US Automobile Accidents - A Technical Report and Analysis

Elan Wilkinson, Parker Christenson, and Melissa Short

Applied Artificial Intelligence, University of San Diego

AAI-500-01: Probability and Statistics for Artificial Intelligence

October 23, 2023

Table of Contents

U.S. Automobile Accidents - A Technical Report and Analysis

The US Accidents data set, utilized for this technical report, can be found at Kaggle.com. This data set contains detailed information on millions of accidents across the United States, including attributes such as location, weather conditions, and severity. By leveraging this comprehensive data set, a classification model can be developed that can contribute to improving road safety and accident prevention efforts.

The US Accidents data set is a comprehensive compilation of traffic accident data in the United States. It is a valuable resource for understanding the patterns and factors that contribute to accidents on American roads. In this technical report, the emphasis has been placed on building a classification model using this data set from the years 2016 to 2019. The goal is to develop a model with our data analysis and model that can accurately predict the severity of accidents based on various features such as weather conditions, road surface conditions, and location. There are additional features and applications that could potentially be leveraged to analyze this data set, but for the purposes of this report, focus is limited to attempted prediction of the severity of accidents given the available data collected and potential to predict the number of automobile accidents in the future.

Data Cleaning and Preparation

Data cleaning is a crucial step in preparing the data set for analysis. It involves identifying and handling missing values, removing duplicate entries, and addressing any inconsistencies or errors in the data.The data consisted of information about US car accidents with 6,599, 480 rows and 46 columns of data. With the volume of data to process, the choice was made to exclude data after 2019 due to the excessively large size of the .csv file. The dataset originally contained data from 2016 to 2023. Loading data into the model and simple pandas data frame manipulation was a challenge to load, due to the size of the CSV file. Ensuring the quality and cleanliness of the data enhances the reliability and accuracy of our classification model. The performance of some alterations, processing and cleaning of the data was necessary, taking such values in the data frame such as night or day and converting them to 0 or 1, to facilitate usage of the data in a model. Changes were made to the data types of some of the columns as well to convert to more usable format, such as to allow the model to use data, decrease unnecessary segmentation of subcategories when one-hot encoding, and conversion to floats to decrease processing time and reduces the computational power needed to train the model. In preparation for running the model and later usage, adjustments were made to compensate for missing values,and error values within the data file in order to achieve an accurate representation for the prediction models. The prediction models also needed to be adjusted and sorted through to ensure that the data models demonstrated usefulness in depiction of severity, supporting the prediction of future accidents.

Exploratory Data Analysis

Before delving into the model-building process, it is important to perform exploratory data analysis (EDA) to gain insights into the data set. EDA involves examining the distribution of variables, identifying any patterns or trends, and detecting outliers or missing values. By conducting EDA, better understanding can be gained of the characteristics of the data and make informed decisions during the modeling phase. The goal of our exploratory data analysis is the exploration of the dataset, patterns were identified in accident severity based on features in the environment such as turning loops, traffic signals, traffic calming, stop sign, bus station, roundabout, railway station, no exit, junction giveaway, crossing, bump, and proximity to an amenity.

In addition to exploration of relationships between features in the environment and severity, relationships between severity and weather conditions and location were investigated. It was found that California and Florida had exceptionally high levels of observations in the dataset, although this could be explained by larger than average populations. The most common weather condition per accident observation was “fair” or a very similar label, higher severity representation percentages were found with specific weather conditions, particularly hail. While weather may only present a transient change to likelihood of severity at a given time, the data would support raising premiums in geographic regions that have disproportionate rates of high-severity-risk weather.

Model Selection

The data types of the columns presented a challenge with initial modeling. The data types were primarily int64, floats, with some columns consisting of some objects and strings. During data preprocessing, the decision was made to drop some of the unnecessary columns that were not of significant relevance for the chosen classification of severity. Categories such as Zip Code, Case ID, and Airport Code were discarded. All rows that contained missing values were dropped, as the relevance of the affected columns was minimal. One hot encoding was used to ensure that the data was in format that was ingestible by the model, and was included in some of the models explored for classification. Data was split into 80% testing set and a 20% testing set. A random forest classification model was chosen for severity prediction given weather, location, and presence of aforementioned traffic features. The random forest model was chosen for its high level of accuracy as well as its robustness.

Model Analysis

In order to explore accident data, the libraries Seaborn, Matplotlib, and Pandas were utilized. These libraries facilitated effective visualization and analysis of the data, and resulted in valuable insights. The primary objective was the meaningful prediction of severity of accidents given available data and determination of risk factors in such a way that could be leveraged for use cases such as adjusted rates for different regions or times of year for insurance companies.

Models were created using exclusively weather related variables, exclusively environment details such as the presence of crosswalks, junctions, railway crossings, and the like, and of a combination of these as well as location data. The severity of accidents in the dataset was heavily skewed toward accidents classified as severity level 2. The source of the data does not elaborate as to the exact criteria used for determination of severity, and whether this skew might be the result of being representative or real-word average model severity, a broad definition of level 2 such that it would encapsulate an unbalanced portion of accident, or due to the way 1he observations themselves were gathered in a way that caused the imbalance. The volume of the data was sufficiently large that it allowed for reducing observations in the dataset in categories that were overrepresented, to compare model performance with a more balanced dataset.

With both the unbalanced and severity balanced datasets, the model accuracy was significantly higher for predictions of accidents of Severity 2. Reduction of overrepresented severity categories decreased model accuracy.

Data was standardized using a standard scalar with the random forest classification model. The model had an overall accuracy of 95% with the highest precision at severity 2 of 97% and dramatically weaker precision of severity 4 at 41%. The confusion matrix of predicted and true labels reflected this, with the greatest values at correct predictions of severity 2 accidents. A balanced accuracy score was calculated from the sensitivity and specificity of the model, yielding a value of 0.61. The calculated ROC curves reflected the varying levels of accuracy for the different severities.

The calculated precision scores had a weighted average of 0.94, with precisions ranging from 0.97 for severity 2 to 0.41 for severity 4.

Conclusion and Recommendations

The scikit-learn random forest model utilized uses decision tree classifiers for categorization. While TensorFlow or PyTorch would have aided in implementation of integration into a data pipeline, the sci-kit learn model was well suited for making our model for the analysis, and the development for this final analysis. Their use was considered, but it was ultimately determined that the scikit-learn library offered an extensive model evaluation which was used in evaluation and analysis of the model. Creation of the model was fairly straight-forward, with splits of the data set into a training set and a testing set. When splitting the data set into two objects, our predictive feature was ‘severity.’

During development of the model, the overall run time of the model was around ten minutes. This was primarily due to the size of the dataset. The dataset in the CSV file was approximately 3.5 gigabytes, which did present a challenge in processing at times prior to pruning to a selected time range. In the future, if a full reference list of all accidents in the United States, offloading the computational task to a AWS or Lambda labs GPU cluster would be more practical. The model created for this analysis helped illuminate some of the likely challenges that would be encountered in an expanded later study.

With some strong accuracy scores exceeding 90%, careful note should be made of the severe imbalance in the data and weaker performance for severities of other categories. Further data as to the nature of the discrepancy in severity counts and its cause would influence further corrections to enhance the model's performance.

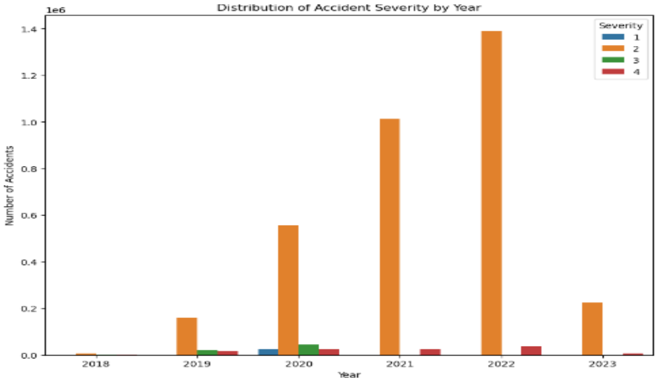
References

Moosavi, S. (2023, May 28). *US accidents (2016 - 2023)*. Kaggle. N https://www.kaggle.com/datasets/sobhanmoosavi/us-accidents

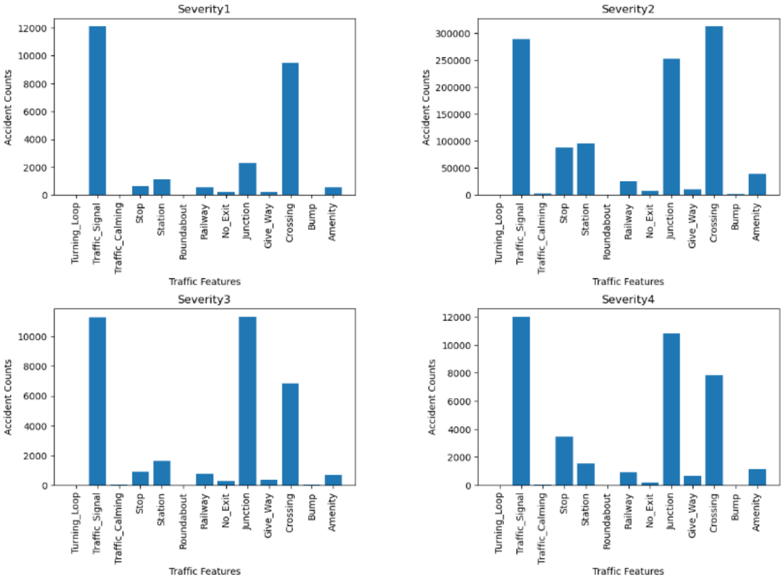
Agresti, A., & Kateri, M. (2022). *Foundations of statistics for data scientists: With R and python*. CRC Press, Taylor & Francis Group.

Scikit-Learn. (n.d.). GitHub - scikit-learn/scikit-learn: scikit-learn: machine learning in Python. GitHub. https://github.com/scikit-learn/scikit-learn

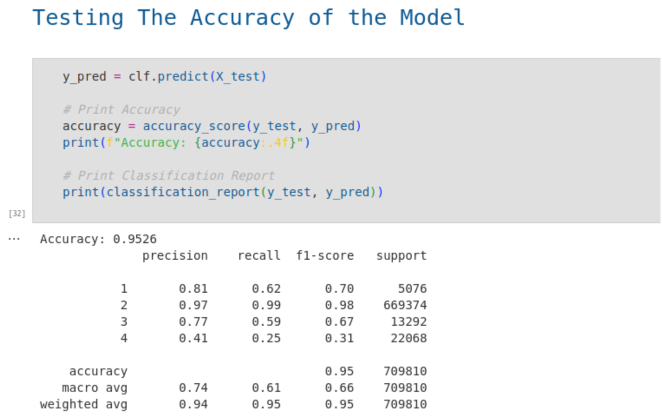
Appendix



This graphic shows the amount of accidents that occured per year. It also shows the counts of the severity of accidents that happened per year as well. It shows us the best way to look at our data, and how the group was able to Identify that there was a very large count of level two severity levels.



The Diagram above shows the amount of accidents based on their severity, and by the traffic features. This was another key indicator that we should switch our approach for refining the model, and developing the model even further to include the environmental conditions. We assumed that the weather and traffic variable might give us a leg up on the accuracy on the model, and having the model predict some of the less accurate severities.

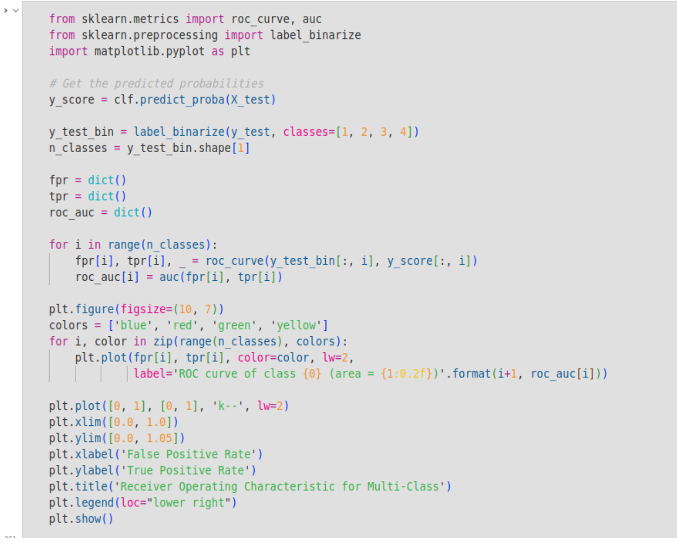


Testing The Accuracy of the Model

```
y_pred = clf.predict(X_test)

# Print Accuracy
accuracy = accuracy_score(y_test, y_pred)
print(f"Accuracy: {accuracy:.4f}")

# Print Classification Report
print(classification_report(y_test, y_pred))
```

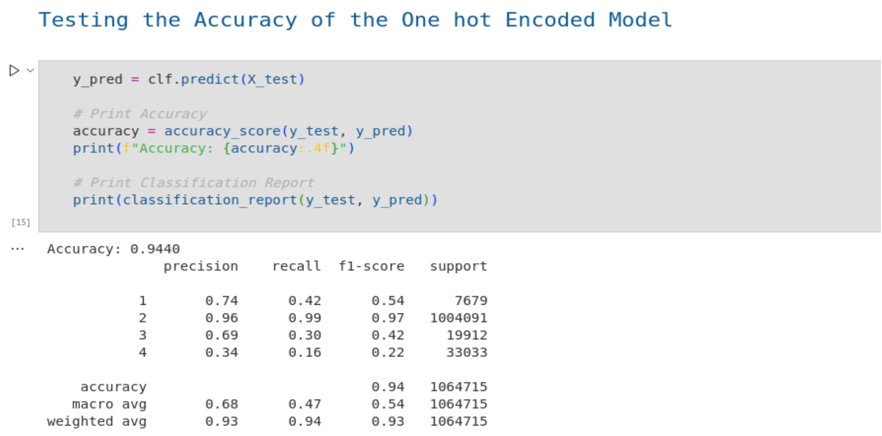
```
Accuracy: 0.9526
              precision    recall  f1-score   support

           1       0.81      0.62      0.70      5076
           2       0.97      0.99      0.98    669374
           3       0.77      0.59      0.67     13292
           4       0.41      0.25      0.31     22068

    accuracy                           0.95    709810
   macro avg       0.74      0.61      0.66    709810
weighted avg       0.94      0.95      0.95    709810
```

This is the classification report of the first model our group had created. The model created the whole dataset, in which we did not use any hot encoding or any kinds of data manipulation. This model achieved the greatest accuracy, out of the three models .



```
from sklearn.metrics import roc_curve, auc
from sklearn.preprocessing import label_binarize
import matplotlib.pyplot as plt

# Get the predicted probabilities
y_score = clf.predict_proba(X_test)

y_test_bin = label_binarize(y_test, classes=[1, 2, 3, 4])
n_classes = y_test_bin.shape[1]

fpr = dict()
tpr = dict()
roc_auc = dict()

for i in range(n_classes):
    fpr[i], tpr[i], _ = roc_curve(y_test_bin[:, i], y_score[:, i])
    roc_auc[i] = auc(fpr[i], tpr[i])

plt.figure(figsize=(10, 7))
colors = ['blue', 'red', 'green', 'yellow']
for i, color in zip(range(n_classes), colors):
    plt.plot(fpr[i], tpr[i], color=color, lw=2,
             label='ROC curve of class {0} (area = {1:0.2f})'.format(i+1, roc_auc[i]))

plt.plot([0, 1], [0, 1], 'k--', lw=2)
plt.xlim([0.0, 1.0])
plt.ylim([0.0, 1.05])
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('Receiver Operating Characteristic for Multi-Class')
plt.legend(loc="lower right")
plt.show()
```

The code above is our ROC curves used to evaluate the model, and see how the model's accuracy performed over time. We had ROC curves for all four of the severity levels. With that being said, the for loop is able to perform the calculation for the classes.

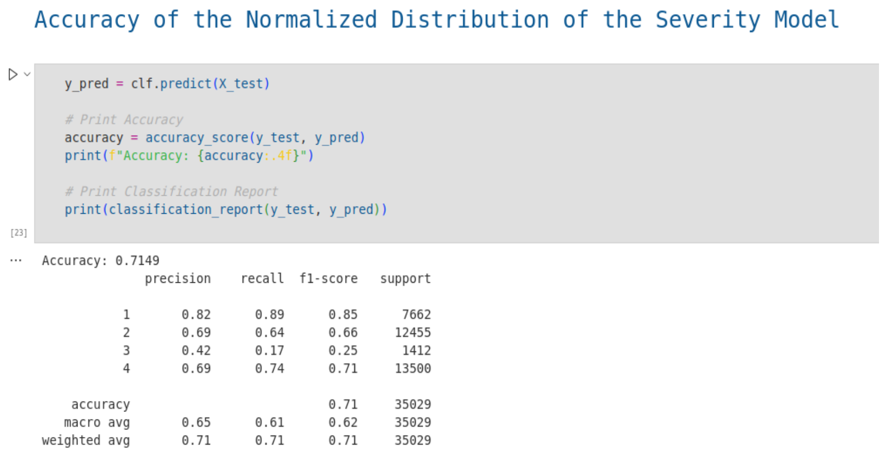


Testing the Accuracy of the One hot Encoded Model

```
y_pred = clf.predict(X_test)

# Print Accuracy
accuracy = accuracy_score(y_test, y_pred)
print(f"Accuracy: {accuracy:.4f}")

# Print Classification Report
print(classification_report(y_test, y_pred))
```

[15]

```
Accuracy: 0.9440
              precision    recall  f1-score   support

           1       0.74      0.42      0.54      7679
           2       0.96      0.99      0.97   1004091
           3       0.69      0.30      0.42     19912
           4       0.34      0.16      0.22     33033

    accuracy                           0.94   1064715
   macro avg       0.68      0.47      0.54   1064715
weighted avg       0.93      0.94      0.93   1064715
```



Accuracy of the Normalized Distribution of the Severity Model

```
y_pred = clf.predict(X_test)

# Print Accuracy
accuracy = accuracy_score(y_test, y_pred)
print(f"Accuracy: {accuracy:.4f}")

# Print Classification Report
print(classification_report(y_test, y_pred))
```

[23]

```
Accuracy: 0.7149
              precision    recall  f1-score   support

           1       0.82      0.89      0.85      7662
           2       0.69      0.64      0.66     12455
           3       0.42      0.17      0.25      1412
           4       0.69      0.74      0.71     13500

    accuracy                           0.71     35029
   macro avg       0.65      0.61      0.62     35029
weighted avg       0.71      0.71      0.71     35029
```

In the Two pictures above, we are able to see the evaluation of the two other models. In these efforts we attempted to handle some of the problems we had with our classification model. The top visual is the Accuracy of the one hot encoded classification model. Our group wanted to one hot encode all of the environmental and weather conditions, in order to ensure that we had tried everything that we could have done to ensure the greatest accuracy of the model.