Ye Yint Win (A16688105)

Final Project Report

I. Introduction

According to the CDC, one of the leading causes of fatality in the United States is due to car accidents. In fact, in 2020 there were over 35,000 deaths as a result of car accidents in the United States. Moreover, there were over 1,500,000 people who suffered injuries, and over 3,000,000 people who suffered damage of property.[1] Likewise, car accidents usually cause major traffic delays. These delays can not only be a health hazard for people requiring medical attention, but they can also decrease the overall productivity of Americans in getting to work and doing other societal tasks. There is no doubt that car accidents play a major role in the overall safety and welfare of citizens in the U.S.. Because of this, I am motivated to tackle some key problems regarding this troubling statistics. What conditions play a major role in the severity of car accidents? From weather to the type of roads severe accidents occur on, I want to address the predictive capability of certain conditions against the severity of a car accident. By understanding what conditions pertain to the severity of a car accident, I can take steps to ensure safe driving that increases the health and welfare of Americans.

The dataset I will use to perform my analyses consists of data collected from 49 states across the U.S. from February 2016 until March 2023. The dataset uses data gathered from multiple APIs that report on traffic accidents and includes data gathered from the US and state departments of transportation, law enforcement agencies, traffic cameras, and traffic sensors across road networks in the United States. The data consists of approximately 7.7 million

[1] https://www.forbes.com/advisor/legal/car-accident-statistics/#fatal_accident_statistics_section

accidents records. The features of this dataset include weather conditions, latitude and longitude of the accidents, start/end time of accident, wind conditions, and severity of the accident. The severity labels were broken down into four categories with severity 4 being the worst impact on traffic, which can also be translated as a potentially more fatal accident, and severity 1 being the least impact on traffic and commonly minor accidents with little to no injuries. Since I have labels of severity given certain conditions(my predictors) I will use this to test my hypotheses.

One of my hypotheses is that harsh weather conditions will cause more severe car accidents. I will test this by evaluating the correlation between severity and weather conditions such as temperature, wind speed, visibility and precipitation. Then I will also try to emulate a model that predicts severity given conditions that I find impact severity. I will evaluate my model using the appropriate error metrics to see if my features like wind speed help correctly predict the severity of a car accident. Of course, I will evaluate the nature of car accidents using many more predictors including location, atmospheric conditions, and traffic conditions.

II. Methods

The data analysis approach I am taking includes both inference and prediction. Because I am interested in the conditions that lead to more severe accidents, I want to evaluate which features of my data set impact severity the most. I will then evaluate if these features are enough to correctly predict the severity of a car accident by building a classification model.

Due to the large nature of my dataset and given my available computing resources, I decided to narrow my approach of analyzing car accidents in the United States to analyzing car accidents in the State of California. Apart from computing resource limitations, I decided to only observe car accidents in California due to California being the state with the most accidents in the U.S. followed by Florida with only half of California's (Fig 1 and Fig 2).

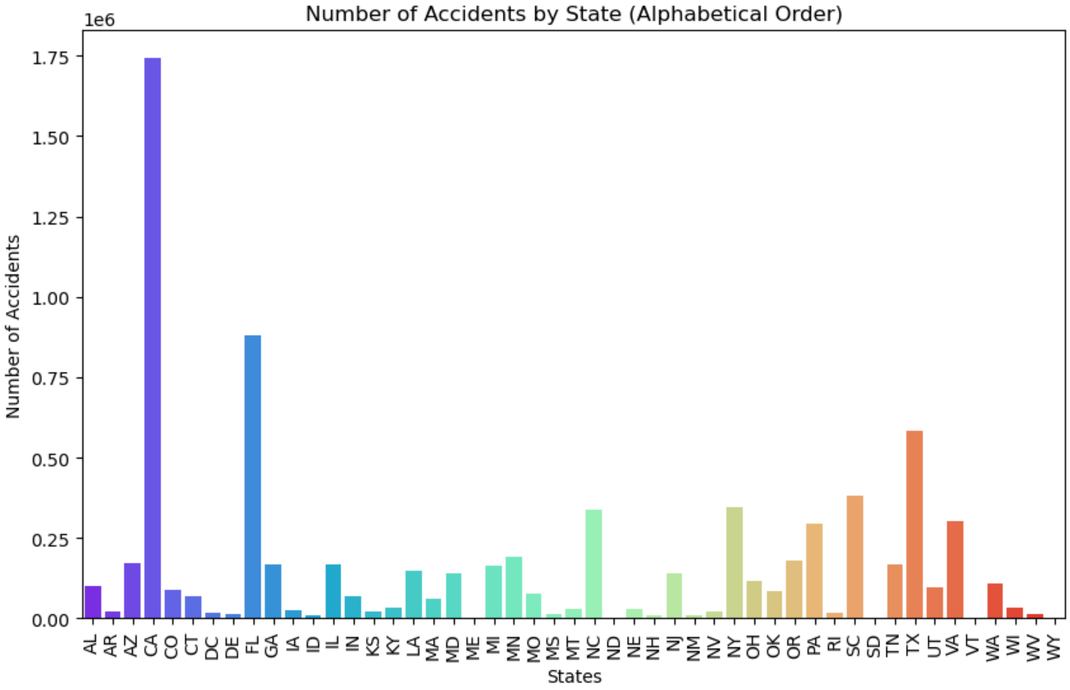


Fig 1: Number of Accidents by State

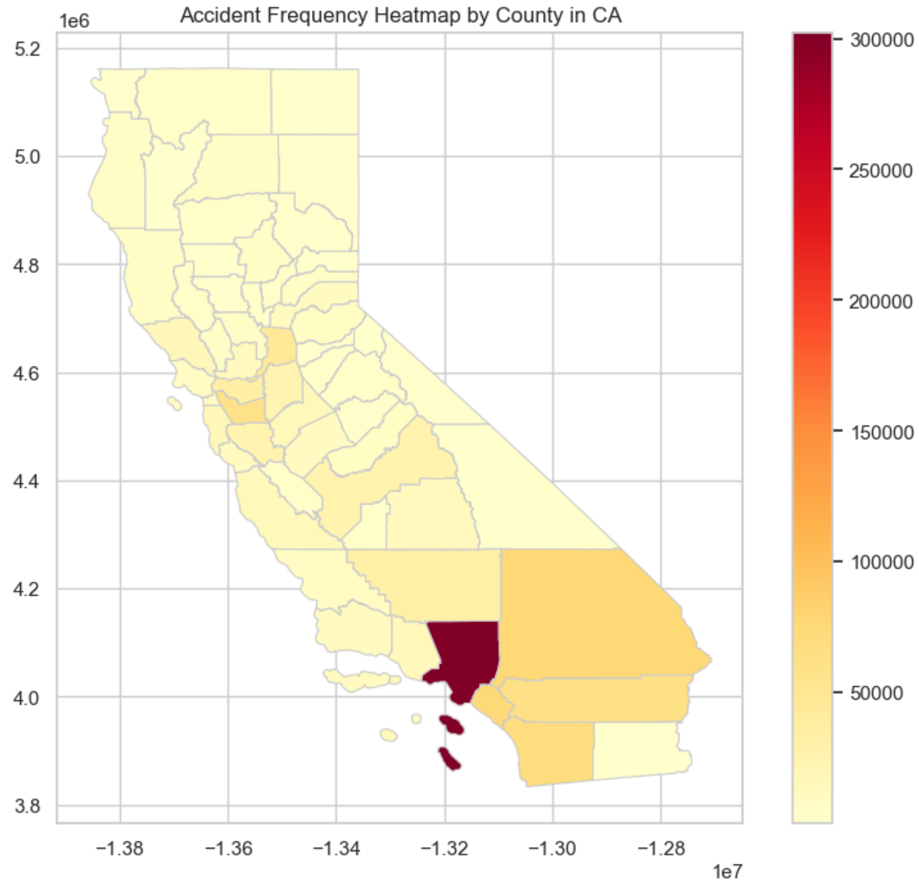
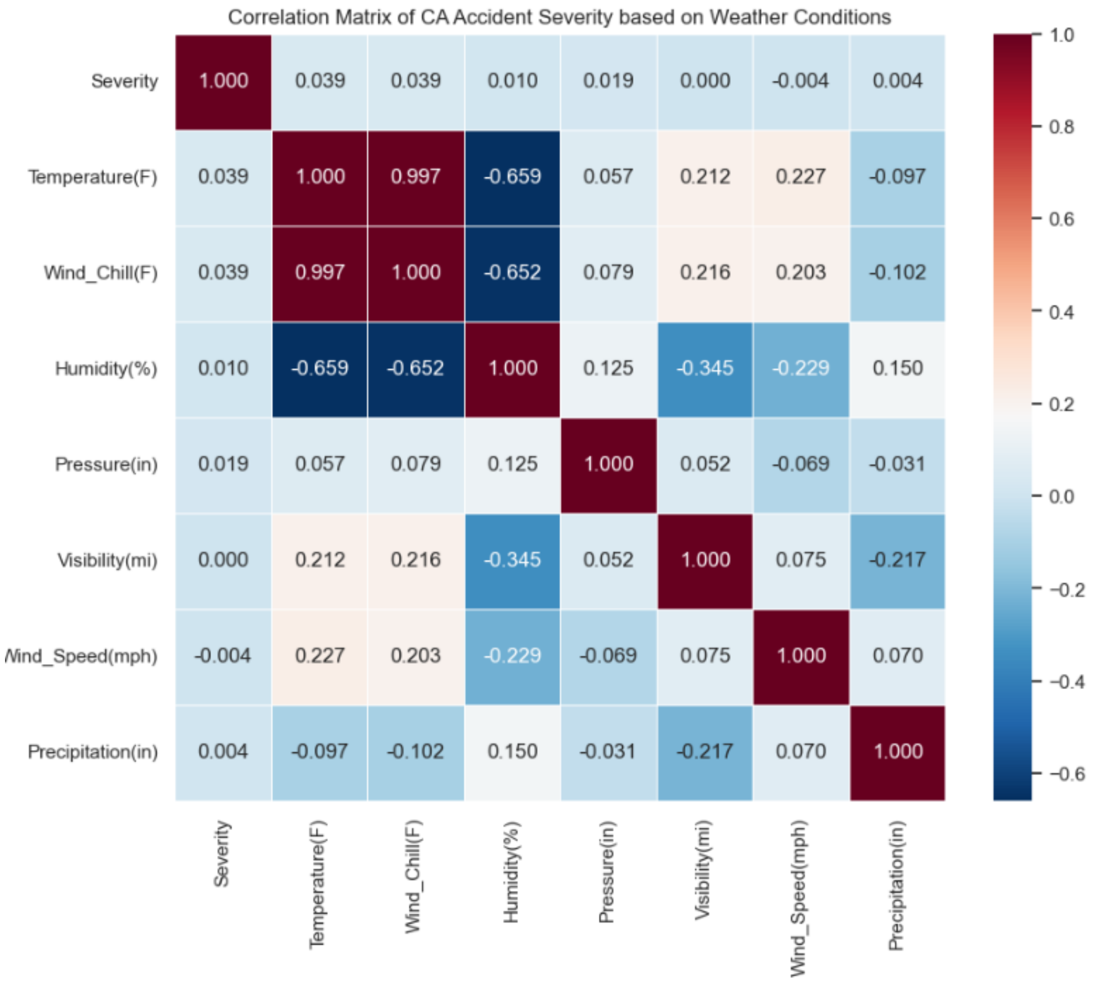


Fig 2: Accident Frequency Heatmap by County in CA

I then decided to narrow and clean the dataset into displaying features and observations I cared about and would be important for my analyses. I removed redundant and irrelevant features such as City, Country, and Timezone. I also detected the outliers in the data using Z-scores and Interquartile Range, however, I decided not to remove them due to them having little to no impact on what I am trying to predict. Moreover, due to the nature of my dataset, I decided to one hot encode categorical data in order to clearly understand what sort of conditions impacted severity of car accidents.



	Severity	Temperature(F)	Wind_Chill(F)	Humidity(%)	Pressure(in)	Visibility(mi)	Wind_Speed(mph)	Precipitation(in)
Severity	1.000	0.039	0.039	0.010	0.019	0.000	-0.004	0.004
Temperature(F)	0.039	1.000	0.997	-0.659	0.057	0.212	0.227	-0.097
Wind_Chill(F)	0.039	0.997	1.000	-0.652	0.079	0.216	0.203	-0.102
Humidity(%)	0.010	-0.659	-0.652	1.000	0.125	-0.345	-0.229	0.150
Pressure(in)	0.019	0.057	0.079	0.125	1.000	0.052	-0.069	-0.031
Visibility(mi)	0.000	0.212	0.216	-0.345	0.052	1.000	0.075	-0.217
Vind_Speed(mph)	-0.004	0.227	0.203	-0.229	-0.069	0.075	1.000	0.070
Precipitation(in)	0.004	-0.097	-0.102	0.150	-0.031	-0.217	0.070	1.000

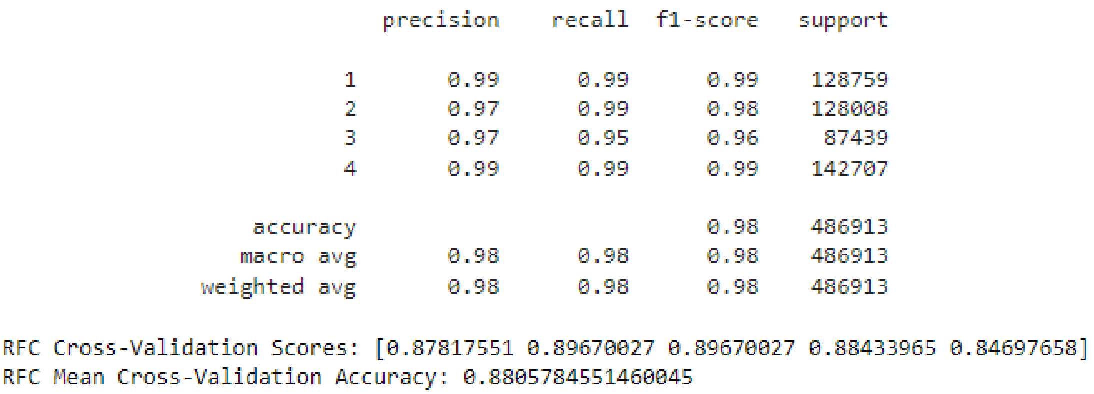
Fig 3: Correlation Matrix of CA Accidents Severity based on Weather Conditions

Once my data was cleaned, I did some inferences of the data using correlation matrices and visual graphs (Fig 3). Next I decided to test one of my hypotheses, which was the prediction of car accident severity using weather conditions. Since I had four categories of severity I decided to build a classification model. Since I had several categories of labels and several features in the dataset, I decided to use a K-Nearest Neighbour Model as well as a Random Forest Classifier to perform my task. Since my data was heavily imbalanced in terms of a high number of severity 2 car accidents, I decided to take several approaches for building the model. my model selection procedures included training both classification models on the original imbalanced dataset, a balanced dataset using undersampling, a balanced dataset using oversampling, and a balanced dataset using both undersampling + oversampling. For each run I got my classification report on the precision, recall, and f1 score for each severity level. Moreover for each run, I performed cross validation with k=5 folds to ensure my model was more generalizable. I scored the models on accuracy of prediction across each fold, and found the mean across the entire dataset. I then repeated this step with the undersampled, oversampled, and undersampled + oversampled fits getting both the cross validation scores across 5 folds as well as the mean across the folds. For the K-Nearest Neighbors model, I decided to find the k that yielded the best results. I tested the model on k = 1, 2, 4, 6, 8 neighbors.

Of course, my 3 fits would have varying complexities due to the nature of the data that trained the model. The model using the underbalanced data would be less complex and more simple because I are shaving the majority classes to match the amount of data in the other classes. In this scenario, I am losing data that could potentially influence the model and because of this the model might lose the ability to accurately predict the majority classes. Likewise, when I overbalance the data I might expect that the model becomes overly complex because I am creating a lot of artificial data. The data might not accurately represent a class and training on more data might lead to more complex decision boundaries. Lastly, I expect using both the undersampled + oversampled data would provide a middle ground in terms of complexity. Not only will I shave some data from the majority classes but I will also add artificial data to the minority classes. I believe this will create a more balanced model and yield the best results.

III. Results

After fitting my Random Forest Classification model for severity prediction using the raw data as well as the data after oversampling, undersampling, and a combination of both, I found that the model fit for all three methods yielded a similar mean cross validation accuracy value, however using both oversampling and undersampling yielded the a slightly higher accuracy of approximately 0.8806. However, this accuracy value is misleading because the data is unbalanced and is easily skewed by the majority class. Thus, I should look at the f1-score, which accounts for precision and recall to check for false negatives and false positives, to determine the true predictive ability of the model. In doing so, I found that the f1-score of the raw data was quite variable between classes of severity due to the imbalance, with values ranging from 0.10 to 0.95. These values increased notably for the undersampled and oversampled data respectively with varying accuracies. However, as I initially surmised, these resampled datasets had a lower predictive accuracy than the raw data because they introduced too much bias. For the data that used both resampling techniques, the f1-scores were significantly higher, with values between 0.96 and 0.99 between the severity classes, and an accuracy of 0.98 (Fig 4). This shows that the data that was both undersampled and oversampled was able to achieve a balance between being skewed by the majority class and introducing too much bias and is thus able to create the most accurate model.



```
              precision    recall  f1-score   support

           1       0.99      0.99      0.99    128759
           2       0.97      0.99      0.98    128008
           3       0.97      0.95      0.96     87439
           4       0.99      0.99      0.99    142707

    accuracy                           0.98    486913
   macro avg       0.98      0.98      0.98    486913
weighted avg       0.98      0.98      0.98    486913

RFC Cross-Validation Scores: [0.87817551 0.89670027 0.89670027 0.88433965 0.84697658]
RFC Mean Cross-Validation Accuracy: 0.8805784551460045
```

Fig 4: Performance Metrics and Cross-validation Scores of Random Forest Classification using both Oversampling and Undersampling Methods

I also used a KNN model to make my predictions and I compared the efficiencies of my models. I tested a certain number n neighbors starting with 1 in increasing order, however, increasing the number of neighbors considered in the classification process leads to a decrease in model performance (Fig 5). While n_neighbors = 1 gives the best metric scores, it falls short on the cross-validation scores. This behavior may indicate that as the number of neighbors increases, the model becomes overly complex or prone to overfitting the training data. Therefore, I decided to go with n_neighbors = 2 which gives us the best scores in performance metrics and cross-validation scores which helps ensure that the model is not simply memorizing the training data and that it performs well on unseen data. (Fig 6). In doing so, I found that the KNN model had similar results to the RFC model in that the oversampled and undersampled data had both a better f1-score and accuracy performance (compared to only raw data or only oversample or undersampled). In comparison to RFC, however, KNN yielded f1-scores and accuracies that were very slightly less than the RFC model, meaning that the RFC model has a slightly better predictive power. As a result, I chose the RFC model to be my final model.

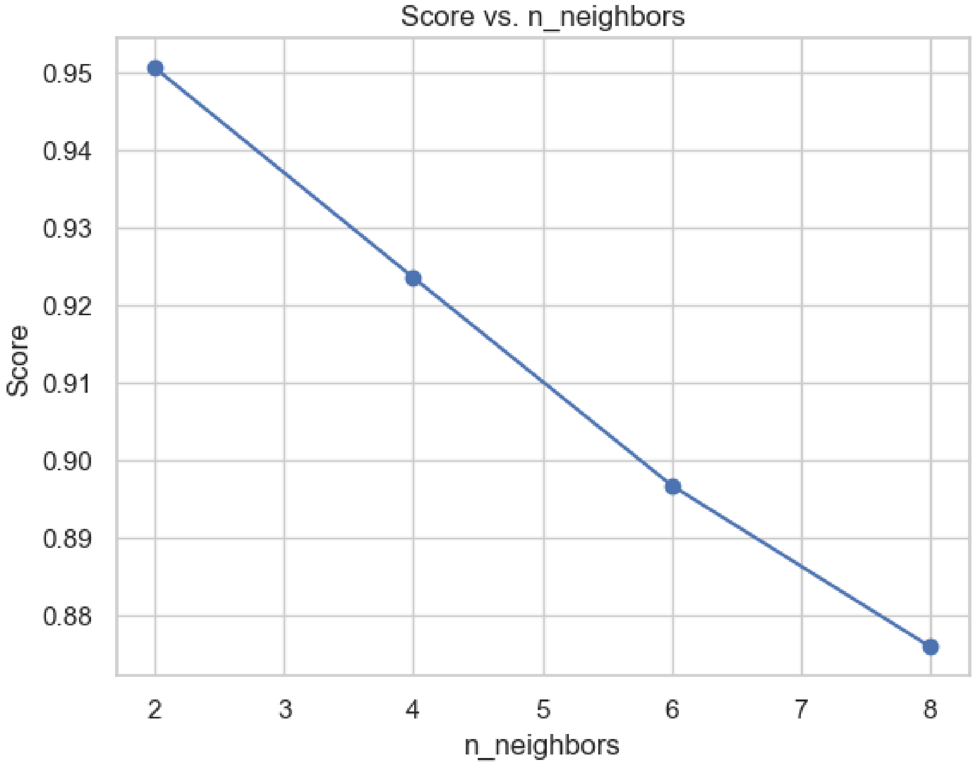
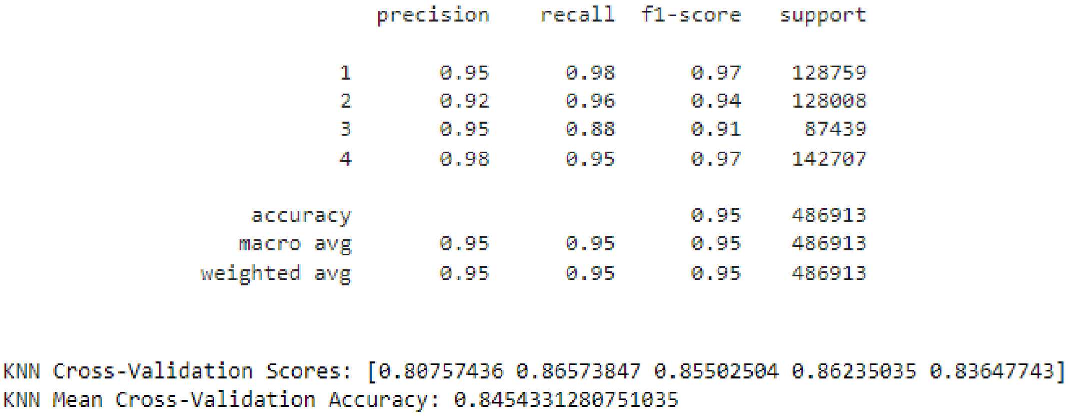


Fig 5: Performance Metrics vs n_neighbors



```
              precision    recall  f1-score   support

           1       0.95      0.98      0.97    128759
           2       0.92      0.96      0.94    128008
           3       0.95      0.88      0.91     87439
           4       0.98      0.95      0.97    142707

    accuracy                           0.95    486913
   macro avg       0.95      0.95      0.95    486913
weighted avg       0.95      0.95      0.95    486913


KNN Cross-Validation Scores: [0.80757436 0.86573847 0.85502504 0.86235035 0.83647743]
KNN Mean Cross-Validation Accuracy: 0.8454331280751035
```

Fig 6: Performance Metrics and Cross-validation Scores of K-Nearest Neighbors (KNN) using both Oversampling and Undersampling Methods

IV. Discussion

Considering the high predictive accuracy of my models, I am able to assert my hypothesis that weather conditions are adequate predictors of car accident severity. Because my model was able to accurately predict how severe a car crash would be given weather based parameters such as temperature, precipitation, and wind speed. This shows that these parameters are important in assessing whether or not a crash will be severe, which demonstrates a correlation between severe weather conditions and severe car accidents. Predictive studies like these are important because it would be difficult, expensive, or even unethical to conduct a true experiment to test which weather conditions are most hazardous for drivers. By creating machine learning models that can predict instances of car accidents from the weather, I can help caution drivers away from roads during certain weather conditions and hopefully prevent severe car accidents. Some further study to build on this could be to include even more parameters to increase the generalizability of the predictions to beyond weather analysis. For example, using location and traffic data to predict which roads in certain areas are more safe than others could help urban planners develop safer roads for drivers.

References:

Christy Bieber, J.D., “Car Accident Statistics For 2023”, Forbes Advisor, Jan 23, 2023, https://www.forbes.com/advisor/legal/car-accident-statistics/#fatal_accident_statistics_section

Dataset:

https://www.kaggle.com/datasets/sobhanmoosavi/us-accidents

Github:

https://github.com/lucaswin89/Predicting-Car-Accident-Severity