# Hazardous Asteroid Prediction


**Group P13 (CSC 522)**

Aakarsh Satish
asatish2@ncsu.edu

Firasat Hussain Mohammed
fmohamm8@ncsu.edu

Utkarsh Sharma
usharma3@ncsu.edu

# Introduction

- Asteroids are celestial objects that orbit the Sun, primarily in the asteroid belt between the orbits of Mars and Jupiter
- Most asteroids benign, but some have the potential to pose a significant threat to our planet
- Around 17,000 meteorites fall to Earth every year*
- NeoWs (Near Earth Object Web Service) allows users to search for near-Earth asteroids, view specific ones, and explore a dataset with 40 attributes.

## Goal:

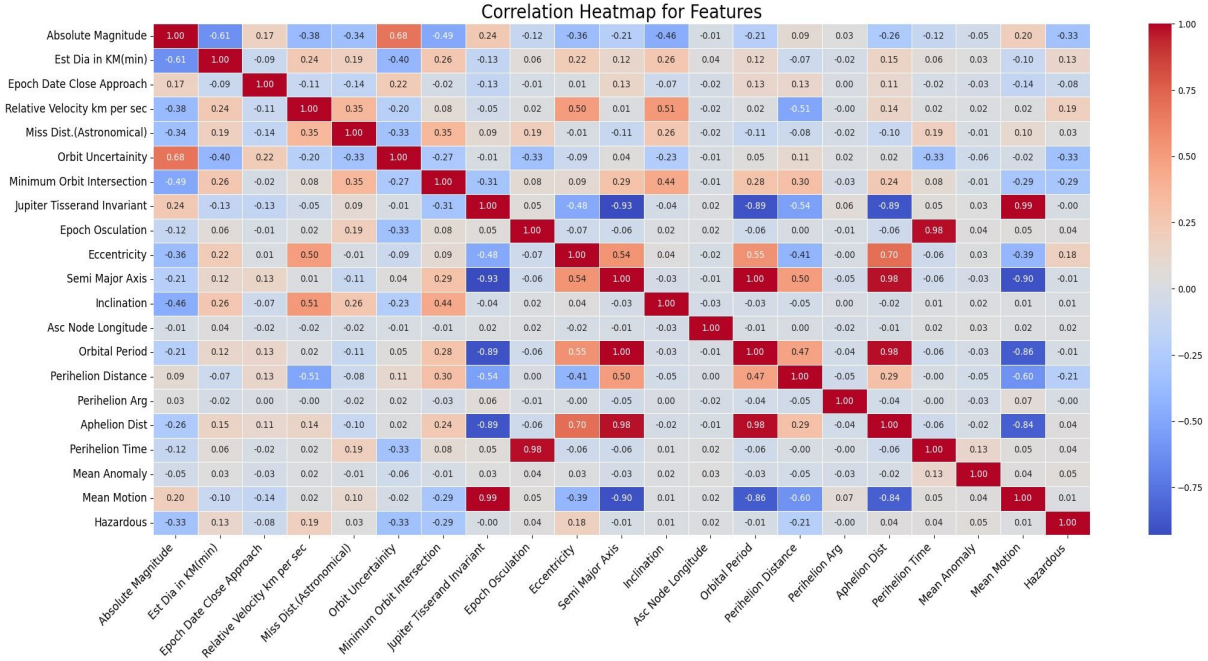
**Predicting the hazardous nature of these near-Earth objects (NEOs) using machine learning models.**

* https://www.iberdrola.com/innovation/meteorites-earth

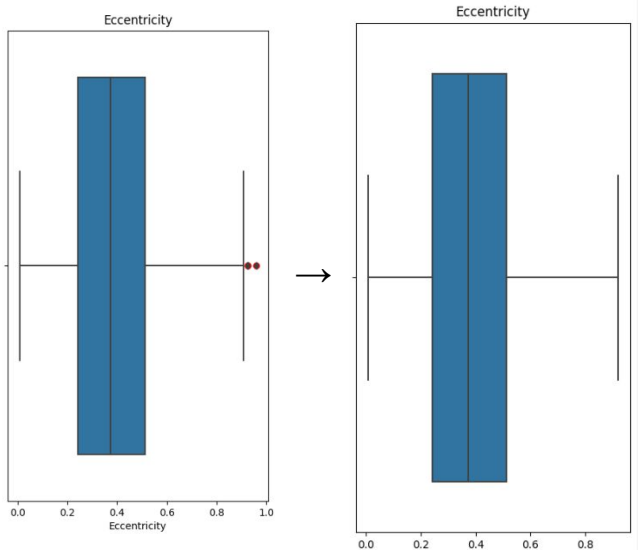
# Correlation heatmap

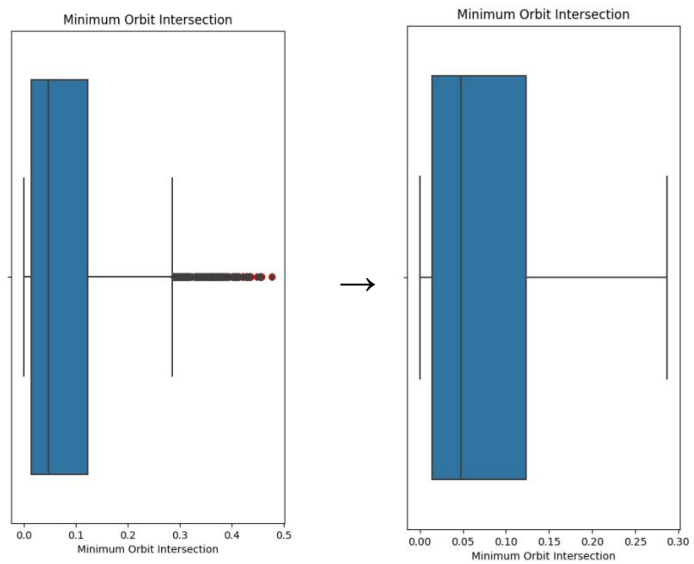
We use the correlation heatmap to narrow down the most important attributes

1. Absolute Magnitude
2. Orbit Uncertainty
3. Minimum Orbit Intersection
4. Eccentricity
5. Perihelion Distance



# Detecting and Treating Outliers

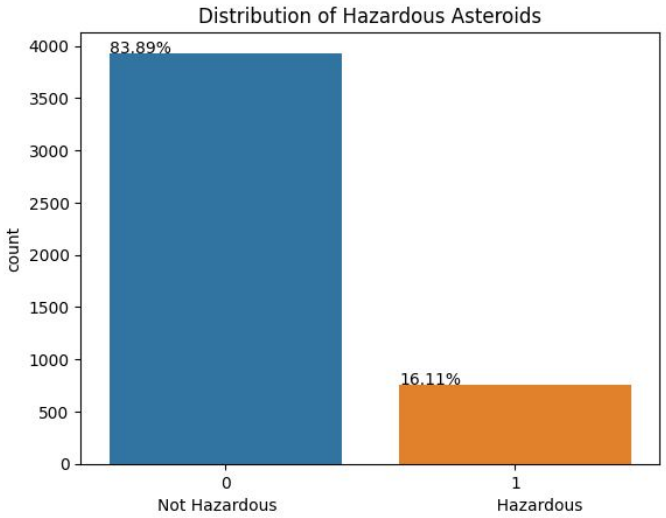


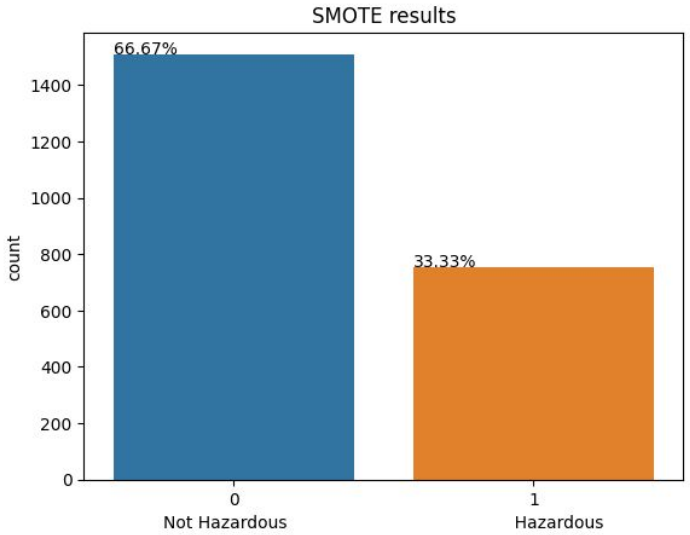
# Data Preprocessing

## SMOTE - Synthetic Minority Oversampling Technique

- To address the issue of uneven distribution among classes in our dataset




vs




* https://machinelearningmastery.com/smote-oversampling-for-imbalanced-classification/

# Models

| | |
|---|---|
| **Decision Tree** | A tree-like model that makes decisions by recursively splitting the dataset based on features, providing a transparent and interpretable representation of decision-making. |
| **Random Forest** | An ensemble learning method that combines multiple decision trees to enhance predictive accuracy by mitigating overfitting. |
| **KNN** | K-Nearest Neighbors (KNN) is a simple, non-parametric algorithm used in machine learning for classification and regression by analyzing the closest k data points in a feature space to make predictions. |
| **AdaBoost** | AdaBoost is an ensemble learning technique that sequentially combines weak learners, adjusting their weights based on their performance, to create a strong and accurate model. |

# PCA

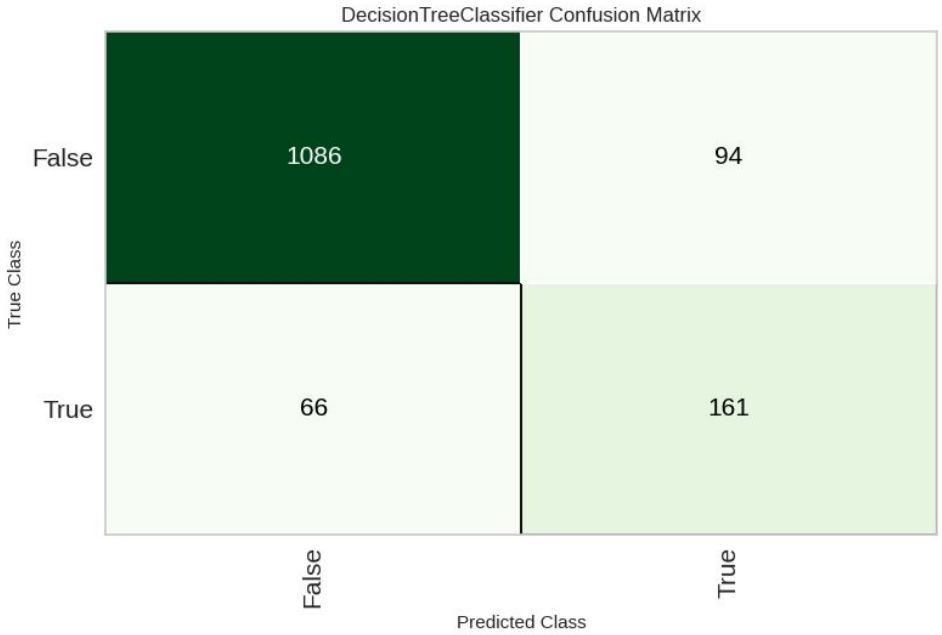
1. PCA reduces data dimensionality by transforming variables into principal components.
2. It orders components by variance captured, prioritizing the most significant features.
3. To reduce the dimensionality we used PCA and then processed with the machine learning models as shown in the next slides.

# Decision Tree

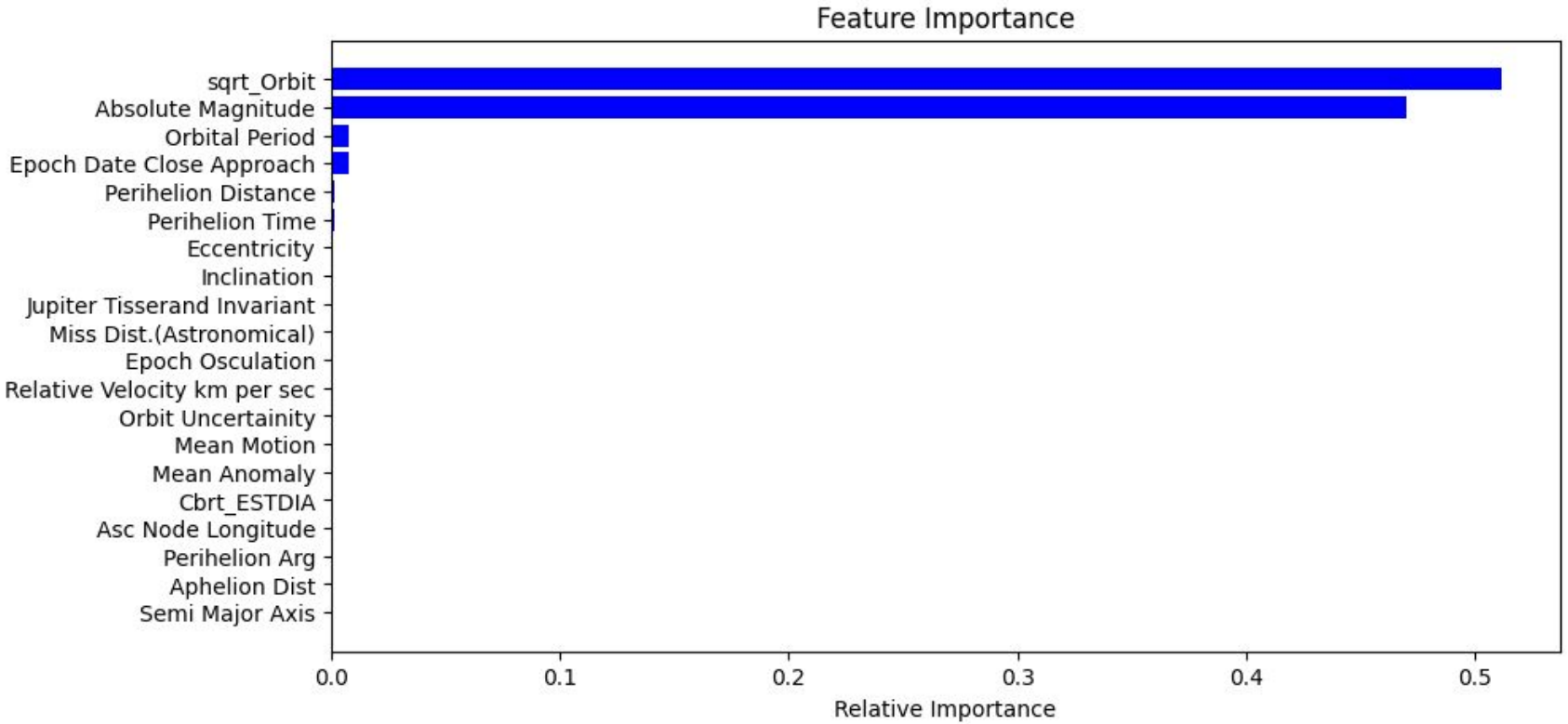
- Key criteria for splits - Gini, Entropy
- Max depth parameter to prevent overfitting
- Experimented because of its interpretability, and versatility in handling both numerical and categorical data.
- However, vulnerable to overfitting and sensitive to small variations in the training data
- Careful parameter tuning is crucial to mitigate potential drawbacks

# Decision Tree

| Fold | Accuracy | AUC | Recall | Prec. | F1 | Kappa | MCC |
|---|---|---|---|---|---|---|---|
| 0 | 0.9116 | 0.8616 | 0.7885 | 0.6949 | 0.7387 | 0.6858 | 0.6878 |
| 1 | 0.8811 | 0.7967 | 0.6731 | 0.6140 | 0.6422 | 0.5711 | 0.5719 |
| 2 | 0.8872 | 0.8261 | 0.7358 | 0.6290 | 0.6783 | 0.6104 | 0.6131 |
| 3 | 0.8811 | 0.7844 | 0.6415 | 0.6296 | 0.6355 | 0.5645 | 0.5645 |
| 4 | 0.8963 | 0.8239 | 0.7170 | 0.6667 | 0.6909 | 0.6287 | 0.6293 |
| 5 | 0.8841 | 0.7634 | 0.5849 | 0.6596 | 0.6200 | 0.5519 | 0.5533 |
| 6 | 0.8902 | 0.7822 | 0.6226 | 0.6735 | 0.6471 | 0.5822 | 0.5828 |
| 7 | 0.9146 | 0.8501 | 0.7547 | 0.7273 | 0.7407 | 0.6897 | 0.6898 |
| 8 | 0.9146 | 0.8729 | 0.8113 | 0.7049 | 0.7544 | 0.7030 | 0.7056 |
| 9 | 0.8537 | 0.7756 | 0.6604 | 0.5385 | 0.5932 | 0.5051 | 0.5090 |
| Mean | 0.8915 | 0.8137 | 0.6990 | 0.6538 | 0.6741 | 0.6092 | 0.6107 |
| Std | 0.0179 | 0.0368 | 0.0705 | 0.0514 | 0.0530 | 0.0631 | 0.0629 |



# Decision Tree

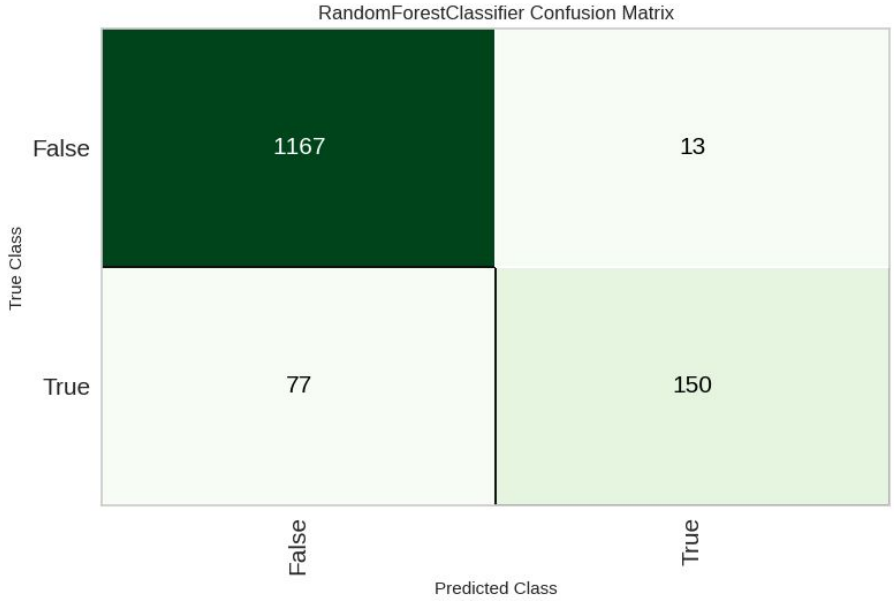


# Random Forest

- Key parameters: n_estimators and class_weight are crucial for optimization.
- Advantages- Insensitivity to feature scaling, easy to handle numerical and categorical data.
- However, has computational complexity and potential overfitting.
- But Random Forests is robust, and a balance between transparency and performance in machine learning applications.

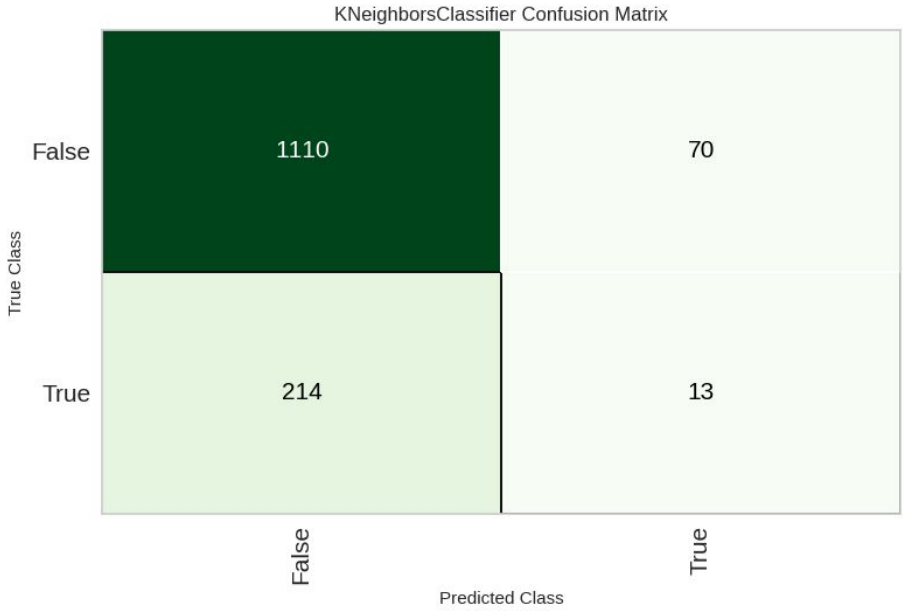
# Random Forest

| Fold | Accuracy | AUC | Recall | Prec. | F1 | Kappa | MCC |
|---|---|---|---|---|---|---|---|
| 0 | 0.9482 | 0.9843 | 0.6923 | 0.9730 | 0.8090 | 0.7800 | 0.7951 |
| 1 | 0.9177 | 0.9654 | 0.6346 | 0.8049 | 0.7097 | 0.6625 | 0.6689 |
| 2 | 0.9268 | 0.9740 | 0.6981 | 0.8222 | 0.7551 | 0.7124 | 0.7157 |
| 3 | 0.9207 | 0.9838 | 0.6038 | 0.8649 | 0.7111 | 0.6668 | 0.6813 |
| 4 | 0.9390 | 0.9809 | 0.7170 | 0.8837 | 0.7917 | 0.7564 | 0.7621 |
| 5 | 0.9268 | 0.9674 | 0.6038 | 0.9143 | 0.7273 | 0.6870 | 0.7068 |
| 6 | 0.9207 | 0.9775 | 0.6415 | 0.8293 | 0.7234 | 0.6780 | 0.6856 |
| 7 | 0.9268 | 0.9790 | 0.6038 | 0.9143 | 0.7273 | 0.6870 | 0.7068 |
| 8 | 0.9451 | 0.9826 | 0.7170 | 0.9268 | 0.8085 | 0.7771 | 0.7858 |
| 9 | 0.9146 | 0.9667 | 0.6038 | 0.8205 | 0.6957 | 0.6473 | 0.6576 |
| Mean | 0.9287 | 0.9762 | 0.6516 | 0.8754 | 0.7459 | 0.7055 | 0.7166 |
| Std | 0.0110 | 0.0070 | 0.0468 | 0.0532 | 0.0404 | 0.0463 | 0.0460 |



# K Nearest Neighbors

| Fold | Accuracy | AUC | Recall | Prec. | F1 | Kappa | MCC |
|---|---|---|---|---|---|---|---|
| 0 | 0.8018 | 0.6063 | 0.0000 | 0.0000 | 0.0000 | -0.0677 | -0.0882 |
| 1 | 0.8171 | 0.5890 | 0.1346 | 0.3182 | 0.1892 | 0.1048 | 0.1172 |
| 2 | 0.8201 | 0.5454 | 0.0377 | 0.2000 | 0.0635 | 0.0129 | 0.0185 |
| 3 | 0.8171 | 0.5645 | 0.0189 | 0.1111 | 0.0323 | -0.0154 | -0.0230 |
| 4 | 0.8354 | 0.5698 | 0.1321 | 0.4667 | 0.2059 | 0.1449 | 0.1815 |
| 5 | 0.7957 | 0.5397 | 0.0755 | 0.1818 | 0.1067 | 0.0131 | 0.0147 |
| 6 | 0.8018 | 0.5597 | 0.0943 | 0.2273 | 0.1333 | 0.0426 | 0.0479 |
| 7 | 0.8232 | 0.5487 | 0.0377 | 0.2222 | 0.0645 | 0.0185 | 0.0277 |
| 8 | 0.8110 | 0.4706 | 0.0189 | 0.0909 | 0.0312 | -0.0257 | -0.0358 |
| 9 | 0.8140 | 0.5121 | 0.0377 | 0.1667 | 0.0615 | 0.0020 | 0.0027 |
| Mean | 0.8137 | 0.5506 | 0.0587 | 0.1985 | 0.0888 | 0.0230 | 0.0263 |
| Std | 0.0111 | 0.0364 | 0.0454 | 0.1215 | 0.0652 | 0.0589 | 0.0728 |

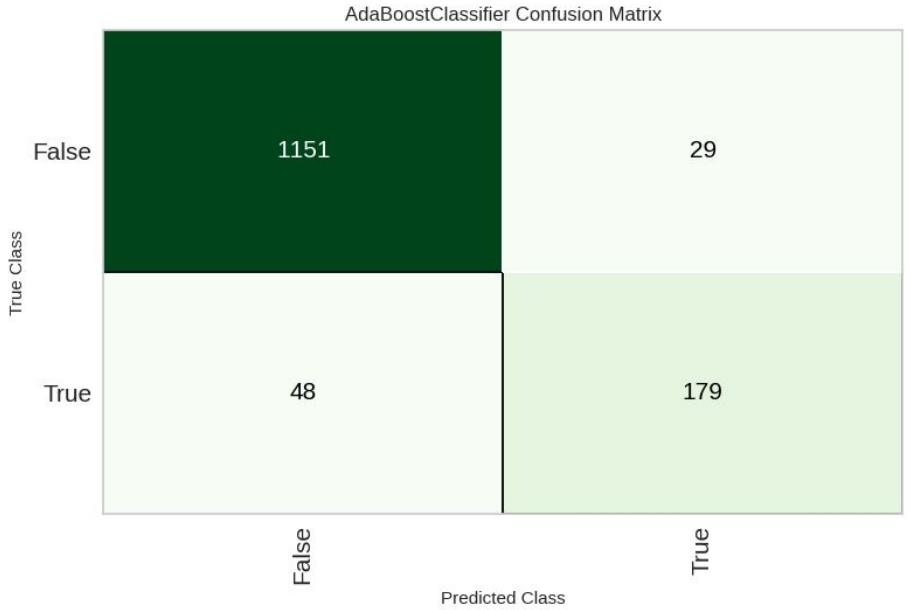


# AdaBoost

- Key parameters:  n_estimators to control the number of weak learners.
- Advantages:  Improve model accuracy by iteratively adjusting for misclassifications.
- However, is sensitive to noisy data and outliers.
- Overall, AdaBoost serves as an effective boosting algorithm for enhancing model performance in classification tasks.

# AdaBoost

| Fold | Accuracy | AUC | Recall | Prec. | F1 | Kappa | MCC |
|---|---|---|---|---|---|---|---|
| 0 | 0.9421 | 0.9798 | 0.7308 | 0.8837 | 0.8000 | 0.7665 | 0.7712 |
| 1 | 0.9421 | 0.9804 | 0.7885 | 0.8367 | 0.8119 | 0.7777 | 0.7782 |
| 2 | 0.9451 | 0.9800 | 0.7925 | 0.8571 | 0.8235 | 0.7911 | 0.7920 |
| 3 | 0.9421 | 0.9828 | 0.8113 | 0.8269 | 0.8190 | 0.7846 | 0.7846 |
| 4 | 0.9451 | 0.9807 | 0.8302 | 0.8302 | 0.8302 | 0.7975 | 0.7975 |
| 5 | 0.9329 | 0.9813 | 0.6981 | 0.8605 | 0.7708 | 0.7320 | 0.7375 |
| 6 | 0.9543 | 0.9851 | 0.8302 | 0.8800 | 0.8544 | 0.8273 | 0.8278 |
| 7 | 0.9604 | 0.9902 | 0.8302 | 0.9167 | 0.8713 | 0.8479 | 0.8494 |
| 8 | 0.9451 | 0.9803 | 0.8491 | 0.8182 | 0.8333 | 0.8005 | 0.8007 |
| 9 | 0.9177 | 0.9716 | 0.7358 | 0.7500 | 0.7429 | 0.6939 | 0.6939 |
| Mean | 0.9427 | 0.9812 | 0.7897 | 0.8460 | 0.8157 | 0.7819 | 0.7833 |
| Std | 0.0109 | 0.0044 | 0.0487 | 0.0431 | 0.0357 | 0.0419 | 0.0415 |

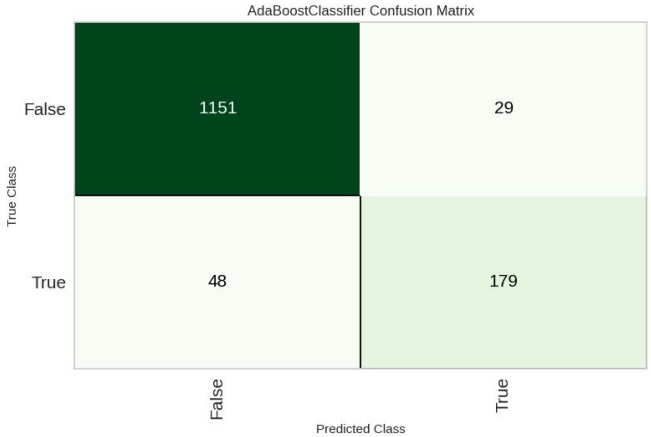


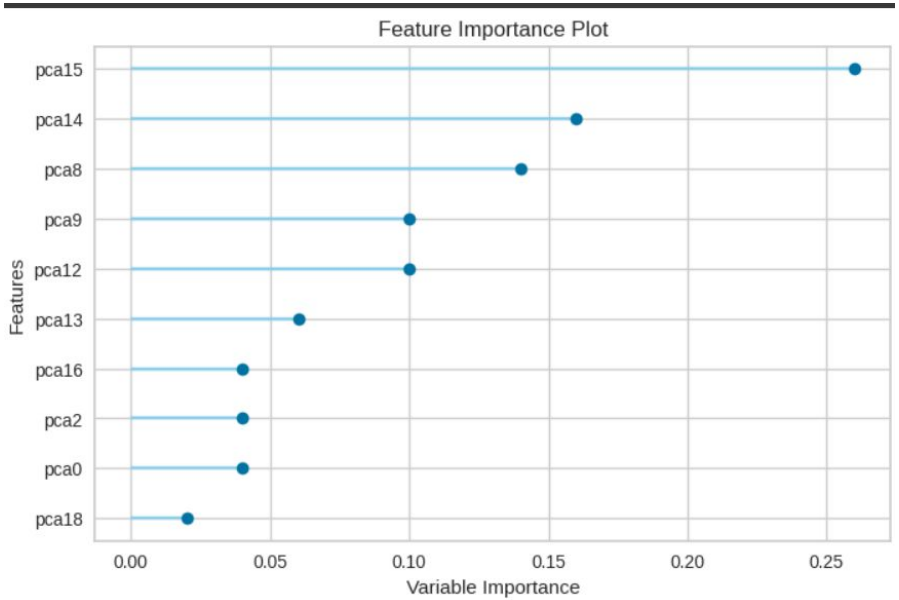
# Model Evaluation & Selection

| | Model | Accuracy | AUC | Recall | Prec. | F1 | Kappa | MCC |
|---|---|---|---|---|---|---|---|---|
| qda | Quadratic Discriminant Analysis | 0.8500 | 0.9698 | 0.8921 | 0.8157 | 0.8519 | 0.8219 | 0.8233 |
| lightgbm | Light Gradient Boosting Machine | 0.9457 | 0.9821 | 0.8087 | 0.8486 | 0.8268 | 0.7947 | 0.7959 |
| xgboost | Extreme Gradient Boosting | 0.9436 | 0.9827 | 0.8067 | 0.8380 | 0.8215 | 0.7881 | 0.7886 |
| ada | Ada Boost Classifier | 0.9427 | 0.9812 | 0.7897 | 0.8460 | 0.8157 | 0.7819 | 0.7833 |
| gbc | Gradient Boosting Classifier | 0.9393 | 0.9767 | 0.7689 | 0.8420 | 0.8028 | 0.7671 | 0.7688 |
| rf | Random Forest Classifier | 0.9287 | 0.9762 | 0.6516 | 0.8754 | 0.7459 | 0.7055 | 0.7166 |
| et | Extra Trees Classifier | 0.9183 | 0.9773 | 0.5208 | 0.9491 | 0.6717 | 0.6296 | 0.6671 |
| lda | Linear Discriminant Analysis | 0.9152 | 0.9632 | 0.7119 | 0.7499 | 0.7297 | 0.6795 | 0.6803 |
| ridge | Ridge Classifier | 0.9003 | 0.0000 | 0.4525 | 0.8626 | 0.5920 | 0.5418 | 0.5791 |
| dt | Decision Tree Classifier | 0.8915 | 0.8137 | 0.6990 | 0.6538 | 0.6741 | 0.6092 | 0.6107 |
| nb | Naive Bayes | 0.8390 | 0.5702 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| dummy | Dummy Classifier | 0.8390 | 0.5000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 | 0.0000 |
| knn | K Neighbors Classifier | 0.8137 | 0.5506 | 0.0587 | 0.1985 | 0.0888 | 0.0230 | 0.0263 |
| lr | Logistic Regression | 0.5579 | 0.5702 | 0.5604 | 0.1962 | 0.2902 | 0.0681 | 0.0874 |
| svm | SVM - Linear Kernel | 0.5146 | 0.0000 | 0.4944 | 0.1673 | 0.2494 | 0.0108 | 0.0095 |

# Conclusion and Inference

We assessed our Machine learning models and compared them with each other. So based on the higher accuracy we chose AdaBoost




Results: Accuracy: 0.942
Recall: 0.789
Precision: 0.653
F1-Score:0.674