Data Preparation II

David Guzmán Leyva - A01706417
Enrique Santos Fraire - A01705746
Leonardo Alvarado Menéndez - A01705998
Oscar Enrique Delgadillo Ochoa - A01705935

Índice

Índice **2**

Bitácora de cambios **5**

Introducción **5**

DataSet **5**
Descripción del data set 5

Seleccionar los datos **6**
Base lógica para la inclusión/exclusión 6

Preparación de los datos **6**

Not moving **7**
Formateo 7
Atributos derivados 7
Formateo 7
Limpieza 7
Atributos derivados 7
Formateo 7
Limpieza 7
Formateo 7
Limpieza 7
Registros generados 7
Código 8

Antenas **8**
Transformar valores de atributos existentes 8
Atributos derivados 8
Registros generados 8
Atributos derivados 8
Limpieza 8
Registros generados 8
Código 8

Cleaning data **9**
Transformar valores de atributos existentes 9
Formateo 9
Registros generados 9
Atributos derivados 9
Limpieza 9
Formateo 9
Limpieza 9
Registros generados 9
Código 10

Merge data **10**
Atributos derivados 10
Transformar valores de atributos existentes 10
Atributos derivados 10
Registros generados 10
Formateo 10
Limpieza 10
Registros generados 10
Código 10

Viajes2 **11**
Transformar valores de atributos existentes 11
Atributos derivados 11
Limpieza 11
Registros generados 11
Código 11

OrigenDestino **11**
Transformar valores de atributos existentes 11
Registros generados 11
Código 12

Heatmap **12**
Limpieza 12
Atributos derivados 12
Registros generados 12
Código 12

Preparación de los datos **14**

Getting_model **14**
Transformar valores de atributos existentes 14
Atributos derivados 15
Limpieza 15
Formateo 15
Registros generados 15
Atributos derivados 15
Formateo 15
Atributos derivados 15
Formateo 15
Registros generados 15
Formateo 15
Atributos derivados 16
Registros generados 16
Código 16

Get_%_of_homeoffice **16**

Limpieza 16
Registros generados 16
Limpieza 16
Atributos derivados 16
Registros generados 16
Limpieza 16
Formateo 17
Registros generados 17
Código 17

obtain_data_Chile 17
Formateo 17
Limpieza 17
Registros generados 17
Transformar valores de atributos existentes 17
Registros generados 17
Formateo 17
Registros generados 18
Transformar valores de atributos existentes 18
Registros generados 18
Formateo 18
Registros generados 18
Transformar valores de atributos existentes 18
Registros generados 18
Formateo 18
Atributos derivados 18
Limpieza 18
Registros generados 19
Código 19

merge_data_Chile 19
Formateo 19
Registros generados 19
Código 19

Bitácora de cambios

Versión	Fecha	Autor(es)	Modificaciones
1.0	04/11/2022	David Guzmán Leyva Enrique Santos Fraire Leonardo Alvarado Menéndez Oscar Enrique Delgadillo Ochoa	Línea base
1.1	17/11/2022	David Guzmán Leyva Enrique Santos Fraire Leonardo Alvarado Menéndez Oscar Enrique Delgadillo Ochoa	Actualización de diagramas Modelado
1.2	23/11/22	David Guzmán Leyva Enrique Santos Fraire Leonardo Alvarado Menéndez Oscar Enrique Delgadillo Ochoa	Actualización de diagramas Introducción

Introducción

Para llegar a una solución adecuada con nuestros datos se tuvo que pasar por un preprocesamiento, el cual dividimos en 2 fases. La primera en la cual llegamos a la matriz de origen-destino que será representada en nuestro mapa de calor y la segunda donde se realizó a la preparación de nuestro dataset que será utilizado para entrenar el modelo a partir de pasos anteriores de la primera fase.

DataSet

Descripción del data set

Nuestro dataset final va a tener los siguientes parámetros:

- PHONE_ID: identificador único del usuario que está realizando el viaje..
- Comuna Origen: lugar en el que empezó el viaje.
- Comuna Destino: destino final.
- Start_time: hora en la que inició el viaje.
- End_time: hora en la que finalizó el viaje.

Seleccionar los datos

Base lógica para la inclusión/exclusión

Tomamos la decisión de no eliminar ningún atributo debido a que los datos proporcionados (timestamp, phone_id, bts_id, lat y lon) por el Centro de Data Science de Chile son útiles para obtener información relevante como, distancias, velocidades, tiempo de movilidad y comunas, que son datos primordiales para obtener la matriz de viajes solicitada.

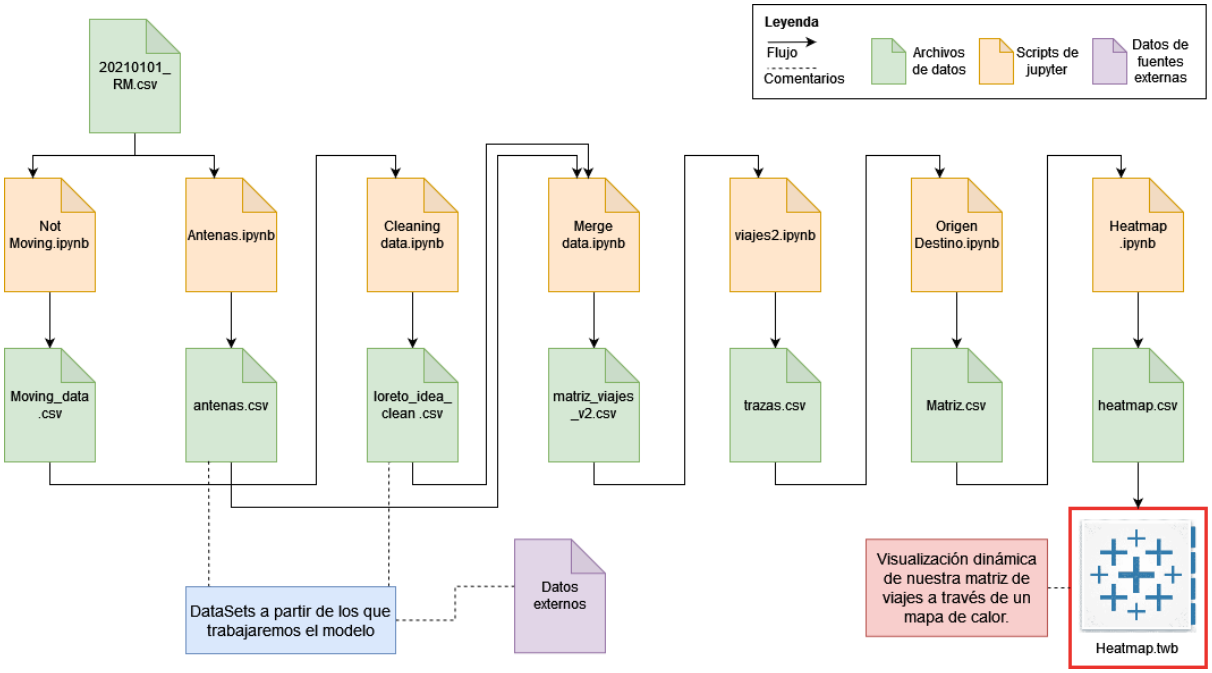
Preparación de los datos

A continuación se muestra un diagrama general sobre el proceso de preparación de nuestros datos. Consta de 7 partes distribuidas en jupyter notebooks, partiendo del csv que se nos otorgó, realizando las fases de preparación que aplicaban en cada uno y generando un csv nuevo por cada jupyter, finalizando con el DataSet limpio para la matriz de viajes, que posteriormente se verá representado en un mapa de calor dentro de Tableau.

En resumen cada fase realizó lo siguiente:

1. Not moving - DataSet original a DataSet con gente que si se mueve.
2. Antenas - Limpieza de antenas repetidas con sus coordenadas y comunas.
3. Cleaning data - Gente que se está moviendo a menos de 150 km/h.
4. Merge - Combinar los datos anteriores en un solo dataset de PHONE_ID, timestamp, coordinates, tiempo_min y comuna.
5. Viajes2 - Agrupar las conexiones a la misma coordenada en un rango continuo de tiempo.
6. OrigenDestino - Genera la matriz de viajes siguiendo las condiciones para que sean posibles.
7. Heatmap -Conteo de viajes que hay de una comuna a otra para poder visualizarse.

En seguida, se describen a detalle las actividades realizadas en cada fase.



Not moving

Formateo

Partimos del DataSet original y los organizamos por *PHONE_ID* y *timestamp*, de manera que sean separados cada usuario y su recorrido a lo largo de todo el día.

Atributos derivados

Mediante la fórmula de Haversine obtenemos la distancia entre dos puntos utilizando la latitud y longitud (columnas lat y lon) en una nueva columna *dis_trans*.

Formateo

Reiniciamos los índices ya que las operaciones los alteran.

Limpieza

Borramos la columna anterior de *index* ya que se reiniciaron anteriormente.

Atributos derivados

Reemplazamos los NaN's por ceros, esto debido a que al inicio de cada id no hay distancias que comparar y los NaN's interfieren en los cálculos.

Mediante una función iterativa y un acumulador, separamos aquellos usuarios que sí tuvieron un recorrido de los que no en 2 arreglos distintos.

Formateo

Asignamos al *PHONE_ID* como index.

Limpieza

Gracias al arreglo de los usuarios que no se mueven, utilizamos dicha variable como parámetro para borrar esos registros del DataSet.

Formateo

Reiniciamos los índices para recuperar el *PHONE_ID* como columna.

Limpieza

Eliminamos la columna de distancia recorrida (*dis_trans*), pues ya no es necesaria.

Registros generados

Se guarda el nuevo DataSet en un csv (*moving_data.csv*).

Código

https://github.com/Davidguzley/Movilidad-Chile/blob/main/Codigo/Not_moving.ipynb

Antenas

Transformar valores de atributos existentes

Partimos del DataSet original y cambiamos el tipo de dato de las columnas de *lat* y *lon* de float a string para poder realizar una concatenación.

Atributos derivados

Creamos una columna de *coordinates*, que es la concatenación de lat y lon separadas por una coma.

Registros generados

Mediante la librería geopy, utilizamos las *coordinates* y en un columna *direction* se le aplica una función reversa con geolocator y obtenemos un objeto de datos en las coordenadas. extrayendo con una función adicional únicamente la comuna y verificando la unicidad de 52 comunas.

Atributos derivados

En una columna *direc* utilizamos una función lambda para extraer la dirección de la columna creada anteriormente.

A partir de direc, utilizamos una función que busca en la dirección la comuna en la que se encuentra, comparándolo con un arreglo que contiene las 52 comunas y guardando el resultado en la columna *comuna*.

Limpieza

Eliminamos las columnas de *direction* y *direc*, pues ya no son necesarias.

Verificamos que no haya datos nulos o inusuales. Encontramos que quedaba un registro nulo, por lo que manualmente buscamos su comuna y se la asignamos.

Registros generados

Se guarda el nuevo DataSet en un csv (*antenas.csv*).

Código

https://github.com/Davidguzley/Movilidad-Chile/blob/main/Codigo/Antenas.ipynb

Cleaning data

Transformar valores de atributos existentes

Partimos del DataSet *moving_data*, generado en el paso de **Not Moving**, y cambiamos el tipo de dato de la columna de *timestamp* a pandas datetime para realizar los cálculos.

Formateo

Organizamos los datos por *PHONE_ID* y *timestamp*, de manera que sean separados cada usuario y su recorrido a lo largo de todo el día.

Registros generados

Se genera una nueva columna llamada *drop* que contiene ceros, esto para identificar los registros que serán borrados bajo cierta condición.

Atributos derivados

Mediante la fórmula de Haversine obtenemos la distancia entre dos puntos utilizando la latitud y longitud (columnas lat y lon) en una variable. Posteriormente se obtiene el tiempo transcurrido restando la hora del registro anterior al actual en otra variable. Finalmente se obtiene la velocidad dividiendo la distancia sobre el tiempo y se guarda el resultado en una tercera variable.

En caso de que la velocidad sea mayor a 150, se registra un 1 en la columna *drop* del registro y se vuelven a hacer los cálculos con el registro siguiente hasta que la velocidad calculada sea menor a 150.

Limpieza

Creamos un nuevo dataset a través de una máscara que nos devuelva únicamente los registros que se hayan mantenido con un *drop* de 0.

Formateo

Reiniciamos los índices ya que las operaciones los alteran.

Limpieza

Borramos la columna anterior de *index* ya que se reiniciaron anteriormente.
Borramos la columna *Unnamed: 0* ya que no es necesaria.

Registros generados

Se guarda el nuevo DataSet en un csv (*loreto_idea_clean.csv*).

Código

https://github.com/Davidguzley/Movilidad-Chile/blob/main/Codigo/Cleaning_data.ipynb

Merge data

Atributos derivados

Partimos de los DataSet *loreto_idea_clean*, generado anteriormente en el paso de **Cleaning data**. Creamos la columna de *tiempo_min* multiplicando *tiempo_trans* por 60, obteniendo así el tiempo en minutos.

Transformar valores de atributos existentes

Cambiamos el tipo de dato de la columna de *timestamp* a pandas datetime para realizar los cálculos.
Cambiamos el tipo de dato de las columnas de *lat* y *lon* de float a string para poder realizar una concatenación.

Atributos derivados

Creamos una columna de *tiempo_min*, que es la resta del tiempo del registro actual menos el anterior en horas.
Creamos una columna de *coordinates*, que es la concatenación de lat y lon separadas por una coma.

Registros generados

Generamos un nuevo DataSet donde conservamos solo *PHONE_ID*, *timestamp*, *coordinates* y *tiempo_min*.

Unimos nuestros datos con el DataSet *antenas*, generado anteriormente en el paso de **Antenas**, utilizando como parámetro de unión la columna *coordinates*.

Formateo

Organizamos los datos por *PHONE_ID* y *timestamp*, de manera que sean separados cada usuario y su recorrido a lo largo de todo el día.

Reiniciamos los índices ya que las operaciones los alteran.

Limpieza

Borramos la columna anterior de *index* ya que se reiniciaron anteriormente.
Borramos la columna *Unnamed: 0* ya que no es necesaria.

Registros generados

Se guarda el nuevo DataSet en un csv (*matriz_viajes_v2.csv*).

Código

https://github.com/Davidguzley/Movilidad-Chile/blob/main/Codigo/Merge_data.ipynb

Viajes2

Transformar valores de atributos existentes

Partimos del DataSet *matriz_viajes_v2*, generado en el paso de **Merge Data**, y cambiamos el tipo de dato de la columna de *timestamp* a pandas datetime para realizar los cálculos.

Atributos derivados

Reemplazamos los NaN's por ceros, esto para facilitar las operaciones al inicio de cada id.

Limpieza

Borramos la columna *Unnamed: 0* ya que no es necesaria.

Registros generados

A través de un diccionario auxiliar se generó un nuevo DataFrame, donde en vez de tener cada registro de los id's conectados a la misma antena en diferentes horas ahora es un registro unico que indica la comuna en la que se encuentra, cuál fue su hora de entrada y su hora de salida de la conexión a esa antena.

Se guarda el nuevo DataSet en un csv (*trazas.csv*).

Código

https://github.com/Davidguzley/Movilidad-Chile/blob/main/Codigo/viajes2.ipynb

OrigenDestino

Transformar valores de atributos existentes

Partimos del DataSet *trazas*, generado en el paso de **Viajes2**, y cambiamos el tipo de dato de las columnas de *Hora_entrada* y *Hora_salida* a pandas datetime para realizar los cálculos.

Registros generados

A través de un diccionario auxiliar se generó un nuevo DataFrame, donde se gardará la matriz de viajes, indicando el traslado del id, desde su comuna de origen hasta su comuna de destino, con su hora de salida y su hora de llegada, así como la distancia que recorrió y la duración del viaje. Esto siguiendo las condiciones de que para terminar un viaje debe mantenerse por lo menos 20 minutos quieto, su duración no debe exceder las 2 horas ni los 25 kilómetros de distancia.

Se guarda el nuevo DataSet en un csv (*Matriz.csv*).

Código

https://github.com/Davidguzley/Movilidad-Chile/blob/main/Codigo/OrigenDestino.ipynb

Heatmap

Limpieza

Partimos del DataSet *Matriz*, generado en el paso de **OrigenDestino**, y borramos la columna Unnamed: 0 ya que no es necesaria.

Atributos derivados

Creamos una columna de *Concatenados*, que es la concatenación de *Comuna_Origen* y *Comuna_Destino* separadas por una coma.

Registros generados

Se crea un arreglo con el conteo de todos diferentes viajes y se guarda en un nuevo DataFrame.

Con un diccionario auxiliar se crea una matriz que guarda el origen, el destino y el número de viajes realizados entre ambos.

Se genera otro DataFrame con este diccionario.

Se guarda el nuevo DataSet en un csv (*heatmap.csv*).

Código

https://github.com/Davidguzley/Movilidad-Chile/blob/main/Codigo/Heatmap.ipynb

Preprocesado para el dataset del Modelado

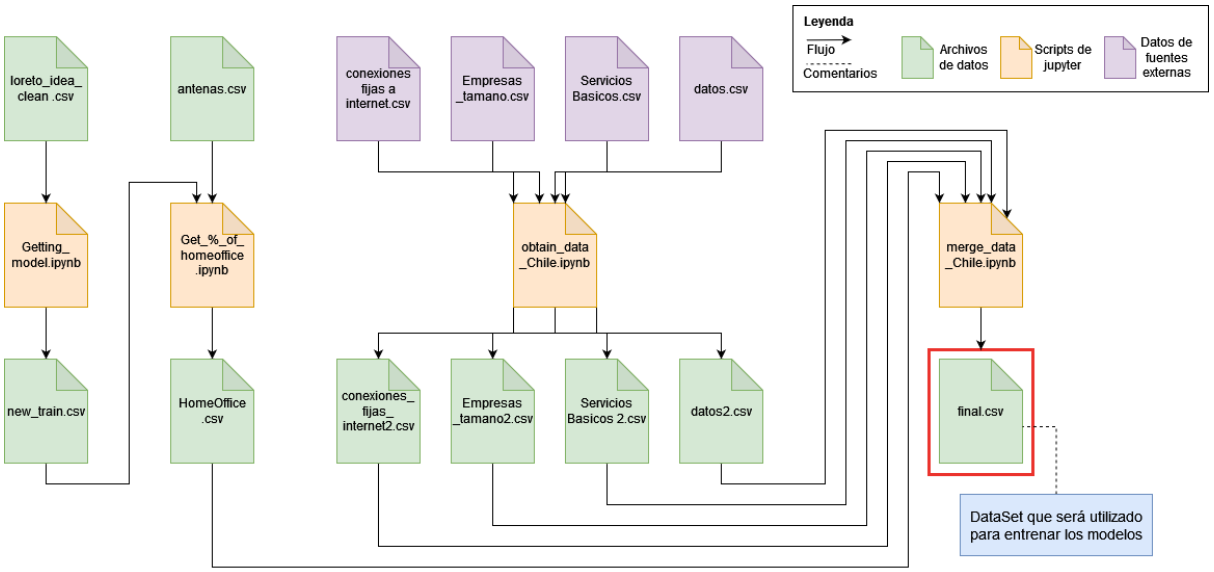
Preparación de los datos

Una vez que llegamos a nuestra matriz de viajes para el mapa de calor, partimos de los DataSets generados en los pasos anteriores para generar la información necesaria para entrenar nuestros modelos de predicción.

A continuación se muestra un diagrama general sobre el proceso de preparación de nuestros datos, así como el entrenamiento de los distintos modelos. Consta de N partes distribuidas en jupyter notebooks, partiendo de dos csv's obtenidos en la primera iteración de preparación de datos.

En resumen cada fase realizó lo siguiente:

1. Getting_model - Se obtienen los labels para identificar si es HomeOffice según las conexiones de los usuarios en horario laboral y en horario de casa.
2. Get_%_of_homeoffice - Se obtiene el porcentaje de HomeOffice por comuna.
3. obtain_data_Chile - Limpieza de datos de fuentes externas para poderlos utilizar.
4. merge_data_Chile - Combinar los datos anteriores en un solo dataset con los indicadores necesarios para entrenar los modelos.



Getting_model

Transformar valores de atributos existentes

Partimos del DataSet *loreto_idea_clean*, generado en el paso de **Cleaning Data**, y cambiamos el tipo de dato de la columna de *timestamp* a pandas datetime para realizar los cálculos.

Cambiamos el tipo de dato de las columnas de *lat* y *lon* de float a string para poder realizar una concatenación.

Atributos derivados

Creamos una columna de *coordinates*, que es la concatenación de lat y lon separadas por una coma.

A través de la función de datetime dt extraemos únicamente la hora de *timestamp* y la guardamos en una nueva columna llamada *Hora*.

Limpieza

Borramos las columnas de *Unnamed: 0, drop, bts_id, lat* y *lon* ya que no son necesarias.

Formateo

Establecemos las columnas de *PHONE_ID* y *timestamp* como los índices.

Registros generados

Se crean dos copias del DataSet para realizar filtros distintos

Atributos derivados

En la primera copia filtramos los datos por horas para establecer el horario de casa, siendo desde la medianoche hasta las siete de la mañana.

Agrupamos datos por *PHONE_ID* y sacamos una lista con todas las coordenadas.

Obtenemos el valor más repetido de todas las *coordinates* aplicando una moda.

Formateo

Reiniciamos los índices para separar *PHONE_ID* y *timestamp*.

Atributos derivados

En la segunda copia filtramos los datos por horas para establecer el horario de trabajo, siendo desde las nueve de la mañana hasta las seis de la tarde.

Agrupamos datos por *PHONE_ID* y sacamos una lista con todas las coordenadas.

Obtenemos el valor más repetido de todas las *coordinates* aplicando una moda.

Formateo

Reiniciamos los índices para separar *PHONE_ID* y *timestamp*.

Registros generados

Unimos las dos copias nuevamente en uno solo usando como llave *PHONE_ID*.

Formateo

Renombramos las columnas de *Coordinates_x* y *Coordinates_y* por *Trabajo* y *Casa* respectivamente para identificar las antenas fácilmente.

Atributos derivados

A través de una función verificamos si las antenas de *Trabajo* y *Casa* son iguales, de ser verdadero se asigna el valor de 1 una nueva columna llamada *HomeOffice?*, de lo contrario es 0.

Registros generados

Se guarda el nuevo DataSet en un csv (*new_train.csv*).

Código

https://github.com/Davidguzley/Movilidad-Chile/blob/main/Codigo/Modeling/getting_model.ipynb

Get_%_of_homeoffice

Limpieza

Partimos del DataSet *new_train*, generado en el paso de **Getting_model**, y del DataSet *antenas*, generado en el paso de **Antenas**. Borramos la columna Unnamed: 0 en ambos ya que no es necesaria.

Registros generados

Unimos nuestros datos utilizando como parámetro de unión la columna *Trabajo* de *new_train* y la columna *coordinates* de *antenas*.

Limpieza

Borramos las columnas de *Trabajo, Casa* y *coordinates* ya que no son necesarias.

Atributos derivados

Con ayuda de un diccionario auxiliar, se cuentan cuántos 1 y 0 hay por cada comuna, para posteriormente obtener el porcentaje de HomeOffice que hay.

Registros generados

Se pasa el diccionario a un dataframe.

Limpieza

Borramos las columnas de *0* y *1* ya que no son necesarias.

Formateo

Renombramos la columna de *2* por *%_of_homeoffice* para identificar claramente los datos.

Registros generados

Se guarda el nuevo DataSet en un csv (*HomeOffice.csv*).

Código

https://github.com/Davidguzley/Movilidad-Chile/blob/main/Codigo/Modeling/Get_%25_of_homeoffice.ipynb

obtain_data_Chile

Formateo

Partimos del DataSet *conexiones fijas a internet*, el cual se obtuvo de manera externa a través del Sistema Integrado de Información Territorial (SIIT) de Chile, siendo una fuente de información pública. Renombramos la columna de *2021* por *Cantidad de Conexiones de internet fijas*.

Limpieza

Borramos las columnas de *Variable* y *Unidad territorial* ya que no son necesarias.

Registros generados

Se guarda el nuevo DataSet en un csv (*conexiones_fijas_internet2.csv*).

Transformar valores de atributos existentes

Partimos del DataSet *Empresas_tamano*, el cual se obtuvo de manera externa a través del Sistema Integrado de Información Territorial (SIIT) de Chile, siendo una fuente de información pública. Se acomodan por medio de un diccionario auxiliar los datos por comuna.

Registros generados

Se pasa el diccionario a un dataframe.

Formateo

Renombramos las columnas del 0 al 9 por la lista de indicadores que se encontraban en el DataSet original.

Registros generados

Se guarda el nuevo DataSet en un csv (*Empresas_tamano2.csv*).

Transformar valores de atributos existentes

Partimos del DataSet *Servicios Basicos*, el cual se obtuvo de manera externa a través del Sistema Integrado de Información Territorial (SIIT) de Chile, siendo una fuente de información pública. Se acomodan por medio de un diccionario auxiliar los datos por comuna.

Registros generados

Se pasa el diccionario a un dataframe.

Formateo

Renombramos las columnas del 0 al 2 por la lista de indicadores que se encontraban en el DataSet original.

Registros generados

Se guarda el nuevo DataSet en un csv (*Servicios Basicos 2.csv*).

Transformar valores de atributos existentes

Partimos del DataSet *datos*, el cual se obtuvo de manera externa a través del Sistema Integrado de Información Territorial (SIIT) de Chile, siendo una fuente de información pública. Se acomodan por medio de un diccionario auxiliar los datos por comuna, tipo de empresa y número de trabajadores.

Registros generados

Se pasa el diccionario a un dataframe.

Formateo

Renombramos las columnas del 0 al 43 por la lista de indicadores que se encontraban en el DataSet original.

Atributos derivados

Agregamos columnas para agrupar empresas y trabajadores por informáticas y no informáticas.

Limpieza

Borramos las columnas originales, dejando únicamente la agrupación de empresas y trabajadores por informáticas y no informáticas.

Registros generados

Se guarda el nuevo DataSet en un csv (datos2.*csv*).

Código

https://github.com/Davidguzley/Movilidad-Chile/blob/main/Codigo/Modeling/obtain_data_Chile.ipynb

merge_data_Chile

Formateo

Partimos del DataSet *HomeOffice*, generado en el paso de **Get_%_of_homeoffice**, de los DataSets *conexiones_fijas_internet2*, *Empresas_tamano2*, *Servicios Basicos 2* y *datos2* generados en el paso de **obtain_data_Chile**.

Unimos los 5 DataSets utilizando como parámetro de unión la columna *Unnamed: 0*.

Establecemos la columna *Unnamed: 0* como índice.

Registros generados

Se guarda el nuevo DataSet en un csv (final.*csv*).

Código

https://github.com/Davidguzley/Movilidad-Chile/blob/main/Codigo/Modeling/merge_data_Chile.ipynb