UdeA

CASO DE ESTUDIO COMPAÑÍA E-CORP

Osiris Contreras
Maritza zapata
Juan Jose Molina
David Toro

CONTENIDO

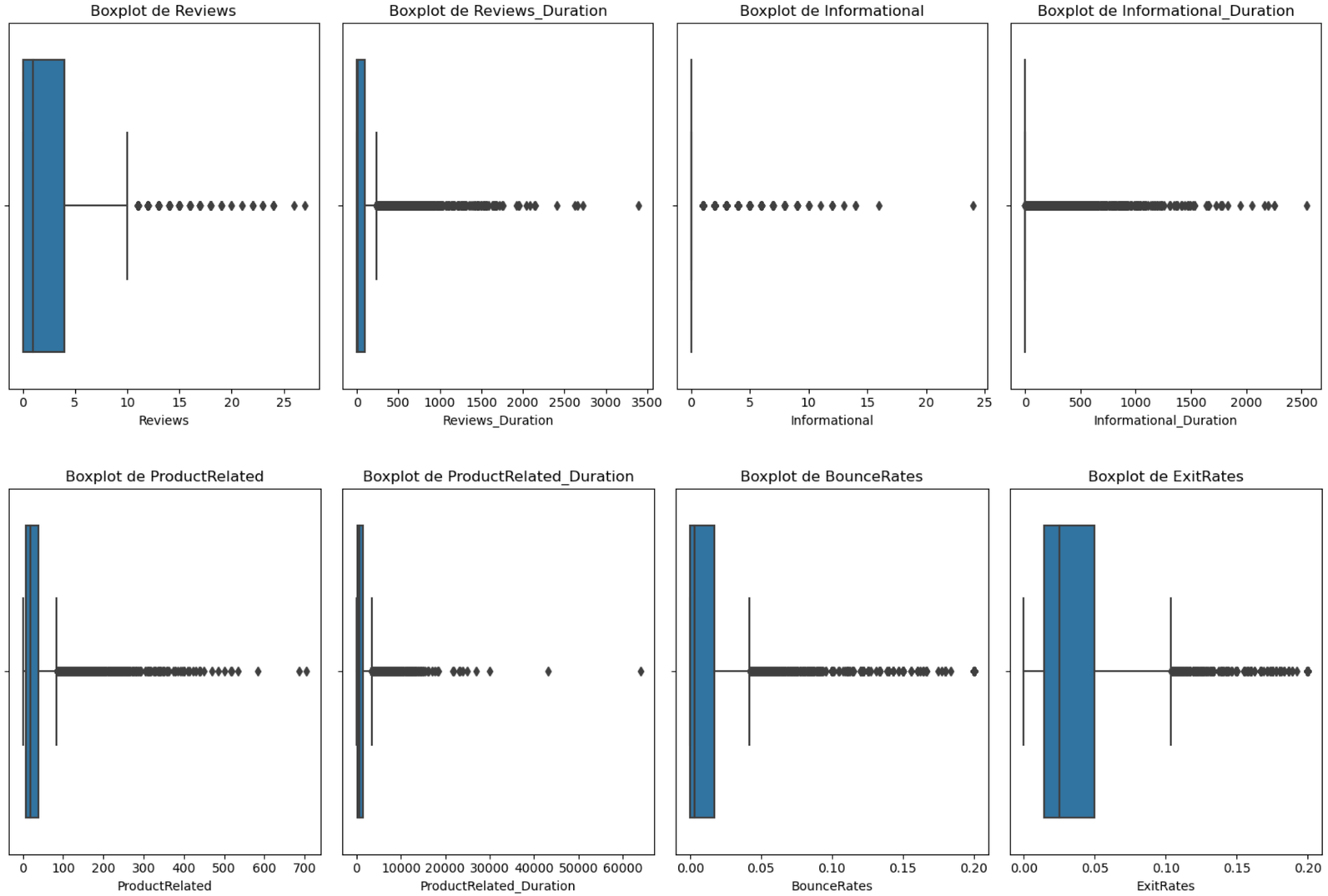
1. Presentación del caso
2. Análisis de los datos
3. Limpieza y transformación de los datos
4. Preparación de los datos
5. Selección de variables
6. Aplicación y comparación de técnicas de modelado
7. Evaluación del mejor modelo
8. Conclusiones

Presentación del caso

E-Corp, líder en productos de lujo, busca expansión digital.

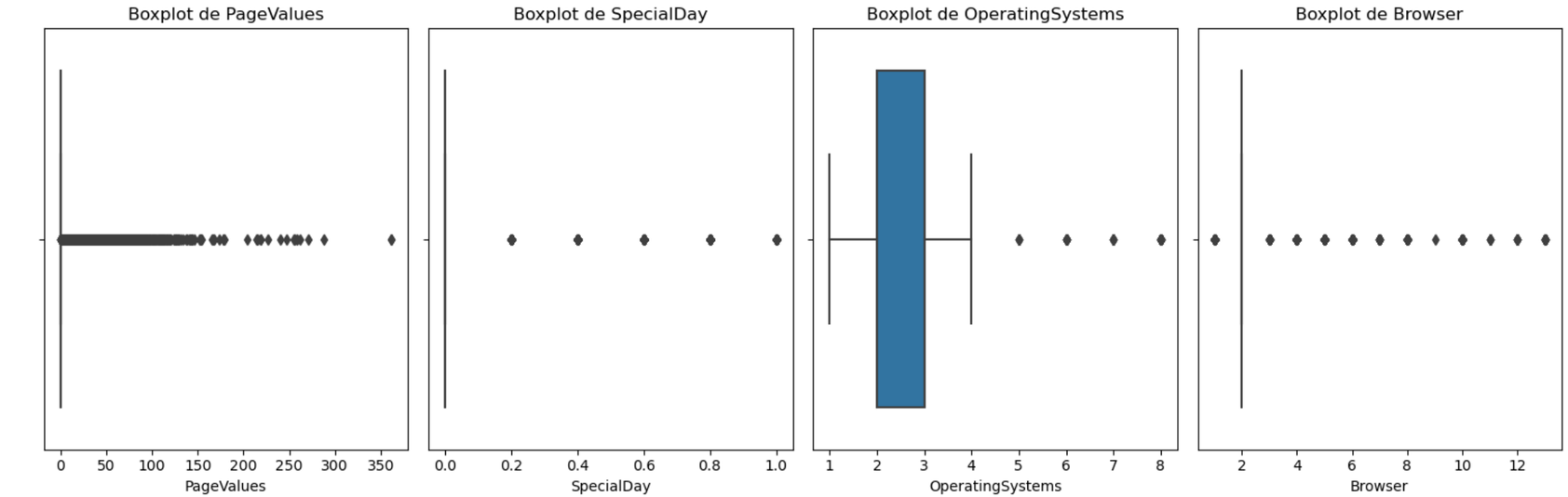
- Problema: Ventas digitales bajas.
- Sospecha: Ineficacia en marketing digital y alcance de audiencia.
- Solución: Contratación de consultores y uso de Machine Learning para:
  - Identificar clientes potenciales.
  - Optimizar inversión publicitaria.
  - Mejorar relevancia de campañas.
  - Incrementar impacto en decisiones de compra.

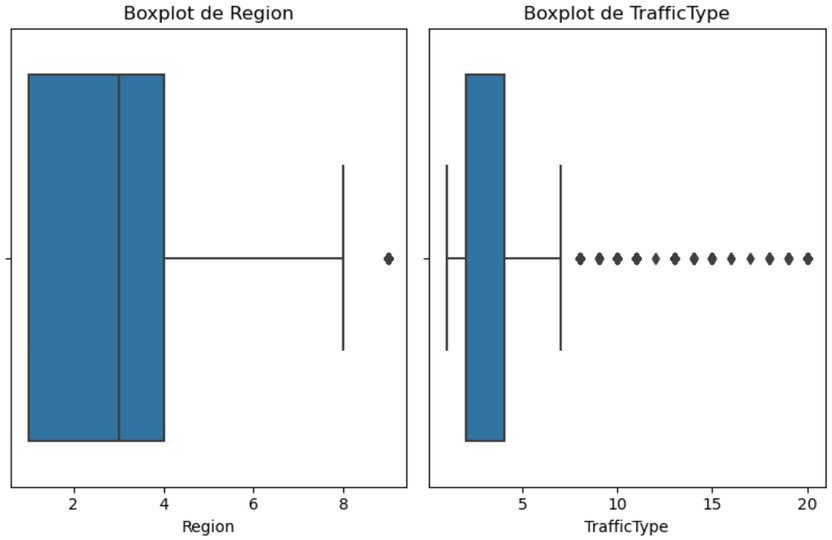
Analisis de los Datos



BOXPLOTS DE VARIABLES NUMÉRICAS

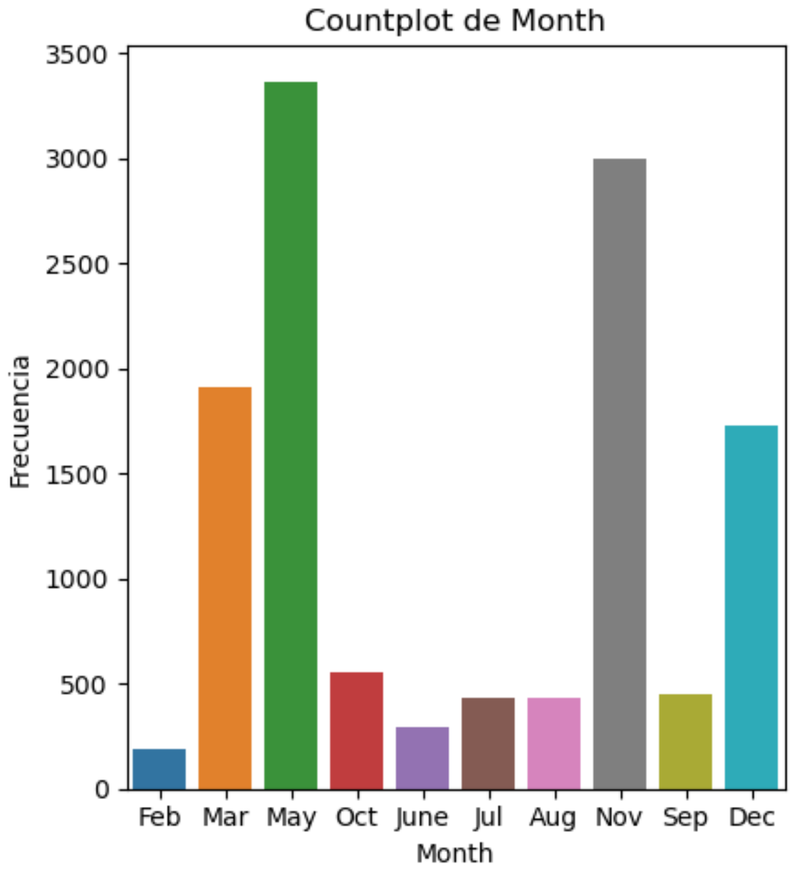
Analisis de los Datos

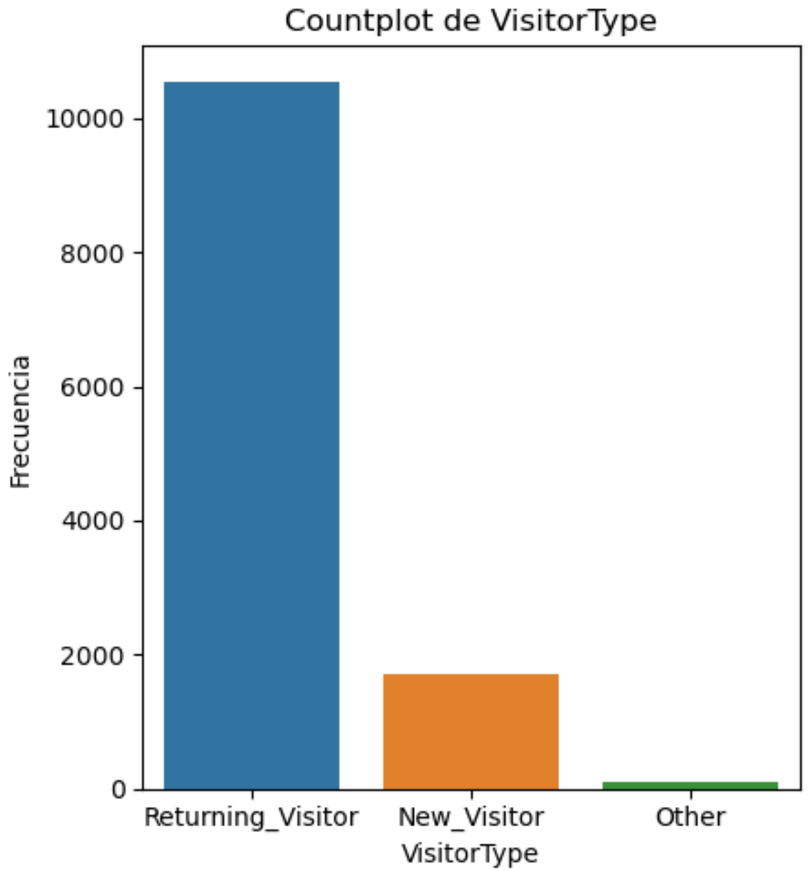




BOXPLOTS DE VARIABLES NUMÉRICAS

Análisis de los datos





COUNTPLOTS DE VARIABLES CATEGORICAS

Análisis de los datos

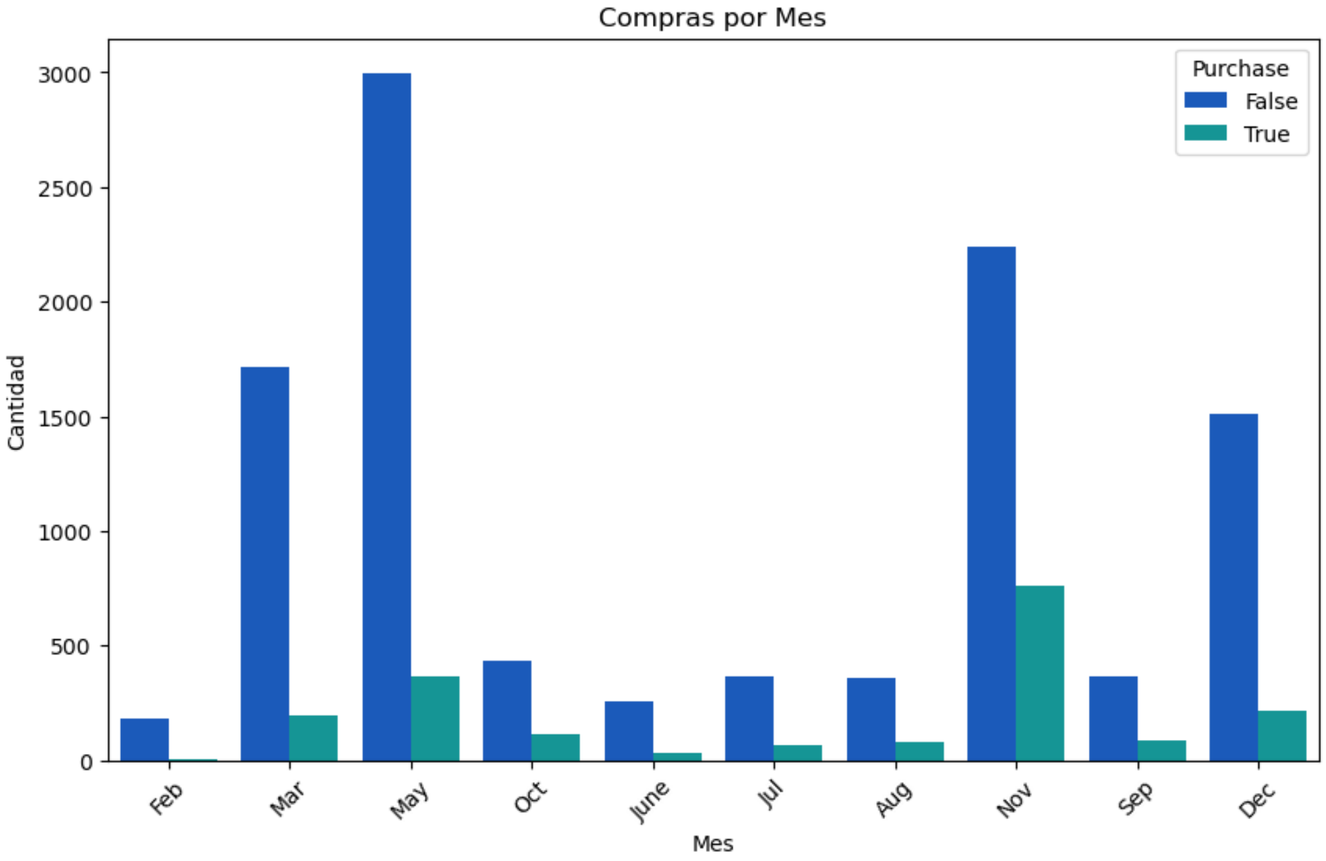


GRÁFICO DE BARRAS DE COMPRAS POR MES

Análisis de los datos

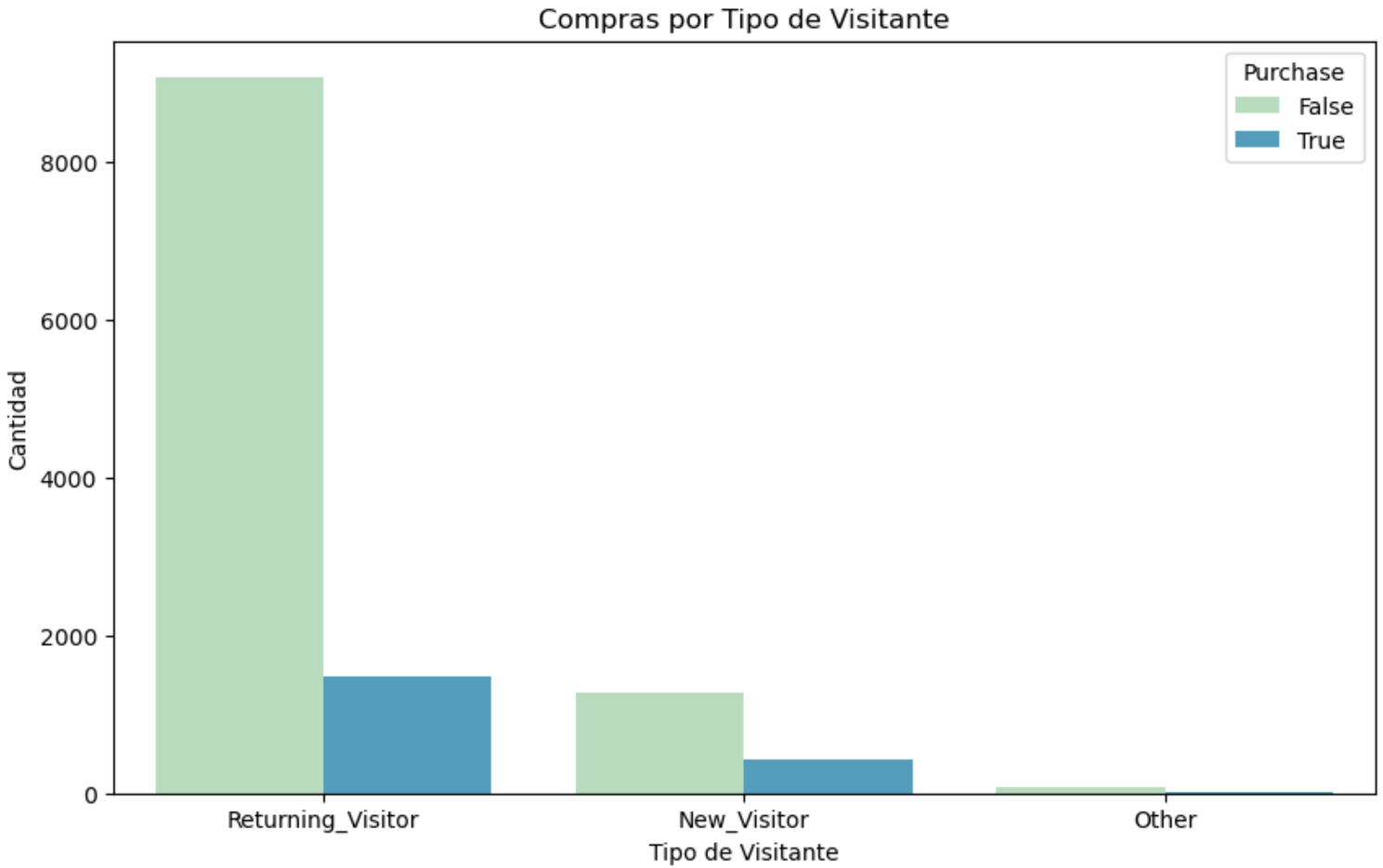


GRÁFICO DE BARRAS DE COMPRAS POR TIPO DE VISITANTE

Limpieza y transformación de los datos

```python
# Información general del dataset
df_original.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 12330 entries, 0 to 12329
Data columns (total 18 columns):
 #   Column                   Non-Null Count  Dtype
---  ------                   --------------  -----
 0   Reviews                  12330 non-null  int64
 1   Reviews_Duration         12330 non-null  float64
 2   Informational            12330 non-null  int64
 3   Informational_Duration   12330 non-null  float64
 4   ProductRelated           12330 non-null  int64
 5   ProductRelated_Duration  12330 non-null  float64
 6   BounceRates              12330 non-null  float64
 7   ExitRates                12330 non-null  float64
 8   PageValues               12330 non-null  float64
 9   SpecialDay               12330 non-null  float64
 10  Month                    12330 non-null  object
 11  OperatingSystems         12330 non-null  int64
 12  Browser                  12330 non-null  int64
 13  Region                   12330 non-null  int64
 14  TrafficType              12330 non-null  int64
 15  VisitorType              12330 non-null  object
 16  Weekend                  12330 non-null  bool
 17  Purchase                 12330 non-null  bool
dtypes: bool(2), float64(7), int64(7), object(2)
memory usage: 1.5+ MB
```

```python
# Copia del dataset para aplicar transformaciones y limpieza de datos
df1 = df_original.copy()
```

```python
from sklearn.preprocessing import LabelEncoder

# Codificación de variables categóricas
label_encoder = LabelEncoder()
df1['Month'] = label_encoder.fit_transform(df1['Month'])
df1['VisitorType'] = label_encoder.fit_transform(df1['VisitorType'])

# Conversión de datos booleanos a numéricos
df1['Weekend'] = df1['Weekend'].astype(int)
df1['Purchase'] = df1['Purchase'].astype(int)

print(df1.head())
```

```python
from sklearn.preprocessing import MinMaxScaler, StandardScaler

# Seleccionar las características numéricas a normalizar o estandarizar
caracteristicas_numericas = df1[names]

# Inicializar el escalador MinMaxScaler
min_max_scaler = MinMaxScaler()

# Normalizar las características utilizando MinMaxScaler
caracteristicas_numericas_normalizadas = min_max_scaler.fit_transform(caracteristicas_numericas)

# Inicializar el escalador StandardScaler
standard_scaler = StandardScaler()

# Estandarizar las características utilizando StandardScaler
caracteristicas_numericas_estandarizadas = standard_scaler.fit_transform(caracteristicas_numericas_normalizadas)

# Convertir las características normalizadas y estandarizadas de nuevo a un DataFrame de pandas
nuevo_df = pd.DataFrame(caracteristicas_numericas_estandarizadas, columns= names)

# Mostrar las primeras filas del DataFrame con características estandarizadas y normalizadas
print("\nCaracterísticas estandarizadas:")
print(nuevo_df.info())
```

```
Características estandarizadas:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 12330 entries, 0 to 12329
Data columns (total 18 columns):
 #   Column                   Non-Null Count  Dtype
---  ------                   --------------  -----
 0   Reviews                  12330 non-null  float64
 1   Reviews_Duration         12330 non-null  float64
 2   Informational            12330 non-null  float64
 3   Informational_Duration   12330 non-null  float64
 4   ProductRelated           12330 non-null  float64
 5   ProductRelated_Duration  12330 non-null  float64
 6   BounceRates              12330 non-null  float64
 7   ExitRates                12330 non-null  float64
 8   PageValues               12330 non-null  float64
 9   SpecialDay               12330 non-null  float64
 10  Month                    12330 non-null  float64
 11  OperatingSystems         12330 non-null  float64
 12  Browser                  12330 non-null  float64
 13  Region                   12330 non-null  float64
 14  TrafficType              12330 non-null  float64
 15  VisitorType              12330 non-null  float64
 16  Weekend                  12330 non-null  float64
 17  Purchase                 12330 non-null  float64
dtypes: float64(18)
memory usage: 1.7 MB
None
```

1.Verificacion de datos nulos y tipos de datos:

2. Codificación de variables categóricas:

3. Normalización y estandarización de características:

Preparacion de los datos

```
# Comprobación de valores duplicados
nuevo_df.duplicated().sum()
```

```
125
```

```
# Eliminación de valores duplicados
nuevo_df = df1.drop_duplicates()

# Comprobación de valores duplicados
nuevo_df.duplicated().sum()
```

```
0
```

```
# Separación de caracteristicas y target (X , y)
y = nuevo_df['Purchase']
X = nuevo_df.drop(['Purchase'],axis=1)

# Separación en conjuntos de entrenamiento y validación con 80% de muestras para entrenamiento
x_train, x_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=0)

#Imprimir Tamaño de dataset
print("Tamaño del conjunto de entrenamiento:", x_train.shape)
print("Tamaño del conjunto de validación:",  x_test.shape)

```

```
Tamaño del conjunto de entrenamiento: (9764, 17)
Tamaño del conjunto de validación: (2441, 17)
```

1.Comprobación y Eliminación de Datos Duplicados:

2. Separación de Características y Objetivo:

3. División de Conjuntos de Entrenamiento y Validación:

Selección de variables

```python
from sklearn.feature_selection import VarianceThreshold

#Función de filtro de caracteristicas
def variance_threshold(X,th):
    var_thres=VarianceThreshold(threshold=th)
    var_thres.fit(X)
    new_cols = var_thres.get_support()
    return new_cols
```

```python
# Para clasificación
# Obtener columnas seleccionadas
X_new_class = variance_threshold(x_train, 0.25)
# Nuevo dataframe
df_classification_new = x_train.iloc[:,X_new_class]
df_classification_new.info()
```

Selección de variables

Variables seleccionadas:

- Reviews: número de páginas de este tipo (Reviews) que visitó el usuario
- Reviews_Duration: que es la cantidad de tiempo dedicado a esta categoría de páginas
- Informational: número de páginas de este tipo (informativas) que visitó el usuario
- Informational_Duration: cantidad de tiempo dedicado a esta categoría de páginas
- ProductRelated: número de páginas de este tipo (relacionadas con productos) que visitó el usuario
- ProductRelated_Duration: cantidad de tiempo dedicado a esta categoría de páginas
- PageValues: Métrica arrojada por Google Analytics que representa el valor medio de una página web que un usuario visitó antes de completar una transacción de comercio electrónico

Selección de variables

- Month: Mes en el que se realizó la visita al sitio web
- OperatingSystems: Sistema operativo usado por el usuario para navegar en el sitio web
- Browser: Navegador usado por el usuario para navegar en el sitio web
- Region: Región (ubicación geográfica personalizada) desde la cual el usuario navega en el sitio web
- TrafficType: Variable que indica el tipo de trafico al cual pertenece el usuario que navega en el sitio web (por ejemplo, si llegó al sitio desde un anunció o a través de una búsqueda)
- VisitorType: Tipo de usuario que ingresa al sitio web

Selección de variables

```
Coeficientes del estimador Lasso:
[ 0.00000000e+00  1.39668775e-05  0.00000000e+00  0.00000000e+00
  0.00000000e+00  2.35134808e-05 -0.00000000e+00 -0.00000000e+00
  5.13265719e-03 -0.00000000e+00  0.00000000e+00 -0.00000000e+00
  0.00000000e+00 -0.00000000e+00 -0.00000000e+00 -0.00000000e+00
  0.00000000e+00]
Index(['Reviews_Duration', 'ProductRelated_Duration', 'PageValues'], dtype='object')
```

Variables seleccionadas:

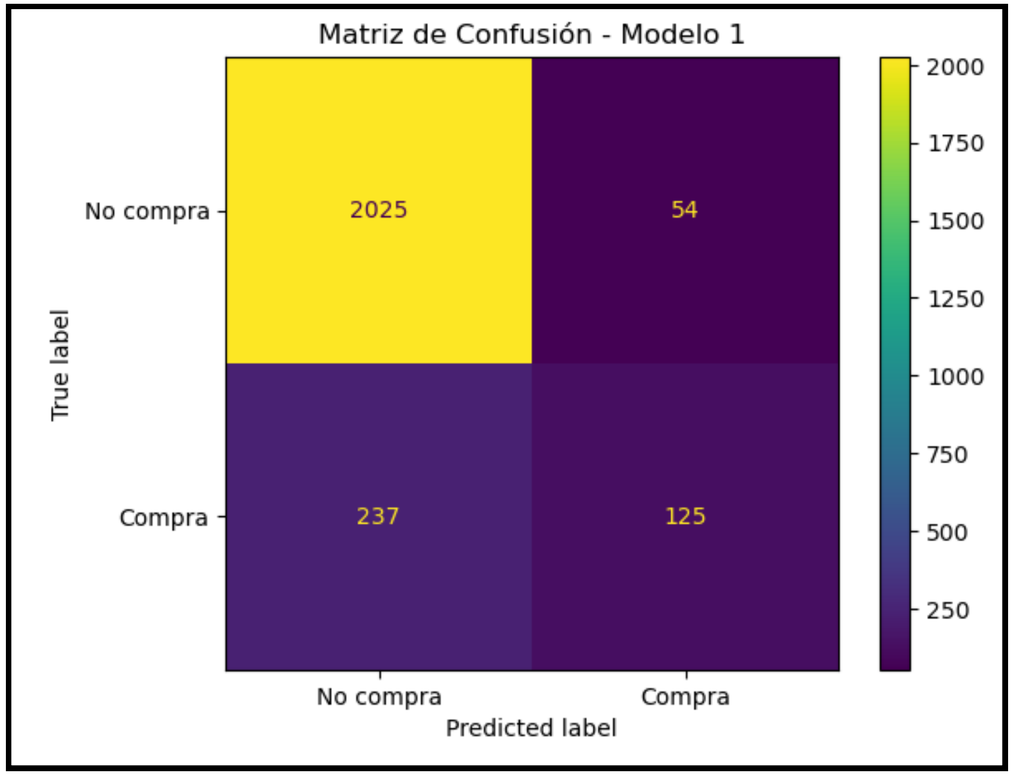
- Reviews_Duration: que es la cantidad de tiempo dedicado a esta categoría de páginas
- ProductRelated_Duration: es la cantidad de tiempo dedicado a esta categoría de páginas
- PageValues: Métrica arrojada por Google Analytics que representa el valor medio de una página web que un usuario visitó antes de completar una transacción de comercio electrónico

Son las más relevantes según el estimador Lasso y tienen un impacto significativo en la variable objetivo

Aplicación y comparación de técnicas de modelado

Modelo 1: Regresión Logística

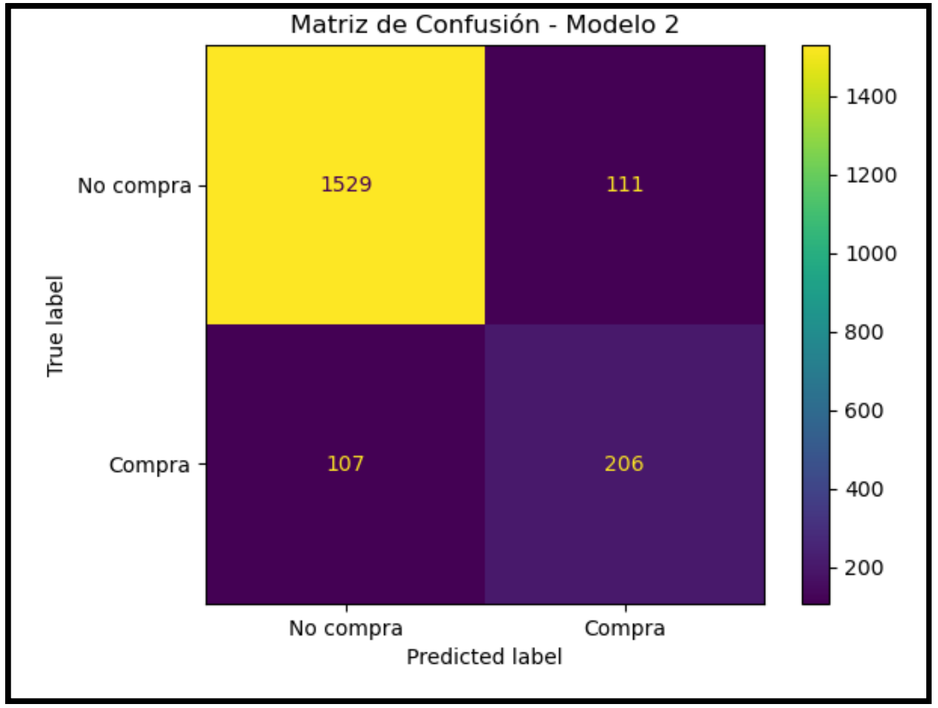
```
Desempeño en el conjunto de entrenamiento:
Accuracy: 0.8786358049979517
Classification Report:
              precision    recall  f1-score   support

           0       0.89      0.97      0.93      8218
           1       0.73      0.37      0.49      1546

    accuracy                           0.88      9764
   macro avg       0.81      0.67      0.71      9764
weighted avg       0.87      0.88      0.86      9764

Exactitud en la validación: 0.881

Desempeño en el conjunto de prueba:
Accuracy: 0.8807865628840639
Classification Report:
              precision    recall  f1-score   support

           0       0.90      0.97      0.93      2079
           1       0.70      0.35      0.46       362

    accuracy                           0.88      2441
   macro avg       0.80      0.66      0.70      2441
weighted avg       0.87      0.88      0.86      2441
```



Aplicación y comparación de técnicas de modelado

Modelo 2: DecisionTreeClassifier

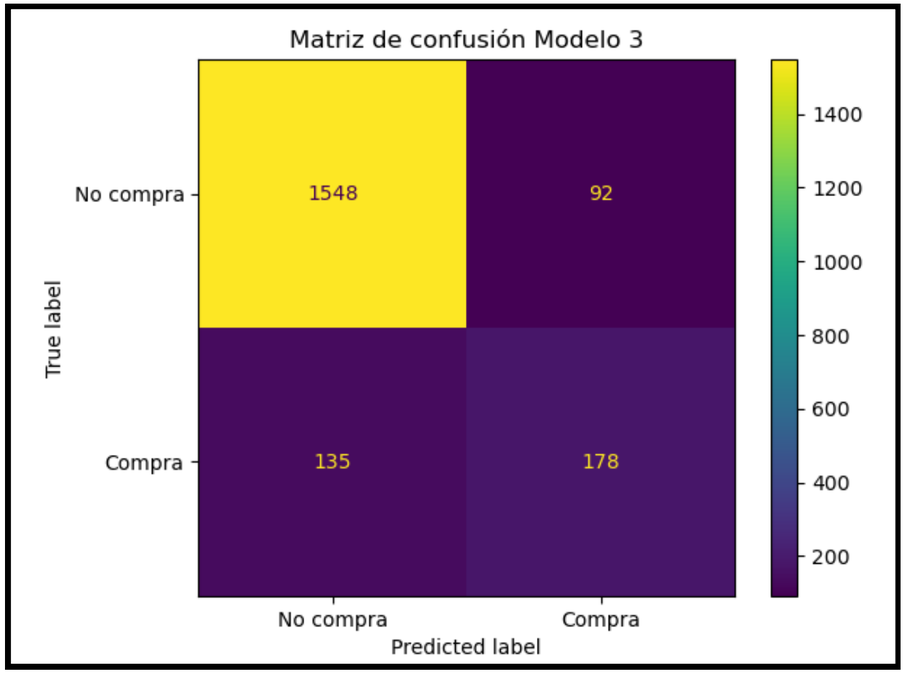
```
Train - Accuracy : 0.9021892203303035
Train - classification report:
              precision    recall  f1-score   support

           0     0.9452    0.9386    0.9419      6594
           1     0.6793    0.7050    0.6919      1217

    accuracy                         0.9022      7811
   macro avg     0.8123    0.8218    0.8169      7811
weighted avg     0.9038    0.9022    0.9029      7811

Test - Accuracy : 0.8883768561187916
Test - classification report :
              precision    recall  f1-score   support

           0     0.9346    0.9323    0.9335      1640
           1     0.6498    0.6581    0.6540       313

    accuracy                         0.8884      1953
   macro avg     0.7922    0.7952    0.7937      1953
weighted avg     0.8890    0.8884    0.8887      1953
```



Aplicación y comparación de técnicas de modelado

Modelo 3: GradientBoostingClassifier

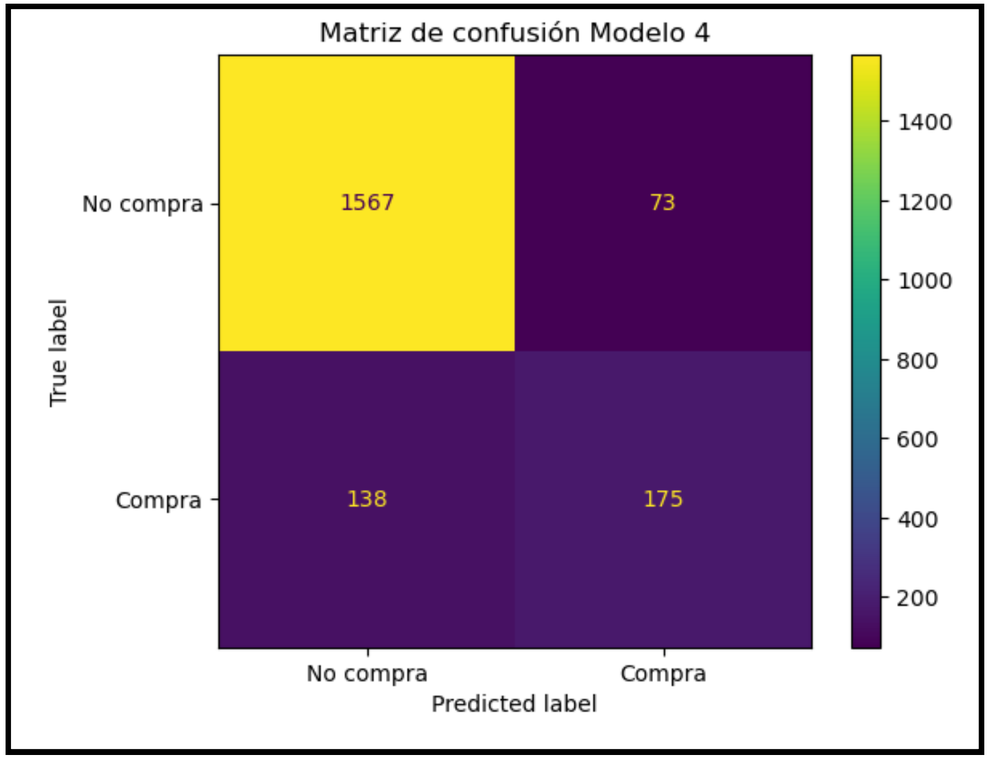
```
Desempeño en el conjunto de entrenamiento:
Accuracy: 0.9183203175009602
Classification Report:
              precision    recall  f1-score   support

           0       0.94      0.96      0.95      6594
           1       0.77      0.67      0.72      1217

    accuracy                           0.92      7811
   macro avg       0.86      0.82      0.84      7811
weighted avg       0.91      0.92      0.92      7811


Desempeño en el conjunto de prueba:
Accuracy: 0.883768561187916
Classification Report:
              precision    recall  f1-score   support

           0       0.92      0.94      0.93      1640
           1       0.66      0.57      0.61       313

    accuracy                           0.88      1953
   macro avg       0.79      0.76      0.77      1953
weighted avg       0.88      0.88      0.88      1953
```



Aplicación y comparación de técnicas de modelado

Modelo 4: Random Forest Classifier

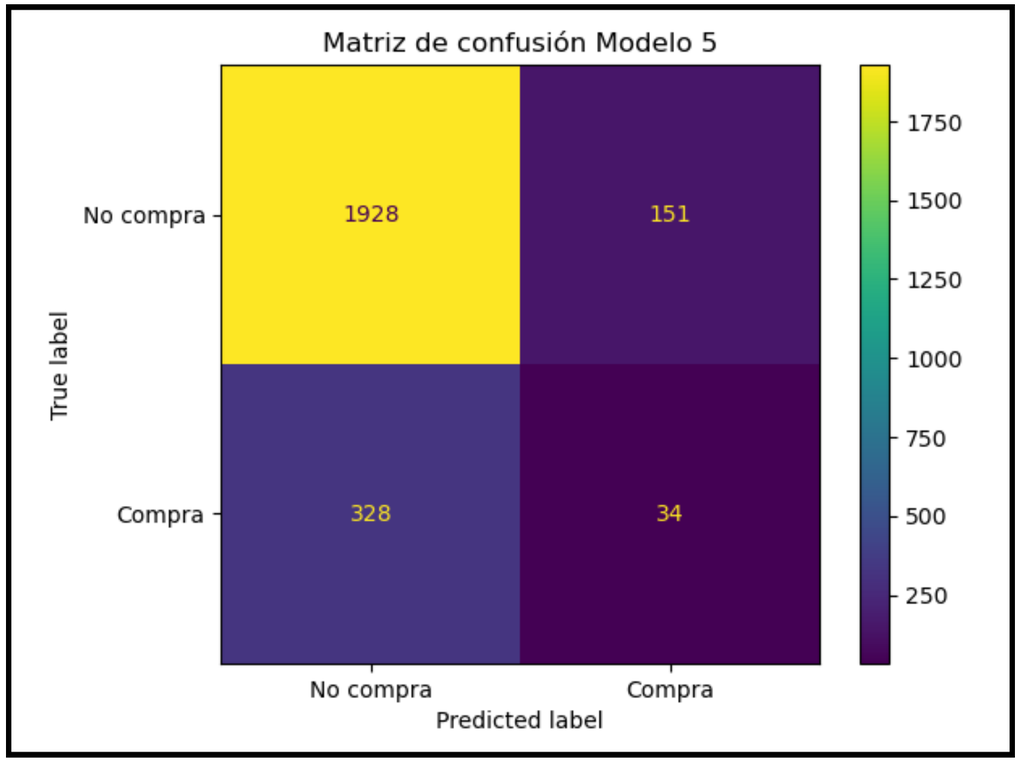
```
Desempeño en el conjunto de entrenamiento:
Accuracy: 0.9526309051337857
Classification Report:
              precision    recall  f1-score   support

           0       0.96      0.99      0.97      6594
           1       0.93      0.76      0.83      1217

    accuracy                           0.95      7811
   macro avg       0.94      0.87      0.90      7811
weighted avg       0.95      0.95      0.95      7811


Desempeño en el conjunto de prueba:
Accuracy: 0.8919610855094726
Classification Report:
              precision    recall  f1-score   support

           0       0.92      0.96      0.94      1640
           1       0.71      0.56      0.62       313

    accuracy                           0.89      1953
   macro avg       0.81      0.76      0.78      1953
weighted avg       0.88      0.89      0.89      1953
```



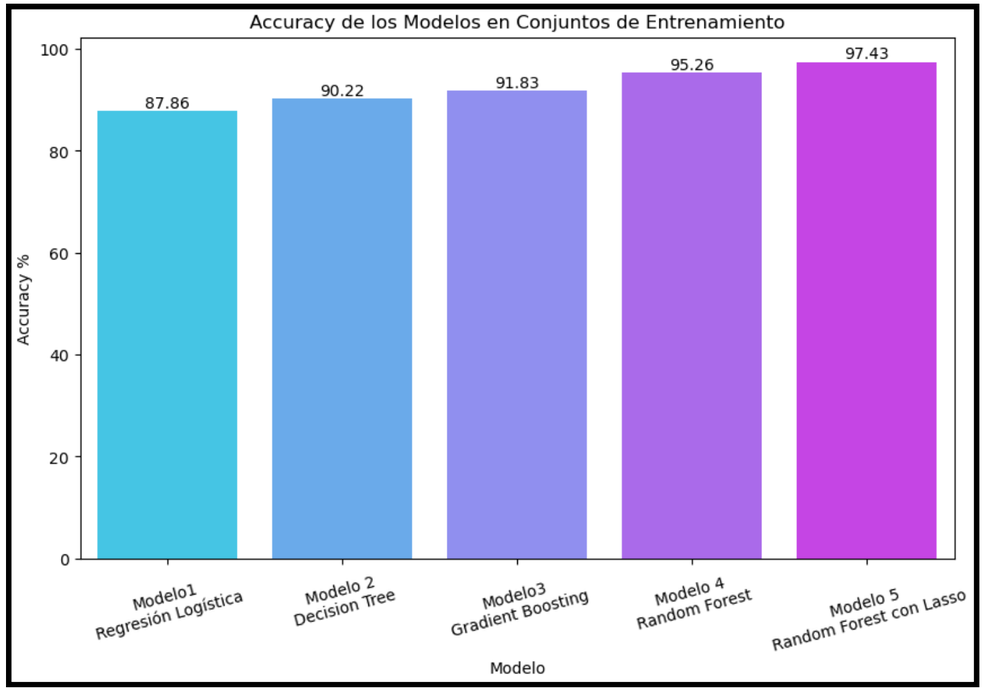
Aplicación y comparación de técnicas de modelado

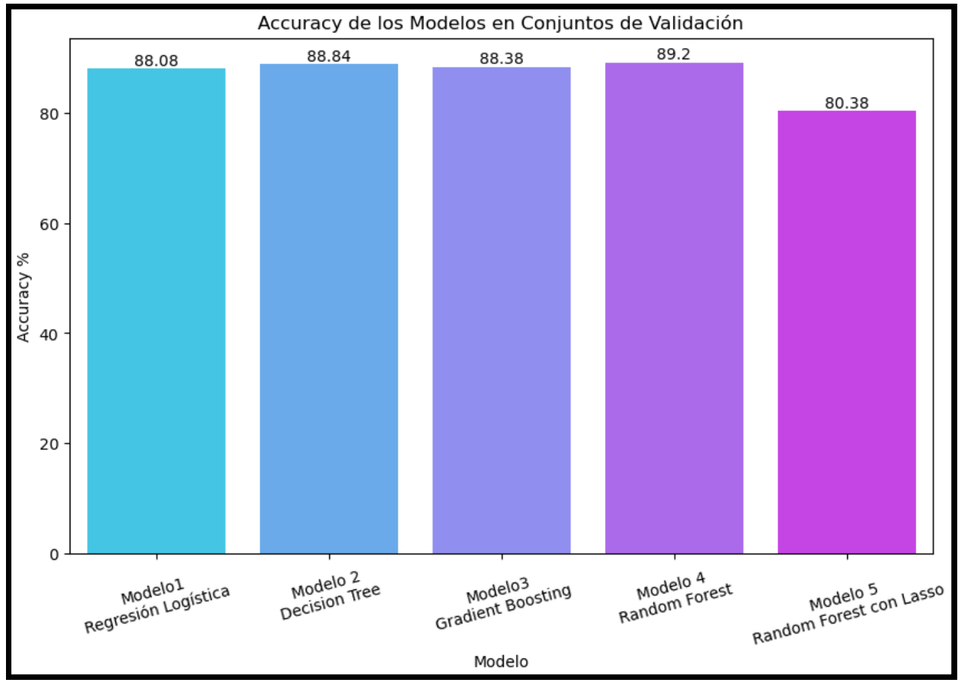
Modelo 5: Random Forest Classifier con estimador Lasso

```
Desempeño en el conjunto de entrenamiento:
Métricas de desempeño en el conjunto de entrenamiento seleccionado:
              precision    recall  f1-score   support

           0       0.97      1.00      0.98      8218
           1       0.98      0.85      0.91      1546

    accuracy                           0.97      9764
   macro avg       0.98      0.92      0.95      9764
weighted avg       0.97      0.97      0.97      9764

Accuracy en el conjunto de entrenamiento seleccionado: 0.9742933224088488

Métricas de desempeño en el conjunto de prueba seleccionado:
              precision    recall  f1-score   support

           0       0.85      0.93      0.89      2079
           1       0.18      0.09      0.12       362

    accuracy                           0.80      2441
   macro avg       0.52      0.51      0.51      2441
weighted avg       0.76      0.80      0.78      2441

Accuracy en el conjunto de prueba seleccionado: 0.8037689471528062
```



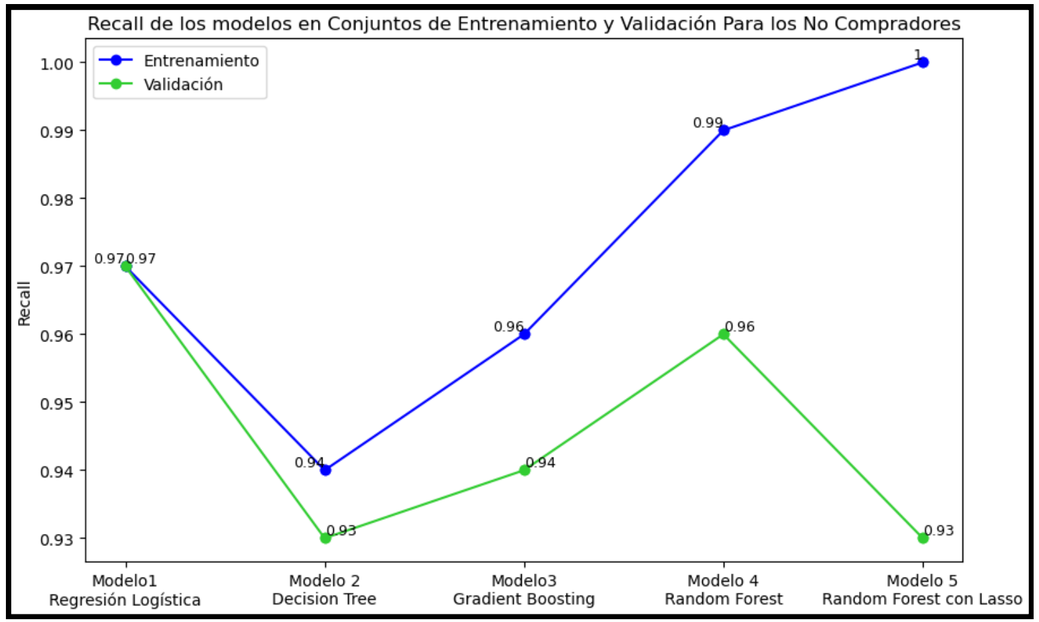
Evaluación del mejor modelo

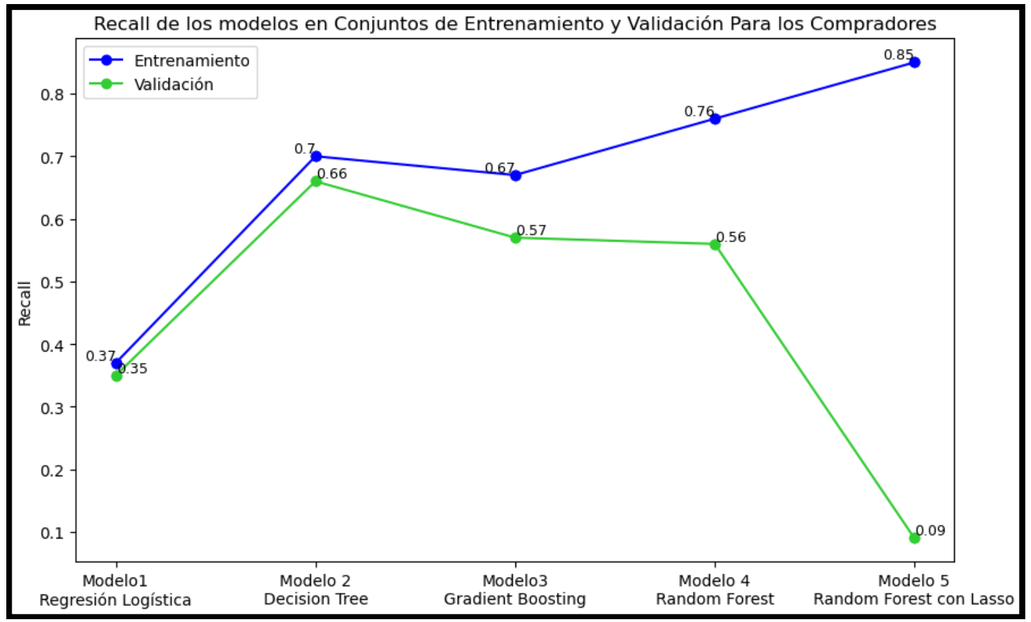




El modelo 4 y el 5 presentan la mayor precisión respecto a los otros modelos para el conjunto de entrenamiento y para el de validación los modelos 2 y 4.

Evaluación del mejor modelo

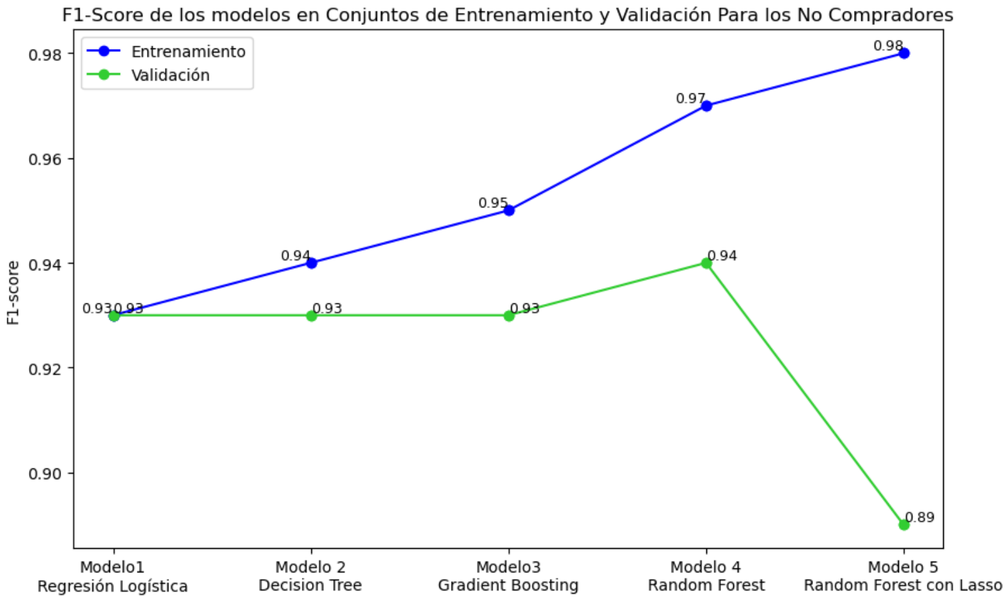


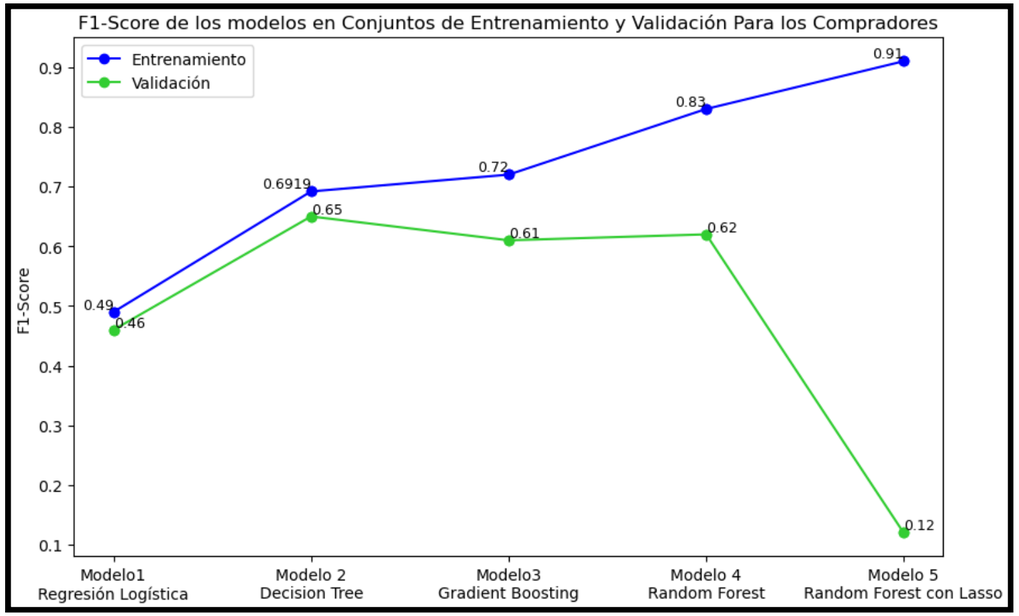


* Recall: Es la habilidad del clasificador para encontrar todas las instancias positivas.Para el modelo 2, el recall para la clase 0 es del 93.86% , y para la clase 1 es del 66% en validación. Para el modelo 4, el recall para la clase 0 es del 96% , y para la clase 1 es del 56% también en validación.

GRÁFICOS DE RECALL

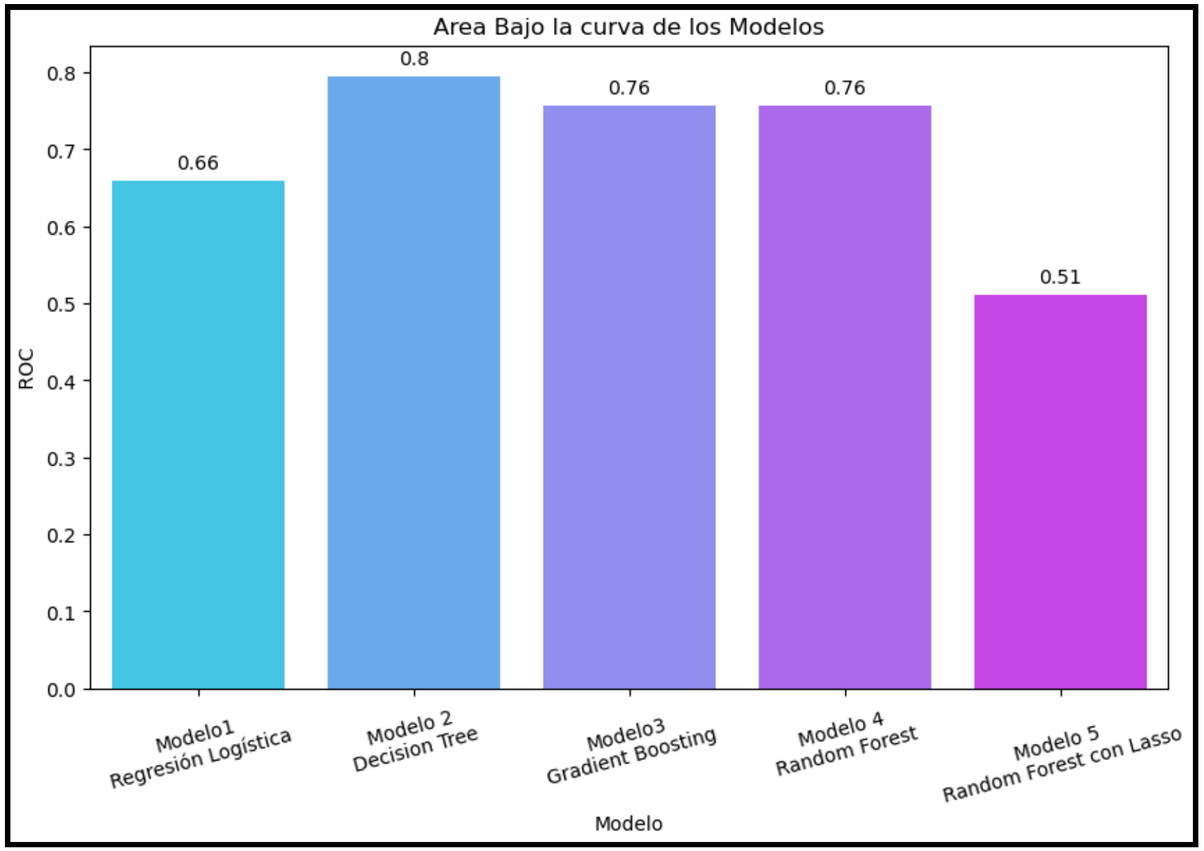
Evaluación del mejor modelo





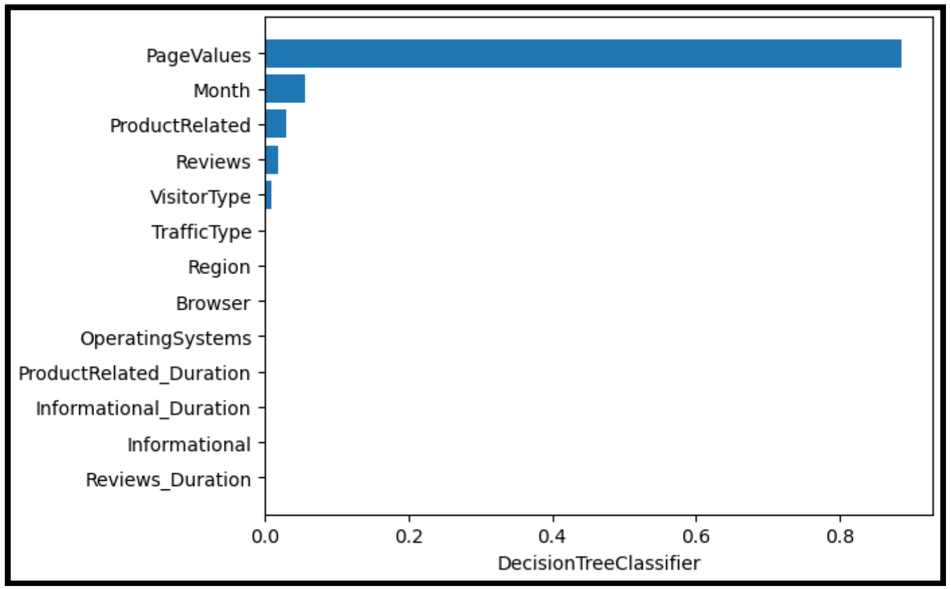
El F1-score es la media armónica de precision y recall. Es útil cuando hay una clase desbalanceada. En este caso para el modelo 2, el F1-score para la clase 0 es de 93%, y para la clase 1 es de 65% en la validación. Mientras que en el modelo 4 para la clase 0 es de 94% y para la clase 1 es de 62% también en los datos de validación.

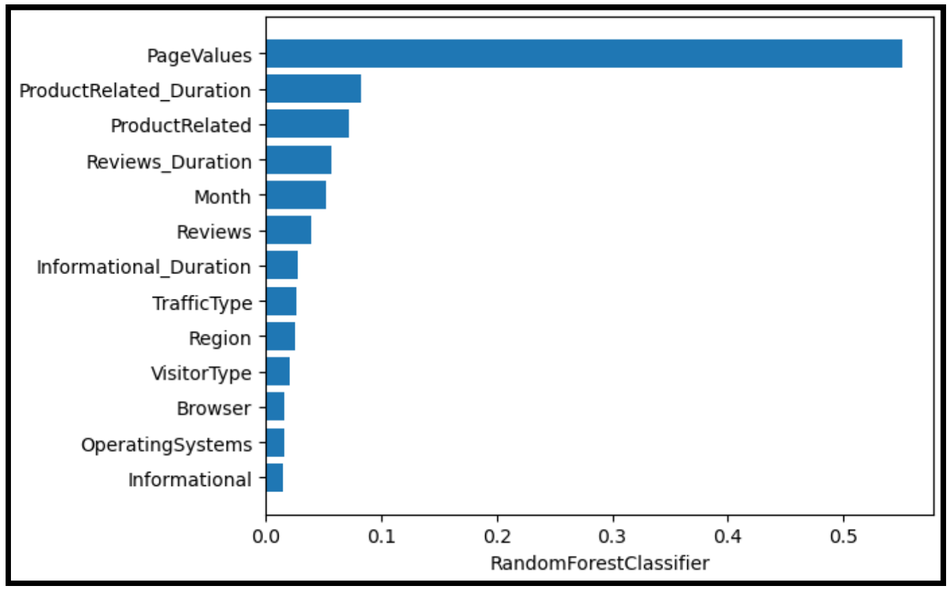
Evaluación del mejor modelo



Entre los modelos, el modelo 2 tiene el AUC más alto (0.8), lo que indica que tiene el mejor rendimiento general en términos de capacidad predictiva en comparación con los otros.

Evaluación del mejor modelo

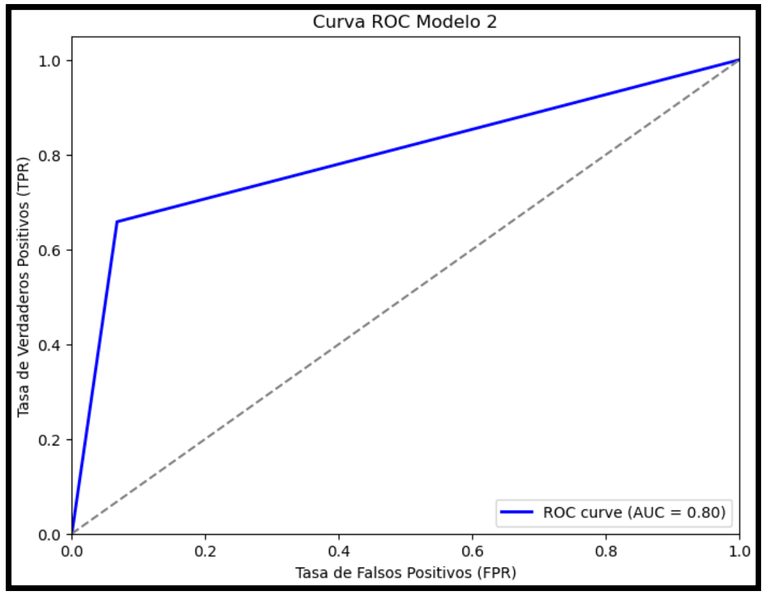




El modelo 2 y el 4 presentan diferencias en las características usadas

GRÁFICOS DE IMPORTANCIA
DE CARACTERISTICAS

Evaluación del mejor modelo



El Modelo 2 (DecisionTreeClassifier) se destaca como el mejor debido a su equilibrio entre una alta exactitud del 89% y la mayor AUC de 0.80 en el conjunto de prueba, lo que indica una fuerte capacidad de discriminación, especialmente valiosa en contextos de clasificación desbalanceada como es nuestro caso.

Conclusiones y recomendaciones

1. Estrategias de Marketing:

Basado en los resultados de los modelos, hay margen de mejora en las estrategias de marketing digital. Los datos sugieren que la nueva estrategia de marketing, implementando el modelo 2 podría ser más efectiva ya que tiene un enfoque en la personalización y la segmentación para captar nuevos clientes con alta intención de compra.

2. Utilización de Modelos de Machine Learning:

Los modelos de ML en este caso son valiosos para identificar clientes potenciales. El Modelo 2, en particular, es prometedor en la identificación de estos clientes, optimizando así la inversión en publicidad digital. Este modelo destaca por su capacidad clasificatoria y debería ser el foco de futuras inversiones y desarrollos.

Conclusiones y recomendaciones

3. Refinamiento del Modelo:

Aunque el Modelo 2 parece ser el más efectivo hasta ahora, presenta oportunidades de mejora. Se pueden explorar otras técnicas de ajuste de hiperparámetros y evaluar la implementación de algoritmos alternativos para maximizar la precisión y el retorno de la inversión.

4. Monitoreo Continuo:

El dinamismo del mercado requiere una adaptación constante dado que las preferencias de los clientes pueden cambiar con el tiempo, esto implicaría la toma de datos y nuevas variables que puedan incidir en la compra del cliente. Es crucial implementar un sistema de monitoreo que permita la actualización continua del modelo en respuesta a las nuevas tendencias de consumo y cambios en el entorno de mercado.