International Journal of Intelligent Information Systems
2024, Vol. 13, No. 6, pp. 117-127
https://doi.org/10.11648/j.ijiis.20241306.11

Research Article

Predictive Hiring and AI: Elevating Recruitment with Optimized Neural Networks and Gradient Descent

Temsamani Khallouk Yassine*, Achchab Said

Engineering School of Information Technology and System Analysis, Mohammed V University, Rabat, Morocco

Abstract

Efficient and accurate hiring processes are critical for organizational success, yet traditional recruitment methods often face challenges such as inefficiencies and delays. This study explores the application of artificial intelligence (AI) and machine learning (ML) techniques to enhance predictive hiring models. A hybrid framework is proposed, integrating neural networks with Stochastic Gradient Descent (SGD) optimization and feature selection methods, including Logistic Regression (LR) and Discriminant Analysis (DA). The approach demonstrates a marked improvement in prediction accuracy and efficiency, with Logistic Regression emerging as a more effective feature selection method for neural networks in this context. By leveraging these techniques, human resource teams can streamline candidate evaluations, enhance decision-making processes, and modernize recruitment workflows. This research underscores the transformative potential of AI in addressing the limitations of traditional hiring practices.

Keywords

Predictive Hiring, HR, Stochastic Gradient Descent, Optimization, Variable Selection

1. Introduction

In today's fast-paced, technology-driven world, organizations are increasingly turning to artificial intelligence (AI) to enhance their decision-making processes across various functions, including human resources (HR). AI has emerged as a game-changer in HR management, offering powerful tools for improving recruitment, employee engagement, and performance evaluation. By analyzing vast amounts of data, AI allows HR professionals to make more informed decisions, streamline workflows, and improve overall efficiency [1, 2].

The application of AI in hiring is particularly transformative. Traditional hiring processes, often bogged down by manual tasks like resume screening and multiple interview rounds, are slow and prone to human bias [3, 4]. AI, on the other hand, can quickly and accurately sift through large datasets, identifying patterns that help predict a candidate's potential success in a role. This predictive power makes AI a valuable tool in modern recruitment, reducing time-to-hire an improving the quality of hiring decisions [5].

Neural networks, a subset of AI, are particularly effective in predictive hiring. They can learn from vast amounts of data to recognize complex relationships between a candidate's qualifications and their likelihood of success. When combined with optimization techniques like Stochastic Gradient Descent (SGD), these models can be fine-tuned to maximize accuracy [2]. However, selecting the right varia-

*Corresponding author: Temsamani.khallouk.yassine@gmail.com (Temsamani Khallouk Yassine)

Received: 18 November 2024; **Accepted:** 3 December 2024; **Published:** 23 December 2024

bles and ensuring that the models are properly optimized is essential for obtaining reliable predictions [6].

In this study, we aim to optimize neural networks for predictive hiring by employing SGD and incorporating variable selection methods such as discriminant analysis and linear regression. This helps reduce the complexity of the data while maintaining the most relevant factors for hiring decisions. By comparing these advanced neural networks with traditional machine learning models, we provide insights into how AI can revolutionize recruitment, making it faster, more accurate, and better aligned with company goals.

2. Predictive Hiring

The integration of artificial intelligence (AI) and machine learning (ML) into human resource management (HRM) has transformed the hiring landscape, particularly with the advent of predictive hiring models. These models leverage AI to analyze vast amounts of data, such as resumes, behavioral assessments, and historical employee performance, to predict how well candidates will perform in future roles. Predictive hiring not only reduces time-to-hire but also enhances accuracy by relying on quantifiable metrics instead of subjective judgment. This data-driven approach aligns hiring decisions with broader organizational goals, ensuring that the right talent is placed in the right roles [1].

One of the key technologies behind predictive hiring is the application of deep learning, specifically neural networks, which excel in identifying complex, non-linear patterns within large datasets. Neural networks, optimized using techniques such as Stochastic Gradient Descent (SGD), continuously improve over time, making them highly effective in predictive hiring scenarios. By learning from previous hiring successes and failures, these models can better match candidates to job roles and predict long-term success [2].

However, despite these benefits, challenges such as data availability, quality, and potential bias persist. Studies have shown that biased historical data can lead to discriminatory hiring practices, which is why fairness and transparency in AI algorithms are critical considerations. Fairness in machine learning models must be addressed to avoid perpetuating historical biases and to ensure compliance with ethical standards [6].

Predictive hiring has demonstrated remarkable potential in improving recruitment outcomes, as evidenced by its growing adoption across industries. It can enhance the quality of hires, reduce turnover rates, and allow companies to scale their talent acquisition strategies efficiently. Companies that leverage these models report better hiring accuracy, faster recruitment cycles, and improved retention rates [5]. With predictive hiring becoming a key part of modern HRM, its ethical application will shape the future of recruitment, aligning talent acquisition with organizational success [3, 4].

2.1. Recent Studies on Machine Learning in Predictive Hiring

Machine learning has significantly transformed predictive hiring by enhancing the efficiency, accuracy, and objectivity of candidate selection processes. This technological advancement addresses common recruitment challenges, such as bias, time consumption, and high costs associated with traditional hiring methods. By leveraging machine learning algorithms, predictive hiring models can analyze vast amounts of data to identify the best candidates, streamline hiring processes, and ultimately improve organizational outcomes.

Applications of Machine Learning in Predictive Hiring:

2.1.1. Resume Screening

Automated resume screening tools employ machine learning algorithms to evaluate candidates’ qualifications based on historical hiring data. These tools can quickly filter out unqualified candidates and highlight those who match the job requirements, significantly reducing the time recruiters spend on manual screening. For example, Guenole and Feinzig (2018) underscore the use of machine learning models to predict candidate success and fit by analyzing historical hiring data and employee performance metrics. Their study demonstrates how these models can identify patterns and predictors of employee success, facilitating more informed hiring decisions (Guenole Feinzig, 2018). [7]

2.1.2. Candidate Ranking

Machine learning algorithms can rank candidates by predicting their potential performance and cultural fit within the organization. By analyzing data from various sources, such as past job performance, educational background, and social media activity, these algorithms can provide a comprehensive assessment of each candidate. Weller et al. (2020) discuss how machine learning models, particularly those utilizing natural language processing (NLP) techniques, can enhance candidate ranking by evaluating textual data from resumes and cover letters. NLP algorithms can identify key skills, experiences, and personality traits that correlate with job success, thereby improving the accuracy of candidate rankings (Weller et al., 2020). [8]

2.1.3. Interview Analysis

Advanced machine learning techniques are also applied to analyze interview data, including audio, video, and text. These models can assess candidates’ verbal and non-verbal cues, such as tone of voice, facial expressions, and speech patterns, to predict their suitability for the role. Studies have shown that incorporating machine learning in interview analysis can reduce human bias and provide a more objective evaluation of candidates’ potential. For instance, AI-driven interview platforms can analyze candidates’ responses in real-time, providing immediate feedback to recruiters (Chen et al., 2019). [9]

2.2. Ethical Implications of Predictive Hiring

As predictive hiring becomes more prevalent, ethical considerations take center stage. The reliance on AI and machine learning in hiring processes raises concerns about bias, transparency, and accountability. Historical data used to train predictive models often contains embedded biases, which can perpetuate discrimination against certain demographic groups. Addressing these biases requires a concerted effort to audit datasets, develop fairness-aware algorithms, and implement rigorous testing protocols.

Another ethical challenge involves transparency. Many predictive hiring models operate as "black boxes," making it difficult for HR professionals to understand how decisions are made. This lack of interpretability can undermine trust in AI-driven hiring tools and complicate efforts to ensure compliance with regulations. Explainable AI (XAI) offers potential solutions by providing insights into how predictions are generated, thereby fostering greater transparency and accountability.

Moreover, the use of AI in hiring raises privacy concerns. Predictive models often rely on personal data, including social media activity and behavioral assessments, which can infringe on candidates' privacy rights. Organizations must establish clear data governance policies and ensure that data collection practices adhere to legal and ethical standards.

3. Artificial Neural Networks in Predictive Hiring

Artificial Intelligence (AI) has become an indispensable tool in recruitment, offering solutions that streamline the hiring process, improve accuracy, and reduce bias. Predictive hiring, driven by AI models, enables companies to forecast a candidate’s job performance using historical data, behavioral traits, and experience. The ability to analyze vast datasets and uncover hidden patterns that humans might overlook makes AI particularly valuable in recruitment. AI-based systems have demonstrated their effectiveness by automating resume screening, ranking candidates, and even conducting initial interviews via AI-driven chatbots. This ensures that companies can quickly identify high-potential candidates and reduce the time-to-hire by up to 50% [6].

3.1. Predictive Hiring with Neural Networks

One of the most promising applications of AI in recruitment is the use of neural networks for predictive hiring. Neural networks, particularly artificial neural networks (ANNs), are powerful machine learning models that mimic the human brain’s ability to process information, recognize patterns, and make decisions. In recruitment, these networks can analyze candidate data—ranging from resumes, interviews, assessments, and even social media behavior—to predict future job performance and fit. By using historical hiring data, ANNs learn to identify complex relationships between candidate attributes (e.g., skills, education, and work experience) and job success.

Neural networks excel at modeling non-linear relationships, making them ideal for recruitment scenarios where multiple factors interact to influence job performance. Unlike traditional statistical methods, which may assume linear relationships between variables, ANNs can process large, complex datasets and handle diverse input types, such as text, numerical data, and even images (e.g., for facial analysis during video interviews). This flexibility allows companies to make more accurate hiring decisions based on the entire candidate profile, rather than just surface-level attributes [10].

3.2. Structure of a Neural Network

An artificial neural network is composed of several layers, each with its specific role in processing the input data. The most common structure includes:

1) Input Layer: This layer receives the raw data about the candidate, such as educational background, previous work experience, test scores, and other measurable factors. Each neuron in the input layer corresponds to a specific feature in the dataset (e.g., years of experience, skills, or keywords from a resume).
2) Hidden Layers: These layers, located between the input and output layers, are where the network learns the relationships between input features. Hidden layers consist of neurons that apply activation functions to the inputs, which introduces non-linearity and allows the network to capture complex patterns in the data. The choice of the number of hidden layers and neurons within them is a critical aspect of designing a neural network, as it directly impacts the network’s performance. Techniques such as Stochastic Gradient Descent (SGD) and backpropagation are used to adjust the weights of these connections based on the error between predicted and actual outcomes [11].
3) Output Layer: The output layer provides the final prediction or decision. In the context of recruitment, the output might be a score representing the predicted job performance or a classification indicating whether the candidate is a good fit for the position. The network’s ability to provide such predictions makes it an invaluable tool in predictive hiring, helping recruiters to focus on high-potential candidates efficiently [12].

3.3. Training a Neural Network for Predictive Hiring

Training a neural network for predictive hiring involves optimizing the network’s weights and biases to minimize the prediction error. This process is done through supervised learning, where the model is trained on historical candidate data labeled with performance outcomes (e.g., whether the candidate was successful in the role). Techniques like backpropagation are used to propagate errors back through the network, allowing the model to adjust its weights and improve accuracy over time [2].

The network's training process requires a large amount of candidate data to generalize effectively. This data includes both positive examples (candidates who were successful in their roles) and negative examples (candidates who underperformed or left the company early). By analyzing these patterns, the neural network learns which features are most predictive of future success and adjusts its predictions accordingly [3].

4. Stochastic Gradient Descent (SGD)

Stochastic Gradient Descent (SGD) is an iterative optimization algorithm used to minimize a loss function by adjusting the parameters of a model, such as the weights and biases in a neural network. In the context of machine learning, the goal is to find the parameters that minimize the difference between predicted and actual outcomes, quantified by the loss function. Unlike traditional gradient descent, which computes the gradient using the entire dataset, SGD updates the model parameters using only one sample at each iteration, making it more efficient for large-scale data.

In predictive hiring, where neural networks must process vast amounts of candidate data, SGD is an effective optimization method due to its ability to handle large datasets efficiently and converge faster. The algorithm's memory-efficient nature, which requires just one data point per iteration to compute parameter updates, is particularly advantageous in scenarios involving extensive candidate profiles and complex features.

The generic form of the optimization problem in neural networks is defined as:

$$min_w l(w) := (\sum_{n=1}^{n} (l_t(h(x_i; w), y_i)) \backslash n$$

where $\ell(w)$ represents the loss function, $h(x_i;\ w)$ is the prediction function, and w denotes the weights of the model. The goal is to minimize the average loss over all training examples (x_i, y_i) by iteratively adjusting the weights.

Key Characteristics of SGD: [13, 14]

1) Efficiency and Scalability: SGD's major advantage lies in its efficiency when working with large datasets. Unlike batch gradient descent, which requires computing gradients across the entire dataset, SGD performs updates with a single randomly selected data point at each iteration. This significantly reduces computational costs and speeds up convergence, which is critical when working with large volumes of candidate data in predictive hiring.
2) Faster Convergence: Due to its stochastic nature, SGD introduces noise into the optimization process. This randomness can accelerate convergence by allowing the algorithm to avoid getting trapped in local minima, which is particularly beneficial for neural networks, whose loss functions are often highly non-convex. This characteristic is essential when training models for predictive hiring, where the complexity and variability of candidate data can result in intricate optimization landscapes.
3) Real-Time and Online Learning: SGD is well-suited for real-time or online learning environments, where data is continuously generated and must be processed sequentially. For example, in predictive hiring, new candidate data may be constantly added to the system, and the model needs to adapt in real-time to new information. SGD's ability to update parameters after each data point allows the model to learn continuously and efficiently without needing to process the entire dataset at once.
4) Variants of SGD: Several variants of SGD have been developed to further improve performance, such as SGD with momentum, AdaGrad, RMSProp, and Adam. These variants introduce adaptive learning rates or momentum terms to accelerate convergence and improve stability during training. For instance, SGD with momentum accumulates a velocity vector, enabling the model to move faster in directions with consistent gradient reductions, while damping oscillations in other directions. This variant is particularly useful in predictive hiring, where consistent, reliable performance is necessary across different candidate data sets.

Algorithm: Stochastic Gradient Descent (SGD) [15]

Algorithm 1: Stochastic Gradient Descent (SGD)

```
1. Input : Learning rate $\eta_t > 0$
2. Initialize: $\theta_0 \in \mathbb{R}_\rho^d, t = 0$
3. While Stop Criteria not me do
    a. Sample $\xi \approx p$ where i $\in \{1 \cdots n\}$
    b. $G_t = \nabla f\ \ (\theta_t; \varepsilon_L)$
    c. $\theta_{t+1} = \theta_t - \eta_t \varsigma_t$
    d. $t = t + 1$
End while=0
```

The main advantage of SGD is its computational efficiency. By updating the weights using a single sample, it reduces the overhead associated with recalculating gradients for the entire dataset. Although SGD introduces some noise into the gradient estimates, this can lead to faster convergence and better generalization performance, especially in high-dimensional spaces like those encountered in predictive hiring models [16, 17].

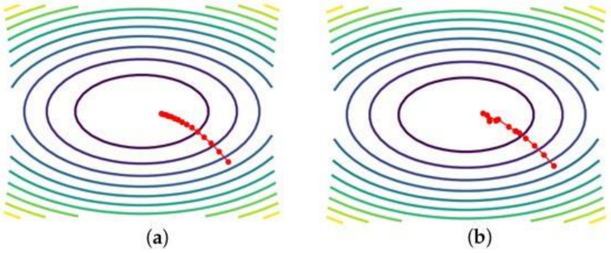


Figure 1. Comparison of Stochastic Gradient Descent (SGD) and Gradient Descent (GD).

Ethical Considerations in AI-Driven Predictive Hiring

Despite the promise of neural networks in predictive hiring, several challenges and ethical concerns must be addressed. One significant issue is the potential for bias in the model's predictions. If the training data includes biases (e.g., past hiring practices that favor certain demographics), the neural network may learn to replicate these biases, leading to discriminatory hiring decisions. This raises concerns about fairness and equity in AI-driven recruitment processes [6].

To mitigate these risks, companies must ensure that their models are regularly audited for bias and that fairness-aware machine learning techniques are implemented. Additionally, explainability is crucial in AI recruitment systems. Neural networks are often seen as "black boxes" due to the complexity of their decision-making process. Using explainability techniques, such as SHAP (SHapley Additive exPlanations) or LIME (Local Interpretable Model-agnostic Explanations), can help HR professionals understand and trust the predictions generated by neural networks [5].

5. Research Framework

To define the predictive model for hiring, a research framework is developed to describe the entire process from data collection to evaluation. The framework outlines the steps involved in building a robust predictive hiring system, using data from a Moroccan bank over a four-year period (2018-2022). The complete research workflow is presented in Figure 5, consisting of four major stages: Data Collection, Data Pre-Processing, Model Building & Optimization, and Model Evaluation.

5.1. Data Collection

The dataset used for this study was provided by a Moroccan bank and includes data on 124,000 individuals collected over a four-year period. The data consists of various attributes, including demographic details, employment history, financial behavior, and more. This rich dataset provides the foundation for training the predictive model to identify successful hires.

5.2. Data Pre-Processing

Before building the model, the raw data undergoes extensive pre-processing to ensure that it is clean, structured, and ready for use in training. The steps include:

1) Data Cleaning: This step involves handling missing values, removing outliers, and standardizing the format of variables. It ensures that the dataset is free of inconsistencies that could skew the model's performance.
2) Variable Selection: To identify the most relevant features, discriminant analysis and linear regression techniques are applied. These methods help reduce the dimensionality of the dataset by selecting variables that have the greatest predictive power.
3) Feature Engineering: Additional features are created from the existing data to enhance the model's ability to learn. For example, categorical data is transformed into numerical representations, and text data (from resumes or feedback) is converted using natural language processing.

5.3. Model Building & Optimization

The predictive model is built using an artificial neural network (ANN), chosen for its ability to model complex, non-linear relationships in the data. The network architecture consists of input layers, hidden layers, and an output layer, where the number of neurons and layers is optimized based on the dataset's complexity.

To optimize the training process, Stochastic Gradient Descent (SGD) is employed, allowing the model to efficiently learn from the data. The optimization process ensures that the model's weights are updated iteratively to minimize the error. Regularization techniques such as dropout and L2 regularization are used to prevent overfitting, ensuring that the model generalizes well to unseen data.

5.4. Model Evaluation

Once the model is built, its performance is evaluated using key metrics, including accuracy, precision, recall, and the F1 score. A confusion matrix (TP, TN, FP, FN) is used to measure the performance of the model in predicting successful hires. These metrics are critical in assessing how well the model performs and in identifying areas for improvement. Cross-validation is applied to further validate the model's robustness.

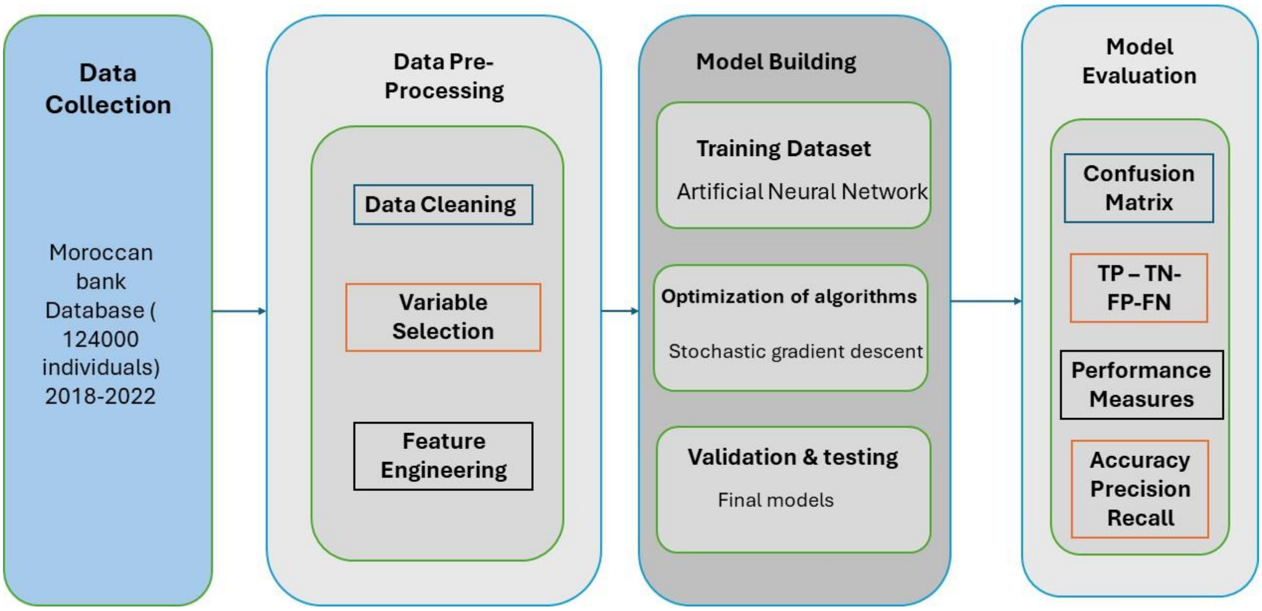


Figure 2. Research Framework.

6. Data Application and Preparation

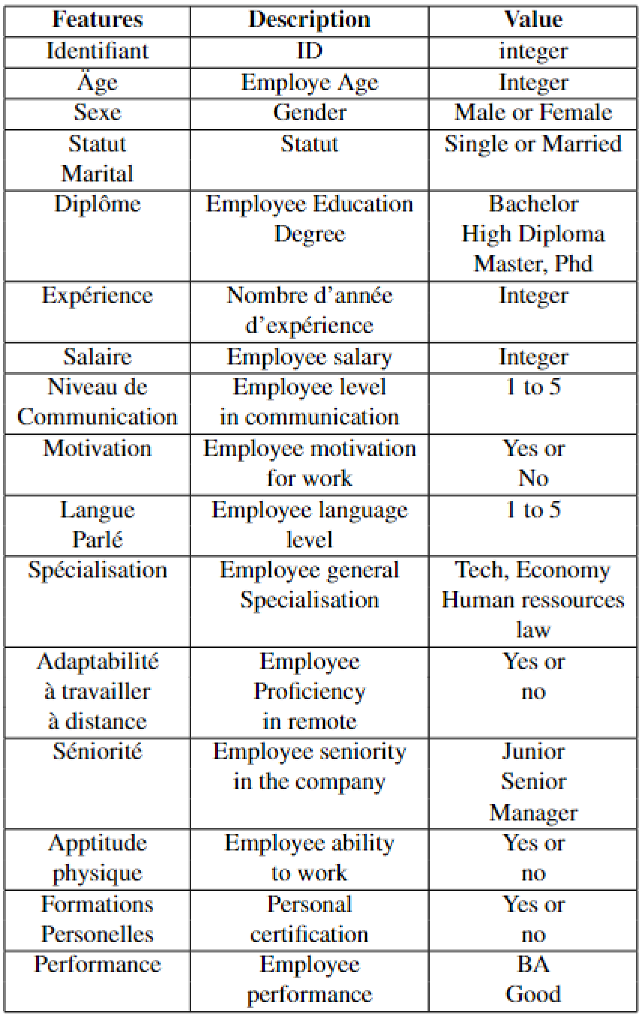
6.1. Data Collection

The dataset used in this study was sourced from a prominent Moroccan corporation. It consists of detailed recruitment data covering more than 10,000 candidates across various departments, including IT, finance, human resources, and operations. The data spans several years and is structured to include critical variables relevant to candidate evaluation. The primary data categories are divided into two groups:

1) "BA" (Below Average): Candidates with qualifications that are assessed to be below the organization's standard thresholds.
2) "Good": Candidates with exceptional qualifications who meet or exceed the organization's criteria for recruitment.

The variables used in this dataset were identified through questionnaires distributed across different departments of the firm. These questionnaires aimed to capture the key factors influencing hiring decisions. By consulting department managers and human resources professionals, we ensured the inclusion of relevant variables, such as technical skills, previous experience, and educational qualifications. This allowed for a more comprehensive view of each candidate's potential.

Data Summary:



Features	Description	Value
Identifiant	ID	integer
Âge	Employe Age	Integer
Sexe	Gender	Male or Female
Statut Marital	Statut	Single or Married
Diplôme	Employee Education Degree	Bachelor High Diploma Master, Phd
Expérience	Nombre d'année d'expérience	Integer
Salaire	Employee salary	Integer
Niveau de Communication	Employee level in communication	1 to 5
Motivation	Employee motivation for work	Yes or No
Langue Parlé	Employee language level	1 to 5
Spécialisation	Employee general Specialisation	Tech, Economy Human ressources law
Adaptabilité à travailler à distance	Employee Proficiency in remote	Yes or no
Séniorité	Employee seniority in the company	Junior Senior Manager
Apptitude physique	Employee ability to work	Yes or no
Formations Personelles	Personal certification	Yes or no
Performance	Employee performance	BA Good

Figure 3. Dataset Overview.

6.2. Data Pre-processing

Before constructing the predictive model, the raw data had to be cleaned and transformed to ensure it was ready for analysis. Pre-processing is a critical step in machine learning that involves converting raw datasets into a usable and structured format. It consists of several sub-tasks, including handling missing data, variable selection, and feature engineering to create additional model performance.

6.2.1. Handling Missing Data

Handling missing data is essential for ensuring the integrity of the model and avoiding bias. In our dataset, missing values were present in various features, such as *Spécialisation* (Specialization), *Langue Parlé* (Spoken Language), and *Formations Personnelles* (Personal Certifications). Simply removing rows with missing data would have led to a reduction in dataset size and introduced potential bias. Instead, we used the k-nearest neighbor (KNN) imputation method to fill in the missing values.

K-nearest neighbor (KNN) imputation is an algorithmic method that replaces missing values by analyzing the most similar instances (neighbors) in the dataset. The distance between the data points is calculated using metrics such as Euclidean distance, and the missing value is filled with the most frequent or average value from the nearest neighbors.

For example, let's assume that an employee's *Langue Parlé* (Spoken Language) value is missing. The KNN algorithm would calculate the similarity between the employee and others based on features such as *Experience*, *Salary*, and *Motivation*. Once the nearest neighbors are identified, the missing language value would be imputed with the most common language value among the closest neighbors.

Mathematically, the KNN imputation process for a missing value x_j can be represented as:

$$x_j = \frac{1}{k}\sum_{i=1}^{nk} x_i$$

Where:

x_j is the missing value being imputed,

k represents the number of nearest neighbors,

x_i represents the values of the nearest neighbors for the feature being imputed.

This approach retains the integrity of the dataset while avoiding the loss of valuable information that would occur with row deletion.

Example: Handling Missing Values in the Dataset Consider the following case where the *Langue Parlé* (Spoken Language) feature has missing values for some employees. For instance, suppose employee ID 205 has a missing value for spoken language. Using the KNN algorithm, the algorithm identifies the three nearest neighbors (based on features like *Experience*, *Salary*, and *Seniority*) who have non-missing values for *Langue Parlé*. If the nearest neighbors have spoken language levels of 4, 5, and 4, the missing value for employee ID 205 is imputed as:

$$\text{Langue Parlé (ID 205)} = \frac{4+5+4}{3} = 4.33$$

The imputed value is then rounded to 4, ensuring that the feature is filled in a way consistent with similar employees.

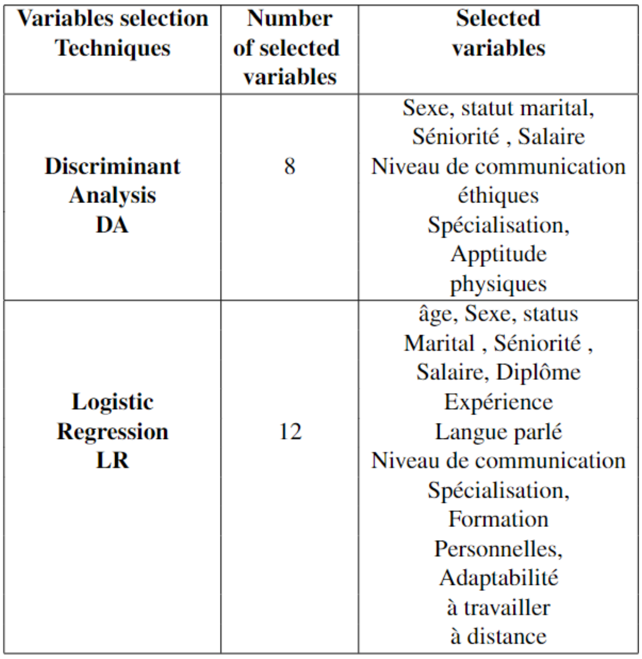
4.3.2. Feature Selection and Transformation

Once the missing values are handled, the dataset undergoes feature selection. This step involves selecting variables that are most relevant to predicting job performance, as discussed in the previous section. After feature selection, feature engineering is performed to transform or create new variables that could improve model performance.

Key features in the dataset include:

1) *Age* (Integer),
2) *Sex* (Male or Female),
3) *Experience* (Number of years),
4) *Diploma* (Education level),
5) *Spoken Language* (Proficiency from 1 to 5),
6) *Salary* (Integer),
7) *Specialization* (Technical, Economics, HR, Law).

These features are standardized and normalized as necessary, ensuring that they are in a suitable format for training the machine learning model.



Variables selection Techniques	Number of selected variables	Selected variables
Discriminant Analysis DA	8	Sexe, statut marital, Séniorité , Salaire Niveau de communication éthiques Spécialisation, Apptitude physiques
Logistic Regression LR	12	âge, Sexe, status Marital , Séniorité , Salaire, Diplôme Expérience Langue parlé Niveau de communication Spécialisation, Formation Personnelles, Adaptabilité à travailler à distance

Figure 4. *Features in the Dataset.*

By effectively handling missing data and applying feature engineering, we ensure that the pre-processed dataset is clean, informative, and ready for machine learning. The next step involves building and training the predictive model based on this data.

6.2.3. Variable Selection and Feature Engineering

In the pre-processing stage, selecting the most important variables is critical to ensuring the accuracy and reliability of the predictive model. Two main techniques were applied: linear regression and discriminant analysis. These methods allowed us to identify and classify variables that had the strongest impact on predicting job performance.

Linear Regression for Variable Selection Linear regression was employed to quantify the relationship between the independent variables (candidate features) and the dependent variable (job performance). This method determines the contribution of each feature to the overall prediction and helps in selecting variables with the greatest predictive power.

The linear regression equation is represented as:

$$Y = \beta_0 + \beta_1X_1 + \beta_2X_2 + ... + \beta_nX_n + \epsilon$$

Where:

Y represents the predicted job performance,

β_i represents the coefficients (weights) that show the strength of the relationship between the independent variable X_i and the outcome Y,

ϵ is the error term.

In this context, the variables with statistically significant β values (p-values less than 0.05) were selected for the model, indicating that these variables had a significant effect on predicting job performance. During the variable selection process, a Variance Inflation Factor (VIF) threshold of 5 was applied to detect multicollinearity among the features. Any variables with a VIF greater than 5 were removed to avoid overfitting and redundancy.

Through this method, the following variables were selected:

1) Age
2) Sex
3) Marital Status
4) Seniority
5) Salary
6) Diploma
7) Experience
8) Spoken Language
9) Communication Skills
10) Specialization
11) Personal Training
12) Adaptability to Remote Work

Discriminant Analysis for Variable Classification Discriminant analysis was applied as a second method to classify the selected variables and validate their effectiveness in predicting job performance. Discriminant analysis seeks to find the combination of variables that best separates predefined groups—candidates who are likely to succeed versus those who are not. The discriminant function is given by:

$$D = \alpha_1X_1 + \alpha_2X_2 + ... + \alpha_nX_n$$

Where:

D is the discriminant score,

X_i are the predictor variables,

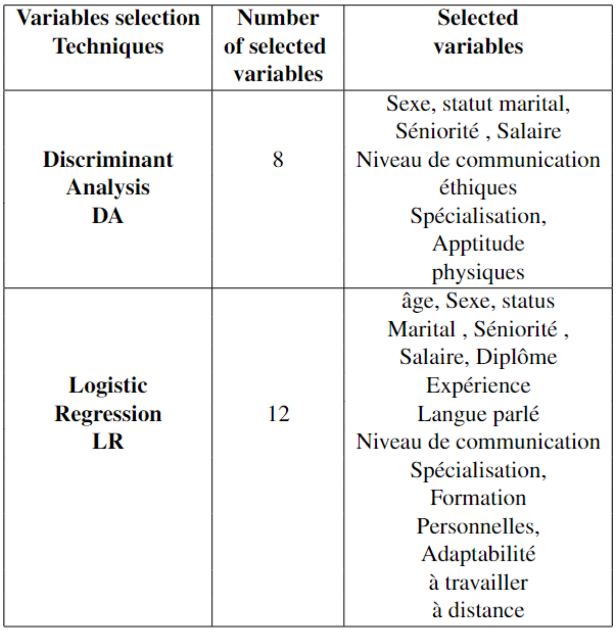
α_i are the coefficients that maximize the separation between groups (e.g., "Good" and "Below Average" candidates).

To determine whether the selected variables had significant discriminating power, we used Wilks' Lambda, which measures the proportion of variance in the discriminant scores not explained by group differences. A Wilks' Lambda value close to 0 indicates strong discriminating power. In our case, we achieved a Wilks' Lambda value below 0.3, confirming that the selected features provided effective classification between high-performing and low-performing candidates.

The discriminant analysis led to the selection of the following variables:

1) Sex
2) Marital Status
3) Seniority
4) Salary
5) Ethics Communication Level
6) Specialization
7) Physical Aptitude

Final Variable Set By combining the results of linear regression and discriminant analysis, a final set of 15 variables was selected. These variables were found to have the most impact on predicting job performance while maintaining statistical robustness and practical relevance.



Variables selection Techniques	Number of selected variables	Selected variables
Discriminant Analysis DA	8	Sexe, statut marital, Séniorité , Salaire Niveau de communication éthiques Spécialisation, Apptitude physiques
Logistic Regression LR	12	âge, Sexe, status Marital , Séniorité , Salaire, Diplôme Expérience Langue parlé Niveau de communication Spécialisation, Formation Personnelles, Adaptabilité à travailler à distance

Figure 5. Selected Variables from Discriminant Analysis and Logistic Regression.

6.3. Model Building

In the model-building phase, we applied supervised machine learning techniques, given that the dataset included labeled outcomes (i.e., whether a candidate was successfully hired and performed well). This made it possible to use supervised learning algorithms, where the model learns from a labeled dataset and predicts outcomes for new data. The dataset was split into training and testing sets, with 80% of the data used for training the model and 20% reserved for testing. This approach ensures that the model's performance can be evaluated on unseen data, which helps generalize predictions to real-world scenarios.

6.3.1. Supervised Learning Approach

The predictive model was developed using an Artificial Neural Network (ANN), which was optimized with the Stochastic Gradient Descent (SGD) algorithm to adjust weights and improve the learning process iteratively. These two methods work together to enhance the accuracy and generalization capability of the model.

Artificial Neural Network (ANN) An Artificial Neural Network (ANN) was chosen for this study due to its capacity to model complex, non-linear relationships between variables. ANNs are inspired by the structure of the human brain, consisting of layers of interconnected nodes (neurons) that work together to process input data and generate predictions.

The architecture of the neural network used for this model consists of:

1) Input layer: This layer takes the selected input features, such as *age*, *experience*, *specialization*, etc.
2) Hidden layers: These layers perform complex transformations on the input data by applying weights, biases, and activation functions. The *ReLU (Rectified Linear Unit)* activation function was used for hidden layers to learn non-linear relationships.
3) Output layer: This layer outputs the predicted label using a *softmax activation function* to assign probabilities to each class.

The forward propagation mechanism allows the input data to flow through the network, while backpropagation calculates the error and adjusts the weights to minimize the error during training.

Stochastic Gradient Descent (SGD) Stochastic Gradient Descent (SGD) was used to optimize the learning process of the ANN by adjusting the model's parameters iteratively. SGD updates model parameters based on a small batch of data, which makes it computationally efficient for large datasets.

The key parameters used for the SGD optimization process in this model include:

Table 1. *SGD Parameters.*

Parameters	Stochastic gradient descent
Learning Rate	0.01
Momentum	0.9
Lot Size	32
Number of epochs	100
Regularization	0.001

The SGD algorithm updates the weights of the model based on the following formula:

$$w_{t+1} = w_t - \eta \cdot \nabla L(w_t)$$

Where:

w_{t+1} represents the updated weights,
w_t represents the current weights,
η is the learning rate,
$\nabla L(w_t)$ is the gradient of the loss function at time step t.

This process is repeated over each mini-batch of data, and the inclusion of momentum ensures that the weight updates are smooth, accelerating learning in the correct direction while preventing oscillation.

6.3.2. Training and Testing Phases

To ensure robustness, the dataset was split into training (80%) and testing (20%) sets. During the training phase, the model learned the relationships between the input features and the target variable (job performance). In each epoch, the model adjusted its weights based on the SGD algorithm. The testing phase involved evaluating the model's performance on unseen data, ensuring that it generalizes well beyond the training set. The following metrics were used to assess the model's performance:

1) Accuracy: The percentage of correctly predicted instances.
2) Precision: The ratio of true positives to the sum of true positives and false positives.
3) Recall: The ratio of true positives to the sum of true positives and false negatives.
4) F1 Score: The harmonic mean of precision and recall, which balances both metrics for imbalanced datasets.

By combining discriminant analysis for feature selection with SGD for optimization, we created a highly efficient and accurate model capable of predicting candidate performance effectively.

7. Results and Discussion

7.1. Study Results

After identifying the optimal variables that significantly impact the target variable (candidate performance), we proceeded to define the architecture for our predictive model. To assess model performance, we compared different learning algorithms, focusing on hybrid neural networks optimized with Stochastic Gradient Descent (SGD) and evaluated against a non-optimized baseline.

We experimented with various neural network architectures. The most effective topology for our hybrid neural network used 12, 18, and 1 neurons across the input, hidden, and output layers, respectively. The 12 input neurons correspond to the variables selected using Logistic Regression (LR), demonstrating their strong discriminatory power in predicting candidate performance. The hidden layer used 18 neurons to capture complex relationships between input features, while the output layer provided a binary classification (Below Average/Good).

As anticipated, the application of SGD as an optimization technique consistently produced favorable outcomes. The hybrid neural network models were evaluated using Logistic Regression (LR) and Discriminant Analysis (DA) for feature selection, followed by training both models with and without SGD. This approach enabled us to assess the comparative performance and efficacy of the hybrid models.

7.2. Performance Evaluation

The performance of each model was evaluated using the following key metrics: Accuracy, Precision, Sensitivity, Specificity, F-measure, and class-specific metrics for Below Average (BA) and Good candidates. These metrics provide a well-rounded evaluation of the predictive model, especially considering the binary nature of the classification problem.

Table 2. Performance Metrics Across Models.

Metric	Value
Accuracy	75.2% (LR&ANN&SGD) 72.5% (LR&ANN) 68.5% (DA&ANN&SGD) 65.0% (DA&ANN)
Precision	73.6% (LR&ANN&SGD) 70.1% (LR&ANN) 52.1% (DA&ANN&SGD) 47.5% (DA&ANN)
Sensitivity	76.2% (LR&ANN&SGD) 73.1% (LR&ANN) 76.3% (DA&ANN&SGD) 74.9% (DA&ANN)
Specificity	74.5% (LR&ANN&SGD) 72.0% (LR&ANN) 62.8% (DA&ANN&SGD) 60.3% (DA&ANN)
F-measure	74.8% (LR&ANN&SGD) 71.6% (LR&ANN) 60.5% (DA&ANN&SGD) 58.1% (DA&ANN)
BA	76.1% (LR&ANN&SGD) 74.9% (LR&ANN) 85.2% (DA&ANN&SGD) 83.4% (DA&ANN)
Good	73.6% (LR&ANN&SGD) 70.1% (LR&ANN) 52.1% (DA&ANN&SGD) 47.5% (DA&ANN)

7.3. Analysis of Results

Accuracy The LR&ANN with SGD model achieves the highest overall accuracy at 75.2%, highlighting its superior ability to correctly classify both "Below Average" and "Good" candidates. The addition of SGD significantly improves accuracy across both the LR and DA models, suggesting that the optimization method plays a crucial role in reducing error and enhancing prediction capabilities.

Precision and Specificity The precision metric, which measures the ability to avoid false positives, is highest for LR&ANN with SGD at 73.6%, indicating that this model makes fewer incorrect predictions when classifying candidates as "Good." This is further supported by the model’s high specificity (74.5%), which measures the ability to correctly identify candidates who do not perform well.

Sensitivity and F-measure Sensitivity refers to the model’s ability to correctly identify true positives. The LR&ANN models achieve high sensitivity, particularly with SGD (76.2%). The F-measure, balancing precision and recall, further confirms LR&ANN with SGD as the most effective model (74.8%).

7.3.1. Class-specific Performance

The BA (Below Average) metric is particularly high in the DA&ANN with SGD model, reaching 85.2%, while the Good measure is highest for LR&ANN with SGD (73.6%).

7.3.2. Impact of Stochastic Gradient Descent

The use of SGD had a notable impact on model performance, improving accuracy, precision, and generalization in both LR and DA models.

7.3.3. Conclusion of Results

The LR&ANN with SGD model emerges as the most reliable for predictive hiring. The study highlights the importance of feature selection and optimization techniques for job performance prediction.

8. Conclusion

his study presents a hybrid approach to predictive hiring, leveraging Artificial Neural Networks (ANNs) optimized with Stochastic Gradient Descent (SGD) and supported by robust feature selection methods. By integrating Logistic Regression (LR) and Discriminant Analysis (DA) for variable selection, we addressed key challenges in predictive hiring, such as model accuracy and computational efficiency. Logistic Regression proved superior in identifying high-impact features, facilitating better predictive outcomes when paired with neural networks.

The inclusion of SGD optimization significantly enhanced the learning process by improving convergence and balancing model precision and recall. These improvements make the framework scalable and adaptable to large, complex datasets often encountered in hiring environments. Although DA demonstrated strengths in specific contexts, its overall performance lagged compared to the LR-based model, reinforcing the importance of selecting appropriate techniques for feature selection.

The practical implications are clear: integrating advanced machine learning methodologies into recruitment workflows enables organizations to refine candidate evaluations, reduce inefficiencies, and improve decision-making. This framework offers a scalable solution for enhancing the quality and efficiency of hiring practices, ultimately contributing to organizational success by increasing the likelihood of selecting the best candidates.

In conclusion, this research validates the potential of combining neural networks, SGD optimization, and effective feature selection methods to modernize and enhance predictive hiring systems. The proposed framework not only improves performance but also lays a foundation for future advancements in AI-driven recruitment strategies.

Abbreviations

AI	Artificial Intelligence
ML	Machine Learning
SGD	Stochastic Gradient Descent
LR	Logistic Regression
DA	Discriminant Analysis
HR	Human Resources
HRM	Human Resource Management
NLP	Natural Language Processing
ANP	Analytical Network Process
SHAP	SHapley Additive exPlanations
LIME	Local Interpretable Model-agnostic Explanations
KNN	K-Nearest Neighbors

Conflicts of Interest

The authors declare no conflicts of interest.

References

[1] Z. H. Zhou, K. Chen, and X. Y. Dai, “Data-driven intelligence in hiring and employee selection,” *International Journal of Data Science and Analytics*, vol. 11, pp. 221–234, 2021.

[2] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016.

[3] T. A. Leopold, V. Ratcheva, and S. Zahidi, “The future of jobs report 2018,” *World Economic Forum*, 2018.

[4] E. Faliagka, A. Tsakalidis, and G. Tzimas, “An integrated e-recruitment system for automated personality mining and applicant ranking,” *Internet Research*, vol. 22, pp. 551–568, 2012.

[5] D. He, X. Yuan, and H. Li, “Ai and talent acquisition: The emerging frontier,” *Journal of Business Research*, vol. 112, pp. 140–148, 2020.

[6] H. Heidari, A. Ferrario, and M. B. Zafar, “Fairness in machine learning: From statistical to causal definitions,” *Data Mining and Knowledge Discovery*, vol. 34, pp. 453–495, 2020.

[7] F. S. Guenole, N. *The Business Case for AI in HR: With Insights and Tips on Getting Started. IBM Smarter Workforce Institute*, 2018.

[8] D. G. W. S.. T. N. Weller, C., “Predictive hiring: Ai applications for predicting success in recruitment.,” *Journal of Machine Learning Research*,, pp. (110), 1–34, 2020.

[9] M. M. E.. X. C. Chen, L., “Intelligent recruiting system by analyzing interview data based on machine learning algorithms.,” *IEEE Access, 7*, pp. 134539–134553, 2019.

[10] D. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *International Conference on Learning Representations (ICLR)*, 2014.

[11] J. A. L. G. Nemirovski, A. and A. Shapiro, “Robust stochastic approximation approach to stochastic programming,” *Siam Journal of Optimization*, vol. 19, pp. 1574–1609, 2009.

[12] P. Tseng, “An incremental gradient (-subgradient) method with momentum term and adaptive stepsize rule,” *SIAM Journal on Optimization*, vol. 10, pp. 323–345, 1998.

[13] L. Bottou, “Large-scale machine learning with stochastic gradient descent.,” *Proceedings of COMPSTAT’2010, Physica-Verlag HD*,, pp. 177–186., 2010.

[14] M. J. D. G.. H. G. Sutskever, I., “On the importance of initialization and momentum in deep learning,” *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*,, pp. 1139–1147, 2013.

[15] H. Robbins and S. Monro, “A stochastic approximation method,” *Ann. Math. Stat.*, vol. 22, pp. 400–407, 1951.

[16] R. Johnson and T. Zhang, “Accelerating stochastic gradient descent using predictive variance reduction,” vol. 26, pp. 315–323, 2013.

[17] L. Bottou and O. Bousquet, “The tradeoffs of large scale learning,” *Advances in Neural Information Processing Systems (NIPS)*, vol. 20, pp. 161–168, 2008.