

# Revolut\_Challenge\_Working\_Document

April 18, 2020

## 1 Revolut Home Task

This notebook contains data formatting in Python and embedded Tableau generated charts. It contains analysis of Revolut KYC data executed by Veritas API

```
In [75]: import numpy as np
import pandas as pnd
import scipy as sp
import matplotlib.pyplot as plt #data visualisation
from matplotlib.pyplot import pie, axis, show
import seaborn as sb
import json
import ast
import sys
```

### 1.0.1 Load and Format Data

```
In [52]: docrep=pnd.read_csv("C:\Revolut_Challenge_Prakhyath\doc_reports_sample.csv")
In [5]: facerep=pnd.read_csv("C:\Revolut_Challenge_Prakhyath\\face_reports_sample.csv")
In [6]: docrep.head()
```

```
Out[6]:
```

	Unnamed: 0	user_id	result	\
0	27241	8190909e566647a5b6afeee9b4ec6c6a	clear	
1	28369	6b62136dfde348a99855e350294aaf5d	clear	
2	27988	73679363dccc46fa9f34a4fef0d76e3	clear	
3	27529	07857065dfa64db386739ec4fff47856	consider	
4	47987	9f887805b2af49069349ff107e0bca01	clear	

  

	visual_authenticity_result	image_integrity_result	face_detection_result	\
0	clear	clear	clear	
1	clear	clear	clear	
2	clear	clear	clear	
3	clear	clear	clear	
4	clear	clear	clear	

  

	image_quality_result	created_at	supported_document_result	\
--	----------------------	------------	---------------------------	---

0	clear	2017-05-25 08:38:56	clear
1	clear	2017-05-31 08:12:51	clear
2	clear	2017-05-29 15:07:04	clear
3	clear	2017-05-26 19:00:35	clear
4	clear	2017-05-29 14:38:21	clear

	conclusive_document_quality_result	colour_picture_result	\
0	NaN	NaN	
1	NaN	NaN	
2	NaN	NaN	
3	NaN	NaN	
4	NaN	NaN	

	data_validation_result	data_consistency_result	data_comparison_result	\
0	clear	NaN	clear	
1	clear	NaN	clear	
2	clear	clear	clear	
3	clear	clear	consider	
4	clear	clear	clear	

	attempt_id	police_record_result	\
0	30e11e95e30748f485a2271ca5e6abb8	clear	
1	4c0bfde8eb2249ed820e1f61d3ec3e33	clear	
2	c5004fd1fc1d4e36a11433b70d960867	clear	
3	e4b26d4ddda545c9931a0a845cd65109	clear	
4	8ead2b23ef664e4d85fae798a7d5d52c	clear	

	compromised_document_result	\
0	NaN	
1	NaN	
2	NaN	
3	NaN	
4	NaN	

	properties	sub_result	
0	{'gender': 'Female', 'document_type': 'driving...	clear	
1	{'gender': 'Male', 'document_type': 'driving_l...	clear	
2	{'gender': 'Male', 'nationality': 'GBR', 'docu...	clear	
3	{'gender': 'Male', 'nationality': 'PER', 'docu...	caution	
4	{'gender': 'Male', 'issuing_date': '2011-03', ...	clear	

In [8]: facerep.head()

Out[8]:	Unnamed: 0	user_id	result	face_comparison_result	\
	0	58	ecee468d4a124a8eafeec61271cd0da1	clear	clear
	1	76	1895d2b1782740bb8503b9bf3edf1ead	clear	clear
	2	217	e71b27ea145249878b10f5b3f1fb4317	clear	clear
	3	221	f512dc74bd1b4c109d9bd2981518a9f8	clear	clear

```

4          251  0685c7945d1349b7a954e1a0869bae4b  clear                                clear

          created_at facial_image_integrity_result \
0  2017-06-20 17:50:43                                clear
1  2017-06-20 13:28:00                                clear
2  2017-06-18 21:18:31                                clear
3  2017-06-18 22:17:29                                clear
4  2017-06-18 19:54:21                                clear

          visual_authenticity_result properties                                attempt_id
0          clear                                {}  9e4277fc1ddf4a059da3dd2db35f6c76
1          clear                                {}  ab259d3cb33b4711b0a5174e4de1d72c
2          clear                                {}  2b7f1c6f3fc5416286d9f1c97b15e8f9
3          clear                                {}  ab5989375b514968b2ff2b21095ed1ef
4          clear                                {}  dd1b0b2dbe234f4cb747cc054de2fdd3

```

```
In [9]: propdf=docrep[['user_id','attempt_id','properties']]
```

```
In [10]: propdfnew = pnd.DataFrame()
for index, row in propdf.iterrows():
    #print(type(row['properties']))
    dict=ast.literal_eval(row['properties'])
    #print(type(dict))
    #print(dict)
    for key in dict:
        # type(dict[key])
        propdfnew.at[index,'attempt_id']=row['attempt_id']
        propdfnew.at[index,'user_id']=row['user_id']
        propdfnew.at[index,key]=dict[key]
```

```
In [15]: docrep = pnd.merge(docrep, propdfnew, on='attempt_id')
```

```
In [14]: propdfnew.head()
```

```

Out[14]:
          attempt_id                                user_id  gender \
0  30e11e95e30748f485a2271ca5e6abb8  8190909e566647a5b6afeee9b4ec6c6a  Female
1  4c0bfde8eb2249ed820e1f61d3ec3e33  6b62136dfde348a99855e350294aaf5d    Male
2  c5004fd1fc1d4e36a11433b70d960867  73679363dccc46fa9f34a4fef0d76e3    Male
3  e4b26d4ddda545c9931a0a845cd65109  07857065dfa64db386739ec4fff47856    Male
4  8ead2b23ef664e4d85fae798a7d5d52c  9f887805b2af49069349ff107e0bca01    Male

          document_type  date_of_expiry  issuing_country  nationality \
0  driving_licence    2023-12-05          GBR          NaN
1  driving_licence    2020-05-13          GBR          NaN
2  passport          2023-11-15          GBR          GBR
3  passport          2021-01-27          PER          PER
4  national_identity_card          NaN          FRA          NaN

```

	issuing_date	issuing_state	document_version
0	NaN	NaN	NaN
1	NaN	NaN	NaN
2	NaN	NaN	NaN
3	NaN	NaN	NaN
4	2011-03	NaN	NaN

```
In [16]: docrep.shape
```

```
Out[16]: (4991, 28)
```

```
In [22]: for index, row in propdf.iterrows():
          print(index)
          print(propdf.loc[index, 'attempt_id'])
          break;
```

```
0
30e11e95e30748f485a2271ca5e6abb8
```

```
In [36]: docrep.to_csv(r'C:\Revolut_Challenge_Prakhyath\document_report_reformat.csv', index =
```

## 1.0.2 Exploratory Data Analysis

**Document Report** Document report has seen a significant upswing in the number of failures

```
In [45]: docrep['result'].value_counts().plot(kind='bar', stacked=True, legend=True)
          #df = pnd.DataFrame(docrep['result'].value_counts(), index=docrep.index)
          #ax = df.plot.bar(rot=0)
          #docrep['result'].apply(pnd.value_counts).plot.pie(subplots=True)
```

```
Out[45]: <matplotlib.axes._subplots.AxesSubplot at 0x29870907ef0>
```