Guía de Laboratorio – Preprocesamiento de datos

Sistema de conocimiento:

- Preprocesamiento de datos
- Evaluación de modelos entrenamos usando métricas.

Objetivos:

- Aplicar técnicas de preprocesamiento de datos a conjuntos de datos como paso previo del aprendizaje automático.
- Evaluar el rendimiento de modelos utilizando diferentes métricas.

Bibliografía:

- García, S., Luengo, J., & Herrera, F. (2015). *Data preprocessing in data mining* (Vol. 72). Springer.
- Jafari, R. (2022). *Hands-On Data Preprocessing in Python: Learn how to effectively prepare data for successful data analytics*. Packt Publishing.

Introducción

En la conferencia pasado estudiamos los conceptos fundamentales relacionados con el preprocesamiento de datos y de igual manera pudimos estudiar los principales tipos de métricas que nos permiten evaluar los modelos entrenados.

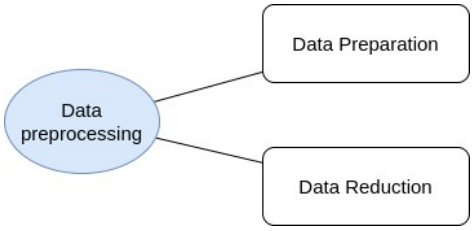
En el laboratorio de hoy veremos dos casos de estudios donde podremos ver ejemplificados estos conceptos.

Preparación de datos

El preprocesamiento de datos es una de las fases por las que hay que transitar cuando se está descubriendo conocimiento en los datos (Figura 1). Podemos definir el procesamiento como:

"… aquellas técnicas de análisis de datos que permite mejorar la calidad de un conjunto de datos de modo que las técnicas de *extracción de conocimiento (minería de datos)* puedan obtener mayor acción (mejor porcentaje de clasificación, reglas con más completitud, etc.)(García et al., 2015)"

El preprocesamiento de datos se clasifican dependiendo del tipo y el conjunto de técnicas que poseen en:



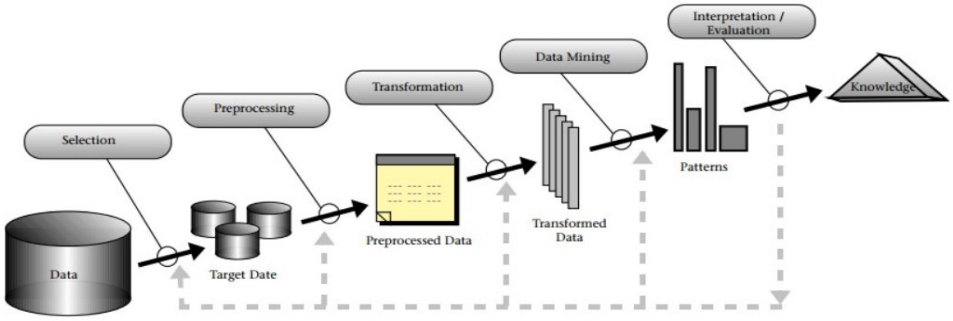


Figura 1: Fases de KDD.

La *preparación de datos* es normalmente un paso obligatorio. Convierte datos previos inútiles en nuevos datos que se ajustan a un proceso de minería de datos (DM). En primer lugar, si los datos no están preparados, es posible que el algoritmo de DM no los reciba para poder funcionar o seguramente informará de errores durante su ejecución. En el mejor de los casos, el algoritmo funcionará, pero los resultados ofrecidos no tendrán sentido o no se considerarán conocimientos precisos(García et al., 2015).

Así pues, ¿cuáles son las cuestiones básicas que deben resolverse en la preparación de datos? A continuación una lista de preguntas con sus respuestas correctas para cada tipo de proceso tipo de proceso que pertenece a la familia de técnicas de preparación de datos (Figura 2):

- ¿Cómo se limpian los datos? - Limpieza de datos.
- ¿Cómo proporciono datos precisos? - Transformación de datos.
- ¿Cómo incorporar y ajustar los datos? - Integración de datos.
- ¿Cómo unifico y amplío los datos? - Normalización de datos.
- ¿Cómo tratar los datos que faltan? - Imputación de datos ausentes.
- ¿Cómo detectar y gestionar el ruido? - Identificación del ruido.

La *reducción de datos* comprende el conjunto de técnicas que, de un modo u otro, obtienen una representación reducida de los datos originales (Figura 3). En el caso de la reducción de datos, los datos producidos suelen mantener estructura esencial y la integridad de los datos originales, pero se reduce la cantidad de datos. Entonces, ¿se pueden utilizar los datos originales, sin aplicar un proceso de reducción de datos, como entrada de un proceso de gestión de datos? La respuesta es sí, pero hay que tener en cuenta otros aspectos importantes, tan cruciales como las que aborda la preparación de datos.

De ahí que, a primera vista, pueda considerarse un paso opcional. Sin embargo, esta afirmación puede ser conflictiva. Aunque se mantenga la integridad de los datos, es bien sabido que cualquier algoritmo tiene una cierta complejidad temporal que depende de varios parámetros.

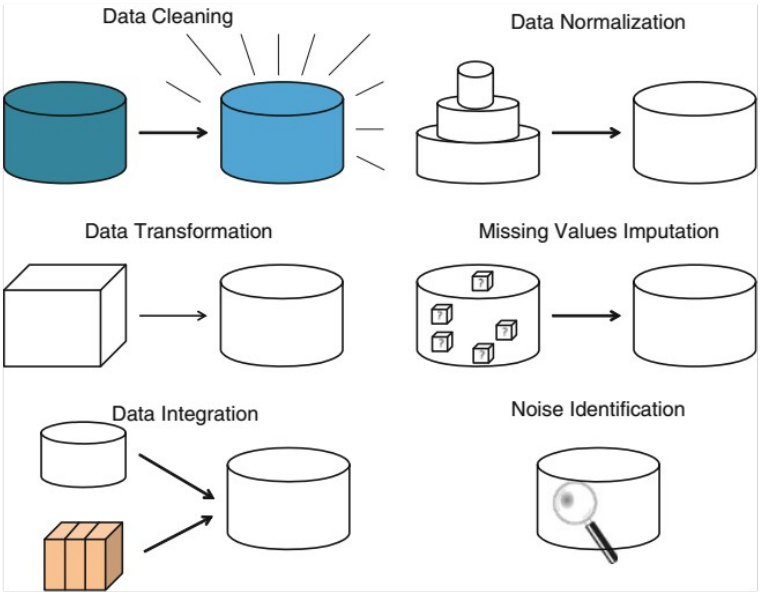


Figura 2: Formas de preparación de datos. Fuente: (García et al., 2015)

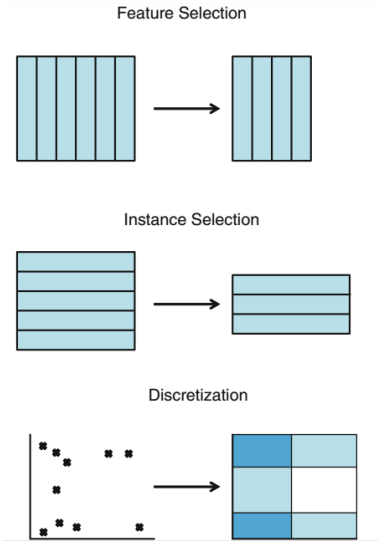


Figura 3: Formas de reducción de datos. Fuente: (García et al., 2015)

En DM, uno de estos parámetros es de algún modo directamente proporcional al tamaño de la base de datos de entrada. Si el tamaño supera el límite, límite que depende mucho del tipo de algoritmos de DM, la ejecución del algoritmo puede ser prohibitiva, y entonces la tarea de reducción de datos es tan crucial como lo es la preparación de los datos. En cuanto a otros factores, como la disminución de la complejidad y la mejora de la calidad de los modelos obtenidos, el papel de la reducción de datos vuelve a ser decisivo.

Como ya se ha mencionado, ¿cuáles son las cuestiones básicas que deben resolverse en la reducción de datos? De nuevo, proporcionamos una serie de preguntas asociadas a la respuesta correcta relacionadas con cada tipo de tarea perteneciente a las técnicas de reducción de datos (García et al., 2015):

- ¿Cómo reducir la dimensionalidad de los datos? - Selección de características (FS).
- ¿Cómo elimino los ejemplos redundantes y/o conflictivos? - Selección de instancias (IS).
- ¿Cómo simplificar el dominio de un atributo? - Discretización.
- ¿Cómo rellenar huecos en los datos? - Extracción de características y/o generación de instancias

Para profundizar en los aspectos teóricos de los diferentes métodos y técnicas de preparación y reducción de datos puede leer los siguientes elementos:

- Data preparation
  - Capítulo 3 Modelos básicos de preparación de datos, página 39 del libro (García et al., 2015).
  - Capítulos 9, 10, 11, 12 y 14 del libro (Jafari, 2022).
- Data reduction
  - Capítulo 6 Reducción de datos, página 147 del libro (García et al., 2015).
  - Capítulo 13 Data reduction, página 404 del libro (Jafari, 2022)

Caso de estudio

Para ejemplificar como es el proceso de preparación de datos necesario en el aprendizaje automático, describiremos el caso mostrado en el Capítulo 15 del libro (Jafari, 2022). Este ejemplo utilizará un conjunto de datos sobre enfermedades mentales recogidas por la organización *Open Sourcing Mental Illness (OSMI)*.

Introducción al caso de estudio

Los trastornos de salud mental, como la ansiedad y la depresión, son intrínsecamente perjudiciales para el bienestar, el estilo de vida y la capacidad de productividad laboral de las personas. Según Mental Health America, más de 44 millones de adultos estadounidenses padecen algún trastorno mental. La salud mental de los empleados de la industria tecnológica es motivo de gran preocupación debido a los entornos competitivos que a menudo se encuentran dentro de estas empresas y entre ellas. Algunos empleados de estas empresas se ven obligados a hacer horas extraordinarias simplemente para conservar su puesto de trabajo. Los directivos de este tipo de empresas tienen buenas razones para desear una mejora de la salud mental de sus empleados, porque las mentes sanas son productivas y las distraídas no.

Los directivos y líderes de empresas tecnológicas y no tecnológicas deben tomar decisiones difíciles sobre si invertir o no en la salud mental de sus empleados y, en caso afirmativo, en qué medida. Hay muchas pruebas de que una mala salud mental puede repercutir negativamente en el bienestar y la productividad de los trabajadores. Cada empresa dispone de una cantidad finita de fondos que puede invertir en la salud física de sus empleados, por no hablar de la salud mental. Saber dónde asignar los recursos es de gran importancia. Esto sirve de introducción general a este estudio de caso. A continuación, trataremos un aspecto muy importante de cualquier análisis de datos: ¿a quién van dirigidos nuestros resultados?

La audiencia de los resultados de la analítica

Siempre, el público principal de los resultados de cualquier análisis son los responsables de la toma de decisiones; sin embargo, es importante tener claro quiénes son exactamente esos responsables. En los proyectos reales, esto debería ser obvio, pero aquí, en este laboratorio, como nuestro objetivo es practicar, tenemos que imaginar un responsable de la toma de decisiones específico y adaptar nuestro análisis para él.

Los responsables de la toma de decisiones en los que nos centraremos son los directivos y los líderes de las empresas tecnológicas encargados de tomar decisiones que pueden repercutir en la salud mental de sus empleados. Aunque la salud mental debería considerarse una prioridad, en realidad, los directivos tienen que navegar por un entorno de toma de decisiones que tiene muchas prioridades contrapuestas, como la salud financiera de la organización, la supervivencia, la maximización de los beneficios, las ventas y el servicio al cliente, así como el crecimiento económico.

Por ejemplo, la siguiente sencilla visualización creada por OSMI (disponible en https://osmi.typeform.com/report/A7mlxC/itVHRYbNRnPqDI9C) nos dice que, aunque el apoyo a la salud mental en las empresas tecnológicas no es terrible, sigue habiendo una gran brecha que mejorar:

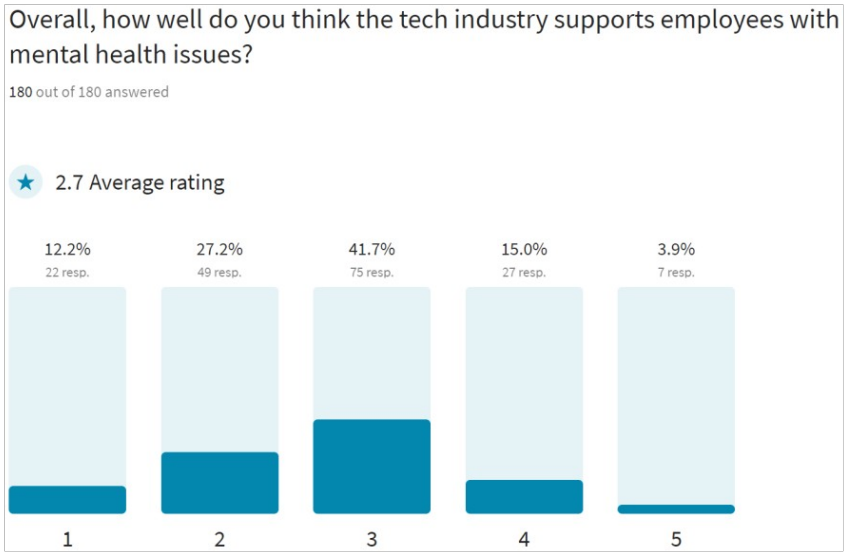


Figura 4: Una visualización sencilla de la encuesta OSMI 2020 sobre salud mental en la tecnología. Fuente: (Jafari, 2022)

Nuestro objetivo en este estudio de caso es cavar un poco más profundo que el informe básico proporcionado por OSMI y ver las interacciones entre más atributos, que puede ser informativo y beneficioso a los tomadores de decisiones descritos.

En concreto, para este estudio de caso, vamos a tratar de responder a las siguientes *preguntas analíticas* (PA) que pueden informar a los responsables de la toma de decisiones descritos sobre la actitud y la importancia de la salud mental en sus empleados:

- PA1: ¿Existe una diferencia significativa entre la salud mental de los empleados en función del atributo de género?
- PA2: ¿Existe una diferencia significativa entre la salud mental de los empleados según el atributo de la edad?
- PA3: ¿Las empresas más favorables tienen empleados más sanos mentalmente?
- PA4: ¿Influye la actitud de los individuos hacia la salud mental en su salud mental y la búsqueda de tratamiento?

Ahora que tenemos claro cómo vamos a analizar estos datos y qué preguntas queremos responder, vamos a ponernos manos a la obra y empezar a conocer la fuente de los datos.

Introducción a las fuentes de los datos

OSMI comenzó a realizar la encuesta sobre salud mental en la tecnología en 2014, y aunque la tasa de participación en sus encuestas ha disminuido a lo largo de los años, han seguido recopilando

datos hasta ahora. Los datos en crudo para los años 2014, y del 2016 al 2022 están accesibles en la siguiente dirección https://osmihelp.org/research.html. Para el ejemplo se utilizarán los conjuntos de datos comprendidos del año 2016 al 2020.

A continuación realizaremos la integración de datos y luego la limpieza de los datos.

Integrando las fuentes de datos

Como se ha comentado, es necesario integrar cinco conjuntos de datos diferentes. Después de haber visto estos cinco conjuntos de datos que recogen información de las encuestas OSMI sobre salud mental en tecnología a lo largo de cinco años diferentes, se dará cuenta de que la encuesta a lo largo de los años ha sufrido muchos cambios. Además, aunque los conjuntos de datos recopilados tratan sobre la salud mental en la tecnología, la redacción de las preguntas y, en ocasiones, la naturaleza de las mismas han cambiado. Por lo tanto, el embudo figurativo de la siguiente figura tiene dos finalidades. En primer lugar, deja pasar las partes de los datos de cada conjunto de datos que son comunes a los seis conjuntos de datos. En segundo lugar, el embudo también filtra los datos que no son relevantes para nuestras PA:

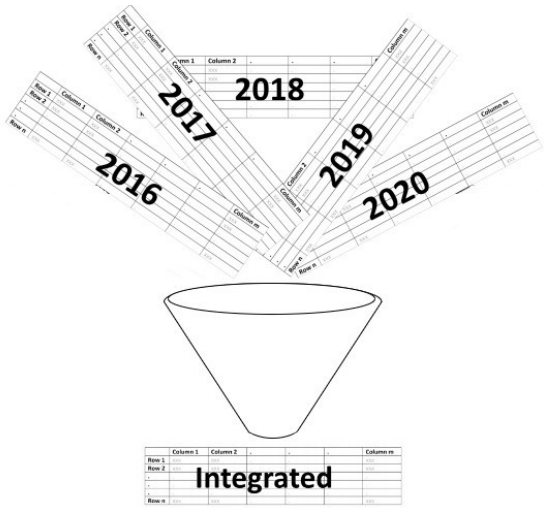


Figura 5: Esquema de la integración de 5 dataset en uno. Fuente:(Jafari, 2022)

Aunque la figura anterior hace que la integración de estos cinco conjuntos de datos parezca sencilla, nos esperan retos importantes.

Si imprimimos la figura de cada uno de estos conjuntos de datos, podremos notar que todos tienen cantidades diferentes de columnas.

[In]	[Out]
`print(df_2016.shape)`	`(1433, 63)`

```
print(df_2017.shape)
print(df_2018.shape)
print(df_2019.shape)
print(df_2020.shape)
```

```
(756, 123)
(417, 123)
(352, 82)
(180, 120)
```

En esta salida podemos ver que los años que más preguntas realizó la encuesta es el 2017 y el 2018.

Lo primero es saber cuáles son los atributos comunes hay entre estos cinco conjuntos de datos si no existe una redacción coherente entre ellos. ¿Tenemos que hacerlo manualmente? Es una forma de hacerlo, pero sería un proceso muy largo. Podemos utilizar SequenceMatcher del módulo *difflib* para encontrar los atributos que son similares entre sí.

Con la función *similar* definida a continuación se puede comprobar el grado de similitud entre dos secuencias.

```
from difflib import SequenceMatcher

def similar(a, b):
    return SequenceMatcher(None, a, b).ratio()
```

Por ejemplo, si se realiza una llamada la función de la siguiente manera: similar("Apple","Appel"), el resultado sería igual a 0.8.

Luego debemos filtrar todos los elementos que son similares en todos los conjuntos de datos. Para lograr esto podemos usar la siguiente secuencia:

```
Columns = pd.DataFrame(False, index = df_2017.columns, columns =['y2016','y2017','y2018','y2019','y2020'])
Columns.y2017 = True

for col in df_2016.columns:
    for incol in Columns.index:
        if(similar(col,incol)>0.7):
            Columns.at[incol,'y2016'] = True

for col in df_2018.columns:
    for incol in Columns.index:
        if(similar(col,incol)>0.7):
            Columns.at[incol,'y2018'] = True

for col in df_2019.columns:
    for incol in Columns.index:
        if(similar(col,incol)>0.7):
            Columns.at[incol,'y2019'] = True

for col in df_2020.columns:
    for incol in Columns.index:
        if(similar(col,incol)>0.7):
            Columns.at[incol,'y2020'] = True

Columns[Columns.y2016 & Columns.y2017 & Columns.y2018 & Columns.y2019&Columns.y2020]
```

Este código devuelve aquellas preguntas que fueron comunes en todas las encuestas. Luego seleccionamos las preguntas comunes y las colocamos en una lista:

```
Selected_columns = ['Does your employer provide mental health benefits as part of healthcare coverage?',
                    'Has your employer ever formally discussed mental health (for example, as part of a wellness campaign or other official communication)?',
                    'Does your employer offer resources to learn more about mental health disorders and options for seeking help?',
                    'Is your anonymity protected if you choose to take advantage of mental health or substance abuse treatment resources provided by your employer?',
                    'If a mental health issue prompted you to request a medical leave from work, how easy or difficult would it be to ask for that leave?',
                    'Would you feel comfortable discussing a mental health issue with your direct supervisor(s)?',
                    'Would you feel comfortable discussing a mental health issue with your coworkers?',
                    'If you have revealed a mental health disorder to a coworker or employee, how has this impacted you or the relationship?',
                    'If you have revealed a mental health disorder to a client or business contact, how has this affected you or the relationship?',
                    'How willing would you be to share with friends and family that you have a mental illness?',
                    'What is your age?',
                    'What is your gender?',
                    'What country do you <strong>live</strong> in?',
                    'What country do you <strong>work</strong> in?',
                    'Have you ever been diagnosed with a mental health disorder?',
                    'Have you ever sought treatment for a mental health disorder from a mental health professional?']
```

Una vez realizado el filtrado basado en lo que es común a los cinco conjuntos de datos, aún necesitamos mantener sólo los atributos que son relevantes para nuestras PA. La manera de hacer esto para el dataset del 2016 se muestra a continuación:

```
# if a column of 2016 data source is not one of the selected ones, this code drops it.
droping_cols = []
for col in df_2016.columns:
    maxScore = 0
    for sel_col in Selected_columns:
        if(similar(col,sel_col)>maxScore):
            maxScore = similar(col,sel_col)
    if (maxScore <0.7):
        droping_cols.append(col)
    else:
        if('previous' in col):
            droping_cols.append(col)


df_2016.drop(columns = droping_cols, inplace=True)
df_2016.head(1)
```

Debe hacer los mismo para los demás conjuntos de datos.

La siguiente lista recoge de los atributos que son comunes a los cinco conjuntos de datos y relevantes para nuestros PA. Para que los datos tengan un aspecto más limpio, se ha asignado un nombre a cada nombre largo de atributo que es una pregunta de la encuesta. Estos nombres se utilizan para crear un diccionario de atributos, *Column_dict*, para que los nombres de atributos sean codificables e intuitivos, y las preguntas preguntas completas:

- *SupportQ1*: Does your employer provide mental health benefits as part of healthcare coverage?

- *SupportQ2*: Has your employer ever formally discussed mental health (for example, as part of a wellness campaign or other official communication)?
- *SupportQ3*: Does your employer offer resources to learn more about mental health disorders and options for seeking help?
- *SupportQ4*: Is your anonymity protected if you choose to take advantage of mental health or substance abuse treatment resources provided by your employer?
- *SupportQ5*: If a mental health issue prompted you to request medical leave from work, how easy or difficult would it be to ask for that leave?
- *AttitudeQ1*: Would you feel comfortable discussing a mental health issue with your direct supervisor(s)?
- *AttitudeQ2*: Would you feel comfortable discussing a mental health issue with your coworkers?
- *AttitudeQ3*: How willing would you be to share with friends and family that you have a mental illness?
- *SupportEx1*: If you have revealed a mental health disorder to a client or businesscontact, how has this affected you or the relationship?
- SupportEx2: If you have revealed a mental health disorder to a coworker or employee, how has this impacted you or the relationship?
- *Age*: What is your age?
- *Gender*: What is your gender?
- *ResidingCountry*: What country do you live in?
- *WorkingCountry*: What country do you work in?
- *Mental Illness*: Have you ever been diagnosed with a mental health disorder?
- *Treatment*: Have you ever sought treatment for a mental health disorder from a mental health professional?
- *Year*: The year that the data was collected.

```
Column_dict = {'SupportQ1': 'Does your employer provide mental health benefits as part of healthcare coverage?',
               'SupportQ2': 'Has your employer ever formally discussed mental health (for example, as part of a wellness campaign or other official communication)?',
               'SupportQ3': 'Does your employer offer resources to learn more about mental health disorders and options for seeking help?',
               'SupportQ4': 'Is your anonymity protected if you choose to take advantage of mental health or substance abuse treatment resources provided by your employer?',
               'SupportQ5': 'If a mental health issue prompted you to request a medical leave from work, how easy or difficult would it be to ask for that leave?',
               'AttitudeQ1': 'Would you feel comfortable discussing a mental health issue with your direct supervisor(s)?',
               'AttitudeQ2': 'Would you feel comfortable discussing a mental health issue with your coworkers?',
               'AttitudeQ3': 'How willing would you be to share with friends and family that you have a mental illness?',
               'SupportEx1': 'If you have revealed a mental health disorder to a client or business contact, how has this affected you or the relationship?',
               'SupportEx2': 'If you have revealed a mental health disorder to a coworker or employee, how has this impacted you or the relationship?',
               'Age': 'What is your age?',
               'Gender': 'What is your gender?',
               'ResidingCountry': 'What country do you <strong>live</strong> in?',
               'WorkingCountry': 'What country do you <strong>work</strong> in?',
               'MentalIlness': 'Have you ever been diagnosed with a mental health disorder?',
               'Treatment' : 'Have you ever sought treatment for a mental health disorder from a mental health professional?'}
Column_dict
```

Después de eliminar los otros atributos, renombrando los nombres largos de los atributos con su clave en el diccionario, los cinco conjuntos de datos se pueden unir fácilmente utilizando la función pandas pd.concat(). He llamado al DataFrame integrado in_df.

Los pasos a seguir con el dataset del 2016 son los siguientes:

```
# Rename the columns - make the shorter
for col in df_2016.columns:
    for key in Column_dict:
        if(similar(col,Column_dict[key])>0.9):
            df_2016.rename({col:key}, axis='columns', inplace=True)
df_2016.head(1)
```

```
Others = {'If a mental health issue prompted you to request a medical leave from work, asking for that leave would be:':'SupportQ5',
          'If you have revealed a mental health issue to a client or business contact, do you believe this has impacted you negatively?':'SupportEx1',
           'If you have revealed a mental health issue to a coworker or employee, do you believe this has impacted you negatively?':'SupportEx2',
          'Have you been diagnosed with a mental health condition by a medical professional?':'MentalIlness',
          'What country do you live in?':'ResidingCountry',
          'What country do you work in?':'WorkingCountry'}
df_2016.rename(Others, axis='columns', inplace=True)
df_2016.head(1)
```

```
df_2016 = df_2016[list(Column_dict.keys())]
df_2016.head()
```

Para el resto de los años debe seguir un procedimiento similar.

Luego para cada conjunto de datos debe agregar el año:

```
# Add the column year to all datasource before combining them all
df_2016['Year'] = 2016
df_2017['Year'] = 2017
df_2018['Year'] = 2018
df_2019['Year'] = 2019
df_2020['Year'] = 2020

Column_dict['Year'] = 'The year this data was collected.'
```

y finalmente concatenar todos los conjuntos de datos en uno solo (in_df):

```
# Integrate all data
in_df = pd.concat([df_2016,df_2017,df_2018,df_2019,df_2020])
in_df.reset_index(inplace=True)
in_df.drop(columns=['index'],inplace=True)
in_df.head(1)
```

Limpieza de datos – Nivel 1

Al proceder a la integración de los datos, también nos hemos ocupado de algunas tareas de limpieza de datos de nivel I, como que los datos estén en una estructura de datos estándar y que los atributos tengan títulos codificables e intuitivos. Sin embargo, dado que in_df se integra a partir de cinco fuentes distintas, es probable que se hayan utilizado prácticas de registro de datos diferentes, lo que puede dar lugar a incoherencias en in_df.

Por ejemplo, la siguiente figura muestra lo variada que ha sido la recogida de datos para el atributo Género:

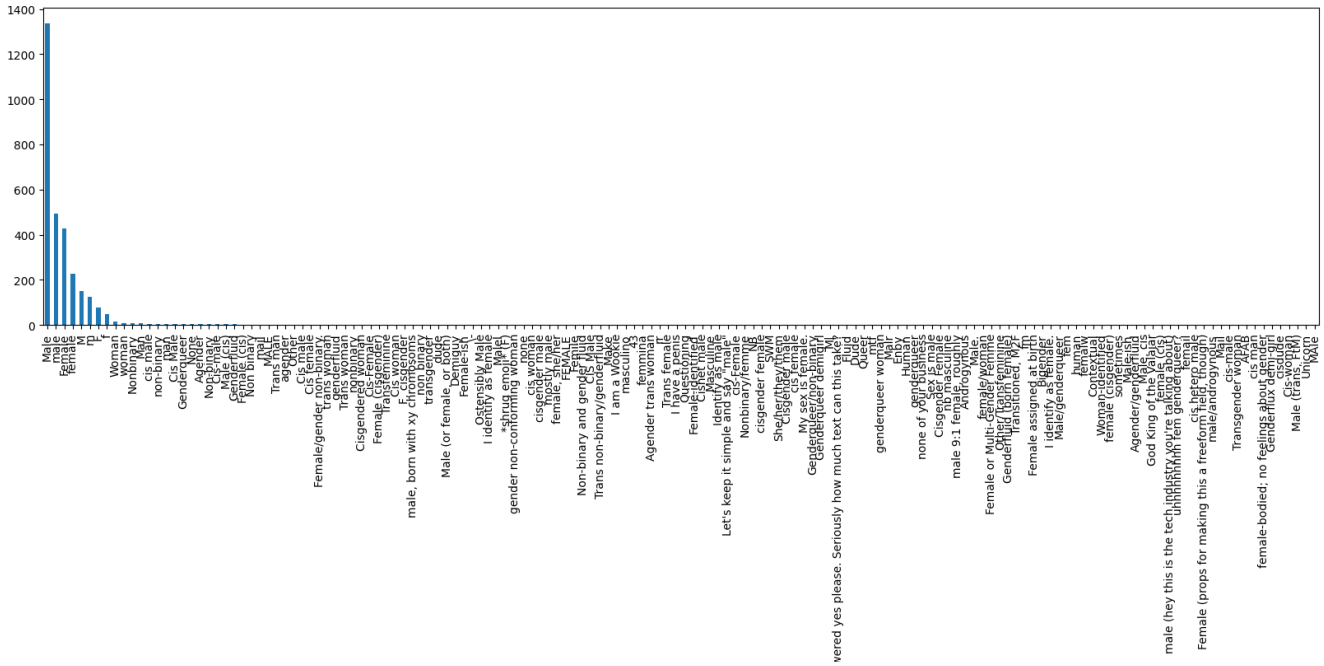


Figura 6: Estado del campo Género antes de la limpieza.

Tenemos que repasar cada atributo y asegurarnos de que no se repiten las mismas posibilidades con una redacción ligeramente distinta debido a una recopilación de datos variable o a errores ortográficos.

A continuación ejemplificamos como sería la limpieza del campo Género.

Inicialmente eliminamos cualquier espacio accidental que pudiese haber quedado en nuestro código.

```
# strip the strings so the acidental spaces do not throw off our codes
in_df.Gender = in_df.Gender.str.strip()
```

Si hacemos el conteo por valores que posee este campo podemos ver lo siguiente:

```
[In]
in_df.Gender.value_counts()
```

```
[Out]
Male 1338
male 495
Female 428
female 229
M 150 ...
cisdude 1
Cis-woman 1
Male (trans, FtM) 1
Unicorn 1
MAle 1
Name: Gender, Length: 150, dtype: int64
```

Existen varias campos con la misma información escrita de diferentes maneras y otros campos únicos que nos dificultan el procesamiento de estos datos. Para resolver esta situación hacemos las siguientes operaciones:

Unificamos todos los tipos de Male que hay en el dataset.

```
rep_dic = {'male':'Male',
        'M':'Male',
        'm':'Male',
        'Man':'Male',
        'cis male':'Male',
        'Cis Male':'Male',
        'Cis-male':'Male',
        'CIS Male':'Male',
        'Male (cis)':'Male',
        'cis hetero male':'Male',
        'Mail':'Male',
        'mail':'Male',
        'Dude':'Male',
        'Male.':'Male',
        'Cis male':'Male',
        "Let's keep it simple and say 'male'":'Male',
        'cis-male':'Male',
        'dude':'Male',
        'cis man':'Male',
```

```
        'Make':'Male',
        'cisdude':'Male',
        'Cisgender male':'Male'}
in_df.Gender = in_df.Gender.replace(rep_dic)
```

Hacemos lo mismo con Female.

```
rep_dic = {'female':'Female',
        'f':'Female',
        'F':'Female',
        'Woman':'Female',
        'woman':'Female',
        'Female (cis)':'Female',
        'Cisgender Female':'Female',
        'female (cis)':'Female',
        'Cisgendered woman':'Female',
        'Cis-Female':'Female',
        'My sex is female.':'Female',
        'fm':'Female',
        'fem':'Female',
        'Female (cisgender)':'Female',
        'female (cisgender)':'Female',
        'Female (props for making this a freeform field, though)':'Female',
        'Cis woman':'Female',
        'female/woman':'Female',
        'F, cisgender':'Female',
        '*shrug emoji* (F)':'Female',
        'Cis-woman':'Female',
        'AFAB':'Female',
        'cis-Female':'Female',
        'cis female':'Female',
        'cisgender female':'Female',
        'cis woman':'Female'}
in_df.Gender = in_df.Gender.replace(rep_dic)
```

Agrupamos todo lo demás en una categoría llamada Others.

```
def Replace_func(row):
    if(row.Gender != 'Male' and row.Gender != 'Female'):
        return 'Other'
    else:
        return row.Gender
```

```
in_df.Gender = in_df.apply(Replace_func,axis=1)
```

Una vez hecho todo podemos comprobar que todo fue bien de la siguiente manera:

[In]	[Out]

`in_df.Gender.value_counts()`	`Male 2148`
	`Female 832`
	`Other 158`
	`Name: Gender, dtype: int64`

De igual manera otros atributos necesitan limpieza, por ejemplo AttitudeQ3:

[In]	[Out]
in_df.AttitudeQ3.unique()	['Somewhat open' 'Neutral' 'Not applicable to me (I do not have a mental illness)' 'Very open' 'Not open at all' 'Somewhat not open' 5 4 10 8 3 6 2 9 7 1 0]

Aquí podemos ver que hay diversos valores que no se corresponden unos con otros. Vamos a unificar estos valores:

```
print(Column_dict['AttitudeQ3'])
in_df.AttitudeQ3.value_counts()
```

✓ 0.0s

```
How willing would you be to share with friends and family that you have a mental illness?

Somewhat open                                          640
10                                                     265
8                                                      264
7                                                      256
Very open                                              251
5                                                      219
Somewhat not open                                      214
6                                                      163
9                                                      162
Neutral                                                141
Not applicable to me (I do not have a mental illness)  112
3                                                      108
4                                                       86
2                                                       79
Not open at all                                         75
0                                                       55
1                                                       48
Name: AttitudeQ3, dtype: int64
```

```
replace_dic = {10:'Very open',
               9:'Very open',
               8:'Somewhat open',
               7:'Somewhat open',
               6:'Somewhat open',
               5:'Neutral',
               4:'Somewhat not open',
               3:'Somewhat not open',
               2:'Somewhat not open',
               1:'Not open at all',
               0:'Not open at all'}

in_df.AttitudeQ3 = in_df.AttitudeQ3.replace(replace_dic)
```

✓ 0.0s

```
print(Column_dict['AttitudeQ3'])
in_df.AttitudeQ3.value_counts()
```

✓ 0.0s

```
How willing would you be to share with friends and family that you have a mental illness?

Somewhat open                                          1323
Very open                                               678
Somewhat not open                                       487
Neutral                                                 360
Not open at all                                         178
Not applicable to me (I do not have a mental illness)   112
Name: AttitudeQ3, dtype: int64
```

Para ver todo el proceso de preprocesamiento seguido en este caso de estudio puede leer el Jupyter Notebook que se encuentra publicado en Github.

De igual manera puede leer sobre otros dos casos de estudio sobre preparación de datos en los capítulos 16 y 17 del libro (Jafari, 2022).

Evaluación de modelos

El desarrollo de modelos es un proceso iterativo, y la etapa de formación del modelo va seguida de las etapas de validación y actualización (Figura 7).

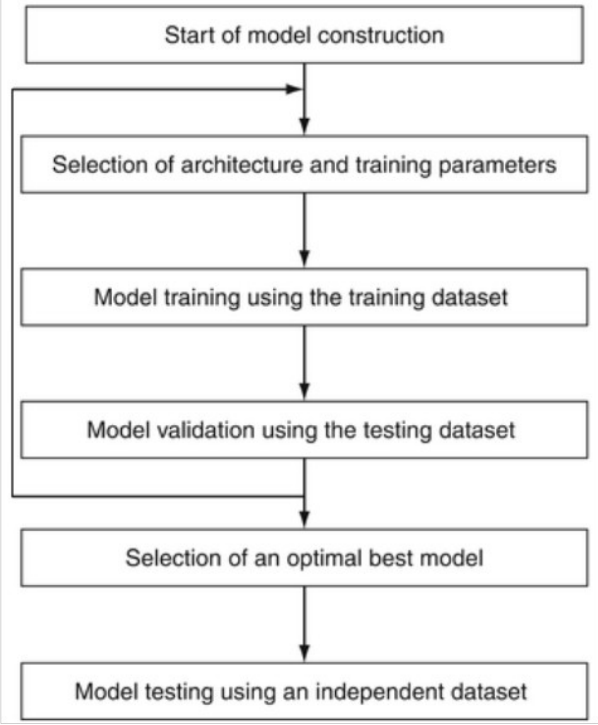


Figura 7: Proceso de desarrollo de modelos de aprendizaje automático.

En esta sección ejemplificaremos el uso de las diferentes métricas que podemos usar para comprobar la calidad de los modelos que entrenemos. Para ello usaremos dos modelos entrenados, uno para resolver un problema de regresión (regresión lineal), y otro para un problema de clasificación (random forest).

Lo primero es cargar los conjuntos de datos que vamos a utilizar. Ambos conjuntos de datos son referentes al acontecimiento del Titanic, uno de ellos fue utilizado para el modelo de regresión lineal y el otro fue utilizado para entrenar el modelo de clasificación.

```
df_reg = pd.read_csv("titanic_regression.csv")
df_reg.head()
```

✓ 0.0s

	Fare	class_ticket__2	class_ticket__3
0	7.2500	0	1
1	71.2833	0	0
2	7.9250	0	1
3	53.1000	0	0
4	8.0500	0	1

```
df_clf = pd.read_csv('titanic_classification.csv')
df_clf.head()
```
✓ 0.0s

	Pclass	Sex	Age	SibSp	Parch	Fare	Emb_C	Emb_Q	Emb_S	Survived
0	3	0	22.0	1	0	7.2500	0	0	1	0
1	1	0	38.0	1	0	71.2833	1	0	0	1
2	3	0	26.0	0	0	7.9250	0	0	1	1
3	1	0	35.0	1	0	53.1000	0	0	1	1
4	3	0	35.0	0	0	8.0500	0	0	1	0

Lo siguiente es cargar los modelos entrenados anteriormente:

```
with open("./saved_models/titanic_regression.pkl", 'rb') as f:
    reg = pickle.load(f)

with open("./saved_models/random_forest_clf.pkl","rb") as f:
    rf = pickle.load(f)
```
✓ 0.0s

Regresión

A continuación utilizaremos las métricas para evaluar el modelo de regresión. Por tanto, utilizaremos el modelo *reg* y el conjunto de datos *df_reg*.

Primero debemos importar las funciones de métricas que nos interesa utilizar.

```
from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score
from math import sqrt
```

Usemos el modelo de regresión cargado para predecir los valores del conjunto de datos.

```
X = df_reg.drop(columns=["Fare"])
y = df_reg['Fare'].values
y_pred = reg.predict(X)
```

Ahora con los valores de calculados usaremos algunas métricas:

```
print('Mean Absolute Error = {}'.format(mean_absolute_error(y, y_pred)))
print('Root Mean Squared Error = {}'.format(sqrt(mean_squared_error(y,
y_pred))))
print('R Squared Score = {}'.format(r2_score(y, y_pred)))
```

✓ 0.0s

```
Mean Absolute Error = 19.628101662449893
Root Mean Squared Error = 41.28596607888486
R Squared Score = 0.36324718649102006
```

Clasificación

Para probar las métricas de evaluación de modelos de clasificación, usaremos el conjunto de datos *df_clf* y el modelo usado para clasificar *rf.* Lo primero es importar las funciones que nos permitirán calcular las métricas.

```
from sklearn.metrics import (accuracy_score, confusion_matrix,
precision_score,recall_score, f1_score)
```

Luego seleccionamos los features y la clase que usaremos para predecir los elementos con el modelo creado.

```
X = df_clf.iloc[:, :-1] #seleccionamos todos los elementos menos el último
y = df_clf.iloc[:, -1]  #seleccionamos el último como la clase

y_pred = rf.predict(X)
y_pred_probs = rf.predict_proba(X)[:, 1]
```

Ahora podemos probar las métricas relacionadas con la evaluación de modelos de clasificación.

```
print('Accuracy Score = {}'.format(accuracy_score(y, y_pred)))
```

✓ 0.0s

```
Accuracy Score = 0.7474747474747475
```

```
print(confusion_matrix(y_pred=y_pred, y_true=y))
```

✓ 0.0s

```
[[497  52]
 [173 169]]
```

```
print('Precision Score = {}'.format(precision_score(y, y_pred)))
print('Recall Score = {}'.format(recall_score(y, y_pred)))
```

✓ 0.0s

```
Precision Score = 0.7647058823529411
Recall Score = 0.49415204678362573
```

```
print('F1 Score = {}'.format(f1_score(y, y_pred)))
```

✓ 0.0s

```
F1 Score = 0.6003552397868561
```

Para graficar el área bajo la curva podemos hacer lo siguiente:

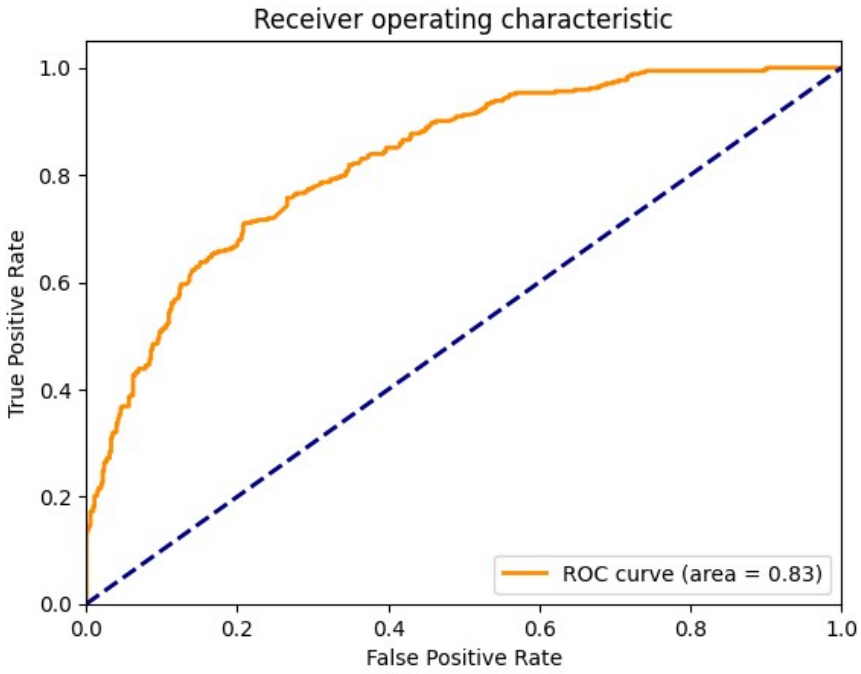
```
import matplotlib.pyplot as plt
from sklearn.metrics import roc_curve, auc

# Compute ROC curve and AUC
fpr, tpr, thresholds = roc_curve(y.values, y_pred_probs)
roc_auc = auc(fpr, tpr)

# Plot ROC curve
plt.plot(fpr, tpr, color='darkorange', lw=2, label='ROC curve (area = %0.2f)' % roc_auc)
plt.plot([0, 1], [0, 1], color='navy', lw=2, linestyle='--')
plt.xlim([0.0, 1.0])
plt.ylim([0.0, 1.05])
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('Receiver operating characteristic')
plt.legend(loc="lower right")
plt.show()
```

✓ 0.2s

Lo cual produce la siguiente gráfica.



Ejercicios

Debe completar los ejercicios que se relacionan a continuación en un Jupyter Notebook, puede usar el que se encuentra en la siguiente URL. Luego debe subirlo al espacio creado en el EVA para su evaluación.

1. Realice el proceso de integración de datos para los siguientes conjuntos de datos:
   - heart-disease.cleveland.csv
   - heart-disease.hungarian.csv
   - heart-disease.switzerland.csv
2. Realice el proceso de preparación de datos aplicando las técnicas que crea necesarias al conjunto de datos *boston.* Para cargar ese conjunto de datos ejecute el siguiente código:

```
import pandas as pd
from sklearn.datasets import load_boston
df = pd.DataFrame(boston['data'], columns = boston['feature_names'])
df['target'] = boston['target']
df
```

3. Evalúe los modelos houseprices_reg.pkl y student_performance_reg.pkl tomando en cuenta que los conjuntos de datos usados para entrenarlos son los siguientes houseprices_regression.csv y students_performance_evaluation.csv respectivamente.
4. Evalúa el rendimiento del modelo de árboles de decisión decision_tree_clf.pkl que fue entrenado usando el conjunto de datos breast_cancer.csv. Para ello utilice métricas de tipo numéricas y gráficos para saber si ese modelo brinda buenos resultados en la clasificación.

Elaborado por: M.Sc. Angel Alberto Vazquez Sánchez
Profesor Auxiliar del Departamento de Inteligencia Computacional.