Build a machine learning model to identify fraudulent credit card

transactions. Preprocess and normalize the transaction data, handle class imbalance issues, and split the dataset into training and testing sets. Train a classification algorithm, such as logistic regression or random forests, to classify transactions as fraudulent or genuine. Evaluate the model'

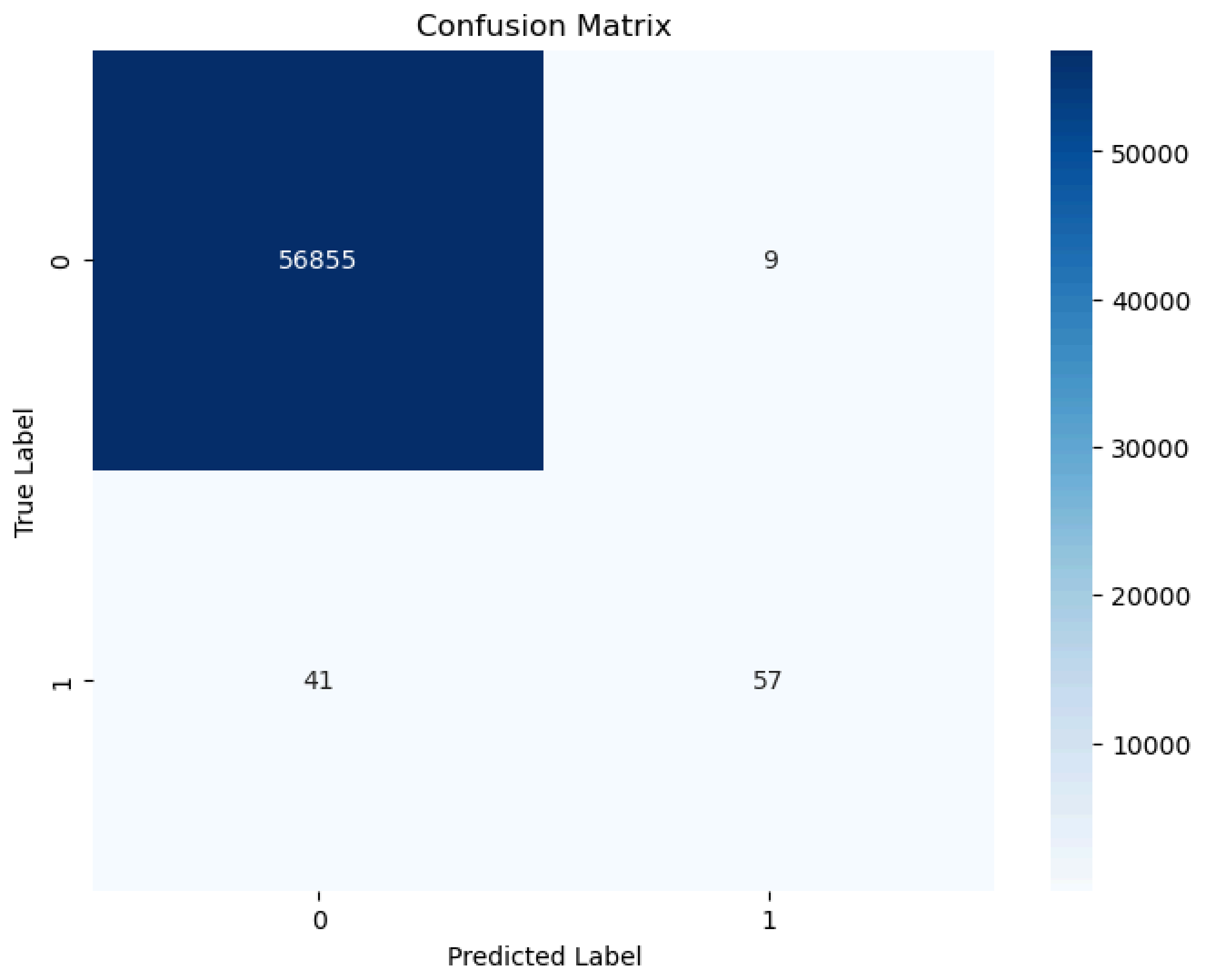
s performance using metrics like precision, recall,

and F1-score, and consider techniques like oversampling or undersampling for improving results.

In [9]:

```python
# Importing 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.linear_model import LogisticRegression
from sklearn.metrics import classification_report, confusion_matrix

# Load the dataset
df = pd.read_csv('creditcard[1].csv')

# Split dataset into features and labels
X = df.drop('Class', axis=1)
y = df['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)

# Standardize features
scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)

# Create a logistic regression classifier
cl = LogisticRegression()

# Train the classifier
cl.fit(X_train, y_train)

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

# Print classification report
print("Classification Report:\n", classification_report(y_test, y_pred))

# Compute confusion matrix
conf_mat = confusion_matrix(y_test, y_pred)

# Plot confusion matrix
plt.figure(figsize=(8, 6))
sns.heatmap(conf_mat, annot=True, fmt='d', cmap='Blues')
plt.title('Confusion Matrix')
plt.xlabel('Predicted Label')
plt.ylabel('True Label')
plt.show()
```

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

           0       1.00      1.00      1.00     56864
           1       0.86      0.58      0.70        98

    accuracy                           1.00     56962
   macro avg       0.93      0.79      0.85     56962
weighted avg       1.00      1.00      1.00     56962
```



DOCUMENATION:

Overview This document outlines the implementation of a machine learning model designed to identify fraudulent credit card transactions. The model utilizes a logistic regression algorithm and evaluates its performance using precision, recall, and F1-score metrics. The dataset is preprocessed and normalized to ensure accurate model training and testing.

Features Data Preprocessing: The dataset is loaded and split into features and labels. Features are standardized using the StandardScaler() method.

Model Training: A logistic regression classifier is trained on the standardized training data to classify transactions as either fraudulent or genuine.

Model Evaluation: The trained model's performance is evaluated on the test dataset using classification metrics:

Precision Recall F1-score Code Structure The code follows a structured approach to achieve the model's objective:

Import Libraries: Necessary libraries for data manipulation, preprocessing, model training, and evaluation are imported.

Data Loading: The dataset containing credit card transaction details is loaded using pd.read_csv().

Data Preprocessing:

Features and labels are separated. Features are standardized using StandardScaler(). Data Splitting: The dataset is split into training and testing sets using train_test_split().

Model Initialization and Training:

A logistic regression classifier (LogisticRegression()) is initialized. The classifier is trained on the standardized training data. Model Evaluation:

Predictions are made on the test dataset. Classification report, including precision, recall, and F1-score, is printed. A confusion matrix heatmap is plotted for visual representation. Recommendations for Enhancement Class Imbalance Handling: Implement oversampling or undersampling techniques to handle class imbalance for improved model performance.

Hyperparameter Tuning: Fine-tune the logistic regression model's hyperparameters to optimize its performance further.

In [ ]: