

Data Visualization Software Using JavaFX

Naga Venkatesh Gavini, Milad Tahmasebi, Rohan Mishra
Software Engineering Systems, Northeastern University
{gavini.n, tahmasebi.m, mishra.roh}@northeastern.edu

Abstract - The saying "a picture is worth a thousand words" accurately captures the aim of data visualization. Data visualization aims to communicate information in a clear and straightforward manner. Visualizations can highlight crucial details and offer the user practical information, such as data summaries. We built software that can be helpful for people who work with a huge amount of data. Basically, we created a data visualization tool to understand the data trends and patterns. This tool is going to help data scientists who can concentrate on building models instead of spending time on fixing the data issues with respect to visualizations. This software can also be useful for people who don't code much and are interested in seeing the data trends. They can easily use this software and visualize the data with the help of graphs, scatter plots, and many more.

Keywords— *data visualization software, javaFX, data analysis, java, data analysis*

I. PROBLEM DESCRIPTION

We are living in an era when every engineering and scientific field is filled with big data gathered by measuring equipment. These datasets can be used for building models and new designs. First step of using a dataset is preprocessing to get familiar with and understand the obvious trends and patterns in dataset. Visualization is the best tool to evaluate and better understand the data. The other importance of visualization is representing the output of models and designs. For example, Figure 1 illustrates the output for a study conducted at Huntington Ave, Boston, MA for measured delay for pedestrians at each intersection and all the networks.

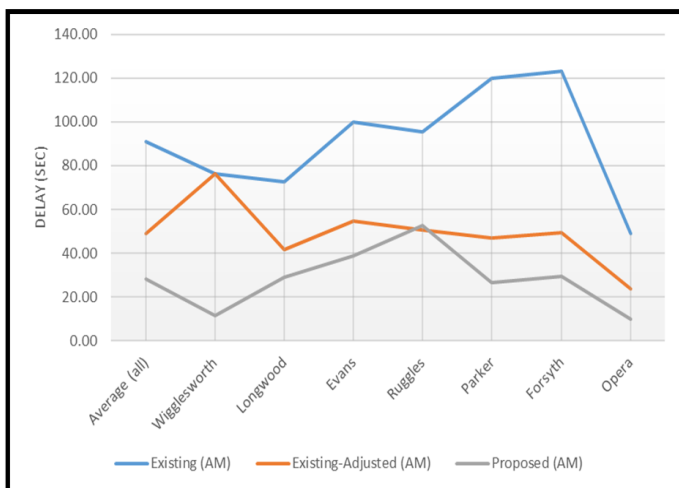


Figure 1- Pedestrian delay at Huntington Ave

There are various types of dataset storage format that are being used by scientists and engineers which listed below:

1. CSV
2. JSON
3. Avro
4. Protocol Buffers

CSV format is one of the most popular data storage formats and is chosen to be used in our application as the data source.

The target market for proposed application consists of two groups of people. First group is engineers in industry which might not have enough knowledge of coding to use programming languages or might not have access to applications like Microsoft office to create graphs from dataset. The second group is data scientists which they might need to save some time and concentrate more on design and building models.

Our solution for this issue was to develop an application to be able to import and read the dataset as a CSV file and provide options for the user to choose which variable and graph they want to have.

II. ANALYSIS (RELATED WORK)

We have tried to use the Tablesaw library to render the graphs to the UI, Unfortunately, this isn't as easy to do as it should be. The JavaFx WebView component is completely fine for the very simple pages, but it is too fragile to handle the JavaScript and CSS-heavy pages for us to support. There are a few other browser components available commercially or as open source if you need to use Tablesaw's visualization tools, but you may be better off with JavaFx Charts. After careful consideration, we have decided to use the JavaFX charts which is a pure java library to render the plots.

JavaFX charts supports the following graphs out of the box

- Pie Chart
- Line Chart
- Area Chart
- Bubble Chart
- Scatter Chart
- Bar Chart

III. SYSTEM DESIGN

The structure of our application is straight forward. Which can be summarized in 5 general steps. It starts from importing the file and will end up with different types of graphs. Figure 2 represents these steps.

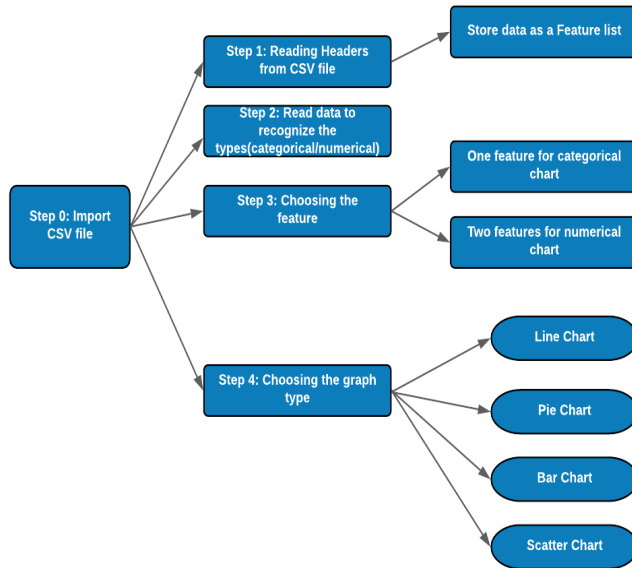


Figure 2 - Design structure of data visualizer app

IV. IMPLEMENTATION

In the final implementation, we have included following plots to implement

1. Line Chart: A line chart, also referred to as a line graph or a line plot, connects a series of data points using a line. This chart type presents sequential values to help you identify trends. Most of the time, the x-axis (horizontal axis) represents a sequential progression of values.

2. Pie Chart: A pie chart is a graphical representation technique that displays data in a circular-shaped graph. It is a composite static chart that works best with a few variables. Pie charts are often used to represent sample data—with data points belonging to a combination of different categories.

3. Bar Chart: Bar charts enable us to compare numerical values like integers and percentages. They use the length of each bar to represent the value of each variable. For example, bar charts show variations in categories or subcategories scaling width or height across simple, spaced bars, or rectangles.

4. Scatter Chart: A scatter chart (aka scatter plot, scatter graph) uses dots to represent values for two different numeric variables. The position of each dot on the horizontal and vertical axis indicates values for an individual data point. Scatter plots are used to observe relationships between variables.

Examples of all the charts explained above are shown in Figure 3.

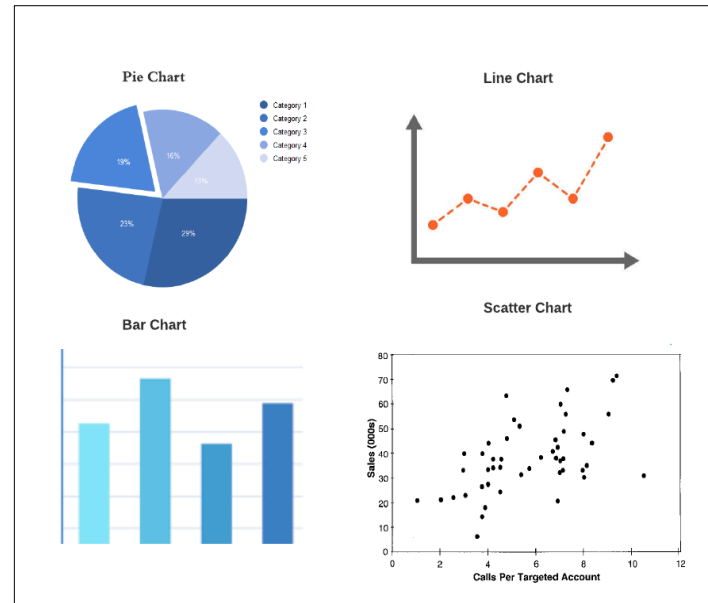


Figure 3. Example Graphs

UML diagram along with the class definitions and their attributes is shown in Figure 4.

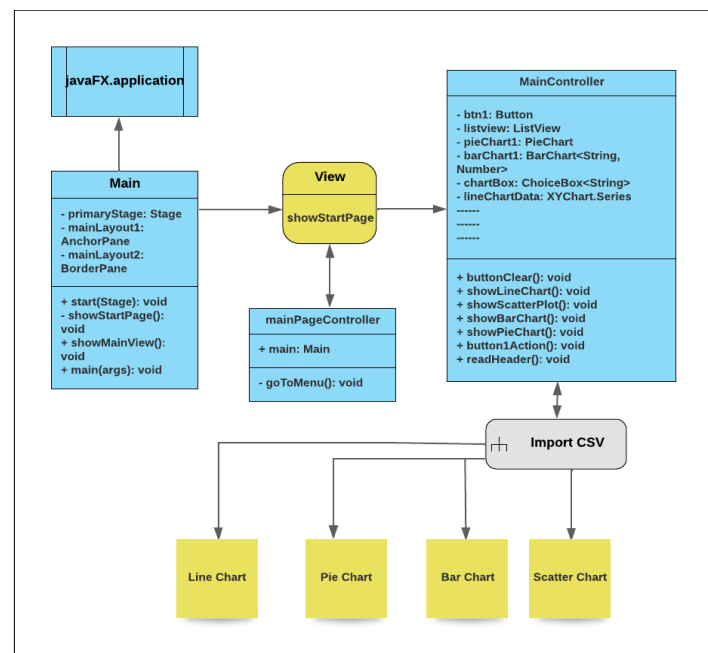


Figure 4. UML Class Diagram

We have implemented the following topics while implementing this project.

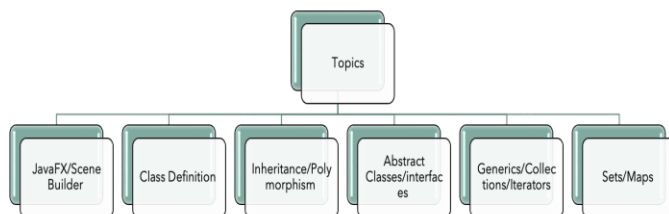


Figure 5. Topics covered

As shown in the class diagram Figure 4. The flow of the application starts with the landing page which will have a “Get Started” button. The main Controller class will load all the sections of the core application upon clicking the “Get Started” button. The Main Controller contains the Graphs section, Variable selection, List of features and upload file section. The user must first import the CSV file and start selecting the variables based on the interest.

V. EVALUATION

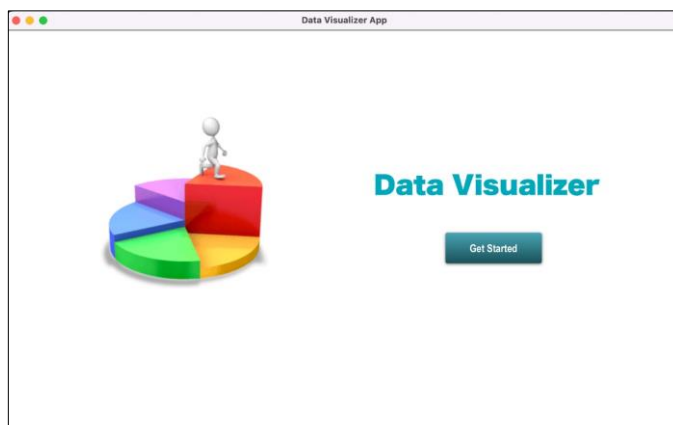
The following screenshots show the complete life cycle of the application. The main application page is the second page where user can choose the csv file based on his/her choice and select the X or Y variables for the data analysis.

Choose X variable to view the following charts

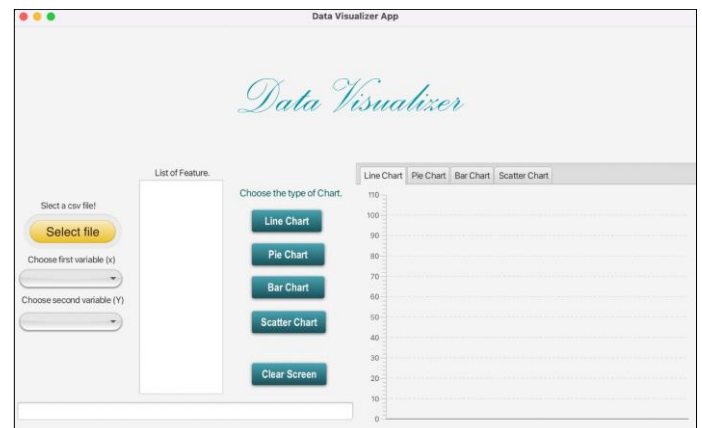
- Line Chart
- Pie Chart
- Bar Chart

Choose X and Y variables to view the Scatter Chart

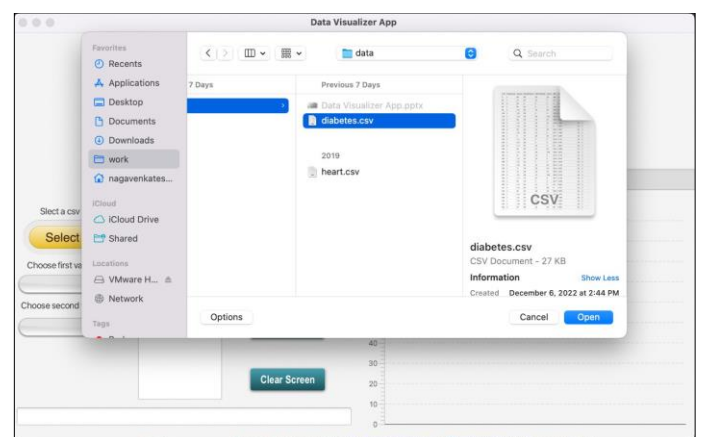
If the user experiences any slowness while using the application, they can simply click on the “Clear Screen” button, which will clear all the charts from the UI.



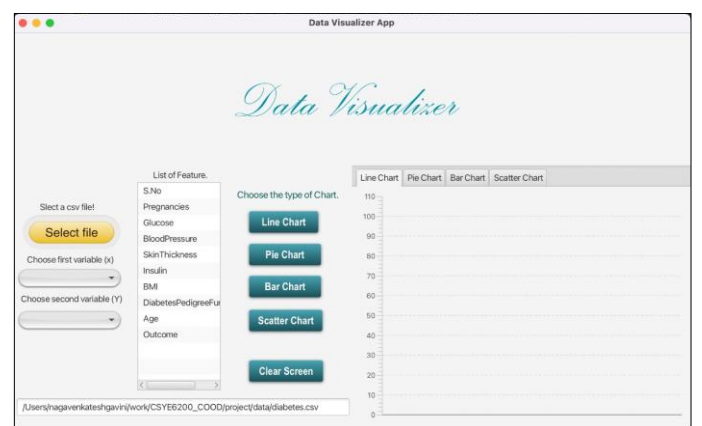
Landing page of the application



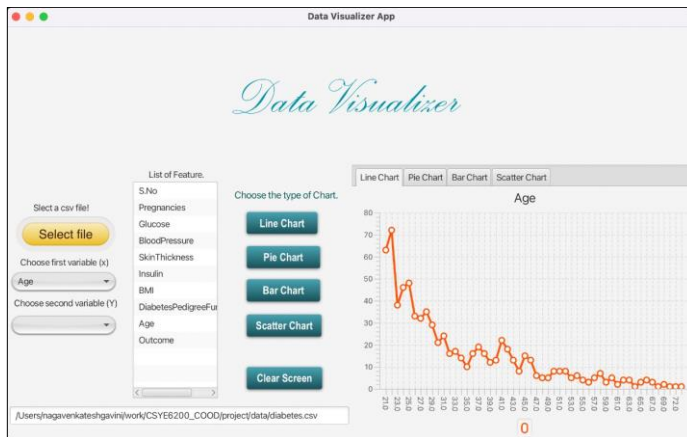
Opening page of the application



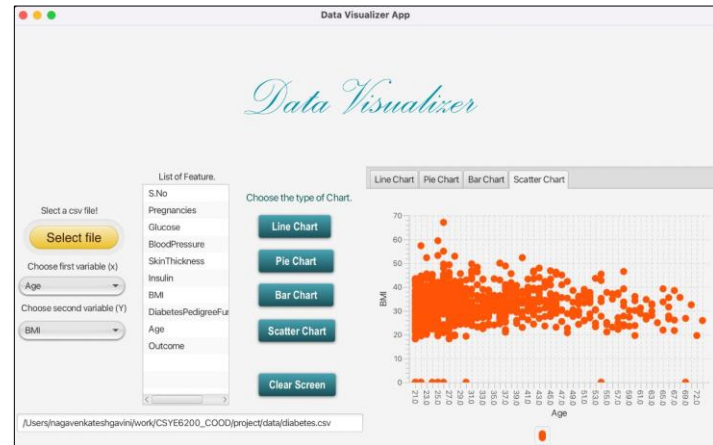
Importing the data through CSV file



loaded data and list of features



Depiction of data with line chart



Depiction of data with scatter chart



Depiction of data with pie chart



Depiction of data with bar chart

VII. DISCUSSION (REFLECTION)

By implementing this project, we got a chance to identify the following areas.

1. Depict data in a visually interactive way
2. Gather sizable amounts of complex data and render its depiction in an easy visual way.

These outcomes help the user to - Reduce the chance of making poor decisions regarding industry related or research related decisions, assists in identifying important factors that influence business outcomes and aid the development of a roadmap for upcoming action.

We also found that JavaFX provides a conducive platform and required libraries for displaying visualizations. Furthermore, unlike popular perception, Java is as effective a language and tool to implement visualization of data through application like any other languages/tools.

VIII. CONCLUSIONS AND FUTURE WORK

Data visualization no longer remains to be viewed as an art form. With Evolving software architecture, complex imaging, and Artificial intelligence, a lot of complex data can be recognized and explored using data visualization. Data visualization has facilitated decision-making for businesses as a digital replacement for visual communication. Data visualizer is made to be incredibly simple to use. The tool fetches data through csv file and presents it to the user through a simple and practical visualization.

During the implementation of this application, we found that we have not yet dealt with one of the issues that appear to cause issues in rendering accurate visualizations. This happens to be the issue of missing values in the data. Our next course of action would be to impute these values

corresponding to the kind of data we are dealing with. Currently, we are looking at ways to deal with this issue through usage of libraries and other tools. Some of the most popular technique is replacing the NA with mean, median or mode. We'll also start adding this feature from this basic technique.

The future of the project includes – Introduction of animation components to charts, incorporation of more categories of charts like – area chart, waterfall chart, ribbon chart, etc. That can enable the application to deal with data of varied nature. Furthermore, we are contemplating usage of third-party libraries such as TableSaw, jFreeChart which can render visualizations that appear to be more aesthetically appealing.

IX. JOB ASSIGNMENT

Naga Venkatesh Gavini worked on Scatter chart, landing page and styles, while Milad Tahmasebi worked on Bar Chart, Pie chart, and reading data. Rohan Mishra worked on Line Chart, UI integration with backend. All three contributed to the power point presentation, report creation, exploration, analysis, bug fixes, testing, planning and execution.

REFERENCES

- [1] Official JavaFX documentation: <https://openjfx.io/>
- [2] Official JavaFX charts documentation: <https://docs.oracle.com/javafx/2/charts/jfxpub-charts.htm>