A Work Project, presented as part of the requirements for the Award of a Master's degree in

Business Analytics from the Nova School of Business and Economics.

LEVERAGING MACHINE LEARNING TO MINIMIZE FRAUDULENT TRANSACTIONS: A STRATEGIC MODELING APPROACH

MAXIMILIAN RACH

Work project carried out under the supervision of:

Qiwei Han

Odhiambo Dormnic

14/12/2024

Abstract

Fraud has increasingly gained prevalence as millions of transactions are done online. Various stakeholders such as governments, organizations, and consumers have developed strategies to detect fraud and other unusual behavior. Machine learning techniques have been leveraged for fraud detection resulting in unique and sustainable solutions in financial transactions. In the modern age, machine learning algorithms have been widely utilized as a data mining technique for identifying issues with transactions. The current research aims to compare the effectiveness of three distinct machine learning models including GaussianNB, XGBoost, and Logistical Regression models by focusing on their precision, recall, and F1 score. Based on the outcomes of the three machine learning models, XGBoost is considered to be the best alternative for fraud detection at WeGoWin.

Keywords: Fraud detection, neural networks, machine learning, data mining

Table of Contents

Table of Figure and Tables

Introduction

The widespread adoption of e-commerce has revolutionized the way business is conducted in contemporary society as most transactions are done over computer-mediated networks. E-commerce has become increasingly prevalent due to its role in expediting orders, invoices, acknowledgements, and payments. According to Mohammed et al. (2017), the proliferation of e-commerce has been accompanied by numerous challenges including fraud. In the modern age, financial transaction fraud, money laundering, and other financial crimes have become rampant. Advancement in technology has made fraud highly dynamic and unpredictable which makes their detection challenging (Sadineni, 2020). Fraudsters have widely utilized emerging technology to bypass security checks resulting in loss of billions of dollars for numerous organizations. Analyzing and identifying fraudulent activities in transactions through data mining techniques is one of the ways of detecting fraudulent activities.

The advancement of digital technology in the present age has resulted in the emergence numerous approaches to fraud detection. Fraud, particularly in transactions, is a growing concern in most businesses and companies with far-reaching and astronomically expensive consequences. Fraud is becoming increasingly rampant as millions of transactions are done online. Various stakeholders such as governments, organizations, and consumers have developed strategies to detect fraud and other unusual behavior. There are different types of fraud transaction including skimming, phishing, account takeover, cards captured during shipment. losing card, card not present fraud, and fake website (Megdad, Abu-Naser, & Abu-Nasser, 2022). However, there are various challenges in fraud detection including inaccessibility, volume, unstructured, and imbalanced data. Emergent digital technologies such as big data analytics, artificial intelligence, and machine learning have resulted in

powerful techniques to identify fraudulent transaction. The financial technology industry has developed numerous techniques for fraud detection.

Machine learning techniques have been leveraged for fraud detection resulting in unique and sustainable solutions with a wide array of applications, particularly in financial transactions. In the modern age, machine learning algorithms have been widely utilized as a data mining technique for identifying issues with transactions. Machine learning is the process through which algorithms are taught to recognize patterns in the world, through automated analysis of large datasets. According to Ali et al. (2022), machine learning has produced systems robustly capable of learning from experience through neural network. Neural network refers to a way of organizing individual processing units into meshes that simulate the workings of the human brain (AbdulSattar & Hammad, 2020). Neural network derives its flexibility, and capacity for training from the numerical weighting of the connection between any two neutrons.

The process of training a machine learning model involves manipulating the weights between neurons to reinforce the particular neural pathways that represent successful recognition while suppressing the activation of the pathways that result in incorrect interpretation of the data. As the weightings between the layers are refined over multiple repetitions, a neural network learns how to identify and isolate complex features from unstructured and chaotic data (Raghavan & El Gayar, 2019). The layering of neural networks equipped with neurons tasked with identifying a specific kind of pattern in the world enables deep learning which involves the modeling of high-level abstractions. Deep learning empowers systems to perform complicated tasks without the explicit instructions to do so (Ali et al., 2022). The training of an algorithm occurs in one of two different ways including supervised learning and unsupervised learning.

In supervised learning an algorithm is given training examples and their accompanying labels while unsupervised deep learning involves setting loose the algorithms on the large datasets hence they are neither prompted nor guided. Ali et al., (2022) asserts that both supervised and supervised methods are used to identify unusual activities during transactions. Tasks to be performed by the supervised algorithm can either be binary or categorical. In binary tasks, the algorithm is given a series of transactions, some of which are recognizably fraudulent and asked to classify the transactions into valid and invalid ones. On the other hand, when dealing with categorical tasks the algorithm is asked to determine which category a specific transaction belongs to. Classification techniques are the most prevalent strategies utilized in detection of fraudulent transaction (Sadineni, 2020). Some of the deep learning methods leveraged in training algorithms for fraud detection include restricted Boltzmann machine (RBM), deep belief network (DBN), and convolutional neural network (CNN). Machine learning has taken center stage in fraud detection across various industries from financial industry to healthcare sector.

Research Significance

The current research is essential as it emphasizes the overarching need for the active detection of the risks of transactions considering its centrality in improving consumer experience and reducing financial loss. Fraudulent transactions have resulted in the loss of billions of dollars across numerous industries particularly the financial industry and the healthcare sector. Therefore, the current research is necessary because it proffers a solution for organizations plagued with fraudulent and other unusual transactions. Moreover, the current research offers incredible insight regarding the body of knowledge regarding the utilization of machine learning, big analytics, and artificial intelligence in addressing contemporary issues in the payment frameworks across different industries. The current research strives to highlight how machine learning techniques can be leveraged in the

detection of fraudulent transaction by using a machine learning model trained and evaluated by Gaussian Naïve Bayes (GNB), XGBoost Classifier (XGB), and logistic regression (LR).

Research Objectives

The current research is a direct research internship (DRI) at WeGoWin striving to compare the effectiveness of three machine learning models in fraud detection including GaussianNB, Logistical Regression, and XGBClassifier models. The research focused on determining the accuracy, recall, and precision of the three machine learning model to determine which one would be the most effective option for the organization.

Literature Review

According to West and Bhattacharya (2016), financial fraud is an issue with profound and far reaching consequences in the modern age detrimentally affecting government, finance, corporations, and consumers. Fraud undermines people's confidence in the financial industry, destabilizes the economy, and unnecessarily elevates people's cost of living. Karpoff (2021) defines fraud as the wrongful or criminal deception meant to result in financial gain, using it interchangeably with cheating, misconduct, and opportunism. The definition classifies fraud as any conduct that explicitly or implicitly violates agreements between parties to an economic transaction, regardless of whether it meets a legal definition of fraud. Fraud is an intricate and multidimensional concept both as a behavioral concept and a legal issue. However, in a financial context such as banking, securities, and insurance, fraud has a specific meaning with defining characteristics and is best aligned with the falsification or exploitation of financial information. Financial information is central to the financial transactions, particularly in the financial markets.

The proliferation of fraud in financial contexts has resulted in the emergence of money laundering, healthcare fraud, insurance fraud, credit card fraud, and securities fraud. The alarming increase in the rates of fraud in financial settings have inspired grave economic concerns. Financial fraud has resulted in the loss of billions of dollars globally every year. In the United States, financial fraud is responsible for the loss of over 400 billion dollars annually. The losses due to financial fraud are commonly borne by the merchants and corporations. For example, the credit card fraud, merchants and companies end up with chargebacks, administrative costs, and loss of trust. Consequently, the impact of fraud are far reaching hence there is an overarching need to design frameworks and technologies to detect and prevent them.

Fraud Detection

The traditional approaches for fraud detection relies on the manual techniques like auditing, which were highly ineffective due to the complexity of the issue. The rapid pace of technological advancement in the modern age have contributed to the increase in the widespread propagation of financial fraud in the modern age. Moreover, social factors such as increased penetration of credit cards and mobile payment frameworks have also provided numerous opportunities for fraud. Fraudsters have progressively improved and reined their strategies and techniques hence it is essential for the fraud detection methods to evolve with the times. Data-mining techniques are already integrated into the financial settings as they are utilized in credit card approval, share markets analysis, and bankruptcy prediction. Fraud detection is a crucial element of the financial industry with low frequency and high costs. Data-mining techniques were shown to be crucial in the detection of fraud due to their innate capacity to identify anomalies in vast amounts of data. Data mining approaches are applicable in fraud detection due to their efficiency at processing vast amount datasets and their capacity to operate without the knowledge of the input variables.

Fraud detection has progressively become a grave concern across various industry from FinTech to healthcare. As financial transactions undergo a paradigm shift due to the proliferation of e-commerce and the widespread integration of digital technologies in the payment methodologies, the risk of fraudulent transactions has increased exponentially. Businesses face substantial financial loss and reputational damage due to fraud hence fraud detection is essential in the highly volatile business landscape. Fraud detection is necessary in the corporate world because of its influence on financial loss, operational efficiency, brand equity, and regulatory compliance. Fraud detection helps organizations stem financial loss due to fraudulent activities such as embezzlement, unauthorized transactions, false claims which detrimentally impacts an organization's profit margins.

Similarly, fraudulent transactions chip at an organization's brand equity and loyalty as it reduces the trust consumers have in an organization. Consequently, fraud detection helps an organization retain its brand equity by securing its reputation. Fraud detection has also gained prevalence because it helps organizations to adhere to the anti-fraud regulation. Failure to detect and prevent fraud in an organization can result in significant losses and legal sanctions for an organization. Finally, fraud detection contributes to tan organization's operational efficiency by creating a culture of accountability.

Data mining approaches is one of the main strategies utilized to widely deal with fraud detection in the modern age. West and Bhattacharya (2016) defines data mining as any method that processes vast amounts of data to acquire the underlying insights. Machine learning is one of the data mining approaches leveraged in fraud detection. According to Greenfield (2017), machine learning is the mechanisms through which algorithms learn from their experiences, generalize from its encounters, and designs versatile solutions. Exploitation of machine learning is essential in the appreciation of the complex patterns and relationships embodied by fraudulent transactions. Leveraging machine learning algorithms in fraud

detection and prevention plays a significant role in deriving accurate inferences about potential threat. Moreover, machine learning algorithms aid businesses to adopt a proactive approach toward various threats by learning from past data.

Traditional Fraud Detection Strategies

Some of the traditional fraud detection machine learning algorithms include linear regression, decision trees, and random forests. The models offered incredible insight about the fraud processes empowering organizations with practical predictive attributes. Traditional fraud detection models fall under the statistical categories off data mining as they rely on traditional mathematical techniques such as logistical regression and Bayesian theory. The traditional algorithms have various benefits including simplicity and efficiency. However, as fraud becomes more complex the traditional models have continually fallen short due to their inability to the non-linear relationships, dynamic strategies and techniques, and high-dimensional data. The traditional fraud detection models are faced with feature engineering challenges, imbalanced data, and dynamic fraud behavior. The traditional fraud detection models depend on customized features which are faced with challenges during data extractions as fraudsters constantly change their strategies and methods.

Moreover, financial data is highly skewed as fraudulent transactions are significantly fewer compared to legitimate transactions. The highly imbalanced data sets result in biased models with limited utility in the volatile corporate landscape. Finally, the statistical machine learning algorithms struggle to keep up with the constantly evolving techniques and technologies adopted by fraudsters. The other classification of data mining techniques part from statistical methods include computational data mining approaches. According to West and Bhattacharya (2016) computational data mining methods are those that leverage modern intelligence techniques such as neural networks and support vector mechanisms.

Computational methods of data mining and the machine learning algorithms based on them are unique as they have the capacity to learn from the problem domain as opposed to the statistical methods which are rigid.

Machine Learning: Algorithm and Neural Networks

Greenfield (2017) reveals that the production of systems with the capacity to learn from experience have only existed since the beginning of the current century due to the limitations of existing hardware and doctrinal disputes, Neural network in the modern age is erected on a layered model of perception. The most basic feature of the neural network is the processing element known as input neurons. The input neurons mimic the workings of the human brain as they react to a particular stimulus. The responses from the input neurons are passed on to the next layer of neurons responsible for integrating them into coherent features. The coherence and specificity increases with each layer of neurons till the conditions for top-level recognition are met and the output neuron is triggered. In the neural network, the algorithm learns to identify and detect the subject by attending to the statistical regularities in the dataset it was trained on. Therefore, the computational machine learning algorithms are built in a cascade of neural responses from the bottom to the top.

Greenfield (2017) argues that neural networks derive their versatility and adaptability from the strength of the connection between two neurons. The strength between two neurons has a numerical weighting which can be altered by whoever is training the algorithm making the neural network highly flexible. Training the neural network involves manipulating the weights to reinforce the successful pathways while suppressing the wrong pathways. Through multiple repetitions, the weightings between the neural layers are improved empowering the neural network to between complex features from unstructured and chaotic data. The computational machine learning algorithms have a similar mechanism to improve

adaptability. Machine learning algorithms utilized in fraud detection can be classified based on their trainings. Consequently, there are different types of machine learning models leveraged in fraud detection including supervised, unsupervised, semi-structured and reinforcement learning.

Machine Learning Algorithms utilized in Fraud Detection

As the traditional fraud detection strategies fell short, new approaches leveraging machine learning emerged. New approaches to fraud detection fundamentally include the utilization of machine learning models which are highly adaptable to the rapidly evolving techniques and strategies employed by fraudsters. Some of the new machine learning algorithms utilized in fraud detection include Gradient Boosting Machines (GBM), neural networks, and extreme gradient boosting (XGBoost), and isolation forest. GBM is a machine learning algorithm m that integrates numerous weak learners like decisions trees into a strong predictive model. GBM fundamentally operates in three distinct steps including initialization, building trees, and weighted aggregations. The model begins by with an initial prediction and computes the differences between the actual and predicted values for each data point then it proceeds to the next step of building decisions trees focused on correcting the residual from the previous tree.

As the different trees are added, the errors are reduced. The final step of the model involves the assignment of weights to the trees depending on their performance in error reduction. GBM yields the final prediction which is the weighted sum of predictions from all trees. Typically, GBM reduces the errors hence the model learns from the previous mistakes, refining its detection techniques and strategies which considerably increases the accuracy pf its inferences and prediction. The ability of GBM to improve with every iteration makes it highly effective in fraud detection as it is highly adaptable. On the other hand, neural networks

is another machine learning model widely utilized in fraud detection as they progressively refine their pathways with every iteration. In fraud detection neural networks are utilized in a wide array of ways including transaction sequences, image based fraud detection, and ensemble approaches.

Neural networks can detect fraud through recurrent neural networks (RNN) which reviews and explores the data set over time to identify patterns and dependencies. Moreover, neural networks can identify anomalies during transaction through the analysis of images such as checks and identification cards through convoluted neural networks (CNN). Neural networks can be utilized in collaboration with other machine learning models to improve their accuracy. The other popular machine learning model utilized in fraud detection is XGBoost. XGBoost is prevalent in fraud detection due to its versatility, accuracy, and effectiveness in anomaly detection in vast datasets. XGBoost have been leveraged in various contexts to detect potential threats and fraud including mobile payment, credit card fraud, and state grid corporation of China dataset. The model has been successfully used in various industries illustrating its centrality in the fraud detection.

Isolation forest, commonly referred to as iForest, is a machine learning algorithm meant for outlier detection. The algorithm is based on the notion that outliers are few and unique hence they are easily isolated compared to other cases. Isolation forest is highly effective due to its capacity to interact with high-dimensional data as it focuses on isolating the outliers. Since the isolation forest depend on isolation of outliers it excels in identifying anomalies and rare behavior which effectively aligns with the nature of fraudulent transactions which are characterized by their low volume and unique attributes. The final reason for the effectiveness of isolation forest is its size and scalability as it does not use memory. Moreover, due to its high speed, isolation forests are widely utilized real-time fraud detection.

Methods

The current research focuses on highlighting how machine learning is leveraged to minimize fraud in financial contexts, particularly during transactions, consequently, the research hinges on critical realism as the research philosophy. A research philosophy primarily describes the development of knowledge and the nature of the knowledge built by the research. The research philosophy for the current research was critical realism which emphasizes the need to explain the observable phenomena through the underlying structures of reality that shape them. The philosophy is highly effective for the current research which strives to explain the role of machine learning in fraud detection through a machine learning model. Rooted in the critical realism research philosophy, the current research adopted deduction as the research approach. According to Saunders and Lewis (2017), deduction refers to the research approach characterized by testing a theoretical proposition through a research strategy designed to perform the test. The deductive approach to research is defined by five distinct steps including establishing research questions from extant theory, establishing the ways in which the questions can be answered, gathering data, data analysis, and confirmation of the initial theory.

The current research strives to illustrate how machine learning models can be leveraged to minimize fraudulent transactions hence it is a descriptive study as it accurately describes a phenomenon in a business context. The research strategy leveraged in portraying how machine learning models can be utilized in the reduction of fraudulent transaction in the modern business landscape was experiment. The primary objective of an experiment was to investigate the causal links between variables to determine whether changing one independent variable can result in changes in the dependent variable. In the current research, a machine learning model is designed and trained on three types of data to show how machine learning identifies errors and anomalies in financial data. Saunders and Lewis (2017) define

experiments as a research strategy that defines a theoretical hypothesis, chooses samples from known populations, allocates samples to varying experiment conditions, introduces changes to one or more variable, and measures the outcome.

An experiment has four distinct elements. First, an experiment involves the manipulation of the independent variable. In the current experiment the machine learning algorithm was trained using data from three data sources. The training of the machine learning model qualifies as the manipulation of the independent variable. The second defining element of an experiment is that it is controlled by holding all other independent variables constant. In the current experiment, the other independent variables were held constant by assuming that there were no other transactions. The third aspect of ann experiment is that the effect of the manipulation of the independent variable are observed on the variable data. In this research, the effect of the machine learning on fraud is evident from the operations of the machine learning model and its ability to identify anomalies in a vast amounts of data. The last element of an experiment is the prediction of events that will occur in the experimental setting. In this research, it is predicted that the machine learning model will increasingly become more precise in identifying anomalies in a large dataset consequently detecting fraud in financial contexts.

As a research strategy, experiment has five sequential steps. First, the identification and definition of the issue that is to be studied. In this research, fraud detection is the issue under scrutiny and it is described in the context of credit card fraudulent transactions. The second step of the experiment is the formulation of the research hypothesis which was that machine learning algorithms reduce fraudulent transactions by identifying anomalies by analyzing large amounts of financial data. The third step involves designing the experiment. in this research the experiment was deigned through the development of a machine learning model through python codes.

Designing the Machine Learning Model

The machine learning model leveraged in the current research is designed in seven main steps including introduction, overview, EDA, preprocessing, feature selection and model, model validation, and conclusion.

Model: Introduction

The introduction of the machine learning model has an empty markdown that is to be filled by some relevant information such as a brief background of the company and the issues the organization is facing. While this part is not filled, the model is meant to address fraudulent transactions however this part is not filled to show that it is an experiment and not a functional model in a company. The introductory part is also meant to include information such as the degree of the problem by clarifying what percentage of the transaction are fraudulent. The introductory part of the model is meant to be explanatory as it includes the classification of the issue as a binary issue as the transactions analyzed can either be legitimate or illegitimate.

The commentary introducing the model was also meant to include a definition of the datasets analyzed by the model. The initial part is a commentary with no functional significance as it is only meant to share information about the organization, its problem, the extent of the problem, and problem classification. The functional aspect of the introduction primarily involves setting up the environment. The model needs an environment which is set up by importing the pertinent python libraries for machine learning and data analysis. After importing all the necessary libraries, the programming environment is completed by loading all the necessary dataset and previewing them.

Overview

The second part of the machine learning model is titled overview and it leverages describe and information methods of data frames to provide the descriptive statistics and data

types of the existing data. Various aspects of the data uploaded is explored revealing crucial information about the data set such as the number of columns in the data, number of entries, the datatypes, and memory usage. Therefore, the overview offers an overall views on the datasets under scrutiny.

Exploratory Data Analysis (EDA)

In the current machine learning model, the third part is titled EDA which means exploratory data analysis. The section is meant to gain a better comprehension of the datasets and their relation with the target. In the section, the data is sampled as it is understood that the resources can not sufficiently be applied on the complete dataset. The EDA is also focused on balancing the data as the model is meant to deal with a binary issue but one side is overwhelmingly larger than the other. An inordinate portion of the data comprises legitimate transactions hence it is essential to balance out the data to ensure that the model is balanced. The EDA also included a generation of new features derived from the pre-existing columns. While these newly-developed features could have been included in the feature engineering, they were included as they are focused on describing the main attributes of the datasets. In the EDA, the correct datatypes were maintained.

Preprocessing

The forth section of the machine learning model is titled preprocessing and it involves various data transformation meant to ensure that the raw data can be easily modelled by the algorithm. Typically, the preprocessing section also involves preparing the features that will be essential for the formatting the raw data in a way that the algorithm can effectively model it. In this section of the model the categorical variables with low cardinality are encoded. Moreover, columns that summarize variables with many elements are created. These actions

are undertaken to reduce the meaningless and extra information in the model. The preprocessing is finished by the encoding of the target variable.

Feature Selection and Modeling

The fifth section of the model is title feature selection and modeling. The section is designed to perform various actions. First, the section is responsible for selecting features and it performs this function by Recursive Feature Elimination, Cross-Validation (RFECV). The next function this section involves splitting the data into training datasets and test datasets. After portioning the data, three models are chosen for testing and comparison. The three machine learning models selected in the current experiment are GaussianNB, Logistic Regression, and XGBClassifies. The section also included the building of pipelines as well as training the models.

Model Validation

The second last part of the machine learning model was titled Model validation and it primarily focused on reviewing the performance of the designed model. The section used the metrics from Python's Sklearn to establish and measure the performance of the model. Balanced accuracy score, confusion matrix, and classification report were the main metrics leveraged in determining the model's performance and utility.

Model: Conclusion

The final part of the machine learning model was titled conclusion and it was predominantly descriptive focusing on concepts like precision, recall, and F1 score. Precision explores the cost associated to predicting that a transaction is fraudulent when it is legitimate. ON the other hand, recall focuses on the costs associated to failing to identify a fraudulent transaction. The last element of the conclusion is referred to as F1 Score which focuses on the balance between precision and recall.

After designing the experiment, the fourth step in any experiment involves running the experiment and collecting the data. In the current research, the machine learning model was tested and run and the resulting data assembled to establish its significance.

Results

The machine learning model was loaded in an integrated development environment (IDE), specifically Virtual Studio Code (VSCode). In running the machine learning model, different sections yielded different results depending on their objectives. The functional elements of the machine learning model each yielded different results. Typically, the machine learning model's functional elements included introduction, overview, EDA, preprocessing, feature selection and modeling, and model validation.

Model: Introduction

In the model, the introduction's primary role was to set up the environment through three cells. The first functional cell imports the relevant python libraries including pandas, matplotlib, datetime, numpy, and seaborn. The second functional cell played a crucial role in importing drive from google.colab. The final cell in the introduction also contributed in setting the environment by loading the data set and revealing the preview. The model shows the previews of the datasets loaded into it including user's data, card data, and data.

User's Data

The first results of the model included the preview of the user's data which comprised various columns including id, current age, retirement age, birth year, birth month, gender, address, latitude, longitude, , per capital income, yearly income, total debt and credit score. And the number of credit cards as shown in figure 1 below.

'Users Data'

	id	current_age	retirement_age	birth_year	birth_month	gender	address	latitude	longitude	per_capita_income	yearly_income	total_debt	credit_score
0	825	53	66	1966	11	Female	462 Rose Lane	34.15	-117.76	$29278	$59696	$127613	787
1	1746	53	68	1966	12	Female	3606 Federal Boulevard	40.76	-73.74	$37891	$77254	$191349	701
2	1718	81	67	1938	11	Female	766 Third Drive	34.02	-117.89	$22681	$33483	$196	698
3	708	63	63	1957	1	Female	3 Madison Street	40.71	-73.99	$163145	$249925	$202328	722
4	1164	43	70	1976	9	Male	9620 Valley Stream Drive	37.76	-122.44	$53797	$109687	$183855	675

Figure 1: Screenshot of the User's data as previewed in the data

'Users Data'

age	retirement_age	birth_year	birth_month	gender	address	latitude	longitude	per_capita_income	yearly_income	total_debt	credit_score	num_credit_cards
53	66	1966	11	Female	462 Rose Lane	34.15	-117.76	$29278	$59696	$127613	787	5
53	68	1966	12	Female	3606 Federal Boulevard	40.76	-73.74	$37891	$77254	$191349	701	5
81	67	1938	11	Female	766 Third Drive	34.02	-117.89	$22681	$33483	$196	698	5
63	63	1957	1	Female	3 Madison Street	40.71	-73.99	$163145	$249925	$202328	722	4
43	70	1976	9	Male	9620 Valley Stream Drive	37.76	-122.44	$53797	$109687	$183855	675	1

Figure 2: Screenshot showing the other columns of the user's data as portrayed in the model

Card's Data

The other data set loaded and previewed in the model was the card's data. The data source had multiple columns including id, client id, card brand, card type, card number, expiry date, cvv, whether the card has a chip, number of cards issues, credit limit, date the account was opened, the year the pin was last changed, and whether the card exists on the dark web. The dataset is previewed as shown in the IDE with the model as shown in fugure2 and figure 3.

'Cards data'

	id	client_id	card_brand	card_type	card_number	expires	cvv	has_chip	num_cards_issued	credit_limit	acct_open_date	year_pin_last_changed
0	4524	825	Visa	Debit	4344676511950444	12/2022	623	YES	2	$24295	09/2002	2008
1	2731	825	Visa	Debit	4956965974959986	12/2020	393	YES	2	$21968	04/2014	2014
2	3701	825	Visa	Debit	4582313478255491	02/2024	719	YES	2	$46414	07/2003	2004
3	42	825	Visa	Credit	4879494103069057	08/2024	693	NO	1	$12400	01/2003	2012
4	4659	825	Mastercard	Debit (Prepaid)	5722874738736011	03/2009	75	YES	1	$28	09/2008	2009

Figure 3: screen shot of the card's data as previewed in the IDE

'Cards data'

ient_id	card_brand	card_type	card_number	expires	cvv	has_chip	num_cards_issued	credit_limit	acct_open_date	year_pin_last_changed	card_on_dark_web
825	Visa	Debit	4344676511950444	12/2022	623	YES	2	$24295	09/2002	2008	No
825	Visa	Debit	4956965974959986	12/2020	393	YES	2	$21968	04/2014	2014	No
825	Visa	Debit	4582313478255491	02/2024	719	YES	2	$46414	07/2003	2004	No
825	Visa	Credit	4879494103069057	08/2024	693	NO	1	$12400	01/2003	2012	No
825	Mastercard	Debit (Prepaid)	5722874738736011	03/2009	75	YES	1	$28	09/2008	2009	No

Figure 4: Screenshot of the card's data showing the hidden columns

Transaction's Data

The third data source loaded and previewed in the model is the transaction data which reveals various details about the transaction as shown by the column including the id, , date of transaction, client id, card id, amount, use chip, merchant id, merchant city, merchant state, zip, mcc, errors, and status. The figure below shows the preview off the transaction data as previewed through IDE.

'Transactions data'

	id	date	client_id	card_id	amount	use_chip	merchant_id	merchant_city	merchant_state	zip	mcc	errors	status
0	7475327	2010-01-01 00:01:00	1556	2972	$-77.00	Swipe Transaction	59935	Beulah	ND	58,523.0	5499	NaN	0
1	7475328	2010-01-01 00:02:00	561	4575	$14.57	Swipe Transaction	67570	Bettendorf	IA	52,722.0	5311	NaN	0
2	7475329	2010-01-01 00:02:00	1129	102	$80.00	Swipe Transaction	27092	Vista	CA	92,084.0	4829	NaN	0
3	7475331	2010-01-01 00:05:00	430	2860	$200.00	Swipe Transaction	27092	Crown Point	IN	46,307.0	4829	NaN	0
4	7475332	2010-01-01 00:06:00	848	3915	$46.41	Swipe Transaction	13051	Harwood	MD	20,776.0	5813	NaN	0

Figure 5: Screenshot of the preview of transaction data as shown in the IDE

Overview

The functional section referred to as overview has seven cells responsible for different functions. The first of the seven cells performs three functions including displaying the summaries of information, describing it, and showing its shape.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 2000 entries, 0 to 1999
Data columns (total 14 columns):
 #   Column             Non-Null Count  Dtype
---  ------             --------------  -----
 0   id                 2000 non-null   int64
 1   current_age        2000 non-null   int64
 2   retirement_age     2000 non-null   int64
 3   birth_year         2000 non-null   int64
 4   birth_month        2000 non-null   int64
 5   gender             2000 non-null   object
 6   address            2000 non-null   object
 7   latitude           2000 non-null   float64
 8   longitude          2000 non-null   float64
 9   per_capita_income  2000 non-null   object
 10  yearly_income      2000 non-null   object
 11  total_debt         2000 non-null   object
 12  credit_score       2000 non-null   int64
 13  num_credit_cards   2000 non-null   int64
dtypes: float64(2), int64(7), object(5)
memory usage: 218.9+ KB

None
```

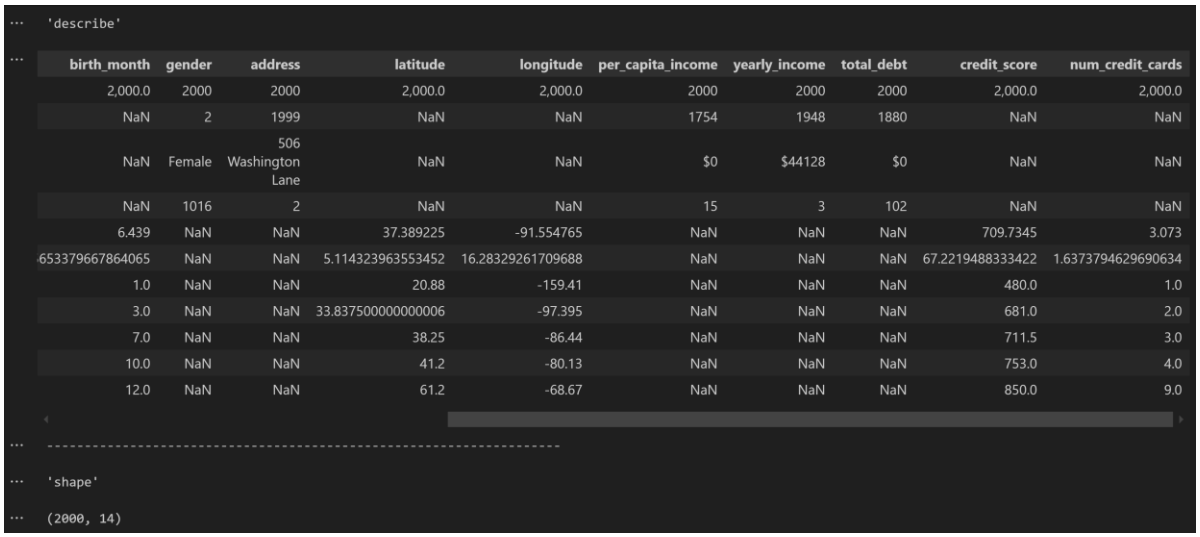
Figure 6: screenshot of the summaries of the user's data as shown in the IDE

'describe'

	id	current_age	retirement_age	birth_year	birth_month	gender	address	latitude	longitu
count	2,000.0	2,000.0	2,000.0	2,000.0	2,000.0	2000	2000	2,000.0	2,00
unique	NaN	NaN	NaN	NaN	NaN	2	1999	NaN	N
top	NaN	NaN	NaN	NaN	NaN	Female	506 Washington Lane	NaN	N
freq	NaN	NaN	NaN	NaN	NaN	1016	2	NaN	N
mean	999.5	45.3915	66.2375	1,973.803	6.439	NaN	NaN	37.389225	-91.554
std	577.4945887192364	18.414091537014993	3.628867328663949	18.42123399588	3.5653379667864065	NaN	NaN	5.114323963553452	16.283292617096
min	0.0	18.0	50.0	1,918.0	1.0	NaN	NaN	20.88	-159
25%	499.75	30.0	65.0	1,961.0	3.0	NaN	NaN	33.837500000000006	-97.3
50%	999.5	44.0	66.0	1,975.0	7.0	NaN	NaN	38.25	-86
75%	1,499.25	58.0	68.0	1,989.0	10.0	NaN	NaN	41.2	-80
max	1,999.0	101.0	79.0	2,002.0	12.0	NaN	NaN	61.2	-68

```
------------------------------------------------------------
'shape'
(2000, 14)
```

Figure 7: screenshot of the description of the summary of user's data



'describe'

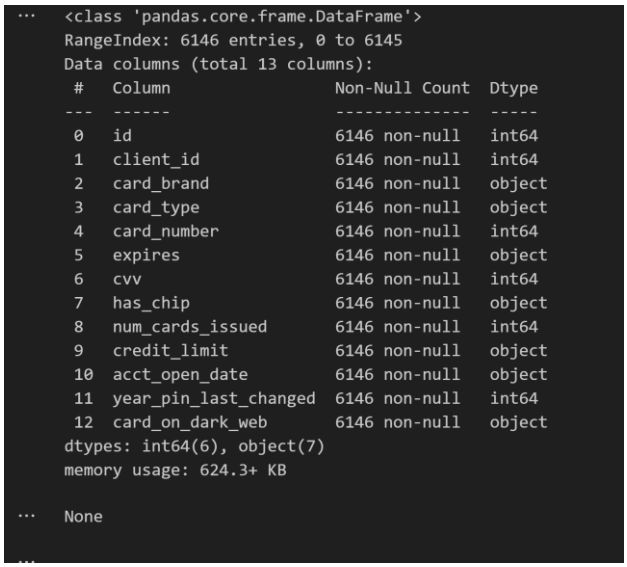
birth_month	gender	address	latitude	longitude	per_capita_income	yearly_income	total_debt	credit_score	num_credit_cards
2,000.0	2000	2000	2,000.0	2,000.0	2000	2000	2000	2,000.0	2,000.0
NaN	2	1999	NaN	NaN	1754	1948	1880	NaN	NaN
NaN	Female	506 Washington Lane	NaN	NaN	$0	$44128	$0	NaN	NaN
NaN	1016	2	NaN	NaN	15	3	102	NaN	NaN
6.439	NaN	NaN	37.389225	-91.554765	NaN	NaN	NaN	709.7345	3.073
653379667864065	NaN	NaN	5.114323963553452	16.28329261709688	NaN	NaN	NaN	67.2219488333422	1.6373794629690634
1.0	NaN	NaN	20.88	-159.41	NaN	NaN	NaN	480.0	1.0
3.0	NaN	NaN	33.837500000000006	-97.395	NaN	NaN	NaN	681.0	2.0
7.0	NaN	NaN	38.25	-86.44	NaN	NaN	NaN	711.5	3.0
10.0	NaN	NaN	41.2	-80.13	NaN	NaN	NaN	753.0	4.0
12.0	NaN	NaN	61.2	-68.67	NaN	NaN	NaN	850.0	9.0

'shape'

(2000, 14)

Figure 8: screenshot of the description of the summary of user's data with the other hidden columns shown

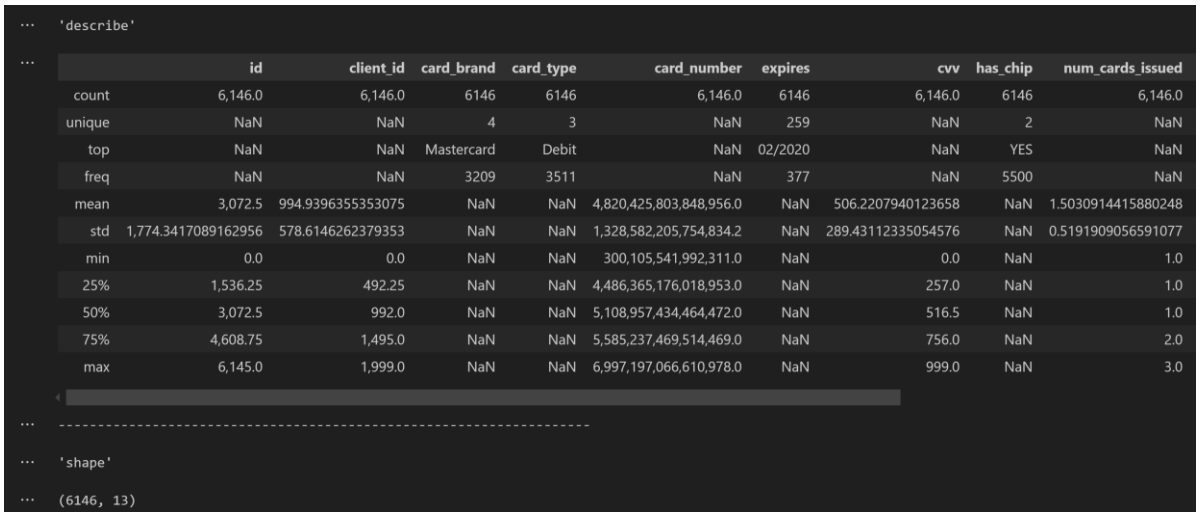
The second of the seven cells also offered a summary of the cards data, the description of the data along with its shape.



```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 6146 entries, 0 to 6145
Data columns (total 13 columns):
 #   Column                Non-Null Count  Dtype
---  ------                --------------  -----
 0   id                    6146 non-null   int64
 1   client_id             6146 non-null   int64
 2   card_brand            6146 non-null   object
 3   card_type             6146 non-null   object
 4   card_number           6146 non-null   int64
 5   expires               6146 non-null   object
 6   cvv                   6146 non-null   int64
 7   has_chip              6146 non-null   object
 8   num_cards_issued      6146 non-null   int64
 9   credit_limit          6146 non-null   object
 10  acct_open_date        6146 non-null   object
 11  year_pin_last_changed 6146 non-null   int64
 12  card_on_dark_web      6146 non-null   object
dtypes: int64(6), object(7)
memory usage: 624.3+ KB

None
```

Figure 9: screenshot of the summaries off the card data and its shape



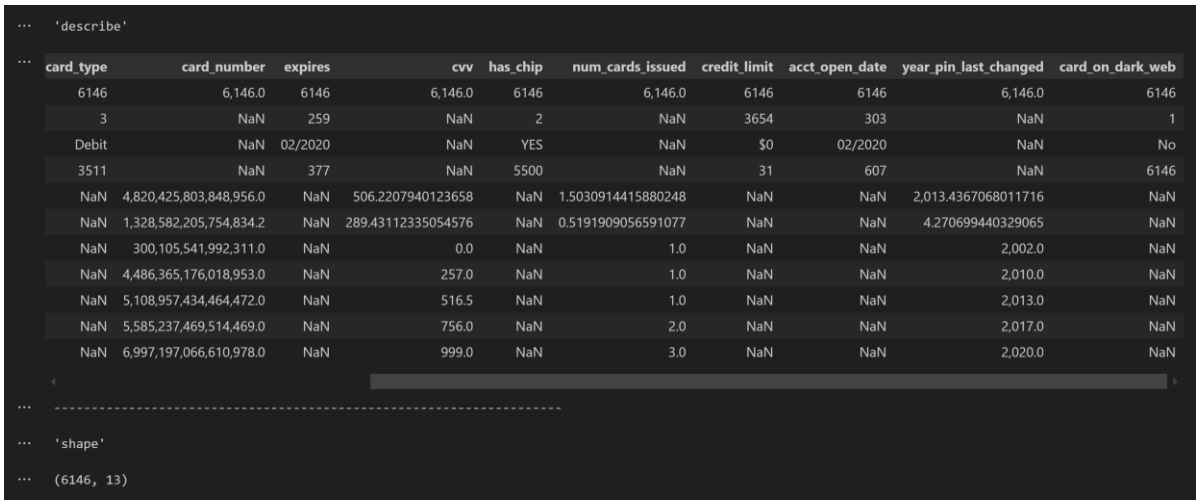
'describe'

	id	client_id	card_brand	card_type	card_number	expires	cvv	has_chip	num_cards_issued
count	6,146.0	6,146.0	6146	6146	6,146.0	6146	6,146.0	6146	6,146.0
unique	NaN	NaN	4	3	NaN	259	NaN	2	NaN
top	NaN	NaN	Mastercard	Debit	NaN	02/2020	NaN	YES	NaN
freq	NaN	NaN	3209	3511	NaN	377	NaN	5500	NaN
mean	3,072.5	994.9396355353075	NaN	NaN	4,820,425,803,848,956.0	NaN	506.2207940123658	NaN	1.5030914415880248
std	1,774.3417089162956	578.6146262379353	NaN	NaN	1,328,582,205,754,834.2	NaN	289.43112335054576	NaN	0.5191909056591077
min	0.0	0.0	NaN	NaN	300,105,541,992,311.0	NaN	0.0	NaN	1.0
25%	1,536.25	492.25	NaN	NaN	4,486,365,176,018,953.0	NaN	257.0	NaN	1.0
50%	3,072.5	992.0	NaN	NaN	5,108,957,434,464,472.0	NaN	516.5	NaN	1.0
75%	4,608.75	1,495.0	NaN	NaN	5,585,237,469,514,469.0	NaN	756.0	NaN	2.0
max	6,145.0	1,999.0	NaN	NaN	6,997,197,066,610,978.0	NaN	999.0	NaN	3.0

'shape'

(6146, 13)

Figure 10: A screenshot of the description of the card's data set as well as its shape



'describe'

card_type	card_number	expires	cvv	has_chip	num_cards_issued	credit_limit	acct_open_date	year_pin_last_changed	card_on_dark_web
6146	6,146.0	6146	6,146.0	6146	6,146.0	6146	6146	6,146.0	6146
3	NaN	259	NaN	2	NaN	3654	303	NaN	1
Debit	NaN	02/2020	NaN	YES	NaN	$0	02/2020	NaN	No
3511	NaN	377	NaN	5500	NaN	31	607	NaN	6146
NaN	4,820,425,803,848,956.0	NaN	506.2207940123658	NaN	1.5030914415880248	NaN	NaN	2,013.4367068011716	NaN
NaN	1,328,582,205,754,834.2	NaN	289.43112335054576	NaN	0.5191909056591077	NaN	NaN	4.270699440329065	NaN
NaN	300,105,541,992,311.0	NaN	0.0	NaN	1.0	NaN	NaN	2,002.0	NaN
NaN	4,486,365,176,018,953.0	NaN	257.0	NaN	1.0	NaN	NaN	2,010.0	NaN
NaN	5,108,957,434,464,472.0	NaN	516.5	NaN	1.0	NaN	NaN	2,013.0	NaN
NaN	5,585,237,469,514,469.0	NaN	756.0	NaN	2.0	NaN	NaN	2,017.0	NaN
NaN	6,997,197,066,610,978.0	NaN	999.0	NaN	3.0	NaN	NaN	2,020.0	NaN

'shape'

(6146, 13)

Figure 11: A screenshot of the description of the card's data set as well as its shape with the other hidden elements

The third cell in the Overview section focuses on summarizing the transaction data with codes meant to portray the information about the data, the description of the data as well as its shape.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 13305915 entries, 0 to 13305914
Data columns (total 13 columns):
 #   Column          Dtype
---  ------          -----
 0   id              int64
 1   date            object
 2   client_id       int64
 3   card_id         int64
 4   amount          object
 5   use_chip        object
 6   merchant_id     int64
 7   merchant_city   object
 8   merchant_state  object
 9   zip             float64
 10  mcc             int64
 11  errors          object
 12  status          int64
dtypes: float64(1), int64(6), object(6)
memory usage: 1.3+ GB

None
```

Figure 12: screenshot of the summary of the transaction data

'describe'

	id	date	client_id	card_id	amount	use_chip	merchant_id	merchant_city	merchant_state
count	13,305,915.0	13305915	13,305,915.0	13,305,915.0	13305915	13305915	13,305,915.0	13305915	11742215
unique	NaN	4136496	NaN	NaN	81161	3	NaN	12492	199
top	NaN	2011-06-09 12:46:00	NaN	NaN	$80.00	Swipe Transaction	NaN	ONLINE	CA
freq	NaN	18	NaN	NaN	132115	6967185	NaN	1563700	1427087
mean	15,584,024.565583952	NaN	1,026.812045845776	3,475.267650589982	NaN	NaN	47,723.763181337024	NaN	NaN
std	4,704,498.64944356	NaN	581.6385589646188	1,674.3559121347575	NaN	NaN	25,815.337690640663	NaN	NaN
min	7,475,327.0	NaN	0.0	0.0	NaN	NaN	1.0	NaN	NaN
25%	11,506,044.5	NaN	519.0	2,413.0	NaN	NaN	25,887.0	NaN	NaN
50%	15,570,866.0	NaN	1,070.0	3,584.0	NaN	NaN	45,926.0	NaN	NaN
75%	19,653,605.5	NaN	1,531.0	4,901.0	NaN	NaN	67,570.0	NaN	NaN
max	23,761,874.0	NaN	1,998.0	6,144.0	NaN	NaN	100,342.0	NaN	NaN

```
----------------------------------------------------------------
'shape'
(13305915, 13)
```

Figure 13: Description of the transaction data

'describe'

	card_id	amount	use_chip	merchant_id	merchant_city	merchant_state	zip	mcc	errors	status
	13,305,915.0	13305915	13305915	13,305,915.0	13305915	11742215	11,653,209.0	13,305,915.0	211393	13,305,915.0
	NaN	81161	3	NaN	12492	199	NaN	NaN	22	NaN
	NaN	$80.00	Swipe Transaction	NaN	ONLINE	CA	NaN	NaN	Insufficient Balance	NaN
	NaN	132115	6967185	NaN	1563700	1427087	NaN	NaN	130902	NaN
	7650589982	NaN	NaN	47,723.763181337024	NaN	NaN	51,327.81983117268	5,565.439814924415	NaN	0.0001458497364893735
	9121347575	NaN	NaN	25,815.337690640663	NaN	NaN	29,404.225233981277	875.7002376867902	NaN	0.01207699072977125
	0.0	NaN	NaN	1.0	NaN	NaN	1,001.0	1,711.0	NaN	0.0
	2,413.0	NaN	NaN	25,887.0	NaN	NaN	28,602.0	5,300.0	NaN	0.0
	3,584.0	NaN	NaN	45,926.0	NaN	NaN	47,670.0	5,499.0	NaN	0.0
	4,901.0	NaN	NaN	67,570.0	NaN	NaN	77,901.0	5,812.0	NaN	0.0
	6,144.0	NaN	NaN	100,342.0	NaN	NaN	99,928.0	9,402.0	NaN	1.0

'shape'

(13305915, 13)

Figure 14: Description of the transaction data with hidden columns shown

The forth cell of the Overview zooms into the status of the transaction data describes the different class weights as shown in the figure 15 below.

```
status
0   13,303,974.0
1        1,941.0
Name: count, dtype: float64
[5.00072948e-01 3.42759274e+03]
```

	id	date	client_id	card_id	amount	use_chip	merchant_id	merchant_city	merchant_state
count	400,000.0	400000	400,000.0	400,000.0	400000	400000	400,000.0	400000	329683
unique	NaN	195284	NaN	NaN	21113	3	NaN	5999	128
top	NaN	2018-04-02 07:07:00	NaN	NaN	$80.00	Swipe Transaction	NaN	ONLINE	CA
freq	NaN	137	NaN	NaN	7544	196821	NaN	70317	39672
mean	15,565,270.01664	NaN	1,009.1209625	3,448.8417025	NaN	NaN	47,580.32039	NaN	NaN
std	4,708,575.807268812	NaN	582.8274142275072	1,651.760749412136	NaN	NaN	25,937.14650250323	NaN	NaN
min	7,475,391.0	NaN	0.0	0.0	NaN	NaN	22.0	NaN	NaN
25%	11,490,618.0	NaN	490.0	2,400.0	NaN	NaN	26,489.75	NaN	NaN
50%	15,576,705.0	NaN	1,034.0	3,528.0	NaN	NaN	44,919.0	NaN	NaN
75%	19,586,041.0	NaN	1,519.0	4,808.0	NaN	NaN	68,671.0	NaN	NaN
max	23,761,809.0	NaN	1,998.0	6,138.0	NaN	NaN	100,340.0	NaN	NaN

Figure 15: summary of the status of the transaction data

```
status
0   13,303,974.0
1        1,941.0
Name: count, dtype: float64
[5.00072948e-01 3.42759274e+03]
```

card_id	amount	use_chip	merchant_id	merchant_city	merchant_state	zip	mcc	errors	status
400,000.0	400000	400000	400,000.0	400000	329683	326,972.0	400,000.0	101515	400,000.0
NaN	21113	3	NaN	5999	128	NaN	NaN	22	NaN
NaN	$80.00	Swipe Transaction	NaN	ONLINE	CA	NaN	NaN	Bad PIN,Insufficient Balance	NaN
NaN	7544	196821	NaN	70317	39672	NaN	NaN	30285	NaN
3,448.8417025	NaN	NaN	47,580.32039	NaN	NaN	51,125.58070721652	5,508.024905	NaN	0.5
I.760749412136	NaN	NaN	25,937.14650250323	NaN	NaN	29,059.544965369525	881.8249168532874	NaN	0.5000006250011719
0.0	NaN	NaN	22.0	NaN	NaN	1,012.0	1,711.0	NaN	0.0
2,400.0	NaN	NaN	26,489.75	NaN	NaN	28,613.0	4,900.0	NaN	0.0
3,528.0	NaN	NaN	44,919.0	NaN	NaN	47,362.0	5,499.0	NaN	0.5
4,808.0	NaN	NaN	68,671.0	NaN	NaN	77,493.0	5,812.0	NaN	1.0
6,138.0	NaN	NaN	100,340.0	NaN	NaN	99,925.0	9,402.0	NaN	1.0

Figure 16: Summary of the status of the transaction data with the hidden columns shown

The fifth cell in the Overview section focuses on outputting the value counts of the status of transaction data as floats data types while the sixth cell imports date time and redefines the variables represented by the columns of the transaction data as shown in the figure 17 below.

```
[107]
```

	0
id	0.0
date	0.0
client_id	0.0
card_id	0.0
amount	0.0
use_chip	0.0
merchant_id	0.0
merchant_city	0.0
merchant_state	70,317.0
zip	0.0
mcc	0.0
errors	0.0
status	0.0
hour	0.0
tran_type	0.0

dtype: float64

Figure 17: the screenshot of the data redefined

The last cell in the overview plays a crucial role in merging the three datasets including users, card, and transaction into a single table with the variables necessary for modeling. Therefore, the code in the cell drops unnecessary variables and outputs a new table with the relevant variables for modeling the machine learning algorithm as shown in figure 18 below.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 400000 entries, 0 to 399999
Data columns (total 20 columns):
 #   Column                Non-Null Count   Dtype
---  ------                --------------   -----
 0   date                  400000 non-null  datetime64[ns]
 1   amount                400000 non-null  float64
 2   use_chip              400000 non-null  object
 3   merchant_city         400000 non-null  object
 4   zip                   400000 non-null  object
 5   mcc                   400000 non-null  object
 6   errors                400000 non-null  object
 7   status                400000 non-null  object
 8   hour                  400000 non-null  int32
 9   tran_type             400000 non-null  object
 10  id_x                  400000 non-null  object
 11  per_capita_income     400000 non-null  object
 12  yearly_income         400000 non-null  object
 13  total_debt            400000 non-null  object
 14  gender                400000 non-null  object
 15  id_y                  400000 non-null  object
 16  card_brand            400000 non-null  object
 17  card_type             400000 non-null  object
 18  credit_limit          400000 non-null  object
 19  year_pin_last_changed 400000 non-null  int64
dtypes: datetime64[ns](1), float64(1), int32(1), int64(1), object(16)
memory usage: 59.5+ MB
```

Figure 18: summary of the new tables with the variables necessary for modeling the machine learning algorithm

EDA

The exploratory data analysis has four functional cells which are focused on understanding the new table that combines the user data, card data, and transaction data. The first cell reveals the distribution of errors in the table as shown in figure 19 below.

```
...   errors
      Complete                                                298,485.0
      Bad PIN,Insufficient Balance                             30,285.0
      Insufficient Balance,Technical Glitch                    25,022.0
      Bad Card Number,Insufficient Balance                      7,258.0
      Bad PIN,Technical Glitch                                  7,238.0
      Bad CVV,Insufficient Balance                              5,769.0
      Bad Expiration,Insufficient Balance                       4,815.0
      Bad Card Number,Bad CVV                                   3,824.0
      Bad Card Number,Bad Expiration                            3,473.0
      Bad Expiration,Bad CVV                                    3,360.0
      Insufficient Balance                                      3,053.0
      Bad Expiration,Technical Glitch                           2,173.0
      Bad Card Number,Technical Glitch                          1,503.0
      Bad CVV,Technical Glitch                                    849.0
      Bad Zipcode,Insufficient Balance                            764.0
      Technical Glitch                                            715.0
      Bad PIN                                                     497.0
      Bad Zipcode,Technical Glitch                                482.0
      Bad Card Number                                             126.0
      Bad CVV                                                     106.0
      Bad Card Number,Bad Expiration,Insufficient Balance         105.0
      Bad Expiration                                               85.0
      Bad Zipcode                                                  13.0
      Name: count, dtype: float64
```

Figure 19: Distribution of errors in the new table combining the user data, card data, and transaction data

The second cell in the section is responsible for showing the number of transactions that were withdrawals and those that were considered fraudulent. Moreover, the cell outputs the graphs for use chip, gender, transaction type, card brand, card type and status as shown in figure 20 below.

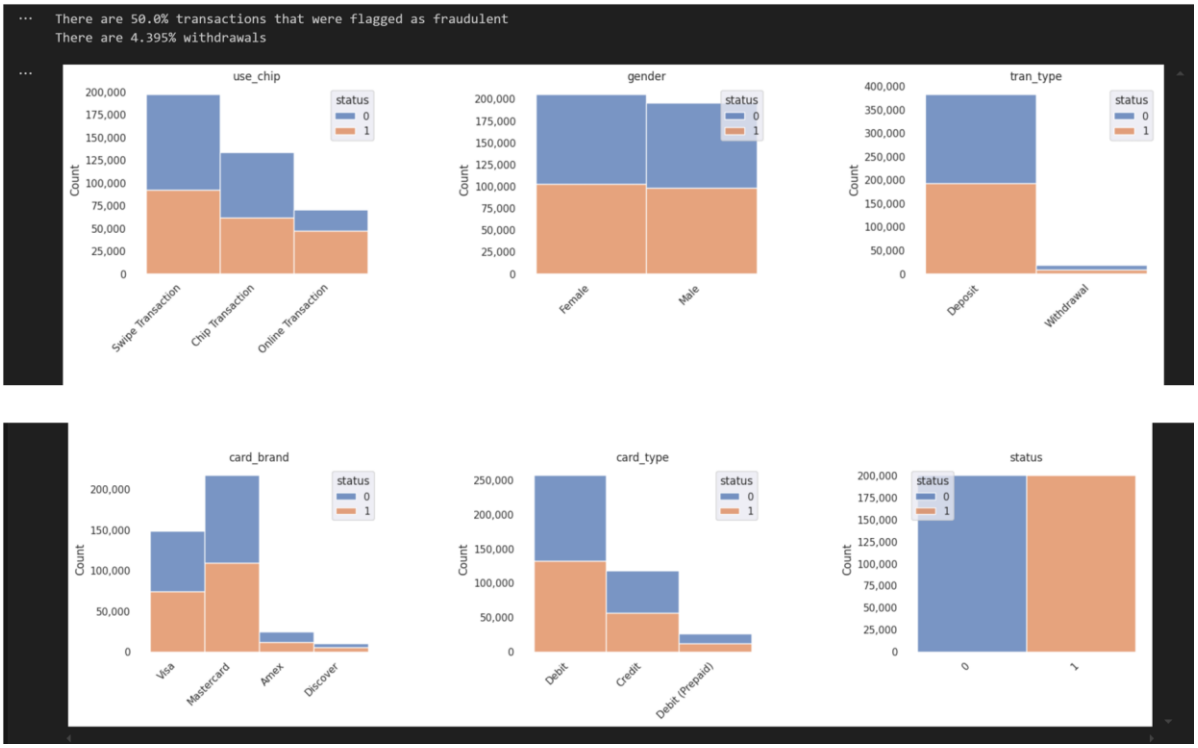


Figure 20: the percentage of transactions that are withdrawals and fraudulent transactions and the plot of various variables in the new table

The third cell in the section primarily outputs the different transaction types in decimals as shown by figure 21 below.

```
tran_type
Deposit      382,420.0
Withdrawal    17,580.0
Name: count, dtype: float64
```

Figure 21Transactoin types in the new table to be modelled

The last cell in the EDA outputs the last year the pins were changed and the accompanying value counts which reveals how many accounts las changed their pins and in which year as shown in figure 22 below.

```
year_pin_last_changed
2011   72,231.0
2010   67,694.0
2009   43,176.0
2012   38,319.0
2013   34,872.0
2014   28,496.0
2008   26,601.0
2015   24,650.0
2007   16,806.0
2016   13,978.0
2006    7,827.0
2018    6,842.0
2017    6,649.0
2019    3,507.0
2005    3,035.0
2020    2,341.0
2003    1,825.0
2004      976.0
2002      175.0
Name: count, dtype: float64
```

Figure 22: the year the pins were last changed and the associated value counts

Preprocessing

The section has five functional cells each tasked with preparing the data for modeling. The first cell in the section focuses on importing the various python tools from Sklearn. The cell is successful in downloading these tools as shown by the output as illustrated in figure 23 below.

```
Requirement already satisfied: category_encoders in /usr/local/lib/python3.10/dist-packages (2.6.4)
Requirement already satisfied: numpy>=1.14.0 in /usr/local/lib/python3.10/dist-packages (from category_encoders) (1.26.4)
Requirement already satisfied: scikit-learn>=0.20.0 in /usr/local/lib/python3.10/dist-packages (from category_encoders) (1.5.2)
Requirement already satisfied: scipy>=1.0.0 in /usr/local/lib/python3.10/dist-packages (from category_encoders) (1.13.1)
Requirement already satisfied: statsmodels>=0.9.0 in /usr/local/lib/python3.10/dist-packages (from category_encoders) (0.14.4)
Requirement already satisfied: pandas>=1.0.5 in /usr/local/lib/python3.10/dist-packages (from category_encoders) (2.2.2)
Requirement already satisfied: patsy>=0.5.1 in /usr/local/lib/python3.10/dist-packages (from category_encoders) (1.0.1)
Requirement already satisfied: python-dateutil>=2.8.2 in /usr/local/lib/python3.10/dist-packages (from pandas>=1.0.5->category_encoders) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.10/dist-packages (from pandas>=1.0.5->category_encoders) (2024.2)
Requirement already satisfied: tzdata>=2022.7 in /usr/local/lib/python3.10/dist-packages (from pandas>=1.0.5->category_encoders) (2024.2)
Requirement already satisfied: joblib>=1.2.0 in /usr/local/lib/python3.10/dist-packages (from scikit-learn>=0.20.0->category_encoders) (1.4.2)
Requirement already satisfied: threadpoolctl>=3.1.0 in /usr/local/lib/python3.10/dist-packages (from scikit-learn>=0.20.0->category_encoders) (3.5.0)
Requirement already satisfied: packaging>=21.3 in /usr/local/lib/python3.10/dist-packages (from statsmodels>=0.9.0->category_encoders) (24.2)
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.10/dist-packages (from python-dateutil>=2.8.2->pandas>=1.0.5->category_encoders) (1.1
```

Figure 23: successful acquisition of the tools necessary for modelling and processing the raw data

The second cell plays a crucial role in confirming the columns of the new transaction table that is to be modelled. The output of the cell is shown in figure 24 below.

```
Index(['date', 'amount', 'use_chip', 'merchant_city', 'zip', 'mcc', 'errors',
       'status', 'hour', 'tran_type', 'per_capita_income', 'yearly_income',
       'total_debt', 'gender', 'card_brand', 'card_type', 'credit_limit',
       'year_pin_last_changed'],
      dtype='object')
```

Figure 24: columns included in the new transaction table meant to be modelled

The third cell in the section leverages the loaded tools to encode the categorical variables iiin the table's column. Moreover, the fourth cell creates the necessary variables based on the categorical variables with high cardinality as shown in figure 25-28 below.

	amount	use_chip_Swipe Transaction	use_chip_Chip Transaction	use_chip_Online Transaction	status	hour	tran_type_Deposit	tran_type_Withdrawal
count	400,000.0	400,000.0	400,000.0	400,000.0	400000	400,000.0	400,000.0	400,000.0
unique	NaN	NaN	NaN	NaN	2	NaN	NaN	NaN
top	NaN	NaN	NaN	NaN	0	NaN	NaN	NaN
freq	NaN	NaN	NaN	NaN	200000	NaN	NaN	NaN
mean	50.179680675	0.4920525	0.3323625	0.175585	NaN	12.4491275	0.95605	0.04395
std	92.68159975294652	0.49993745817544744	0.4710607427284786	0.3804671729118206	NaN	5.094805872785223	0.20498415193809794	0.20498415193809794
min	-500.0	0.0	0.0	0.0	NaN	0.0	0.0	0.0
25%	11.53	0.0	0.0	0.0	NaN	9.0	1.0	0.0
50%	36.72	0.0	0.0	0.0	NaN	12.0	1.0	0.0
75%	76.15	1.0	1.0	0.0	NaN	16.0	1.0	0.0
max	3,363.29	1.0	1.0	1.0	NaN	23.0	1.0	1.0

11 rows × 27 columns

Figure 25: categorical variables with high cardinality (first eight variables)

tran_type_Deposit	tran_type_Withdrawal	per_capita_income	yearly_income	...	card_type_Debit	card_type_Credit	card_type_Debit (Prepaid)
400,000.0	400,000.0	400,000.0	400,000.0	...	400,000.0	400,000.0	400,000.0
NaN	NaN	NaN	NaN	...	NaN	NaN	NaN
NaN	NaN	NaN	NaN	...	NaN	NaN	NaN
NaN	NaN	NaN	NaN	...	NaN	NaN	NaN
0.95605	0.04395	23,947.2154525	46,581.5696675	...	0.6415775	0.293355	0.0650675
0.20498415193809794	0.20498415193809794	12,396.664293165726	24,373.37632984134	...	0.47953767983715045	0.45530029894618396	0.24664523617691209
0.0	0.0	0.0	1.0	...	0.0	0.0	0.0
1.0	0.0	17,179.0	33,076.0	...	0.0	0.0	0.0
1.0	0.0	21,033.0	40,810.0	...	1.0	0.0	0.0
1.0	0.0	27,057.0	53,872.0	...	1.0	1.0	0.0
1.0	1.0	163,145.0	280,199.0	...	1.0	1.0	1.0

Figure 26: categorical variables with high cardinality (next eight variables)

card_type_Debit (Prepaid)	credit_limit	year_pin_last_changed	cityCounts	zips	mccs	errorCounts
400,000.0	400,000.0	400,000.0	400,000.0	400,000.0	400,000.0	400,000.0
NaN	NaN	NaN	NaN	NaN	NaN	NaN
NaN	NaN	NaN	NaN	NaN	NaN	NaN
NaN	NaN	NaN	NaN	NaN	NaN	NaN
0.0650675	15,742.619415	2,011.2938225	12,855.13981	13,527.64852	22,957.30919	227,136.815915
0.24664523617691209	12,260.398964786651	2.9003123742121475	26,546.112277208547	28,120.34269859173	14,035.930710088953	122,487.3643486277
0.0	0.0	2,002.0	1.0	1.0	1.0	13.0
0.0	8,366.0	2,009.0	187.0	112.0	13,227.0	30,285.0
0.0	13,823.0	2,011.0	473.0	238.0	23,726.0	298,485.0
0.0	20,886.0	2,013.0	1,596.0	555.0	39,251.0	298,485.0
1.0	141,391.0	2,020.0	70,317.0	73,028.0	41,699.0	298,485.0

Figure 27: categorical variables with high cardinality (next eight variables)

card_type_Debit (Prepaid)	credit_limit	year_pin_last_changed	cityCounts	zips	mccs	errorCounts	city
400,000.0	400,000.0	400,000.0	400,000.0	400,000.0	400,000.0	400,000.0	400,000.0
NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
0.0650675	15,742.619415	2,011.2938225	12,855.13981	13,527.64852	22,957.30919	227,136.815915	0.1757925
4664523617691209	12,260.398964786651	2.9003123742121475	26,546.112277208547	28,120.34269859173	14,035.930710088953	122,487.3643486277	0.3806440058215598
0.0	0.0	2,002.0	1.0	1.0	1.0	13.0	0.0
0.0	8,366.0	2,009.0	187.0	112.0	13,227.0	30,285.0	0.0
0.0	13,823.0	2,011.0	473.0	238.0	23,726.0	298,485.0	0.0
0.0	20,886.0	2,013.0	1,596.0	555.0	39,251.0	298,485.0	0.0
1.0	141,391.0	2,020.0	70,317.0	73,028.0	41,699.0	298,485.0	1.0

Figure 28: categorical variables with high cardinality (last variables)

The last cell encodes the target variable while training, testing, and splitting the raw data. The final cell does not have any results as it just processes the data for modeling.

Feature Selection and Modeling

The feature selection and modeling section is the most important section in the machine learning model as ti ios responsible for training the algorithm based on the provided data. The section has six cells each tasked with a different function. The first cell in the section involves importing the tools for feature selection and modelling from python luibraries particularly sklearn and xgboost. The second cell is responsible for instantiating the models GaussianNB, Logistic Regression, and XGBClassifier. Next, the third cell has

the code responsible for building the pipelines for the different models. After the pipelines for the different models have been developed, the next cell focused on training the models based on the data. The last model in the section focused on predicting various values based on the models. The section did not have any outputs as the tools focused on the models.

Model Validation

The model validation section has nine cells which reviews the performance of the model in detecting fraud. In this section, the functions of the machine learning in fraud detection becomes apparent unlike the previous section where the operations of the models could not be directly witnessed.

Balanced Accuracy Score

The first three cell of the model validation section imports metrics from sklearn and leverages them to determine the balanced accuracy score of the GaussianNB, XGBClassifier, and Logistic regression model which is shown in the figure 29-31 below.

Figure 29: Balance accuracy score of the GaussianNB model

Figure 30: Balanced accuracy score of XGBClassifier model

Figure 31: Balanced accuracy score of Logistic regression model

Classification Report

The fourth cell of the model validation measures the performance of the models through a classification report imported from sklearn. The classification reports outlines the

precision, recall and f1 scores of the GaussianNB, XGBClassifier, and Logistic Regresion models as shown in figure 32-34

	precision	recall	f1-score	support
class 0	0.66	0.86	0.74	50000
class 1	0.79	0.55	0.65	50000
accuracy			0.70	100000
macro avg	0.72	0.70	0.70	100000
weighted avg	0.72	0.70	0.70	100000

Figure 32: : the classification report of the GaussianNB model

	precision	recall	f1-score	support
class 0	1.00	0.77	0.87	50000
class 1	0.82	1.00	0.90	50000
accuracy			0.89	100000
macro avg	0.91	0.89	0.89	100000
weighted avg	0.91	0.89	0.89	100000

Figure 33: Classification report for the XGBClassifier model

	precision	recall	f1-score	support
class 0	0.66	0.98	0.79	50000
class 1	0.97	0.49	0.65	50000
accuracy			0.74	100000
macro avg	0.81	0.74	0.72	100000
weighted avg	0.81	0.74	0.72	100000

Figure 34: Classification report for the Logistic Regression model

Confusion Matrix

The model validation also utilized another metric known as confusion matrix to review the performance of the machine learning models. The last three cells in the Model validation sector outputs the confusion matrix values and plots a heat map with the predicted value on the x-axis and the true value on the y-axis for GaussianNB model, XGBClassifier, and Logistic Regression models respectively in figure 35-37.

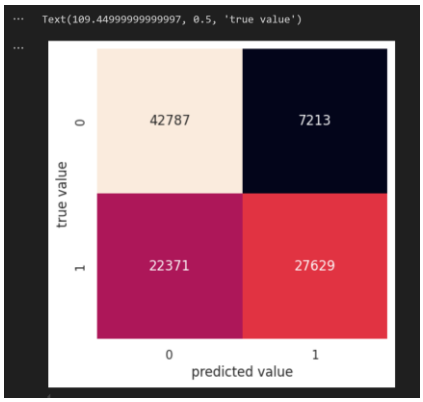


Figure 35: Confusion matrix and heat map for GaussianNB model

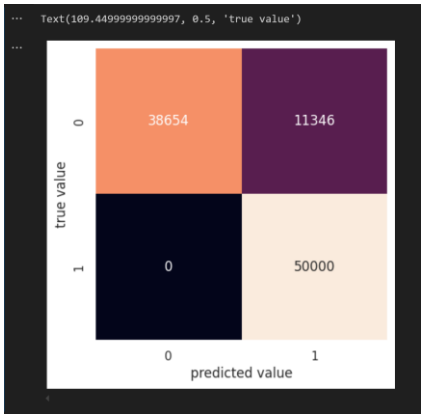


Figure 36: Confusion matrix and heat map for the XGBClassifier model

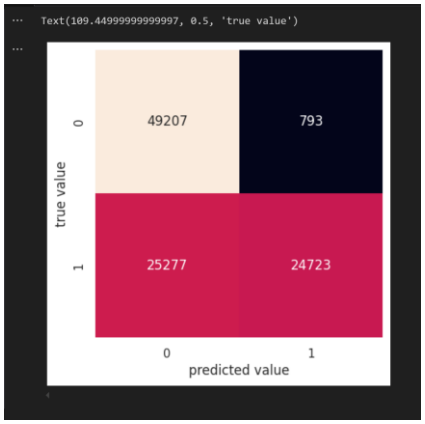


Figure 37: Confusion matrix and heat map for Logistic Regression model

Discussion

The research outlines the various processes involved in the utilization of machine learning models to identify anomalies during transactions. In the initial stages, the raw data acquired is explored and its attributes established to determine which variables are necessary for identifying fraud. Further, in the initial stages, the data is parsed and processed to eliminate meaningless data consequently making it easier for the machine learning models to interact with. The experiment was also crucial in showing the various machine learning models in python such as sklearn. The machine learning model illustrated how different machine learning models interact with the same dataset consequently illustrating the best machine learning model a business should adopt to effectively identify fraud and other anomalies with tis transactions.

The various metrics highlights the different strengths and weaknesses of the Logistic regression model, XGBClassifier, and GaussianNB models. For instance, the classification report for the different models reveal that the GuassianNB has an accuracy of 70%, XGBClassifier has an accuracy of 89% while logistic regression model has an accuracy of 74%. The classification reports therefore illustrates that the XGBclassifier is the best machine learning model compared to Logistic Regression models and GausssianNB models. According to West and Bhattacharya (2016), machine learning methods are categorized into statistical and computational methods depending on the underlying process through which they interact with data. GaussianNB and Logistical Regression are statistical machine learning methods because they are underpinned by traditional mathematical methods. On the other hand, computational methods are highly dynamic and adaptive and are based on programming. XGBoost and neural networks are some of the computational machine learning methods. Computational machine learning methods are considered to be better than statistical methods

due to their dynamics. The research aligns with this conclusion as XGB classifier has a considerably higher accuracy compared to other statistical methods like GaussianNB and Logistical Regression models.

Conclusion

In conclusion, the research has illustrated the role of machine learning models in fraud detection and prevention by exploring the precision, recall, and F1 score of three distinct machine learning models including GaussianNB, Logistical Regression, and XGBClassifier. In the research's context, precision determines the cost associated to the model predicting a transaction as fraudulent when it is legitimate. In banking or other financial settings, the cost involves causing unnecessary delays by adding another layer of validation. Consequently, it is essential to ensure that the model has a high precision to prevent delays in transaction processing. Another significant metric in comparing the different model is the recall. In the current research, recall measure cost associated to the model's failure to identify fraudulent transactions. In the business landscape, a model that fails to identify fraud exposes the business to risk of losing significant amounts of money and damaging its brand equity and reputation. Recall essentially establishes the usefulness of the model as it determines the model's ability to identify fraudulent transactions. Based on the classification report of the GaussianNB, XGBClassifier, and Logistical Regression, XGBClassifier has the most balanced recall compared to the other two. The final parameter through which the machine learning models are compared is the F1 score. The F1 score determines the model's capacity to address the imbalance in the data effectively. Based on these factors, XGBClassifieris the most effective model as it has the highest F1 score.

Proposed Solution

The research has revealed that the XGBClassifier has a higher accuracy, recall, and precision compared to GaussianNB and Logistical Regression Models hence it is crucial for WeGoWin to adopt it as in its fraud detection endeavors. WeGoWin is likely to save cost considerably by eliminating delays during transactions due to extra validation as XGBClassifier has a high precision. Similarly, XGBClassifier has a high recall hence it will help WeGoWin identify any fraudulent transactions consequently creating a culture of accountability and integrity in the organization.

References

Mohammed, M. A., Kothapalli, K. R. V., Mohammed, R., Pasam, P., Sachani, D. K., & Richardson, N. (2017). Machine Learning-Based Real-Time Fraud Detection in Financial Transactions. *Asian Accounting and Auditing Advancement*, *8*(1), 67-76.

Sadineni, P. K. (2020, October). Detection of fraudulent transactions in credit card using machine learning algorithms. In *2020 Fourth International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud)(I-SMAC)* (pp. 659-660). IEEE.

Megdad, M. M., Abu-Naser, S. S., & Abu-Nasser, B. S. (2022). Fraudulent financial transactions detection using machine learning.

AbdulSattar, K., & Hammad, M. (2020, December). Fraudulent transaction detection in FinTech using machine learning algorithms. In *2020 International Conference on Innovation and Intelligence for Informatics, Computing and Technologies (3ICT)* (pp. 1-6). IEEE.

Ali, A., Abd Razak, S., Othman, S. H., Eisa, T. A. E., Al-Dhaqm, A., Nasser, M., ... & Saif, A. (2022). Financial fraud detection based on machine learning: a systematic literature review. *Applied Sciences*, *12*(19), 9637.

Raghavan, P., & El Gayar, N. (2019, December). Fraud detection using machine learning and deep learning. In *2019 international conference on computational intelligence and knowledge economy (ICCIKE)* (pp. 334-339). IEEE.

West, J., & Bhattacharya, M. (2016). Intelligent financial fraud detection: a comprehensive review. *Computers & security*, *57*, 47-66.

Karpoff, J. M. (2021). The future of financial fraud. *Journal of Corporate Finance*, *66*, 101694.

Greenfield, A. (2017). *Radical technologies: The design of everyday life*. Verso Books.

Saunders, M., & Lewis, P. (2017). *Doing research in business and management*. Pearson.