**FORMATO PARA EL DESARROLLO DE COMPONENTE FORMATIVO**

|  |  |
| --- | --- |
| **PROGRAMA DE FORMACIÓN** | Limpieza y transformación de datos con Python |

|  |  |  |  |
| --- | --- | --- | --- |
| **COMPETENCIA** | **220501114** sistematizar datos masivos de acuerdo con métodos de analítica y herramientas tecnológicas. | **RESULTADOS DE APRENDIZAJE** | **220501114-2.** Preparar los datos para el análisis de acuerdo con los requerimientos técnicos. |

|  |  |
| --- | --- |
| **NÚMERO DEL COMPONENTE FORMATIVO** | 01 |
| **NOMBRE DEL COMPONENTE FORMATIVO** | Depurando los datos |
| **BREVE DESCRIPCIÓN** | En este componente formativo, se describe la importancia de la limpieza y transformación de los datos, además, se dan a conocer los pasos necesarios para realizar estos procesos; por otro lado, se reconocerán los comandos de las principales librerías para el análisis y manipulación de los datos del lenguaje de programación *Python*. |
| **PALABRAS CLAVE** | dato, atipicidad, *dummy*, discreto, continuo, escalamiento, significancia, *machine learning* |

|  |  |
| --- | --- |
| **ÁREA OCUPACIONAL** | 6 - Ventas y servicios |
| **IDIOMA** | Español |

1. **TABLA DE CONTENIDOS**

**Introducción**

**1. Limpieza de los datos**

1.1 Describir los datos

1.2 Datos nulos

1.3 Imputación de valores nulos

1.4 Datos sucios

1.5 Datos atípicos

**2. Transformación de los datos**

2.1 Nuevas columnas

2.2 Filtrado

2.3 Agrupación

2.4 Columnas *dummy*

2.5 Discretización o *Binning*

2.6 Escalamiento

**Síntesis**

1. **INTRODUCCIÓN**

La limpieza de datos implica observar más de cerca los problemas en los datos que se han seleccionado para incluir en un análisis determinado. En este componente se brindarán los elementos necesarios para utilizar las principales librerías de *Python*, indispensables para la limpieza y transformación de los datos. El siguiente video expone a modo general en qué consiste la limpieza de datos.

|  |
| --- |
| **CF02\_video\_introducción** |

1. **DESARROLLO DE CONTENIDOS**
2. **Limpieza de datos**

Este proceso consiste en detectar, corregir o eliminar registros corruptos o inexactos de un conjunto tablas o base de datos; dentro de dicha limpieza, se identifican las partes incompletas, incorrectas, inexactas o irrelevantes de los datos,para su posterior sustitución, modificación o eliminación.

Realizar este proceso en el mundo de la ciencia de datos es fundamental, pues no puede confiarse a ciegas en la veracidad de los datos que se toman de las diferentes fuentes de información. Es necesario depurarlos y tomar los que realmente contribuyan al cumplimiento de los objetivos organizacionales propuestos.

Desde esta perspectiva, es importante tener claridad sobre los problemas más comunes que se pueden llegar a encontrar durante este proceso. Estos se presentan a continuación:

|  |
| --- |
| **CF02\_1\_infografía\_problemas comunes** |

Para entender mejor este tema, en el siguiente video se expondrá un ejemplo que contribuirá a comprender a qué se enfrenta alguien cuando recolecta información desde diversas fuentes de información.

|  |
| --- |
| **CF02\_1\_video\_problemas en la recolección de la información** |

Si bien no se conocen todas las numerosas anomalías que se pueden encontrar, es importante tener claridad en los tipos y la estructura de datos de la colección, de tal manera que se puedan quitar, modificar o eliminar; lo cierto es que los datos deben mantener siempre una consistencia de lo que representan.

Es importante tener en cuenta que, para iniciar con el proceso de limpieza de datos y búsqueda de inconsistencias, se debe realizar la importación de librerías, proceso que se explicó en el componente anterior; pues es necesario trabajar con datos unificados en los tres tipos de archivos de diferentes formatos, para posteriormente guardarlos en un archivo de formato CSV, sobre el cual se basará el desarrollo de este componente. Observe la siguiente figura:

**Figura 1**

*Archivos unificados*

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Sobre este archivo unificado, se empezará a realizar la búsqueda de inconsistencias, pues la información aquí recolectada se tomó desde diferentes fuentes de información, por lo que seguramente se presentarán anomalías en la estructura de los datos.

A continuación, se empezará con la descripción de los datos, para determinar la identificación de cada columna y los tipos de datos; de esta forma, se podrá determinar la consistencia de los mismos.

* 1. **Describir los datos**

Existen muchas formas de describir los datos, pero normalmente se tienen en cuenta estos aspectos:

* **Cantidad de datos:** número de registros, campos y tablas seleccionadas en la etapa anterior.
* **Tipo de valores:** pueden ser numéricos, categóricos o booleanos. Se debe prestar atención que al momento de cargar los datos no se modifiquen estos tipos.
* **Esquemas de codificación:** en algunas ocasiones, los valores de las bases de datos son representaciones de características, como, por ejemplo, el género o tipo de producto.

El siguiente recurso educativo amplía la información sobre el proceso para realizar la descripción de los datos:

|  |
| --- |
| **CF02\_1\_1\_video\_experto\_** **Proceso para realizar la descripción de los datos.** |

A continuación, se empezarán a manipular los datos nulos, para lo cual es necesario conocer diferentes técnicas para corregir este tipo de situaciones.

* 1. **Datos nulos**

Lo primero que se debe contemplar es que, cuando no se encuentran ciertos datos de un campo, al cargar los datos, a estos se les debe asignar un valor *NaN,* que indica que son datos nulos. Es importante tener en cuenta que el tratamiento de los datos depende del objetivo del estudio de los mismos.

En la mayoría de los casos, se opta por ignorar toda la fila; en otros casos, dependerá de la cantidad de valores nulos, y, a partir de las filas ignoradas, se establece si es posible continuar con el estudio de los datos o si se hace inviable debido a la falta de registros.

Otra opción es completar los valores vacíos, sin embargo, se debe considerar con detalle con qué valor se va a completar, puesto que cualquier valor que se ponga podrá aumentar el sesgo del conjunto de datos.

Para ver la cantidad de valores nulos del conjunto de datos que se ha venido trabajando, es necesario utilizar el siguiente comando.

*df.isnull().sum()*

Este comando permitirá detectar y contar cada uno de los valores nulos encontrados en cada columna de los datos que se están estudiando, tal como se muestra en la siguiente figura.

**Figura 2**

*Valores nulos por columna*

Interfaz de usuario gráfica

Descripción generada automáticamente con confianza media

Es importante tener en cuenta que:

Una salida rápida para la solución de valores nulos puede ser eliminar todas las filas donde se presente un valor nulo en cualquiera de las columnas, pero observe que, en la figura anterior, la variable *color* presenta un número alto de valores nulos, de manera que, al realizar la eliminación con esa técnica, se estaría descartando mucha información importante.

Pero, ¿cómo determinar lo que se considera mucha información?

Para responder a esta pregunta, es necesario realizar el cálculo porcentual de los valores nulos con respecto al total de registros. Este procedimiento debe contar con todos los valores nulos y dividirlos sobre el total; se utiliza el siguiente comando para obtener ese resultado.

*porcentaje = ((df.isnull().sum() / len(df))\*100).sort\_values(ascending = False)*

*print(porcentaje)*

Al aplicar dicho comando, se puede determinar cuál sería el porcentaje de eliminación si se realiza un proceso de descarte general.

Tal como se observa en la siguiente figura 3 el porcentaje que se estaría eliminando, solamente teniendo en cuenta la variable *color*, sería de aproximadamente el 60 % del total de los datos; si dentro de ese porcentaje no se cruza con registros de la edad, entonces se eliminaría otro 12 % adicional, lo que resultaría en un error, ya que se estaría descartando una muestra bastante representativa, equivalente a un 72 %, y si se procede con la eliminación de esta manera, seguramente los resultados estarán bastante sesgados o insuficientes para realizar la analítica.

**Figura 3**

*Cálculo del porcentaje de valores nulos*

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente con confianza media

Para entender mejor este procedimiento, observe con atención el documento anexo denominado: **Casos datos nulos.**

* 1. **Imputación de datos nulos**

En caso de que la cantidad de valores nulos sea lo suficientemente baja para reemplazar todos los valores con un valor específico, se puede realizar la imputación de valores nulos, la cual toma valores estadísticos, como la media, la mediana o la moda, para realizar el cambio; tal como se presenta en el siguiente recurso:

|  |
| --- |
| **CF02\_1\_3\_pestañas\_** **Imputación de datos nulos.** |

Llevando estos métodos a la práctica, se debe definir cuál sería el método correcto para realizar la imputación de los valores nulos para la variable *edad*, que aún presenta registros inferiores con relación a las demás.

Se sabe que la descripción de los datos arrojó valores máximos atípicos y realizar la imputación con las medias puede generar inconsistencias, ya que la media es muy sensible a los valores atípicos altos.

Para efectos prácticos, a continuación, se darán a conocer todos los comandos para los métodos vistos que se pueden aplicar a la variable *edad*, y al final, se escogerá el método definitivo que se aplicará a los datos.

|  |
| --- |
| **CF02\_1\_3\_infografía\_** **comandos para cada método** |

Lo que realizan estos métodos es revisar, dentro de la columna *edad*, los valores que se encuentran vacíos y reemplazarlos con el método seleccionado; para este caso, el método que se utilizará es el de k vecinos. Después de ejecutar el método para k vecinos y visualizar los resultados con el comando *df.info()*, se obtiene la siguiente información.

**Figura 4**

*Resultado sin valores nulos*

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Se sigue avanzando en el perfeccionamiento de los datos y, como se puede observar, ya no se tiene presencia de valores nulos. Primeramente, se eliminaron muestras pequeñas, que no ponían en riesgo la consistencia de los datos; se descartaron columnas con valores superiores al 50 % de ausencia de datos y se realizó la imputación de los datos para así complementar los datos faltantes a la edad. Todas las columnas quedaron con un total de 2993, de los 3000 registros con que se empezó, y una columna menos, que tenía mucha información faltante.

Recuerde que en la imputación de los datos es importante conocer muy bien el comportamiento de las variables; no se escoge un método u otro simplemente por gusto, aquí se deben tener en cuenta factores como la variabilidad de los datos o los valores permitidos reales que se puedan asignar cuando se habla de imputación de valor fijo.

Ya terminado el proceso del tratamiento de datos nulos, ahora, se procede con la búsqueda de datos sucios.

* 1. **Datos sucios**

Cuando un dato es erróneo o inconsistente se le llama dato sucio**.** Este tipo de dato es mucho más difícil de encontrar, pues estos pueden derivar de errores generados al momento de ser registrados. Para estos casos, existen métodos preventivos, que minimizan la cantidad de datos sucios, ya que los procesos correctivos, y en especial la limpieza de datos, suelen ser muy costosos y laboriosos.

Para ahondar de manera detallada en este proceso, se invita a visualizar el siguiente recurso:

|  |
| --- |
| **CF02\_1\_4\_video\_experto\_** **Datos sucios** |

* 1. **Datos atípicos**

Cuando se registra un dato cuyos valores son muy diferentes a las observaciones del mismo grupo de datos, se le llama dato atípico**.** Este tipo de dato se diferencia de los datos nulos y los datos sucios porque existe y no es consecuencia de un error en el registro o compilación de datos.

Un valor atípico puede ser indicativo de que el dato pertenece a una población diferente del resto de las muestras establecidas. Normalmente, son excluidos del análisis debido a que distorsionan los resultados.

Para valores numéricos, el método más impartido académicamente, por su sencillez y resultados, es el *test de Tukey*, que toma como referencia la diferencia entre el primer cuartil y el tercer cuartil de los datos.

Una forma de visualizar este test es usando el diagrama de caja, el cual se genera usando el siguiente comando:

*sns.catplot(x = "Modelo", data = df, kind = "box", aspect = 1.5);*

**Figura 5**

*Diagrama de caja y bigotes*

Gráfico

Descripción generada automáticamente

Tal como se muestra en la Figura 5, se selecciona la columna *modelo* para este ejemplo. Se puede observar que aquellos vehículos anteriores al año 1978 son muy pocos comparados con el resto de vehículos; por ende, son considerados datos atípicos.

Si se desea eliminar los datos atípicos de esta columna, se puede utilizar un filtro que seleccione aquellos cuyo modelo sea menor a 1978. Para realizar esa acción, ejecute el siguiente comando:

*seleccion=df[df['Modelo']< 1978].index*

*de=df.drop(seleccion, inplace = True)*

*df.shape*

Como se observa a continuación, se realizó la selección de los registros que cumpliesen el parámetro, y posteriormente, se eliminaron, como se refleja en el total, que pasó de 2993 a 2925.

**Figura 6**

*Eliminación de registros atípicos*

Interfaz de usuario gráfica, Texto, Aplicación, Word

Descripción generada automáticamente

Entonces, con base en lo anterior, si se grafica nuevamente el resultado, se obtiene lo siguiente, aquí ya no se observan aquellos valores que se han descartado.

**Figura 7**

*Gráfica sin datos atípicos*

Gráfico

Descripción generada automáticamente

Para entender mejor el tema, se realizará el mismo procedimiento para las variables *Edad* e *Hijos*. Para obtener los resultados gráficos, se deben ejecutar los siguientes comandos:

|  |
| --- |
| **CF02\_1\_5\_sliders\_Edad e hijos** |

Para ambos campos, se observan datos atípicos, por lo que se procederá a eliminarlos, estableciendo un rango en el que solo se permiten los datos que cumplan la condición.

Los siguientes comandos ayudan con el procedimiento para descartar esos registros atípicos. En cuanto a la variable *edad*, sólo se dejarán registros que estén en el rango de 15 a 80 años, y para *hijos*, datos que sean menores a 5.

También, se deben quitar las columnas que no se vayan a utilizar; en este caso, la que se creó automáticamente al unificar los archivos, llamada *Unnamed: 0* y la columna *id;* tal como se presenta a continuación:

|  |
| --- |
| **CF02\_1\_5\_infografía\_comandos para registros atípicos** |

El resultado de la ejecución de los comandos anteriores se ve reflejado en la Figura 10, donde el total de registros también va disminuyendo y ya no se visualizan las columnas que ya se han quitado.

**Figura 10**

*Comandos para eliminar datos atípicos*

Interfaz de usuario gráfica, Texto, Aplicación, Chat o mensaje de texto

Descripción generada automáticamente

Para el caso de las variables categóricas, como *Marca*, se puede usar un diagrama de frecuencia, el cual se puede generar usando el siguiente comando:

*plot = df['Marca'].value\_counts().plot(kind='bar', figsize=(16, 6), title='Marcas de Automovil')*

**Figura 11**

*Histograma de frecuencias por marca*

Gráfico, Histograma

Descripción generada automáticamente

Como se puede ver en la Figura 11, se seleccionó la columna *Marca*para este ejemplo, y el resultado es que la mayoría de los autos son de la marca Ford, y muy pocos, de las marcas Foose, Daihatsu, Tesla, Citroen, entre otras.

Si se desean eliminar los datos atípicos de esta columna, se debe establecer una cantidad mínima de casos, para filtrar aquellas categorías que no son frecuentes. Para este filtro, se usará un mínimo de 10 repeticiones, en cuyo caso aquellas marcas que tengan un número inferior de repeticiones serán consideradas como datos atípicos.

**Figura 12**

*Marcas cuya frecuencia sea menor de 10*

Interfaz de usuario gráfica, Texto, Aplicación

Descripción generada automáticamente

Como se muestra en la Figura 12, haciendo uso de la función *value\_counts*, se obtiene la frecuencia de cada valor categórico dentro de la columna *Marca* y se asigna el valor *True* a aquellas cuyo conteo es menor que 10 y *False* a aquellas cuyo conteo sea mayor o igual que 10. Luego, se asigna a la variable *condición*.

Ya que lo que interesa es la lista de marcas que cumplan la *condición*, se extraen de la lista *condición* aquellas marcas cuyo valor asignado fue *True* y se asignan a la variable *filtrarmarcas*.

**Figura 13**

*Filtrado de marcas con menos de 10 registros*

Interfaz de usuario gráfica

Descripción generada automáticamente con confianza media

Entonces, haciendo uso de la función *isin*, se filtran todas aquellas filas cuya marca se encuentre en lista *filtrarmarcas*, dejando solo aquellas que no se encuentran en la lista de filtrado, usando el comando:

*df[~df["Marca"].isin(filtrarmarcas)]*

En la anterior figura se observa que el total, hasta este punto, es de 2004; todo el proceso de limpieza tuvo un costo de 996 registros menos, a pesar de que se fue cuidadoso con las exclusiones o eliminaciones, pero se presentó un gran número de datos vacíos y atípicos en la edad.

Tenga en cuenta que cada caso es particular y se debe conocer muy bien el negocio para determinar qué se puede o no dejar dentro de las variables; en ocasiones, puede que solo se requiera un parte de la limpieza, lo importante es conocer muy bien el dato y darle la integridad para que sea consistente.

1. **Transformación de los datos**

La transformación de datos es el proceso de convertir datos crudos en un formato o estructura que se acomode mejor al objetivo del estudio.

|  |  |
| --- | --- |
| Por ejemplo, se pueden transformar valores numéricos a categóricos, como las edades a *mayor de edad* o *menor de edad.* También se puede transformar un valor sin cambiar el tipo de dato, como lo sería pasar *Edad* a *Año de nacimiento*. | Multinational group of people isolated on white background. Children, adults and teenagers stand together. Vector illustration |

Asimismo, los datos transformados pueden remplazar los datos que se encontraban en una columna o pueden ser colocados en nuevas columnas, como se aprecia en el siguiente ejemplo.

**Tabla 1**

*Ejemplo de transformación de datos*

|  |  |  |
| --- | --- | --- |
| Datos original: **Edad** | Nueva columna: **Clasificación** | Nueva columna: **Año de Nacimiento** |
| 17 | Menor de edad | 2005 |
| 35 | Mayor de edad | 1986 |

Si se presenta mucha variabilidad en los datos, estos pueden ser agrupados en rangos, por ejemplo, la edad puede que esté dispersa en edades de 1 a 80 años, por lo que serían muchas variables independientes sueltas, lo que dificulta un poco el análisis, por lo que se puede recurrir a crear rangos para edades, que vayan de 10 en 10; y en vez de analizar 80 variables independientes, se analizarían 8.

Como se puede ver en el siguiente ejemplo, el objetivo es agrupar en un rango las edades pertenecientes a ese segmento:

**Tabla 2**

*Rango de edades por segmento*

|  |  |  |
| --- | --- | --- |
| **Edades** | **Rango** | **Nombre Rango** |
| 1, 3,5,6 | 1-10 | A |
| 12, 15 | 10-20 | B |
| 25 | 20-30 | C |
| 31, 38, 39 | 30-40 | D |
| 42, 43 | 40-50 | E |
|  | 50-60 | F |
| 62, 63, 65 | 60-70 | G |
| 79 | 70-80 | H |

Desde la información base, se pueden crear muchas otras variables que contribuyan a realizar un análisis completo.

|  |  |
| --- | --- |
| Por ejemplo, si se tienen la cantidad y el valor que representan ventas de artículos, se puede crear una nueva variable multiplicándolas, para obtener el total de venta para cada registro.  O también, en caso de discriminar un valor con base en el impuesto, pero solamente se cuenta con el total y se sabe el porcentaje de impuesto, se pueden realizar estos cálculos y ponerlos en nuevas columnas. |  |

Son múltiples las acciones que se pueden realizar con la información base, la cual permite crear nuevas variables que facilitarán un mejor entendimiento de los datos. A continuación, se conocerán algunos ejemplos de cómo se pueden realizar estas transformaciones.

* 1. **Nuevas columnas**

Son una forma de transformación de los datos iniciales, ya que, a partir de uno o varios campos, se puede crear una nueva columna, con información nueva; estas columnas nuevas dependen de los requerimientos que se haya solicitado.

|  |
| --- |
| **CF02\_2\_3\_video\_experto\_Nuevas columnas** |

* 1. **Filtrado**

El filtrado de datos se utiliza cuando se desea retirar o clasificar, del conjunto de datos, aquellos registros que cumplan ciertas condiciones.

Recuerde que esta variable fue construida anteriormente, en el apartado denominado nuevas columnas, desde el campo *Modelo*, cumpliendo con el filtro de clasificación, para el cual la condición era que, para modelos inferiores al año 2000, se guardaba una nueva etiqueta denominada *Modelo Viejo,* y para los superiores a ese año, el valor asignado fue *Modelo Nuevo.*

Para realizar una gráfica que permita determinar el valor porcentual que representa cada una de estas etiquetas, se deben ejecutar las siguientes líneas de comando:

*total\_rango\_modelo = df['ModeloTransformado'].groupby(df['ModeloTransformado']).count()*

*labels = ["Modelo Nuevo", "Modelo Viejo"]*

*colors = sns.color\_palette('Set2')[0:5]*

*plt.pie(total\_rango\_modelo, labels =labels , colors = colors, autopct='%.1f%%')*

*plt.show()*

Como se observa en la Figura 14, el resultado es la cantidad porcentual de automóviles nuevos y viejos.

**Figura 14**

*Gráfica porcentual variable ModeloTransformado*

Interfaz de usuario gráfica, Gráfico, Aplicación

Descripción generada automáticamente

Ese filtro de clasificación se utilizó para crear una nueva columna, que en estos momentos permite determinar porcentualmente el número de automóviles viejos y nuevos, continuando con el siguiente filtro que se utilizará para descartar información; por ejemplo, si solo se desea trabajar con los modelos que representan una mayor presencia, se tendría que dejar los registros que se identifiquen como modelo nuevo, ya que son el 58 % total de los datos.

Esto se puede lograr usando una lista de condicionales que responden si el auto es *Modelo Nuevo o Modelo Viejo*. Luego, esta lista de condicionales es pasada al comando *loc*, el cual se encargará de devolver cada una de las posiciones donde, según la lista de condicionales, la condición haya sido verdadera.

Para obtener ese resultado, se debe ejecutar la siguiente línea de comandos, donde se determina el filtro de registros que cumplan con el valor *Modelo Nuevo* de la columna *ModeloTransformado.*

*filtros=df['ModeloTransformado']=='Modelo Nuevo'*

Y para descartar el resto de los datos, se debe ejecutar el filtro dentro de la colección de los datos de la siguiente manera:

*df.loc[filtros]*

**Figura 15**

*Filtrando para descartar registros*

Interfaz de usuario gráfica, Tabla

Descripción generada automáticamente

En la Figura 15, se muestra cómo, al filtrar los datos, se han dejado solo los registros que cumplen con la condición Modelo Nuevo, y también se observan los cambios en el total de los registros de la parte inferior de la imagen, pasando de 2004 a 1206.

* 1. **Agrupación**

La agrupación de datos se realiza cuando no es significativo el detalle de un registro sino su valor como grupo, ya sea su sumatoria o frecuencia. Esta se aplica para variables categóricas o variables numéricas no continuas. Al igual que con los casos anteriores, se va a utilizar como ejemplo el calcular el valor promedio por marca de auto en la tabla de vehículos, explicado a través del siguiente recurso:

|  |
| --- |
| **CF02\_2\_5\_pasos horizontales\_Agrupación** |

* 1. **Columnas *dummy***

Estas variables son datos ficticios que necesitan ser creados, ya que los algoritmos de entrenamiento o predicción solo trabajan con valores numéricos. En estos casos, se deben transformar los valores que sean categóricos a valores numéricos. Por ejemplo, si se usa la columna *Marca* y se desea transformar a numérica, se pueden crear columnas para cada marca, y luego asignar 1 si pertenece a esa marca o 0 si no pertenece.

Para entender un poco mejor, observe el siguiente cuadro, donde se tiene la variable *Género,* con los valores *F* y *M* en la columna original; cuando se pasa a *dummie*s, se crearía una columna nueva para cada dato diferente, si este coincide, se coloca el número 1, de lo contrario, se coloca un 0.

|  |
| --- |
| **Genero** |
| F |
| F |
| M |
| F |
| M |
| M |
| F |

|  |  |
| --- | --- |
| **Dummy\_F** | **Dummy\_M** |
| 1 | 0 |
| 1 | 0 |
| 0 | 1 |
| 1 | 0 |
| 0 | 1 |
| 0 | 1 |
| 1 | 0 |

Siguiendo con el ejemplo de la variable *Marca,* se puede hacer usando el comando *get*\_*dummies,* para obtener una tabla donde se indica a qué marca pertenece el auto; para tal fin, se debe hacer uso del siguiente comando. (Véase Figura 16)

*dfdummy=pd.get\_dummies(df['Marca'], prefix='Marca')*

*dfdummy*

**Figura 16**

*Columnas Dummy según la marca*

Tabla

Descripción generada automáticamente con confianza media

Luego, se toma esta tabla y se concatena a la tabla inicial, usando el comando *concat*. Para esto, se debe hacer uso de los siguientes comandos.

*df=pd.concat([df,dfdummy], axis=1)*

*df*

**Figura****17**

*Columnas dummy concatenadas al conjunto de datos inicial*

Interfaz de usuario gráfica, Aplicación

Descripción generada automáticamente

Como se puede observar en la figura anterior, la cantidad de columnas aumentó a 65, de las cuales 55 son las columnas *dummy* que se generaron; es allí donde se debe pensar si es conveniente usar esta columna para el análisis de los datos; como siempre, todo dependerá de los requerimientos del negocio.

Ahora, observe el mismo procedimiento, pero, esta vez, para la variable *Género*. El proceso es el mismo, sólo debe cambiar por la columna *Género.*

*dfdummy=pd.get\_dummies(df['Genero'], prefix='Genero')*

*dfdummy*

El resultado se puede ver en la Figura 18, donde se realiza la transformación de una variable categórica, como lo es el género, representado en F o M, a un valor numérico, de 0 donde no encuentre la coincidencia y 1 para cuando corresponda a la variable.

**Figura 18**

*Columnas Dummy para el Género*

Tabla

Descripción generada automáticamente con confianza media

En este caso, se crearon dos columnas nuevas, lo que quiere decir que el número de columnas dependerá de la variabilidad que exista en cada una de las filas de las columnas categóricas que se quieran transformar a *dummy.*

* 1. **Discretización o *Binning***

El proceso de discretización corresponde a la transformación de una variable numérica en una categórica. Esta técnica se utiliza para reducir los efectos de los errores de pequeñas observaciones.

Para comprender esta parte, se tiene una encuesta en la que se pregunta por el nivel de satisfacción, y el resultado es capturado de 1 a 5. Los resultados se necesitan agrupados de la siguiente manera:

Los valores 0 y 1 se etiquetan como Muy bajo; el 2, Bajo; el 3, Medio; y 4 y 5, Alto. El resultado quedaría de la siguiente manera:

**Tabla 3**

*Discretización de datos*

|  |  |
| --- | --- |
| **Respuesta** | **Variable discreta** |
| 0 | Muy bajo |
| 1 | Muy bajo |
| 4 | Alto |
| 3 | Medio |
| 2 | Bajo |
| 3 | Medio |
| 4 | Alto |
| 5 | Alto |
| 2 | Bajo |
| 1 | Muy bajo |

Cuando existe mucha dispersión de los datos, es recomendable recurrir a la discretización, de forma que se permita un mejor entendimiento de ellos.

Previamente, se utilizó este método en la creación de nuevas columnas, donde se convierte un valor numérico discreto, como es el año del modelo, a una variable categórica, como es *Modelo Nuevo* o *Modelo Viejo*.

Otra forma de discretizar los datos es utilizando etiquetas que identifiquen rangos de ciertas variables, como lo sería *Menor de edad* y *Mayor de edad* para el campo *Edad***;** los menores de 18 años tomarían la primera etiqueta, el resto, la segunda etiqueta. (Véase Figura 19)

Para realizar este procedimiento, ejecute las siguientes líneas de comando.

*df['EdadDiscreta']=np.where(df["Edad"]<18,"Menor de Edad", "Mayor de Edad")*

df.head()

**Figura 19**

*Discretización del campo Edad*

Interfaz de usuario gráfica, Texto

Descripción generada automáticamente

|  |  |
| --- | --- |
| La discretización de variables es muy útil en aprendizaje supervisado. El científico de datos puede decidir qué variables conviene que sean discretizadas antes de aplicar los modelos y cuáles pueden funcionar mejor de forma continua. Aquellas variables que puedan ser descritas, desde el punto de vista del algoritmo, en forma de intervalos son buenas candidatas a la discretización, es decir, aquellas que convenga describir en grupos (tales como bajo, medio y alto) podrían ser discretizadas. | Ilustración del concepto de consultoría vector gratuito |

Para visualizar todas las variables discretas, se debe ejecutar el siguiente comando, que permitirá ver las columnas categóricas nuevas que se obtuvieron desde una variable numérica.

*df[['ModeloTransformado','Rango\_Edad','EdadDiscreta']]*

Este comando selecciona solo las columnas que se coloquen dentro de dobles corchetes. En este caso, se usa para visualizar las nuevas variables categóricas, pero puede ser utilizado para ver un conjunto de columnas cualquiera, así:

**Figura 20**

*Variables discretas*

Interfaz de usuario gráfica

Descripción generada automáticamente con confianza media

* 1. **Escalamiento**

Algunos algoritmos diferencian la importancia de ciertas columnas con base en la escala que tengan sus valores para poder generar un resultado, es por esto que es importante que todos los datos estén dentro de un mismo rango y tengan una misma desviación media.

El método que se implementará funciona de la siguiente manera: por ejemplo, se tienen dos columnas con 10 datos, estos con diferentes valores; para el valor 1, se puede decir que son rangos para el precio de un artículo; mientras que para el valor 2, se establece la representación de las unidades vendidas; claramente existirá una amplia diferencia en sus rangos.

Para normalizar estos dos rangos, lo primero es determinar el valor máximo, que sería 1500 para la columna denominada valor 1, y 20, para la columna denominada valor 2. Posteriormente, se divide cada valor de la fila por el valor máximo, como se muestra a continuación.

**Tabla 4**

*Escalamiento de datos*

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **No** | **Valor 1** | **Operación 1** | **Valor normalizado 1** | **Valor 2** | **Operación 2** | **Valor normalizado 2** |
| 1 | 1200 | 1200/1500 = | 0.800 | 5 | 5/20 = | 0.250 |
| 2 | 200 | 200/1500 = | 0.133 | 2 | 2/20 = | 0.100 |
| 3 | 300 | 300/1500 = | 0.200 | 4 | 4/20 = | 0.200 |
| 4 | 400 | 400/1500 = | 0.267 | 5 | 5/20= | 0.250 |
| 5 | 300 | 300/1500 = | 0.200 | 7 | 7/20 = | 0.350 |
| 6 | 100 | 100/1500 = | 0.067 | 20 | 8/20 = | 1.000 |
| 7 | 50 | 50/1500 = | 0.033 | 9 | 9/20 = | 0.450 |
| 8 | 20 | 20/1500 = | 0.013 | 10 | 10/20 = | 0.500 |
| 9 | 5 | 5/1500 = | 0.003 | 15 | 15/20 = | 0.750 |
| 10 | 1500 | 1500/1500 = | 1.000 | 5 | 20/20 = | 0.250 |

A pesar de que las columnas *Valor 1* y *Valor 2* en sus datos originales presentan diferencias importantes, cuando se normalizan se comportan de la misma manera, ya que se encuentran en el mismo rango, y este es el resultado que se busca cuando se realiza dicho proceso: que todas las variables tengan la misma importancia.

Pasando al ejemplo práctico, su objetivo es llevar todas las variables numéricas a una misma escala, pues, si se observan los valores de cada una de las columnas, sus datos varían significativamente. Para observar estos datos, se debe utilizar el siguiente comando:

*df.describe()*

Al ejecutarlo, se genera el resultado que se aprecia en la Figura 21, donde los datos varían significativamente. Si se dejan las variables de esta manera, los algoritmos de entrenamiento para predicción de datos le estarían dando mayor importancia a la variable *Valor*, que es la que representa datos más altos, mientras que a *Hijos,* le daría menor relevancia, al ser la de menor valor.

**Figura 21**

*Valores mínimos y máximos de campos numéricos*

Interfaz de usuario gráfica

Descripción generada automáticamente con confianza media

Lo que se quiere es que se tenga igual importancia para todas las variables, como se veía anteriormente en las columnas *dummy,* que se representan todas por igual, ya sea 0 o 1; y este también sería el objetivo para las variables numéricas, como en el ejemplo descrito anteriormente, en la cual se llevan a una escala entre 0 y 1; la diferencia con las *dummy* es que existirá más variabilidad en ese rango.

Para este proceso, existen diferentes técnicas; para esta ocasión, se utilizará el máximo absoluto, descrito en el ejemplo anterior, el cual consiste en dividir cada dato entre el valor máximo absoluto, el cual se puede calcular usando las funciones *max*() y *abs*(), por lo que, para realizar el escalamiento a todas las variables numéricas, se deben ejecutar los siguientes comandos.

*df\_max=df[['Modelo','Valor', 'Hijos', 'Edad']]*

*for column in df\_max.columns:*

*df\_max[column]=df\_max[column]/df\_max[column].abs().max()*

*display(df\_max*)

**Figura 22**

*Escalamiento de las columnas numéricas*

Interfaz de usuario gráfica, Tabla

Descripción generada automáticamente

El resultado que se muestra en la Figura 22 permite ver que cada una de las variables numéricas se encuentran en una escala de 0 a 1. Al obtener el valor máximo de cada columna y dividir cada valor por ese valor máximo, se tiene una normalización de los datos. Así, los algoritmos tendrán menos problemas para realizar un correcto reconocimiento de la importancia de estas columnas, ya que serán tratadas por igual.

Para estar seguros del procedimiento, se debe usar el comando que se usó anteriormente para la descripción de los datos.

df\_max.describe()

Como se puede observar en la Figura 23, el valor mínimo para cada una de las variables es 0, de la misma manera que el máximo es 1.

**Figura 23**

*Variables numéricas normalizadas*

Tabla

Descripción generada automáticamente con confianza media

Al tener las variables en la misma escala, los algoritmos de entrenamiento no le darán mayor importancia a una variable específica, por lo que todas serán tratadas con la misma importancia.

1. **síntesis**

Existen diversos medios por los cuales una empresa puede acceder a diferentes tipos de información, pues estos se han convertido en el insumo principal para la toma de decisiones, la planeación de procesos y la ejecución de tareas. En este orden de ideas, se debe tener claro que existen diferentes fuentes y distintos tipos de datos que pueden influir en el resultado del análisis.

Desde esta perspectiva, es fundamental conocer todos aquellos elementos que permiten seleccionar la información, categorizarla, limpiarla y transformarla, con el fin de que su óptimo manejo permita el alcance de los objetivos y requerimientos del negocio.

Una breve revisión de los temas vistos se encuentra en el siguiente esquema:

|  |
| --- |
| **CF02\_infografía\_Síntesis** |

1. **ACTIVIDADES DIDÁCTICAS**

|  |  |
| --- | --- |
| **DESCRIPCIÓN DE ACTIVIDAD DIDÁCTICA** | |
| Nombre de la Actividad | Tipos de archivos |
| Objetivo de la actividad | Identificar los principales comandos utilizados para limpieza y transformación de los datos. |
| Tipo de actividad sugerida | Preguntas de selección múltiple con única respuesta. |
| **Archivo de la actividad**  **(Anexo donde se describe la actividad propuesta)** | Formatos DI: CF02\_Actividad didáctica 01 |

|  |  |
| --- | --- |
| **DESCRIPCIÓN DE ACTIVIDAD DIDÁCTICA** | |
| Nombre de la Actividad | Complete el comando |
| Objetivo de la actividad | Escribir correctamente los comandos necesarios para la limpieza y la transformación de datos según los casos presentados. |
| Tipo de actividad sugerida | Arrastre |
| **Archivo de la actividad**  **(Anexo donde se describe la actividad propuesta)** | Formatos DI: CF02\_Actividad didáctica 02 |

1. **MATERIAL COMPLEMENTARIO**

|  |  |  |  |
| --- | --- | --- | --- |
| **Tema** | **Referencia APA del Material** | **Tipo de material**  **(Video, capítulo de libro, artículo, otro)** | **Enlace del Recurso o**  **Archivo del documento o material** |
| 1. Limpieza de Datos | Torres, A. (2021). *Limpieza de datos en Pandas: Explicado con ejemplos*. freeCodeCamp. <https://www.freecodecamp.org/espanol/news/limpieza-de-datos-en-pandas-explicado-con-ejemplos/> | Página Web | <https://www.freecodecamp.org/espanol/news/limpieza-de-datos-en-pandas-explicado-con-ejemplos/> |
| 1. Limpieza de Datos | Skills For Industry. (2021). *Python desde cero: Limpieza de Datos* [Video]. YouTube. <https://www.youtube.com/watch?v=rOzxoJYoWj8> | Video | <https://www.youtube.com/watch?v=rOzxoJYoWj8> |
| 2. Transformación de los datos | McKinney, W. y Pandas Development Team. (2022). *Pandas: powerful Python data analysis toolkit. Release 1.4.2*. Pandas. <https://pandas.pydata.org/pandas-docs/version/1.4.2/pandas.pdf> | PDF | <https://pandas.pydata.org/pandas-docs/version/1.4.2/pandas.pdf> |
| 2.3. Agrupación | InteractiveChaos. (2022). *Agrupaciones en dataframes*. <https://interactivechaos.com/es/manual/tutorial-de-pandas/agrupaciones-en-dataframes> | Página Web | <https://interactivechaos.com/es/manual/tutorial-de-pandas/agrupaciones-en-dataframes> |

1. **GLOSARIO**

|  |  |
| --- | --- |
| **TÉRMINO** | **SIGNIFICADO** |
| **Tipo de dato** | espacio en memoria que especifica una restricción, por ejemplo, *int64*, *float64*, *object*, *bool*. |
| **Atípico** | valor que no corresponde a un conjunto de valores definidos. |
| ***Dummy*** | variable ficticia utilizada para convertir valores categóricos a numéricos. |
| **Escala** | valores definidos por un valor mínimo o máximo. |
| **Imputación** | reemplazar registros vacíos mediante técnicas como valor fijo, promedio, k vecinos entre otros. |
| **Valores nulos** | ausencia de información en algún registro de los campos de una colección de datos. |
| **Filtrado** | acción realizada para obtener información detallada, clasificar o agrupar. |
| **Agrupar** | acción utilizada para unir por rangos que permitan entender mejor la información. |

1. **REFERENCIAS BIBLIOGRÁFICAS**

González, R. (s. f.). *Python para todos*. <https://persoal.citius.usc.es/eva.cernadas/informaticaparacientificos/material/libros/Python%20para%20todos.pdf>

López, L. (2021). *Limpieza de datos con Python*. Medium. <https://lauralpezb.medium.com/limpieza-de-datos-con-python-48d436ca9ace>

Ministerio de Tecnologías de la Información y las Comunicaciones [MinTIC]. (2018). *Herramientas de analítica para la explotación de datos*. <https://herramientas.datos.gov.co/sites/default/files/2020-11/Inventario%20herramientas%20anal%C3%ADtica_0.pdf>

Rodríguez, S. y Rivera, A. (2018). *La importancia de la limpieza de datos en Big Data*.

1. **CONTROL DEL DOCUMENTO**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Nombre** | **Cargo** | **Dependencia** | **Fecha** |
| **Autor(es)** | Rónald Alexánder Vacca Ascanio | Experto Temático | Regional Distrito Capital - Centro de Diseño y Metrología | Abril 2022 |
| Jeimy Lorena Romero Perilla | Diseñadora Instruccional | Regional Norte de Santander - Centro de la Industria, la Empresa y los Servicios | Abril 2022 |
| Carolina Coca Salazar | Asesora Metodológica | Regional Distrito Capital - Centro de Diseño y Metrología | Abril 2022 |
| Darío González | Corrector de Estilo | Regional Distrito Capital - Centro de Diseño y Metrología | Junio 2022 |
| Miroslava González Hernández | Diseñadora Instruccional | Regional Santander - Centro Industrial del Diseño y la Manufactura | Julio 2023 |
| Rafael Neftalí Lizcano Reyes | Responsable Equipo de Desarrollo Curricular | Regional Santander - Centro Industrial del Diseño y la Manufactura | Julio 2023 |

1. **CONTROL DE CAMBIOS**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Nombre** | **Cargo** | **Dependencia** | **Fecha** | **Razón del cambio** |
| **Autor(es)** |  |  |  |  |  |