Web Application: Stay Alive - Think and Drive

CSE6242 | Team 001 | Ashish Dhiman, Cassidy Gasteiger, Paul Jordan, Siddharth Solanki, Yibei Hu

1 Introduction and Problem

Driving is among the most dangerous activities most Americans do every day[1]. Besides their incalculable emotional and physical toll, crashes cost the U.S. an estimated $340B each year[2]. Our goal was to assist drivers in planning safer routes as well as first responders in understanding where and when accidents are most likely to occur by building an interactive web app that visualizes accident risk. We implemented our analysis on a large-scale highway traffic accident dataset for the continental U.S., and deployed a prototype for the state of Georgia.

2 Literature Review

To usher in the era of smart cities, we must be able to predict and reduce traffic accidents. Past research has laid out the critical role of deep learning and big data in surveying for, predicting, and detecting traffic accidents to minimize damage, reduce costs, and save lives[3].

In the U.S., the National Highway Traffic Safety Administration aggregates annual crash data and creates dashboards revealing common trends[4]. However, these dashboards lack a map-based component and only include fatal crashes. More sophisticated geographic visualizations can reveal accident hotspots to aid policymakers in designing safer streets[5]. Innovative interactive heat-mapping approaches have revealed the power of mapping algorithms to link machine learning models and crime data[6]. Yet, there are no publicly available interactive risk maps that employ this approach for traffic accident data.

Many researchers have attempted to predict traffic accidents and accident risk using machine learning models. Deep learning random forest models[7], differential time-varying graph neural networks[8], and deep spatio-temporal graph convolutional neural networks[9] have all been used to calculate risk for accident-prone areas in specific geographies, and can be deployed in real time to provide minute-by-minute accident risk predictions. Researchers have also used k-modes clustering combined with association rule mining and trend analysis to uncover road accident trends, explore potentially related factors, and predict future accidents[10][11]. All of these models suffer from limited prediction accuracies in geographies not covered by the original dataset. A recent model develops high-resolution risk maps based on satellite imagery and GPS trajectories, incorporating real-time speed and traffic density information to provide predictions in areas even without historic accident data[12]. Although it has better prediction accuracy, it is still limited to the four cities in its training dataset.

Understanding the modeling limitations of accurate accident prediction, our analysis focuses on calculating risk of highways for which we have historic data, and how risk changes based on current environmental conditions. We used a 2.5M row U.S. traffic accident dataset, which covers 2016-2022 and most of the continental U.S. with data scraped from two large traffic APIs[13]. This dataset's robustness and size allows for larger-scale predictions; researchers have used ensemble learning models and deep neural network approaches to accurately predict accident risk and accident duration[14][15].

We quantified the above risk with the joint density of the various environmental factors in the accidents data and applied Bayesian inference[16]. Previous works have shown good results using Bayesian models in accident settings[17]. Because there was not a fixed parametric distribution which fit this joint density over the factors, we used Kernel Density estimation[18][19] to approximate joint density. However, given the large size of our dataset and the in-memory nature of KDE, we also experimented with fast variations of KDE[20].

3 Proposed Methods

3.1 Intuition

There are two state-of-the-art tools available right now: Google Maps, which provides the user with an optimal route between two locations, and RSF EuroRap 2022, which provides risk calculations for roadways based on historic accident data and road traffic. RSF EuroRap provides a more comprehensive and detailed calculation of accident risk than our tool in its current form. However, it is only publicly available for the UK, and it is not interactive; the user can simply view every road in the UK and its associated accident risk. Our tool combines Google Map's state-of-the-art optimization

technology with a risk calculation, and integrates real-time weather conditions from a weather API to provide a visualization of risk to the user based on current conditions. This powerful combination can aid the user in understanding where and when during their route to be most alert given the risk of an accident. During weather conditions with the greatest log-odds risk of an accident, it can also provide first responders with the stretches of roadway most likely to see a crash so they can ensure emergency vehicles are stationed nearby.

3.2 Approach

We began by conducting exploratory data analysis on the U.S. Traffic Accident dataset, then cleaned the dataset to impute or fill missing values.

From here, we conducted a two-pronged computational analysis to:

- Develop an accident propensity index for each stretch of roadway based on historic frequency and spatial proximity.
- Calculate the odds ratio of in-/decrease in expected accident risk based on relative frequency of real-time environmental factors, normalized by frequency of those factors in that locality[21].

The user interface follows these steps:

(1) The user enters a source and destination on the application.
(2) The application fetches the optimal path using Google Maps Directions API, which returns the coordinates of the path.
(3) The application calculates the accident propensity index of each route segment.
(4) The application fetches the current weather for the user's starting location.
(5) The application calculates the log-odds risk of an accident based on starting location, current weather condition, and historic weather data for that location.
(6) The application returns an overlay of color-coded information using the Google Maps JavaScript API on the selected route, along with summary statistics of historic accidents along that route and log-odds risk of an accident given current weather conditions and very bad weather conditions.

3.2.1 Accident Propensity Index With this feature, we provide the user with the accident propensity index of a given roadway based on historical accident frequency. The user inputs their starting location and end destination. Our application uses the Google Maps API to map the optimal route from start to finish, and returns a color on a continuous scale representing the relative accident propensity of that route.

We first divide the Google Maps-provided route into five segments. Then we calculate the accident propensity of each segment as follows:

$$Propensity = \frac{\sum_{i=1}^{accidentsonsegment} accident * severity}{\sum_{i=1}^{allUSaccidents} accident * severity}$$

Because the accident dataset is not comprehensive (we have extensive data for some cities or regions and no data for others), normalizing based on all accidents available in the dataset will ensure that every accident propensity index is subject to the same random variation in available data as every other index.

Our web application returns the optimal route for the user on a map based on the Google Maps API, with segments of the route colored to represent relative risk of each segment compared to the rest of their route. The most dangerous segment will be colored red and the least dangerous segment will be green. The segments in-between follow a gradient from red to green.

In addition to the color of the route, the user receives a sidebar with aggregated outputs of historical accidents along that route. This sidebar displays the total number of accidents that have occurred on their route based on the historic accident dataset. It also displays a histogram with an aggregation of time of day of historic accidents, so the user has a better understanding of how the current time of day may impact likelihood of an accident. Finally, the sidebar displays the log-odds risk of an accident given current weather conditions and how those odds would change if the weather took a turn for the worse. This calculation is described in more detail in the next section.

3.2.2 Odds Ratio of Accident Increase With this particular feature, we provide the user with odds of accident risk given the real-time weather at their location relative to the average weather conditions at that location. Mathematically, we define odds as:

$$Odds = \frac{P(accident|weather_{now}, time_{now}, location_{current})}{P(accident|weather_{avg}, time_{now}, location_{current})}$$

where $weather_{now}$, $time_{now}$ are the real-time weather and time at user's current location $location_{current}$, while $weather_{avg}$ for same conditions. We can then simplify the terms in the above equation using Bayes inference:

$$\begin{aligned} & P(accident|weather, time, location) \\ &= \frac{P(accident, weather, time, location)}{P(weather, time, location)} \\ &= \frac{P(weather|time, acc, loc) * P(time, acc, loc)}{P(weather|time, loc) * P(time, loc)} \end{aligned} \tag{1}$$

Thus, the calculation of odds simplifies to

$$Odds = \frac{P(weather_{now}|time, acc, loc)}{P(weather_{avg}|time, acc, loc)} * \frac{P(weather_{avg}|time, loc)}{P(weather_{now}|time, loc)} \tag{2}$$

In the above equation, the terms in the first fraction are obtained from our accidents dataset, while the terms in the second fraction are obtained from Web API. The weather data in the accidents dataset is collected from the weather station at the nearest airport. To maintain consistency, our application uses the same approach. Specifically, we:

(1) find airport weather stations closest to $location_{current}$
(2) find two weathers:
  (a) realtime weather $weather_{now}$ from the s and t_{now}
  (b) avg weather $weather_{avg}$ from the s and t_{now}
(3) get density of two weathers in data set
  (a) use $weather_{now}$ to get $P(weather_{now}|time_{now}, loc, acc)$ from accident dataset
  (b) use $weather_{avg}$ to get $p(weather_{now}|time_{now}, loc, acc)$ from accident dataset
(4) get density of two weathers in the area
  (a) $P(weather_{now}|time_{now}, loc)$ from API
  (b) $P(weather_{avg}|time_{now}, loc)$ from API

We consider our weather as multivariate RV with the the following covariates:

$$weather = \begin{bmatrix} Temp \\ Wind - Chill \\ Humidity \\ Pressure \\ Visibility \\ Wind - Speed \\ Precipitation \end{bmatrix}^T \tag{3}$$

3.2.3 Technology We conducted our back-end analyses of the Accident Propensity Index and log-odds accident risk using Python packages: numpy 1.24.2, sckikitlearn 1.2.2, scipy 1.10.1, and pandas 2.0.0. We used the Google Maps Developer API to render the map and call the optimal route between a user's inputted starting point and destination. We used Flask to integrate our back-end functions with our front-end user interface, which is built in React, a JavaScript-based component library. The webpage is hosted on Vercel, while the back-end static data files are hosted in a Github repository. All code files are available for further exploration, development, testing, and viewing on Github.

4 Experiments and Evaluation

4.1 Testbed

As we developed this project, we iteratively created experiments to test its effectiveness designed to answer the following questions:

- What are the limitations of our dataset? How can we handle skewed data?
- How can we clean the dataset to handle important missing values?
- How can we rapidly calculate the accident propensity of a given segment while searching a very large database of historic accidents?
- What geography is most effective to fetch real-time weather and calculate log-odds accident risk, given the original methodology for assembling the dataset?
- How can we handle the large number of covariates in our dataset to make accurate probability density estimates?
- How can we design the web app so it is easily accepted by the user?

4.2 Description of Experiments

4.2.1 Data Limitations To obtain a better understanding of the dataset, we performed EDA using Python and Tableau (see image of the dashboard below). The most important insight was that the dataset only includes accidents that happened on major highways. Furthermore, we observed that our data is skewed towards later years. The states with the most records are California (28%), Florida (14%), and Texas (5%), and the most mentioned traffic accident reasons are junction (291k accidents), traffic signal (265k), and crossing (200k accidents). This experimentation confirmed that we should limit calculations to highways and major roadways alone, and normalize accident propensity index by the whole dataset so all routes were subject to the same randomness, to adjust for incomplete data.

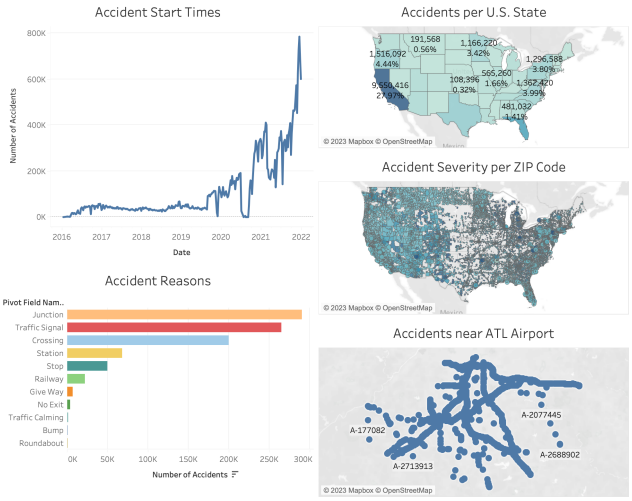

Figure 1: Tableau EDA

4.2.2 Data Cleaning Since we are not using address information for our analysis, we ignored all missing address values. To impute missing weather variables, which were critical for the log-odds calculation, we took the following steps:

(1) Wind Chill: imputed using outside temperature
(2) Precipitation:
  (a) if the categorical weather condition variable contained precipitation words, imputed using mean precipitation when precipitation was not 0 for that state during that season
  (b) if the categorical weather condition variable did not contain those keywords, imputed to 0
(3) Pressure, Wind Speed, Humidity, Temperature: imputed using hourly historical weather conditions from Meteostat API for airport code closest to the accident location
(4) Visibility: imputed using linear regression (.87 R^2 score)

We finally dropped about 0.7% of the dataset's rows, which had remaining null values.

4.2.3 Accident Propensity Index To calculate the accident propensity index, we ran a number of experiments to determine the most effective approach. Most importantly, we had to reduce calculation time to ensure a fast response for the user. Overall, the time increases linearly with the length of the route. After extensive experimentation, we divided the accidents into several thousands of csv files by latitude and longitude. Each of these csv files includes the accident data for a square of 11km by 11km (1/10th of a latitude respectively longitude). To retrieve the number of accidents between two points on a segment, we would only load the applicable csv file and search within this file. This decreased the running speed by multiple 100x. Now, retrieving the accidents and performing all related calculations for a route from, e.g., GT to UGA takes less than 10 seconds. Moreover, we limited the scope of our application to Georgia for now to decrease the needed storage speed. To expand to further states, we would only need to create the respective csv files and the computation of the accident propensity index would not increase.

4.2.4 Odds Ratio Our method for calculating real-time accident odds is based on Bayes Inference. We calculated the weather distribution for each location during different times when accidents occurred, then update the odds using real-time information provided by users.

Given that weather has high number of covariates, and due to the curse of dimensionality, this increased the requirement of data (at each location) for accurate probability density estimates. To relax this requirement, we performed PCA on the weather covariates to reduce it to two components.

Experimentation revealed that we could capture 90 to 95% variance in the accident dataset with principal component analysis.

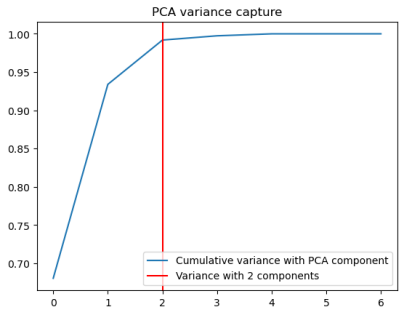

Figure 2: Explained Variance for a sample location

The odds equation 2 has two component distributions. Distribution 1 $p(weather|accident, time, location)$ is calculated using the accident data. Given time is a continuous variable, we discretized the time domain into relevant time steps. Our initial experiments suggested four categories based on summer/winter and day/night to effectively distinguish between weather conditions and best capture variability with a distinguishable kernel density distribution of weather conditions.

We used distribution 2, $p(weather|time, location)$, to normalization for the accident data, and used a similar method as distribution 1 to calculate it. We collected 3 years of historical weather data from the MeteoStat API for all 2000 weather stations in our dataset, which is used to calculate the weather distribution for a specific area/location as described in section 3.2.2

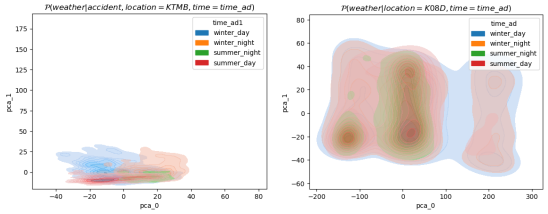

Figure 3: KDE 1 and 2

The above figures provide the density estimates for two sample locations. The next step is to calculate the same densities for each of the weather stations in our dataset. We also introduce the notion of $weather_{avg}$, which we calculate based on the Maximum Likelihood estimate from Distribution 2.

To complete the odds calculation, we query real-time weather at a given location through the same MeteoStat API. We evaluate the odds calculation by calculating

Location	Odds	
	current weather	*bad weather*
Atlanta	0.421849036	0.556300289
New York	2.24932	3.526061751
Indianopolis	0.232212393	1.644284629

Table 1: Estimated Odds values at 4PM EST Apr 22

odds for a large combination of real-time weather conditions, and use the results as feedback to update the calculation our original density estimates.

Our time discretization method defines "Summer" as the period from June to September, while "Winter" encompasses the rest of the year. "Day" is defined as the period from 9 am to 9 pm, and "Night" is defined as the period from 9 pm to 9 am the next day. We experimented with other discretization grids, but this method best captures the variability of weather and results in a distinguishable kernel density distribution of weather conditions.

To provide more context to the user around the odds calculation, we have decided to include odds both based on current weather conditions, and showcasing how worse the odds could be if the current weather turned bad. We define bad weather as the point with the highest density in $p(weather|accident, time, location)$. The second version of odds thus serves as a benchmark for the user, and helps establish the baseline. We have tested our odds calculation using empirical evaluations at different locations, and the experiments suggest intuitive results.

4.2.5 Usability Evaluation We prioritized using Google Maps as our optimal route-finding tool to ensure easy acceptance by the user, since it is the industry-leading mapping tool. We used a simple design and as few words as possible on the site to ensure ease of understanding of the user. We also used familiar Google colors for the user input section so it would be easy to understand.

We also conducted user testing to gather feedback on our web application. We conducted five one-on-one sessions to garner more detailed feedback, and also received 17 responses to a remote survey that tested usability.

Initial feedback indicated that response time for the application still needed work, which led to us limiting the geographic route search to the state of Georgia alone for this prototype model. Further feedback

indicated that the application is intuitive and easy-to-understand, and that it returned useful information that had the potential to change users' driving behavior so they would focus more during the most dangerous segments of their drive. User feedback also indicated the tool would be more helpful if it covered a larger geographic area and/or local roadways, and if it allowed them to compare the danger of several possible routes to each other. For version 2.0, as we develop this app further to create a public product, we aim to include a few additional functionalities based on user feedback. Specifically, we will include:

(1) functionality for all of the continental United States, not just Georgia
(2) data for backroads and local road accidents to encompass all possible routes
(3) comparison routes - Google Maps loads the most optimal route along with calculating all possible routes, and we will display the top three most optimal routes along with accident risk so the user can assess if there is a safer alternative to the optimal path
(4) a relativity score that adjusts accident propensity of a route for relative vehicle flow on that route, so the user can compare accident propensity across all paths, not just the current path

5 Conclusion and Discussion

Our application provides clear and accurate information regarding relative accident risk for a driver embarking on a given route. It provides three key sources of information: the relative accident propensity of each segment of a driver's route, summary statistics regarding past accidents on the given route, and the log-odds risk of an accident given real-time weather conditions at the starting location.

The potential for impact is significant: large-scale uptake of this tool could reduce accidents if drivers are more alert and attuned during the most dangerous parts of their drive, and under the most dangerous conditions, and first responders may have faster response time if they use this tool to identify the most dangerous sections of roadway and post emergency vehicles nearby during dangerous conditions. Both of these results have the potential to save lives.

The "Stay Alive - Think and Drive" web application proves the feasibility of integrating a state-of-the-art route finding mapping tool and an accident risk calculation so users have real-time information about the safety of their chosen path. This application is simple, user-friendly, and according to our user testing, fulfills its intended purpose to make drivers more aware of the risks associated with their upcoming drive.

All team members have contributed a similar amount of effort to the project. Please note that one of our team members, Richik Sen, dropped the course.

References

[1] Marina Bolotnikova. America's car crash epidemic. https://www.vox.com/22677892/road-safety-traffic-accidents-us-pedestrians-cyclists, September 19 2021. [Online; accessed March 5, 2023].

[2] National Highway Traffic Safety Administration. The economic and societal impact of motor vehicle crashes, 2019. https://crashstats.nhtsa.dot.gov/Api/Public/ViewPublication/814160, January 10 2023. [Online; accessed March 5, 2023].

[3] Safa Ben Atitallah, Maha Driss, Wadii Boulila, and Henda Ben Ghézala. Leveraging deep learning and iot big data analytics to support the smart cities development: Review and future directions. *Computer Science Review*, 38:100303, 2020.

[4] National Highway Traffic Safety Administration. Fatality Analysis Reporting System. https://cdan.nhtsa.gov/DataVisualization/DataVisualization.htm, 2022. Accessed: March 5, 2023.

[5] Muhammad Babar Rabbani, Muhammad Ali Musarat, Wesam Salah Alaloul, Ahsen Maqsoom, Hamna Bukhari, and Waqas Rafiq. Road traffic accident data analysis and its visualization. *Civil Engineering and Architecture*, 9(5):1603–1614, 2021.

[6] Sandhya Harikumar, Viveka Mannam, Chiranjeeb Mahanta, Mounika Smitha, and Shazia Zaman. Interactive map using data visualization and machine learning. In *2020 6th IEEE Congress on Information Science and Technology (CiSt)*, pages 104–109, 2020.

[7] Honglei Ren, You Song, Jingwen Wang, Yucheng Hu, and Jinzhi Lei. A deep learning approach to the citywide traffic accident risk prediction. In *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, pages 3346–3351, 2018.

[8] Zhengyang Zhou, Yang Wang, Xike Xie, Lianliang Chen, and Hengchang Liu. Riskoracle: A minute-level citywide traffic accident forecasting framework. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(01):1258–1265, 2020.

[9] Le Yu, Bowen Du, Xiao Hu, Leilei Sun, Liangzhe Han, and Weifeng Lv. Deep spatio-temporal graph convolutional network for traffic accident prediction. *Neurocomputing*, 423:135–147, 2021.

[10] Sachin Kumar and Durga Toshniwal. A data mining framework to analyze road accident data. *Journal of Big Data*, 2(1), 2015.

[11] Mingchen Feng, Jiangbin Zheng, Jinchang Ren, and Yanqin Liu. Towards big data analytics and mining for uk traffic accident analysis, visualization and prediction. *Proceedings of the 2020 12th International Conference on Machine Learning and Computing*, 2020.

[12] Lingyu Liu, Yong Zhou, Jiaoe Wang, Shuai Wang, and Qian Chen. Inferring high-resolution traffic accident risk maps based on satellite imagery and gps trajectories. *IEEE Transactions on Intelligent Transportation Systems*, 2021.

[13] Sobhan Moosavi, Mohammad Hossein Samavatian, Srinivasan Parthasarathy, and Rajiv Ramnath. A countrywide traffic accident dataset, 2019.

[14] Sobhan Moosavi, Mohammad Hossein, Srinivasan Parthasarathy, Ruxandra Teodorescu, and Rajiv Ramnath. Accident risk prediction based on heterogeneous sparse data: New dataset and insights. In *Proceedings of the 27th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*. ACM, 2019.

[15] Yuting Zhao and Weihua Deng. Prediction in traffic accident duration based on heterogeneous ensemble learning. *Applied Artificial Intelligence*, 36(1), 2022.

[16] Andrew Gelman, John B. Carlin, Hal S. Stern, and Donald B. Rubin. *Bayesian Data Analysis*. Chapman and Hall/CRC, 2nd ed. edition, 2004.

[17] Markus Deublein, Matthias Schubert, Bryan T. Adey, Jochen Köhler, and Michael H. Faber. Prediction of road accidents: A bayesian hierarchical approach. *Accident Analysis Prevention*, 51:274–291, 2013.

[18] Yen-Chi Chen. A tutorial on kernel density estimation and recent advances, 2017.

[19] Dequan Gao. Spatial patterns analysis of urban road traffic accidents based on gis. *IET Conference Proceedings*, pages 1898–1901(3), January 2012.

[20] Joseph A. Gallego, Juan F. Osorio, and Fabio A. González. Fast kernel density estimation with density matrices and random fourier features, 2022.

[21] Yahia Halabi, Hu Xu, Danbing Long, Yuhang Chen, Zhixiang Yu, Fares Alhaek, and Wael Alhaddad. Causal factors and risk assessment of fall accidents in the u.s. construction industry: A comprehensive data analysis (2000–2020). *Safety Science*, 146:105537, 2022.