Group 4 - Killer Plot: Decision Tree Heat Map

Jian Ruan, Zach Yu, Carrie Hashimoto, Sophia Lyu (Undergraduate), Patrick Yee (Graduate)

April 11, 2023

1 Introduction

Through previous drafts, we have identified several contributing factors for NYC car crashes with COVID-19 as a turning point. However, data analysis is limited in a way that it only captures patterns in the past. As a result, we decide to **do machine learning on the primary and secondary datasets to predict future accidents given the past pattern**.

Our first round of training includes variables like weather, time of the day, and mentioning of COVID in NYT articles. These variables are **not comprehensive** for car crashes related factors, but they give us a **starting point** to generate predictions so that police officers can take proactive measures to prevent them. Ultimately, ML on NYC car crashes can lead to improved traffic safety and a reduction in injuries and fatalities on the road.

2 Inspiration: Gradient Boosting

We use **Gradient Boosting** as the algorithm for ML training. It involves the iterative development of multiple weak models to create a stronger, more accurate model. In gradient boosting, a model is created by taking a base algorithm, such as decision trees, and sequentially fitting new models to the errors made by the previous models. This process involves optimizing a loss function, such as mean squared error, and adjusting the weights of the data points to minimize the error in the predictions. The final model is an aggregation of all the weak models, each one improving the accuracy of the previous models. Gradient boosting is a popular technique for machine learning because it can handle large datasets and is relatively robust to overfitting, a common problem in other machine learning techniques.

The inspiration for the final killer plot comes from **the need to communicate complex ML algorithm in a simple and visually appealing way**. It is important to visualize the decision process of a Gradient Boosting model for several reasons.

First, visualizing the decision process can help us understand **how the model is making predictions** and **which features are most important** in the model. This can help us identify **potential biases or errors** in the model, and it can also help us explain the model to others who may not be familiar with machine learning techniques.

Second, visualization can help us identify any **potential overfitting or underfitting** in the model. Overfitting occurs when the model is too complex and fits the training data too closely, while underfitting occurs when the model is too simple and fails to capture the complexity of the data. By visualizing the decision process, we can see how well the model is fitting the data and whether it is generalizing well to new data.

Finally, visualizing the decision process can help us **optimize the model** by identifying areas where it can be improved. By understanding how the model is making predictions, we can identify areas where we can add more features or data, or where we may need to adjust the model parameters to improve its performance. Overall, visualization is an important tool for understanding, improving, and communicating the results of

Gradient Boosting models.

3 Common Parameters And Metrics For Gradient Boosting

In Gradient Boosting, some of the common hyperparameters and evaluation metrics are:

eta (learning rate): This is a hyperparameter that controls the step size at each iteration of the gradient boosting algorithm. A smaller learning rate may result in better performance but at the cost of slower convergence.

N (number of boosting iterations): This is the number of trees that will be built during the training process. A higher number of trees may lead to better performance but also increases the risk of overfitting.

MAD (mean absolute deviation): This is an evaluation metric that measures the average absolute difference between the predicted values and the true values.

RMSE (root mean squared error): This is another evaluation metric that measures the square root of the average squared difference between the predicted values and the true values.

labels: These are the target variables or labels that the model is trying to predict.

num.leaves: This is another hyperparameter that determines the maximum number of leaves or terminal nodes in each tree. Increasing this parameter may result in higher model capacity, but also increases the risk of overfitting.

4 Killer Plot - Decision Tree Heat Map

The killer plot is consited of two part: heat map and decision trees.

The heat map visualizes the accuracy of ML prediction. The less MAD (mean absolute deviation), the greener a pixel. The greenest area represents the highest accuracy.

The decision tree visualizes the decision making process of the Gradient Boosting algorithm. The tree starts from the bottom node and grows upward with more nodes added. We choose depth = 3 to ensure enough information is presented while keeping the graph clean and easy to read.

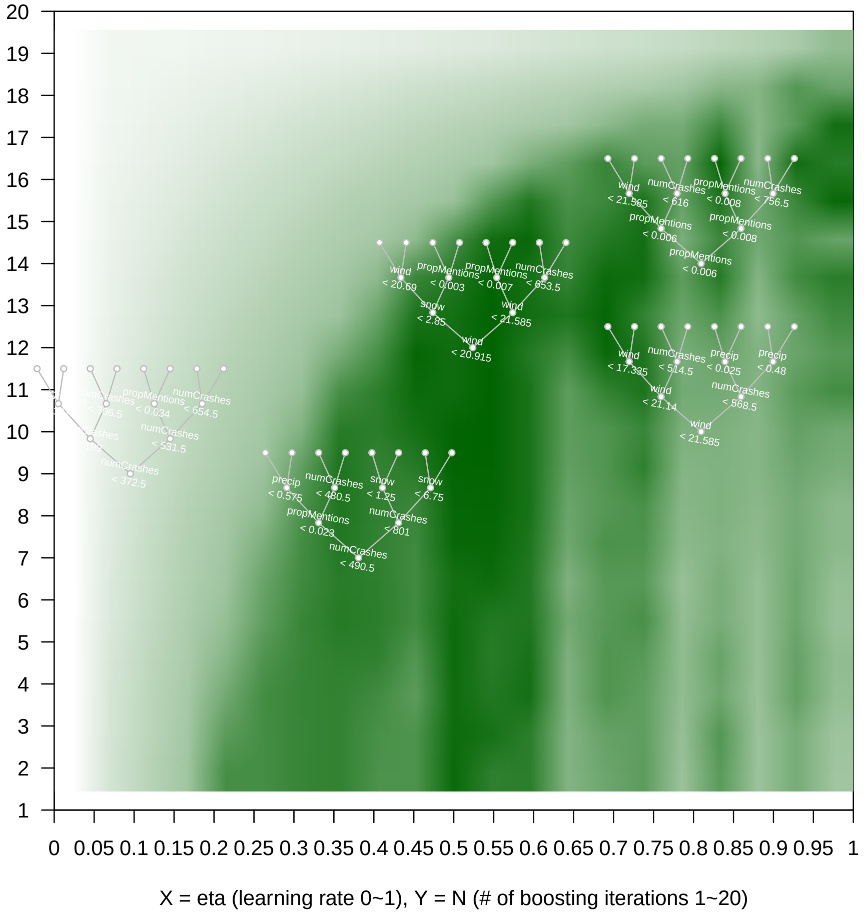

Figure 1: Decision Tree Map