Alex Beaumier, Daniel Haller

INFO SCI 412

Professor Anam

May 8, 2022

Project: Tornado Data Analysis

Abstract

In this report we will take our knowledge gained from datamining to apply the research tactics and organizational techniques to tornado data collected in the United States and through the National Oceanic and Atmospheric Administration (NOAA). Through an analysis of the data, we can search for correlations between the different tornado attributes and develop classifiers that can predict further attributes of tornadoes given our data.

Introduction

The purpose of this project is to further increase our understanding of data mining techniques, develop skills taught in and outside of the classroom to explore new methods of excavating hidden answers from a variety of sources. For this project, we tested our knowledge by applying what we learned in class to tornado data and determine if the data provides us with the ability to predict a couple of data patterns form the existing dataset. The data being utilized are tornado data collected by National Oceanic and Atmospheric Administration (NOAA). The reason we chose to research this topic was due to the natural unpredictability of natural weather conditions on a daily basis and a curiosity regarding tornado paths, seasons, and timings according to recorded data. Definitive conclusions we would like to gather are average fatalities per year, predicting deaths given a tornado occurrence, a visualization of total damages in dollars per tornado type and of crop loss, and where tornados are more frequent compared to other geographic locations. We will use various python-based libraries like Matplotlib, Pandas, NumPy, Seaborn, and Scikit-learn. These tools will help us in both the fields of cleaning and preprocessing the data, as well as helping us process possible correlations, attributes, and even classifiers regarding our data being researched. We believe understanding the patterns behind this data to be important in a wide array of field including real estate, travel, construction, and safety.

Experiment Methods

For our methods, we based them solely off of the orientation of the tornado dataset. Previously we had mentioned the use of two other datasets: hailstorm data and wind data. After a complete analysis of the tornado data, we opted to discard the former two datasets as the work would exceed the bounds of the original objective: to utilize our skills to analyze one dataset and report our findings in a five to 12 page report format.

The dataset named “1950-2020_torn.csv” contains 67,504 records across 29 attributes. The attributes (listed in the csv) are unique tornado IDs (om), ranges of dates and times (yr, mo,

dy, date, time, tz), state identification (st, stf, stn, ns, sn, sg, f1, f2, f3, f4), tornado statistics (mag, slat, slon, elat, elon, len, wid), and outcomes resulting of the tornado occurrence (inj, fat, loss, closs). All attribute labels were relabeled to a more descriptive title that could be readily understood (figure 1.1) without consulting a legend (specifically the pdf titled "SPC_severe_database_description.pdf").

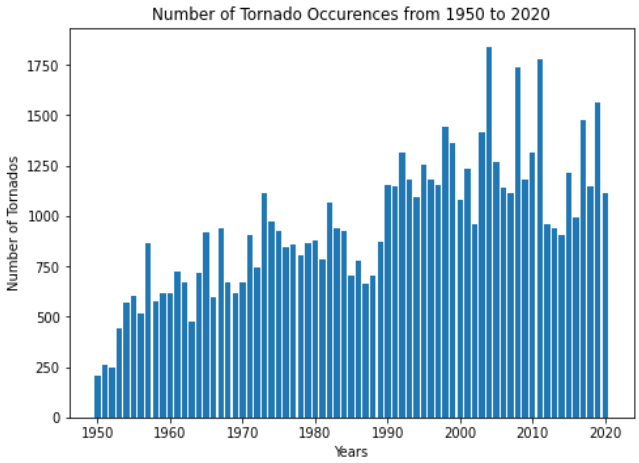
```
torn = torn.rename(columns={"om": "tornado_id",
                            "yr": "year",
                            "mo": "month",
                            "dy": "day",
                            "date": "date",
                            "time": "time",
                            "tz": "time_zone",
                            "st": "state",
                            "stf": "state_fips",
                            "stn": "tornado_id_by_state",
                            "mag": "ef_scale",
                            "inj": "injuries",
                            "fat": "deaths",
                            "loss": "prop_loss",
                            "closs": "crop_loss",
                            "slat": "start_lat",
                            "slon": "start_lon",
                            "elat": "end_lat",
                            "elon": "end_lon",
                            "len": "path_length",
                            "wid": "tornado_width",
                            "ns": "states_affected",
                            "sn": "state_affected_number",
                            "sg": "state_affected_segment",
                            "f1": "fips1",
                            "f2": "fips2",
                            "f3": "fips3",
                            "f4": "fips4",
                            "fc": "was_ef_scale_altered"})
```

"Column Rename", Figure 1.1

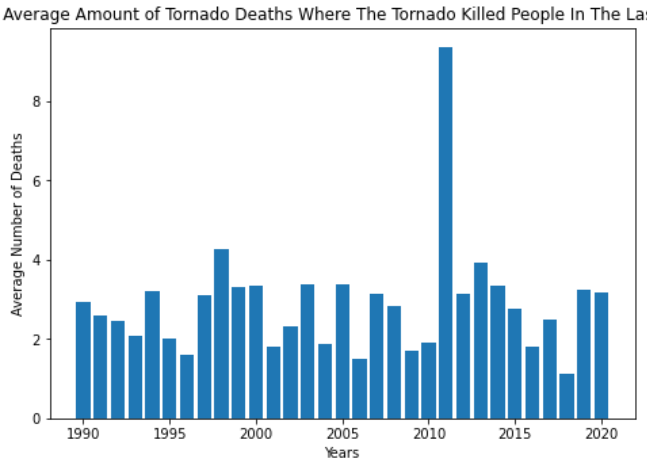
The evaluation metrics used to measure the outcomes of the prediction models are confusion matrices to visualize the number of cases that were correctly and incorrectly predicted, accuracy scores to measure the accuracy of the predictions, and finally whether or not the visualizations of the desired graphs are understandable and easy to interpret. The data mining classifiers used are per industry norm: KNN (K Nearest Neighbors), Decision Trees, Random Forest, and Multi-Layered Perception. To analyze certain columns and determine which KNN/Random Forest parameters were a best fit, combinations of each parameter were tested to identify the best overall arguments. Towards the end of basic testing, the classifiers were pitted against each other to model and test the same datasets and report their scores so we could see which classifier performed the best and why.

Results

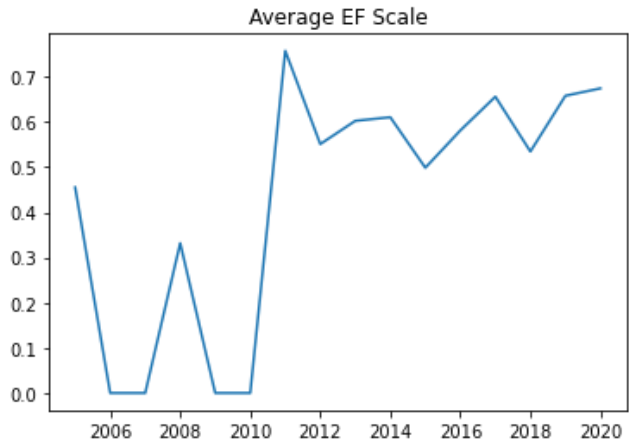
The first goal of data mining the tornado data was to create some basic visualizations and not only get a better understanding of the data, but also to use knowledge learned in the first half of the semester to demonstrate our abilities. Below are the basic visualizations created.



"Tornado Occurrences", Fig. 1.2



"Average Deaths", Fig. 1.3



"Average EF Scale", Fig. 1.4

To predict the number of deaths from any given tornado, we used the KNN classifier to model the data and test for the best KNN arguments. Below is a table of both death predictions, accuracy scores, and respective KNN arguments.

	Train Score	Test Score
Accuracy with deaths = or > 0	97.75%	97.46%
Accuracy with deaths only > 0	57.23%	48.71%

	Score	Metric	# Neighbors	weights
Model with deaths = or > 0	49.06%	Manhattan	7	uniform
Model with deaths only > 0	49.06%	Manhattan	7	uniform

From the tables we can interpret that using the classifier's optimal arguments does not interfere with it's ability to predict, albeit the low score. However, the accuracy score does drop significantly once all records with deaths = 0 are removed, therefore forcing the model to try and predict deaths from tornados that are deadly enough to kill. To make sure the best classifier was used, we ran the same dataset consisting of ef scale, injuries, property loss, crop loss, latitude, longitude, tornado path length, and tornado width. This data frame was than tested against the four classifiers KNN, Random Forest, Decision Trees, and Multi-Layered Perception to show which would be the best for modeling. The results are as follows:

Decision Tree

The accuracy of the model is: 0.6627906976744186

Classification Report:

	precision	recall	f1-score	support
1	0.83	0.75	0.79	127
3	0.10	0.19	0.13	16
5	0.62	0.55	0.58	29
accuracy			0.66	172
macro avg	0.51	0.50	0.50	172
weighted avg	0.72	0.66	0.69	172

[[95 25 7]

[10 3 3]

[10 3 16]]

Random Forest

The accuracy of the model is: 0.8081395348837209

Classification Report:

	precision	recall	f1-score	support
1	0.81	0.98	0.89	127
3	0.00	0.00	0.00	16
5	0.83	0.52	0.64	29
accuracy			0.81	172
macro avg	0.55	0.50	0.51	172
weighted avg	0.74	0.81	0.76	172

[[124 1 2]

[15 0 1]

[14 0 15]]

Best Random Forest

The accuracy of the model is: 0.7616279069767442

Classification Report:

	precision	recall	f1-score	support
1	0.79	0.94	0.86	127
3	0.00	0.00	0.00	16
5	0.69	0.38	0.49	29
accuracy			0.76	172
macro avg	0.49	0.44	0.45	172
weighted avg	0.70	0.76	0.72	172

[[120 3 4]

[15 0 1]

[17 1 11]]

KNN

The accuracy of the model is: 0.7383720930232558

Classification Report:

	precision	recall	f1-score	support
1	0.80	0.91	0.85	127
3	0.00	0.00	0.00	16
5	0.67	0.41	0.51	29
accuracy			0.74	172
macro avg	0.49	0.44	0.45	172
weighted avg	0.70	0.74	0.71	172

[[115 7 5]

[15 0 1]

[14 3 12]]

MLP

The accuracy of the model is: 0.7441860465116279

Classification Report:

	precision	recall	f1-score	support
1	0.74	1.00	0.85	127
3	0.00	0.00	0.00	16
5	1.00	0.03	0.07	29
accuracy			0.74	172
macro avg	0.58	0.34	0.31	172
weighted avg	0.72	0.74	0.64	172

[[127 0 0]

[16 0 0]

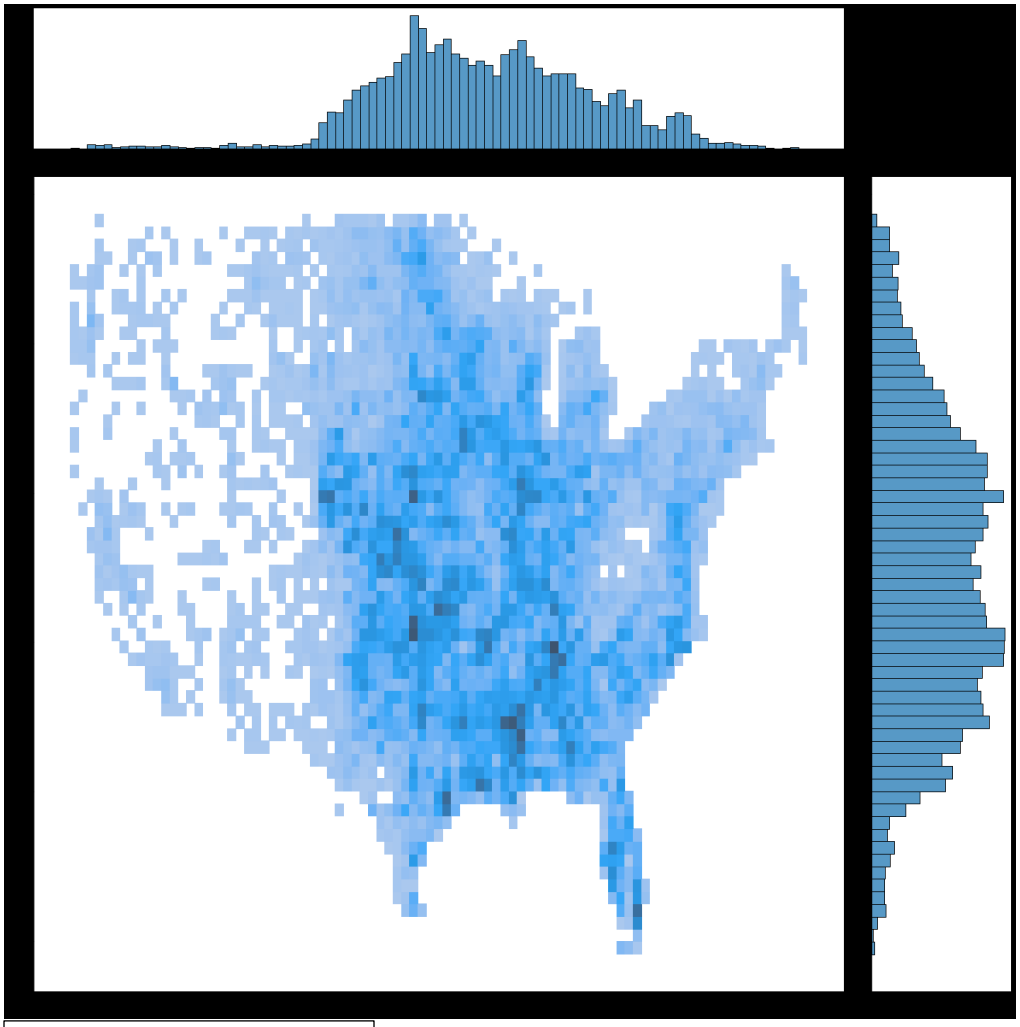
[28 0 1]]

Interestingly, the best classifier to use given the data frame was Random Forest as it had 80.81% accuracy rate for the model, whilst the worst performing model was Decision Trees with 66.28%. Another note is that these classifiers were trained without the deaths attribute, as it was the target label. Further research would be needed to see the accuracy of classifiers with records that contain deaths < 0. I believe that the accuracies would be much lower as the "deaths" attribute is not categorical. Making ranges of deaths rather than the model predicting a single number on an infinite scale would be the most likely fix.

One interesting graph we were able to plot were the locations of tornados based on their starting latitude and longitude, and their ending latitude and longitude. With these 4 values we can pinpoint the exact location on the map a tornado appeared, and compare it with the position, time, damage, or other values that could have a possible correlation to geographical location.

Before we could properly analyze any possible results through the graph, it's important to note that some of the data had to be cleaned before we could apply it to the graph. For example, our data had information from offshore states and provinces such as Alaska, Hawaii, Puerto Rico, and the US Virgin Islands. We decided to focus on tornados in the mainland united states rather include these offshore locations because it would allow us to make a more concise map using the given geographical data.

There were also null values present in the geographical data we chose to ignore rather than incorporate into the data, as we had enough data points to allow us to disregard these. It would have been difficult to incorporate them due to the fact that they are geographical locations and do not allow us to incorporate means, medians, and modes in the same manner other data might have.

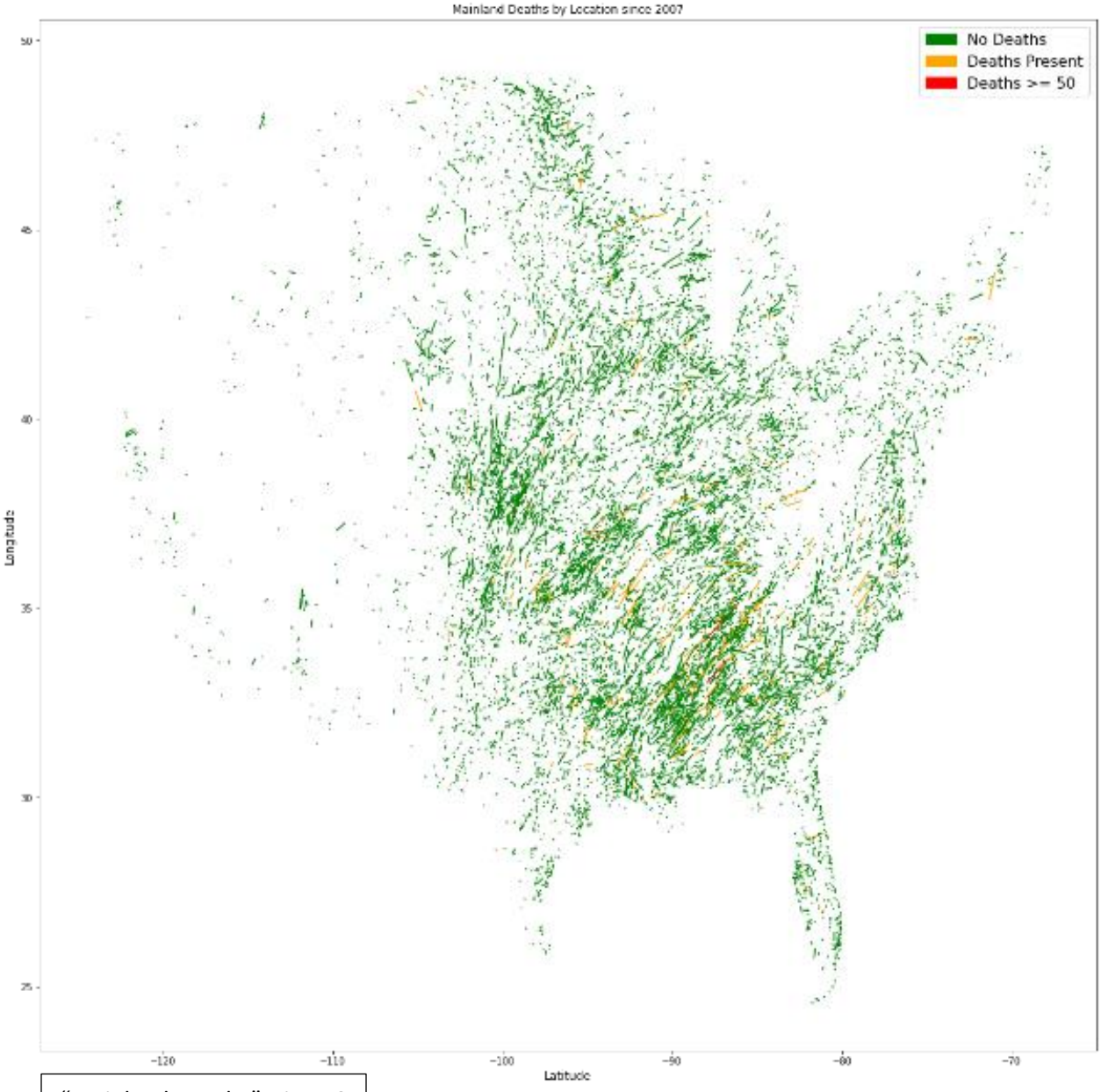


"U.S. Heatmap", Fig. 1.5

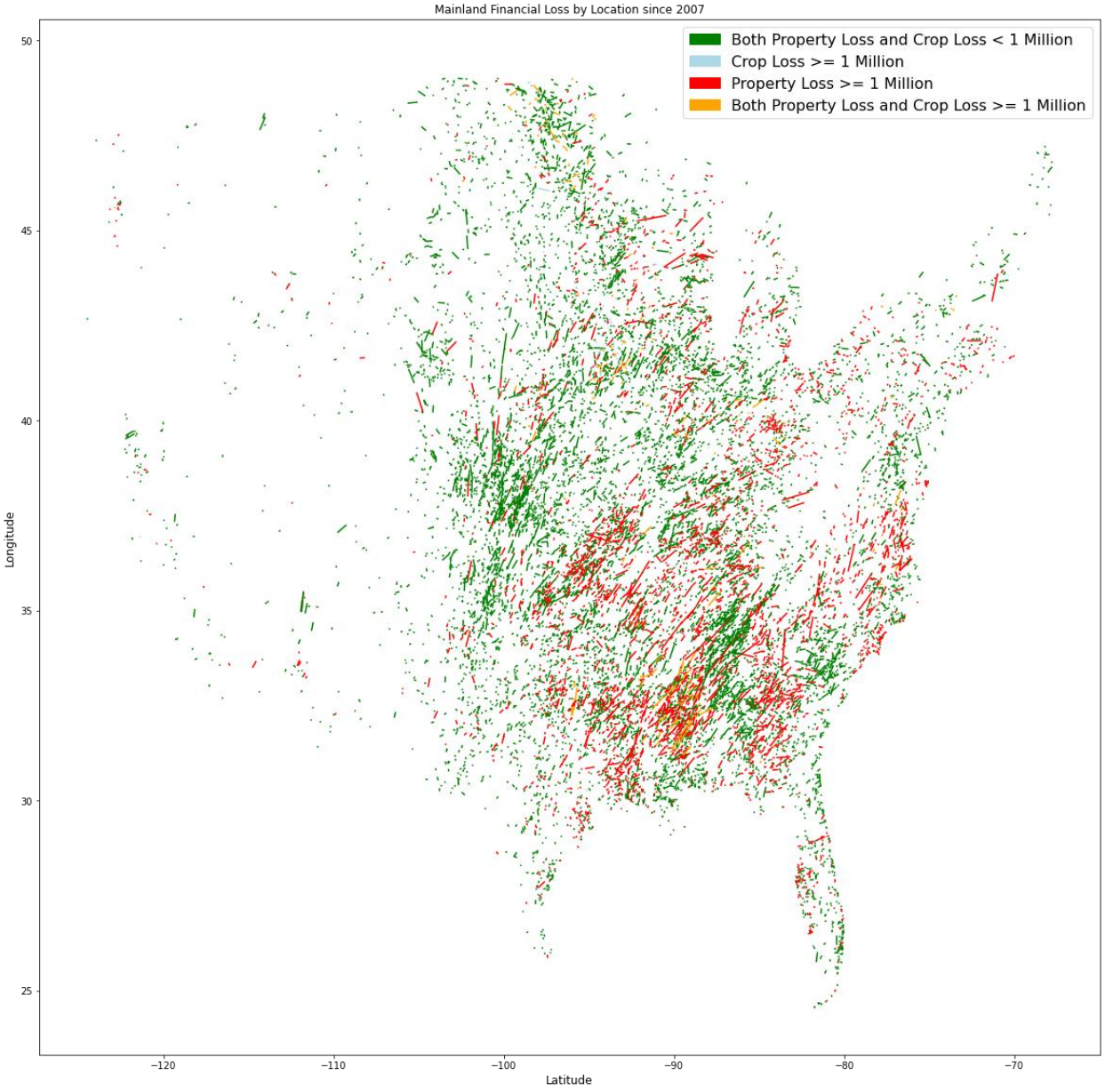
The graphic above is a joint plot of the mainland United States. The sides of the graph showcase histograms of both the starting latitude and starting longitude of tornadoes while the middle encompasses those results in a three-dimensional space create a sort of density selection of the location of tornadoes. It helps showcase areas that get larger number of tornadoes in an easy-to-understand way.

We also created a similar graph that graphs both the starting and ending positions of each tornado to better trace the path of a tornado. We were also able to change the color of the line plotted to help showcase differences between different variables in accordance with the path of the tornado.

These graphs can also be used to compare the latitude and longitude to other variables to see if there is any correlation between the geographical location and the variable in question. The results of some of our notable graphs will be shown below.



"Mainland Deaths", Fig. 1.6

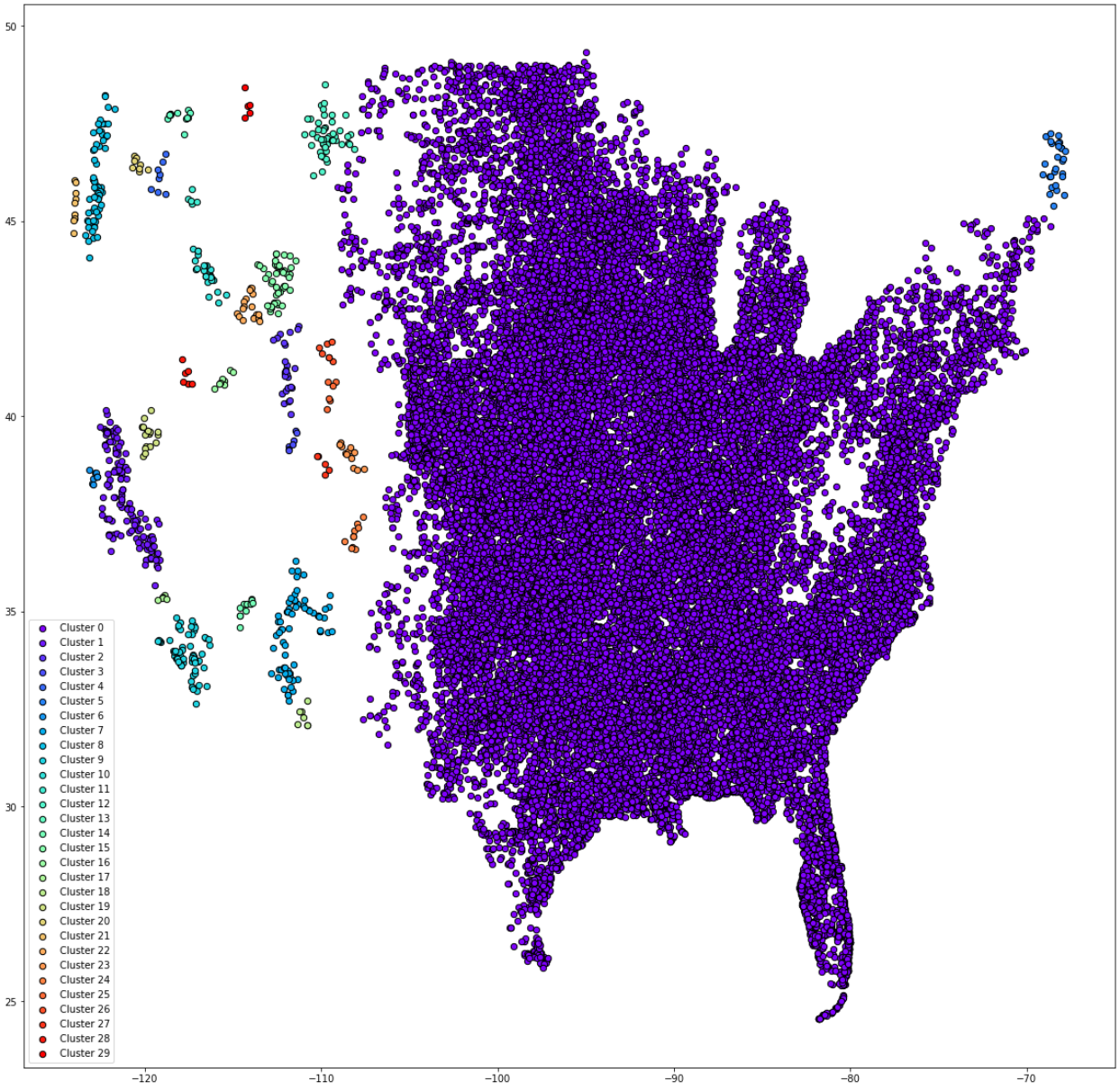


"Crop/Property Loss", Fig. 1.7

Our starting point of 2007 was chosen for a variety of reasons regarding making our graphs look readable while also showcasing some of the rarities that may have happened in the earlier years of our timeline. One important thing to mention is that the storage of crop loss was not kept track of until the year 2007, which is one of the reasons the graph starts at this location.

One final thing we were able to do with the geographical data of tornadoes is perform a density-based scan (DBSCAN) to find clusters of the United States where variables appear. The data use was the same as the data used in the joint and line plots above, but with all the data rather than starting from the year 2007. This was done because more data was present to be used,

as well as attempts in changing the amount of data provided result in a less satisfying graph considering the results collected. There is a possibility that changing the amount of data could change the results due to the possibility of overfeeding, but our methods being based on density deemed we should include all available data.



"DBSCAN Map", Fig. 1.8

The results of our graph showcase a large area of tornados being present in the eastern side of the United States, with a small gap in the Appalachian Mountains, while the western United States has smaller clusters where tornadoes appear, with probable gaps around the Rocky Mountains.

Another goal of ours was to use different classifiers to predict the amount of damage a tornado could cause. To keep our classifiers simple, our continuous value of property damage was converted into a simple Boolean for the methods to predict. The value in question was divided in the method of tornados that did over one million dollars' worth of property damage, and tornados that contributed to under 1 million dollars in property damage. This number was chosen to simplify the prediction process and was easy to implement into the data due to the data being stored in millions of dollars each tornado caused.

The attributes we chose to feed the classifier were based on prior methods to measure positive correlations between attributes as well as our assumptions and inputs regarding the subject. The 4 classifiers we chose to test were a decision tree, random forest, k-nearest-neighbor, and multi-layered perception classifier. We also chose one random forest that was created through a grid-search function to find the best parameters for creating a decision tree through a random forest method. This item is labeled "Best Random Forest".

Below are the results of the classifiers:

Decision Tree

The accuracy of the model is: 0.8283779494233414

Classification Report:

	precision	recall	f1-score	support
0	0.91	0.89	0.90	8075
1	0.42	0.45	0.43	1376
accuracy			0.83	9451
macro avg	0.66	0.67	0.67	9451
weighted avg	0.83	0.83	0.83	9451

[[7208 867]

[755 621]]

Random Forest

The accuracy of the model is: 0.8692201883398583

Classification Report:

	precision	recall	f1-score	support
0	0.87	0.99	0.93	8075
1	0.78	0.14	0.24	1376

accuracy			0.87	9451
macro avg	0.82	0.57	0.58	9451
weighted avg	0.86	0.87	0.83	9451

[[8019 56]
[1180 196]]

Best Random Forest

The accuracy of the model is: 0.8800126970690932

Classification Report:

	precision	recall	f1-score	support
0	0.89	0.98	0.93	8075
1	0.72	0.29	0.41	1376
accuracy			0.88	9451
macro avg	0.80	0.64	0.67	9451
weighted avg	0.86	0.88	0.86	9451

[[7916 159]
[975 401]]

KNN

The accuracy of the model is: 0.8605438577928262

Classification Report:

	precision	recall	f1-score	support
0	0.89	0.96	0.92	8075
1	0.54	0.30	0.39	1376
accuracy			0.86	9451
macro avg	0.71	0.63	0.65	9451
weighted avg	0.84	0.86	0.84	9451

[[7720 355]

[963 413]]

MLP

The accuracy of the model is: 0.8698550417945191

Classification Report:

	precision	recall	f1-score	support
0	0.88	0.98	0.93	8075
1	0.65	0.23	0.34	1376
accuracy			0.87	9451
macro avg	0.77	0.61	0.63	9451
weighted avg	0.85	0.87	0.84	9451

[[7902 173]

 [1057 319]]

Out of all the classifiers the optimized random forest performed the best with a score of 88% success rate. Though this doesn't necessarily make it the best classifier due to other methods having the ability to differentiate larger property damages to lower property damages. The optimized random forest was able to correctly guess 401 large property damage tornadoes while the normal decision tree was able to guess 621. Still, the normal decision tree performed the worst in regard to accuracy.

What intrigued us was that multi-layered perception performed as low as it did. Further research would indicate that there are some methods we could optimize the functions of the classifier to better encapsulate the data. While it's accuracy score of 86.98%, when we look at the accuracy matrix it reveals that it got 1057 large property damage tornados wrong, while only missing 173 lower property damage tornadoes. These results could be from a result of overfeeding or poor implementation of the classifier features, or even possibly from the classifier not being fit for our scenario we are trying to use it in.

Conclusion

In conclusion, the results of our project display the knowledge and skills achieved throughout the semester. We did achieve our initial goals of predicting death outcomes, tornado location by geographic latitude and longitude, crop loss and property loss visualizations, as well as using DBSCAN to try and cluster tornados based on location. Overall, this project allowed us to flex our data mining muscles and try to emulate what concepts will be done in a data oriented career.

References

D. Hunter, "Matplotlib: A 2D Graphics Environment", Computing in Science & Engineering, vol. 9, no. 3, pp. 90-95, 2007.

Harris, C.R., Millman, K.J., van der Walt, S.J. et al. *Array programming with NumPy*. Nature 585, 357–362 (2020). DOI: 10.1038/s41586-020-2649-2.

Jeff Reback, Wes McKinney, jbrockmendel, Joris Van den Bossche, Tom Augspurger, Phillip Cloud, gfyoung, Sinhrks, Adam Klein, Matthew Roeschke, Simon Hawkins, Jeff Tratner, Chang She, William Ayd, Terji Petersen, Marc Garcia, Jeremy Schendel, Andy Hayden, MomIsBestFriend, … Mortada Mehyar. (2020). pandas-dev/pandas: Pandas 1.0.3 (v1.0.3). Zenodo. https://doi.org/10.5281/zenodo.3715232

Scikit-learn: Machine Learning in Python, Pedregosa *et al.*, JMLR 12, pp. 2825-2830, 2011.

Waskom, M. L., (2021). seaborn: statistical data visualization. Journal of Open Source Software, 6(60), 3021, https://doi.org/10.21105/joss.03021