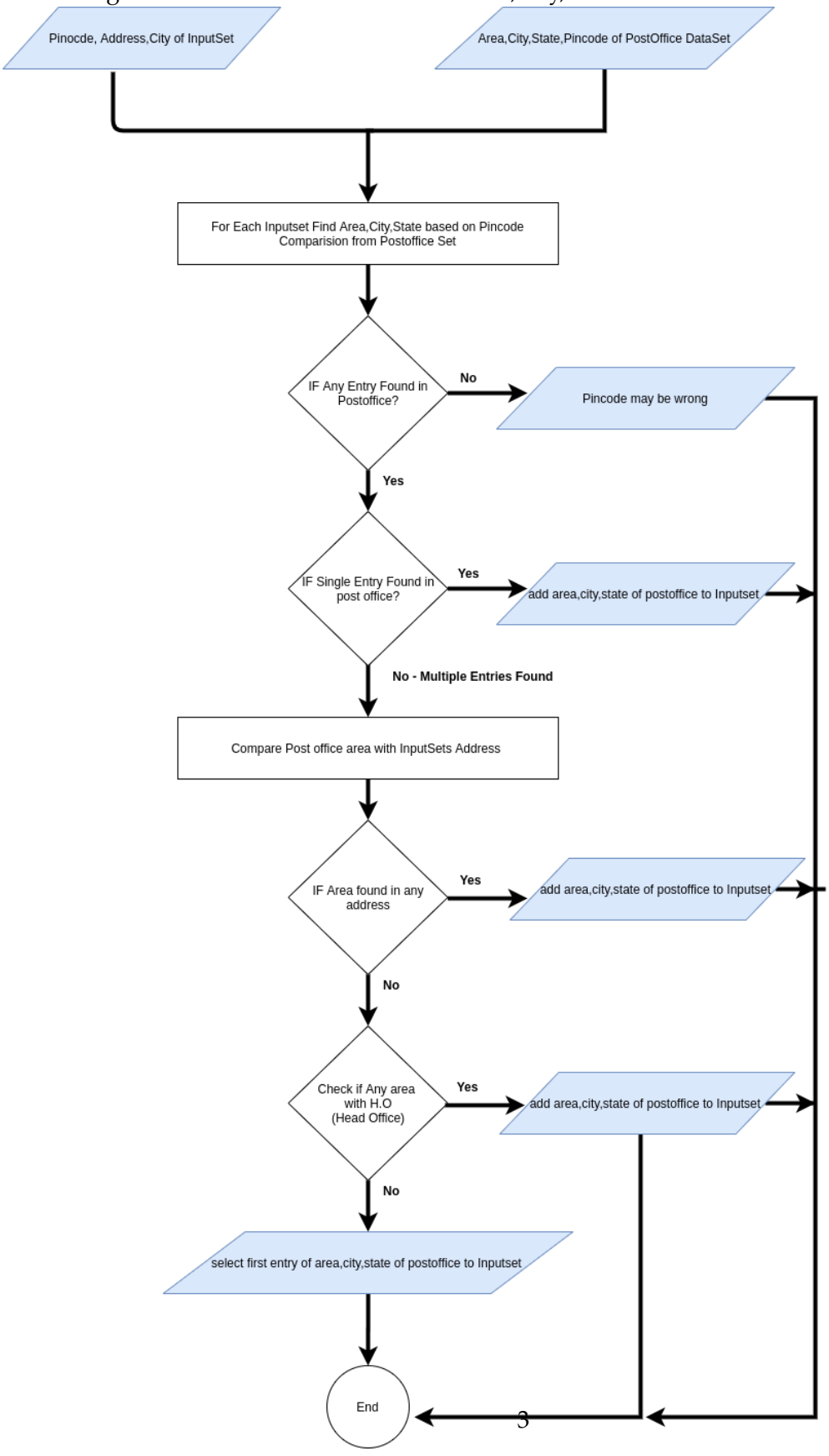
find area_city_state_basedon_postoffice_with_pinocde

June 3, 2020

1 Find Area City State Based on Pincode

This algo is used to find Area,City,State based on available pincode



[3]:
```
#########
# Import Libraries
#########
import pymysql
import re
import pandas as pd
import numpy as np
import pandasql as ps
import numpy as np
from sqlalchemy import create_engine
import pymysql

from fuzzywuzzy import fuzz
from fuzzywuzzy import process

## import warnings
warnings.filterwarnings("ignore", 'This pattern has match groups')
```

1.0.1 Inputset with Pincode

InputSet which contains pincode, based on pincode we need to find area,city,state from the postoffice set

[19]:
```
my_df = pd.read_csv("DATA/Data for area finding.csv",dtype=str,low_memory=False)
## Change the column Names
query1 = """
          SELECT Id as Id,(REPLACE(lower(coalesce(Address2,'')),"\\n",'') ||␣
 ↪REPLACE(lower(coalesce(Address2,'')),"\\n",'') ||␣
 ↪REPLACE(lower(coalesce(Area,'')),"\\n",'') ||␣
 ↪REPLACE(lower(coalesce(City,'')),"\\n",'') )as␣
 ↪Address,REPLACE(lower(Pincode),"\\n",'') as Pincode FROM my_df
         """
my_df = ps.sqldf(query1, locals())
my_df.head()
```

[19]:

	Id	Address	Pincode
0	8000	shop no. 78 first floorshop no. 78 first floor...	495001
1	331	udhampurudhampurudhampur	None
2	4090	thanethanethane	401107
3	4091	ganesh nagar, udaipurganesh nagar, udaipurudaipur	313001
4	4092	junnar, punejunnar, punepune	410502

1.0.2 Postoffice set with Pincode,area,city,state

Postoffice set is like master set which contains all the pincodes area,city,state info.

[20]:
```
df_postoffice = pd.read_csv("DATA/MasterFiles/postofficedata.
 ↪csv",low_memory=False)
df_postoffice['area'] = df_postoffice['area'].str.lower()
```

```
df_postoffice.head()
```

[20]:
```
   id                officename                   area pincode office_type  \
0   1             Achalapur B.O              achalapur  504273         B.O
1   2                  Ada B.O                    ada  504293         B.O
2   3               Adegaon B.O                 adegaon  504307         B.O
3   4  Adilabad Collectorate S.O  adilabad collectorate  504001         S.O
4   5              Adilabad H.O                adilabad  504001         H.O

  deliverystatus divisionname regionname      circlename     taluk  \
0       Delivery     Adilabad  Hyderabad  Andhra Pradesh  Asifabad
1       Delivery     Adilabad  Hyderabad  Andhra Pradesh  Asifabad
2       Delivery     Adilabad  Hyderabad  Andhra Pradesh     Boath
3   Non-Delivery     Adilabad  Hyderabad  Andhra Pradesh  Adilabad
4       Delivery     Adilabad  Hyderabad  Andhra Pradesh  Adilabad

  districtname  statename     telephone related_suboffice related_headoffice  \
0     Adilabad  TELANGANA           NaN       Rechini S.O     Mancherial H.O
1     Adilabad  TELANGANA           NaN      Asifabad S.O     Mancherial H.O
2     Adilabad  TELANGANA           NaN        Echoda S.O       Adilabad H.O
3     Adilabad  TELANGANA  08732-226703               NaN       Adilabad H.O
4     Adilabad  TELANGANA  08732-226738               NaN               NaN

   longitude  latitude
0        NaN       NaN
1        NaN       NaN
2        NaN       NaN
3        NaN       NaN
4        NaN       NaN
```

[22]:
```
lst_df = pd.DataFrame(columns=['MasterId','Area','City','State'])
cnt = 1
for i,r in my_df.iterrows():
    print(cnt)
    cnt= cnt+1
    masterid = str(r["Id"])
    pincode = str(r["Pincode"])
    address = (re.sub('[^a-zA-Z ]','',str(r['Address']))).replace(" ", "")
    query = """
            SELECT area,office_type,regionname as city,statename FROM␣
 ↪df_postoffice where pincode = '"""+(pincode)+"""'
         """
    my_df_main = ps.sqldf(query, locals())
    if(len(my_df_main)==1):
        # if only 1 entry found for particular pinocde then use that
        tmp_df = my_df_main
    else:
```

```
        query = "SELECT area,office_type,city,statename FROM my_df_main where␣
↪'"+address+"' like '%'||area||'%'"
        tmp_df = ps.sqldf(query, locals())
        if len(tmp_df)==0:
            tmp_df = (my_df_main[(my_df_main.apply(lambda row: fuzz.
↪partial_ratio(row['area'], address), axis=1) > 92)])
        if len(tmp_df)==0:
            if len(my_df_main)>1:
                tmp_df = my_df_main[my_df_main['office_type'] == 'H.O']
                if len(tmp_df)==0:
                    tmp_df = (my_df_main[(my_df_main.apply(lambda row: fuzz.
↪partial_ratio(row['area'], address), axis=1) > 60)])
                    tmp_df = tmp_df[:1]
                    if len(tmp_df)==0:
                        tmp_df = my_df_main[:1]


    if(len(tmp_df)>0):
        for index, row in tmp_df[:1].iterrows():
            lst_df = lst_df.append({'MasterId':str(masterid),'Area':
↪str(row['area']),'City':str(row['city']),'State':str(row['statename'])},␣
↪ignore_index=True)
    else:
        lst_df = lst_df.append({'MasterId':str(masterid),'Area':"",'City':
↪"","State":""}, ignore_index=True)

#print(lst_df)
lst_df.to_csv('/home/juned/PythonWork/Mapping/DATA/5.8 L retailer list␣
↪(Cleaned)-AreaCityState-560000-565000.csv', header=True, index=False)
```