Variable Aleatoria II

Clase sincrónica

{desafío} latam_

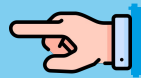
Aplica herramientas de Python para calcular indicadores estadísticos y probabilidades.

- **Unidad 1: Estadística descriptiva y probabilidades**
  (Parte I)

  (Parte II)
- **Unidad 2: Variable aleatoria**
  (Parte I)

  (Parte II)
- **Unidad 3:Estadística inferencial**
- **Unidad 4: Regresión**
  (Parte I)

  (Parte II)



Te encuentras aquí

{desafío} latam_

¿Qué aprenderás en esta sesión?

Crea visualizaciones que permiten identificar la distribución de variables de un dataset

¿Qué importancia tiene la visualización de datos?
¿Con qué herramientas contamos para ello?

{desafío} latam_

/*Gráficos con Seaborn*/

Gráficos con Seaborn

Seaborn

- Seaborn es una biblioteca de visualización de datos en Python que se basa en la popular biblioteca Matplotlib.

- Una de las principales características de Seaborn es su capacidad para generar automáticamente gráficos estilizados con solo unas pocas líneas de código.

Fuente: https://seaborn.pydata.org/

Manos a la obra
¡A graficar en Seaborn!

¡A graficar en Seaborn!

Vamos a aprender a utilizar comandos básicos de Seaborn para construir gráficos de acuerdo a diferentes necesidades y requerimientos. Para ello, sigue la presentación que te mostrará tu profesor

{desafío}
latam_

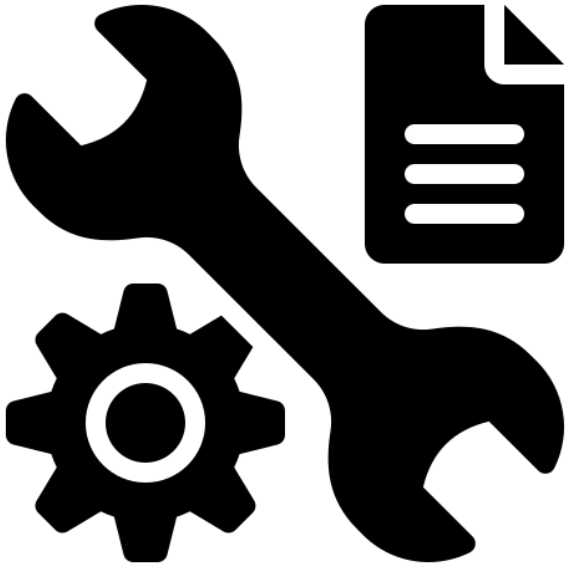
/* Preparación de datos*/

Preparación de datos

¿En qué consiste?

La preparación de datos, también conocida como preprocesamiento de datos, es una etapa crítica en el análisis de datos y el desarrollo de modelos de aprendizaje automático.

Consiste en una serie de pasos y técnicas que se aplican a los datos crudos (raw data) con el objetivo de limpiarlos, transformarlos y estructurarlos adecuadamente para su posterior análisis.

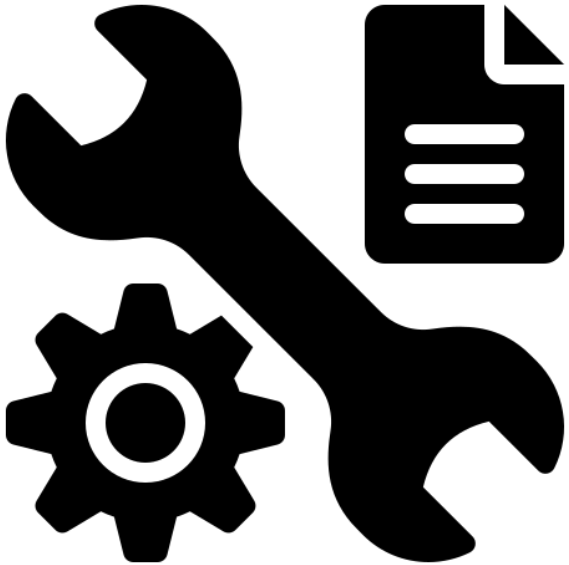


Preparación de datos

¿Para qué?

La preparación de datos es fundamental porque los datos crudos pueden ser desorganizados, contener ruido, datos faltantes, valores atípicos o inconsistencias, lo que podría afectar negativamente los resultados del análisis y los modelos de aprendizaje automático.

Por lo tanto, esta etapa tiene como objetivo mejorar la calidad y la utilidad de los datos.



Preparación de datos

¿Cómo se hace?

Algunos de los pasos comunes en la preparación de datos incluyen:

1. Limpieza de datos
2. Transformación de datos
3. Codificación de variables categóricas

Preparación de datos

Opciones con Python -Valores faltantes

missing.values=df.apply(lambda x: summ(x.isnull()),axis=0)

En esta línea, utilizamos el método apply() de Pandas para aplicar una función lambda a cada columna (axis=0) del DataFrame df. La función lambda lambda x: sum(x.isnull()) se aplica a cada columna x, y cuenta la cantidad de valores faltantes en esa columna utilizando x.isnull(). La función sum() luego suma los valores True (1) que representan los valores faltantes, proporcionando el recuento total de valores faltantes para cada columna.

El resultado será una serie de Pandas llamada missing_values, donde los índices son los nombres de las columnas del DataFrame df y los valores son la cantidad de valores faltantes en cada columna.

Preparación de datos

Valores faltantes - ¿qué hacemos?

df['columna'].fillna(df['columna'].mean(), inplace=True)

En esta línea, reemplazamos los valores faltantes de ‘columna’ por el valor promedio de los datos de ‘columna’.

¿Qué otra solución podríamos utilizar?

Preparación de datos

Valores repetidos - duplicates

df_sin_duplicados = df.drop.duplicates(subset=[‘columna1’,’columna2’])

En esta línea, construimos otro dataFrame eliminando los valores que se encuentren repetidos considerando las columnas ‘columna1’ y ‘columna2’. Si utilizamos solo drop.duplicates(), el valor repetido solo se eliminará si la coincidencia es en todas las columnas.

Preparación de datos

Valores repetidos - ejemplo

```
data = {'Nombre': ['Alice', 'Bob', 'Alice', 'Charlie', 'Bob', 'David', 'Luis'],
        'Edad': [25, 30, 25, 35, 30, 22,25],
        'Ciudad': ['New York', 'Los Angeles', 'New York', 'Chicago', 'Los Angeles', 'Boston','New
York']}
df = pd.DataFrame(data)
```

- **df.drop.duplicates()** eliminará los datos repetidos correspondientes a Alice y Bob.
- **df.drop.duplicates(subset=['Ciudad'])** eliminará los datos correspondientes a Alice, Bob y Luis. Este último es eliminado porque vive en la misma ciudad que Alice

Preparación de datos

Valores atípicos

Si estamos analizando estaturas de un grupo de personas, una primera verificación sería asegurarnos de que todos los datos sean numéricos (de tipo int o float, dependiendo de lo que nos interese). En el mejor de los casos, podríamos convertirlos a este tipo de datos.

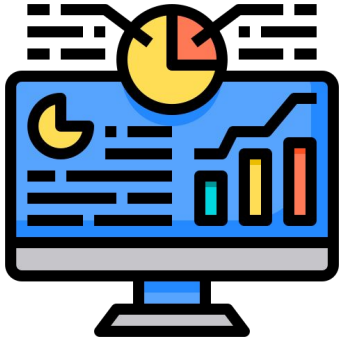
Pero, ¿qué sucede si hay valores numéricos, que son poco razonables?

- ¿Qué podría ser, en este contexto, un valor poco razonable?

Preparación de datos

Valores atípicos

Los valores atípicos corresponden a valores de una variable que se escapan sensiblemente del conjunto. En ocasiones puedes ser muy valiosos y puede ser necesario tenerlos en cuenta, pero en otras ocasiones distorsionan sensiblemente la observación y el objetivo que tenemos para ella.

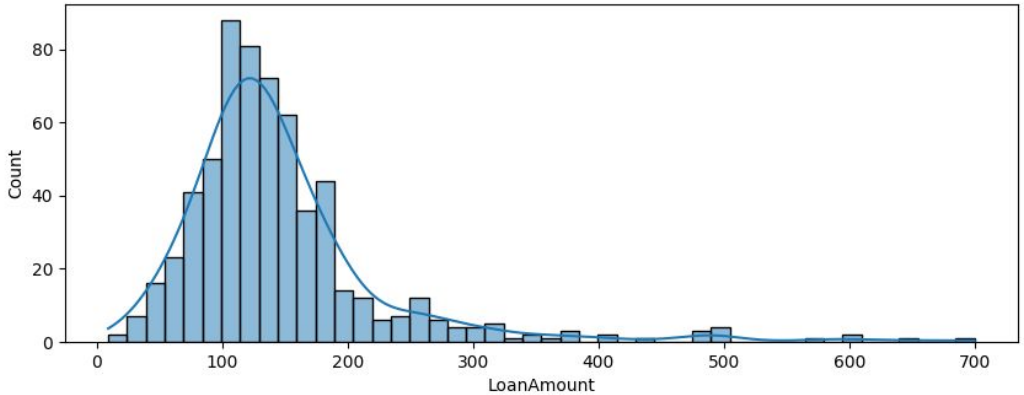


Ejemplo: si estamos analizando la estatura de los habitantes de un país o región, y escogemos una muestra, una persona con gigantismo o enanismo va a distorsionar mucho nuestra observación, haciendo que nuestras conclusiones puedan ser erróneas.Del mismo modo, si escribimos mal un valor que era con decimales y no lo pusimos así.

Preparación de datos

Valores atípicos

Podemos detectar valores atípicos fácilmente utilizando un boxplot o incluso un histograma

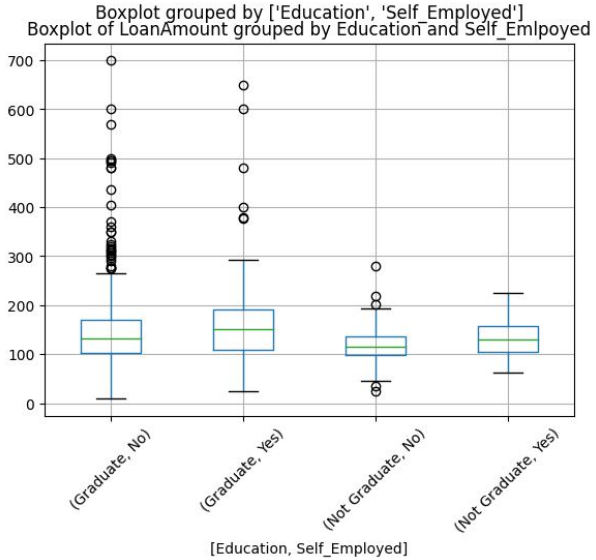


{desafío}
latam_

Preparación de datos

Valores atípicos

Podemos detectar valores atípicos fácilmente utilizando un boxplot o incluso un histograma



Preparación de datos

Valores atípicos - filtrando

Un criterio habitual para seleccionar solo los datos no atípicos es utilizar como referencia el rango intercuartil, y los cuartiles 1 y 3. Así, nos quedamos solo con los valores que se encuentran en el intervalo [Q1 - 1,5*IQR, Q3 + 1,5*IQR]

```
data = df['columna']
Q1 = np.percentile(data, 25)
Q3 = np.percentile(data, 75)
IQR = Q3 - Q1
lower_bound = Q1 - 1.5 * IQR
upper_bound = Q3 + 1.5 * IQR

df=df.iloc[np.where((data>= lower_bound) * (data <= upper_bound))]
```

Desafío - Preparación de datos y gráficos

{desafío} latam_

Desafío

"Preparación de datos y gráficos"

- Descarga el archivo "Desafío".
- Tiempo de desarrollo asincrónico: desde 2 horas.
- Tipo de desafío: individual.

¡AHORA TE TOCA A TI! 💪

{desafío} latam_

Ideas fuerza

*El paquete **Seaborn** nos permite crear diferentes **gráficos** a partir de datos organizados en **DataFrame***

*Una adecuada **preparación de datos** nos permite realizar un mejor **análisis** y obtener mejores **conclusiones.***

*Para la preparación de datos nos podemos apoyar en el **análisis exploratorio** de ellos, para comprender mejor su **contexto**.*

¿Qué aspectos de la clase necesito reforzar?

Recursos asincrónicos

¡No olvides revisarlos!

Para esta semana deberás revisar:

- Guía de estudio.
- Desafío “Preparación de datos y gráficos”.

Próxima sesión...

- *Aplicar los conceptos de estadística y probabilidad en Python para describir datasets y validar hipótesis.*

{desafío}
latam_

Academia de
talentos digitales