COMP 562 Final Project

Pooja Mathur, Sanaa Dalvi, Siya Yeolekar, Ananya Gode

I. Project Summary

This project aimed to create a model to predict the severity level of traffic incidents within the contiguous United States, and identify relevant variables that are most influential in causing accidents. Understanding these variables and finding patterns regarding time of day, weather, and other incident descriptors is vital in protecting public and road safety in less-than-ideal conditions. Utilizing R, we conducted our exploratory data analysis and initial data analysis, but due to computational limitations, when creating our models we switched to using Python. In our project, we tested logistic regression, support vector classifiers, Naive Bayes, and random forest models. The goal of this project is to advocate for road safety measures and other features that help protect drivers, passengers, and others across the country, as well as exploring seasonal, weather, and other accident patterns that are revealed by the data.

Our dataset, created by Sobhan Moosavi for a Cornell University paper [1] is pulled from multiple APIs that provide traffic incident data (sources include US-DOT, law enforcement, traffic cameras/sensors, etc.) from January 2016 until March 2023. There are 46 variables and 7,728,394 observations. Due to computational limits, we used a 500,000 observation random sample for our analysis.

We investigated what factors are most related to accident severity, where severity is categorical on a scale from 1 (least) to 4 (most). Key features we analyzed and engineered include time of day, affected road distance, geographical location, and weather categories. We used the existing severity column for our train and test data, and implemented additional methods including under-sampling and PCA to address class imbalance and high dimensionality.

Techniques Used

Logistic Regression is a linear classification algorithm used for binary and multi-class classification tasks. It models the probability of an instance belonging to a particular class using the logistic function. Logistic Regression is computationally efficient, interpretable, and well-suited for cases where the decision boundary between classes is relatively simple and linear.

Random Forest constructs many decision trees during training and outputs the class that is the mode of the classes (classification only) of the individual trees. It is highly robust and capable of handling non-linear relationships and high-dimensional data. Random Forest is useful for avoiding overfitting, dealing with missing values, and providing feature importance rankings.

Naive Bayes is a probabilistic classification algorithm based on Bayes' theorem. It assumes that features are conditionally independent given the class label, which simplifies computation. Naive Bayes is computationally efficient, easy to implement, and performs well in high-dimensional spaces.

Support Vector Classifier is a powerful classification algorithm that aims to find a hyperplane that best separates data into different classes. It is effective in high-dimensional spaces, can handle non-linear decision boundaries through kernel functions, and is robust against overfitting.

II. Data Cleanup & Processing

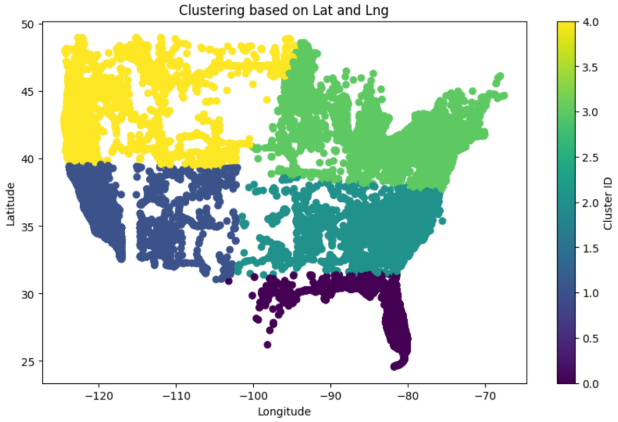
In our pre-processing, we took a random sample of 500,000 observations, performed a train-valid-test split and under-sampling on our data set. We also engineered features including converting timestamps, categorizing rush hour times, label encoding, binning weather conditions, and clustering geographical coordinates. We then measured variable importance, and applied principal component analysis to reduce our dimensions. One thing to note is our binning of weather conditions may impact model performance, due to its subjective nature.

Feature Engineering

We utilized feature engineering to convert the datetime variable to hour, day, week, and month variables to allow us to label encode and categorize a certain time of day as rush-hour as well as weekday or weekend. Our dataset also utilized very specific weather condition descriptions that were often unique to a specific observation. We decided to bin all of these descriptions into more specific weather categories to make classification much easier.

Geographical Category Clustering

By clustering geographical coordinates and location variables, we are able to group data points based on their proximity. These variables include start latitude, start longitude, zip code, country, city, street, airport code. This allows us to compress spatial data, improve visualization, make computation easier, and help identify spatial patterns in geographical data which can in turn help identify outliers or anomalies. The following displays the clusters of longitude and latitude features, after first scaling the data and using k-means clustering.



III. Exploratory Data Analysis

Location

From our exploratory data analysis we see that the highest city-wide volume of accidents occurs in Miami, but the most accidents by far are in California, then Florida, Texas, Virginia, and New York. The high amount of Virginia traffic can be mostly attributed to commuter traffic in and out of Washington, DC, which is not a distinct territory counted for in our dataset. Interestingly, we see that Raleigh makes the list despite not having a large metro area compared to the other cities.

Time/Date

Hourly accident distribution on its own is trimodal with groupings in twilight, morning commute, and mid afternoon/evening commute times. This is also shown in the weekdays plot, but not so much in the weekend, which has a high skew towards twilight hours and also just peaks in mid afternoon. Some information is missing in the yearly distribution - 6/2020 and 7/2020 have no recorded accidents. Additionally, accidents tend to be lowest in summer months and highest in fall/winter. 2019 marks the beginning of a gradual increase as months go by each year, but 2022 has an unclear distribution and January 2021 is an outlier. While changes in trends from the beginnings of the COVID-19 pandemic are not as visible in these plots, it is important to mention that an increase in working from home (WFH) and the discouragement of carpooling with those not in the same household may have changed the frequency of accidents during the last 3 years.

Weather

Low/normal pressure and clear visibility is most frequently observed in each condition, and weather conditions look generally mild overall. The data looks disproportionate when modeling the comparison of severity classes due to the severity class imbalance. Interestingly, harsh and dangerous weather conditions are not as frequently represented in the more severe classes (3 and 4) as our original assumption.

IV. Feature Selection

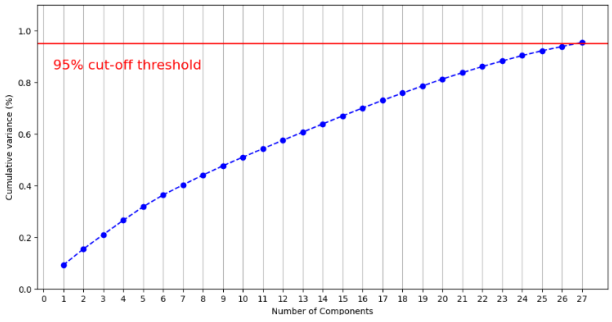
In building our models, we want to make sure our features are not highly correlated with each other which can lead to unstable coefficient estimates and reduce model interpretability. Using VIF test, we can reduce multicollinearity and evaluate which features could be removed in order to preserve orthogonal variables. We found that the Wind_Chill and Humidity variables were highly correlated with the Temperature variable, and features like Astronomical_Twilight and Civil_Twilight could be explained by (and are thus highly correlated with) features such

as Sunrise_Sunset. We thus decided to remove several of these features to simplify the data and avoid unstable estimates.

V. Handling Class Imbalance

Principal Component Analysis

PCA is a dimensionality reduction technique that aims to retain as much of the original variance in the data. This allows us to reduce computational complexity and mitigate multicollinearity as well as improve visualization and reduce noise. After standardizing the features, we determined that approximately the first 27 principal components explain about 95% of the variance in the data using a plot of cumulative variance vs. the number of principal components.



Random Undersampling

```
##   Severity_Level Severity_Count
## 1              2         216678
## 2              4           7185
## 3              3           4463
## 4              1           1601
```

Our dataset is largely unbalanced between severity classes, with the majority of observations being classified as Severity = 2. In our data, Severity = 4 is the smallest and Severity = 2 is the largest. After random under-sampling, a significant chunk of our data is lost due to the small size of class 4 (1601 observations). This significant class imbalance can lead to biased models that perform poorly when it comes to the minority class. By under-sampling, we randomly remove samples from the majority class to create a more balanced data set. This prevents bias, improves model generalization, reduces training time, and mitigates model sensitivity to outliers. A consequence of under-sampling this dataset is that it severely reduces the number of observations used in our analysis.

VI. Model Evaluation

By splitting our data into training, validation, and test sets, we are able to evaluate the classification ability of our models more accurately. Here, the training set is used to train the model, and the model learns patterns within the data and begins to adjust parameters to minimize error. The validation set is very important in evaluating the models performance throughout training, especially when tuning parameters and trying to prevent over fitting. The validation set helps us choose the best-performing model and confirms that the model generalizes well to new data.In the final model evaluation, the test set is a completely independent dataset that the model has not yet used. It provides unbiased estimates of the models performance.

Model Construction

We created a pipeline to create effective training, validation, and test sets by implementing under-sampling and PCA in the best order. and applying undersampling only to the training set ensures our model is evaluated on realistic data. Finally, PCA with 27 components is applied to the training data. This fit is then used to transform the validation and test sets. After splitting our data set into a training, validation, and test set, undersampling is applied to the training set to help our model learn from better representations of the minority class. Executing undersampling after the PCA transformation helps prevent data leakage. All four models were fitted with and without PCA to evaluate whether dimension reduction is beneficial for our data set. Trying alternative methods also helped assess the robustness of our models, and their sensitivity to feature representation.

After our data set undergoes one of two pipelines, the four models we selected are fitted for evaluation. Our team also experimented with the number of principal components for PCA, and found that 27 principal components yielded the best results.

Performance

Logistic Regression

The logistic regression model without PCA performed mediocre, with a 44% accuracy on both the validation and test set. The confusion matrix is more

balanced after under-sampling than before, which is true for all of the proceeding models. However, the model still struggles with class imbalances, apparent by the low precision, recall, and F1-scores for classes 2 and 3. Hyperparameter tuning was also applied to both logistic regression models, but did not improve model performance notably. With PCA, the model performs slightly better with a 46% accuracy rate across both the validation and test set. Again, class imbalance persists apparent by significantly low precision, recall, and F1-scores for classes 2 and 3.

Random Forest

Random forest without PCA performs pretty well compared to the other models, with a 48% accuracy for both validation and test sets. Precision, recall, and f1-score values are also higher than the previously seen values. Although this model still suffers from class imbalance, it is slightly better at classifying the minority class. The random forest model with PCA performs better than the model without PCA. This model has a 50% accuracy test and validation set, the only model to reach an average accuracy of 50% or above.

Naive Bayes

The Naive Bayes model without PCA performs similarly to logistic regression, with a 45% accuracy between the validation and test sets. Looking at the precision and recall values for class 2, it seems like that when this model makes predictions for class 2, it is often correct but it misses many instances of class 2. Surprisingly, the Naive Bayes model with PCA applied performs worse than without PCA, and in fact performs the worst out of all the models. The model produces a 39% accuracy rate for the validation and test set, and features some of the lowest precision, recall, and f-1 scores for classes 2 and 3.

This could indicate that PCA removes important feature information, or that even with attempts to remove multicolinearity, there is still dependence present that violates the Naive Bayes assumptions of independence. PCA aims to maximize variance, but not necessarily class separability, and the performance of this model could be a symptom of issues in finding the best way to separate the Severity classes out.

Support Vector Classification

The SVC model without PCA performs surprisingly well, with an accuracy of 49% and the second-highest weighted F-1 score. Interestingly, the model faces a 0.71 precision rate for class 1. We see a reduction in accuracy again to 48% after applying PCA, indicating that our approach to PCA may be affecting kernel choice and hyperparameter matching.

VII. Conclusion

The Naive Bayes model with PCA yielded the best results, with a 50% accuracy rate across validation and test sets and an 50% weighted F1-score. The SVC model without PCA yielded the second best results, with a slightly lower accuracy rate and weighted F1-score.

Areas for Improvement

Overall, due to the high class imbalances and severe reduction in the count of observations after employing random undersampling, none of the models produced significantly high accuracies or managed to mitigate the effects of these imbalances. After conducting these tests, we still believe that dimension reduction and some version of PCA is still appropriate for this data, however, it is also possible that the full dataset yields completely different patterns.

We recommend using the larger data set in future analysis if resources permit, and experimenting with more models using hyperparameter tuning. Subjective weather bins may have also impacted our model performance, and therefore the implementation of a weighted average or a better grouping method could improve model performance. Experimentation with different numbers of principal components is also encouraged, as our team mainly experimented with 24-29 principal components. A different number of principal components may improve models that use PCA. The largest issue to address in further analysis is the class imbalance prevalent in all of the models we created. It is possible that a different size for undersampled groups would improve class imbalance, in combination with a larger data set size. Underlying relationships in the data set could be explored more thoroughly by using the entire data set. Improved feature engineering may also result in better results.