proj2

April 2, 2021

1 Machine Learning in Python - Project 1

Due Friday, April 9th by 5 pm UK local time.

include contributors names here

1.1 0. Setup

[1]:
```
# Install required packages
!pip install -q -r requirements.txt
```

[2]:
```
# Add any additional libraries or submodules below

# Display plots inline
%matplotlib inline

# Data libraries
import pandas as pd
import numpy as np
import geopy.distance as gpy
import plotly.express as px
from datetime import datetime

#Web Scraping Requirement
import datapackage

# Plotting libraries
import matplotlib.pyplot as plt
import seaborn as sns

# Plotting defaults
plt.rcParams['figure.figsize'] = (8,5)
plt.rcParams['figure.dpi'] = 80

# sklearn modules
import sklearn
```

```
[3]: # Load data
d = pd.read_csv("hotel.csv")
cur_code = pd.read_csv("curr_codes.csv")
countries_name = pd.read_csv("ISO 3155.csv")
coords = pd.read_csv("countries_coords.csv")
fx_rates = pd.read_csv("/work/currency_exchange_rates_02-01-1995_-_02-05-2018.
 ↪csv")
```

```
[ ]: data_url = 'https://datahub.io/core/country-codes/datapackage.json'
# to load Data Package into storage
package = datapackage.Package(data_url)
# to load only tabular data
resources = package.resources
for resource in resources:
    if resource.tabular:
        comp_countries = pd.read_csv(resource.descriptor['path'])
```

Dataframe	description
d	Hotel.csv - as provided in question
cur_code	Country Names & associated Currencyfor that country
countries_name	ISO 3155 Country Names & 3 character currency code to join with hotel.csv
coords	central coordinates of each country & ISO 3155 Country code
fx_rates	Daily Global Currecny Exchange rates from 1995 - 2018 (in USD)
comp_countries	Comprehensive country code information, including ISO 3166 codes, ITU dialing codes, ISO 4217 currency codes,

```
[ ]: #d['country'].unique()
#comp_countries['ISO3166-1-Alpha-2']
```

1.2 1. Introduction

This section should include a brief introduction to the task and the data (assume this is a report you are delivering to a client). If you use any additional data sources, you should introduce them *here and discuss why they were included.*

Briefly outline the approaches being used and the conclusions that you are able to draw.

1.3 2. Exploratory Data Analysis and Feature Engineering

Include a detailed discussion of the data with a particular emphasis on the features of the data that are relevant for the subsequent modeling. Including visualizations of the data is strongly encouraged - all code and plots must also be described in the write up. Think carefully about whether each plot needs to be included in your final draft - your report should include figures but they should be as focused and impactful as possible.

Additionally, this section should also implement and describe any preprocessing / feature engineering of the data. Specifically, this should be any code that you use to generate new columns in the data frame `d`. All of this processing is explicitly meant to occur before we split the data in to training and testing subsets. Processing that will be performed as part of an sklearn pipeline can be mentioned here but should be implemented in the following section.

All code and figures should be accompanied by text that provides an overview / context to what is being done or presented.

From data collected by Antonio, Almeida and Nunes, 2019, we found that "Data source location Both hotels are located in Portugal: H1 at the resort region of Algarve and H2 at the city of Lisbon". This can be used with the ISO code and compare to the location of the guests, and how far they are travelling roughly using average co-ordinates.

'ISO 3155.csv' - Gives ISO 3155 ISO codes and names - https://en.wikipedia.org/wiki/ISO_3166-1_alpha-3

'countries_coords.csv' - Gives location of Countries with ISO 2 letter code ISO 3155. https://developers.google.com/public-data/docs/canonical/countries_csv

Comprehensive country codes: ISO 3166, ITU, ISO 4217 currency codes and many more: https://datahub.io/core/country-codes

1.3.1 Plan for alternative Data:

Currency exchange rated

- expand date into day/month/years.
- delete data out of date range - Minus 3 months of the initial hotel booking date range (so we can keep in for 3 month average)
- across the row (axis = 0) divide all by the value of Euros to give exchange rate with respect to Euros (as opposed to USD, which data is currently in) -Take a 3 month/90 day average of the exchange rate in that currency
- Pivot Data so that the columns are: Day, Month, Year, Exchange rate, Currency code

use curr_codes-all_csv.csv to join Currency to Country

- Join curr_codes-all_csv.csv to Currencies table, giving a "Country code" column
- Now join Exchange rate & 90 day average from exchange rate data set to the main "d" dataset via:

– d.day = curr.day *d.month = curr.month* d.year = curr.year *d.country code= curr.country code

Geographical location I have some prebuilt code for this which will hopefully work Create a new column that calculates the distance of each country from Portugal (where the hotel is). The have a Column with "Distance from"

1.3.2 Formatting Arrival & Booking Date

Formatting to allow for joining of external data sources

[ ]:
```
#format arrival date
d['month'] = pd.to_datetime(d.arrival_date_month, format='%B').dt.month
d['day'] = pd.to_datetime(d.arrival_date_day_of_month, format='%d').dt.day
d['Year'] = pd.to_datetime(d.arrival_date_year, format='%Y').dt.year
d['arrival_date']  = pd.to_datetime(d[['Year','month','day']], format =␣
 ↪'%Y%m%d').dt.date
d['booking_date'] = d['arrival_date'] - pd.to_timedelta(d['lead_time'],␣
 ↪unit='d')
min_date = d['arrival_date'].min()
max_date = d['arrival_date'].max()
min_booking_date = d['booking_date'].min()
```

```
/usr/local/lib/python3.7/site-packages/pandas/core/arrays/datetimelike.py:1111:
PerformanceWarning: Adding/subtracting object-dtype array to TimedeltaArray not
vectorized
  PerformanceWarning,
```

[ ]:
```
#Reformat Date column
fx_rates['Date']  = pd.to_datetime(fx_rates['Date']).dt.date

#Cut Dates so that theres only the date from the earliest booking to the last␣
 ↪booking
fx_rates = fx_rates[fx_rates['Date'].between(min_booking_date,max_date)]

#Divide all by Euros - All FX Rates are: "Currency" per Euro
fx_rates.iloc[:,1:] = fx_rates.iloc[:,1:].div(fx_rates.Euro, axis=0)
```

[ ]:
```
#fx_rates.iloc[:,1:]
# Get all names
#for col_name in fx_rates.iloc[:,1:]:
#    np.array["90_ave_"+col_name]



#fx_rates.iloc[:,1:].div(fx_rates.Euro, axis=0)
print(fx_rates.iloc[:,1:].rolling(window=50).mean())
fx_rates.head()
```

```
      Algerian Dinar  Australian Dollar  Bahrain Dinar  Bolivar Fuerte  \
4764             NaN                NaN            NaN             NaN
4765             NaN                NaN            NaN             NaN
4766             NaN                NaN            NaN             NaN
4767             NaN                NaN            NaN             NaN
4768             NaN                NaN            NaN             NaN
…                  …                  …              …               …
5807             NaN                NaN            NaN             NaN
5808             NaN                NaN            NaN             NaN
5809             NaN                NaN            NaN             NaN
5810             NaN                NaN            NaN             NaN
5811             NaN                NaN            NaN             NaN

      Botswana Pula  Brazilian Real  Brunei Dollar  Canadian Dollar  \
4764            NaN             NaN            NaN              NaN
4765            NaN             NaN            NaN              NaN
4766            NaN             NaN            NaN              NaN
4767            NaN             NaN            NaN              NaN
4768            NaN             NaN            NaN              NaN
…                 …               …              …                …
5807            NaN             NaN            NaN              NaN
5808            NaN             NaN            NaN              NaN
5809            NaN             NaN            NaN              NaN
5810            NaN             NaN            NaN              NaN
5811            NaN             NaN            NaN              NaN

      Chilean Peso  Chinese Yuan  …  South African Rand  Sri Lanka Rupee  \
4764           NaN           NaN  …                 NaN              NaN
4765           NaN           NaN  …                 NaN              NaN
4766           NaN           NaN  …                 NaN              NaN
4767           NaN           NaN  …                 NaN              NaN
4768           NaN           NaN  …                 NaN              NaN
…                …             …  …                   …                …
5807           NaN      5.842053  …                 NaN              NaN
5808           NaN      5.831427  …                 NaN              NaN
5809           NaN      5.819257  …                 NaN              NaN
5810           NaN      5.807250  …                 NaN              NaN
5811           NaN      5.796285  …                 NaN              NaN

      Swedish Krona  Swiss Franc  Thai Baht  Trinidad And Tobago Dollar  \
4764            NaN          NaN        NaN                         NaN
4765            NaN          NaN        NaN                         NaN
4766            NaN          NaN        NaN                         NaN
4767            NaN          NaN        NaN                         NaN
4768            NaN          NaN        NaN                         NaN
…                 …            …          …                           …
5807            NaN          NaN        NaN                         NaN
5808            NaN          NaN        NaN                         NaN
```

```
5809            NaN            NaN        NaN                           NaN
5810            NaN            NaN        NaN                           NaN
5811            NaN            NaN        NaN                           NaN

      Tunisian Dinar  U.A.E. Dirham  U.K. Pound Sterling  U.S. Dollar
4764             NaN            NaN                  NaN          NaN
4765             NaN            NaN                  NaN          NaN
4766             NaN            NaN                  NaN          NaN
4767             NaN            NaN                  NaN          NaN
4768             NaN            NaN                  NaN          NaN
…                …            …                  …          …
5807             NaN            NaN             1.119664     0.865540
5808             NaN            NaN             1.118416     0.864402
5809             NaN            NaN             1.117049     0.863143
5810             NaN            NaN             1.116028     0.862000
5811             NaN            NaN             1.115052     0.860971

[1048 rows x 51 columns]
```

[ ]:
```
            Date  Algerian Dinar  Australian Dollar  Bahrain Dinar  \
4764  2013-06-24       60.358551           0.703194       0.287330
4765  2013-06-25       60.255444           0.704431       0.286280
4766  2013-06-26       61.209690           0.711686       0.288698
4767  2013-06-27       61.180786           0.715009       0.288521
4768  2013-06-28       61.179817           0.709098       0.287462

      Bolivar Fuerte  Botswana Pula  Brazilian Real  Brunei Dollar  \
4764        4.802231       0.088644        1.730705       0.975470
4765        4.784681       0.088473        1.714253       0.971372
4766        4.825092       0.088990        1.703394       0.976351
4767        4.822130       0.089319        1.686234       0.973297
4768        4.804434       0.088838        1.670183       0.967278

      Canadian Dollar  Chilean Peso  …  South African Rand  Sri Lanka Rupee  \
4764         0.804830    393.076570  …            7.762876        98.441464
4765         0.800442    390.893863  …            7.610020        98.086950
4766         0.803824    390.479115  …            7.746583        98.941186
4767         0.804174    387.523020  …            7.657228        99.207950
4768              NaN    385.214067  …            7.641628        99.358257

      Swedish Krona  Swiss Franc  Thai Baht  Trinidad And Tobago Dollar  \
4764       5.123873     0.714351  23.778083                    4.904937
4765       5.125019     0.711436  23.599056                    4.879473
4766       5.145424     0.722666  23.853655                    4.933200
4767       5.171578     0.725675  23.869705                    4.926412
4768       5.132722     0.722324  23.805046                    4.911544
```

```
      Tunisian Dinar  U.A.E. Dirham  U.K. Pound Sterling  U.S. Dollar
4764        1.249809       2.806434             1.173315     0.764175
4765        1.252551       2.796178             1.176184     0.761383
4766        1.261594       2.819794             1.179208     0.767813
4767        1.264426       2.818063             1.172805     0.767342
4768        1.266514       2.807722             1.164450     0.764526

[5 rows x 52 columns]
```

Geographical location The distance in kilometers was calculated for all countries, based on their central location, to base the distance of each traveller. this will also act as act as a proxy for the cost of travel to the location.

[ ]:
```
#Reform Co-Ordinates into list within DF
coords['co_ords'] =  coords[['latitude', 'longitude']].values.tolist()
coords = coords.dropna()

# Set Portugal as basis
portugal = coords['co_ords'].loc[coords["name"] == 'Portugal'].values.tolist()

#Compute the distance in KM from all countries to Portugal
coords['distance(km)'] = coords.apply(lambda coords: gpy.great_circle(portugal,
                                                            coords['co_ords']).
 ↪km,
                                                           axis = 1).
 ↪round(decimals=2)


coords
```

[ ]:
```
    country   latitude  longitude                  name  \
0        AD  42.546245   1.601554               Andorra
1        AE  23.424076  53.847818  United Arab Emirates
2        AF  33.939110  67.709953           Afghanistan
3        AG  17.060816 -61.796428   Antigua and Barbuda
4        AI  18.220554 -63.068615              Anguilla
..      …          …          …                   …
240      YE  15.552727  48.516388                 Yemen
241      YT -12.827500  45.166244               Mayotte
242      ZA -30.559482  22.937506          South Africa
243      ZM -13.133897  27.849332                Zambia
244      ZW -19.015438  29.154857              Zimbabwe

                     co_ords  distance(km)
0      [42.546245, 1.601554]        895.35
1     [23.424076, 53.847818]       6031.03
2      [33.93911, 67.709953]       6596.55
```

```
3    [17.060816, -61.796428]       5707.20
4    [18.220554, -63.068615]       5738.08
..                         …          …
240   [15.552727, 48.516388]       6077.68
241     [-12.8275, 45.166244]      8010.16
242  [-30.559482, 22.937506]       8419.60
243  [-13.133897, 27.849332]       6933.48
244  [-19.015438, 29.154857]       7567.33

[243 rows x 6 columns]
```

1.4 3. Model Fitting and Tuning

[ ]:
```

```

In this section you should detail your choice of model and describe the process used to refine and fit that model. You are strongly encouraged to explore many different modeling methods (e.g. logistic regression, classification trees, SVC, etc.) but you should not include a detailed narrative of all of these attempts. At most this section should mention the methods explored and why they were rejected - most of your effort should go into describing the model you are using and your process for tuning and validatin it.

This section should also include the full implementation of your final model, including all necessary validation. As with figures, any included code must also be addressed in the text of the document.

1.5 4. Discussion & Conclusions

In this section you should provide a general overview of your final model, its performance, and reliability. You should discuss what the implications of your model are in terms of the included features, predictive performance, and anything else you think is relevant.

This should be written with a target audience of the client who is with the hotel data and university level mathematics but not necessarily someone who has taken a postgraduate statistical modeling course. Your goal should be to convince this audience that your model is both accurate and useful.

Keep in mind that a negative result, i.e. a model that does not work well predictively, that is well explained and justified in terms of why it failed will likely receive higher marks than a model with strong predictive performance but with poor or incorrect explinations / justifications.

1.6 5. Convert Document

[ ]:
```
# Run the following to render to PDF
!jupyter nbconvert --to pdf proj2.ipynb
```

```
[NbConvertApp] Converting notebook proj2.ipynb to pdf
[NbConvertApp] Writing 46696 bytes to notebook.tex
[NbConvertApp] Building PDF
[NbConvertApp] Running xelatex 3 times: ['xelatex', 'notebook.tex', '-quiet']
[NbConvertApp] Running bibtex 1 time: ['bibtex', 'notebook']
```

```
[NbConvertApp] WARNING | bibtex had problems, most likely because there were no
citations
[NbConvertApp] PDF successfully created
[NbConvertApp] Writing 69652 bytes to proj2.pdf
```

[ ]:

```

```

Created in Deepnote