**Product Type Prediction: Machine learning model using Neural Network**

***Divya Jain***

***Himadri Patel***

***Edwin Jacob***

***Daegan Pimenta***

***Suryansh Gupta***

**ABSTRACT**

*This project examines the impact of varying hidden layer sizes in a Multi-Layer Perceptron (MLP) neural network on the accuracy of predicting product types in ‘supply\_chain\_dataset.csv’ dataset. By experimenting with different hidden layer sizes, ranging from 1 to 100, we observe significant variations in model performance. The results indicate that a moderate hidden layer size (around 10) yields the highest accuracy, suggesting that an optimal balance between model complexity and predictive performance is crucial. This finding emphasizes the importance of hyperparameter tuning in neural network models for supply chain applications.*

**INTRODUCTION**

In the modern supply chain landscape, predictive analytics plays a pivotal role in optimizing operations and improving decision-making. Machine learning models, particularly neural networks, have gained prominence due to their ability to capture complex patterns in large datasets. However, the effectiveness of these models is highly dependent on the appropriate selection of hyperparameters, such as the number of hidden layers and neurons within those layers. This study aims to investigate the effect of varying hidden layer sizes on the accuracy of a Multi-Layer Perceptron (MLP) classifier in predicting product types within a supply chain context.

**LITERATURE REVIEW**

The use of neural networks in supply chain management has been extensively explored in recent years. Neural networks can effectively handle the non-linear relationships and interactions inherent in supply chain data. The importance of hyperparameter optimization enhances the predictive performance of neural network models. They argue that inappropriate selection of hyperparameters, such as the number of hidden layers, can lead to suboptimal performance and increased computational costs.

Previous studies have shown mixed results regarding the optimal number of hidden layers. Hornik et al. (1989) demonstrated that a single hidden layer is theoretically sufficient to approximate any continuous function, given a sufficient number of neurons. However, in practical applications, multiple hidden layers often yield better results by enabling the model to learn more complex features (LeCun et al., 2015).

In the context of supply chain management, neural networks with more hidden layers outperformed traditional statistical methods in demand forecasting. Nevertheless, it was also observed that beyond a certain point, additional hidden layers did not significantly improve accuracy and sometimes even degraded performance due to overfitting. Similarly, the need for a balanced approach in selecting the number of hidden layers to achieve optimal performance without incurring unnecessary computational costs was highlighted.

Building on this body of work, our study aims to provide empirical evidence on the relationship between hidden layer size and model accuracy in a supply chain dataset. By systematically varying the hidden layer sizes, we seek to identify an optimal configuration that maximizes predictive accuracy while maintaining computational efficiency.

**DESCRIPTION OF DATASET AND DATA PROCESSING**

**DATASET SUMMARY**

This study aims to explore product type, for different products, customers. We used the “supply\_chain\_dataset.csv" dataset, a detailed collection of 100 records across 23 variables like Price, products sold, revenue generated, availability. These variables offer insights into the factors influencing the product type. Our initial analysis lays the groundwork for understanding product type.

**DETAILED VARIABLE OVERVIEW**

**Variables Explained**

The dataset includes variables like SKU, Price, Availability, Number of products sold etc. These factors are crucial in order to predict the product type.

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Product\_type** | **SKU** | **Price** | **Availability** | **Number\_of\_products\_sold** | **Revenue\_generated** | **Customer\_demographics** | **Stock\_levels** | **Lead\_times** | **Order\_quantities** | **...** | **Location** | **Lead\_time** | **Production\_volumes** | **Manufacturing\_lead\_time** | **Manufacturing\_costs** | **Inspection\_results** | **Defect\_rates** | **Transportation\_modes** | **Routes** | **Costs** |  |
| **0** | haircare | SKU0 | 69.808006 | 55 | 802 | 8661.99679 | Non-binary | 58 | 7 | 96 | ... | Mumbai | 29 | 215 | 29 | 46.279879 | Pending | 0.22641 | Road | Route B | 187.752075 |
| **1** | skincare | SKU1 | 14.843523 | 95 | 736 | 7460.90007 | Female | 53 | 30 | 37 | ... | Mumbai | 23 | 517 | 30 | 33.616769 | Pending | 4.854068 | Road | Route B | 503.065579 |
| **2** | haircare | SKU2 | 11.319683 | 34 | 8 | 9577.74963 | Unknown | 1 | 10 | 88 | ... | Mumbai | 12 | 971 | 27 | 30.688019 | Pending | 4.580593 | Air | Route C | 141.920282 |
| **3** | skincare | SKU3 | 61.163343 | 68 | 83 | 7766.83643 | Non-binary | 23 | 13 | 59 | ... | Kolkata | 24 | 937 | 18 | 35.624741 | Fail | 4.746649 | Rail | Route A | 254.776159 |
| **4** | skincare | SKU4 | 4.805496 | 26 | 871 | 2686.50515 | Non-binary | 5 | 3 | 56 | … | Delhi | 5 | 414 | 3 | 92.065161 | Fail | 3.14558 | Air | Route A | 923.440632 |

*Table 1.Data description*

**ANALYSIS OF PRELIMINARY DATA AND INITIAL OBSERVATIONS**

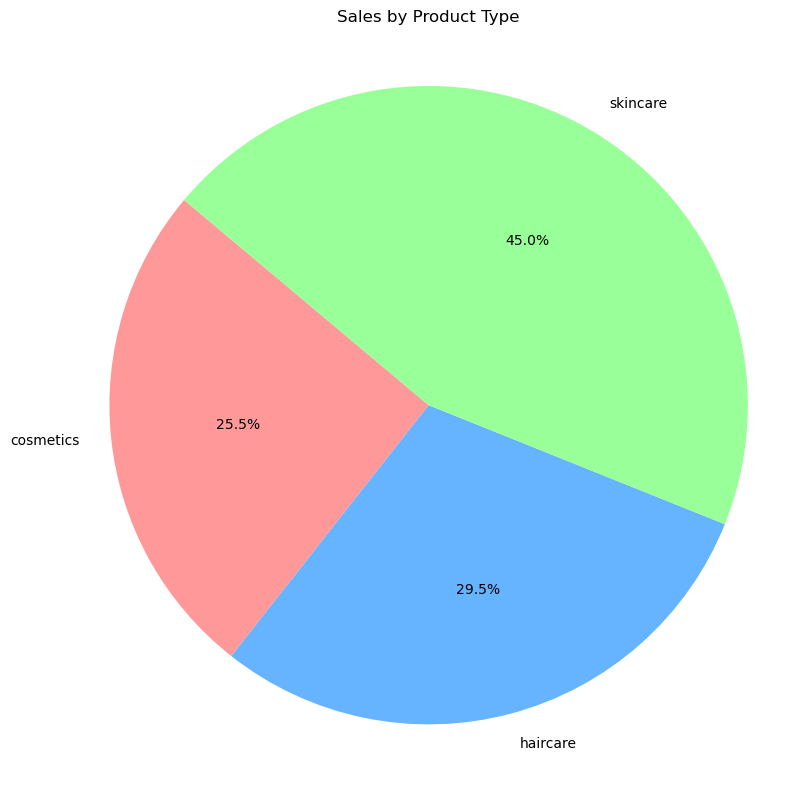
The initial analysis of the dataset indicates a meticulously curated selection with no missing values, ensuring high data integrity. This eliminates the need for imputation, preserving the dataset's accuracy. Certain column headers were replaced with \_ in order to avoid mishandling.

**DESCRIPTIVE STATISTICS**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | **Price** | **Availability** | **Number\_of\_products\_sold** | **Revenue\_generated** | **Stock\_levels** | **Lead\_times** | **Order\_quantities** | **Shipping\_times** | **Shipping\_costs** | **Lead\_time** | **Production\_volumes** | **Manufacturing\_lead\_time** | **Manufacturing\_costs** | **Defect\_rates** | **Costs** |
| **count** | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 | 100 |
| **mean** | 49.462461 | 48.4 | 460.99 | 5776.04819 | 47.77 | 15.96 | 49.22 | 5.75 | 5.548149 | 17.08 | 567.84 | 14.77 | 47.266693 | 2.277158 | 529.245782 |
| **std** | 31.168193 | 30.743317 | 303.780074 | 2732.84174 | 31.369372 | 8.785801 | 26.784429 | 2.724283 | 2.651376 | 8.846251 | 263.046861 | 8.91243 | 28.982841 | 1.461366 | 258.301696 |
| **min** | 1.699976 | 1 | 8 | 1061.61852 | 0 | 1 | 1 | 1 | 1.013487 | 1 | 104 | 1 | 1.085069 | 0.018608 | 103.916248 |
| **25%** | 19.597823 | 22.75 | 184.25 | 2812.84715 | 16.75 | 8 | 26 | 3.75 | 3.540248 | 10 | 352 | 7 | 22.983299 | 1.00965 | 318.778455 |
| **50%** | 51.239831 | 43.5 | 392.5 | 6006.35202 | 47.5 | 17 | 52 | 6 | 5.320534 | 18 | 568.5 | 14 | 45.905622 | 2.141863 | 520.430444 |
| **75%** | 77.198228 | 75 | 704.25 | 8253.97692 | 73 | 24 | 71.25 | 8 | 7.601695 | 25 | 797 | 23 | 68.621026 | 3.563995 | 763.078231 |
| max | 99.171329 | 100 | 996 | 9866.46546 | 100 | 30 | 96 | 10 | 9.929816 | 30 | 985 | 30 | 99.466109 | 4.939255 | 997.41345 |

*Table 2.Descriptive Statistics*

Table 2. provides a statistical snapshot highlighting key aspects such as Price, Availability, products sold, revenue generated, stock levels. The average number sold is approximately 461 units, with a minimum of 8 and a maximum of 996. On average, stock levels are about 47.77%, ranging from 0% to 100%. Average shipping time is 5.75 days, with a range from 1 day to 10 days. Average shipping costs are $5.55, ranging from $1.01 to $9.93. This is around 14.77 days on average, with a range from 1 day to 30 days. The average cost is $529.25, with a range from $103.92 to $997.41.



*Figure 1.Sales by Product type*

According to Figure 1, the pie chart visualizes the distribution of total sales across different product types in the supply chain dataset. the proportion of sales each product type contributes to the overall sales. Skincare contributing to maximum sales of the all product type, with cosmetics and haircare at an almost equal contribution. 45% of the business comes from skincare products, 29.5% from haircare, and 25.5% from cosmetics.

A graph of a bar chart

Description automatically generated with medium confidence

*Figure 2.Average Defect rates by product type*

According to Figure 1, the defect rate of haircare is higher calculating an average of 2.5. The bar graph highlights a critical area for improvement within the supply chain of haircare products. By focusing on reducing the defect rate, companies can enhance product quality, customer satisfaction, and overall business performance.

**MACHINE LEARNING FOR CLASSIFICATION USING NEURAL NETWORK**

The primary objective of this study is to classify product types using an MLP classifier and evaluate the impact of different hidden layer sizes on model accuracy. The methodology involves the following steps:

1. Data Splitting:
   * The dataset is split into training and testing sets using an 80-20 split. The training set is used to train the model, and the testing set is used to evaluate its performance.
2. Model Training:
   * An MLPClassifier from scikit-learn is used for classification. The neural network architecture is varied by changing the hidden layer sizes to study their impact on classification accuracy. The hidden layer sizes tested are [1, 2, 3, 4, 5, 6, 8, 10, 20, 40, 60, 80, 100].
   * The activation function used is ReLU (Rectified Linear Unit), and the solver for weight optimization is Adam, which is known for its efficiency and good convergence properties.
3. Model Evaluation:
   * Accuracy: The primary metric used to evaluate model performance is accuracy, which is the proportion of correctly classified instances in the testing set.
   * Precision, Recall, and F1-Score: These metrics provide a more nuanced understanding of the model’s performance, particularly in the presence of class imbalance. Precision measures the accuracy of positive predictions, recall measures the ability of the model to identify all positive instances, and F1-score is the harmonic mean of precision and recall.
   * Confusion Matrix: A confusion matrix is used to visualize the performance of the classification model by displaying the true positives, true negatives, false positives, and false negatives.

**Results and Analysis**

A graph with a red line

Description automatically generated*Figure 3.Accuracy score by hidden layer sizes*

For small hidden layer sizes (1, 2, 3), the accuracy scores are relatively low (around 0.2 to 0.35), suggesting that the neural network may not capture the complexity of the data adequately with fewer hidden layers.

As the hidden layer size increases (4 to 10), there is an improvement in accuracy, with scores ranging from 0.35 to 0.5. This indicates that adding more hidden layers allows the model to learn more complex patterns in the data, leading to better predictive performance.

However, beyond a certain point (around hidden layer size 10), the accuracy scores start to fluctuate and may not show significant improvement, as seen with hidden layer sizes 20, 40, 60, 80, and 100.

**CONCLUSION**

Based on the analysis, it is evident that the choice of hidden layer size plays a crucial role in determining the accuracy of the neural network model.

A moderate increase in hidden layer size (from 1 to 10) leads to improved accuracy, indicating that a neural network with a sufficient number of hidden layers can better capture the underlying patterns in the data.

However, increasing the hidden layer size beyond a certain threshold does not necessarily guarantee a proportional increase in accuracy. It is essential to find a balance between model complexity (hidden layer size) and performance (accuracy) to avoid overfitting and computational overhead.

**REFERENCES**

[1] S. E. Fahlman and C. Lebiere, “The cascade-correlation learning architecture,” Neural Networks, vol. 2, no. 5, pp. 523–545, 1989.

[2] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” Nature, vol. 521, no. 7553, pp. 436–444, 2015.