Instituto Tecnológico y de Estudios Superiores de Monterrey

Inteligencia Artificial Avanzada para la Ciencia de Datos II

Grupo 101

9 de noviembre de 2024

Modulo 2 - Feature Selection

Autor:

Catherine Johanna Rojas Mendoza - A01798149

Rodolfo Jesús Cruz Rebollar - A01368326

Rogelio Lizárraga Escobar - A01742161

Adrián Pineda Sánchez - A00834710

Luis Maximiliano López Ramírez - A00833321

Profesor:

Sebastián Ulises Adán Saldívar

Índice

1. Introducción

Arca Continental enfrenta el desafío de optimizar sus esfuerzos comerciales al identificar a aquellos clientes, con mayor probabilidad de adoptar nuevos productos exitosamente. La introducción de productos novedosos representa un proceso de alto riesgo, ya que no todos los clientes muestran el mismo interés o capacidad para adoptarlos. Para abordar esta situación, el presente proyecto aplicará modelos de inteligencia artificial que analicen patrones de compra, datos de clientes y características específicas de los productos. Esto permitirá a Arca Continental orientar sus estrategias de marketing y ventas hacia clientes con mayores probabilidades de aceptación, maximizando el uso eficiente de recursos y mejorando el éxito en los lanzamientos de productos.

2. Selección y Justificación de las Características

El objetivo de este análisis es construir un modelo que anticipe la probabilidad de adopción de un nuevo producto por parte de los clientes. Para lograrlo, se realizó una cuidadosa selección de características, enfocándose en aquellas variables que mejor representan patrones de comportamiento de compra, preferencias de productos y atributos clave de los clientes. A continuación, se presenta una lista de las características seleccionadas para el modelo:

- **Meses de compra (mes_1 a mes_12)**:
    - Estas columnas representan la actividad de compra mensual de cada cliente, reflejando su consistencia y lealtad en el consumo de productos de la marca. Al incluir los datos de compras mensuales, el modelo puede capturar la regularidad y frecuencia de consumo de cada cliente, lo cual es crucial para entender su disposición a probar nuevos productos. Clientes que compran regularmente podrían tener una mayor probabilidad de probar nuevos lanzamientos.
- **Mes inicial de compra (mes_inicial_1 a mes_inicial_12)**:
    - Las columnas dummies que indican el mes de la primera compra de un cliente permiten analizar patrones estacionales en el comportamiento de adopción. Los clientes que realizan sus primeras compras en determinados meses pueden mostrar una tendencia a explorar productos nuevos, debido a factores como campañas promocionales o estacionalidad de ventas.
- **Tipo de contenedor (bolsa, lata, lata sleek, plástico, tetra pack, vidrio)**:

  - Estas características identifican el envase preferido por cada cliente y permiten al modelo captar las preferencias de consumo en cuanto a presentación del producto. Los clientes que prefieren ciertos tipos de envases pueden mostrar más disposición hacia nuevos productos que se lanzan en estos formatos. Además, el tipo de envase puede estar asociado con determinados estilos de vida o contextos de uso, que son factores importantes en la adopción de nuevos productos.

- **Tipo de producto (AGUA_FUNCIONAL, AGUA_MINERAL, AGUA_PURIFICADA, AGUA_SABORIZADA, BEBIDA_ALCOHOLICA, etc.)**:

  - Incluir el tipo de producto en el modelo permite capturar la afinidad del cliente hacia categorías de productos específicas. Los clientes que consumen productos dentro de una misma categoría podrían estar interesados en probar nuevas variantes o lanzamientos dentro de esa categoría. Esta característica ayuda al modelo a predecir si un cliente mostrará interés en un nuevo producto relacionado con sus hábitos de compra actuales.

- **Número de productos distintos probados (`Num_productos_distintos`)**:

  - Esta variable refleja la diversidad de productos que un cliente ha probado, lo cual es un indicador de su curiosidad o apertura hacia diferentes productos. Un número alto en esta característica puede sugerir que el cliente es un .explorador"de la marca y, por lo tanto, más propenso a probar productos de reciente lanzamiento. Esta variable es clave para evaluar la flexibilidad del cliente en sus hábitos de compra y su disposición a experimentar con nuevos productos.

- **Producto Exitoso (variable objetivo)**:

  - La columna `Producto Exitoso` es la variable que se intenta predecir en el modelo y representa si un producto ha sido adoptado con éxito por un cliente. Para fines de análisis y entrenamiento, esta variable fue derivada evaluando si el cliente ha mantenido compras regulares de un producto durante un tiempo específico, (5 meses consecutivos) lo cual indica aceptación y preferencia.

2.1. Justificación de la Exclusión de Otras Características

Durante el proceso de selección, se probaron variables adicionales relacionadas con los establecimientos alrededor, categorías de edad, hogares familiares, tipo de establecimiento del cliente, sabores y tamaños de productos, entre otras. Sin embargo,

estas características no mejoraron significativamente el rendimiento de los modelos implementados. La decisión de excluir estas variables se basó en los siguientes criterios:

- **Redundancia o baja correlación**: Algunas variables, como las categorías de edad y los tipos de establecimientos cercanos (parques, gimnasios, etc.), presentaban baja correlación con la adopción de nuevos productos. Además, estas variables no mostraron un impacto relevante en el comportamiento de adopción de productos.
- **Complejidad adicional sin beneficio predictivo**: Incluir características como los diferentes sabores y tamaños del producto introducía una mayor complejidad sin aportar beneficios sustanciales al modelo. Esto aumentaba el riesgo de sobreajuste sin mejorar el poder predictivo.
- **Relevancia limitada en la decisión de compra**: Variables como el tipo de establecimiento cliente o la marca del producto no aportaban información significativa en la predicción de adopción de nuevos lanzamientos, ya que no necesariamente determinan la disposición del cliente a probar productos nuevos.

Categoría de Columnas Eliminadas	**Ejemplos de Columnas**
Datos de población y demografía	PADRON_HOMBRES_300m, LISTA_18_HOMBRES_300m
Movilidad diaria y por hora	mov_lunes, mov_8_00_9_59, autos_hora_12
Gastos promedio y específicos	gasto_promedio_300m, pc_gasto_salud_300m
Ingresos y gastos por tipo	ingreso_promedio_300m, ingreso_rentas_300m
Datos de vivienda y entorno	viviendas_300m, prob_VPH_TV_300m
Flujos de personas	flo_sem_tot_300m, flo_finde_e_300m
Datos de accesibilidad y características del entorno	accesibilidad, arboles_300m
Modelos de hogar	modelos_una_persona_conteo_hogares_300m
Descripción del producto y clasificación	Material_desc, GlobalCategory, GlobalFlavor
Información de marca y presentación	Brand, Presentation, Pack
Segmentación y grupo de mercado	SegAg, SegDet, BrandPresRet

Cuadro 1: Categorías y ejemplos de columnas eliminadas

Característica	Descripción
Meses de compra (mes_1 a mes_12)	Representa la actividad de compra mensual del cliente, indicando frecuencia y regularidad en el consumo.
Mes inicial de compra (mes_inicial_1 a mes_inicial_12)	Indica el mes en el que el cliente realizó su primera compra, capturando patrones estacionales de compra.
Tipo de contenedor (bolsa, lata, lata sleek, plástico, tetra pack, vidrio)	Identifica el envase preferido por el cliente, asociando la adopción de productos nuevos en formatos específicos.
Tipo de producto (AGUA_FUNCIONAL, AGUA_MINERAL, etc.)	Refleja la afinidad del cliente hacia categorías específicas, sugiriendo interés en nuevos productos similares.
Número de productos distintos probados (Num_productos_distintos)	Indica la diversidad de productos probados por el cliente, reflejando su apertura a experimentar con nuevos lanzamientos.

Cuadro 2: Resumen de características seleccionadas para el modelo de adopción de productos nuevos

La selección de características se realizó con un enfoque en las variables que mejor representan el comportamiento de compra y las preferencias de producto del cliente. Al limitar el modelo a estas características clave, se logró un equilibrio entre simplicidad y efectividad, optimizando el rendimiento del modelo para predecir la adopción de productos de lanzamiento buscando maximizar el f1 score. Este enfoque asegura que el modelo capture de manera efectiva los patrones de consumo relevantes y pueda ser aplicado para identificar clientes con alta probabilidad de adoptar nuevos productos en futuros lanzamientos.

3. Métodos y Técnicas Utilizadas

3.1. Wrapper Method: Recursive Feature Elimination

Recursive Feature Elimination (RFE) es un método de selección de características que se clasifica dentro de los Wrapper Methods. La característica principal de estos métodos es que utilizan un modelo de aprendizaje para evaluar la importancia o el desempeño de cada subconjunto de características.

3.1.1. Funcionamiento de Recursive Feature Elimination (RFE)

- **Inicio con todas las características:** RFE comienza entrenando un modelo (como regresión logística, SVM o Random Forest) utilizando todas las características disponibles en el conjunto de datos.
- **Evaluación de la importancia de las características:** Una vez que el modelo se ha entrenado, RFE evalúa la importancia de cada característica. Dependiendo del modelo, esta importancia puede calcularse mediante pesos (coeficientes en una regresión) o medidas específicas del modelo (importancia de características en un árbol de decisión, por ejemplo).
- **Eliminación iterativa de características:** RFE elimina la característica menos relevante o con menor importancia en cada iteración. Esto se hace de manera iterativa, donde el modelo se entrena de nuevo después de cada eliminación.

  El proceso se repite hasta que se alcanza un número específico de características deseado o cuando se cumple algún criterio predefinido.
- **Selección del mejor subconjunto:** RFE finaliza cuando solo quedan las características más relevantes, proporcionando un subconjunto de características que, según el modelo y las iteraciones, tienen el mayor impacto en el rendimiento.

Ejemplo de implementación

```
from sklearn.datasets import load_iris
from sklearn.feature_selection import RFE
from sklearn.linear_model import LogisticRegression

# Cargar el conjunto de datos
iris = load_iris()
X = iris.data
y = iris.target

# Definir el modelo base
modelo = LogisticRegression()

# Configurar RFE para seleccionar 2 caracter sticas
selector = RFE(estimator=modelo, n_features_to_select=2, step=1)
selector = selector.fit(X, y)

# Mostrar las caracter sticas seleccionadas
print("Caracter sticas seleccionadas:", selector.support_)
print("Ranking de caracter sticas:", selector.ranking_)
```

El RFE es útil para la selección de características en machine learning porque permite identificar las variables más influyentes en la predicción de la variable objetivo, ayudando a:

- **Reducir el ruido:** Al eliminar características irrelevantes, mejora la interpretabilidad del modelo y reduce la interferencia del ruido".
- **Evitar el sobreajuste:** Disminuir el número de características ayuda a que el modelo se generalice mejor, evitando el ajuste excesivo a los datos de entrenamiento.
- **Mejorar la precisión y la eficiencia:** Entrenar con menos características hace que el modelo sea más rápido y eficiente, manteniendo su precisión.

3.2. Métodos Intrínsecos: Árboles de Decisión con Regularización Lasso

El enfoque combinado de Árboles de Decisión y Regularización Lasso es una técnica avanzada para la selección de características. Esta metodología aprovecha la capacidad de Lasso para realizar una selección inicial de características al penalizar aquellas menos relevantes y, posteriormente, utiliza Árboles de Decisión para refinar la importancia y jerarquía de las características seleccionadas.

3.2.1. Funcionamiento del Método Combinado

- **Regularización Lasso para la Selección Inicial:** Lasso (*Least Absolute Shrinkage and Selection Operator*) aplica una penalización L_1 a los coeficientes del modelo, forzando algunos de ellos a ser exactamente cero. Esto permite identificar un subconjunto inicial de características relevantes al eliminar aquellas que tienen una menor contribución al modelo.
- **Entrenamiento del Árbol de Decisión:** Después de la selección inicial con Lasso, se entrena un Árbol de Decisión utilizando únicamente las características seleccionadas. Este modelo construye su estructura dividiendo iterativamente los datos con base en las características que mejor separan las clases o minimizan la impureza.
- **Cálculo y Refinamiento de la Importancia de las Características:** El Árbol de Decisión asigna un puntaje de importancia a cada característica seleccionada. Este puntaje se basa en la contribución de cada característica a la reducción de la impureza en las divisiones del árbol. Las características menos relevantes pueden ser eliminadas en esta etapa para optimizar aún más el modelo.

- **Interpretación del Modelo Resultante:** La combinación de Lasso y Árboles de Decisión no solo proporciona un modelo eficiente y preciso, sino también interpretable. Las características finales seleccionadas son aquellas que tienen un impacto significativo tanto en el modelo regularizado como en la estructura jerárquica del árbol.

3.2.2. Ventajas del Enfoque Combinado

- **Selección Robustas:** Lasso elimina características irrelevantes en la primera etapa, reduciendo la dimensionalidad y el riesgo de sobreajuste.
- **Refinamiento de Importancia:** Los Árboles de Decisión asignan una jerarquía adicional, mejorando la precisión y eficiencia del modelo.
- **Equilibrio entre Interpretabilidad y Rendimiento:** Este enfoque produce modelos que son altamente interpretables sin sacrificar el rendimiento predictivo.

3.3. Ejemplo de Implementación

```
from sklearn.datasets import load_iris
from sklearn.linear_model import Lasso
from sklearn.tree import DecisionTreeClassifier
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score
from sklearn.preprocessing import StandardScaler

# Cargar el conjunto de datos
iris = load_iris()
X = iris.data
y = iris.target

# Escalar los datos para aplicar Lasso
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)

# Aplicar Lasso para la selecci n inicial de caracter sticas
lasso = Lasso(alpha=0.01)
lasso.fit(X_scaled, y)
caracteristicas_seleccionadas = lasso.coef_ != 0
X_lasso = X[:, caracteristicas_seleccionadas]

# Dividir los datos en conjuntos de entrenamiento y prueba
X_train, X_test, y_train, y_test = train_test_split(X_lasso, y, test_size
    =0.3, random_state=42)

```

```
# Entrenar el modelo de  rbol  de Decisi n
modelo = DecisionTreeClassifier()
modelo.fit(X_train, y_train)

# Obtener la importancia de las caracter sticas
importancia_caracteristicas = modelo.feature_importances_
print("Importancia de las caracter sticas:", importancia_caracteristicas)

# Predecir y evaluar el modelo
y_pred = modelo.predict(X_test)
print("Precisi n del modelo:", accuracy_score(y_test, y_pred))
```

Listing 1: Implementación Combinada de Lasso y Árboles de Decisión para la Selección de Características

3.3.1. Ventajas del Enfoque Combinado de Lasso y Árboles de Decisión

El uso combinado de Lasso y Árboles de Decisión para la selección de características en machine learning ofrece los siguientes beneficios:

- **Selección inicial robusta:** Lasso reduce la dimensionalidad al eliminar características irrelevantes.
- **Refinamiento adicional:** Los Árboles de Decisión priorizan y asignan importancia jerárquica a las características seleccionadas.
- **Mejora en la generalización:** Este enfoque combinado ayuda a evitar el sobreajuste, especialmente en conjuntos de datos con muchas variables.
- **Eficiencia computacional:** Entrenar con un subconjunto optimizado de características reduce el tiempo de cómputo, sin sacrificar el rendimiento predictivo.

4. Implementación

Implementación del Método Intrinsic con Árboles de Decisión y Regularización Lasso

1. Preparación de los Datos

Para la implementación del método Intrinsic, se seleccionaron las características de entrada X y la variable objetivo y. El conjunto X incluye todas las columnas numéricas y booleanas del `DataFrame df_final`, excluyendo las columnas de identificación y algunas irrelevantes, mediante el filtro:

```
X = df_final.drop(df_final.columns[[0, 1, 2, 3, 4, 18, 31, 32]], axis=1)
X = X.select_dtypes(include=[np.number, 'bool'])
```

La variable objetivo y es la columna `Producto Exitoso`, que representa el éxito del producto en términos de aceptación o ventas.

Luego, los datos se dividieron en conjuntos de entrenamiento y prueba, con una proporción del 70 % y 30 %, respectivamente. Además, se aplicó una estandarización a las características mediante el método `StandardScaler`.

2. Configuración de Lasso para la Selección de Características

Se utilizó la Regularización Lasso (*Least Absolute Shrinkage and Selection Operator*) para reducir inicialmente el espacio de características. Este método aplica una penalización L_1 que fuerza algunos coeficientes a ser cero, eliminando características con menor contribución al modelo. El parámetro α se ajustó iterativamente para seleccionar un máximo de 60 características:

```
alpha = 0.0001
while True:
    lasso = Lasso(alpha=alpha, random_state=42)
    lasso.fit(X_train_scaled, y_train)
    selected_features = X.columns[lasso.coef_ != 0]

    if len(selected_features) <= 60:
        print(f"Alpha  ptimo : {alpha}")
        break
    alpha *= 2
```

Con un α óptimo de 0.0064, se seleccionaron 43 características, incluidas variables temporales (`mes_4`, `mes_9`, etc.), datos demográficos (`pob_ab_300m`, `pob_cmas_300m`) y categorías de productos (`AGUA_PURIFICADA`, `BEBIDAS_DE_SOYA`).

3. Entrenamiento de Árboles de Decisión con Umbrales de Importancia

Tras la selección inicial de características con Lasso, se aplicó un modelo de Árbol de Decisión con criterio de *entropía* para refinar las características y evaluar su importancia. Se probaron tres umbrales de importancia (0.005, 0.01, 0.02) para filtrar características de bajo impacto en el rendimiento predictivo.

```
thresholds = [0.005, 0.01, 0.02]
for threshold in thresholds:
    model = DecisionTreeClassifier(criterion='entropy', max_depth=15,
    min_samples_split=5, random_state=42, class_weight="balanced")
```

```
    model.fit(X_train, y_train)
    feature_importances = model.feature_importances_
    selected_features = X.columns[feature_importances > threshold]
    X_train_selected = X_train[selected_features]
    X_test_selected = X_test[selected_features]
    # Evaluaci n del modelo
```

```
Alpha óptimo para un máximo de 60 características: 0.0064
Características seleccionadas por Lasso: Index(['mes_4', 'mes_9', 'mes_10', 'mes_11', 'mes_12', 'pob_ab_300m',
       'pob_cmas_300m', 'industry_customer_size',
       '_Abarrotes / Almacenes / Bodegas / Víveres',
       'TDC_Proximidad_Independiente', 'bolsa', 'lata', 'plasticos',
       'AGUA_PURIFICADA', 'AGUA_SABORIZADA', 'BEBIDAS_DE_SOYA',
       'COLAS_REGULAR', 'LECHE_UHT_ESPECIALIZADA', 'SABORES_REGULAR',
       '_AGUA MINERAL CON SABOR', '_& NADA', '_ADES', '_COSTA', '_DEL VALLE',
       '_JOYA', '_AGUA', '_CITRUS', '_COLA', '_DURAZNO', '_ENERGY', '_LIMON',
       '_MANGO', '_MANZANA', '_MANZANA CANELA', '_MORAS', '_NARANJA',
       '_NARANJA MANDARINA', '_SABILA', '_UVA MENTA', '_NO RETORNABLE',
       '_RETORNABLE', 'Num_productos_distintos', 'mes_inicial'],
      dtype='object')
Número de características seleccionadas: 43
Precisión del modelo con Lasso: 0.13836704191926474

Resultados del método empírico con métricas detalladas:
   Threshold  Num_Features  Test_Score (Accuracy)  Cross_Val_Score (Accuracy)  \
0      0.005            32               0.783546                    0.776646
1      0.010            20               0.769344                    0.773589
2      0.020             6               0.774651                    0.776646

   Cross_Val_Precision  Cross_Val_Recall  Cross_Val_F1_Score
0             0.437612          0.738961            0.549663
1             0.433528          0.741100            0.547023
2             0.436858          0.728841            0.546252
```

Figura 1: Características determinadas por Lasso

Características Temporales

- **mes_4, mes_9, mes_10, mes_11, mes_12**: Estas características temporales corresponden a los meses en los que se registraron patrones de compra o comportamiento significativo. Por ejemplo, *mes_12* podría estar relacionado con aumentos en las ventas durante la temporada navideña, mientras que *mes_4* y *mes_9* podrían reflejar campañas promocionales o comportamientos estacionales específicos.

Datos Demográficos

- **pob_ab_300m, pob_cmas_300m**: Estas variables capturan información sobre la densidad poblacional en áreas específicas alrededor de 300 metros del punto de venta o distribución. La inclusión de estas variables indica que la proximidad y el entorno poblacional influyen directamente en la probabilidad de éxito de los productos.

Categorías de Producto

- **AGUA_PURIFICADA, BEBIDAS_DE_SOYA, COLAS_REGULAR**: Estas categorías representan tipos específicos de productos que fueron identificados como claves en el modelo. *AGUA_PURIFICADA*, por ejemplo, puede ser un producto de alta rotación en áreas donde la calidad del agua es un problema. Las *BEBIDAS_DE_SOYA* y *COLAS_REGULAR* podrían representar tendencias saludables o preferencias consolidadas en el mercado.

Características de Empaque y Retorno

- **bolsa, lata, plásticos**: La preferencia por estos tipos de empaques puede estar correlacionada con costos de producción, tendencias de sostenibilidad o la conveniencia del producto para el consumidor.
- **_NO_RETORNABLE, _RETORNABLE**: Estas características refieren si el empaque del producto es retornable o no, reflejando un posible impacto en la percepción del cliente respecto al valor, reciclaje o sostenibilidad del producto.

Otras Características Importantes

- **TDC_Proximidad_Independiente**: Esta variable probablemente agrupa puntos de venta independientes en áreas de proximidad, lo que podría representar canales clave de distribución.
- **Num_productos_distintos**: Este campo indica la diversidad de productos adquiridos por un cliente o área, reflejando la amplitud de oferta o preferencia por variedad en el mercado.
- **mes_inicial**: La inclusión de *mes_inicial* puede ayudar a capturar la estacionalidad o los ciclos de lanzamiento y promoción de productos.

La precisión del modelo utilizando únicamente las características seleccionadas por Lasso fue de 0.1383, indicando la necesidad de un refinamiento adicional en el modelo y en las que combinado con Desicion Tree obtenemos las siguientes metricas.

4. Evaluación del Modelo

Se evaluaron los modelos generados en función de su precisión en el conjunto de prueba, además de realizar validación cruzada con métricas detalladas como precisión, recall y F1-score.

Posterior a la selección inicial de características, se entrenaron Árboles de Decisión para evaluar la importancia de las variables y optimizar el modelo. La Tabla 3 resume las métricas clave para diferentes umbrales de importancia de características.

Threshold	Num. Características	Test Score (Accuracy)	Cross Val Score (Accuracy)	Precision	Recall	F1-Score
0.005	32	0.7835	0.7766	0.4376	0.7390	0.5497
0.010	20	0.7693	0.7735	0.4335	0.7411	0.5470
0.020	6	0.7747	0.7766	0.4368	0.7288	0.5463

Cuadro 3: Métricas del modelo basado en Árboles de Decisión para diferentes umbrales de importancia

A partir de los resultados, el mejor balance entre las métricas de evaluación se obtuvo con un umbral de importancia de 0.005, seleccionando 32 características. Este umbral permitió alcanzar un F1-score de 0.5497, resaltando la importancia de mantener un conjunto de variables suficientemente representativo para garantizar un buen rendimiento del modelo.

5. Experimentación con las características obtenidas con Lasso en el proyecto

Modelo de Regresión Logística

El modelo de Regresión Logística se entrenó con diferentes umbrales de clasificación para optimizar la precisión y el F1-score. Los mejores resultados se alcanzaron con un umbral de 0.60, obteniendo una precisión de 0.83 y un F1-score de 0.65. Este modelo se destacó por su balance entre precisión y recall, lo que permite una clasificación adecuada sin sacrificar mucho en la detección de falsos positivos.

- **Umbral óptimo**: 0.60
- **Precisión**: 0.83
- **F1-score**: 0.65

Modelo de Árboles de Decisión

Para los Árboles de Decisión, se probaron múltiples umbrales, logrando los mejores resultados con un umbral de 0.70, donde se alcanzó una precisión de 0.82 y un F1-score de 0.61. Este modelo resultó eficiente en términos de precisión en comparación con otros modelos más complejos, demostrando la efectividad de los Árboles de Decisión en la selección de características relevantes.

- **Umbral óptimo**: 0.70
- **Precisión**: 0.82
- **F1-score**: 0.61

Modelo de Random Forest

El modelo de Random Forest se optimizó utilizando diferentes umbrales de importancia. El mejor desempeño se obtuvo con un umbral de 0.50, alcanzando una precisión de 0.87 y un F1-score de 0.70. Random Forest mostró un rendimiento superior en comparación con la Regresión Logística y los Árboles de Decisión, gracias a su capacidad de generalización y resistencia al sobreajuste.

- **Umbral óptimo**: 0.50
- **Precisión**: 0.87
- **F1-score**: 0.70

Modelo de CatBoost

Se configuraron cinco variantes de CatBoost, cada una con diferentes hiperparámetros. La mejor configuración fue la número 4, con un umbral de 0.40, logrando una precisión de 0.87 y un F1-score de 0.70. Este modelo fue el más eficiente en términos de precisión y equilibrio entre las métricas de rendimiento, destacándose por su manejo de características categóricas y complejidad en comparación con los modelos anteriores.

- **Configuración óptima**: Configuración 4 con umbral 0.40
- **Precisión**: 0.87
- **F1-score**: 0.70

6. Resultados de la Implementación

Modelo	Mejor F1 Score	Umbral	Comentarios Extra
Regresión Logística 1	0.65	0.60	Logística con buen equilibrio entre precisión y recall.
Regresión Logística 2	0.65	0.60	Muestra mejoras consistentes; similares métricas en diferentes umbrales.
Árboles de Decisión 1	0.61	0.70	Árboles de Decisión ligeramente más bajos en comparación con Random Forest, pero más simples.
Random Forest 4	0.70	0.50	Mejor desempeño general; demuestra que los bosques aleatorios manejan bien la complejidad.
Random Forest 5	0.70	0.50	Similar a RF4, pero usando diferentes hiperparámetros para mejorar robustez.
CatBoost	0.70	0.40	Mejor configuración con hiperparámetros optimizados, destacando en datos categóricos.

Cuadro 4: Comparación de los mejores modelos con sus respectivos F1 scores y umbrales óptimos.

Conclusiones

Conclusión

El método combinado de **Lasso** y **Árboles de Decisión** demostró ser altamente eficaz para la selección de características en el desarrollo de modelos predictivos. Inicialmente, **Lasso** permitió realizar una selección preliminar de las variables más relevantes mediante la regularización, lo que ayudó a reducir la complejidad del modelo eliminando características redundantes o de baja contribución. Este proceso resultó en un conjunto de 43 características clave de forma inicial, que sirvieron como base para el refinamiento posterior.

Posteriormente, los **Árboles de Decisión** fueron utilizados para evaluar la importancia relativa de las características seleccionadas por Lasso, refinando aún más el

modelo. Este doble enfoque no solo redujo el riesgo de sobreajuste, sino que también mejoró la capacidad del modelo para generalizar sobre nuevos datos.

En cuanto a los modelos evaluados, destacaron las siguientes configuraciones:

- **Random Forest 4** y **Random Forest 5** lograron los mejores resultados con un F1-Score de 0.70, mostrando que los bosques aleatorios son particularmente eficaces para manejar la complejidad de los datos gracias a su capacidad para trabajar con múltiples subconjuntos de características.
- **CatBoost**, con un F1-Score de 0.70 y un umbral de 0.40, demostró ser una alternativa eficiente para el manejo de datos categóricos, maximizando la precisión sin perder generalización.
- **Regresión Logística** con características seleccionadas también mostró un desempeño competitivo, alcanzando un F1-Score de 0.65 en diversas configuraciones. Esto subraya su capacidad para ofrecer modelos interpretables con un rendimiento sólido.

El enfoque de selección de características utilizando Lasso y Árboles de Decisión no solo mejoró el rendimiento de los modelos, sino que también redujo significativamente el tiempo de cómputo al trabajar con un subconjunto más manejable de datos. En conclusión, la combinación de técnicas de regularización y selección de características con modelos avanzados como Random Forest y CatBoost resulta en soluciones robustas, precisas y escalables, adaptables a distintos contextos de negocio y tipos de datos.

Wrapper Method: Recursive Feature Elimination

1. Preparación de los Datos

Se seleccionan las características de entrada X y la variable objetivo y.

X contiene todas las columnas de nuestro DataFrame df_final que incluye información de las ventas, de los productos y de los clientes, a excepción de algunas características de identificación. Solo se seleccionan las columnas numéricas y booleanas usando `X.select_dtypes(include=[np.number, 'bool'])`.

La variable objetivo y es la columna `Producto Exitoso`, que representa el éxito del producto en términos de ventas o aceptación. Luego, los datos se dividen en conjuntos de entrenamiento ($X_{\text{train}}, y_{\text{train}}$) y de prueba ($X_{\text{test}}, y_{\text{test}}$) con un 70 % de los datos en el conjunto de entrenamiento y un 30 % en el de prueba. Esto es importante para evaluar el modelo en datos no vistos y asegurar que la selección de características sea generalizable.

2. Configuración de RFE

Definición del Modelo Base: Para RFE, se selecciona `RandomForestClassifier` como el modelo subyacente. Los árboles de decisión en el bosque aleatorio pueden evaluar la importancia de las características, lo cual es útil en la selección de características mediante RFE.

Parámetros de RFE:

- `n_features_to_select=60:` Esto indica que RFE seleccionará 60 características, reduciendo el número total de variables hasta dejar las más relevantes. Este número fue seleccionado tras un análisis exhaustivo de los resultados de las métricas obtenidas en los modelos implementados en nuestro proyecto, luego de realizar múltiples pruebas.
- `step=1`: RFE eliminará una característica en cada iteración, evaluando en cada paso qué conjunto de características maximiza el rendimiento del modelo base.

Ejecución de RFE:

`rfe.fit(X_train, y_train)`: Este paso entrena el RFE en el conjunto de datos de entrenamiento ($X_{\text{train}}, y_{\text{train}}$). Durante el proceso, RFE elimina de manera iterativa las características menos importantes, refina el modelo, y recalcula la importancia de las características hasta alcanzar las 60 variables más relevantes.

3. Entrenamiento del Modelo con las Características Seleccionadas

Se obtienen las características seleccionadas por RFE a través de `X.columns[rfe.support_]`.

Se entrena el modelo `RandomForestClassifier` usando solo estas 60 características en el conjunto de entrenamiento (`X_train[selected_features]`, `y_train`).

4. Evaluación del Modelo

Predicciones: El modelo realiza predicciones en el conjunto de prueba utilizando solo las características seleccionadas por RFE.

Cálculo del F1 Score: Finalmente, se calcula el F1 Score ponderado del modelo en el conjunto de prueba. Este valor mide la precisión y la sensibilidad del modelo, lo que permite evaluar si las características seleccionadas son efectivas para predecir `Producto Exitoso`.

4.1. Resultados de la Implementación de RFE con `RandomForestClassifier`

```
Características seleccionadas: Index(['mes_1', 'mes_2', 'mes_3', 'mes_4', 'mes_5', 'mes_6', 'mes_7', 'mes_8',
       'mes_9', 'mes_10', 'mes_11', 'mes_12', 'Cluster', 'pob_ab_300m',
       'pob_cmas_300m', 'pob_c_300m', 'pob_cmen_300m', 'pob_dmas_300m',
       'pob_d_300m', 'pob_e_300m', 'industry_customer_size', 'MLSize',
       'infancia_0_11', 'adolescencia_12_17', 'joven_Adulto_18_29',
       'adulto_30_49', 'adulto_mayor_50_mas', 'hogar_familiar_conteo',
       'hogar_familiar_no_conteo', 'lata', 'plasticos', 'vidrio',
       'AGUA_PURIFICADA', 'COLAS_LIGHT', 'COLAS_REGULAR', 'SABORES_REGULAR',
       '_AGUA', '_CATEGORIAS EN EXPANSION', '_REFRESCOS', '_COCA-COLA',
       '_DEL VALLE', '_FANTA', '_POWERADE', '_TOPO CHICO', '_CITRUS', '_COLA',
       '_DURAZNO', '_FRESA', '_LIMA LIMON', '_MANZANA', '_MORAS', '_NARANJA',
       '_TORONJA', '_UVA', '_FAMILIAR', '_INDIVIDUAL', '_NO RETORNABLE',
       '_RETORNABLE', 'Num_productos_distintos', 'mes_inicial'],
      dtype='object')
F1 Score del modelo con características seleccionadas: 0.8496040852185983
```

Figura 2: Características determinadas por RFE

En el resultado de la implementación de RFE con el `RandomForestClassifier`, se observan:

- **Características Seleccionadas**: Después de aplicar RFE, se seleccionaron 60 características de un conjunto mucho mayor. Estas características incluyen tanto variables de frecuencia mensual de compras (`mes_1`, `mes_2`, etc.) como otras variables relevantes de contexto de cliente y producto, como `Cluster`, `pob_ab_300m`, variables sociodemográficas (`infancia_0_11`, `joven_Adulto_18_29`, etc.), y variables de tipo y tamaño del producto (como `_COLAS_REGULAR`, `_COCA-COLA`, `_TOP CHICO`, `_INDIVIDUAL`). Esto muestra que el modelo priorizó una combinación de información temporal, demográfica y de producto, lo cual es lógico dada la naturaleza del problema de predicción del éxito del producto.

- **Puntaje F1**: El modelo obtuvo un F1 Score ponderado de aproximadamente **0.8496**. Este valor es un buen indicador de rendimiento, especialmente si consideramos que el F1 Score ponderado toma en cuenta tanto la precisión como la sensibilidad y es útil en contextos con clases desbalanceadas. Un puntaje cercano a 0.85 sugiere que el modelo tiene un rendimiento balanceado, sin inclinarse excesivamente hacia un solo tipo de características.

- **Significado del Puntaje**: Un F1 Score alto indica que el modelo es efectivo para capturar tanto los verdaderos positivos como los verdaderos negativos. Esto sugiere que el proceso de selección de características realizado por RFE fue adecuado para reducir la dimensionalidad del conjunto de datos, manteniendo únicamente aquellas variables que aportan significativamente a la predicción de la variable objetivo.

- **Implicación de las Características Seleccionadas**: Las características seleccionadas incluyen información detallada sobre el cliente (ubicación, tipo de establecimiento, características demográficas) y características específicas del producto (tipo, marca, tamaño). Esto muestra que el éxito de un producto no solo depende de su categoría y marca, sino también de factores del cliente y contexto en el que se vende. Esta selección de características proporciona una visión detallada de qué factores son críticos para el éxito de un producto en el mercado.

5. Experimentación con las características obtenidas con RFE en el proyecto

Se probaron diversos modelos de clasificación para predecir la variable objetivo, evaluando su desempeño con diferentes umbrales de decisión.

1. Regresión Logística

- **Descripción**: Es un modelo lineal que calcula probabilidades y utiliza un umbral para clasificar las observaciones en clases. Es ideal para problemas linealmente separables.
- **Características**:
  - Evalúa la relación lineal entre las características y la variable objetivo.
  - Se probaron cinco versiones de la regresión logística.
  - El mejor resultado fue un **F1 Score de 0.47** para el modelo con un umbral de 0.60 en las versiones Regresión Logística 2, 3, 4 y 5.
- El modelo tiene un buen desempeño general, pero no logra superar otros métodos más complejos como los árboles de decisión o random forests.

2. Árboles de Decisión

- **Descripción**: Son modelos no lineales que dividen el espacio de características en regiones basadas en condiciones de decisión. Son interpretables y manejan relaciones no lineales entre las variables.
- **Características**:
  - Se probaron tres versiones del modelo.

  - El mejor resultado fue un **F1 Score de 0.54** con umbrales entre 0.10 y 0.70 en la versión Árboles de Decisión.
  - Aunque el modelo tiene un buen equilibrio entre precisión y sensibilidad, no logró alcanzar el desempeño de Random Forest en este problema.

- Útil en problemas donde se necesita interpretabilidad. Sin embargo, puede sobreajustarse fácilmente a los datos.

3. Random Forest

- **Descripción**: Es un conjunto de múltiples árboles de decisión entrenados en subconjuntos aleatorios de los datos, lo que reduce el sobreajuste y mejora la generalización.
- **Características**:
  - Se probaron cinco versiones del modelo.
  - El mejor resultado fue un **F1 Score de 0.60** para los modelos Random Forest 4 y 5 con un umbral de 0.50.
  - Este modelo demostró ser el más robusto, manteniendo un equilibrio constante entre precisión y sensibilidad.
- Es el modelo más efectivo para este problema, especialmente con el umbral de 0.50.

4. K-Nearest Neighbors (KNN)

- **Descripción**: Clasifica las observaciones según los puntos de entrenamiento más cercanos en el espacio de características. Es sensible al escalado de las variables y funciona mejor con datos equilibrados.
- **Características**:
  - Se probaron tres versiones del modelo.
  - El mejor resultado fue un **F1 Score de 0.50** con umbrales de 0.30 y 0.40 en KNN 3.
  - Tiene un rendimiento moderado, pero no logra superar a Random Forest.
- Es simple y efectivo en problemas pequeños, pero menos eficiente en problemas con muchas dimensiones o características.

5. Naive Bayes

- **Descripción**: Es un modelo probabilístico basado en el teorema de Bayes, que asume independencia entre las características. Es eficiente y funciona bien con datos categóricos o textuales.
- **Características**:
  - Se evaluó con varios umbrales.
  - El mejor resultado fue un **F1 Score de 0.45** con un umbral de 0.90.
  - Aunque eficiente, el modelo es limitado debido a la suposición de independencia de las características.
- Es un modelo sencillo, pero no se adapta bien a problemas complejos con alta correlación entre las variables.

5.1. Comparación General

Modelo	Mejor F1 Score	Umbral	Comentarios
Regresión Logística	0.47	0.60	Buen desempeño, pero limitado para relaciones no lineales.
Árboles de Decisión	0.54	0.10-0.70	Buen equilibrio, pero menor que Random Forest.
Random Forest	**0.60**	0.50	Mejor modelo probado, balanceado y robusto.
KNN	0.50	0.30-0.40	Competitivo, pero superado por Random Forest.
Naive Bayes	0.45	0.90	Simple y eficiente, pero limitado en problemas complejos.

Cuadro 5: Comparación General de Modelos de Clasificación

El modelo **Random Forest 4** con un umbral de **0.50** destacó como el más robusto y efectivo en términos de desempeño general, alcanzando un **F1 Score de 0.60**. Este resultado evidencia su capacidad para mantener un equilibrio óptimo entre precisión y sensibilidad, superando a otros modelos que, si bien presentan una precisión comparable (como Árboles de Decisión o KNN en ciertos umbrales), no logran igualar su F1 Score.

Al analizar el comportamiento del modelo bajo diferentes umbrales, se observó que su mejor desempeño ocurre con valores intermedios, específicamente con umbrales

de **0.40** y **0.50**, donde alcanzó un **F1 Score máximo de 0.66**. Este resultado refleja un balance excepcional entre precisión y sensibilidad en este rango, lo que lo convierte en una opción ideal para problemas en los que ambos aspectos son críticos. Además, este rendimiento refuerza la efectividad de Random Forest como una herramienta robusta y versátil para problemas de clasificación complejos con múltiples características y posibles clases desbalanceadas.

5.2. Probando CatBoost

Al analizar los resultados de los modelos implementados con **CatBoost**, se observan varias tendencias interesantes.

Modelo 1:

- **Mejor F1 Score**: 0.60 (umbrales 0.50 y 0.60)
- El modelo alcanza un desempeño razonable, pero es superado por otros modelos de CatBoost en este conjunto.

Modelo 2:

- **Mejor F1 Score**: 0.63 (umbral 0.60)
- Este modelo es competitivo, con un buen balance entre precisión y sensibilidad. Es el mejor modelo de este grupo, alcanzando el F1 Score más alto.

Modelo 3:

- **Mejor F1 Score**: 0.61 (umbrales 0.40 y 0.50)
- El modelo tiene un buen desempeño, pero no alcanza el nivel de Modelo 2.

Modelo 4:

- **Mejor F1 Score**: 0.63 (umbral 0.60)
- Este modelo es comparable al Modelo 2, con un desempeño consistente y uno de los mejores F1 Scores de todos los modelos probados.

Modelo 5:

- **Mejor F1 Score**: 0.60 (umbral 0.60)
- Aunque es competitivo, este modelo tiene un desempeño inferior al Modelo 2 y Modelo 4.

5.2.1. Análisis por Umbral

- En general, los mejores F1 Scores se alcanzan en umbrales de **0.60** o cercanos, lo que sugiere que este rango es ideal para equilibrar precisión y sensibilidad.
- Los umbrales más bajos (0.10 - 0.30) tienden a favorecer la sensibilidad a costa de la precisión, resultando en un menor F1 Score.
- Los umbrales altos (0.80 - 0.90) sacrifican sensibilidad, lo que disminuye significativamente el F1 Score.

Los modelos **Modelo 2** y **Modelo 4** son los mejores implementados con CatBoost, alcanzando un **F1 Score de 0.63** con un umbral de **0.60**. Esto los convierte en una excelente opción para maximizar el balance entre precisión y sensibilidad en este problema. Si se buscara una mayor precisión con un ligero sacrificio de sensibilidad, un umbral de 0.50 podría ser una alternativa viable.

5.3. Conclusión RFE

La comparación de modelos mostró que **Random Forest 4** y **CatBoost Modelos 2 y 4** fueron los más efectivos, logrando un **F1 Score de 0.66 y 0.63**, respectivamente, con un umbral de 0.60. Estos resultados destacan la importancia de una adecuada selección de características para optimizar el rendimiento del modelo.

La selección de características, implementada con **Recursive Feature Elimination (RFE)**, permitió reducir la dimensionalidad, eliminando variables irrelevantes o redundantes. Esto no solo mejoró la capacidad de generalización de los modelos, evitando sobreajustes, sino que también redujo costos computacionales, acelerando el entrenamiento. Además, la interpretación del modelo fue más sencilla, facilitando la identificación de patrones clave, como los relacionados con patrones de compra mensuales, demografía y propiedades del producto.

La selección adecuada de variables impacta directamente en el rendimiento de los modelos. Random Forest aprovechó la importancia de las características seleccionadas para alcanzar un equilibrio excepcional entre precisión y sensibilidad. Por su parte, CatBoost demostró su capacidad para manejar datos complejos y correlaciones, destacando en escenarios donde las clases están desbalanceadas.

En problemas reales, como la predicción del éxito de productos, métricas como el **F1 Score** son críticas, ya que equilibran precisión y sensibilidad, asegurando modelos robustos y confiables.

Bibliografía

[1] scikit-learn developers. $sklearn.feature_selection.RFE^{-}scikit-learn1,5,2documentation$. Disponible en `https://scikit-learn.org/stable/modules/generated/sklearn.feature_selection.RFE.html`. 2024.