Nicholas English

CIS-635-03

December 13, 2023

Studying Correlations Between Calls-For-Service and Weather

Introduction

Nature plays a pivotal role in our everyday lives, which includes the weather it brings with it. An icy day may result in increased traffic incidents, and rising temperatures can elevate human emotions. Given this idea, how much impact does weather have on calls-for-service? These two unique domains are therefore the primary areas of focus when it comes to answering this question. Finding any correlations found between these two domains could lead to better utilization of emergency services in any given weather scenario.

To answer this question, weather data was gathered from the National Oceanic and Atmospheric Administration (NOAA) and calls-for-service data was retrieved from the National Institute of Justice (NIJ), which was provided to them from the Portland Police Bureau. This raw data is then loaded into a Python project, which begins pre-processing by binning, cleaning, and interpolating the data. The pre-processed data then enters the data transformation phase by performing actions to normalize the data. This now normalized can then enter data mining and pattern evaluation phases, which includes learning and testing the data via KNN and decision trees, graphing the decision trees, and performing correlation analysis. The decision trees and correlation analysis results are stored into separate files, which allows easy access to the learned data after having executed the pipeline.

The results of this research suggest that there does exist a correlation between some calls-for-service with some weather conditions. The correlation analysis output, which is stored in a CSV, marks which call-for-services are correlated with which weather conditions. This gives us direction for when we open the decision tree graphs, which should give us a better understanding of what that correlation is. For example, “Assault” is suggested to be correlated with multiple weather conditions, which includes snowfall. The decision tree corroborates that and shows that higher snowfall appears to generally be more likely to be classified with the “Below Normal” classification.

Related Work

The effects of weather on human behavior have been an area of interest for some time. However, studies generally focus on specific types of calls-for-service, or at least more generalized types of crime. For my research, I didn’t focus on just violent crimes or domestic violence, but rather I searched for correlations between all provided types of calls-for-service and weather. Regardless, we’ll briefly compare what other researchers have experienced who examined this domain.

A thesis out of Canada, from Nipissing University and drafted by Ysabel Castle (see the references section for the fully styled MLA reference), sought to find the relationship between weather and calendar variables with violent and property crimes. This is a more granular approach compared to mine since Castle was focused on two areas of crime, however, she also expanded beyond my approach by including features related to the season. Castle's approach to handling the raw data was like mine, as we both gathered the calls-for-service data and weather data independently, and then performed cleanup and binning during the merging process. When it comes to results, Castle mentioned that the temperature variable was associated with the majority of violent crimes, which my study supports as it shows temperature has the most correlations of all weather conditions. Castle goes further to discuss the negative association between precipitation and snowfall compared to violent crimes. While I did not directly conduct a test for positive/negative associations, that does seem like a valuable piece of info that should have been included, rather than just finding if a correlation exists and then checking the decision tree for what that correlation causes.

Another academic paper, from Kent State University and authored by Paul Butke and Scott Sheridan (see the references section for the fully styled MLA reference), studied the relationship between weather and aggressive crime. This study also includes seasonal information, but that is due in part to them targeting summer versus winter crime analytics. Something unique that these authors do in their study, that I eventually opted to not do, is they binned their temperatures as either "hotter conditions" or "colder conditions." This makes sense for their study since part of their focus is how the summer season compares to the winter season. Butke and Sheridan's results showed that the summer seasons do have more aggressive crime compared to the winter seasons, which they state is proven by the linear relationship between the number of cases of aggressive crimes and temperature change. While I don't disagree that the temperature is playing a role, my research and the research from Castle suggests that temperature does play a role, but so do other weather factors such as precipitation and snowfall. Regardless, this study does still support the arguments from both me and Castle that temperature is a factor in the amount of crime that occurs.

The final academic paper is out of Northwestern University and authored by Alexandar Stec and Diego Klabjan (see the references section for the fully styled MLA reference). This study uses deep neural networks and attempts to predict the next day crime count predictions. This paper uses the Portland data similar to how I do, but they took it a bit further by also training on data out of Chicago. They also break the crime counts down into bins, again just like me, however they opted for 10 bins whereas I simply used 3 bins. Additionally, they used more than just weather to merge their data on; they also used census data and public transportation details. Using neural networks, their best model successfully predicted the next day's binning with an overall accuracy of 75.6%. After having seen this and knowing that I too was storing, but admittedly not using, my own models' accuracies, I went back into my own project and quickly added a snippet of code to output the average of my KNN and decision tree models. Both models

scored averages of just over 90%, which is better than the neural networks from this paper. However, this could have been attributed to my models having it easier since they only need to predict against 3 bins rather than 10 bins like the neural network models, or it could have been attributed to their neural network models having additional noise from the census data and/or the public transportation details.

Methods

This section will go in depth into how the data was gathered, and all the steps required from there to get to the knowledge learned.

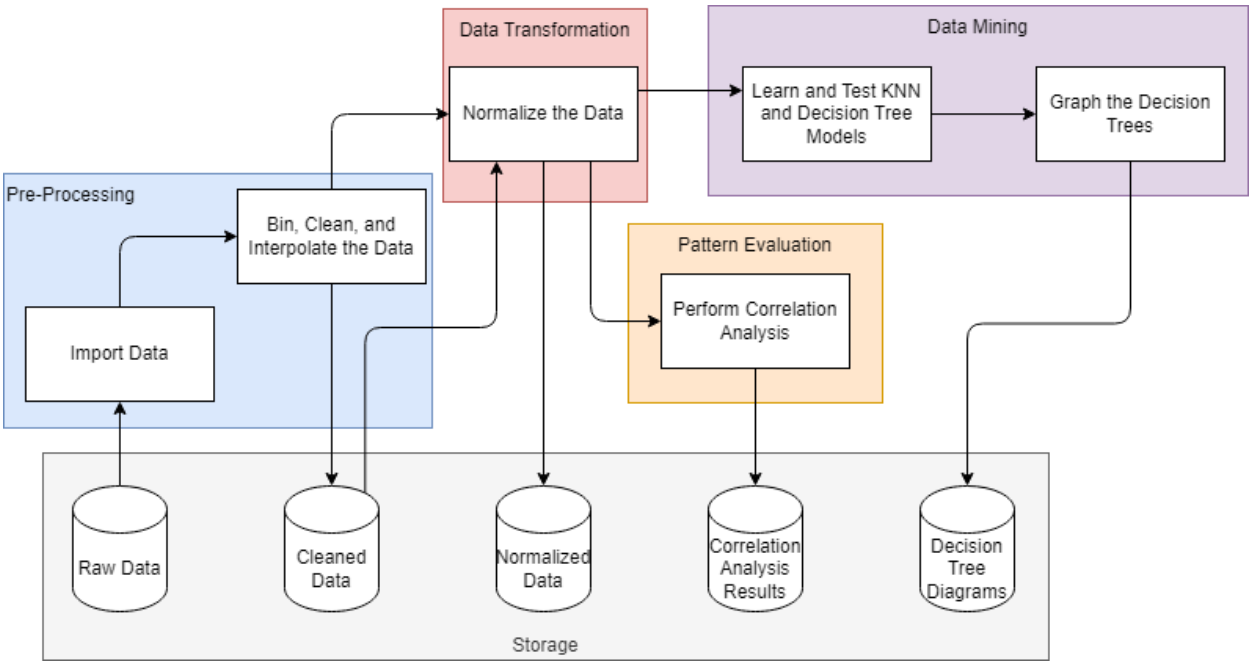
Data Collection

As mentioned previously, the weather data was collected from the NOAA, who received the measured values from Portland International Airport. The calls-for-service data was collected from the NIJ, who received the data from the Portland Police Bureau. This specific calls-for-service data was selected due to it being presented to the class at the start of the project, which meant there was no need to scan through countless search results to find an appropriate dataset. Knowing that Portland resides in the United States and that I wanted to use weather data, the source of this data was easy to pick since it could be grabbed from a federal agency, the NOAA. However, picking a weather station in Portland wasn't as straight forward since I found multiple stations weren't always reliable for measuring weather. When I discovered Portland International Airport's weather dataset, they were an easy choice since they had hardly any missing data, which meant to me that it should be reliable.

Once the sources were solidified, records were directly downloaded from the NIJ, and an order request was submitted to the NOAA for the desired weather data. All CSVs were then stored as raw data into the repository. It's also worth noting that the data was filtered down to only include data points that occurred between January 1, 2013, and December 31, 2016.

Data Mining Pipeline

When running the pipeline from Main, the user is prompted to enter a specific process to be run, which spans from simply importing the data, to graphing the decision trees. Most processes require another process to have been run previously, which could involve directly running an earlier process or simply importing data. The pipeline itself is divided into 4 main sections: pre-processing, data transformation, data mining, and pattern evaluation. Every section also saves data to local storage, which can either be used in the following section or can be analyzed on its own. The main purpose of saving this data between sections is it allows one to continue or repeat any phase of the pipeline without having to run the pipeline in its entirety. For a deeper dive into what happens in each of these sections, a detailed description of each will be provided below. Throughout the explanations, feel free to reference the below figure, which shows the general outline of the pipeline.



Pre-Processing

This section of the pipeline involves importing the raw data and then performing actions to bin, clean, and interpolate the data.

Importing the raw data involves reading multiple calls-for-service CSV files into a single DataFrame object. The weather data is much simpler with a single CSV file being expected, which is read into a different DataFrame object.

Cleaning the data is a multi-step process. First, calls-for-service data's text goes through a simple text cleanup operation, which involves stripping whitespace, replacing "&" with "AND", replacing "/" with "-", and removing suffixes from case types that involve priority levels. This text cleanup allows for consistent formatting and allows for fewer unique case types since priority is not a factor when considering how weather plays a role on the types of calls. Once the calls-for-service data has been cleaned, it then goes through the first phase of binning, which groups the data based on date and the case type. Both data sets then go through interpolation. Calls-for-service handles this by assuming a case type that does not appear on a specific date should have a value of 0. Weather handles interpolation by filling missing average temperatures with the average of the max and min temperatures, which was surprising that the data would sometimes be missing the average but never the max and min. Additionally, the weather type fields, which hold a Boolean value for if a specific weather phenomenon had occurred, was interpolated to assume the event should be set to false when missing. The rationale behind this is that while missing data does not guarantee the absence of the weather event, it's better to assume it did not occur than assuming it did occur. Following interpolation, the calls-for-service data is again binned, but this time so that each entry can be assigned a "Case Count" category, instead of relying on the numerical value for the number of events. This category is used to classify if

the number of events for that case type is above, below, or about normal. This classification is determined by taking the median, max, and min datapoints. If the number of events was below the halfway point of the median and min data point, it was considered “Below Normal.” If the number of events was higher than the midpoint between the median and max datapoints, then it was classified as “Above Normal.” Everything else was considered to fall in the “About Normal” category. After this, we then remove features that are considered unnecessary, which includes weather station details, weather values that we already have an average for, weather values that the weather station did not measure, and the number of events feature that was just binned. Finally, this section concludes that by merging the weather and calls-for-service data by joining on the date feature, and then saving the entire DataFrame into a CSV.

Data Transformation

This section begins either by using the DataFrames generated in the previous section, or by importing the CSV that that section would have saved. From there, we enter data normalization. This section seeks to ensure that all numerical values are on the same scale, which should help keep the data consistent. The min-max normalizer function was selected as the best choice to perform this operation since it would put the data on the 0 to 1 scale. Features that this was applied to include: average windspeed, precipitation, snowfall, snow depth, and average temperature. Once this is completed, the normalized DataFrame is saved to a new CSV file.

Data Mining

This section requires that the data transformation section has been completed, or the results have at least been imported. The data mining section has two distinguishable parts: learning and testing models, and then creating decision tree graphs.

The learning and testing part focuses on having two models learn the data, which are the KNN model and decision tree model. Importantly, there is a unique model used to learn and train for each case type, which was done because we wanted to focus on the relationship between case types and weather. If we didn’t have this distinction and used a single model, then it would also compare case types against each other. Before the program begins to learn each case type’s model, it sets up K-fold cross validation, which is used to split the data into smaller trainable parts and should result in reducing the risk of overfitting. The setup for the K-folds includes using 4 folds, setting shuffle to true, and using a random state value of “616.” Additionally, the decision tree models are set up to pre-prune the trees with a max depth of 5. Both KNN and decision tree models are trained simultaneously while iterating through the folds. Once training has completed for all case types models, the average score for both models is computed across all case types and is then printed to the console.

Once the models have all been trained, graphing the decision trees is quite simple. Iterating through each case type, the decision tree model is individually graphed and then saved to local storage as an image.

Pattern Evaluation

The pattern evaluation section has a dependency that the data transformation section has been completed. As stated before, this can be solved by either performing that process or by importing the CSV that that section would have saved.

This section of the pipeline is focused on generating any correlations that could be detected between the calls-for-service and weather. To achieve this, it first creates a new DataFrame that just holds the case types and the weather attributes. The program then goes through each case type and attempts to detect a correlation between it and each weather attribute. For numerical weather attributes (average windspeed, precipitation, snowfall, snow depth, and average temperature), correlation is detected by using Pearson (pearsonr) and is considered correlated if the p-value is below a threshold of 0.05. Additionally, for binary weather attributes (WT*, which are the weather type attributes), Chi2 (chi2_contigency) is used to detect correlations and is considered correlated if the p-value is below the same threshold of 0.05. When any correlation is detected, it is marked in a DataFrame which is then saved to a CSV once the process has completed.

Model Evaluation

Two models were used for data mining, which includes KNN and decision tree models. These models were trained and tested by implementing K-folds and having parameters set to use 4 splits and to shuffle the data. Across all calls-for-service case types, the models performed extremely well as they both obtained average scores over 90%. KNN models achieved an average score of 91.035% and decision tree models scored on average 91.138%. The scoring technique used to grade the models for each case type was the F1 micro-average, which was chosen due to its popularity for evaluating the performance of classification models and its ability to handle models where multi-class classification occurs.

Software Used

This pipeline was developed in Python (3.10) and utilizes many third-party libraries to achieve the result. Libraries used includes:

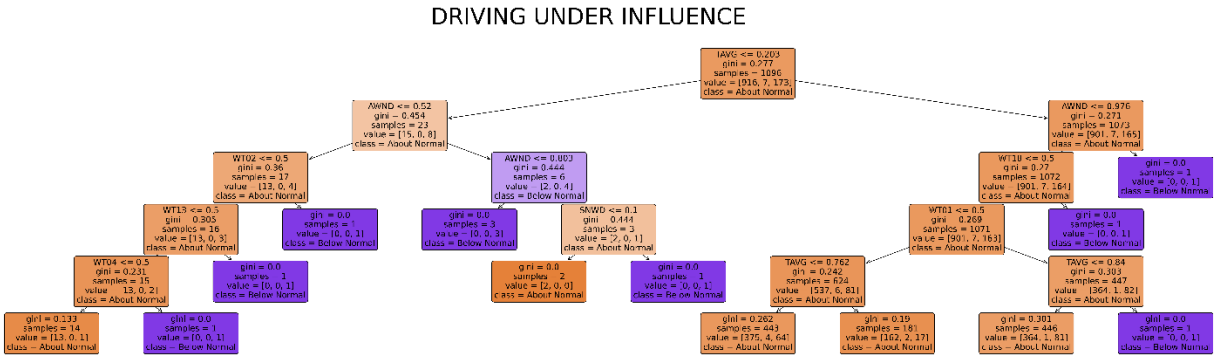
- Matplotlib
- Numpy
- Pandas
- Scipy
- Sklearn

Additionally, the pipeline was developed from the Visual Studio Code IDE, and the source code is managed and stored in a repository on GitHub.

Results and Discussion

The results of this research problem found that correlations between calls-for-service and weather do exist, and the decision tree models do help to display some of these correlations. The

two largest weather patterns that show correlations are average temperature and the weather type WT01 (which represents heavy fog, heavy freezing fog, and fog). Previous research supports the suggestion that average temperatures do play a role in correlations with crime, so it comes as no surprise that the average temperature feature has the most correlations. The addition that WT01, which we'll refer to as "fog" from here on, plays a higher role is something that hasn't gained as much popularity for influencing calls-for-service. Despite that, average temperatures and fog share most case types that they are correlated with, and they don't differ much. One example of a case that they share a correlation for is "Driving Under Influence." If we look at the decision tree graph that was generated, we can see that average temperature appears multiple times as a split point, and it was also the first split point. This helps reinforce that average temperature is playing a role, but fog is a bit tougher to see. In the graph, we can see fog (WT01) in the middle of the right-hand side of the tree, but it doesn't seem like it does anything special for adding correlation, especially when we can see many other weather conditions are causing "Below Normal" events. This is when pre-pruning might be hiding more of the information, since it's very well possible that a few layers deeper may reveal how much of a role fog plays for this case type.



To help aid in seeing how this pipeline came to its conclusions, please visit the demo that's listed in the "Data and Software Availability" section.

Conclusion

Out of this research, we discovered strong evidence that correlations do exist between calls-for-service and weather. Like previous studies, temperatures appear to have the most correlations, but the weather type WT01 (heavy fog, heavy freezing fog, and fog) was a very close second for number of correlations. Additionally, while using a decision tree model was nice since it allows for visualizing the model, it doesn't always help with displaying correlations, which is likely due to the pre-pruning to such a short depth.

What didn't go so well for this research was using such a broad approach for case types. Attempting to view correlations via decision trees is difficult when there is 80+ case types, each with their own decision tree. In the future, scope should be limited just like previous research studies, which would allow more focus on fewer case types and would allow time for examining decision trees with much greater depth. Furthermore, while using the KNN model was nice as a reference of performance compared to the decision tree model, not much was used from it and there probably should have been.

In the future, this research project could be expanded by including more data points. This data could come from additional cities in the United States and by expanding to include additional years. When expanding to include more years, ideally the project would be updated to handle this as a time series problem. It would also be recommended to perform additional binning by both expanding the number of bins for the number of events, and by binning case types into more generic categories. And finally, implementing an output for if the association between calls-for-service and weather is positive or negative would be very beneficial.

Data and Software Availability

- GitHub / Source Code
  - https://github.com/NWEenglish/GVSU-CIS635-DataMiningTermProject

- GitHub / Correlation Analysis Results
  - https://github.com/NWEenglish/GVSU-CIS635-DataMiningTermProject/blob/main/Learned%20Data/Correlation%20Analysis%20Results.csv

- GitHub / Decision Tree Graphs
  - https://github.com/NWEenglish/GVSU-CIS635-DataMiningTermProject/tree/main/Learned%20Data/Decision%20Tree%20Graphs

- YouTube / Video Demo
  - https://youtu.be/TIlogrxA3Ic

- Calls-for-Service (NIJ)
  - https://nij.ojp.gov/funding/real-time-crime-forecasting-challenge-posting#data

- Weather (NOAA)
  - https://www.ncei.noaa.gov/cdo-web/datasets#GHCND

References

Butke, Paul, and Scott C. Sheridan. *An Analysis of the Relationship between Weather and Aggressive Crime in Cleveland, Ohio*. Kent State University, 2010, doi:10.1175/2010WCAS1043.1.

Castle, Ysabel, and John Kovacs. *The Potential Influence of Environmental Variables on Spatial and Temporal Crime Patterns in a Small Canadian City : A Case Study of North Bay, Ontario, Using Call-for-Service Data, 2015-2019*. Nipissing University, 2021, https://library-archives.canada.ca/eng/services/services-libraries/theses/Pages/item.aspx?idNumber=1265302202.

Stec, Alexander, and Diego Klabjan. *Forecasting Crime with Deep Learning*. Northwestern University, 2018, doi:10.48550/arXiv.1806.01486.