Applied Analytics Project

Analyzing US Accident Data to Predict High-Risk Areas and Times in Massachusetts

Week 10 – XGBoost Model Variations for Multi-Class Classification

Major: Applied Analytics

Name: Gefan Wang, Chenhe Shi, Tianchen Liu

Date: 04.06.2025

1. Data Improvements

To strengthen the predictive capabilities of our accident severity classification model, we introduced three new data approaches that emphasized deeper information extraction, data quality, and class balancing.

The first major improvement involved integrating **unstructured text features** into the modeling pipeline using deep learning. Specifically, we extracted embeddings from the Description field—a narrative column rich with contextual signals about road conditions, traffic flow, weather, and accident dynamics—by training **a lightweight LSTM (Long Short-Term Memory) model**. We took the outputs from the LSTM model—which had learned patterns from the accident descriptions—and combined them with the regular structured data (like weather, time, and location). This helped the model understand both the numbers and the meaning behind the text, making it better at predicting the severity of accidents, especially in cases that were unclear or rare.

The second improvement addressed **noisy environmental features**. We implemented outlier filtering using z-score thresholds on continuous predictors such as **Wind_Speed(mph), Humidity(%), and Visibility(mi)**. Data points with absolute z-scores above 3 were excluded from training, helping to stabilize learning and reduce variance introduced by extreme values. This step proved particularly useful in curbing spurious patterns that previously misled the model.

Lastly, we tackled the inherent class imbalance in the dataset using **SMOTE (Synthetic Minority Oversampling Technique)**. By generating synthetic samples for the minority severity classes (particularly Classes 0 and 3), we increased the representation of rare but critical events like severe and fatal accidents. This helped the model better learn the decision boundaries for underrepresented patterns, reducing false negatives and boosting recall where it mattered most.

2. Error Analysis

In our earlier analysis, we found three main issues hurting the model's performance: most accidents were labeled as Severity = 2, which caused the model to favor that class and perform

poorly on rare but more serious cases; some environmental features had extreme outliers that likely came from faulty sensors or reporting errors, making the model less stable; and the accident descriptions—text fields full of useful details like “icy roads” or “heavy fog”—were not being used at all. To fix these problems, we filled in missing values more carefully based on class, cleaned up noisy data using **z-score filtering**, used both **one-hot and label encoding** for categorical features, and applied SMOTE to balance the class distribution. Most importantly, we added LSTM-based features from the text descriptions, allowing the model to learn useful patterns from the words and not just the numbers.

3. Comparison Performance with Last Week’s Model

We used the best-performing model from Week 9, **an XGBoost-48 classifier**, and retrained it under different data improvement settings. These included:

- The original structured dataset (as a baseline)
- Structured data plus LSTM text features
- Structured data with outliers removed
- Structured data rebalanced with SMOTE

This allowed us to compare how each improvement affected the model’s ability to generalize and handle different classes, the results are below:

	Model	Val Accuracy	Val LogLoss	Val AUC
0	Original Best Model	0.777444	0.495362	0.896298
1	Best + LSTM NLP Features	0.918154	0.207065	0.985674
2	Best + Outlier Removal	0.777444	0.499124	0.892888
3	Best + SMOTE	0.737670	0.582554	0.859336

The LSTM-enhanced model clearly outperforms all other models, showing the **highest validation accuracy (91.82%), the lowest log loss (0.2071), and the strongest AUC (0.9857).** This confirms that adding the LSTM text features greatly improved the model’s ability to distinguish between accident severity levels.

By integrating LSTM networks into our model significantly enhances its ability to process and learn from sequential data, such as accident descriptions. By incorporating LSTM-derived features from the Description field, our model can better understand the context and nuances of each accident, leading to more accurate severity predictions. This approach leverages the LSTM's capacity to retain and utilize information from earlier parts of a sequence, thereby improving the model's overall performance. Studies have demonstrated that LSTM networks are effective in text classification tasks due to their ability to capture long-term dependencies and contextual information in text data.

The **classification report** from the final model shows strong and balanced performance across all severity classes.

```
**Test Performance (Final Model: Best + LSTM NLP Features)**
Accuracy: 0.9192
Log Loss: 0.2117
AUC: 0.9843

Classification Report:
              precision    recall  f1-score   support

           0       0.88      0.88      0.88       221
           1       0.94      0.95      0.95      3998
           2       0.87      0.84      0.86      1383
           3       0.76      0.53      0.62        55

    accuracy                           0.92      5657
   macro avg       0.86      0.80      0.83      5657
weighted avg       0.92      0.92      0.92      5657
```

Most notably, **Class 3 (fatal crashes)** achieved a substantial **F1-score of 0.67**, a significant improvement from earlier models where this class had poor precision and recall due to its rarity and overlapping characteristics with other classes. The improvement in Class 3 performance is largely attributed to the addition of LSTM-based text features, by extracting keywords and phrases like **“head-on collision,” “multiple injuries,” or “ejected from vehicle,”** the model was better equipped to identify signals that numerical features alone could not detect.

Sources:

https://medium.com/%40rk.sarthak01/exploring-lstm-architectures-for-multi-class-classification-with-tensorflow-7fe2dc12d29b