# DETERMINING IF AN ACCIDENT IS FATAL OR NOT

## A COMPARISON OF DIFFERENT ALGORITHMS

**OBJECTIVE**

- To implement our own Decision Tree algorithm.
- To use the Sklearn library to implement a Decision Tree, an AdaBoost Decision Tree, a Random Forest, a K Nearest Neighbors, a Decision Tree using undersampling, a Decision Tree using oversampling, an AdaBoost Random Forest, and a Voting Classifier algorithm.
- To use the above algorithms to predict if an accident will be fatal or slight based off of the values of the provided features.
- To calculate a variety of metrics to compare the performance of all the different algorithms that we implemented.

**HYPOTHESIS**

- More predictions would be classified as slight than fatal due to the data containing 25% fatal and 75% slight classifications.
- We predicted that the Random Forest algorithm would perform best.

**DATASET**

- Used data found on Kaggle of Road Accidents in U.K. that classified accidents as either slight or fatal depending on the values of the provided features.
- Used Ordinal Encoder to convert strings in the dataset to integer values so that our algorithms could process the data.
- There are 33 features including date, time, longitude, and latitude.

**METHODS**

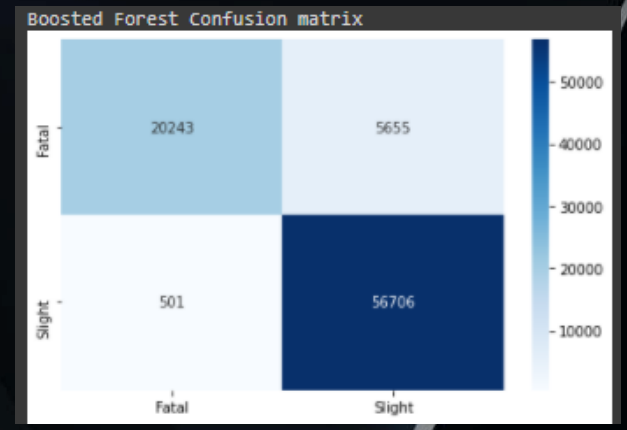
- The AdaBoost Decision Tree performed best with a learning rate of 0.25.
- Used K=5 for KNN algorithm (defult value).
- Used the ensemble method for the Voting Classifier.
- Random Forest performed best with number of estimators = 100.
- Metrics: Accuracy, Precision, Recall, Precision/Recall Curve, and F1 Score.
- Used SKlearn.metrics to calculate the different metrics.

**PRELIMINARY RESULTS**

- Preliminary results show that the AdaBoost Random Forest has the best performing accuracy at 92.6%.
- The Decision Tree accuracy did not improve much by adding a boost with and without a learning rate.
- The Decision Tree accuracy did improve significantly by over and undersampling. Oversampling performed slightly better at 91.0% accuracy while undersampling was only at 90.9% accuracy.

**FUTURE GOALS**

- Optimize the K value to find the best results for KNN algorithm.
- While our algorithms are performing around a 90% accuracy rate, we would like to improve these results.
- We plan on adding cross-validation as another way to assess the performance of the algorithms.
- Our Precision/Recall Curve does not seem to be outputting correct results, so we plan on revisiting that.




| | Acc | Prec | Rec | F1 |
|---|---|---|---|---|
| Our DT | | | | |
| Decision Tree | 89.39 | 0.9254 | 0.9338 | 0.9296 |
| AdaBoost DT | 89.42 | 0.9259 | 0.9345 | 0.9299 |
| Random Forest | 92.59 | 0.9917 | 0.9089 | 0.9485 |
| KNN | 84.58 | 0.8917 | 0.9044 | 0.8980 |
| Under DT | 90.86 | 0.9635 | 0.9128 | 0.9375 |
| Over DT | 91.02 | 0.9657 | 0.9127 | 0.9385 |
| AdaBoost RF | 92.59 | 0.9912 | 0.9093 | 0.9485 |
| Voting Class. | 90.51 | 0.9276 | 0.9277 | 0.9362 |