Tennessee Accident Severity Prediction

Team 9 - COSC522: Machine Learning

Clay Shubert, Nagendra Upadhyay, Robert Williams, and Zach Williams

cshubert@vols.utk.edu, nupadhy3@vols.utk.edu, rwill166@vols.utk.edu, zwilli13@vols.utk.edu

December 14, 2023

https://github.com/clayshubert/USAccidentAnalysisML

THE UNIVERSITY OF
TENNESSEE
KNOXVILLE

Table of Contents

Abstract 1

Introduction 2

- Background and Motivation 2
- Challenge and Achievement 2
- Task Assignment 2

Technical Approach 3

- Pipeline 3
- Algorithms 3

Experiments and Results 6

- Experimental Design 6
- Result and Comparison 7
- Discussion of Results 8
- Comparison to State-of-the-Art 8

Summary and Future Work 10

References 11

Appendix A: Performance Evaluation Table 12

Abstract

The goal of this project is to develop a predictive model for car accident severity based on a detailed analysis of a large U.S. car accident dataset spanning from 2016 to 2023. Motivated by the significant impact of car accidents on public safety and the economy, our team decided to employ machine learning techniques to identify influential features, understand factors that are related to the road, and determine the most relevant correlations within the dataset.

We applied dimensionality reduction techniques such as Principal Component Analysis (PCA) and Fisher's Linear Discriminant (FLD) to reduce the space of the feature. We also used classification algorithms like Minimum Mahalanobis Distance Classifier, K-means, Random Forest, and Decision Tree. We also used fusion techniques like Majority Voting, AdaBoost, and XGBoost, which were used to enhance the prediction accuracy. The results of our efforts gave us a respectable accuracy of 92.21%, which could be further improved by making some improvements in the overall process and also doing more experimentation with the data to get better results.

Despite achieving good enough accuracy, our efforts fell a bit short to the state-of-the-art benchmark set by Dr. Ronghui Zhou, who achieved a remarkable accuracy of 97.4% using a much smaller subset of data. Although our overall accuracy was lower, we used a larger subset of the data which introduced more variability which likely decreased the accuracy.

In this report, we have presented our methodology, the challenges that we faced, achievements, revaluation of the results, and a comparative analysis with the state-of-the-art benchmark. The collaborative efforts of each team member played an important role in dealing with the complexities of predicting car accident severity, leading to the successful completion of our project.

Introduction

Background and Motivation

Car accidents are one of the biggest concerns for public safety since they are one of the leading causes of death globally. Analyzing the factors that influence accidents can help by providing some very valuable insights into ways to prevent them and mitigate their impact on traffic. In this project, our team employed multiple machine learning algorithms to predict the severity of the accidents using a vast dataset spanning from 2016 to 2023 across the United States.

The dataset that we used comprised over 7.7 million records across the 49 states included. It also contained 46 features which we would need to evaluate. Our primary objective was to attempt to create a predictive model that accurately assessed the accident severity based on key features to help community leaders to make the best decisions to enhance road safety.

The state-of-the-art benchmark that we referenced was set by Dr. Ronghui Zhou's Kaggle notebook which had a remarkable 97.4% accuracy. This incredible accuracy inspired us to test a variety of machine learning algorithms to attempt to do even better with the dataset.

Challenge and Achievement

The challenges that we encountered during this project were diverse. Handling the large dataset posed time challenges, requiring us to carefully select the features that were most relevant to severity and preprocessing the data to deal with outliers and missing data. In order to overcome this challenge, we chose to use a subset of the overall dataset for just the state of Tennessee, which is a larger region than used by the state-of-the-art example. We hope that by adding data points and additional pre-processing and post-processing methods, we will be able to contribute to a higher accuracy overall. Identifying the most important features for prediction was another challenge that we encountered, and we had to use various skills to determine the most important features. Finally, we were very optimistic about the number of prediction models that we could test, which became a challenge in terms of processing time.

Despite these challenges, we hope to add to the overall classification ability for this dataset by showing a high accuracy of a larger subset of the data and introducing additional steps including dimensionality reduction and classifier fusion to allow for more complex uses of the data.

Task Assignment

In order to overcome the challenges of this project, the entire group actively contributed in every project stage, working together as a group from preprocessing all the way through to post-processing while actively collaborating every step of the way. We decided against splitting the work because we all wanted to have a full understanding of every step of the project to enhance our learning.

Technical Approach

Pipeline

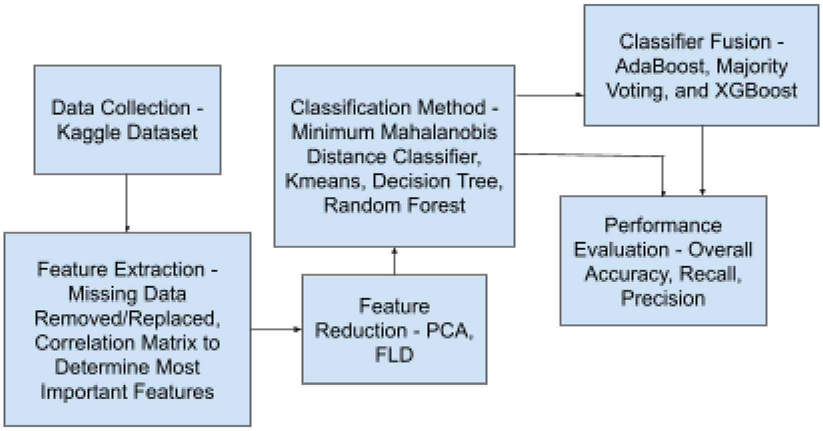

Data Collection - Kaggle Dataset
Feature Extraction - Missing Data Removed/Replaced, Correlation Matrix to Determine Most Important Features
Feature Reduction - PCA, FLD
Classification Method - Minimum Mahalanobis Distance Classifier, Kmeans, Decision Tree, Random Forest
Classifier Fusion - AdaBoost, Majority Voting, and XGBoost
Performance Evaluation - Overall Accuracy, Recall, Precision

Figure 1: Pipeline

Our project followed a specific pipeline that was built throughout the machine learning class. First, we collected our dataset from Kaggle. It had over 7.7 million entries of accidents that occurred across the United States. There were 48 features each giving a piece of important information about the situation surrounding the accident and details about the accident. After gathering our data, the next step was to extract features from the dataset. This included selecting the most correlated features to the accident severity and removing repeating data/missing data. After we moved on to feature reduction. In this step, we retained the non-reduced data as a control and applied both PCA (principal component analysis) and FLD (Fisher's linear discriminant). Once we had these three versions of our data, we began applying classifiers on all three PCA, FLD, and the non-reduced datasets. We experimented with the minimum Mahalanobis distance classifier, K-means clustering classification, Decision Tree Regressor, and Random Forest Regressor. After looking at the performance evaluation of each classification and regression method attempted, we used classifier fusion in an effort to improve our accuracy, precision, and recall. The classifier fusion methods tried included Adaboost with a decision tree base estimator, Majority voting between similar performing models, and XGBoost with a decision tree base estimator.

Algorithms

For PCA, we tested many parameters for the number of features to reduce to. We got our best accuracy across all of our classification methods for reducing to 30 features. This also provided a visualization of only the first two parameters that look a bit separable.

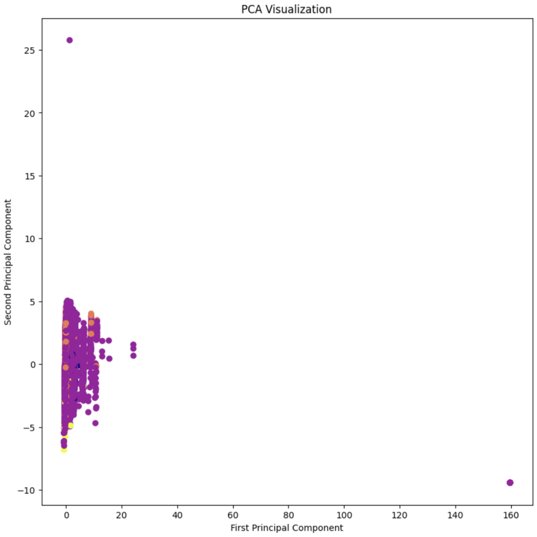


Figure 2: PCA Visualization

For FLD, after testing many different component numbers, we decided to reduce it to 3 components. We also achieved better accuracy across all of our classification methods with these parameters. The visualization of the first two components looks very separable and clustered.

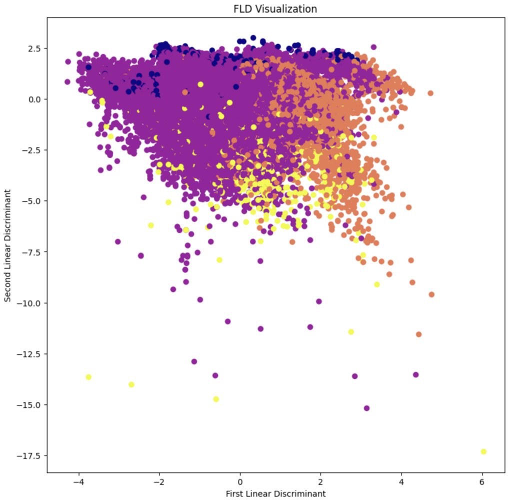


Figure 3: FLD Visualization

The minimum Mahalanobis distance classifier was created using the NearestCentroid() scikit-learn library using a custom Mahalanobis distance function we created to precompute the Mahalanobis distance between all data points.

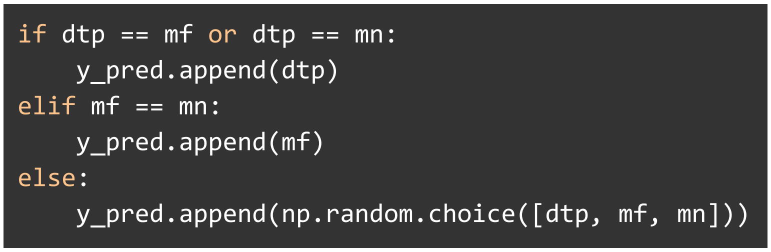
K-means was implemented using the scikit-learn library and we tested different target clusters for all the different reduced datasets. We found that 6 clusters worked best for PCA data, 5 clusters worked best for FLD data, and 8 clusters worked best for the non-reduced data. To better compare the three approaches we used a random state parameter of 21 across all three implementations.

Decision Tree was implemented using the scikit-learn library; there were no special parameters used.

We implemented Random Forest with the same parameters as Dr. Zhou to compare with our use of reduced datasets and larger datasets. His parameters for random forest were 100 estimators.

For AdaBoost, we used the scikit-learn library. The base estimator was chosen to be 50 decision trees, testing with more and less led to poorer performance. The random state was chosen to be 21 keeping consistent with our k-means approach for better comparison.

For our majority voting classifier, we decided to use three of our models that had similar performance to see if we could use the fusion to improve the performance of all three. Our basic comparison algorithm was to loop through the lists of classifications and choose the label that the majority of the classifiers identified. Below is the logic used.



```
if dtp == mf or dtp == mn:
    y_pred.append(dtp)
elif mf == mn:
    y_pred.append(mf)
else:
    y_pred.append(np.random.choice([dtp, mf, mn]))
```

For XGBoost we used the scikit-learn library and used typically used parameters found from most implementations. These parameters were 400 estimators, a learning rate of 0.1, and max depth of 5. Something interesting to note about this implementation is that the library requires the labels to begin with class zero. We had to run label encode on the labels to be severity 0 to 3 in order to apply XGBoost.

Finally, to evaluate all of our classifiers, regressions, and fusion models we used the scikit-learn library clasification_report function which reports the accuracy, precision, recall, and f1 score for each class label as well as the overall accuracy, precision, recall, and f1 score. We chose to just report overall accuracy, precision, and recall in our Appendix A evaluation table.

Experiments and Results

Experimental Design

To begin the experiment, we first imported the Kaggle dataset and analyzed the data types present in the dataset using a pandas data frame. As mentioned in the challenges section, we decided to extract only the Tennessee accident values to allow our algorithms to run in a reasonable time. To begin the feature extraction step, we extracted the fields 'Year', 'Month', 'Day', 'Hour', and 'Weekday' from 'Start_Time' to create new independent features. We then created a column called 'Time_Duration' by subtracting the ‘Start_Time’ from the ‘End_Time’. After this step, we expanded our data from 48 to 54 features.

In the feature reduction step, we started by inspecting the newly created 'Time_Duration' field for any negative values or major outlying values and replaced them with the median value. We identified that Precipitation and Wind_Chill had a large amount of data missing, so we removed the features because they had less than 50% data present. We then label encoded any string object fields using scikit-learn LabelEncode() to be integers for future steps.

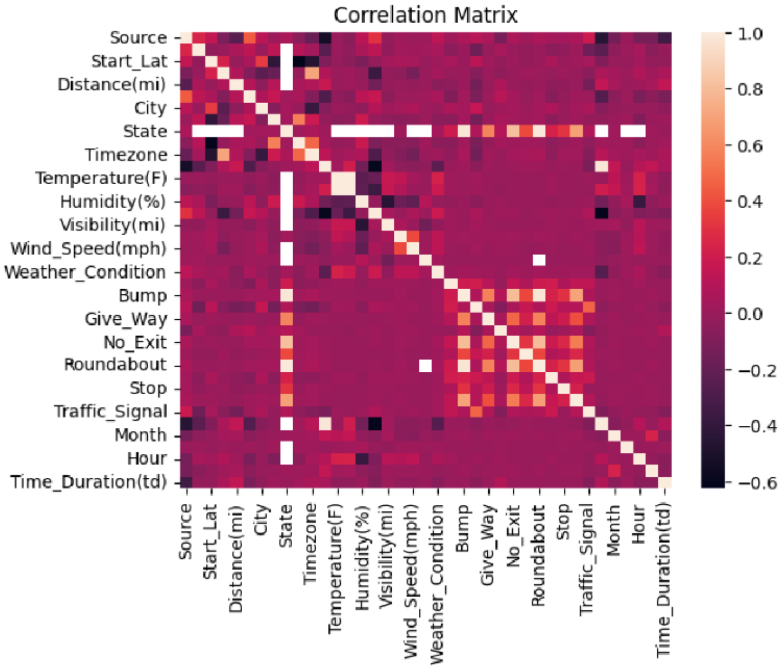


Figure 4: Correlation Matrix

Following that, we created the above correlation matrix based on the 'Severity' value to determine the most important features of the final output classification. One note about the correlation matrix is that it is only able to show a subset of the feature labels at a time, but we looked into the output to determine the following columns to remove. After seeing these results, we removed multiple columns such as 'Civil_Twilight', 'End_Lat', 'End_Lng', 'Country', and 'Airport_Code' because they had very little positive or negative correlation to the accident severity, were duplicate data from other features, or had only one value among all samples.

Before applying dimensionality reduction techniques, we created training, testing, and validation sets by splitting the data to be comprised of 10% testing, 10% validation, and 80% training data points. Finally, we applied both PCA and FLD dimensionality reduction to reduce the dataset to 30 and 3 features, respectively, down from 36.

The next stage of the pipeline was classification/regression methods. For this project, we tested Minimum Mahalanobis Distance, K-means, Decision Tree, and Random Forest methods for predicting crash severity. We applied each of these methods to the PCA and FLD reduced datasets as well as the non-reduced dataset to try and achieve the highest possible accuracy. Finally, we applied Majority Voting, AdaBoost, and XGBoost fusion methods onto subsets of our results. For Majority Voting, we first applied it to the three results of our Decision Tree classifier but found that it only decreased the overall accuracy. To try and test its capabilities, we then applied it to three unimpressive classifiers from our testing, which were the Minimum Mahalanobis Distance classifier on both the FLD reduced dataset and the non-reduced dataset, as well as the Decision Tree classifier on the PCA reduced dataset. Finally, we applied both AdaBoost and XGBoost to the Decision Tree classifier on the non-reduced dataset to compare the performances of the two fusion approaches.

The final step of the pipeline was performance evaluation. For this step, we tested the accuracy, precision, and recall of each classifier on each dataset.

Result and Comparison

Overall, our experiments yielded very promising results. We performed a comprehensive valuation of multiple machine learning algorithms applied to predict car accident severity. The primary focus was on achieving high accuracy, precision, and recall in our predictions. Here is a summary of the key outcomes and a comparative analysis of our methodologies.

For our feature extraction and reduction we utilized Principal Component Analysis (PCA) and Fisher's Linear Discriminant (FLD). These two algorithms were employed for dimensionality reduction. Interestingly, FLD consistently outperformed PCA across various classifiers, indicating that FLD better preserved discriminative features crucial for predicting accident severity.

For our classification algorithms, we used Decision Tree, Random Forest, K-means, and Minimum Distance Mahalanobis. Decision Tree and Random Forest on the non-reduced dataset emerged as standout performers, achieving approximately 90% accuracy. Random Forest exhibited a slight edge reaching an accuracy of 92%. This result supports the efficacy of Random Forest to mitigate overfitting. K-means clustering presented some challenges, especially with the uneven distribution of accident severities in the dataset. It consistently yielded suboptimal results, dropping below the accuracy of random guessing. This underscored the impact of dataset distribution on the effectiveness of clustering algorithms. Minimum Distance Mahalanobis did not do much better and was unable to cross the 50% threshold either.

We were also able to run fusion techniques with AdaBoost and XGBoost. These techniques were applied to enhance classification performance. XGBoost demonstrated superiority over AdaBoost, aligning with the current state-of-the-art trends in machine learning.

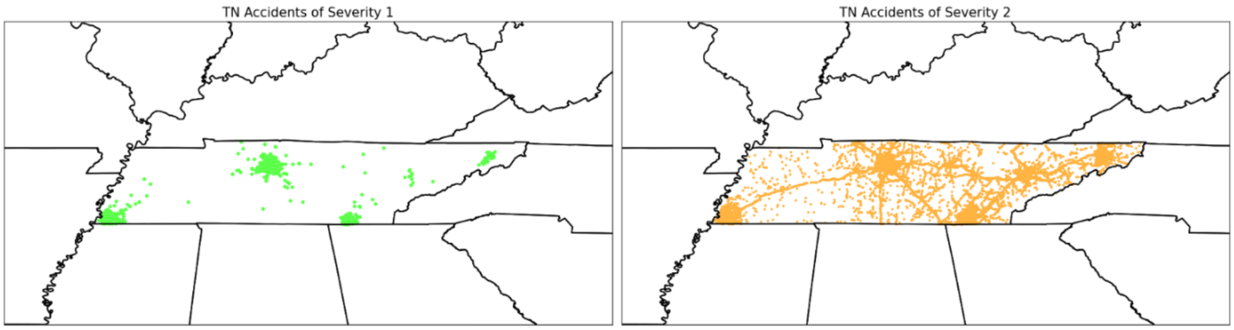
Despite our considerable efforts, we fell short of the benchmark set by Dr. Ronghui Zhou, who achieved an impressive 97.4% accuracy using Random Forest. Our best-performing model Random Forest, achieved a commendable accuracy, but further enhancements are required to approach or surpass the benchmark.

Our analysis revealed notable trends, such as the limitation of PCA in predictive tasks and the challenges associated with K-means clustering in the context of imbalanced datasets. Decision Tree and Random Forest showcased robust performance emphasizing the importance of ensemble methods in the predictive modeling scenario. Additionally, the nuanced performance differences between AdaBoost and XGBoost shed light on the evolving landscape of fusion techniques in machine learning.

Discussion of Results

There were a few major trends and standout performances that we noticed in the prediction results from this project. The first trend that we noticed was that the PCA-reduced dataset almost always resulted in the lowest performance. This trend makes sense because of the nature of our project. When trying to predict/classify the output of a system, PCA is not the best option as it is made to better represent the data, rather than allow for discrimination like with FLD.

The only case in which our performance was not the worst with the PCA dataset was with the K-means clustering algorithm. This, however, was another trend that we noticed in our results. K-means consistently produced the significantly worst results among the prediction algorithms, even dropping below the accuracy of a random guess with the non-reduced dataset. We believe that this is due to the dataset not having a uniform, or even close to uniform, distribution of accident severities. There were far more ‘Severity 2’ accidents than any other severity. This can be seen in the figure below.



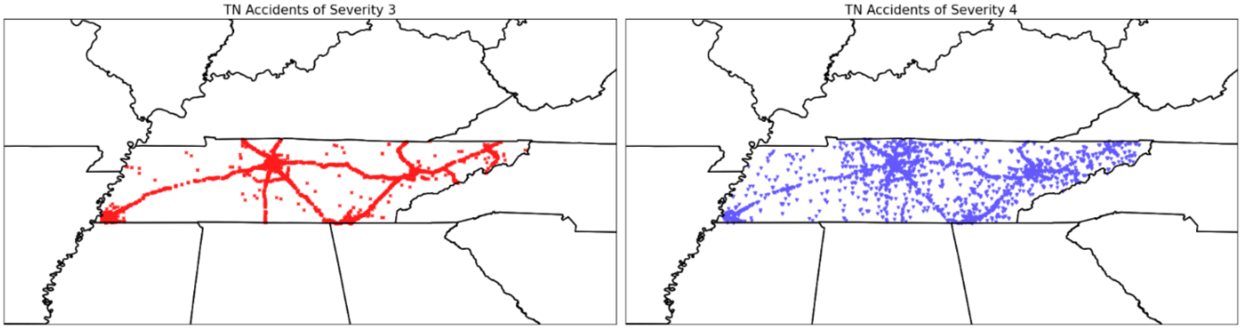


Figure 5: Severity Distribution in Tennessee

Some standout performances among the classifiers that we tested were both Decision Tree and Random Forest on the non-reduced dataset. They were both very accurate with Random Forest edging slightly ahead. We believe that the Random Forest achieves a higher accuracy because it is a combination of decision trees which helps to prevent overfitting.

The final insight that we gained from this project was that XGBoost performed better than AdaBoost, a similar fusion technique. This is likely due to the fact that XGBoost is currently state-of-the-art, while AdaBoost is slightly more outdated.

Comparison to State-of-the-Art

The state-of-the-art performance that we are comparing is by Dr. Ronghui Zhou. As stated in the background section, Dr. Zhou used a variety of prediction methods but ultimately was able to achieve an accuracy of 97.4% using Random Forest as well as over 95% performance with various regression models. Despite our best efforts, we fell short of that performance. Our best performance was achieved with Decision Tree combined with XGBoost on the non-reduced dataset and was 92.21%. Although this is over 5% worse overall accuracy than the State-of-the-Art example, we feel that we were not far from it due to our use of a larger area in our analysis. We think that this difference in scale likely negatively impacts the performance of the models, causing our accuracy to be slightly lower than we had hoped.

Summary and Future Work

Our project provided valuable insights, yet several avenues for improvement and exploration remain open for future work. Further experimentation is warranted to discern features with stronger correlations to accident severity. A deeper analysis of feature importance could guide the selection of key attributes for improved predictive modeling. The uneven distribution of accident severities, particularly the prevalence of 'Severity 2,' poses a challenge. Future work should explore techniques to handle class imbalance, potentially enhancing the accuracy of algorithms like K-means.

Regression models demonstrated promising accuracy; hence, future investigations should delve into the optimization of regression techniques. Fine-tuning parameters and exploring novel regression approaches may contribute to more accurate severity predictions. Considering the impact of dataset distribution on K-means' performance, future work could involve generating or transforming datasets to achieve a more uniform distribution of accident severities. This may uncover hidden patterns and improve clustering accuracy. The scope of experimentation can be broadened by testing additional classification and regression algorithms. Incorporating emerging techniques and algorithms may lead to further performance enhancements.

In summary, future work should focus on refining feature selection, addressing class imbalance issues, exploring regression techniques, and optimizing dataset distribution to advance the accuracy and applicability of car accident severity prediction models.

References

Moosavi, Sobhan, Mohammad Hossein Samavatian, Srinivasan Parthasarathy, and Rajiv Ramnath. “A Countrywide Traffic Accident Dataset.”, arXiv preprint arXiv:1906.05409 (2019).

Moosavi, Sobhan, Mohammad Hossein Samavatian, Srinivasan Parthasarathy, Radu Teodorescu, and Rajiv Ramnath. “Accident Risk Prediction based on Heterogeneous Sparse Data: New Dataset and Insights.” In Proceedings of the 27th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, ACM, 2019.

Ronghui, Zhou. December 31, 2019 “ML to Predict Accident Severity_PA_Mont.” Retrieved 12/5/2023 from https://www.kaggle.com/code/phip2014/ml-to-predict-accident-severity-pa-mont

Appendix A: Performance Evaluation Table

Classifier	Reduction Method	Accuracy	Precision	Recall
Minimum Mahalanobis Distance Classifier	PCA	30.91%	61%	31%
	FLD	48.40%	83%	48%
	None	48.79%	83%	49%
K-Means	PCA	28.57%	65%	29%
	FLD	34.69%	81%	35%
	None	21.38%	71%	21%
Decision Tree	PCA	54.66%	64%	55%
	FLD	76.25%	76%	76%
	None	88.88%	89%	89%
Random Forest	PCA	78.76%	64%	79%
	FLD	81.75%	80%	82%
	None	92.19%	92%	92%
Decision Tree + AdaBoost	None	91.68%	91%	92%
Decision Tree + XGBoost	None	92.21%	92%	92%
Majority Voting	Maha. FLD/None and DT PCA	50.21%	83%	50%