# CREDIT CARD FRAUD DETECTION

MEMBERS:

NIRALEE KOTHARI

PRACHI BHOSALE

PRAGATI GUNAI

SACHIN LADE

MOUNICA CHIRRAVANU

- With the increasing trend of digital transactions, the prevalence of credit card fraud has become a serious issue.
- The financial loss that results from credit card fraud impacts both consumers and financial institutions.
- The detection of fraudulent activities using credit cards is crucial, which led us to develop a credit card fraud detection system using big data analytics.
- Our aim is to analyze and process vast amounts of transaction data using Decision Tree and K Nearest Neighbor algorithms to identify potentially fraudulent transactions.

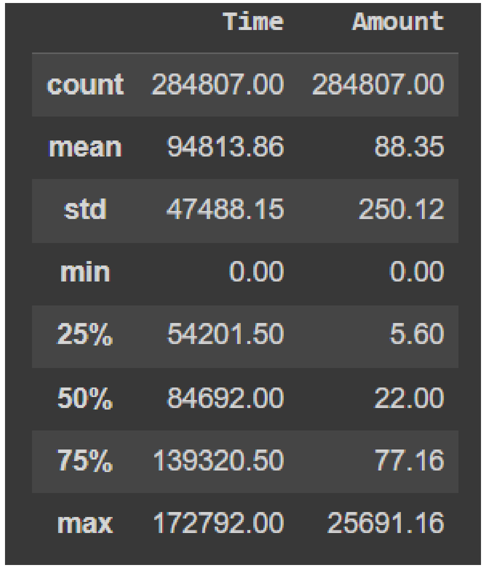
# Introduction

- The "Credit Card Fraud Detection" dataset contains a total of 284,807 transactions, of which only 492 (0.17%) are labelled as fraudulent. Each transaction in the dataset includes various features such as the transaction amount, time, and 28 anonymized variables. The dataset was generated by transforming the original features using Principal Component Analysis (PCA) due to privacy concerns. It is a highly unbalanced dataset, with the majority of transactions being non-fraudulent.
- One of the main challenges faced while working with the dataset is its highly unbalanced nature, with only 0.17% of the transactions being fraudulent. This makes it challenging to train a model that can accurately detect fraudulent transactions without being biased towards the majority class. Another challenge is that the dataset features have been anonymized using PCA, which makes it difficult to interpret the data and understand the relationship between the features and the target variable.

# Data Set Overview

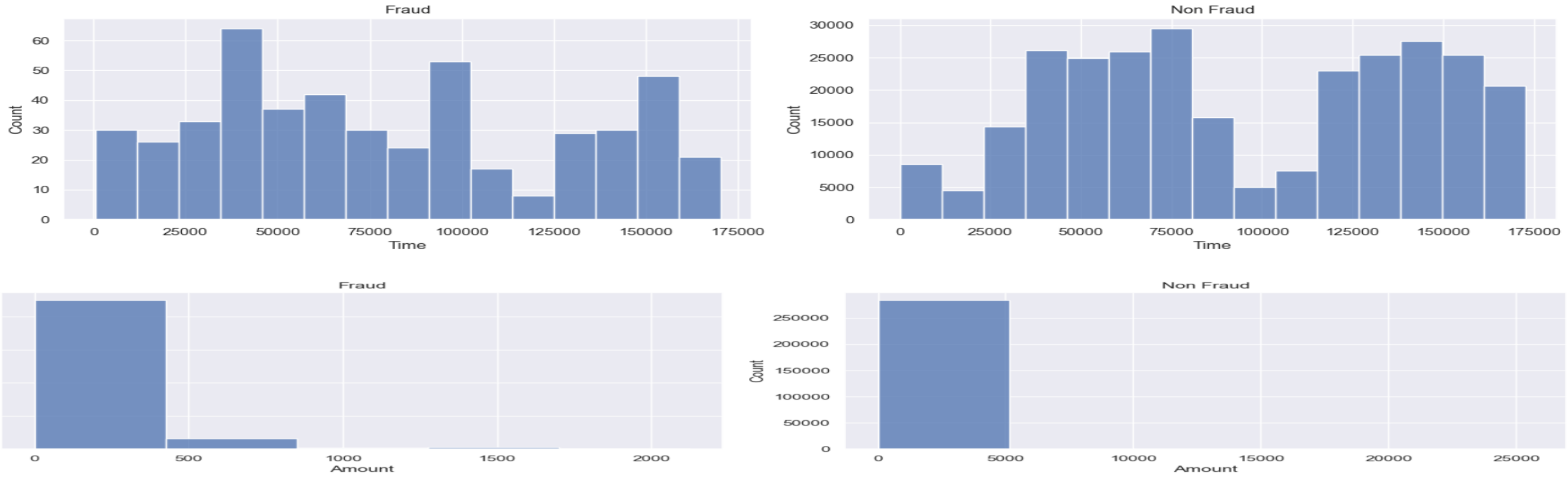
# Methodology

- Firstly we read all the data and did some analysis on that such as check if there are any missing values or null values present which resulted in value zero.
- Analysis said 'Time' and 'Amount' are the only features that have not been transformed where is time is in seconds describing the transaction time and amount is the transaction amount.



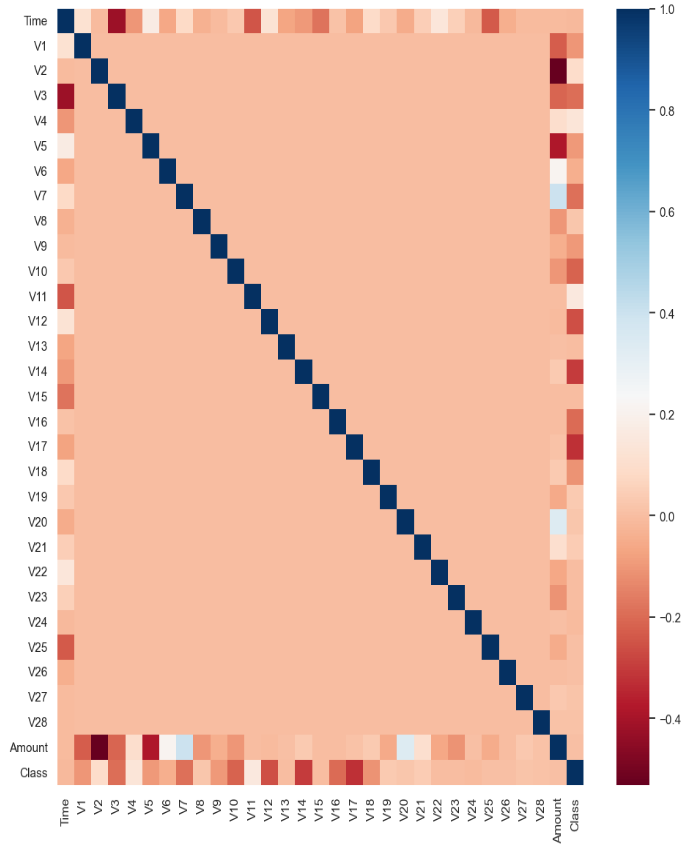
|  | Time | Amount |
|---|---|---|
| **count** | 284807.00 | 284807.00 |
| **mean** | 94813.86 | 88.35 |
| **std** | 47488.15 | 250.12 |
| **min** | 0.00 | 0.00 |
| **25%** | 54201.50 | 5.60 |
| **50%** | 84692.00 | 22.00 |
| **75%** | 139320.50 | 77.16 |
| **max** | 172792.00 | 25691.16 |

- From the analyzed data we developed an histogram for Time and Amount features considering the column that represents the frauds.



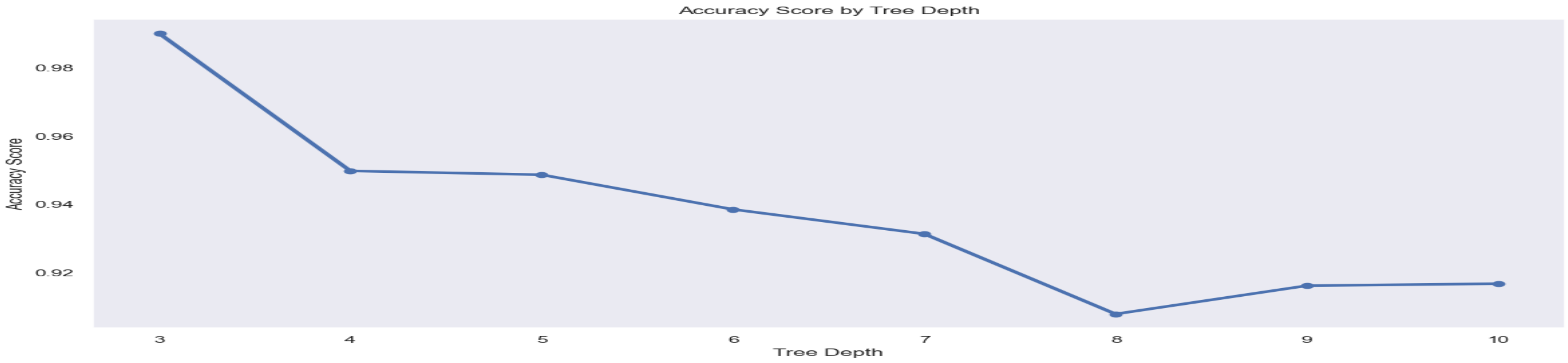
# Methodology

- Since the data is very imbalanced with a majority of transaction being non-fraudulent.
- Using this imbalanced dataset as the basis for predictive models and analysis can lead to errors and overfitting of algorithms, as they will assume that most transactions are non-fraudulent. It is important to develop a model that can accurately detect patterns indicative of fraud, despite the imbalanced nature of the dataset.
- In order to identify the features that heavily influence whether a transaction is fraudulent or not, it is crucial to use the correct Data Frame (subsample) for analysis.This helps us to determine which features have a high positive or negative correlation with fraudulent transactions.To achieve this, we utilize a correlation matrix.



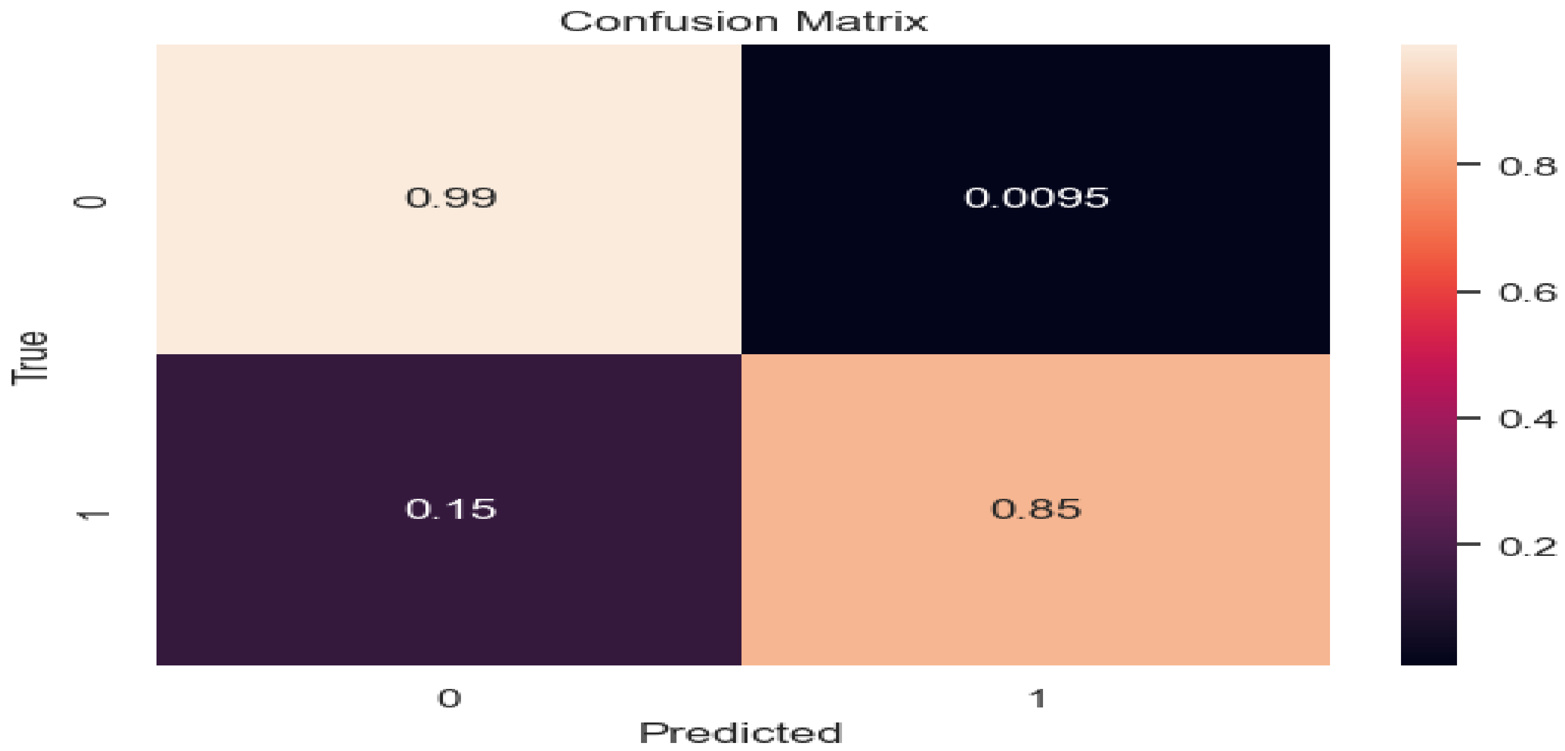
# Big Data Technologies Used

- To prevent overfitting issues, it is important to set the maximum depth of a Decision Tree. To determine the optimal depth, we generate models with varying depths and evaluate their performance.
- By selecting the model with the best results, we can identify the optimal depth and use it to create and assess the final decision tree model.
- The best accuracy was 0.9902 with depth =3.



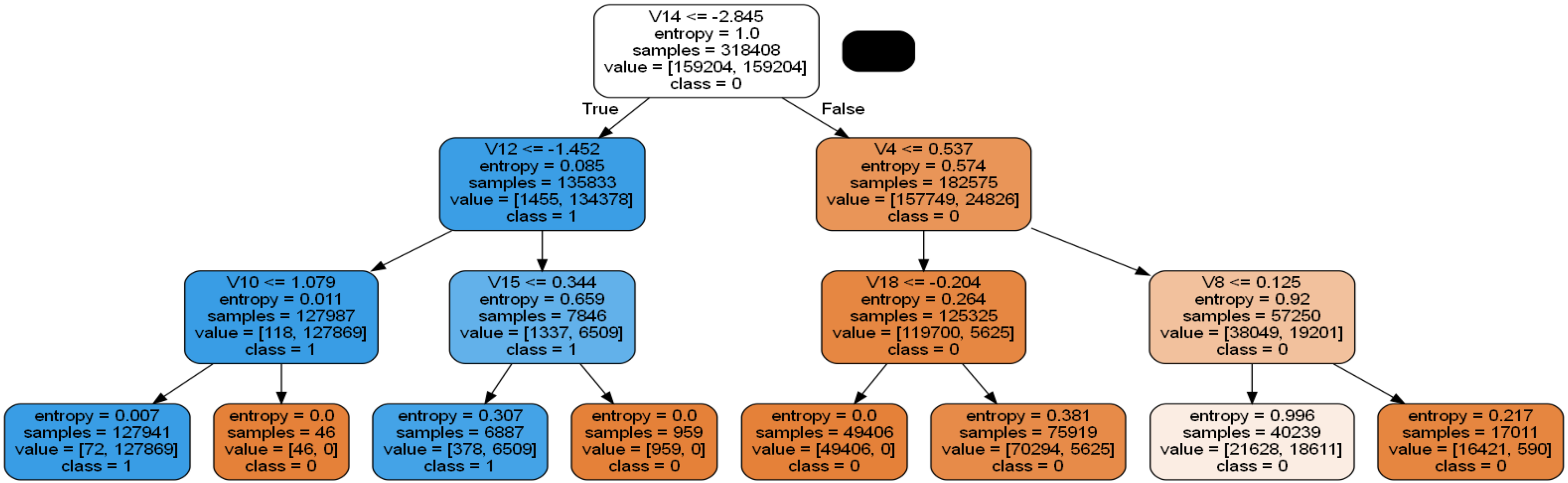
# Model Used

- Then we trained the model using underbalanced data which resulted in this graph.



# Model Used

- Visualization resulted in this graph



V14 <= -2.845
entropy = 1.0
samples = 318408
value = [159204, 159204]
class = 0

[redacted]

True

False

V12 <= -1.452
entropy = 0.085
samples = 135833
value = [1455, 134378]
class = 1

V4 <= 0.537
entropy = 0.574
samples = 182575
value = [157749, 24826]
class = 0

V10 <= 1.079
entropy = 0.011
samples = 127987
value = [118, 127869]
class = 1

V15 <= 0.344
entropy = 0.659
samples = 7846
value = [1337, 6509]
class = 1

V18 <= -0.204
entropy = 0.264
samples = 125325
value = [119700, 5625]
class = 0

V8 <= 0.125
entropy = 0.92
samples = 57250
value = [38049, 19201]
class = 0

entropy = 0.007
samples = 127941
value = [72, 127869]
class = 1

entropy = 0.0
samples = 46
value = [46, 0]
class = 0

entropy = 0.307
samples = 6887
value = [378, 6509]
class = 1

entropy = 0.0
samples = 959
value = [959, 0]
class = 0

entropy = 0.0
samples = 49406
value = [49406, 0]
class = 0

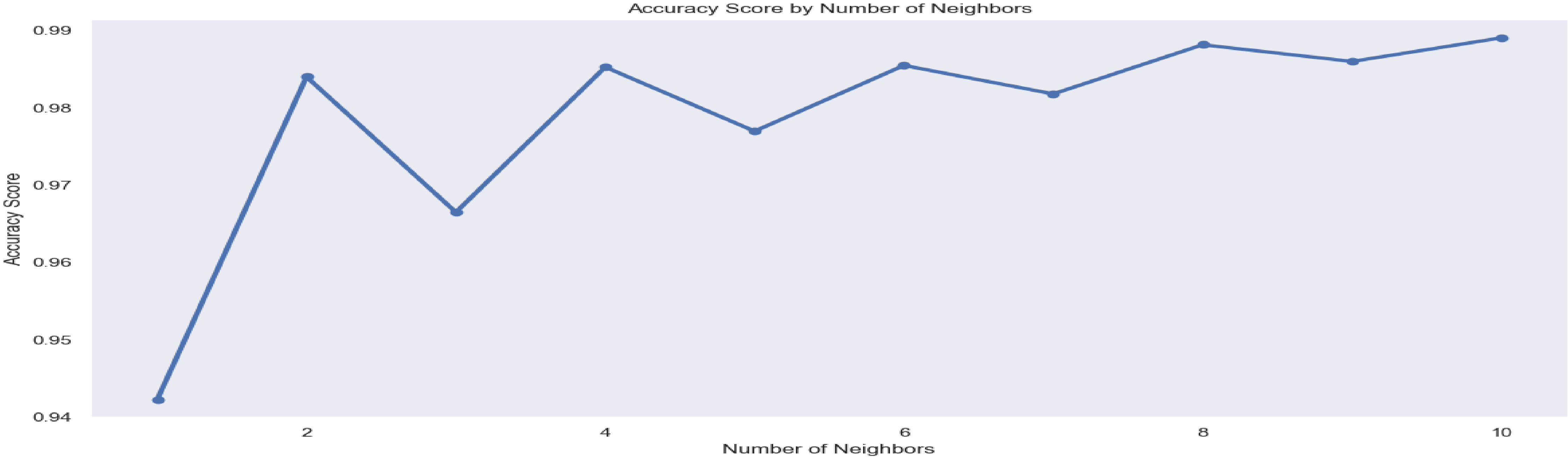
entropy = 0.381
samples = 75919
value = [70294, 5625]
class = 0

entropy = 0.996
samples = 40239
value = [21628, 18611]
class = 0

entropy = 0.217
samples = 17011
value = [16421, 590]
class = 0

# Model Used

- Then we used the K-nearest neighbor algorithm to detect the non-linear relationship between the data.
- The best accuracy was 0.9891 with k =10.



# Model Used

- The decision tree model was trained on the Credit Card Fraud Detection dataset, which consisted of a total of 284,807 transactions.
- The model was implemented using a maximum depth of 3 to prevent overfitting, which could lead to poor performance on new data.
- After evaluating the model, we achieved an accuracy of 0.9902, indicating that the model was able to correctly identify fraudulent transactions with a high degree of accuracy.
- This level of accuracy demonstrates the effectiveness of the decision tree model in detecting fraudulent transactions in credit card data.
- The KNN model was implemented with a value of k=10 and achieved an accuracy of 0.9891, which is slightly lower than the accuracy achieved by the decision tree model with a depth of 3.
- The model can be used as a reliable tool for detecting fraudulent transactions, which is crucial for preventing financial losses for both consumers and financial institutions.

# Results

- The project uses the "Credit Card Fraud Detection" dataset available on Kaggle, which consists of credit card transactions labelled as fraudulent or non-fraudulent. The project team has implemented decision tree and k-NN algorithms to train machine learning models to detect fraudulent transactions. The team plans to leverage big data technologies such as Hadoop and Spark for real-time data processing and analysis.
- **Potential Impact:** Credit card fraud has become an increasingly prevalent problem, leading to significant financial losses for both consumers and financial institutions. The project aims to address this problem by developing a credit card fraud detection system that can accurately detect fraudulent transactions in real-time. If successful, the project could potentially save financial institutions and consumers billions of dollars in losses due to fraudulent transactions.

# Conclusion

# References

- Kaggle dataset: https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud
- Scikit-learn. (2021). KNeighborsClassifier. Revised from https://scikitlearn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html
- Some concepts revised from: Big Data Made Accessible. 2020. Anil Maheshwari.

# THANK YOU

niralee@umich.edu
prachibh@umich.edu
pragatig@umich.edu
mounicac@umich.edu
sachya@umich.edu