College Dublin
Computing • IT • Business

Detecting Credit Card Fraud with Machine Learning

Stefany Alves- 2019307

Introduction

The Credit Card Fraud Detection Project is a data-driven initiative that leverages the power of machine learning algorithms to tackle the critical issue of credit card fraud. With the increasing prevalence of online transactions and digital payments, credit card fraud has become a significant concern for individuals, businesses, and financial institutions alike. Fraudulent activities not only cause financial losses but also erode trust and confidence in electronic payment systems. The Credit Card Fraud Detection Project aims to develop a machine learning algorithm that can effectively identify fraudulent transactions from legitimate ones.

In the world of machine learning, credit card fraud detection is approached as a binary classification problem. The goal is to accurately classify transactions into two categories: legitimate or fraudulent. This classification is based on various features and attributes associated with each transaction, such as transaction amount, location, time of day, and user behavior. These features act as crucial indicators, helping machine learning models distinguish between genuine and fraudulent activities.

Machine learning algorithms play a pivotal role in fraud detection, as they continuously learn from historical transaction data. Models like Logistic Regression, Random Forest, and Support Vector Machines (SVM) are commonly used to process and analyze vast amounts of data, allowing them to identify patterns and anomalies associated with fraudulent transactions. With each new transaction, the model refines its understanding, adapting to emerging fraud patterns and staying one step ahead of fraudsters. (Science, 2022)

One of the key challenges in credit card fraud detection is dealing with imbalanced data. Fraudulent transactions are relatively rare compared to legitimate ones, resulting in an imbalanced dataset. In such cases, the minority class (fraudulent transactions) is significantly outnumbered by the majority class (legitimate transactions). To address this issue, various techniques are employed, such as oversampling the minority class, undersampling the majority class, or using synthetic data generation methods like SMOTE (Synthetic Minority Over-sampling Technique). These approaches enable the model to learn from a more balanced dataset, enhancing its ability to accurately detect fraudulent transactions.

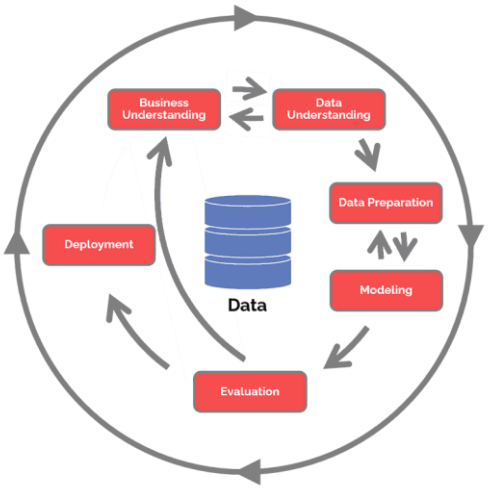
The success of credit card fraud detection with machine learning hinges on feature engineering. Extracting meaningful features from transaction data is critical to capturing the subtle patterns indicative of fraudulent activities. For instance, analyzing the time of day when transactions occur can reveal abnormal behaviors, such as transactions made at odd hours. Combining multiple features, such as transaction frequency and geographical location, can provide deeper insights into user behavior and transaction patterns, further improving fraud detection accuracy.

Detecting credit card fraud with machine learning represents a crucial advancement in the fight against financial crime. By leveraging the capabilities of machine learning algorithms, financial institutions can build proactive and efficient fraud detection systems that safeguard the interests of their customers and protect the integrity of electronic payment systems. As machine learning techniques continue to advance, the battle against credit card fraud is likely to shift in favor of the defenders, empowering consumers and institutions to transact with confidence and security.

Data and Features

Methodology

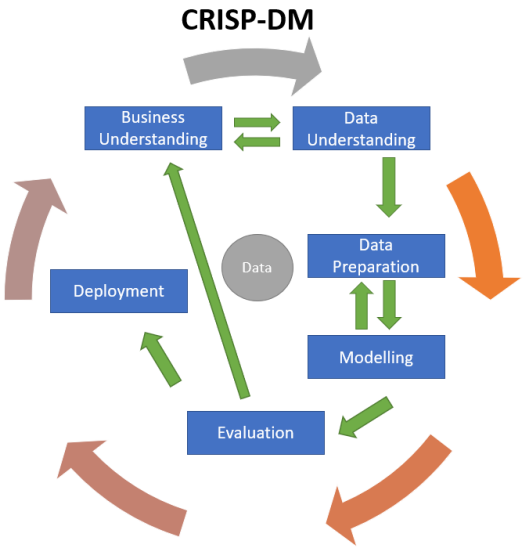
The Cross-Industry Standard Process for Data Mining (CRISP-DM) is a comprehensive and widely adopted methodology for guiding data mining and machine learning projects. The primary goal of CRISP-DM is to facilitate the efficient and effective development of data-driven solutions by breaking down the entire project lifecycle into well-defined stages. Each stage encompasses specific tasks and objectives, ensuring that data scientists and analysts follow a clear path from initial business understanding to model deployment and beyond. The CRISP-DM methodology consists of six key stages Business Understanding, Data Understanding, Data Preparation, Modelling, Evaluation, and Deployment (IBM, 2022)



(Saltz, 2017)

Business Understanding

In order to develop this project, the mean focus is to understand the objectives of the project. Our prime goal in this research work is to classify fraudulent credit card transactions from legitimate ones by using supervised machine learning techniques. The idea of this project is to create a credit card fraud detection model which would be an easier way to resolve the class imbalance problems. By developing this model, it would be useful for certain group pf users, such as credit card providers and other financial institutes. They would be able to provide faster, more scalable yet reliable credit card fraud detection. This project would provide a huge contribution in the field of credit card fraud detection, and it would also help the stakeholders to minimize the number of frauds. The primary aim is to protect credit cardholders and financial institutions from potential financial losses due to fraudulent transactions. Understanding the impact of credit card fraud on both consumers and businesses is essential to develop an effective fraud detection system that safeguards electronic payment systems. The key business objective is to build a robust and accurate machine learning algorithm that can accurately classify transactions as legitimate or fraudulent. By comprehending the business aspects of credit card fraud detection, people can develop a targeted and effective machine learning solution that addresses the specific needs of the financial industry and protects customers from potentially fraudulent activities. (IBM, 2022)



. (Tabladillo, n.d.)

Data Understanding

Data exploration is the initial step in the data analysis process, where you interactively examine and understand the characteristics, patterns, and structure of the dataset. The main goal of data exploration is to gain insights into the data, identify potential issues, and prepare it for further analysis, modeling, or visualization. This process involves various techniques and tools to better understand the data, detect anomalies, and make informed decisions about data preprocessing and analysis strategies. During the next step, one of the crucial phases when developing the CRISP-DM methodology is understanding the selected data. Understanding the data allows data practitioners to assess the data quality, identify potential errors, inconsistencies, or missing values. Being able to get the required data ethically from a reliable source itself is a big challenge. (IBM, 2022)

For this research, we are using open-source data for European credit cardholders recorded in 2013, which can be found on Kaggle. When analysing this dataset, you will notice some confidentiality issues, the data providers have changed the sensitive information in the form of transformed PCA components, for the data protection. The dataset used, contains one categorical feature and 30 other continuous features, including 'Time' and 'Amount' of the given transaction. (Kaggle, 2013)

```
1 credit_card_data = pd.read_csv('creditcard_dataset.csv')
2 print(credit_card_data.shape)
3 print(credit_card_data.columns)
4 credit_card_data.head(3)
```

```
(284807, 31)
Index(['Time', 'V1', 'V2', 'V3', 'V4', 'V5', 'V6', 'V7', 'V8', 'V9', 'V10',
       'V11', 'V12', 'V13', 'V14', 'V15', 'V16', 'V17', 'V18', 'V19', 'V20',
       'V21', 'V22', 'V23', 'V24', 'V25', 'V26', 'V27', 'V28', 'Amount',
       'Class'],
      dtype='object')
```

	Time	V1	V2	V3	V4	V5	V6	V7	V8	V9	...	V21	V22	V23	V24	V25
0	0.0	-1.359807	-0.072781	2.536347	1.378155	-0.338321	0.462388	0.239599	0.098698	0.363787	...	-0.018307	0.277838	-0.110474	0.066928	0.128539
1	0.0	1.191857	0.266151	0.166480	0.448154	0.060018	-0.082361	-0.078803	0.085102	-0.255425	...	-0.225775	-0.638672	0.101288	-0.339846	0.167170
2	1.0	-1.358354	-1.340163	1.773209	0.379780	-0.503198	1.800499	0.791461	0.247676	-1.514654	...	0.247998	0.771679	0.909412	-0.689281	-0.327642

3 rows × 31 columns

Data Exploration

Data exploration is a crucial initial step in the data analysis process that involves gaining insights and understanding from the dataset. Visualization is one of the efficient ways to explore and understand the structures of the data.
The dataset used for this project contains a large number of credit card transactions, both genuine and fraudulent. The goal is to train a model that can learn patterns and features from the data to accurately classify these transactions. In this applied project, I implement and assess the performance of various machine learning models, including logistic regression, random forests, and neural networks, using a rich dataset from Kaggle. The dataset contains approximately 300,000 credit card transactions occurring over two days in Europe.
Data preparation helps identify and handle data quality issues, such as missing values, outliers, duplicates, and inconsistencies. By addressing these issues, the overall data quality improves, leading to more robust analysis and modelling, it also allows the creation of new features or transformations of existing features to capture important information and relationships withing the data.
The low-quality data preparation approach could result in high computational time and cost, poor results of models.
Due to all these factors, data preparation is one of the most challenging and time-consuming stage in the data processing. During this project, we were able to performance, Summary Statistics, histograms, Bar charts, heatmaps and more.
Below a few examples of our findings:

- Data types of all features (“Credit_card_data)

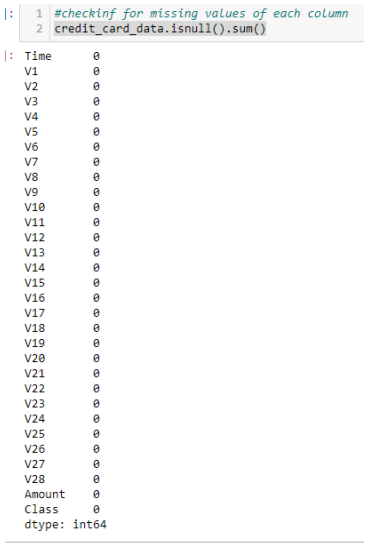
```
1 credit_card_data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 284807 entries, 0 to 284806
Data columns (total 31 columns):
 #   Column  Non-Null Count   Dtype  
---  ------  --------------   -----  
 0   Time    284807 non-null  float64
 1   V1      284807 non-null  float64
 2   V2      284807 non-null  float64
 3   V3      284807 non-null  float64
 4   V4      284807 non-null  float64
 5   V5      284807 non-null  float64
 6   V6      284807 non-null  float64
 7   V7      284807 non-null  float64
 8   V8      284807 non-null  float64
 9   V9      284807 non-null  float64
 10  V10     284807 non-null  float64
 11  V11     284807 non-null  float64
 12  V12     284807 non-null  float64
 13  V13     284807 non-null  float64
 14  V14     284807 non-null  float64
 15  V15     284807 non-null  float64
 16  V16     284807 non-null  float64
 17  V17     284807 non-null  float64
 18  V18     284807 non-null  float64
 19  V19     284807 non-null  float64
 20  V20     284807 non-null  float64
 21  V21     284807 non-null  float64
 22  V22     284807 non-null  float64
 23  V23     284807 non-null  float64
 24  V24     284807 non-null  float64
 25  V25     284807 non-null  float64
 26  V26     284807 non-null  float64
 27  V27     284807 non-null  float64
 28  V28     284807 non-null  float64
 29  Amount  284807 non-null  float64
 30  Class   284807 non-null  int64  
dtypes: float64(30), int64(1)
memory usage: 67.4 MB
```

Analysing missing values is important because missing data can have a significant impact on data analysis and modelling. Understanding the patterns and reasons for missing values helps data practitioners make informed decisions on how to handle them effectively. A high percentage of missing values may indicate potential data collection issues or data entry errors. The Impact on Analysis can affect statistical analysis and machine learning models. Ignoring missing data or using inappropriate methods to handle them can lead to biased results and inaccurate predictions (Darwin., 1998).

Addressing missing values is an essential part of data preprocessing. Imputing missing data or removing instances with missing values ensures that the data is in a suitable format for further analysis and modelling. Missing values can lead to biased estimates and inflated standard errors, impacting the validity of statistical analyses and the generalizability of results. (Saltz, 2017)

Analyzing missing values is an essential step in understanding the data's completeness and quality, enabling researchers to make informed decisions on how to handle the missing data appropriately for accurate and reliable analysis.



```
#checkinf for missing values of each column
credit_card_data.isnull().sum()
```

```
Time      0
V1        0
V2        0
V3        0
V4        0
V5        0
V6        0
V7        0
V8        0
V9        0
V10       0
V11       0
V12       0
V13       0
V14       0
V15       0
V16       0
V17       0
V18       0
V19       0
V20       0
V21       0
V22       0
V23       0
V24       0
V25       0
V26       0
V27       0
V28       0
Amount    0
Class     0
dtype: int64
```

Data Preparation

Data preparation is another important step in the data analysis process.t It involves cleaning, transforming, and organizing the data to make it suitable for analysis and modelling. Proper data preparation is essential for accurate and reliable results. During these analyses, we can performance different types of analyses, bay cleaning and transforming the data, the overall data quality is improved, leading to more reliable and trustworthy results. In machine learning, the performance and generalization of models heavily depend on the quality of the data used for training. Proper data preparation ensures that the model is trained on a clean, consistent, and relevant dataset, leading to more accurate predictions.

Missing values or outliers can introduce bias into the analysis and modelling process. Biased data can lead to erroneous conclusions and impact decision-making based on the results. Real-world data can be messy and vary in format and structure. Proper data preparation ensures that the data is organized and formatted consistently, making it easier to analyze and interpret.

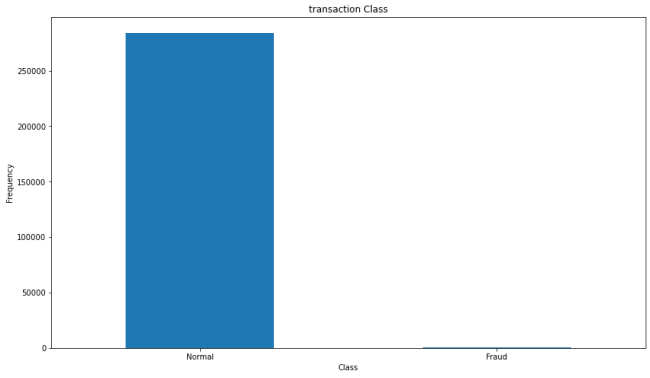
When Dealing with Imbalanced Data in Machine learning algorithms, they are designed to perform best when trained to adequate samples of both classes. In our dataset, we could analyse that, only 0.17% of the total Credit Card transaction is fraudulent. Therefore, it needs to be handled before applying any algorithm to it. By Combining Oversampling and Under sampling we were able to combine of oversampling the minority class (fraudulent) and under sampling the majority class (normal) in order to achieve more balanced dataset.

- Investigating target class distribution: Understanding the distribution of the target class is crucial for Class Imbalance, Model performance and much more.

Before:

```
count_classes = pd.value_counts(credit_card_data['Class'], sort = True)
count_classes.plot(kind = 'bar', rot=0)
plt.title("transaction Class")
plt.xticks(range(2), LABELS)
plt.xlabel("Class")
plt.ylabel("Frequency")
```

```
Text(0, 0.5, 'Frequency')
```



Overall, proper data preparation lays the foundation for accurate, reliable, and meaningful data analysis and machine learning. It ensures that the data used for modelling is of high quality, representative, and suitable for the task at hand, leading to more informed decision-making and valuable insights from the data.
33s

Dealing with Unbalance Data Building a sample dataset from the dataset, it will contain a similar distribution of normal and Fraudulent Transaction.

```
1 legit_sample = legit.sample(n=492) #taking the radom saple of the dataset
```

Concatenating Dataframes

```
1 new_data = pd.concat([legit_sample,fraud], axis=0)
```

```
1 new_data.head() #showing the new dataset
```

	Amount	Time	V1	V2	V3	V4	V5	V6	V7	V8	...	V20	V21	V22	V23
155052	0.76	104280.0	2.111327	0.356141	-2.361005	0.381346	1.125613	-1.098481	0.854416	-0.566408	...	-0.253324	0.194597	0.994540	-0.183938
203429	23.84	134813.0	-1.083670	0.143245	2.339141	1.504218	0.756779	0.459225	-0.292890	0.324416	...	0.435989	0.242507	0.512431	-0.244281
1394	0.76	1079.0	-1.114721	1.076515	-0.695010	-2.570389	2.311144	2.962518	0.082177	0.881409	...	0.490382	-0.199782	-0.528762	-0.069945
247360	15.80	153551.0	-1.070648	0.788632	0.122755	-1.560835	0.863830	0.307210	0.613635	-0.006253	...	-0.325888	-0.512067	-1.105043	-0.279858
146052	17.77	87395.0	-0.865070	0.400865	3.934537	4.858052	-0.202272	1.854239	-0.670641	0.211515	...	0.528313	-0.024958	0.806938	-0.421682

5 rows × 31 columns

```
1 new_data.tail() #showing the new dataset
```

	Time	V1	V2	V3	V4	V5	V6	V7	V8	V9	...	V21	V22	V23	V24
279863	169142.0	-1.927883	1.125653	-4.518331	1.749293	-1.566487	-2.010494	-0.882850	0.697211	-2.064945	...	0.778584	-0.319189	0.639419	-0.294885
280143	169347.0	1.378559	1.289381	-5.004247	1.411850	0.442581	-1.326536	-1.413170	0.248525	-1.127396	...	0.370612	0.028234	-0.145640	-0.081049
280149	169351.0	-0.676143	1.126366	-2.213700	0.468308	-1.120541	-0.003346	-2.234739	1.210158	-0.652250	...	0.751826	0.834108	0.190944	0.032070
281144	169966.0	-3.113832	0.585864	-5.399730	1.817092	-0.840618	-2.943548	-2.208002	1.058733	-1.632333	...	0.583276	-0.269209	-0.456108	-0.183659
281674	170348.0	1.991976	0.158476	-2.583441	0.408670	1.151147	-0.096695	0.223050	-0.068384	0.577829	...	-0.164350	-0.295135	-0.072173	-0.450261

Result after concatenation

5 rows × 31 columns

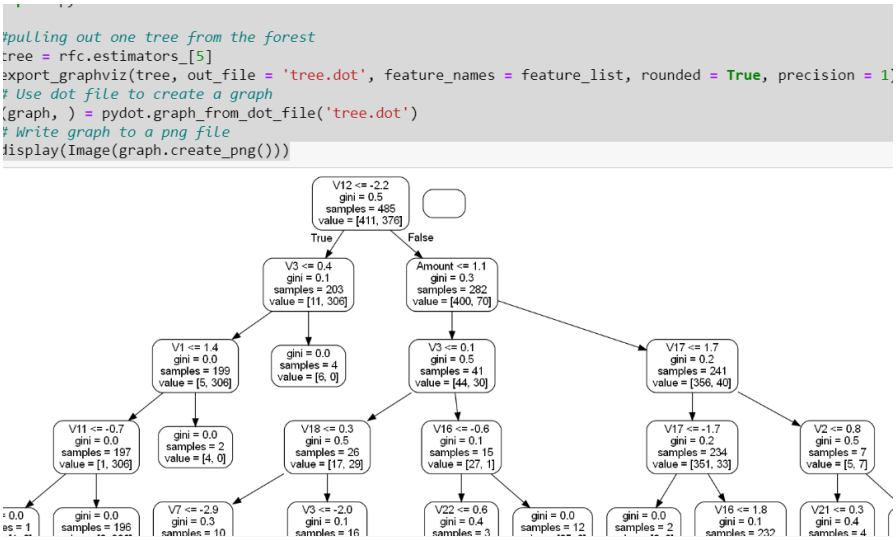
```
[36]: 1 new_data['Class'].value_counts()

[36]: 0    492
      1    492
      Name: Class, dtype: int64
```

Modelling Approach

After data preparation steps like feature selection and class balancing, the proposed modes are implemented on processed data, The effect of various preprocessing techniques on proposed machine learning models is evaluated and compared. Below a few examples of the tools used during the Modelling approach.

Random Forest: The random forest model is an ensemble of many decision trees to solve classification problems. It is a powerful ensemble learning method used in machine learning for both classification and regression tasks. It is an extension of the decision tree algorithm that builds multiple decision trees and combines their predictions to improve overall accuracy and reduce overfitting. The random forest has two-step processing. Random forest has advantages like the ability to handle unbalanced data, high capability to handle large data. It uses a collection of decision trees to make predictions. Each decision tree in the forest is trained independently on a different subset of the data and features, and their predictions are combined to produce the final result. By combining multiple decision trees, the model can effectively handle class imbalance and achieve higher predictive performance compared to individual decision trees.



Implementation

Implementing a credit card fraud detection system involves several steps, including data preprocessing, model selection, and evaluation. In this section Ill get into details on how it was created. All implementation of the proposed methodology has been carried out using Python language. For our implementation, Python emerged as the most optimal choice due to its numerous advantages and support in the data science. Its excellent code readability making it an ideal programming language for our credit card fraud detection project. Python's availability of powerful packages for data handling and pre-processing data, its position as a top choice for machine learning projects. With Python's, it enables us to streamline our development process, making it efficient and effective in tackling the complexities of credit card fraud detection. The data used for our research is publicly in a CSV format 10. It consists of credit card transactions since 2013 for European cardholders. In the dataset, there are 31 features in total, including the variable class, which means whether the given sample transaction is a fraudulent or legitimate. After chosen the dataset, we had to clean and scale the data. After this process was done, I had to analyse the data by using visualization and the patterns and correlations.

While exploring the data, we were able to see, that the data was imbalance. In order to achieve a better result and to avoid overfitting, I had to use a modelling technique called CT-GAN. Conditional Tabular Generative Adversarial Network is particularly useful when dealing with tabular datasets, where each row represents a separate observation, and columns represent different features or attributes. This type of data is common in various domains, including credit card fraud detection allows for the conditional generation of synthetic data. It can generate data samples conditioned on specific values or ranges for certain attributes. This makes it well-suited for generating data with specific characteristics. In our approach, we have leveraged popular classifiers such as Random Forest, Logistic Regression, from the Python sklearn library. Our training dataset is carefully balanced, comprising augmented data samples to ensure fair representation of both classes. For validating the training results, we have used a validation set of data. We are testing our models on a testing set of data only after fine-tuning our models.

Evaluation

This research aims to explore a novel approach that combines supervised machine learning techniques with Generative Adversarial Networks (GAN) to improve model performance significantly. We seek to evaluate the effectiveness of our proposed method in comparison to existing state-of-the-art approaches.

To conduct a comprehensive comparative study, we have designed two experiments. The first experiment involves evaluating model performances using unbalanced data. This scenario simulates real-world situations where credit card fraud is relatively rare compared to legitimate transactions, resulting in an imbalanced dataset.

Through this research, we hope to demonstrate the advantages of our combined approach and showcase its potential to elevate model performances, particularly in addressing the challenges posed by imbalanced data. By contributing to the field of credit card fraud detection, we aim to enhance security measures and protect users from fraudulent activities, thereby fostering trust and confidence in electronic payment systems.

Personal findings

The Credit Card Fraud Detection Project leverages the power of machine learning to the critical issue of credit card fraud. By developing an effective fraud detection algorithm, the project aims to protect credit cardholders and financial institutions from potential losses caused by fraudulent transactions. With machine learning algorithm to combat the rising threat of credit card fraud. The primary objective of this project is to safeguard credit cardholders and financial institutions from potential losses caused by fraudulent transactions.

Following the CRISP-DM methodology, the project began by understanding the challenges of imbalanced data and exploring historical credit card transaction datasets. Data preprocessing techniques were employed to handle missing values, outliers, and address class imbalance using oversampling and under sampling. I adopted the CRISP approach encompassing six phases: Business Understanding, Data Understanding, Data Preparation, Modelling, Evaluation, and Deployment. This methodology provided a structured framework for the project, ensuring a systematic and efficient approach.

The implementation of machine learning algorithms in credit card fraud detection has immense potential in minimizing financial losses and enhancing security for consumers and financial institutions alike. Through this project, I have not only honed my technical skills but also gained a deeper appreciation for the impact of data-driven solutions in addressing real-world challenges.

References

Darwin., C. (1998). *Applied multiple regression/correlation analysis for the behavioral sciences*. Retrieved from https://en.wikipedia.org/wiki/Linear_regression

IBM. (2022). Retrieved from https://www.ibm.com/docs/en/spss-modeler/saas?topic=dm-crisp-help-overview

Kaggle. (2013). Retrieved from Credit Card Fraud Detection: https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud

Saltz, S. &. (2017). Retrieved from Data Science Process: https://www.google.com/search?sxsrf=AB5stBgRoHZY6OrBNo7AbOfFCA2YiuNrHA:1690896107444&q=crisp-dm+implementation+credit+card+fraud&tbm=isch&source=lnms&sa=X&ved=2ahUKEwje-9fuxruAAxVZh1wKHaNYBWIQ0pQJegQIDBAB&biw=1280&bih=563&dpr=1.5#imgrc=6YKkW8XCZfns4M

Science, J. o. (2022). Retrieved from https://www.sciencedirect.com/science/article/pii/S1877050915007103

Tabladillo, M. (n.d.). *Microsoft*. Retrieved from https://learn.microsoft.com/en-us/azure/architecture/data-science-process/lifecycle-business-understanding