

Patient Service Analysis Tool User Manual

The following is a guide on how the system works, its features and how to use it. In this manual is the following:

- System Architecture diagram and explanation
- Explanation of the example data collection form
- Naming and data entry formats for the Excel files
- How to upload the Excel files to your Azure storage account
- Explanation of all the API endpoints
- Explanation of Power BI pages and visualisations
- Power BI features
- Price guide for operating costs

Before continuing to read this manual, make sure you have completed all the steps in the deployment manual first.

System Architecture

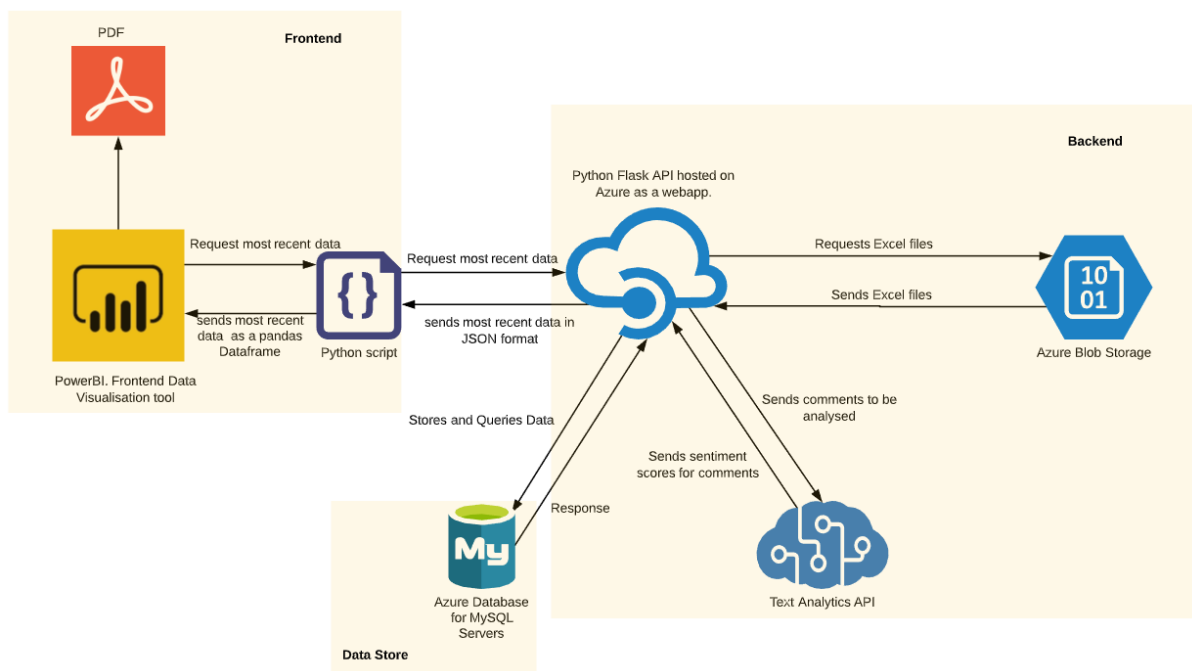


Figure 1: System Architecture Diagram

The key components of the system are the following:

- **Power BI:** the visualisation tool.
- **Python script:** Used by Power BI to make a request to the API and refresh its data .
- **MySQL database:** This is used to store the data that has already been analysed, including its sentiment scores. This is so that if we want to retrieve data that we've already analysed we don't have to analyse it again and can simply retrieve it from the database.
- **Text Analytics API:** Azure service used to carry out sentiment analysis.
- **Azure Blob Storage:** Azure storage resource where excel files containing data is uploaded to.
- **API:** Our API that uses all the above to process and analyse the data

Figure 1 is the system architecture diagram for the patient service analysis tool. For our frontend we used Power BI, but the system is not dependant or reliant on using Power BI so other visualisation tools can be used if needed. Power BI contains a python script that is run whenever Power BI wants to refresh the data. This python script makes a request to the API, from here one of two things can happen.

The first scenario is when the data being requested is already stored in the database. The API processes data one months' worth at a time. So, if the most recent years' worth of data is requested it will analyse the first months' worth of data, then the next and so on, each time building up a table of analysed data to be returned. Upon receiving a request, the API first checks to see if the data it needs is already in the database. If the data is already in the database then it is retrieved from the database and converted into a pandas dataframe. This dataframe is then returned by the API as JSON. The python script in Power BI then converts this JSON back into a dataframe, this can then be read by Power BI and the data is used to populate the visualisations.

The second scenario is when the data has not already been analysed before and isn't stored in the database. The API receives the request and checks the database to see if the data it needs is already stored there. If it's not then it goes to the Azure storage container where the excel files are uploaded to and retrieves the required excel files. These files are then converted to pandas dataframes by the API and are then checked for missing rows, invalid entries and general sanity checks. For each month there are 2 excel files, one for positive and one for negative data. The dataframes for positive and negative data are appended together and a new column is added to indicate if the row contains positive or negative data. The comments are then sent to the Text Analytics API for sentiment analysis and the scores for the comments are returned and added as a new column to the dataframe. All this data is then written to the database in case it's needed again in the future before being returned by the API as JSON.

The data collection form

In the folder called Data_Collection_Form_Example there is a word document and pdf which show an example of a form that can be used to collect patient feedback. The form is based off of the Friends and Family Test (FFT) used by the NHS. You are free to use these form templates to collect your patient data or variants of them. The data that is needed is how likely the patient is to recommend the service they received to someone needing similar services as well as a comment based on their likeliness to recommend the service. The likeliness must be one of the 4 following options: Extremely Likely, Likely, Unlikely, Extremely Unlikely. Extra options can't be added as the Power BI file used has visualisations for these 4 options only.

By collecting data using a form similar to this you should end up with data in the correct format to be used with the rest of the system.

Sample Excel files and Naming formats.

In the folder called Excel there are 2 excel files. These files show how data collected from feedback forms should be stored in the excel file. There are 3 columns: CLINIC, RESPONSE, COMMENT. Please do not modify the column titles. The CLINIIC column should contain the name of the clinic from which the data was collected from. Make sure the same clinic is spelt the same throughout as the system is case sensitive. Orthodontics and orthodontics would register as 2 different clinics. Similarly, for the Response column make sure entries are one of: Extremely Likely, Likely, Unlikely, Extremely Unlikely. Notice the capitalisation of the words, they are also case sensitive so in order for visualisations to

recognise them as valid data they must be capitalised correctly. No such restrictions are in place for the COMMENTS column.

The other important thing to note with the excel file is the naming format given to them. When creating new excel files to store other months data in, you must ensure they follow the naming format.

For excel files containing the months negative data the naming format is

Negative Comments - <Month> <YY>.xlsx

For excel files containing the months positive data the naming format is

Positive Comments - <Month> <YY>.xlsx

The names of the files are again case sensitive. <Month> should be replaced by the full name of the month the data in the excel file is for. <YY> should be the last 2 digits of the year for which the data is for, so for 2019 it would be 19. For example, the names of the files containing data for October 2021 would be:

Positive Comments - October 21.xlsx

Negative Comments - October 21.xlsx

It must have the .xlsx file extension. You can ensure it has this extension by using 'save as' and choosing the .xlsx file extension.

If these criteria are not met then the API will not recognise the file as a valid file containing data to be analysed.

Uploading data to Azure

Once you have collected your data and stored them on the excel files in the correct format with the correct naming format, you will then need to upload them to Azure. To do this go to your Azure portal homepage. From here you then need to go to your Azure storage account resource. Then click on 'Containers'. Open the container you made earlier and in the top left you'll see a button that says 'Upload'. Click upload, select the excel files you want to add and then upload them. The data is now ready to be analysed.

API endpoints: types of requests that can be made

The API has 3 main endpoints:

1. /psat/pastyear/
2. /psat/specificmonth?month=<mm>&year=<yy>/
3. /psat/mostrecentmonths/<int:no_of_months>

Endpoint 1

This endpoint will return data for the most recent 12 months. If less than 12 months of data is available (e.g. only 7 months' worth of data uploaded to Azure) then it will return however much data it can (7 months' worth in this example).

Endpoint 2

This endpoint will return data for a specific month and year. In the URL you must pass the month and year for which you want data for. <mm> should be replaced with the numerical representation

of the month with no leading 0s and <yy> should be replaced with the last 2 digits of the year. So, if I wanted data for only May 2023 I'd make a request to /psat/specificmonth?month=5&year=23.

Endpoint 3

This endpoint will return data for the most recent x number of months where x is number of months for which you want data for. You simply need to replace <int:no_of_months> with the number you want. For example, if I wanted to get data from Azure for the most recent 2 years' I'd make a request to /psat/mostrecentmonths/24.

For more information and examples of the API endpoints see the Requests.py file.

Loading data into Power BI and visualising it

In order to now visualise the data, you have uploaded to Azure, you will need to run the API on your localhost (or if you have publicly hosted it, make sure it is up and running). To run the API on localhost first open up a command prompt in the same directory where the project code is stored. Then enter your virtual environment by using .\venv\Scripts\activate.

Then type the following:

```
set FLASK_APP=application.py
```

press enter, then type:

```
flask run
```

Press enter and this should start up the API on your localhost on port 5000 and it is now ready to receive requests from Power BI and provide responses containing the data to be visualised.

Now open up the Power BI file. In the deployment manual you set it up so that hitting refresh makes a request for the most recent years' worth of data. All you have to do now is click on 'Refresh', This will run the python script to get the data, process it, and visualise it.

For data that has not been analysed before and is not stored in the database, allow 5 minutes per 1000 rows of data being analysed. This is due to the sentiment analysis and writing the data into the database taking time. Hitting refresh when data is already in the database is much faster as no processing is needed, and we simply retrieve the data from the database. Once it is done the visualisation should come up and you can analyse your data.

Make sure the query being run by Power BI to refresh data uses the correct URL. If you are using localhost then it would be something like 'http://127.0.0.1:5000/psat/pastyear'.

If you were instead hosting it on Azure it would look something like 'https://your-chosen-API-name.azurewebsites.net/psat/pastyear/'

For information on using localhost to run the API and configuring the Power BI query to retrieve data from localhost, please see the section titled *Adding python request script to Power BI with API running on localhost* in the Deployment manual.

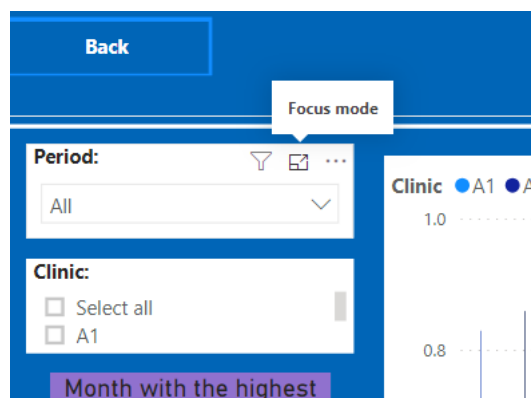
For information on hosting the API on a service such as Azure and configuring the Power BI query to retrieve data from the hosted API, please see the section titled *Adding python request script to Power BI with API running on Azure as WebApp* in the Deployment manual.

Power BI visualisations

This section provides details and tips on how to use the Power BI file provided, what all the visualisations represent and how they can help.

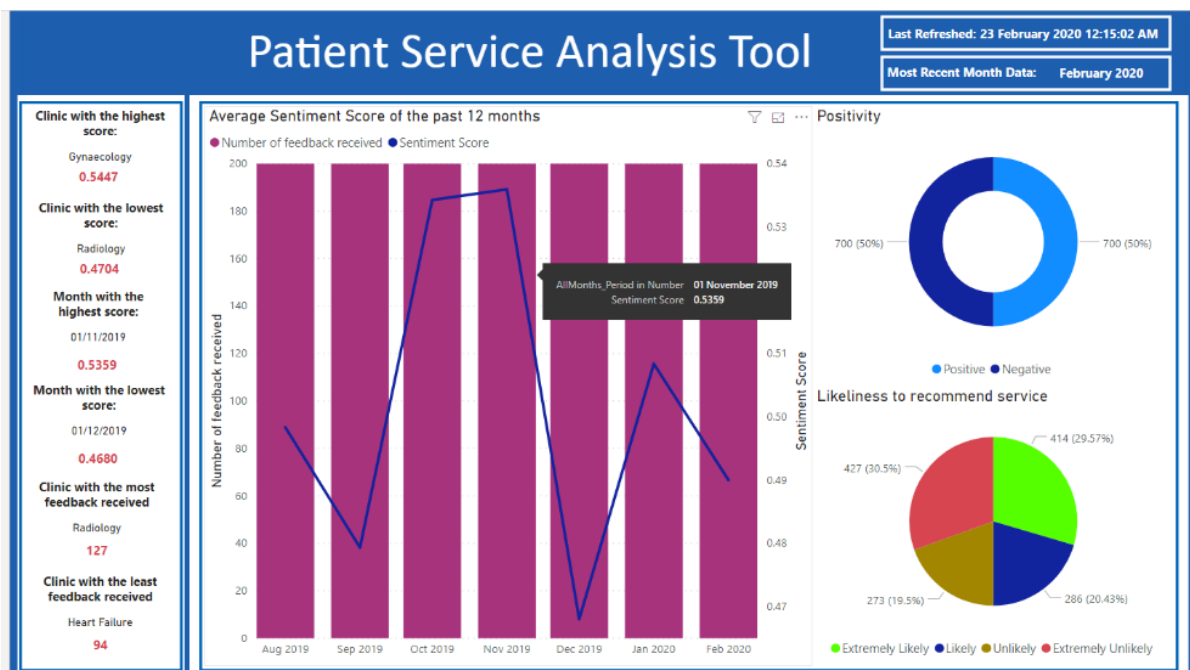
Filters and Focus Mode

Many of the visualisations offer filters to further narrow down the data you are visualising. The 4 main filters used are period, clinic, positivity and response. Period allows you to filter the data you are viewing by month so you can analyse data for a specific time range. Clinic is to filter what departments and clinics you wish to see data for. Positivity is to filter whether you want to visualise only positive data or only negative data. Response allows you to filter by response type, for example you could filter the data to only show entries that were Likely to recommend the service.



All of the visualisations and filter selectors support what's called focus mode. Focus mode is one of the options on the top right when you hover over a visualisation or filter. By entering this mode, you can more easily select items from drop down lists and view visualisations in full screen. When selecting multiple values from a dropdown list you will need to ctrl click the options.

Home page



The home page provides a high-level view of how the hospital is doing. The main feature of this page is the time-sentiment graph with volume of responses overlaid on top. The purple bars represent the volume of data received for that month. The line going across the visualisation is the average

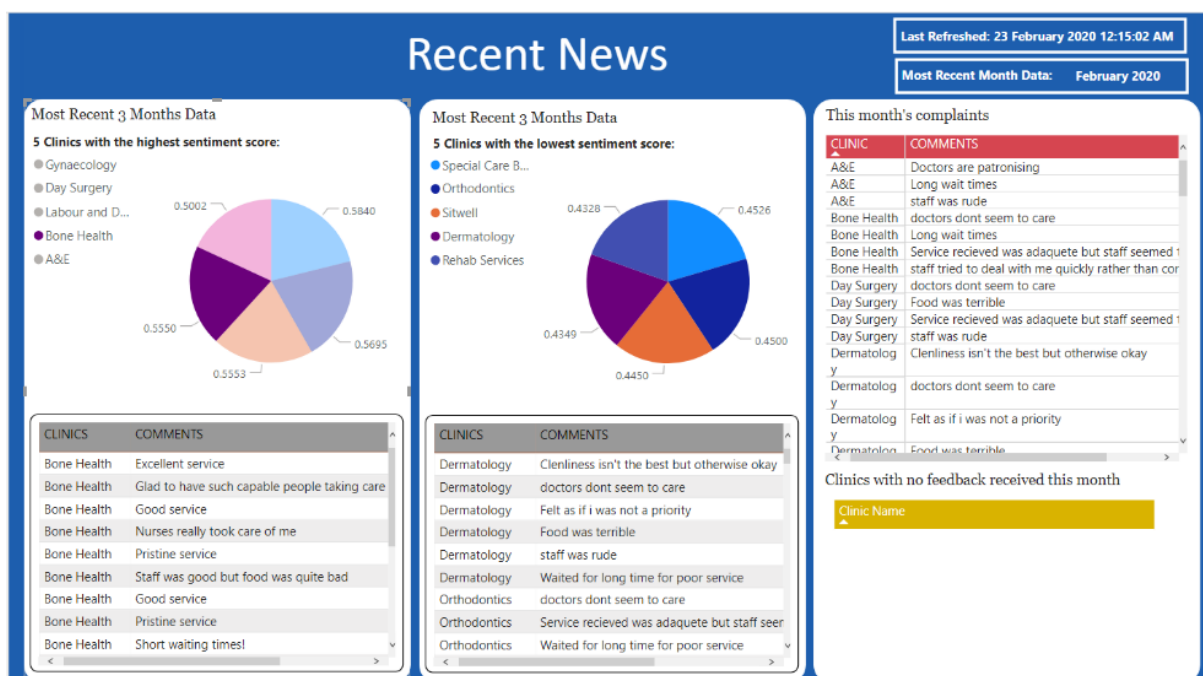
sentiment score for that month. This allows hospitals to see how sentiment has changed over time and lets them know if they are improving their services or not.

On the top right there is a doughnut chart which breaks down the proportion of positive to negative responses. The bottom right is a pie chart that breaks down the proportion of response types received. Down the left are some key facts that may be commonly asked such as the best performing clinic and best performing month.

By clicking on bars on the time-sentiment graph you are able to filter all the data on this page to the selected months. So, if you were to select October, November and December, all the visualisations and key facts would change to only analyse data from the last quarter of the year.

There are also 2 boxes at the very top right of the page. One of them shows the latest month for which data is available for and the other shows you the last time the data was refreshed.

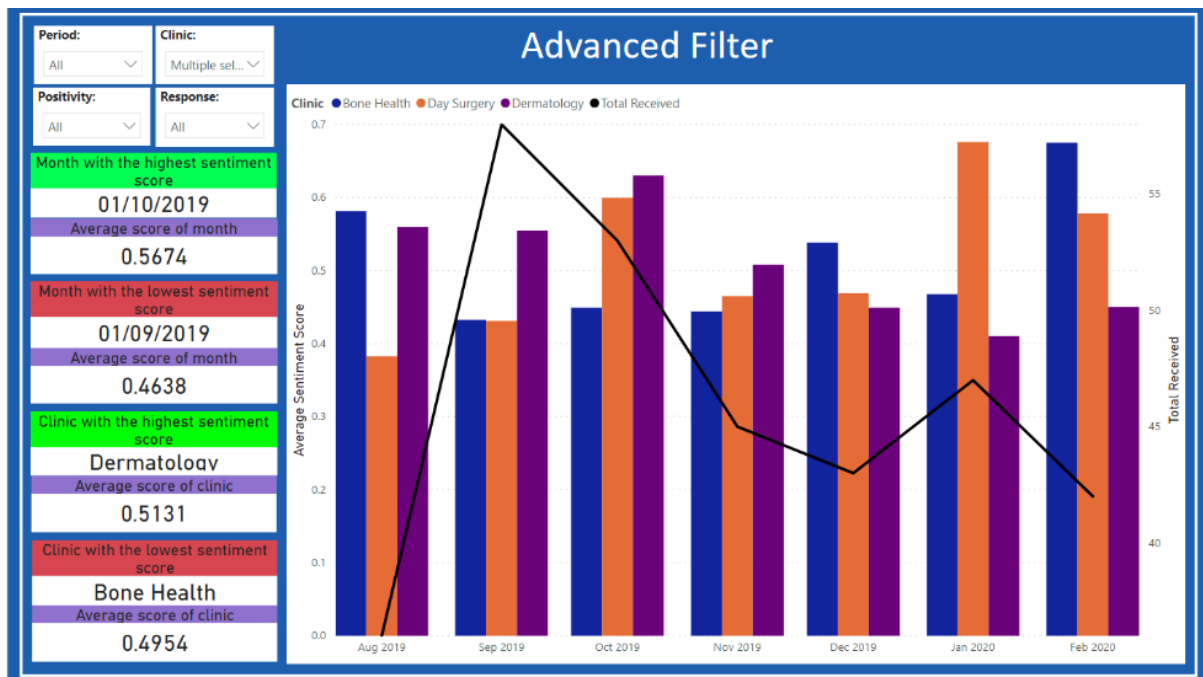
Recent News Page



The recent news page shows you information about data for the past 3 months. The left section shows you data on the top 5 best performing clinics. By clicking on a section of the pie chart you will filter the table below it to show comments received from that clinic as well as their sentiment scores. Similarly, the middle section shows you data on the 5 worst performing clinics and has the same filtering capability. This allows you to see what parts of the hospital are exceeding expectations and doing well and also where improvements need to be made.

The right section shows you a list of all the complaints received in the past month and also shows you all the clinics for which no data was received. This shows you which departments need to improve their methods of patient feedback collection and for what departments the rest of the data does not consider due to not having any data on them.

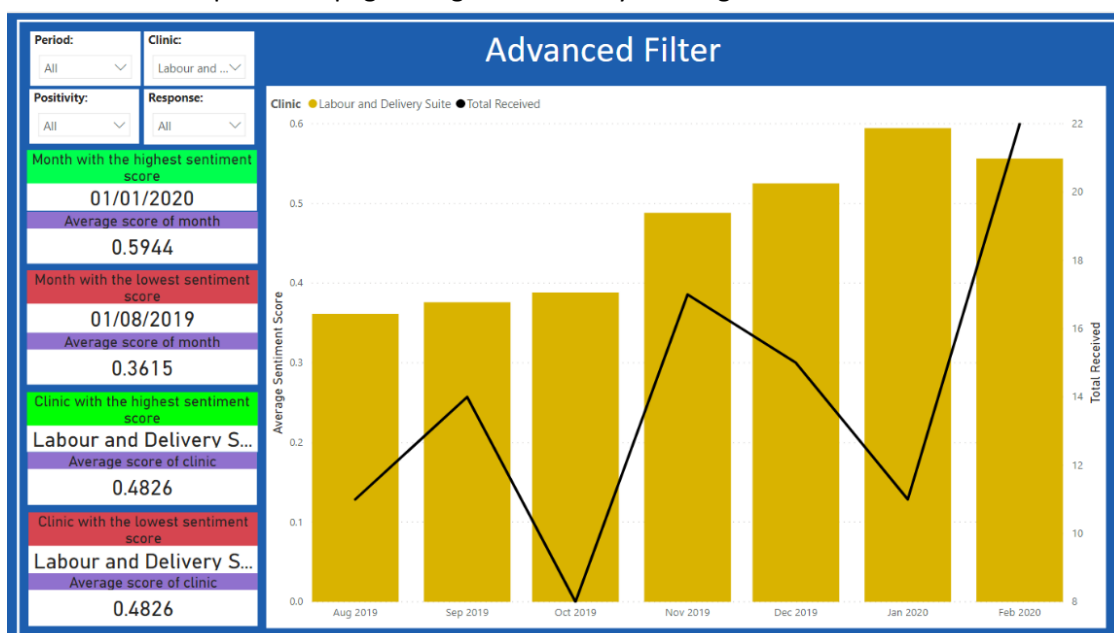
Advanced Filter Page



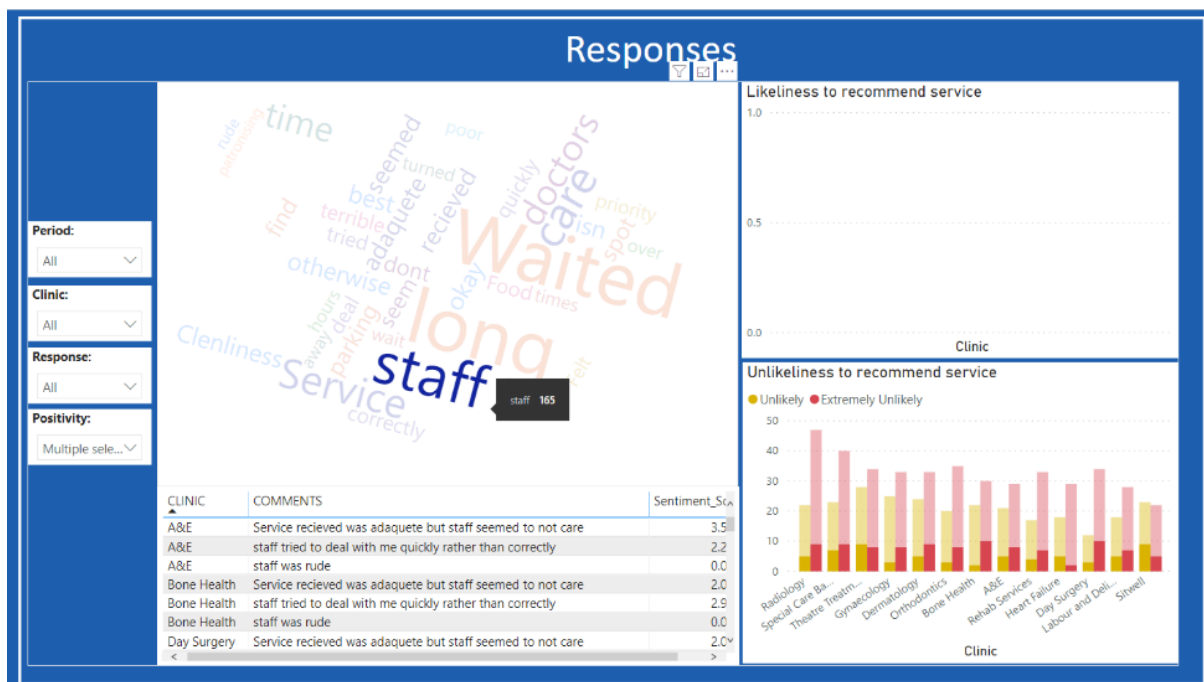
This page is used to see how a specific clinic has been performing or to compare clinics against each other. In the above picture we can see the page being used for the latter. Here we are comparing 3 departments against one another. The bars are colour coded with the key being at the top. The height of the bars represent the average sentiment score for that month from that clinic based on whatever filters have been applied. Here only clinics have been filtered. The black line represents the number of responses received. This allows you to check if the results you are seeing are based off enough data to be valid.

Down the left is some key information that answers some common questions that may be asked. This includes best and worst month and best and worst clinic.

Below is an example of the page being used to analyse a single clinic.



Responses Page



This page allows you to see clinics with the most positive and most negative response types. On the right side on the page there are 2 visualisations. The top left shows the clinics with the most Likely and Extremely Likely responses. Conversely, the bottom right shows the clinics with the most Unlikely and Extremely Unlikely responses. The visualisations shows each clinic with 2 bars, one for each response type.

The left side of the page features a word cloud that allows you to see commonly said key words in patient comments. If you click on one of the words it will show you the number of times that word came up. It also filters the list of comments below it to show you all of the comments that include the selected word. This allows you to find common issues that people are having then look at individual cases of the issue in the comments to see if you can fix it. Clicking on a word also modifies the visualisations on the right to show the proportions of comments from those departments that include the selected word.

This page also has the 4 filters to the left. These filters will apply to all visualisations on the page.

Pricing guide

Below is information on the prices and costs of all parts needed for the system to operate.

The database on Azure can vary in cost depending on what pricing tier you choose. Use [this link](#) to work out how much you would be spending per month on the database.

The Text Analytics API again depends on how much data you plan to analyse each month. If its less than 5000 records a month then choose the free tier and there should be no costs. Else you will have to use the Standard tier which is pay as you go. On the Standard tier, analysis of 1000 records will cost £1.49. [This link](#) provides more information on costs for this API.

Storing the excel files on Azure also incurs costs. Per month it costs £0.0144 per GB of data stored on the cloud. It also costs £0.0036 per 10,000 read operations.

As an example, say you had a database on the Basic tier with 2 vCores, you were analysing 2000 records per month and each month the files containing that data was 150 KiB.

Database - £41.459

Text Analytics API – Free

Cloud storage – < £1

Another example but with more data. Say you had a General-Purpose database on azure with 2 vCores, were analysing 20,000 records per month and each month the files containing the data were 2GB in size.

Database - £110.557

Text Analytics - £30

Cloud storage - < £1