Laboratorio N°3 Machine Learning

I. Introducción

El presente informe se enmarca como el resultado del Laboratorio N°3, implementando y aplicando algoritmos supervisados y no supervisados de Machine Learning (ML).

Para el desarrollo de este trabajo, se ha seleccionado el siguiente dataset del repositorio de UC Irvine Machine Learning (UCI): 'Online Shoppers purchasing intention dataset' (https://archive.ics.uci.edu/dataset/468/online+shoppers+purchasing+intention+dataset).

La elección de este conjunto de datos se basa en su relevancia para abordar uno de los desafíos más habituales en el campo de la ciencia de datos: determinar la conversión, es decir, si un cliente realiza o no una compra. Esta variable objetivo es central en muchos análisis y representa un problema recurrente para los Data Scientists. Por otra parte, este dataset consta de 12.330 instancias y 17 características o variables explicativas, que abarcan tanto datos numéricos como categóricos, lo que proporciona una fuente variada de información para el análisis a desarrollar.

II. Objetivo

Realizar un análisis del dataset 'Online Shoppers purchasing intention', considerando al menos un análisis de ML no supervisado y otro supervisado.

III. Metodología

En nuestro análisis, hemos decidido emplear una variedad de algoritmos de clasificación para modelar y entender nuestro conjunto de datos. Los algoritmos seleccionados incluyen Decision Tree, Random Forest, Support Vector Machine (SVM), PCA, XG Boost y AdaBoost. Cada uno de estos modelos tiene características únicas que los hacen adecuados para diferentes tipos de datos y problemas. Al probar una variedad de algoritmos, buscamos identificar el modelo que mejor se adapte a nuestras necesidades específicas y proporcione los resultados más precisos y confiables.

Dada la naturaleza de nuestro conjunto de datos y el objetivo de nuestro análisis, hemos decidido utilizar el "recall" específicamente para Revenue True como nuestra principal métrica de evaluación. El recall es una medida de la capacidad del modelo para capturar y clasificar correctamente todos los casos relevantes. En nuestro contexto, un "caso relevante" se refiere a un cliente que realiza una compra (Revenue True). Al maximizar el recall para esta clase, buscamos asegurarnos de que nuestro modelo identifique a tantos clientes potenciales como sea posible, minimizando así el riesgo de perder oportunidades de venta.

Nuestro criterio principal para seleccionar el mejor modelo se basa en maximizar el recall de la clase positiva (Revenue True), ya que esto se alinea directamente con nuestro objetivo de negocio de identificar a los clientes potenciales. Sin embargo, también estamos conscientes de la importancia de mantener un equilibrio, ya que un recall extremadamente alto podría venir a expensas de clasificar incorrectamente a muchos no clientes como clientes potenciales. Por lo tanto, también consideraremos otras métricas como la

precisión, el F1-score y la exactitud para asegurarnos de que nuestro modelo proporciona un rendimiento equilibrado.

Para optimizar el rendimiento de nuestros modelos, hemos realizado ingeniería de características en nuestro conjunto de datos. Las variables categóricas VisitorType, Weekend, Month y Region fueron transformadas en variables dummy para facilitar su uso en los modelos de machine learning. Posteriormente, aplicamos PCA a estas variables para reducir la dimensionalidad de nuestros datos y mejorar la eficiencia computacional de nuestros modelos. Los componentes principales resultantes fueron utilizados como entradas para todos los modelos de clasificación.

Con estas estrategias, buscamos no solo identificar a los clientes potenciales de manera efectiva, sino también optimizar nuestro modelo para que sea computacionalmente eficiente y fácil de interpretar. Creemos que esta metodología nos proporcionará una comprensión profunda de nuestros datos y nos ayudará a tomar decisiones informadas basadas en los resultados de nuestro análisis.

IV. Resultados

EDA

- Estamos ante un dataset de 12.330 instancias y 18 variables: 1 target (REVENUE) + 17 variables independientes o explicativas.
- Las variables independientes tratan sobre: visita a ciertas secciones de la web (informativa, administrativa o de los productos), características del shopper como región de la que proviene, tipo de usuario, su navegador y tipo de tráfico. Además EXITRATE y BOUNCERATE dan cuenta de las tasas de salida y rebote mientras está en la web. Por último hay columnas que describen la época en que la persona navega en la web como si es fin de semana o no, es una fecha cercana a alguna festividad, y el mes en curso.
- Al realizar una análisis de frecuencia sobre la variable target REVENUE (Imagen N°1) Se tiene una base en que es mucho más preponderante la clase False con un 84,5%, vs la clase True con un 14,5%.

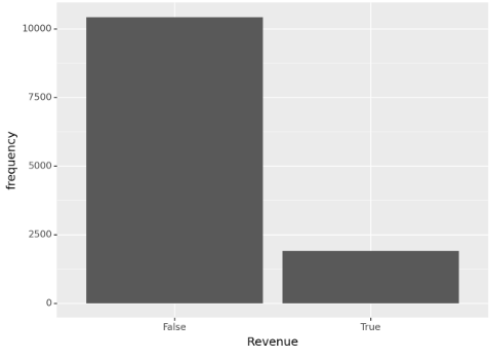

Imagen N°1: Histograma de Revenue catalogado como True y False.

- Al analizar las distribuciones del resto de las variables vale la pena mencionar que están ausente de la base los meses Enero y Abril, en cuanto a SPECIALDAYS se tiene que gran parte de las visitas a la web es durante las festividades o cercano a ellas. Por último,en cuanto al tipo de páginas más visitados, por lejos las relacionadas a productos son las más frecuentadas por los shoppers, por sobre las informativas y las administrativas.

Preparación de los datos

- Proceso de Encoding de columnas (Dummy):

La mayoría de los algoritmos supervisados y no supervisados requieren como input variables numéricas. En este dataset, como vimos anteriormente, tenemos una serie de variables categóricas que convendrá convertir en dummy para no tener inconveniente a la hora de ejecutar los modelos.

Las variables a "dummizar" fueron : VisitorType, Weekend, Month y Región, convirtiéndo estas 4 features en 24 nuevas.

- Normalización de las variables numéricas:

Para empezar a trabajar con los algoritmos, se necesita eliminar el sesgo de las variables numéricas, para así no sesgar posteriormente los modelos. Para deshacerse de esto y otorgarle un mejor input a los modelos, las variables numéricas se normalizaron y escalaron utilizando la herramienta "StandarScaler" de Python. Esto permite transformar las variables para que tengan media 0 y desviación estándar de 1, osea, las variables numéricas tendrán una distribución normal estándar, lo que facilita el uso de modelos de Machine Learning.

Análisis ML No Supervisado:

- Análisis de Componentes Principales (PCA):

En primer lugar, se analizó cuánta varianza aporta cada componente principal individualmente (imagen N°1 izquierda). El primero de los componentes es el que más varianza explica y así sucesivamente, no obstante, mediante este gráfico no está claro qué componentes capturan la mayoría de la información en el dataset.

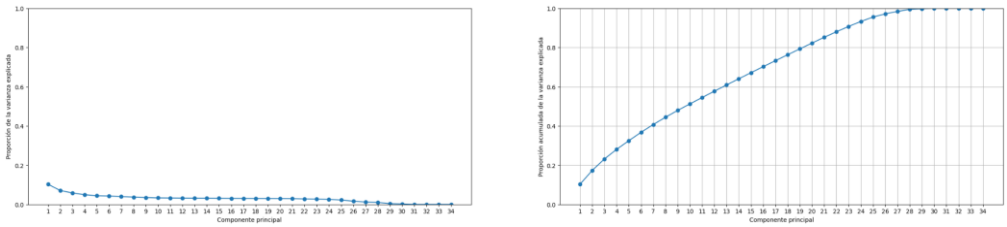

Imagen N°1: Izquierda, Gráfico de varianza explicada por cada componente principal. Derecha, Gráfico de varianza acumulada explicada.

En la imagen N°1 derecha, se muestra cuánta varianza se ha capturado en total hasta cada componente. Mediante este gráfico, se puede visualizar que mediante 20 componentes se ha capturado el 83% de la varianza total, por lo tanto se consideraron 20 componentes en

lugar de todos, reduciendo así la dimensionalidad de los datos de 34 a 20, sin perder mucha información.

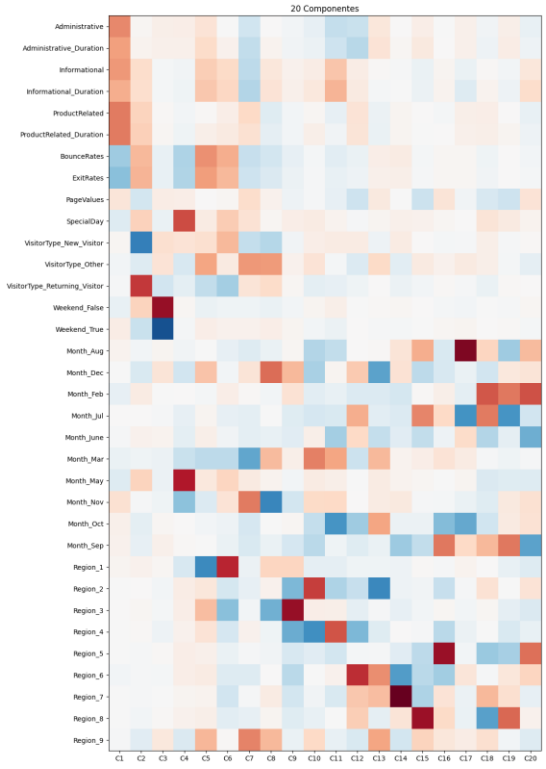

Imagen N°2: Heatmap de Componentes Principales con PCA.

En la imagen N°2 se puede visualizar la relación que tiene cada componente principal resultante del PCA con las respectivas características del dataset. El detalle de cada relación se observa en la tabla N°1.

Componente	Nombre	Explicación
C1	uso_pagina	Esta componente tiene relación positiva con el uso de la web del shopper, independ tipo (administrativa, informativa o de producto). Además tiene relación negativa co y salidas.
C2	visitante_frecuente	Indica cuán frecuente en la web es el shopper (relacionada positivamente con " "ret visitor" y negativamente a "new visitor")
C3	no_findesemana	Indica si visita a página web fue en días laborales (relacionada positivamente con "weekend_false" y negativamente a "weekend_true")
C4	periodo_mayo	Indica si visita fue en época de Mayo (relacionada positivamente con " "Month_Day "SpecialDay")
C5	fuera_region1	Indica la no pertenencia a la Región 1 (relacionada negativamente con variable dum "Region_1")
C6	dentro_region1	Indica la pertenencia a la Región 1 (relacionada positivamente con variable dummy "Region_1")
C7	noviembre_region9	Componente fuerte en shoppers que pertenece a Región 9 y visitan la web en marz personas de esa región en ese mes es más probable que compren por alguna festivi particular).
C8	en_diciembre	Componente relacionada positivamente con quienes visitan el sitio en diciembre y negativamente con quienes visitan el sitio en noviembre
C9	dentro_region3	Indica la pertenencia a la Región 3 (relacionada positivamente con variable dummy

		"Region_3")
C10	dentro_region2_fuera_region	Indica si el shopper pertenece a la Región 2 y si visitó la página en marzo, a su vez, i no pertenencia a la región 4 y la no visita en agosto, diciembre, octubre o septiembr (Relacionada positivamente con 'Region_2', seguida de 'Month_Mar' y negativamer 'Region_4', seguida en menor medida con 'Month_Aug','Month_Dec','Month_Oct' y 'Month_Sep')
C11	dentro_region4_informativa_ ion2_no_octubre	Indica si el shopper pertenece a la Región 4, si visitó la página en marzo y está relaci uso de la web de tipo informativa, a su vez con la no pertenencia a la Región 2, la no octubre, agosto, junio y el no uso de la web de tipo administrativa. (relacionada positivamente con 'Region_4', en menor medida con 'Month_Mar', 'Informational' y 'Informational_Duration'. A su vez, indica la no pertenencia con 'Month_Oct' en me medida con 'Month_Aug', 'Month_June', 'Region_2', 'Administrative' y 'Administrative_Duration')
C12	dentro_region6y7_fuera_regi _administrativas	Indica si el shopper pertenece a la Región 6 y en menor medida a la Región 7 y 8, si página en julio, diciembre y junio, a su vez con la no pertenencia a la Región 4, (rela positivamente con 'Region_6', en menor medida con 'Region_7', 'Month_Jul' y relac negativamente con 'Region_2' y 'Region_4','Administrative' y 'Administrative_Durat
C13	ni_diciembre_ni_region2	Relacionada negativamente con shoppers de Región 2 y con visitas efectuadas en D
C14	dentro_region7	Relacionada fuertemente con la pertenencia a la Región 7 y negativamente con la pertenencia la Región 6.
C15	dentro_region8_julio	Relacionada a la pertenencia a la Región 8 y a la visita en julio (relacionada positivar con variable dummy "Region_8" y en menor medida a 'Month_Jul')
C16	dentro_region5_septiembre	Relacionada a la pertenencia a la Región 5 (relacionada positivamente con variable "Region_5" y en menor medida a 'Month_Sep')
C17	en_agosto_no_julio_ni_octub	Componente relacionada positivamente con quienes visitan el sitio en agosto (quizá importante pues en este mes es el día del Niño).
C18	en_febrero_julio	Componente relacionada positivamente con quienes visitan el sitio en febrero o juli importante pues en este mes es San Valentín).
C19	region8_febsep_no_julio	Relacionada positivamente con pertenencia a la región 8 y si visita a la web ocurre e Febrero o Septiembre. A su vez relación negativa con visitas en Julio
C20	region5_feb_no_sep_ni_junio	Relacionada positivamente con pertenencia a la región 5 y si visita a la web ocurre e Febrero. A su vez relación negativa con visitas en Junio y Septiembre.

Tabla N°1: Desglose de Componentes Principales y su relación con features del dataset.

Esta técnica no sólo permitió disminuir la dimensionalidad, sino también eliminar la multicolinealidad presente en los datos. Los componentes principales resultantes serán las características o features para el entrenamiento del modelo supervisado que se entrenará en la siguiente fase del proyecto.

Análisis ML Supervisado

Luego de haber reducido la dimensionalidad y eliminar la multicolinealidad de la base, se implementaron los siguientes algoritmos supervisados:

- **Decision Tree Classifier:** El algoritmo de árboles de decisión se entrenó con el objetivo de encontrar la mejor forma de dividir los datos y tomar decisiones precisas sobre la conversión de clientes, dividiendo los datos en nodos según las características claves.
- **Random Forest Classifier:** Este algoritmo es una extensión del árbol de decisión que combina múltiples árboles para así, mejorar la predicción. Este modelo se implementó para intentar mejorar el desempeño del árbol de decisión entrenado previamente, pero ahora buscando reducir el sobre ajuste y mejorar la generalización del modelo.

- **Support Vector Classifier:** Este algoritmo se implementó buscando clasificar los datos en dos categorías distintas (Cliente convertido y cliente no convertido), esto se logra buscando el hiperplano que maximice la separación entre estas dos clases.
- **XGBoost:** Este algoritmo al combinar múltiples modelos más débiles para formar un modelo más fuerte, lo hace ideal para la predicción de clientes, ya que puede identificar características importantes y patrones de comportamiento que influyen en la conversión, además de ayudar a reducir el sesgo de la base.
- **AdaBoost:** Este algoritmo se implementó buscando mejorar la precisión del modelo de clasificación, ya que le da más peso a las instancias clasificadas incorrectamente, lo que enfoca el modelo a corregir sus errores.

V. Validación:

Una vez implementados los modelos supervisados descritos anteriormente, se busca elegir el algoritmo que mejor desempeño maneje en las métricas de rendimiento seleccionadas para la conversión de clientes, los resultados se ven a continuación:

Algoritmo	Accuracy	Target	Precision	Recall	F1-Score	AUC
Decision Tree Classifier	0.8264	Convertido	0.48	0.43	0.45	0.66
Random Forest Classifier	0.8702	Convertido	0.73	0.35	0.47	0.86
Support Vector Machine	0.8658	Convertido	0.76	0.28	0.41	0.86
XGBoost	0.869	Convertido	0.68	0.40	0.50	0.85
AdaBoost	0.8621	Convertido	0.72	0.29	0.41	0.82

Tabla N°2: Resumen de resultados con clientes convertidos.

Algoritmo	Accuracy	AUC	Target	Precision	Recall	F1-Score
Decision Tree Classifier	0.8264	0.6600	Convertido	0.48	0.43	0.45
			No convertido	0.89	0.91	0.90
Random Forest Classifier	0.8702	0.86	Convertido	0.73	0.35	0.47
			No convertido	0.88	0.97	0.93
Suport Vector Machine	0.8658	0.86	Convertido	0.76	0.28	0.41
			No convertido	0.87	0.98	0.92
XGBoost	0.869	0.85	Convertido	0.68	0.40	0.50
			No convertido	0.89	0.96	0.92
AdaBoost	0.8621	0.82	Convertido	0.72	0.29	0.41
			No convertido	0.87	0.98	0.92

Tabla N°3: Resumen de resultados con ambos tipos de clientes.

Como se puede ver en la tabla N°2 y 3, los resultados de la evaluación de las métricas para cada algoritmo, el modelo supervisado con el mejor rendimiento escogido fue XG Boost, dado que presenta el mayor equilibrio (comprobable con el F1-Score) entre la precisión y el recall, métricas que son clave para nuestro contexto, con el fin de tener el equilibrio entre evitar falsos positivos y falsos negativos.

Si bien, XG Boost no presenta el Recall más elevado (presenta el segundo más alto) ni la precisión más alta entre los modelos, con el equilibrio logrado gracias a su capacidad de reducir el sesgo de la base, se logra obtener un accuracy del 87%.

Otro de las razones por las cuales se optó por este algoritmo fue por su capacidad para manejar el desbalance en la proporción de clientes que convierten o no convierten dentro de la base, lo que ayuda a reducir los errores en la identificación a la hora de predecir.

A continuación se mostrarán las matrices de confusión obtenidas con XG Boost, algoritmo elegido y la de Random Forest, algoritmo que presentó el mejor accuracy en la comparación.

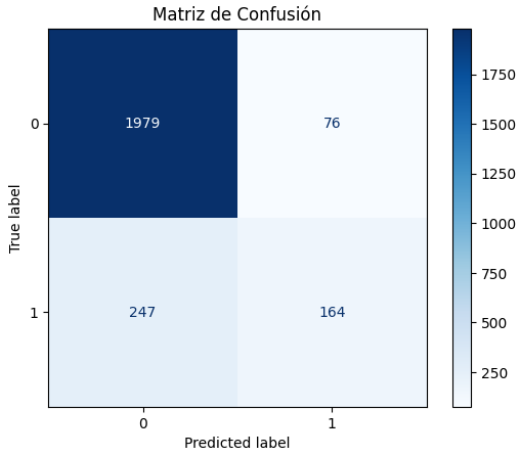

Imagen N°3: Matriz de confusión XG Boost (1: Convertido, 0: No convertido).

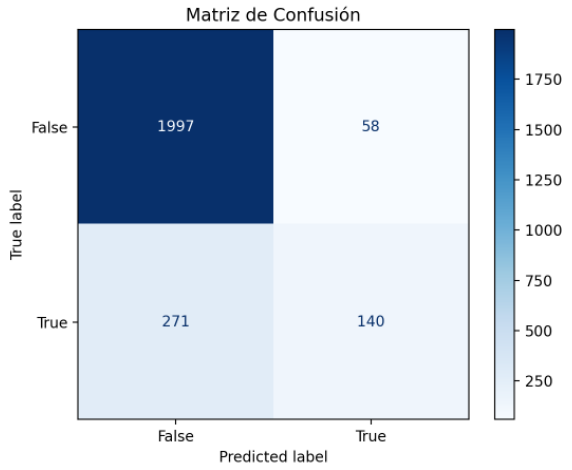

Imagen N°4: Matriz de confusión Random Forest (True: Convertido, False: No convertido)

Podemos notar que XG Boost presenta una cantidad más balanceada de verdaderos positivos y de verdaderos negativos que el modelo Random Forest, esto nos ayuda a comprender a simple vista el desempeño en la predicción de las conversiones, destacando la razón por la cúal se optó por XG Boost, que como se mencionó anteriormente, no fue el modelo con el accuracy más elevado, pero dada sus características y capacidades en la predicción, es el algoritmo que mejor se adapta a la necesidad presentada en este caso,

clasificar correctamente la conversión de clientes. La dificultad presentada fue la naturaleza misma de la conversión, la proporción de convertidos y no convertidos está desbalanceada, condición que en nuestra base de datos estaba presente, sin embargo este algoritmo trata bastante bien el desbalance gracias a su proceso de combinar varios modelos más débiles para formar uno más fuerte.

VI. Conclusiones

Mediante el uso de PCA, se logró reducir la dimensionalidad del dataset de 34 a 20 componentes principales, capturando el 83% de varianza de los datos. Esta reducción no solo permitió una mejora en la eficiencia computacional, sino que también eliminó la multicolinealidad presente, preparando la data para un análisis supervisado más efectivo.

En cuanto a la implementación de algoritmos supervisados, se implementaron diversos métodos de clasificación, incluyendo Decision Tree, Random Forest, SVM, XG Boost y AdaBoost, cada uno con sus características únicas y aplicaciones específicas. El enfoque principal fue maximizar el recall para la clase positiva (Revenue True), buscando identificar a tantos clientes potenciales como fuera posible.

XG Boost se destacó como el modelo con el mejor rendimiento, proporcionando un equilibrio entre recall y precisión, además de manejar eficientemente el desbalance en la proporción de clientes que realizan una compra. Con un accuracy de 87%, este modelo demostró ser robusto y eficaz para predecir la conversión de clientes, lo cual llevado a un caso práctico, puede traducirse en una asignación más eficiente de recursos de marketing, una mejor segmentación de clientes y, en última instancia, un aumento de las ventas.

VII. Discusión

La alta dimensionalidad del dataset y la presencia de multicolinealidad fueron obstáculos que requieren especial atención al inicio del proyecto. A través de la implementación de PCA, se logró abordar este problema, además de comprender cómo la reducción de dimensionalidad puede mejorar la eficiencia del modelo y la interpretación de resultados.

La elección del recall como muestra métrica principal fue una decisión estratégica, alineada con el objetivo de maximizar la identificación de clientes potenciales. Sin embargo, esta elección trajo consigo la necesidad de equilibrar otras métricas para evitar un modelo sesgado. La elección del modelo, en particular la elección de XG Boost, destaca la importancia de considerar el equilibrio entre varias métricas, en lugar de optimizar solo una, asegurando que el modelo sea robusto y confiable.

Sin duda, el hecho que la base haya estado desbalanceada desde un inicio, influyó en el resultado de los algoritmos supervisados. Existen distintas técnicas que permiten equilibrar las clases de una base de entrenamiento desbalanceada. En nuestro caso, posteriormente intentamos utilizando el módulo SMOTE, balanceando la base de entrenamiento sobre el algoritmo Random Forest, obteniendo:

```
Accuracy: 0.8576642335766423
Classification Report:
              precision    recall  f1-score   support

       False       0.90      0.93      0.92      2055
        True       0.59      0.50      0.54       411

    accuracy                           0.86      2466
   macro avg       0.74      0.72      0.73      2466
weighted avg       0.85      0.86      0.85      2466
```

Aquí se debe tener cuidado, pues si bien se logró mejorar el recall de la clase True (de 0.35 a 0.50) la precisión de la misma decae desde 0.73 a 0.59 (es decir, con el balanceo logró detectar más casos True, pero habrán más casos Falsos Verdaderos). El optar por este modelo o el anterior dependerá de qué es más relevante para quien desarrolla el modelo: quizá detectar más True, independiente que me equivoque más (mayor recall) o asegurar de que lo predicho por True realmente lo sea (mejor precisión).

Finalmente, sería beneficioso profundizar en el proceso fine-tunning, de modo de probar diversas combinaciones de hiperparámetros y ver de esta manera si mejoran las métricas de los modelos supervisados.