UNIVERSIDAD
DE ANTIOQUIA
1 8 0 3

Predicciones de recomprar en H&M

FACULTAD DE INGENIERÍA
BIOINGENIERÍA

- Miguel López Vélez Cc: 1001014378
- María Fernanda Villarreal Teherán Cc: 1233345251

El grupo H&M al ofrecer una amplia selección de productos busca desarrollar recomendaciones a los clientes las cuales son basadas en datos transaccionales anteriores y metadatos de clientes y productos, para que de esta manera puedan encontrar rápidamente lo que buscan. En este modelo se intentará predecir si un cliente va a recomprar un artículo utilizando como métrica principal RMS, el dataset posee información sobre los clientes, artículos, imágenes del artículo, transacciones entre otros datos.

Los datos se encuentran clasificados principalmente en tres grandes grupos: artículos, clientes y transacciones, en este caso cada grupo contaba con diferentes variables las cuales entregaban información sobre las compras anteriores de los clientes. En este caso como nos interesa conocer el orden y las posibilidades de compras de los clientes nos centramos en las transacciones realizadas para tener una idea de cuáles son los ítems y los compradores más frecuentes.

De archivos se cuenta con una carpeta de imágenes correspondiente a cada artículo, sin embargo, se debe tener en cuenta que no todos los artículos tienen una imagen correspondiente, se tienen metadatos pata cada artículo disponible para la compra y para cada cliente y finalmente datos de entrenamiento que consisten en compras de cada cliente para cada fecha en donde se encuentran filas duplicadas las cuales corresponden a compras múltiples del mismo artículo. En este caso no se utilizaron los archivos de las imágenes.

La meta principal del proyecto es predecir si un cliente aleatorio recomprara un artículo, con un porcentaje de error final de menos del 10% para ser viable.

1. Análisis de la información

Luego de la lectura de datos se analizó el archivo más relevante para las predicciones el cual es, como se había dicho anteriormente, las transacciones, este cuenta con datos como el artículo, el precio y los canales de venta, a los cuáles se les calculó la moda, se identificaron los 10 valores más representativos de cada categoría y se realizó un análisis de simetría concluyendo que contaba con una distribución asimétrica hacia la derecha.

	t_dat	customer_id	article_id	price	sales_channel_id
0	2019-09-28	be1981ab818cf4ef6765b2ecaea7a2cbf14ccd6e8a7ee9...	706016001	0.016932	2

Figura 1. Moda de cada conjunto de datos.

Debido a que nos interesa predecir la recompra hay que tener en cuenta cuales son los artículos más comprados y los días de mayor compra como se puede observar a continuación:

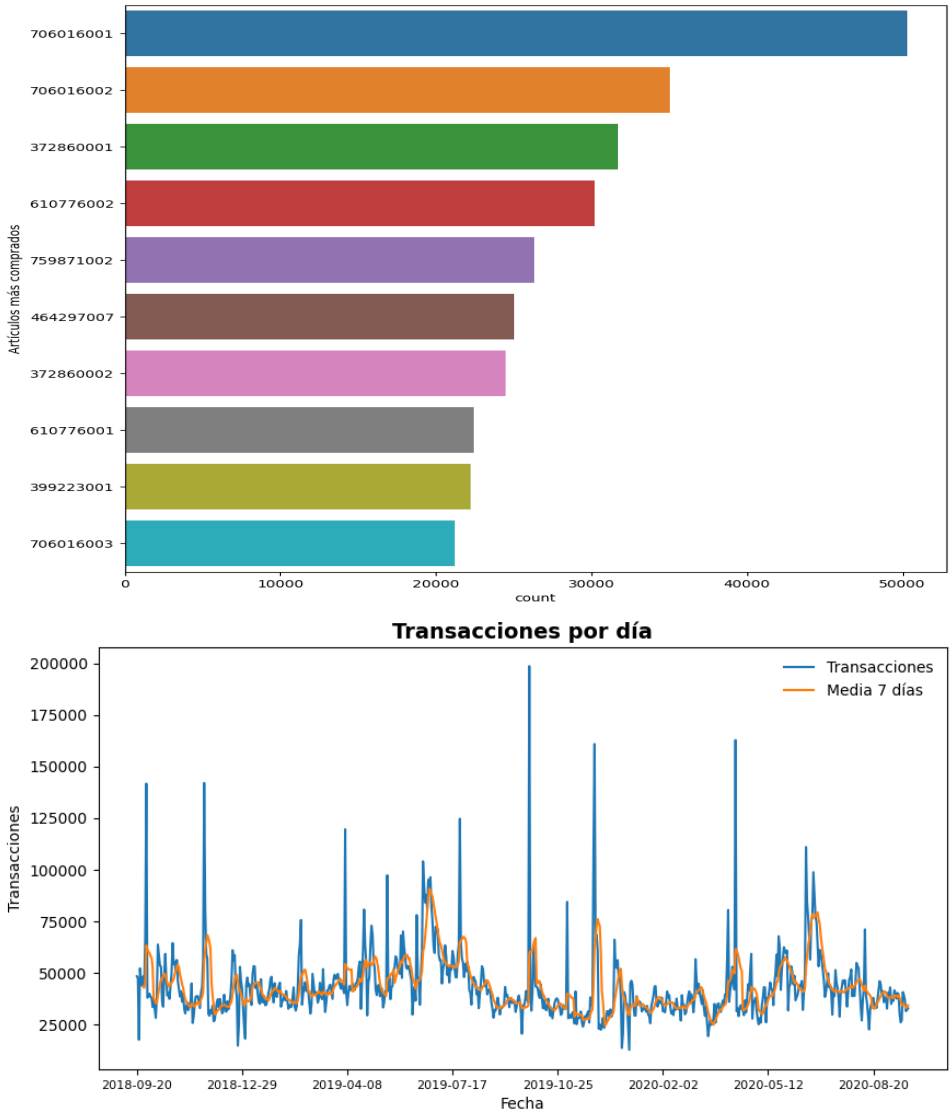

Figura 2. Artículos más comprados y días de mayor compra del dataset.

2. Preprocesamiento de los datos:

Se procede entonces a identificar los datos faltantes en donde luego de analizar el dataset se crearon artificialmente el 5% de los datos en columnas de interés, para esto primero se desorganiza el dataset luego se elimina el 5% de cada variable con la función "sample()" para finalmente restablecer el índice del DataFrame con la función "reset_index()", sin embargo, el dataset queda muy grande por lo que se reducen aún más el número de transacciones.

Cuando se tuvo el nuevo data set se procedió a llenar los datos faltos, en el caso del artículo y el fashion news con la moda, el FN y active con 0 y la edad con el promedio y se procedió a filtrar los artículos que se consideran que afectan más las compras futuras. Debido a que el dataset no tiene una columna explicita de recompra se

procede a crear una analizando los clientes y los article_id en el caso de encontrar una repetición de article_id por parte del cliente se le adiciona un 1 a esa columna de lo contrario se coloca un 0.

Para intenta predecir la recompra de artículos primero se realiza una correlación, lo cual se selecciona los 11 mejores valores que se pueden observar en los resultados arrojando en la matriz mostrada en la figura .

Adicionalmente solo se exporta la mitad de los datos, y en la etapa de generación de modelos solo se toman 221.350 de las filas de manera que sea similar el porcentaje de no recompra 0 o de recompra.

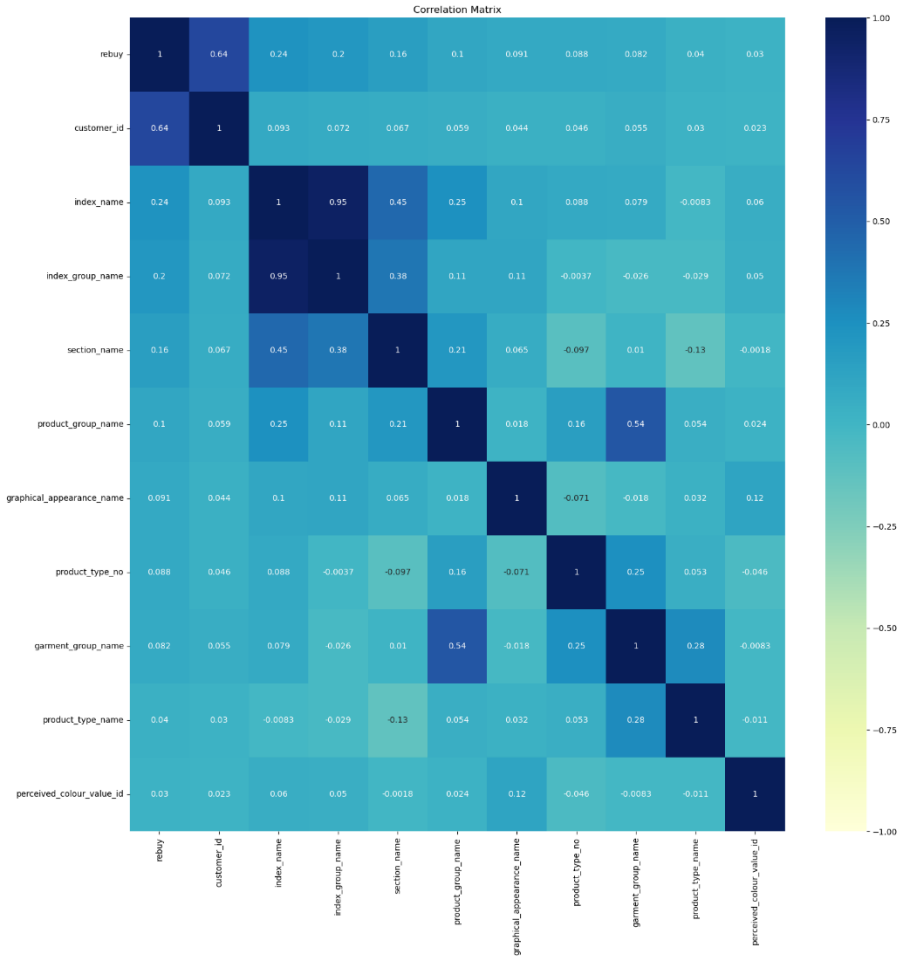

Figura 3. Matriz de correlación

3. Generación de modelos y comparaciones

Se puede observar que los modelos obtienen mejores resultados dependiendo de la cantidad de datos, sin embargo, se debe tener en cuenta que entre más datos se

adicionen más gasto computacional se presenta, por lo que es ideal encontrar un punto medio en el ajuste de los datos. Para esto se utilizan curvas de aprendizaje como se pueden observar en las figuras 10, 11 y 12.

Por otro lado, se usará como métrica principal la RMSE (error de raíz cuadrática media), debido a su validez y uso para problemas de regresión, esta métrica mide la cantidad de error que hay entre dos conjuntos de datos, es decir, compara el valor predicho con un valor conocido.

Adicionalmente, se construye un dataframe con MAE, MSE, R2 y los ratios de MAE y RMSE para su comparación final.

Específicamente MAE (error absoluto medio) es una medida de la diferencia entre el valor pronosticado y el valor real en cada punto pronosticado, mientras que, el MSE es similar a la medida estadística de la varianza, la cual permite medir la incertidumbre alrededor del más probable pronóstico, es decir, se puede ver como la varianza del error del pronóstico y el R2 es un coeficiente de determinación el cual examina cómo las diferencias en una variable pueden ser explicadas por la diferencia en una segunda variable, al predecir el resultado de un evento determinado, es decir, evalúa la fuerza de la relación lineal entre dos variables.

Se procede a analizar entonces cuál es el mejor estimador entre regresión lineal, árbol de decisiones, random forest y SVR lineal, se puede observar en la figura 4 los resultados de la comparación entre las métricas y cada estimador y en la figura 5 que el mejor modelo es Random Forest, el cual es un algoritmo que combina la salida de múltiples árboles de decisión para alcanzar un solo resultado, es decir, por medio de una serie de preguntas cuya respuesta es si o no lleva a tomar una decisión final.

	Model	MAE	MSE	RMSE	R2 Score	MAE Ratio	RMSE Ratio
0	LinearRegression inicial	0.307354	0.136098	0.368915	0.451427	0.565355	0.678590
1	DecisisionTree inicial	0.154772	0.077465	0.278325	0.687761	0.284692	0.511958
2	Random Forest inicial	0.154549	0.076912	0.277330	0.689989	0.284281	0.510127
3	LinearSVR inicial	0.164400	0.164400	0.405463	0.337349	0.302402	0.745818

Figura 4. Comparación entre las métricas y los estimadores

```
best_estimator
```

```
RandomForestRegressor
RandomForestRegressor(max_depth=5, n_estimators=2)
```

Figura 5. Mejor estimador

Luego de seleccionar el mejor estimador se seleccionan los mejores hiperparámetros para este utilizando GridSearchCV con los parámetros mostrados a continuación y dando como resultados los valores de la figura 6.

```
GridSearchCV
GridSearchCV(cv=ShuffleSplit(n_splits=5, random_state=None, test_size=0.4285714285714286,
       train_size=None),
             estimator=RandomForestRegressor(max_depth=5, n_estimators=2),
             n_jobs=-1,
             param_grid={'max_depth': [11, 13, 15, 17, 19],
                         'min_samples_leaf': [1, 2],
                         'n_estimators': [20, 40, 60], 'random_state': [23]},
             scoring='neg_mean_squared_error', verbose=2)
estimator: RandomForestRegressor
RandomForestRegressor(max_depth=5, n_estimators=2)
RandomForestRegressor
RandomForestRegressor(max_depth=5, n_estimators=2)
RandomForestRegressor
RandomForestRegressor(max_depth=13, min_samples_leaf=2, n_estimators=60,
                      random_state=23)
```

Figura 6. Mejores parámetros para Random Forest

A pesar de ya tener seleccionado los mejores parámetros para el mejor estimador se procede a buscar los mejores parámetros para los otros estimadores, esto con el fin de realizar una comparación que nos pueda dar información sobre posibles mejores resultados. Los valores encontrados se observan en la figura 7.

```
Fitting 5 folds for each of 9 candidates, totalling 45 fits
Mejores parámetros para Desission tree:  {'max_depth': 8}
Fitting 5 folds for each of 3 candidates, totalling 15 fits
Mejores parámetros para SVR:  {'C': 1}
```

Figura 7. Mejores parámetros en el resto de estimadores.

Se realizó entonces un análisis de componentes principales PCA (figura 8), el cual es un método muy utilizado de aprendizaje no supervisado y que permite reducir el número de dimensiones tratando de recoger la mayor parte de la variabilidad de los datos originales, en otras palabras, reduce la dimensionalidad de conjuntos de datos

complejos, su objetivo principal es identificar patrones y estructuras subyacentes en los datos, al tiempo que minimiza la pérdida de información, esta comparación se hace inicialmente con Random Forest con los mejores hiper parámetros.

```
------------
Mejor RMSE:  0.25840   ; obtenido con  9  componentes para PCA
```

Figura 8. Resultados de PCA

Luego, se analizó el método NMF, el cual es una técnica de aprendizaje que se utiliza para la reducción de dimensionalidad y la extracción de características en con juntos de datos, a diferencia del PCA que se basa en la descomposición en valore singulares de una matriz, el NMF busca descomponer una matriz no negativa en dos matrices no negativas de menor rango, los resultados de este modelo se observan en la figura 9.

```
------------
Mejor RMSE:  0.30334   ; obtenido con  10  componentes para NMF
```

Figura 9. Resultados de NMF

De igual forma que el PCA se compara el NMF con el árbol de decisión.

En resumen, los mejores hiper parámetros para cada iteración de Random Forest obtenido se puede observar en la siguiente tabla:

	max_depth	min_samples_leaf	n_estimators	random_state
Best Random Forest	13	2	60	23
Best Random Forest+PCA	13	2	60	23
Best Random Forest+NMF	15	2	60	23

Tabla 1. Mejores hiper parámetros Random Forest

Similar para Decision Tree:

	max_depth
Best Decision Tree	8
Best Decision Tree+PCA	8
Best Decision Tree+NMF	8

Tabla 2. Mejores hiper parámetros Random Forest

Para conocer el valor de datos necesarios para entrenar un buen modelo, o ver como responden los diferentes algoritmos a la cantidad de datos de este problema, se realizarán curvas de aprendizaje para observar su comportamiento y así evaluar si se necesita agregar más datos.

Las curvas de aprendizaje muestran cómo evoluciona el rendimiento de un modelo a medida que se incrementa la cantidad de datos de entrenamiento utilizados, representan la relación entre el tamaño del conjunto de datos de entrenamiento y la precisión o el error del modelo de ese conjunto de datos, estas entregan información sobre el ajuste del modelo, la capacidad de generalización y la presencia de problemas sobreajuste o subajuste.

En las figuras 10, 11 y 12 se observan las curvas de aprendizaje para random forest, random forest con PCA y random forest con NMF respectivamente.

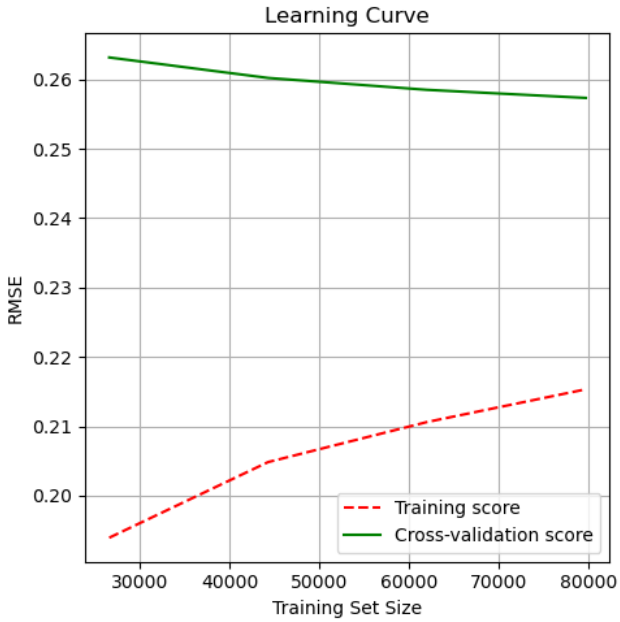

Figura 10. Curva de aprendizaje de Random Forest

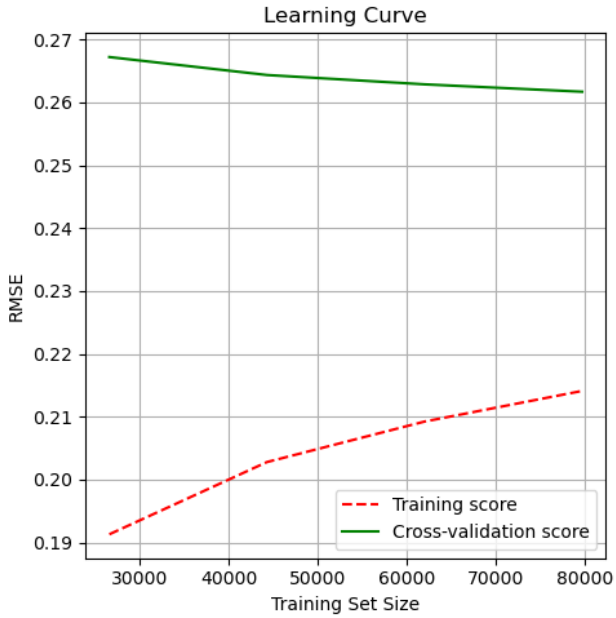

Figura 11. Curva de aprendizaje de Random Forest y PCA

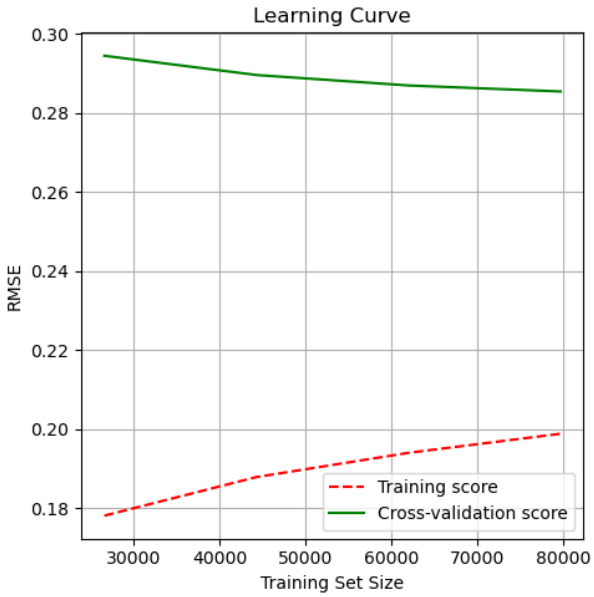

Figura 12. Curva de aprendizaje de Random Forest y NMF

Finalmente, luego de todos los análisis se evalúa cual es el modelo con mejor desempeño a través de una matriz comparativa mostrada en la tabla 3, en este caso el mejor desempeño en general lo tiene el random forest con los cambios a sus hiper parámetros, así que procederemos a hacer una comparación más exhaustiva de sus métricas.

4. Selección de mejor modelo y consideraciones finales:

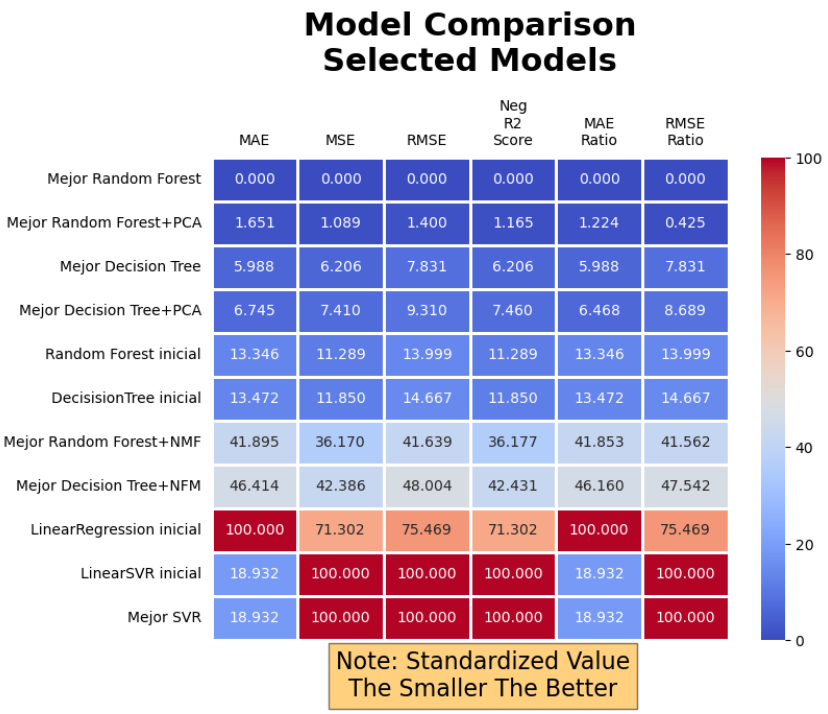

Tabla 3. Comparación para seleccionar el mejor modelo.

Esta matriz fue construida con los valores normalizados mínimos y máximos encontrados para las métricas expuestas, como se mencionó anteriormente en casi todas las métricas es mejor el random forest, pero pese a eso los métodos no supervisados no muestran una mejora considerable y en el caso de NMF para los dos estimadores evaluados tienen un peor desempeño del aproximadamente el 40%, también podemos observar el por modelo en general que sería el SVR y en nuestro caso particular las técnicas no supervisadas no muestran una mejora considerable, a pequeña excepción del PCA con los mejores hiper parámetros y usando random forest.

Se hace una adecuación de los datos predichos porque la idea es tener un 0 o 1 dependiendo de si se realiza la recompra o no, en este caso random forest daba resultados cercanos a 0 o 1 entonces se realiza una modificación con un criterio de 0.55, si el valor es menor a este se toma como 0 y si es superior o igual se toma como un 1. En este caso si se evalúa el RMSE bajo este y predicho modificado su valor incrementara, pero en este caso es para validad la veracidad de la predicción y también es bajo la suposición que al ser el mejor modelo si se hace con los modelos anteriores esta modificación dicho valor de error también subirá debido a que esta métrica mide la distancia del dato real al dato predicho.

Finalmente, ya decidido el mejor modelo y sus mejores hiper parámetros procedemos a hacer un análisis más profundo de su desempeño. Primeramente, se hace un reporte de clasificación como se puede ver a continuación:

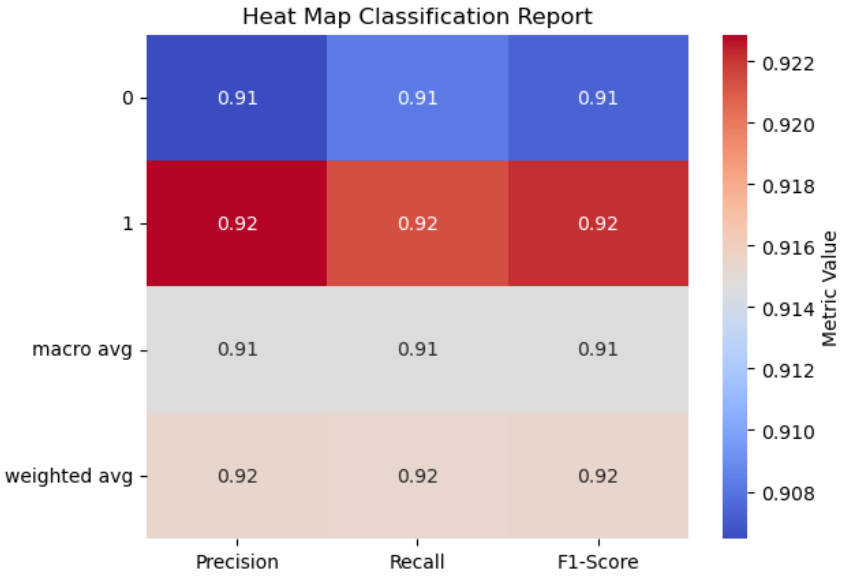

Tabla 4. Reporte de clasificación para el mejor Random Forest.

A partir de esto se observa un excelente porcentaje de predicción de 92% teniendo en cuenta la puntuación F1, pese a que esta se utilice para problemas de clasificación en este caso al ser binomial tiene buenos resultados. Adicionalmente se realiza un matriz de confusión para terminar de evaluar el desempeño del estimador como se puede ver a continuación:

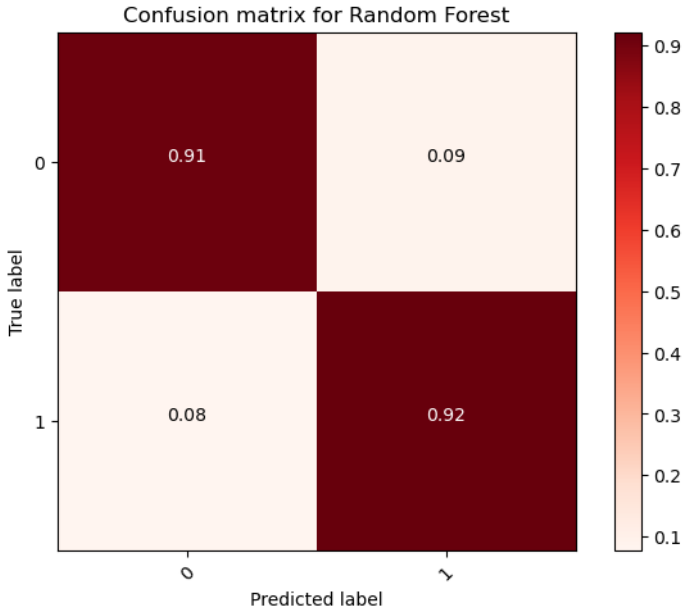

Figura 13. Curva de aprendizaje de Random Forest y NMF.

En este caso se cumple el objetivo del proyecto el cual era intentar predecir la recompra de un artículo por un cliente con la información del dataset y un porcentaje de error menor al 10% ya que, en este caso, el porcentaje de error seria de un 9% y de acierto un 91%.

5. Retos y consideraciones de despliegue

Uno de los principales problemas iniciales fue la formulación del problema, al inicio se intentaba predecir que articulo compraría un cliente con respecto a 7 días después de la compra anterior. Pero esto presentaba un reto bastante grande por la increíble variación de los datos y la cantidad necesaria para entrenar un modelo que diera resultados viables, seguidamente se intentó solo predecir qué articulo compraría un cliente a base de las características con mejor coherencia del dataset, pero en este caso se seguían teniendo valores de error muy grandes y necesitando agregar mucha información. Que en este caso contiene el dataset original, pero por términos computacionales era inviable para el proyecto.

Finalmente se decide por generar una nueva columna de recompra como se menciona en la sección de preprocesamiento, se hace la generación de modelos y optimización de parámetros sobre esta, lo cual da resultados satisfactorios.

Principalmente el problema se puede resumir en la capacidad computacional necesaria para la cantidad de datos necesarios para generar un modelo exitoso con las formulaciones anteriores. Al necesitar replantear el problema también fue necesario cambiar la formulación de las métricas necesarias lo cual se necesitó hacer búsquedas adicionales para obtener un grupo de métricas viables para realizar la predicción de los datos.

En términos de despliegue es necesario conocer los patrones de compra de los clientes recolectando la información necesaria para la predicción, pero en este caso, se puede observar que principalmente se necesita conocer al cliente, y no sus hábitos de compra sino su frecuencia y las secciones de compra habitual de este, estas asunciones se hacen a base de la figura 3 que es la matriz de correlación y estos son los mejores valores.

Finalmente, los procesos iterativos consumen bastante tiempo de cómputo y de corrección de errores entonces el tiempo invertido en el proyecto por sus diferentes modificaciones fue un factor a tener en cuenta en el desarrollo del proyecto.

6. Conclusiones

Primeramente, se puede observar el gran desempeño de random forest en este tipo de problemas de regresión, quedando en los primeros lugares y con buenos valores con respecto a todas las métricas incluso con sus valores iniciales si por ejemplo lo comparamos con SVR o Decision Tree, por otra parte, podemos ver el pobre desempeño de SVR, en este caso siendo la versión lineal que funciona mejor para datasets más largos lo mismo que para el linear regression.

En este caso particular las técnicas no supervisadas no dieron una mejora de desempeño a los algoritmos de una manera significativa, en cambio empeoraron su desempeño en el caso de NFM como se puede observar en tabla 3, no obstante, el PCA presento una mejora sustancial comparándolo con el Decision Tree inicial y el random forest inicial para ambos modelos, pero presentan peor desempeño si lo comparamos con sus versiones con mejores hiper parámetros.

En el caso particular de ramdom forest, pese a ser el mejor modelo se tuvieron que realizar modificaciones finales debido a que los datos dados no estaban bajo el criterio de 1 y 0 necesarios para una estimación real, pese a ello, a través de una simple aproximación con un umbral se pudo observar el potencial del modelo para este tipo de problemas. No solo con los resultados más bajos de error si no con un resultado de exactitud necesario para cumplir los objetivos como se puede ver en la figura 13 y la tabla 4.

En términos de información podemos observar, una adecuada elección de datos basándonos en las curvas de las figuras 10,11,12. Ya que los valores de cross validation disminuyen y se acercan a los valores de train score a medida que se incrementan el número de datos.

Finalmente, gracias las buenas métricas obtenidas y porcentajes de error menores al 10% se puede predecir la recompra de artículos de H&M a base de los parámetros seleccionados y con el mejor random forest posible con los parámetros de búsqueda ingresados. Además de poder comparar su desempeño con otros modelos y poder asegurar la elección certera de un buen modelo para este proyecto además de cumple las metas dispuestas.

Referencias

[1] J. M. Heras, “Error Cuadrático Medio para Regresión”, *Iartificial.net*, 28-dic-2018.
[2] L. Gonzalez, “Evaluando el error en los modelos de regresión”, *Aprende IA*, 23-nov-2018. [En línea]. Disponible en: https://aprendeia.com/evaluando-el-error-en-los-modelos-de-regresion/.
[3] “R2 (R cuadrado) o Coeficiente de Determinación”, *Estrategiasdeinversion.com*. [En línea]. Disponible en: https://www.estrategiasdeinversion.com/herramientas/diccionario/fondos/r2-r-cuadrado-o-coeficiente-de-determinacion-t-1163.
[4] J. M. Heras, “Random Forest (Bosque Aleatorio): combinando árboles”, *Iartificial.net*, 10-jun-2019.
[5] “IBM Documentation”, *Ibm.com*, 17-ago-2021. [En línea]. Disponible en: https://www.ibm.com/docs/es/spss-modeler/saas?topic=mining-oracle-nonnegative-matrix-factorization-nmf.