Group 5 Final Project Report

“Separating Driver Classes to Estimate the Safety Benefits of AI-Corrected and Autonomous Vehicles”

Micah Simmerman
Computer Science
Post-Baccalaureate
Program (CSPB),
University of Colorado at
Boulder,
jasi9001@colorado.edu

Nathan Palmer
Computer Science
Post-Baccalaureate
Program (CSPB),
University of Colorado at
Boulder,
napa8745@colorado.edu

Nolan Ollada
Computer Science
Post-Baccalaureate
Program (CSPB),
University of Colorado at
Boulder,
nool0624@colorado.edu

1 Abstract

Artificial intelligent (AI) driving systems are a relatively new facet of automotive research based on extending the “awareness” of onboard computer systems with information collected from a sensor harness. Camera, radar, lidar, and infrared sensors are some of the technologies commonly used to suit this purpose. The information collected from these sensor harnesses are processed by an onboard logical control unit (LCU) and handed to an onboard software system that enables the vehicle to make split-second decisions to preserve the life and safety of the driver and the potential victims of automobile collisions.

The automotive AI market is positioned for explosive capital growth according to a recent market estimation by marketsandmarkets.com[9], the current valuation of this industry is expected to climb from $2.2 billion in 2022 to a whopping *$7.7 billion by 2027*. Companies such as Argo AI, Tractable, BMW, and Intel are rapidly emerging as key players in a primitive automotive AI market that appears to be anyone’s game.

While the prospect of owning (and selling) an AI-Autonomous or AI-Corrected automobile is an alluring prospect, the subject of AI vehicles has raised certain ethical concerns about the implications of autonomous vehicles on public traffic and environmental safety laws. In particular, the question of “*who should be held responsible in the event of an accident (involving an AI vehicle)*”, comes up a great deal, according to

a recent nasscom community article [8]. There is still much to learn, therefore, about where these AI technologies may be applied to achieve the greatest benefit to public safety and consumer satisfaction associated with AI vehicles.

2 Problem Statement and Motivation

The motivation of this investigation is to uncover the attribute qualities of data points hidden within the massive SWITRS California Traffic Collision data set. We use the insights we develop to gather an intuition about the primary factors involved in common automobile accidents and use the information we uncover to determine which of the driver categories would benefit most from an automotive AI technology.

We can then build vector-based classification models that expand upon the patterns and trends we observe. We can use these insights to derive new automotive AI-safety features that compensate for negative driving patterns observed in various consumer driver classes as well.

This project investigates data mining techniques for the efficient isolation of principle classes hidden inside massive datasets (i.e., those containing millions of data points or more). Throughout this article, we will discuss programming and memory-management techniques for handling and sampling large datasets. We will also discuss effective methods for data cleaning, smoothing, and other forms of data processing required to produce several common classification models.

Principal Component Analysis (PCA) and Data Cube techniques were used to uncover these driving trends, and all model-based approximations were produced from a n=10,000 random sample data set, which was collected from the three tables of the 'switrs.sqlite' source file (see the repository for code and documentation for more information).

The driving trends our group uncovered during numerical and categorical PCA have enabled us to identify a set of common driver categories that would appear to benefit greatly from one or more safety-based automotive AI technology. According to these findings, the most at-risk driver populations (i.e., the subjects of our search), would appear to benefit most from some form of an AI copilot system or on-board tutor, as we shall see.

3 Literature Survey

Our investigation into the prior work conducted in this field provided a few interesting analyses that we plan to reference throughout the scope of our work. One such resource is the "US Traffic - Getting to the Bottom of it", by Kaggle user "silversurf" (kaggle.com/silversurf)[6]. This analysis uses a data set called US Traffic and Fatality Records, and contains data from Fatal car accidents recorded in the United States from 2015-2016.

Our analysis is performed in a python notebook, and analyzes several data attributes collected from such fatal car accidents. This resource presents key findings that display the attributes that are most likely to contribute to fatal car accidents. These attributes consist of variables such as weather conditions, lighting, types of roadways and intersections, vehicle types, and driver impairment level (if applicable).

These attributes may act as a sounding board for the observations and conclusions drawn throughout this analysis, and the active reader is encouraged to have a look at them here. We also encourage you to visit https://tims.berkeley.edu/help/SWITRS.php to locate term definitions for the attribute columns presented in the SWITRS data set.

Another resource is "Tesla Deaths" (Tesla Deaths data set)[7], a dataset consisting of accidents involving Tesla vehicles that include the death of a driver, occupant, cyclist, or pedestrian, as well as whether Tesla Autopilot was in use at the time of the accident. We plan to reference this dataset during our analysis on the potential advantages and disadvantages of Artificial intelligent (AI) driving systems.

4 Data Set

This project analyzes data collected from the California Highway Patrol (CHP) system, specifically, the Statewide Integrated Traffic Records System (SWITRS) database. This database consists of almost 10 million data points, representing every traffic collision recorded within gathering system limitations in the state of California, dated from January 1st 2001 to mid-December 2020. The database was assembled by Kaggle user Alex Gude (visit kaggle.com/alexgude)[2] by numerous requests to CHP for data made over the course of the last 20 years.

The format of this groups' data in the form of an SQLite file, and categorizes the data through dozens of attributes. These attributes consist of factors such as crash location, crash severity, weather and lighting conditions, road and intersection types, vehicle types, and victim information. (Link to data set)[2]

5 Data Extraction and Preprocessing

Numerous SQL and noSQL methods can be used to compose a sample data frame. This team has decided to utilize an sqlite source file of the SWITRS California Traffic Collision Dataset. This SQLite format was selected on the basis of the portability, the structured nature of SQL query dialects, and the relative simplicity of managing the database in file format.

The uncompressed 'switrs.sqlite' source file is on the order of *9.71 Gigabytes* in size. Therefore, a great deal of memory management is required to drop the unnecessary source columns from the sqlite database file. These memory-management considerations include "hushing" the sqlite journaling file (note that the switrs.sqlite file sets the journaling mode to "ON" by default). Silencing this output is critical for avoiding a local memory crash, as the generation of excess journaling scripts is observed to rapidly overload the main memory buffers of a common (e.g., laptop) computer.

After dropping the unnecessary attribute columns from the SWITRS source file, the resulting database is composed of some 47,157,168 data points, distributed across the three tables of the SWITRS database (i.e., ‘collisions’, ‘parties’, and ‘victims’). Among these, some 9,639,334 data points exist in the ‘collisions’ table alone. The cardinality of the collisions table represents the maximum sampling size based on real results.

This reduced-dimension file proves much easier to manage on a typical desktop computer system. Please refer to the python notebook for more information about the columns that were removed prior to random sampling.

Data Extraction and Preprocessing:

The reduced switrs.sqlite source file was next used as the subject of an python-SQLite random sampling technique based on the generation of a common case_id vector. A series of similar SQLite queries were then constructed to produce the following random-sample data frames.

- *collisions_n_10000_test_set_df*,
- *parties_matching_n_10000_test_set_df*
- *victims_matching_n_10000_test_set_df*

Each of these analytical test frames contains data points associated with a common set of 10,000 case_ids analytical data frames. See “SWITRS_Python_Notebook.ipynb” for more information about how this task was accomplished.

SQLite string queries were constructed and passed to an SQLite SQLAlchemy cursor to draw in six n=10,000 random sample data frames, created by two mutually exclusive draws of 10,000 randomly selected sample case_id’s which were used to draw in data from each of the three SWITRS database tables to create 6 PANDAS dataframes.

The contents of these data frames were examined for indications of general data quality, and no-response values. Data cleaning was performed iteratively until the data was of sufficient quality to examine with the sklearn library packages.

Over the course of standard data processing, attributes containing binary values had their values mapped to a value of 0 or 1 in preparation for numerical PCA. These attribute values indicate whether the topic is related to the event tuple or not.

Non-response values with support in excess of ~50% were annotated to alert team members to the possible defects in data quality associated with a certain region of data.

An example of a non-response value could be the result of an error-prone field in California standard traffic tickets. In particular, the ‘not_private_property’ attribute column has a non-null response ratio of less than 35%!

If the ratio of non-null data points exceeds a heuristically determined, user-specified minimum threshold, the data points of these columns can be restored using the approximation methods afforded by a reliable smoothing function. If the data quality has deteriorated below a user-defined threshold, then it is often best to view the data in an "as-is" condition and simply notate the observations on that column. The information collected from these attributes can still be useful however, and it is often a mistake to simply discard them.

6 Data Cleaning and Examination

PANDAS provides a highly-reliable tool suite for examining dataframes. These tools proved useful for examining the progress of our data cleaning efforts, and the information that was collected from these tools assisted our team tremendously in preparing the data for subsequent analysis.

To effect the necessary changes in data quality, an assortment of null-replacement, data-smoothing, and sklearn.preprocessing tools were used to shape the dataframes for categorical and numerical PCA. These data processing stages were also valuable for grooming the data sets in preparation for the sklearn machine learning classification algorithms. The type of transformation applied depends on the column itself. Please see "SWTIRS_Python_Notebook" for more detail about data transformation.

6.1 Null Responses

A variety of data replacement techniques were used to replace null values with the appropriate response. Often this included replacing null values with a zero response or annotating the presence of missing value in certain sections of the database-derived sets.

7 Principle Component Analysis (PCA): General Methodologies

This section focuses on the methods we used to mine the most interesting and important attributes in our dataset. At this point of the analysis, data cleaning and extraction have reduced the size of the SQLite database file to a manageable size by eliminating unnecessary attributes. Conducting PCA on these columns is vital for determining the attributes that deserve the greatest focus under the context of the current analysis.

7.1 Isolating Attributes using Numerical PCA

A O(n^2) algorithm was developed to procure a list of cross-plots that constitute an exhaustive numerical PCA of the attributes of a combined data frame composed of the following NoSQL pseudo query:

INNER JOIN

'*collisions_n_10000_test_set_df'*
'*parties_matching_n_10000_test_set_df'*

ON 'case_id'

Some 992 cross plots were generated over the 86 numerical or converted-numerical attribute columns within this combined dataframe.

According to our experiments, this algorithm assumes a standard wall-time of approximately 9 minutes on a Lenovo x86-64 ThinkPad laptop computer and generates plots in a .pdf file format located in the same directory location as the .ipynb notebook.

Numerical PCA is as useful for determining attribute based clustering features as it is for identifying regression. Numerous observations have been made in each of these categories.

(See “Numerical_PCA_Report.docx” for more information.)

As we shall see, attributes extracted from numerical PCA can be given (i.e., “fed”) directly to the sklearn classifiers for the concise generation of class-based rules.

7.2 Assessing Class Qualities with Categorical PCA

Categorical PCA was conducted on the joined collisions and parties dataframe. All pairs of categorical variables were run through chi-squared testing to determine the strongest associations between attributes. Hundreds of associations were tested, but were then filtered to only include those we deemed to be strong or interesting. Attribute associations with low p values were extracted to be further analyzed. Some attributes had to be further removed manually, due to them not being interesting or relevant.

8 PCA Findings

This section focuses on the results of our PCA efforts. We will outline important findings at a high level, as well as how to find more in-depth results through files in our repository.

8.1 Numerical Attribute PCA Findings

The numerical PCA effort was a resounding, (albeit tedious) success, enabling us to identify the attribute dimensions containing some of the most interesting trends in our investigation. This method works as a flashlight would a dark cellar, enabling the analyst to rapidly search through all possible combinations of available attribute dimensions in order to make inferences about the underlying data structure.

Insights from numerical PCA enabled us to rapidly identify the attributes of at-risk driver classes. We can also use these inferences to make well-founded assumptions about the likely needs and perspectives of potential automotive AI safety consumers.

8.2 Categorical Attribute PCA Findings

Our efforts in PCA of the categorical attributes of the dataset yielded some key insights. After manually sifting through the pdf generated by our chi-squared testing, we were able to compile a list of pairs of highly associated attributes.

Using these relationships helped us determine the most interesting attributes to use when generating rules from our FP Trees by radically reducing the dimensionality of our data.

It was also reassuring to see that most of almost all of the categorical attributes highlighted by our PCA were ones that were most intuitively associated with collision causes and outcomes.

9 Classification Methodology

This section outlines the implementations of our classification techniques at a high level.

9.1 Rule Mining with Decision Tree Classifiers (DTC)

The sklearn module package offers a variety of DTC library tools that are covered by the extensive documentation on their website. This documentation includes refreshingly concise examples of how to invoke common sklearn library functions.

The Decision Tree Classifier (see SWITRS_Python_Notebook for more information) is modeled after one such example in the sklearn documentation, found here. This exemplary sklearn tutorial includes boilerplate source code for determining the regression method best suited for answering a user-specified analytical query based on a defined attribute vector 'X' and a monitored outcome variable 'y'. The tutorial also explains how to construct tables that can be used to assess the accuracy and relative support of each generated model tested under the DTC Classification scheme.

Indeed, this tool represents a rule-mining **query engine**, which can be used to generate countless queries to mine any of the association rules presented in the findings of the Numerical_PCA_Report (see this report for more information on discovered trends).

In contrast to the authors' original intention, this sklearn-based decomposition is designed on the classic 'test-train-split' model for regression fitting. As a result, the number of training points used in developing the DTC classifier models is lower than we had originally anticipated, using ~5,000 training points, instead of the 10,000 samples included in the original dataframes.

The resulting accuracy of the models presented for the queries tested so far appear very good (precision values close to 1.0 with recall values as high as 0.886 have been observed) however, and so this reduced level of sampling does not appear to cause any issue.

The figures generated by this code present side-by-side comparisons for several DTC classifier regression models at once, enabling the analyst to select the model that best describes the data under the context of his/her query. See "Numerical_PCA_Report" and "SWITRS_Python_Notebook "for further information regarding DTC rule generation using the classifier tools created in this project.

9.2 Generating Rules from Frequent Pattern Trees (FP-Trees)

The mlxtend python library was utilized to encode entries of the collision parties joined dataframe into a transactional format that can be used with a FP growth algorithm. Using this algorithm, our team built a function that allows users to find frequent patterns of attributes given a classification rule. For example, using this function allows a user to find frequent patterns associated with collisions where the collision severity attribute had the value of “fatal”, in order to determine patterns of factors that were most commonly associated with fatal accidents.

9.3 Assessing Conditional Class Probabilities using Bayesian Belief Networks

Bayesian Belief Networks (BBNs), are a type of statistical model that represent the probabilistic relationships among a set of variables. They are used for various tasks such as classification, prediction, and decision making in uncertain environments. A BBN consists of a directed acyclic graph (DAG) where nodes represent random variables, and the edges between nodes represent the conditional dependencies between variables. Each node has an associated probability table that quantifies the relationships between a variable and its parents in the graph. The library that was imported for the BBN was pgmpy.models and pgmpy.estimators. BBN can be used to model the likelihood of dying in a motorcycle collision by identifying key variables such as helmet usage, speed, road condition, and more. These variables are structured into a directed graph, and their probabilistic relationships are quantified using statistical data. By inputting specific conditions into the network, one can calculate the relative risk of death in different scenarios. This model offers valuable insights for safety regulations and public awareness campaigns, relying on the quality of the data and expert knowledge used in its construction.

9.4 Mining Temporal and Spatial Trends with Data Cubes

Mining temporal and spatial trends with data cubes is a multifaceted technique used in data analysis to explore patterns across time and space. Temporal trends focus on patterns and changes in data over time, while spatial trends concentrate on the geographic distribution of data. A data cube, a multi-dimensional array of values, facilitates this analysis by allowing for the aggregation of data across different dimensions. Within a data cube, operations such as slicing and dicing, drill-down, and roll-up enable analysts to extract specific parts, view data at different levels of granularity, and quickly summarize large datasets. When applied to temporal and spatial trends, data cubes allow for the efficient comparison of trends across various time periods or geographic regions, such as comparing traffic accidents in different conditions across California. Various Business Intelligence (BI) tools and data analytics platforms support data cubes, making this method accessible for multiple applications including different factors such as weather, time of day, age, and other factors. This approach offers a powerful means for understanding complex datasets, revealing insights into patterns and relationships, and providing valuable support for decision-making.

10 Classification Results

The following sections focus on how we extracted our findings from our classification methods, and how we used them to come to the conclusions we found, included in section 10.5.

10.1 Decision Tree Classification (DTC) Rules

The analytical tools developed with the help of the sklearn online documentation example produce an honest-to-life DTC class rule model generator, complete with statistics about the test performance of competing sub-models.

The DTC classifier works by separating data points based on a series of entropy or Gini-index driven split decisions that isolate tuples based on their attribute vector qualities. The result is a tree-based data structure that represents the sequence of decisions that was applied to achieve the greatest separation of data points in the set. Rules are then generated by combining the relative weight of each decision to produce a model that can be used to classify incoming data points.

The authors encourage the interested reader to investigate his/her own assumptions about the hidden data structures within the SWITRS sqlite source file, by experimenting with the analytical tools included in this repository.

Please see the documentation provided under the "Constructing the Decision Tree Classifier (DTC)" of "SWITRS_Python_Notebook" in this project repository for information about modeling rules with DTC Classifiers.

10.2 FP-Tree Based Classification Rules

Using our FP-Tree function, we were able to derive frequent patterns associated with given classification rules. The frequent patterns were then sorted by their support, and written into a pdf file for easier reading.

An interested reader will be able to utilize the "Constructing Frequent Pattern Tree" section of the "SWITRS_Python_Notebook" to find frequent patterns associated with any given classification rule.

For the purposes of our project, we decided to focus on the frequent patterns associated with collisions resulting in fatal outcomes for one or more parties involved. This enabled us to find patterns in the most severe collisions so that we could determine the most potentially useful applications and features of AI driving systems.

Many of these patterns involved factors such as drunk driving, type of location the collision occurred at, and airbag capabilities of vehicles involved in collisions.

10.3 Bayesian Belief Network (BBN Classification Rules)

Under our Bayesian Belief Network model, we are able to ascertain the probabilistic relationships between the variables related to specific outcome events in the data set. These relationships were then represented graphically and summarized in a pdf file for easier understanding.

For the purposes of our project, we decided to concentrate on the relationships associated with party age leading to motorists killed in a fatal accident. This enabled us to uncover the underlying patterns in the most critical results with age associated with fatal accidents. Strong statistical significance was discovered with men driving at night under the age of 30 and fatal accidents. The trends also increased the rate of fatal accidents when adding inclement weather as a factor. Identifying this information can prove helpful for future potential benefits for AI driving in hazardous conditions as well as certain times of day that reflect higher risk of accidents.

10.4 Data Cube Trends

Traffic collision data in California reflects a complex interplay of factors including weather, time of day, road conditions, and human behavior. In recent years, the state has seen a notable number of accidents during wet conditions and fog, particularly in certain regions. Collisions tend to spike during rush hours and are more fatal at night. Urban areas report a higher number of incidents but often with less severity, while highways experience more serious crashes. Pedestrian involvement in accidents is significant, especially in densely populated areas and during nighttime. The state's unique challenges, like wildfires affecting visibility and the legal practice of motorcycle lane-splitting, also contribute to the collision statistics. Identifying these attributes may lead to potential safety enhancements in AI-driven vehicles in the future. Understanding the areas that have the most significant impact on vehicle fatalities can pave the way for improvements in collision prevention and accident mitigation.

10.5 Summary- Insights About the Driver Classes of the SWITRS Data Set

- Throughout this analysis, we have uncovered numerous safety trends that have led us to adopt the following observation-based conclusions about the SWITRS data set.
- party_age is inversely correlated with the probability of being involved in a collision of any kind.
- Younger parties are more likely to use a cellphone when driving.
- Elderly drivers are more likely to accrue both a greater number and greater severity of injuries as a result of a motor vehicle collision than younger parties are.
- Solo drivers carry an increased risk of a collision-involvement compared to drivers who travel with a single passenger. This observation appears to suggest that solo drivers and parents of novice drivers would benefit most from an “AI copilot” type of technology, based on the relative frequency of collisions within these classes.
- Bicyclists are among the top three vehicle types involved in collisions

in the California SWITRS database. Collisions involving bicyclists appear to present support levels close to 50% across the database! AI technologies appear well-suited for reducing collisions with bicyclists and, by extension, the number of bicycle injuries and fatalities.

- Fatal traffic collisions of all types are more likely to occur on a highway than anywhere else.
- Many fatal highway collisions involve an airbag that failed to deploy.
- A car running off the road is a common occurrence in fatal accidents.
- While proceeding straight has the highest association with fatal accidents, an actual vehicle maneuver that is common among fatal accidents is turning left.
- Many fatal accidents that occur on the highway also involve one more parties being under the influence of alcohol.

11 Tools and Libraries

1. SQLite

SQLite offers a highly portable and compact file representation that is very useful for handling large quantities of information. The structured nature of the SQL dialect queries also provides a powerful basis for random sample generation, and supports indexing techniques to enhance the speed and efficiency of large SQL queries.

2. PANDAS

PANDAS offers an extensive library of dataframe construction tools that were nothing short of indispensable for the data cleaning and preprocessing steps required for this project. PANDAS makes it easy to read data from a variety of source files, convert dataframes to alternative data structures, and apply transformations on the resulting structures in preparation for sklearn.

3. sklearn

The python sklearn library package provides many useful tools for data preprocessing and standardization, as well as the construction of tree-based classification and regression models that can be used to generate rules based on the statistical analyses and observed data trends. The machine learning packages provided by sklearn are highly compatible with PANDAS dataframes.

This compatibility with PANDAS appears to improve the usefulness of compiler messages when preparing dataframes for subsequent use in machine learning algorithms.

4. mlxtend

The mlxtend library was utlized for our FP-Tree implementation. It helped us to encode our data into a transactional format which allowed us to to run the FP-growth algorithms on our collision_parties dataframe.

12 Proposed Future Work

To enhance the depth and accuracy of our analysis on collisions, it's pivotal to incorporate data from diverse sources. By integrating information from AI vision systems, autopsy reports, and weather monitoring systems, we can achieve a more holistic understanding of each incident's contributing factors. Furthermore, for a more streamlined numerical analysis, we can conduct a Principal Component Analysis (PCA) using a dataframe formed by an inner join between the ‘collisions’ and ‘victims’ tables, offering a concise representation of the most significant features in the combined dataset.

REFERENCES

[1] Mubarak Almuntairi, Kashif Muneer, and AqeelUrRehman. 2022. Vehicles Auto Collision Detection & Avoidance Protocol. IJCSNS International Journal of Computer Science and Network Security, VOL.22 No.3, March 2022. http://paper.ijcsns.org/07_book/202203/20220315.pdf

[2] Alex Gude. 2021. California Traffic Collision Data from SWITRS. Retrieved from https://www.kaggle.com/datasets/alexgude/california-traffic-collision-data-from-switrs

[3] Hovannes Kuhandjian. 2022. AI-based Pedestrian Detection and Avoidance at Night Using an IR Camera, Radar, and a Video Camera. CSU Transportation Consortium.Project 2127. https://scholarworks.sjsu.edu/mti_publications/430/

[4] Hazem H. Refai and Fadi Basma. 2009. Collision Avoidance System at Intersections FINAL REPORT - FHWA-OK-09-06. Electrical and Computer Engineering Department. https://www.odot.org/hqdiv/p-r-div/spr-rip/library/reports/fhwa-ok0906.pdf

[5] github.com/agude. 2021. SWITRS-to-SQLite. SWITRS data dictionary retrieved from https://github.com/agude/SWITRS-to-SQLite/blob/master/switrs_to_sqlite/value_maps.py.

[6] kaggle.com/silversurf. 2018. US Traffic - Getting to the Bottom of it. kaggle.com data analysis. Retrieved from https://www.kaggle.com/code/silversurf/us-traffic-getting-to-the-bottom-of-it/notebook.

[7] Tesla Deaths. 2023. Retrieved from https://www.tesladeaths.com/.

[8] Brijesh Saluja. 2023. “What are the Ethical and Safety Concerns with Autonomous Vehicles?”.

https://community.nasscom.in/communities/digital-transformation/what-are-ethical-and-safety-concerns-autonomous-vehicles.

[9] August, 2023. “Automatic Artificial Intelligence Market Report”. marketsandmarkets.com; Report Code: SE 5533. https://www.marketsandmarkets.com/Market-Reports/automotive-artificial-intelligence-market-248804391.html