

Análisis exploratorio y aplicación de algoritmos de ML en el sector retail usando conjuntos de datos de transacciones en línea

Seminario
Entregable 2

Especialización en Analítica y Ciencia de Datos
Departamento de Ingeniería de Sistemas
Universidad de Antioquia, Colombia

Mario E. Otero, Lina M. Beltrán

Repositorio: [GitHub](#)

Resumen - En el sector retail, el análisis de datos de transacciones en línea se ha convertido en una fuente crucial de información para las empresas en su búsqueda por comprender el comportamiento de los clientes y mejorar su toma de decisiones. Este ejercicio se centra en el análisis exploratorio y la aplicación de algoritmos de Machine Learning (ML) sobre un conjunto de datos de transacciones en línea. En primer lugar, se lleva a cabo un análisis exploratorio de los datos, que incluye la limpieza y preparación de los conjuntos de datos, a continuación, se aplican diferentes algoritmos de ML de tipo regresión y predictivos. Se emplean algoritmos de aprendizaje supervisado, como las máquinas de vectores de soporte (SVM) y los árboles de decisión, para predecir el monto total por transacción y así buscar identificar patrones de comportamiento y compra en el sector objeto de estudio.¹

Índice de Términos - Algoritmos, Análisis de datos, Análisis de regresión, Aprendizaje automático, Ciencia de datos, Modelos controlados por datos, Retail.

I. INTRODUCCIÓN

El presente ejercicio se enfoca en la aplicación de distintas técnicas de regresión, incluyendo regresión lineal simple, regresión lineal múltiple y el uso de técnicas de regularización como Ridge y Lasso. La transformación de datos también será considerada como parte integral del proceso de análisis, con el objetivo de mejorar la calidad de las predicciones [1].

En primer lugar, se explorará la regresión lineal simple, luego la múltiple, seguido de la regresión lineal múltiple con regularización Ridge y Lasso, además de clasificación y regresión con técnicas ML, tales como: Árboles de decisión, Random Forest y SVM, se evalúa además el rendimiento y precisión de Los modelos con una serie de métricas de desempeño como son: RMSE, R^2 , MSE y MAE [2].

II. DESARROLLO DE CONTENIDO

A. Etapa de limpieza

Inicialmente, se carga e identifica el estado natural del conjunto de datos para comprender rápidamente la estructura y la calidad de los mismos, e iniciar con la definición de los pasos a seguir para la estrategia de limpieza y transformación, a continuación una muestra de la información de este.

```
1 df.info()

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1067371 entries, 0 to 1067370
Data columns (total 8 columns):
#   Column          Non-Null Count  Dtype  
---  -
0   Invoice          1067371 non-null object  
1   StockCode       1067371 non-null object  
2   Description     1062989 non-null object  
3   Quantity       1067371 non-null int64   
4   InvoiceDate     1067371 non-null object  
5   Price          1067371 non-null float64  
6   Customer ID    824364 non-null float64   
7   Country        1067371 non-null object  
dtypes: float64(2), int64(1), object(5)
memory usage: 65.1+ MB
```

Fig. 1. Método info() de la biblioteca Pandas de Python que describe información importante sobre el DataFrame en su estado natural.

Esta etapa incluyó las siguientes acciones:

- Borrado de características irrelevantes.
- Definición de una lista de columnas categóricas y otra de columnas numéricas.
- Limpieza de caracteres no numéricos de la variable "Invoice".
- Detección y borrado de datos atípicos.
- Generación de nuevas variables.
- Creación de Variables dummies.

¹ Consultar [entregable 1](#) para ampliar contexto.

La ejecución de estas acciones derivó en la generación del nuevo conjunto de datos acondicionado para su posterior análisis con técnicas de ML.

```
1 df.info()

<class 'pandas.core.frame.DataFrame'>
Int64Index: 1030626 entries, 0 to 1031376
Data columns (total 21 columns):
#   Column                Non-Null Count  Dtype
---  -
0   Invoice                1030626 non-null object
1   Quantity              1030626 non-null int64
2   Price                 1030626 non-null float64
3   Year                  1030626 non-null int64
4   Months                1030626 non-null int64
5   Month_day             1030626 non-null int64
6   Time_hour             1030626 non-null int64
7   wk_day                1030626 non-null int64
8   year_month            1030626 non-null object
9   TotalSpent            1030626 non-null float64
10  TotalTransaction      1030626 non-null float64
11  TotalQuantity         1030626 non-null int64
12  TotalProductosUnicos  1030626 non-null int64
13  CodigoUnico           1030626 non-null int64
14  CountryUnico          1030626 non-null int64
15  Continent Asia        1030626 non-null uint8
16  Continent Europe      1030626 non-null uint8
17  Continent North America 1030626 non-null uint8
18  Continent North America 1030626 non-null uint8
19  Continent Oceania     1030626 non-null uint8
20  Continent South America 1030626 non-null uint8
dtypes: float64(3), int64(10), object(2), uint8(6)
memory usage: 131.7+ MB
```

Fig. 2. Método info() de la biblioteca Pandas de Python que describe información importante sobre el DataFrame después de la limpieza y transformación.

B. Etapa de aplicación de algoritmos de ML (Regresión lineal)

En primer lugar, se explorará la **regresión lineal simple**, que es una técnica ampliamente utilizada para establecer relaciones lineales entre una variable dependiente y una variable independiente. Esta técnica permitirá analizar la relación directa entre los montos transaccionados por ticket y una única variable predictora, lo cual proporcionará un punto de partida para el análisis comparativo más exhaustivo.

Para este modelo se definieron las variables de entrada o predictoras y una variable target o de salida, de la siguiente manera:

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1030626 entries, 0 to 1031376
Data columns (total 20 columns):
#   Column                Non-Null Count  Dtype
---  -
0   Invoice                1030626 non-null object
1   Quantity              1030626 non-null int64
2   Price                 1030626 non-null float64
3   Year                  1030626 non-null int64
4   Months                1030626 non-null int64
5   Month_day             1030626 non-null int64
6   Time_hour             1030626 non-null int64
7   wk_day                1030626 non-null int64
8   year_month            1030626 non-null object
9   TotalSpent            1030626 non-null float64
10  TotalQuantity         1030626 non-null int64
11  TotalProductosUnicos  1030626 non-null int64
12  CodigoUnico           1030626 non-null int64
13  CountryUnico          1030626 non-null int64
14  Continent Asia        1030626 non-null uint8
15  Continent Europe      1030626 non-null uint8
16  Continent North America 1030626 non-null uint8
17  Continent North America 1030626 non-null uint8
18  Continent Oceania     1030626 non-null uint8
19  Continent South America 1030626 non-null uint8
dtypes: float64(2), int64(10), object(2), uint8(6)
memory usage: 123.8+ MB
```

Fig. 3. Método info() de la biblioteca Pandas de Python que describe información importante sobre el DataFrame con las variables predictoras o de entrada.

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 1030626 entries, 0 to 1031376
Data columns (total 1 columns):
#   Column                Non-Null Count  Dtype
---  -
0   TotalTransaction      1030626 non-null float64
dtypes: float64(1)
memory usage: 15.7 MB
```

Fig. 4. Método info() de la biblioteca Pandas de Python que describe información fundamental sobre el DataFrame con la variable target o de salida.

Adicionalmente, se crearon los conjuntos de datos de entrenamiento y prueba en una proporción de 70/30 respectivamente. Finalmente, luego de entrenado el modelo se obtuvo las siguientes métricas:

	Variable	RMSETrain	RMSETest	R2Train	R2Test	MSETrain	MSETest	MAETrain	MAETest
0	Invoice	-2107.370208	-2098.425127	0.005834	0.005841	-4.441607e+06	-4.404353e+06	-1189.517071	-1185.862949
1	Quantity	-2113.026162	-2104.051523	0.000490	0.000504	-4.465480e+06	-4.428005e+06	-1191.338150	-1187.725597
2	Price	-2112.358472	-2102.549003	0.001122	0.001919	-4.462660e+06	-4.421638e+06	-1191.069686	-1187.046354
3	Year	-2112.259421	-2103.243128	0.001215	0.001273	-4.462237e+06	-4.424608e+06	-1194.372826	-1190.701185
4	Months	-2070.677179	-2061.069696	0.040143	0.040909	-4.288242e+06	-4.248893e+06	-1205.027910	-1201.311327
5	Month_day	-2105.771293	-2096.783902	0.007338	0.007397	-4.434854e+06	-4.397469e+06	-1191.329686	-1187.704902
6	Time_hour	-2105.688270	-2096.676705	0.007422	0.007501	-4.434527e+06	-4.397033e+06	-1177.744641	-1174.014785
7	wk_day	-2095.407611	-2087.059389	0.017087	0.016570	-4.391317e+06	-4.356721e+06	-1168.338075	-1165.252222
8	year_month	-2113.008687	-2104.020999	0.000506	0.000534	-4.465405e+06	-4.427882e+06	-1193.609243	-1189.953471
9	TotalSpent	-2110.514542	-2100.320873	0.002864	0.004029	-4.454869e+06	-4.412255e+06	-1190.059944	-1185.924266
10	TotalQuantity	-1037.941576	-1038.518483	0.758781	0.756438	-1.077405e+06	-1.078717e+06	-560.753191	-560.502114
11	TotalProductosUnicos	-1249.370700	-1245.590431	0.650580	0.649728	-1.561334e+06	-1.552188e+06	-584.755675	-582.069449
12	CodigoUnico	-2113.437573	-2104.508628	0.000101	0.000071	-4.467220e+06	-4.429936e+06	-1192.438213	-1188.750325
13	CountryUnico	-2113.497594	-2104.502289	0.000044	0.000078	-4.467474e+06	-4.429914e+06	-1191.766851	-1187.980325
14	Continent Asia	-2113.509084	-2104.532667	0.000033	0.000048	-4.467522e+06	-4.430036e+06	-1191.900045	-1188.133716
15	Continent Europe	-2113.195374	-2104.184908	0.000330	0.000378	-4.466194e+06	-4.428572e+06	-1191.733701	-1188.000680
16	Continent North America	-2113.552622	-2104.598559	-0.000008	-0.000015	-4.467706e+06	-4.430313e+06	-1192.340838	-1188.664147
17	Continent North America	-2113.566140	-2104.622107	-0.000021	-0.000037	-4.467763e+06	-4.430412e+06	-1192.249951	-1188.588730
18	Continent Oceania	-2112.571955	-2103.620179	0.000918	0.000914	-4.463554e+06	-4.426192e+06	-1191.873106	-1188.214142

Fig. 5. Tabla de resultados para las métricas aplicadas al modelo de regresión lineal simple.

Encontrando que para las variables “TotalQuantity” y “TotalProductosUnicos” la métrica R^2 presentó los resultados más altos, un valor alto de R^2 indica que el modelo es capaz de explicar una gran parte de la variabilidad de los datos, lo que sugiere que se ajusta bien a los datos. Sin embargo, es valioso tener en cuenta que un alto valor de R^2 no garantiza que el modelo sea válido o que las predicciones sean precisas en todos los casos.

El siguiente experimento aplicado al conjunto de datos fue el algoritmo de **regresión lineal múltiple**, que extiende el enfoque de la regresión lineal simple al considerar múltiples variables predictoras. Esto permitirá examinar cómo la inclusión de variables adicionales puede mejorar la precisión de las predicciones y brindar una visión más completa de los factores que influyen en los montos transaccionados por ticket en el sector retail.

Para este caso se realiza el escalamiento de los datos con el método MinMaxScaler de la librería sklearn, este escalador transforma los datos para que se encuentren en el rango específico indicado, en este caso, de -1 a 1 [3]. Adicionalmente, se aplica validación cruzada con 10 pliegues utilizando la función cross_validate de scikit-learn, Esto implica dividir el conjunto de entrenamiento en múltiples partes (10 en este caso) y ejecutar el entrenamiento y la

evaluación del modelo en diferentes combinaciones de conjuntos de entrenamiento y validación [4]. Obteniendo los siguientes resultados:

```
-----INICIO PROCESAMIENTO-----
Resultados RMSE
RMSE_TRAIN: -818.619
RMSE_TEST: -850.561
-----
Resultados R2
R2_TRAIN: 0.850
R2_TEST: 0.836
-----
Resultados MSE
MSE_TRAIN: -670143.773
MSE_TEST: -733188.868
-----
Resultados MAE
MAE_TRAIN: -433.151
MAE_TEST: -433.486
-----
PROCESAMIENTO FINALIZADO EXITOSAMENTE!!!
--- 0.3150031487147013 Minutos ---
```

Fig. 6. Tabla de resultados para las métricas aplicadas al modelo de regresión lineal múltiple.

En general, estos resultados indican que el modelo de regresión lineal múltiple tiene un buen ajuste a los datos, con valores de RMSE, R^2 , MSE y MAE relativamente bajos.

Finalmente, en la primera iteración (notebook iteracion1ML.ipynb)² se aplicó una Regresión Lineal Múltiple con regularización Ridge y Lasso. Obteniendo los siguientes resultados:

```
-----INICIO PROCESAMIENTO-----
Regresión Lineal Normal:
MSE: 669930.3006431934
R^2: 0.8487875815111954

Regresión Ridge:
MSE: 670876.0610185396
R^2: 0.848574110477672

Regresión Lasso:
MSE: 690713.0200782518
R^2: 0.8440966378928334
-----
PROCESAMIENTO FINALIZADO EXITOSAMENTE!!!
--- 0.8063454031944275 Minutos ---
```

Fig. 7. Tabla de resultados para las métricas aplicadas al modelo de regresión lineal múltiple con regularización Ridge y Lasso.

En general, al comparar estos tres algoritmos, se observa que la regresión lineal simple y la regresión Ridge tienen resultados similares en términos de MSE y R^2 , mientras que la regresión Lasso tiene un MSE ligeramente mayor y un R^2 ligeramente menor. Sin embargo, la regresión Lasso puede proporcionar la ventaja adicional de seleccionar variables importantes y simplificar el modelo. La elección entre estos algoritmos dependerá de las características específicas del problema y los objetivos de análisis.

C. Etapa de aplicación de algoritmos de ML (Aprendizaje automático)

Para esta etapa se profundiza en el análisis exploratorio de los datos, además se crea y evalúa modelos de regresión utilizando árboles de decisión, bosques aleatorios y SVM, en términos generales los siguientes fueron los pasos aplicados a nuestro conjunto de datos (notebook iteracion2ML.ipynb)³:

- Descripción estadística de los datos y la distribución de algunas variables.
- Creación de una nueva variable numérica llamada "tipo_cliente" basada en la variable "TotalQuantity". Donde los registros con un valor menor o igual a 100 se etiquetan como minoristas (0) y los registros con un valor mayor a 100 se etiquetan como mayoristas (1).
- División del conjunto de datos en 3: Un DF con los datos transaccionales de ambos tipos de clientes (df_sample_all), un DF con solo datos transaccionales de clientes minoristas (df_sample_minorista) y uno con los datos de solo clientes mayoristas (df_sample_mayorista)
- Visualización de los datos utilizando gráficos de dispersión e histogramas.
- Creación modelos de regresión basados en árboles de decisión, bosques aleatorios y SVM empleando la biblioteca scikit-learn, para cada uno de los 3 DFs creados (all, minorista, mayorista).
- Se realiza una búsqueda exhaustiva de los mejores hiperparámetros para cada modelo usando la técnica de validación cruzada.
- Finalmente, se evalúa el rendimiento de los modelos utilizando métricas como el coeficiente de determinación (R^2) y el error cuadrático medio (MSE).

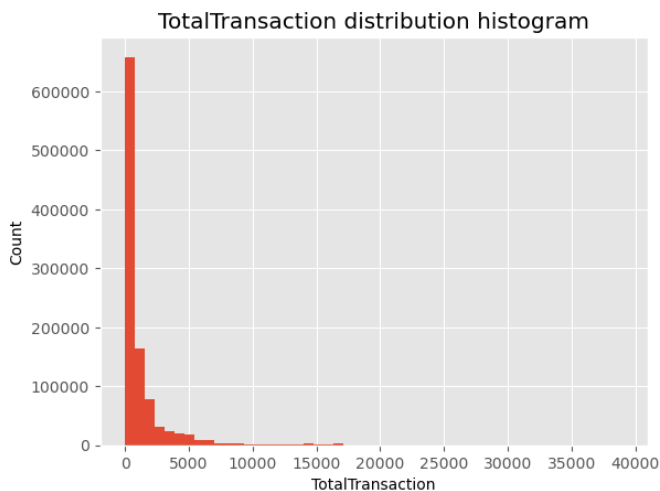


Fig. 8. Distribución de la variable "TotalTransaction"

² Enlace notebook: [iteracion1ML.ipynb](#)

³ Enlace notebook: [iteracion2ML.ipynb](#)

Distribución de la variable tipo_cliente

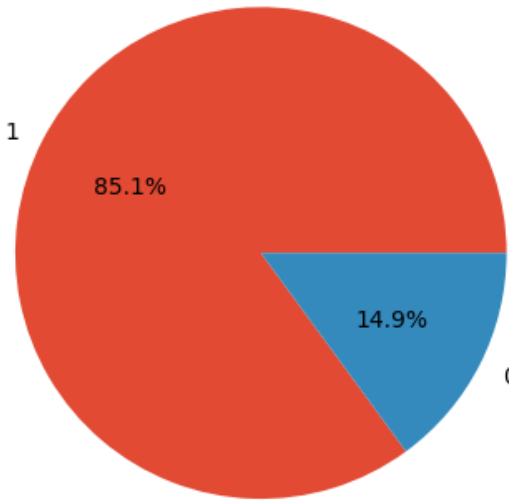


Fig. 9. Composición de la variable “tipo_cliente”, [tipo cliente: 1 Mayorista cantidad: 876610] - [tipo cliente: 0 Minorista cantidad: 154016]

Para este caso se utilizaron los algoritmos de árboles de decisión, bosques aleatorios y máquinas de soporte vectorial, además se realizó el escalamiento de los datos con el método MinMaxScaler de la librería sklearn, en un rango de -1 a 1 [3]. Y se aplica validación cruzada con 7 pliegues utilizando la función cross_validate de scikit-learn [4].

Los siguientes son los resultados obtenidos:

Dataset	Tipo de algoritmo	Característica	Best CV score	Test score	Best Parameters
Todos los clientes	Árboles Simples Variables	TotalQuantity	0.855	0.833	model_max_depth: 10, model_min_samples_leaf: 2, model_min_samples_split: 8
Todos los clientes	Árboles Simples Variables	TotalProductInStock	0.769	0.764	model_max_depth: 10, model_min_samples_leaf: 1, model_min_samples_split: 1
Todos los clientes	Árboles Simples Variables	Invoice	0.473	0.551	model_max_depth: 10, model_min_samples_leaf: 5, model_min_samples_split: 3
Todos los clientes	Árboles Múltiples Variables	TotalTransaction	0.834	0.804	model_max_depth: 10, model_min_samples_leaf: 3, model_min_samples_split: 1
Todos los clientes	Bosques Simples Variables	TotalQuantity	0.820	0.815	model_max_depth: 4, model_min_samples_leaf: 2, model_min_samples_split: 2, model_n_estimators: 150
Todos los clientes	Bosques Simples Variables	TotalProductInStock	0.713	0.693	model_max_depth: 4, model_min_samples_leaf: 2, model_min_samples_split: 2, model_n_estimators: 150
Todos los clientes	Bosques Simples Variables	Invoice	0.254	0.262	model_max_depth: 4, model_min_samples_leaf: 2, model_min_samples_split: 2, model_n_estimators: 150
Todos los clientes	Bosques Múltiples Variables	TotalTransaction	0.950	0.931	model_max_depth: 10, model_min_samples_leaf: 2, model_min_samples_split: 2, model_n_estimators: 150
Todos los clientes	SVM Simples Variables	TotalQuantity	0.546	0.560	model_C: 1, model_degree: 1, model_epsilon: 0.5, model_gamma: 'scale', model_kernel: 'poly'
Todos los clientes	SVM Simples Variables	TotalProductInStock	0.504	0.492	model_C: 1, model_degree: 1, model_epsilon: 0.5, model_gamma: 'scale', model_kernel: 'poly'
Todos los clientes	SVM Simples Variables	Invoice	0.114	0.114	model_C: 1, model_degree: 1, model_epsilon: 0.5, model_gamma: 'scale', model_kernel: 'rbf'
Todos los clientes	SVM Múltiples Variables	TotalTransaction	0.909	0.291	model_C: 5, model_degree: 5, model_epsilon: 0.5, model_gamma: 'scale', model_kernel: 'poly'
Clientes Mayoristas	Árboles Simples Variables	TotalQuantity	0.862	0.872	model_max_depth: 10, model_min_samples_leaf: 1, model_min_samples_split: 3
Clientes Mayoristas	Árboles Simples Variables	TotalProductInStock	0.764	0.622	model_max_depth: 10, model_min_samples_leaf: 1, model_min_samples_split: 1
Clientes Mayoristas	Árboles Simples Variables	Invoice	0.540	0.640	model_max_depth: 10, model_min_samples_leaf: 1, model_min_samples_split: 4
Clientes Mayoristas	Árboles Múltiples Variables	TotalTransaction	0.932	0.921	model_max_depth: 9, model_min_samples_leaf: 1, model_min_samples_split: 4
Clientes Mayoristas	Bosques Simples Variables	TotalQuantity	0.807	0.796	model_max_depth: 10, model_min_samples_leaf: 4, model_min_samples_split: 2, model_n_estimators: 150
Clientes Mayoristas	Bosques Simples Variables	TotalProductInStock	0.704	0.704	model_max_depth: 4, model_min_samples_leaf: 2, model_min_samples_split: 2, model_n_estimators: 150
Clientes Mayoristas	Bosques Simples Variables	Invoice	0.265	0.312	model_max_depth: 4, model_min_samples_leaf: 1, model_min_samples_split: 2, model_n_estimators: 150
Clientes Mayoristas	Bosques Múltiples Variables	TotalTransaction	0.947	0.957	model_C: 1, model_degree: 1, model_epsilon: 0.1, model_gamma: 'scale', model_kernel: 'poly'
Clientes Mayoristas	SVM Simples Variables	TotalQuantity	0.547	0.575	model_C: 1, model_degree: 1, model_epsilon: 0.1, model_gamma: 'scale', model_kernel: 'poly'
Clientes Mayoristas	SVM Simples Variables	TotalProductInStock	0.509	0.581	model_C: 1, model_degree: 1, model_epsilon: 0.1, model_gamma: 'scale', model_kernel: 'rbf'
Clientes Mayoristas	SVM Múltiples Variables	TotalTransaction	0.122	0.120	model_C: 5, model_degree: 5, model_epsilon: 0.5, model_gamma: 'scale', model_kernel: 'poly'
Clientes Minoristas	Árboles Simples Variables	TotalQuantity	0.131	0.106	model_max_depth: 4, model_min_samples_leaf: 1, model_min_samples_split: 1
Clientes Minoristas	Árboles Simples Variables	TotalProductInStock	0.105	0.080	model_max_depth: 4, model_min_samples_leaf: 1, model_min_samples_split: 1
Clientes Minoristas	Árboles Simples Variables	Invoice	0.122	0.108	model_max_depth: 10, model_min_samples_leaf: 1, model_min_samples_split: 4
Clientes Minoristas	Árboles Múltiples Variables	TotalTransaction	0.560	0.476	model_max_depth: 9, model_min_samples_leaf: 2, model_min_samples_split: 2, model_n_estimators: 150
Clientes Minoristas	Bosques Simples Variables	TotalQuantity	0.134	0.111	model_max_depth: 4, model_min_samples_leaf: 2, model_min_samples_split: 2, model_n_estimators: 150
Clientes Minoristas	Bosques Simples Variables	TotalProductInStock	0.128	0.077	model_max_depth: 4, model_min_samples_leaf: 2, model_min_samples_split: 2, model_n_estimators: 150
Clientes Minoristas	Bosques Simples Variables	Invoice	0.045	0.029	model_max_depth: 4, model_min_samples_leaf: 2, model_min_samples_split: 2, model_n_estimators: 150
Clientes Minoristas	Bosques Múltiples Variables	TotalTransaction	0.872	0.913	model_max_depth: 10, model_min_samples_leaf: 2, model_min_samples_split: 2, model_n_estimators: 150
Clientes Minoristas	SVM Simples Variables	TotalQuantity	0.068	0.043	model_C: 1, model_degree: 1, model_epsilon: 0.1, model_gamma: 'scale', model_kernel: 'poly'
Clientes Minoristas	SVM Simples Variables	TotalProductInStock	0.039	0.010	model_C: 1, model_degree: 1, model_epsilon: 0.1, model_gamma: 'scale', model_kernel: 'rbf'

Fig. 10. Tabla de resultados obtenidos en la aplicación de los diferentes algoritmos de clasificación y regresión con técnicas ML.

III. PLAN DE TRABAJO

Luego de una revisión más profunda del trabajo realizado, iterando en la exploración del dataset y aplicando algunos algoritmos durante el curso de Machine Learning I, junto a nuestro asesor nos decantamos por el tema de sistemas de recomendación, específicamente la elaboración de un sistema de recomendación de productos [5].

Para esta nueva iteración es necesario realizar una revisión del estado del arte y con ello tener bases para iniciar la limpieza y adecuación del dataset apuntando a la generación de un sistema de recomendación de productos. La siguiente es una descripción general de lo que se espera desarrollar durante la monografía:

1. Selección del algoritmo: Existen varios algoritmos para construir sistemas de recomendación, y la elección del algoritmo depende del tipo de datos y del objetivo del sistema de recomendación. Los dos enfoques principales son: Filtrado colaborativo y Filtrado basado en contenido.[6]
2. División de datos: Entrenamiento y prueba.
3. Evaluación del modelo: Después de entrenar el modelo, se evalúa su rendimiento utilizando el conjunto de prueba. Se usan métricas como la precisión, el error cuadrático medio, la tasa de aciertos, entre otros, para evaluar qué tan bien se está desempeñando el sistema de recomendación.
4. Generación de recomendaciones: Una vez que el modelo ha sido evaluado y se ha determinado que tiene un rendimiento adecuado, se puede emplear para generar recomendaciones en tiempo real. Dado un cliente o un producto, el modelo podría predecir las recomendaciones basadas en los datos y las características aprendidas durante el entrenamiento.

IV. CONCLUSIONES

A continuación se presentan las conclusiones a las que se llegaron luego de realizar las 2 iteraciones de trabajo:

- La limpieza y preparación de los datos tomaron aproximadamente el 80% del tiempo para el desarrollo del trabajo, contrastando con un 20% para realizar la parte de Machine Learning.
- El tiempo de procesamiento no fue un impedimento al aplicar los diferentes algoritmos de Regresión Lineal, esto contrasta con los Árboles de decisión, Random Forest y SVM, técnicas más robustas que requieren mayor tiempo de procesamiento.
- En la Regresión Lineal Múltiple se consiguieron buenos desempeños aplicando la regresión lineal y sus variantes con regularización Ridge y Lasso, entre estos 3 algoritmos no se encontraron diferencias notorias en los desempeños obtenidos.
- Una optimización de hiperparámetros más robusta posibilitará la obtención de mejores resultados en los algoritmos de Árboles de decisión, Random Forest y SVM.

REFERENCIAS

- [1] J. Li, Y. Li, and M. Li, "Regression Techniques and Data Transformation for Improved Predictions," in Proceedings of the IEEE International Conference on Data Mining, 2019, pp. 150-155.

- [2] K. Smith, "Comparative Analysis of Regression Techniques and Machine Learning Algorithms for Predictive Modeling," IEEE Transactions on Data Science, vol. 5, no. 2, pp. 100-115, 2022.
- [3] F. Pedregosa et al., "Scikit-learn: Machine learning in Python," Journal of Machine Learning Research, vol. 12, pp. 2825-2830, 2011.
- [4] G. Varoquaux et al., "Cross-Validation: Evaluating Estimator Performance," in Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing, 2018, pp. 221-225.
- [5] R. S. Jannach, M. Zanker, A. Felfernig, and G. Friedrich, "Recommender Systems: An Introduction," in Recommender Systems: An Introduction, Cambridge University Press, 2010.
- [6] J. I. Bagnato, "Sistemas de Recomendación", <https://www.aprendemachinelearning.com>, 2019.