Datos de Contactabilidad

Data Cleaning - Mails

1. Lectura del dataset recibido.
    a. En muchos casos, el proveedor envía los datos divididos en 2 archivos; por lo tanto, hay que leer ambos y unirlos.
2. Se dropean filas y columnas que no tienen ningún valor.
3. Se obtienen datos de mails de las columnas correspondientes.
4. Se analiza la columna “Tel. de contacto e email”, que contiene teléfonos y mails en la misma celda.
    a. Se generan máscaras para identificar qué celdas contienen @.
5. Se analiza la columna Email.
6. Se genera un diccionario por campo del dataset que contenga mails.
    a. Los diccionarios contendrán para cada moroso todos los mails que se encuentren en los campos anteriormente mencionados.
    b. Esto se realiza ya que el proveedor nos otorga los datos en formato multifila y multicolumna; si el moroso tiene más de un mail, puede que los mails adicionales aparezcan en filas o columnas subsiguientes, sin seguir un patrón definido.
7. Se agrupa por DNI.
8. Se unen los diccionarios.
9. Se convierte a dataframe, y se concatena con el dataframe original que se estaba analizando.
10. Lectura del .csv enviado. Siendo más estrictos, no sería el .csv “enviado” sino aquel que se extrae y contiene IDMOROSO, NOMBRE, ESTADO, FILTROA e IDCLIENTE (el .csv enviado sólo tiene IDMOROSO o Documento y NOMBRE). Refiere al .csv de casos que extraemos de Active para enviar a enriquecer.
11. Se crea una columna NRO_DOC, que contendrá los DNIs limpios para hacer la unión con el dataframe recibido.
    a. En caso que se hayan enviado a enriquecer casos de AMEX, se deberán descomentar las líneas correspondientes que indican que si el IDCliente empieza con “AMEX”, el número de documento está ubicado en el campo DATOADIC en lugar de IDMOROSO.
12. DNI Cleaning: se generan procesos de limpieza de los IDMorosos/DATOADIC para extraer el número de DNI y el tipo de Documento (este punto podría llegar a optimizarse redefiniendo ciertas funciones).
13. Se unen ambos dataframes ya limpios (dataset recibido + dataset enviado).
14. Se generan campos adiiconales (tipo, subtipo, propietario del dato).
    a. Para identificar al propietario de cada mail (si es propio o ajeno), se hace uso de distancias de Levenshtein.
15. Se renombran y preparan las columnas para adecuarse a la tabla donde será insertada la data.
16. Se revisan los duplicados que puedan existir.
    a. Por ejemplo, el proveedor nos puede indicar que el mail test@test.com es el mail del moroso, pero también nos indica que es el mail de algún familiar; como no sabemos a quién atribuirle, se generan ciertas reglas.

    b. Tenemos dos opciones:
        i. Los casos son pocos y los podemos dropear manualmente. En este caso, la regla general es que si el dato es otorgado como Propio y como Ajeno, queda como Propio y se dropean las filas que indican que es Ajeno.
        ii. Los casos son muchos y debemos pasar al notebook de Duplicados. Las reglas están definidas dentro de este notebook.
17. Se exporta el archivo, cambiando simplemente la fecha que está al final del nombre.

Data Cleaning - Teléfonos (fijos, celulares, vecinos)

1. Lectura del dataset recibido.
    a. En muchos casos, el proveedor envía los datos divididos en 2 archivos; por lo tanto, hay que leer ambos y unirlos.
2. Se dropean filas y columnas que no tienen ningún valor.
3. Se obtienen datos de mails de las columnas correspondientes.
4. Se analizan las columnas que pueden contener datos de teléfonos:
    a. Teléfonos celulares: celular, celular 1,2,3,4 y 5.
    b. Teléfonos fijos: teléfonos propios, tel. de contacto e email, teléfono línea 1,2,3,4 y 5.
    c. Teléfonos vecinos.
5. Se genera una expresión regular para poder identificar teléfonos en los campos indicados.
6. Se genera un diccionario por cada campo del dataset que contenga teléfonos.
    a. Los diccionarios contendrán para cada moroso todos los teléfonos que se encuentren en los campos anteriormente mencionados, según el caso.
    b. Esto se realiza ya que el proveedor nos otorga los datos en formato multifila y multicolumna; si el moroso tiene más de un teléfono, puede que los teléfonos adicionales aparezcan en filas o columnas subsiguientes, sin seguir un patrón definido.
7. Se agrupa por DNI.
8. Se unen los diccionarios.
9. Se convierte a dataframe, y se concatena con el dataframe original que se estaba analizando.
10. Lectura del .csv enviado. Siendo más estrictos, no sería el .csv “enviado” sino aquel que se extrae y contiene IDMOROSO, NOMBRE, ESTADO, FILTROA e IDCLIENTE (el .csv enviado sólo tiene IDMOROSO o Documento y NOMBRE). Refiere al .csv de casos que extraemos de Active para enviar a enriquecer.
11. Se crea una columna NRO_DOC, que contendrá los DNIs limpios para hacer la unión con el dataframe recibido.
    a. En caso que se hayan enviado a enriquecer casos de AMEX, se deberán descomentar las líneas correspondientes que indican que si el IDCliente empieza con “AMEX”, el número de documento está ubicado en el campo DATOADIC en lugar de IDMOROSO.
12. DNI Cleaning: se generan procesos de limpieza de los IDMorosos/DATOADIC para extraer el número de DNI y el tipo de Documento (este punto podría llegar a optimizarse redefiniendo ciertas funciones).

13. Se unen ambos dataframes ya limpios (dataset recibido + dataset enviado).
14. Se generan campos adiiconales (tipo, subtipo, propietario del dato).
15. Se renombran y preparan las columnas para adecuarse a la tabla donde será insertada la data.
16. Se revisan los duplicados que puedan existir.
    a. Por ejemplo, el proveedor nos puede indicar que el teléfono 1111-1111 es el teléfono del moroso, pero también nos indica que es el número de algún familiar; como no sabemos a quién atribuirle, se generan ciertas reglas.
    b. Tenemos dos opciones:
        i. Los casos son pocos y los podemos dropear manualmente. En este caso, la regla general es que si el dato es otorgado como Propio y como Ajeno, queda como Propio y se dropean las filas que indican que es Ajeno.
        ii. Los casos son muchos y debemos pasar al notebook de Duplicados. Las reglas están definidas dentro de este notebook.
17. Se exporta el archivo, cambiando simplemente la fecha que está al final del nombre.

DAM_LOG

1. Lectura del .csv enviado. Siendo más estrictos, no sería el .csv “enviado” sino aquel que se extrae y contiene IDMOROSO, NOMBRE, ESTADO, FILTROA e IDCLIENTE (el .csv enviado sólo tiene IDMOROSO o Documento y NOMBRE). Refiere al .csv de casos que extraemos de Active para enviar a enriquecer.
    a. En el caso que el envío incluya datos de AMEX, descomentar la cuarta línea que indica que nos quedamos además con el campo DATOADIC.
2. Se agregan las fechas de envío e inserción (fecha de cambio de estados es algo que ya no se utiliza, por lo que siempre se agrega un equivalente a la fecha de inserción).
3. Se engloban todos los estados en las categorías correspondientes (Referencia: ver documento de Glosario de Estados).
4. Lectura del .csv recibido por parte del proveedor. La idea es cruzar contra los IDs y ver cuáles están enriquecidos.
5. Reducimos el .csv recibido a CUIL y DNI, para hacer el join.
    a. En caso que se hayan enviado a enriquecer casos de AMEX, se deberán descomentar las líneas correspondientes que indican que si el IDCliente empieza con “AMEX”, el número de documento está ubicado en el campo DATOADIC en lugar de IDMOROSO.
6. DNI Cleaning: se generan procesos de limpieza de los IDMorosos/DATOADIC para extraer el número de DNI y el tipo de Documento (este punto podría llegar a optimizarse redefiniendo ciertas funciones).
7. Se unen ambos dataframes ya limpios (dataset recibido + dataset enviado).
8. Se seleccionan las columnas del dataframe (aquellas booleanas que generamos y las que queremos puntuar).
9. Se asigna un booleano a los casos que devuelve la unión, ya que son los que se enriquecieron.
10. Se exporta la data en formato .csv, cambiando la fecha al final del nombre.