NATIONAL HEALTH SERVICES(NHS)

APPROACHES AND INSIGHTS

Part 1 – Background and context

National Health Services (NHS), a publicly funded healthcare system in England. The NHS incurs significant, potentially avoidable, costs when patients miss general practitioner (GP) appointments.

This data analysis report will provide NHS with better insights of how they can eliminate this avoidable cost. This report will further assist to address the two business questions below

- Has there been adequate staff and capacity in the networks?
- What was the actual utilisation of resources?

To explore the main business problem, we have sub-divided this to build a framework of questions as below.

- What is the number of locations, service settings, context types, national categories, and appointment statuses in the data sets?
- What is the date range of the provided data sets, and which service settings reported the most appointments for a specific period?
- What is the number of appointments and records per month?
- What monthly and seasonal trends are evident, based on the number of appointments for service settings, context types, and national categories?
- What are the top trending hashtags (#) on Twitter related to healthcare in the UK?
- Were there adequate staff and capacity in the networks?
- What was the actual utilisation of resources?
- What possible recommendations does the data provide for the NHS?

Part 2 – Analytical approach

Firstly, imported the various libraries i.e., pandas and NumPy

Importation of three data sets on python, followed by a read function to sense-check the data set for any unusual items. Missing values verified using the isna().sum() function to see if any columns miss a value in the ad, nc and ar datasets.

To address the first sub-set question, we decided to use the national category data set and used sub-location column as well as the value_count function to count the no of locations. Refer to the syntax code in appendix 1. The total number of locations are 106 locations. To further analyse this question, we performed separate analysis per service setting, context type and national categories, appointment status by sorting by columns to obtain the output. **Refer to appendix 2 syntax**.

In order to identify the date range, imported the datetime module and used the ad dataset to change the 'appointment date' to this format. This enabled us to identify the minimum and maximum date in each data set. **Refer to appendix 3.** To further analyse the date range in a specific period we decided to use the loc function to filter out data sub location name and created a set for date range to determine the number of service settings in that location within the date time frame. **Refer to appendix 4.** Value_counts used to determine the total count per service setting in that location.

To determine the total number of appointment and records, firstly we created two separate columns for month and year using the dt. year and dt. month in the nc database. Then, used the group by function to aggregate function to display the column for year and month column and total sum of monthly count of appointments. **Refer to appendix 5.** To further explore the output to determine the highest number of appointment sort_value and sum function was used to count the appointments in ascending order. **Refer to appendix 6**. The highest number of appointments were in November 2021.

Based on the analysis, General practice setting had the most number of appointments in all months. This category has the most fluctuation month-wise i.e., November 2021 was the highest number of appointments in the general category practice whereas extended access provisions had the higher number of appointments in the month of March 2022. Furthermore, for context type month-wise fluctuations were evident in context type "Care Related Encounter" which resulted in highest number of appointments in November 2021 while other context types unmapped and inconsistent mapping slightly fluctuated over the months.

To determine the top trending hashtags, we imported the tweets.csv file and filter the first 5 columns using the. head () function. Decision was made to change the tweets data to string. Loop in function was used to display results with # only. **Refer to appendix 7.** Further, created a new panda series for hashtags and .value_count used to count the number of times the hashtags were used. A decision was made to use iloc function to determine the top 30 hashtags. **Refer to appendix 8.** Use of rename function to rename the # column to "word" and "count". To provide further insights we used the query function to filter hashtags count over 10. **Refer to appendix 9.** The most popular hashtag was #healthcare. To obtain better analysis we also excluded the highest count hashtag to identify other hashtags that were popular. From our analysis we excluded counts over 10 and obtained results of #AI being the most popular hashtag.

The decision was made to use the lambda function to create an additional column of utilisation per day.

PART 3 – Dashboard design and development

The aim of the dashboard is to provide NHS better insights to answer business problem around adequacy of resources and utilization.

The decision was made to import sea born and matpotlib library to create these visualizations.

The first visualization was a line plot, that assisted in identifying the monthly trends between the service settings. The parameters used to build the line plot are count of appointments and appointment month with an interactive legend (hue) used to determine the service settings. The syntax used for this visualization was the group by function which provided the month-wise count of appointment per service setting. General practice category had the highest number of appointments.

The second visualization was also a line plot to determine the count of appointment per contextual type. The outcome with the most appointments was care-related encounter. Similarly, a line plot was created to visualize the national category with structured medical review category having the highest number of appointments. Interactive legends are better way to present data.

To identify the seasonal trends the decision was made to create line plots for each season month-wise. The group by function and appointment date range was used as parameters to understand the seasonal trends, various features such as the colour palette, the hue function, titles were used to make

the visualizations user-friendly to present to the stakeholders to provide them better insights on the number of appointments per service setting category date wise.

To identify the most popular twitter trend we used bar plots. This made visualization clear. We created two bar plot one identifying the most popular hashtag and another excluding the highest count hashtags. Stakeholders can gain a key output by identifying a new trend i.e., # AI which was twitted more in the category after #healthcare and #medicine.

Furthermore, visualisation was built to identify trends in the count of appointments after Aug 2021 to understand the most appointments. We used the syntax code group by to the ar dataset to include additional columns for the data types and specified the appointment month range between Aug 2021 to June 2022. Line plots were used to present these visualizations with Nov 2021 showing the highest number of appointments. Further, a new data frame was created to add a column for utilisation per day. This will enable stakeholders to resolve the business problem comparing the total number of appointments per day to the current maximum capacity.

Other visualization i.e., line plots were used to answer business questions to verify the health care professionals' overtime, appointment mode, time between the slot and appointment and appointment status. Hue was used to create a legend enabling the stakeholders to address these questions.

PART 4 – Patterns, trends and insights

General practice by far has the greatest number of appointments. The number of appointments fluctuated occasionally with November 2021 being the highest number of appointments**.** Extended Access provisions had the most appointments in March 2022, how-ever the graph remained consistent throughout the period. NHS should be curious on unmapped category so that stakeholders can see if there are any unusual items in that category in terms of the count of appointments.

The contextual type that had most appointments were "Care related encounter". Stakeholder would wish to dig down further questions on why there is inconsistent mapping and unmapped category to identify any unusual trends in contextual type and no of appointments in these categories. On the other hand, the national category line plot was also used to identify any trends, general consultation routine, general consultation acute, service provided by organisation external to the practice. Other categories remained constant were home visit, unplanned clinical activity, unmapped and patient contact during care home round.

Seasonal trends were seen in the months of Aug (Summer days) etc with some months seeing higher number of appointments. Refer to the presentation slides for further explanation.

The highest number of tweets was for hashtags #healthcare with 716 counts. This means that people tweeted for healthcare visited NHS often. NHS should focus on their social media handle to identify any such negative tweets to improvise on their services. Furthermore, customers also tweeted on AI which could been automatic required in medical services provided.

PART 5 – Next steps

To address the main business questions further investigation is required

Staff data- The number of staff employed per unit would provide better insights on the actual utilisation to the capacity.

Customer reviews through market research would provide better insights on the quality of services provide insights to better understand how NHS can improve on their services and how satisfied the customers with the current services.

Other financial data such as revenue outcomes and cost in each category would be able to provide better insights of performing and non-performing departments. This would enable NHS to inject resources.

Appendix

1) # Selecting few columns.
   Locations = pd. read_excel ('national_categories.xlsx',
   usecols=['sub_icb_location_name'])
   # Print the DataFrame
   Locations.head()
   #Determine the number of locations
   Locations = nc['sub_icb_location_name'].value_counts()
   Locations.count()

2) #Number of service stations, context type and national_category
   ## Selecting few columns.
   Locations = pd. read_excel('national_categories.xlsx',
   usecols=['service_setting','context_type','national_category'])
   # Print the DataFrame.
   Locations.head()

Syntax per category and output

```
In [10]: #Number of service setting
         Locations = nc['service_setting'].value_counts()
         Locations.count()

Out[10]: 5
```

```
In [11]: #Number of context type
         Locations = nc['context_type'].value_counts()
         Locations.count()

Out[11]: 3
```

```
In [12]: #Number of national category
         Locations = nc['national_category'].value_counts()
         Locations.count()

Out[12]: 18
```

```
In [13]: #Number of appointment status
         # Selecting few columns.
         appointment_status = pd.read_csv('appointments_regional.csv',
                                          usecols=['appointment_status'])

         # Print the DataFrame.
         appointment_status.head()
```

In [13]:
```
#Number of appointment status
# Selecting few columns.
appointment_status = pd.read_csv('appointments_regional.csv',
                                 usecols=['appointment_status'])

# Print the DataFrame.
appointment_status.head()
```

Out[13]:

	appointment_status
0	Attended
1	Attended
2	Attended
3	Attended
4	Attended

In [14]:
```
#Number of appointment status
appointment_status = ar['appointment_status'].value_counts()
appointment_status.count()
```

Out[14]: 3

3) Date and time module in ad and nc data sets

```
# converting the string to# Import modules and classes.

# Import modules and classes.
# Import the datetime module and datetime class.
from datetime import datetime

ad['appointment_date'] = pd.to_datetime(ad['appointment_date'])

# printing dataframe
ad.head()

print(ad.dtypes)
```

```
sub_icb_location_code               object
sub_icb_location_ons_code           object
sub_icb_location_name               object
icb_ons_code                        object
region_ons_code                     object
appointment_date            datetime64[ns]
actual_duration                     object
count_of_appointments                int64
dtype: object
```

In [22]:
```
# converting the string to# Import modules and classes.

# Import modules and classes.
# Import the datetime module and datetime class.
from datetime import datetime

nc['appointment_date'] = pd.to_datetime(nc['appointment_date'])

# printing dataframe
nc.head()

print(nc.dtypes)
```

```
appointment_date         datetime64[ns]
icb_ons_code                     object
sub_icb_location_name            object
service_setting                  object
context_type                     object
national_category                object
count_of_appointments             int64
appointment_month                object
dtype: object
```

4) USE OF LOC FUNCTION TO SUBSET A DATA SET AND DATE RANGE

In [26]:
```
#Use loc function to create a subset
nc_sub = nc[['service_setting','sub_icb_location_name','appointment_date']]
nc_sub2 = nc_sub.loc[nc_sub['sub_icb_location_name']=='NHS North West London ICB - W2U3Z']
nc_sub3 = nc_sub2[(nc_sub2['appointment_date']>='2022-01-01') &(nc_sub2['appointment_date']<='2022-06-01')]

#view the output
print (nc_sub3['service_setting'].value_counts())
```

```
General Practice            2104
Other                       1318
Primary Care Network        1272
Extended Access Provision   1090
Unmapped                     152
Name: service_setting, dtype: int64
```

5) USE OF GROUPBY AND AGGREGATE FUNCTION

```
nc['year']=pd.to_datetime(nc['appointment_month']).dt.year
nc['month']=pd.to_datetime(nc['appointment_month']).dt.month


#view
nc
```

```
transaction_groupby_obj = nc.groupby(['year','month'])
transaction_groupby_obj.agg({'count_of_appointments':['sum']})
```

:

		count_of_appointments
		sum
year	month	
2021	8	23852171
	9	28522501
	10	30303834
	11	30405070
	12	25140776
2022	1	25635474
	2	25355260
	3	29595038
	4	23913060
	5	27495508
	6	25828078

6) USE OF SORT VALUE AND ASCENDING

```
#The month with highest appointments
transaction_groupby_stock_code_obj = nc.groupby('month')
transaction_groupby_stock_code_obj.agg({'count_of_appointments':['sum']}).sort_values(by=('count_of_appointments','sum'),ascendir
```

7) LOOP FUNCTION TO CREATE A SPLIT

```
In [55]: tags=[]
         for y in [x.split(' ') for x in tweets['tweet_full_text'].values]:
             for z in y:
                 if '#' in z:
                     # Change to lowercase.
                     tags.append(z.lower())
         #view the output
         tags
```

```
Out[55]: ['#healthcare',
          '#premisehealth',
          '#hiring',
          '#healthcare',
          '🚨#new:🚨',
          'look!\n\n#blogs',
          '#digitaltransformation',
          '#cybersecurity',
          '#accounting',
          '#finance',
          '#healthcare',
          'https://t.co/jrgqeqdme4\n.\n#firstcoastcna',
          '#cnaexam',
          '#cnaexampreparation',
          '#jacksonville',
          '#cnatraining',
```

8) USE OF ILOC FUNCTION TO IDENTIFY TOP TRENDING HASHTAGS

```
In [56]: #Create a panda series to count the values in the list
         hashtags = pd.Series(tags).value_counts()
```

```
In [57]: #Display the first 30 records
         hashtags.iloc[:30]
```

```
In [57]: #Display the first 30 records
         hashtags.iloc[:30]
```

```
Out[57]: #healthcare                  716
         #health                       80
         #medicine                     41
         #ai                           40
         #job                          38
         #medical                      35
         #strategy                     30
         #pharmaceutical               28
         #digitalhealth                25
         #pharma                       25
         #marketing                    25
         #medtwitter                   24
         #biotech                      24
         #competitiveintelligence      24
         #meded                        23
         #vaccine                      18
         #hiring                       18
         #news                         17
         #machinelearning              17
         #technology                   17
         #coronavirus                  16
         #womeninmedicine              16
         #covid                        16
         #competitivemarketing         16
```

9) COUNT LARGER THAN 10

```
#Display records where count is larger then 10
filtered_df = data.query('count >= 10')


filtered_df
```

	word	count
0	#healthcare	716
1	#health	80
2	#medicine	41
3	#ai	40
4	#job	38
...	...	...
58	#saskatchewan	10
59	#nodejs	10
60	@ingliguori\n\n#python	10
61	#data	10
62	#healthcare!\n\ninfographic	10

COUNT EXCLUDING OVERPRESENTED HASHTAGS

```
#Display records where count is larger then 10 but less than 41
filtered_df1 = data.query('count >= 10 &count<41')


filtered_df1
```

In [132]:
```
#Display records where count is larger then 10 but less than 41
filtered_df1 = data.query('count >= 10 &count<41')


filtered_df1
```

Out[132]:

	word	count
3	#ai	40
4	#job	38
5	#medical	35
6	#strategy	30
7	#pharmaceutical	28