Deliverables

Process/ Project Management

Use revision control system

We used Github as our version control repository. After any major updates to the code, the updated codes are pushed to the Github repository. To assist with that, Git Bash program is being used.

Documented end-to-end

All the functions in the codes have been documented to show their functionalities, type of input required, type of output expected etc. It's been further explained in detail in the methodology section of the project report.

Continuous Integration

The code is updated twice or thrice every week. It depends on the how much work there has been done

Backup Procedures

We took backup quite seriously. First it was constantly uploaded in Github. Second it was also upload in Google drive. Third, we have 3 laptops in our team for every individual user where the workings are stored in all laptops. Codes have also been occasionally stored in USB drives.

Coding

Code Structures

The code using to build a database for the web app has been divided into 5 section which are given as follows:

1) Fitting Dataset
2) Clean Reviews
3) AS Sentiment Analysis
4) Build Database
5) Web Integration

Coding Readability

The functions are coded in python which is quite easy to understand. The variable names for all the functions are meaningful and the user can understand the purpose of that variable.

Deliverables

Coding Comments

All the functions in both Python and Java files have been thoroughly documented using both in-line comments, explaining purposes of individual lines, and block comments which explain the purpose of a block of code of a function.

Coding Maintainability

The folder is structured in an appropriate order. All the files are named meaningfully. Only the files which are necessary are present. Each folder name implies its purpose.

Coding Robustness

Most of the function that we created had an worst case time completely of O(N). Complexity has been improved wherever possible including some algorithm and data structure decisions. This makes the system quite efficient.

Documentation

A readme.md has been created in the GIthub repository explaining the user on how to install the web app.

Modularity

Ease of adding features

The web app is made using MVC framework which is one of the easiest web development frameworks to work with. On top of that the backend is built in Python. So adding addition feature will be quite easy. Jupyter Notebook was installed using Conda which comes prepackaged with a lot of libraries which can be used to further extend, enhance and expand the project in the future.

Ease of adding a new data set will vary according to the database of the web app. If the new dataset has all the characteristics of the old dataset, then it will merge seamlessly.

Scalability of software

Visitors
The current web app is hosted on a cloud application platform called Heroku. As this is a free version, too many visitor will cause the usage limit to run out. However the web app has been created using a Python framework called Flask which can handle large applications. Based on that if we host the web app in a good server, it will be able to handle a decent amount of visitors. A key point to be noted here is that the current state of the software is

that of a prototype and that is very much suited to the current frameworks being used. However, the core logic of the program is independent of that of the deployment toolkit and therefore in future, if a different application other than Heroku was to be used to deploy the software, it would be relatively easy.

Limitation of software

Database
For now we are using Python dictionaries as database for the web app but the data inside is of small size. There might be a memory issue if the dataset were to be large enough. This issues can be solved by using a SQL server as the database instead.
At the moment, the software only works with English language. The website is also only optimized for desktop and doesn't work well with mobile phones.
The data set on which our project is based on had some limitations which manifested in our project. Due to lack of punctuation and punctuation restoration not being 100% accurate, aspect extraction may not be as accurate.

Degree of portability

Project is coded in languages which are available across all the operating systems. None of the modules and libraries used are exclusive to a single OS so the web app is platform independent. Application has been tried in all three major operating systems (Windows, Mac, Linux) and it has worked without any issues.

Usability and UX

Aesthetic

We used a lightweight CSS stylesheet framework called Materialize. This way we maintained a user friendly and minimalistic design. With hotels and tourism being our domain, some graphics were used in animation which implies the nature of the web app to an uninformed user.

Ease of use

We made sure that our web app produces output that is not confusing for the users and at the same time does not have a clustered design. User needs to only use a drop-down menu for location selection and interact with a simple text field to input the aspect. The map shown is zoomable by means of mouse wheel and shows the areas with great detail. The top hotels are labelled clearly and a conveniently located Reviews button will immediately yield the user the top reviews for their aspect. For each hotel, the relevant information including hotel's name, hotel's address, city name are shown clearly.