International Journal of Science and Research Archive

eISSN: 2582-8185

Cross Ref DOI: 10.30574/ijsra

Journal homepage: https://ijsra.net/

(Research Article)

Check for updates

AI-ML algorithm for enhanced performance management: A comprehensive framework using Backpropagation (BP) Algorithm

Sunil Basnet *

Chief Human Resource Officer (iCHRO), Virtuosway, Kathmandu, Nepal.

International Journal of Science and Research Archive, 2024, 11(01), 1111–1127

Publication history: Received on 28 November 2023; revised on 27 January 2024; accepted on 1 February 2024

Article DOI: https://doi.org/10.30574/ijsra.2024.11.1.0118

Abstract

In the era of economic globalization and heightened market competition, organizations face the imperative to establish robust performance evaluation mechanisms that drive both organizational development and individual employee motivation. This article delves into the multifaceted factors influencing employee performance, encompassing personal attributes, interpersonal relations, and work standards. The study takes a deep dive into the transformative integration of AI-ML algorithms, proposing a comprehensive framework for elevated performance management. Through the application of machine learning algorithms, this research seeks to revolutionize performance appraisals, impacting crucial HR processes such as employee selection, promotions, terminations, training initiatives, and remuneration adjustments. The investigation provides nuanced insights into the synergy between artificial intelligence, machine learning, and traditional performance evaluation methodologies, offering profound perspectives on contemporary organizational practices amid evolving challenges.

Keywords: Performance Management; Artificial Intelligence (AI); Machine Learning (ML); Algorithmic Evaluation; Human Resource Management

1. Introduction

The inception of Artificial Intelligence (AI) in the 1950s aimed at instilling machines with human-like intelligence, envisioning a future where advanced AI technologies not only create new employment opportunities and skill sets but also address societal challenges by significantly improving efficiency (McCarthy et al., 2006; Pillai & Sivathanu, 2020; Gherhes, 2018). Machine Learning (ML), a subset of AI, operates within computer science, focusing on extracting knowledge from data through statistical methods. Notably, Deep Learning (DL), an advanced form of ML, employs a hierarchical approach to transform information into intricate data representations (Goodfellow et al., 2016; LeCun et al., 2015).

In the domain of Human Resource Management (HRM), the integration of AI technologies has ushered in the era of electronic HR management (E-HRM) (Ma & Ye, 2015). Expanding on this, Jiang et al. (2019) delve into the potential of AI to enhance the efficiency and quality of human resource management, emphasizing its significance and relevant indicators. The impact of AI on HRM effectiveness is explored, addressing core challenges in its application.

As organizations navigate through this transformative era, the exploration of AI and Machine Learning assumes a pivotal role in reshaping conventional approaches to performance management. This article delves into the theoretical framework and empirical analysis of an AI/ML algorithm designed for optimizing performance management, offering mathematical equations for backpropagation and sample codes to provide practical insights into its potential applications and benefits in the contemporary organizational landscape.

* Corresponding author: Sunil Basnet

1.1. Statement of Problem

In contemporary organizational landscapes, the traditional methods of performance management often encounter challenges such as subjectivity, recency biases, and limitations in effectively harnessing diverse data sources. Cappelli and Tavis (2016) emphasized a central challenge in the intricate landscape of HR outcomes, particularly the nuanced definition of a "good employee." This concept encompasses multiple dimensions, and precise measurements, especially for diverse roles, prove to be a significant challenge. Traditional metrics such as performance appraisal scores, widely utilized for assessments, face considerable criticism due to issues related to validity, reliability, and bias. Consequently, many organizations are moving away from these conventional metrics. The complexity deepens when considering that numerous jobs are interdependent, making it challenging to discern individual performance from group dynamics (Pfeffer and Sutton, 2006).

As organizations strive for enhanced efficiency and more data-driven decision-making, there exists a critical need to integrate Machine Learning (ML) algorithms into performance management systems. This integration aims to address the limitations of conventional approaches and unlock the full potential of data analytics for comprehensive and objective performance assessments.

1.2. Research Objectives

This study aims to contribute to the evolving field of performance management, providing insights and solutions for organizations seeking to leverage the power of ML in optimizing their workforce's performance assessment processes.

- Design and implement a robust machine learning algorithm specifically tailored for performance management.
- Consider various performance metrics and contextual factors in the algorithm's design.
- Investigate methods to mitigate subjective biases inherent in traditional performance appraisal methods and ensure the developed ML algorithm promotes fair and objective evaluations.
- Explore the integration of diverse data sources, including employee productivity metrics, project outcomes, and feedback loops.
- Aim to create a holistic performance management framework that captures multifaceted aspects of employee performance.
- Evaluate the predictive capabilities of the ML algorithm to anticipate future performance trends.
- Assess how these insights can contribute to proactive decision-making and talent development strategies.
- Develop an intuitive and user-friendly interface for the ML-based performance management system.
- Prioritize ease of adoption and acceptance among end-users to ensure successful implementation.
- Investigate and implement ethical guidelines within the ML algorithm.
- Address concerns related to privacy, algorithmic transparency, and fairness in performance evaluations.

2. Literature Review

The evolution of Human Resource Management (HRM) gained momentum with the Michigan Matching Model and the Harvard Framework, particularly Boxall's redefinition in 1992. This broadened HRM beyond selection and compensation, emphasizing a strategic approach aligned with organizational goals. The Harvard Framework highlighted the need for alignment between HR systems, organizational structure, and overall strategy, suggesting that historical personnel issues could be addressed through proactive employee development.

Strategic HRM, according to Wright et. al. (2001), is vital for effective human resource allocation within organizations. Schuler (1994) stressed the dynamic integration of HRM with an organization's development strategy, emphasizing alignment with organizational structures and gaining acceptance among managers and employees.

Zhang et al. (2020) advocated for a strategic HRM approach that adapts to internal and external opportunities and threats, contributing to overall enterprise development. Becker and Gerhart (1996) introduced the concept of a high-performance work system within HRM, aiming to enhance job satisfaction and improve enterprise performance by combining motivation and ability.

2.1. Performance Management

The integration of performance management within an organization is imperative, providing valuable insights for crucial decision-making processes like promotions, merit raises, transfers, and training and development. Moreover, it extends beyond these tangible aspects to foster increased commitment and job satisfaction among employees (Wiese and Buckley, 1998). This comprehensive practice not only propels employee productivity, thereby enhancing overall organizational performance, but also nurtures professional growth by pinpointing areas for improvement (Sels et. al., 2003).

Research conducted by Nayab et al. (2011) emphasizes that the current competitive economic landscape and rapidly evolving external environment necessitate organizations to shift from reactive performance appraisals to proactive performance management. This transition seeks to amplify productivity and elevate overall organizational performance. However, Pareek and Rao (2006) argue that the focus should extend beyond mere performance appraisal, placing a more concentrated emphasis on defining, strategizing, and orchestrating performance. Transparent evaluation of performance, as highlighted by Singh (2004), emerges as a potent motivational tool, compelling employees to exert additional effort to achieve organizational goals.

The performance appraisal system stands out as a crucial instrument in assessing organizational work performance, determining the level of success or failure. Past research by Arora and Arora (2010), Aggarwal and Thakur (2013) has delved into various techniques of performance appraisal from both managerial and employee perspectives, evaluating each type for its advantages and disadvantages.

2.2. Artificial Intelligence in Performance Management

Artificial Intelligence (AI) has transformed the landscape of performance management, introducing continuous workforce monitoring and tailored feedback generation through the analysis of extensive datasets (Tong et al., 2020). At its core, AI embodies a system's ability to accurately interpret external data, learn iteratively, and accomplish specific objectives through adaptable adjustments (Haenlein & Kaplan, 2019).

The real-time analysis and rating capabilities brought by AI contribute to objectivity in data collection, amplifying its impact on performance management (Ewenstein et al., 2016). It's crucial to acknowledge that the effectiveness of AI is intricately tied to the quality of its training data, influencing the precision of algorithmic outcomes. The successful integration of AI-generated feedback stands to redefine performance management, contingent on organizational adoption and the recognition of its software potential by managers. The process of sensemaking, particularly in interpreting disruptive events, plays a pivotal role in facilitating this transition (Helms Mills et al., 2010). The perception of AI feedback by individuals, whether seen as an opportunity or a threat, significantly shapes its successful implementation.

AI applications demonstrate notable efficiency in monitoring workplace behavior, evaluating performance, and delivering personalized feedback, marking a significant departure from conventional approaches (Tong et al., 2020). By harnessing algorithms to identify patterns and classify data, AI facilitates well-informed decisions concerning employee recruitment, dismissal, and promotion, providing a distinct advantage in the realm of performance management (Kalischko & Riedl, 2021).

2.3. Artificial Neural Network (ANN)

The conceptual foundation of neural networks was established by McCulloch and Pitts in 1943, with the first operational neural network achieved by Farley and Clark in 1954.

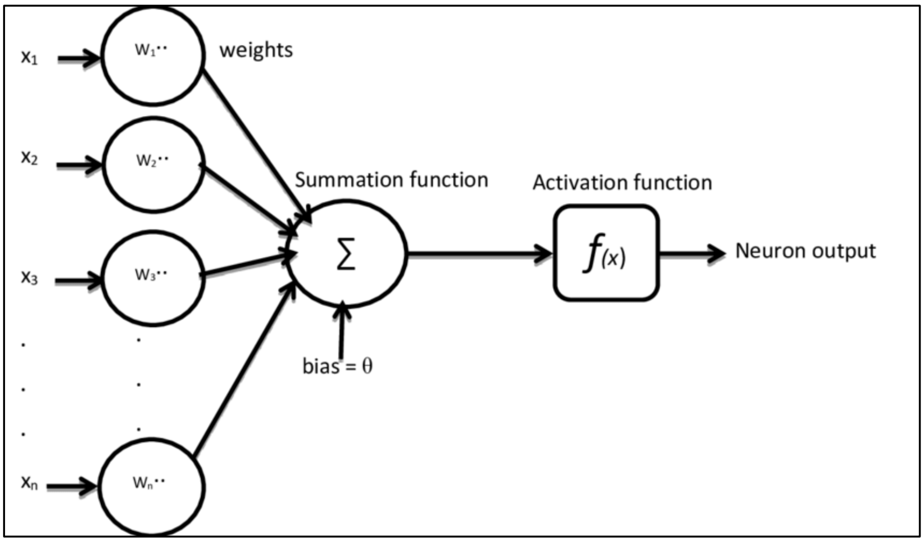
The mathematical representation of the artificial neuron is given by:

$$y_j=f(\sum_{i=1}^{n} w_{ij}x_i) \qquad (1)$$

Where,

- y_j: The output of the artificial neuron in the j^{th} layer.
- $f(\cdot)$: The activation function applied to the weighted sum of inputs.
- $\sum_{i=1}^{n} w_{ij}x_i$: The weighted sum of inputs, where w_{ij} is the connection weight between the i^{th} input and the j^{th} neuron, and x_i is the i^{th} input

Neural networks are renowned for their ability to emulate the pattern-recognition capabilities of the human brain, finding applications in diverse fields like investment decision-making, handwriting recognition, and bomb detection. Nicoletti (2000) defines neural networks as computational networks consisting mainly of massively parallel processing units that interact through weighted interconnections.



Source: Yacim and Boshoff (2018): online

Figure 1 A Simple Neural Network

Altay and Satman (2005) conducted a comparative analysis of forecasting performance, pitting artificial neural network (ANN) against linear regression strategies in the Istanbul Stock Exchange. The study presented compelling evidence supporting the statistical and financial superiority of ANN models. Roh (2007) proposed hybrid models integrating neural networks and time series models for forecasting stock price index volatility from both deviation and direction perspectives. Empirical results showcased the significant enhancement of predictive accuracy for both aspects using ANN-time series models.

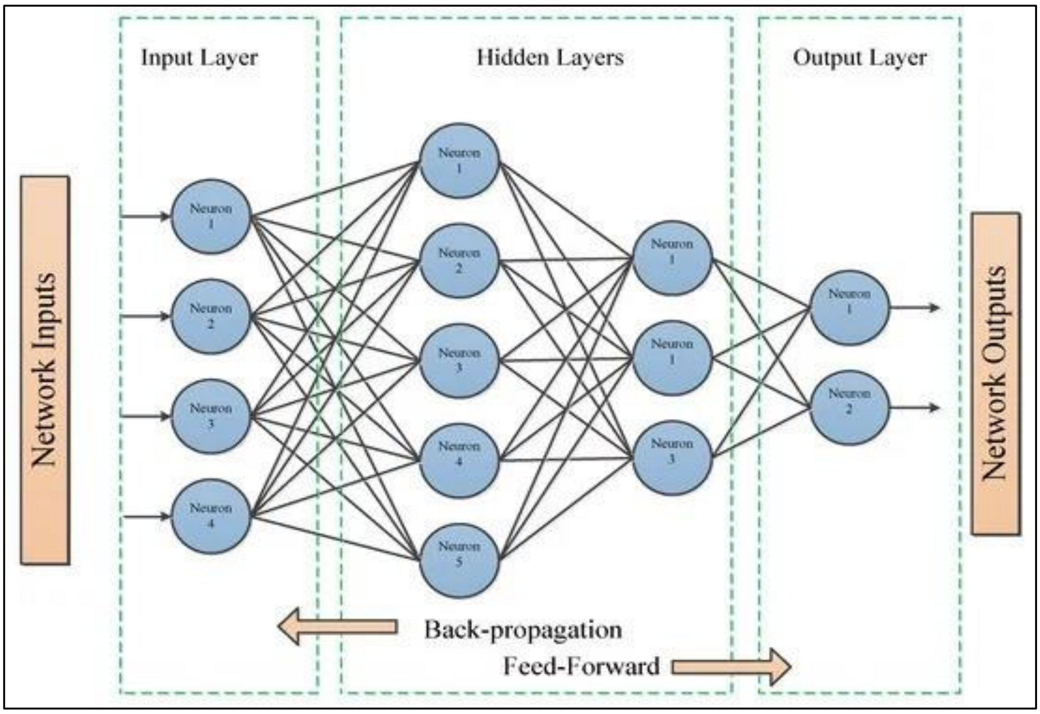
Richards (2006) outlines the organizational structure of neural networks into input, hidden, and output layers. A typical neural network architecture, aligned with the Multilayer Perceptron (MLP) model, comprises these three layers. The input layer distributes and scales inputs to the subsequent layer, the hidden layer processes inputs, and the single hidden layer usually suffices. The output layer provides class labeling information.

In a related study, Kumar and Thenmozhi (2006) investigated the effectiveness of models, including ARIMA, ANN, SVM, and random forest regression, in predicting and trading the S&P CNX NIFTY Index return. The Probabilistic Neural Network (PNN) exhibited superior predictive power over generalized methods of moments (GMM) with Kalman filter and random walk prediction models. In another context, Liao and Wang (2010) applied Stochastic Time Effective Neural Networks to predict the China global index, demonstrating superior predictive accuracy over traditional regression models.

2.4. Back Propagation (BP) Algorithm

The backpropagation (BP) algorithm is a cornerstone in the realm of neural network training, playing a pivotal role in optimizing the weights of connections between neurons. Originally proposed by Paul Werbos in the 1970s and later rediscovered by Rumelhart and McClelland in 1986, BP has become widely adopted for its effectiveness in supervised learning tasks. This algorithm relies on the principles of error-correction learning, utilizing an error function to iteratively adjust weights and minimize the disparity between actual and desired network outputs.

Training a neural network involves adjusting its internal parameters, known as weights, to minimize the difference between predicted and actual outcomes. In this section, we delve into the mathematics behind one of the foundational algorithms for neural network training—Backpropagation. Through rigorous derivations, we aim to demystify the intricacies of weight updates in hidden and output layers, shedding light on the principles that govern the learning process



Source: Abdolrasol et. al., 2021: Online

Figure 2 Sample illustration of Artificial Neural Network with feed forward and Back Propagation Algorithm

2.5. Previous Studies

Nasr et al. (2019) predicted employee's performance using data from the Ministry of Egyptian Civil Aviation, through a questionnaire filled by 145 employees. Collected data included Rank, years of experience, Service period, number of period companies, salary range, working in comfortable conditions or not, satisfaction with salary, professional training, satisfaction with job, age, gender, marital status, educational degree, General Specialization, Type of the University and grade. Data mining techniques, Decision Tree (DT), Naïve Bayes, and Support Vector Machine (SVM) were implemented to build the classification model and choose the most influencing factors. An application model can be built from the several trials in order to support the human resources management.

Singh et al. (2023) assessed the prediction of employee performance and leave management using Bayesian Classification algorithm, on a self-collected dataset of 430 employees. Data included revenue, pay rate, attendance, expense and employee remark.

Qasem and Al-Radaideh (2012) built a classification model to predict the performance of new applicants and employees. For this purpose, they adopted the CRISP-DM data mining methodology and they used the decision tree as the main data mining tool with several experiments. Data was collected via a questionnaire filled by employees in IT companies. Collected data was age, gender, marital status and number of kids, university type, general specialization, degree and grade, number of experience years, number of previous companies worked for, job title, rank, service period in the current company, salary, finding the working conditions uncomfortable and dissatisfaction of salary or rank.

Another study by Thakre et. Al. (2021) aimed to predict employee retention. They created a system that forecasts employee attrition from data Kaggle's Employee dataset. Within a sample of 1470 employees, data included employee age, role, daily rate, job happiness, years at the company and years in present function, etc. Four different machine learning methods, KNN (K-Nearest Neighbor), SVM (Support Vector Machine), Decision Tree, and Random Forest, were conducted.

Gao et al. (2019) used a random forest machine learning model to help human resources departments forecast employee turnover. Weighted quadratic random forest algorithm was applied. Features were monthly salary, percent of salary increase, overtime, years at the company, age and distance from home.

Lather et al. (2020) adopted supervised learning to predict employee performance denoted as three 3 classes (low to high). After trying Support Vector Machines, Random Forest, Naive Bayes, Neural Networks and Logistic Regression, with and the 10-fold validation technique, the best accuracy was obtained with Support Vector Machines.

Chein and Chen (2006) used several attributes to predict the employee performance. They specified age, gender, marital status, experience, education, major subjects and school tires as potential factors that might affect the performance. Then they excluded age, gender and marital status, so that no discrimination would exist in the process of personal selection. As a result for their study, they found that employee performance is highly affected by education degree, the school tire, and the job experience.

Sadath (2013) used Data Mining (DM) techniques for automated intelligent decisions from rich employee database for predictions of employee performance implementing the finest KM strategies, thus achieving stable HR system and brilliant business

Jantan et al. (2010) also propose the potential data mining techniques for talent forecasting. Data mining technique is the best balanced estimator, decision tree and neural network and is found useful in developing predictive models in many fields. In this study, they attempt to use classifier algorithm C4.5 and Random Forest for decision tree; and Multilayer Perceptron (MLP) and Radial Basic Function Network for neural network. They focus on the accuracy of the techniques to find the suitable classifier for HR data. The data are for management and professional employees from higher education institution.

Salleh et al. (2011) tested the influence of motivation on job performance for state government employees in Malaysia. The study showed a positive relationship between affiliation motivation and job performance. As people with higher affiliation motivation and strong interpersonal relationships with colleagues and managers tend to perform much better in their jobs.

3. Mathematical Model

The derivation of the backpropagation algorithm is fairly straightforward. It follows from the use of the chain rule and product rule in differential calculus.

3.1. Input to Neuron (z):

The input to a neuron is a weighted sum of its inputs. For a neuron in layer j with inputs and $x_1, x_2, \ldots, x_n$ weights $w_1, w_2, \ldots, w_n$, the total input z_j is calculated as:

$$z_j = \sum_{i=1}^{n} w_i \cdot x_i \qquad (2)$$

Where,

- z_j : Total input to neuron in layer j.
- x_i : Input from the i-th neuron in the previous layer.
- w_i : Weight associated with the connection from the i-th neuron in the previous layer to the neuron in layer j.
- n: Number of neurons in the previous layer.

3.2. Output from Neuron (o):

The output of a neuron is the result of applying an activation function to its input. The output o_j for the neuron in layer j is given by:

$$O_j = f(z_j) \qquad (3)$$

Where,

- o_j: Output of the neuron in layer j.
- $f(z_j)$: Activation function applied to the total input z_j.

3.3. Activation Function (f):

The activation function f(z) introduces non-linearity to the model. Common activation functions include the rectified linear unit (ReLU), sigmoid (logistic) function and the hyperbolic tangent (tanh) function. For example, the ReLU function is defined as:

$$\mathrm{ReLU}(z) = \max(0, z) \qquad (4)$$

The sigmoid function is another common choice, given by:

$$f(z) = \frac{1}{1+e^{z}} \qquad (5)$$

Where,

- f(z): Activation function applied to the total input z.
- e: Euler's number (approximately 2.71828).
- z: Total input to the neuron.

3.4. Error at the j^{th} layer (output layer)

$$e_j = t_j - o_j \qquad (6)$$

Where,

- e_j: Error at neuron j.
- t_j: Desired or target output for neuron j.
- o_j: Actual output of neuron j.

This equation calculates the numerical difference between the desired output (t_j) and the actual output (o_j) of a neuron, representing the error.

3.5. Error Function for normalization:

$$E = \frac{1}{2}\sum_j (e_j)^2 \qquad (7)$$

Where,

- E: Error function.
- e_j: Error at neuron j.

The error function sums the squared errors across all neurons, providing a measure of overall network performance. This form of error function is commonly used in training neural networks using techniques like gradient descent to minimize the total error.

3.6. Gradient Descent Weight Update for Output Layer (k):

$$\Delta w_{kj} = \eta \delta_j o_j \qquad (8)$$

Where,

- Δw_{kj}: Change in weight for the connection between the hidden layer and the output layer
- η: Learning rate, a constant that controls the size of weight updates.
- δ_j: Local gradient of the error at the output layer.
- o_j: Output from the hidden layer neuron.

This equation represents how the weight w_{kj} is updated based on the error at neuron j and the output of neuron k in the previous layer.

3.7. Gradient Descent Weight Update for Hidden Layer (j):

$$\Delta w_{ij} = \eta\ \delta_j\ o_i \qquad (9)$$

Where,

- Δw_{ij}: Change in weight for the connection between the input layer and the hidden layer.
- η: Learning rate.
- δ_j: Local gradient of the error at the hidden layer.
- o_i: Output from the input layer neuron.

Similar to Equation 9, this equation updates the weights for connections between neurons in hidden layers.

3.8. Partial Derivative for Output Layer (K):

$$\delta_j = f'\ (z_j)\ \Sigma_k\ \delta_k\ w_{kj} \qquad (10)$$

Where,

- δ_j: Local gradient of the error at the output layer.
- $f'(z_j)$: Derivative of the activation function at the output layer.
- z_j: Activation input at neuron j.
- δ_k: Local gradient of the error at the next layer.
- w_{kj}: Weight connecting the hidden layer neuron to the output layer neuron.

This equation calculates the local gradient for neurons in the output layer.

3.9. Partial Derivative for Hidden Layer (j):

$$\delta_j = f'(z_j)\ \Sigma_k\ \delta_k\ w_{kj} \qquad (11)$$

Where,

- δ_j: Local gradient of the error at the hidden layer.
- $f'(z_j)$: Derivative of the activation function at the hidden layer.
- z_j: Activation input for the j^{th} neuron.
- δ_k: Local gradient of the error at the next layer.
- w_{kj}: Weight connecting the hidden layer neuron to the output layer neuron.

This equation calculates the local gradient for neurons in hidden layers by considering the contribution from connected neurons in the next layer.

3.10. Weight update for output layer (k) combining all:

$$\Delta w_{kj} = \eta\delta_j o_j \qquad (12)$$

This is a consolidated form, integrating the components of the weight update for the output layer.

Where,

- Δw_{kj}: Change in weight for the connection between the hidden layer and the output layer.
- η: Learning rate.
- δ_j: Local gradient of the error at the output layer.
- o_j: Output from the hidden layer neuron.

3.11. Weight update for hidden layer (j) combining all:

$$\Delta w_{ij} = \eta\delta_j o_i \qquad (13)$$

Similarly, this equation combines the elements for the weight update in the hidden layer

- Δw_{ij}: Change in weight for the connection between the input layer and the hidden layer.
- η: Learning rate.
- δ_j: Local gradient of the error at the hidden layer.
- o_i: Output from the input layer neuron.

4. Implementation of Code

4.1. Import Section

In this section essential libraries, including NumPy for numerical operations, Scikit-learn for data preprocessing, and Matplotlib for data visualization, are imported. This set's up the required libraries and tools for data manipulation, preprocessing, and visualization throughout the code.

```
# Import Section
import numpy as np
from sklearn.preprocessing import MinMaxScaler
from sklearn.metrics import mean_squared_error
import matplotlib.pyplot as plt
```

4.2. Data Preparation

The dataset consists of employee information, including employee code, education level, work experience, training hours and performance scores. This helps to provide a realistic set of data for training and evaluating the neural network model.

```
# Dataset Section
data = np.array([
   ['E1', 'Masters', 3, 30, 10],
   ['E2', 'Masters', 2, 50, 7],
   ['E3', 'Bachelors', 3, 20, 6],
   ['E4', 'Bachelors', 2, 40, 5],
   ['E5', 'PHD', 10, 50, 9],
   ['E6', 'PHD', 10, 60, 8],
   ['E7', 'Bachelors', 3, 70, 5],
   ['E8', 'Bachelors', 3, 40, 6],
   ['E9', 'Bachelors', 3, 90, 6],
   ['E10', 'Diploma', 2, 10, 4]
], dtype=object)
```

4.3. Data Preprocessing

In this section categorical variables, such as education level, are converted to numerical values and the data is normalized. This preprocess the data to ensure compatibility with the neural network model.

```
# Data Preprocessing Section
# Separate features (X) and performance scores (Y)
names = data[:, 0]
education_levels = data[:, 1]
X = data[:, 2:-1].astype(float)  # Convert features to float (excluding the
first and last columns)
Y = data[:, -1:].astype(float)   # Convert performance scores to float

# Convert categorical education levels to numerical values
education_mapping = {'Diploma': 1, 'Bachelors': 2, 'Masters': 3, 'PHD': 4}
X[:, 0] = [education_mapping[edu] for edu in education_levels]

# Normalize the data
scaler = MinMaxScaler()
X_normalized = scaler.fit_transform(X)
```

4.4. Neural Network Initialization

In this section neural network parameters, including input, hidden, and output layer sizes, learning rate, and epochs, are defined. Weights and biases are initialized using Xavier/Glorot initialization. Purpose of this is to set up the architecture and initial conditions for the neural network.

```
# Section: Neural Network Initialization
# Neural Network Parameters
input_size = X.shape[1]
hidden_size = 10
output_size = 1
learning_rate = 0.01
epochs = 1000

# Initialize Weights and Biases with Xavier/Glorot initialization
weights_input_hidden = np.random.randn(input_size, hidden_size) /
np.sqrt(input_size)
weights_hidden_output = np.random.randn(hidden_size, output_size) /
np.sqrt(hidden_size)
bias_hidden = np.zeros((1, hidden_size))
bias_output = np.zeros((1, output_size
```

4.5. Activation Function (Rectified Linear Unit - ReLU)

ReLU is an activation function that outputs the input directly if it is positive, otherwise, it returns zero. The derivative of ReLU is 1 for positive inputs and 0 for negative inputs. Introducing non-linearity into the neural network, allowing it to learn complex patterns and relationships in the data. ReLU is chosen for its simplicity and effectiveness in mitigating the vanishing gradient problem.

```
# Activation Function (Rectified Linear Unit - ReLU)
def relu(x):
   return np.maximum(0, x)

def relu_derivative(x):
   return np.where(x > 0, 1, 0)
```

4.6. Training Model (Forward and Backward Propagation)

The neural network is trained using a specified number of epochs. In each epoch, forward propagation is performed, where input data is passed through the network to generate predictions. Error is calculated by comparing predicted and actual outputs. Backward propagation then updates the weights and biases based on the error to improve the model's performance.

```
# Training the Neural Network
for epoch in range(epochs):
   # Forward Propagation with ReLU activation in the hidden layer
   hidden_layer_input = np.dot(X_normalized, weights_input_hidden) +
bias_hidden
   hidden_layer_output = relu(hidden_layer_input)

   output_layer_input = np.dot(hidden_layer_output, weights_hidden_output) +
bias_output
   predicted_output = relu(output_layer_input)

   # Calculate Error
   error = Y - predicted_output

   # Backward Propagation with ReLU derivative
   output_error = error * relu_derivative(output_layer_input)
   hidden_layer_error = output_error.dot(weights_hidden_output.T) *
relu_derivative(hidden_layer_input)

   # Update Weights and Biases
   weights_hidden_output += hidden_layer_output.T.dot(output_error) *
learning_rate
   bias_output += np.sum(output_error, axis=0, keepdims=True) *
learning_rate

   weights_input_hidden += X_normalized.T.dot(hidden_layer_error) *
learning_rate
   bias_hidden += np.sum(hidden_layer_error, axis=0, keepdims=True) *
learning_rate

    # Print intermediate information for debugging
   if epoch % 100 == 0:

       print(f"Epoch {epoch}, Mean Squared Error: {mean_squared_error(Y,
predicted_output)}")
```

Iteratively adjusts the model's parameters to minimize the difference between predicted and actual outputs, enhancing the model's ability to generalize to unseen data. The ReLU activation in the hidden layer introduces non-linearity, enabling the neural network to learn more complex relationships in the data. Intermediate information, such as Mean Squared Error, is printed periodically for monitoring training progress.

4.7. Testing The model

This section train model which is used to predict performance scores on the same dataset. This assess the model's ability to generalize and make accurate predictions on new data.

```
# Testing the Neural Network
hidden_layer_test = relu(np.dot(X_normalized, weights_input_hidden) +
bias_hidden)
predicted_scores = relu(np.dot(hidden_layer_test, weights_hidden_output) +
bias_output)
```

4.8. Performance Evaluation

In this section Mean Squared Error is calculated to evaluate the model's performance. Results are visualized using a scatter plot and a table displaying employee information. This provides a visual representation of the model's predictions and its accuracy.

```
# Evaluate Performance
mse = mean_squared_error(Y, predicted_scores)
print(f"Mean Squared Error: {mse}")
```

4.9. Data Visualization

This section helps to visualize result in scatter plot and table. Scatter plot visualizes the relationship between actual and predicted performance scores. A table displays employee information, actual scores, predicted scores, and model suggestions based on a predefined threshold. This offers a comprehensive view of model performance and aid in decision-making for employee management.

```
# Visualize Results and Display Information
# Scatter Plot
plt.scatter(Y, predicted_scores)
for i, name in enumerate(names):
   plt.annotate(name, (Y[i], predicted_scores[i, 0]))
plt.xlabel('Actual Performance Scores')
plt.ylabel('Predicted Performance Scores')
plt.title('Actual vs. Predicted Performance Scores')
plt.show()
# Display Employee Information in Table
fig, ax = plt.subplots()
table_data = [['Employee', 'Actual Score', 'Predicted Score', 'Suggestion']]
threshold = 7  # Set threshold value
for i, name in enumerate(names):
   actual_score = Y[i, 0]
```

```
    predicted_score = round(predicted_scores[i, 0], 2)

    row_data = [name, actual_score, predicted_score]

    # Compare predicted score with the threshold
    if predicted_score < threshold:
        suggestion = "The predicted performance score is lower than
expected.So, additional support is needed and areas for improvement should
be identified."
        row_data.append(suggestion)
    else:
        suggestion = "The predicted performance score within the expected
range."
        row_data.append(suggestion)

    table_data.append(row_data)

table = ax.table(cellText=table_data, loc='center', cellLoc='center',
colLabels=None)
table.auto_set_font_size(False)
table.set_fontsize(8)
# Adjust the initial scale based on preference
table.scale(1, 1.5)

# Auto adjust table height based on content
table.auto_set_column_width([0, 1, 2, 3])
table.auto_set_column_width([-1])

# Turn off the axis for the table subplot

ax.axis('off')
plt.show()
```

5. Results

The achieved Mean Squared Error (MSE) of 0.409 reflects the model's ability to capture the underlying patterns within the dataset. Due to the squaring of errors in the Mean Squared Error (MSE), the resulting value is always non-negative, and its range spans from 0 to infinity. As mistakes accumulate, MSE exhibits exponential growth, making it sensitive to outliers or large errors. In the context of model evaluation, a well-performing model is indicated by an MSE value approaching zero (Allen, 1971). The results of the neural network predictions, when compared to actual performance scores, indicate a satisfactory overall alignment. However, it is crucial to address instances where the model underestimated employee performance.

The consistently observed lower predictions suggest potential areas for model refinement. This discrepancy could be attributed to the complexity of real-world performance factors that extend beyond the captured features. Future iterations of the model may benefit from incorporating additional relevant variables or fine-tuning hyperparameters to enhance predictive accuracy.

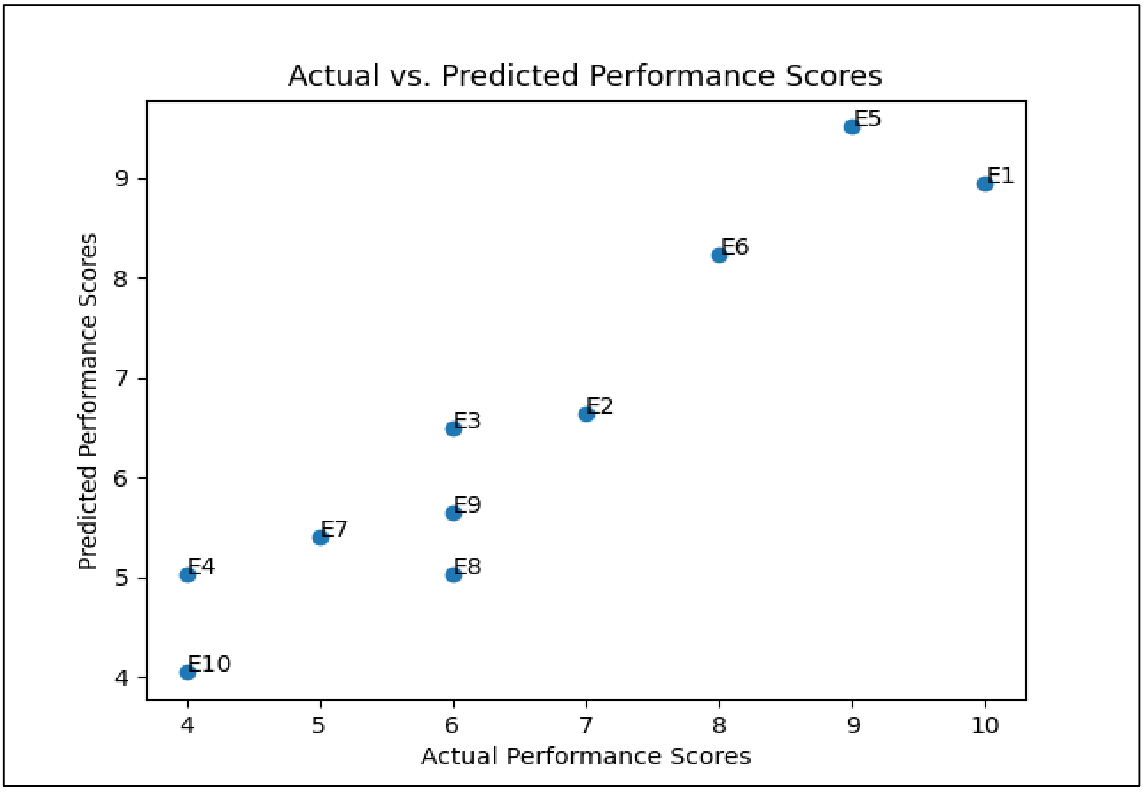
Employee	Actual Score	Predicted Score	Suggestion
E1	10.0	8.95	The predicted performance score within the expected range.
E2	7.0	6.63	The predicted performance score is lower than expected.So, additional support is needed and areas for improvement should be identified.
E3	6.0	6.5	The predicted performance score is lower than expected.So, additional support is needed and areas for improvement should be identified.
E4	4.0	5.04	The predicted performance score is lower than expected.So, additional support is needed and areas for improvement should be identified.
E5	9.0	9.51	The predicted performance score within the expected range.
E6	8.0	8.23	The predicted performance score within the expected range.
E7	5.0	5.41	The predicted performance score is lower than expected.So, additional support is needed and areas for improvement should be identified.
E8	6.0	5.04	The predicted performance score is lower than expected.So, additional support is needed and areas for improvement should be identified.
E9	6.0	5.65	The predicted performance score is lower than expected.So, additional support is needed and areas for improvement should be identified.
E10	4.0	4.05	The predicted performance score is lower than expected.So, additional support is needed and areas for improvement should be identified.

Source: Own Study, 2024

Figure 3 Actual, Predicted Score and Suggestion Output

Moreover, the individualized suggestions generated based on the predicted scores provide actionable insights for employee development. The model's ability to pinpoint areas for improvement or support aligns with the broader objective of enhancing performance management strategies within the organizational context.

while the current model demonstrates promising predictive capabilities, ongoing refinement efforts and a deeper exploration of influential factors will contribute to its robustness in guiding effective decision-making and talent management practices.



Source: Own Study, 2024

Figure 4 Scatter Plot of actual and predicted performance Score

6. Conclusion

This study leveraged a neural network model to predict employee performance scores based on relevant features such as education level, work experience, and training hours. The achieved Mean Squared Error (MSE) of 0.409 indicates a commendable level of predictive accuracy. The model effectively aligns with actual performance scores, offering valuable insights for talent management and decision-making processes.

While the model excels in capturing overall trends, the observed instances of underestimation highlight the intricacies of real-world performance dynamics. These nuances suggest opportunities for refinement, encouraging the inclusion of additional features or the fine-tuning of parameters to enhance predictive precision.

The personalized suggestions generated by the model contribute practical guidance for employee development, aligning with contemporary performance management strategies. This study underscores the potential of machine learning in augmenting HR practices, offering a data-driven approach to talent assessment and improvement strategies.

In conclusion, the current model stands as a promising tool for performance prediction, but ongoing efforts towards refinement and expansion will further solidify its utility in fostering a proactive and adaptive approach to talent management within organizational contexts.

6.1. Limitation of the study

- The study's predictive model was trained on a relatively small dataset, potentially limiting its generalizability to larger and more diverse populations.
- The current model relies on a limited set of features such as education level and work experience. The exclusion of certain factors, like specific job roles or interpersonal skills, may impact the comprehensiveness of predictions.
- The model assumes static relationships between features and performance, overlooking potential dynamic changes in employee capabilities over time.
- The linear nature of the neural network architecture assumes linear relationships between input features and performance, potentially oversimplifying the underlying complexities.
- The model does not account for external factors influencing performance, such as changes in industry trends, economic conditions, or organizational culture shifts.

6.2. Future Research

While this study provides valuable insights into predicting employee performance using neural network, several avenues for future research could enhance the depth and applicability of predictive models. Investigating the impact of additional contextual factors, such as organizational climate, leadership styles, and team dynamics, could contribute to a more holistic understanding of performance determinants.

To advance the field of performance management, future research could explore the integration of predictive models with Human Resource Information Systems (HRIS). This integration aims to create a comprehensive solution that seamlessly incorporates employee performance predictions into the HRIS infrastructure. Such a system could automate performance evaluation processes, assist in talent management, and provide actionable insights for strategic decision-making.

In addition to neural networks, more sophisticated machine learning algorithms, such as ensemble methods (e.g., Random Forests, Gradient Boosting), deep learning architectures (e.g., deep neural networks, convolutional neural networks), or even natural language processing (NLP) models for sentiment analysis, could be explored. These advanced models might capture intricate patterns and nuances in employee data, offering enhanced accuracy and interpretability in predicting performance.

Furthermore, the integration of predictive analytics into HRIS could enable real-time monitoring of employee performance, facilitating dynamic adjustments to talent development strategies. The combination of advanced ML models and HRIS integration holds the potential to revolutionize performance management practices, fostering a data-driven and adaptive approach to workforce optimization.

References

[1] Abdolrasol, M. G. M., Hussain, S. M. S., Ustun, T. S., Sarker, M. R., Hannan, M. A., Mohamed, R., Ali, J. A., Mekhilef, S., & Milad, A. (2021). Artificial Neural Networks Based Optimization Techniques: A Review. Electronics, 10(21), 2689. https://doi.org/10.3390/electronics10212689

[2] Aggarwal, A., & Thakur, G. S. M. (2013). Techniques of performance appraisal-a review. International Journal of Engineering and Advanced Technology (IJEAT), pp. 617-621.

[3] Allen, D. M. (1971). Mean square error of prediction as a criterion for selecting variables. Technometrics, 13(3), 469–475.

[4] Altay, E., & Satman, M. H. (2005). Stock Market Forecasting: Artificial Neural Network and Linear Regression Comparison in an Emerging Market. Journal of Financial Management and Analysis, 18. Available at SSRN

[5] Arora, N., & Arora, P. (2010). Performance Appraisal: A Managerial Viewpoint. Indiastat. com September-October-2010, Socio-economic voice P, (1).

[6] Becker, B. E., & Gerhart, B. (1996). The impact of human resource management on organizational performance: Progress and prospects. Academy of Management Journal, 39, 779-801.

[7] Cappelli P. and Tavis A. "The performance management revolution." Harvard Business Review, 94(10), 58-67.

[8] Chein, C., & Chen, L. (2006). Data mining to improve personnel selection and enhance human capital: A case study in the high technology industry. Expert Systems with Applications, In Press.

[9] Farley, B. G., & Clark, W. A. (1954). Simulation of Self-Organizing Systems by Digital Computer. IRE Transactions on Information Theory, 4(4), 76–84. doi:10.1109/TIT.1954.1057468

[10] Gao, X., Wen, J., & Zhang, C. (2019). An Improved Random Forest Algorithm for Predicting Employee Turnover. Math. Probl. Eng., 2019, e4140707. doi: 10.1155/2019/4140707

[11] Jantan, H., Hamdan, A. R., & Othman, Z. A. (2010). Human Talent Prediction in HRM using C4.5 Classification Algorithm. International Journal on Computer Science and Engineering, 2(08-2010), 2526-2534.

[12] Kumar, M., & Thenmozhi, M. (2006). Forecasting Stock Index Movement: A Comparison of Support Vector Machines and Random Forest. Indian Institute of Capital Markets 9th Capital Markets Conference Paper. (Available at SSRN or http://dx.doi.org/10.2139/ssrn.876544)

[13] Lather, S., Malhotra, R., Saloni, P., Singh, P., & Mittal, S. (2020). Prediction of employee performance using machine learning techniques. In Proceedings of the 1st International Conference on Advanced Information Science and System, New York, NY, USA, pp. 1–6. doi: 10.1145/3373477.3373696.

[14] Liao, Z., & Wang, J. (2010). Forecasting Model of Global Stock Index by Stochastic Time Effective Neural Network. Expert Systems with Applications, 37, 834-841. http://dx.doi.org/10.1016/j.eswa.2009.05.086

[15] McCulloch, W., & Pitts, W. (1943). A Logical Calculus of Ideas Immanent in Nervous Activity. Bulletin of Mathematical Biophysics, 5(4), 115–133. doi:10.1007/BF02478259

[16] Nasr, M., Shaaban, E., & Samir, A. (2019). A proposed Model for Predicting Employees' Performance Using Data Mining Techniques: Egyptian Case Study. Int. J. Comput. Sci. Inf. Secur., 17(1), 31–40. doi:10.5815/ijcsis.2019.01.04.

[17] Pfeffer and Robert Sutton. Hard Facts, Dangerous Half-Truths, and Total Nonsense: Profiting From Evidence-Based Management. (2006). Harvard Business Press; and Denise Rousseau(Ed.) The Oxford Handbook of Evidence-Based Management. (2012). Oxford University Press.

[18] Qasem, A., & Al-Radaideh, E. (2012). Using Data Mining Techniques to Build a Classification Model for Predicting Employees Performance. Int. J. Adv. Comput. Sci. Appl., 3(2)*. doi: 10.14569/IJACSA.2012.030225

[19] Richards, J. (2006). Remote Sensing Digital Image Analysis. Berlin: Springer-Verlag.

[20] Roh, T. H. (2007). Forecasting the volatility of stock price index. Expert Systems with Applications, 33(4), 916-922.

[21] Sadath, L. (2013). Data Mining: A Tool for Knowledge Management in Human Resource. International Journal of Innovative Technology and Exploring Engineering, 2(6), April 2013.

[22] Schuler, R. S. (1994). Human resource management: Domestic to global. In M. Warner (Ed.), International encyclopedia of business and management. London: Routledge.

[23] Singh, H., Vishnavat, K., & Srinivasan, R. (2018). Employee performance and leave management using data mining technique. International Journal of Pure and Applied Mathematics, 118, 2063.

[24] Thakre, G., Wankhede, P., Patle, S., Joshi, S., & Chauhan, A. (2021). Implementation of Machine Learning Model for Employee Retention Prediction. Int. J. Sci. Res. Sci. Technol., 8(3), 503–508. doi:10.32628/IJSRST2183122

[25] Wright, P., Gardner, T., Moynihan, L., Park, H., Gerhart, B., & Delery, J. (2001). Measurement Error in Research on Human Resources and Firm Performance: Additional Data and Suggestions for Future Research. CAHRS Working Paper Series, 54. 10.1111/j.1744-6570.2001.tb00235.x

[26] Yacim, J., & Boshoff, D. (2018). Impact of Artificial Neural Networks Training Algorithms on Accurate Prediction of Property Values. Journal of Real Estate Research, 40, 375-418. https://doi.org/10.1080/10835547.2018.12091505.

[27] Zhang, L., Li, X., & Li, D. (2022). Construction and Simulation of a Strategic HR Decision Model Based on Recurrent Neural Network. Journal of Mathematics, 2022.