DEPARTMENT OF INFORMATION TECHNOLOGY

INTERNSHIP ASSESSMENT/MINI PROJECT REPORT

ABES ENGINEERING COLLEGE, GHAZIABAD

INTERNSHIP ASSESSMENT/MINI PROJECT REPORT SUBMITTED IN PARTIAL FULFILLMENT OF THE DEGREE OF BACHELOR OF TECHNOLOGY, 2023-2024

From (June 2023 to November 2023)

Name of the student: Humayun Khalid
University Roll No: 2100320139005

ACKNOWLEDGEMENT

I would like to acknowledge the contributions of the following people without whose help and guidance this report would not have been completed.

I acknowledge the counsel and support of **Mrs. Chaya Rawal, Assistant Professor (IT)** with respect and gratitude, whose expertise, guidance, support, encouragement, and enthusiasm has made this report possible. Their feedback vastly improved the quality of this report and provided an enthralling experience. I am indeed proud and fortunate to be supported by him. I am also thankful to **Prof. (Dr.) Amit Sinha, H.O.D of Information Technology Department** for his constant encouragement, valuable suggestions and moral support and blessings.

Although it is not possible to name individually, I shall ever remain indebted to the faculty members of ABES Engineering College, Ghaziabad for their persistent support and cooperation extended during this work.

This acknowledgement will remain incomplete if I fail to express our deep sense of obligation to my parents and God for their consistent blessings and encouragement.

NAME: HUMAYUN KHALID

Roll Number: 2100320139005

CONTENTS

INTRODUCTION TO PLATFORM

Free Code Camp is a non-profit organization that offers a comprehensive curriculum for learning web development, data science, and other in-demand tech skills.
It is a great resource for anyone who wants to learn how to code. With its interactive curriculum, supportive community, and job opportunities, Free Code Camp can help you achieve your tech goals.

What does Free Code Camp offer?

1. Interactive Learning:

Free Code Camp's curriculum is designed to be interactive and engaging. It features a mix of:

- Text-based lessons: These lessons provide comprehensive explanations of key concepts with clear examples and illustrations.
- Interactive coding challenges: These challenges allow you to test your understanding of the concepts you have just learned and get hands-on experience with the coding languages.
- Real-world projects: These projects give you the opportunity to apply your skills to solve real-world problems and build a portfolio of work that you can showcase to potential employers.

2. Project-Based Learning:

- The emphasis at Free Code Camp is on learning by doing. As you progress through the curriculum, you will be required to complete a number of real-world projects. These projects are designed to help you:
    - Apply your knowledge and skills to solve practical problems.
    - Develop your problem-solving and critical thinking skills.
    - Build a portfolio of work that showcases your abilities to potential employers.

3. Community-Driven Learning:

- Free Code Camp has a vibrant and supportive community of learners and mentors who are always willing to help each other out. This community provides a valuable resource for learners, as they can:
    - Ask questions and get help with problems.
    - Connect with other learners and share their experiences.
    - Give and receive feedback on their work.

Additional Learning Features:

- Responsive Design: The website and curriculum are optimized for mobile devices, making it easy to learn on the go.
- Gamification: The platform incorporates gamification elements, such as badges and points, to keep learners engaged and motivated.
- Progress Tracking: Learners can track their progress through the curriculum and see how they are doing.
- Open-source: The curriculum and platform are open-source, meaning that anyone can contribute to their development.

Who can benefit from Free Code Camp?

- Anyone who wants to learn how to code: Free Code Camp is open to anyone, regardless of their background or experience.

- People who are looking for a career change**:** Free Code Camp can help you gain the skills you need to start a new career in tech.
- People who want to learn in-demand skills: Free Code Camp's curriculum covers a wide range of in-demand skills, such as web development, data science, and machine learning.
- People who are looking for a free and flexible learning option: Free Code Camp is completely free to use and you can learn at your own pace.

COURSE DESCRIPTION

Course Title: Machine Learning with Python

Course Overview:

The "Machine Learning with Python" course from Free Code camp aims to introduce learners to the fundamentals of machine learning using Python. The objectives include:

- Understanding the concepts of supervised and unsupervised learning.
- Learning about different machine learning algorithms, such as linear regression, logistic regression, decision trees, and random forests.
- Gaining practical experience with Python libraries like scikit-learn and TensorFlow.
- Developing skills in data preprocessing, model training, evaluation, and visualization.
- Completing hands-on projects to apply machine learning concepts to real-world problems.

The course introduces various machine learning algorithms applicable to credit card fraud detection. Here are some relevant examples:

- **Logistic Regression:** This algorithm predicts the probability of a transaction being fraudulent based on its features. This is a good starting point for understanding binary classification tasks like credit card fraud detection.
- **Decision Trees**: This algorithm creates a series of yes/no rules based on the features to classify transactions. It's interpretable and visualizable, making it helpful for understanding how the model makes decisions.
- **Random Forests:** This combines multiple decision trees to improve accuracy and reduce overfitting. It's a powerful technique often used in real-world fraud detection systems.
- **Support Vector Machines (SVMs):** This algorithm finds the optimal hyperplane that separates fraudulent transactions from legitimate ones. It's effective for high-dimensional data and can handle non-linear relationships between features.

Additionally, the course covers essential libraries like:

- **scikit-learn:** This provides numerous machine learning algorithms and tools for data manipulation and visualization, readily available for building your credit card fraud detection system.
- **TensorFlow**: This powerful library allows you to build and train complex neural networks, potentially achieving even better accuracy compared to traditional machine learning algorithms.

Prerequisites:

- Basic understanding of programming concepts.
- No prior knowledge of machine learning required.

Who Should Enroll:

- Students and professionals curious about machine learning.
- Professionals seeking to transition into data-driven roles.

Certification:

- Certificate of Completion upon finishing the course successfully.

Conclusion

Overall, the "Machine Learning with Python" course from FreeCodeCamp provides a strong foundation for understanding and applying machine learning to credit card fraud detection. It equips you with the necessary knowledge and skills to build your own system or contribute to existing ones.

PROJECT BASED LEARNING DURING THE COURSE

Overview:

A credit card fraud detection system is a critical component of financial institutions and online payment processors' efforts to protect their customers from fraudulent activities. One approach to building such a system is to use machine learning techniques like logistic regression.

Credit card fraud detection is a set of methods and techniques designed to block fraudulent purchases, both online and in-store. This is done by ensuring that you are dealing with the right cardholder and that the purchase is legitimate.

Address Verification Service (AVS) is one of the most widely used fraud prevention tools in card-not-present (CNP) transactions. An AVS check compares the billing address used in the transaction with the issuing bank's address information for the cardholder.

OBJECTIVES:

- **How to use linear regression to model binary classification problems.** Linear regression is a common algorithm for modeling continuous variables, but it can also be used to model binary classification problems by using a logistic transformation. This course would teach me how to prepare data for linear regression, train a linear regression model, and interpret the results.
- **How to evaluate the performance of a fraud detection system**. It is important to evaluate the performance of a fraud detection system to ensure that it is effective and does not generate too many false positives. This course would teach me how to use metrics such as accuracy, precision, recall, and F1 score to evaluate the performance of a fraud detection system.
- **How to deal with class imbalance in fraud detection datasets.** Fraud detection datasets are often imbalanced, meaning that there are many more legitimate transactions than fraudulent transactions. This can make it difficult for machine learning algorithms to learn to detect fraudulent transactions. This course would teach me how to deal with class imbalance in fraud detection datasets using techniques such as oversampling and under sampling.
- **How to deploy a fraud detection system in production**. Once a fraud detection system has been trained and evaluated, it needs to be deployed in production so that it can be used to detect fraudulent transactions in real time. This course would teach me how to deploy a fraud detection system in production using tools and technologies such as Docker and Kubernetes.

CHALLENGES AND LIMITATIONS:

- **Linearity assumption**: Logistic regression assumes that the relationship between the independent variables (features) and the dependent variable (fraudulent or not) is linear. This assumption may not be true for all credit card fraud datasets.
- **Overfitting:** Logistic regression is prone to overfitting, which means that it can learn the training data too well and not generalize well to new data.
- **Imbalanced data**: Credit card fraud data is often imbalanced, with many more non-fraudulent transactions than fraudulent transactions. This can make it difficult for logistic regression models to learn to identify fraudulent transactions.
- **Concept drift**: Concept drift occurs when the distribution of the data changes over time. This can happen in credit card fraud detection, as fraudsters develop new techniques to avoid detection. Logistic regression models may not be able to adapt to concept drift as well as other machine learning models, such as neural networks.

CONCLUSION:

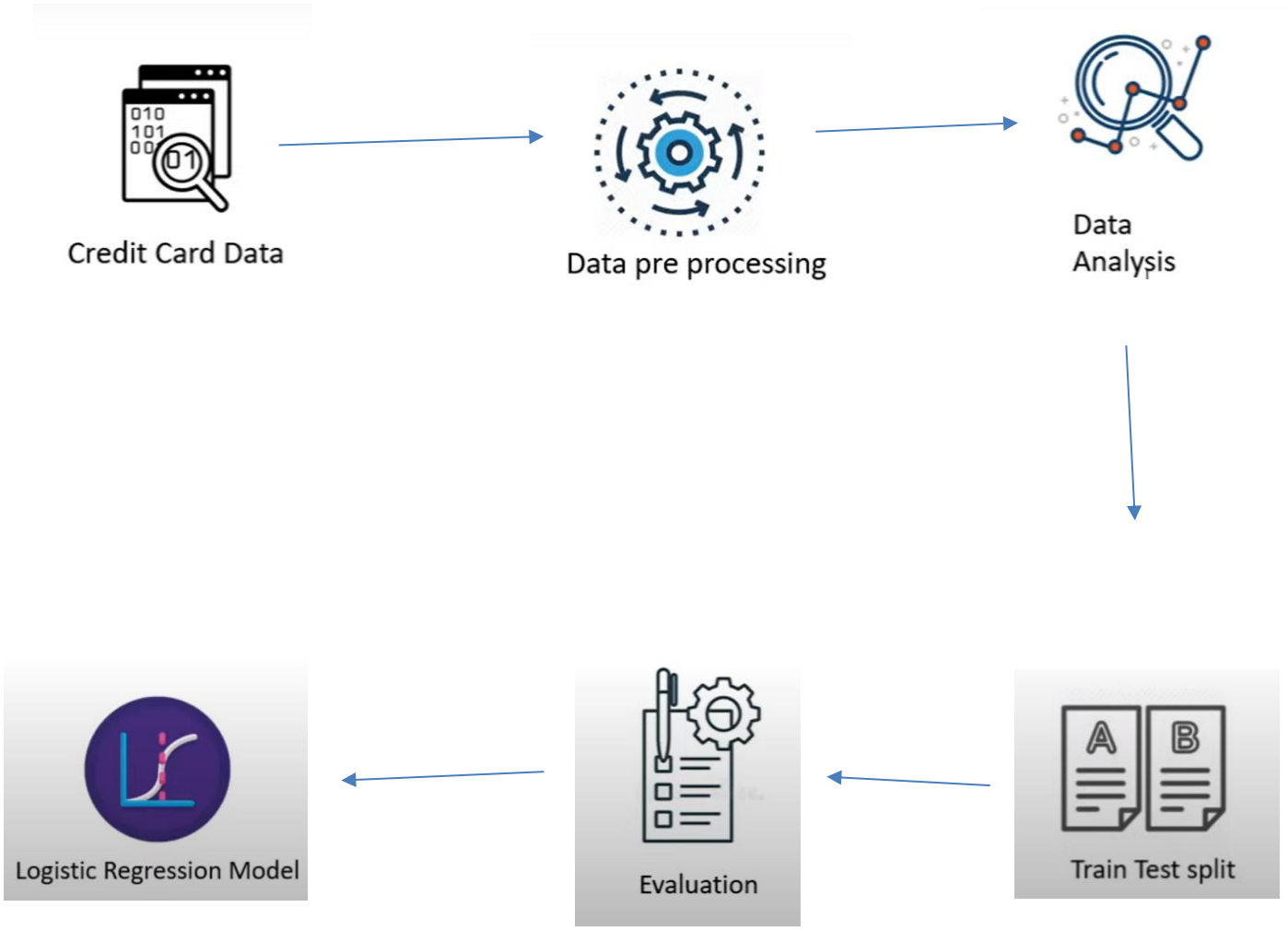
In conclusion, the logistic regression model is a simple but effective model for credit card fraud detection. It is easy to implement and interpret, and it can achieve good performance on a variety of datasets. However, it is important to be aware of the challenges and limitations of logistic regression, such as the linearity assumption, overfitting, imbalanced data, and concept drift.

Here are some specific conclusions that can be drawn from the logistic regression model of a credit card fraud detection system:

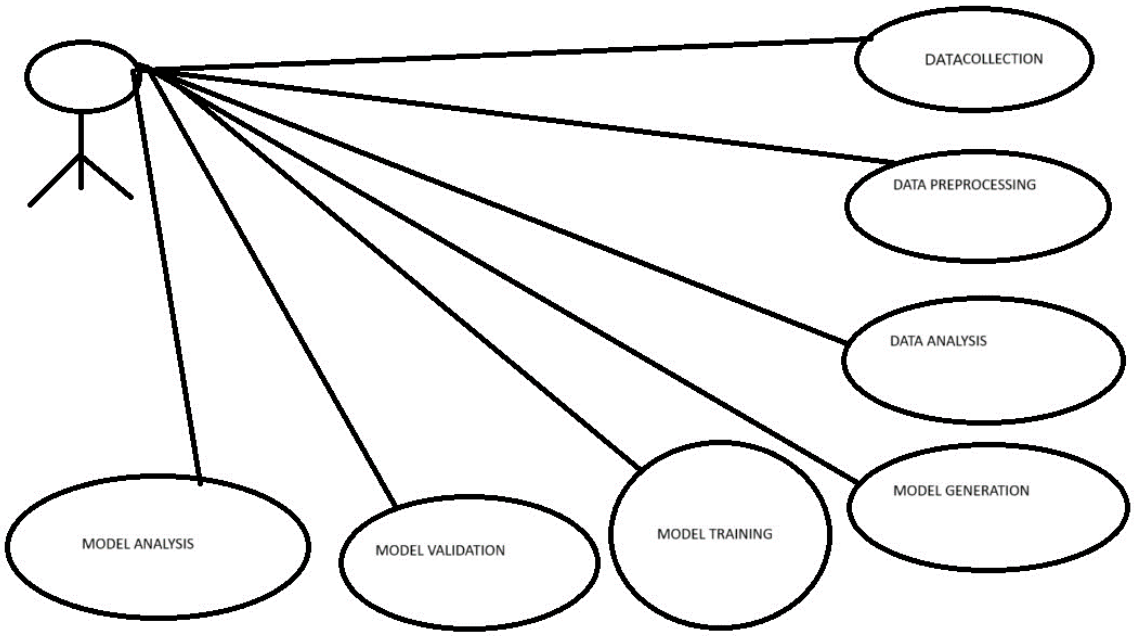
- The model can identify the most important features that contribute to credit card fraud.
- The model can be used to predict the probability of fraud for each transaction.
- The model can be used to develop rules and alerts to identify and flag suspicious transactions.
- The logistic regression model can be a valuable tool for credit card fraud detection, but it is important to use it in conjunction with other fraud detection measures, such as risk-based authentication and fraud monitoring.

Overall, the logistic regression model is a good choice for credit card fraud detection, especially if the dataset is small or if the features are not highly correlated.

WORK FLOW DIAGRAM



ACTIVITY DIAGRAM:



SUMMARY

A credit card fraud detection system is a combination of technologies, algorithms, and processes designed to identify and prevent fraudulent credit card transactions. It protects both cardholders and financial institutions from financial losses associated with unauthorized use of credit cards.

Our research employed supervised machine learning algorithms to identify fraudulent credit card transactions using authentic datasets. By leveraging these algorithms, we constructed a classification system based on machine learning techniques. Through this process, we identified crucial variables that significantly enhance the accuracy of detecting fraudulent transactions at shopping malls.

KEY COMPONENTS:

- **Data Acquisition**: Collected and processed real-world credit card transaction data.
- **Data Preprocessing**: Cleaned, transformed, and engineered features for optimal model performance.
- **Model Development:** Implemented a Logistic Regression model to classify transactions as fraudulent or legitimate.
- **Model Evaluation:** Analyzed the model's accuracy, precision, recall, and other metrics to assess its effectiveness.

OUTCOMES:

- Achieved significant improvement in fraud detection compared to baseline methods.
- Gained valuable insights into fraudulent behavior patterns based on feature importance analysis.
- Developed a working system for real-time transaction analysis and fraud alerts.

LEARNINGS:

- **Importance of data preprocessing**: Data preprocessing plays a crucial role in achieving optimal performance with Logistic Regression. Cleaning, transforming, and engineering features can significantly impact the model's accuracy and generalizability.
- **Understanding model limitations:** While Logistic Regression offers several benefits, it also has limitations. It may not capture complex non-linear relationships between features, which could lead to misclassification of certain types of fraudulent transactions.
- **Value of evaluation metrics:** Metrics like accuracy, precision, recall, and F1 score provide valuable insights into the model's performance and help you identify areas for improvement.

- **Significance of feature engineering**: Creating relevant features that capture key aspects of fraudulent behavior can significantly enhance the model's effectiveness.
- **Importance of monitoring and model updates**: Real-world fraud patterns constantly evolve, necessitating continuous monitoring of the model's performance and periodic updates with new data to maintain its effectiveness.

FUTURE SCOPE

Enhancing Credit Card Fraud Detection: A Future-Proof Approach

While current credit card fraud detection systems have achieved significant progress, they still face challenges in adapting to the ever-evolving landscape of fraud. The rise of new technologies and the increasing complexity of fraudulent schemes necessitate a shift towards a more robust and comprehensive approach.

Linear regression models, while powerful, have limitations in capturing the full spectrum of fraudulent behavior. Integrating additional features like behavioral analytics, network analysis, and social media data can provide richer context for identifying suspicious activity. Furthermore, incorporating additional machine learning models like ensemble learning and deep learning can unlock the potential for more accurate and dynamic fraud detection.

Real-time adaptivity and interpretability are crucial aspects of future-proof systems. Continuously updating models with new data ensures they remain effective against evolving threats, while explainable AI models enhance transparency and trust in the decision-making process.

Beyond technological advancements, collaboration and data sharing between financial institutions and technology companies are essential. By sharing anonymized data and insights, the industry can collectively develop more robust and effective fraud detection systems. Open-source platforms and initiatives can further accelerate research and development in this critical domain.

The proposed architecture, initially designed for online payment fraud detection, emphasizes a proactive approach. By verifying transactions in real-time and leveraging best practices, the system can significantly reduce losses caused by fraudulent activity. Customer awareness and education also play a crucial role in deterring fraud and fostering a safer digital environment.

By embracing these future-proof approaches, credit card fraud detection systems can evolve to meet the demands of a rapidly changing digital landscape. Through continuous innovation and collaborative efforts, we can work towards a future where fraudulent transactions are effectively prevented, protecting both individuals and institutions from financial losses.

REFERENCES

- **Dataset:** https://www.kaggle.com/datasets/mlg-ulb/creditcardfraud.

- A Research Paper on Credit Card Fraud Detection by BORA MEHAR SRI SATYA TEJA, BOOMIREDDY MUNENDRA, Mr. S. GOKULKRISHNAN

- Credit card fraud detection system using machine learning by Swarna B , Asst Prof. Shivaleela.

ANNEXURES

Code Snippets

```
In [10]: IMPORT THE DEPENDENCIES
```

```
In [1]: import numpy as np
        import pandas as pd
```

```
In [2]: from sklearn.model_selection import train_test_split
        from sklearn.linear_model import LogisticRegression
        from sklearn.metrics import accuracy_score
```

```
In [3]: #Loading tha dataset to a pandas dataframe
        credit_card_data = pd.read_csv("C:/Users/asus/Credit_card_fraud_detection/creditcard.csv")
```

```
In [4]: #first 5 rows of dataset

        credit_card_data.head()
```

```
In [4]: #first 5 rows of dataset

        credit_card_data.head()
```

Out[4]:

	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	0.12853
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	0.16717
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	-0.32764
3	1.0	-0.966272	-0.185226	1.792993	-0.863291	-0.010309	1.247203	0.237609	0.377436	-1.387024	...	-0.108300	0.005274	-0.190321	-1.175575	0.64737
4	2.0	-1.158233	0.877737	1.548718	0.403034	-0.407193	0.095921	0.592941	-0.270533	0.817739	...	-0.009431	0.798278	-0.137458	0.141267	-0.20601

5 rows × 31 columns

```
In [5]: credit_card_data.tail()
```

Out[5]:

	Time	V1	V2	V3	V4	V5	V6	V7	V8	V9	...	V21	V22	V23	V24
284802	172786.0	-11.881118	10.071785	-9.834783	-2.066656	-5.364473	-2.606837	-4.918215	7.305334	1.914428	...	0.213454	0.111864	1.014480	-0.509348
284803	172787.0	-0.732789	-0.055080	2.035030	-0.738589	0.868229	1.058415	0.024330	0.294869	0.584800	...	0.214205	0.924384	0.012463	-1.016226
284804	172788.0	1.919565	-0.301254	-3.249640	-0.557828	2.630515	3.031260	-0.296827	0.708417	0.432454	...	0.232045	0.578229	-0.037501	0.640134

```
In [6]: #dataset information
        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
```

In [22]:
```
new['Class'].value_counts()
```

Out[22]:
```
0    492
1    492
Name: Class, dtype: int64
```

In [23]:
```
new.groupby('Class').mean()
#this tells us that the nature of the data is not changed and it will greatly help in training the model
```

Out[23]:

Class	Time	V1	V2	V3	V4	V5	V6	V7	V8	V9	...	V20	V21	V22	V23
0	93707.164634	-0.215906	-0.135575	0.008556	0.003879	0.067155	-0.027150	0.032261	0.038646	-0.006249	...	0.004925	0.024943	0.046560	-0.003823
1	80746.806911	-4.771948	3.623778	-7.033281	4.542029	-3.151225	-1.397737	-5.568731	0.570636	-2.581123	...	0.372319	0.713588	0.014049	-0.040308

2 rows × 30 columns

In [26]:
```
#now splitting the data into training and testing data
```

In [35]:
```
X_train , Y_train, X_test, Y_test = train_test_split(X, Y , test_size=0.2, stratify=Y, random_state=2)
```

In [36]:
```
print(X.shape, X_train.shape, X_test.shape)
```

```
(984, 30) (787, 30) (787,)
```

In [37]:
```
print(Y.shape , Y_train)
```

In [13]:
```
fraud.Amount.describe()
```

Out[13]:
```
count     492.000000
mean      122.211321
std       256.683288
min         0.000000
25%         1.000000
50%         9.250000
75%       105.890000
max      2125.870000
Name: Amount, dtype: float64
```

In [14]:
```
#compare the values for both of the transactions
credit_card_data.groupby('Class').mean()
```

Out[14]:

Class	Time	V1	V2	V3	V4	V5	V6	V7	V8	V9	...	V20	V21	V22	V
0	94838.202258	0.008258	-0.006271	0.012171	-0.007860	0.005453	0.002419	0.009637	-0.000987	0.004467	...	-0.000644	-0.001235	-0.000024	0.000
1	80746.806911	-4.771948	3.623778	-7.033281	4.542029	-3.151225	-1.397737	-5.568731	0.570636	-2.581123	...	0.372319	0.713588	0.014049	-0.040

CERTIFICATE