PREDICTING TORNADO DANGER LEVELS

NYU CENTER FOR DATA SCIENCE ♦ DS-GA 1001 ♦ FALL 2018

ZANE DENNIS
SREE GOVINDAPRASAD
ORION TAYLOR

I. INTRODUCTION & BUSINESS UNDERSTANDING

Tornadoes often destroy buildings, burying people alive and necessitating a time-sensitive search for victims in need of urgent medical care. One the most crucial aspects of emergency response to these severe weather events, therefore, is search and rescue (SAR). For example, in the aftermath of the deadly 2011 Joplin, Missouri tornado, "emergency crews drilled through concrete at a ruined Home Depot, making peepholes in the rubble in hopes of finding lost shoppers and employees."[1]

As such it is essential to deploy SAR teams as soon as possible after an incident so that victims can receive medical attention sooner. This duty is often carried out by firefighters. A 2011 article by Mike Walker of the Oklahoma City Fire Department describes some of the logistical challenges these teams face:

> "When a major tornado occurs, the tendency is to send every resource in the jurisdiction to the area. If allowed, this practice will create multiple problems. Too many responders can be worse than too few, because it leaves the remainder of the jurisdiction without emergency services and accountability of the excess responders can become problematic. Just like any other incident, responders should not respond until requested."[2]

This leaves us with the questions of when, where and to what extent to request SAR teams. Current measures of tornado intensity — most notably the Enhanced Fujita (EF) Scale — focus solely on the strength of a tornado and how dangerous it might be to a given structure. To our knowledge, however, there is no standard measure of the threat level a tornado poses to a given populated area. We aim to solve this problem by building a model to classify tornadoes into threat levels based on the number of injuries and deaths they would be expected to cause.

According to Walker, current best practices for assessing the necessity of SAR rely on 911 calls and manual reconnaissance, but these methods can only be used "once the tornado leaves the area." With our solution, we make predictions using data captured in real-time as the tornado is over the target area. While this sort of model can never be as accurate as manually checking the affected areas, being able to approximate damage levels while the storm is still occurring will allow emergency coordinators to prepare the appropriate responses before necessary rather than after. This capability would enable SAR teams to begin finding victims sooner, saving lives.

Hospitals and other medical institutions that treat victims could also use our model. Tornadoes and other severe weather events can cause what the St. Francis Health System of Oklahoma's Emergency Operations Plan defines as a Surge Event: "the arrival of a large number of individuals from an internal or external event, such as the result of a fire, explosion, train wreck, or

[1] Younker, Emily. The Joplin Globe. https://tinyurl.com/yc9zcb7m. 25 May 2011. 6 Dec 2018.
[2] Walker, Mike. Fire Rescue Magazine. https://tinyurl.com/y7w8f23o. 1 May 2011. 6 Dec 2018.

bioterrorism event that may require treatment."[3] These Surge Events can force hospitals to quickly acquire additional resources and sometimes even call in extra staff. Our model would enable them to begin their emergency planning even earlier, helping them prepare to better handle the patient influx.

II. DATA UNDERSTANDING

The NOAA Storm Events Database[4] includes storm events data for all years from 1950 to present. Given the inconsistent availability of county demographic data (discussed below), and in order to limit the effects of potential changes in collection methodology or other linear time fluctuation, we narrowed our scope of interest to 2008-2018, filtering for tornadoes to reduce our merged dataset to n=14634 events.

With this task complete, we joined supplementary data on median household income from the 2016 Small Area Income and Poverty Estimates dataset[5] and population data from the 2010 U.S. Census[6] by state and county to obtain a complete picture of tornado- and location-characteristic data for each event.

A preliminary assessment of the distribution of tornadoes by casualties (derivation discussed in the Data Preparation section) showed an imbalanced dataset; this had downstream implications for sampling (also discussed below).

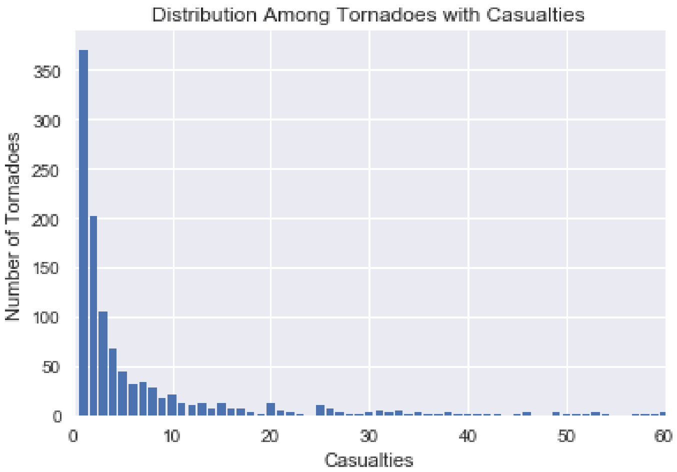


Figure 1: Distribution of casualties across tornadoes

[3] Permission not given to share complete document.

[4] NOAA. Index of /pub/data/swdi/stormevents/csvfiles. https://tinyurl.com/yb69tbdc. 6 Dec 2018.

[5] United States Census Bureau. SAIPE State and County Estimates for 2016. https://tinyurl.com/yd8th3qr. 6 Dec 2018.

[6] United States Census Bureau. County Population by Characteristics: 2010-2017. https://tinyurl.com/y823z8jy. 6 Dec 2018.

We chose to treat each tornado-county event as an independent instance. In reality, tornadoes occasionally traverse multiple counties (and even states), and there are likely some geophysical characteristics that correlate destructiveness across locales for a given tornado, but given our inclusion of demographic information that might vary widely from one county to the next, it was logical to treat these tornadoes piecewise.

III. DATA PREPARATION

Our final feature set could be grouped roughly into four topics: tornado time, trajectory and intensity information, and county demographic information. Target variable derivation and subsequent data balancing are also discussed below.

Time Data

To capture the cyclicality of certain time aspects of each tornado event, we used trigonometric functions to generate sine- and cosine-based coordinates. This transformation was applied to both time of day and day of the year. We also derived a binary feature indicating whether the incident occurred on a weekday or on the weekend, in hopes of capturing information about changes in population concentration or behavior. Finally, we derived a 'Duration' feature, which would be tracked as a tornado is ongoing to update the model in real time.

Trajectory Data

The NOAA data included beginning and ending latitude and longitude coordinates and beginning and ending range values (measures of distance from the nearest "particular village/city, airport, or inland lake, providing that the reference point is documented in the Storm Data software location database"[7]). From these we derived average latitude and longitude and average and minimum range, the latter pair in hopes of capturing the relation between casualties and population proximity in greater detail.

Intensity Data

Current forecasting capacity for tornadoes is limited by the brevity of their formation process and their sheer force (which makes it difficult to get close enough to secure accurate measurements). As such, 'Magnitude' (measured as wind speed) and other data available for some storm events in the NOAA database are noticeably absent for tornadoes. Some of the features that are available (e.g., EF Scale, length, width) present data leakage risk; in particular the EF Scale, which

[7] NOAA. Index of /pub/data/swdi/stormevents/csvfiles. https://tinyurl.com/yb69tbdc. 6 Dec 2018.

categorizes tornadoes by estimated wind speed, is measured through post-event surveillance of affected areas (assigning potential wind speeds retroactively based on observed damage).[8]

Nevertheless, after reviewing previous studies and consulting with Dr. Chris Porter, Radar Team Lead at Norman, Oklahoma-based Weather Decision Technologies/DTN, we concluded that a derivative intensity feature was viable as an anterior metric. The NOAA's Storm Prediction Center, Dr. Porter writes, may use convective outlooks to "try to forecast severity and include language in their outlooks addressing number of tornadoes (if expecting an outbreak or not) and if tornadoes are expected to be violent (higher on the EF scale) or on the lower-end (EF-0, EF-1)." Accordingly, we de-granularized EF Scale up to a binary feature, approximating this "violent" vs. "lower-end" dynamic.

Similarly, we found that techniques currently in development — including rotation-track mosaics produced using radar data — are able to provide near-real-time analysis and forecasting of tornado path-length and width. While these technologies are currently crude estimators, we note that future refinement may strengthen their informative value for preemptive models.[9] As such we included a 'Tornado Area' feature derived from length and width measurements.

A final method we employed, borrowed from a July 2018 study that used similar data to predict tornado financial damage[10], was to featurize an 'Event Narrative' column by mining its text for variations of the term 'multivortex' (indicative of the outbreaks referenced above), and creating a binary indicator (we found this property in only ~75 events).

County Demographic Data

These features were relatively straightforward, and required little engineering beyond normalization. Median household income was measured at the county level, as were population and housing densities. We derived one additional feature 'Percent Land' (vs. water) by dividing land area by total area for each county.

Target Variable & Data Balancing

To derive our target variable, we first chose to sum injuries and deaths (both direct and indirect) as 'Casualties' and then binned tornadoes into three groups by number of casualties: Class 0 (0 casualties), Class 1 (1-19 casualties) and Class 2 (20+ casualties), to distinguish tornadoes that necessitate substantial emergency responses, small responses or no response. To address our class

[8] Enhanced F Scale for Tornado Damage. https://tinyurl.com/yasj58ba. 6 Dec 2018.

[9] Porter, Chris "Re: Tornado Prediction." 19 Nov 2018. E-mail.

[10] Diaz, Jeremy and Maxwell Joseph. Predicting property damage from tornadoes with deep learning. https://tinyurl.com/ydgvnzpt. 6 Dec 2018.

imbalance, we applied the *SMOTE* (Synthetic Minority Over-Sampling Technique) algorithm to our training data.

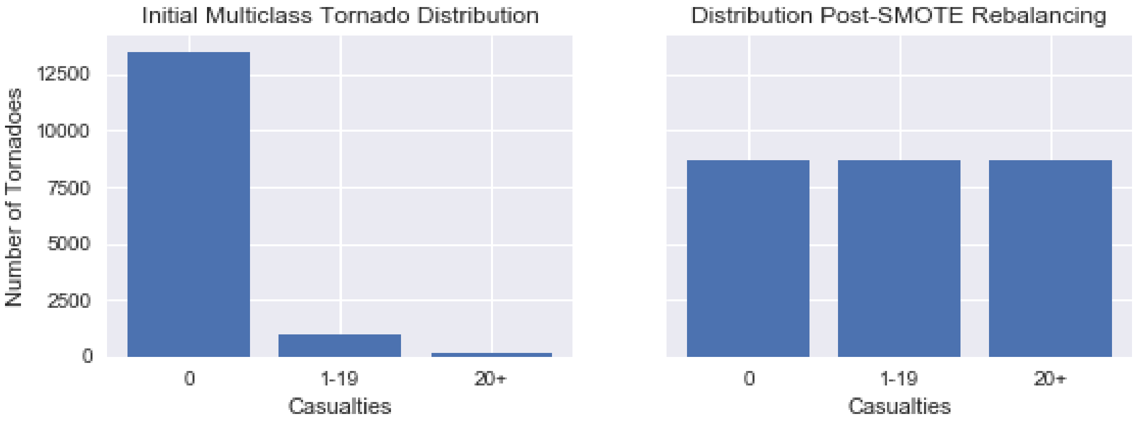


Figure 2: Casualties class distribution before and after SMOTE balancing

SMOTE operates "by taking each minority class sample and introducing synthetic examples along the line segments joining any/all of the *k* minority class nearest neighbors," randomly chosen depending on the oversampling requirement.[11]

A quick comparison of confusion matrices for an untuned Logistic Regression with and without balancing justifies the use of SMOTE: without balancing, the model trains poorly against the minority classes, correctly predicting harmless tornadoes in ~99%+ of cases but only predicting moderate and severe tornadoes in ~9% and ~10% of cases respectively (untenable in our use case); after applying SMOTE the model's recognition of severe tornadoes improved dramatically, correctly identifying ~86% of validation-set Class 2 tornadoes. While its performance on Class 1 tornadoes improved, we continued to find misclassification in this range problematic (more information in the following section). Most importantly, the false negative percentage of severe tornadoes predicted to be benign shrank from ~62% to 0. Given the potential loss of life avoided by eliminating this misclassification from our model, this is a very good indicator of the general value of this exercise from a safety standpoint.

[11] Chawla, Nitesh et al. SMOTE: Synthetic Minority Over-sampling Technique. Jun 2002. https://tinyurl.com/ybvub4v5. 6 Dec 2018.

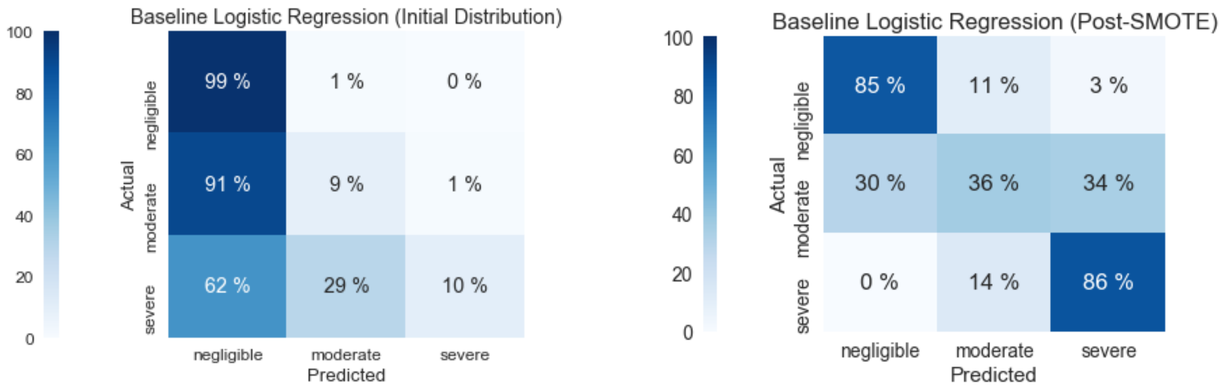


Figure 3: Confusion matrices from Logistic Regression with and without SMOTE balancing

Given the size of our training set and the reasonably small volume of available features, we opted to determine feature importances for exploratory purposes only, using an untuned Decision Tree. Only three of our features (all related to the tornadoes' scale and path) returned Gini importances >10%; conversely, only one feature (the derived multi-vortex indicator) was immediately discardable.

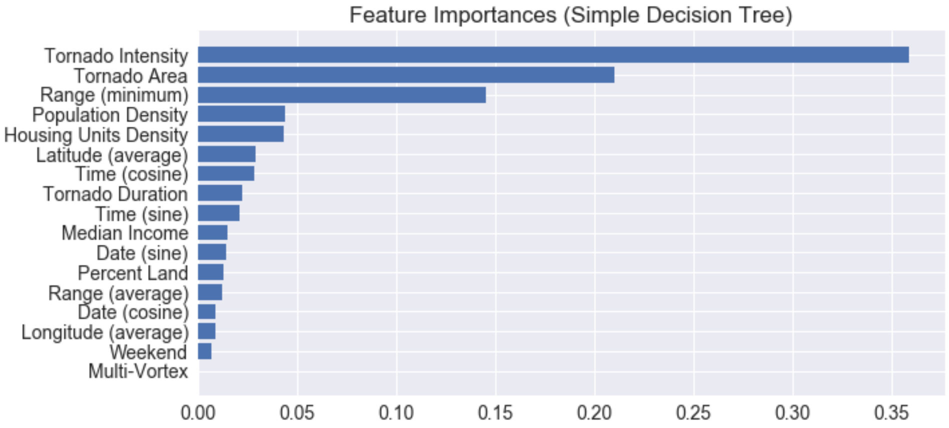


Figure 4: Feature importances from Decision Tree model

IV. MODELING & EVALUATION

Choice of Evaluation Metric

We chose recall score and confusion matrix as our evaluation metrics for this three-class classification problem. We could not rely on accuracy score as an evaluation metric because it averages true positive rates across classes and thus doesn't provide insight about predictions for minority classes. For example, our baseline Logistic Regression model achieved an accuracy score of 0.92 prior to SMOTE balancing and an accuracy of 0.82 after using SMOTE; however, as seen in Section III Figure 3, it is clear that the post-SMOTE Logistic Regression model is much better at predicting Classes 1 and 2.

Recall, Confusion Matrix and the Business Problem

We use the confusion matrix and recall value as evaluation metrics to select the best models for each algorithm and to make comparisons of the performance of different algorithms. Since lives could be at stake, we use models that are less likely to underestimate tornado damage.

Given that this is a three-class problem, in the case of Class 1, for example, a false negative prediction could be an instance that is moderate (Class 1) predicted as being negligible (Class 0) or severe (Class 2). A false positive prediction could be an instance that is negligible (Class 0) or severe (Class 2) predicted as being moderate (Class 1). Hence, both false positives and false negatives could underestimate the level of damage and have to be considered. Using the confusion matrix and recall value enables us to do that.

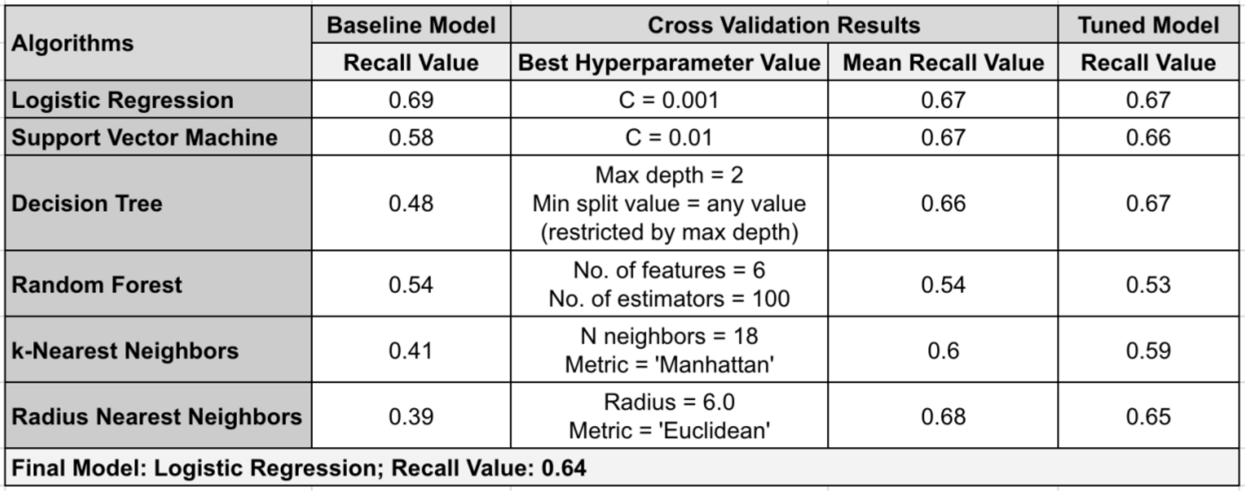
Baseline Model Performance

We chose Logistic Regression as our baseline model because it is robust on small sample sizes and imbalanced classes. Our baseline model had a recall score of 0.69.

Model Choice, Cross Validation and Hyperparameter Tuning

In addition to Logistic Regression, we train SVM, k-Nearest Neighbors, Radius Nearest Neighbors, Decision Tree and Random Forest models. To improve upon the baseline, we perform hyperparameter tuning for each model to prevent overfitting. We use stratified 3-fold cross-validation to select the best hyperparameter values (by highest mean recall). We set a seed random state on the folds to reduce noise in our results.

Results



Algorithms	Baseline Model	Cross Validation Results		Tuned Model
	Recall Value	Best Hyperparameter Value	Mean Recall Value	Recall Value
Logistic Regression	0.69	C = 0.001	0.67	0.67
Support Vector Machine	0.58	C = 0.01	0.67	0.66
Decision Tree	0.48	Max depth = 2 Min split value = any value (restricted by max depth)	0.66	0.67
Random Forest	0.54	No. of features = 6 No. of estimators = 100	0.54	0.53
k-Nearest Neighbors	0.41	N neighbors = 18 Metric = 'Manhattan'	0.6	0.59

Radius Nearest Neighbors	0.39	Radius = 6.0
		Metric = 'Euclidean'
			0.68	0.65

Final Model: Logistic Regression; Recall Value: 0.64				

Figure 5: Baseline models, cross validation results, tuned models and final model for deployment

Our key findings are summarized as follows:

- *SVM* — we did not find SVM to be an improvement over our Logistic Regression models
- *k-Nearest Neighbors* — there was relatively little variation in the cross-validation scores of kNN, regardless of distance metric (Euclidean or Manhattan); attempts at filtering feature inclusion by importance only made results worse
- *Radius Nearest Neighbors* — determining neighbors by radius (rather than k) was more successful, and an optimum was found at radius=6.0 with Euclidean distance; attempts at filtering feature inclusion by importance only made results worse
- *Decision Tree* — a Decision Tree did not perform significantly better than the baseline Logistic Regression; going deeper than a max depth of 2 did not improve model performance (this could indicate that a small subset of features affected the predictions)
- *Random Forest* — Random Forest fits a number of Decision Tree classifiers and aggregates the results; this model did not perform well, probably due to presence of few dominant features in the model (this is supported by the feature importances graph in Section III Figure 4)

The recall scores for Logistic Regression, Support Vector Machine and Decision Tree were all similar. As underestimation of Class 2 tornadoes carries the greatest consequences, we chose Logistic Regression for our final model because of its superiority in predicting Class 2 tornadoes (81%).

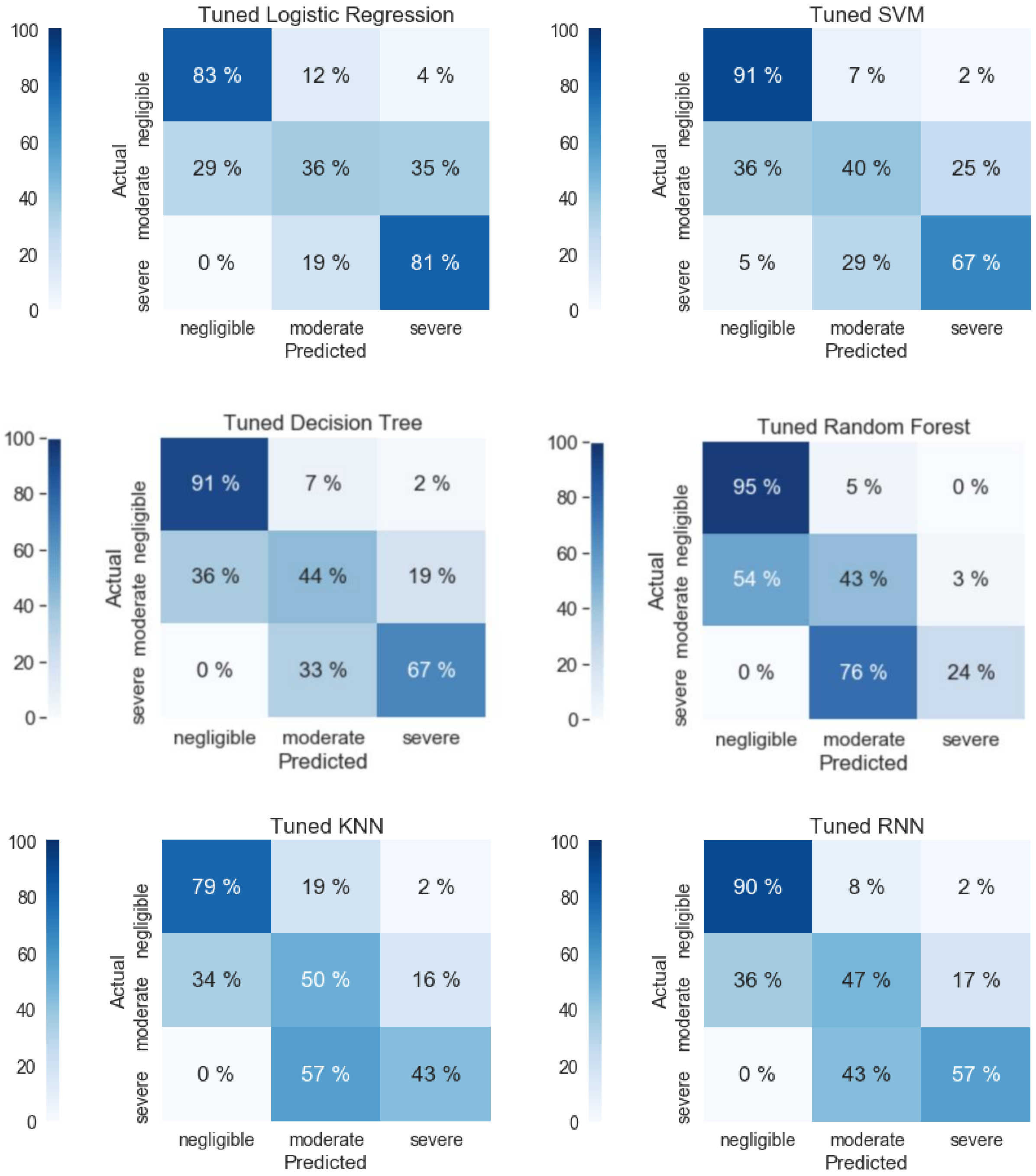


Figure 6: Confusion matrices generated from different algorithms

We combined the training and validation sets and used Logistic Regression with C=0.001 to make a prediction on our unseen test set using our best model. Since our Logistic Regression model produced consistent results and did well on predicting Class 2 damage, we decided that this model was suitable for deployment.

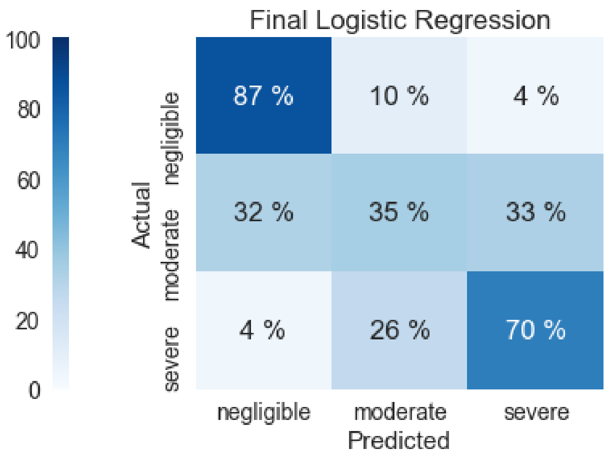


Figure 7: Final result from Logistic Regression model

V. DEPLOYMENT & CONCLUSION

Our model would best be deployed by meteorologists with direct access to real-time data about an ongoing tornado. They would pre-establish channels of communication with hospitals, fire departments, EMS and other local authorities involved in disaster response. The meteorologists would then discuss the output of the model with those authorities to decide upon course of action. The model would then be updated and possibly provide new outputs as the situation changes.

In its current state, our model should be considered a proof-of-concept. While its ability to distinguish between harmless and severe tornadoes is encouraging, the poor scores with regards to moderate tornadoes prevent it from being of high enough quality for immediate deployment. Additionally, many of the features used in our model are ex post facto measurements which might differ slightly from the imperfect measurements determined by the rotation-track mosaics discussed by Dr. Porter.

However, we believe that performance can still be greatly improved. As a new technology, rotation-track mosaics will only continue to improve, assuaging that concern. One way to improve the model might be additional feature engineering; our feature importances show that data such as population density are useful, but the data we used only represent county-level measurements. Future studies might aim to use more exact measurements of the population in a tornado’s crosshairs.

VI. CONTRIBUTIONS

- Zane: modeling (K Nearest Neighbors, Radius Nearest Neighbors), writeup (Business Understanding, Deployment)

- Sree: data preparation (with Orion), data balancing, modeling (Decision Tree, Random Forest), writeup (Modeling, Evaluation)
- Orion: data merging, data preparation (with Sree), modeling (Logistic Regression, Support Vector Machine), writeup (Data Understanding, Data Preparation)