Understanding Traffic Accidents

Jaini Patel
Rutgers University
Piscataway, NJ, USA
Email: jp1891@scarletmail.rutgers.edu

Arun Subbaiah
Rutgers University
Piscataway, NJ, USA
Email: as3590@scarletmail.rutgers.edu

Moriah Walker
Rutgers University
Piscataway, NJ, USA
Email: maw305@scarletmail.rutgers.edu

***Abstract*— Traffic accidents are the 9th leading cause of death in the United States resulting in more than 38,000 deaths per year, on an average. Moreover, an additional 4.4 million people sustain serious injuries that require medical attention. By age group, it remains the number one leading cause of death for people under 44. Hence, it becomes vital to gain a deeper understanding of the circumstances in which accidents happen and patterns if any. This will help policy makers to implement steps that effectively increase the safety standards of our transport system and reduce the fatalities due to traffic accidents.**

PROJECT DESCRIPTION

In this project, we explore and visualize a massive dataset on traffic accidents made available by National Highway Traffic Safety Administration (NHTSA). This helps us intuitively understand the circumstances in which traffic accidents occur and correlations between different factors. This project will help both lay persons and policy makers comprehend the data.

I. DATASET

A. Data Description

Additionally we used the Crash Report Sampling Systems (CRSS) that was also published by the NHTSA to normalize our data, so that we could have a fair picture of the actual relationships we wanted to showcase, which we will go into more detail with later on in this report [2]. In this project, we used one dataset that had been divided into 3 sub-datasets based on the category of its data and what attributes were collected. Fatality Analysis Reporting System (FARS) Data is published by National Highway Transport Safety Authority (NHTSA) [1]. This includes all the Traffic Accidents that involved one or more deaths over a time range of about 20 years from 2000 to 2019.

Additionally we used the Crash Report Sampling Systems (CRSS) that was also published by the NHTSA to normalize our data, so that we could have a fair picture of the actual relationships we wanted to showcase, which we will go into more detail with later on in this report [2].

B. Data Reduction

Before we can delve into the current size of data we should dive into the how this data size was achieved. The originally data collectively goes over 3 GB. At this size, the Django backend we used for the project would take a long time to retrieve the data we needed to render a page because of the huge data size. The best way to reduce this load time is to reduce the data size thatâs being rendered. We took a few steps to reduce the data's size.

1. Reduce Number of Columns - From examining the data, we realized that we did not need all the attributes in the data set, so we made a list of all the useful attributes that would answer our questions and removed the others, which is how we were able to reduce our data to minimize the load time in general for our website.
2. Custom Sets for Each Page - The load time was still not satisfactory with loading the data this way and on top of that we had to find a way to combine these subsets into one. We also realized that not all of the pages were using the attributes we selected. To not load excessive data for each page, we further subdivided the sets into what attributes were needed for each page and merged these new sets whenever possible.
3. OLAP Cube - Creating special sets for each page of website still caused us some issues. Ultimately, the size of all these smaller datasets combined is still equal if not greater than the original dataset. In order to combat this issue and to really get the best load time possible, we decided to use the OLAP Cube concept. We found the OLAP Cubes concept to be very powerful in reducing data load time [3]. To implement this concept, we aggregated the data based on the attributes used in the respective plots. This reduced the size of the datasets greatly. We found that, using these smaller aggregated datasets versus using the entire original dataset as a whole, the former reduced the rendering time of the page by 4 to 5 folds. Hence, we uploaded these new datasets to our database and changed the required configurations in our web app. This helped us decrease the load time of our web application significantly.

C. Data Size

Fatal Accidents
Size: 600,000 rows. 156 columns. 800 MB.
Person
Size: 1,700,000 rows. 255 columns. 1.2 GB.
Vehicle
Size: 1,000,000 rows. 204 columns. 1.4 GB.

D. Data Representation

The dataset is static, time variant data and in CSV format. Time is essential to processing the data as it plays a huge role

as a pimary key for each datasubset. Each row in the dataset represents one instance of the accident. The years the data was recorded spans from 2000 to 2019.

E. Central Entities and Relationships

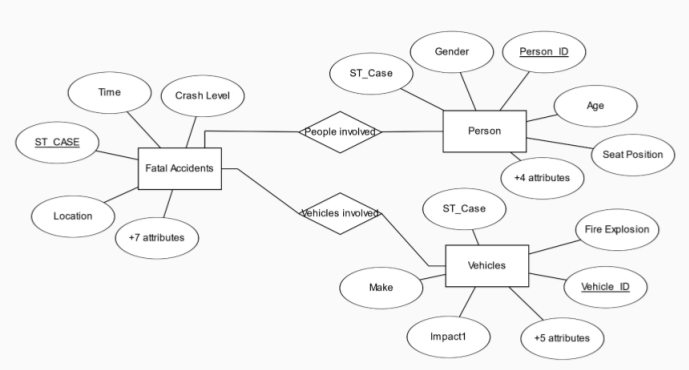
The central entities of this project are:

1. Fatal Accidents
2. Person
3. Vehicle

Primary keys: The ID of the accident and the year

Relationships: People involved is the relation between Fatal Accidents and Person -

Vehicle involved is the relation between Fatal Accidents and Vehicles



II. Questions that this project answers

Traffic accidents are one of the leading causes of death in the US and worldwide, so we wanted to make use of visual analytics to make trivial sense of the data to help lawmakers and stakeholders understand the prevalence of the situation. This those end users and what goals they may have when using our visual analytics tool in mind, we crafted a list of areas we wanted to provide answers for questions in those realms. Those main areas of focus were...

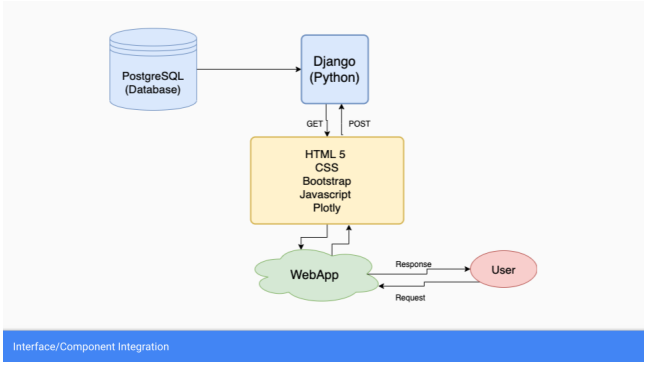
1. Spatio-temporal distribution of accidents across the United States
2. Accident-prone regions/states/counties
3. Unfavorable weather conditions for driving. i.e. weather conditions at which the maximum accidents happen
4. Duration of the day when the accidents happen the most
5. Fatal accidents due to drug/alcohol usage
6. Age and gender distribution in the fatal accidents
7. Common impact directions
8. Correlation (if any) between fatal accidents and the age of the car

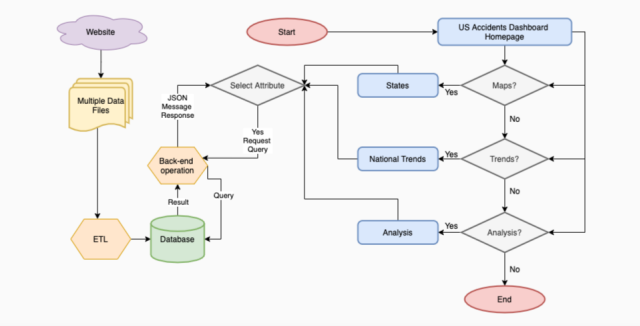
Additionally end users might be curious to see if any of these fields share any relationship with each other and what the data looks like when you narrow it down based on a value, such as time or a specific state. In order to answer these questions properly, we set out to not only show interesting graphs, but to allow for user interactivity as well as interconnectivity between graphs that responded properly to this interactivity.

With these goals for our project in mind, we decided to keep it simple and straight to the point and named our project *Understanding Traffic Accidents in the US*.

III. Mode of Processing

1. Database: Since the dataset is in CSV format, it was more appropriate to use an RDMS to manage the data. In order to make the data set more easily easily accessible and manipulable, we treated the data as key-value pairs throughout each level of processing. After the data has been added to the dataset, the data continues to be static. We chose PostgreSQL for this project because of the size of this data and how well PostgreSQL handles data of this size. Even though MySQl could handle the size of our data, we still chose PostgreSQL instead, because it is faster than MySQL.
2. Backend: The backend of this project was built using Python's Django Framework. The backend was used to retrieve data from the database properly according to which page the user was on and convert it into a JSON to be used by Plotly in the frontend.
3. Frontend: The frontend of this project uses HTML, CSS, Plotly and amCharts for the plots, Three.js, and Bootstrap. Plotly was chosen, because it possessed all the features that we were looking for and more and we found it very easy-to-use[4]. We added amCharts into our project a little later to help create charts with greater room for interactivity that could not be made using Plotly [5]. Towards the end of our project, we got the idea to use a 3D car model to represent our data, so using Three.js was for implementing this interactive 3D plot [6][7]. For the finishing touches of our project, we used Bootstrap so that our pages were more stylish and in line with the current standards for website frontend design[8].



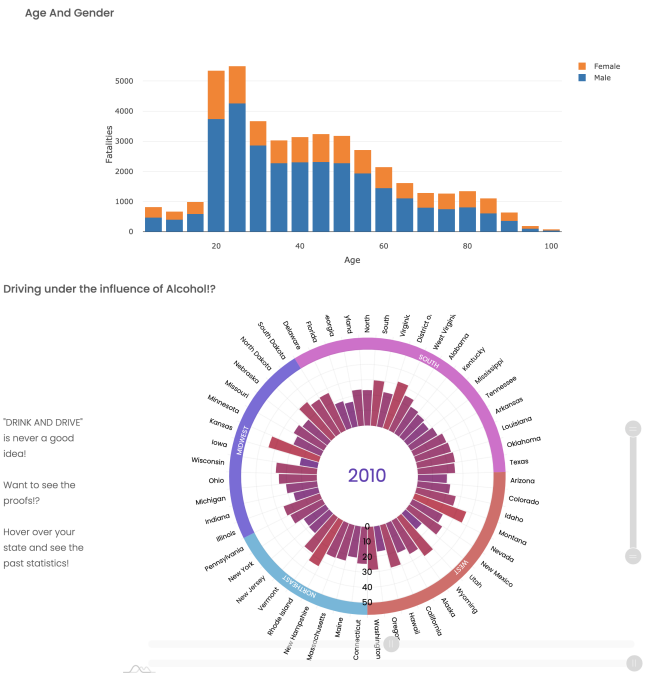


Conceptual Flow Chart

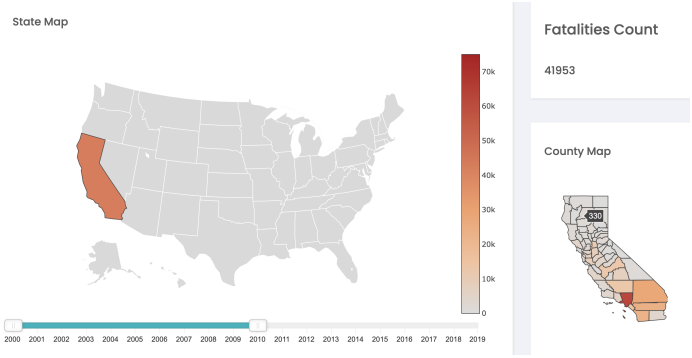
IV. Visual Representation

The main idea of this project is to make it easy for policy makers, stakeholders and lay people to understand the data and the trends intuitively. Hence, this project makes use of easy-to-understand visual elements like line charts and geographic choropleth maps

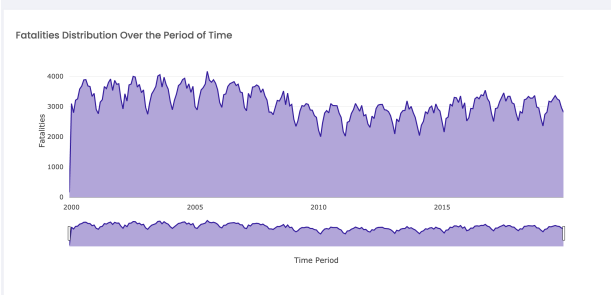
- Barcharts & Radar Barcharts: Barcharts are basic types of intuitive plots that can be used to compare categorical data. We have used it to show difference of accident fatalities based on different days of the week, different age groups along with different sexes, light conditions, and presence vs the absence of alcohol. Radar barcharts allow a similar experience but are presented in a circular form, which show more information about the data, such as the grouping of variables into categories. We use it show fatalities of different accidents for each state and divide it into the categories of their region in the US, which can be selected to show a comparison of states only in that region.



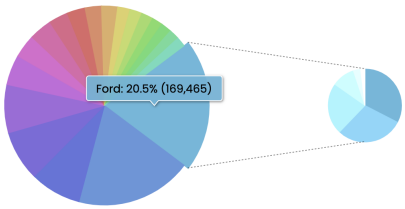
- Geographic Choropleth Maps: Geographic choropleth maps are color-coded maps that show the level of the presence of some attribute based on the values' association with a color scale. The distribution of traffic accidents across the US can be best visualized with the help of Choropleth Maps as shown below. We also got into further detail about the geographic distribution by showing a geographic choropleth at the state level for whichever state the user selects. Furthermore, the maps in the way we have set them up serve as an interface that allows the user to select a state, so that all other plots on the page are focused on only that state's data.



- Line Plots: These are the best way to visualize trend data. We used them to show the number of accident fatalities that occurred over a range of months, years and what the distribution of accident fatalities looks like at various hours of the day.

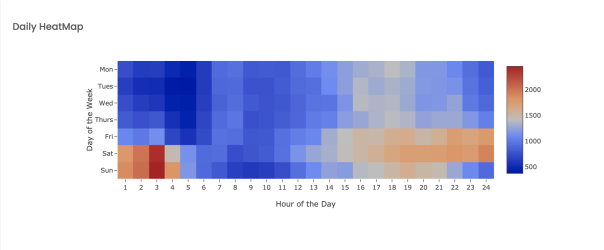


- Pie of Pie: Shows the percentage distribution of different brands involved in fatal accident that is linked to another pie chart that breaks down the distribution of car model's age for that specific model.

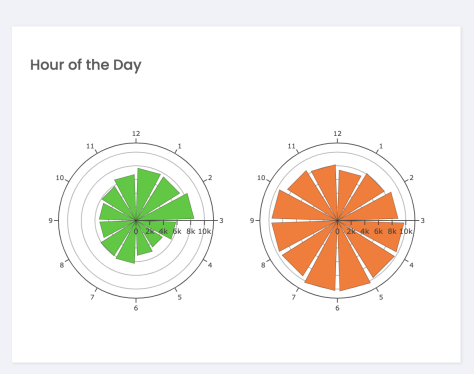


- Heatmap: The day of the week and duration of the day during which the accidents happen the most can be found using the help of Heatmaps. Time of the day and day of the week form the two axes of the plot. The number of

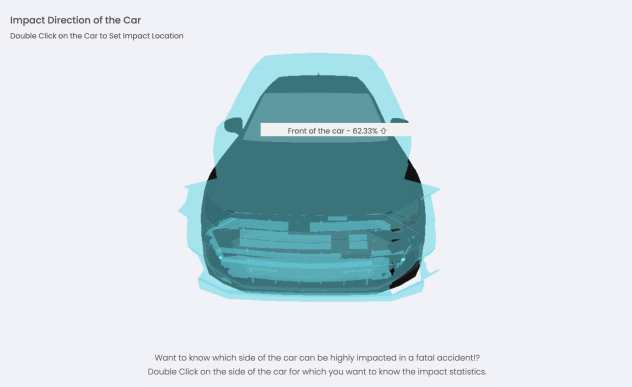
accidents is indicated with the help of a color scale.



- Wind-Rose Chart: Two different wind-rose charts on our website show a comparison of the number of fatal accidents that occur as separate AM and PM clocks. Another wind-rose chart shows the likelihood of getting involved in a fatal car accident when the impact occurs at the indicated area of the car.



- 3D Model: A 3D car model that displays information regarding the percentage of accidents that occur when the collision occurs at that impact point.

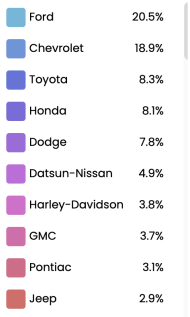


- Legend: Legends are used throughout all the plots to make sure the user is able to instantly make sense of the data they are looking at. The labels in legends can actually be clicked on to filter out the corresponding data.

V. Interactivity

This project makes use of interactive elements like zooming/panning, bruising, tooltip, etc. to help users focus on a particular area of the data, breakdown the data, understand the data on a deeper level and analyse it in their terms.

- Zooming Panning: This option is provided in the charts like geographic map and line plots to help the user focus on a particular area of the plot that they select using a selection box. This is one of the amazing features, which Plotly already has implemented for its plots and one of the reasons we chose to use Plotly. It is very useful for our line charts, because they cover such a large spread of data and panning and zooming allows the user to really zoom on the time periods they want to focus on and to what degree do they want to focus on that data.
- Mouse Hovering & Tooltip: This option applies to all the plots in this project including the 3D car model such that whenever the user hovers the mouse pointer over the plot, a tooltip pops up to show the data values of the the plot at that point. This is very useful for providing exact information instead of the general visual indicators from the image and the y-axis, which only provides general apporximations for what the data's value is.
- Mouse Hovering & Highlighting: A feature exclusive to the car model that shows a highlight over the car area that is being hovered over by the mouse, so that the user gets a visual picture of what that section of impact indicated in the tooltip really looks like.
- Checkboxes: Originally only a few care models are selected in the pie of pie char based on which car models were the most frequency, since there is such a large variety of car models in the dataset. To still give users the ability to access this data or deselect the ones that are most frequent, we add in the checklist for selecting or deselecting from a list of car models to determine which car models are represented in the pie of pie chart.



- Brushing & Selection: In the geographic choropleth map, whenever a particular state is selected, all the other related plots will show the data corresponding to the particular state instead of the whole of the United States. The linkage for all of the brushing is done by listening for an event, in this case selection. When this event occurs, the frontend takes the data that was initially loaded when the page loaded and goes through that data or the map created by that data to update the values used to generate each plot.
- Brushing & Sliders: Used throughout the project to choose a year or a certain range of time to have the data focus on. Affects all plots related to the slider. For the

geographic choropleth and its related plots, its slider's range is based completely on year, while for the year line charts and its related plots its slider's range is more granular and allows the the user to get more into detail with their selection down to the month level.The slider is also used on the bar plot dealing with drunk driving in order to allow the user to change how the bar chart is displayed (in its circular format or in a straight line).
- Summary Tab: Used to navigate the different pages of the website, so that the user can access all the plots and information our website provides.

VI. Analytics and Evaluation

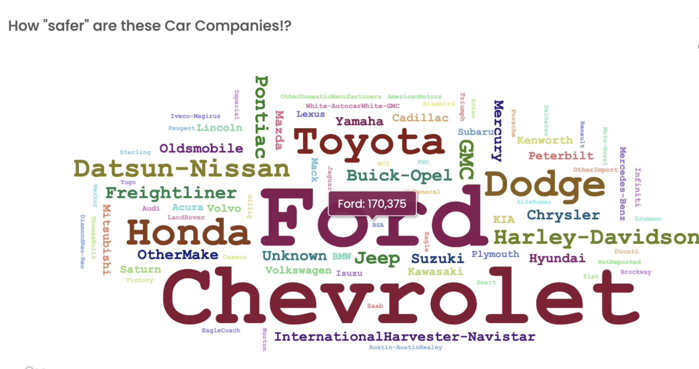
Analytics is an important part of this project. Since the data was initially very extensive with a large number of attributes, we decided to use to evaluate and identify insights such as

- dangerous regions for motorists,
- age group who need to be more cautious while driving,
- unfavorable weather conditions for driving,
- direction they are more likely to face the impact from
- at what age a car becomes more prone to accidents

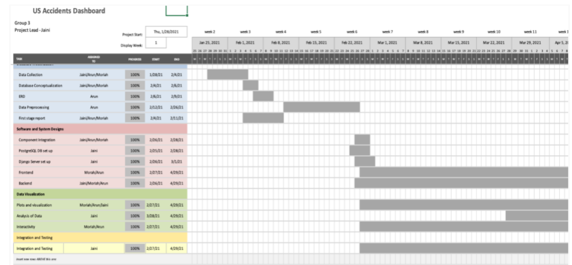
The insights were able to gain were...

- The most frequent states for accidents by a very high rate were California, Texas, and Florida.
- People in the age range of 15-20 got in the most number of fatal accidents.
- You are most likely to get into an accident when there are severe crosswinds.
- The impact direction of the majority of fatal accidents is from the front of the car.
- Saturday and Sunday during the time range 2:00-3:00AM were when the majority of accidents occurred.

We have also used a Word Cloud to analyze the vehicle data and show which cars are most commonly involved in fatal accidents, which we found the top 3 to be Ford, Chevrolet, and Toyota.



We were able to answer all of our questions and implement all of desired features in a timely matter as can be seen in this Gantt Chart.



From reviewing our tool from the perspective of a user, we believe that our expected users will want to and can reuse our tool because of the large interconnectivity of the data which each page demonstrates with multiple different factors. Take the Stats page for example. A person wanting to construct new effective policies for a state could examine what other states' data look in multiple scenarios and coordinate these findings with policy changes in other states and how policies affected different age groups and sexes as well as the time of day and day of the week when accidents happen the most.

VII. Further Work and Conclusion

In conclusion, we discovered some very interesting insights about the conditions which surround the majority of accidents and that supported statements, such as the dangers of drinking and driving. To further improve the results of our dataset in the future we could find other datasets to further normalize our data in certain areas, such as a car model dataset that provides the age and models driven thoughout the entire US. Beyond normalization, we could also integrate other datasets to provide other interesting information that would specifically help road developers, such as data about national highways, junctions, and intersections. It would also be interesting to integrate crash testing and recall data into our current and observe the relationship with these data in order to guide the search for the safest car. One of the most beneficial things we could provide through an increase in non-accident related car and road datasets would be the development of a prediction model that would help drivers, policy makers, and road developers alike in determining whether or not an accident's occurrence could be predicted and therefore further prevented.

References

[1] Fatality Analysis Reporting System (FARS) data published by National Highway Transport Safety Authority (NHTSA)
[2] Crash Report Sampling System (CRSS) data published by National Highway Transport Safety Authority (NHTSA)
[3] https://olap.com/learn-bi-olap/olap-bi-definitions/olap-cube/
[4] https://plotly.com/javascript
[5] https://www.amcharts.com/
[6] https://threejs.org/
[7] 3D Car Model https://www.cgtrader.com/free-3d-models/car/luxury/low-poly-car-no-1
[8] https://getbootstrap.com/