

Proyecto de analisis de ventas en una tienda de ropa

Recopilación y limpieza de datos: Se recolectaron datos de una pequeña Tienda de Ropa en la ciudad de Pasto, los datos de ventas desde el mes de mayo 2022 hasta diciembre 2022, incluyendo información sobre fechas, productos, precios, etc. Estos datos se limpiarán para eliminar cualquier información faltante o incorrecta.

Análisis Exploratorio de datos: Se analizarán los datos recolectados para identificar patrones y tendencias en las ventas, principalmente con ventas totales por mes y colores vendidos por mes. Se crearán gráficos y tablas para visualizar esta información.

Predicción: Con los datos historicos de las ventas se hará una proyección de valor de ventas por mes, esto con un algoritmo de machine learning de power bi, así también se hará una proyección de la cantidad de prendas vendidas por mes.

Carga de Datos

```
In [28]: import pandas as pd

# Cargar el archivo csv en un DataFrame
data = pd.read_csv('Datos/VentasTiendaA.csv')

# Imprimir las primeras filas del DataFrame
data.head()
```

```
Out[28]:
```

	fecha_venta	hora_venta	Producto	Valor_sugerido	venta_final	venta_inventario	valor_compra	c
0	2022-05-31	1330.0	Gaban Negro Totas	89000.0	82000	NaN	NaN	
1	2022-05-31	1330.0	Blusa rosa Tulcan	66000.0	58000	NaN	NaN	
2	2022-06-01	1600.0	Blusa tejida caracol	59000.0	50000	NaN	NaN	
3	2022-06-02	1000.0	Gaban Murcielago	72000.0	60000	NaN	NaN	
4	2022-06-06	1400.0	Chaqueta Totas	93000.0	72500	NaN	NaN	

Consultas Exploratorias

Aquí se utiliza el atributo shape de un DataFrame de Pandas para obtener la cantidad de filas y columnas. El primer elemento de la tupla devuelta es el número de filas y el segundo elemento es el número de columnas.

¿Cuántas filas tiene el DataFrame?

```
In [29]: num_rows = data.shape[0]
         print(num_rows)
```

396

¿Cuántos atributos tiene el DataFrame?

```
In [30]: num_columns = data.shape[1]
         print(num_columns)
```

16

¿Cuales son los nombres de los 16 atributos?

Con el atributo columns de un DataFrame de Pandas se puede obtener una lista con los nombres de las columnas.

```
In [31]: col_names = data.columns.tolist()
         print(col_names)
```

['fecha_venta', 'hora_venta', 'Producto', 'Valor_sugerido', 'venta_final', 'venta_inventario', 'valor_compra', 'contribucion', 'Comprador', 'fecha_ingreso_inventario', 'NOMBRE', 'sexo_producto', 'color_prenda', 'talla_prenda', 'proveedor', 'OBSERVACIONES']

¿Cuál es la fecha de la primera y última venta?

Para validar el rango de fecha en que se encuentran los datos recolectados se usará min() y max() de la columna de 'fecha_venta' en el DataFrame de Pandas.

```
In [32]: first_sale_date = data['fecha_venta'].min()
         print(first_sale_date)

         last_sale_date = data['fecha_venta'].max()
         print(last_sale_date)
```

2022-05-23
2022-12-31

Se tuvo en cuenta que la columna fecha_venta debe ser de tipo datetime para poder utilizar min() y max().

También se puede asegurar de que tenga un formato valido utilizando previamente pd.to_datetime([]) así como se realizará en la siguiente sección.

¿Cuál es el valor total de las ventas realizadas por cada mes?

Se crea la columna de 'month' que contiene solo el numero del mes de la columna 'fecha_venta' y con el método groupby() del DataFrame se agrupan las ventas por mes y luego se utiliza sum() para obtener el valor total de las ventas por mes. En este caso, se devolverá una serie con un índice correspondiente al numero de cada mes y el valor siendo el total de las ventas para cada mes.

```
In [33]: data['fecha_venta'] = pd.to_datetime(data['fecha_venta'])
data['month'] = data['fecha_venta'].dt.month
sales_by_month = data.groupby(['month'])['venta_final'].sum()
print(sales_by_month)
```

```
month
5      200000
6     2869000
7     2415000
8     1977000
9     2300000
10    1999000
11    2828000
12    7344000
Name: venta_final, dtype: int64
```

¿Cuál es el mes de mayor ventas y su valor?

Se utiliza el método idxmax() para obtener el índice del mes con la mayor venta y su valor, después de haber agrupado las ventas por mes y sumado el valor de ventas para cada mes.

```
In [34]: max_sales_month = sales_by_month.idxmax()
print(max_sales_month)
print("Total de Ventas en el mes de mayor venta: ", sales_by_month.loc[max_sales_month])
```

```
12
Total de Ventas en el mes de mayor venta: 7344000
```

¿Cuál es el mes de menor ventas y su valor?

Y se utiliza idxmin() para obtener el índice del mes con menor venta.

```
In [35]: min_sales_month = sales_by_month.idxmin()
print(min_sales_month)
print("Total de Ventas en el mes de menor venta: ", sales_by_month.loc[min_sales_month])
```

```
5
Total de Ventas en el mes de menor venta: 200000
```

¿Cual es el promedio de ventas por mes?

Con el método mean() de una serie se obtiene el promedio de las ventas de los 8 meses.

```
In [36]: average_sales = sales_by_month.mean()
print(average_sales)
```

```
2741500.0
```

El método mean() devuelve el valor promedio de los valores disponibles en el DataFrame. Para calcular el promedio solo para los 7 meses específicos (los cuales son los más representativos

para este caso), se utiliza `.loc[]` antes de calcular el promedio.

```
In [37]: average_sales = sales_by_month.loc[[6, 7, 8, 9, 10, 11, 12]].mean()
print(average_sales)

3104571.4285714286
```

Redondear el valor

`round()` para redondear el valor promedio resultante a un número específico de decimales. El siguiente código redondeará el valor promedio a 2 decimales:

```
In [38]: rounded_average = round(average_sales, 2)
print(rounded_average)

3104571.43
```

Gráfica #1

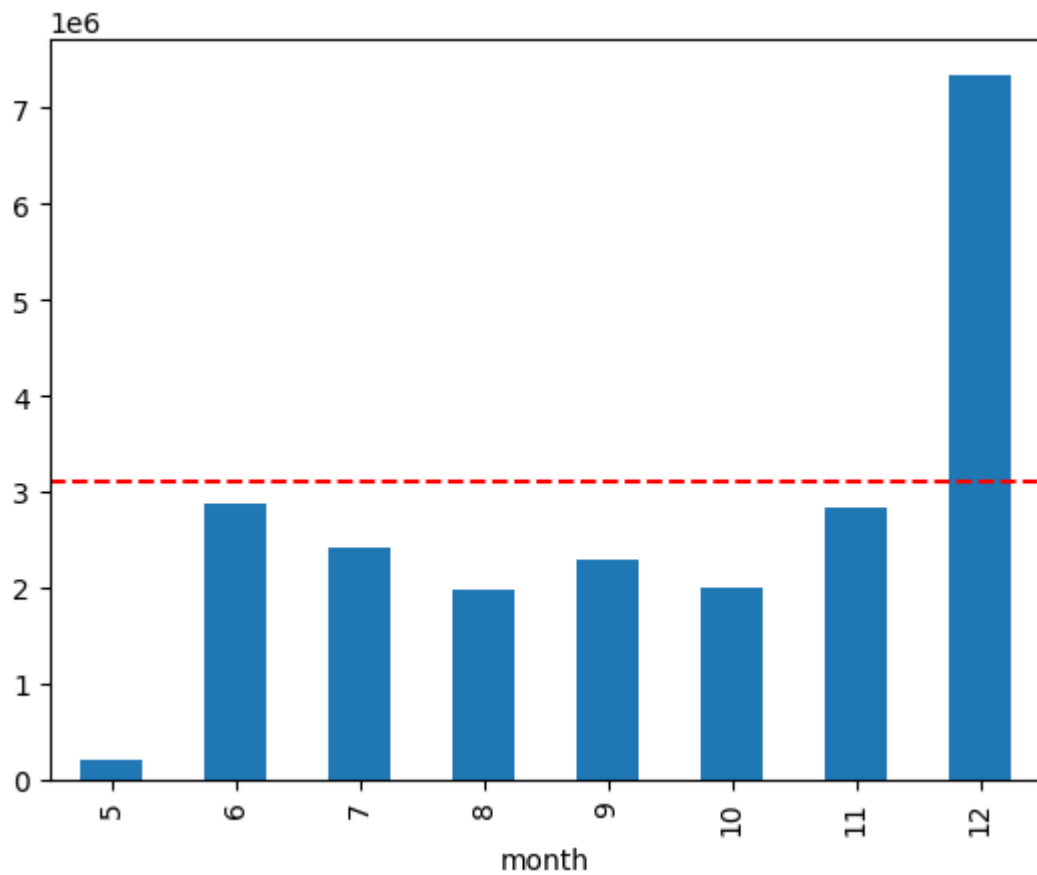
Ahora se comenzó instalando e importando la librería `matplotlib` para graficar las ventas por cada mes y una línea transversal que indique el valor promedio desde el mes 6.

```
In [39]: ! pip install matplotlib

Requirement already satisfied: matplotlib in c:\users\principal\anaconda3\envs\diana\lib\site-packages (3.6.3)
Requirement already satisfied: pyparsing>=2.2.1 in c:\users\principal\anaconda3\envs\diana\lib\site-packages (from matplotlib) (3.0.4)
Requirement already satisfied: fonttools>=4.22.0 in c:\users\principal\anaconda3\envs\diana\lib\site-packages (from matplotlib) (4.38.0)
Requirement already satisfied: kiwisolver>=1.0.1 in c:\users\principal\anaconda3\envs\diana\lib\site-packages (from matplotlib) (1.4.4)
Requirement already satisfied: contourpy>=1.0.1 in c:\users\principal\anaconda3\envs\diana\lib\site-packages (from matplotlib) (1.0.6)
Requirement already satisfied: cyclor>=0.10 in c:\users\principal\anaconda3\envs\diana\lib\site-packages (from matplotlib) (0.11.0)
Requirement already satisfied: pillow>=6.2.0 in c:\users\principal\anaconda3\envs\diana\lib\site-packages (from matplotlib) (9.4.0)
Requirement already satisfied: packaging>=20.0 in c:\users\principal\anaconda3\envs\diana\lib\site-packages (from matplotlib) (21.3)
Requirement already satisfied: python-dateutil>=2.7 in c:\users\principal\anaconda3\envs\diana\lib\site-packages (from matplotlib) (2.8.2)
Requirement already satisfied: numpy>=1.19 in c:\users\principal\anaconda3\envs\diana\lib\site-packages (from matplotlib) (1.22.3)
Requirement already satisfied: six>=1.5 in c:\users\principal\anaconda3\envs\diana\lib\site-packages (from python-dateutil>=2.7->matplotlib) (1.16.0)
```

Para graficar las ventas por cada mes, se puede utilizar el método `.plot()` de una serie de ventas por mes y el argumento `kind='bar'` el cual indica que se desea graficar un gráfico de barras. También se va a agregar una línea que indique el promedio de ventas por mes.

```
In [40]: import matplotlib.pyplot as plt
sales_by_month.plot(kind='bar')
plt.axhline(average_sales, color='r', linestyle='--')
plt.show()
```



Se desea obtener una grafica con más detalles por lo cual vamos a instalar altair y codificar con `alt.Chart().mark_bar().encode(x,y)`

```
In [41]: ! pip install altair
```

Requirement already satisfied: altair in c:\users\principal\anaconda3\envs\diana\lib\site-packages (4.2.0)
 Requirement already satisfied: entrypoints in c:\users\principal\anaconda3\envs\diana\lib\site-packages (from altair) (0.4)
 Requirement already satisfied: Jinja2 in c:\users\principal\anaconda3\envs\diana\lib\site-packages (from altair) (3.1.2)
 Requirement already satisfied: toolz in c:\users\principal\anaconda3\envs\diana\lib\site-packages (from altair) (0.12.0)
 Requirement already satisfied: numpy in c:\users\principal\anaconda3\envs\diana\lib\site-packages (from altair) (1.22.3)
 Requirement already satisfied: pandas>=0.18 in c:\users\principal\anaconda3\envs\diana\lib\site-packages (from altair) (1.4.2)
 Requirement already satisfied: jsonschema>=3.0 in c:\users\principal\anaconda3\envs\diana\lib\site-packages (from altair) (4.16.0)
 Requirement already satisfied: pyparsing!=0.17.0,!=0.17.1,!=0.17.2,>=0.14.0 in c:\users\principal\anaconda3\envs\diana\lib\site-packages (from jsonschema>=3.0->altair) (0.18.1)
 Requirement already satisfied: attrs>=17.4.0 in c:\users\principal\anaconda3\envs\diana\lib\site-packages (from jsonschema>=3.0->altair) (22.1.0)
 Requirement already satisfied: python-dateutil>=2.8.1 in c:\users\principal\anaconda3\envs\diana\lib\site-packages (from pandas>=0.18->altair) (2.8.2)
 Requirement already satisfied: pytz>=2020.1 in c:\users\principal\anaconda3\envs\diana\lib\site-packages (from pandas>=0.18->altair) (2022.1)
 Requirement already satisfied: MarkupSafe>=2.0 in c:\users\principal\anaconda3\envs\diana\lib\site-packages (from Jinja2->altair) (2.1.1)
 Requirement already satisfied: six>=1.5 in c:\users\principal\anaconda3\envs\diana\lib\site-packages (from python-dateutil>=2.8.1->pandas>=0.18->altair) (1.16.0)

Gráfica #2

```
In [42]: import altair as alt

chart = alt.Chart(sales_by_month.reset_index()).mark_bar().encode(x='month',y='venta_t
(chart).display()
```

COLORES

¿Cual es la cantidad de diferentes colores vendidos por mes? Aquí también se utiliza .groupby() así como el conteo de colores unicos y los meses como indices.

```
In [43]: # Grupos de datos por colores y conteo de colores unicos
sales_by_month = data.groupby('month')['color_prenda'].nunique().reset_index()
sales_by_month = sales_by_month.rename(columns={'color_prenda':'cantidad_colores'})
# Imprime el resultado en una tabla
print(sales_by_month)
```

	month	cantidad_colores
0	5	0
1	6	3
2	7	15
3	8	13
4	9	15
5	10	12
6	11	15
7	12	34

Se quiere saber los diferentes colores vendidos, para la cual se toma como muestra el mes 11.

```
In [44]: month_11_data = data.query("month == 11")
colors_month_11 = month_11_data["color_prenda"].unique()
print(colors_month_11)
```

```
['CAFE' 'NEGRO' 'VERDE' 'NARANJA' 'GRIS CLARO' 'ROSADO' 'MOSTAZA' 'CRUDO'
'GRIS' 'ROJO' 'CAMEL' 'AZUL' 'NEGRO BLANCO' 'NEGRO-BEIGE' 'CREMA']
```

También se quiere que esta lista este en orden Descendente según la cantidad de prendas vendidas por color, para esto se usa `.sort_values(" ",ascending=False)` y `.reset_index(drop=True)`

```
In [45]: month_11_data = data[data["month"] == 11]
color_month_11 = month_11_data.groupby("color_prenda").size().reset_index(name="cantidad_prendas")
color_month_11 = color_month_11.sort_values("cantidad_prendas",ascending=False)
color_month_11 = color_month_11.reset_index(drop=True)
colors_month_11 = color_month_11["color_prenda"].tolist()
print(colors_month_11)
```

```
['NEGRO', 'GRIS', 'ROJO', 'AZUL', 'CAMEL', 'CRUDO', 'GRIS CLARO', 'MOSTAZA', 'VERDE',
'CAFE', 'CREMA', 'NARANJA', 'NEGRO BLANCO', 'NEGRO-BEIGE', 'ROSADO']
```

Se prueba de que estos datos se puedan mostrar como una tabla

```
In [46]: month_11_data = data.query("month == 11")
colors_month_11 = month_11_data.groupby(["color_prenda"]).size().reset_index(name="cantidad_prendas")
colors_month_11 = colors_month_11.sort_values("cantidad_prendas",ascending=False)
colors_month_11 = colors_month_11.reset_index(drop=True)
print(colors_month_11)
```

	color_prenda	cantidad_prendas
0	NEGRO	17
1	GRIS	7
2	ROJO	5
3	AZUL	3
4	CAMEL	2
5	CRUDO	2
6	GRIS CLARO	2
7	MOSTAZA	2
8	VERDE	2
9	CAFE	1
10	CREMA	1
11	NARANJA	1
12	NEGRO BLANCO	1
13	NEGRO-BEIGE	1
14	ROSADO	1

¿Cuales son los colores más vendidos en los últimos 7 meses de 2022?

Para esto se utiliza '.query' con un rango de fechas.

```
In [47]: month_6_12_data = data.query("month >= 6 & month <= 12")
colors_month_6_12 = month_6_12_data.groupby(["color_prenda"]).size().reset_index(name=
colors_month_6_12 = colors_month_6_12.sort_values("cantidad_prendas",ascending=False)
colors_month_6_12 = colors_month_6_12.reset_index(drop=True)
print(colors_month_6_12)
```

	color_prenda	cantidad_prendas
0	NEGRO	96
1	GRIS	59
2	AZUL	28
3	ROSADO	24
4	ROJO	19
5	GRIS CLARO	13
6	BLANCO	13
7	VERDE	11
8	BEIGE	7
9	CAFE	7
10	CRUDO	7
11	MOSTAZA	5
12	CREMA	5
13	CAMEL	4
14	GRIS OSCURO	4
15	MORADO	4
16	TERRACOTA	3
17	AGUA MARINA	3
18	NARANJA	3
19	VINOTINTO	3
20	GRIS MEDIO	3
21	AZUL OSCURO	3
22	NEGRO BLANCO	2
23	ROSA GUAVA	1
24	VINO TINTO	1
25	VERDE OSCURO	1
26	VERDE CLARO	1
27	TOMATE	1
28	BEIS	1
29	TABACO	1
30	BLANCO NEGRO	1
31	MENTA	1
32	ROSA	1
33	GRIS NEGRO	1
34	PIEL	1
35	PERLA	1
36	PALO ROSA	1
37	NEGRO-GRIS	1
38	NEGRO-BLANCO	1
39	NEGRO-BEIGE	1
40	AMARILLO	1
41	CIAN	1

Gráfica #3

Se desea ver la anterior tabla en una grafica, para lo cual utilizamos matplotlib


```
In [48]: # Datos para graficar
x = colors_month_6_12['color_prenda']
y = colors_month_6_12['cantidad_prendas']

fig, ax = plt.subplots()
ax.plot(x, y)

ax.set(xlabel='Colores', ylabel='Cantidad de prendas vendidas',
       title='Ventas de prendas por color desde el mes 6 al 12')
ax.grid()
plt.xticks(rotation = 90)
plt.tight_layout()
plt.show()
```

