

ITESM

# MAESTRÍA EN INTELIGENCIA ARTIFICIAL APLICADA

SEGUNDO TRIMESTRE

APUNTES DATA ANALYSIS CON  
PYTHON

MODULO 2

NOMBRE: VILLALPANDO GUERRERO  
JIRAM CESAR

## MODULO 2

Dato de preprocesamiento o data cleaning

El preprocesamiento de datos es un paso preliminar durante el proceso de minería de datos. Se trata de cualquier tipo de procesamiento que se realiza con los datos brutos para transformarlos en datos que tengan formatos que sean más fáciles de utilizar.

En Python se pueden realizar operaciones en dataframes

`df["symboling"]`      `df["body-style"]`



|   | symboling | normalized-losses | make        | fuel-type | aspiration | num-of-doors | body-style  | drive-wheels | engine-location | wheel-base | ... | engine-size | fuel-system | bore | stroke | compression-ratio |
|---|-----------|-------------------|-------------|-----------|------------|--------------|-------------|--------------|-----------------|------------|-----|-------------|-------------|------|--------|-------------------|
| 0 | 3         | ?                 | alfa-romero | gas       | std        | two          | convertible | rwd          | front           | 88.6       | ... | 130         | mpfi        | 3.47 | 2.68   | 9.0               |
| 1 | 3         | ?                 | alfa-romero | gas       | std        | two          | convertible | rwd          | front           | 88.6       | ... | 130         | mpfi        | 3.47 | 2.68   | 9.0               |
| 2 | 1         | ?                 | alfa-romero | gas       | std        | two          | hatchback   | rwd          | front           | 94.5       | ... | 152         | mpfi        | 2.68 | 3.47   | 9.0               |
| 3 | 2         | 164               | audi        | gas       | std        | four         | sedan       | fwd          | front           | 99.8       | ... | 109         | mpfi        | 3.19 | 3.40   | 10.0              |
| 4 | 2         | 164               | audi        | gas       | std        | four         | sedan       | 4wd          | front           | 99.4       | ... | 136         | mpfi        | 3.19 | 3.40   | 8.0               |
| 5 | 2         | ?                 | audi        | gas       | std        | two          | sedan       | fwd          | front           | 99.8       | ... | 136         | mpfi        | 3.19 | 3.40   | 8.5               |
| 6 | 1         | 158               | audi        | gas       | std        | four         | sedan       | fwd          | front           | 105.8      | ... | 136         | mpfi        | 3.19 | 3.40   | 8.5               |
| 7 | 1         | ?                 | audi        | gas       | std        | four         | wagon       | fwd          | front           | 105.8      | ... | 136         | mpfi        | 3.19 | 3.40   | 8.5               |
| 8 | 1         | 158               | audi        | gas       | turbo      | four         | sedan       | fwd          | front           | 105.8      | ... | 131         | mpfi        | 3.13 | 3.40   | 8.3               |
| 9 | 0         | ?                 | audi        | gas       | turbo      | two          | hatchback   | 4wd          | front           | 99.5       | ... | 131         | mpfi        | 3.13 | 3.40   | 7.0               |

Se pueden agregar valores a las columnas a través de: `df["x"]=df["x"]+1`

Missing Values

Los valores perdidos (missing values en inglés) están presente en la mayoría de los conjuntos de datos con los que trabajemos en nuestro día a día. Son aquellos en los que no se almacena ningún valor de datos en una observación.

Para interactuar con los valores faltantes de puede hacer:

- Eliminar los datos
- Reemplazar los datos con la media o haciendo una aproximación
- Reemplazar con la frecuencia
- Reemplazar en otras funciones

También a veces se pueden dejar los datos como faltantes

## ¿Cómo eliminar los datos faltantes?

Se puede usar lo siguiente

```
dataframes.dropna () :
```

Ejemplo: `df.dropna(subset) ["x"], axis=0, inplace= True)`

## Para reemplazar

Pandas tiene la siguiente función

```
dataframe.replace(missing_value, new_value):
```

Ejemplo: `df["x"].replace(np.nan, mean)`

## Data Formatting

Ayuda a estandarizar los datos que se tienen en las distintas bases de datos.

- Si se trata de alguna operación equivalente se puede escribir en una línea de código.
- Se debe tener cuidado de no asignar datos equivocados

Tipos de datos en pandas como: objetos, int64, float64, etc

Para saber que tipo de datos usamos se puede usar: `dataframe.dtypes()`

Para convertir un tipo de datos se puede usar: `dataframe.astype()`

## Normalización de datos

Normalizar la estructura de datos de los pandas con la normalización min-max. Este es uno de los métodos ampliamente utilizados para la normalización. El resultado de la normalización resta el valor mínimo de Dataframe y lo divide por la diferencia entre el valor más alto y el más bajo de la columna correspondiente.

Ejemplo:

| age | income |
|-----|--------|
| 20  | 100000 |
| 30  | 20000  |
| 40  | 500000 |

Not-normalized



| age | income |
|-----|--------|
| 0.2 | 0.2    |
| 0.3 | 0.04   |
| 0.4 | 1      |

Normalized

De esta manera se puede trabajar de una mejora manera

## Métodos de normalización de datos

- Simple escala

$$x_{new} = \frac{x_{old}}{x_{max}}$$

Ejemplo:



```
df["length"] = df["length"]/df["length"].max()
```

- Min-max

$$x_{new} = \frac{x_{old} - x_{min}}{x_{max} - x_{min}}$$

Ejemplo:

```
df["length"] = (df["length"] - df["length"].min()) /  
(df["length"].max() - df["length"].min())
```

- Z-Score

$$x_{new} = \frac{x_{old} - \mu}{\sigma}$$

Ejemplo:


```
df["length"] = (df["length"] - df["length"].mean()) / df["length"].std()
```

## Clasificación

Sucede cuando se agrupan los valores en contenedores

Por ejemplo:

| price |  |
|-------|--|
| 13495 |  |
| 16500 |  |
| 18920 |  |
| 41315 |  |
| 5151  |  |
| 6295  |  |
| ...   |  |



| price | price-binned |
|-------|--------------|
| 13495 | Low          |
| 16500 | Low          |
| 18920 | Medium       |
| 41315 | High         |
| 5151  | Low          |
| 6295  | Low          |
| ...   | ...          |

El primer DF, en el segundo.

A través del siguiente código:

```
bins = np.linspace(min(df["price"]), max(df["price"]), 4)
```

```
group_names = ["Low", "Medium", "High"]
```

## Convertir variables categóricas en variables cuantitativas

Los algoritmos matemáticos no pueden incluir objetos o cadenas como entrada ni para el entrenamiento del modelo, acá solo toma números, pero justamente es, en estos casos, que se puede hacer una pequeña modificación en la data para cambiar los datos categóricos en numéricos.



## Variables Dummy

Se puede usar el método: `pandas.get_dummies()` method.

Cara poder categorizar en 0 o 1

```
Pd.get_dummies(df['x'])
```



## Graded Review Questions

### Question 1

1/1 point (graded)

Consider the dataframe `df`. What is the result of the following operation: `df['symboling'] = df['symboling'] + 1`?

☒ Every element in the column "symboling" will increase by one.

☐ Every element in the row "symboling" will increase by one.

☐ Every element in the dataframe will increase by one.



Save | Show answer

Submit

You have used 1 of 2 attempts

### Question 2

1/1 point (graded)

Consider the dataframe `df`. What does the command `df.rename(columns={'a':'b'})` change about the dataframe `df`?

☐ Renames column "a" of the dataframe to "b".

☐ Renames row "a" to "b".

☒ Nothing. You must set the parameter "inplace = True".



Show answer

Submit

You have used 2 of 2 attempts

### Question 3

1 point possible (graded)

Consider the dataframe "df". What is the result of the following operation `df['price'] = df['price'].astype(int)`?

☐ Convert or cast the row 'price' to an integer value.

☒ Convert or cast the column 'price' to an integer value.

☐ Convert or cast the entire dataframe to an integer value.

Save

Submit

You have used 0 of 2 attempts



## Question 4

1/1 point (graded)

Consider the column of the dataframe `df['a']`. The column has been standardized. What is the standard deviation of the values as a result of applying the following operation: `df['a'].std()`?

☒ 1

☐ 0

☐ 3



Show answer

Submit

You have used 2 of 2 attempts

## Question 5 a)

1/1 point (graded)

Consider the column of the dataframe, `df['Fuel']`, with two values: 'gas' and 'diesel'. What will be the name of the new columns `pd.get_dummies(df['Fuel'])`?

☐ 1 and 0

☐ Just 'diesel'

☐ Just 'gas'

☒ 'gas' and 'diesel'



Show answer

Submit

You have used 2 of 2 attempts

✓ Correct (1/1 point)

## Question 5 b)

1/1 point (graded)

What are the values of the new columns from part 5a)?

☒ 1 and 0

☐ Just 'diesel'

☐ Just 'gas'

☐ 'gas' and 'diesel'



Save | Show answer

Submit

You have used 1 of 2 attempts

✓ Correct (1/1 point)