# Credit Card Fraud Detection using Machine learning Algorithms.

# PARTICIPANTS AND WORKFLOW

Names and Respective Roles:

Hari Krishna

Roles: Code Development, Model Prediction & Evaluation and StreamLIT

kavya

Roles: Code Development, Exploratory Data Analysis and Git Hub

Hepzibah

Roles: Code Development, Feature Engineering, Algorithm Selection and Model Prediction

Pavan

Roles: Project Documentation, Dataset Selection, References from Existing Projects

Deepak

Project Documentation, Project PPT, Dataset Selection, References from Existing Projects

Our team meets during after-class hours and when everybody is available on campus to discuss the ideas, progress, and upcoming agendas that we need to work on. We also use Google Meets for online team communication to share our work and make changes in any code or documentation related to the project. We have also created a group UNT-Email id for communication and sharing purposes.

Moreover, we use the GitHub platform for software project storage, tracking, and collaboration. It allows us to easily share code files and collaborate with other team members on project code building. GitHub also functions as a social networking site, where we can freely network, collaborate, and pitch our work.

# Abstract

Credit card fraud detection is currently the most common problem in the modern world. This is due to an increase in online transactions as well as e-commerce platforms. According to the Nilson Report, global losses from card fraud are expected to total $397.4 billion over the next ten years, with $165.1 billion of those losses occurring in the United States. Credit card fraud occurs when a card is stolen and used for unauthorized purposes, or when the fraudster uses the credit card information for his own benefit.

We are currently dealing with a slew of credit card issues. The credit card fraud detection system was introduced to detect fraudulent activities. In real time, a machine learning model can detect any deviations from regular transactions and user behaviours. ML algorithms can reduce the risk of fraud and ensure more secure transactions by detecting anomalies such as a sudden increase in transactional amount or location change.

The primary goal is to concentrate on machine learning algorithms. The random forest algorithm, Decision tree algorithm and the logistic regression algorithm were used. The accuracy, precision, recall, and F1-score of the three algorithms are used to calculate the results. The Random Forest, Decision tree algorithm and Logistic regression algorithms are compared, and the algorithm with the highest accuracy, precision, recall, and F1-score are considered the best fraud detection algorithm.

# Mission:

The main goal of our project is to create a user-friendly interface for an accurate and efficient machine learning model where they can detect any transactions which were not made by them, and which makes them capable of finding out if any deceptive transactions are going on with their credit cards. This will allow people to be more aware of any mis-happenings in their credit cards and be able to protect their money. By implementing this we hope to make a good effort in helping society as a lot of duplicitous activities are going on in this world. We are dedicated to applying data science and machine learning techniques to deliver a beneficial tool for anyone interested in tracking their fitness progress and enhancing their exercises.

# Vision:

One of the major challenges in applying ML methods to the problem of credit card fraud detection is that most of the published work is impossible to replicate. This is due to the fact that credit card transactions are extremely private. As a result, the datasets used to train ML models for credit card fraud detection contain anonymised features.

We think that by using the power of data science and machine learning, we can effectively and efficiently assist people and businesses in achieving 100% fraud-free transactions. Our purpose is to detect fraudulent transactions in order to prevent the loss of money or an organization through online payments.

Data Specification (Explanation about the dataset. - 0.2, If merging two different datasets,what is the merging point, whether they are in same time frame and the source - 0.05)

# Flowchart of design:

Diagram

Description automatically generated

# Project Design:

Overview of the tools and frameworks:

Visual Studio Code – Visual Studio Code, also commonly referred to as VS Code, is a source-code editor made by Microsoft with the Electron Framework, for Windows, Linux and macOS. Features include support for debugging, syntax highlighting, intelligent code completion, snippets, code refactoring, and embedded Git.

Python – Python is a high-level, general-purpose programming language. Its design philosophy emphasizes code readability with the use of significant indentation via the off-side rule

Git Hub - GitHub, Inc. is an Internet hosting service for software development and version control using Git.

Lucid Chart - Lucidchart is a web-based diagramming application that allows users to visually collaborate on drawing, revising, and sharing charts and diagrams, and improve processes, systems, and organizational structures.

Scikit-learn - Scikit-learn is a popular machine learning library in Python that offers a range of tools for data preprocessing, feature selection, and model selection. It provides a variety of algorithms for classification, including logistic regression, decision trees, and random forests, which can be used for fraud detection.

Streamlit - Open-source framework for building web applications in Python. It provides a range of tools for creating interactive data visualizations and dashboards, which can be used for visualizing the results of fraud detection models.

Data Collection:

Collect a dataset of credit card transactions containing both fraudulent and non-fraudulent transactions. The dataset used was from Kaggle users Dhanush Narayanan R[4], a CSV dataset containing over 10 Lakh legitimate and fraud credit card transactions. Each column stood for one aspect of the distance from home, distance from last transaction, ratio to median purchase price, repeat retailer, used chip, used pin number, online order and fraud these features are used to create the network in a simple method of the random forest, logistic regression and Decision tree. Then we have loaded the dataset into our panda’s data frame.

Graphical user interface, application

Description automatically generated

 ·

Graphical user interface, application

Description automatically generated

Exploratory Data Analysis (EDA):

Exploratory Data Analysis (EDA) is an essential step in the machine learning process as it helps in gaining a better understanding of the data.

Performing EDA for checking for missing values and handle them appropriately, Missing data can significantly affect the accuracy of machine learning models, and it is important to address it appropriately.

Understanding the distribution of data, as using EDA bar graphs we have identified the number of fraud transactions and non-fraud transactions from our dataset.

Graphical user interface, application

Description automatically generated

Graphical user interface, application

Description automatically generated

Data Pre-processing and feature Engineering:

Data pre-processing, which comprises cleaning, manipulating, and preparing data for analysis, is an important stage in machine learning. It seeks to verify that the data used in machine learning models is correct, consistent, and relevant, with the goal of improving model performance.

We performed missing value and duplicate removal, as well as null value identification and removal, in order to determine the most significant features that contribute to fraud detection from the dataset.

Feature engineering is an essential part of building effective machine learning models, especially for credit card fraud detection. Computing the correlation matrix and identify highly correlated features for feature selection, so that these features can help us in getting more accuracy.

Graphical user interface, application

Description automatically generated

Model Building:

Once we have our dataset and features ready after clearing null values and unnecessary features, the next step is to select a machine learning model that can accurately predict whether a transaction is fraudulent or not.

Popular models for credit card fraud detection include logistic regression, decision trees, random forests. We have compared performance of these three models and chosen the one that gave us the best results.

Model training and Evaluation:

After selecting the three machine learning models logistic regression, decision tree and random forest have train it on our dataset and evaluate its performance.

We split the dataset into training and testing dataset to evaluate the performance of the models and we have included the precision, recall, f2-score and accuracy which are typically used for cred card fraud detection.

Chart

Description automatically generated

Graphical user interface, application

Description automatically generated

Model Deployment:

Streamlit Integration

Integrate the trained machine learning model into a Streamlit web application. Using Streamlit to create an interactive user interface that allows users to input credit card transactions and receive a prediction on whether the transaction is fraudulent or not.

And we deploy the Streamlit web application to our localhost and which give a proper User-Interface for end-users. And by monitor the performance of the deployed model to ensure that it is working correctly and providing accurate predictions.

# Project Results:

In this paper we developed a model for credit card fraud detection using machine learning resulted in high accuracy, precision, recall, and F1 scores using three different algorithms - decision tree, logistic regression, and random forest.

After training and testing the models on a dataset containing credit card transactions, the results showed that the random forest algorithm achieved the highest accuracy and F1 score as compared with other machine learning models (decision tree and Logistic regression model).

The conclusions drawn from this project are that machine learning algorithms, particularly random forest, can effectively detect credit card fraud with high accuracy and precision. However, it is important to note that the choice of algorithm may depend on the specific needs and resources of the organization using the model.

If given more time and/or data, further improvements to the models could be made. For example, additional features could be added to the dataset, or the hyperparameters of the algorithms could be tuned to optimize performance. Additionally, the models could be tested on new and unseen data to ensure their generalizability and robustness. And the finest possible user interface for the end user that interacts with our model via an application or a website.

Graphical user interface, application

Description automatically generated

Graphical user interface, application

Description automatically generated

# Project Milestone:

The project milestone was achieved when we successfully built and trained three machine learning algorithms, namely decision tree, logistic regression, and random forest, to detect credit card fraud. The algorithms were trained on a dataset of credit card transactions and evaluated on their accuracy, precision, recall, and f1 score.

The project milestone was reached when we identified the best-performing algorithm, random forest, and concluded the project by suggesting possible improvements to further enhance the performance of the algorithms. On top of that, we use streamlit to construct a user interface for our fraud detection machine learning model which was a unique way to represent from other referred project.

**Repository / Archive:**

Our project is available on Git Hub, where it may be readily collaborated with in teams and by individuals interested in credit card fraud detection.

<https://github.com/HariKrishnaJammula/Credit-Card-Fraud-Detection-using-Machine-Learning.git>

The link above provides everyone with access to our code, dataset, and Project report, which they can use as a reference.

**Appendix – Code:**

import streamlit as st

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

# Title of the app

st.title("Credit Card Fraud Detection Using Machine Learning Algorithms")

# Upload CSV file

uploaded\_file = st.file\_uploader("Upload your input CSV file", type=["csv"])

# Check if a file has been uploaded

if uploaded\_file is not None:

# Read the file

data = pd.read\_csv(uploaded\_file)

# Show the first 5 rows of the data

st.write("First 5 rows of the data:")

st.write(data.head())

# Show the shape of the data

st.write("Shape of the data:")

st.write(data.shape)

# Show the data types of the columns

st.write("Data types of the columns:")

st.write(data.dtypes)

# Show the summary statistics of the data

st.write("Summary statistics of the data:")

st.write(data.describe())

# Show the missing values in the data

st.write("Missing values in the data:")

st.write(data.isnull().sum())

# Show the correlation matrix of the data

st.write("Correlation matrix of the data:")

st.write(data.corr())

# Plot scatterplot matrix for all numeric columns

# st.write("Scatterplot matrix for all numeric columns:")

# sns.pairplot(data.select\_dtypes(include=['int64', 'float64']))

# st.pyplot()

fraud\_counts = data['fraud'].value\_counts()

# Create a pie chart with the fraud counts

fig, ax = plt.subplots()

ax.pie(fraud\_counts, labels=fraud\_counts.index, autopct='%1.1f%%')

ax.set\_title('Fraudulent Transactions')

ax.axis('equal')

st.pyplot(fig)

# Count the number of fraud and non-fraud records

fraud\_count = len(data[data['fraud'] == 1])

non\_fraud\_count = len(data[data['fraud'] == 0])

# Calculate the percentages

fraud\_percentage = fraud\_count / len(data) \* 100

non\_fraud\_percentage = non\_fraud\_count / len(data) \* 100

# Create the bar graph

labels = ['Fraud', 'Non-Fraud']

percentages = [fraud\_percentage, non\_fraud\_percentage]

# Display the chart using Streamlit

st.title('Percentage of Fraud and Non-Fraud Records')

st.bar\_chart({'Labels': labels, 'Percentages': percentages})

# Create your heatmap

fig, ax = plt.subplots()

sns.heatmap(data.corr().round(3), annot=True, vmin=-1, vmax=1, cmap="coolwarm", ax=ax)

sns.set(rc={"figure.figsize":(10,10)})

# Display the plot in Streamlit app

st.pyplot(fig)

import streamlit as st

from sklearn.model\_selection import train\_test\_split

st.title("Fraud Detection App")

# Define the feature columns and target variable

feature\_columns = ["distance\_from\_home", "distance\_from\_last\_transaction",

"ratio\_to\_median\_purchase\_price", "repeat\_retailer",

"used\_chip", "used\_pin\_number", "online\_order"]

target\_variable = "fraud"

# Split the data into training and testing sets

X = data[feature\_columns]

y = data[target\_variable]

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=39)

# Display the dataset

st.subheader("Dataset")

st.write(data)

# Display the feature columns and target variable

st.subheader("Feature Columns")

st.write(feature\_columns)

st.subheader("Target Variable")

st.write(target\_variable)

# Display the training and testing sets

st.subheader("Training and Testing Sets")

st.write("X\_train:", X\_train.shape)

st.write("y\_train:", y\_train.shape)

st.write("X\_test:", X\_test.shape)

st.write("y\_test:", y\_test.shape)

import streamlit as st

from sklearn.linear\_model import LogisticRegression

from sklearn import metrics

# Create a logistic regression model and fit it to the training data

logreg = LogisticRegression(max\_iter=200)

logreg.fit(X\_train, y\_train)

# Make predictions on the test data

y\_pred = logreg.predict(X\_test)

# Calculate the accuracy of the model

accuracy = metrics.accuracy\_score(y\_test, y\_pred)

# Display the accuracy score

st.subheader("Accuracy of logistic regression classifier on test set:")

st.write("{:.5f}".format(accuracy))

# Display the classification report

st.subheader("Classification Report")

report = metrics.classification\_report(y\_test, y\_pred, digits=6)

st.code(report, language="text")

import streamlit as st

from sklearn.ensemble import RandomForestClassifier

from sklearn import metrics

from sklearn.metrics import classification\_report

import streamlit as st

from sklearn.ensemble import RandomForestClassifier

from sklearn import metrics

# Create a Random Forest Classifier and fit it to the training data

rfc = RandomForestClassifier(n\_estimators=100)

rfc.fit(X\_train, y\_train)

# Make predictions on the test data

y\_pred = rfc.predict(X\_test)

# Calculate the accuracy of the model

accuracy = metrics.accuracy\_score(y\_test, y\_pred)

# Display the accuracy score

st.subheader("Accuracy of Random Forest Classifier on test set:")

st.write("{:.5f}".format(accuracy))

# Display the classification report

st.subheader("Classification Report")

report = metrics.classification\_report(y\_test, y\_pred, digits=6)

st.code(report, language="text")

RELATED PROJECTS

The editor of the blog Credit card fraud detection: how machine learning can protect your business from scams[2] describes a detailed review of credit card fraud detection of an organization or a single person. The blog also discussed who becomes a victim of credit card scams and how machine learning aids in fraud detection. When and how should machine learning be used in fraud detection processes? It covers all the fundamentals of credit card fraud detection.

The website analyticsvidhya.com [3] a brief description of machine learning algorithms and illustrates how the random forest algorithm and other required algorithms for our project can be used.

The research paper Enhanced credit card fraud detection based on attention mechanism and LSTM deep model[1] was published in springer, the goal of this paper is to create a novel system for credit card fraud detection based on sequential data modeling and LSTM deep recurrent neural networks which are very advance related to our project.

REFERENCES

[1] Ibtissam Benchaji, Samira Douzi, Bouabid El Ouahidi, and Jaafar Jaafari. Enhanced credit card fraud detection based on attention mechanism and deep model. Journal of Big Data, 8(1), 2021.

[2] Editor. Credit card fraud detection: How machine learning can protect your business from scams, Aug 2020.

[3] Sruthi E R. Understand random forest algorithms with examples (updated 2023), Feb 2023.

[4] <https://www.geeksforgeeks.org/a-beginners-guide-to-streamlit/>

[5] Dhanush Narayanan R. Credit Card Fraud Detection Dataset –

“ <https://www.kaggle.com/datasets/dhanushnarayananr/credit-card-fraud> “

[6] <https://discuss.streamlit.io/t/streamlit-server-1-5-1-from-command-line-hard-to-stop-with-ctrl-c/22277>

[7] <https://towardsdatascience.com/machine-learning-basics-decision-tree-regression-1d73ea003fda>

[8] <https://www.geeksforgeeks.org/random-forest-regression-in-python/>

[9] <https://github.com/benedekrozemberczki/awesome-fraud-detection-papers>