. Siendo una parte importante de todo el proceso el entender y comprender las características de los datos. El proceso de iniciación, generación, selección y evaluación de individuos en el algoritmo genético.

INTRODUCCIÓN.

Con la intención de lograr el tuning de un algoritmo seleccionado se busca realizar un análisis completo de los hiperparámetros del modelo, en un análisis previo lo que se pretende explorar toda la información que puedan aportar las variables antes de ser utilizados en el algoritmo, con esto se divide el trabajo varias partes, teniendo una importancia considerable el entendimiento de los datos y la extracción de las características que estos puedan poseer, para una vez ya se tenga un dataset que esté en condiciones óptimas, se pueda realizar así el entrenamiento del modelo y obtener resultados mejores resultados.

Resultados y Análisis.

Para este problema lo que se pretende es más que la implementación del algoritmo de regresión logística que realice la clasificación de un conjunto de datos, se busca aplicar un algoritmo que ayude a encontrar los mejores hiperparámetros del algoritmo de regresión para obtener los mejores resultados.

Es importante resaltar que, si bien los datos son importantes para todo este proceso, se selecciona un dataset que permita resultados buenos y ver como varían estos según los cambios en los hiperparámetros del algoritmo, siendo estos un data set de información bancaria con las columnas que se pueden ver en la ilustración 1, cullas variables se ven explicadas de la siguiente manera.

1 - edad (numérico)
2 - trabajo: tipo de trabajo (categórico: "administrador", "obrero", "empresario", "criada", "administración", "jubilado", "autónomo", "servicios",

"estudiante","técnico","desempleado","desconocido")

3 - civil: estado civil (categórico: "divorciado","casado","soltero","desconocido"; nota: "divorciado" significa divorciado o viudo)

4 - educación (categórico: "básico.4y","básico.6y","básico.9y","bachillerato","analfabetos","curso.profesional","título.universitario", "desconocido")

5- mora: ¿tiene crédito en mora? (categórico: "no","sí","desconocido")

6 - vivienda: ¿tiene préstamo de vivienda? (categórico: "no","sí","desconocido")

7 - préstamo: ¿tiene préstamo personal? (categórico: "no","sí","desconocido")

relacionado con el último contacto de la campaña actual:

8 - contacto: tipo de comunicación del contacto (categórico: "celular","teléfono")

9 - mes: último mes de contacto del año (categóricos: "ene", "feb", "mar", ..., "nov", "dec")

10 - day_of_week: último día de contacto de la semana (categórico: "lunes", "martes", "miércoles", "jueves", "vie")

11 - duración: duración del último contacto, en segundos (numérico). Nota importante: este atributo afecta en gran medida al objetivo de salida (por ejemplo, si la duración es igual a 0, entonces y="no"). Sin embargo, la duración no se conoce antes de que se realice una llamada. Además, después del final de la llamada y es obviamente conocido. Por lo tanto, esta entrada solo debe incluirse con fines de referencia y debe descartarse si la intención es tener un modelo predictivo realista.

otros atributos:

12 - campaña: número de contactos realizados durante esta campaña y para este cliente (numérico, incluye último contacto)

13 - pdays: número de días que transcurrieron desde la última vez que se contactó al cliente de una campaña anterior (numérico; 999 significa que el cliente no fue contactado previamente)

14 - anterior: número de contactos realizados antes de esta campaña y para este cliente (numérico)

15 - poutcome: resultado de la campaña de marketing anterior (categórico: "fracaso","inexistente","éxito")

atributos del contexto social y económico

16 - emp.var.rate: tasa de variación del empleo - indicador trimestral (numérico)

17 - cons.price.idx: índice de precios al consumidor - indicador mensual (numérico)

18 - cons.conf.idx: índice de confianza del consumidor - indicador mensual (numérico)

19 - euribor3m: tasa euribor 3 meses - indicador diario (numérico)

20 - nr.employed: número de empleados - indicador trimestral (numérico)

Variable de salida (objetivo deseado):

21 - y - ¿El cliente ha suscrito un depósito a plazo? (binario: "sí","no")

```
Data columns (total 20 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   age             4119 non-null   int64
 1   job             4119 non-null   object
 2   marital         4119 non-null   object
 3   education       4119 non-null   object
 4   default         4119 non-null   object
 5   housing         4119 non-null   object
 6   loan            4119 non-null   object
 7   contact         4119 non-null   object
 8   month           4119 non-null   object
 9   day_of_week     4119 non-null   object
 10  duration        4119 non-null   int64
 11  campaign        4119 non-null   int64
 12  pdays           4119 non-null   int64
 13  previous        4119 non-null   int64
 14  poutcome        4119 non-null   object
 15  emp.var.rate    4119 non-null   float64
 16  cons.price.idx  4119 non-null   float64
 17  cons.conf.idx   4119 non-null   float64
 18  euribor3m       4119 non-null   float64
 19  nr.employed     4119 non-null   float64
dtypes: float64(5), int64(5), object(10)
```

Ilustración 1 variables del dataset

Siendo la variable objetivo la 21 “y”. es de importancia resaltar que el dataset presenta un mix entre variables descriptivas y variables numéricas, lo cual implica que se debe realizar un ajuste, en caso de que los posibles valores que pueda tomarla variable características sean ordinales, se podría enumerar y conservar ese orden o de no serlo se puede optar por realizar una Binarización de los datos convirtiendo así cada descripción en una variable binaria.

	count	mean	std	min	25%	50%	75%	max
age	4119.0	40.113620	10.313362	18.000	32.000	38.000	47.000	88.000
duration	4119.0	256.788055	254.703736	0.000	103.000	181.000	317.000	3643.000
campaign	4119.0	2.537266	2.568159	1.000	1.000	2.000	3.000	35.000
pdays	4119.0	960.422190	191.922786	0.000	999.000	999.000	999.000	999.000
previous	4119.0	0.190337	0.541788	0.000	0.000	0.000	0.000	6.000
emp.var.rate	4119.0	0.084972	1.563114	-3.400	-1.800	1.100	1.400	1.400
cons.price.idx	4119.0	93.579704	0.579349	92.201	93.075	93.749	93.994	94.767
cons.conf.idx	4119.0	-40.499102	4.594578	-50.800	-42.700	-41.800	-36.400	-26.900
euribor3m	4119.0	3.621356	1.733591	0.635	1.334	4.857	4.961	5.045
nr.employed	4119.0	5166.481695	73.667904	4963.600	5099.100	5191.000	5228.100	5228.100

Ilustración 2 descripciones variables numéricas

Si bien el número de variables descriptivas que se tienen es alto, no se debe pasar por alto las variables numéricas, ya que dependiendo de las magnitudes de esta se deberán realizar o no otras actividades como puede ser la normalización en caso de tener magnitudes diferentes entre sí, para este dataset se encuentra que las variables con mayor magnitud esta alrededor de los 5000 como se puede apreciar en la ilustración 2, lo cual es más grande que el resto, pero no es significativamente mayor. Esto no implica que se deba dejar de realizar una normalización de los datos.

	job	marital	education	default	housing	loan	contact	month	day_of_week	poutcome
count	4119	4119	4119	4119	4119	4119	4119	4119	4119	4119
unique	12	4	8	3	3	3	2	10	5	3
top	admin.	married	university.degree	no	yes	no	cellular	may	thu	nonexistent
freq	1012	2509	1264	3315	2175	3349	2652	1378	860	3523

Ilustración 3 descripción análisis descriptivas

Un análisis parecido se realiza para las variables descriptivas, viendo en la ilustración 3 cuantas variaciones tiene cada una y la mayor frecuencia que tienen estas, es importante conocer esto dado que las variables descriptivas no siguen un orden en sus valores, así que la cantidad de valores que puedan tomar todas estas variables implicara que tantas columnas se le agregaran a la data set lo que aumentara considerablemente el tamaño de este, al tiempo que aumenta la complejidad del problema.

Toda la exploración anterior es necesaria pero aún falta conocer si el dataset tiene variables con variables nulos y si cuenta con registros atípicos. El problema de los nulos no presenta gran inconveniente dado que no se presenta ninguno en las variables, pero los datos atípicos si están presentes al momento de ver los diagramas de cajas y bigotes, como se aprecia en la ilustración 4, pero esto en si mismo no aporta más que la información individual de las variables, detectar un valor atípico de todo el registro es un problema que no se abordara en este informe y por lo tanto se trabajara con todos los registros, implicando esto que se podrían afectar los resultados del modelo.

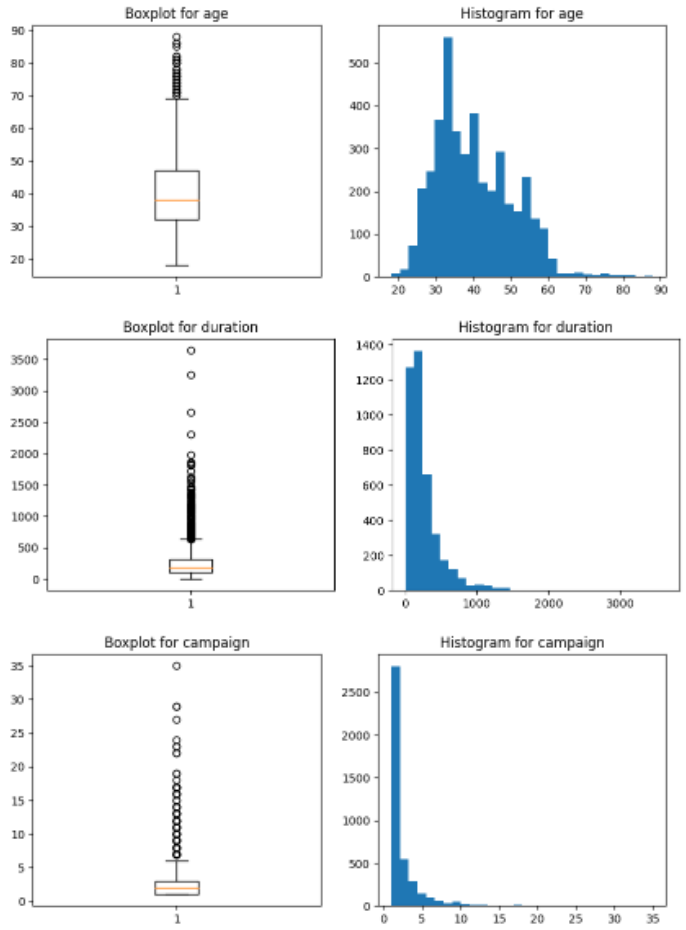

Ilustración 4 histogramas y cajas de bigotes

Ahora es de utilidad revisar la correlación entre las variables dado que se pretende evitar la dependencia entre variables, evitando así la redundancia de la información, como se ve en la ilustración 5, hay variables muy dependientes entre sí, por lo que se podría optar por quitar una de ellas y solo conservar una, pero esto sería muy apresurado dado que todavía no se realiza la Binarización de las variables categóricas, por lo que revisar esta matriz de correlaciones una vez realizado este paso también es necesario para el proceso.

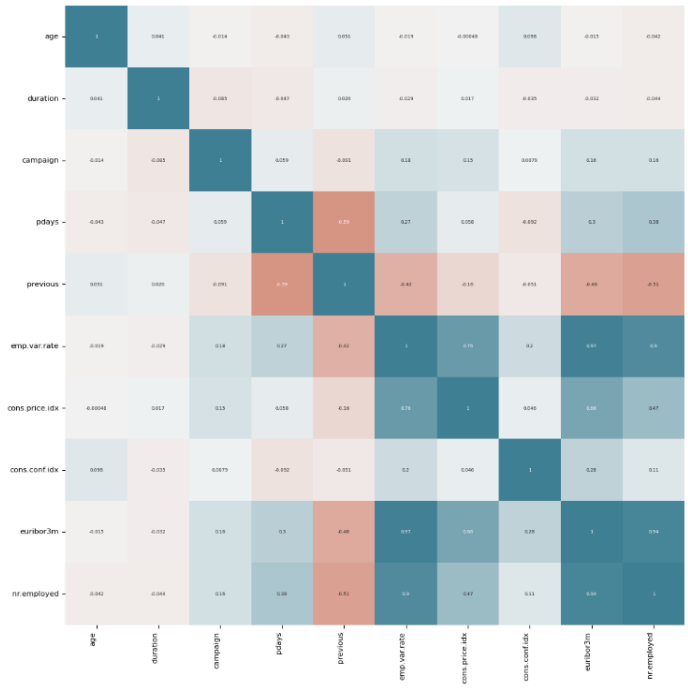

Ilustración 5 matriz de correlación

Antes de realizar este paso se busca relaciones que puedan ser visibles entre las variables categóricas y las variables numéricas como se muestra en la ilustración 6. Pero no se encuentra algo que sea digno de mencionar ya que todos los valores son bastantes parecidos impidiendo así el poder obtener más información en este proceso.

education	age	duration	campaign	pdays	previous	emp.var.rate
Basic	42.337124	253.898457	2.429732	978.815597	0.149472	0.237368
High School	38.097720	258.534202	2.630836	958.022801	0.206298	-0.002497
Illiterate	42.000000	146.000000	4.000000	999.000000	0.000000	-2.900000
Professional Course	40.207477	278.816822	2.512150	958.211215	0.194393	0.163925
University Degree	39.017405	247.707278	2.583070	947.900316	0.207278	-0.009731
Unknown	42.826347	267.281437	2.538922	939.700599	0.263473	-0.074251

Ilustración 6 relación entre variables

Ahora si se procede con la Binarización de las variables categóricas y para no afectar los resultados con las dimensiones, realizar una normalización de los datos es fundamental, si bien no es estrictamente necesario en la regresión logística, permite la convergencia más rápida del modelo. Por último, en la preparación de los datos se da una última ojeada a la matriz de correlación es ya con todo el data set extendido y como se puede observar en la ilustración 7, esas dependencias tan altas entre las variables ya no están presente por lo que es de utilidad usar todo el data set completo.

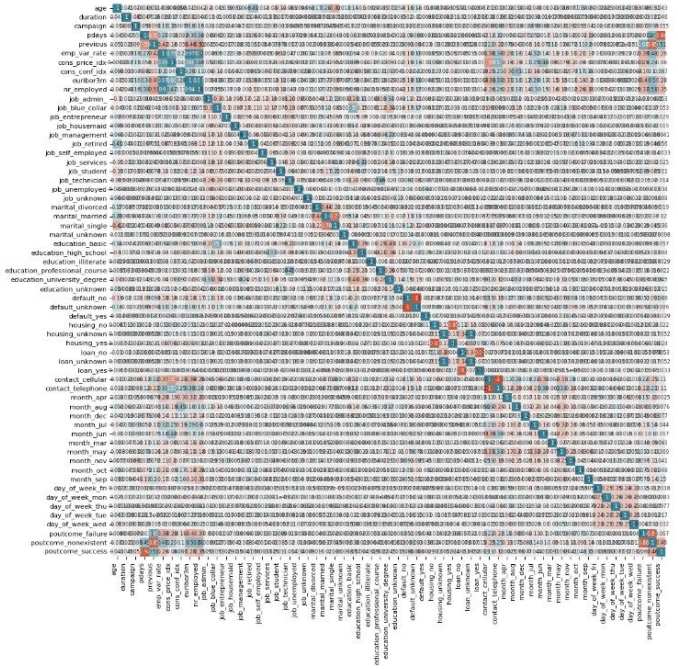

Ilustración 7matriz de correlación

Una vez realizados los procesos previos de ingeniería de características y transformación de variables, se procede a iniciar con la etapa de modelación, inicialmente se realiza una división de los datos con el objetivo de particionar la base en dos conjuntos de train-test y validación, con un porcentaje de 70/30. La metodología para implementar es realizar una validación cruzada que permita encontrar un error preliminar de prueba que se pueda comparar con el error de validación y tener una idea acerca del error de forma generalizada sobre el conjunto completo de los datos.

```
X_train, X_test, y_train, y_test = train_test_split(df_scaled, df_y,
                                                    train_size=0.7,
                                                    random_state=seed)
```

```
X = X_train.values
y = np.ravel(y_train.values)
```

Ilustración 8. Definición de conjunto de train-test y validación

Conociendo la naturaleza del problema, como una clasificación binaria, se decide implementar una regresión logística para determinar la probabilidad de una observación pertenecer a una clase u otra, considerando un umbral de decisión-por defecto del 0.5.

```
df_y.value_counts()
```

```
y
0    3668
1     451
dtype: int64
```

Ilustración 9. Variable objetivo: cantidad por clase

Ahora, la problemática que se presenta al momento de entrenar este modelo es ¿Cómo encontrar el conjunto de hiperparámetros que optimiza los resultados?, si bien existen muchas formas de encontrar la mejor combinación de hiperparámetros (desde "fuerza bruta" a optimizaciones bayesianas) en este ejercicio se presenta la implementación de una búsqueda a través de una malla de hiperparámetros que varía o "muta" a partir de sus mejores individuos (conjuntos de hiperparámetros) teniendo como métrica de referencia, en este caso, la precisión (la cual definimos al observar el comportamiento que tiene la variable objetivo de un desbalanceo de clases en una proporción del 90/10 %). A continuación se procede a explicar el funcionamiento del concepto del algoritmo evolutivo/genético sobre la tarea de aprendizaje automático implementada en este problema.

```
lrmodel_tuning = LogisticRegression(random_state=seed)

param_grid = {'tol'      : Continuous(0.00005, 0.0005),
              'solver'   : Categorical(['lbfgs', 'liblinear', 'saga']),
              'max_iter' : Integer(50, 200)}

cv = KFold(n_splits=5, shuffle=True)
```

Ilustración 10. Malla de parámetros

Iniciamos preliminarmente con la definición del modelo a implementar y un espacio de valores en los cuales nuestros parámetros se moverán. Particularmente la regresión logística entre varios parámetros ajustables cuenta con los parámetros de "tol", "solver" y "max_iter", los cuales nos permiten tener una tolerancia como criterio de parada, varias opciones de algoritmos de resolución de problemas y el número máximo de iteraciones en las que el algoritmo realiza una ejecución. Este espacio de parámetros tiene sus propias particularidades y deben ser definidas a través de su naturaleza, es decir, continuas, discretas o categóricas. Adicionalmente tenemos la definición del número de folds a utilizar (5) para realizar la validación cruzada (la cual nos permite tener un poco de independencia y tranquilidad con respecto a la dependencia del modelo respecto al conjunto de train-test y validación).

```
evolved_estimator = GASearchCV(estimator=lrmodel_tuning,
                               cv=cv,
                               scoring='precision',
                               population_size=20,
                               generations=50,
                               tournament_size=5,
                               elitism=True,
                               mutation_probability=0.70,
                               crossover_probability=0.15,
                               param_grid=param_grid,
                               n_jobs=-1)

evolved_estimator.fit(X, y)
```

Ilustración 11. Parametrización tuning del modelo

Posteriormente a través de la función GASearchCV (la cual realiza una búsqueda de hiperparámetros por medio de evolución genética y validación cruzada) se establecen las condiciones iniciales y probabilísticas que tendrá el espacio de optimización. Entre los valores que se establecen en la función son relevantes y asociados al concepto de evolución genética aquellos de:

Population_size: establece la población inicial con la que empieza la búsqueda de hiperparámetros, siendo un individuo de dicha población una combinación particular de los hiperparámetros (tol, solver y max_iter) del algoritmo.

Generations: Son el numero de generaciones o iteraciones realizadas en la búsqueda de hiperparámetros, también se puede interpretar como la cantidad de veces que se reproducen o mutan los individuos de la población inicial.

Elitism: es una variable booleana que habilita la selección de los mejores individuos con los mejores genes (Aquellos que obtienen el mayor valor de la métrica-precisión en los entrenamientos).

Tournament: Define la cantidad de individuos seleccionados por cada iteración o generación, se

establece solo cuando elitism es verdadero, de lo contrario se tiene una selección aleatoria de los individuos en cada generación.

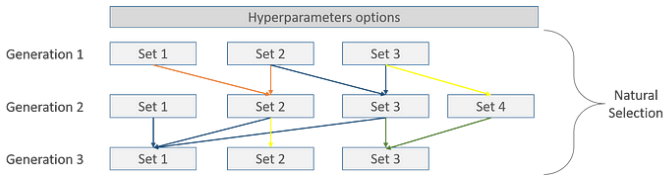

Ilustración 12. Proceso de mutación, cruce y selección de individuos (set's). Imagen tomada de: https://towardsdatascience.com/tune-your-scikit-learn-model-using-evolutionary-algorithms-30538248ac16

Mutation_probability: es la probabilidad que tiene un individuo de una generación de sufrir una mutación en sus genes.

Crossover_probability: es la probabilidad que tiene un individuo de cruzarse con otro individuo de su generación.

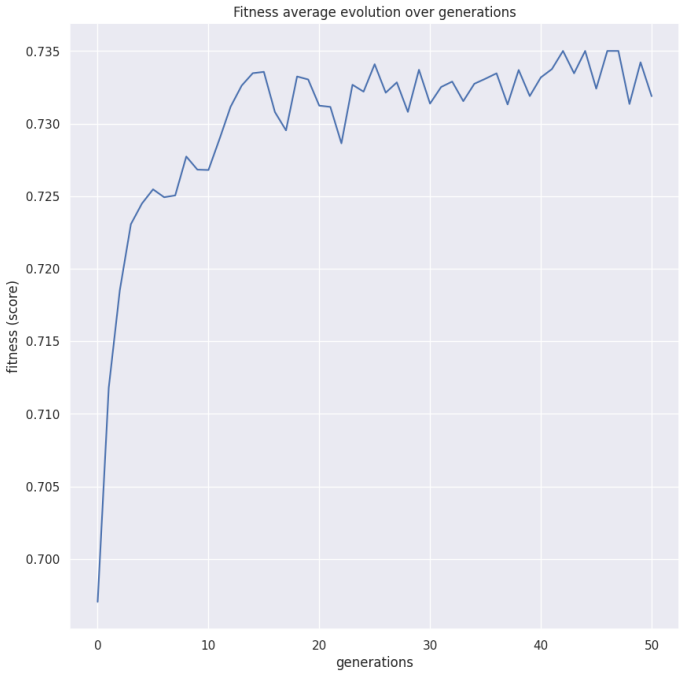

Ilustración 13. Resultados de precisión por generación en el conjunto de entrenamiento

Es importante aclarar que la regresión logística se va entrenando con cada individuo (set de hiperparámetros), calculando sus métricas (siendo en este caso la precisión la métrica objetivo de mejora) y evaluándose cada vez en la siguiente generación con los nuevos individuos, finalmente retornando el algoritmo de búsqueda, el mejor set de hiperparámetros para el algoritmo (aquel que consiguió la mayor métrica) en el conjunto de entrenamiento.

```
evolved_estimator.best_params_
```

```
{'tol': 0.0002292196008409039, 'solver': 'saga', 'max_iter': 184}
```

Ilustración 14. Resultado de mejor set de hiperparámetros

Finalmente se entrena el modelo con la base completa de train con el mejor set de hiperparámetros y se evalúa sobre el conjunto de validación. La matriz de confusión para las clases en el conjunto de validación se muestra a continuación:

	precision	recall	f1-score	support
0	0.94	0.96	0.95	1105
1	0.56	0.44	0.50	131
accuracy			0.90	1236
macro avg	0.75	0.70	0.72	1236
weighted avg	0.90	0.90	0.90	1236

Ilustración 15. Matriz de confusión en conjunto de validación

Como se pueden observar en las ilustraciones 12 y 14, a medida que van pasando las generaciones se va obteniendo una convergencia al posible valor máximo de precisión para el modelo.

Conclusiones.

los algoritmos genéticos son una técnica eficaz para encontrar los mejores hiperparámetros para un modelo de aprendizaje automático. Los hiperparámetros son importantes para el rendimiento del modelo y pueden tener un impacto significativo en la calidad de los resultados. Los algoritmos genéticos imitan la selección natural en la evolución biológica para generar y evaluar soluciones candidatas, seleccionando las mejores para reproducirse y crear una nueva generación de soluciones. Este proceso se repite hasta que se encuentra una solución óptima. Los algoritmos genéticos son una herramienta valiosa para manejar problemas de optimización complejos y explorar una gran cantidad de soluciones candidatas de manera eficiente.

Si a esto le sumamos la ingeniería de características la cual es esencial para mejorar el rendimiento de los modelos de aprendizaje automático, ya que permite seleccionar y transformar las características de entrada para mejorar la precisión y reducir el sobreajuste. También puede mejorar la eficiencia y la interpretabilidad del modelo, y adaptarlo a diferentes conjuntos de datos y problemas de predicción. Por lo que, la ingeniería de características y los algoritmos genéticos pueden ayudar a mejorar considerablemente los resultados que se puedan obtener.

Referencias.

[1] https://statics.teams.cdn.office.net/evergreen-assets/safelinks/1/atp-safelinks.html

[2] https://towardsdatascience.com/tune-your-scikit-learn-model-using-evolutionary-algorithms-30538248ac16