Sukanya

Final Capstone Project

Credit Card Fraud Detection: A Machine Learning Approach

Sukanya

Credit Card Fraud

- Credit Card fraud detection in a timely fashion is crucial to prevent financial losses and maintain customers trust with banks

- Increased online transactions has resulted in rising credit card fraudulent transactions

Sukanya

Credit Card Fraud Statistics

Key Credit Card Fraud Statistics

- **Total value of credit card fraud**: $246 million (2023)
- **Annual global fraud losses (credit & debit card)**: $34.36 billion (2022)
- **Total volume of credit card and debit card fraud losses**: 6.81 cents per $100 (2020) & 6.78 cents per $100 (2019)
- **U.S. share of global payment card fraud**: 38.83% (payment card fraud losses) and 22.40% (transaction value)
- **Projection for total losses due to payment card fraud from 2021-2023**: $408.50 billion

Source: https://wallethub.com/edu/cc/credit-card-fraud-statistics/25725

Dataset Source

Credit Card Fraud Detection Dataset 2023

url: https://www.kaggle.com/datasets/nelgiriyewithana/credit-card-fraud-detection-dataset-2023/data

Why Credit Card Fraud Detection

"In a world where fraud is rampant, trust is the most valuable currency."
- Robert T. Kiyosaki

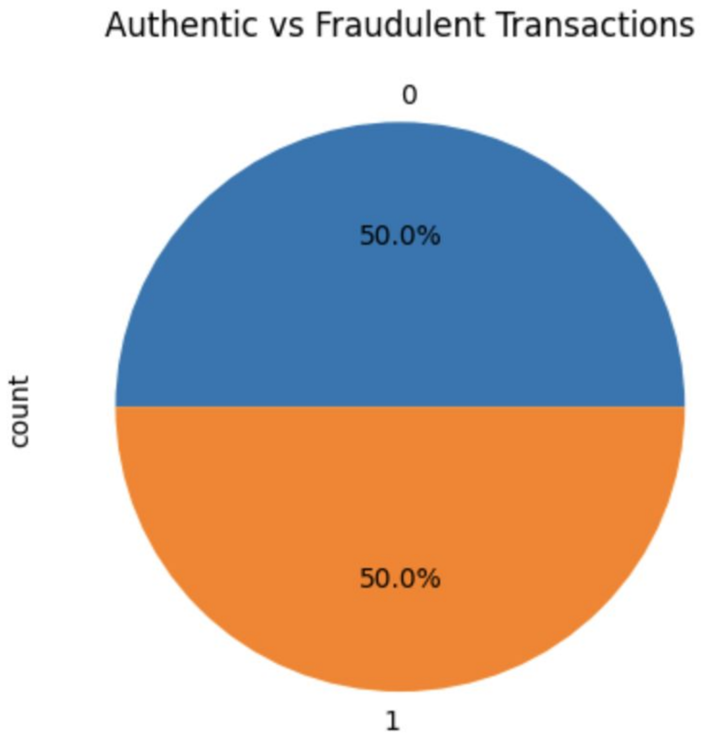
Sukanya

Exploration & Analysis

Sukanya

Fraudulent Transaction Proportion

- The fraudulent transaction % from the given set of data is 50%
- The dataset has 284,315 normal and 284,315 fraudulent transactions
- This in based on the 568,630 records



Sukanya

Data Exploration

- The dataset has 11 columns and 568630 clean records
- Target Value is Class, which has 1 and 0 Values. It is an integer type variable.
- 1 indicates Fraud while 0 indicates normal transaction
- There are no duplicate values
- There are no null values

```
data.shape
```

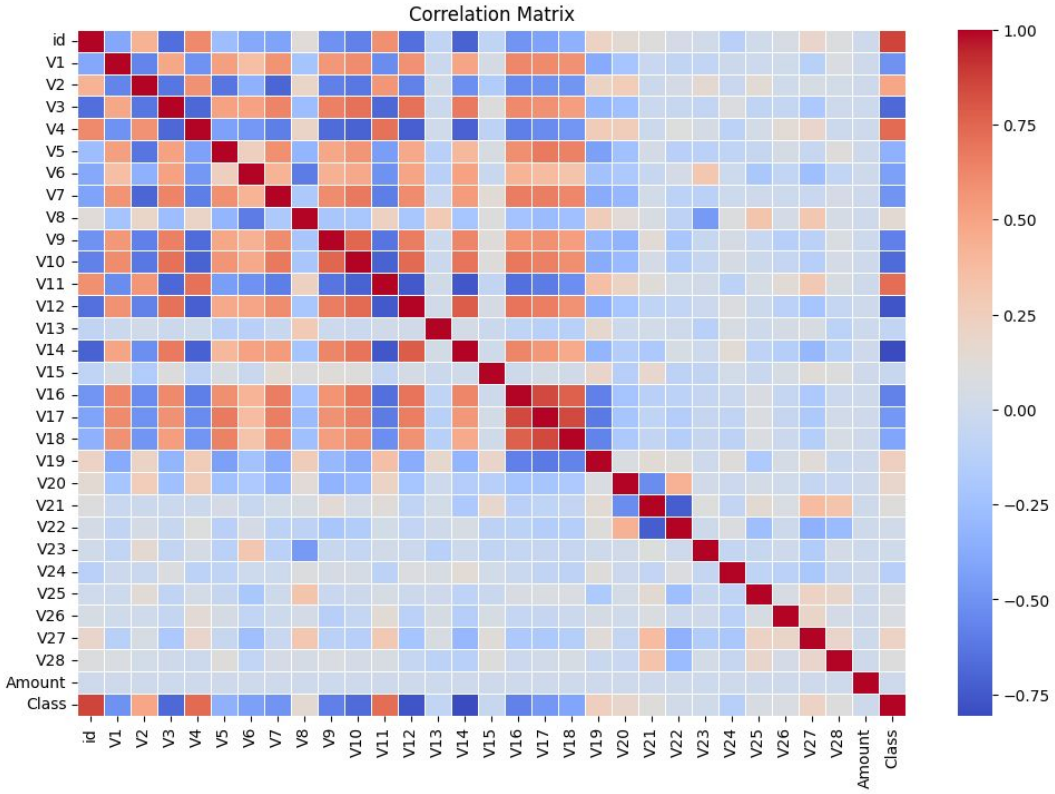
```
(568630, 31)
```

```
#Identifying duplicate records
print(df.duplicated().sum())
```

```
0
```

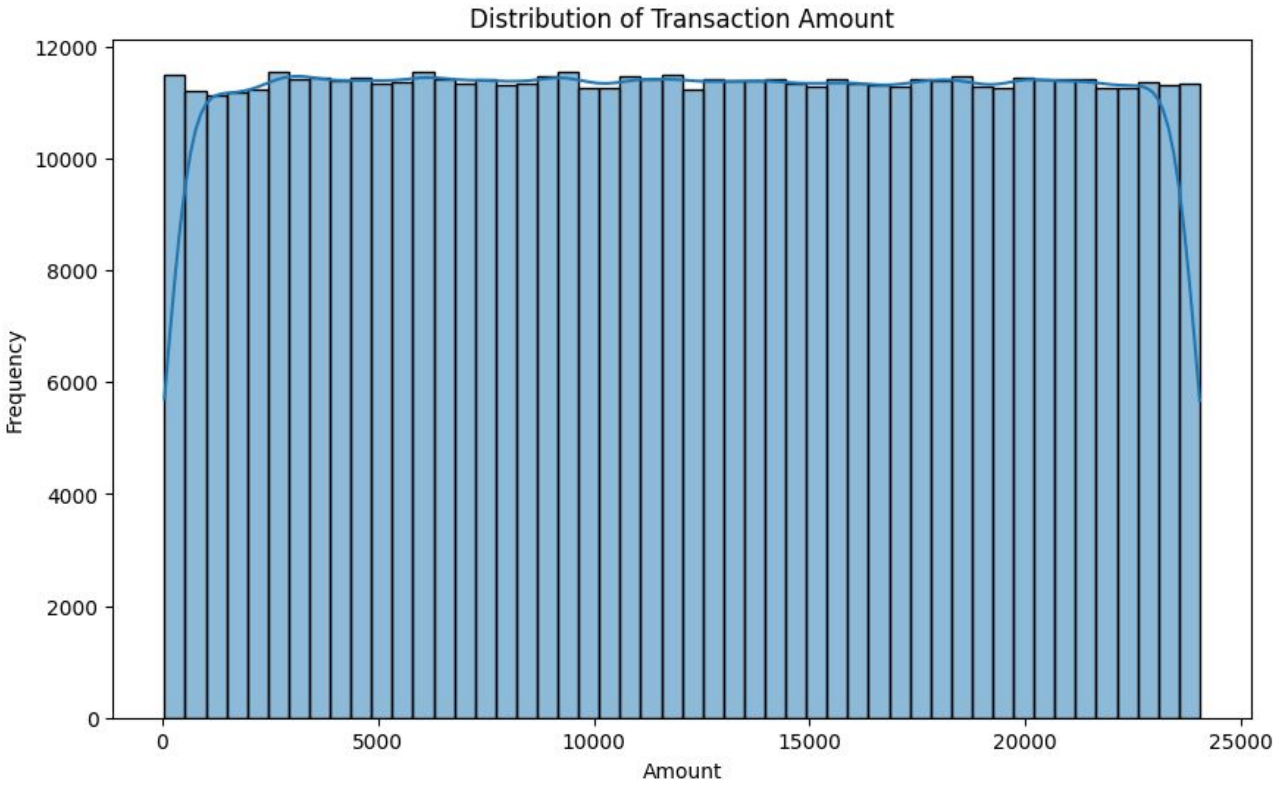
Sukanya

Relationship between Attributes

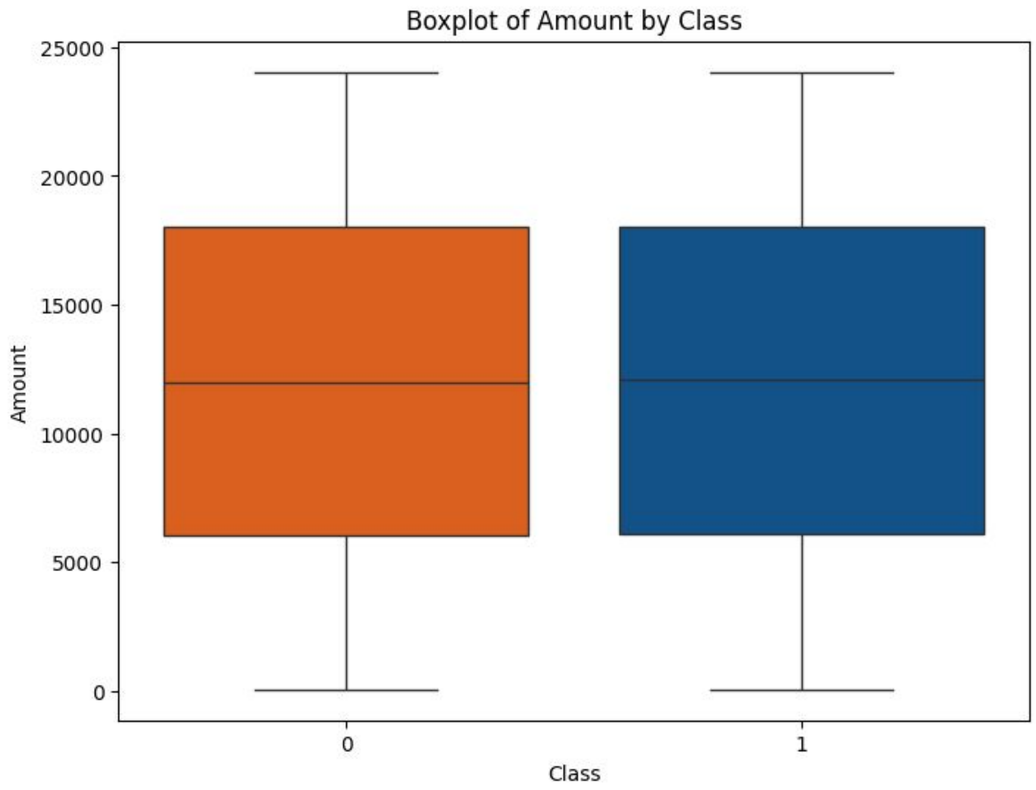


- The attributes V1-V28 are features representing anonymized transaction attributes
- They are values like time, location and PII information derived from PCA transformation
- Masked for user privacy

Distribution of transaction amount



Class with respect to amount



- Identical class separation between the amounts
- Well balanced dataset

Sukanya

Supervised Machine Learning

Sukanya

Targets and Features

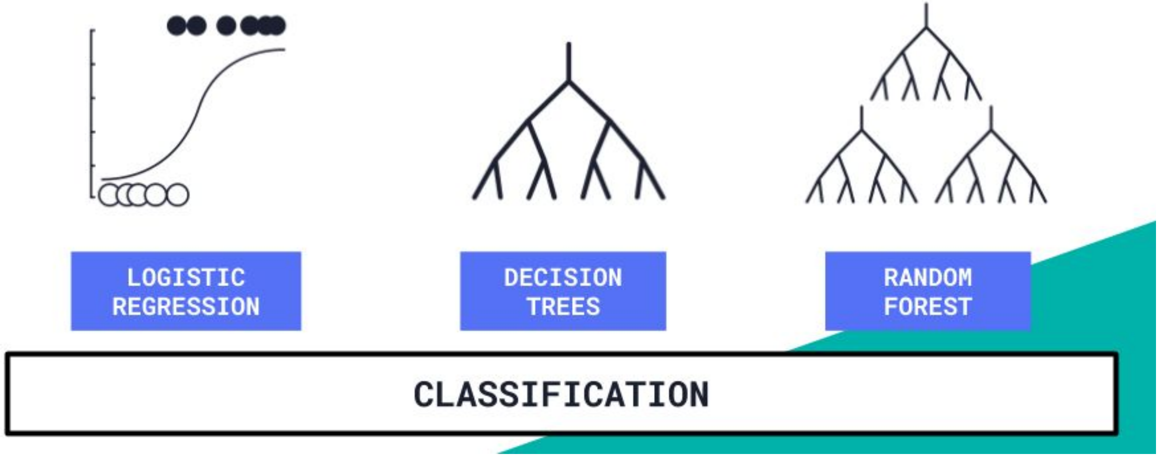
```
# Split features and target variable
X = data.drop(columns=["Class"])  # Features variable
y = data["Class"]  # Target variable
```

```
# Split data into train and test sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
```

No data processing and data wrangling needed as there are no nulls.

Supervised Machine Learning

- Decision Tree
- Random Forest
- Logistic Regression



Supervised Machine Learning Outputs

Sukanya

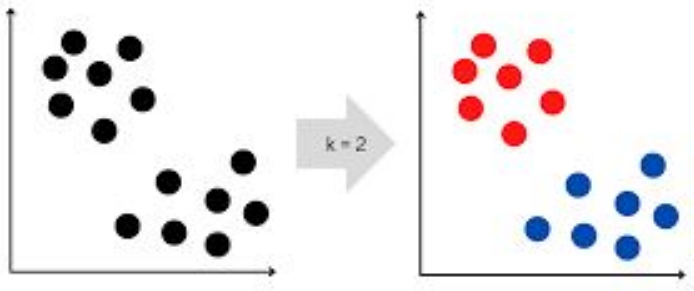
Model	Accuracy	Precision	Recall	F1 Score
Decision Tree	0.8508	0.9000	0.8991	0.8662
Random Forest	0.7528	0.8348	0.8329	0.7787
Logistic Regression	0.8343	0.8167	0.8801	0.8358

Unsupervised Machine Learning

K-Means

- Data Standardization
- Applied K-Means Clustering

```
K-Means Accuracy: 0.1037827761461759
```



K-Means Clustering Algorithm

Sukanya

Deep Learning

Sukanya

Configurations

- Simple deep learning model-Sequential model

- Sequential model is a linear stacks of layers, where each layer has one input and output tensor

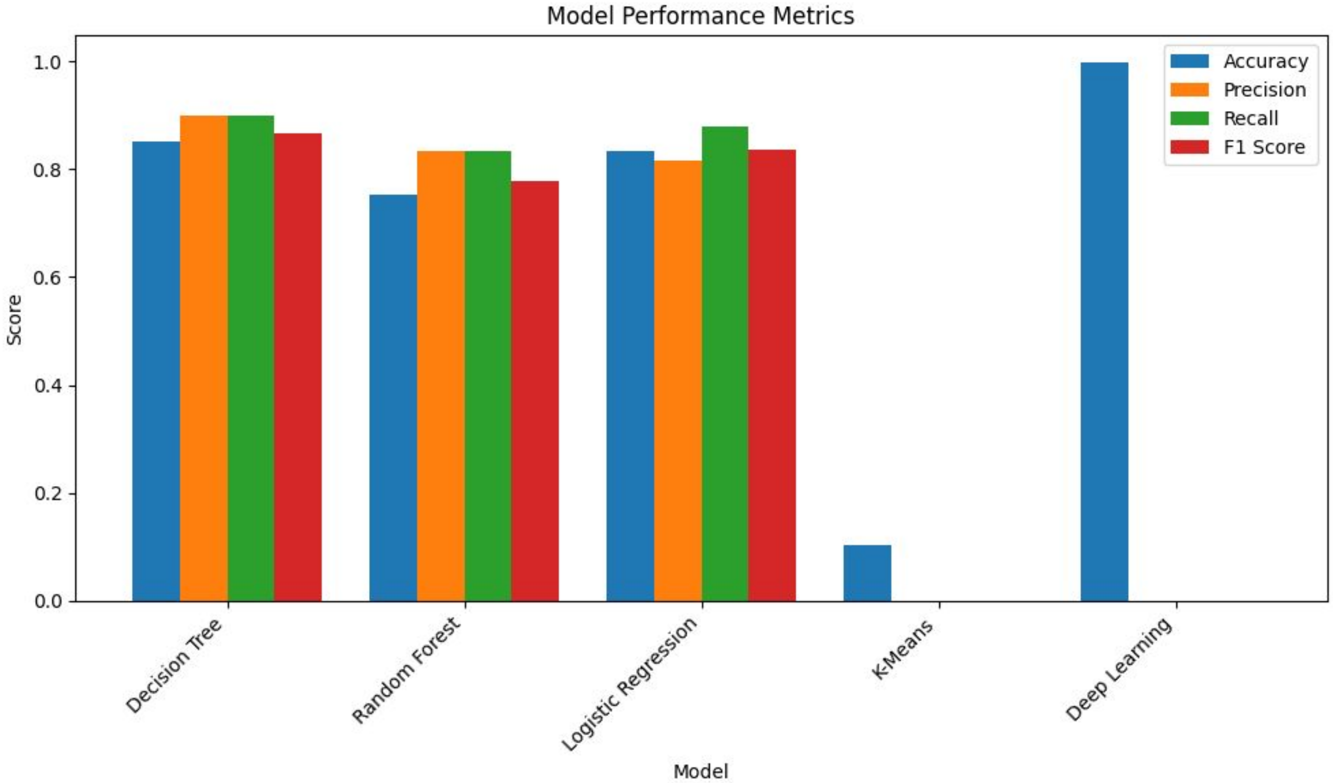
Configurations

```python
# Define a simple deep learning model
model = Sequential()
model.add(Dense(64, input_dim=X_train.shape[1], activation='relu'))
model.add(Dense(32, activation='relu'))
model.add(Dense(1, activation='sigmoid'))
```

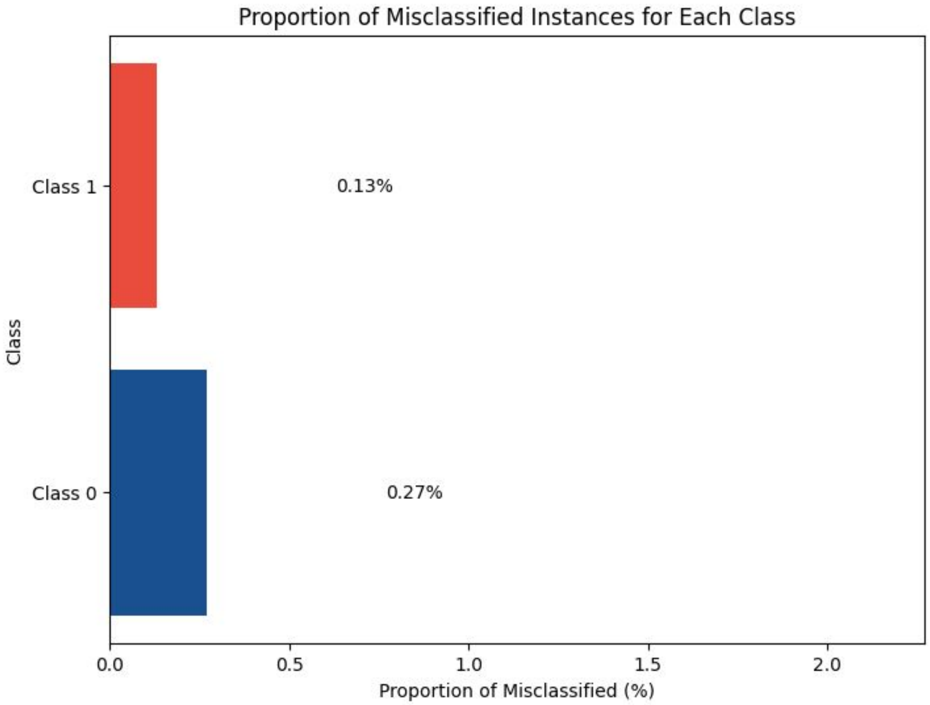
```python
# Compile the model
model.compile(loss='binary_crossentropy', optimizer='adam', metrics=['accuracy'])

# Train the model
model.fit(X_train, y_train, epochs=20, batch_size=32, validation_split=0.1, verbose=1)

# Evaluate the model on the test set
loss, accuracy = model.evaluate(X_test, y_test, verbose=0)
print("Test Accuracy:", accuracy)
```

Best Model Outcome

Model Performance Comparison



Simple Deep Learning - Outcome Analysis




```
Number of Misclassified Instances: 231
```

```
Deep Learning Mean Accuracy: 0.9979687929153442
```

Sukanya

Challenges

- Feature Engineering: Identifying relevant features to improve model performance was a challenge as the feature names anonymized and difficult to interpret
- Computational Resources: Models required high GPU and faced issues with time out.
- Reduced cross validation factor from 5 to 3 to circumvent this issue. Hence hyperparameter optimization was a challenge.
- Dataset comprised of only 2023 data
    - Lack of historical context
    - Potential overfitting of 2023 trends
    - Limited diversity of fraud cases and inadequate training data

Sukanya

Recommendations

- Ensemble methods such as bagging, boosting, or stacking can be utilized to aggregate the predictions of individual models, leveraging their diverse strengths and mitigating the weaknesses inherent in each model.
- Use a diverse dataset that spans over several years to get the historical context
- Continuous monitoring leveraging feedback from real-world performance for iteratively improving performance

Sukanya

Conclusion

- The detection of fraud greatly helps banks and customers to take effective timely fraud prevention measures.

- Analyzing the dataset with transaction features, the project was aimed at building models to accurately predict fraudulent transactions

- Leveraging insights from the credit card transaction dataset, banks can develop and implement fraud reduction strategies thereby mitigating fraudulent transactions.

Sukanya

Thank You