

BTS – MBDS – Data Driven Business

Assignment 2:

2.A: Data Science Team

2.B: Creating Value From Data

2.C: C-Python RFM Analysis

27 Febrero, 2021

2.A: Data Science Team

Who Am I?: In Data Science Team I would be a **Data Scientist** who can deploy models to productions (Machine Learning Engineer)

Why?: I enjoy the work, I like the combination of technical and business (not the reporting part but the thinking about adding value to the business part), and the possibility of a potentially different problem every time, to think in new different ways.

What are the requirements?: Knowledge of statistical methods, knowledge of machine learning and deep learning techniques, knowledge of ingesting data into development environments (distributed or on memory), knowledge of production deployments.

2.B: Creating Value From Data

Insights

- 1) Value before Data: Less about what data and more about where can we create more value
- 2) Seven Practical Steps:
 - 1) Understand how you create value today
 - 2) Develop investment options
 - 3) Review your data assets
 - 4) Confirm the options
 - 5) Develop realistic aspirations for monetisation
 - 6) Trial it, learn and refine
 - 7) Do it, scale it
- 3) Organizations should avoid the initial hype and think carefully about value added, and define a strategy for investing resources so that the output has a direct impact in the profit margins and only then execute and scale.

2.C: C-Python Exercise

```
In [25]: # RFM segmentation mapping
seg_map = {
    r'[1-2][1-2]': 'Hibernating',
    r'[1-2][3-4]': 'At_Risk',
    r'[1-2]5': 'Cant_Loose',
    r'3[1-2]': 'About_to_Sleep',
    r'33': 'Need_Attention',
    r'[3-4][4-5]': 'Loyal_Customers',
    r'41': 'Promising',
    r'51': 'New_Customers',
    r'[4-5][2-3]': 'Potential_Loyalists',
    r'5[4-5]': 'Champions'
}
```

```
In [28]: # Agg segmentation
rfm_df['Segment'] = rfm_df['RecencyScore'].astype(str) + rfm_df['FrequencyScore'].astype(str)
rfm_df['Segment'] = rfm_df['Segment'].replace(seg_map, regex=True)
rfm_df.head()
```

Out[28]:

	Recency	Frequency	Monetary	RecencyScore	FrequencyScore	MonetaryScore	RFM_SCORE	Segment
Customer ID								
12346.0	65	46	578.36	3	3	5	335	Need_Attention
12608.0	38	16	26.04	3	2	1	321	About_to_Sleep
12745.0	120	22	67.98	2	2	2	222	Hibernating
12746.0	161	20	73.95	2	2	2	222	Hibernating
12747.0	3	162	716.11	5	5	5	555	Champions

```
In [29]: rfm_df[["Segment", "Recency", "Frequency", "Monetary"]].groupby("Segment").agg(["mean", "count", "std"])
```

Out[29]:

Segment	Recency			Frequency			Monetary		
	mean	count	std	mean	count	std	mean	count	std
About_to_Sleep	50.471875	320	10.191850	15.459375	320	8.642947	157.263844	320	1002.325505
At_Risk	159.421053	570	80.158553	58.215789	570	24.071308	185.023512	570	110.579890
Cant_Loose	123.534884	86	50.772780	232.046512	86	162.221048	705.643500	86	686.948763
Champions	4.732899	614	4.512118	261.314332	614	370.497612	851.894502	614	1413.504487
Hibernating	207.668410	956	90.729319	13.424686	956	8.644677	157.522742	956	1387.520343
Loyal_Customers	34.721358	707	15.875762	173.205092	707	154.443657	520.840983	707	442.555837
Need_Attention	48.276316	152	10.264191	45.730263	152	8.535838	411.011908	152	3227.238742
New_Customers	5.783333	60	4.333909	7.216667	60	3.987658	33.589500	60	43.270695
Potential_Loyalists	15.157895	475	9.308945	36.709474	475	13.253445	126.599411	475	118.514052
Promising	22.421053	95	5.158300	8.200000	95	3.891070	33.052632	95	32.215465