Guía de estudio - Preprocesamiento de datos para Data Science

¡Hola! Te damos la bienvenida a esta nueva guía de estudio.

¿En qué consiste esta guía?

Bienvenido a nuestra guía de Data Science, donde explicaremos en detalle la metodología completa de este emocionante campo. Data Science se ha convertido en una disciplina fundamental en la era de la información, permitiendo a las organizaciones extraer información valiosa a partir de datos brutos. En esta guía, nos centraremos en las dos primeras etapas del proceso de Data Science: el Entendimiento de Datos y el Preprocesamiento de Datos.

El Entendimiento y Preprocesamiento de Datos son las bases sobre las que se construye cualquier proyecto de Data Science exitoso. Estas etapas son cruciales para garantizar que los datos que utilizamos sean confiables, relevantes y estén listos para ser analizados. Aquí es donde se realiza una inmersión profunda en los datos para comprender su naturaleza y calidad, y se aplican técnicas para limpiarlos y transformarlos en una forma adecuada para el análisis.

En esta guía, explicaremos cada paso de las etapas de Entendimiento y Preprocesamiento de Datos en profundidad. Aprenderás cómo abordar un conjunto de datos desconocido y cómo aplicar técnicas para la exploración, limpieza y transformación de datos, que nos permita realizar una adecuada selección de los mismos.

Nuestro objetivo es proporcionar una comprensión sólida de estas etapas fundamentales que servirán como base para proyectos más avanzados de Data Science.

Estamos emocionados de acompañarte en este viaje para convertirte en un experto en el arte de comprender y preparar datos para el análisis. ¡Comencemos con el fascinante mundo del Entendimiento y Preprocesamiento de Datos!

¡Vamos con todo!

Tabla de contenidos

¡Comencemos!

Metodología para Data Science

Las metodologías en Data Science son marcos estructurados y enfoques sistemáticos diseñados para guiar a los profesionales a través de cada etapa de un proyecto de análisis de datos. Estas metodologías se han convertido en faros confiables en el vasto mar de datos, proporcionando direcciones y mejores prácticas para obtener resultados precisos y significativos.

1. **CRISP-DM (Cross-Industry Standard Process for Data Mining):** Esta metodología, ampliamente utilizada, abarca desde la comprensión del negocio y los datos hasta la implementación de modelos y la evaluación de resultados. Su enfoque en capas asegura una comprensión completa y una toma de decisiones bien fundamentada.

2. **TDSP (Team Data Science Process):** Desarrollada por Microsoft, TDSP es un marco colaborativo que pone un énfasis especial en el trabajo en equipo y la colaboración en proyectos de Data Science. Ofrece un conjunto de guías, prácticas recomendadas y plantillas para todas las fases del ciclo de vida del proyecto.

Las metodologías en Data Science son esenciales por varias razones clave:

- **Estructura y Enfoque:** Proporcionan una estructura sólida para proyectos, lo que garantiza que no se pasen por alto etapas críticas y se mantenga un enfoque claro en los objetivos.
- **Reproducibilidad:** Facilitan la reproducibilidad al estandarizar el proceso, lo que permite a otros científicos de datos comprender y replicar los resultados.
- **Optimización de Recursos:** Ayudan a optimizar el uso de recursos al enfocarse en las tareas que tienen el mayor impacto en los resultados finales.
- **Comunicación Efectiva:** Facilitan la comunicación entre equipos interdisciplinarios al proporcionar un lenguaje común y una estructura compartida.
- **Mejores Prácticas:** Incorporan mejores prácticas de la industria, lo que reduce la probabilidad de errores y aumenta la confiabilidad de los resultados.

A continuación nos adentraremos en la metodología CRISP DM que es la más utilizada en el campo del data science.

CRISP DM

La metodología CRISP-DM (Cross-Industry Standard Process for Data Mining) es un marco ampliamente utilizado para llevar a cabo proyectos de minería de datos de manera estructurada y eficiente. Diseñada para ser aplicada en diversos sectores de la industria, la Metodología CRISP-DM se ha convertido en un estándar de facto en el campo de la minería de datos.

La metodología CRISP-DM se divide en seis fases principales, cada una de las cuales representa una etapa crítica en un proyecto de minería de datos:

a. **Comprensión del Negocio (Business Understanding)**: En esta fase inicial, se busca comprender los objetivos del negocio y cómo la minería de datos puede contribuir a alcanzarlos. Se establecen metas claras para el proyecto y se definen los criterios de éxito.
b. **Comprensión de los Datos (Data Understanding):** Aquí, se recopila y explora el conjunto de datos disponible. Se identifican las fuentes de datos, se realiza una evaluación inicial de su calidad y se exploran las características de los datos.
c. **Preparación de los Datos (Data Preparation):** En esta fase, se llevan a cabo tareas de limpieza, transformación y selección de datos para prepararlos para el modelado. Esto incluye la codificación de variables, la imputación de valores faltantes y la creación de nuevas características si es necesario.
d. **Modelado (Modeling):** En esta etapa, se seleccionan y aplican técnicas de modelado de datos, como regresión, clasificación o agrupación, según los objetivos del proyecto. Se ajustan y validan los modelos para lograr un rendimiento óptimo.
e. **Evaluación (Evaluation):** Aquí se evalúa el rendimiento de los modelos desarrollados en términos de las metas comerciales establecidas en la fase de Comprensión del Negocio. Se pueden utilizar métricas como la precisión, el F1-score o el error cuadrático medio (MSE) según el tipo de problema.
f. **Despliegue (Deployment):** En esta última fase, se implementan los modelos en un entorno operativo para su uso en el negocio. Esto puede implicar la integración en sistemas existentes o la creación de aplicaciones específicas.

CRISP-DM es un enfoque iterativo, lo que significa que las fases no son necesariamente lineales. Puede ser necesario retroceder y revisar fases anteriores a medida que se adquiere una comprensión más profunda del problema o se recopila nueva información.

¿Por qué CRISP-DM?

1. **Estructura Clara:** CRISP-DM proporciona una estructura clara y bien definida para los proyectos de minería de datos, lo que facilita la comunicación y la gestión del proyecto.

2. **Orientación al Negocio:** La metodología se enfoca en la comprensión de los objetivos comerciales, lo que asegura que los resultados sean relevantes para la organización.

3. **Flexibilidad:** CRISP-DM es lo suficientemente flexible como para adaptarse a una variedad de problemas y tipos de datos.

Aunque CRISP-DM es un marco sólido, algunas organizaciones optan por agregar elementos específicos, como la ética de datos y la privacidad, para abordar cuestiones adicionales en la minería de datos.

En resumen, la Metodología CRISP-DM es una guía efectiva para llevar a cabo proyectos de minería de datos desde la comprensión del negocio hasta la implementación de modelos. Su enfoque estructurado y su capacidad para adaptarse a diferentes contextos hacen que sea una herramienta valiosa en el mundo de la minería de datos.

Entendimiento del Negocio (Business Understanding)

La fase de "Comprender el Negocio" (Business Understanding) en la metodología CRISP-DM es crucial para establecer una base sólida para tu proyecto de minería de datos. Aquí hay algunas subetapas clave y consejos para abordar esta fase de manera efectiva:

Subetapas

1. **Establecer Objetivos Claros**

- Define claramente los objetivos comerciales que deseas lograr con el proyecto de minería de datos. ¿Qué problema deseas resolver o qué oportunidad deseas aprovechar?
- Asegúrate de que los objetivos sean específicos, medibles, alcanzables, relevantes y con un plazo definido (conocidos como criterios SMART).

2. **Comprensión del Contexto**

- Investiga a fondo la industria y el mercado en los que opera tu organización. Comprende cómo se relaciona tu proyecto con las tendencias y los desafíos actuales.
- Identifica a las partes interesadas clave dentro de la organización y consulta sus opiniones y preocupaciones. Comprende sus expectativas y necesidades.

3. **Evaluación de Recursos:**

- Evalúa los recursos necesarios para el proyecto, incluidos datos, personal, hardware y software.
- Asegúrate de que haya un presupuesto adecuado para respaldar el proyecto y que los recursos estén disponibles en el momento adecuado.

4. **Definición de Éxito**

- Establece criterios claros para medir el éxito del proyecto. ¿Qué indicadores clave de rendimiento (KPI) utilizarás para evaluar los resultados?

- Considera el impacto financiero y estratégico de los resultados. ¿Cómo se traducirán los resultados en términos de beneficios comerciales?

5. **Planificación Inicial**

- Crea un plan de alto nivel que describa las etapas principales del proyecto, los plazos y los entregables esperados.

- Define las métricas clave que se utilizarán para evaluar el progreso a lo largo del proyecto.

6. **Documentación**

Documenta toda la información recopilada en esta fase. Esto incluye los objetivos, el contexto del negocio, las partes interesadas, los recursos necesarios y los criterios de éxito.

Consejos Adicionales

Fomenta la comunicación abierta y regular con las partes interesadas. Mantenlos informados sobre el progreso del proyecto y asegúrate de que sus comentarios sean tenidos en cuenta.

- Sé flexible. A medida que avanzas en la fase de comprensión del negocio, es posible que debas ajustar los objetivos o las expectativas en función de lo que descubras.

- No subestimes la importancia de esta fase. Una comprensión incompleta o incorrecta de los objetivos comerciales puede llevar a resultados insatisfactorios en las etapas posteriores del proyecto.

- Colabora estrechamente con expertos en el dominio del negocio. Comprender los detalles del negocio es esencial para desarrollar soluciones efectivas.

- La fase de "Comprender el Negocio" sienta las bases para el éxito de tu proyecto de minería de datos. Invertir tiempo y esfuerzo en esta etapa inicial puede ahorrar problemas y asegurar que tu trabajo tenga un impacto positivo en la organización.

Entendimiento de Datos (Data Understanding)

La fase de "Entendimiento de los Datos" (Data Understanding) en la metodología CRISP-DM es esencial para comprender los datos con los que trabajarás en tu proyecto de minería de datos. Aquí hay algunas subetapas clave y consejos para abordar esta fase de manera efectiva:

Subetapas

1. **Recopilación de Datos**

- Identificación de Fuentes: Enumera todas las fuentes de datos disponibles, ya sean bases de datos, archivos, sensores u otras fuentes.
- Adquisición de Datos: Obtén acceso a los datos y recogelos de acuerdo con los requisitos del proyecto. Esto puede incluir la descarga de datos, la conexión a bases de datos, la adquisición de datos en tiempo real, etc.

2. **Evaluación de la Calidad de los Datos:**

- Limpieza de Datos: Identifica y trata los valores atípicos, valores faltantes y otros problemas de calidad de datos.
- Detección de anomalías: Utiliza técnicas de detección de anomalías para identificar posibles errores en los datos.

3. **Exploración Inicial:**

- Exploración de Datos: Realiza una exploración inicial de los datos para comprender su estructura y contenido. Esto incluye la identificación de variables, su tipo y su calidad.
- Resumen de Datos: Genera estadísticas descriptivas, como la media, la mediana, la desviación estándar y los percentiles, para obtener una idea general de los datos.

4. **Exploración Detallada:**

- Análisis Exploratorio: Realiza un análisis más detallado de las relaciones entre las variables, utilizando gráficos y estadísticas más avanzadas.
- Visualización de Datos: Crear visualizaciones efectivas para representar los datos y resaltar patrones o tendencias importantes.

5. **Muestreo de Datos:**

- Selección de Muestra: Si los datos son muy grandes, considera la posibilidad de trabajar con una muestra representativa en lugar de la población completa.

- Validación de Muestra: Asegúrate de que la muestra sea representativa y que no introduzca sesgos en el análisis.

6. **Documentación de Datos:**

- Documenta el Diccionario de Datos: Crea una descripción detallada de cada variable, incluidos su nombre, tipo, significado y cualquier transformación realizada.
- Registros de Origen: Documenta el origen de los datos, incluyendo detalles como la fecha de adquisición y las personas responsables.

Consejos Adicionales

Comprende completamente el contexto de los datos. Conocer el dominio de negocio te ayudará a interpretar mejor los datos y a realizar análisis más significativos.

La calidad de los datos es crítica. Dedica tiempo a la limpieza y la detección de problemas de calidad antes de avanzar a las etapas posteriores.

Colabora estrechamente con expertos en el dominio del negocio y con el equipo de adquisición de datos. Ellos pueden proporcionar información valiosa sobre la calidad y el significado de los datos.

Utiliza herramientas de visualización y análisis de datos para explorar y comprender mejor los patrones y relaciones en los datos.

No subestimes la importancia de esta fase. Un entendimiento sólido de los datos es fundamental para tomar decisiones informadas en las etapas posteriores del proyecto.

La fase de "Entendimiento de los Datos" sienta las bases para el análisis y la modelización posteriores. Invertir tiempo y esfuerzo en esta etapa puede garantizar que los datos sean una base sólida para la toma de decisiones en tu proyecto de minería de datos.

Identificando Outliers

La detección de valores atípicos (outliers) es un paso importante en la preparación de datos, ya que los outliers pueden distorsionar los modelos de machine learning y sesgar los resultados. Aquí tienes algunos métodos comunes para detectar outliers, junto con una breve descripción de cómo funcionan:

1. **Método de IQR (Rango Intercuartil):**
    a. El IQR es la diferencia entre el tercer cuartil (Q3) y el primer cuartil (Q1) de un conjunto de datos.

   b. Los valores atípicos se identifican como aquellos que están por debajo de Q1 - 1.5 * IQR o por encima de Q3 + 1.5 * IQR.
   c. Este método se basa en la distribución de los datos y es robusto frente a valores extremos.

2. **Método de Z-Score (Puntuación Z):**
   a. Calcula la puntuación Z para cada punto de datos, que mide cuántas desviaciones estándar un punto está del promedio.
   b. Los valores atípicos suelen definirse como aquellos con un valor absoluto de Z mayor que un umbral (por ejemplo, 2 o 3).
   c. Este método asume que los datos siguen una distribución normal.

3. **Método de Caja y Bigotes (Box Plot):**
   a. Los valores atípicos se identifican visualmente en un gráfico de caja y bigotes.
   b. Los puntos que están por encima o por debajo de los bigotes se consideran outliers.
   c. Es útil para una rápida inspección visual de la distribución de los datos.

4. **Métodos basados en Machine Learning:**
   a. Algoritmos de clustering como el DBSCAN pueden detectar valores atípicos al agrupar datos y considerar puntos aislados como outliers.
   b. Modelos de regresión o clasificación pueden identificar valores atípicos observando los residuos.

5. **Otros:**
   a. Hay varios métodos para encontrar outliers, otra opción son algoritmos de detección de anomalías como svm one class o isolation forest.

Es importante destacar que la elección del método de detección de outliers debe basarse en la naturaleza de los datos y el problema en cuestión. Además, la eliminación o el tratamiento de outliers debe realizarse con precaución y teniendo en cuenta el impacto en los resultados finales. En algunos casos, los outliers pueden contener información importante o ser legítimos, por lo que no siempre deben eliminarse.

¡Manos a la obra! - Identificando Outliers

Ahora estudiaremos la parte práctica de algunos de los métodos para calcular outliers con python.

IQR

Lo primero que vamos a hacer es importar las librerías necesarias y cargar los datos.

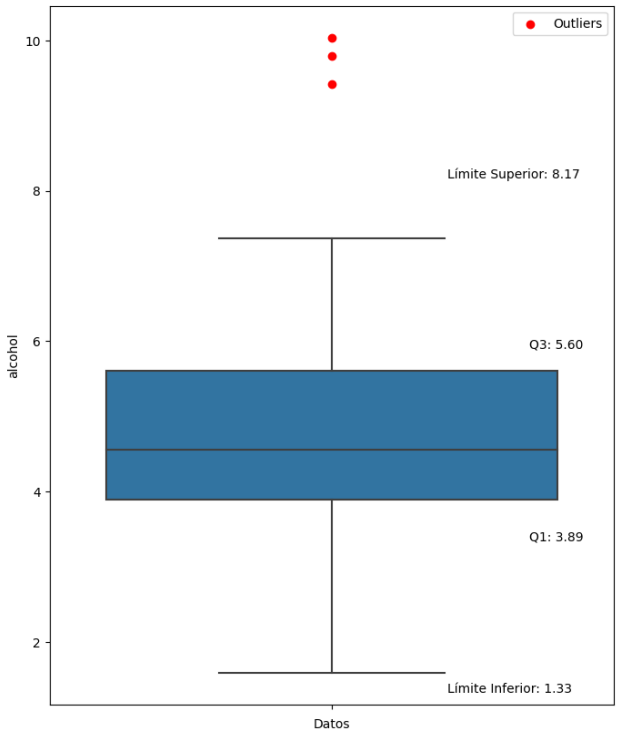
```
import seaborn as sns
import matplotlib.pyplot as plt
import pandas as pd

# Cargar un DataFrame de ejemplo desde Seaborn
df = sns.load_dataset('car_crashes')
df.head()
```

	total	speeding	alcohol	not_distracted	no_previous	ins_premium	ins_losses	abbrev
0	18.8	7.332	5.640	18.048	15.040	784.55	145.08	AL
1	18.1	7.421	4.525	16.290	17.014	1053.48	133.93	AK
2	18.6	6.510	5.208	15.624	17.856	899.47	110.35	AZ
3	22.4	4.032	5.824	21.056	21.280	827.34	142.39	AR
4	12.0	4.200	3.360	10.920	10.680	878.41	165.63	CA

Ahora vamos a generar una función que al entregarle un dataset y una columna utiliza el método de IQR para detectar los outliers y generar un gráfico de boxplot.

```
def plot_boxplot_with_outliers(data, column_name, outlier_color='red',
factor=1.5):
    # Calcular los estadísticos clave
    Q1 = data[column_name].quantile(0.25)
    Q3 = data[column_name].quantile(0.75)
    IQR = Q3 - Q1

    # Calcular los límites para los valores atípicos
    lower_bound = Q1 - factor * IQR
    upper_bound = Q3 + factor * IQR



    # Crear un gráfico de diagrama de caja (boxplot)
    plt.figure(figsize=(8,10))
    sns.boxplot(y=data[column_name], showfliers=False)
    # plt.title(f'Diagrama de Caja de {column_name}')
    plt.ylabel(column_name)
    plt.xlabel('Datos')

    # Resaltar los valores atípicos en un color diferente
```

```
    outliers = data[(data[column_name] < lower_bound) | (data[column_name] >
upper_bound)]
    plt.scatter(y=outliers[column_name], x=[0]*len(outliers),
color=outlier_color, label='Outliers')

    # Anotar los valores de límites y cuartiles en el gráfico
    plt.text(0.35, Q1-0.55, f'Q1: {Q1:.2f}', fontsize=10)
    plt.text(0.35, Q3+0.3, f'Q3: {Q3:.2f}', fontsize=10)
    plt.text(0.205, lower_bound, f'Límite Inferior: {lower_bound:.2f}',
fontsize=10)
    plt.text(0.205, upper_bound, f'Límite Superior: {upper_bound:.2f}',
fontsize=10)

    plt.legend()
    plt.show()
    return outliers


plot_boxplot_with_outliers(df, 'alcohol', outlier_color='red', factor=1.5)
```



Boxplot con outliers según método IQR

Se observa en el gráfico los cuartiles Q1 y Q3 dentro del cajón, como los límites superior e inferior según el método IQR, lo que permite destacar los valores alejados que son considerados Outliers y se marcan en rojo.

Z Score

El Z-score es una medida estadística que se utiliza para identificar valores atípicos en un conjunto de datos. Se calcula tomando la diferencia entre un valor específico y la media del conjunto de datos, y luego dividiendo esta diferencia por la desviación estándar. Matemáticamente, el Z-score de un valor x se calcula de la siguiente manera:

$$Z = \frac{x - \mu}{\sigma}$$

Donde:

- Z es el Z-score
- x es el valor que se está evaluando
- μ es la media del conjunto de datos
- σ es la desviación estándar del conjunto de datos.

Utilizando los mismos datos vamos a generar una función que detecte los outlier utilizando este método y un umbral de 2 y una función que reciba este indicador de outliers y los grafique para identificar los valores anómalos.

```
def detect_outliers_zscore(data, threshold=2):
    # Calcular el Z-score para cada punto de datos
    z_scores = (data - np.mean(data)) / np.std(data)

    # Encontrar valores atípicos basados en el umbral
    outliers = np.abs(z_scores) > threshold

    return outliers


def plot_outliers(data, outliers):
    # Crear un gráfico de dispersión para los valores normales
    plt.figure(figsize=(10, 6))
    plt.scatter(data[~outliers], [1] * len(data[~outliers]), label='Valores
Normales', color='blue', s=50)

    # Crear un gráfico de dispersión para los valores atípicos
    plt.scatter(data[outliers], [1] * len(data[outliers]), label='Valores
Atípicos', color='red', marker='x', s=100)
```

```
    plt.title('Detección de Outliers con Z-score (tresh=2)')
    plt.xlabel('Datos')
    plt.yticks([])
    plt.legend()
    plt.grid(True)
    plt.show()

outliers = detect_outliers_zscore(df['alcohol'], threshold=2)
plot_outliers(df['alcohol'], outliers)
```

Gráfico de outliers con método de Z-Score

Es importante destacar que los outliers detectados por los métodos no necesariamente tienen que ser eliminados o que sean anómalos, pero son indicadores de valores alejados de la distribución de la muestra, al final se debe decidir caso a caso según el problema que se desee resolver y la naturaleza de la variable.

Correlación entre variables

La medición de la correlación entre variables depende del tipo de datos y el contexto del análisis. A continuación se muestran técnicas de correlación que posiblemente se pueden utilizar.

Variable numérica v/s variable numérica

Coeficiente de Correlación de Pearson (r)

Mide la relación lineal entre dos variables numéricas. Varía entre -1 y 1, donde 1 indica una correlación positiva perfecta, -1 indica una correlación negativa perfecta y 0 indica ausencia de correlación.

Pros

- Fácil de entender y ampliamente utilizada.
- Proporciona una medida cuantitativa de la relación lineal.
- Varía entre -1 y 1, lo que facilita la interpretación.

Contras

- Solo mide relaciones lineales, por lo que no detecta relaciones no lineales.
- Sensible a los valores atípicos (outliers).
- No captura relaciones de dependencia complejas.
- Recuerda que la correlación de Pearson es una herramienta útil, pero no siempre es suficiente para comprender la relación entre variables, especialmente cuando esta relación es no lineal o involucra más factores. En esos casos, es importante complementar el análisis con otras técnicas estadísticas y gráficas.

Coeficiente de Correlación de Spearman

Similar al coeficiente de Pearson, pero no asume una relación lineal. Es útil cuando las relaciones no son lineales o cuando hay valores atípicos.

Coeficiente de Correlación de Kendall

Mide la correlación entre dos variables en función de la concordancia y discordancia de los pares de datos en comparación con el rango completo de datos. Es útil cuando se tienen datos ordenados o clasificados.

Variable numérica v/s variable categórica

Chi cuadrado

Se utiliza para medir la asociación entre dos variables categóricas en forma de tablas de contingencia. Puede ser calculado como la raíz cuadrada del estadístico chi-cuadrado dividido por el tamaño de la muestra.

Pros

- Detecta relaciones entre variables categóricas.
- Es fácil de entender y de aplicar.
- No asume normalidad en los datos.

Contras

- Solo es aplicable a variables categóricas.
- No proporciona información sobre la fuerza o la dirección de la relación.
- Puede no ser adecuado para tablas de contingencia con frecuencias pequeñas.
- El test de chi-cuadrado es útil en diversos campos, como la epidemiología, la investigación social, la biología y otros, cuando se necesita evaluar la independencia o la asociación entre dos variables categóricas.

Coeficiente de Cramer's V

Es una versión normalizada del coeficiente de contingencia y se utiliza para medir la fuerza de la asociación entre dos variables categóricas. Varía entre 0 y 1, donde 1 indica una asociación perfecta.

Variable categórica v/s variable categórica

ANOVA (Análisis de Varianza)

Se utiliza para medir la relación entre una variable numérica y una variable categórica con más de dos grupos. Calcula si hay diferencias significativas en la variable numérica entre los grupos de la variable categórica.

Pros

- Puede manejar múltiples grupos a la vez.
- Proporciona una prueba estadística sólida para determinar las diferencias significativas entre grupos.

Contras

- Supone que las varianzas son iguales entre grupos (homocedasticidad).
- No indica cuáles grupos son diferentes entre sí; se necesita una prueba post hoc para eso.

Coeficiente de Correlación Punto-Biserie (Point-Biserial Correlation)

Mide la correlación entre una variable numérica y una variable binaria. Puede ser visto como una versión especializada del coeficiente de Pearson para esta situación.

Coeficiente de Correlación de Eta (η)

Mide la relación entre una variable numérica y una variable categórica ordinal (categorías ordenadas). Es una variante del coeficiente de contingencia.

Recuerda que la elección de la métrica de correlación depende del tipo de datos que estés analizando y de tus objetivos específicos. Además, es importante recordar que la **correlación no implica causalidad**; sólo indica relaciones estadísticas entre variables.

Siempre es necesario interpretar los resultados en el contexto del problema que estás abordando.

¡Manos a la obra! - Midiendo la relación entre variables

Ahora analizaremos el cálculo de correlaciones con Python

Correlación de Pearson

Para Calcular la correlación de Pearson se realiza el siguiente cálculo:

$$r = \frac{\sum(X_i - \bar{X})(Y_i - \bar{Y})}{\sqrt{\sum(X_i - \bar{X})^2 \sum(Y_i - \bar{Y})^2}} = \frac{COV(X,Y)}{VAR(X)VAR(Y)}$$

Donde:

r es la correlación de pearson

X_i, Y_i son los valores individuales de las 2 variables

$\bar{X}, \bar{Y}$ son las medias de las 2 variables

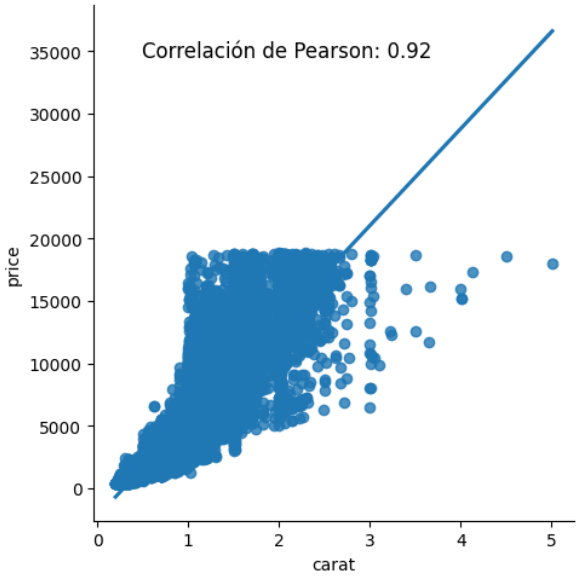
Para calcular primero vamos a importar las librerías y el dataset necesario

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

# Cargar un DataFrame de ejemplo desde Seaborn
df = sns.load_dataset('diamonds')
print(df.shape)
df.head()
```

	carat	cut	color	clarity	depth	table	price	x	y	z
0	0.23	Ideal	E	SI2	61.5	55.0	326	3.95	3.98	2.43
1	0.21	Premium	E	SI1	59.8	61.0	326	3.89	3.84	2.31
2	0.23	Good	E	VS1	56.9	65.0	327	4.05	4.07	2.31
3	0.29	Premium	I	VS2	62.4	58.0	334	4.20	4.23	2.63
4	0.31	Good	J	SI2	63.3	58.0	335	4.34	4.35	2.75

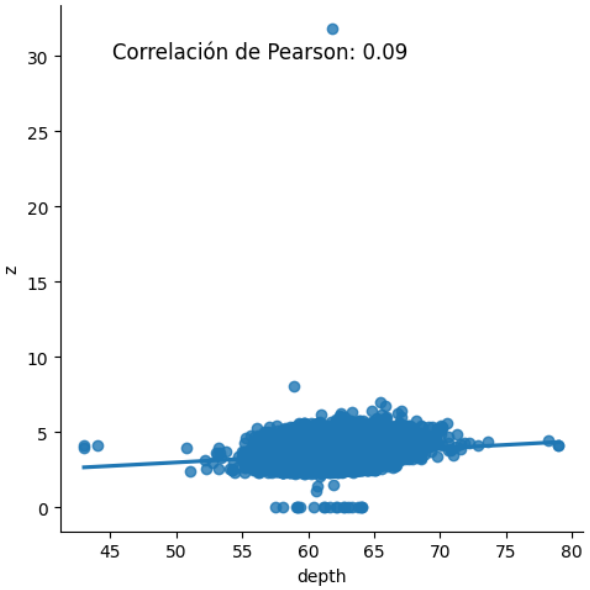
Ahora vamos a generar una función que utilizando el método .corr() de pandas vamos a calcular la correlación de pearson y la vamos a anotar en un gráfico de dispersión entre las 2 variables seleccionadas.

```
def plot_lmplot_with_pearson(dataframe, variable1, variable2):
    # Calcula la correlación de Pearson
    corr = dataframe[variable1].corr(dataframe[variable2])

    # Crea el gráfico de dispersión con una recta de regresión lineal
    sns.lmplot(x=variable1, y=variable2, data=dataframe)

    # Anota la correlación en el gráfico
    plt.text(0.1, 0.9, f'Correlación de Pearson: {corr:.2f}',
transform=plt.gca().transAxes, fontsize=12)

    # Muestra el gráfico
    plt.show()

plot_lmplot_with_pearson(df, 'carat', 'price')
```



Se observa la correlación entre la variable “carat” que corresponde al peso de los diamantes y la variable “price” que es el precio de los diamantes. La correlación de pearson es de 0.92, lo que corresponde a una relación fuerte entre ambas variables, que tiene mucho sentido dada la naturaleza de las variables.

```
plot_lmplot_with_pearson(df, 'depth', 'z')
```



Ahora observamos la correlación entre la profundidad del diamante y la variable z, donde se observa un coeficiente de 0.09, lo que corresponde a una relación débil entre ambas variables.

Chi Cuadrado

El Test de Chi-cuadrado es una prueba estadística que se utiliza para determinar si existe una asociación significativa entre dos variables categóricas en una tabla de contingencia. Su nombre proviene de la distribución chi-cuadrado, que es la distribución de probabilidad que se utiliza en este test.

1. **Creación de la tabla de contingencia:** Primero, se crea una tabla que muestra la distribución conjunta de dos variables categóricas. Esta tabla muestra cuántas observaciones pertenecen a cada combinación de categorías de las dos variables.

2. **Cálculo de las frecuencias esperadas:** Se calculan las frecuencias esperadas para cada celda de la tabla de contingencia bajo la hipótesis nula de independencia entre las dos variables. Esto se hace multiplicando las sumas marginales de las filas y columnas de cada celda y dividiendo por el tamaño total de la muestra. Las frecuencias esperadas representan lo que se esperaría en cada celda si las dos variables fueran independientes.

3. **Cálculo de la estadística de chi-cuadrado:** La estadística de chi-cuadrado (χ^2) se calcula utilizando la fórmula:

$$\chi^2 = \sum \frac{(O-E)^2}{E}$$

Donde:

- χ^2 es el estadístico chi cuadrado
- O Es la frecuencia observada en cada celda de la tabla de contingencia.
- E representa las frecuencias esperadas en cada celda de la tabla de contingencia.

4. **Grados de libertad (df):** Los grados de libertad se calculan como (filas-1)*(columnas-1), donde "filas" es el número de categorías en una variable y "columnas" es el número de categorías en la otra variable.

5. **Valor p:** Una vez calculada la estadística de chi-cuadrado y los grados de libertad, se busca en una tabla de la distribución chi-cuadrado para determinar el valor p correspondiente. El valor p indica la probabilidad de obtener una estadística de chi-cuadrado tan extrema como la observada bajo la hipótesis nula de independencia entre las variables.

6. **Prueba de hipótesis:** Finalmente, se compara el valor p con un nivel de significancia predefinido (por ejemplo, 0.05). Si el valor p es menor que el nivel de significancia, se rechaza la hipótesis nula y se concluye que existe una asociación significativa entre las dos variables categóricas. Si el valor p es mayor que el nivel de significancia, no se rechaza la hipótesis nula y se concluye que no hay evidencia suficiente para afirmar una asociación significativa.

El objetivo del test de chi-cuadrado es determinar si existe una asociación significativa entre las dos variables categóricas en lugar de establecer una relación causal.

La interpretación es:

1. Si el valor de chi-cuadrado calculado es significativo, esto sugiere que existe una relación entre las dos variables.

2. Si el valor de chi-cuadrado es pequeño y no significativo, esto sugiere que las dos variables son independientes entre sí.

Ahora vamos a cargar las librerías necesarias y un dataset de prueba para calcular el valor de chi cuadrado paso a paso:

```
import seaborn as sns
import matplotlib.pyplot as plt
import pandas as pd
from scipy.stats import chi2
```

```
from scipy.stats import chi2_contingency

data = {
    'Género': ['Masculino', 'Masculino', 'Femenino', 'Femenino'],
    'Preferencia': ['Acción', 'Romance', 'Acción', 'Romance'],
    'Frecuencia': [12, 8, 5, 15]
}

df = pd.DataFrame(data)
tabla_contingencia = pd.crosstab(df['Género'], df['Preferencia'],
df['Frecuencia'], aggfunc='sum')
tabla_contingencia
```

Preferencia	Acción	Romance
Género		
Femenino	5	15
Masculino	12	8

A continuación haremos paso a paso lo descrito en la descripción anterior para calcular el estadístico chi cuadrado:

```
# Paso 1: Calcular la tabla de contingencia
tabla_contingencia = pd.crosstab(df['Género'], df['Preferencia'],
df['Frecuencia'], aggfunc='sum')

# Paso 2: Calcular las frecuencias esperadas
expected_data = np.outer(tabla_contingencia.sum(axis=1),
tabla_contingencia.sum(axis=0)) / tabla_contingencia.values.sum()
expected_data = pd.DataFrame(expected_data, index=['Femenino', 'Masculino'],
columns=['Acción', 'Romance'])

# Paso 3: Calcular la estadística de chi-cuadrado # Aca se puede aplicar un
factor de correcion
chi_squared_statistic = (tabla_contingencia - expected_data)**2 /
expected_data
chi_squared = chi_squared_statistic.sum().sum()

# Paso 4: Calcular los grados de libertad
degrees_of_freedom = (tabla_contingencia.shape[0] - 1) *
(tabla_contingencia.shape[1] - 1)

# Paso 5: Calcular el valor p
p_value = 1 - chi2.cdf(chi_squared, degrees_of_freedom)
```

```
# Paso 6: Realizar la prueba de hipótesis
alpha = 0.05  # Nivel de significancia
if p_value < alpha:
    conclusion = "Se rechaza la hipótesis nula, hay una asociación significativa."
else:
    conclusion = "No se rechaza la hipótesis nula, no hay evidencia de asociación significativa."
```

Ahora imprimimos los resultados:

```
# Imprimir resultados
print(f"Tabla de contingencia:\n{tabla_contingencia}\n")
print(f"Frecuencias esperadas:\n{expected_data}\n")
print(f"Tabla de chi cuadrado:\n{chi_squared_statistic}\n")
print(f"Estadística de chi-cuadrado: {chi_squared}\n")
print(f"Grados de libertad: {degrees_of_freedom}\n")
print(f"Valor p: {p_value}\n")
print(f"Conclusión: {conclusion}")
```

```
Tabla de contingencia:
Preferencia  Acción  Romance
Género
Femenino          5       15
Masculino        12        8

Frecuencias esperadas:
           Acción  Romance
Femenino      8.5     11.5
Masculino     8.5     11.5

Tabla de chi cuadrado:
Preferencia    Acción   Romance
Género
Femenino     1.441176  1.065217
Masculino    1.441176  1.065217

Estadística de chi-cuadrado: 5.012787723785166

Grados de libertad: 1

Valor p: 0.025160759200408833

Conclusión: Se rechaza la hipótesis nula, hay una asociación significativa.
```

Observamos que el p valor es menor a 0.05, por lo que se rechaza la hipótesis nula de la independencia entre las variables, por lo que hay una asociación significativa.

Vamos a contrastar el valor con la función chi2_contingency de stats.

```
chi2_stat, p, _, _ = chi2_contingency(tabla_contingencia, correction=False)

print(f"Valor de Chi-cuadrado: {chi2_stat}")
print(f"Valor p: {p}")
```

```
Valor de Chi-cuadrado: 5.012787723785166
Valor p: 0.025160759200408785
```

Vemos como el valor que calculamos anteriormente coincide con el calculado con la función.

Test de ANOVA

El Análisis de Varianza, o ANOVA, es una técnica estadística utilizada para analizar la variación en datos de manera que puedas determinar si existen diferencias significativas entre las medias de tres o más grupos o poblaciones. Se basa en comparar las varianzas entre grupos y dentro de grupos para tomar decisiones sobre si las diferencias observadas son estadísticamente significativas.

El cálculo paso a paso de ANOVA es el siguiente:

1. **Calcular la media de cada grupo:** Comienza calculando la media de cada grupo que deseas comparar.

2. **Calcular la media total:** Luego, calcula la media general de todos los datos, independientemente del grupo al que pertenezcan.

3. **Calcular la suma de cuadrados entre grupos (SSB):** Esta suma de cuadrados mide la variación entre las medias de los grupos y se calcula como la suma de las diferencias al cuadrado entre la media de cada grupo y la media total, ponderadas por el tamaño de cada grupo.

4. **Calcular la suma de cuadrados dentro de grupos (SSW):** Esta suma de cuadrados mide la variación dentro de cada grupo y se calcula como la suma de las diferencias al cuadrado entre cada valor de datos y la media de su grupo respectivo.

5. **Calcular el grado de libertad entre grupos (DFB):** Este valor se refiere al número de grupos menos uno (DFB = número de grupos - 1).

6. **Calcular el grado de libertad dentro de grupos (DFW):** Este valor se refiere al número total de observaciones menos el número de grupos (DFW = número total de observaciones - número de grupos).

7. **Calcular la media cuadrática entre grupos (MSB):** Se obtiene dividiendo la SSB por los DFB (MSB = SSB / DFB).

8. **Calcular la media cuadrática dentro de grupos (MSW):** Se obtiene dividiendo la SSW por los DFW (MSW = SSW / DFW).

9. **Calcular el valor F:** El valor F es la relación entre la MSB y la MSW (F = MSB / MSW).

10. **Calcular el valor p:** Utiliza el valor F y los grados de libertad para calcular el valor p utilizando una tabla de distribución F o una función en Python.

11. **Comparar el valor p con un nivel de significancia (alfa):** Si el valor p es menor que alfa, se rechaza la hipótesis nula y se concluye que al menos un grupo es significativamente diferente de los demás.

La interpretación de ANOVA es la siguiente:

- **Hipótesis Nula (H0)**: Supone que no hay diferencias significativas entre los grupos.

- **Hipótesis Alternativa (H1):** Sugiere que al menos un grupo es significativamente diferente de los demás.

Ahora vamos a realizar un ejemplo en python, donde primero importamos las librerías importantes y el dataset a utilizar, un dataset de propinas.

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

# Cargar el conjunto de datos de muestra de Seaborn
data = sns.load_dataset("tips")
data.head()
```

	total_bill	tip	sex	smoker	day	time	size
0	16.99	1.01	Female	No	Sun	Dinner	2
1	10.34	1.66	Male	No	Sun	Dinner	3
2	21.01	3.50	Male	No	Sun	Dinner	3
3	23.68	3.31	Male	No	Sun	Dinner	2
4	24.59	3.61	Female	No	Sun	Dinner	4

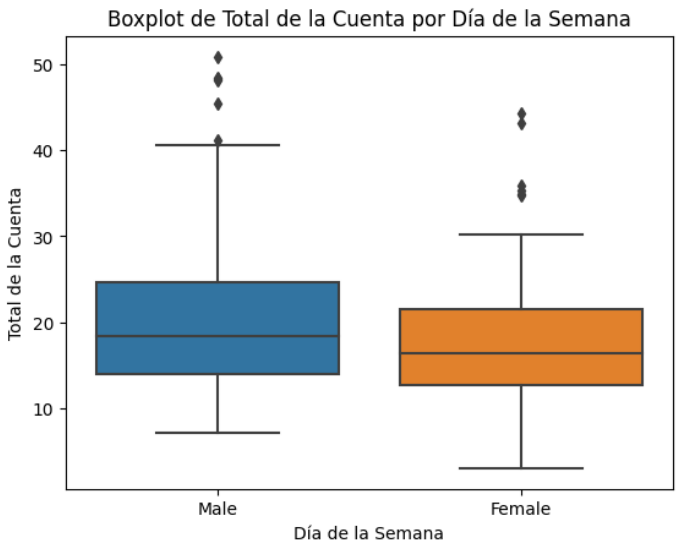
Ahora separaremos en una lista la variable del total de la cuenta (total_bill) para diferentes categorías de 'sex' y 'day', para ver si hay al menos un grupo diferente según el test de ANOVA.

```
# Agrupar los datos por día
grouped_data = [data['total_bill'][data['sex'] == sex] for sex in
data['sex'].unique()]

# Realizar el test de ANOVA
statistic, p_value = stats.f_oneway(*grouped_data)

# Imprimir el resultado del test
alpha = 0.05
if p_value < alpha:
    print("El test de ANOVA es significativo, al menos un grupo es
diferente.")
else:
    print("No hay evidencia suficiente para rechazar la hipótesis nula.")

Output:
El test de ANOVA es significativo, al menos un grupo es diferente.
```

En lo anterior se busca una relación entre la variable 'total_bill' y 'sex', donde no se encuentra que algún grupo sea significativamente diferente al resto.

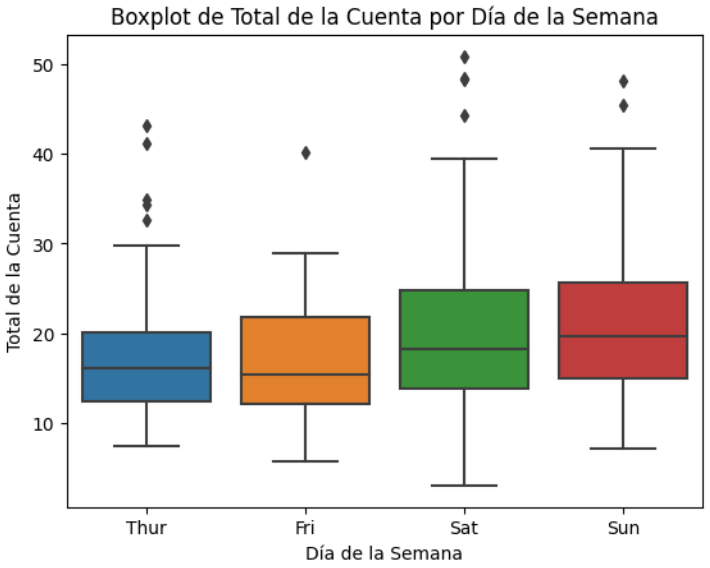
Ahora vamos a graficar los boxplot de los diferentes grupos, para ver como se observan sus distribuciones.

```
# Graficar los datos con un boxplot
sns.boxplot(x='sex', y='total_bill', data=data)
plt.xlabel('Día de la Semana')
plt.ylabel('Total de la Cuenta')
plt.title('Boxplot de Total de la Cuenta por Día de la Semana')
plt.show()
```



Procedemos a realizar el mismo procedimiento para la variable 'day'.

```
# Agrupar los datos por día
grouped_data = [data['total_bill'][data['day'] == day] for day in
data['day'].unique()]

# Realizar el test de ANOVA
statistic, p_value = stats.f_oneway(*grouped_data)

# Imprimir el resultado del test
alpha = 0.05
if p_value < alpha:
    print("El test de ANOVA es significativo, al menos un grupo es
diferente.")
else:
    print("No hay evidencia suficiente para rechazar la hipótesis nula.")

# Graficar los datos con un boxplot
sns.boxplot(x='day', y='total_bill', data=data)
plt.xlabel('Día de la Semana')
plt.ylabel('Total de la Cuenta')
plt.title('Boxplot de Total de la Cuenta por Día de la Semana')
plt.show()

Output:
El test de ANOVA es significativo, al menos un grupo es diferente.
```



En este último caso si se observa que hay al menos un grupo diferente entre los incluídos en el análisis.

Preparación de Datos (Data Preparation)

La etapa de preparación de datos (Data Preparation) es una de las fases fundamentales en el proceso de minería de datos y machine learning, ya que la calidad y adecuación de los datos influye significativamente en la efectividad de los modelos. Aquí tienes un desglose de las subetapas que son importantes revisar en esta fase, junto con algunos consejos.

Subetapas

1. **Limpieza de Datos**
    a. Identificación y manejo de valores atípicos (outliers).
    b. Tratamiento de datos faltantes (missing data imputation).
    c. Eliminación de duplicados.
    d. Conversión de datos a un formato uniforme.

2. **Transformación de Datos**
    a. Normalización o estandarización de características para que tengan la misma escala.
    b. Codificación de variables categóricas en variables numéricas (One-Hot Encoding, Label Encoding, etc.).
    c. Aplicación de transformaciones matemáticas a características (logaritmos, raíces cuadradas, etc.) para reducir la asimetría.

3. **Creación de Características (Feature Engineering)**

   a. Generación de nuevas características a partir de las existentes que puedan ser más informativas para el modelo.
   b. Extracción de características relevantes de los datos originales.

4. **Selección de Características (Feature Selection)**
   a. Identificación de características irrelevantes o redundantes.
   b. Uso de técnicas de selección de características como la correlación, la importancia de las características, etc.
   c. Consideración de la dimensionalidad de los datos y la posible reducción de características.

5. **Documentación**
   a. Mantener un registro detallado de todas las transformaciones realizadas en los datos, para que el proceso sea reproducible.

6. **Automatización**
   a. Considera la posibilidad de automatizar tareas repetitivas o procesos de limpieza utilizando scripts y herramientas adecuadas.

Consejos Adicionales

- Comprende profundamente tus datos antes de realizar cualquier transformación.
- Realiza un análisis exploratorio exhaustivo para identificar patrones, relaciones y problemas en los datos.
- Mantén un registro de todas las decisiones tomadas durante la preparación de datos.
- Mantén un conjunto de datos de prueba sin procesar para validar el modelo final.
- Utiliza bibliotecas de programación como Pandas, Scikit-Learn y herramientas de visualización para facilitar estas tareas.

La preparación adecuada de datos es esencial para garantizar que los modelos de machine learning sean precisos y útiles en la toma de decisiones.

Maldición de la Dimensionalidad

La "maldición de la dimensionalidad" es un término utilizado en estadísticas y aprendizaje automático para describir los desafíos y problemas que surgen cuando trabajamos con conjuntos de datos de alta dimensionalidad, es decir, conjuntos de datos que tienen un gran número de características o variables en comparación con el número de observaciones. Esta situación puede llevar a una serie de problemas y dificultades en el análisis de datos y en la construcción de modelos de machine learning.

Algunos de los efectos de la maldición de la dimensionalidad son los siguientes:

- **Espacio de características disperso:** A medida que aumenta la dimensionalidad, el espacio de características se vuelve extremadamente disperso. Esto significa que los puntos de datos están muy separados en el espacio, lo que puede dificultar la detección de patrones y relaciones en los datos.

- **Requerimientos computacionales:** A medida que aumenta la dimensionalidad, los algoritmos de machine learning requieren más recursos computacionales y tiempo de procesamiento. Esto puede hacer que el análisis sea lento e ineficiente.

- **Sobreajuste:** Con un número insuficiente de observaciones en comparación con las características, los modelos de machine learning pueden sobreajustarse fácilmente a los datos de entrenamiento, lo que significa que pueden funcionar bien en los datos de entrenamiento pero mal en datos no vistos.

Algunas de las formas de trabajar los dataset para superar la maldición de la dimensionalidad son los siguientes:

- **Selección de características:** Una de las estrategias más efectivas es seleccionar cuidadosamente las características más relevantes y eliminar las menos importantes. Esto reduce la dimensionalidad y mejora la eficiencia del modelo. Puedes utilizar técnicas como análisis de componentes principales (PCA) o métodos de selección de características.

- **Reducción de dimensionalidad:** Otra técnica es reducir la dimensionalidad a través de técnicas como PCA o reducción de dimensionalidad basada en árboles. Estas técnicas crean nuevas características que capturan la mayor parte de la variabilidad de los datos originales en un espacio de dimensionalidad inferior.

- **Incrementar el tamaño de la muestra:** Recolectar más datos puede ayudar a contrarrestar el problema de la maldición de la dimensionalidad. A medida que se aumenta el tamaño de la muestra, la relación entre el número de observaciones y características mejora.

- **Utilizar algoritmos apropiados:** Algunos algoritmos de machine learning son menos sensibles a la alta dimensionalidad que otros. Por ejemplo, las máquinas de soporte vectorial (SVM) con kernel lineal pueden funcionar bien incluso en conjuntos de datos de alta dimensionalidad.

En resumen, la maldición de la dimensionalidad es un desafío común en el aprendizaje automático y la estadística cuando se trabaja con conjuntos de datos de alta dimensionalidad. Sin embargo, mediante la selección de características adecuadas, técnicas de reducción de dimensionalidad y otras estrategias, es posible mitigar sus efectos y desarrollar modelos efectivos en tales escenarios.

Selección de variables

La selección de variables en modelos de machine learning se refiere a identificar las características (variables) más relevantes para predecir el objetivo de manera precisa. La selección adecuada de variables puede mejorar la eficiencia del modelo, reducir la complejidad y el sobreajuste, y acelerar el tiempo de entrenamiento. Aquí hay algunos métodos comunes de selección de variables y cómo funcionan:

1. **Selección hacia atrás (Backward Selection)**

    a. Comienza con todas las características y elimina iterativamente la menos significativa.
    b. Utiliza un criterio de evaluación, como el AIC (Criterio de Información de Akaike) o el BIC (Criterio de Información Bayesiana).
    c. Funciona bien para conjuntos de datos pequeños, pero puede ser costoso en términos computacionales para conjuntos de datos grandes.

2. **Selección hacia adelante (Forward Selection)**

    a. Comienza con un conjunto vacío y agrega características una por una.
    b. En cada paso, selecciona la característica que mejor mejora el rendimiento del modelo (según el mismo criterio de evaluación).
    c. También puede ser costoso computacionalmente en conjuntos de datos grandes.

3. **Métodos de filtro (Filter Methods)**

    a. Evalúan cada característica de forma independiente en función de una métrica, como la correlación con la variable objetivo o la varianza.
    b. Las características se seleccionan o se eliminan según un umbral predefinido.
    c. Rápido y eficiente, pero no considera las interacciones entre características.

4. **Métodos integrados (Embedded Methods)**

    a. Incorporan la selección de variables en el proceso de entrenamiento del modelo.
    b. Algunos algoritmos de machine learning, como Lasso (L1 regularization) y Random Forest, pueden realizar selección de variables internamente.
    c. Estos métodos penalizan o ponderan automáticamente las características menos importantes.

5. **Métodos de importancia de características**

a. Algunos modelos, como los árboles de decisión y Random Forest, proporcionan una medida de la importancia de cada característica.
b. Puedes seleccionar las características más importantes según estas medidas.

La elección del método de selección de variables depende del conjunto de datos, el algoritmo de machine learning utilizado y los objetivos del modelo. Es importante realizar una validación cruzada y evaluar el rendimiento del modelo después de la selección de variables para asegurarse de que no se haya perdido información relevante. Además, es posible que la selección de variables no sea necesaria en todos los casos, ya que algunos algoritmos pueden manejar conjuntos de datos con características irrelevantes sin problemas.

¡Manos a la obra! - Selección de características

Vamos a aplicar lo aprendido seleccionando características con Python. Lo primero que haremos es cargar las librerías necesarias y la data que se va a trabajar para los distintos métodos:

```
import pandas as pd
from sklearn.datasets import load_wine
import warnings

from sklearn.feature_selection import SequentialFeatureSelector
from sklearn.linear_model import LogisticRegression
from sklearn.linear_model import LassoCV

warnings.filterwarnings('ignore')

wine = load_wine()
data = pd.DataFrame(data=wine.data, columns=wine.feature_names)
target = wine.target
```

Se va a utilizar el dataset de calidad de vino de sklearn.

Forward Selection

Este método utiliza un modelo como base, en este caso vamos a utilizar la regresión logística.

```
# Crear un modelo base (por ejemplo, Regresión Logística)
base_model = LogisticRegression()

# Utilizar Forward Selection para seleccionar características
sfs = SequentialFeatureSelector(base_model, n_features_to_select=3,
direction='forward', scoring='accuracy', cv=5)
sfs.fit(data, target)
selected_features = data.columns[sfs.support_]
print("Características seleccionadas por Forward Selection:")
print(selected_features)



Output:
Características seleccionadas por Forward Selection: Index(['alcohol',
'alcalinity_of_ash', 'flavanoids'], dtype='object')
```

Este método se elige cual métrica se utiliza, por ejemplo accuracy y la cantidad de características que se desean seleccionar.

Filtro por correlaciones

Se calcula la correlación, por ejemplo la correlación de pearson y se elige un umbral (0.7) en base a esto se mantienen aquellas características que cumplan la condición.

```
# Calcular matriz de correlación
correlation_matrix = data.corr().abs()

# Crear máscara para seleccionar características altamente correlacionadas
upper = correlation_matrix.where(np.triu(np.ones(correlation_matrix.shape),
k=1).astype(bool))
to_drop = [column for column in upper.columns if any(upper[column] > 0.7)]

# Eliminar características altamente correlacionadas
filtered_data = data.drop(to_drop, axis=1)
print("Características después del filtrado por correlaciones:")
print(filtered_data.columns)

Output:

Características después del filtrado por correlaciones: Index(['alcohol',
```

```
'malic_acid', 'ash', 'alcalinity_of_ash', 'magnesium', 'total_phenols',
'nonflavanoid_phenols', 'proanthocyanins', 'color_intensity', 'hue',
'proline'], dtype='object')
```

Uso de modelos para selección (Lasso)

```
# Usar LassoCV para seleccionar características
lasso_model = LassoCV(alphas=[0.001, 0.01, 0.1, 1.0])
lasso_model.fit(data, target)
lasso_coefs = lasso_model.coef_
selected_features_lasso = data.columns[lasso_coefs != 0]
print("Características seleccionadas por Lasso:")
print(selected_features_lasso)

Output:

Características seleccionadas por Lasso: Index(['alcohol', 'malic_acid',
'alcalinity_of_ash', 'flavanoids', 'color_intensity',
'od280/od315_of_diluted_wines', 'proline'], dtype='object')
```

¡Manos a la obra! - ¿Cual método es mejor?

A partir de los 3 métodos implementados anteriormente ¿cuál consideras que es mejor? ¿Por qué?

1. Compara las variables seleccionadas para cada método. ¿Coinciden?
2. Utiliza un algoritmo y entrenalo con los 3 set de variables? ¿Cuál es mejor? ¿Cuál se demora menos?
3. Concluye en base a lo anterior.

Preguntas de proceso

Reflexiona:

- ¿A qué propósito sirve la fase de Business Understanding en la metodología CRISP-DM?
- ¿Por qué es importante involucrar a los stakeholders desde el principio en un proyecto de ciencia de datos?
- ¿Cuál es el objetivo principal al definir los objetivos del negocio en un proyecto de ciencia de datos?
- ¿Cuál es la diferencia entre datos estructurados y no estructurados? ¿Puede un proyecto de ciencia de datos trabajar con ambos tipos?
- ¿Por qué es crucial comprender la calidad de los datos antes de comenzar el análisis?
- ¿Qué es una variable categórica y cómo se diferencia de una variable numérica en el contexto de análisis de datos?
- ¿Qué pasos incluye la fase de Data Preparation en CRISP-DM y por qué es fundamental?
- ¿Cuál es el propósito de la limpieza de datos en la preparación de datos?
- ¿Cómo se pueden manejar los valores faltantes en un conjunto de datos?
- ¿Por qué es importante detectar y manejar los outliers en un conjunto de datos?
- ¿Cuáles son algunas técnicas comunes para detectar outliers en un conjunto de datos numéricos?
- ¿Cuál es el enfoque de "IQR" en la detección de outliers y cómo se calcula?
- ¿Qué estrategias se pueden utilizar para manejar los outliers una vez que se detectan?
- ¿Por qué es esencial realizar un análisis exploratorio de datos antes de construir modelos de machine learning? ¿Qué se puede descubrir a través de este análisis?

¡Continúa aprendiendo y practicando!