PRESTIGE
INSTITUTE OF MANAGEMENT, GWALIOR
UGC NAAC ACCREDITED 'A' GRADE INSTITUTE
UGC APPROVED AUTONOMOUS INSTITUTE

Project Report

Topic :- Credit Card Detection Machine Learning Project

Submitted To:-

Prestige Institute of Management, Gwalior

Submitted By:-

Pradumn

Surya Pratap Singh

BCA 6th SEM (A)

DECLARATION

We, **Pradumn and Surya Pratap Singh Chauhan,** students of BCA- 6th semester of Prestige Institute of Management, Gwalior, hereby declare that the project report entitled "**Credit Card Detection Machine Learning Project**" is submitted by us in the line of partial fulfilment of course objectives for the BACHELOR of COMPUTER APPLICATIONS.

We assure that this project report is the result of our own efforts and that any other institute for the award of any degree or diploma has not submitted it.

Date : May 11, 2024 Pradumn

Place: PIMRG Surya Pratap Singh

CERTIFICATE

This is to certify that **Pradumn and Surya Pratap Singh Chauhan** of BCA 6th (A) of Prestige Institute of Management, Gwalior, have successfully completed their Project Report. They have prepared this report entitled "**Credit Card Detection Machine Learning Project**" under my direct supervision and guidance.

Dr. Nitin Paharia

(Faculty guide)

Acknowledgement

We express our sincere gratitude to **Dr. Nitin Paharia** for giving us the opportunity to work under her guidance on the report entitled "**Credit Card Detection Machine Learning Project**".

We are grateful to our Director Dr. Nishant Joshi, Asst. Prof. Nitin Paharia (project Coordinator), Faculty Members and other friends for their valuable suggestions in the execution of report preparation.

We are also thankful to other staff that guided and helped us very kindly at each and every step whenever we required.

We also acknowledge & convey thanks to the library staff, computer department of PIMG for their kind and valuable support.

1: Credit Card Detection Machine Learning Project with Report:

Project Report:

Introduction

Credit card fraud is a major concern for both consumers and financial institutions. Fraudulent transactions can lead to financial losses and damage to the reputation of financial institutions. Machine learning techniques have been used extensively to detect fraudulent transactions. In this project, we use logistic regression to classify transactions as either legitimate or fraudulent based on their features.

Data

The data used in this project is a CSV file containing credit card transaction data. The data has 31 columns and 284,807 rows. The "Class" column is the target variable, which indicates whether the transaction is legitimate (Class = 0) or fraudulent (Class = 1).

Preprocessing

Before training the model, we first separate the legitimate and fraudulent transactions. Since the data is imbalanced, with significantly more legitimate transactions than fraudulent transactions, we undersample the legitimate transactions to balance the classes. We then split the data into training and testing sets using the train_test_split () function.

Model

We use logistic regression to classify transactions as either legitimate or fraudulent based on their features. Logistic regression is a widely used classification algorithm that models the probability of an event occurring based on input features. The logistic regression model is trained on the training data using the LogisticRegression () function from scikit-learn. The trained model is then used to predict the target variable for the testing data.

Evaluation

The performance of the model is evaluated using the accuracy metric, which is the fraction of correctly classified transactions. The accuracy on the training and testing data is calculated using the accuracy_score() function from scikit-learn.

Streamlit Application

We use Streamlit to create a user interface for the credit card fraud detection project. The Streamlit application allows the user to upload a CSV file containing credit card transaction data, and the uploaded data is used to train the logistic regression model. The user can also input transaction features and get a prediction on whether the transaction is legitimate or fraudulent.

Conclusion

In this project, we used logistic regression to detect fraudulent credit card transactions. We achieved a high accuracy on both the training and testing data, indicating that the model is effective at detecting fraudulent transactions. The Streamlit application provides an easy-to-use interface for detecting fraudulent transactions in real-time.

```
import numpy as np
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import accuracy_score
import streamlit as st

st.set_page_config(page_title="Credit Card Fraud Detection", layout="wide")


@st.cache(allow_output_mutation=True)
def load_data(file):
    return pd.read_csv(file)


def train_model(data):
    # separate legitimate and fraudulent transactions
    legit = data[data.Class == 0]
    fraud = data[data.Class == 1]

    # undersample legitimate transactions to balance the classes
    legit_sample = legit.sample(n=len(fraud), random_state=2)
    data = pd.concat([legit_sample, fraud], axis=0)

    # split data into training and testing sets
    X = data.drop(columns="Class", axis=1)
    y = data["Class"]
    X_train, X_test, y_train, y_test = train_test_split(
        X, y, test_size=0.2, stratify=y, random_state=2
    )

    # train logistic regression model
    model = LogisticRegression()
    model.fit(X_train, y_train)

    # evaluate model performance
    train_acc = accuracy_score(model.predict(X_train), y_train)
    test_acc = accuracy_score(model.predict(X_test), y_test)

    return model, train_acc, test_acc


def parse_transaction_string(transaction_string, feature_names):
    values = transaction_string.split(",")
    transaction = {}
    for i in range(len(values)):
        transaction[feature_names[i]] = float(values[i])
    return transaction


st.title("Credit Card Fraud Detection")
```

```
file = st.file_uploader("Upload a CSV file containing credit card transaction data:")
if file is not None:
    data = load_data(file)
    st.write("Data shape:", data.shape)

    model, train_acc, test_acc = train_model(data)

    st.write("Training accuracy:", train_acc)
    st.write("Test accuracy:", test_acc)

    # allow user to input transaction features and get a prediction
    st.subheader("Check a transaction")
    feature_names = data.drop(columns="Class", axis=1).columns
    transaction_string = st.text_input(
        "Enter transaction features (comma-separated)",

"0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0
.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0
,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0,0.0")
```

About The Code:

This is a machine learning project for credit card fraud detection. The project uses a logistic regression model to classify transactions as either legitimate or fraudulent based on their features.

The code begins with importing necessary libraries such as numpy, pandas, scikit-learn, and Streamlit. Then, the layout of the Streamlit application is set using the st.set_page_config() function. The load_data() function is defined to load data from a CSV file, which is uploaded by the user using the file_uploader() function. The train_model() function is defined to train a logistic regression model on the uploaded data. The function first separates the legitimate and fraudulent transactions, undersamples the legitimate transactions to balance the classes, and then splits the data into training and testing sets using the train_test_split() function. The logistic regression model is then trained on the training data and evaluated on the training and testing data using the accuracy_score() function.

The parse_transaction_string() function is defined to parse a comma-separated string of transaction features and convert it into a dictionary with feature names as keys and feature values as values.

Finally, the Streamlit application is created using the st.title() function to set the title, the file_uploader() function to allow the user to upload a CSV file, and the text_input() function to allow the user to input transaction features and get a prediction. The uploaded data is loaded using the load_data() function, and the model is trained and evaluated using the train_model() function. The training and testing accuracies are displayed using the st.write() function, and the user can input transaction features using the text_input() function.