Assignment Lab 2: Credit card fraud detection using Decision Tree Classifier

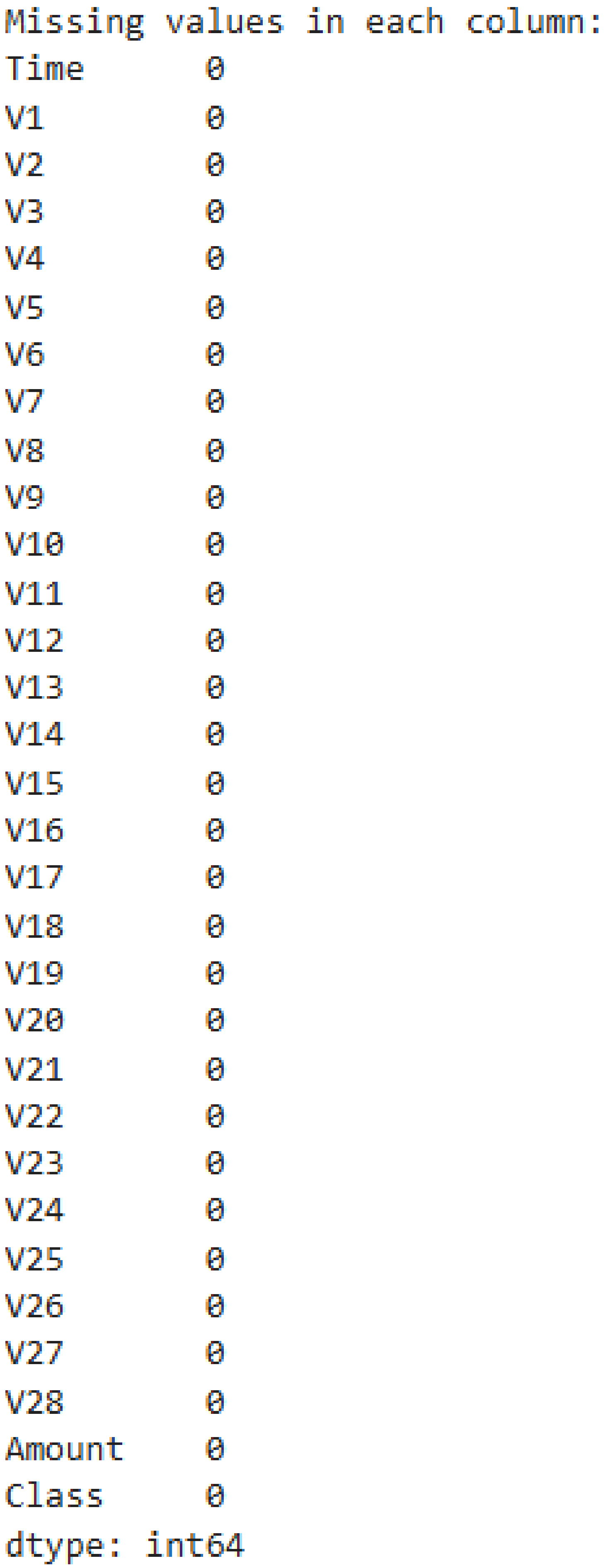
Karthikeyan G
Roll No: CB.SC.P2.CYS24008

January 2, 2025

1 Detection using Decision Tree

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.tree import DecisionTreeClassifier, plot_tree
from sklearn.metrics import confusion_matrix, ConfusionMatrixDisplay,
    classification_report
from sklearn.preprocessing import StandardScaler
from sklearn.feature_selection import VarianceThreshold

# Load the dataset
data = pd.read_csv(’/kaggle/input/creditcardfraud/creditcard.csv’)

# Check for missing values
print("Missing values in each column:")
print(data.isnull().sum())

# Separate features and target
X = data.drop(columns=[’Class’])
y = data[’Class’]

#using VarianceThreshold to remove low-variance features
selector = VarianceThreshold(threshold=0.1)
X_selected = selector.fit_transform(X)

#To Scale the features
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)

# Split the data into Training and Test sets
X_train, X_test, y_train, y_test = train_test_split(X_scaled, y, test_size=0.2,
    random_state=42)

# Training the Decision Tree Classifier
credit = DecisionTreeClassifier(random_state=42)
credit.fit(X_train, y_train)

# evaluating the model
y_pred = credit.predict(X_test)
cm = confusion_matrix(y_test, y_pred)
disp = ConfusionMatrixDisplay(confusion_matrix=cm, display_labels=clf.classes_)
disp.plot(cmap=plt.cm.Reds)
plt.title("Confusion Matrix")
plt.show()

# Classification Report
print("Classification Report:")
print(classification_report(y_test, y_pred))

# Feature & Importance
importances = credit.feature_importances_
feature_names = data.drop(columns=[’Class’]).columns
plt.figure(figsize=(10, 6))
plt.barh(feature_names, importances, color=’Orange’)
plt.xlabel("Importance")
plt.ylabel("Feature")
plt.title("Feature Importance")
plt.show()

# visualizing based on the the Decision Tree
plt.figure(figsize=(15, 10))
```

```
plot_tree(credit, feature_names=feature_names, class_names=["Non-Fraud", "Fraud"
    ], filled=True, fontsize=12)
plt.title("Decision Tree Visualization")
plt.show()
```



```
Missing values in each column:
Time      0
V1        0
V2        0
V3        0
V4        0
V5        0
V6        0
V7        0
V8        0
V9        0
V10       0
V11       0
V12       0
V13       0
V14       0
V15       0
V16       0
V17       0
V18       0
V19       0
V20       0
V21       0
V22       0
V23       0
V24       0
V25       0
V26       0
V27       0
V28       0
Amount    0
Class     0
dtype: int64
```

Figure 1: Missing values in each column

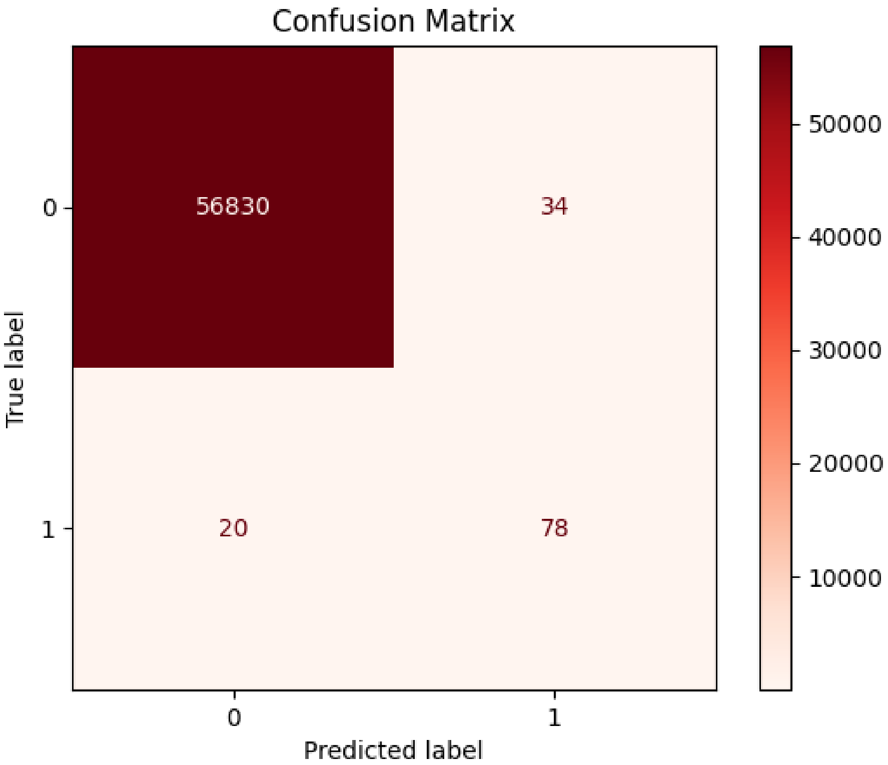
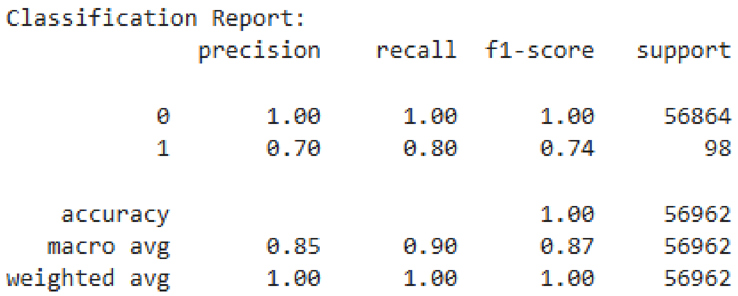


Figure 2: evaluating the model



```
Classification Report:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00     56864
           1       0.70      0.80      0.74        98

    accuracy                           1.00     56962
   macro avg       0.85      0.90      0.87     56962
weighted avg       1.00      1.00      1.00     56962
```

Figure 3: Classification Report

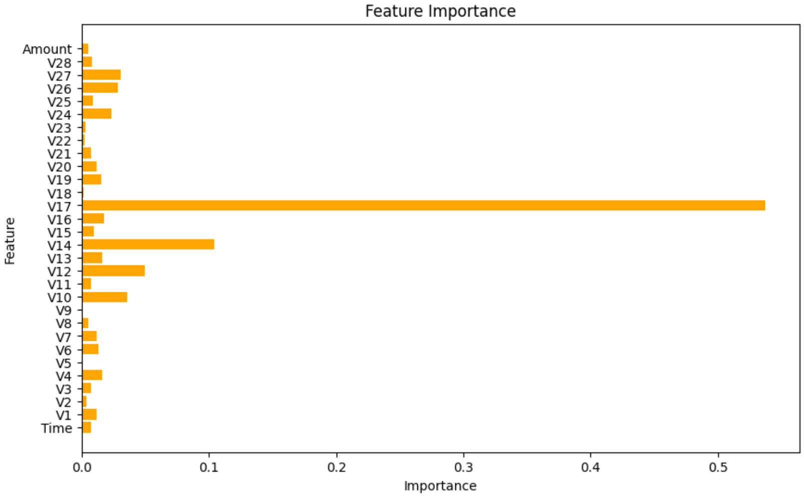


Figure 4: Featureimportance

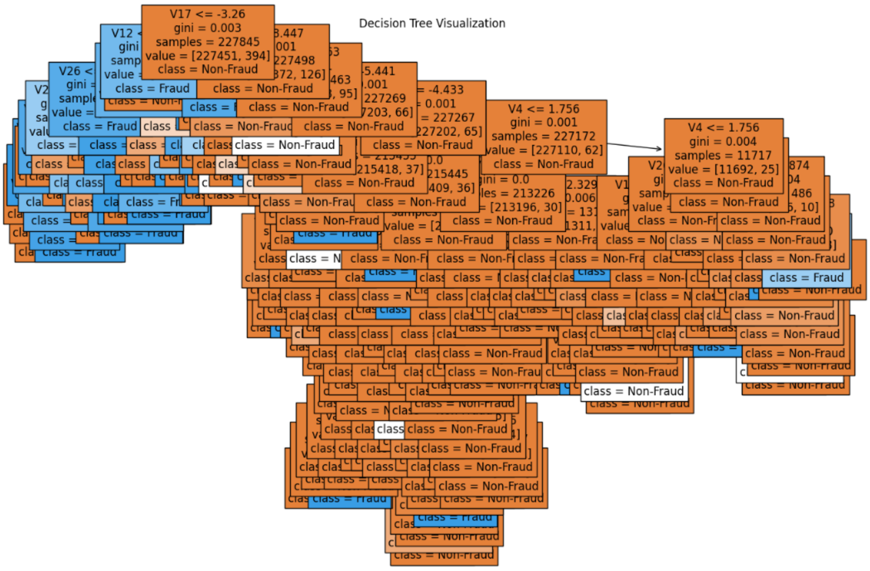


Figure 5: visualizing based on the the Decision Tree