Journey App Deliverable 3 Final Deliverable

Andrew Whittum

Jaewoo Kang

Giacomo Ieronutti

Along with this pdf we are submitting out Jupyter Notebook as part of this deliverable (JourneyDeliverble3Team2.ipynb). The Jupyter Notebook is fairly straightforward to run and all code blocks are commented. Nevertheless, we will give some instructions, commentary and explanation below.

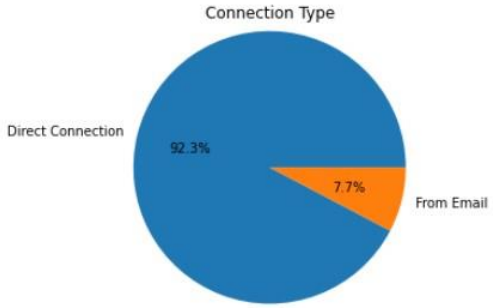
The first several blocks of code are focused exclusively on processing the data. There were a few corrupted lines that we came across so we deleted those. We also altered the formatting slighting on some of the rows (for example which change the data type on some of the rows). The biggest change we applied to the data was for the "Customer Id" column. There was a lot of other information besides just the company id, so we extracted and kept on the company id.

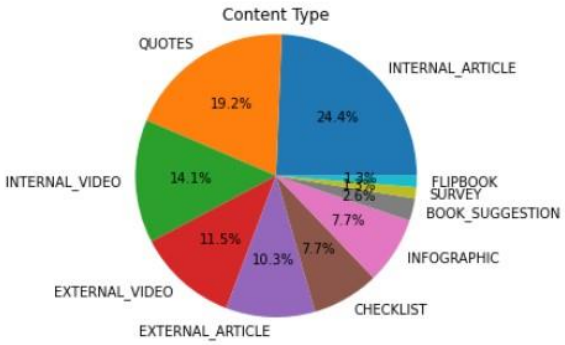
We also ended up adding four new columns to the dataset. We included the day of the week for both the Activation Time and the Activity Time. Then we extracted just the time from the Activity Time column and made that it's own column (the Activity Time column also included the date, so we just isolated the hour, minute, and second for this new column). Finally we added a column titled "Activity Delay" which records the number of hours elapsed between when a user received a piece of content and when they opened it.

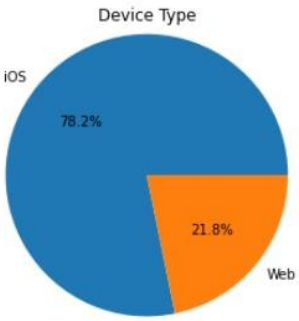
After this we defined six functions. The first four are for our analysis and the last two are for visualization. The first function is "eliminate_repeats" this function takes data for one user and looks to see if they opened a piece of content more than once. We had noticed that some users had opened a piece of content many times (sometimes more than 20 or 30 times). This function only retains the first time that a user opened a piece of content. We defined a repeated content as any piece of content that had the same name and the same activation date. There were some instances where a content with the same name was sent out more than once by Vibons; we considered these instances to be separate pieces of content. If this type function isn't desired by the client, it can be removed by deleting the line of code that calls thi sfunciton. The next function is called "sort". This function takes the input for one user and sorts all of the rows by day of the week, and within each day of the week it sorts them by time from earliest to latest. The next function "process_user" eliminates the columns that we aren't considering in this analysis, reorders the columns and applies the two functions mentioned above. After this function is applied, the data for the user is ready to be passed to the function which will generate the recommended times. This function is called "generate_user_recommendations" and will only consider times when the completion rate is 85% or higher. If there are no such times, it will then look for any times between 50% and 85% percent. If it still can't find any times, then it will return times that are below 50%. The output from this function is a dataframe for one user. The first four columns included are User Id, Consumer Id, Activity Day, Time, Name, Content Type, Journey Name, Action, Device, Channel, and Activity Delay. A sample output is given below.

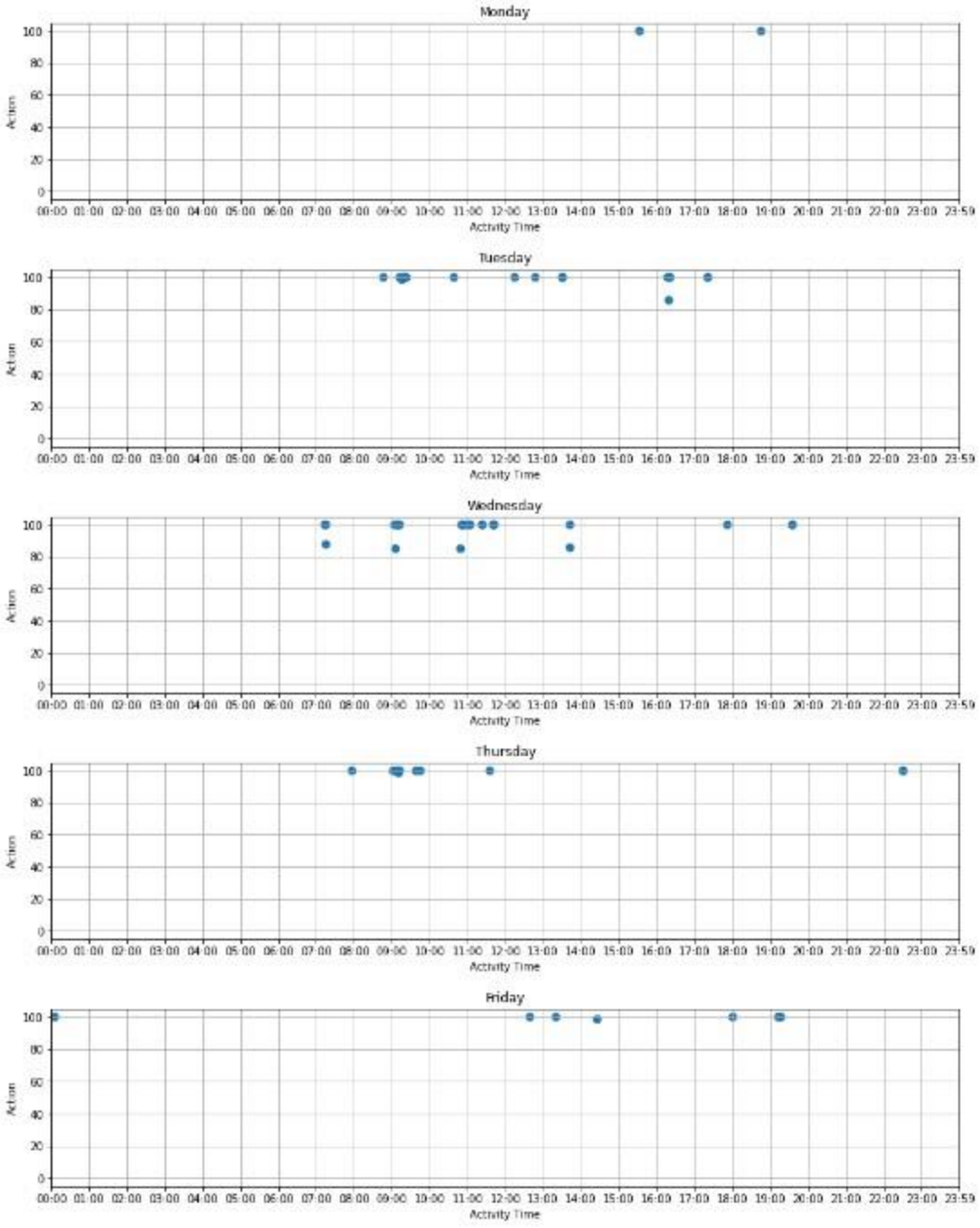
	Customer Id	User Id	Activity Day	Time	Name	Content Type	Journey Name	Action	Device	Channel	Activity Delay
3	34	15000	Monday	15:32:21	OKRs Nedir?	INTERNAL_VIDEO	Yönlendirme ve Yönetme	100	iOS	Direct Connection	3007.32
7	34	15000	Monday	18:44:48	Canlı Yayında Hangi Konu Konuşulsun?	SURVEY	Zor Zamanlarla Başa Çıkma	100	iOS	Direct Connection	96.75
17	34	15000	Tuesday	08:45:38	Bill Gates'den İlham Verici Fikir	QUOTES	Yönlendirme ve Yönetme	100	iOS	Direct Connection	0.76
21	34	15000	Tuesday	09:11:35	Destekleyen liderlik nasıl uygulanır?	INTERNAL_ARTICLE	Yönlendirme ve Yönetme	100	iOS	Direct Connection	1.19
22	34	15000	Tuesday	09:15:22	Liderlerin Verimliliğini Artıran İletişim Takt...	INTERNAL_ARTICLE	Yönlendirme ve Yönetme	99	Web	From Email	1.26
23	34	15000	Tuesday	09:18:56	Bilişsel Yanlılıklar - Dikkat Etmeniz Gerekenler	CHECKLIST	Zor Zamanlarla Başa Çıkma	100	iOS	Direct Connection	91.32
24	34	15000	Tuesday	09:19:07	Hızlı Düşünen Beynimiz Oyun Mu Oynuyor?	EXTERNAL_VIDEO	Zor Zamanlarla Başa Çıkma	100	iOS	Direct Connection	96.99
25	34	15000	Tuesday	09:19:53	Performansı Yönetirken Etkili Koçluk Nasıl Ver...	INTERNAL_ARTICLE	Yönlendirme ve Yönetme	100	iOS	Direct Connection	3865.33
26	34	15000	Tuesday	09:19:56	Bill Gates'den İlham Verici Fikir	QUOTES	Yönlendirme ve Yönetme	100	iOS	Direct Connection	4033.33
27	34	15000	Tuesday	09:20:33	Başarılı İletişim için 3 CEO'nun "3 Kuralı"	INTERNAL_ARTICLE	İletişim	100	iOS	Direct Connection	3985.34

After these functions we have included to functions to visualize the data. The first function is called “graph_user” and takes the recommendations from the format above and prints out a graph for each day of the week showing time on the x-axis and the completion on the y axis. The next graphing function is called “pie_charts” and generates a pie chart for the “Content Type,” “Device,” and “Channel” columns.








Monday
Tuesday
Wednesday
Thursday
Friday
Action
Activity Time
0
20
40
60
80
100
00:00 01:00 02:00 03:00 04:00 05:00 06:00 07:00 08:00 09:00 10:00 11:00 12:00 13:00 14:00 15:00 16:00 17:00 18:00 19:00 20:00 21:00 22:00 23:00 23:59

After this we have included a block of code which will apply the analysis to every user in the database. First, we get a list of all the unique user ids in the database. Then the four analysis functions listed previously will be applied to the user, and the output will be saved in a csv file, where the title of the csv file is the user id number. We included this because to generate the output in the picture above, anyone using this notebook would have to put in an individual username manually, and given that there are tens of thousands of users in the database, this would take a very long time. This function should take about three to four hours to run in total.

The final block of the code applies the basic Natural Language Processing procedure to the Name column that we mentioned earlier in the semester. This was not strictly necessary but we thought we would include it if it may be of interest to the client in the future. This calculates the average completion for each word that the user saw in the “Name” column.

29	bulma	100.000000
30	meslek	100.000000
31	yıldızı	100.000000
32	makas	100.000000
33	journey	100.000000
34	üretici	98.636364
35	pos	97.857143
36	limit	97.666667
37	ten	96.250000
38	denizcilerimize	96.250000
39	ateş	96.250000
40	hakan	96.250000