In [ ]:

```
# Import necessary libraries
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import StandardScaler
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import classification_report, confusion_matrix,
from sklearn.model_selection import GridSearchCV

# Step 1: Load and explore the dataset
data = pd.read_csv('Fraud.csv')

# Display the first few rows of the dataset
print(data.head())

# Check for missing values
print(data.isnull().sum())



# Check the distribution of the target variable
sns.countplot(data['fraudulent_column'])
plt.title('Distribution of Target Variable')
plt.show()

# Step 2: Data preprocessing
# In this step, you should handle missing values, encode categorical
# For example, if you have missing values:
# data = data.dropna()  # Drop rows with missing values
# If you have categorical variables, you can use one-hot encoding or

# Step 3: Feature Engineering
# Create or extract relevant features based on the dataset.
# For example, you could create a feature for the time difference be

# Create a DataFrame to visualize feature importances
feature_importance_df = pd.DataFrame({'Feature': feature_names, 'Imp
feature_importance_df = feature_importance_df.sort_values(by='Import

# Plot feature importances
plt.figure(figsize=(10, 6))
sns.barplot(x='Importance', y='Feature', data=feature_importance_df)
plt.title('Feature Importances')
plt.show()
```

```
# Step 6: Actionable Plan
# Develop an actionable plan based on model insights.
# For example, you could create alert thresholds based on feature im

# Step 7: Fine-Tuning (Optional)
# You can perform hyperparameter tuning using GridSearchCV or other
# For example, you can tune the number of trees in the Random Forest

param_grid = {
    'n_estimators': [100, 200, 300],
    'max_depth': [10, 20, 30],
    'min_samples_split': [2, 5, 10]
}

grid_search = GridSearchCV(estimator=model, param_grid=param_grid,
                           scoring='accuracy', cv=3, n_jobs=-1)
grid_search.fit(X_train, y_train)

best_model = grid_search.best_estimator_

# Evaluate the model on the validation set
y_pred = model.predict(X_val)

# Evaluate the model using relevant metrics
accuracy = accuracy_score(y_val, y_pred)
precision = precision_score(y_val, y_pred)
recall = recall_score(y_val, y_pred)
f1 = f1_score(y_val, y_pred)
roc_auc = roc_auc_score(y_val, model.predict_proba(X_val)[:, 1])

print("Accuracy:", accuracy)
print("Precision:", precision)
print("Recall:", recall)
print("F1 Score:", f1)
print("ROC AUC Score:", roc_auc)

# Print the classification report
print(classification_report(y_val, y_pred))

# Plot the confusion matrix
conf_matrix = confusion_matrix(y_val, y_pred)
sns.heatmap(conf_matrix, annot=True, fmt="d")
plt.xlabel('Predicted')
plt.ylabel('True')
```

```
plt.title('Confusion Matrix')
plt.show()
```

In [ ]:

```
Addressing the proactive detection of fraud involves multiple steps



There were no missing values .

The fraud detection model is a binary classification model designed

Algorithm: Random Forest Classifier.

Variable selection involves choosing the most relevant features for

Domain Knowledge:** Consult with domain experts to identify importan
Feature Importance:** Use techniques like feature importance from tr
Correlation Analysis:** Examine correlations between features and th



To demonstrate the performance of the model, use Python with librari




To prevent fraud, the company can consider these actions during infr

- Implement real-time transaction monitoring and anomaly detection.
- Use machine learning models for continuous fraud detection.
- Strengthen authentication and authorization mechanisms.
- Enhance customer education about fraud prevention.
- Collaborate with cybersecurity experts to identify and mitigate th
- Regularly update and patch security vulnerabilities.



To determine the effectiveness of prevention measures:

- Continuously monitor fraud detection rates and false positive rate
- Conduct A/B testing to compare the effectiveness of new prevention
- Analyze the frequency and severity of fraud incidents over time.
- Review feedback from customers and employees regarding the effecti
- Adjust and refine prevention measures based on ongoing analysis an
```