

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.

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, Random

Forest y SVM, se evalúa además el rendimiento y precisión de Los modelos con una serie de métricas de desempeño como son: RMSE, R2, MSE y MAE.

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.

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

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.

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	-1187.704802
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	-1165.225222
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	0.756436	-1.077405e+06	-1.078717e+06	-560.753191	-560.502114
11	TotalProductosUnicos	-1249.370700	-1245.590431	0.850580	0.649728	-1.561334e+06	-1.552188e+06	-584.755675	-582.059449
12	CodigoUnico	-2113.437573	-2104.508628	0.000101	0.000071	-4.467220e+06	-4.429936e+06	-1192.438213	-1188.750328
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	-4.430039e+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	-4.430213e+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 R2 presentó los resultados más altos, un valor alto de R2 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 R2 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. 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. 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, R2, 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 modelos, se observa que la regresión lineal simple y la regresión Ridge tienen resultados similares en términos de MSE y R2, mientras que la regresión Lasso tiene un MSE ligeramente mayor y un R2 ligeramente menor. Sin embargo, la regresión Lasso puede proporcionar la ventaja adicional de seleccionar variables importantes y simplificar el modelo. La elección entre estos modelos 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 y bosques aleatorios, en términos generales los siguientes fueron los pasos aplicados a nuestro conjunto de datos:

- 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 y bosques aleatorios empleando la biblioteca scikit-learn, para cada uno de los 3 DFs creados (all, minorista, mayorista).

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

- 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).

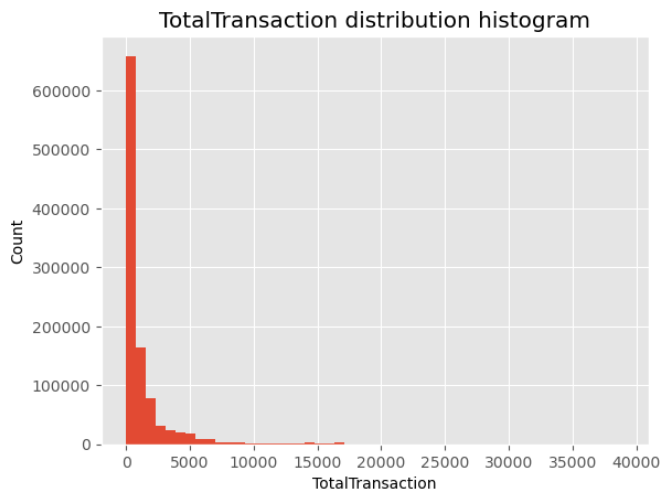


Fig. 8. Distribución de la variable “TotalTransaction”

Distribución de la variable tipo_cliente

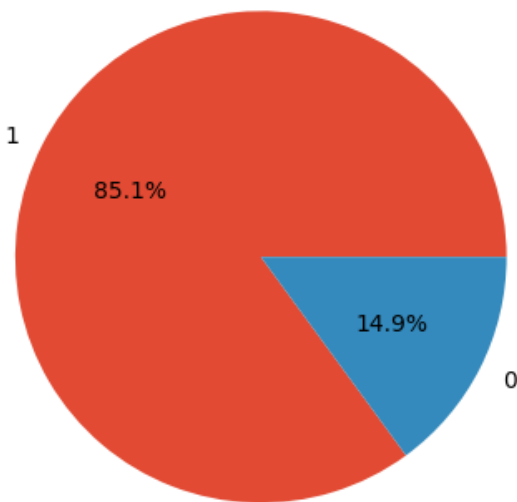


Fig. 9. Composición de la variable “tipo_cliente”, [tipo cliente: 1 Mayorista cantidad: 876610] - [tipo cliente: 0 Minorista cantidad: 154016]

- III. PLAN DE TRABAJO
- IV. CONCLUSIONES

Use SI (MKS) o CGS como unidades primarias. (Se prefieren las unidades del SI.) Pueden usarse las unidades

inglesas como unidades secundarias (en paréntesis). **Esto se aplica a los documentos en el almacenamiento de información.** Por ejemplo, escriba “15 Gb/cm² (100 Gb/in²).” Una excepción es cuando se usan las unidades inglesas como los identificadores en el comercio, como “3½ en la unidad de disco.” Evite combinar SI y unidades de CGS, como la corriente en los amperios y el campo magnético en oersteds. Esto lleva a menudo a confusión porque las ecuaciones no cuadran dimensionalmente. Si usted debe usar unidades mixtas, claramente declare las unidades para cada cantidad en una ecuación.

La unidad del SI para la fuerza del campo magnético H es A/m. Sin embargo, si usted desea usar unidades de T, o referirse a densidad de flujo magnético B o la fuerza del campo magnético simbolizadas como $\mu_0 H$. Use un punto en el centro para separar las unidades compuestas, por ejemplo, “A·m².”

REFERENCIAS

- [1] G. O. Young, “Synthetic structure of industrial plastics (Book style with paper title and editor),” in *Plastics*, 2nd ed. vol. 3, J. Peters, Ed. New York: McGraw-Hill, 1964, pp. 15–64.
- [2] W.-K. Chen, *Linear Networks and Systems* (Book style). Belmont, CA: Wadsworth, 1993, pp. 123–135.
- [3] H. Poor, *An Introduction to Signal Detection and Estimation*. New York: Springer-Verlag, 1985, ch. 4.
- [4] B. Smith, “An approach to graphs of linear forms (Unpublished work style),” unpublished.
- [5] E. H. Miller, “A note on reflector arrays (Periodical style—Accepted for publication),” *IEEE Trans. Antennas Propagat.*, to be published.
- [6] J. Wang, “Fundamentals of erbium-doped fiber amplifiers arrays (Periodical style—Submitted for publication),” *IEEE J. Quantum Electron.*, submitted for publication.
- [7] C. J. Kaufman, Rocky Mountain Research Lab., Boulder, CO, private communication, May 1995.
- [8] Y. Yoroazu, M. Hirano, K. Oka, and Y. Tagawa, “Electron spectroscopy studies on magneto-optical media and plastic substrate interfaces(Translation Journals style),” *IEEE Transl. J. Magn.Jpn.*, vol. 2, Aug. 1987, pp. 740–741 [*Dig. 9th Annu. Conf. Magnetics Japan*, 1982, p. 301].