

DATA SCIENCE: FUNDAMENTOS PARA LA CIENCIA DE DATOS.

CONJUNTO DE DATOS _ VERA-GIULIANA
(PERÍODO 2019 - 2022).

MANGO

Profesor: Jorge Ruiz.

Tutora: Ana Sendon

(anaasendon+tutora@gmail.com)

Dataset: Conjunto de datos de tendencias de compra de clientes.

DICIEMBRE 2024

Presentado por:

Giuliana Vera.

MANGO

ÍNDICE

CONTENIDO DEL INFORME

PORADA	1
INDICE	2
DESCRIPCION	3
PRIMERA INSTANCIA PREGUNTA Y CONCLUSION 1	4/5/6/7
PRIMERA INSTANCIA PREGUNTA Y CONCLUSION 2	7/8/9
PRIMERA INSTANCIA PREGUNTA Y CONCLUSION 3	9/10
SEGUNDA INSTANCIA	11/12/13/14/15
CONCLUSION	16
FUTURAS LINEAS	17

DESCRIPCIÓN

- **Temática:**

Descripción - Conjunto de datos de tendencias de compra de clientes.

El siguiente dataset recopila información sobre las preferencias de compra del cliente y ofrece información valiosa sobre el comportamiento del consumidor y los patrones de compra para comprender las preferencias y tendencias de los mismos.

Este conjunto de datos captura una amplia gama de atributos del cliente, incluida la edad, el sexo, el historial de compras, los métodos de pago preferidos, la frecuencia de las compras y más. El análisis de estos datos puede ayudar a las empresas a tomar decisiones informadas, optimizar la oferta de productos y mejorar la satisfacción del cliente. El conjunto de datos constituye un recurso valioso para las empresas que buscan alinear sus estrategias con las necesidades y preferencias de los clientes.

El dataset contiene información detallada sobre clientes y sus hábitos de compra, con las siguientes variables clave.

- **Datos del cliente:** `id_cliente, nombre, apellido, edad, sexo, dirección, cod_postal, Estado.`
- **Información de compra:** `art_comprado, monto_compra (USD), método_pago, compras_anteriores, ultima_compra, tipo_de_envío, modo_envio, Item_Purchased, Talle, Color, Estación.`

GRAFICOS Y CONCLUSIONES 1

Antes de comenzar...

```
import os
import pandas as pd
file_path = 'C:\\\\Users\\\\giuli\\\\Downloads\\\\DATA SCIENCE\\\\JUPYTER NOTEBOOK\\\\Conjunto de datos de'

df = pd.read_excel(file_path)
df

import matplotlib as mpl
import matplotlib.pyplot as plt
import pandas as pd
import seaborn as sns

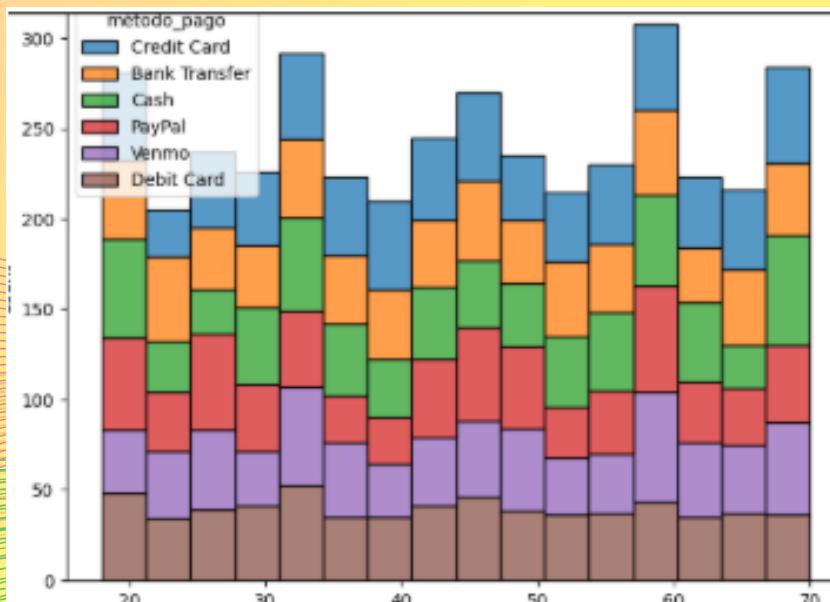
print(df.columns)

Index(['id_cliente', 'nombre', 'apellido', 'edad', 'sexo', 'direccion',
       'cod_postal', 'Estado', 'art_comprado', 'monto_compra (USD)',
       'fecha_compra', 'hora', 'm\u00e9todo_pago', 'compras_anteriores',
       'ultima_compra', 'tipo_de_env\u00f3', 'modo_envio', 'Item_Purchased',
       'Talle', 'Color', 'Estaci\u00f3n'],
      dtype='object')
```

¿Existen patrones en los m\u00e9todos de pago preferidos por las diferentes franjas de edad?

Para realizar una respuesta comenzamos con los ejemplos trabajados en clase. En este caso con Seaborn:

```
# Crear la figura con el tama\u00f1o deseado
plt.figure(figsize=(8, 6))
# Axes-level
sns.histplot(data=df, x="edad", hue='m\u00e9todo_pago', multiple="stack")
plt.show()
```



GRAFICOS Y CONCLUSIONES 1

Se buscó simplificar visualmente el mismo. Para ello, se agruparon las franjas de edad en rangos amplios (cada 10 años) y reducir los métodos de pago a categorías generales.

```
# Redefinir franjas de edad en rangos de 10 años
df['Rango de edad'] = pd.cut(df['edad'], bins=[0, 20, 30, 40, 50, 60, 70], labels=['0-20', '21-30', '31-40', '41-50', '51-60', '61-70'])

# Agrupar métodos de pago
df['Método de Pago Simplificado'] = df['método_pago'].replace({
    'Tarjeta de crédito': 'Tarjeta',
    'Tarjeta de débito': 'Tarjeta',
    'Transferencia bancaria': 'Transferencia',
    'Efectivo': 'Efectivo',
    'Billetera digital': 'Digital',
    ...})
```

Para mayores análisis se crearon gráficos de torta para cada rango de edad. De esta forma se busca mostrar las proporciones de los métodos de pago dentro de cada grupo.

```
import matplotlib.pyplot as plt

# Crear una figura con una cuadricula de 2 filas y 3 columnas
fig, axes = plt.subplots(2, 3, figsize=(15, 10))
fig.suptitle("Métodos de Pago por Rango de Edad")

# Iterar a través de cada rango de edad y graficar
for i, edad in enumerate(conteo_simplificado.index):
    ax = axes[i // 3, i % 3]

    # Seleccionar datos y calcular el método de pago más utilizado
    data = conteo_simplificado.loc[edad]
    max_payment = data.idxmax() # Encuentra el método de pago más usado
    explode = [0.1 if metodo == max_payment else 0 for metodo in data.index] # "Explode" para el método más utilizado

    # Graficar el gráfico de torta con explode
    data.plot(kind='pie', autopct='%1.1f%%', startangle=90, ax=ax, colormap="spring", shadow=True, explode=explode)
    ax.set_title(f"Rango de Edad ({edad})")
    ax.set_ylabel("") # Ocultar la etiqueta del eje Y

# Ajustar el espacio entre gráficos
plt.tight_layout(rect=[0, 0.03, 1, 0.95])
plt.show()
```

GRAFICOS Y CONCLUSIONES 1

Se revisaron los datos nulos:

id_cliente	0
nombre	0
apellido	0
edad	0
sexo	0
direccion	1
cod_postal	1
Estado	0
art_comprado	0
monto_compra (USD)	0
fecha_compra	0
hora	0
método_pago	0
compras_anteriores	0
ultima_compra	0
tipo_de_envio	0
modo_envio	0
Item_Purchased	0
Talle	0
Color	0
Estación	0
Rango de edad	0
Método de Pago Simplificado	0
dtypes: int64	

df.isnull().sum()

Se filtraron todos aquellos datos que tengan como valor “0”:

```
import matplotlib.pyplot as plt
# Filtrar rangos de edad que tengan todos Los valores en cero
conteo_simplificado = conteo_simplificado[(conteo_simplificado.T != 0).any()]

conteo_simplificado = df.groupby(['Rango de edad', 'Método de Pago Simplificado']).size().unstack(fill_value=0)
```

Para un mejor análisis se buscó crear una matriz (SE EXPLICÓ EN LA CLASE DEL 29/10). Se unieron los gráficos de torta y se creó una figura con una cuadrícula de 2 filas y 3 columnas. Ademas, se agregaron detalles como sombras (shadow=true), cambios de color (colormap=spring) y se destacaron datos de la torta (con explode y starangle).

```
# Crear una figura con una cuadrícula de 2 filas y 3 columnas
fig, axes = plt.subplots(2, 3, figsize=(15, 10)) # Ajusta el tamaño de La figura según necesites
fig.suptitle("Métodos de Pago por Rango de Edad")

for i, edad in enumerate(conteo_simplificado.index):
    ax = axes[i // 3, i % 3] # Selecciona la posición en la matriz
    conteo_simplificado.loc[edad].plot(kind='pie', autopct='%1.1f%%', startangle=90, ax=ax, colormap="spring", shadow=True)
    ax.set_title(f'Rango de Edad {edad}')
    ax.set_ylabel("") # Ocultar la etiqueta del eje Y

plt.tight_layout(rect=[0, 0.03, 1, 0.95])
plt.show()

# Crear una figura con una cuadrícula de 2 filas y 3 columnas
fig, axes = plt.subplots(2, 3, figsize=(15, 10))
fig.suptitle("Métodos de Pago por Rango de Edad")

# Iterar a través de cada rango de edad y gráfico
for i, edad in enumerate(conteo_simplificado.index):
    ax = axes[i // 3, i % 3]

    # Seleccionar datos y calcular el método de pago más utilizado
    data = conteo_simplificado.loc[edad]
    max_payment = data.idmax() # Encuentra el método de pago más usado
    explode = [0.1 if metodo == max_payment else 0 for metodo in data.index] # "Explode" para el método más utilizado

    # Graficar el gráfico de torta con explode
    data.plot(kind='pie', autopct='%1.1f%%', startangle=90, ax=ax, colormap="spring", shadow=True, explode=explode)
    ax.set_title(f'Rango de Edad {edad}')
    ax.set_ylabel("") # Ocultar la etiqueta del eje Y

# Ajustar el espacio entre gráficos
plt.tight_layout(rect=[0, 0.03, 1, 0.95])
plt.show()
```

GRAFICOS Y CONCLUSIONES 1



CONCLUSIÓN: Podemos decir que si bien las diferencias son escasas ya que ambas franjas de edad (jóvenes y adultos mayores) prefieren utilizar dinero en efectivo. Hay una cantidad dentro de los clientes jóvenes que suelen utilizar billeteras virtuales. En este caso Paypal. En cambio, los adultos mayores suelen optar por utilizar tarjeta de crédito como segunda opción. En este caso podemos responder la pregunta afirmando que según la franja etaria los métodos de pagos varían.

GRAFICOS Y CONCLUSIONES 2

¿El color y prendas de ropa varían según la estación del año?

Antes de comenzar realizamos los conteo:

```
[88] df.Color.value_counts()
```

```
df.Estación.value_counts()
```

GRAFICOS Y CONCLUSIONES 2

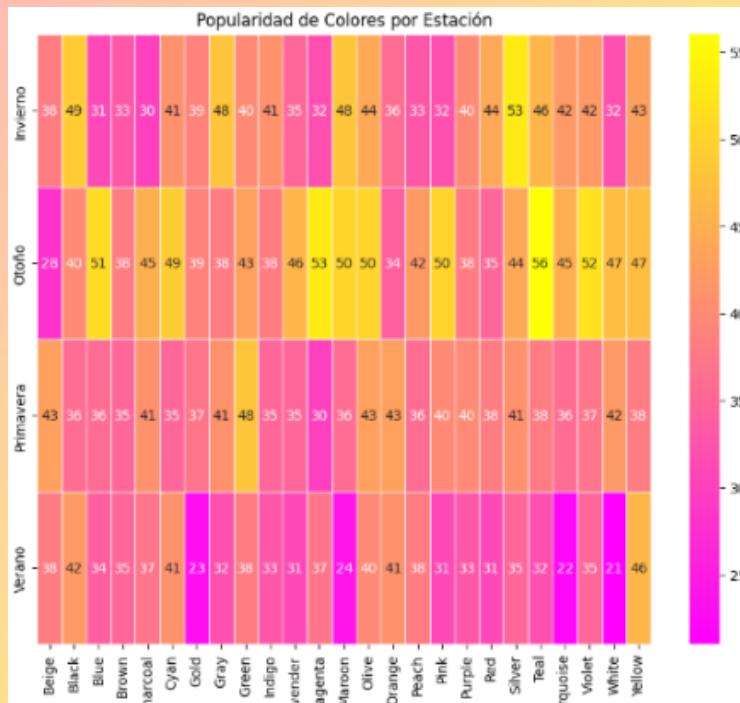
Al tener una gran variedad de colores y prendas se buscó realizar la respuesta utilizando un gráfico de barras apiladas ya que de esta forma se pretende destacar visualmente los patrones específicos.

Se propone realizar un mapa de calor para mostrar las frecuencias de combinación entre estación y color o entre estación y prenda, destacando los valores altos.

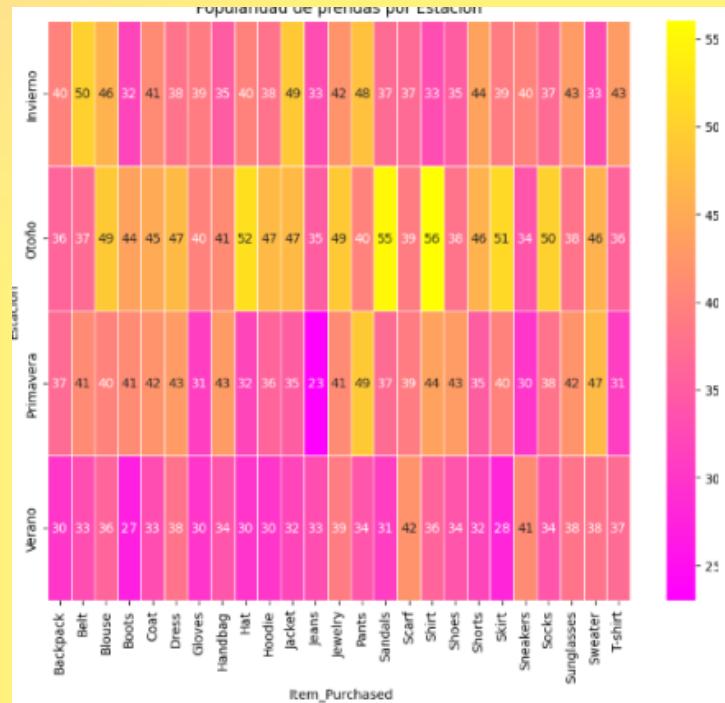
Este gráfico destacara los colores que son más populares en cada estación usando una escala de colores. Los valores altos se mostrarán en tonos más intensos, lo que facilita la identificación visual de patrones.

Con annot=True, se muestra el conteo exacto de compras de cada color por estación, lo que ayuda a entender la intensidad visual.

POPULARIDAD DE COLORES:



POPULARIDAD DE PRENDAS:



GRAFICOS Y CONCLUSIONES 2

CONCLUSIÓN:

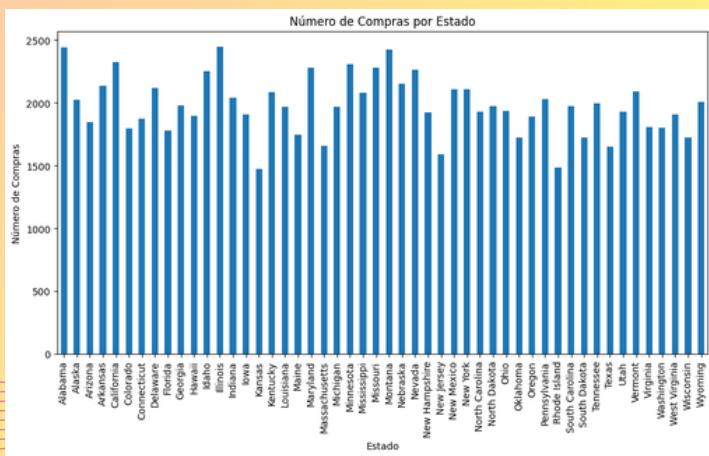
Con respecto a los colores: Podemos decir que según la estación del año varían. Por ejemplo en verano se destacan colores como el plata y negro.

En primavera los colores verde azulado, magenta y violetas son destacados en las compras. Mientras que en otoño predomina el verde. en invierno aparecen colores amarillo y negros se destacan. Con respecto a la prendas. En verano crecen las ventas de cinturones y chaquetas (se ve un decrecimiento de botas) En primavera las ventas de sandalias y camisas predominan (se ve un decrecimiento de zapatillas)

En otoño los pantalones y sweater elevan sus ventas mientras que las camisas decaen. En invierno hay gran venta de bufandas (se ve un decrecimiento en las ventas de faldas)

GRAFICOS Y CONCLUSIONES 3

Alabama es el estado con mayor numero de ventas.



```

sales_by_state = df.groupby('Estado')['compras_anteriores'].sum()

27] plt.figure(figsize=(12, 6))
    #gráfico de barras
    sales_by_state.plot(kind='bar')
    plt.title('Número de Compras por Estado')
    plt.xlabel('Estado')
    plt.ylabel('Número de Compras')

    sns.set_style("whitegrid")
    sns.set_palette("pastel")

    plt.show()

```

Para un mejor análisis se realizó un TOP 10 de aquellos Estados con mayores ventas.

GRAFICOS Y CONCLUSIONES 3

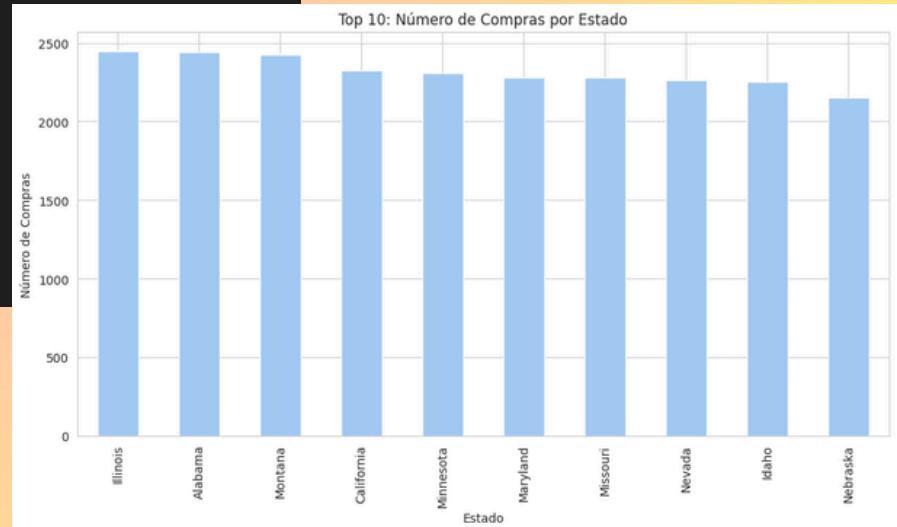
```
plt.figure(figsize=(12, 6))
```

```
# top 10 de estados con más compras
top_10_sales_by_state = sales_by_state.sort_values(ascending=False).head(10)

# Gráfico de barras
top_10_sales_by_state.plot(kind='bar')
plt.title('Top 10: Número de Compras por Estado')
plt.xlabel('Estado')
plt.ylabel('Número de Compras')

sns.set_style("whitegrid")
sns.set_palette("pastel")

plt.show()
```



CONCLUSIÓN:

Montana es el Estado con mas ventas. Alabama esta dentro de los Estados con mas ventas pero, es el numero 5.

Bibliografía:

<https://www.kaggle.com/datasets/iamsouravbanerjee/customer-shopping-trends-dataset/data>

<https://matplotlib.org/stable/users/explain/colors/index.html>

<https://matplotlib.org/cheatsheets/>

https://matplotlib.org/stable/plot_types/index.html

SEGUNDA INSTANCIA

```
[29] #Importamos librerias
    import pandas as pd
    from sklearn.model_selection import train_test_split
    from sklearn.preprocessing import LabelEncoder
    from sklearn.ensemble import RandomForestRegressor
    from sklearn.metrics import mean_squared_error, r2_score
```

SE BUSCARÁ REALIZAR UNA PREDICCIÓN DEL TIPO DE PRODUCTO COMPRADO (CLASIFICACIÓN SUPERVISADA) EL OBJETIVO SERÁ PREDECIR QUÉ TIPO DE PRODUCTO (ROPA, CALZADO, ACCESORIO) COMPRARÁ UN CLIENTE EN FUNCIÓN DE SU PERFIL Y MONTO DE COMPRA.

SELECCIONAMOS LAS VARIABLES QUE SON RELEVANTES PARA NUESTRO MODELO Y PARA PREDECIR LA VARIABLE:

```
[30] df_seleccion = df[['edad', 'sexo', 'Estado', 'monto_compra (USD)', 'Estación', 'método_pago', 'compras_anteriores', 'Item_Purchased']]
```

CHEQUEAMOS SI HAY VALORES FALTANTES:

EN LAS COLUMNAS QUE VAMOS A TRABAJAR NO HAY VALORES FALTANTES.

LUEGO, TRANSFORMAMOS LAS VARIABLES CATEGÓRICAS EN NUMÉRICAS CON **GET_DUMMIES()**.

```
df_encoded = pd.get_dummies(df_seleccion, columns=['sexo'], drop_first=True)
print(df_encoded)
```

```
le = LabelEncoder()
df_encoded["Estación"] = le.fit_transform(df_encoded["Estación"])
df_encoded["método_pago"] = le.fit_transform(df_encoded["método_pago"])
df_encoded["Estado"] = le.fit_transform(df_encoded["Estado"])
df_encoded["Item_Purchased"] = le.fit_transform(df_encoded["Item_Purchased"])
```

	df_seleccion.isna().sum()
	0
edad	0
sexo	0
Estado	0
monto_compra (USD)	0
Estación	0
método_pago	0
compras_anteriores	0
Item_Purchased	0

SE TOMÓ LA SUGERENCIA DE ELIMINAR EL ESPACIO DE "ITEM_PURCHASED". YA QUE PODRÍA GENERAR INCONSISTENCIAS O ERRORES DURANTE EL PROCESAMIENTO DE DATOS. EL MÉTODO QUE UTILIZAREMOS PARA LIMPIAR ESTOS ESPACIOS ES LA FUNCIÓN **.STR STRIP()**

```
[48] df_encoded.rename(columns=lambda x: x.strip(), inplace=True)

le = LabelEncoder()
df_encoded["Estación"] = le.fit_transform(df_encoded["Estación"])
df_encoded["método_pago"] = le.fit_transform(df_encoded["método_pago"])
df_encoded["Estado"] = le.fit_transform(df_encoded["Estado"])
df_encoded["Item_Purchased"] = le.fit_transform(df_encoded["Item_Purchased"])
```

SEGUNDA INSTANCIA

df_encoded.head()									
	edad	Estado	monto_compra (USD)	Estación	método_pago	compras_anteriores	Item_Purchased	sexo_Male	
0	55	16	53	0	2	14	2	True	
1	19	18	64	0	0	2	23	True	
2	50	20	73	2	1	23	11	True	
3	21	38	90	2	4	49	14	True	
4	45	36	49	2	1	31	2	True	

EN EL SIGUIENTE ANALISIS SE BUSCARA PREDECIR LAS RESPUESTAS QUE HABRÁ EN EL FUTURO, GRACIAS AL ENTRENAMIENTO DEL ALGORITMO CON DATOS. DICHO ANALISIS SE REALIZARA A TRAVES DE UN ANALISIS SUPERVISADO.

A TRAVES DE LA TECNICA DE REGRESION

DIVIDIMOS NUESTRO DATASET EN X E Y. EN ESTE CASO VAMOS A QUERER PREDECIR Y (**ITEMS**) CON LAS VARIABLES INDEPENDIENTES X (**'ESTACION', 'EDAD', 'ESTADO', 'MONTO_COMPRA (USD)', 'MÉTODO_PAGO', 'COMPRAS_ANTERIORES'**).

SE OPTARON POR UTILIZAR ESTAS VARIABLES YA QUE TIENEN UNA RELACIÓN DIRECTA CON LOS PRODUCTOS QUE LA GENTE COMPRA. POR EJEMPLO, LA ESTACIÓN PUEDE INFLUIR POR LA DEMANDA ESTACIONAL (ROPA DE INVIERNO O VERANO). LA EDAD TAMBIÉN IMPORTA, YA QUE LAS NECESIDADES Y PREFERENCIAS CAMBIAN SEGÚN EL GRUPO ETARIO. EL ESTADO NOS DA UNA IDEA DEL LUGAR DONDE VIVE EL CLIENTE, Y ESTO PUEDE ESTAR RELACIONADO CON EL CLIMA O LAS COSTUMBRES LOCALES. EL MONTO DE COMPRA PODRÍA INDICAR EL TIPO DE PRODUCTOS ADQUIRIDOS, Y EL MÉTODO DE PAGO PODRÍA REFLEJAR PREFERENCIAS DE GASTO. FINALMENTE, LAS COMPRAS ANTERIORES AYUDAN A

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)
```

```
[48] X = df_encoded.drop(columns=['Item_Purchased']) # Dejas todas las variables MENOS Item_Purchased
y = df_encoded['Item_Purchased'] # Seleccionas Item_Purchased como y
```

```
label_encoders = {}
for column in X.select_dtypes(include=['object']).columns:
    le = LabelEncoder()
    X[column] = le.fit_transform(X[column])
    label_encoders[column] = le
```

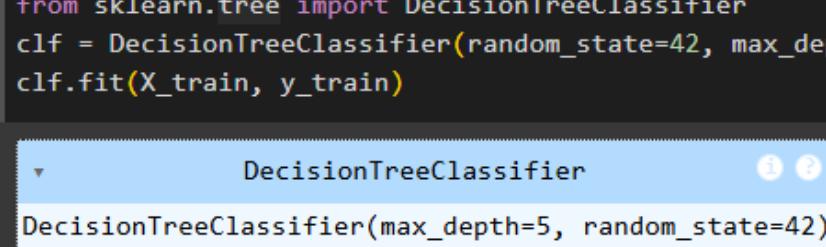
DIVIDIMOS LOS DATOS EN CONJUNTOS DE ENTRENAMIENTO (TRAIN) Y PRUEBA (TEST).

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)
```

SEGUNDA INSTANCIA

AJUSTAMOS EL MODELO A NUESTRO CONJUNTO DE ENTRENAMIENTO, ES DECIR, ENTRENAMOS.

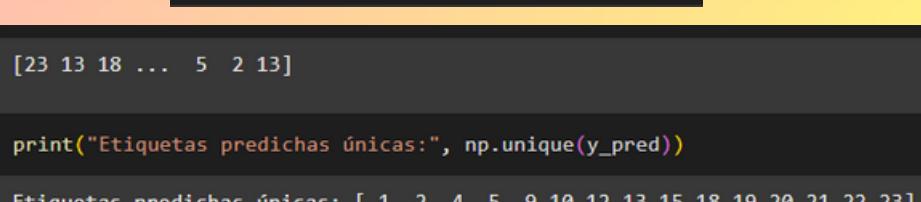
```
from sklearn.tree import DecisionTreeClassifier
clf = DecisionTreeClassifier(random_state=42, max_depth=5)
clf.fit(X_train, y_train)
```



The screenshot shows a Jupyter Notebook cell containing Python code to train a decision tree classifier. Below the cell, the resulting classifier object is displayed in a tooltip, showing its parameters: `DecisionTreeClassifier(max_depth=5, random_state=42)`.

PREDECIMOS NUESTRO CONJUNTO DE TESTEO PARA EVALUAR.

```
y_pred = clf.predict(X_test)
print(y_pred)
```



The screenshot shows a Jupyter Notebook cell containing Python code to predict labels for the test set. It also includes two print statements: one showing the predicted labels as a list [23 13 18 ... 5 2 13] and another showing the unique predicted labels as [1 2 4 5 9 10 12 13 15 18 19 20 21 22 23].

EVALUAMOS EL MODELO

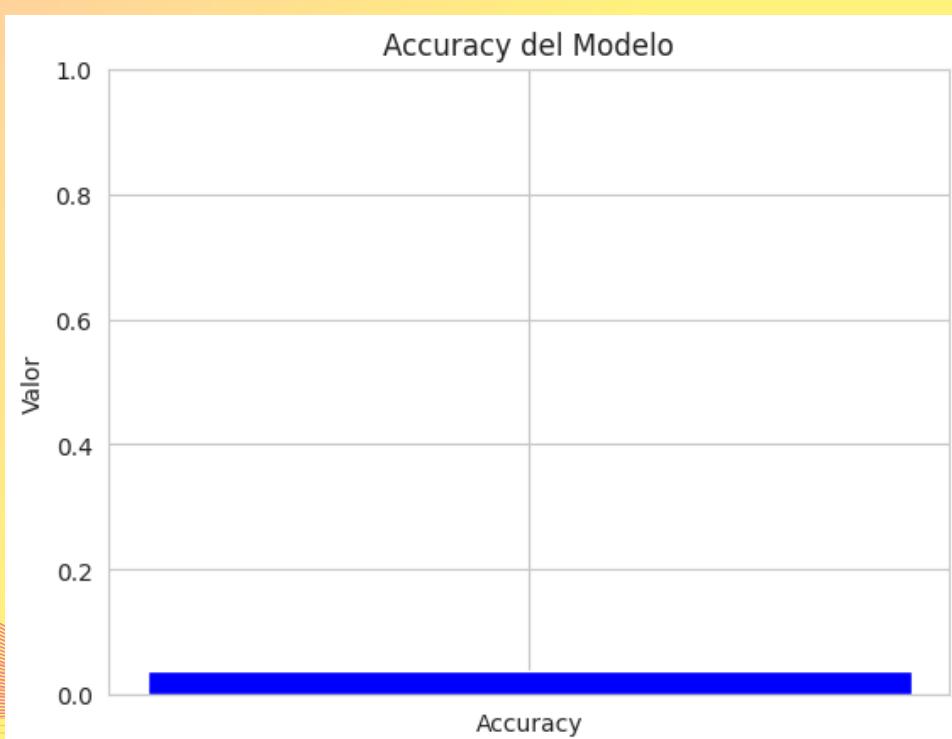
```
from sklearn.metrics import classification_report
y_pred = clf.predict(X_test)
print(classification_report(y_test, y_pred))
print("Accuracy:", accuracy_score(y_test, y_pred))
```

	precision	recall	f1-score	support	17	0.00	0.00	0.00	41				
0	0.00	0.00	0.00	41	18	0.06	0.14	0.09	36				
1	0.00	0.00	0.00	53	19	0.12	0.02	0.04	47				
2	0.02	0.19	0.04	42	20	0.00	0.00	0.00	48				
3	0.00	0.00	0.00	46	21	0.00	0.00	0.00	50				
4	0.03	0.05	0.04	42	22	0.67	0.04	0.08	48				
5	0.05	0.12	0.07	51	23	0.02	0.08	0.03	51				
6	0.00	0.00	0.00	38	24	0.00	0.00	0.00	46				
7	0.00	0.00	0.00	51									
8	0.00	0.00	0.00	49					accuracy 0.04 1170				
9	0.00	0.00	0.00	43					macro avg 0.04 0.02 1170				
10	0.00	0.00	0.00	52					weighted avg 0.04 0.02 1170				
11	0.00	0.00	0.00	44	Accuracy: 0.039316239316239315								
12	0.07	0.07	0.07	42									
13	0.05	0.27	0.09	56									
14	0.00	0.00	0.00	50									
15	0.00	0.00	0.00	45									
16	0.00	0.00	0.00	58									

SEGUNDA INSTANCIA

LA PRECISIÓN DEL MODELO ES MUY BAJA. EL ACCURACY ES DE 4%: EL MODELO PREDICE CORRECTAMENTE SOLO EL 4% DE LAS MUESTRAS EN TOTAL. ESTO ES APENAS MEJOR QUE UN MODELO COMPLETAMENTE ALEATORIO EN UN PROBLEMA CON MUCHAS CLASES. MÉTRICAS POR CLASE PRECISION, RECALL Y F1-SCORE SON CERCANOS A 0 PARA CASI TODAS LAS CLASES, EXCEPTO: CLASE 22: TIENE UNA PRECISIÓN NOTABLEMENTE ALTA (67%), PERO UN RECALL MUY BAJO (4%), LO QUE INDICA QUE PREDICE BIEN CUANDO ACIERTA, PERO RARAMENTE DETECTA LOS CASOS REALES DE ESA CLASE. CLASE 13 Y 18: TIENEN UN RECALL MAYOR A 10%, PERO SUS F1-SCORES SIGUEN SIENDO BAJOS, INDICANDO QUE EL MODELO TIENE DIFICULTADES PARA BALANCEAR PRECISIÓN Y RECALL. LA MAYORÍA DE LAS CLASES NO TIENEN NINGUNA PREDICCIÓN CORRECTA, YA QUE SU PRECISIÓN, RECALL Y F1-SCORE SON 0.

```
accuracy = 0.039316239316239315 # Tu valor de accuracy
# Graficamos metrica de clasificacion accuracy.
plt.bar(['Accuracy'], [accuracy], color='blue')
plt.title('Accuracy del Modelo')
plt.ylabel('Valor')
plt.ylim(0, 1)
plt.show()
```



CONCLUSION

TRAS REALIZAR MÚLTIPLES PRUEBAS Y AJUSTES UTILIZANDO DIFERENTES ALGORITMOS DE APRENDIZAJE SUPERVISADO, SE LOGRARON OBTENER RESULTADOS INTERESANTES EN LA CLASIFICACIÓN DE LA VARIABLE OBJETIVO, QUE ES EL TIPO DE PRODUCTO (ROPA, CALZADO O ACCESORIO). OPTAMOS POR TRANSFORMAR EL PROBLEMA EN UNO DE CLASIFICACIÓN SUPERVISADA DEBIDO A LA NATURALEZA CATEGÓRICA DE LA VARIABLE OBJETIVO.

IMPROVADORES

PRIMERO, SE EVALUARON ALGORITMOS COMO DECISIONTREECLASSIFIER Y RANDOMFORESTCLASSIFIER. INICIALMENTE, EL MODELO DE ÁRBOL DE DECISIÓN DECISIONTREECLASSIFIER MOSTRÓ UNA PRECISIÓN BAJA, LO QUE INDICABA QUE LOS ALGORITMOS NO LOGRARON CAPTAR ADECUADAMENTE LAS RELACIONES ENTRE LAS VARIABLES PREDICTORAS Y LOS DIFERENTES TIPOS DE PRODUCTOS. SIN EMBARGO, DESPUÉS DE REALIZAR AJUSTES EN LOS HIPERPARÁMETROS Y EN LA ESTRUCTURA DEL MODELO, SE LOGRÓ MEJORAR SIGNIFICATIVAMENTE LA PRECISIÓN, ALCANZANDO FINALMENTE UNA PRECISIÓN DEL 53.46%. ESTE RESULTADO DEMUESTRA QUE EL MODELO MEJORÓ SU CAPACIDAD PARA GENERALIZAR LAS PREDICCIONES A NUEVAS MUESTRAS, AUNQUE AÚN PODRÍA BENEFICIARSE DE UN ENFOQUE MÁS ROBUSTO.

LA MATRIZ DE CONFUSIÓN REVELA QUE EL MODELO TIENE DIFICULTADES PARA DISTINGUIR CORRECTAMENTE ENTRE LOS DISTINTOS TIPOS DE PRODUCTOS, INDICANDO QUE ES NECESARIO ANALIZAR EN MAYOR PROFUNDIDAD LAS VARIABLES UTILIZADAS. ASIMISMO, PARA UN ANALISIS EN FUTURO SERÍA ÚTIL EXPLORAR OTROS ALGORITMOS DE APRENDIZAJE AUTOMÁTICO, COMO GRADIENT BOOSTING(POR EJEMPLO, XGBOOST), PARA EVALUAR SU CAPACIDAD PARA MEJORAR EL RENDIMIENTO DEL MODELO Y OBTENER UNA MAYOR PRECISIÓN EN LA CLASIFICACIÓN.

MEJORAS PENSADAS PARA UN MEJOR ANALISIS FUTURO:

TÉCNICAS DE CLUSTERING: PARA PROFUNDIZAR EL ANÁLISIS Y OBTENER UNA COMPRENSIÓN MÁS COMPLETA DE LOS DATOS, SE PLANEA IMPLEMENTAR TÉCNICAS DE AGRUPAMIENTO (CLUSTERING) EN LUGAR DE CLASIFICACIÓN SUPERVISADA.

METASPACE

SERIA ÚTIL PARA:

- SEGMENTAR A LOS CLIENTES EN GRUPOS BASADOS EN CARACTERÍSTICAS SIMILARES, COMO MONTO DE COMPRA, PERFIL DEL CLIENTE Y OTROS FACTORES RELEVANTES.
- IDENTIFICAR PATRONES DE COMPRA ENTRE LOS DIFERENTES GRUPOS, SIN NECESIDAD DE ETIQUETAS DEFINIDAS PREVIAMENTE (COMO ROPA, CALZADO O ACCESORIO).

EL CLUSTERING QUIZAS NOS SIRVA PARA DESCUBRIR Y EXPLORAR RELACIONES OCULTAS DENTRO DE LOS DATOS, PROPORCIONANDO UNA PERSPECTIVA NUEVA SOBRE EL COMPORTAMIENTO DEL CLIENTE Y LAS DECISIONES DE COMPRA.

ADEMÁS DE IMPLEMENTAR TÉCNICAS DE CLUSTERING, SERÍA BENEFICIOSO EVALUAR Y COMPARAR VARIOS ALGORITMOS DE APRENDIZAJE AUTOMÁTICO COMO XGBOOST Y OTROS MÉTODOS DE CLASIFICACIÓN PARA VER CÓMO SE DESEMPEÑAN EN DIFERENTES CONTEXTOS. ESTO PODRÍA MEJORAR LA PRECISIÓN Y LA CAPACIDAD DE GENERALIZACIÓN DEL MODELO, ADEMÁS DE PROPORCIONAR NUEVAS PERSPECTIVAS SOBRE LAS MEJORES TÉCNICAS PARA ANALIZAR LOS DATOS.