Project Title: Data Analytics Platform

Project Detail Document

Version: 1

Group No. : 50

TA Name: Anshul Bhaware

Date: 25 March 2022

Group Members

Ridham Sonani	201901001
Bhargav Patel	201901037
Nirav Parmar	201901132
Krunal Lukhi	201901449
Mitul Vaghela	201901403
Raj Patel	201901306

Objectives of the Project:

The major objectives of the project are as follows:

- The data analytics platform should be able to visualize data, perform analysis on data that helps the user to retrieve, combine, interact with, explore, and visualize data from the dataset.
    - The major aim of the platform is to provide freedom to the user to analyze his/her dataset first by uploading the dataset followed by choosing the various data fields of interest he/she would like to visualize in the form of graphs.
    - The platform should also help the user with the analysis by providing different types of visualization options.
- The user upon providing a dataset to the platform should be able to get various relations between the different features of the dataset.
    - The relations between the data fields should be in an easy-to-understand format, for instance, in certain cases, a pie chart may provide better insights than a bar chart. With this fact in mind, it is essential that the platform should also provide freedom to the user about the choice of chart/graph he/she would like to add to the dashboard.
- The user should be provided with a visualization of the provided data and also at times predict the values for different features based on the relationship among the features.
    - The predicted model should also be displayed in an easy-to-understand format and hence if possible should be able to give a graphical representation of the prediction.
    - The prediction model in the initial versions should at least be able to give the user an understanding of the general trend (increasing/decreasing relation) among the chosen fields, i.e. the prediction model will be based on linear regression.
- The functions provided by the platform should provide the user with a good understanding of the data and will help the user to make informed conclusions.

Functionalities:

The major functionalities provided by the platform will be as follows:

- The user should be able to register and thereafter login to the platform.
    - A new user should first be asked to register himself/herself on the platform. This will enable the platform to keep a track of the datasets the user has uploaded and will be working on.
    - A registered user should be asked to log in at the start of a new session. This will help the platform identify the user and his/her datasets. Also, this will also enable the platform to display the work saved by the user in his/her previous session.
- The user should be able to upload the dataset file on the platform.
    - This is an essential feature as without a dataset the platform will not be able to provide visualizations and predictions to the user.
    - This functionality should only be provided to the registered users as without user information, the dataset should not be displayed. This is to ensure that only the user that uploaded the dataset should be able to view and perform analysis on it.
- The user should be able to obtain a graphical representation of the data uploaded.
    - The major aim of the platform is to help the user understand his/her dataset better and the best way to do that is by providing graphical representations of the data fields.
- The graphs on the platform should provide the user an option to select the attributes among which the graph will be displayed.
    - This functionality will allow the user to perform an analysis of data according to his/her will. The user, according to his/her requirements can choose the data fields relevant to his/her goals and work only on those particular data fields.
- The user should have access to add, delete or update the graphs for a dataset.
    - The ability to add, delete or update the graphs will add another layer to the freedom provided by the platform to the user in carrying out analysis.
    - The user can add as many graphs as he/she desires, delete some irrelevant graphs and also make changes in the existing graphs like changing the data fields among which the graph is plotted.
    - This feature will come in handy when the user’s preferences of the data fields he/she wishes to work upon will change. To elaborate further, let’s say for a certain period of time the user would like to work only with certain data fields of

  - the dataset, so he/she adds graphs related to those fields. After a certain period of time, the user feels the need to change the data fields he/she wants to analyze and so the functionality to add, delete and update the graphs will provide the user the required tools to carry out the analysis without the need to upload the dataset again.
- The platform should save the changes made by the user.
  - Since the no. of data fields can be large and based on the preference of the user, the ability to save the graphs would help the user to have the option of reviewing the trends in the dataset for later.
- The platform should also display the uploaded data in a tabular form.
  - Along with the graphical representation, the user should be able to view the data in a tabular form so that he/she doesn’t have to remember the data field names and this also helps the user to look for the overall information about the dataset.
  - In the tabular representation, the platform should enable the user to make general queries about the data such as edit a particular cell, sort with respect to a particular data field, search in a particular data field, apply filters to the dataset. The user should also be able to choose which data fields to view and also download the modified table for further use.
- The platform should provide the user to visualize the data in the form of different types of charts.
  - The different types of charts can be a bar chart, pie-chart, line-chart, scatter chart, heat-map, radar chart, and others.
- The platform should also have a provision for its users to upgrade to the premium tier. The premium tier will have additional features like providing a classification of data fields, increased dataset file size limit(to upload on the platform), and developer support.
- The platform should be able to provide the functionality to predict values taking the dataset as input.
  - For non-premium users, the prediction model available will be linear regression and premium users will be provided with advanced models like polynomial regression.
- The platform should provide various statistics like mean, variance, standard deviation of the various data columns.

Project Deliverables

Milestones

- From the date of allotment of the project till this week, we have had several group discussions about the topic and have done research on the functionalities and features that a data analytics platform provides. This research has helped us to get a general idea of the project topic and enabled us to define the tasks to be undertaken.
- In the next week, the milestone we wish to achieve is to create a DFD diagram of the project topic and will carry out other design-related tasks.
- In further weeks (Once the design is complete), we will build the feature of sign-in and sign-up. Also, we would be providing the user with the feature of uploading the datasets.
- For our next milestone, we will be implementing all functionalities related to graph plotting.
- The next milestone would be implementing features that provide statistics related to the data and the feature of model prediction using linear regression.
- The next milestone would be the implementation of the premium tier where users have the ability to upload large datasets and have an additional feature of classification of data fields.

Estimated Total Time

- Based on the milestones stated above, we estimate that the total time required should be approximately 75-80 hours.
- The first 30-35 hours are estimated to be spent on design purposes.
- The remaining time is to be spent on the implementation of the design. We estimate that it would take 5-6 hours to complete the implementation of a feature. This estimate can again be changed depending on the problems faced during the implementation.

H/W and S/W requirements

- **Hardware requirements:** Apart from a working PC, there is no such specific hardware requirement as it is a web application.

- **Software requirements:** The only software requirement is that the system should be an OS-based system and should have a browser that is compatible with the execution environment.

Technology/Architecture:

The initial versions of the platform will be built using the following technologies:

- Web-based application built using React JS at the front-end and Node JS at the back-end.
    - The front end will be built using React JS. The major user interface components will be based on material UI and apexCharts and the front-end will have the need to utilize the local storage of the browser.
    - The back-end will be built using Node JS. The backend will communicate with the database and as per the user requests from the front-end, it will fetch the required data. In order to fetch the datasets, the backend will require user authentication.
- Database for backend will be stored on MongoDB cloud and Firebase
    - MongoDB will be utilized to store the user information and the information regarding the datasets uploaded on the platform.
    - Firebase will store the actual dataset file uploaded by the user in csv format. It will also provide a URL for the dataset file which will, in turn, be stored along with the dataset information on the MongoDB cloud. Hence, when required the backend upon accessing the dataset link will be able to get the dataset file.
- Code development will be done using VS Code, Postman, Git, GitHub
    - The development of the code for the front and back end will be done on VS Code.
    - Postman will be used for testing the server requests
    - Git and GitHub will be used to keep a track of various features added during a time period. The Git repository will also contain the code and other documentation related to the platform.

Standard to be Followed:

- The implementation code should be divided into modules and sub-modules so that it becomes easy to navigate through the code. Hence the code will be divided based on react components.
- The variables/ functions used in the code should be named appropriately to enhance code readability.
- The project should be supported by relevant documentation.