Influence of weather on accidents in New York

Anonymous

1. INTRODUCTION

Vehicular accidents play a major role in the realm of transport. Safety standards are put in place to ensure passenger safety and governing bodies, along with police authorities, ensure that road rules are followed to prevent a breakdown of the transport system. Although these rules have been put in place to prevent accidents, an ability to predict them will help reduce the same. Studying data related to accidents will help us understand these factors. Moreover, a deeper understanding of how these factors are influenced will provide us the ability to predict accidents.

While there are several factors that influence accidents, the one that plays a huge role is the weather. The study intends to find a correlation between various weather data points like temperature, dew point, wind speed and precipitation and major accidents. To do this, we investigate the accident data reports for the years 2014, 2015 and 2016 in New York state and merge data available from weather data sets of different New York counties.

We retrieved ordinal weather data from a website[4] which maintains historical weather datasets for each county in New York and merge it with the accident data set. This allows us to closely study weather data and relate it to prevalent conditions when the accident took place.

2. DATA ASSEMBLY

The data on vehicular accidents[5] is presented by New York state open data[5]. This data set presents the crash description, the time of the incident, road conditions and county where the crash occurred. The dataset also contains a single column that describes weather conditions. However, this is a nominal data type that describes four types of the weather conditions – "clear", "rain", "snow" and "cloudy". There is no ordinal data to describe the weather.
To improve our analytics we sought weather data with numerical values. We needed a weather dataset that gave us data on precise weather conditions at each county when the accident occurred. We looked at several weather data sources but most of them required us to pay to obtain historical weather data. Moreover, we needed weather data at the time of the accident and not a weather summary for the day.

The vehicular accident data set[5] has data on weather conditions and road conditions at the time of the accident. However, we cannot use the same for analysis as the two columns provide nominal data. We need ratio and interval type data to supplement the vehicular accident dataset and this was found on Weather Underground[4].

Historical data is published on their website and they provide several data points such as temperature, wind speed, wind direction, humidity and precipitation in a county for a given date. Moreover, it has data that was recorded every hour (from 1:00 AM to 12:00 AM) in the county for the given date. The website is structured in such a way that the weather information can be accessed by modifying the URL template. For example, if we wish to obtain the weather conditions for Rochester on 12/25/2016, we use the template URL for Rochester and append the date as shown:

```
https://www.wunderground.com/history/daily/us/ny/
rochester/KROC/date/2016-12-25
```

We have written a python script that goes through every row of the vehicular accident data, extracts the date and time when the accident occurred and appends it to the template URL of the county where the accident occurred. This script uses Selenium[2] to open a web browser and extract the table data by using its Xpath property and append the weather conditions at the time of the accident to a Pandas[1] dataframe. This dataframe is written to a CSV file to give us the weather dataset. To focus on a few counties, this script removes vehicular accident data records of all counties that have less than 10,000 accidents over the three years.

To increase the rate of data collection from the website, several parameters had to be tuned through trial and error. Firstly, multiple instances of the script were run to obtain data as quickly as possible. The Selenium[2] web driver object had to be closed after every 50 URLs to prevent cache issues. Each 50 URL chuck was instantiated as a thread to ensure web exceptions didn't crash the program.
As the script extracts information from the Weather Underground website[4], we decided to go through the website's terms and conditions of use. Under their "PERMITTED USE" policy, information from their website can be used for personal and non-commercial purposes. Users are permitted to access, view and make copies of data for personal and non-commercial purposes. According to their "OWNERSHIP/TRADEMARK" policy, any use of their data should be acknowledged by stating Weather Underground as the source.

Finally, data from the two sources was stored in CSV files and we used Python Pandas[1] dataframe object to perform analytics. Other analytics and visualization was done through Weka[3].

3. DATA CLEANING AND PREPARATION

The data preparation and cleaning process required us to clean data in two stages. We first cleaned the two individual datasets and then cleaned the combined dataset after the merge. We also dropped a few columns and combined certain values that had a similar meaning.

3.1 Cleaning Before Merge

Before merging the two data sets, we needed to ensure the data was clean. The first step we took was to remove duplicate data. All duplicate rows were removed using Pandas[1]. We eliminated all rows where the number of vehicles involved in an accident were not available as this was our target column. As our script extracted data from a website, there were several errors in a few columns. For example, the column "Weather Condition" had values like "Partly" or "Mostly". This was due to the script leaving out the term "Cloudy" while extracting the data. We used Pandas[1] to replace these terms as "Partly Cloudy" and "Mostly Cloudy".

Once this was done, we had to clean several columns that had numerical data with their units in the string. Columns that described precipitation had "in" next to the numerical data to represent inches, wind speed and gust had "mph" and the temperature had "F". There were several fields where the units were before the numerical data like "mph10". We cleaned these columns by removing these units and renamed the column names to include the units. For example, the column "Wind Speed" after cleaning was renamed as "Wind Speed (MPH)".

3.2 Drop Columns

Once the data was cleaned there were several columns we had to drop as we felt they had no bearing on the target column. As we had a date column, redundant columns like "Day of the week" and "Year" were dropped. Considering we could only extract weather data based on the column "Municipality", we dropped the column "County". Columns like 'Police Report', 'Collision Type Descriptor', 'Road Descriptor' (which described the road elevation and curve), 'Traffic Control Device', 'Pedestrian Bicyclist Action' and 'DOT Reference Marker Location' were dropped as they were descriptive data. We could not quantify these descriptions and definitively weren't able to link these columns to the target column.

From the data that was extracted from Weather Underground[4], we dropped "Wind Direction" and "Pressure". Though wind direction could influence an accident, we could not quantify it using the data available to us and atmospheric pressure has no visible influence on accidents. Once the columns were dropped and data was cleaned, we merged the two data sets. The merge process was a left join where the extracted weather dataset was on the left and the accident data on the right. We merged on the common columns "Date", "Time" and "Location". This resulted in a merged dataset with approximately 106,000 rows.

3.3 Cleaning After Merge

Once the datasets were merged, we had assessed the values various columns had. We ran a unique value count on each column and tried to reduce values that were similar. Terms like "Muddy" and "Slush" were combined under "Slush", "Drizzle" was combined with "Rain", "Flooded Water" was combined with "Wet" and "Mist" and "Haze" were combined with "Fog" (as they all represented visibility). Considering we had two weather description columns, we replaced the value "Unknown", that was found in the accident dataset, with the one found under the weather description column from the extracted dataset. The weather description column from the extracted dataset was then dropped.

Finally, the target column was grouped under 4 numbers. 1 represented an accident where only one vehicle was involved, 2 and 3 represented accidents where two and three vehicles were involved in accident respectively. 4 represented accidents where 4 or more vehicles were involved. This reduced the number of values from 23 to just 4. We chose 4 and above as a number to classify a major accident. Figure 3.3 and figure 3.3 plots show a comparison of the unique values before and after this process. We see that this reduces the number of outliers and brings them under one classification.

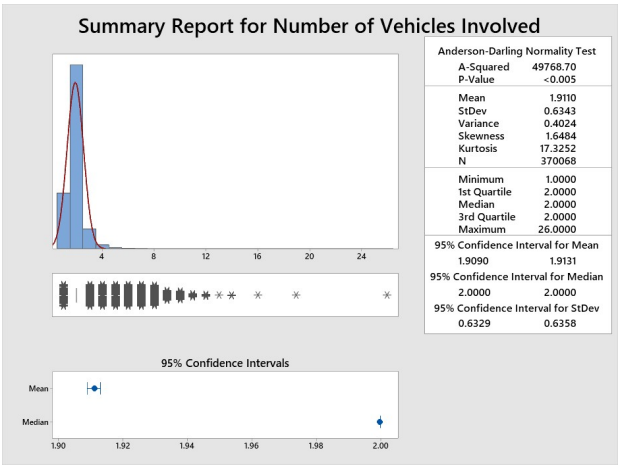

Figure 1: Column number of vehicles before cleaning

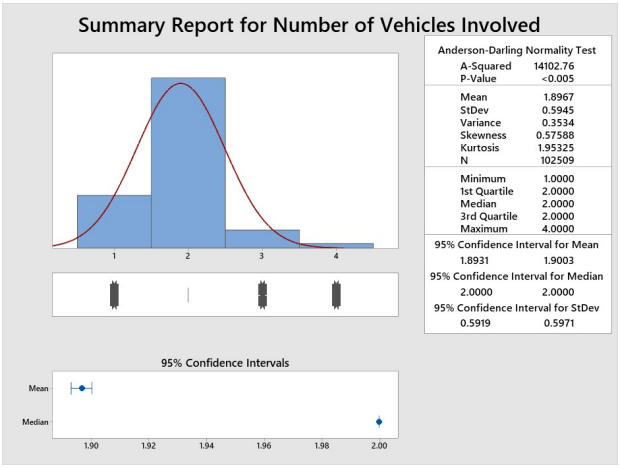

Figure 2: Column number of vehicles after cleaning

3.4 Pre-Data Mining Cleaning

Once the merged dataset was created, we felt we could tune the classifier by excluding certain columns from the data mining process. For this, we turned to Weka[3] to help us select features. Figure 3.4 shows the attribute ranking generated by Weka[3] on the combined data. We dropped features with a score lesser than 0.003 and the rest were chosen as data columns for the classifier. From the columns that were chosen, three columns ("Lighting Conditions", "Weather Conditions", "Road Surface Conditions") had nominal data. We performed one-hot encoding on these columns and this resulted in 17 additional columns. Figure 3.4 gives the descriptive statistics of the selected attributes being fed to the classifier.

```
Search Method:
       Attribute ranking.

Attribute Evaluator (supervised, Class (numeric): 10 Number of Vehicles Involved):
       Correlation Ranking Filter
Ranked attributes:
 0.05699   14 Lighting Conditions_Daylight
 0.02582   18 Weather Conditions_Cloudy
 0.02181   16 Lighting Conditions_Unknown
 0.02053   23 Road Surface Conditions_Dry
 0.01825   26 Road Surface Conditions_Unknown
 0.01704    4 Temperature(F)
 0.01612    3 Municipality
 0.01057    7 Wind Speed(mph)
 0.01052    5 Dew Point(F)
 0.00448   17 Weather Conditions_Clear
 0.00411    2 Time
 0.00326    8 Wind Gust(mph)
 0.00298    1 Date
 0.0018    20 Weather Conditions_Other
-0.0058    24 Road Surface Conditions_Slush
-0.0069     9 Precipitation(in)
-0.00724   15 Lighting Conditions_Dusk
-0.00806   19 Weather Conditions_Fog
-0.01504   27 Road Surface Conditions_Wet
-0.01605    6 Humidity(%)
-0.01666   13 Lighting Conditions_Dawn
-0.02182   21 Weather Conditions_Rain
-0.03124   22 Weather Conditions_Snow
-0.03848   25 Road Surface Conditions_Snow/Ice
-0.04255   11 Lighting Conditions_Dark-Road Lighted
-0.09032   12 Lighting Conditions_Dark-Road Unlighted
```

Figure 3: Feature selection ranking from Weka[3]

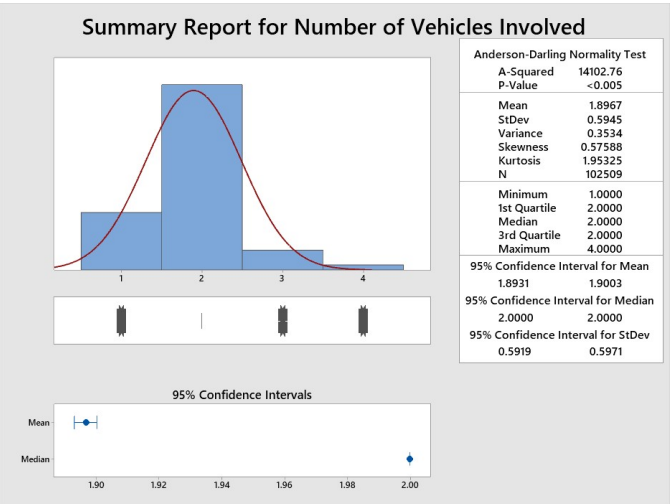

Figure 4: Descriptive statistics of the attributes provided to the classifier

4. DATA MINING

As the target column has 4 values where 4 represents accidents involving 4 or more vehicles, this column represents a classification. Thus, we chose to use a classification classifier and a linear regression classifier. We chose support vector machines as the classifier and ran a k fold cross-validation (k=5) for data mining. We obtained an accuracy of 69.27%.

5. FUTURE SCOPE

Given the time constraints, we could not extract all the historical data. There were close to 900,000 rows in the accident dataset[5] and the extraction process alone would have taken months. If this data could be obtained by purchasing data or accessing the paid API, we could perform a better analysis. Moreover, quantifying other attributes like road type (with curve, elevation and road condition) will help us improve accident prediction.

6. REFERENCES

[1] Pandas: Open source data analysis tool for python.

[2] Selenium for python: Python language bindings for selenium webdriver.

[3] Weka 3 - data mining with open source machine learning software in java.

[4] Wunderground: Local weather forecast, news and conditions.

[5] Motor vehicle crashes - case information: Three year window - dataset by data-ny-gov, Jun 2019.