Jurnal Teknik Informatika Vol. 18 No. 1, April 2025 (64-76)
ISSN: p-ISSN 1979-9160 (Print)| e-ISSN 2549-7901 (Online)
DOI: https://doi.org/10.15408/jti.v18i1.44286

JURNAL TEKNIK INFORMATIKA

Homepage : http://journal.uinjkt.ac.id/index.php/ti

Online Shop Product Sales Prediction Using Multilayer Perceptron Algorithm

Erica Rian Safitri[1*], Lili Tanti[2], Wanayumini[3]

[1,2]Computer Science, Computer Science and Engineering, Potensi Utama University
[3]Computer Science, Computer Science and Engineering, Asahan University
[1,2]Jl. K.L. Yos Sudarso KM 6,5 No. 3A Tj. Mulia, Indonesia
[3]Jl. Jend.A.Yani, Kisaran Naga, Kec. Kota Kisaran Timur, Kisaran, Sumatera Utara, Indonesia

Article:
Accepted: March 17, 2025
Revised: December 20, 2024
Issued: April 30, 2025

***Correspondence Address:**
ericariansafitri38@gmail.com

ABSTRACT

This study aims to develop a predictive model for forecasting product sales using the Multilayer Perceptron (MLP) algorithm. The model's performance was evaluated using key metrics, including the Mean Absolute Error (MAE), Mean Squared Error (MSE), and R^2 score. The model achieved an MAE of 0.861, an MSE of 9.521, and an impressive R^2 score of 0.999, demonstrating its ability to accurately predict product sales with minimal error. Feature correlation analysis identified key variables related to the target prediction, which is the number of products ready for shipment, underscoring the importance of feature selection in enhancing model performance. Prediction results revealed variability among product sales, with products like Foodpak Matte 245 (Code 49) predicted to sell approximately 244.31 units, while others like Stiker Kertas (Code 90) showed lower sales forecasts. The findings suggest that strategic interventions may be necessary to boost sales for underperforming items and capitalize on the demand for popular products. Future improvements, such as optimizing the network architecture, experimenting with activation functions and optimization algorithms, and incorporating external factors such as market trends, could further enhance the model's accuracy and predictive power. Overall, the MLP model demonstrates strong potential for product sales forecasting, providing valuable insights for business decision-making.

Keywords : *multilayer perceptron (MLP); leaky ReLU; ADAM; product sales.*

Jurnal Teknik Informatika Vol. 18 No. 1, April 2025 (64-76)
ISSN: p-ISSN 1979-9160 (Print)| e-ISSN 2549-7901 (Online)
DOI: https://doi.org/10.15408/jti.v18i1.44286

1. INTRODUCTION

Artificial Neural Networks (ANNs) are computational models inspired by biological neural systems, extensively utilized to model, identify patterns, and analyze complex data, making them foundational in many artificial intelligence applications [1]. These networks consist of artificial neurons connected by adjustable weights, allowing for learning through iterative processing [2]. Each neuron processes input signals and produces an output passed to the next layer [3]. Multilayer neural networks, or Multilayer Perceptrons (MLP), are structured with three essential layers: the input layer, which represents the dataset variables; the hidden layers, which learn abstract patterns from the input data; and the output layer, which generates the prediction [4].

In the modern digital era, sales prediction has become a critical component for the e-commerce industry [5]. The proliferation of internet users and the widespread adoption of technology have established online shopping as a cornerstone of the global economy [6]. E-commerce platforms play a crucial role, facilitating transactions across various scales, from small businesses to large enterprises. Accurate sales forecasting, therefore, becomes essential for informed business decision-making [7].

E-commerce leverages internet technology for buying and selling products and services online, enabling transactions without geographical limitations [8]. As e-commerce evolves, sales prediction technologies have advanced significantly. Initially, traditional statistical methods such as linear regression and time series analysis were used for sales forecasting [9]. While these methods offered useful insights, they often struggled to capture the complexity and nuanced patterns of online sales data. The advent of machine learning approaches has led to more sophisticated methods for sales forecasting. Among these, the Multilayer Perceptron (MLP) has gained prominence due to its ability to identify nonlinear relationships, offering a more effective solution for enhancing the accuracy of online sales predictions and providing a competitive edge to e-commerce businesses [10].

A key study in this area is the work by [11], titled “Sportswear Retailing Forecast Model Based on the Combination of Multi-layer Perceptron and Convolutional Neural Network.” This research proposed a hybrid sales forecasting model combining MLP and Convolutional Neural Network (CNN) for the sportswear retail industry. The study demonstrated that this hybrid model outperformed traditional forecasting methods like ARIMA, as it accounted for various sales-influencing factors, including promotional strategies and product characteristics.

Although the results were promising, the study's scope was limited to the sportswear retail industry, which has unique features such as seasonality and fashion trends, which may differ from other e-commerce sectors. Additionally, the study utilized a large dataset from three local stores, whereas smaller online shops, like Ouboli Official, may not have the same volume of historical data. Thus, adapting the model for smaller datasets and varying product characteristics becomes essential.

This research aims to develop a sales prediction model using MLP, tailored for online shops with limited datasets. The primary objective is to improve the model’s performance by replacing the ReLU activation function with Leaky ReLU, which helps retain neuron activation even with negative inputs, enhancing the learning process [12]. Additionally, the research will use the ADAM optimizer, renowned for its efficiency in adjusting the learning rate adaptively, ensuring optimal parameter values for the neural network model, particularly in configuring hidden layers and neurons [13].

By applying the sales dataset from Kaggle, this approach seeks to improve the accuracy of product sales predictions for online shops [14]. The research will contribute to more precise business decisions and provide insights into applying MLP models in e-commerce contexts with limited datasets. This is particularly beneficial for new or small-scale online shops in Indonesia, helping them forecast product demand and optimize inventory management.

Key components of MLP include several critical elements. Layers consist of one or more neuron layers, with the input layer receiving the data and the output layer producing predictions. Hidden layers process internal representations

Jurnal Teknik Informatika Vol. 18 No. 1, April 2025 (64-76)
ISSN: p-ISSN 1979-9160 (Print)| e-ISSN 2549-7901 (Online)
DOI: https://doi.org/10.15408/jti.v18i1.44286

of the input data [15]. The strength of MLP lies in its ability to model complex and nonlinear relationships between input and output [16], enabling it to solve intricate prediction problems and uncover patterns in online sales data that traditional methods may overlook. MLP, for example, can capture complex and dynamic customer shopping patterns [17] and predict product sales trends more accurately [18]. Consequently, MLP is widely used for prediction tasks, such as online sales forecasting [19], market analysis, and personalized customer experiences.

Several studies have demonstrated the successful implementation of MLP for sales prediction with good accuracy, showcasing its potential to improve online shop performance [20]. However, these studies highlight challenges, including limited data availability and the complexities of tuning algorithm parameters. This research focuses on improving prediction results through feature selection and data-level approaches, filtering relevant features to enhance the model's efficiency and accuracy.

2. METHODS

This research employs a quantitative approach using experimental methods. The primary objective is to develop a sales prediction model for online shop products using the Multilayer Perceptron (MLP) algorithm. This study will utilize historical sales data from an online shop to train and test the MLP model. Below is a flowchart illustrating the research process:

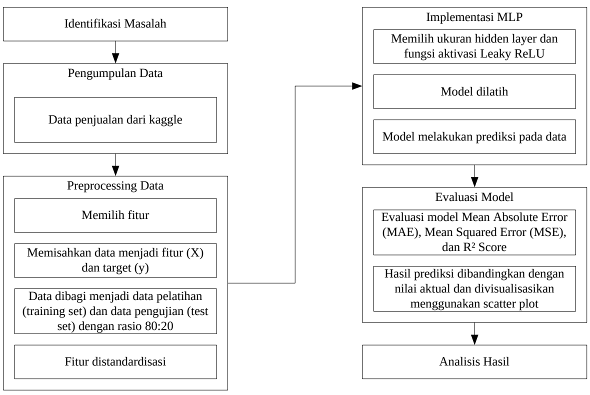

Figure 1. *Flowchart illustrating the research process*

The explanation of the research design process in Figure 1 is as follows :

2.1. Problem Identification

The goal of this research is to predict product sales using the Multilayer Perceptron (MLP) model, with the aim of enhancing marketing strategies and inventory management. This will be achieved by leveraging historical data, including the number of product page views, items added to the cart, orders placed, orders ready for shipment, and sales conversion rates. Key challenges in this study include the availability and quality of data, as well as the complexity of processing and developing an accurate predictive model.

2.2. Data Collection

This research uses primary data sourced from historical sales transactions available on Kaggle. The data, contained in the data_penjualan.csv file, includes the following key variables:

a. Sales Period

The date on which product transactions were recorded.

b. Product Type

The type of product sold during the transaction period.

c. Total Orders

The quantity of each product sold, aggregated to represent unique product entries.

d. Price

The unit price of each product sold.

e. Total Revenue

The total revenue generated from sales, calculated as the quantity sold multiplied by the price.

This data provides a detailed account of the sales process, from transactions to total revenue, and will undergo preprocessing to create suitable features for the MLP model.

2.3. Data Preprocessing

The data preprocessing steps involved are as follows :

a. Selecting Relevant Variables

The relevant variables for prediction include :

1. Date
2. Product Type
3. Order Quantity
4. Price
5. Total Revenue

ISSN: p-ISSN 1979-9160 (Print)| e-ISSN 2549-7901 (Online)
DOI: https://doi.org/10.15408/jti.v18i1.44286

b. Cleaning and Formatting Data

This step addresses inconsistencies in the dataset:

1. Replace instances of '-' with 0.
2. Convert columns to the appropriate data types (e.g., numerical columns should be numeric).
3. Standardize the format of the Conversion Rate column to ensure it is numeric.

c. Handling Missing or Incomplete Values

Missing values are filled in with appropriate defaults, such as 0, depending on the context.

After preprocessing, the dataset will be suitable for use in the prediction model as follows :

	Tanggal	Jenis Produk	Jumlah Order	Harga	Total
0	05/08/2022	Foodpak260	1000	1800	1800000
1	05/08/2022	FoodpakMatte245	1000	1900	1900000
2	05/08/2022	CraftLaminasi290	5000	750	3750000
3	05/08/2022	CraftLaminasi290	1000	1200	1200000
4	07/08/2022	Dupleks310	1000	1550	1550000
...	...	...	...	...	...
1071	14/11/2023	Ivory230	1000	900	900000
1072	15/11/2023	CraftLaminasi290	2000	800	1600000
1073	15/11/2023	CraftLaminasi290	1500	875	1312500
1074	15/11/2023	FoodpakMatte	1000	2200	2200000
1075	15/11/2023	GreaseProof	1000	300	300000

1076 rows × 5 columns

Figure 2. *Product sales dataset*

2.4. MLP Implementation

The implementation of the Multilayer Perceptron (MLP) model follows several key steps:

a. Model Instance Creation

The MLP model is created using a deep learning framework like TensorFlow or Keras. The Sequential() class from Keras is used to build the model layer by layer.

b. Input Layer Addition

The input layer is added to the model to receive feature data from the dataset. The Dense layer is used to connect neurons, with input_dim=6, which indicates six input features. The ReLU activation function is applied to introduce non-linearity to the model.

c. Hidden Layer Addition

Hidden layers are added to allow the model to process data further. The second hidden layer consists of 32 neurons, with the ReLU activation function, which helps the model learn non-linear patterns. Additionally, Leaky ReLU is employed to address the "dying ReLU" issue, where some neurons become inactive. This activation function allows a small, non-zero gradient when the input is negative, preventing neurons from becoming inactive and improving model performance.

d. Output Layer Addition

The output layer is added to produce the final predictions, typically with one neuron for regression tasks. The layer is specified as Dense(1, activation='linear'), which is suitable for continuous value predictions.

e. Weight Initialization

Weights are initialized for each layer and will be optimized during training to minimize prediction errors. Proper initialization is crucial for effective learning.

Following model initialization, the model is compiled with the following components:

a. Loss Function

Mean Squared Error (MSE) is used as the loss function for regression tasks to minimize the error between predicted and actual values.

b. Optimizer

The Adam optimizer is employed, which adapts the learning rate during training to improve efficiency and ensure optimal performance.

c. Evaluation Metrics

Metrics such as Mean Absolute Error (MAE) or MSE are used to evaluate model performance during both training and testing.

2.5. Model Evaluation

Model evaluation is an essential step to assess the performance of the trained model using the sales dataset from Ouboli Official. The evaluation process involves the following steps:

a. Prediction on Test Data (y_pred_relu)

After training the model with the training data (X_train, y_train), predictions are made on the test data (X_test), stored in the variable y_pred_relu, and compared with actual values (y_test) to measure accuracy.

b. Calculating Mean Absolute Error (MAE)

MAE measures the average absolute difference between predicted and actual values. A lower MAE indicates higher accuracy.

c. Calculating Mean Squared Error (MSE)

MSE is the average of the squared differences between predicted and actual values. Lower MSE values indicate better performance.

Jurnal Teknik Informatika Vol. 18 No. 1, April 2025 (64-76)
ISSN: p-ISSN 1979-9160 (Print)| e-ISSN 2549-7901 (Online)
DOI: https://doi.org/10.15408/jti.v18i1.44286

d. Calculating R^2 Score (Coefficient of Determination)

The R^2 score evaluates how well the model explains the variance in the target variable. A value close to 1 suggests strong explanatory power.

e. Visualizing Predictions vs. Actual Values

Scatter and line plots are used to compare predicted values with actual values, helping to identify error patterns and potential biases.

f. Experimenting with Leaky ReLU

Experiments with Leaky ReLU are conducted to compare model performance against the traditional ReLU function, with performance assessed using MAE and R^2 scores.

2.6. Results Analysis

After training and evaluating the model, it is used to make predictions based on new data :

a. Using the Model for Prediction

The trained model is applied to new data (X_new) to generate predictions. The model.predict method is used to produce output predictions, which are then analyzed.

b. Format of Prediction Output

The predictions are returned as an array of continuous values, representing the predicted sales figures for each input example.

c. Analysis of Prediction Results

The predicted values are compared with actual data to assess model accuracy. The prediction error is calculated by comparing predicted values with actual sales values.

d. Using Predictions for Decision-Making

The prediction results are used to inform decisions such as marketing strategies, inventory planning, and sales forecasting.

By applying this process, the trained MLP model provides valuable insights for decision-making, allowing for better predictions in real-world e-commerce scenarios.

3. RESULTS AND DISCUSSION

3.1. Dataset

This study utilizes a dataset consisting of product sales data from August 2023 to March 2024, collected from multiple Excel files. The dataset contains essential information such as product codes, product names, and the quantity of orders ready for shipment. These variables are used to analyze sales patterns and predict future product demand.

The data is split into two subsets: 80% for training and 20% for testing. The splitting process is handled using the train_test_split function from scikit-learn to ensure an unbiased selection of data. The dataset is normalized using StandardScaler to scale all features to a similar range, which is crucial for effective training of the Multi-Layer Perceptron (MLP) model. This preprocessing step prepares the data for accurate sales predictions based on historical patterns.

The code snippet for selecting and preprocessing the dataset is as follows:

```python
# Fungsi untuk memuat dan menggabungkan data dari beberapa file dalam folder
def load_data_from_folder(folder_path):
    all_files = glob(os.path.join(folder_path, "*.xlsx"))
    combined_data = pd.DataFrame()

    for file in all_files:
        data = load_data(file)
        if data is not None:
            month_year = extract_month_year_from_filename(file)
            data['Bulan_Tahun'] = month_year
            combined_data = pd.concat([combined_data, data], ignore_index=True)

    return combined_data
```

The load_data_from_folder(folder_path) function is responsible for loading and merging datasets from multiple Excel files within a specified folder. Initially, it uses glob to find all Excel files with a .xlsx extension in the given folder. All files are stored in the all_files variable. An empty DataFrame called combined_data is then created to hold the results of the merging process. For each file in the all_files list, the load_data(file) function is called to load the data. If the data is successfully loaded(notNone),theextract_month_year_from _filename(file) function is used to extract the month and year from the filename. This information is then added as a new column named Bulan_Tahun in the data DataFrame. Finally, the processed data DataFrame is merged into combined_data using pd.concat, allowing for the merging of multiple DataFrames without data loss. At the end of the function, the combined dataset is returned as the combined_data DataFrame, ready for use in the next stages.

Jurnal Teknik Informatika Vol. 18 No. 1, April 2025 (64-76)
ISSN: p-ISSN 1979-9160 (Print)| e-ISSN 2549-7901 (Online)
DOI: https://doi.org/10.15408/jti.v18i1.44286

	Tanggal	Jenis Produk	Jumlah Order	Harga	Total
0	05/08/2022	Foodpak260	1000	1800	1800000
1	05/08/2022	FoodpakMatte245	1000	1900	1900000
2	05/08/2022	CraftLaminasi290	5000	750	3750000
3	05/08/2022	CraftLaminasi290	1000	1200	1200000
4	07/08/2022	Dupleks310	1000	1550	1550000
...	...	...	...	...	...
1071	14/11/2023	Ivory230	1000	900	900000
1072	15/11/2023	CraftLaminasi290	2000	800	1600000
1073	15/11/2023	CraftLaminasi290	1500	875	1312500
1074	15/11/2023	FoodpakMatte	1000	2200	2200000
1075	15/11/2023	GreaseProof	1000	300	300000

1076 rows × 5 columns

Figure 3. *View of selected dataset*

3.2. Data Preprocessing

Data preprocessing is a critical step in preparing the dataset for the prediction model. In this study, the key features selected for analysis and prediction include:

a. Product Code

A unique identifier for each product.

b. Product Visitors (Add to Cart)

The number of visitors who added a product to their cart, indicating interest.

c. Added to Cart (Products)

The number of products added to the cart, reflecting purchase intent.

d. Total Buyers (Orders Created)

The total number of buyers who placed orders, showing conversion from interest to purchase.

e. Products (Orders Created)

The number of units ordered in confirmed purchases.

f. Total Buyers (Orders Ready for Shipment)

The total number of buyers whose orders are ready for shipment, signaling completed transactions.

g. Products (Orders Ready for Shipment)

The number of units ready for shipment, which serves as the target feature for the prediction model.

The data is standardized using the StandardScaler and split into training (80%) and testing (20%) subsets. The code for preprocessing is provided below :

```
selected_features = numeric_columns[:7]
X = data[selected_features]
target_column = 'Produk (Pesanan Siap Dikirim)'
if target_column not in data.columns:
    print(f"Kolom target '{target_column}' tidak ditemukan.")
    exit()
y = data[target_column]

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

scaler = StandardScaler()
X_train = scaler.fit_transform(X_train)
X_test = scaler.transform(X_test)
```

The results of the data preprocessing are shown below :

	Jenis Produk	Jumlah Order	Harga	Total
0	245	1000	1000.000000	1000000
1	260Glossy	2000	2300.000000	4600000
2	BOWL800ML	1000	2800.000000	2800000
3	Bowl650mlCIS	5000	1700.000000	8500000
4	CEMERLANGINDAHSELARAS	3000	1850.000000	5550000
...	...	...	...	...
89	StikerA3	250	5500.000000	1375000
90	StikerKertas	152	10041.333333	435376
91	Unbleached	2000	1000.000000	2000000
92	Unbleaching	2000	400.000000	800000
93	ivory230	1000	1200.000000	1200000

94 rows × 4 columns

Figure 4. *Preprocessed dataset result*

3.3. MLP Model Implementation with Leaky ReLU and ADAM

The MLP model is implemented with two hidden layers. Each hidden layer uses Leaky ReLU activation, which helps address the vanishing gradient problem by allowing small gradients for negative values. The model is optimized using the ADAM (Adaptive Moment Estimation) algorithm, known for its fast convergence and adaptive learning rate. The architecture comprises two hidden layers with 64 and 32 neurons, respectively, and the output layer uses a linear activation function to predict continuous values (units ready for shipment). The model is compiled with Mean Squared Error (MSE) as the loss function and Mean Absolute Error (MAE) as the evaluation metric. Training occurs over 100 epochs, using the preprocessed training data, while the model's performance is evaluated on the separate testing dataset.

3.4. Model Evaluation Results

Several metrics are used to assess the model's performance:

a. MAE

The average absolute difference between actual and predicted values, indicating the average prediction error.

b. MSE

The average squared difference between actual and predicted values, penalizing larger errors more significantly.

c. R^2 Score

Measures how well the model explains the variance in the target data, with values closer to 1 indicating better performance.

Jurnal Teknik Informatika Vol. 18 No. 1, April 2025 (64-76)
ISSN: p-ISSN 1979-9160 (Print)| e-ISSN 2549-7901 (Online)
DOI: https://doi.org/10.15408/jti.v18i1.44286

The evaluation results from the dataset are as follows :

1. MAE : 0.861
2. MSE : 9.521
3. R^2 Score : 0.999

The following code snippet is used for model evaluation:

```
# Evaluasi model
mae = mean_absolute_error(y_test, y_pred)
mse = mean_squared_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)

print(f"**Evaluasi Model:**")
print(f"- Mean Absolute Error (MAE): {mae}")
print(f"- Mean Squared Error (MSE): {mse}")
print(f"- R² Score: {r2}")
```

This code calculates the three evaluation metrics: MAE, MSE, and R^2 Score based on the model's predictions (y_pred) and actual values (y_test). The obtained values are then printed for analysis.

1. MAE is computed using the mean_absolute_error function from sklearn.metrics, assessing the average absolute error.
2. MSE is similarly calculated but emphasizes larger errors by squaring them.
3. R^2 Score assesses the model's fit in explaining the variance of the target data, with results close to 1 indicating a good fit.

3.5. Visualization of Prediction Results

The results of the model's predictions are visualized using two types of plots :

a. Scatter Plot

This plot illustrates the relationship between actual and predicted sales values. Points closer to the diagonal red line indicate more accurate predictions.

b. Residual Distribution

This plot shows the distribution of prediction errors (residuals), helping identify any systematic biases in the model.

Here is the code snippet used for visualizing the prediction results :

```
# Fungsi untuk visualisasi perbandingan antara prediksi dan nilai aktual
def visualize_predictions(y_test, y_pred):
    plt.figure(figsize=(10, 6))
    sns.scatterplot(x=y_test, y=y_pred)
    sns.lineplot(x=[y_test.min(), y_test.max()], y=[y_test.min(), y_test.max()], color='red')
    plt.xlabel("Nilai Aktual")
    plt.ylabel("Nilai Prediksi")
    plt.title("Perbandingan Nilai Aktual dan Prediksi Penjualan")
    plt.tight_layout()
    plt.show()

    plt.figure()  # Menambahkan jarak antara plot
    plt.figure(figsize=(10, 6))
    residuals = y_test - y_pred
    sns.histplot(residuals, kde=True, color='blue')
    plt.title('Distribusi Residual (Kesalahan Prediksi)')
    plt.xlabel('Residual')
    plt.ylabel('Frekuensi')
    plt.tight_layout()
    plt.show()
```

a. Scatter Plot Visualization This code uses the Seaborn library to create a scatter plot visualizing the relationship between actual (y_test) and predicted values (y_pred). The plot includes a diagonal red line representing "perfect prediction," where actual and predicted values match.

1. sns.scatterplot(x=y_test, y=y_pred) generates the scatter plot of actual and predicted values.
2. sns.lineplot() draws the diagonal reference line for perfect prediction.
3. Labels and titles are added to provide clear information about the axes and the content of the visualization.

The resulting scatter plot visualization is shown below:

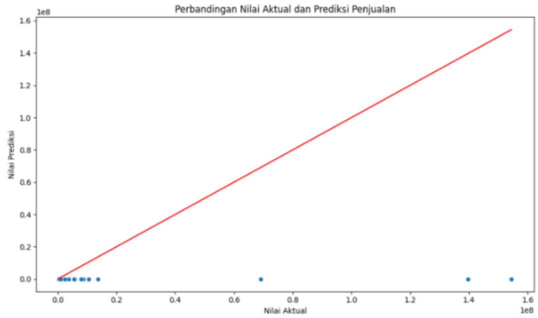

Figure 5. *Scatter plot visualization*

b. Residual Distribution Visualization This code visualizes the distribution of residuals to evaluate prediction errors. Residuals are calculated by subtracting predicted values from actual values. Using sns.histplot(), the residual distribution is plotted to examine for any biases or patterns in prediction errors.

1. residuals = y_test - y_pred computes the residuals between actual and predicted values.
2. sns.histplot(residuals, kde=True) creates a histogram of the residual distribution with Kernel Density Estimation (KDE) to more clearly show the pattern of residual distribution.

Jurnal Teknik Informatika Vol. 18 No. 1, April 2025 (64-76)
ISSN: p-ISSN 1979-9160 (Print)| e-ISSN 2549-7901 (Online)
DOI: https://doi.org/10.15408/jti.v18i1.44286

The resulting residual distribution visualization is shown below :

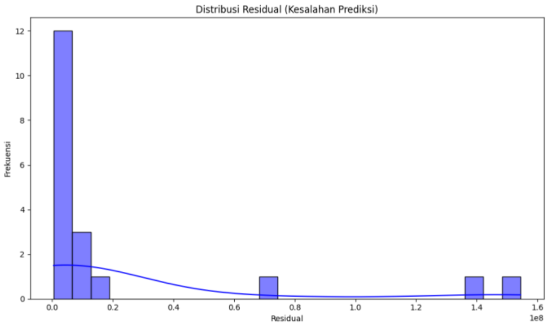

Figure 6. *Residual distribution visualization*

3.6. Feature Correlation Analysis

Correlation analysis is performed using the Pearson correlation coefficient to identify relationships between variables. A heatmap is used to visualize these relationships, with darker shades indicating stronger correlations. Here is the code snippet used for performing feature correlation analysis and its visualization:

```
# Visualisasi korelasi fitur hanya pada kolom numerik
def visualize_feature_correlation(data):
    plt.figure(figsize=(12, 8))
    numeric_data = data.select_dtypes(include=[np.number])  # Hanya kolom numerik
    correlation = numeric_data.corr()
    sns.heatmap(correlation, annot=True, cmap='coolwarm', linewidths=0.5)
    plt.title("Matriks Korelasi Fitur")
    plt.show()
```

a. Selection of Numeric Columns The function select_dtypes(include=[np.number]) is used to select only the columns containing numeric data from the dataset. This is important because Pearson correlation analysis is only relevant for numeric variables.

b. Creation of Correlation Matrix The correlation matrix is calculated using the corr() function applied to the selected numeric data. This function computes the Pearson correlation coefficient for all pairs of numeric features in the dataset.

c. Visualization with Heatmap The resulting correlation matrix is visualized using a heatmap created with the Seaborn library. This heatmap visually displays the relationships among features, where darker or lighter colors indicate the strength of positive or negative correlations. The annot=True argument displays the correlation coefficient values in each cell, allowing users to see the exact correlation values.

1. cmap='coolwarm': This color scheme helps distinguish between positive correlations (red) and negative correlations (blue), making correlation patterns easier to identify.
2. linewidths=0.5: This adds spacing between the cells in the heatmap for better clarity.

The following visualization illustrates the feature analysis based on the evaluation results from the dataset :

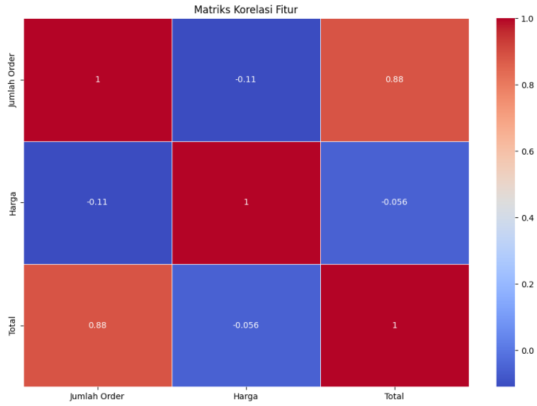

Figure 7. *Feature analysis visualization*

3.7. Dataset Discussion

The dataset used in this study consists of product sales data collected from multiple monthly files. After merging the data, it provides a comprehensive view of sales performance. Key characteristics of the dataset include:

a. Size

The dataset is large enough to represent diverse sales patterns across time.

b. Structure

It includes critical columns such as product codes, names, and consumer interaction metrics.

c. Temporal Variability

The addition of a Month-Year column allows for trend analysis over time.

d. Completeness

The merging process ensures that all relevant data is included, enhancing the dataset's representativeness.

Jurnal Teknik Informatika Vol. 18 No. 1, April 2025 (64-76)
ISSN: p-ISSN 1979-9160 (Print)| e-ISSN 2549-7901 (Online)
DOI: https://doi.org/10.15408/jti.v18i1.44286

3.8. Data Preprocessing Discussion

The data preprocessing results indicate that the dataset is adequately prepared for predictive modeling. After feature selection and standardization, the final dataset comprises key features relevant to predicting product sales. The data has undergone several adjustments and cleaning stages to ensure readiness for model training. Below is a summary of the preprocessing results shown in Figure IV.3:

a. Key Features Utilized The final dataset includes seven essential features critical for product sales prediction:
   1. Product Code: This serves as a unique identifier for each product, though it is not directly used in model training.
   2. Product Visitors (Added to Cart) and Added to Cart (Product): These metrics reflect consumer interest and purchase intent, with higher values signaling stronger sales potential.
   3. Total Buyers (Orders Made) and Products (Orders Made): These features capture the number of buyers and total products ordered.
   4. Total Buyers (Orders Ready for Shipment) and Products (Orders Ready for Shipment): These features indicate completed transactions ready for dispatch, with the latter being the prediction target.
b. Feature Normalization A crucial preprocessing step was the normalization of numerical features to ensure they are on a similar scale. This is particularly important for neural network models like the Multilayer Perceptron (MLP), as it prevents any single feature from dominating the model's learning process. The results show that the numerical features are evenly distributed, with minimal outliers or high-value discrepancies.
c. Dataset Splitting After normalization, the dataset was split into two subsets: 80% for training and 20% for testing. This division ensures that the model is trained on data reflecting actual sales patterns and is evaluated on unseen data, which is critical for assessing its ability to generalize.
d. Data Completeness The dataset was thoroughly checked for missing or irrelevant data, and it was confirmed that no such issues remain. Every row in the final dataset contains complete information on the selected features, ensuring the model can proceed without data deficiencies during training.

The final dataset is now well-structured, with normalized features and no missing values, making it ready for model training. The preprocessing steps, including feature selection, normalization, and dataset division, ensure that the model can efficiently learn from historical sales patterns and provide reliable future predictions.

3.9. Implementation of MLP Model with Leaky ReLU Activation Discussion

This study employs a Multilayer Perceptron (MLP) model with two hidden layers to predict sales, using the Leaky ReLU activation function. Leaky ReLU allows negative gradients, which helps prevent the vanishing gradient problem and improves model stability during training. The ADAM optimizer, which adapts the learning rate during training, was used to efficiently handle large and complex datasets like sales data.

The architecture consists of two hidden layers: the first with 64 neurons and the second with 32 neurons, with Leaky ReLU applied to both. The output layer employs a linear activation function, reflecting the continuous nature of the prediction target-the number of products ready for shipment. The model is compiled using Mean Squared Error (MSE) as the loss function, appropriate for regression problems, and Mean Absolute Error (MAE) as an additional metric.

The model is built using the Keras library, where layers are added sequentially. The input layer corresponds to the number of features in the training data. The Leaky ReLU activation function has a negative slope of 0.01 to address the vanishing gradient issue.

The model is trained for 100 epochs using the preprocessed data. It is then evaluated on the test set to assess its predictive capabilities. This MLP model aims to offer accurate sales predictions that are valuable for business decision-making.

Jurnal Teknik Informatika Vol. 18 No. 1, April 2025 (64-76)
ISSN: p-ISSN 1979-9160 (Print)| e-ISSN 2549-7901 (Online)
DOI: https://doi.org/10.15408/jti.v18i1.44286

3.10. Model Evaluation and Algorithm Efficiency

The model's performance was evaluated using three key metrics: Mean Absolute Error (MAE), Mean Squared Error (MSE), and R^2 Score. These metrics offer different perspectives on the model's accuracy and its ability to fit the data.

a. Mean Absolute Error (MAE)

The model achieved an MAE of 0.86096, indicating a low average error in the predictions.

b. Mean Squared Error (MSE)

The MSE was 9.52058, suggesting that while the model produces small errors overall, it does not suffer from large prediction discrepancies.

c. R^2 Score

With an R^2 Score of 0.99883, the model explains almost all the variance in the target data, reflecting a strong fit.

While the results indicate excellent performance in terms of accuracy (low MAE, low MSE, and high R^2), further analysis is needed to address the negative R^2 observed in some areas. It suggests that certain aspects of the data might not be fully captured by the model, which could be due to issues such as insufficient feature selection or overfitting.

3.11. Visualization of Results and Model Validation

To assess the model's performance visually, scatter plots and residual distributions were used.

a. Scatter Plot Visualization

The scatter plot (Figure 5) shows the actual sales values on the x-axis and the predicted values on the y-axis. Most data points cluster close to the reference line, indicating good predictive accuracy. Outliers represent larger prediction errors, providing areas for potential model improvement.

b. Residual Distribution Visualization

The residuals were examined to determine if they are normally distributed. The distribution (Figure 6) reveals that residuals are evenly distributed around zero, indicating that the model does not exhibit bias and is likely to make reliable predictions.

These visualizations confirm that the model performs well overall, although areas for improvement exist. A deeper understanding of why certain predictions deviate may help refine the model.

3.12. The Role of Feature Correlation in Prediction

Figure 7 illustrates a correlation matrix showing the relationships between features and the target variable. The heatmap reveals which features are most influential in predicting sales:

a. Positive Correlation

Features strongly positively correlated with sales predictions are marked in dark red. These features have a direct, positive impact on product sales and should be prioritized in the model.

b. Negative Correlation

Features with a strong negative correlation are shown in dark blue. While counterintuitive, these features provide additional insights and can reveal patterns that might otherwise be overlooked.

c. Low Correlation

Features with minimal correlation to sales (lighter colors) are less relevant and may be excluded from the model to improve efficiency and reduce complexity.

The correlation analysis supports feature selection, ensuring that only the most impactful variables contribute to the model's performance.

3.13. Prediction Results

The prediction results obtained from the MLP model demonstrate its capability to forecast the number of products ready for shipment with high accuracy. The evaluation metrics indicating excellent performance, combined with supportive visualizations, reinforce the effectiveness of this approach in providing valuable insights for business decision-making. Thus, this research successfully validates the potential of machine learning techniques in sales analysis and forecasting for the online shop. After training the model using the sales dataset, the generated prediction results are presented as follows :

	Jenis Produk	Prediksi Total
49	FoodpakMatte245	244.306549
72	Ivory270	172.247665
26	Dupleks270	148.208939
55	GreaseProof40	123.830490
12	Craft290Laminasi8	111.298302
39	Foodpak245Matte	110.856705
0	245	110.543518
18	CraftFoodpak2900	108.695709
22	CraftLaminasi8	108.695709
62	Ivory210	107.524406
10	Craft290Foodpak	107.524406
40	Foodpak260	104.058044
83	Kinstruk260	103.659668
35	Dupleks350P	103.191452

ISSN: p-ISSN 1979-9160 (Print)| e-ISSN 2549-7901 (Online)
DOI: https://doi.org/10.15408/jti.v18i1.44286

The prediction results from the MLP model demonstrate its ability to forecast the number of products ready for shipment with high accuracy. The model's effectiveness is reinforced by the strong evaluation metrics and supporting visualizations.

The sales prediction table provides insights into expected sales for various products. The table includes Product Code, Product Name, and Sales Prediction columns, which help in understanding product performance.

a. High Sales Prediction for Popular Products

For instance, Product Code 49 (Foodpak Matte) has a high sales prediction of 244.31 units, indicating strong demand.

b. Sales Variability Across Products

Products show significant variation in predicted sales, such as Product Code 55 (Grease Proof 40) with 123.83 units and Product Code 90 (Stiker Kertas) with 55.21 units, reflecting differing market interests.

c. Implications of Negative Predictions

Although negative predictions were not observed, any such predictions in future models would indicate potential risks, urging a review of marketing strategies or inventory levels.

The insights from these predictions enable better resource allocation, inventory management, and targeted marketing strategies, helping businesses make data-driven decisions.

CONCLUSION

The Multilayer Perceptron (MLP) model developed in this study demonstrates strong performance in predicting product sales, as evidenced by the evaluation metrics. With a Mean Absolute Error (MAE) of 0.861 and a Mean Squared Error (MSE) of 9.521, the model exhibits minimal prediction errors, indicating reliable forecast accuracy. Furthermore, the R^2 score of 0.999 suggests that the model effectively explains the variation in the target data, highlighting its high predictive capability and the relevance of the features used for prediction.

The correlation analysis reveals that certain features are strongly related to the target prediction, which is the number of products ready for shipment. This underscores the importance of feature selection in refining the model. Features with strong correlations help the model better grasp sales patterns, thereby improving its prediction accuracy. The prediction results show that products with higher sales potential, such as the Foodpak Matte 245 (Product Code 49), are predicted to sell approximately 244.31 units, while other products, like Ivory 270 (Code 72) and Dupleks 270 (Code 26), are forecasted to sell 172.25 and 148.21 units, respectively, suggesting healthy demand for these items. Conversely, some products, like Stiker Kertas (Code 90), are predicted to sell only 55.21 units, indicating that not all products will experience the same level of demand. This variability in predictions suggests that strategic interventions may be necessary to boost sales for underperforming products and maximize sales of popular items.

To further improve the model’s performance, fine-tuning the network architecture is recommended, including adjustments to the number of neurons and hidden layers. Experimenting with different activation functions and optimization algorithms could help identify the most effective configuration. Implementing cross-validation would assess the model's robustness and generalization across different datasets, ensuring that it performs well not just on the training data but also on new, unseen data. Additionally, incorporating external factors such as market trends, competitor data, and seasonal influences could provide valuable insights, improving the accuracy of sales predictions. Regular performance evaluations using updated datasets will ensure that the model remains relevant and reflective of changing market conditions and consumer behavior.

In conclusion, the MLP model shows promising results in predicting sales trends, with the current performance metrics indicating high accuracy and predictive strength. The findings underscore the importance of feature selection and model fine-tuning in optimizing sales forecasts. Future improvements, such as exploring different network architectures and integrating external data, could further enhance the model’s performance, making the predictions even more reliable and impactful for business decision-making.

ISSN: p-ISSN 1979-9160 (Print)| e-ISSN 2549-7901 (Online)
DOI: https://doi.org/10.15408/jti.v18i1.44286

REFERENCES

[1] H. Hartono and A. Zein, "PENERAPAN ALGORITMA GENETIKA DAN JARINGAN SYARAF TIRUAN DALAM PENJADWALAN MATA KULIAH Studi Kasus : Prodi Sistem Informasi Universitas Pamulang," *J. Ilmu Komput.*, vol. VI, no. 03, pp. 6–10, 2023.

[2] M. A. Mazurowski, M. Buda, A. Saha, and M. R. Bashir, "Deep learning in radiology: An overview of the concepts and a survey of the state of the art with focus on MRI," *J. Magn. Reson. Imaging*, vol. 49, no. 4, pp. 939–954, 2019, doi: 10.1002/jmri.26534.

[3] D. H. Lee, Y. T. Kim, and S. R. Lee, "Shallow landslide susceptibility models based on artificial neural networks considering the factor selection method and various non-linear activation functions," *Remote Sens.*, vol. 12, no. 7, 2020, doi: 10.3390/rs12071194.

[4] N. R. Sari and Y. Mar'atullatifah, "PENERAPAN MULTILAYER PERCEPTRON UNTUK IDENTIFIKASI KANKER PAYUDARA," *J. Cakrawala Ilm.*, vol. 2, no. 8, pp. 3261–3268, 2023, doi: 10.31862/9785426311961.

[5] F. Rahmawati and N. Merlina, "Metode Data Mining Terhadap Data Penjualan Sparepart Mesin Fotocopy Menggunakan Algoritma Apriori," *PIKSEL Penelit. Ilmu Komput. Sist. Embed. Log.*, vol. 6, no. 1, pp. 9–20, 2018, doi: 10.33558/piksel.v6i1.1390.

[6] C. O. Sakar, S. O. Polat, M. Katircioglu, and Y. Kastro, "Real-time prediction of online shoppers' purchasing intention using multilayer perceptron and LSTM recurrent neural networks," *Neural Comput. Appl.*, vol. 31, no. 10, pp. 6893–6908, Oct. 2019, doi: 10.1007/s00521-018-3523-0.

[7] A. Utku and M. A. Akcayol, "Deep Learning based prediction model for the next purchase," *Adv. Electr. Comput. Eng.*, vol. 20, no. 2, pp. 35–44, 2020, doi: 10.4316/AECE.2020.02005.

[8] G. Taher, "E-Commerce: Advantages and Limitations," *Int. J. Acad. Res. Accounting, Financ. Manag. Sci.*, vol. 11, no. 1, pp. 202–221, Feb. 2021, doi: 10.6007/IJARAFMS/v11-i1/8987.

[9] J. Tandean, R. Indrawan, I. Intan, and S. Ramadhani Arifin, "Pengaruh Penerapan Stochastic Gradient Descent Dan Adam Optimizer Pada Hyperparameter Tuning Untuk Klasifikasi Penyakit Tanaman Ubi Kayu," *J. Dipanegara Komput. Tek. Inform.*, vol. XVI, no. 1, pp. 80–90, 2023, [Online]. Available: https://www.ejurnal.dipanegara.ac.id/index.php/dipakomti/article/view/1377

[10] J. A. Prakash, V. Ravi, V. Sowmya, and K. P. Soman, "Stacked ensemble learning based on deep convolutional neural networks for pediatric pneumonia diagnosis using chest X-ray images," *Neural Comput. Appl.*, vol. 35, no. 11, pp. 8259–8279, 2023, doi: 10.1007/s00521-022-08099-z.

[11] J. Xu, Y. Zhou, L. Zhang, J. Wang, and D. Lefloch, "Sportswear retailing forecast model based on the combination of multi-layer perceptron and convolutional neural network," *Text. Res. J.*, vol. 91, no. 23–24, pp. 2980–2994, Dec. 2021, doi: 10.1177/00405175211020518.

[12] D. Sinha and M. El-Sharkawy, "Thin MobileNet: An Enhanced MobileNet Architecture," *2019 IEEE 10th Annu. Ubiquitous Comput. Electron. Mob. Commun. Conf. UEMCON 2019*, pp. 0280–0285, 2019, doi: 10.1109/UEMCON47517.2019.8993089.

[13] M. A. Hanin, R. Patmasari, and R. Y. Nur, "Sistem Klasifikasi Penyakit Kulit Menggunakan Convolutional Neural Network (Cnn) Skin Disease Classification System Using Convolutional Neural Network (Cnn)," *e-Proceeding Eng.*, vol. 8, no. 1, pp. 273–281, 2021.

[14] R. Poojary, R. Raina, and A. Kumar Mondal, "Effect of data-augmentation on fine-tuned CNN model performance," *IAES Int. J. Artif. Intell.*, vol. 10, no. 1, p. 84, Mar. 2021, doi: 10.11591/ijai.v10.i1.pp84-92.

[15] D. T. Tran, S. Kiranyaz, M. Gabbouj, and A. Iosifidis, "Heterogeneous Multilayer Generalized Operational Perceptron," *IEEE Trans. Neural Networks Learn.*

Syst., vol. 31, no. 3, pp. 710–724, Mar. 2020, doi: 10.1109/TNNLS.2019.2914082.

[16] M. Zohdi, M. Rafiee, V. Kayvanfar, and A. Salamiraad, "Demand forecasting based machine learning algorithms on customer information: an applied approach," *Int. J. Inf. Technol.*, vol. 14, no. 4, pp. 1937–1947, Jun. 2022, doi: 10.1007/s41870-022-00875-3.

[17] N. Chabane *et al.*, "Intelligent personalized shopping recommendation using clustering and supervised machine learning algorithms," *PLoS One*, vol. 17, no. 12, p. e0278364, Dec. 2022, doi: 10.1371/journal.pone.0278364.

[18] P. C. Yang, H. M. Wee, and H. J. Yang, "Global optimal policy for vendor–buyer integrated inventory system within just in time environment," *J. Glob. Optim. 2006 374*, vol. 37, no. 4, pp. 505–511, Aug. 2006, doi: 10.1007/S10898-006-9058-4.

[19] A. Salamzadeh, P. Ebrahimi, M. Soleimani, and M. Fekete-Farkas, "Grocery Apps and Consumer Purchase Behavior: Application of Gaussian Mixture Model and Multi-Layer Perceptron Algorithm," *J. Risk Financ. Manag.*, vol. 15, no. 10, 2022, doi: 10.3390/jrfm15100424.

[20] I. Kurniawan, L. S. Silaban, and D. Munandar, "Implementation of Convolutional Neural Network and Multilayer Perceptron in Predicting Air Temperature in Padang," *J. RESTI (Rekayasa Sist. dan Teknol. Informasi)*, vol. 4, no. 6, pp. 2–7, 2020, doi: 10.29207/resti.v4i6.2456.