

# CREDIT CARD FRAUD DETECTION



Building a credit card fraud detection system involves using machine learning algorithms to identify patterns indicative of fraudulent activity. Below is a simplified example using Python and the scikit-learn library. Note that a real-world implementation would require a more sophisticated approach and a larger dataset.

python

[Copy code](#)

```
# Import necessary libraries
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import accuracy_score, confusion_matrix

# Load your dataset (replace 'your_dataset.csv' with your actual file)
data = pd.read_csv('your_dataset.csv')

# Explore the dataset
print(data.head())

# Split the data into features (X) and target variable (y)
X = data.drop('Class', axis=1)
y = data['Class']

# Split the dataset into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,
random_state=42)
```

```
# Initialize the RandomForestClassifier
model = RandomForestClassifier(n_estimators=100, random_state=42)

# Train the model
model.fit(X_train, y_train)

# Make predictions on the test set
y_pred = model.predict(X_test)

# Evaluate the model
accuracy = accuracy_score(y_test, y_pred)
conf_matrix = confusion_matrix(y_test, y_pred)

print(f'Accuracy: {accuracy}')
print('Confusion Matrix:')
print(conf_matrix)
```

This is a basic example using a RandomForestClassifier. In practice, you might want to explore other algorithms, perform hyperparameter tuning, and deal with imbalanced datasets.

Additionally, feature scaling and engineering are crucial steps in building an effective fraud detection model.



Note:

Ensure that your dataset is appropriately preprocessed, and features are anonymized or encoded to protect sensitive information. Also, the dataset used here is assumed to have a 'Class' column where 1 represents fraud and 0 represents non-fraudulent transactions.

## Part 2

could involve further enhancements, such as handling imbalanced data, feature engineering, and deploying the model for real-time monitoring. Let me know if you'd like more details on any specific aspect

The dataset used for this project was the Credit Card Fraud Detection dataset, available on Kaggle, and it contains credit card transactions that were made during the month of September, 2013 by European clients during two days. It has 284,807 transactions and 30 variables.

The variable time contains the seconds elapsed between each transaction and the first transaction in the dataset. Amount contains the value of each transaction and lastly, we have class, which is a binary feature that tells us if that certain transaction was genuine or a fraud.

We also have other features V1, V2, ... V28 that are numerical inputs result of a PCA transformation whose content couldn't be displayed due to their confidential nature.

During exploration analysis, it was possible to see that the dataset contains only numerical inputs and no null values were present. I proceeded to an analysis on the amount feature with the describe( ) method.