cs7641-team60

Proposal Report

Introduction and Background

Credit card fraud is a substantial financial problem that causes immense losses to individuals and businesses around the world. Utilizing the Credit Card Fraud Detection Dataset 2023 that contains over 550 thousand records of credit card transactions by European card holders, we will develop advanced fraud detection models. The selected dataset contains information on the various transaction attributes like time, place, etc, and this information is anonymized to protect the individual's credit card data. Furthermore, the data is classified and labeled as either fraudulent or not. This dataset encompasses the necessary data points to create advanced fraud detection models.

Problem Definition

Credit card fraud is a growing problem that continues to cause substantial financial losses globally. In 2019, card fraud losses amounted to $28.65 billion, and by 2021, this number increased to $32.34 billion. [1] Projections estimate global losses from card fraud to reach $397.4 billion over the next decade, with $165.1 billion expected in the U.S. alone. Beyond the direct financial impact, U.S. retail and e-commerce businesses face a significant burden, incurring $3.75 in costs for every $1 of fraud, a 20% increase from 2019. These trends highlight the need for effective fraud detection and prevention measures. Companies like Visa and PayPal have made strides using machine learning and AI, with Visa's system preventing up to $25 billion in annual losses and PayPal improving detection accuracy by 50% [2, 3]. This is why, developing and deploying advanced fraud detection models is crucial to minimizing these escalating risks.

Methods

The dataset contains the transaction amount for each entry, as well as 28 quantitative, anonymized features such as time, location, etc. There are a couple steps we can take to prepare our data for exploratory analysis and then modeling:

Cleaning steps:

1. Data cleaning: we will replace missing feature entries either with 0, or if that results in a bias, we will delete the rows instead
2. Standard scaling: we will scale our data to a distribution with mean 0 and standard deviation 1, helping us in eliminating bias between feature distributions and in accounting for outliers that can potentially impact model training and inference

3. Dimensionality reduction: we will perform Principal Component Analysis (PCA) on the 28 features to see if we can minimize the features we would need

ML Algorithms/Models:

1. Supervised: K-nearest neighbors
2. Supervised: Random forest classifier
3. Unsupervised: k=2 clustering

Potential Results and Discussion

We intend to evaluate our model(s) efficacy using the following metrics:

- Success rate (%) at predicting an entry's legitimacy or fraudulent status;
- False positive rate;
- False negative rate.

The successful identification rate can yield a generalized benchmark reference. More significant for consumers, false positive and false negative rates will play significantly in determining the potential of our project. False negatives indicate fraudulent behavior that successfully fool model our without being detected, meaning potential losses for either the user or the employing financial institution. False positives signify legitimate transactions incorrectly flagged. While fine and potentially confidence instilling in small frequencies, in larger numbers can result in dissatisfied customers due to wasted time and inconveniences. Too high of a rate would make the process inconvenient for large-scale use.

We aim to generate a model that minimizes the false negative rate first and foremost, along with minimizing the false positive rate. Given success in our development, we expect to observe a high success rate and low rates for both false positives and false negatives with an emphasis on a low false negative rate. Given current information, we'll be aiming for a success rate of at least 85% with false positive and negative rates below 10%.

Results and Discussion

As of the midterm date, we've completed 2 out of the 3 models and have run one visualization.

Why These Models

To solve the problem of credit card fraud detection, we selected **Random Forest**, **k=2 Clustering**, and **KNN (k-Nearest Neighbors)** for the following reasons:

1. **Random Forest** is perfect for handling labeled data, making it ideal for detecting fraud based on specific interactions among a large number of anonymized features. RF also mitigates overfitting, something important we wanted to keep in mind.

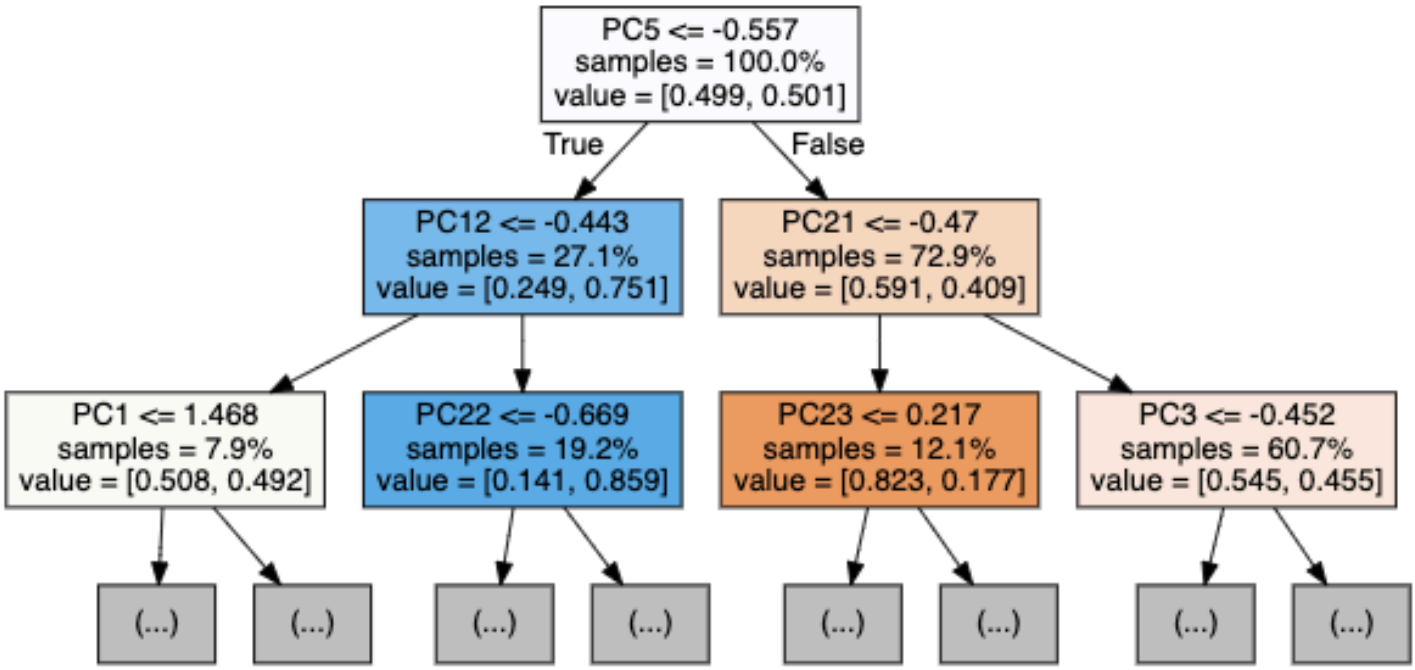
2. **k=2 Clustering** is best suited to separate transactions into the two groups you'd expect– fraudulent and legitimate— based on the similarity of other features. A k=2 approach is also useful in identifying anomalies when there are sparse ground truth labels or a higher need for validation.

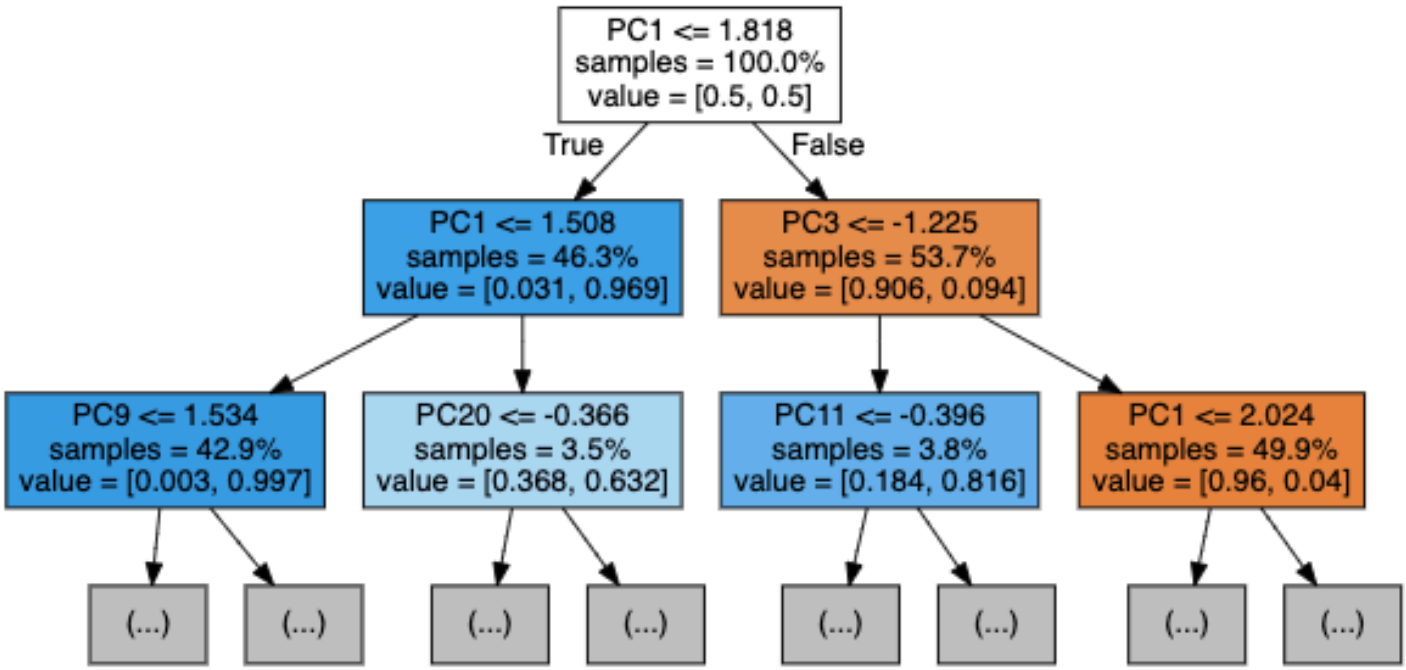
3. **KNN** quickly assigns labels to varying transactions based on their similarity to their fraudulent or legitimate neighbors, which speedily assesses any patterns in the data.

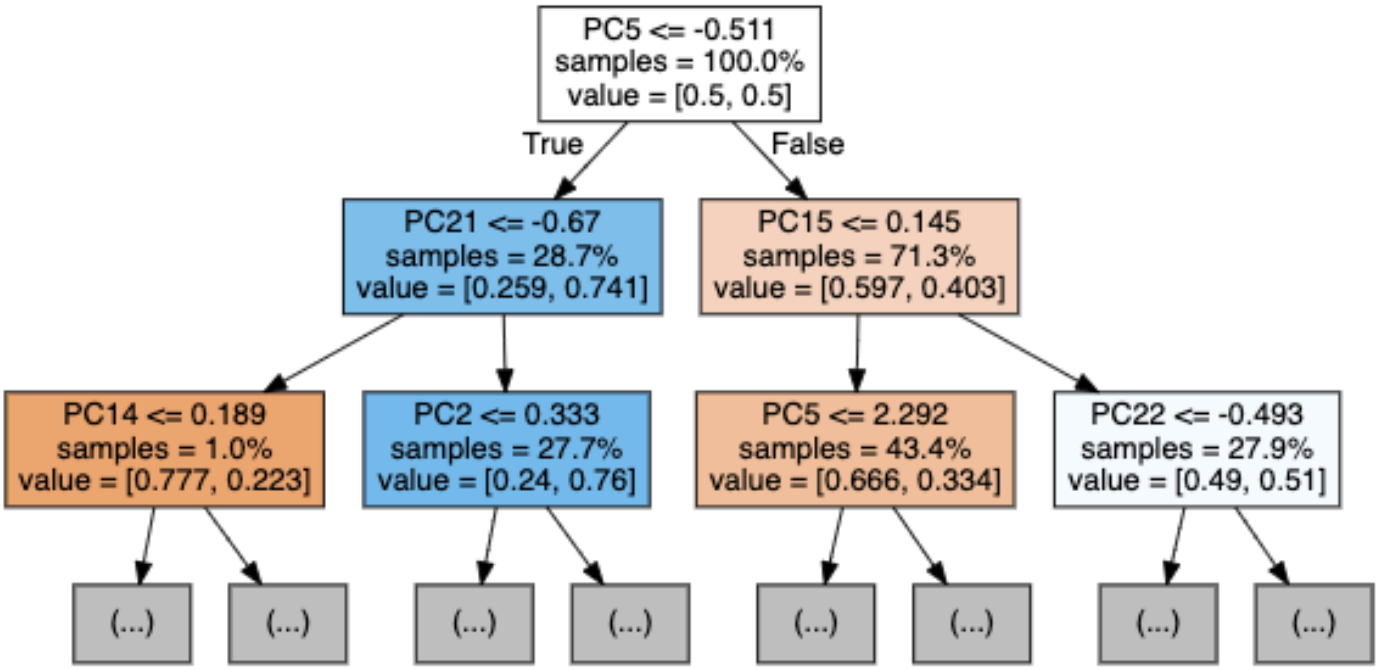
Random Forest

We'll start by discussing our "complete" model, Random Forest, which exhibited a 99.9% accuracy, and the decision tree we implemented is attached as well, with an image down below.

Seeing as how we've achieved such a high accuracy, it's safe to say that our model performed quite well, although we believe that some of these results might be the cause of overfitting. Random forest can overfit data as a result of there being too many trees present (this could be due to our 28 parameters) or if each tree is allowed to grow deeply, which captures both noise and data patterns that are specific to data rather than something that broadly applies to the trends present themselves. It goes without saying that this can lead to very high accuracy on training data due to the model becoming overly tailored on the sample data's nuances, so we acknowledge that, as it stands, our current model may struggle to generalize well to new, unseen samples. With that said, we have no other overwhelming concerns about the model's performance as things stand.


PC5 <= -0.557
samples = 100.0%
value = [0.499, 0.501]
True
False
PC12 <= -0.443
samples = 27.1%
value = [0.249, 0.751]
PC21 <= -0.47
samples = 72.9%
value = [0.591, 0.409]
PC1 <= 1.468
samples = 7.9%
value = [0.508, 0.492]
PC22 <= -0.669
samples = 19.2%
value = [0.141, 0.859]
PC23 <= 0.217
samples = 12.1%
value = [0.823, 0.177]
PC3 <= -0.452
samples = 60.7%
value = [0.545, 0.455]
(...)
(...)
(...)
(...)
(...)
(...)
(...)
(...)


PC1 <= 1.818
samples = 100.0%
value = [0.5, 0.5]
True
False
PC1 <= 1.508
samples = 46.3%
value = [0.031, 0.969]
PC3 <= -1.225
samples = 53.7%
value = [0.906, 0.094]
PC9 <= 1.534
samples = 42.9%
value = [0.003, 0.997]
PC20 <= -0.366
samples = 3.5%
value = [0.368, 0.632]
PC11 <= -0.396
samples = 3.8%
value = [0.184, 0.816]
PC1 <= 2.024
samples = 49.9%
value = [0.96, 0.04]
(...)
(...)
(...)
(...)
(...)
(...)
(...)
(...)


PC5 <= -0.511
samples = 100.0%
value = [0.5, 0.5]
True
False
PC21 <= -0.67
samples = 28.7%
value = [0.259, 0.741]
PC15 <= 0.145
samples = 71.3%
value = [0.597, 0.403]
PC14 <= 0.189
samples = 1.0%
value = [0.777, 0.223]
PC2 <= 0.333
samples = 27.7%
value = [0.24, 0.76]
PC5 <= 2.292
samples = 43.4%
value = [0.666, 0.334]
PC22 <= -0.493
samples = 27.9%
value = [0.49, 0.51]
(...)
(...)
(...)
(...)
(...)
(...)
(...)
(...)

As for what our visualizations tell us: despite us utilizing anonymous data, our decision tree revealed some underlying patterns and relationships between features while showing which attributes most strongly influence outcomes. As we can see, our tree was split based on various principal components (PCs), indicating how each component contributes to classifying the data points using varying probabilities that assess each class' impact.

The tree shows a balanced root node, where samples are almost evenly split, but subsequent splits are apparent in showing distinct paths and probabilities that emphasize certain principal components as key differentiators between the classes. We see that the root node splits on **PC5** at a threshold of -0.557, indicating that this component has the highest initial influence in separating the dataset. The root node contains all the samples, with a nearly even class distribution (49.9% for one class and 50.1% for the other), and the split based on PC5 leads to two distinct paths:

1. **Left path (True for PC5 ≤ -0.557):**
    - This branch captures 27.1% of the samples, with a class distribution of roughly 25% and 75% that clearly favors one class.
    - The next split on **PC12 ≤ -0.443** narrows down samples with high confidence toward a specific class (around 86% for one class).
    - Later nodes continue to filter samples by splitting into components like **PC1**, **PC22**, and **PC9** while refining the sample subsets to achieve a high probability for each terminal node's dominant class.
2. **Right path (False for PC5 ≤ -0.557):**
    - This branch has 72.9% of the samples with a majority class distribution of about 59% to 41%.
    - The next split on **PC21 ≤ -0.47** pushes 60.7% of samples further down the right, with a more balanced class distribution (around 54% and 45%), indicating that PC21 is a generally weaker separator.
    - Further splits occur on **PC23**, **PC3**, **PC1**, and others. The most distinct of these splits occur at one node, **PC3** (at -1.225), which creates a high skew in class distribution, making one class highly dominant at 91%.

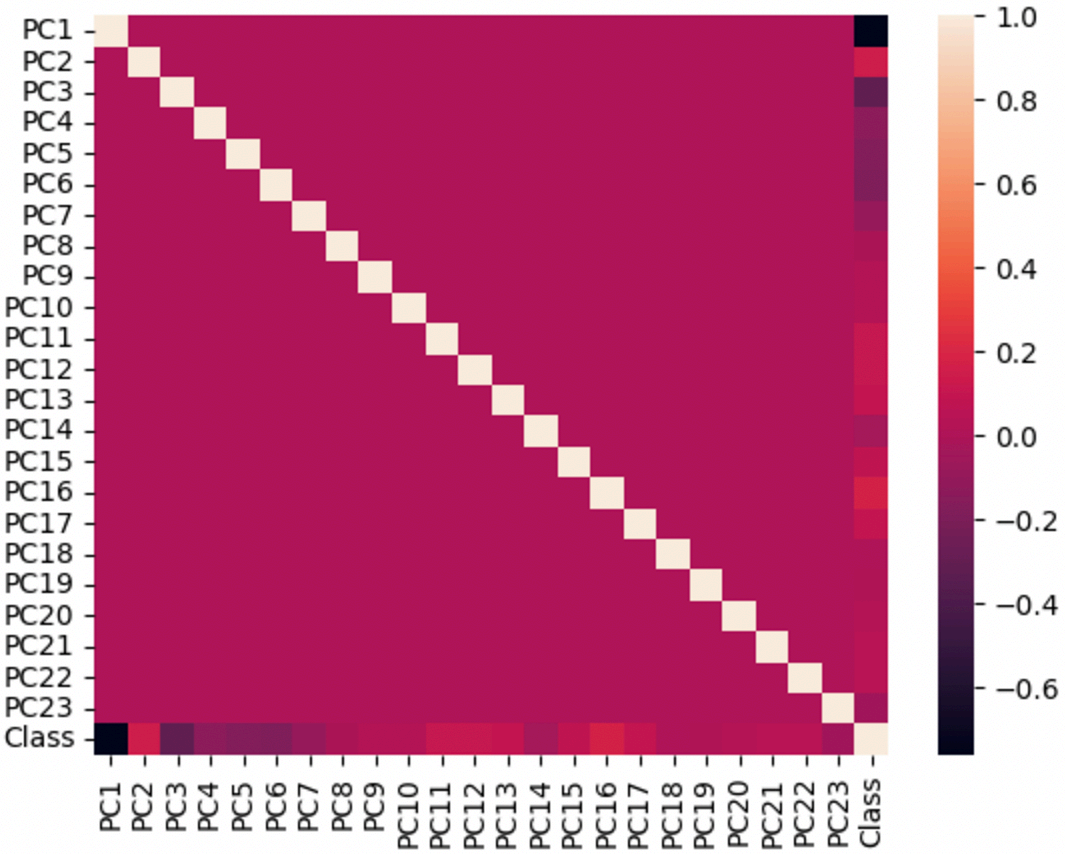
Based on this analysis, we see that **PC5**, **PC1**, **PC21**, and **PC3** appear multiple times in critical splits, indicating that they're likely the most essential features in the model's classification decisions. Even though the features are anonymous, we know that, with further analysis and the deanonymization of the data in the future, the insight that the decision tree has provided is critical to our understanding of how RF made the decisions it did.

k=2 Clustering

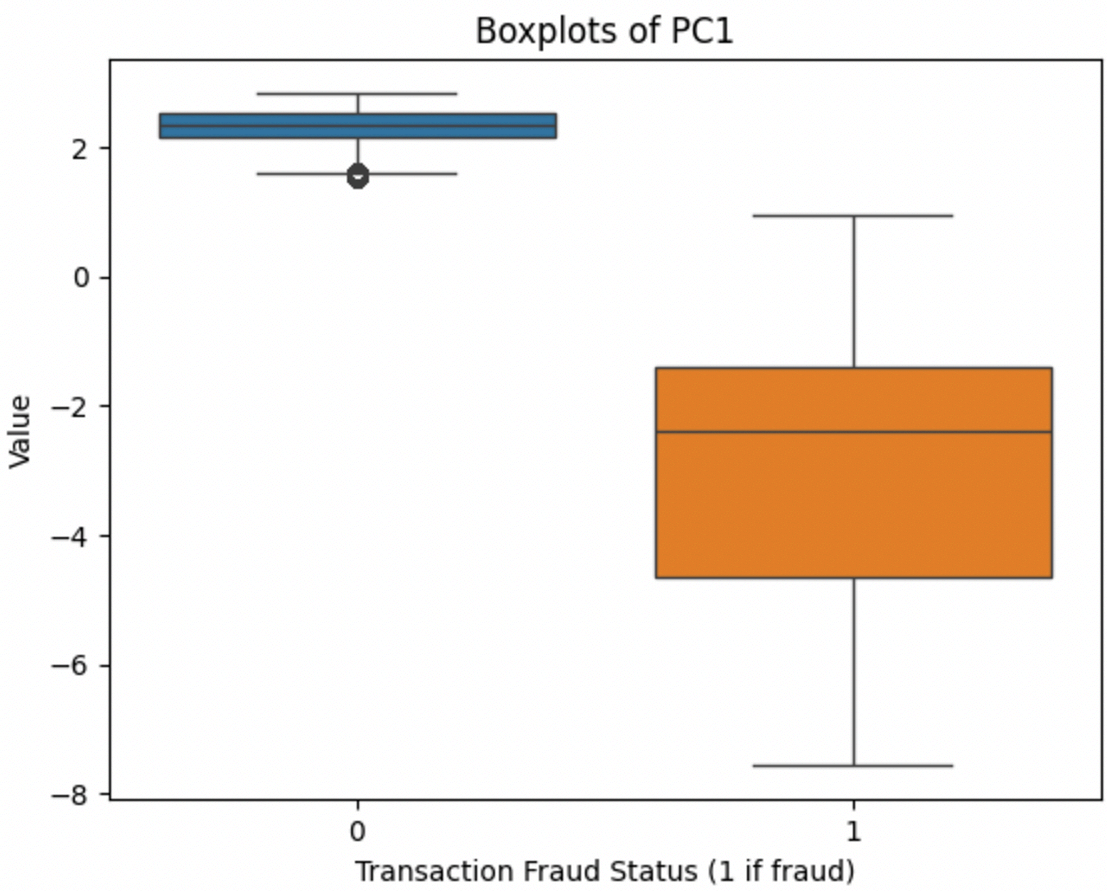
Our k=2 Clustering model yielded a clustering accuracy of approximately 89.6%, telling us that the algorithm was able to distinguish between fraudulent and legitimate transactions with reasonably high precision. We see in our visualizations that k=2 worked by separating transactions into two distinct groups (clusters) based on similarities in their anonymized features; these clusters are likely those of fraudulent and valid transactions. Despite the lack of labeled data in the clustering phase, the clustering accuracy highlights our model's robustness in identifying meaningful patterns in anonymized data, but we do acknowledge that,

given our accuracy, there is still room to improve. Further advancements, like testing alternative clustering algorithms (e.g., DBSCAN), could likely increase our accuracy overall.

We see in the heatmap generated below the representations of correlation between our dataset's features after preprocessing. This visualization illustrates the relationships among the principal components (PCs), revealing which patterns likely influenced the clustering results. The heatmap shows how the PCs are not correlated with one another, proving that they are principal components. Additionally, the last column indicates each component's correlation with the class, so PC1 is the most correlated followed by PC3, PC6, PC5, etc. (the level of correlation depends on the darkness of the color according to the legend).



Additionally, boxplots of specific principal components (in specific, PC1 through PC22) were generated to analyze how these PCs corresponded to results across our 2 clusters. The most significant example is that of PC1, with the boxplot attached below; we see that the separation between clusters is clearly pronounced (the most out of any of the components), highlighting PC5's strong influence in differentiating between fraudulent and valid transactions.



As we can see, PC1 shows a clear distinction between fraudulent and valid transactions; the valid transactions have a tightly concentrated range around +2, whereas the fraudulent transactions exhibit a wider range, with a median at about -2 and outliers going down to -8. This large difference in central tendency and variability suggests that PC1 is a critical feature in distinguishing between the two classes, and the PC's high separation suggests that it is a valuable predictor for fraud detection.

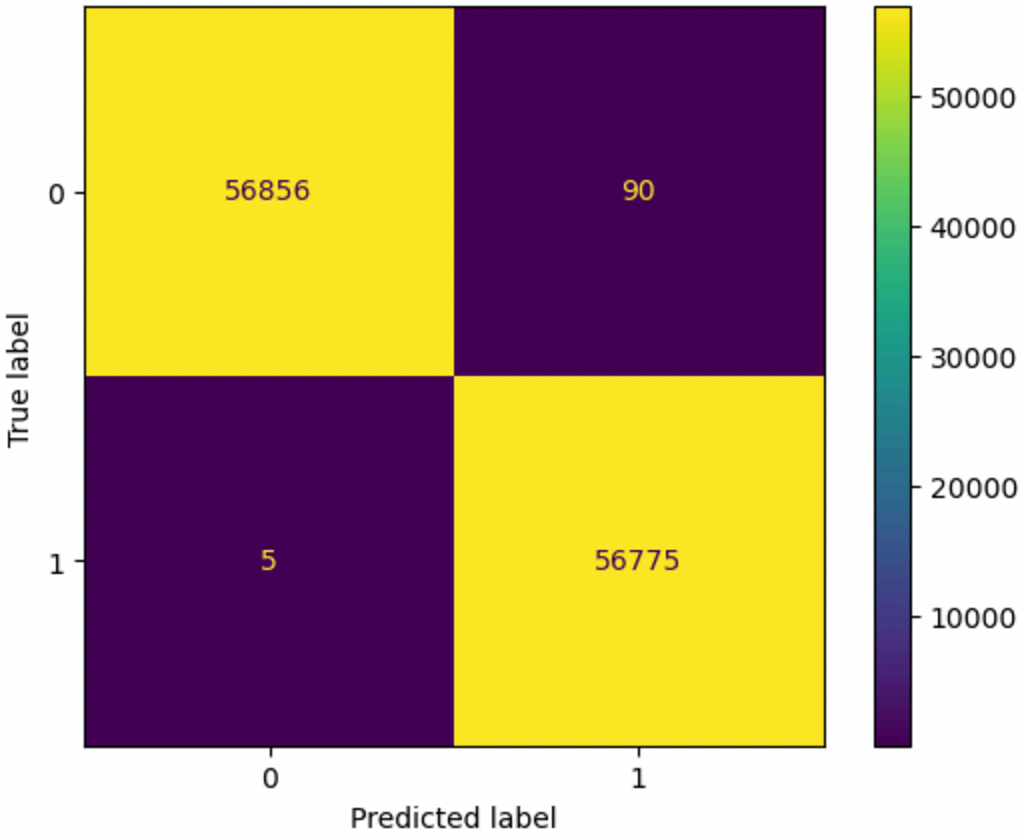
For brevity's sake, we'll note here that other critical features appear to be PC3 and PC5, with other PCs largely middling in their range. These key features, despite being anonymous, tell us what the most important variables are in distinguishing between our binary options and are important to keep in mind for further analyses.

KNN (K-Nearest Neighbors)

Our KNN model achieved an accuracy of 99.92% on the test data. The high accuracy suggests that the model performed well in correctly classifying most instances, and as we know, KNN's accuracy often depends heavily on the chosen value of the number of neighbors present and the distribution of the data at hand. We don't want to deign from noting that a very high accuracy might also indicate overfitting as occurred in our RF model, especially in complex datasets where KNN may memorize patterns rather than generalize them, which is likely what happened here. Unlike more robust models like Random Forest, KNN has a simple algorithmic structure, which might limit its ability to distinguish nuanced variations in our anonymized and large dataset.

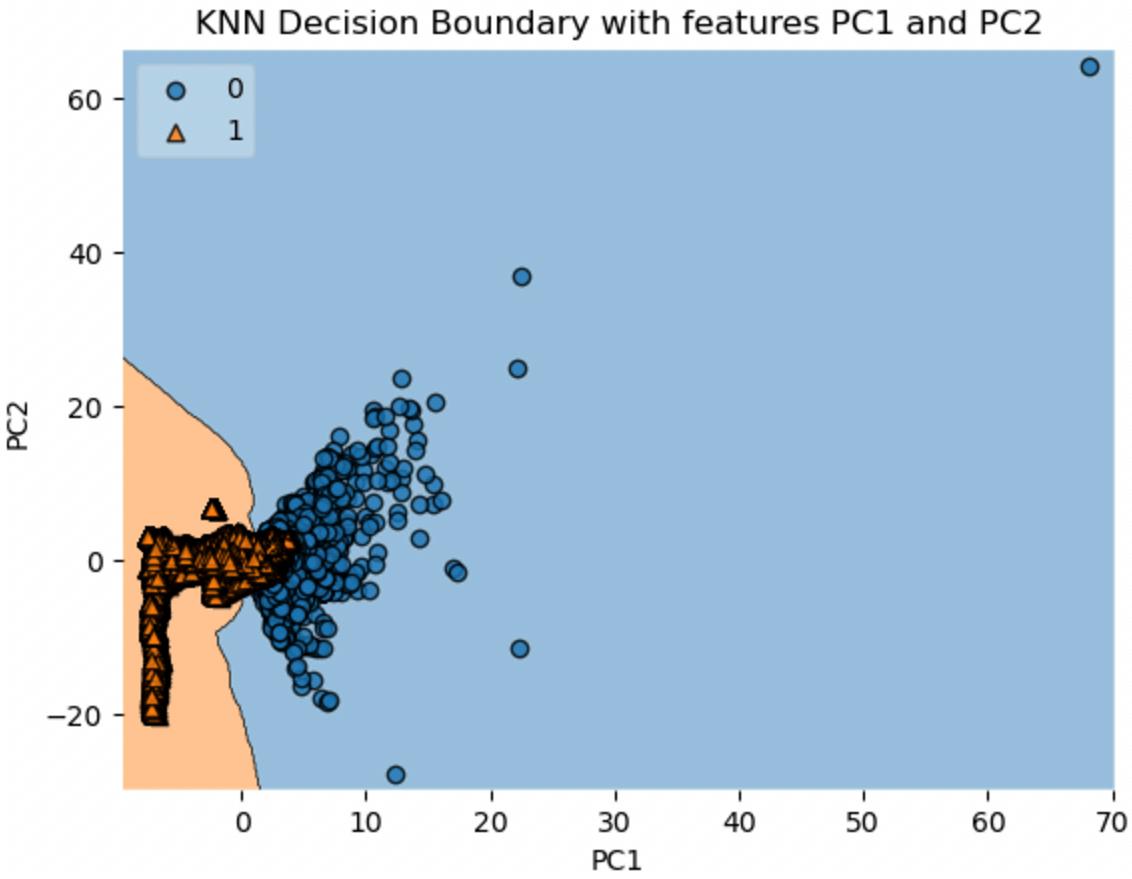
Here is our confusion matrix, which shows minimal misclassification. Most instances are clustered along the diagonal, showing that the model predicted the correct class for both majority and minority classes. We

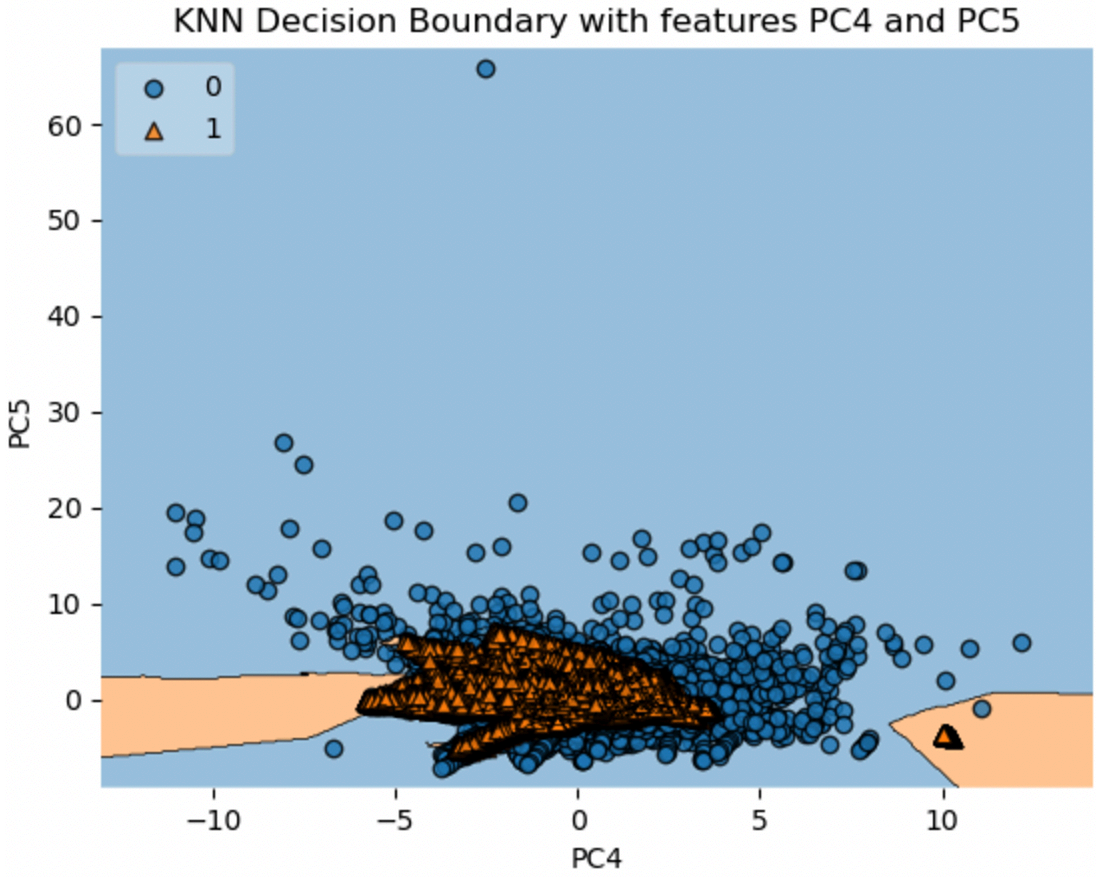
believe that indicates strong classification performance, with only a few false positives and false negatives, thereby showcasing our model's ability to accurately separate the given classes.



Additionally, we also generated decision boundary plots for selected pairs of principal components (such as PC1 vs. PC2, PC3 vs. PC4), and they reveal clear, distinct regions where the KNN model predicts each class. KNN forms regions based on Euclidean distance, and as we can see in our visualizations, decision regions adapt to the data structures present. In a lot of cases, the boundaries appear complex and jagged, which is a signature characteristic of KNN while it adjusts its regions based on the local density of data points.

Several PCs play key roles in defining the class boundaries, with sharp separations noted, especially around PC1 and PC5; thus, we've included these two plots as you can see here. Overall, the plots indicate the influence of these specific features in distinguishing between the classes, highlighting their importance in the dataset.





References

[1] B. F. Caminer, “Credit Card Fraud: The Neglected Crime,” The Journal of Criminal Law and Criminology (1973-), vol. 76, no. 3, p. 746, 1985, doi: https://doi.org/10.2307/1143521.

[2] “Visa Prevents Approximately $25 Billion in Fraud Using Artificial Intelligence,” usa.visa.com. https://usa.visa.com/about-visa/newsroom/press-releases.releaseId.16421.html

[3] “Harnessing the power of machine learning for payment fraud detection,” www.paypal.com. https://www.paypal.com/us/brc/article/payment-fraud-detection-machine-learning

Gantt Chart

Contributions Chart

Name	Proposal Contributions
Khushi	Write-Up, Final Video
Anthony	K-Nearest Implementation
Ansh	k=2 Clustering Implementation
Priyanshu	Data Pre-Processing
Niranjan	RF Implementation