# Credit Card Transaction Fraud Detection Using Machine Learning

A Project Report submitted to the
Department of Software Engineering, Daffodil International University
in partial fulfillment of the requirements for the degree of
M.Sc. in Software Engineering under FSIT.


**Submitted By**

Md. Mehedi Hassan

ID: 0242220005343008

**Supervised By**

Md. Shohel Arman

Assistant Professor



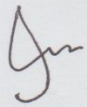

**Daffodil Smart City (DSC)**

**Birulia, Savar, Dhaka-1216, Bangladesh**

# Approval of Acceptance

This project titled on **Credit Card Transaction Fraud Detection,** submitted by **Md. Mehedi Hassan,** ID: 0242220005343008 to the Department of Software Engineering, Daffodil International University has been accepted as satisfactory for the partial fulfillment of the requirements for the degree of Master of Science in Software Engineering and approval as to its style and contents.

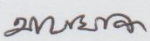
**BOARD OF EXAMINERS**



---

Chairman

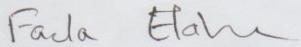
**Dr. Imran Mahmud**
Associate Professor and Head
Department of Software Engineering
Daffodil International University



---

Internal Examiner 1

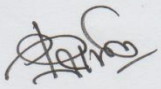
**Afsana Begum**
Assistant Professor
Department of Software Engineering
Daffodil International University



---

Internal Examiner 2

**Dr. Md. Fazla Elahe**
Assistant Professor and Associate Head
Department of Software Engineering
Daffodil International University



---

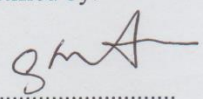
External Examiner

**Md. Tanvir Quader**
Senior Software Engineer
Technology Team
a2i Program

# Declaration

I, hereby, declare that the work presented in this report is the outcome of the investigation performed by me under the supervision of **Md. Shohel Arman**, Department of Software Engineering, Daffodil International University. The work was spread over one final semester course: Research Project, in accordance with the course curriculum of the department for the Professional Masters of Science in Software Engineering.

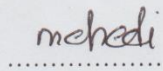
Certified by:



.....

**Md. Shohel Arman**
Assistant Professor
Department of Software Engineering
Daffodil International University

Submitted by:



.....

**Md. Mehedi Hassan**
ID: 0242220005343008
Enrolled Semester: Fall-2022
Department of Software Engineering
Daffodil International University

# Acknowledgment

First, I express my heartiest thanks and gratefulness to almighty for his divine blessing makes me possible to complete the final year project successfully.

I am really grateful and wish my profound indebtedness to my supervisor **Md. Shohel Arman,** Department of Software Engineering, Daffodil International University for his endless patience, scholarly guidance, continual encouragement, constant and energetic supervision at all stage have made it possible to complete this project.

I would like to express my heartiest gratitude to **Afsana Begum,** Coordinator, M.Sc. Program 2022-2023, Department of Software Engineering, for her kind help to finish our project and the honorable faculty members and the staffs of my department.

Finally, I must acknowledge with due respect the constant support and patients of my family.

# Abstract

Credit card fraud is a major challenge for Bank, business owners, credit card users, and transactions services companies, causing every year substantial and growing financial losses. Detecting fraud patterns in credit card transactions is very common problem which is hard to solve. With the ever-growing amount of data generated by credit card transactions, it has become impossible for a human analyst to detect fraudulent patterns in transaction data sets. As a result, the design of credit card fraud detection techniques has increasingly focused in the last decade on approaches based on machine learning (ML) techniques that automate the process of identifying fraudulent patterns from large volumes of data. This project focuses on predictions of whether a credit card transaction is fraudulent or no. To solve this problem, we first build a machine learning model. Then, use existing training data to train the model and evaluate how good its accuracy is.

**Keywords**: Machine Learning, Credit Card Transaction and Data Set.

# List of Abbreviations

| | |
|---|---|
| **ML** | Machine Learning |
| **AutoML** | Automated Machine Learning |
| **PCA** | Principal Component Analysis |
| **API** | Application Programming Interface |
| **SVD** | Singular Value Decomposition |
| **FIS** | Fuzzy Interference System |
| **SQL** | Structured Query Language |
| **AUC** | Area Under the Curve |
| **OOP** | Object Oriented Programming |
| **VS** | Visual Studio |

# Table of Contents

# List of Figures

**Figure**

# List of Tables

**Table**

# Project Outline

**Chapter I**- Provides information about Introduction, objective of the project.
**Chapter II**- Introduce the concept of detecting credit card fraud, Detection Credit Card Fraud with ML, Existing System, description about Problem, Dataset.
**Chapter III**- Provides information about Technology used, Machine Learning Process
**Chapter IV**- Implementation
**Chapter V**- Conclusion

# CHAPTER I

# Introduction

Now a days worldwide online purchases, bills, fees, and many other payments people can pay using credit card, so some peoples find out new ways to do transactions illegally. This is a very big issue in the modern world, anyone can do transactionseasily by entering credit card information which may be fraud or legal. To improve performance based on computerized collective data or previously acquired data is known as ML. A machine learning algorithm has two tracks: training and testing. Ifwe train data set using algorithm ML can predict with accuracy transactions is fraudor legal. There are ways to predict and find out fraud transactions using ML.

## 1.1 Overview

The main goal of this project is to implement ML Algorithms using Binary Classification, Anomaly detection to find out fraud transactions with credit cards apply with ML.NET and find the best algorithm with fraud detection accuracy.

## 1.2 Problem Statement

This problem focuses on predicting whether an initiated credit card transaction(with related information) is trustworthy or fraudulent. The input dataset of the transactions contains only numeric input variables that are the result of the previous Principal Component Analysis transformations. Unfortunately, for confidentiality issues, the main features and additional background info are not available, but theway I build the model doesn't change. V1 to V28 are the principal features agreedwith PCA, 'Amount' and 'Time', which were not transformed by PCA. 'Time' carriesthe elapsed seconds of the dataset. Credit card transaction 'Amount' example usefor dependent cost sensitive learning. Response variable 'Class' takes value 1 Fraud and otherwise 0. [1]. This dataset is very unstable, Class positive (fraud) accounts for 0.172Using those datasets, when predicting it will analyze a transaction's input variables and predict a fraud value of false or true.

## 1.3 Motivation

With ML Detect Credit Card Fraud is a data process exploration by a Data Scientist that will provide the best results in development of an anti-fraudulent transaction model. This is accomplished by uniting All the convenient features of credit card user's transactions, where Amount, Product Category, User Zone, Date, Provider, Client's Behavioral Patterns, etc. Those data then run through a subtly trained model that patterns find and rules so that it can categorize a transaction as fraudor as legal.

## 1.4 Objective

Detecting fraud is a set of operation that are taken to prevent money or property from being obtained through false pretenses. Cheating can happen in ways and across many industries. Most detection methods combine numerous fraud detection records to form a coherent summary of valid and invalid payment data to make a decision. This decision must consider IP address, geolocation, device identification, BIN data, global longitude or latitude, historic transaction patterns, and the actual transaction information. In practice, this means that traders and issuers use inside and outside information to apply a set of analytical algorithms or business rules to provide analytics-based responses that detect fraud.

Specific goals of this thesis that should be mentioned:

- Provides information about Introduction, objective of the project.
- Introduce the concept of detection credit card fraud, detecting Credit Card Fraud with ML, Existing System, description about Problem, Dataset.
- Provides information about Technology used, Machine Learning Process.

## 1.5 Assumptions & Limitations

- Credit card fraud not detect real time there are a fixed data set.
- Need to calculate and evaluate manually click there are no auto systems

# CHAPTER II

# Literature Review

## 2.1 Overview

Most of the existing system is based on Python or R Language. Here I tried to solve this problem using ML.Net.

## 2.2 Problem

This problem focuses on predicting whether a credit card transaction (including related information/variables) is fraudulent or not. The transaction input dataset carry numerical input variables only that are previous result of PCA transformations. Unfortunately for privacy reasons, the original attributes and other background dataare not available, but the way I build the model does not change. Features V1 to V28are main components acquired by PCA Amount, Time. property of Time containsthe elapsed seconds between every transaction and record of the first transaction. The 'Amount' is the Card transaction amount. This property used for example, for cost aware learning. The Class property is a feedback variable and has a value of 1 if true and 0 if false. [1]. Considering the positive class (fraud), the dataset is significantly imbalanced 0.172Using these datasets, I build a model that analyzes the transactional input variables while making predictions and predicts false or true fraudulent values.

## 2.3 Dataset

Training and test data were based on public datasets available on Kaggle, originally from Worldline and ML Group at ULB (http://mlg.ulb.ac.be ) collected by.The dataset holds credit card payment transactions from Cardholders of European in

Table 2.1: European Attributes dataset

| SL. | Features | Description |
|---|---|---|
| 1 | Time | between the cutting-edge transaction and the first transaction. |
| 2 | Amount | Transactions amount |
| 3 | Class | 0 - no fraudulent<br>1 – fraudulent |

Sep-2013. This dataset represents transactions made over 2 days, 4,444 out of 284,808transactions with 492 fraud cases.

### 2.3.1 Principal Component Analysis

Main factor analysis is an unmanaged mastering set of rules used for dimensional reduction in ML. that is a statistical manner that transforms correlated characteristic observations right into a linearly uncorrelated set of capabilities with the use ofan orthogonal transformation. those new converted capabilities are known as main components. it's far one of the maximum common gear used for exploratory facts evaluation and predictive modeling. a method to lessen variance and extract strong patterns from a given dataset

# CHAPTER III

# Technology Used & Machine Learning Process

## 3.1 Technology Used

Here I attempted to illuminate this issue utilizing .net system with computer program machine learning library ML.NET.

## 3.2 Prerequisites

- Visual Studio (I'm utilizing VS2022)
- ML.NET Show Builder (I'm utilizing v1.5.5)
- ASP.NET Core (I'm utilizing v5.0.0)
- Credit Card Extortion Discovery Dataset (I'm utilizing kaggle dataset)

## 3.3 .NET Framework

The .NET System could be a computer program system created by Microsoft. DOT NETFramework is totally free, open-source, cross platform, designer stage for building numerous distinctive application type using .NET, Create for web, portable, desktop, recreational and IoT with dozens of dialects, editors and libraries.

## 3.4 Languages

.NET applications can be written in C#, F#, or Visual Essential. Modern, simple, object-oriented, and kind-secure programming languages include C#. A programming language like F# might make it simple to write concise, active and effective code.

For developing kind-secure, OOP programs, Visual Essential is an approachable dialect with a true language structure.

## 3.5 Cross Platform

Cross Platform Regardless of whether I'm writing my code in C#, F#, or Visual Basic, any reliable OS will be able to execute it natively. Different.NET executions handle the heavy lifting for me. For example,.NET Center is a cross-platform.NET execution that can run console apps, servers, and websites on Windows, Linux, and macOS. Websites, services, mobile apps, and more are supported by Windows'.NET System. For walking programs on all of the essential portable running frameworks, Xamarin/Mono is a.NET application.

## 3.6 Libraries

Microsoft and other companies maintain a robust package environment based on.NET Standard to increase utility. Microsoft and other companies maintain a robustpackage environment based on.NET Standard to increase utility. Over 90,000 bundles are included in the package management NuGet, which was created specifically for this purpose.

## 3.7 Why C#

C# is one of GitHub's top five programming languages and a cutting-edge, creative, open-source, cross-platform object-oriented programming language.

## 3.8 ML.NET

A free, open-source, cross-platform device learning framework built just for is called ML.Net. Net developers. A free machine learning library for the C# and F# programming languages is called ML.Net. I developed and incorporated unique ML models into my.Net applications using ML.Net without the need for prior ML expertise. ML.Net is an extensible platform that powers diagnosis and has VS tooling as well as a cross-platform CLI. Microsoft uses technologies such as Windows, Power-Point Plan Ideas, Bing Advertisements, and much more [6]. The ML preview release. Internet changes include focused designs N-gram creation, newbie relief, etc. Regression tasks, multiple classifications, and double classification. Additional ML tasks

like unusual location and advice structures have already been handed, and alternative strategies like profound mastering may be covered in future modifications.

## 3.9 Machine Learning Process

- **Information:** The information can be crude, organized, and unstructured. It can be in any form as long as, the information is of tall quality to work upon.
- **Information Arrangement:** Within the information planning stage, the information are cleaned and wrangled and set up for the modeling.
- **Model Training:** The models are prepared utilizing different calculations as best suited for the reason of usage.
- **Evaluate:** The demonstrate is tested and assessed and the method of preparing and testing is iterated to deliver a brilliant demonstrate. With this an appropriate higher prediction model is created for use.
- **Deploy:** They assessed and tried information is presently conveyed as per the requirement of the framework.

## 3.10 ML Task - Binary Classification

Parallel or binomial classification is the assignment of classifying the componentsof a given set into two bunches (foreseeing which gather each one has a place to) onthe premise of a classification run the show. Settings requiring a choice as to whetheror not a thing has some subjective property, a few indicated characteristics.

## 3.11 ML Task - Anomaly Detection

Peculiarity (or exception) Location is recognizable evidence of something unusualand an event or observation that raises suspicion by varying basically from the majority of the data. Typically, the atypical things will decipher to a few kind of issue such as bank fraud, a basic deformity, restorative issues or blunders in a content.

## 3.12 Binary classification

An administered ML errand that's utilized to foresee which of two classes (categories) an occasion of information has a place to. The input of a classification

Calculation is a set of named cases, where each name is a numbers of either or 1. Output of the algorithm binary classification could be a classifier, which I can utilize to anticipate the class of unused unlabeled occurrences. Cases of parallel classification scenarios incorporate:

- Understanding opinion of Twitter comments as either" positive" either" negative".
- Diagnosing whether a quiet encompasses a certain infection or not.
- Making a choice to check an email as" spam" or not.
- Deciding on the off chance that a photo contains a specific thing or not, such as a pooch or natural product.

## 3.13 Binary classification trainers

One can prepare a binary classification show utilizing the taking after calculations algorithms [2]:

- PriorTrainer
- AveragedPerceptronTrainer
- GamBinaryTrainer
- SdcaNonCalibratedBinaryTrainer
- LightGbmBinaryTrainer
- LbfgsLogisticRegressionBinaryTrainer
- FastTreeBinaryTrainer
- FastForestBinaryTrainer
- SdcaLogisticRegressionBinaryTrainer
- FieldAwareFactorizationMachineTrainer
- SymbolicSgdLogisticRegressionBinaryTrainer
- LinearSvmTrainer

Table 3.1: Binary classification inputs and outputs

| Output Column Name | Column Type | Description |
|---|---|---|
| Score | Single | The crude score that was calculated by the show |
| Predicted Label | Boolean | The anticipated name, stands on the hints of the score.<br>A negative scores are equal to wrong and a positive scores charts to genuine. |

## 3.14 Inputs and outputs of Binary Classification

For best comes about with parallel classification, the preparing information oughtto be balanced (that is, rise to numbers of positive and negative preparing information). Lost values should be dealt with some time recently preparing. The input name column information must be Boolean. The input features column information must be a fixed-size vector of Single. These coaches yield the following columns:

## 3.15 Training Algorithm Details

Choice trees are non-parametric models that perform an arrangement of basic tests on inputs. This choice method maps them to yields found within the preparing dataset whose inputs were comparable to the occasion being handled. A choice ismade at each node of the double tree information structure based upon a degree of similitude that maps each occasion repetitively through the branches of the trees until the fitting leaf node is come to and the yield choice returned.
Decision trees have a few points of interest:

- They are productive in both computation and memory utilization amid preparing and prediction.
- They can speak to non-linear choice boundaries.
- They perform coordinates highlight determination and classification.
- They are strong within the nearness of boisterous highlights.

Quick woodland may be an irregular woodland usage. The show comprises of a package of choice trees. Every tree in a choice timberland yields a Gaussian dispersion by way of expectation. An accumulation is accomplished over the outfit of trees to

discover a Gaussian conveyance closest to combination dissemination all trees in the model. This choice timberland classifier comprises of a gathering of choice trees [7]. Generally, ensemble models give way better scope and exactness than single choice trees. Each tree in a choice timberland yields a Gaussian dispersion.

### 3.15.1 Random Forest

Subjective Forest can be an energetic machine learning calculation that can be utilized for an assortment of errands checking backslide and classification. It is an equip strategy, 10 meaning that a subjective timberland appear is made up a giganticnumber of small selection trees, called estimators, which each make their claim desires. The arbitrary forest show combines the figures of the estimators to make a more precise expectation.

### 3.15.2 Anomaly detection

This assignment makes an irregularity discovery show by utilizing Central Component Analysis (PCA). PCA-Based Peculiarity Discovery makes a difference me to construct a show in scenarios where it is simple to get preparing information from one course, such as substantial transactions, but troublesome to get adequate tests of the focused on peculiarities. An established technique in ML, PCA is habitually utilized in explorative information analysis forasmuch as it uncovers the inward formation ofthe information and clarifies the change in data. PCA works by analyses information that take on numerous factors. It looks for interrelations among the factors and decides the amalgamation of standards that better captures contrast in results. Theseassociated highlight standards are utilized to make a much compact highlight space called the central components.

Anomaly detection envelops numerous vital assignments in machine learning:

- Distinguishing exchanges that are possibly fraudulent.
- Learning designs that demonstrate that an organize interruption has occurred.
- Finding anomalous clusters of patients.
- Checking values entered into a framework.

Since irregularities are uncommon occasions by definition, it can be troublesome to gather a delegate sample of information to utilize for modeling. The calculations

Table 3.2: Anomaly Detection inputs and outputs

| Output | Type | Description |
|---|---|---|
| Score | Single | The non-negative, boundless score that was computed by the model of anomaly detection |
| Predicted Label | Boolean | A true/false grade illustrating if the input is an anomaly (Predicted Label=true) or not (Predicted Label=false) |

included in this category have been particularly outlined to address the center challenges of building and training models by utilizing imbalanced information sets.

### 3.15.3 Trainer of Anomaly Detection

I can prepare an irregularity discovery show utilizing the taking after algorithm:

- Randomized Pca Trainer

### 3.15.4 Inputs and outputs of Anomaly detection

Input highlights essential to be a static-sized vector of individual. This coach outputs the following:

### 3.15.5 Training Algorithm Details

This coach employment the beat eigenvectors to surmised the subspace containing the normal lesson. For each unused occurrence, it computes the standard contrast between the raw include vector and the anticipated include on that subspace. In casethe blunder is close to 0, the occasion is considered ordinary (non-anomaly). More particularly, this trainer trains a surmised PCA employing a randomized strategy for computing the singular value decay (SVD) of the framework whose columns are the input vectors. The model created by this coach contains three parameters:

- A projection lattice UU
- The cruel vector within the unique include space mm
- The cruel vector within the anticipated include space pp

For an input include vector xx, the inconsistency score is computed by comparingthe L2L2 standard of the first input vector and the L2L2 standard of the anticipated vector.

# CHAPTER IV

# Implementation

## 4.1 Proposed System Overview

Here I tried to solve this problem using ML.Net. In ML.Net Two ways I tried to find out the fraud Transaction and its accuracy.

- Binary Classification
- Anomaly Detection

Because ML.NET also has feature like Auto Machine learning features. Some scenario like Binary/Data classification, value prediction, Image Classification, Recommendation, Object Detection are included in Auto ML but scenario like Anomaly Detection, Forecasting & Clustering are not included in Auto ML. So I tried both way with Auto ML using Binary classification & without Auto ML using Anomaly Detection.

## 4.2 Fraud detection in credit cards (based on anomaly detection)

First I need to build a machine learning model. Then I train model with existing training data, evaluated its accuracy how good is, and lastly I consume the model to

Table 4.1: Fraud detection in credit cards (based on anomaly detection)

| ML.Net Type | API | Status | App | Data | Scenarios | ML Tasks | Algorithm |
|---|---|---|---|---|---|---|---|
| v 1.6.1 | Dynamic-API | Updated | Two console apps | .csv file | Fraud Detection | Anomaly Detection | Randomized PCA |

predict a fraud for a sample credit card transaction.

**ML (Machine Learning) Technique**

Machine learning is used for dynamic analysis for traffic and uses **WEKA** tool for detection method. It has two techniques for classification- supervised and unsupervised. In **Supervised** technique there is a training data set as input to train the system model for the expected output but in **unsupervised** technique there is no training/known data set and it works based on the prior knowledge or the statistical information.

Supervised technique has so many classifiers in order to classify data such as - KNN, LR, SVM, Random Forest, Naive Bayes, Bayesian Network etc. To achieve our system purpose efficiently we are using the supervised machine learning technique and trained our proposed FIS (Fuzzy Interference System) with the NSL-KDD dataset and the selected classifier.

## 4.3 Procedure

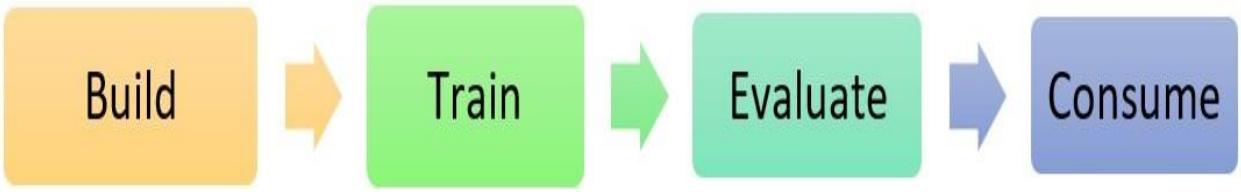



Figure 4.1: ML.NET process (anomaly detection)

In figure 4.1 shows ML.Net process/ steps for Anomaly Detection. Steps are -

### 4.3.1 Model Build

Building a model includes:

- Prepare data and split data for training and tests.
- Load the data with Text Loader by specifying the type name that holds data's schema to be mapped with datasets.
- Create an Estimator and transform the data with a Concatenate() and Normalize by LP Norm.
- Choosing a trainer/learning algorithm Randomized PCA to train the model with.

### 4.3.2 Train model

Preparing the show may be a preparation of running the chosen calculation on a preparing data to tune the parameters of the show. It is executed within the Fit () strategy from the Estimator protest. To perform preparing, I got to call the Fit () strategy while providing the preparing dataset (creditcard.csv) in a DataView object.

### 4.3.3 Evaluate model

I require this step to conclude how precise the show is. To do so, the model from the past step is run against another dataset that was not utilized in training (creditcard.csv). Assess () compares the anticipated values for the test dataset and produces different measurements

### 4.3.4 Consume model

After the demonstrate is prepared, I can utilize the Anticipate () API to anticipateon the off chance that a transaction is an extortion, employing a IDataSet.

## 4.4 Trainer Console Result

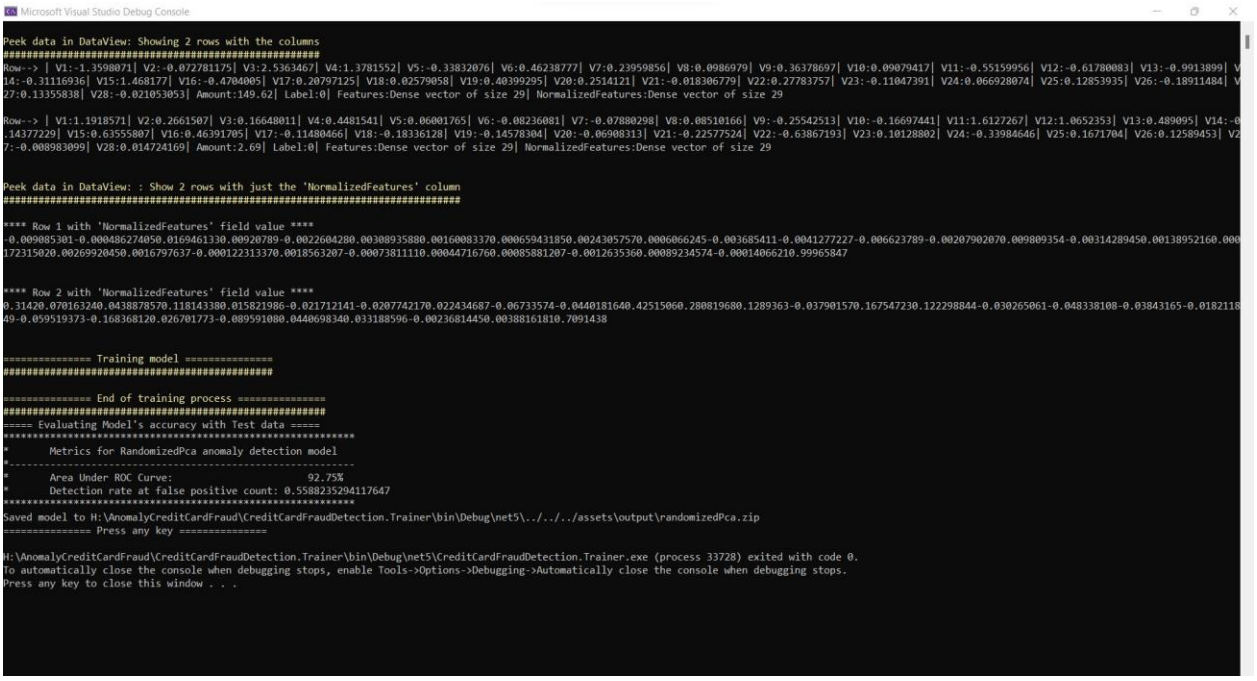



Figure 4.2: Trainer Console Result

In figure 4.2 shows Anomaly Detections Trainer Results. Showing 2 rows with the columns which data peek in DataView and shows ROC curve with Detection rate at false positive count.

## 4.5 Prediction Console result

In figure 4.3 Predictions from saved model: Test 5 transactions, from the test data source, that should be predicted as fraud (true).

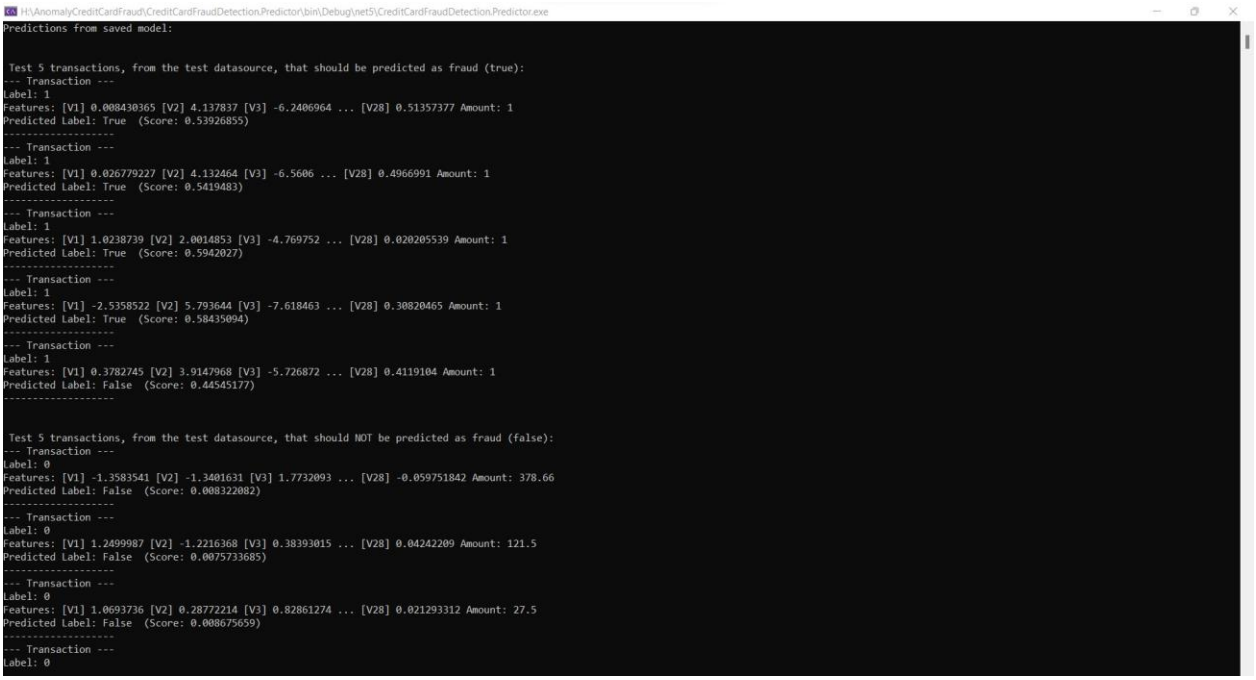



Figure 4.3: Prediction Console result screen 1

In figure 4.4 Predictions from saved model: Test 5 transactions, from the test data source, that should be predicted as fraud (false).

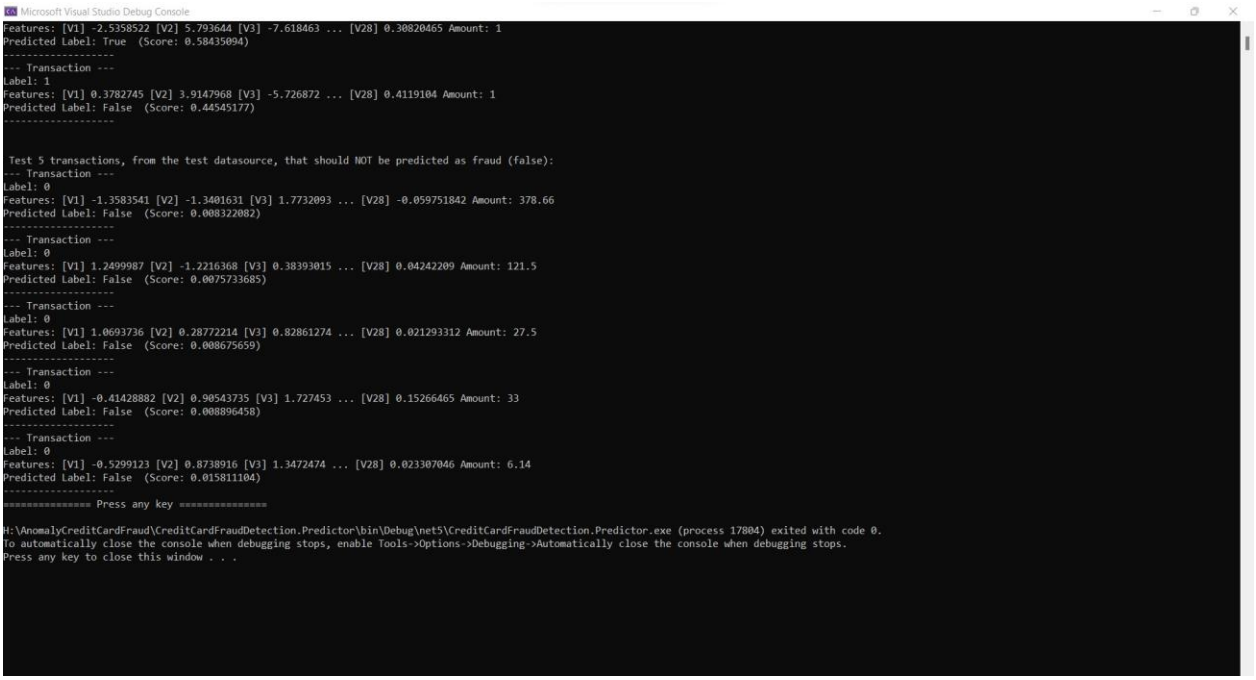



Figure 4.4: Prediction Console result screen 2

## 4.6 Fraud detection in credit cards (binary classification)

Table 4.2: Fraud detection in credit cards (binary classification)

| ML. NET | API | Status | App Type | Data | Scenarios | ML Task | Algorithm |
|---|---|---|---|---|---|---|---|
| v 1.6.1 | Dynamic-API | Updated | Web app | .csv file | Fraud Detection | Two-class classification | FastTree Binary Classification |

## 4.7 Solution

To illuminate this issue, to begin with construct a machine learning demonstrates.At that point I prepared the model on existing preparing information, assess how great its precision is, and in conclusion I consumed the demonstrate to anticipate an extortion for a test credit card exchange.


Figure 4.5: ML.NET process (Binary Classification)

In figure 4.5 shows ML.Net process/ steps for Binary Classification.

## 4.8 Procedure

ML.Net Builder for Model is an intuitive VS extension that builds, trains, and deploys custom gadgets that learn about models. Model Builder can use automated machine learning (AutoML) to discover my own gadgets, gain knowledge about algorithms and settings, and find better answers to the scenarios. Model Builder applications do not require gadgets for knowledge and understanding. All I want is the stats and the hassle is solved. Model Builder generates code that includes the versionin your .Net application. Model Builder is currently in preview.

### 4.8.1 Making a Demonstrate Builder Extend

Creating the Model Builder Project When first launched the Model Builder, myself were asked in the project name. This creates an mbconfig composition file in the project. [mbconfig] file tracks all operations in the Model Builder so myself restartthe session. After training mbconfig file:

- **[Model.consumption.cs]:** This file contains a [ModelInput] and Model Out- put schemas and the prediction functions generated to consume the model. It contains.
- **[Model.training.cs]:** This file holds the training pipeline (info transformations, algorithms, algorithmic hyper parameters) selected by A model maker to train the model. This pipeline can be used for model retraining.
- **[Model.zip]:** This is a sequential ZIP file represents a trained ML.NET model.When creating the mbconfig file, I was asked for name. This name applies to training, consumption and model file. In this situation the name used is model.

When I made the mbconfig record, I was provoked for a title. This title applies to utilization, preparing, and demonstrate records. In this case the title utilized is the model.

### 4.8.2 Scenario

Scenarios describe prediction myself want to make from the data. Each scenario is associated with a different machine learning task, including: Classifying fraudas true or false falls under the binary classification task. Therefore, choose a data classification scenario.

- Binary
- Multiclass
- Clustering
- Regression
- Recommendation
- Inconsistency detection
- Ranking
- Determining

For illustration, the situation of classifying extortion as genuine or wrong falls beneath the binary classification assignment. For this reason, select the information classification situation.

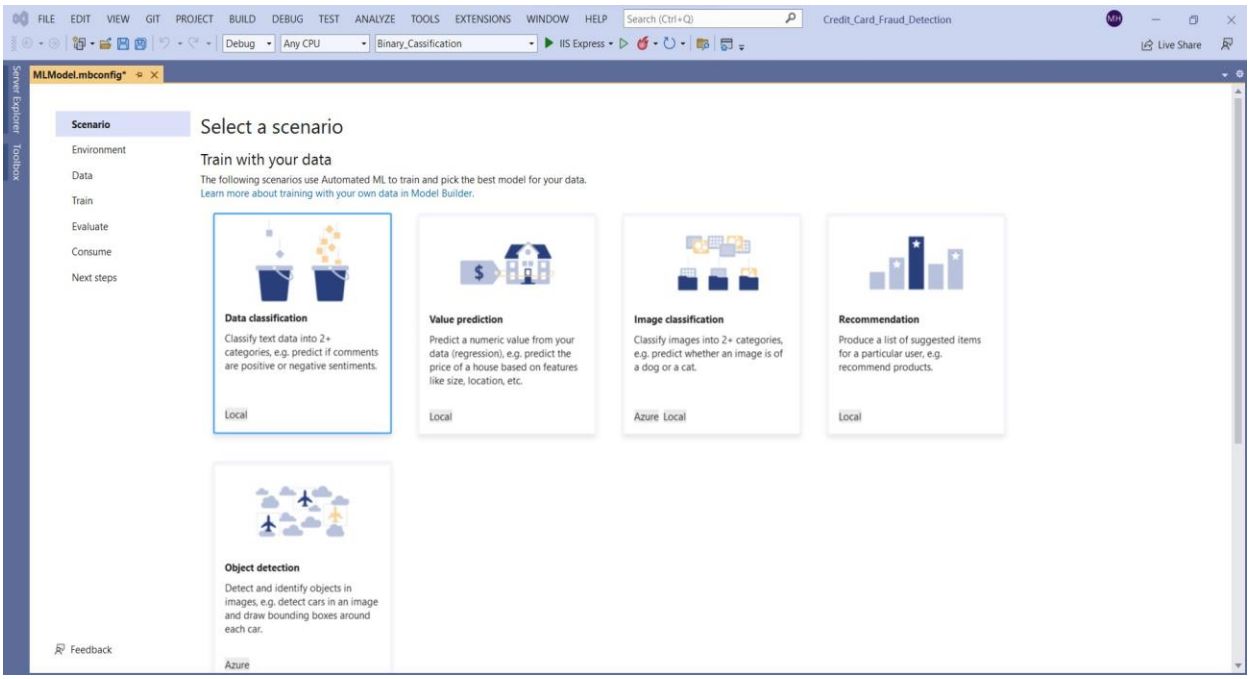



Figure 4.6: Select Scenario (Data Classification).

In figure 4.6 shows Data Classification Selection for Train data.

### 4.8.3 Environment

Depending at the scenario, a person or a user can educate his gadget getting to know the version domestically on his pc or with inside the cloud in Azure. When education domestically, paintings within side the limits of the pc sources (CPU, memory, disk). Training with inside the cloud lets in you scale your sources to satisfy the wishes of the scenario, particularly for huge datasets.

- Local CPU training is supported for all scenarios except object detection
- Image classification supports local GPU training.
- Azure training is supported for image classification and object detection.

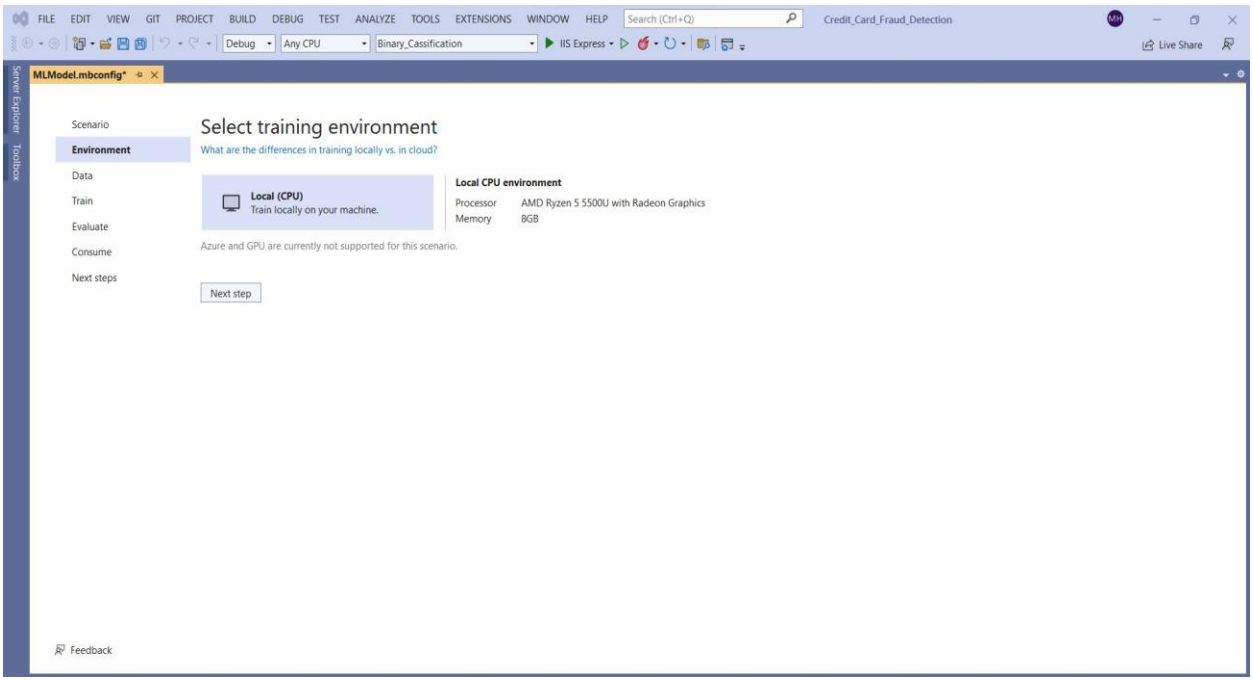



Figure 4.7: Select Training Environment (Local CPU).

In figure 4.7 shows Select training environment for Train data.

### 4.8.4 Data

After selecting a scenario, the model builder asks for a data set data is used for training, scoring and selection for the finest model of a scenario. The Model builder support set of records in .txt, .tsv, .csv and SQL DB formats. If .txt file has, the columns will have ,, ; or : If the data set consists images files, they support .jpg and .png types.

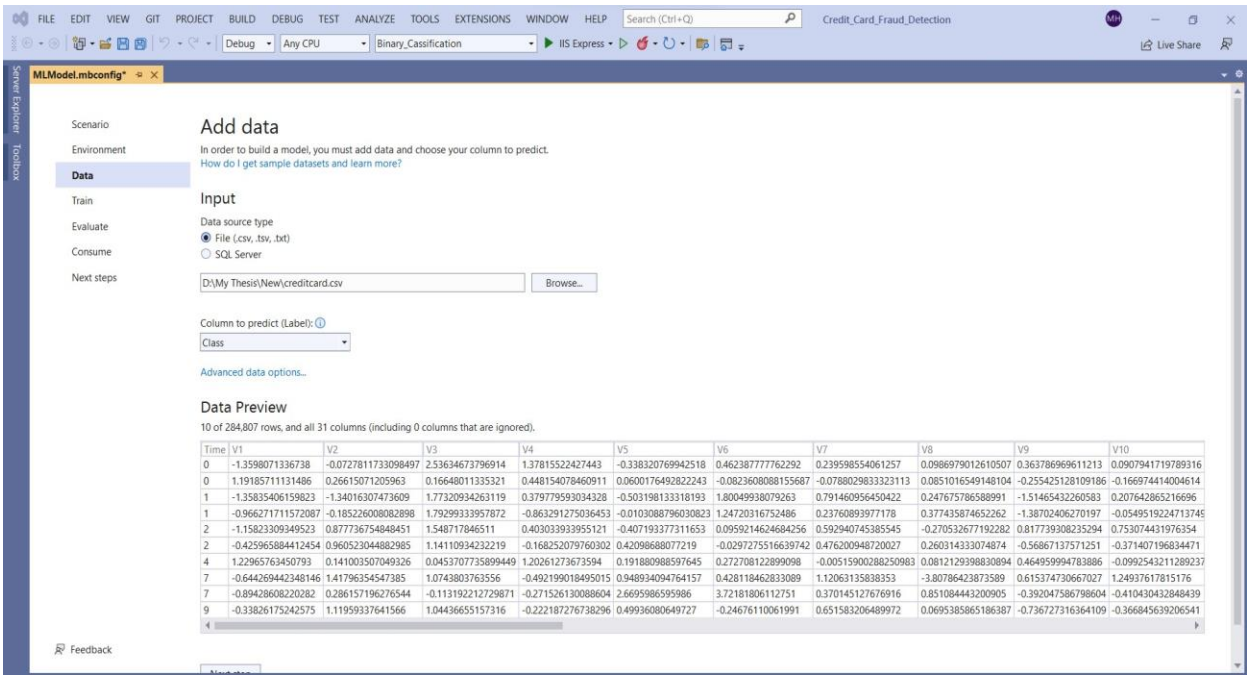



Figure 4.8: Add Data (Select creditcard.csv File from Local Path)

In figure 4.8 shows Add data file (.csv) for build a model.

### 4.8.5 Choose the output to predict (label)

Selecting Outputs (Labels) for Prediction A dataset table containing training samples rows and qualities of columns. Each line has:

- Labels (attributes we want to Prediction)
- Features (used as attributes input for predicting labels).

This is my output label class and the traits are the quantities, V1-V28 (anonymized traits).

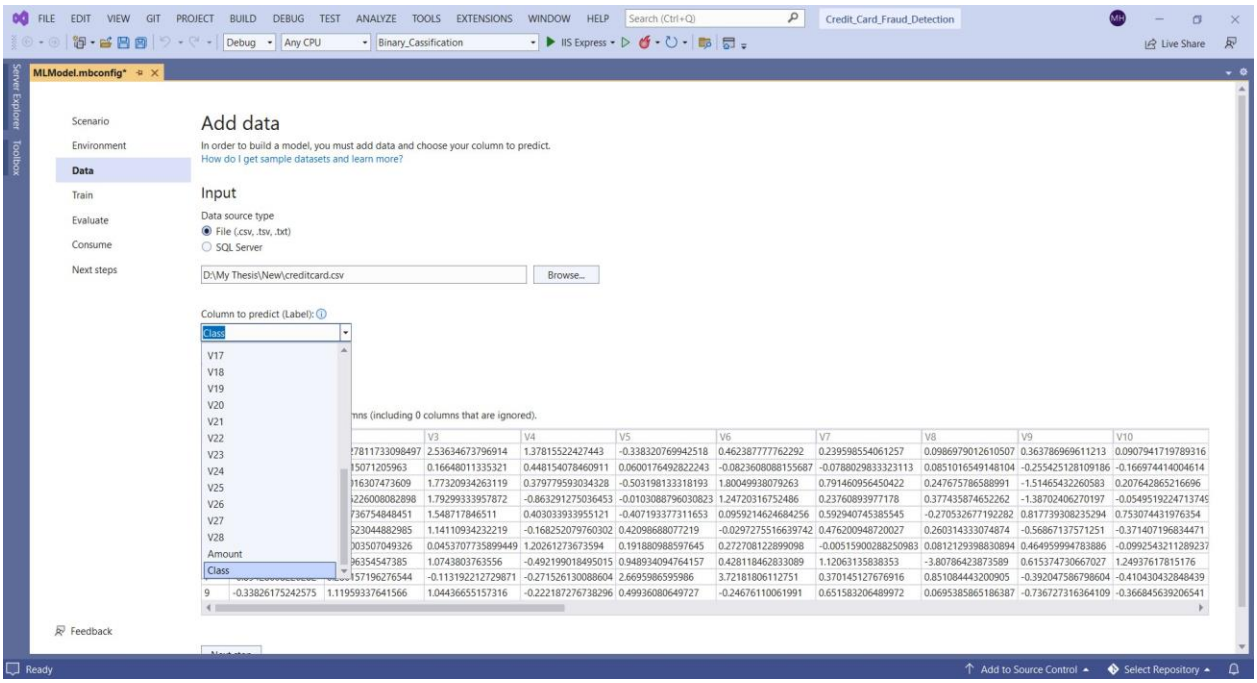



Figure 4.9: Label and Features (Column to Predict Label)

In figure 4.9 shows select column to predict (label) for response.

### 4.8.6 Train

After select scenario, data, environment, labels, Model Builder model can train model.

## 4.9 What is Training?

Training is defined as the procedure in which a model builder trained a model on how to response my own queries in a given scenario. Once model is trained, then the model is able to make assumptions on never-before-seen input data. Model Builder uses automated machine learning (AutoML), so no input or configuration is required during training.

In figure 4.10 shows time to train (seconds).

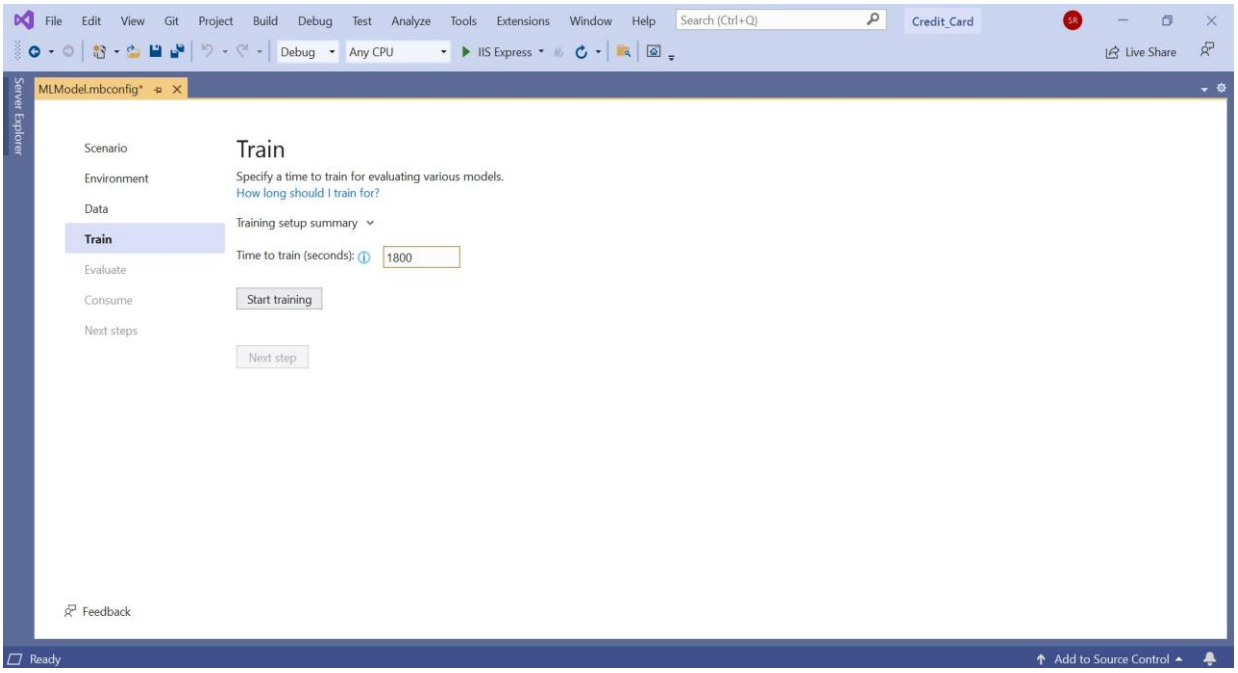



Figure 4.10: Train (Model Builder Trains the Model)

Table 4.3: Model Builder Dataset Wise Average Train Time

| Size of Dataset | Average time for train |
|---|---|
| 0 - 20 MB | 20 sec |
| 20 - 200 MB | 20 min |
| 200 - 700 MB | 35 min |
| 700 - 1.5 GB | 80 min |
| 1.5 GB+ | 3.5+ hours |

## 4.10 How long ought to I prepare?

Model Builder uses AutoML to examine different models and determine the better outcomes model. Longer preparation times permit AutoML to pursue many models with a settings of broad range. The following table outputs the typical time it takesto achieve better performance on a set of sample datasets on nearby computers.
These numbers are a direct as it were. The precise length of preparing is subordinate on:

- The input of the model is highlighted column
- The list of columns
- ML task

- In this training execution of machine, CPU, disk, and memory are used

It's by and large prompted that I can utilize more than 100 columns as datasets with less than that may not deliver any comes about.

In figure 4.11 shows training time (seconds) with models accuracy.

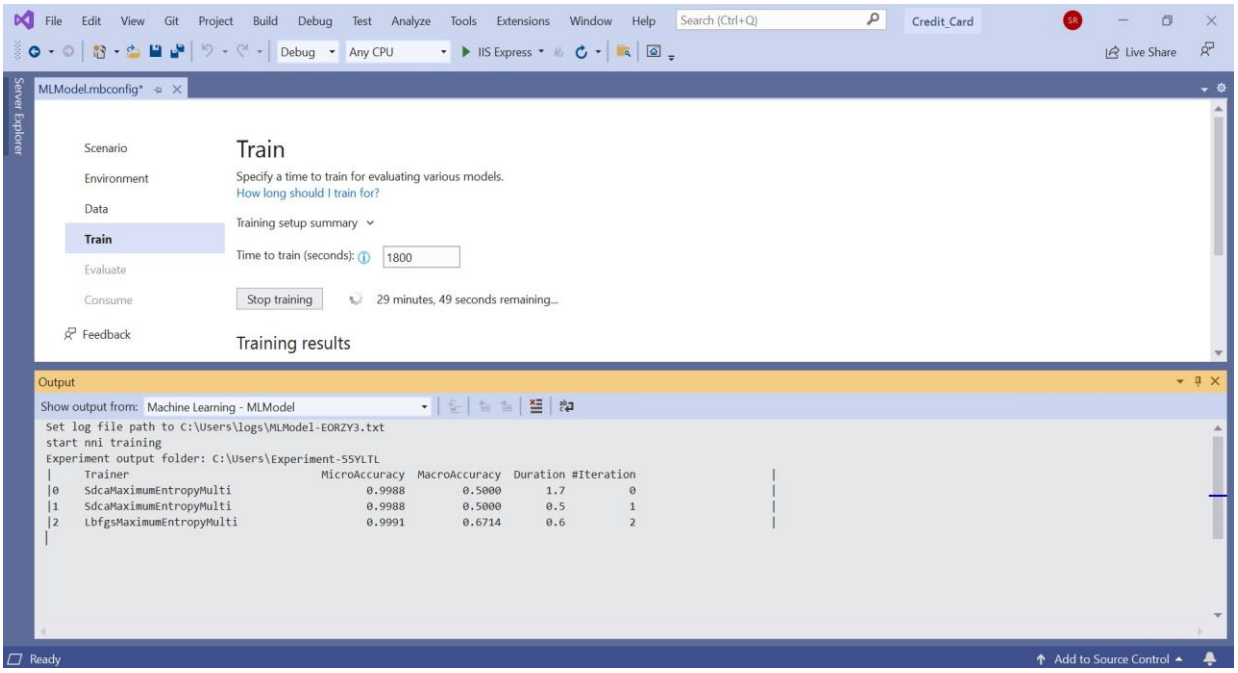



Figure 4.11: Training Time (Specify Time to Train for Evaluating Model)

In my scenario it will take 1800 seconds (30 minutes) time to train the machine.

Table 4.4: Top 5 Models

| Trainer | Micro Accuracy | Macro Accuracy | Duration | Iteration |
|---|---|---|---|---|
| 499 FastForestOva | 0.9998 | 0.9286 | 12.5 | 499 |
| 430 FastForestOva | 0.9998 | 0.9428 | 36.1 | 430 |
| 472 FastForestOva | 0.9998 | 0.9428 | 39.3 | 472 |
| 447 FastForestOva | 0.9998 | 0.9428 | 42.8 | 447 |
| 451 FastForestOva | 0.9998 | 0.9428 | 76.2 | 451 |

## 4.11 Top 5 models explored

- Best accuracy: 99.98
- Best Model: FastForestOva
- Training time: 1789.36 seconds
- Models explored (Total): 568

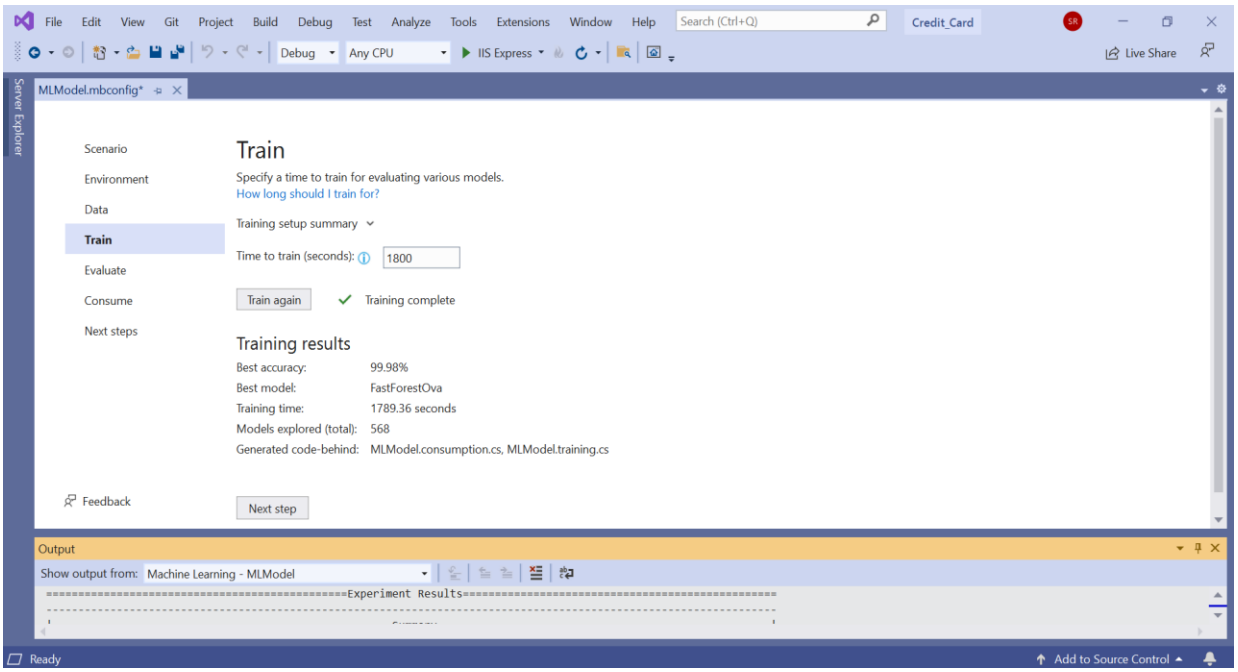



Figure 4.12: Training Results

In figure 4.12 shows training results with accuracy and best model (FastForestOva).

## 4.12 Evaluation

Evaluation is defined as the method of measured how excellent a model is. Model Builder uses a model trained on previously collected data to make predictions on new test data. Then evaluate the model to see how well its predictions match reality. A modeler divided the training data into a training and test set. Train model on training data (80%) and evaluate the model on test data (20%).

## 4.13 How can I understand the performance of my model?

Scenarios are a type of data that can be used in machine learning tasks. Every MLtask has its own performance metric. A standard metric for classifying problems is accuracy. A model's accuracy is defined by its ability to correctly predict the outcome of a test on a data set. If the model accuracy is judged to be good based on other metrics like AUC, the model accuracy should be at least 50%.

## 4.14 Consume

After the evaluation phase, the version generator outputs version documentationwith code. We use this to display the version in our application. ML.NET Fashion is saved as a .ZIP file. The code to load utilizing the version is introduced as a new task in the answer. Model Builder further provides a sample console application that we can run to peer versions with actions. Additionally, we can use our version to create tasks using the model builder. Model Builder is currently developing the following projects:

- **Console Application:** Create a console application .NET Core console application to predict from the version.
- **Web API:** Create an ASP.NET Core Web API that can be used to consume Releases over the web.

## 4.15 Implementation (binary classification)-

After consume the Machine Learning Model in my project I build simple user interface to detect the transaction is fraud or not. Here figure 4.13 firstly I enter the

1st transaction data from kaggle data set as input.

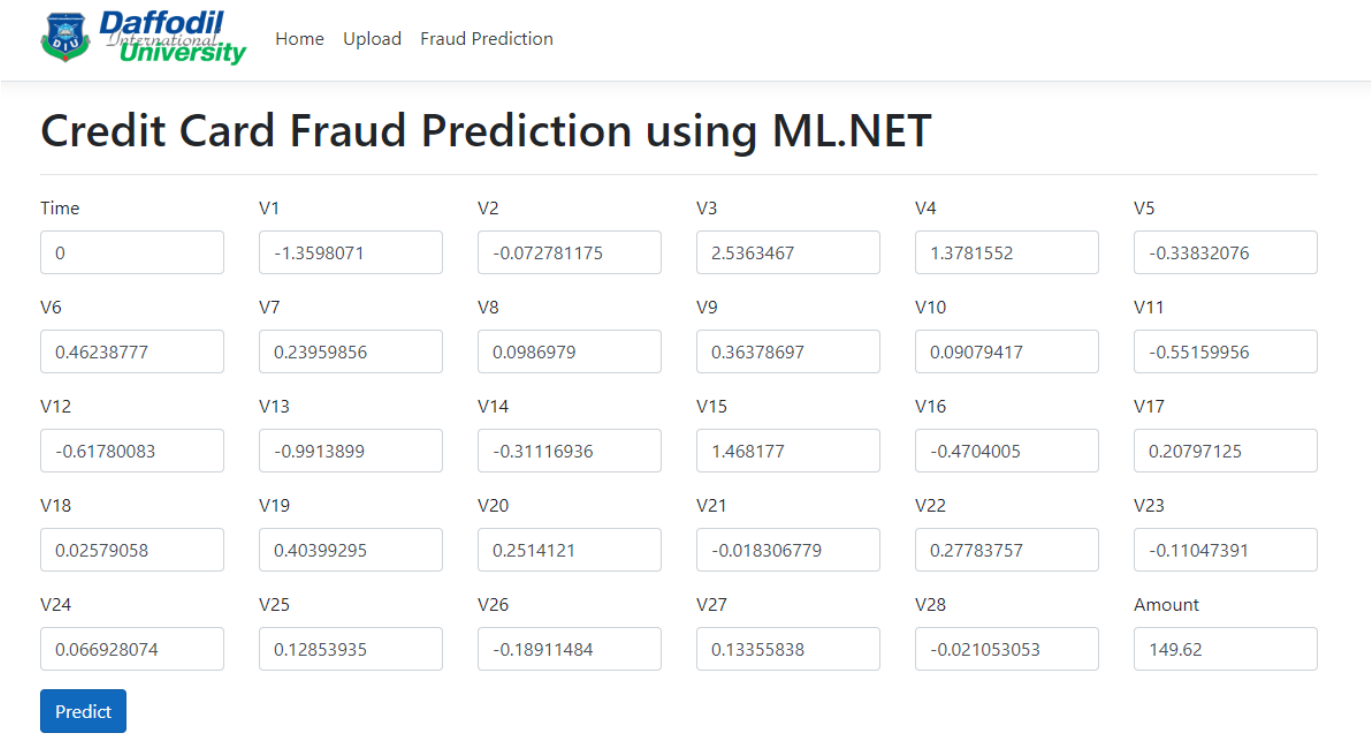



Figure 4.13: Implementation screen 1

After click on the predict button in figure 4.14 I get prediction result as below
Prediction: 0
Score: 0.9996613
Its mean the transaction is not fraud and its accuracy is 0.9996613.

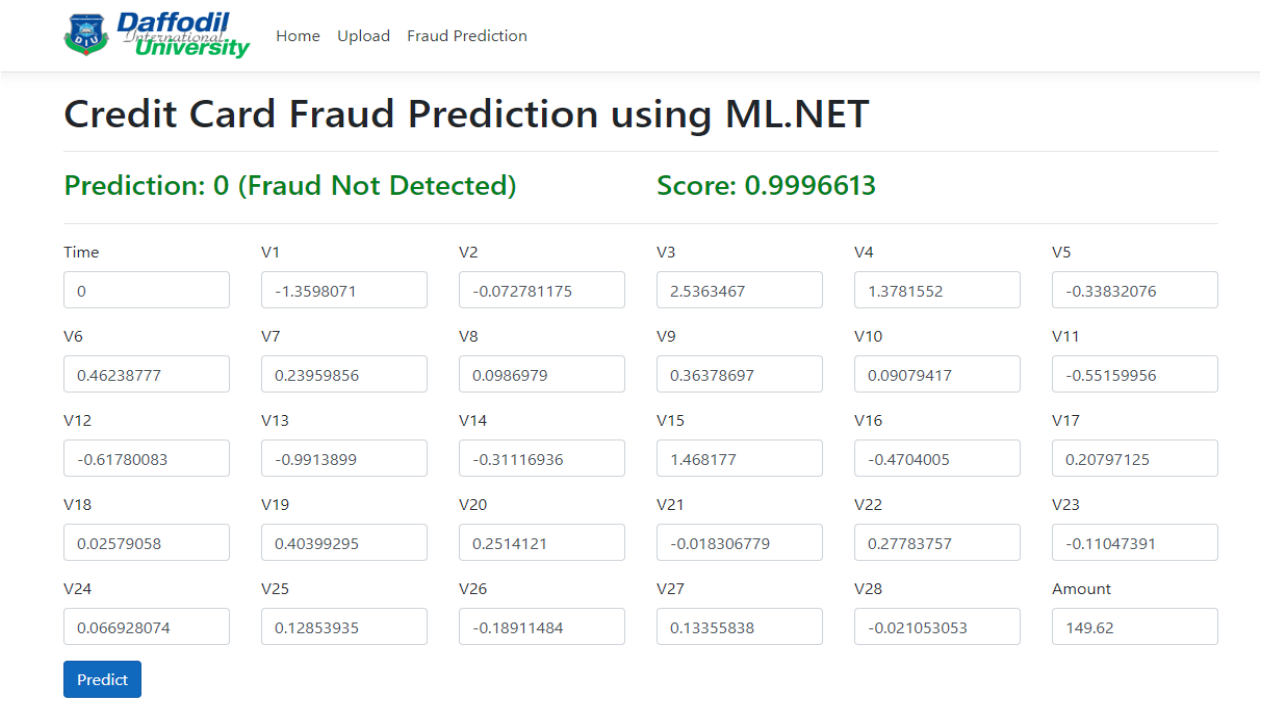



Figure 4.14: Implementation screen 2

Then I enter the 542nd transaction data from kaggle data set as input in figure 4.15.

After click on the predict button I get prediction result as below

Prediction: 1

Score: 0.0021387264

Its mean this transaction is fraud and its accuracy is 0.0021387264.

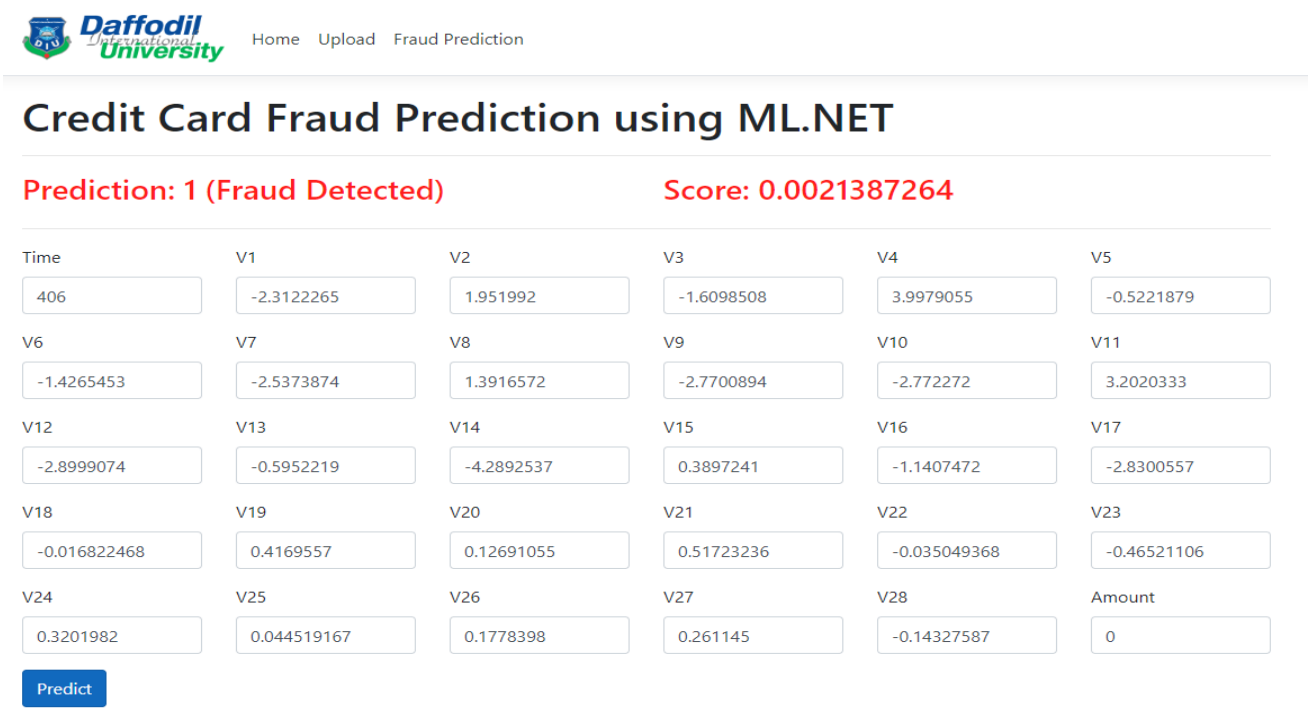



Figure 4.15: Implementation screen 3

In this Kaggle data set I found 492 Transaction is fraud from 284807 transactions.
Then I randomly change the input data to see is it give the accurate result.
After click on the predict button I get prediction result as below in figure 4.16
Prediction: 1
Score: 0.001042977
Its mean this transaction is fraud and its accuracy is 0.001042977

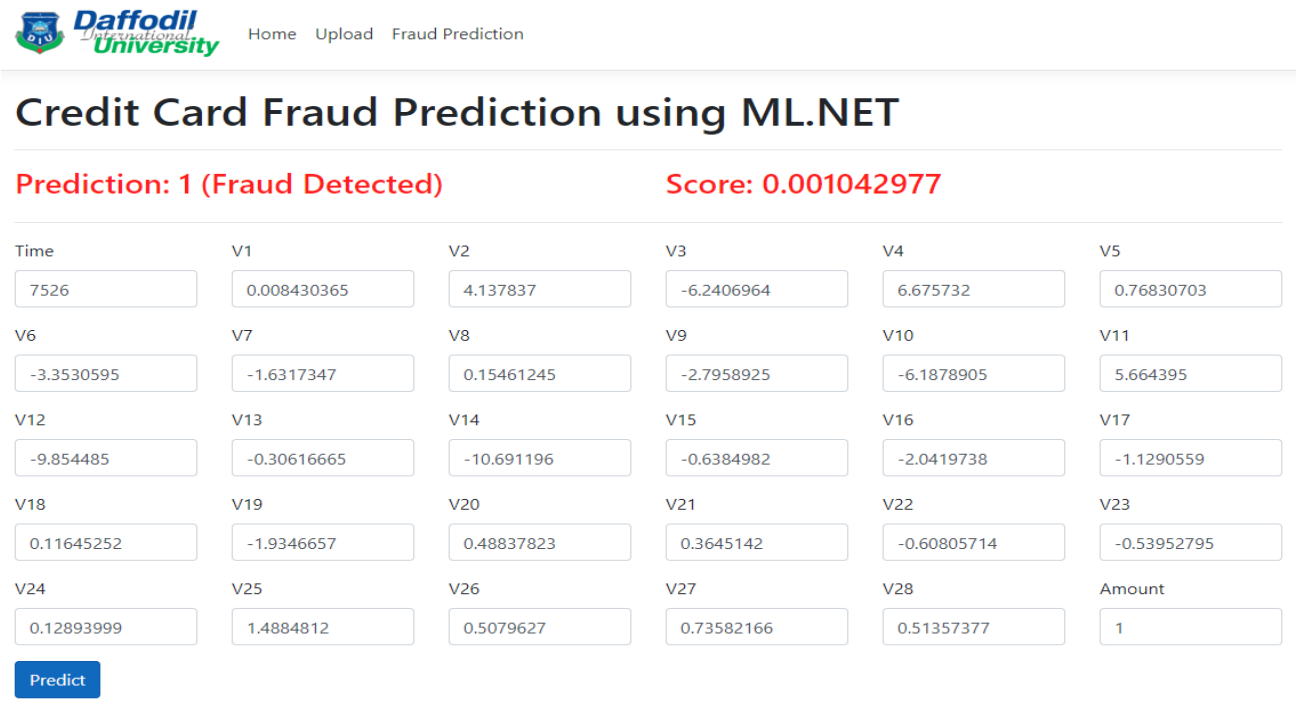



Figure 4.16: Implementation screen 4

# CHAPTER V

# Future Work & Conclusion

## 5.1 Overview

This chapter will conclude summary our future scope to further develop of my project.

## 5.2 Future Work

In the future, I will do more studies and will try to consume the best model in real time applications. Will try to integrate this process with a live payment gatewaysystem to predict fraud and stop if anyone can try for Credit Card fraud transactionspayment in Core Banking System and payment gateway on both ends with notify theclient. I will try to build an application with this process in the banking industry forpredicting and overcoming fraudulent transactions.

## 5.3 Conclusion

In this project, we explored machine learning applications such as anomaly detection, binary classification, random forests, and randomized PCA. The purpose of this project is to use ML.Net to create a ML algorithm for credit card fraud detection anddetermine the most accurate algorithm. There are two methods of fraud detection here. One method is anomaly detection and the other is binary classification. After completing the entire machine learning process in ML.NET the program can detect fraud in any credit card transaction and it also show the result accuracy.

# References


[1] "Credit Card Fraud Detection — kaggle.com." https://www.kaggle.com/mlg-ulb/creditcardfraud.

[2] "Ml.net: Machine Learning Made for .net — dotnet.microsoft.com." https://dotnet.microsoft.com/en-us/apps/machinelearning-ai/ml-dotnet.

[3] "Tutorials, API reference Microsoft docs — docs.microsoft.com." https://docs.microsoft.com/en-us/dotnet/machine-learning/?WT.mc_id=dotnet-35129-website.

[4] "Dotnet, Machinelearning-samples: Samples for ML.net, an open source and cross-platform Machine Learning Framework for .net.— dhttps://github.com." https://github.com/dotnet/machinelearning-samples.

[5] "kaggle.com." https://www.kaggle.com.

[6] "What is ML.net? an Open-Source Machine Learning Lramework.— https://dotnet.microsoft.com." https://dotnet.microsoft.com/learn/ ml-dotnet/what-is-mldotnet.

[7] "Fastforestbinarytrainer Class.— https://microsoft.ml.trainers.fasttree." https://docs.microsoft.com/en-us/dotnet/api/microsoft.ml.trainers.fasttree.fastforestbinarytrainer?view=ml-dotnet.

[8] "What is Model Builder and How Does it Work? - ML.net.— https://microsoft.ml.trainers.fasttree." https://docs.microsoft.com/en-us/dotnet/machine-learning/automate-training-with-model-builder.

[9] L. Delamaire, H. Abdou, and J. Pointon, "Credit card fraud and detection techniques: a review," *Banks and Bank systems*, vol. 4, no. 2, pp. 57–68, 2009.

[10] J. O. Awoyemi, A. O. Adetunmbi, and S. A. Oluwadare, "Credit card fraud detection using machine learning techniques: A comparative analysis," in *2017 international conference on computing networking and informatics (ICCNI)*, pp. 1–9, IEEE, 2017.

[11] R. Kumari and S. K. Srivastava, "Machine learning: A review on binary classification," *International Journal of Computer Applications*, vol. 160, no. 7, 2017.

[12] B. Mahesh, "Machine learning algorithms-a review," *International Journal of Science and Research (IJSR).[Internet]*, vol. 9, pp. 381–386, 2020.

[13] F. K. Sufi and M. Alsulami, "Automated multidimensional analysis of global events with entity detection, sentiment analysis and anomaly detection," *IEEE Access*, vol. 9, pp. 152449–152460, 2021.

[14] B. Mulloy, "Web api design," 2013.

[15] C. Notes, "Introducing the ml.net – a machine learning library for .net developers," Nov 2018.

[16] "Machine learning in ml.net.— https://custom software & it outsourcing." https://softwarehut.com/blog/tech/machine-learning.

[17] "Where developers learn, share, & build careers." https://stackoverflow.com/.

[18] V. Li, "Differences between classification and anomaly detection," Jul 2019.

[19] "Create a web api with asp.net core." https://docs.microsoft.com/en-us/aspnet/core/tutorials/first-web-api?view=aspnetcore-6.0\&tabs= visual-studio.

[20] "How to load data in ml.net model." https://devindeep.com/how-to-load-data-in-ml-net/.

# Account Clearance