MSPA PREDICT 420

Graded Exercise 1: Flying Around

Introduction

This document presents the results of first graded exercise for the Masters of Science in Predictive Analytics course: PREDICT 420. This assessment required the student to perform some data w rangling exercises on airline/flight data retrieved from OpenFlights.org.

Assessment

1. Loading the Data

Load datasets into pandas dataframes.

In [1]:
```
import pandas as pd

df_airlines = pd.read_csv("data/airlines.dat", header = None, encoding = "latin-1")
df_airports = pd.read_csv("data/airports.dat", header = None, encoding = "latin-1")
df_routes = pd.read_csv("data/routes.dat", header = None, encoding = "latin-1")
```

2. Pre-process the Data

Import field names for each dataframe as lists from relevant text files.

In [2]:
```
import csv

f = open("data/airlines_fields.txt", "r")
reader = csv.reader(f)
airlines_fields = list(reader)

f = open("data/airports_fields.txt", "r")
reader = csv.reader(f)
airports_fields = list(reader)

f = open("data/routes_fields.txt", "r")
reader = csv.reader(f)
routes_fields = list(reader)
```

Show each of the imported field name lists.

In [3]:
```
airlines_fields
```

Out[3]:
```
[['airlID',
  'airlName',
  'airlAlias',
  'airlIATA',
  'airlICAO',
  'airlCallsign',
  'airlCountry',
  'airlActive']]
```

In [4]:
```
airports_fields
```

Out[4]:
```
[['airpID',
  'airpName',
  'airpCity',
  'airpCountry',
  'airpIATAFAA',
  'airpICAO',
  'airpLat',
  'airpLong',
  'airpAlt',
  'airpTimezone',
  'airpDST',
  'airpTz']]
```

In [5]:
```
routes_fields
```

Out[5]:
```
[['airlName',
  'airlID',
  'sourceAirpName',
  'sourceAirpID',
  'destAirpName',
  'destAirpID',
  'airlCodeshare',
  'airlStops',
  'airlEquip']]
```

Use the imported field name lists to set column names for each dataframe.

In [6]:
```
df_airlines.columns = airlines_fields
df_airports.columns = airports_fields
df_routes.columns = routes_fields
```

Show first three rows of each dataframe.

In [7]:
```
df_airlines.head(3)
```

Out[7]:

	airlID	airlName	airlAlias	airlIATA	airlICAO	airlCallsign	airlCountry	airlActive
0	1	Private flight	\N	-	NaN	NaN	NaN	Y
1	2	135 Airways	\N	NaN	GNL	GENERAL	United States	N
2	3	1Time Airline	\N	1T	RNX	NEXTIME	South Africa	Y

In [8]:
```
df_airports.head(3)
```

Out[8]:

	airpID	airpName	airpCity	airpCountry	airpIATAFAA	airpICAO	airpLat	airpLong	airpAlt	airpTimezone	airpDST	airpT
0	1	Goroka	Goroka	Papua New Guinea	GKA	AYGA	-6.081689	145.391881	5282	10	U	Pacific
1	2	Madang	Madang	Papua New Guinea	MAG	AYMD	-5.207083	145.788700	20	10	U	Pacific
2	3	Mount Hagen	Mount Hagen	Papua New Guinea	HGU	AYMH	-5.826789	144.295861	5388	10	U	Pacific

In [9]:
```
df_routes.head(3)
```

Out[9]:

	airlName	airlID	sourceAirpName	sourceAirpID	destAirpName	destAirpID	airlCodeshare	airlStops	airlEquip
0	2B	410	AER	2965	KZN	2990	NaN	0	CR2
1	2B	410	ASF	2966	KZN	2990	NaN	0	CR2
2	2B	410	ASF	2966	MRV	2962	NaN	0	CR2

Pickle final dataframes, then re-read.

In [10]:
```
import pickle

df_airlines.to_pickle('data/airlines.p')
df_airports.to_pickle('data/airports.p')
df_routes.to_pickle('data/routes.p')

df_airlines = pd.read_pickle("data/airlines.p")
df_airports = pd.read_pickle("data/airports.p")
df_routes = pd.read_pickle("data/routes.p")
```

3. What is three letter airport code for the airport that is closest to your home?

Import 'distance_on_unit_sphere' function to calculate distance between two latitude/longitude pairs.

In [11]:
```
#Source: http://www.johndcook.com/blog/python_longitude_latitude/
import math

def distance_on_unit_sphere(lat1, long1, lat2, long2):
    degrees_to_radians = math.pi/180.0
    phi1 = (90.0 - lat1)*degrees_to_radians
    phi2 = (90.0 - lat2)*degrees_to_radians
    theta1 = long1*degrees_to_radians
    theta2 = long2*degrees_to_radians

    cos = (math.sin(phi1)*math.sin(phi2)*math.cos(theta1 - theta2) +
           math.cos(phi1)*math.cos(phi2))
    arc = math.acos( cos )

    return arc*6373
```

Apply 'distance_on_unit_sphere' function to each airport within the df_airports dataframe, in order to find the distance between each airport and my hometown 'Rivervale, Western Australia' (latitude -31.9610 and longitude 115.9170).

In [12]:
```
import numpy as np

rivervale_lat = -31.9610
rivervale_long = 115.9170

df_airports["distToRivervale"] = np.vectorize(distance_on_unit_sphere)(df_airports["airpLat"],
                                                                       df_airports["airpLong"],

                                                                       rivervale_lat,
                                                                       rivervale_long)

min_rivervale_row_index = df_airports["distToRivervale"].idxmin()
```

Return the airport which minimizes the distance to my hometown.

In [13]:
```
min_rivervale_airport_name = df_airports["airpName"].iloc[min_rivervale_row_index]
min_rivervale_airport_name
```

Out[13]:
```
'Perth Intl'
```

Return the airport code associated with 'Perth Intl'.

In [14]:
```
min_rivervale_airport_code = df_airports["airpIATAFAA"].iloc[min_rivervale_row_index]
print("Three letter airport code for the airport closest to my home:", min_rivervale_airport_code)
```

```
Three letter airport code for the airport closest to my home: PER
```

4. How many departing routes are there from this airport?

Create a new dataframe based on matches of original 'df_routes' dataframe where 'Source_airport' is equal to 'PER'.

In [15]:
```
min_rivervale_airport_routes = df_routes[df_routes.sourceAirpName == min_rivervale_airport_code]
```

Count rows of matched dataframe.

In [16]:
```
min_rivervale_routes_count = len(min_rivervale_airport_routes.index)
print("Number of departing routes from PER:", min_rivervale_routes_count)
```

```
Number of departing routes from PER: 92
```

5. How many routes are there coming into the airport with the three letter code "EGO?"

Create a new dataframe based on matches of original 'df_routes' dataframe where 'Source_airport' is equal to 'EGO'.

In [17]:
```
ego_airport_routes = df_routes[df_routes.destAirpName == "EGO"]
```

Count rows of matched dataframe.

In [18]:
```
ego_airport_routes_count = len(ego_airport_routes.index)
print("Number of arriving routes at EGO:", ego_airport_routes_count)
```

```
Number of arriving routes at EGO: 11
```