Fabien Gueret – 28 March 2018 – Iwoka Business Finance

Call Centre / Data Samurai

Questions / Answers

1. Which agent made the most calls? [1]
   Agent Orange with 2234 calls.

2. For the leads that received one or more calls, how many calls were received on average? [2]
   The leads called at least once received 1.84 calls on average.

3. For the leads that signed up, how many calls were received, on average? [2]
   The signed up were called on average 2.10 times before they were convinced.

4. Which agent had the most signups? Which assumptions did you make? (note that there is a many-to-one relationship between calls and leads) [4]
   The agent Orange had 560 sign ups. We assume un-successful calls are all calls except the “INTERESTED” ones. We assume one phone number has one lead. We assume the sign-ups table corresponds to the interested in the calls table.

5. Which agent had the most signups per call? [2]
   But the most efficient agent for the most signups per call is agent Blue.

6. Was the variation between the agents’ signups-per-call statistically significant? Why? [5]
   To determine whether a result is statistically significant, we calculate the p-value associated with a binomial test, which is the probability of observing an effect given that the null hypothesis is true. The null hypothesis is that the Success rate of each agent does not differ from the overall average success rate of the team.
   Our results:

Agent	Calls	Signups	Success Rate	P value
Black	750	157	0.209	0.0018
Blue	199	58	0.291	0.2939
Green	339	97	0.286	0.2646
Orange	2234	560	0.251	0.3717
Red	1478	424	0.287	0.0162

We reject the null hypothesis for Black and Red, ie their performance is average

7. A lead from which region is most likely to be “interested” in the product? [3]

The most interested region is London

8. A lead from which sector is most likely to be "interested" in the product? [1]
The most interested sector is Consultancy

9. Given a lead has already expressed interest and signed up,
    a. signups from which region are most likely to be approved? [2]
    The region with the most approved sign-ups is the North-West

    b. Is this statistically significant? Why? [5]
    To determine whether a result is statistically significant, we calculate the p-value associated with a binomial test, which is the probability of observing an effect given that the null hypothesis is true. The null hypothesis is that the approval rate of each region does not differ from the country wide approval rate of sign-ups.
    Our results:

Region	Approval Rate	Countrywide	P value
North-West	0.45	0.33	0.0004

We reject the null hypothesis for the North-West ie the difference in approval rate is not significant.

10. Suppose you wanted to pick the 1000 leads most likely to sign up (who have not been called so far), based only on age, sector and region.
    a. What criteria would you use to pick those leads? [10]
    We would use a function of the age, sector and region that would give us a probability of interest/success. We have chose to fit a logistic regression model considering the sample sizes and the number of categories in the predictor variables. We have created bins for the age of the leads first then dichotomised it together with the region and sector
    Below the coefficients calculated:

Intercept	-0.08446591
Region_london	1.417680
Region_midlands	-0.654595
Region_north-east	-0.182676
Region_north-west	0.499336
Region_northern-ireland	-0.157100
Region_scotland	-0.210190
Region_south	-0.301667
Region_south-east	-0.078173
Region_south-west	0.359215
Region_wales	-0.776297
Sector_agriculture	-0.272434
Sector_construction	-0.871275
Sector_consultancy	0.854160
Sector_entertainment	0.262466
Sector_food	-0.079287
Sector_retail	-0.528669
Sector_wholesale	0.550573
Age Bucket_(0, 25]	0.150940
Age Bucket_(25, 50]	0.223575
Age Bucket_(50, 75]	0.178159
Age Bucket_(75, 100]	-0.029826

b. In what sense are those an optimal criteria set? [3]
The criteria is optimal in the sense that it minimises the error/cost function for the training data.

c. How many signups would you expect to get based on those called leads, assuming they were being called by random agents? [3]
According to our model the number of sign ups would be 5263 out of 9994 which is too high and comes from the leads (our test data) and training data overlapping. There are other models that could be used and should be tested to see which one is the most practical at this moment with the possibility to change once the data available is bigger

d. If you could choose the agents to make those calls, who would you choose? Why? [3]
Owing to the already dubious performance of our first model, adding another set of variables might not be a good idea. Furthermore we could make the hypothesi that the impact of the agent on the call is not region/age/sector specific. In that view, preselecting leads according to the 3 variables first then dividing them amongst the agent according to performance (question 4) and availability (question 1) could be a first strategy

Nevertheless we have a logistic model including the agent impact and out of the coefficients for the agents, Agent Blue seems to be the one with the most positive impact. We would have to test the significance of those coefficients.

Agent_black	-0.733619
Agent_blue	0.726713
Agent_green	0.132170
Agent_orange	-0.306021
Agent_red	0.246731

Below Python code and output

Code

```
# -*- coding: utf-8 -*-
"""
Created on Tue Mar 27 11:45:31 2018

@author: Fabien Gueret 4

Data Samurai Challenge

Call Centre

Data
1.leads.csv. This is a list of fictitious company directors, with some basic data about them and their company.
2.calls.csv. This is a list of fictitious calls made by an outbound call centre. The call centre consists of several agents,
  who make calls one after the other. They don't get to choose who to call, the system does. The objective of the call is to
  get the lead to signup on the website. When they finish a call, they mark down the outcome, from a fixed list of possible outcomes.
  Note that a single lead may be called multiple times.
3.signups.csv. This is a list of leads who signed up after being called by someone from the call centre.
  Each signup was risk assessed and either approved or rejected for a loan.

"""
# Dependencies

#Data management library
import pandas as pd
#Time management library
import datetime as dt
# Database Library
import numpy as np
# Statistics
from scipy import stats


#### Open csv files and save data in Dataframes ####
# paths
```

```
leadfile = 'leads.csv'
callfile = 'calls.csv'
signupfile = 'signups.csv'
# inflow of data
leads = pd.read_csv(leadfile,header = 0,index_col=0, converters ={'Age':int})
print(leads.head())
calls = pd.read_csv(callfile,header = 0,index_col=3)
print(calls.head())
signups = pd.read_csv(signupfile,header = 0,index_col=0)
print(signups.head())
```

1.Which agent made the most calls?

```
agents_activity = calls.groupby('Agent').count().drop(['Call Outcome'],axis=1)
#print(agents_activity)
prolific_agent = agents_activity.sort_values(['Phone Number'], ascending=False).head(1)
print(prolific_agent)
```

2.For the leads that received one or more calls, how many calls were received on average?

```
lead_contacts = calls.groupby('Phone Number').count()
avg_calls_number = lead_contacts['Call Outcome'].mean()
print(avg_calls_number)
```

3.For the leads that signed up, how many calls were received, on average?

```
signed_ups_phone_number= pd.merge(signups, leads , how='left', left_index= True,
right_index=True)
signed_up_calls = pd.merge(signed_ups_phone_number, calls, how='left', left_on= 'Phone
Number', right_on= 'Phone Number')
signed_up_call_counts = signed_up_calls.groupby('Phone Number').count()
signed_up_call_avg = signed_up_call_counts['Call Outcome'].mean()
print(signed_up_call_avg)
```

4.Which agent had the most signups? Which assumptions did you make? (note that there

is a many-to-one relationship between calls and leads) ####

success_calls = calls[calls['Call Outcome']=='INTERESTED']
agents_success = success_calls.groupby('Agent').count().drop(['Call Outcome'],axis=1)
best_agent = agents_success.sort_values(['Phone Number'], ascending=False).head(1)
print('Best Agent ', best_agent)

5.Which agent had the most signups per call?

agent_effort_to_signup = pd.merge(agents_activity,agents_success,how='left', left_index= True, right_index=True)
agent_effort_to_signup['Success Rate']= agent_effort_to_signup['Phone Number_y']/agent_effort_to_signup['Phone Number_x']
efficient_agent = agent_effort_to_signup.sort_values(['Success Rate'], ascending=False).drop(['Phone Number_y','Phone Number_x'],axis=1).head(1)
print('Efficient Agent', efficient_agent)

6.Was the variation between the agents' signups-per-call statistically significant? Why?

Ho : p1 = avg Ha p1!= avg

avg_success = agent_effort_to_signup['Phone Number_y'].sum()/ agent_effort_to_signup['Phone Number_x'].sum()
agent_effort_to_signup['p_value']=[stats.binom_test(row[1],row[0],avg_success) for index , row in agent_effort_to_signup.iterrows()]
print(agent_effort_to_signup)

7.A lead from which region is most likely to be "interested" in the product?

lead_regions= pd.merge(leads, calls, how='right', left_on= 'Phone Number', right_on='Phone Number')
# calculate the number of interest number by region
lead_interested_by_regions = lead_regions.loc[lead_regions['Call Outcome']=='INTERESTED'].groupby('Region').count()
# calculate the number of unique phone numbers called by region (many calls for one number!)
lead_phone_number = calls.groupby('Phone Number').count()
all_phone_numbers_regions=pd.merge(lead_phone_number,leads,how='left', left_index= True,

```
right_on='Phone Number')
all_leads_called_by_regions = all_phone_numbers_regions.groupby('Region').count()
interested_region_data=pd.merge(lead_interested_by_regions,all_leads_called_by_regions,how
='inner', left_index= True, right_index=True)
interested_region_data['InterestedvsAll']=interested_region_data['Age_x']/interested_region_dat
a['Age_y']
most_interested_region = interested_region_data.sort_values(['InterestedvsAll'],
ascending=False).head(1)
print('Interested region : ',most_interested_region['InterestedvsAll'])


#### 8.A lead from which sector is most likely to be “interested” in the product? ####

# calculate the number of interest number by sectors
lead_interested_by_sectors = lead_regions.loc[lead_regions['Call
Outcome']=='INTERESTED'].groupby('Sector').count()
all_leads_called_by_sectors = all_phone_numbers_regions.groupby('Sector').count()
interested_sector_data=pd.merge(lead_interested_by_sectors,all_leads_called_by_sectors,how='i
nner', left_index= True, right_index=True)
interested_sector_data['InterestedvsAll']=interested_sector_data['Age_x']/interested_sector_data[
'Age_y']
most_interested_sector = interested_sector_data.sort_values(['InterestedvsAll'],
ascending=False).head(1)
print('Interested sector : ', most_interested_sector['InterestedvsAll'])


#### 9.Given a lead has already expressed interest and signed up, ####
#### 9.a.signups from which region are most likely to be approved? ####

signups_info = pd.merge(signups , leads , how ='left', left_index = True, right_index = True)
signups_region = signups_info.groupby('Region').count()
approved_signups_region = signups_info.loc[signups_info['Approval
Decision']=='APPROVED'].groupby('Region').count()
approved_region_data=pd.merge(signups_region, approved_signups_region,how='inner',
left_index= True, right_index=True)
approved_region_data['ApprovedvsAll']=approved_region_data['Age_y']/approved_region_dat
a['Age_x']
most_approved_region = approved_region_data.sort_values(['ApprovedvsAll'],
ascending=False).head(1)
print('Approved region : ', most_approved_region['ApprovedvsAll'])
```

```
#### 9.b.Is this statistically significant? Why? ####
# Ho : p1 = avg Ha p1!= avg

avg_approved = signups.loc[signups['Approval
Decision']=='APPROVED'].count()/signups.count()
avg =avg_approved.sum()
approved_region_data['average']=avg
approved_region_data['p_value']=[stats.binom_test(row[4],row[0],avg) for index , row in
approved_region_data.iterrows()]
print(approved_region_data)


#### 10 Suppose you wanted to pick the 1000 leads most likely to sign up (who have not been
called so far), based only on age, sector and region.####
#### 10.a.What criteria would you use to pick those leads? ####

all_called = pd.merge(calls,leads, how ='left', left_on = 'Phone Number', right_on = 'Phone
Number')
conditions = [
    (all_called['Call Outcome'] == 'INTERESTED'),
    (all_called['Call Outcome'] == 'NOT INTERESTED')]
choices = [1,0]
all_called['Signup']=np.select(conditions, choices, default='rid')
all_called = all_called.loc[all_called['Signup']!='rid']
# Bin the Age
all_called['Age Bucket']= pd.cut(all_called['Age'],range(0,125,25))
#clean up before binarisation
all_called=all_called.drop(['Phone Number','Age', 'Agent','Call Outcome'],axis=1)
# Binaries predictors and target
training_data = pd.get_dummies(all_called)
training_data= training_data.drop(['Signup_0'],axis=1)
#print(training_data.columns.values)

#Import Library
from sklearn.linear_model import LogisticRegression

# Create logistic regression object
model = LogisticRegression()
# X (predictor) and Y (target) for training data set and x_test(predictor) of test_dataset
```

```
X = training_data.drop(['Signup_1'],axis=1).values
X_data=training_data.drop(['Signup_1'],axis=1)
y= training_data['Signup_1'].values
# Train the model using the training sets and check score
model.fit(X, y)
model.score(X, y)
#Equation coefficient and Intercept
coeff =model.coef_[0]
features =X_data.columns.values
print('Intercept: \n', model.intercept_)

Results = pd.DataFrame(list(zip(features,coeff)),columns= ['features', 'estimated_Coefficients'])
print(Results)
#### 10.b.In what sense are those an optimal criteria set?


#### 10.c.How many signups would you expect to get based on those called leads, assuming
they were being called by random agents?

all_leads = leads[['Region','Sector','Age']]
#print(all_leads)

# Bin the Age
all_leads['Age Bucket']= pd.cut(all_leads['Age'],range(0,125,25))
#clean up before binarisation
all_leads=all_leads.drop(['Age'],axis=1)
# Binaries predictors and target
testing_data = pd.get_dummies(all_leads)
#print(testing_data.columns.values)

#Predict Output
x_test = testing_data.values

predicted= model.predict(x_test)

print('Predicted signps',sum(predicted),' on ', len(leads))

#### 10.d.If you could choose the agents to make those calls, who would you choose? Why?

all_called = pd.merge(calls,leads, how ='left', left_on = 'Phone Number', right_on = 'Phone
```

```
Number')
conditions = [
    (all_called['Call Outcome'] == 'INTERESTED'),
    (all_called['Call Outcome'] == 'NOT INTERESTED')]
choices = [1,0]
all_called['Signup']=np.select(conditions, choices, default='rid')
all_called = all_called.loc[all_called['Signup']!='rid']
# Bin the Age
all_called['Age Bucket']= pd.cut(all_called['Age'],range(0,125,25))
#clean up before binarisation
all_called=all_called.drop(['Phone Number','Age','Call Outcome'],axis=1)
# Binaries predictors and target
training_data = pd.get_dummies(all_called)
training_data= training_data.drop(['Signup_0'],axis=1)
#print(training_data.columns.values)


# Create 2nd logistic regression object
model2 = LogisticRegression()
# X (predictor) and Y (target) for training data set and x_test(predictor) of test_dataset
X = training_data.drop(['Signup_1'],axis=1).values
X_data=training_data.drop(['Signup_1'],axis=1)
y= training_data['Signup_1'].values
# Train the model using the training sets and check score
model2.fit(X, y)
model2.score(X, y)
#Equation coefficient and Intercept
#print('Coefficient: \n', model2.coef_)
print('Intercept: \n', model2.intercept_)

coeff =model2.coef_[0]
features =X_data.columns.values

Results = pd.DataFrame(list(zip(features,coeff)),columns= ['features', 'estimated_Coefficients'])
print(Results)
```

```
Output
Python 3.6.3 |Anaconda custom (64-bit)| (default, Oct 15 2017, 03:27:45) [MSC v.1900 64 bit (AMD64)]
Type "copyright", "credits" or "license" for more information.

IPython 6.1.0 -- An enhanced Interactive Python.

runfile('F:/New folder/Job Search/iwoca/callcentre.py', wdir='F:/New folder/Job Search/iwoca')
                 Phone Number      Region       Sector  Age
Name
Isabela MEZA       175718505368  north-west    wholesale   19
Deangelo LEE       937521423043  north-west       retail   38
Rosia MENDEZ       403640999962    midlands  agriculture   40
Jeremiah GALLOWAY  946740713605    scotland         food   23
Sarah POPE         264176984341    midlands       retail   18
             Phone Number     Call Outcome   Agent
Call Number
0             83473306392   NOT INTERESTED  orange
1            762850680150  CALL BACK LATER  orange
2            476309275079   NOT INTERESTED  orange
3            899921761538  CALL BACK LATER     red
4            906739234066  CALL BACK LATER  orange
                  Approval Decision
Lead
Tyree TERRY            APPROVED
Ansel WOOD             REJECTED
Ludwig DIAZ            APPROVED
Mack ARELLANO          APPROVED
Judy HENDRICKS         REJECTED
        Phone Number
Agent
orange          2234
1.839587932303164
2.0989583333333335
Best Agent          Phone Number
Agent
orange           560
Efficient Agent        Success Rate
Agent
```

```
blue      0.291457
       Phone Number_x  Phone Number_y  Success Rate   p_value
Agent
black            750             157      0.209333  0.001754
blue             199              58      0.291457  0.293865
green            339              97      0.286136  0.264620
orange          2234             560      0.250671  0.371723
red             1478             424      0.286874  0.016189
Interested region :  Region
london    0.756757
Name: InterestedvsAll, dtype: float64
Interested sector :  Sector
consultancy    0.651515
Name: InterestedvsAll, dtype: float64
Approved region :  Region
north-west    0.452381
Name: ApprovedvsAll, dtype: float64
                 Approval Decision_x  Phone Number_x  Sector_x  Age_x  \
Region
london                            25              25        25     25
midlands                          91              91        91     91
north-east                        82              82        82     82
north-west                       210             210       210    210
northern-ireland                  24              24        24     24
scotland                          82              82        82     82
south                             32              32        32     32
south-east                        86              86        86     86
south-west                       102             102       102    102
wales                             34              34        34     34

                 Approval Decision_y  Phone Number_y  Sector_y  Age_y  \
Region
london                             2               2         2      2
midlands                          26              26        26     26
north-east                        20              20        20     20
north-west                        95              95        95     95
northern-ireland                   6               6         6      6
scotland                          37              37        37     37
south                             12              12        12     12
south-east                        29              29        29     29
```

south-west	25	25	25	25
wales	5	5	5	5

	ApprovedvsAll	average	p_value
Region			
london	0.080000	0.334635	0.005102
midlands	0.285714	0.334635	0.374400
north-east	0.243902	0.334635	0.100593
north-west	0.452381	0.334635	0.000424
northern-ireland	0.250000	0.334635	0.517108
scotland	0.451220	0.334635	0.034372
south	0.375000	0.334635	0.708273
south-east	0.337209	0.334635	1.000000
south-west	0.245098	0.334635	0.058888
wales	0.147059	0.334635	0.018305

Intercept:
[-0.08446591]

	features	estimated_Coefficients
0	Region_london	1.417680
1	Region_midlands	-0.654595
2	Region_north-east	-0.182676
3	Region_north-west	0.499336
4	Region_northern-ireland	-0.157100
5	Region_scotland	-0.210190
6	Region_south	-0.301667
7	Region_south-east	-0.078173
8	Region_south-west	0.359215
9	Region_wales	-0.776297
10	Sector_agriculture	-0.272434
11	Sector_construction	-0.871275
12	Sector_consultancy	0.854160
13	Sector_entertainment	0.262466
14	Sector_food	-0.079287
15	Sector_retail	-0.528669
16	Sector_wholesale	0.550573
17	Age Bucket_(0, 25]	0.150940
18	Age Bucket_(25, 50]	0.223575
19	Age Bucket_(50, 75]	0.178159
20	Age Bucket_(75, 100]	-0.029826

Predicted signps 5263 on 9994

Intercept:
[0.06597395]

	features	estimated_Coefficients
0	Agent_black	-0.733619
1	Agent_blue	0.726713
2	Agent_green	0.132170
3	Agent_orange	-0.306021
4	Agent_red	0.246731
5	Region_london	1.492626
6	Region_midlands	-0.665632
7	Region_north-east	-0.184671
8	Region_north-west	0.537006
9	Region_northern-ireland	-0.194053
10	Region_scotland	-0.187785
11	Region_south	-0.235415
12	Region_south-east	-0.069980
13	Region_south-west	0.402555
14	Region_wales	-0.828678
15	Sector_agriculture	-0.224853
16	Sector_construction	-0.906547
17	Sector_consultancy	0.901235
18	Sector_entertainment	0.291083
19	Sector_food	-0.062458
20	Sector_retail	-0.526341
21	Sector_wholesale	0.593855
22	Age Bucket_(0, 25]	0.114009
23	Age Bucket_(25, 50]	0.202811
24	Age Bucket_(50, 75]	0.126109
25	Age Bucket_(75, 100]	-0.039033