HUMBER INSTITUTE OF TECHNOLOGY AND ADVANCED LEARNING (HUMBER COLLEGE)

Group Assignment 1 Report

Machine Learning 2 - BIA-5402-0GA

Group - 5

Artificial Neural Networks for Fraud Detection in Supply Chain Analytics: A Study on MLPClassifier and Keras

Submitted by:

Last Name	First Name	Student Number
Amodkar	Sweta	N01472800
Das	Subhanjan	N01431473
Patel	Hardi	N01480409
Shah	Divyansh	N01472284
Shand	Yuvraj	N01479401
Tyagi	Abhi	N01474042

Submitted to: **Professor Salam Ismaeel**

Submission Date: **2023-01-31**

Abstract

In this study, we aimed to detect fraudulent activities in the supply chain through the use of neural networks. The study focused on building two machine learning models using the MLPClassifier algorithm from the scikit-learn library and a custom neural network using the Keras library in Python. Both models were trained and tested on the DataCo Supply Chain dataset. The results showed that the custom neural network achieved an accuracy of 97.67% in detecting fraudulent transactions, demonstrating its potential to minimize financial losses for organizations.

Introduction

In recent years, the use of neural networks in supply chain analytics has gained considerable traction as organisations look for ways to improve their operations and make more informed decisions. One area where neural networks can have a significant impact is in the detection of fraud before shipments are processed. Fraudulent activities can cause significant financial losses, and early detection is essential for minimising any damage.

We present our study on the use of neural networks for detecting fraud in the supply chain. Two models were developed as part of this study: one using the MLPClassifier algorithm from the scikit-learn library, and another using a custom neural network built using the Keras library in Python. These models were developed using open-source libraries, including NumPy for numerical computation, Pandas for data manipulation, Seaborn for statistical data visualisation, matplotlib for plotting, and the machine learning frameworks SciKit Learn, Keras, and Tensorflow (backend). The scikit-learn library is a widely used machine learning library in Python, and the MLPClassifier algorithm from this library is a type of multi-layer perceptron classifier that has been shown to perform well on various classification tasks. The custom neural network, on the other hand, was designed to provide a deeper level of control over the architecture and training process, allowing for a more customised solution. The objective of our study is to identify potential fraudulent activities in the supply chain before shipments are processed, thus reducing the risk of financial loss for the organisation. Some of the most important attributes in our dataset are as follows with their description:

- **Type:** Kind of transaction made
- **Sales per customer:** Total sales made for each customer
- **Delivery Status:** State of the order delivery: Early shipment, delayed delivery, cancelled shipment, or on-time shipment
- **Late_delivery_risk:** A categorical variable indicating if the delivery is late (1) or not (0)
- **Category Id:** Code for the product category
- **Category Name:** Description of the product category
- **Market:** Delivery destination for order: Africa, Europe, LATAM, Pacific Asia, USCA
- **Sales:** Sales value

Other critical attributes in our dataset are mentioned below in a tabular format:

Customer Information	Customer City, Customer Country, Customer Email, Customer Fname, Customer Id, Customer Lname, Customer Password, Customer Segment, Customer State, Customer Street, Customer Zipcode
Order Information	Order City, Order Country, Order Customer Id, Order Date (DateOrders), Order Id, Order Item Cardprod Id, Order Item Discount, Order Item Discount Rate, Order Item Id, Order Item Product Price, Order Item Profit Ratio, Order Item Quantity, Order Item Total, Order Profit Per Order, Order Region, Order State, Order Status, Benefit per order
Product Information	Product Card Id, Product Category Id, Product Description, Product Image, Product Name, Product Price, Product Status
Department Information	Department Id, Department Name
Shipping Information	Shipping Date (DateOrders), Shipping Mode, Days for shipping (real), Days for shipment (scheduled)

Table.1: *List of Attributes*

Literature Review

Supply chain management has become increasingly data-driven, with organisations seeking ways to optimise their operations and gain a competitive advantage. One area that has garnered significant attention is the use of neural networks in supply chain and logistics. These algorithms have proven to be highly effective in solving complex problems and uncovering valuable insights from data.

Here we explore the applications of neural networks in supply chain analytics and how they are being used to address the challenges facing the field. Accurate demand forecasting is one of the most critical challenges in supply chain management. Fluctuations in demand can have a significant impact on an organisation's ability to meet customer requirements. Neural networks have shown promise in this area by learning from historical data and uncovering patterns that may not be noticeable through traditional statistical methods. For instance, Chawla et al., (2019) observed that an artificial neural network outperformed traditional time series methods in forecasting demand in a retail supply chain.

In addition to demand forecasting and inventory management, neural networks have also been applied to other areas of supply chain analytics, including supplier selection, logistics optimization, and risk management. For example, Yan et al. (2020) used a deep reinforcement learning approach to optimise logistics operations. These findings suggest that neural networks can be valuable tools for organisations looking to enhance their supply chain operations.

Data Cleaning and Handling Missing Data Values

In order to effectively analyze the supply chain data, it was obligatory to perform data cleaning and handling on the dataframe (df). We start by identifying and removing any irrelevant columns that would not contribute to the analysis. Columns such as 'Customer Email', 'Product Status', 'Customer Password', 'Customer Street', 'Customer Fname', 'Customer Lname', 'Latitude', 'Longitude', 'Product Description', 'Product Image', 'Order Zipcode', and 'shipping date (DateOrders)' were deemed unnecessary and were therefore dropped from the data set. The data cleaning process was done in-place, which means that the original data frame was modified and the shape of the modified data frame was printed, revealing that there were 180519 rows and 42 columns.

Next, we used the df.isnull().sum() function to check for missing values in the DataFrame and found that only the 'Customer Zipcode' column had three missing values. These missing values were filled with zeros to ensure that no information was lost during the analysis process. Additionally, a new column named ‘Cust_Full_Name’ was created by joining ‘Customer Fname’ and ‘Customer Lname’.

Finally, we created new columns from the 'order date (DateOrders)' column to allow for a more in-depth analysis of the data. By using the 'pd.DatetimeIndex' function, the date string was converted into year, month, day, and hour and stored in separate columns, namely 'order_yr', 'order_month', 'order_day', and 'order_hour'. This provided valuable information about the timing of the orders and allowed us to better understand the trends and patterns in the supply chain data.

By performing these data cleaning and handling steps, we ensured that the data was ready for analysis and modelling and also that we had the information we needed to make informed decisions about the supply chain processes

Exploratory Data Analysis

In order to gain further insights into the data and identify trends, we conducted an exploratory data analysis (EDA) of our DataCo Global Supply Chain data. The first step in our EDA was to create a heatmap that revealed some products have a negative benefit per order, which indicates that these orders are causing losses for the company. To further investigate this issue, we created two bar graphs using the Python data visualisation libraries ‘plotly’ and ‘seaborn’.

We filtered the data to only include rows where the 'Benefit per order' was negative, and saved this filtered data in a separate DataFrame named "loss". The first bar graph shows the top-10 most loss-making products based on their category name, while the second bar graph depicts the top-5 most loss-making markets. These graphs helped us understand how different features, like product categories and markets, impacted the outcome variable 'Benefit per order'.

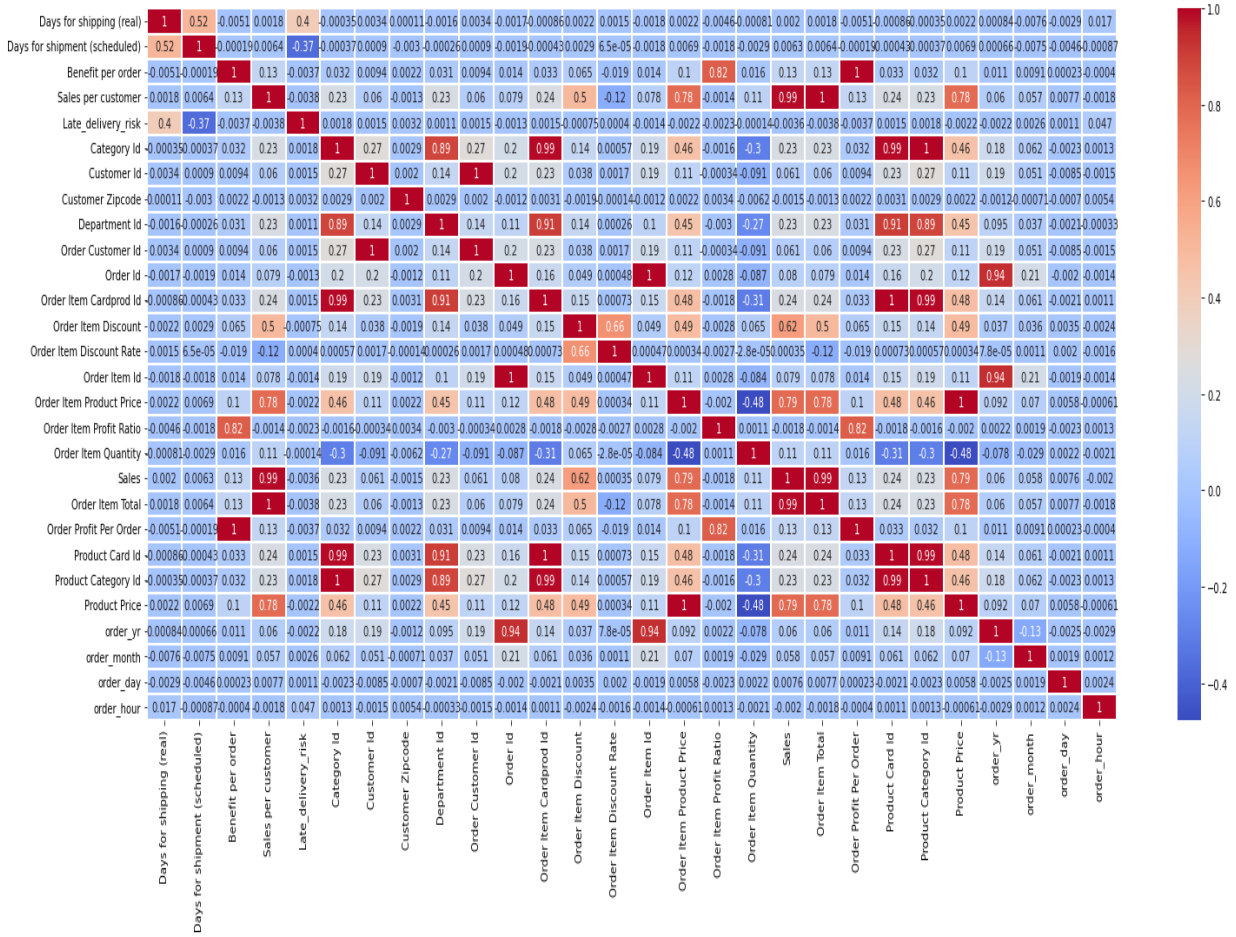

Fig.1: *Correlation Heatmap in CoolWarm*

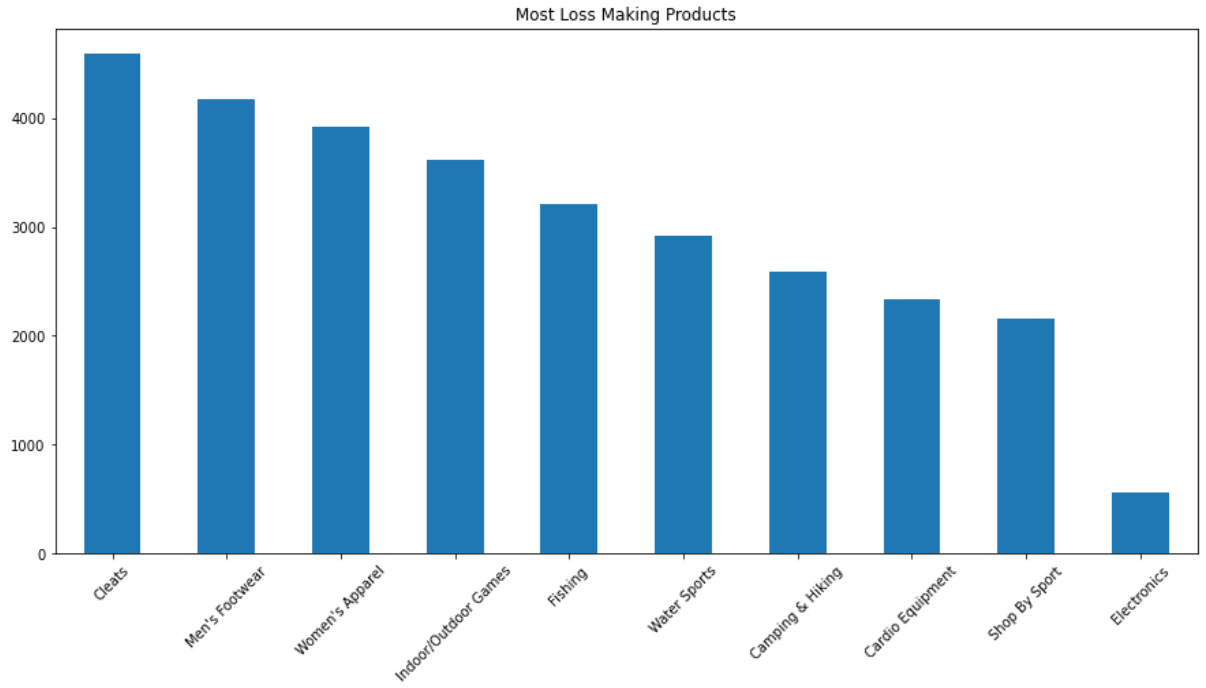

Fig.2: *Bar Plot - Top 10 Most Loss Making Products*

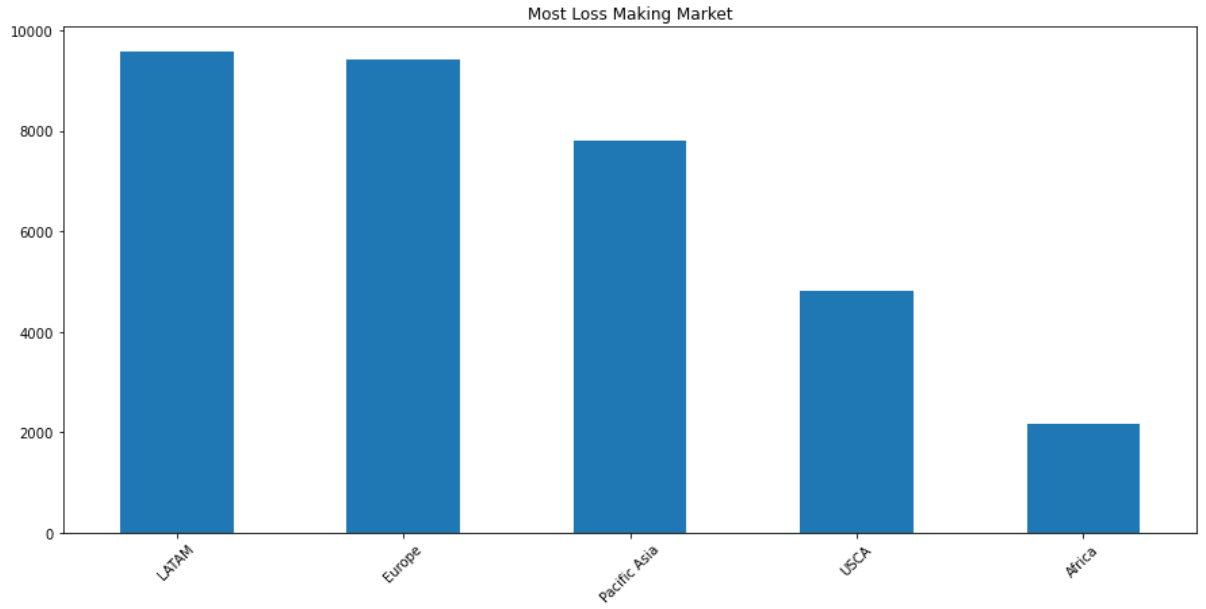

Fig.3: *Top-5 Most Loss Making Markets*

Data Modelling - Neural Networks

As part of this study, two machine learning models were developed to predict the outcome of sales orders. One model was developed using the MLPClassifier algorithm from the scikit-learn library, while the other was developed using a custom neural network built with the Keras library in Python.

Data Pre-Processing

Before building the models, all object-type variables were encoded, and the data was prepared for modelling. To do this, 10% of the dataset was selected for validation, while the remaining 90% was split into training data (70%) and test data (30%). The StandardScaler() function from scikit-learn was used to fit-scale the training data, and it was then ready for modeling.

This indicates that the machine learning model has high accuracy in both training and test datasets. The model's performance on the training set is 97.77% accurate, which suggests that it has learned the patterns in the data well. The performance on the test set (97.68%) is slightly lower but still high, indicating that the model generalises well to unseen data. However, it's important to consider if the model may be overfitting the training data, in which case the accuracy on the test set may be lower than expected.

MLPClassifier

MLP neural network transforms inputs into outputs using a layered structure. It's composed of interconnected nodes, where all layers except the input layer contain non-linear neurons with activation functions. The hidden layers, which can be one or many, introduce non-linearity and enable the network to learn complex input-output relationships (Klein, 2022).

Keras - Custom

The second model was a custom neural network, which had 1024 nodes. Keras is a Python-based deep learning API. The custom neural network was trained using the fit method, with an initial number of 10 epochs. The results showed that the train and test accuracy was high, at approximately 97%, while the loss was approximately 8.88%. However, increasing the number of epochs to 15 significantly reduced the loss and prevented overfitting of the data. The final F1 score calculated was 96.58%.

Model Validation and Evaluation

It is imperative to assess the performance of our models to ensure their effectiveness and reliability. One commonly used method for this purpose is the confusion matrix. A confusion matrix is a tool that summarises the true positive, false positive, true negative, and false negative predictions made by a classifier. This information is used to calculate the accuracy of the model, which is an important metric in determining the performance of the algorithm. Our confusion matrix model has a remarkable accuracy of 97.89%.

The difference in accuracy between the two models is small, with the MLPClassifier model having a slightly higher accuracy (97.87%) compared to the custom Neural Network (97.84%). However, overfitting can still occur in both models and it's important to consider other evaluation metrics such as precision, recall, and F1-score, as well as techniques to prevent overfitting such as early stopping, regularisation, and cross-validation.

```
Confusion Matrix (Accuracy 0.9787)

       Prediction
Actual     0     1
     0 17532   130
     1   254   136
```

Fig.4: *MLPClassifier - Confusion Matrix*

Conclusion

The focus of our study was to investigate the feasibility of using neural networks for detecting fraud in DataCo Global's supply chain. Two models were developed as part of this study: the MLPClassifier from the scikit-learn library and a custom neural network built with the Keras library in Python. The results showed that both models demonstrated high accuracy and a strong F1 score. In conclusion, the results of this study highlights the potential for neural networks to play a key role in the detection of fraud in the supply chain, and serves as a stepping stone for further research in these areas.

References

Chawla, A., Singh, A., Lamba, A., Gangwani, N., & Soni, U. (2019). Demand forecasting using artificial neural networks—a case study of American retail corporation. In Applications of Artificial Intelligence Techniques in Engineering: SIGMA 2018, Volume 2 (pp. 79-89). Springer Singapore.

Yan, Yimo & Chow, Andy & Ho, Chin Pang & Kuo, Yong-Hong & Wu, Qihao & Ying, Chengshuo. (2021). Reinforcement Learning for Logistics and Supply Chain Management: Methodologies, State of the Art, and Future Opportunities.

Klein, B. (2022, February 17). *Neural Networks with Scikit*. Retrieved January 28, 2023, from

https://python-course.eu/machine-learning/neural-networks-with-scikit.php