



Software for Digital Innovation Module

Coronavirus And Stop and Search Data
Analysis Using Python GUI

By

SALISU ABDULLAHI

C2224671

Table of Contents

Introduction	3
Software Tools and Libraries.....	3
Limitations.....	4
Real World Application in Fintech	4
Appendix: Black Box Testing	6
References.....	9

Introduction

This report is designed to examine the software tools and libraries used in the analysis of the coronavirus and stop and search information of the police. The analysis focused on two Data sets; firstly, the coronavirus infections in England, specifically focusing on two low-tier-local-authorities (LTLA) namely Hartlepool and Stockton-on-Tees; and the second part focused on stop and search information sourced from the Cleveland Police Department.

Software Tools and Libraries

To do this analysis, software tools and libraries were used. These include Python programme used to source and analyse the data and a Graphical User Interface (GUI) which allows the user to interact with the software programme to access the visualisation of the analysis done. Performing the analysis using Python, the Jupyter Notebook IDLE was the environment used to write the code programme used for analysis, as well as the code used to create the GUI. Writing the code for analysis, software libraries were accessed to perform different function and tasks in the programme. Libraries accessed include the following:

- **Pandas Library** – This library was accessed to enable me to read the data into python and assisted in viewing the data, cleaning, and slicing the data which allowed me to get only the data that I required for my analysis without writing many codes. The library comes with different functionalities which were used such as 'groupby ()', 'diff ()', 'reset_index ()' amongst others which were used in the processing and cleaning on the data.
- **Matplotlib Library** – this library was accessed and used in creating visualisations of the analysis done. Creation of Charts and Graphs were seamlessly done by the library with the library allowing for a great level of customization such as setting the colour, creating axis limits, labelling of axis and so many more. The matplotlib library was an important library for this project as its functionalities were used in integrating the analysis done into the GUI.

- **Numpy Library** – this library was mainly accessed for use during visualisation as it was used to arrange the ticks of the graph axis. This library can be used to support multi-dimensional arrays and perform mathematical functions of high-level.
- **Requests Library** – This library was the most important library in the stop and search analysis as it allowed me to source the stop and search data from the internet through the stop and search API.
- **JSON Library** – The Json library was particularly useful during the stop and search data analysis. This is because upon sourcing the data from the API by the request library, JSON converts the data into human readable format.
- **Tkinter** – this was the toolkit I used to apply my programming code into a GUI for user interface.
- **Python Imaging Library** – this provided support for opening and inserting images into our program. This was the library I used to insert the respective logos in my interface.

Limitations

With regards to the IDE used, the Jupyter notebook gave errors or hangs at intervals while writing and testing the code which prompted restarting and running the cell again to correct. Also, while writing and testing the code, I noticed that Jupyter notebook does not possess code-style correction. Also, it takes longer to test long written codes. Some pandas' library functions were also difficult to understand and required some external reading for me to be able to properly use. The data fetched from the API were in bits (monthly) which allowed for rewriting of code to get data for each month instead of fetching data for a period (from April to June). Also, JSON used is not as extensive as XML which allows for a larger data and verifies correctness of data.

Real World Application in Fintech

In emerging financial technologies, the use of software tools and libraries is increasingly becoming important as companies use them to speed up analysis of data. Financial services use Python for activities such as pricing, risk assessment, Data automation and analytics, and so much more. One of the reasons it is being used more than other programming

languages is because of its easier syntax which makes it faster to program when compared to other languages such as C++ (Smyrnova, T. 2022). Python also contains many libraries that are relevant to fintech companies such as NumPy, Pandas, SciPy, Quandl, PyFolio (Singh, T., *no date*). Other libraries relevant to financial markets include finmarkeppy, scikit-learn, pybitcointools, pynance and so many more (Smyrnova, T. 2022). Considering this project and the libraries applied to it, it is worthy to note that while they can complete the work, they have some limitations. One of the limitations involves the safety of the Jupyter Notebook. While Python programme is one of the top used languages by fintech, developments decisions are influenced by factors such as safety (Luiza, 2020) which Jupyter Notebook to an extent has a lower security when compared with other IDEs such as Python IDLE, PyCharm and Atom IDLE. Recent article by Gupta, A. (2022) ranks Jupyter at number 6 amongst the top 10 integrated development environment. Worthy of note is that while it ranks 6th, one of Jupyter notebook main features includes the intergeneration of data science libraries allow for easy data analysis. Also, the request tool used to get the data from the stop and search API cannot deal large sizes of data and as a result is not mostly used in situations where the company deals with a large size of data. Another limitation to the tools used is the limitations of the Tkinter which has a limited number of widgets and has an older design compared to other python GUI such as PyQt (*Python GUI, PyQt vs Tkinter Online*, 2020).

Appendix: Black Box Testing

Note: In some cases, please give interface few extra seconds to run outcome successfully

Testing the app for visualisations on Corona Virus data, the below black box tests provide a guide into expected outcomes

Action	Expected Outcome	Actual Outcome	Remark
Click Run on Jupyter Notebook	Application Launches to landing page asking user to select data to analyse	Application Launches to landing page asking user to select data to analyse	Okay
Click on 'Coronavirus'	Opens window showing coronavirus analysis visualization options	Opens window showing coronavirus analysis visualization options	Okay
If the first radio button is selected	Opens window visualising the Daily Infection Chart	Opens window visualising the Daily Infection Chart	Okay
If the second radio button is selected	Opens window visualising the Weekly Infection Chart	Opens window visualising the Weekly Infection Chart	Okay
If the third radio button is selected	Opens window visualising the Monthly Infection Chart	Opens window visualising the Monthly Infection Chart	Okay
If the fourth radio button is selected	Opens window visualising the	Opens window visualising the	Okay

	Comparing infection charts	Comparing infection charts	
Click on 'Back' button	Returns user to coronavirus analysis visualization window	Returns user to coronavirus analysis visualization window	Okay
Click on 'Back to Homepage' button	Returns user to landing page window	Returns user to landing page window	Okay
Click on 'Quit' button	Closes the window	Closes the window	okay

Testing the app for visualisations on Cleveland Police stop and search data, the below black box tests provide a guide into expected outcomes

Action	Expected Outcome	Actual Outcome	Remark
Click Run on Jupyter Notebook	Application Launches to landing page asking user to select data to analyse	Application Launches to landing page asking user to select data to analyse	Okay
Click on 'Stop and Search'	Opens window showing Cleveland police stop and search analysis visualization options	Opens window showing Cleveland police stop and search analysis visualization options	Okay
If the first radio button is selected	Opens window visualising the Outcome chart	Opens window visualising the Outcome chart	Okay

If the second radio button is selected	Opens window visualising the Gender chart	Opens window visualising the Gender chart	Okay
If the third radio button is selected	Opens window visualising the Age Count Chart	Opens window visualising the Age Count Chart	Okay
If the fourth radio button is selected	Opens window visualising the Objects searched charts	Opens window visualising the Objects searched charts	Okay
If the fifth radio button is selected	Opens window visualising the age comparison chart	Opens window visualising the age comparison chart	Okay
Click on 'Back' button	Returns user to coronavirus analysis visualization window	Returns user to coronavirus analysis visualization window	Okay
Click on 'Back to Homepage' button	Returns user to landing page window	Returns user to landing page window	Okay
Click on 'Quit' button	Closes window	Window closed	Okay

References

Gupta, A. (2022) *Top 10 Python IDEs in 2023: Choosing the best one*. Available at: <https://www.simplilearn.com/tutorials/python-tutorial/python-ide> (Accessed 03 January 2023)

Luiza (2020) *Why Successful Fintech Apps are Built with Python*. Available at: <https://www.merixstudio.com/blog/why-successful-fintech-apps-are-built-python/> (Accessed 03 January 2023)

Python GUI, PyQt vs Tkinter Online (2020). Available at: <https://www.geeksforgeeks.org/python-gui-pyqt-vs-tkinter/> (Accessed 03 January 2023)

Singh, T. (No date) *Python in Finance: How Python is powering the Fintech Universe*. Available at: <https://www.trio.dev/blog/python-in-finance-how-python-is-powering-the-fintech-universe> (Accessed 03 January 2023)

Smyrnova, T. (2022) *Best Python Libraries for Fintech* <https://upplabs.com/blog/best-python-libraries-for-fintech-development/>