Credit Card Fraud Detection Using Machine Learning in the Cloud

Balazs Barcza
x19190638

Christoph Kratz
x21111898

Wislan Alandes De Lima
x21126151

Abstract—**Credit card fraud has been a problem for businesses and financial institutions for decades, resulting, in recent years, in billions of dollars in losses on a yearly basis. To take on the large amount of data generated around financial transactions, large computing resources will be required. Additionally, to review large numbers of transactions in an efficient and timely manner, human review would not be suitable. Therefore, to address these challenges machine learning in the cloud seems to be the solution. With this project, we cover many aspects of fraudulent transactions, as well as a model based on supervised learning techniques such as Decision Tree (DT), and Logistic Regression (LR). It makes use of the Simulated Credit Card Transactions generated using Sparkov. It simulates the transactions of 1000 customers doing transactions with a pool of 800 merchants that was run from the duration 1st Jan 2019 to 31st Dec 2020. The purpose of this study is to predict the likelihood of transactions being fraudulent using machine learning models and deploy it to the cloud. The findings show that Decision Tree Model achieves the best recall and accuracy scores (94%).**

Keywords—credit card, fraud, cloud computing, machine learning, Amazon Web Services (AWS)

I. Introduction

For decades, credit card fraud has been an ongoing problem for businesses and financial institutions who have been losing up to several billion dollars on a yearly basis. Building robust defenses to mitigate this problem is paramount as otherwise business will not only lose large sums of money, but also get a negative brand reputation due to the traumatic experiences their customers might be facing. Indeed, some victims have a feeling of powerlessness as all they can do is to watch their money exit their account. Therefore, for some, it might be difficult to have trust in a brand again after such an incident. The recent Covid-19 pandemic has only contributed to increasing this issue. Indeed, more financial transactions than ever must be processed by credit card either because cash payments are not accepted due to fear of virus contamination or business had to shift to an online model to be able to get at least some revenue during periods of lockdown. As a result, the amount of data surrounding credit card transactions has reached an all-time high, which is impossible to be reviewed for fraud solely by humans or by limited computing resources. To address these issues, we will have to rely on machine learning (ML), which will help us analyze enormous amounts of data in an accurate and efficient manner, as well as cloud computing which will help us get the required computing resources to evaluate this data in a timely manner. In this project we will create a classification model that accurately detects fraudulent credit card transactions with minimal impact on genuine transactions and demonstrate that such models can be deployed into a cloud environment with limited computing resources.

II. Methodology

A. Dataset

This project is based on a dataset published by Kartik Shenoy on Kaggle in 2020. It "is a simulated credit card transaction dataset containing legitimate and fraud transactions from the duration 1st Jan 2019 - 31st Dec 2020. It covers credit cards of 1000 customers doing transactions with a pool of 800 merchants." [1]

B. Data Cleaning, Exploratoruy Data Analysis, and Model Creation

For our pre-processing tasks we leveraged Google Colab, a cloud-based tool that fosters easier collaboration. In our Colab Notebook we used different Python libraries to initially clean the data (Pandas), then proceed with visually exploring the data (Matplotlib, Seaborn), and finally creating our ML model (Scikit-learn).

C. Design

For this project our initial plan was to develop the architecture shown in Figure 1, which would leverage more cloud technologies. However, due to time constraints and access limitations in our AWS accounts provided by the National College of Ireland we decided to implement the architecture shown in Figure 2.

1) Complete Architecture

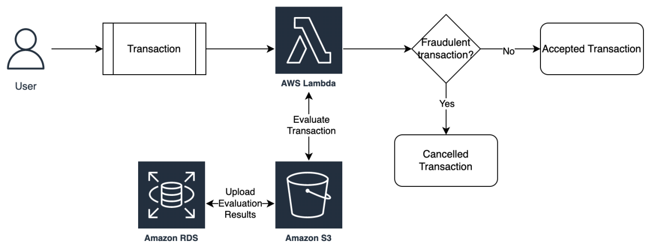


Figure 1: Complete Architecture

In this architecture, the ML model would be saved to a S3 bucket. Whenever a credit card transaction is created, a Lambda function would be triggered to evaluate if said transaction is fraudulent or not. Once evaluated, these results would be uploaded to a relational database, which could be used to monitor the model's performance over time, and the transaction would either get accepted or declined depending on the outcome.

The benefit of this architecture over the simplified one is, firstly, that the model is stored separately from user traffic which increases security. Secondly, the Lambda function allows for scalability as we do not have to worry about provisioning enough computing resources and finally, the relational database allows us to store data about the transactions we received and continuously monitor whether our model is continuing to accurately identify fraudulent transactions or whether it needs to get retrained.

2) *Simplified Architecture*

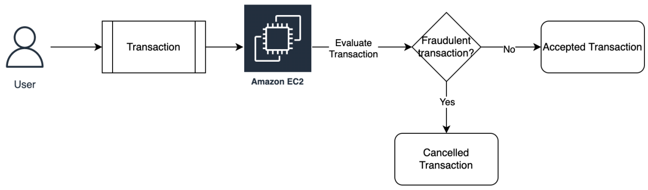


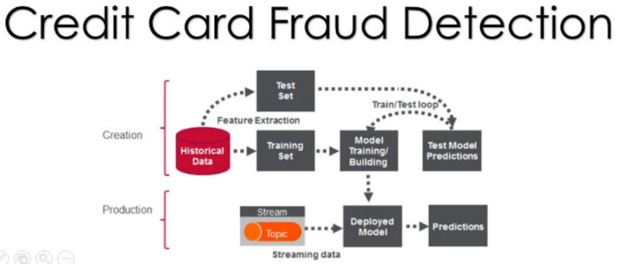
Figure 2: Simple Architecture

In this simplified architecture, a ML model is deployed to an EC2 instance which will take data for the model's features as input and provide the prediction whether a transaction is fraudulent or not as output. While such an architecture is easy to implement, it is not desirable to deploy it to production for the following reasons:

- **Lack of scalability:** using only one EC2 instance provides only a finite amount of resources. This might be sufficient for a small business launching a new product, but as demand will increase more computing resources will need to be provisioned.
- **Weak security:** having the model stored on the same instance which is going to receive user traffic is not a good idea as, in case of security breach, bad actors could get a hold of the model and try to reverse engineer it, resulting, potentially, in large losses to the business.

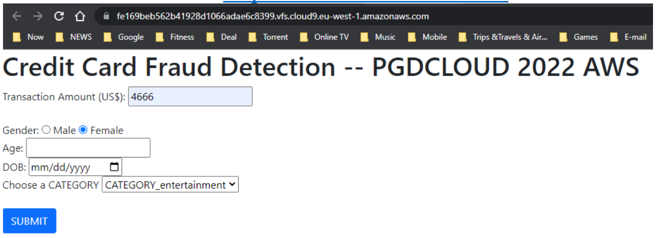
III. Evaluation

One of the hottest segments of the software business is being driven by the increasing popularity of machine learning technologies. Machine learning algorithms are used by data scientists to harvest information and insights from a given data collection. Applications and business analytics systems can use this information to inform decision-making. Nowadays, a variety of industrial applications use machine learning tools created on cloud infrastructures. Many different industries fall under this category, including e-commerce, retail, cybersecurity, life sciences, gaming, technical diagnostics, healthcare, information protection, entertainment, education, digital signal processing, manufacturing, telecommunications, logistics, government, financial services, and many more. We have to ask a question: What are the Benefits of Machine Learning in the Cloud? Google, Amazon, and Microsoft have all made considerable investments in artificial intelligence (AI) and machine learning over the past several years, from the introduction of new services to the implementation of significant organizational reorganizations that strategically incorporate AI. The benefits are Thanks to the cloud, businesses can quickly experiment with machine learning techniques and scale up when projects are placed into production and demand increases. The cloud's pay-per-use model is good for machine learning workloads. You do not need to use a cloud service provider to build a machine learning solution. After all, organizations may run a variety of open-source machine learning frameworks, including TensorFlow, on their hardware. However, as real-world model training frequently necessitates enormous compute clusters, firms building robust machine learning models internally are likely to experience issues with workload scalability. All three of the major cloud providers have their machine learning solutions: Google Vertex AI, Amazon SageMaker, and Azure Machine Learning. Along with various cutting-edge data analytics capabilities, the cloud offers BigQuery a multi-cloud data warehouse that is serverless, highly scalable, and affordable for businesses.



In our project, we have used a couple of cloud services. It was the original plan we wanted to use the Amazon SageMaker. "Amazon SageMaker is a fully-managed service that enables data scientists and developers to quickly and easily build, train, and deploy machine learning models at any scale. Amazon SageMaker includes modules that can be used together or independently to build, train, and deploy your machine learning models." [3] Unfortunately, We could not build our project with SageMaker because We have used the education Amazon Web Services. We did not get access to the server. We wanted to create our project with Amazon Lambda, but we had the same problem. "AWS Lambda is an event-driven, serverless computing platform provided by Amazon as a part of Amazon Web Services. It is a computing service that runs code in response to events and automatically manages the computing resources required by that code."[4] So We have chosen another way to deploy our ML model in the cloud We have used the Amazon EC2. "Amazon Elastic Compute Cloud (Amazon EC2) provides scalable computing capacity in the Amazon Web Services (AWS) Cloud."[5] We have created a simple Flask web application and we have run this web service on the EC2. The project is available in port=8080.

Link to the website: http://18.202.218.32:8080/



Credit Card Fraud Detection -- PGDCLOUD 2022 AWS

Transaction Amount (US$): 4666

Gender: ○ Male ● Female

Age:

DOB: mm/dd/yyyy

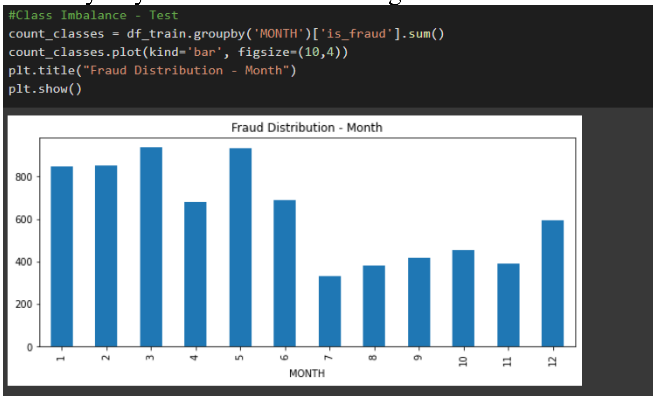
Choose a CATEGORY CATEGORY_entertainment

We have developed our web servers with Cloud9. "AWS Cloud9 is a cloud-based integrated development environment (IDE) that lets you write, run, and debug your code with just a browser. It includes a code editor, debugger, and terminal."[6] This IDE helped to use collocated when we created the HTML and py files. We have used GitHub version control. It helped us work together on the project.

GitHub link: https://github.com/Balays33/Credit-Card-Fraud-Detection-PGDCLOUD-2022-AWS.git

In our project We have use Google Colab. Colab is particularly well suited to machine learning, data analysis, and education. It enables anyone to create and execute arbitrary Python code through the browser. Google account is necessary to use Google Colab, a service offered by Google Research. It enables the free use of Python notebooks on Google Cloud.

Visualization Project link: https://colab.research.google.com/drive/1CguQcdI4lzQKXIt5dUzRepmYx9PQJWtB?usp=sharing
If you visit this link, you can see how we have prepared the data.
We have cleaned the data, removed unwanted data, missing values, rows and columns, duplicate values, data type conversion. We have restructured the dataset and changed the rows and columns or index of rows and columns. We have visualized the data to understand how it is structured and understand the relationship between various variables and classes present. The team spit the cleaned data into two sets / training and test data. The training set is the set your model learns from. A testing set is used to check the accuracy of your model after training.



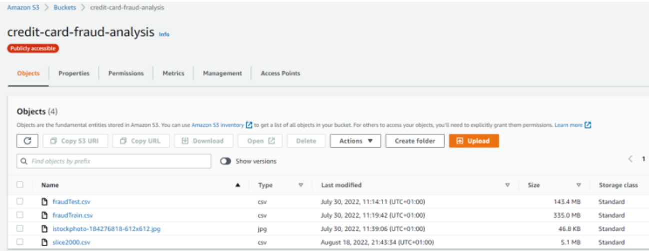
```
#Class Imbalance - Test
count_classes = df_train.groupby('MONTH')['is_fraud'].sum()
count_classes.plot(kind='bar', figsize=(10,4))
plt.title("Fraud Distribution - Month")
plt.show()
```

Google Colab offers a free GPU, albeit the model may change depending on the resources available. Nvidia K80s, T4s, P4s, or P100s are frequently among the models of provided GPUs.
We have uploaded our dataset to the Amazon S3 bucket, and We have connected the G Colab with the data. After the connection, We can do the data mining and data transformation and cleaning and create a model.

```
#train = pd.read_csv('fraudTrain.csv')
train=pd.read_csv('https://credit-card-fraud-analysis.s3.eu-west-1.amazonaws.com/fraudTrain.csv')

#test = pd.read_csv('fraudTest.csv')
test = pd.read_csv('https://credit-card-fraud-analysis.s3.eu-west-1.amazonaws.com/fraudTest.csv')
```

This code shows how we called the data from the Amazon S3



Amazon S3 > Buckets > credit-card-fraud-analysis

credit-card-fraud-analysis Info

Publicly accessible

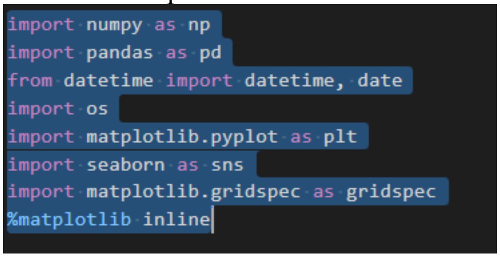
Objects (4)

Objects are the fundamental entities stored in Amazon S3. You can use Amazon S3 inventory to get a list of all objects in your bucket. For others to access your objects, you'll need to explicitly grant them permissions. Learn more

□	Name	Type	Last modified	Size	Storage class
□	fraudTest.csv	csv	July 30, 2022, 11:14:11 (UTC+01:00)	143.4 MB	Standard
□	fraudTrain.csv	csv	July 30, 2022, 11:19:42 (UTC+01:00)	335.0 MB	Standard
□	istockphoto-184276818-612x612.jpg	jpg	July 30, 2022, 11:39:06 (UTC+01:00)	46.8 KB	Standard
□	slice2000.csv	csv	August 18, 2022, 21:43:34 (UTC+01:00)	5.1 MB	Standard

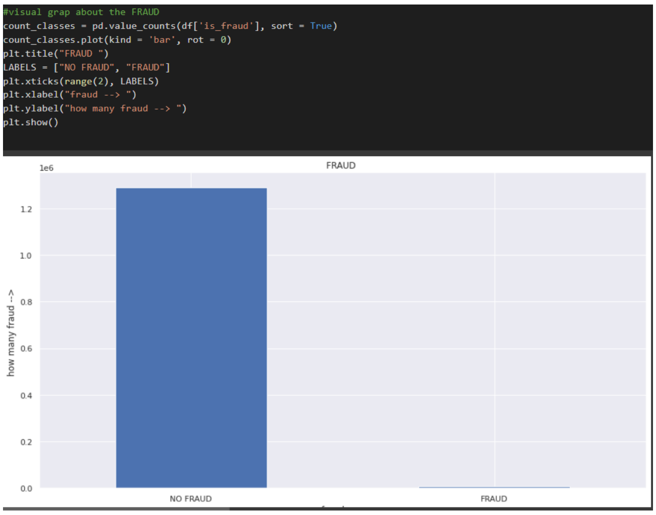
S3 is public to access to the Train and Test data

We have import Machine learning liberates. Multidimensional arrays and matrices are included in the NumPy, along with a substantial number of advanced mathematical operations that can be used on these arrays.



```
import numpy as np
import pandas as pd
from datetime import datetime, date
import os
import matplotlib.pyplot as plt
import seaborn as sns
import matplotlib.gridspec as gridspec
%matplotlib inline
```

We see we have NO Fraud : 1289169 rows and Fraud : 7506 rows so our dataset is unbalance

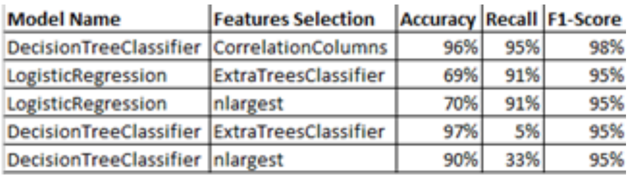


```
#visual grap about the FRAUD
count_classes = pd.value_counts(df['is_fraud'], sort = True)
count_classes.plot(kind = 'bar', rot = 0)
plt.title("FRAUD ")
LABELS = ["NO FRAUD", "FRAUD"]
plt.xticks(range(2), LABELS)
plt.xlabel("fraud --> ")
plt.ylabel("how many fraud --> ")
plt.show()
```

Using a histogram, we can see how the data is distributed. Having data that follows a normal distribution is necessary for some of the statistical techniques used to detect outliers. If the data doesn’t follow a normal distribution, the z-score calculation shouldn’t be used to find the outliers

The model depends on the classification of previous fraudulent transaction in order to predict if the future transaction can potentially be positive. In this case, the classification problem requires a machine learning that can classify the fraudulent transaction and predict the outcome. In this project, we have used Decision Tree and Logistic Regression to create a machine learning model. Table 1 compare the models using different techniques to select the relevant features from the dataset.

Model Accuracy. Table 1 shows the two different models' classification results



Model Name	Features Selection	Accuracy	Recall	F1-Score
DecisionTreeClassifier	CorrelationColumns	96%	95%	98%
LogisticRegression	ExtraTreesClassifier	69%	91%	95%
LogisticRegression	nlargest	70%	91%	95%
DecisionTreeClassifier	ExtraTreesClassifier	97%	5%	95%
DecisionTreeClassifier	nlargest	90%	33%	95%

It clearly shows that Decision tree is the best evaluation among the models. It has higher accuracy and recall rate.

In real world scenario, the finance institution would have access to the customer transaction. This would allow the institution to create models based on the user's behaviour and improve the user's experience and provide better products. Although, this can create advantages to the institution, it generates challenges when dealing with sensitive and classified information. For example, a machine learning model that does not consider the privacy and ethics during the development stage, can cause discrimination bias when dealing with sensitive attributes such as in a delicate environment for the identification of Image based Sexual Abuse cases [7]. Based on that, we had to transform and remove sensitive attributes and any other type of social economic prejudice from the ML algorithm to avoid a result that could lead to damage to the integrity of the individual.

It is also important to remember that Machine Learning Model can produce damage to society if a proper ethics guideline is not followed and adhered to. According to a real-world fraud detection case in The Netherlands that wrongly accused around 26.000 parents of making fraudulent benefit claims for day-care [8], which was based on a ML model that was predicting bad outcomes due to bad machine learning design.

IV. Conclusion and Future Work

The K-Nearest Neighbor algorithms, Support Vector Machines (SVM), and Naive Bayes (NB) techniques are the most popular methods for fraud detection (KNN). To build classifiers using ensemble or meta-learning techniques, these techniques can be employed separately or in combination. Even though there are now a number of fraud detection techniques available, none of them were sufficient to uncover the fraud as it was occurring. They learned about past frauds that had taken place. The disadvantage of all current approaches is that they can only consistently deliver results when applied to a particular dataset, occasionally with particular unique traits.

With this project we were able to demonstrate that it is possible to deploy simple ML models, which can accurately detect fraudulent credit card transactions, into a cloud environment with limited computing resources. Had we had more time and access in our AWS account to work on this project, it would have been interesting to build a more scalable solution and compare performance metrics (e.g. CPU usage...) between both solutions.

There are also some improvement opportunities that could be explored:

- **Monitoring:** firstly for the cloud resources to ensure that they are running as expected and, secondly, on the models performance to ensure that it continues to accurately detect fraudulent transactions.
- **Security:** some of the datapoints in the dataset contain sensitive information (e.g. credit card number, names, address), to protect these from data breaches, more consideration towards encryption and hashing should be placed.
- **Ethics:** more consideration would have to be put on how data is being handled and for how long it is being retained for to prevent the model from having any bias towards specific populations for example and to ensure that data protection considerations are respected.

V. Contributions

	Balazs	Christoph	Wislan
Discovery	33	33	33
Design	20	20	60
Data Analysis	33	33	33
Development	40	40	20
Report	33	33	33

References

[1] K. Shenoy, Credit Card Transactions Fraud Detection Dataset, Kaggle, 2021, https://www.kaggle.com/datasets/kartik2112/fraud-detection

[2] A. Shen, R. Tong and Y. Deng, "Application of Classification Models on Credit Card Fraud Detection," 2007 International Conference on Service Systems and Service Management, 2007, pp. 1-4, doi: 10.1109/ICSSSM.2007.4280163.

[3] https://aws.amazon.com/about-aws/whats-new/2017/11/introducing-amazon-sagemaker/

[4] https://aws.amazon.com/lambda/

[5] https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/concepts.html

[6] https://aws.amazon.com/cloud9/

[7] A. Eusebi, M. Vasek, E. Cockbain, and E. Mariconti, "The Ethics of Going Deep: Challenges in Machine Learning for Sensitive Security Domains," in *2022 IEEE E. Symposium on Sec. and Privacy Workshops (EuroS&PW)*, Genoa, Italy, Jun 22, 2022, pp. 533-537. doi: 10.1109/ICAIS53314.2022.9743014

[8] X. V. Bruxvoort, and M V. Keulen, "Framework for Assessing Ethical Aspects of Algorithms and Their Encompassing Socio-Technical System," *MDPI AG.*, vol. 11, no. 11187, pp. 11187, Nov. 2021. doi: 10.3390/app112311187