



TRABAJO DE GRADO
Opción Seminario-Diplomado.

**ESTRATEGIA COMPUTACIONAL PARA ANALIZAR LA SATISFACCION DE UN
CLIENTE EN LA EXPERIENCIA DE COMPRAR DE UN PRODUCTO, UTILIZANDO
ALGORITMOS DE MACHINE LEARNING**

Corporación Universitaria Remington.
Nombre de la facultad: Ingenierías
Nombre del programa académico: Ingeniería Industrial

Estudiante:
Jean Carlos Corredor Ferro

Tutor: Juan Carlos Briñez de León

Opción de Trabajo de grado Seminario-Diplomado.
2025.

Dedicatoria

A MI ESPOSA Y FAMILIA, QUIENES HAN ESTADO PRESENTE EN ESTE SUEÑO Y
META DE SER PROFESIONAL.

Tabla de Contenidos

Resumen.....	4
Palabras clave.....	5
Marco conceptual y contextual	6
1.1.1 Sistemas de regresión.....	6
1.1.2 Algoritmos de Machine learning en sistemas de regresión.	6
1.1.2.1 Algoritmo KNN (modelo de predicción para series de tiempo)	6
1.1.2.2 Algoritmo ANN (modelo de red neuronal artificial)	7
1.1.2.3 Algoritmo de regresión lineal	8
1.1.2.4 Algoritmo árbol de decisión	8
1.1.2.5 Algoritmo SVR (máquinas de vectores de soporte para regresión)	9
1.2 Descripción de caso de estudio.	9
1.3 Pregunta problema:	9
1.4 Hipótesis:	10
Objetivos.....	11
2.1 Objetivo general.....	11
2.2 Objetivos específicos.	11
Desarrollo e implementación del aprendizaje.....	12
3.1 Preparación y análisis de los datos.....	12
3.1.1 Variables involucradas	12
3.1.2 Carga de datos	14
3.1.3 Analizar Datos	15
3.2 Modelo de toma de decisiones	27
3.3 Análisis de desempeño.....	28
Conclusiones y trabajos futuros.....	29
Referencias bibliográficas.....	30

Resumen

El trabajo “Estrategia computacional para analizar la satisfacción de un cliente en la experiencia de comprar de un producto, utilizando algoritmos de machine learning”, tiene como finalidad analizar el nivel de satisfacción del cliente, así como identificar los factores que más influyen tras la compra de productos. Para ello, se trabajó con un conjunto de datos obtenidos a partir de encuestas aplicadas a los consumidores de la marca, donde se aplicaron técnicas de análisis de datos utilizando el lenguaje de programación Python.

En la etapa de exploración, se utilizaron herramientas como pandas para la lectura y manipulación de los datos y librerías de visualización. Se generaron diferentes gráficos descriptivos entre ellos: tabla de frecuencia, gráfico de pastel, gráfico de densidad, gráfico de caja y bigote y matriz de correlación. Todo esto para analizar la calidad del producto, niveles de satisfacción, precio, experiencias de compra y relaciones significativas entre variables.

Los resultados indican que todas las variables seleccionadas son factores determinantes en la satisfacción del cliente. La aplicación práctica de estos algoritmos y visualizaciones permitió no solo validar conocimientos teóricos en ciencia de datos, sino también obtener conclusiones útiles para mejorar estrategias de atención, calidad, marketing, fidelización y mejora continua en el entorno comercial.

Este estudio demuestra que el uso de herramientas computacionales basadas en datos puede transformarse en un recurso estratégico para la toma de decisiones empresariales informadas, orientadas al cliente u con base en evidencia real.

Palabras clave

- Sistemas de regresión,
- Satisfacción,
- Experiencia,
- Compra,
- Productos,
- Clientes
- Python
- Análisis de datos
- Algoritmos
- Factores de influencia
- Calidad
- Conjunto de datos

Marco conceptual y contextual

1.1 Contexto:

1.1.1 Sistemas de regresión.

En el actual entorno empresarial altamente competitivo, comprender el nivel de satisfacción del cliente se ha convertido en una prioridad estratégica. Las marcas buscan no solo evaluar la experiencia de compra, sino también anticiparse a sus necesidades y expectativas.

El presente estudio se desarrolló mediante análisis de información recolectada a través de una encuesta de satisfacción aplicada a clientes que realizaron compras de alguna prenda, con el objetivo de identificar los factores que más influyen en la percepción general de su experiencia. Las variables evaluadas incluyeron aspectos como la calidad de la prenda, precio, atención recibida, tiempo de entrega, entre otros.

Para analizar estos datos, se implementó un modelo de regresión utilizando Python, lo cual permitió interpretar de manera cuantitativa la relación entre cada factor y la calificación final que otorgan los clientes. Además, el modelo, no solo explica el comportamiento actual de los clientes, sino que también permite predecir una futura satisfacción de nuevos compradores partiendo de respuestas en la encuesta. Esto transforma el modelo en una herramienta predictiva para evaluar el impacto potencial de cambios en el producto y atención antes de implementarlos.

1.1.2 Algoritmos de Machine learning en sistemas de regresión.

1.1.2.1 Algoritmo KNN (modelo de predicción para series de tiempo).

```
from sklearn.neighbors import KNeighborsRegressor as KNNR
Modelo_1=KNNR()
Modelo_1.fit(Entradas,Salida)
Prediction1 = Modelo_1.predict(Entradas)
RMSE1 = np.sqrt(np.mean((Prediction1-Salida)**2))
print('El error RMSE del modelo KNN es: ',RMSE1)
```

Este algoritmo nos ayuda a predecir valores futuros o presentes, evaluar el rendimiento del modelo usando el error cuadrático medio (RMSE) y funciona mejor cuando los datos siguen patrones repetitivos y no se salen del rango de entrenamiento (útil en datos estacionarios o acotados).

En contexto el algoritmo KNN nos permite saber el nivel de satisfacción en una escala de 1 a 5 a partir de factores como:

- Precio
- Calidad
- Variedad
- Estética
- Atención al cliente

El algoritmo KNN ha sido validado en investigaciones recientes para predecir la satisfacción del cliente en contextos reales, Srividya y Akila aplicaron el algoritmo sobre datos obtenidos de encuestas en una empresa de servicio, logrando una predicción precisa del puntaje de satisfacción. Este enfoque respalda el uso de KNN en nuestro estudio.

1.1.2.2 Algoritmo ANN (modelo de red neuronal artificial).

```
from sklearn.neural_network import MLPRegressor
Modelo_2 = MLPRegressor()
Modelo_2.fit(Entradas,Salida)
Prediction2 = Modelo_2.predict(Entradas)
RMSE2 = np.sqrt(np.mean((Prediction2-Salida)**2))
print('El error RMSE del modelo ANN es: ',RMSE2)
```

Este algoritmo es ideal en problemas no lineales, donde modelos simples como regresión lineal o KNN no funcionan bien, nos ayuda a predecir valores continuos y es útil en serie de tiempo, ingeniería entre otros.

Cuando aplicas una ANN a datos de encuestas, ingresas variables como precio, calidad, atención, entre otros factores relevantes. La red se entrena con estos datos para predecir el puntaje de satisfacción general otorgado por el cliente tras la compra.

En un estudio realizado por Zeinalizadeh, Shojaie y Shariatmadari (2015), se empleó una red neuronal MLP para predecir la satisfacción de clientes bancarios luego de un servicio. El modelo fue entrenado con datos de encuestas que contenían variables como tarifas, velocidad del servicio, tecnología y comportamiento del personal. Al compararlo con un modelo de regresión lineal, la ANN obtuvo una precisión superior en aproximadamente un **73 %**, y permitió identificar los factores con mayor influencia, como tarifas y atención rápida

1.1.2.3 Algoritmo de regresión lineal.

```
from sklearn.linear_model import LinearRegression

modelo_3 = LinearRegression()
modelo_3.fit(Entradas, Salida)
Prediction3 = modelo_3.predict(Entradas)
RMSE3 = np.sqrt(np.mean((Prediction3 - Salida) ** 2))
print("Modelo 3 - Regresión Lineal, RMSE:", RMSE3)
```

Este algoritmo puede ser simple o múltiple, todo esto dependiendo de estradas que tenga, nos ayuda ha hacer predicciones simples y rápidas, útil como modelo base y es fácil de interpretar ya que puedes la influencia de cada variable con los coeficientes.

Cuando se aplica el algoritmo a un conjunto de datos obtenidos por una encuesta, en este caso para la satisfacción de un cliente, la regresión lineal nos ayuda a:

Identificar los factores que mas influyen en la satisfacción

Cuantificar el impacto de cada uno de los factores

Predecir niveles de satisfacción en función de características observables

1.1.2.4 Algoritmo árbol de decisión.

```
from sklearn.tree import DecisionTreeRegressor

modelo_4 = DecisionTreeRegressor(random_state=0)
modelo_4.fit(Entradas, Salida)
Prediction4 = modelo_4.predict(Entradas)
RMSE4 = np.sqrt(np.mean((Prediction4 - Salida) ** 2))
print("Modelo 4 - Árbol de Decisión, RMSE:", RMSE4)
```

Este es un algoritmo adaptado para regresión, a diferencia del árbol usado para clasificación, este modelo predice valores continuos, no categorías. Este fue diseñado para realizar predicciones sobre la variable de salida a partir de los datos de entrada y captura relaciones no lineales sin necesidad de escalar datos.

Wei-Yin Loh presenta en este artículo una visión integral y crítica de los métodos de árboles de decisión, tanto para clasificación como para regresión. El foco central es el árbol de regresión, ideal para variables objetivo continuas como la satisfacción del cliente.

1.1.2.5 Algoritmo SVR (máquinas de vectores de soporte para regresión).

```
from sklearn.svm import SVR

modelo_6 = SVR(kernel='rbf')
modelo_6.fit(Entradas, Salida)
Prediction6 = modelo_6.predict(Entradas)
RMSE6 = np.sqrt(np.mean((Prediction6 - Salida) ** 2))
print("Modelo 6 - SVR (kernel RBF), RMSE:", RMSE6)
```

Es fundamental contar con modelos predictivos que permitan anticipar el nivel de satisfacción en función de múltiples variables como el precio, calidad, estética, atención al cliente entre otros. El algoritmo SVR se presenta como alternativa robusta y efectiva, especialmente cuando los datos muestran relaciones no lineales complejas.

En encuestas de satisfacción, el uso de este algoritmo no responde preguntas claves como: ¿Qué tan satisfecho estará el cliente si el precio baja y la variedad sube?

¿Cuánto impacta la estética en la satisfacción del cliente, en comparación de la atención recibida?

Según Smola y Schölkopf, SVR es especialmente útil cuando se trabaja con datos limitados y se requiere un modelo que pueda generalizar bien sin sobre ajustarse, lo cual es frecuente en estudios de satisfacción donde la muestra puede ser relativamente pequeña pero multivariable.

1.2 Descripción de caso de estudio.

En este caso se centra en la recopilación y análisis de un conjunto de datos obtenido a través de una encuesta de satisfacción aplicada a clientes que adquirieron algún producto.

El objetivo principal es evaluar la experiencia del cliente durante y después de la compra, considerando aspectos importantes como la calidad, precio, estética, durabilidad, atención al cliente, durabilidad, rapidez en el envío, experiencia de compra, confianza de marca, funcionalidad, utilidad percibida y la percepción general del valor recibido.

1.3 Pregunta problema:

¿Cómo puedo utilizar algoritmos de análisis de datos en Python para identificar los factores que influyen en la satisfacción del cliente a la hora de obtener o comprar los productos?

1.4 Hipótesis:

Después de aplicar algoritmos de análisis de datos en Python a los resultados de la encuesta de satisfacción por parte del cliente, se podrán identificar factores que tienen mayor influencia en la calificación final de la compra del producto, permitiendo conocer el nivel de satisfacción con un alto grado de precisión.

Objetivos

2.1 Objetivo general.

Analizar y evaluar los factores que influyen en la satisfacción del cliente con respecto a la compra de alguna prenda, a partir de un conjunto de datos obtenido mediante encuestas, con el fin de identificar patrones, relaciones significativas y oportunidades de mejora en la experiencia de los consumidores.

2.2 Objetivos específicos.

- Identificar y describir las variables que influyen en la impresión del cliente, tales como el precio, atención al cliente y la calidad del producto.
- Importar y limpiar datos en bibliotecas como pandas, creando gráficos para representar visualmente la satisfacción del cliente.
- Analizar correlaciones entre variables.
- Comparar y analizar niveles de satisfacción por periodos a largo tiempo.
- Proponer recomendaciones basadas en los hallazgos para una mejora en la experiencia del cliente.

Desarrollo e implementación del aprendizaje

El trabajo tiene como objetivo conocer el nivel de satisfacción de un cliente luego de realizar la compra de un producto, y analizar qué factores influyen en dicha satisfacción. Para ello se utilizó un conjunto de datos recolectados mediante una encuesta, aplicada a 150 consumidores, donde se midieron variables como calidad, precio, estética, durabilidad, experiencia, funcionabilidad, entre otras, medidas en una escala de 1 a 10.

El problema de estudio responde a la necesidad de las empresas de conocer con precisión que aspectos de la experiencia del cliente son determinantes en su nivel de satisfacción, lo cual es clave para la fidelización y retención de consumidores.

3.1 Preparación y análisis de los datos

3.1.1 Variables involucradas.

3.1.1.1 Variables de entrada.

1. Facilidad de uso
2. Calidad
3. Precio (valoración del cliente, no el precio real)
4. Estética
5. Durabilidad
6. Atención al cliente
7. Variedad
8. Rapidez de envío
9. Experiencia de compra
10. Confianza en la marca
11. Funcionalidad
12. Utilidad percibida

3.1.1.1.1 Significado detallado variable de entrada.

Valor Significado

1	Extremadamente insatisfecho / Muy deficiente
2	Muy insatisfecho
3	Insatisfecho
4	Ligeramente insatisfecho
5	Neutral / Ni satisfecho ni insatisfecho
6	Ligeramente satisfecho
7	Satisfecho
8	Muy satisfecho
9	Casi perfecto
10	Excelente / Totalmente satisfecho

3.1.1.2 Variable de salida.

1. Calificación final (salida, escala de 1 a 5)

3.1.1.2.1 Significado detallado variable de salida.

Calificación Final	Significado
1	Muy insatisfecho — el producto no cumplió expectativas en ningún aspecto.
2	Insatisfecho — varios aspectos negativos, aunque alguno puede haber sido aceptable.
3	Neutral o aceptable — experiencia mixta, ni buena ni mala.
4	Satisfecho — el producto cumple con lo esperado, con pequeños detalles a mejorar.
5	Muy satisfecho / Excelente — superó expectativas, experiencia muy positiva en general.

3.1.2 Carga de datos

Este algoritmo nos permite acceder, virtualizar y manipular la información que podremos analizar mediante algoritmos estadísticos. En el contexto de la encuesta realizada de satisfacción a clientes, este proceso me permite trasladar información recolectada en formularios o base de datos a un entorno automatizado. Cuando se realizó la encuesta de satisfacción después de cada compra de un producto, los datos recolectados se almacenaron en un archivo Excel.

```
#Para cargar los datos
import pandas as pd
from google.colab import files
uploaded = files.upload()
for filename in uploaded.keys():
    #Conjunto_Datos = pd.read_csv(filename,sep=',')
    Conjunto_Datos = pd.read_excel(filename)
Conjunto_Datos.head(15)
```

Le pedimos que nos deje visualizar las primeras 15 filas.

ID	Calidad	Precio	Estética	Durabilidad	Atención cliente	Variedad	Rapidez envío	Experiencia compra	Confianza marca	Funcionalidad	Utilidad percibida	Calificación final	
0	1	9	7	6	9	8	7	9	8	9	8	9	5
1	2	7	5	5	6	7	6	6	7	6	7	6	4
2	3	9	8	9	9	9	9	9	9	9	9	9	5
3	4	5	4	5	5	4	5	4	5	5	5	5	3
4	5	3	2	2	3	3	2	3	3	2	3	2	2
5	6	7	9	9	6	8	9	6	8	9	9	4	5
6	7	8	7	9	8	4	6	7	3	3	5	7	4
7	8	4	4	9	7	4	8	8	3	9	5	8	4
8	9	9	3	4	6	8	7	6	3	3	5	4	3
9	10	7	6	6	7	7	9	3	8	8	6	3	4
10	11	3	6	8	6	4	4	3	7	5	9	6	4
11	12	8	8	4	8	4	6	8	5	4	5	6	4
12	13	7	3	4	8	9	3	3	7	8	7	8	4
13	14	3	3	7	4	4	2	8	3	4	3	4	3
14	15	6	7	4	5	9	9	7	3	4	6	4	4

3.1.3 Analizar Datos

3.1.3.1 Información de datos.

Este algoritmo nos muestra un resumen técnico y general del conjunto de datos.

```
#Información de la estructura de datos
Conjunto_Datos.info()
```

El algoritmo nos indicó que tenemos:

- 150 filas, es decir de 0 a 149.
- 13 columnas.
- Entradas de datos completas.
- Tipos de datos numéricos completos de 64 bits

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 150 entries, 0 to 149
Data columns (total 13 columns):
#   Column                Non-Null Count  Dtype
---  -
0   ID                    150 non-null   int64
1   Calidad               150 non-null   int64
2   Precio               150 non-null   int64
3   Estética              150 non-null   int64
4   Durabilidad          150 non-null   int64
5   Atención cliente     150 non-null   int64
6   Variedad              150 non-null   int64
7   Rapidez envío        150 non-null   int64
8   Experiencia compra   150 non-null   int64
9   Confianza marca      150 non-null   int64
10  Funcionalidad         150 non-null   int64
11  Utilidad percibida   150 non-null   int64
12  Calificación final    150 non-null   int64
dtypes: int64(13)
memory usage: 15.4 KB
```

3.1.3.2 Eliminar columnas por nombres.

Este algoritmo nos ayuda a eliminar columnas o filas de nuestro conjunto de datos.

```
#Quitando columnas indeseadas
Conjunto_Datos = Conjunto_Datos.drop(['ID'],axis=1)
#resumen de los datos
Conjunto_Datos.head()
```

En este cajo eliminamos la columna ID para reducir un poco el tamaño de datos y por qué es la más irrelevante.

	Calidad	Precio	Estética	Durabilidad	Atención cliente	Variiedad	Rapidez envío	Experiencia compra	Confianza marca	Funcionalidad	Utilidad percibida	Calificación final
0	9	7	6	9	8	7	9	8	9	8	9	5
1	7	5	5	6	7	6	6	7	6	7	6	4
2	9	8	9	9	9	9	9	9	9	9	9	5
3	5	4	5	5	4	5	4	5	5	5	5	3
4	3	2	2	3	3	2	3	3	2	3	2	2
5	7	9	9	6	8	9	6	8	9	9	4	5
6	8	7	9	8	4	6	7	3	3	5	7	4
7	4	4	9	7	4	8	8	3	9	5	8	4
8	9	3	4	6	8	7	6	3	3	5	4	3
9	7	6	6	7	7	9	3	8	8	6	3	4
10	3	6	8	6	4	4	3	7	5	9	6	4
11	8	8	4	8	4	6	8	5	4	5	6	4
12	7	3	4	8	9	3	3	7	8	7	8	4
13	3	3	7	4	4	2	8	3	4	3	4	3
14	6	7	4	5	9	9	7	3	4	6	4	4

3.1.3.3 Eliminar valores con faltantes.

Este algoritmo nos ayuda a eliminar datos, eliminando filas incompletas que podría afectar el análisis de los modelos.

```
#Elimina filas que tengan datos nulos
Conjunto_Datos = Conjunto_Datos.dropna()
Conjunto_Datos.head(15)
```

En el conjunto de datos tenemos toda la información completa, en este caso no fue necesario utilizar este algoritmo.

	Calidad	Precio	Estética	Durabilidad	Atención cliente	Variedad	Rapidez envío	Experiencia compra	Confianza marca	Funcionalidad	Utilidad percibida	Calificación final
0	9	7	6	9	8	7	9	8	9	8	9	5
1	7	5	5	6	7	6	6	7	6	7	6	4
2	9	8	9	9	9	9	9	9	9	9	9	5
3	5	4	5	5	4	5	4	5	5	5	5	3
4	3	2	2	3	3	2	3	3	2	3	2	2
5	7	9	9	6	8	9	6	8	9	9	4	5
6	8	7	9	8	4	6	7	3	3	5	7	4
7	4	4	9	7	4	8	8	3	9	5	8	4
8	9	3	4	6	8	7	6	3	3	5	4	3
9	7	6	6	7	7	9	3	8	8	6	3	4
10	3	6	8	6	4	4	3	7	5	9	6	4
11	8	8	4	8	4	6	8	5	4	5	6	4
12	7	3	4	8	9	3	3	7	8	7	8	4
13	3	3	7	4	4	2	8	3	4	3	4	3
14	6	7	4	5	9	9	7	3	4	6	4	4

3.1.3.4 Tabla de frecuencia.

Este algoritmo agrupa datos numéricos en rangos y etiqueta cada valor según el intervalo al que pertenece.

```
# Realizando tabla de frecuencia según el numero de rangos
numero_de_rangos = 4 #el número de rangos es aproximadamente la raíz cuadrada del número de muestras

# Calcular la tabla de frecuencia absoluta con bins
Tabla_frecuencia = pd.cut(Conjunto_Datos['Calidad'], bins=numero_de_rangos, include_lowest=True, right=False).value_counts().reset_index()
Tabla_frecuencia.columns = ['Intervalo', 'Frecuencia Absoluta']

# Calcular la frecuencia acumulada
Tabla_frecuencia['Frecuencia Acumulada'] = Tabla_frecuencia['Frecuencia Absoluta'].cumsum()

# Mostrar la tabla de frecuencia
print("Tabla de Frecuencia:")
print(Tabla_frecuencia)
```

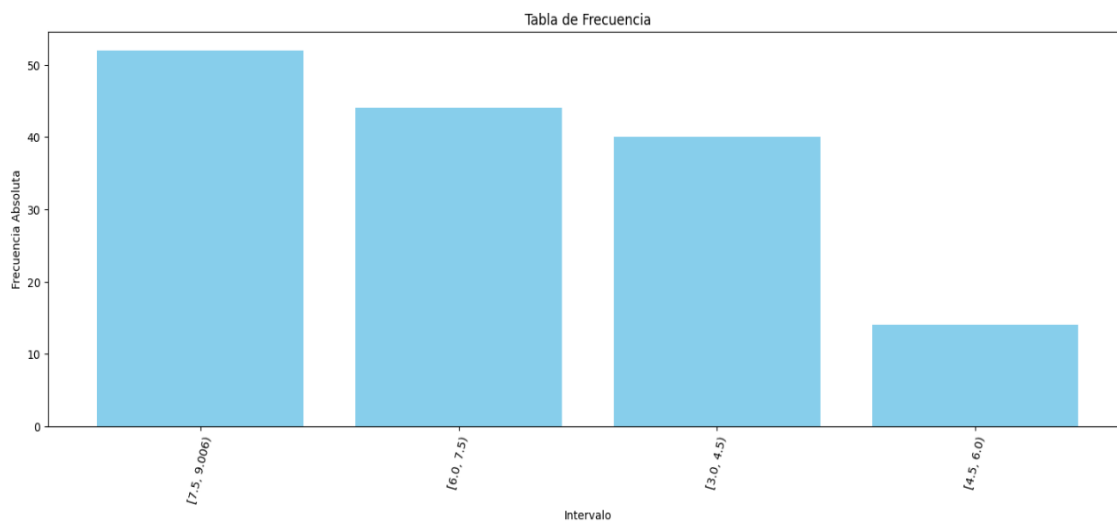
Para el análisis de este algoritmo trabajamos con la variable de calidad dividiéndola en cuatro rangos. En esta observamos que la mayoría de las respuestas se concentraron en un rango de 7,5 a 9, lo cual indica que en cuanto experiencia, los clientes perciben la calidad del producto como satisfactoria a casi perfecta. Esto sugiere un buen desempeño del producto en cuanto a la percepción del cliente, lo que puede ser considerado un punto fuerte para la empresa.

Tabla de Frecuencia:

	Intervalo	Frecuencia Absoluta	Frecuencia Acumulada
0	[7.5, 9.006)	52	52
1	[6.0, 7.5)	44	96
2	[3.0, 4.5)	40	136
3	[4.5, 6.0)	14	150

3.1.3.5 Grafico Tabla de frecuencia.

Mediante este algoritmo podemos visualizar gráficamente los datos de la tabla de frecuencia. En esta ocasión podemos observar que 40 clientes calificaron la calidad del producto en el intervalo de 3.0 a 4.5 esto queriendo decir que su percepción está a un nivel de “insatisfecho” a “neutro”. Este resultado es muy importante resaltarlo ya que representa una oportunidad critica de mejora, ya que puede afectar a la fidelidad del cliente y reputación de la marca. Es recomendable tomar medidas correctivas que permitan elevar la percepción general de calidad en futuras evaluaciones.



3.1.3.6 Grafico para análisis.

Esta representación visual permite identificar con claridad el nivel general de satisfacción, resaltar posibles áreas de mejora y facilitar la toma de decisiones empresariales.

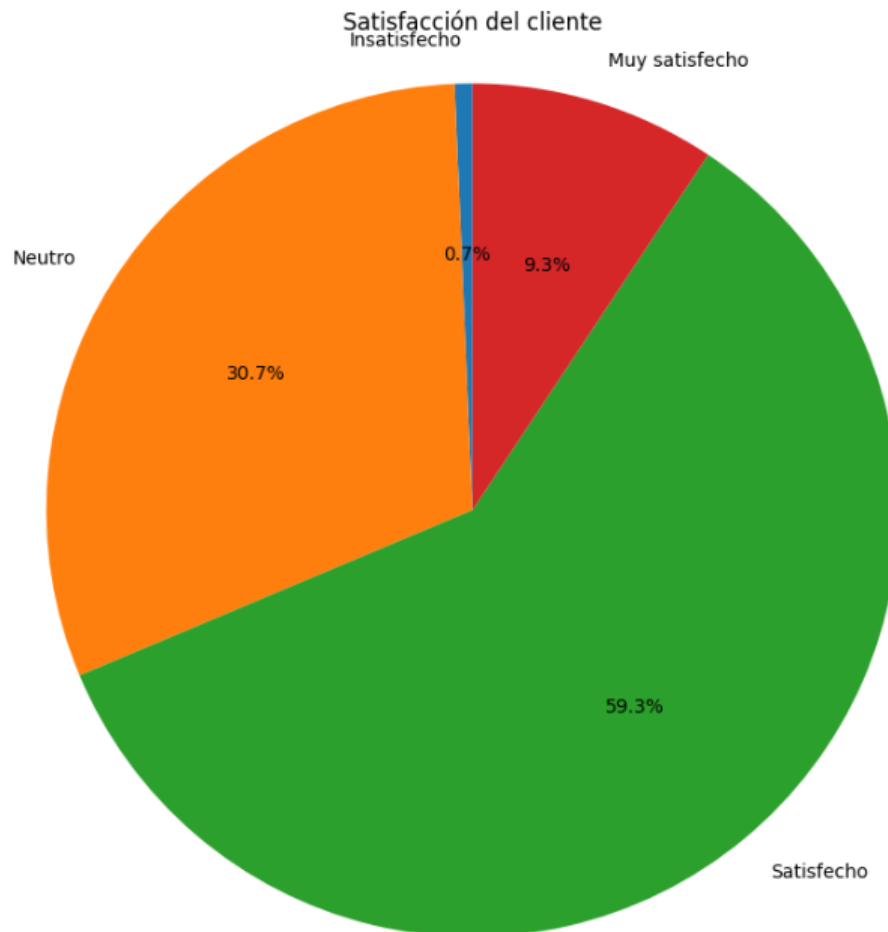
```
#Aquí podemos hacer un montón de análisis...
Conjunto_Datos['Rango_satisfaccion'] = pd.cut(Conjunto_Datos['Calificación final'], bins=[1, 2, 3, 4, 5], labels=['Insatisfecho', 'Neutro', 'Satisfecho', 'Muy satisfecho'])
frecuencias = Conjunto_Datos['Rango_satisfaccion'].value_counts().sort_index()
plt.figure(figsize=(25, 9))
plt.pie(frecuencias,
        labels=frecuencias.index,
        autopct='%1.1f%%',
        shadow=False,
        startangle=90)

plt.title('Satisfacción del cliente')
plt.axis('equal')
plt.show()
```

En la gráfica podemos visualizar que con un (59.3%) de los clientes se sienten satisfechos con el producto o servicio recibido, lo cual indica que la marca esta cumpliendo las expectativas de los consumidores. Además, un 9.3% se mostro muy satisfecho, lo cual representa una percepción positiva y destaca aspectos bien valorados en la experiencia del cliente.

También es importante resaltar que un 30.7% de los clientes se encuentra en una posición neutral, lo que sugiere que hay margen de mejora. Por otro lado, y menos importante un 0.7% de clientes están insatisfechos, tema que podría afectar la marca si no se toman acciones correctivas.

En resumen, la marca presenta un alto nivel de satisfacción por parte de los consumidores, pero también esta la oportunidad de fortalecer la experiencia de los clientes consolidando así su fidelidad.



3.1.3.7 Gráfico de densidad.

Con este algoritmo realizamos un gráfico de densidad el cual nos muestra como se distribuyen los valores de una variable numérica e identificando patrones y anomalías, es decir si hay datos extremos o múltiples picos.

```
# Gráfico de densidad para una variable (Almacenada en una columna)
sns.kdeplot(Conjunto_Datos['Experiencia compra'], fill=True)
plt.title('Gráfico de Densidad para experiencia de compra')
plt.xlabel('Experiencia de compra')
plt.ylabel('Densidad')
plt.show()
```

La grafica de densidad nos revela que la mayoría de los clientes encuestados evaluaron su experiencia de compra con valores entre 3 y 9, lo que indica una amplia diversidad de percepciones, sin embargo, el pico más alto se encuentra en el valor numero 9, lo cual representa que una gran parte de los clientes calificaron su experiencia como casi perfecta. Aunque existen opiniones variadas, la tendencia es altamente positiva. No obstante, la presencia de valores bajos indica que hay un segmento de clientes que no quedaron totalmente satisfechos, por lo que es recomendable identificar y mejorar puntos específicos.



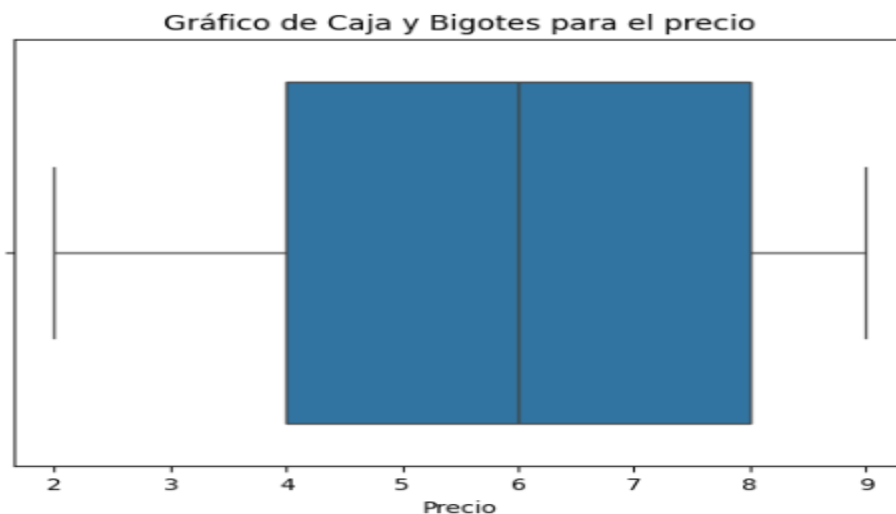
3.1.3.8 Grafico de caja y bigotes.

Este algoritmo tiene como objetivo representar gráficamente la distribución de una variable, utilizando un grafico de caja y bigotes en la que visualizaremos la dispersión de los datos, identificar la mediana, los cuartiles y detectar la presencia de valores atípicos.

```
# Gráfico de caja y bigotes para una variable (Almacenada en)
sns.boxplot(x=Conjunto_Datos['Precio'])
plt.title('Gráfico de Caja y Bigotes para el precio')
plt.xlabel('Precio')
plt.show()
```

A través del grafico generado por la variable de precio, se logra observar que los datos se distribuyen en una tendencia estable dentro un rango de 2 a 9, lo que muestra cierta variabilidad, no obstante, se destaca que la mayor concentración de los datos se encuentra entre los valores 4 “Ligeramente satisfecho” y 8 “Muy satisfecho”.

Sin embargo, los valores extremos, especialmente cercanos al rango 2, podrían representar casos puntuales en la percepción negativa por parte del precio o situaciones de insatisfacción del consumidor, que serían recomendable investigar con mayor profundidad, a fin de identificar posibles causas y oportunidades de mejora en la estrategia de precios como el posicionamiento de la marca.



3.1.3.9 Matriz de correlaciones.

Este algoritmo de matriz de correlaciones nos permite medir y visualizar la relación entre variables numéricas dentro de un conjunto de datos.

Los valores oscilan entre -1 y 1:

- 1 → correlación positiva perfecta.
- -1 → correlación negativa perfecta.
- 0 → sin correlación lineal.

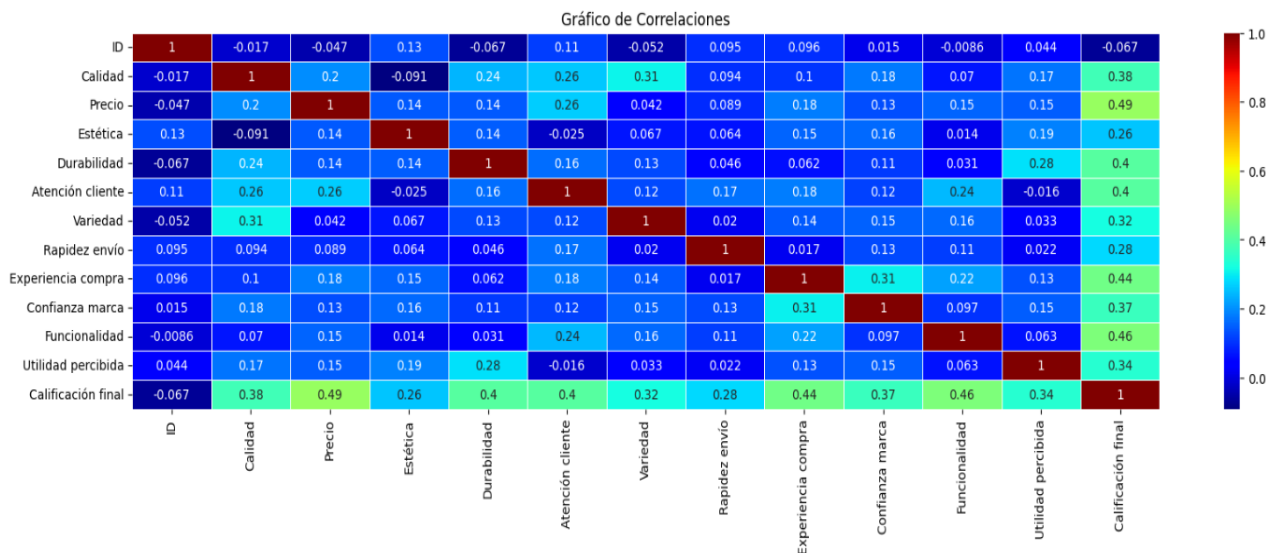
```
# Matriz de correlaciones
matriz_correlaciones = Conjunto_Datos.corr()
plt.figure(figsize=(20,5))

# Gráfico de correlaciones
sns.heatmap(matriz_correlaciones, annot=True, cmap='jet', linewidths=0.5)
plt.title('Gráfico de Correlaciones')
plt.show()
```

En la matriz podemos observar que hay muchos tipos de relación positiva débil.

Realizando un análisis entre la variable de calidad y variedad se encuentra una correlación de 0.31, lo que indica una relación positiva débil entre ambas variables. Esto sugiere que, a mayor variedad, puede haber una ligera tendencia a una mayor calidad percibida. Por tanto, la variedad un es un factor determinante en la calidad y debería analizarse otras variables que influyan de manera más significativa.

Por otra parte, encontramos las variables de estética y atención al cliente cuya correlación es de -0.025, lo cual indica una ausencia de relación significativa, sugiriendo que las percepciones sobre la estética de un producto o servicio no están asociadas con la calidad de atención al cliente, es decir que actúan de forma independiente.



3.1.3.10 Grafico KDE (Histograma con curva de densidad)

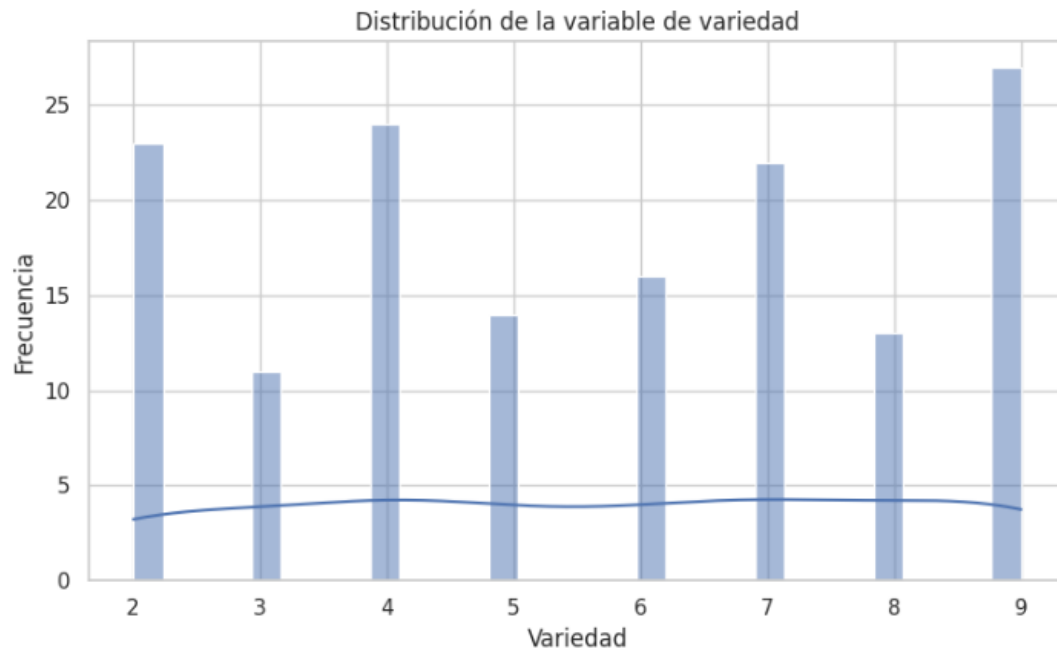
Este algoritmo lo utilizamos para exploración de datos, detección de patrones o problemas y como base para modelado es decir saber como se comporta la variable te ayuda a decidir el modelo estadístico o de machine learning aplicar.

```
import seaborn as sns
import matplotlib.pyplot as plt

sns.set(style="whitegrid")
plt.figure(figsize=(8, 5))
sns.histplot(Conjunto_Datos['Variedad'], bins=30, kde=True)
plt.title("Distribución de la variable de variedad")
plt.xlabel("Variedad")
plt.ylabel("Frecuencia")
plt.grid(True)
plt.tight_layout()
plt.show()
```

En la gráfica la variable variedad, se observa evaluada de una escala de 2 a 9, esto refleja la percepción del cliente sobre la diversidad de los productos ofrecidos en la tienda. Según la distribución en la gráfica se observa que la mayor concentración de respuestas se encuentra en un nivel alto, destacándose el valor 9 “casi perfecto”, esta tendencia sugiere que gran parte de los clientes que la tienda ofrece una gran variedad de prendas. También

se evidencia la alta frecuencia en niveles 2 “muy insatisfecho”, 3 “insatisfecho” y 4 “ligeramente insatisfecho” lo que indica que hay varios clientes que tienen una percepción negativa en este aspecto.



3.1.3.11 Grafico de una Entrada vs Salida

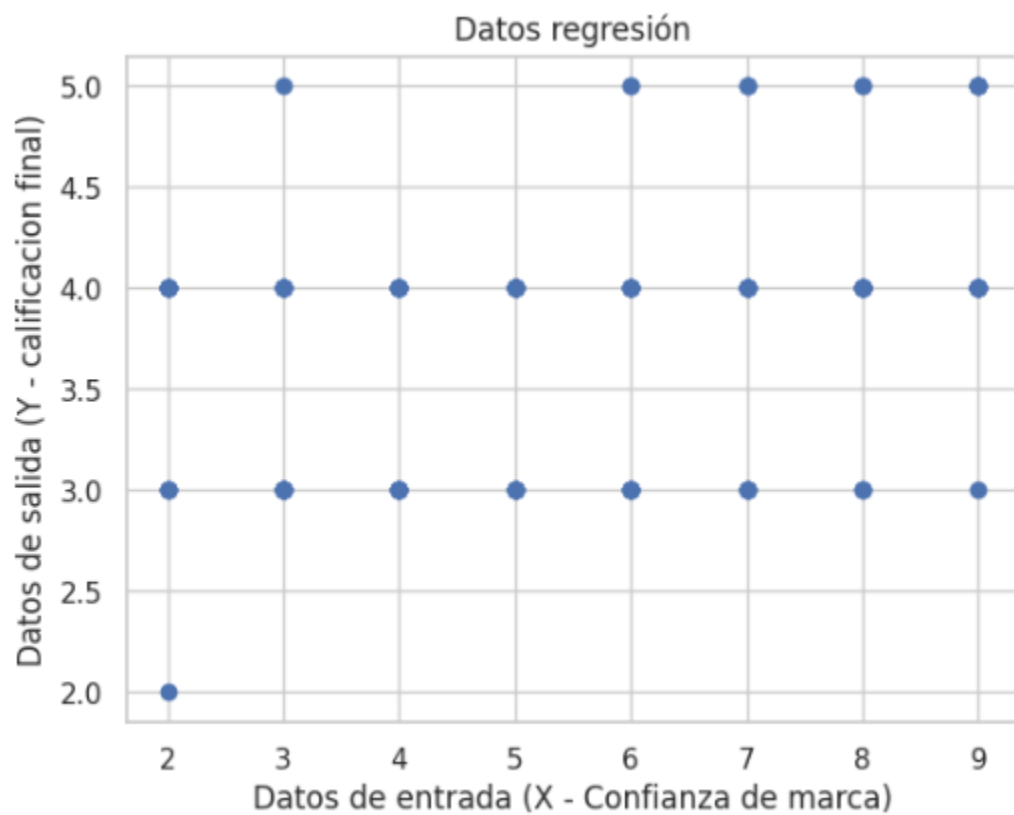
Con este grafico podemos explorar la relación entre dos variables numéricas y evaluar si hay tendencia lineal, exponencial o nula entre variables.

```
import numpy as np
Mis_datos_Array = np.array(Conjunto_Datos)
X_real = Mis_datos_Array[:,9]
Y_real = Mis_datos_Array[:,12]

# Graficando datos
import matplotlib.pyplot as plt
plt.scatter(X_real, Y_real)
plt.title("Datos regresión")
plt.xlabel("Datos de entrada (X - Confianza de marca)")
plt.ylabel("Datos de salida (Y - calificacion final)")
plt.grid(True)
plt.show()
```

El análisis gráfico entre una variable de entrada como la confianza de la marca y una salida como calificación final revela una relación positiva a pesar que la confianza del cliente vario entre valores de 2 “muy insatisfecho” a 9 “casi perfecto, la satisfacción del cliente siempre otorgo calificaciones finales entre 3 “neutral” y 5 “muy satisfecho”, sin alcanzar puntuaciones muy bajas.

En resumen, una baja confianza en la marca no garantiza una mala satisfacción en cliente, pero si contribuye a mantener una percepción positiva moderada. Esta información es clave para las estrategias comerciales.



3.2 Modelo de toma de decisiones

En lugar de aplicar modelos predictivos complejos, se utilizó un enfoque descriptivo y visual para interpretar la información obtenida a través de encuestas aplicadas a consumidores. Para ello, se implementaron herramientas computacionales de Python, como Pandas para el tratamiento de datos y Matplotlib para generar representaciones visuales

El modelo se estructuró en las siguientes etapas:

- 1) Recolección y limpieza de datos: se cargaron los datos desde un archivo Excel con respuestas de clientes sobre experiencia de compra.
- 2) Visualización y análisis gráfico
 - ✓ Tabla de frecuencia para la distribución de variables
 - ✓ Gráfico de pastel para representar porcentajes de nivel de satisfacción
 - ✓ Gráfico de caja y bigotes para identificar valores atípicos y la dispersión de los datos.
 - ✓ Matriz de correlación para explorar relaciones entre variables numéricas.
 - ✓ Gráfico de densidad (KDE) para observar la concentración de respuestas en variables.
- 3) Interpretación de resultados: el análisis visual permitió detectar patrones importantes, como la asociación entre la percepción de la calidad de la ropa, confianza en la marca y calificación final del cliente. También permitió identificar que aspectos presentaban mayor variabilidad o menor satisfacción.

- 4) Apoyo a la toma de decisiones: con base a los resultados gráficos, se generaron recomendaciones dirigidas a mejorar áreas específicas, contribuyendo a mejorar la experiencia del cliente.

3.3 Análisis de desempeño

El desempeño del proyecto fue evaluado tanto desde el punto de vista técnico como analítico. Durante la implementación del análisis de datos y modelos de machine learning, se utilizaron herramientas efectivas como Pandas para la manipulación de los datos y Matplotlib para visualización, lo que permitió una comprensión clara y rápida de las características del conjunto de datos.

Las visualizaciones como la matriz de correlación, los gráficos de densidad, gráfico de caja y bigotes y tablas de frecuencia permitieron identificar zonas críticas, valores extremos y patrones de comportamiento del consumidor, lo que fortaleció la interpretación de los resultados.

En conjunto, el desempeño de la estrategia computacional aplicada fue satisfactoria, ya que permitió no solo confirmar hipótesis teóricas, sino también generar información útil para la toma de decisiones estratégicas en el ámbito comercial.

Este resultado demuestra que el uso de modelos de aprendizaje supervisado, apoyado con análisis exploratorio y visualización de datos, puede ser una herramienta altamente efectiva para conocer y mejorar la experiencia del cliente.

Conclusiones y trabajos futuros

En el actual entorno empresarial, comprender y anticipar la satisfacción del cliente se ha convertido en un componente esencial para la mejora continua y la ventaja competitiva. Este estudio logro abordar dicha necesidad mediante un enfoque analítico basado en datos reales obtenidos de encuestas aplicadas a consumidores de la marca.

En el proyecto se aplicaron técnicas de análisis exploratorio de datos, visualización y modelos de machine learning en Python, lo que permitió identificar los principales factores que inciden en la percepción del cliente.

Las visualizaciones y análisis descriptivos implementados con bibliotecas como Pandas y Matplotlib facilitaron la interpretación de los datos, detectando patrones clave, zonas críticas de insatisfacción y oportunidades de mejora. Este enfoque visual y cuantitativo permitió traducir los resultados técnicos en acciones prácticas para la optimización de la experiencia del cliente.

En resumen, el estudio confirmo que el análisis de datos y el aprendizaje automático supervisado so herramientas poderosas para mejorar la comprensión del comportamiento del consumidor. Al implementar modelos adecuados y validar resultados, se logro desarrollar una base solida para diseñar estrategias orientadas a elevar los niveles de satisfacción, fidelización y retención del cliente en el sector comercial.

Referencias bibliográficas

Smola, A. J., & Schölkopf, B. (2004). *A tutorial on support vector regression*. *Statistics and Computing*, 14(3), 199–222.
<https://doi.org/10.1023/B:STCO.0000035301.49549.88>

Srividya, N., & Akila, B. (2024). *Predicting Customer Satisfaction Score (CSS) for Urban Company Utilizing the K-Nearest Neighbours (KNN) Algorithm: A Machine Learning Approach*. *International Journal of Intelligent Systems and Applications in Engineering*, 12(22s), 467–476.

Loh, W.-Y. (2011). *Classification and Regression Trees*. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 1(1), 14–23.
<https://doi.org/10.1002/widm.8>