Analyzing Credit Card Fraud Detection Datasets

Cyber 207 Final Project

Presentation by:
Coleman Davis, Amiya Prasad,
Naeemah Robert, and Ramana Venkatesh

Berkeley
UNIVERSITY OF CALIFORNIA

Hypothesis

In our project, we aim to address several key questions:

1. How does the type of model affect the accuracy and precision of fraud detection models?

2. To what extent does age and class imbalance affect false positive and false negative rates in model predictions?

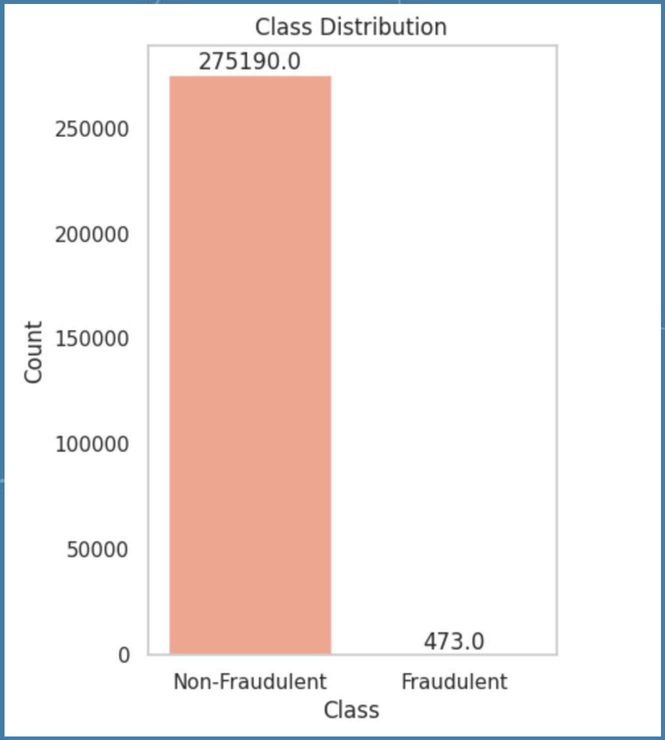
3. What is the relationship between the number of features and the overall performance metrics of these models?

Berkeley
UNIVERSITY OF CALIFORNIA

Datasets

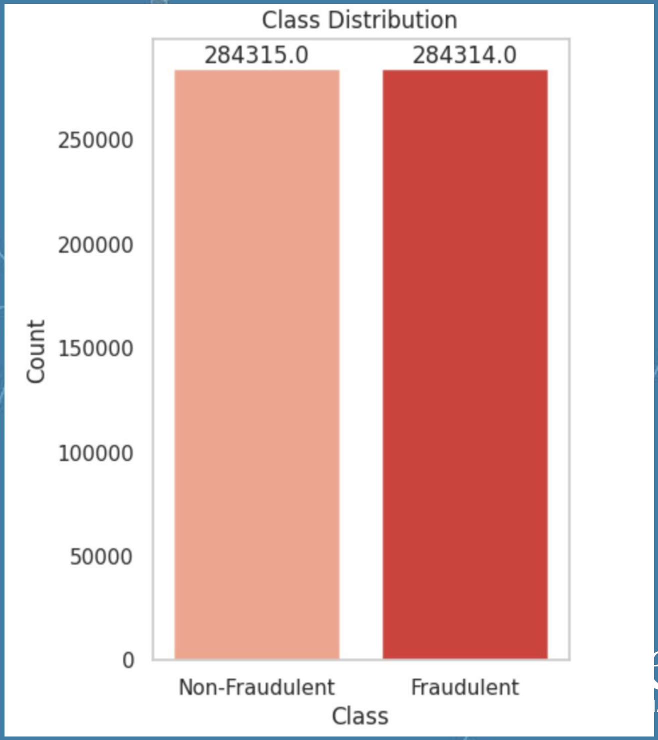
2013 Dataset

Dataset contains 284,807 credit card transactions from 2013, made by European cardholders.



2023 Dataset

Dataset contains 550,000 credit card transactions from 2023, also by European cardholders.



Models

For our analysis, we evaluated the performance of 4 models.

We chose **K-Nearest Neighbors** and **Logistic Regression** due to our familiarity with them from class and previous projects, as well as their proven effectiveness in credit card fraud detection.

We chose **Random Forest Classifier** as it is known to provide very high accuracy and is useful in handling complex data.

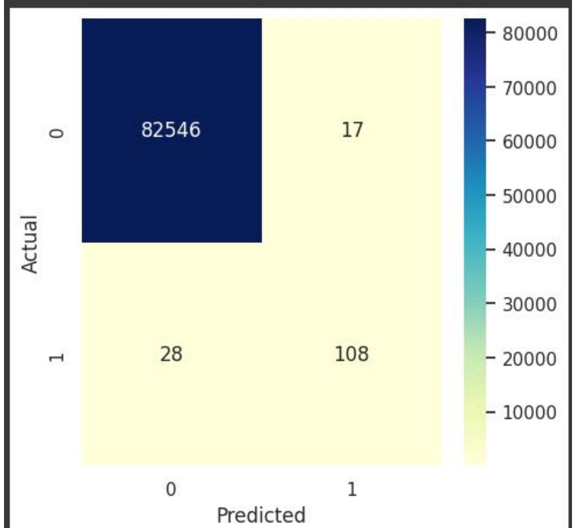
XGboost was recommended to us as it is known for handling class imbalances and large feature sets

Berkeley
UNIVERSITY OF CALIFORNIA

KNN Model Performance – 2013 Data

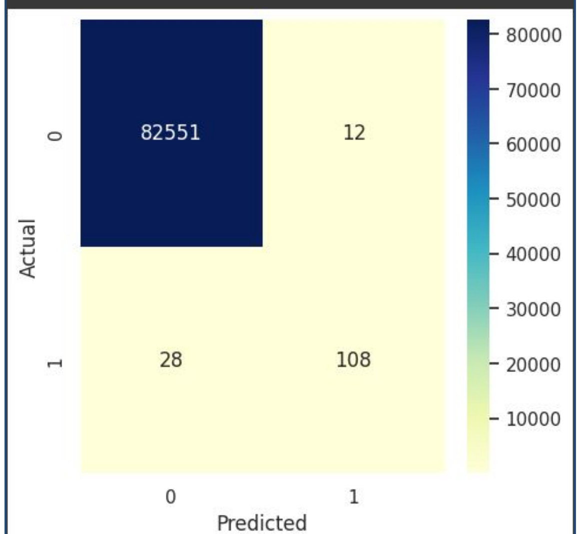
k = 1

```
Recall (2013-2013):  0.7941176470588235
Precision (2013-2013):  0.864
Accuracy (2013-2013):  0.9994558579910277

Classification Report - K Nearest Neighbors - k=1 (2013-2013):
              precision    recall  f1-score   support

           0     0.9997    0.9998    0.9997     82563
           1     0.8640    0.7941    0.8276       136

    accuracy                         0.9995     82699
   macro avg     0.9318    0.8970    0.9137     82699
weighted avg     0.9994    0.9995    0.9994     82699
```



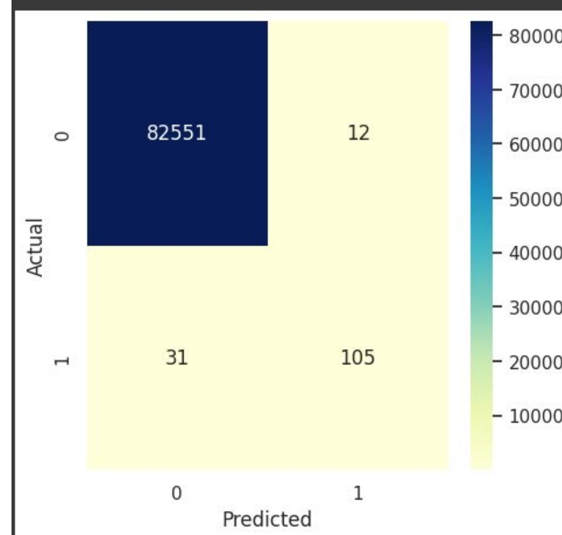
k = 3

```
Recall (2013-2013):  0.7941176470588235
Precision (2013-2013):  0.9
Accuracy (2013-2013):  0.9995163182142468

Classification Report - K Nearest Neighbors - k=3 (2013-2013):
              precision    recall  f1-score   support

           0     0.9997    0.9999    0.9998     82563
           1     0.9000    0.7941    0.8438       136

    accuracy                         0.9995     82699
   macro avg     0.9498    0.8970    0.9218     82699
weighted avg     0.9995    0.9995    0.9995     82699
```



k = 5

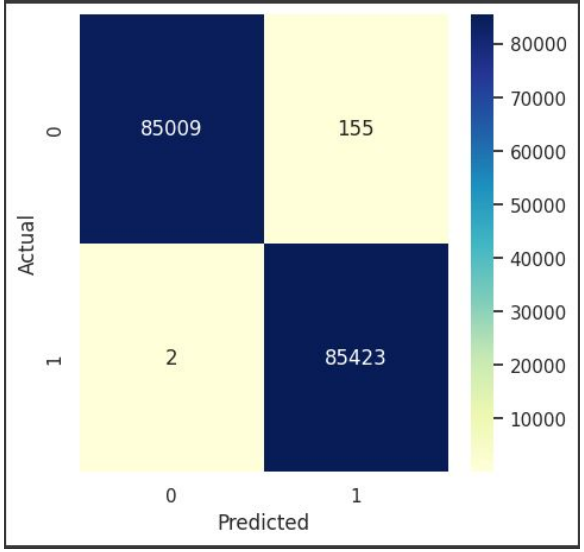
```
Recall (2013-2013):  0.7720588235294118
Precision (2013-2013):  0.8974358974358975
Accuracy (2013-2013):  0.9994800420803154

Classification Report - K Nearest Neighbors - k=5 (2013-2013):
              precision    recall  f1-score   support

           0     0.9996    0.9999    0.9997     82563
           1     0.8974    0.7721    0.8300       136

    accuracy                         0.9995     82699
   macro avg     0.9485    0.8860    0.9149     82699
weighted avg     0.9995    0.9995    0.9995     82699
```



KNN Model Performance – 2023 Data

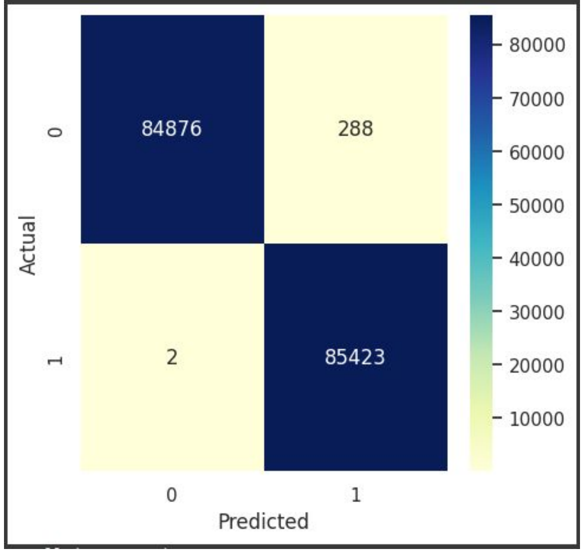
k = 1

```
Recall (2023-2023):  0.9999765876499853
Precision (2023-2023):  0.9981887868377387
Accuracy (2023-2023):  0.999079659298079

Classification Report - K Nearest Neighbors - k=1 (2023-2023):
              precision    recall  f1-score   support

           0     1.0000    0.9982    0.9991     85164
           1     0.9982    1.0000    0.9991     85425

    accuracy                         0.9991    170589
   macro avg     0.9991    0.9991    0.9991    170589
weighted avg     0.9991    0.9991    0.9991    170589
```



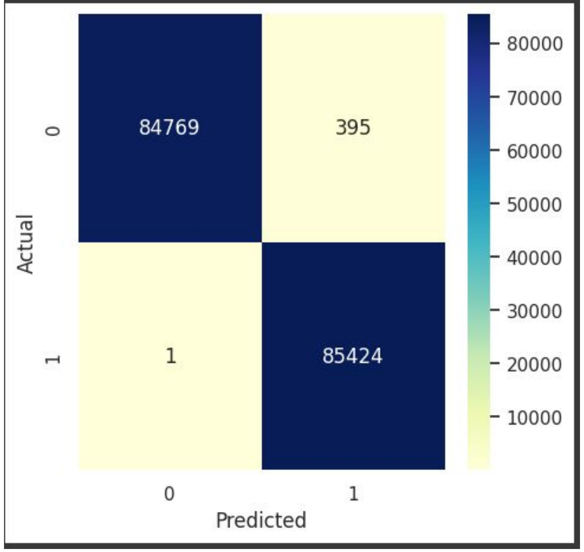
k = 3

```
Recall (2023-2023):  0.9999765876499853
Precision (2023-2023):  0.9966398711950625
Accuracy (2023-2023):  0.9983000076206555

Classification Report - K Nearest Neighbors - k=3 (2023-2023):
              precision    recall  f1-score   support

           0     1.0000    0.9966    0.9983     85164
           1     0.9966    1.0000    0.9983     85425

    accuracy                         0.9983    170589
   macro avg     0.9983    0.9983    0.9983    170589
weighted avg     0.9983    0.9983    0.9983    170589
```



k = 5

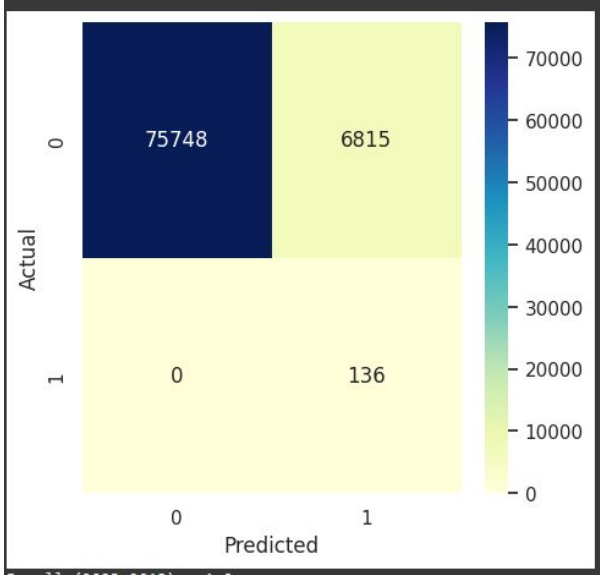
```
Recall (2023-2023):  0.9999882938249927
Precision (2023-2023):  0.9953972896444844
Accuracy (2023-2023):  0.9976786310957917

Classification Report - K Nearest Neighbors - k=5 (2023-2023):
              precision    recall  f1-score   support

           0     1.0000    0.9954    0.9977     85164
           1     0.9954    1.0000    0.9977     85425

    accuracy                         0.9977    170589
   macro avg     0.9977    0.9977    0.9977    170589
weighted avg     0.9977    0.9977    0.9977    170589
```



KNN Model Performance – Trained 2023, Tested 2013

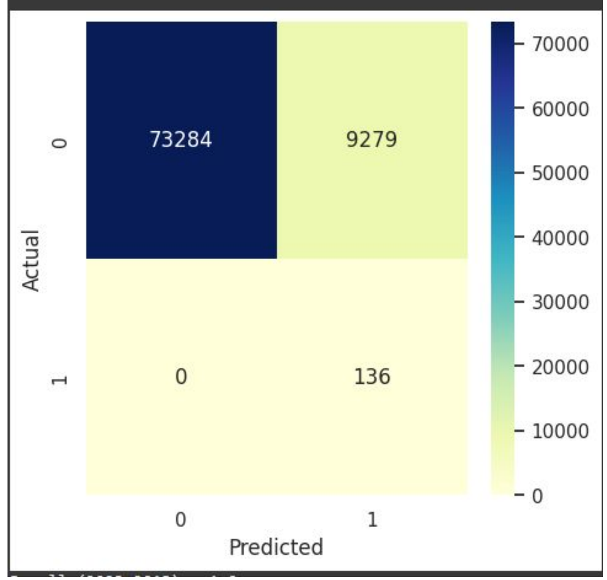
k = 1

```
Recall (2023-2013):  1.0
Precision (2023-2013):  0.019565530139548266
Accuracy (2023-2013):  0.9175927157523066

Classification Report - K Nearest Neighbors - k=1 (2023-2013):
              precision    recall  f1-score   support

           0     1.0000    0.9175    0.9570     82563
           1     0.0196    1.0000    0.0384       136

    accuracy                         0.9176     82699
   macro avg     0.5098    0.9587    0.4977     82699
weighted avg     0.9984    0.9176    0.9554     82699
```



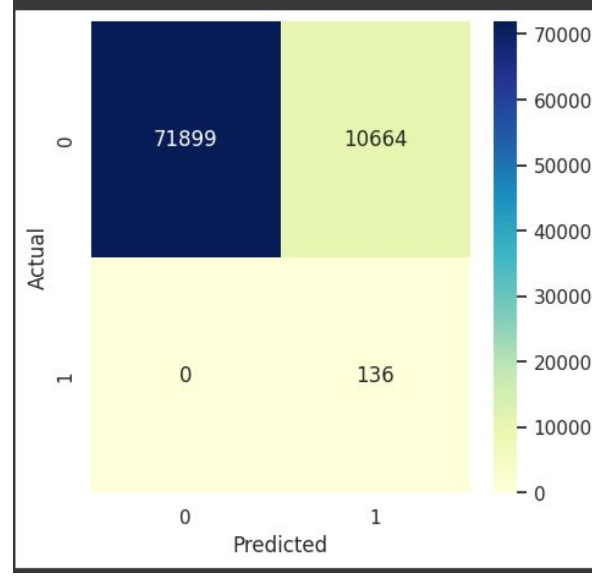
k = 3

```
Recall (2023-2013):  1.0
Precision (2023-2013):  0.014445034519383962
Accuracy (2023-2013):  0.8877979177499123

Classification Report - K Nearest Neighbors - k=3 (2023-2013):
              precision    recall  f1-score   support

           0     1.0000    0.8876    0.9405     82563
           1     0.0144    1.0000    0.0285       136

    accuracy                         0.8878     82699
   macro avg     0.5072    0.9438    0.4845     82699
weighted avg     0.9984    0.8878    0.9390     82699
```



k = 5

```
Recall (2023-2013):  1.0
Precision (2023-2013):  0.012592592592592593
Accuracy (2023-2013):  0.8710504359182094

Classification Report - K Nearest Neighbors - k=5 (2023-2013):
              precision    recall  f1-score   support

           0     1.0000    0.8708    0.9310     82563
           1     0.0126    1.0000    0.0249       136

    accuracy                         0.8711     82699
   macro avg     0.5063    0.9354    0.4779     82699
weighted avg     0.9984    0.8711    0.9295     82699
```

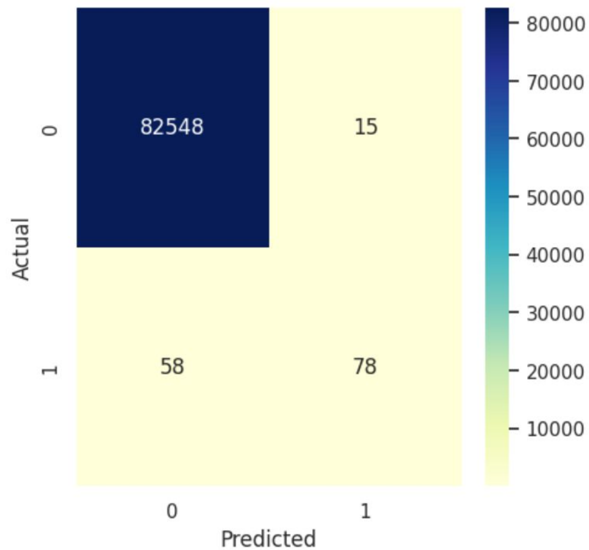


Logistic Regression Model Performance

Train and Test on 2013 Dataset

Classification Report - Logistic Regression (2013-2013):

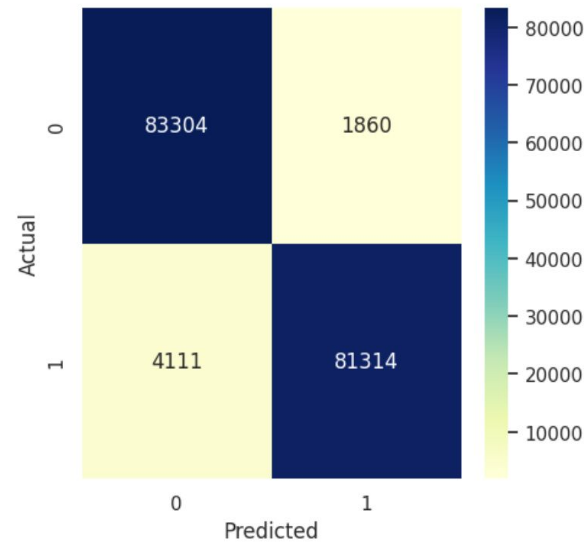
	precision	recall	f1-score	support
0	0.9993	0.9998	0.9996	82563
1	0.8387	0.5735	0.6812	136
accuracy			0.9991	82699
macro avg	0.9190	0.7867	0.8404	82699
weighted avg	0.9990	0.9991	0.9990	82699



Train and Test on 2023 Dataset

Classification Report - Logistic Regression (2023-2023):

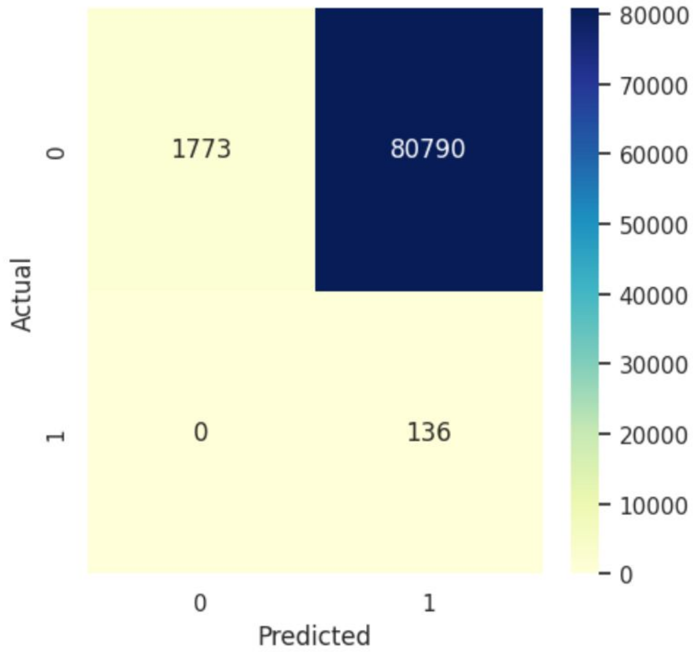
	precision	recall	f1-score	support
0	0.9530	0.9782	0.9654	85164
1	0.9776	0.9519	0.9646	85425
accuracy			0.9650	170589
macro avg	0.9653	0.9650	0.9650	170589
weighted avg	0.9653	0.9650	0.9650	170589



Logistic Regression Model Performance

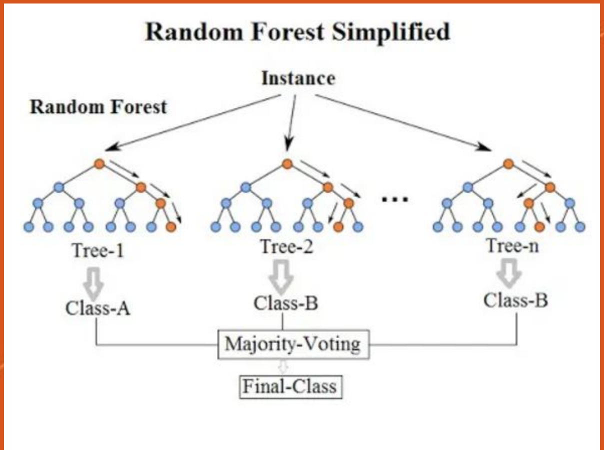
Train on 2023 dataset and Test on 2013 Dataset

```
Classification Report - Logistic Regression (2023-2013):
              precision    recall  f1-score   support

           0     1.0000    0.0215    0.0420     82563
           1     0.0017    1.0000    0.0034       136

    accuracy                         0.0231     82699
   macro avg     0.5008    0.5107    0.0227     82699
weighted avg     0.9984    0.0231    0.0420     82699
```

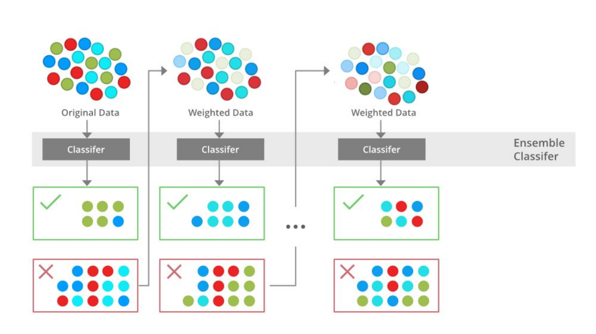


Random Forest & XGBoost

- **Random Forest**
  - ML algorithm that utilizes the output of multiple decision trees to reach one result
  - Able to handle both regression and classification problems
  - Works well with large datasets and imbalanced data



- **XGBoost (Extreme Gradient Boosting)**
  - Based on the Gradient Boosting framework
  - Able to handle large datasets with high-dimensional features
  - Includes L1 and L2 regularization

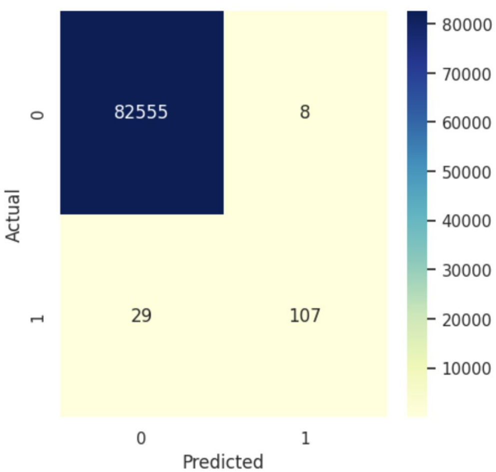


Berkeley
UNIVERSITY OF CALIFORNIA

XGBoost Model Performance

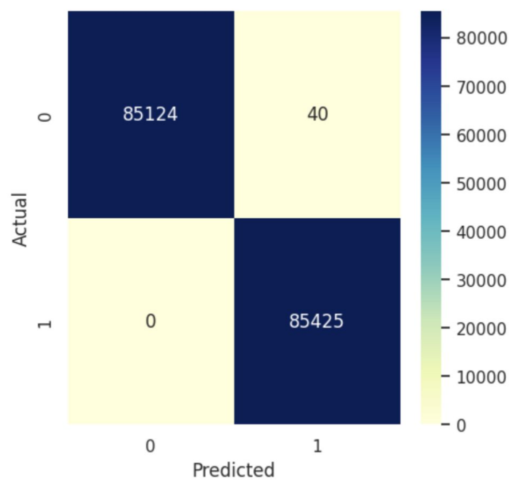
Train 2013, Test 2013

```
Recall (2013-2013):  0.7867647058823529
Precision (2013-2013):  0.9304347826086956
Accuracy (2013-2013):  0.9995525943481783

Classification Report - XGBoost (2013-2013):
              precision    recall  f1-score   support

           0     0.9996    0.9999    0.9998     82563
           1     0.9304    0.7868    0.8526       136

    accuracy                         0.9996     82699
   macro avg     0.9650    0.8933    0.9262     82699
weighted avg     0.9995    0.9996    0.9995     82699
```



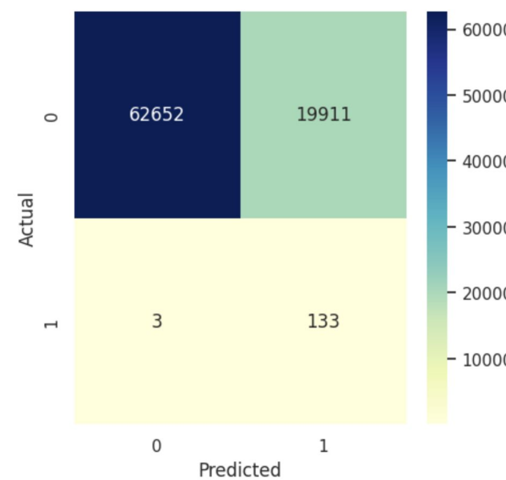
Train 2023, Test 2023

```
Recall (2023-2023):  1.0
Precision (2023-2023):  0.999531972152343
Accuracy (2023-2023):  0.9997655182925042

Classification Report - XGBoost (2023-2023):
              precision    recall  f1-score   support

           0     1.0000    0.9995    0.9998     85164
           1     0.9995    1.0000    0.9998     85425

    accuracy                         0.9998    170589
   macro avg     0.9998    0.9998    0.9998    170589
weighted avg     0.9998    0.9998    0.9998    170589
```



Train 2023, Test 2013

```
Recall (2023-2013):  0.9779411764705882
Precision (2023-2013):  0.006635402115346238
Accuracy (2023-2013):  0.7591990229627927

Classification Report - XGBoost (2023-2013):
              precision    recall  f1-score   support

           0     1.0000    0.7588    0.8629     82563
           1     0.0066    0.9779    0.0132       136

    accuracy                         0.7592     82699
   macro avg     0.5033    0.8684    0.4380     82699
weighted avg     0.9983    0.7592    0.8615     82699
```

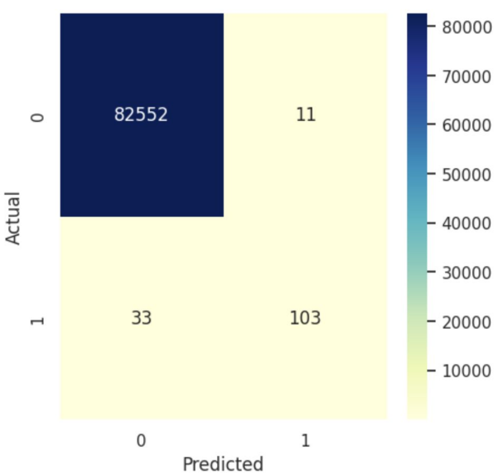


Berkeley
UNIVERSITY OF CALIFORNIA

Random Forest Model Performance

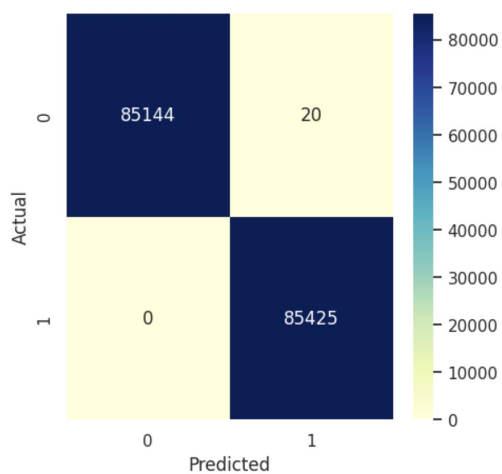
Train 2013, Test 2013

```
Recall (2013-2013):  0.7573529411764706
Precision (2013-2013):  0.9035087719298246
Accuracy (2013-2013):  0.9994679500356716

Classification Report - Random Forest (2013-2013):
              precision    recall  f1-score   support

           0     0.9996    0.9999    0.9997     82563
           1     0.9035    0.7574    0.8240       136

    accuracy                         0.9995     82699
   macro avg     0.9516    0.8786    0.9119     82699
weighted avg     0.9994    0.9995    0.9994     82699
```



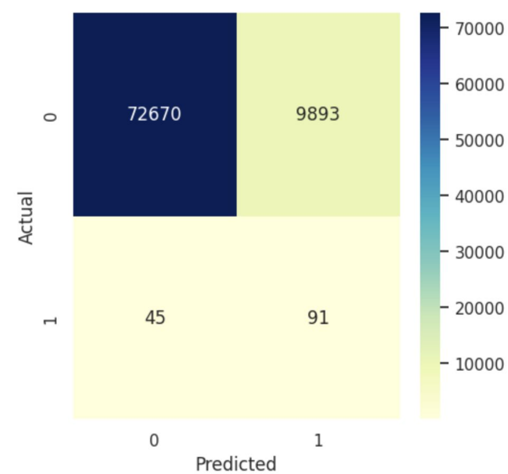
Train 2023, Test 2023

```
Recall (2023-2023):  1.0
Precision (2023-2023):  0.9997659313008368
Accuracy (2023-2023):  0.9998827591462521

Classification Report - Random Forest (2023-2023):
              precision    recall  f1-score   support

           0     1.0000    0.9998    0.9999     85164
           1     0.9998    1.0000    0.9999     85425

    accuracy                         0.9999    170589
   macro avg     0.9999    0.9999    0.9999    170589
weighted avg     0.9999    0.9999    0.9999    170589
```



Train 2023, Test 2013

```
Recall (2023-2013):  0.6691176470588235
Precision (2023-2013):  0.009114583333333334
Accuracy (2023-2013):  0.8798292603296292

Classification Report - Random Forest (2023-2013):
              precision    recall  f1-score   support

           0     0.9994    0.8802    0.9360     82563
           1     0.0091    0.6691    0.0180       136

    accuracy                         0.8798     82699
   macro avg     0.5042    0.7746    0.4770     82699
weighted avg     0.9978    0.8798    0.9345     82699
```



Feature Removal Impact on Model Performance

We used Recursive Feature Elimination (RFE) to select the top 15 features that contribute to the model's performance.

Selected features: ['V1', 'V3', 'V4', 'V7', 'V8', 'V9', 'V10', 'V11', 'V12', 'V14', 'V16', 'V17', 'V18', 'V22', 'V23']

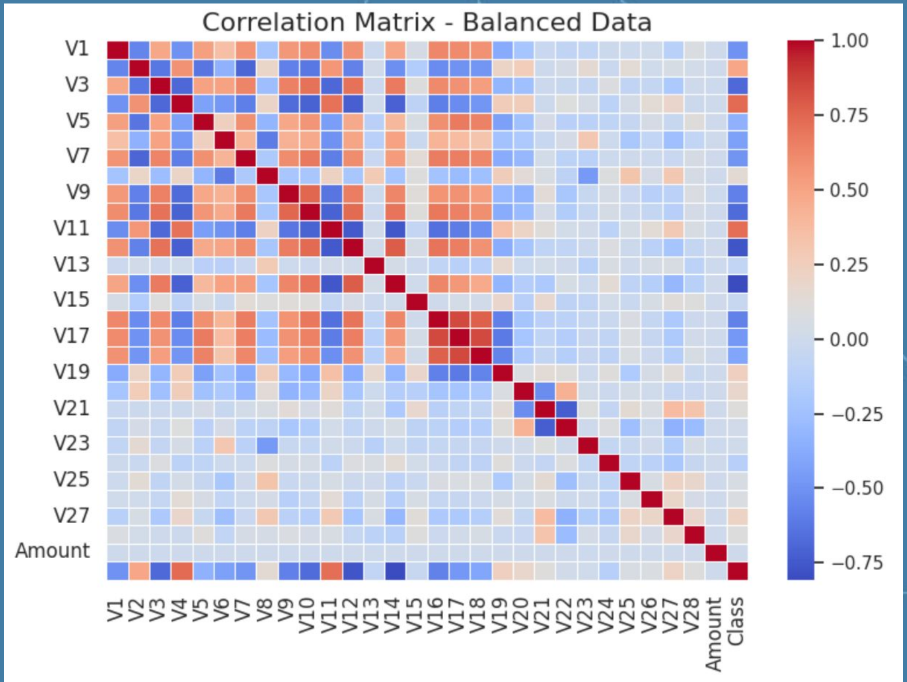
Logistic Regression Model Performance

```
Accuracy:  0.9649977431135653
F1 Score:  0.9649923544424698

Classification Report:
              precision    recall  f1-score   support

           0       0.95      0.98      0.97     85164
           1       0.98      0.95      0.96     85425

    accuracy                           0.96    170589
   macro avg       0.97      0.97      0.96    170589
weighted avg       0.97      0.96      0.96    170589
```

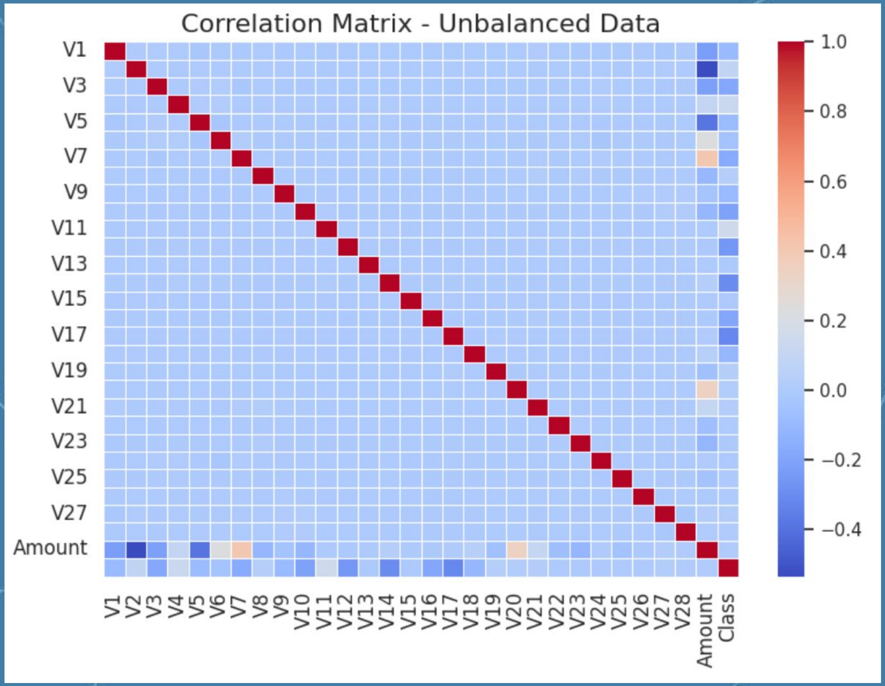
Berkeley
UNIVERSITY OF CALIFORNIA

Feature Removal Impact on Model Performance

2023 Dataset Correlation Matrix



2013 Dataset Correlation Matrix



Feature Removal Impact on Model Performance

[Initial Model Performance Metrics] Accuracy: 0.9649 ; F1 score: 0.9649

Impact of removing Feature 'V17'

Accuracy: 0.9619729290868696
F1 Score: 0.9619623606798576

Classification Report:

	precision	recall	f1-score	support
0	0.9463	0.9794	0.9626	85164
1	0.9787	0.9446	0.9614	85425
accuracy			0.9620	170589
macro avg	0.9625	0.9620	0.9620	170589
weighted avg	0.9625	0.9620	0.9620	170589

Impact of removing Feature 'V16'

Accuracy: 0.9649977431135653
F1 Score: 0.964991341503686

Classification Report:

	precision	recall	f1-score	support
0	0.9520	0.9793	0.9654	85164
1	0.9787	0.9508	0.9645	85425
accuracy			0.9650	170589
macro avg	0.9654	0.9650	0.9650	170589
weighted avg	0.9654	0.9650	0.9650	170589

Impact of removing Feature 'V17', 'V18', 'V16'

Accuracy: 0.9593701821336663
F1 Score: 0.9593563082999921

Classification Report:

	precision	recall	f1-score	support
0	0.9422	0.9786	0.9601	85164
1	0.9778	0.9402	0.9586	85425
accuracy			0.9594	170589
macro avg	0.9600	0.9594	0.9594	170589
weighted avg	0.9601	0.9594	0.9594	170589

Impact of removing Feature 'V4'

Accuracy: 0.9516264237436177
F1 Score: 0.9515987457520952

Classification Report:

	precision	recall	f1-score	support
0	0.9303	0.9763	0.9527	85164
1	0.9751	0.9270	0.9505	85425
accuracy			0.9516	170589
macro avg	0.9527	0.9517	0.9516	170589
weighted avg	0.9527	0.9516	0.9516	170589

Conclusion

- **Overall, models with the ability to perform feature importance calculations performed better**
  - This trend heavily depends on the type of dataset and the balance
- **The balance of the dataset drastically changes the outcome of the models**
  - The nuance of age was unable to be determined due to the imbalance
- **Feature Engineering has the potential to slightly improve the performance of the models**
  - More research required to improve marginal performance increase

Berkeley
UNIVERSITY OF CALIFORNIA

References

[1] Baishya, Manjit. "Does Removal of Highly Correlated Features Always Improve Model Performance?" Medium, June 17, 2024. https://medium.com/@datacodedesign/does-removal-of-highly-correlated-features-always-improve-model-performance-8d820d30b71d#:~:text=Understanding%20Feature%20Correlation,impact%20generalization%20to%20new%20data .

[2] Brownlee, Jason. "Recursive Feature Elimination (RFE) for Feature Selection in Python." MachineLearningMastery.com, August 27, 2020. https://machinelearningmastery.com/rfe-feature-selection-in-python/ .

[3] "Credit Card Fraud Detection." Kaggle, March 23, 2018. https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud?resource=download .

[4] "Copy of Cyber Attack Detection with Random Forest ." Google colab. Accessed December 8, 2024. https://colab.research.google.com/drive/1AxXUNXAI6Edggr9ailj9g6kBPYIqVoj-?usp=drive_link .

[5] Elgiriyewithana, Nidula. "Credit Card Fraud Detection Dataset 2023." Kaggle, September 18, 2023. https://www.kaggle.com/datasets/nelgiriyewithana/credit-card-fraud-detection-dataset-2023 .

Berkeley
UNIVERSITY OF CALIFORNIA