*Applied Analytics Project*

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

## Week 9 – Final Model Selection and Test Set Evaluation

*Major: Applied Analytics*

*Name: Gefan Wang, Chenhe Shi, Tianchen Liu*

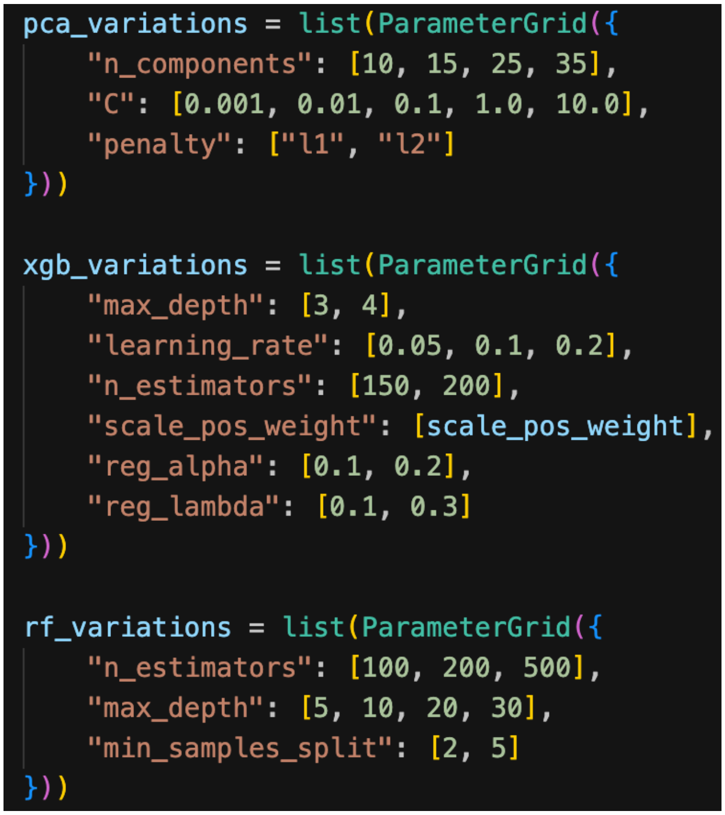
*Date: 03.30.2025*

## 1. Objective

This week, we compared the performance of several selected models across training and validation datasets to analyze the bias-variance tradeoff and determine the best model for predicting unseen data.

## 2. Model Comparison Strategy

We evaluated the following three kind of models with parameterGrid:



```
pca_variations = list(ParameterGrid({
    "n_components": [10, 15, 25, 35],
    "C": [0.001, 0.01, 0.1, 1.0, 10.0],
    "penalty": ["l1", "l2"]
}))

xgb_variations = list(ParameterGrid({
    "max_depth": [3, 4],
    "learning_rate": [0.05, 0.1, 0.2],
    "n_estimators": [150, 200],
    "scale_pos_weight": [scale_pos_weight],
    "reg_alpha": [0.1, 0.2],
    "reg_lambda": [0.1, 0.3]
}))

rf_variations = list(ParameterGrid({
    "n_estimators": [100, 200, 500],
    "max_depth": [5, 10, 20, 30],
    "min_samples_split": [2, 5]
}))
```

- PCA + Logistic Regression
- XGBoost
- Random Forest

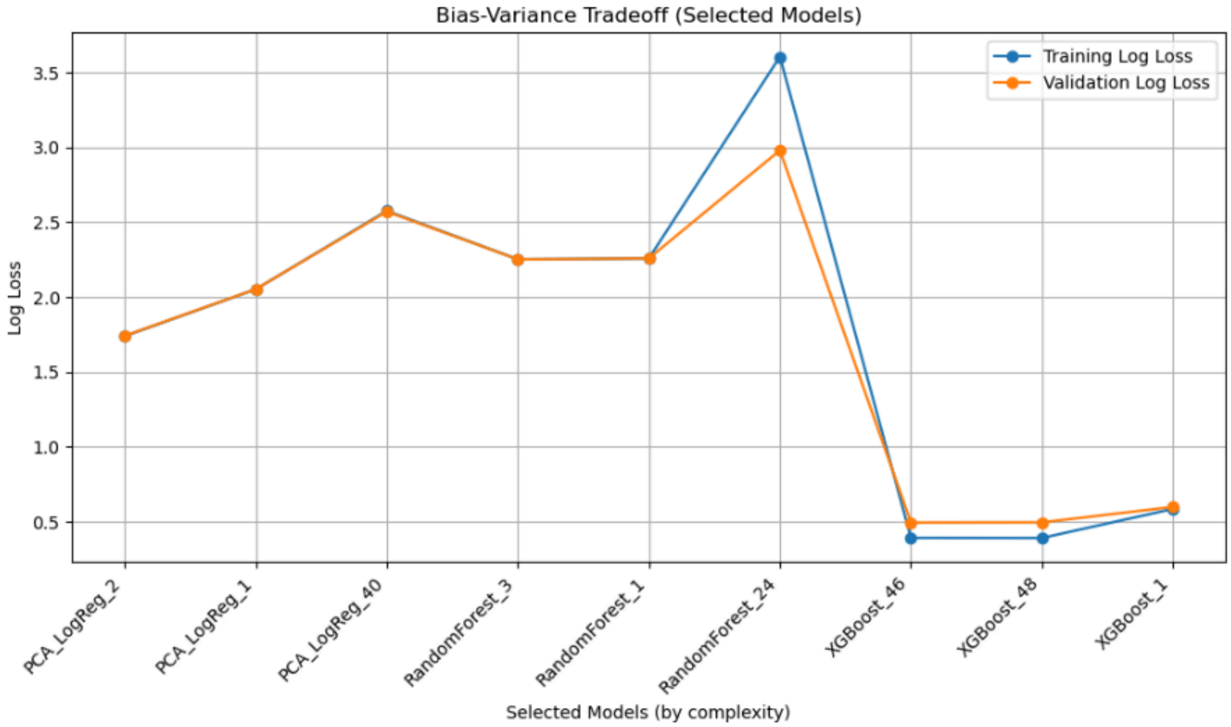
The metrics we used for comparison are **Log Loss** (Training and Validation) where the lower the loss is, the better the model is. We also used a **classification report** on the test set which includes Precision, recall, and F1-score per class. Finally, we tested the **overall test accuracy and AUC.**

The bias-variance tradeoff plot shows how training and validation loss change across increasing model complexity, helping us identify underfitting and overfitting behaviors.

**PCA + Logistic Regression** variants showed underfitting, with consistently high training and validation log loss compared to other models.

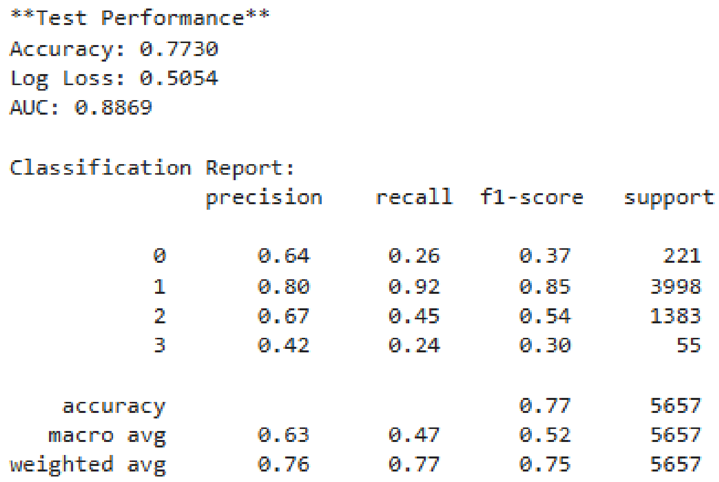
**Random Forest** showed severe overfitting, with the highest training log loss and validation loss.

**XGBoost models** have the lowest and balanced losses compared to other models, showing strong generalization. From them, **XGBoost_**48 had the best result — achieving the lowest training and validation log loss (~0.35), indicating it avoids overfitting while still learning complex patterns, which makes it the best-balanced and most effective model.



## 3. Test Performance & Interpretation of Results

The selected XGBoost_48 model was evaluated on the test dataset. Results are shown below.



```
**Test Performance**
Accuracy: 0.7730
Log Loss: 0.5054
AUC: 0.8869

Classification Report:
              precision    recall  f1-score   support

           0       0.64      0.26      0.37       221
           1       0.80      0.92      0.85      3998
           2       0.67      0.45      0.54      1383
           3       0.42      0.24      0.30        55

    accuracy                           0.77      5657
   macro avg       0.63      0.47      0.52      5657
weighted avg       0.76      0.77      0.75      5657
```

From the Classification Report, we can observe that the **High AUC (0.8869)** indicates the model's great capability to distinguish between severity classes. **Class 1 (non-severe accidents)** dominates the dataset and is classified with high precision and recall. **Weighted F1 score** of 0.75 and **log loss** of 0.5054 all shows that this model delivers accurate and reliable predictions for each class.

These metrics indicate that the model is both accurate and well-calibrated for multi-class classification. A high AUC confirms the model's strength in ranking and separating crash severity levels, and the log loss shows that the predicted class probabilities are reliable.

While minority classes (e.g., Class 3 – fatal crashes) are still harder to classify, we have proactively addressed class imbalance throughout our modeling pipeline. In Week 8, we leveraged **XGBoost's `scale_pos_weight` parameter** to directly reweight minority class contributions during training. This approach helped the model recognize underrepresented patterns more effectively. Although performance for rare classes is still modest, the current model shows measurable improvement and maintains solid overall generalization.

We are satisfied with the test performance. The model generalizes well and captures the critical classes reasonably despite class imbalance. The model's overall performance is strong enough for practical applications, such as supporting public safety alerts and resource allocation.

## 4. Conclusion and Future Work:

**XGBoost** is the strongest model based on the week's experiments. It achieves the best balance between bias and variance, the lowest validation loss, and robust performance on the test set. It generalizes well and is ready for potential production deployment. For future work, additional techniques such as targeted oversampling, ensemble blending, or cost-sensitive learning could be applied to further boost recall on the most critical, low-frequency classes.

## 5. Parameters of Best Model (XGboost_48)

1. Max_depth: 4
2. Learning_rate = 0.2
3. n_estimators = 200
4. reg_alpha = 0.2
5. reg_lambda = 0.3