Journal of Information Systems and Informatics
Vol. 7, No. 2, June 2025 e-ISSN: 2656-4882 p-ISSN: 2656-5935
DOI: 10.51519/journalisi.v7i2.1129 Published By DRPM-UBD

Integration of Hash Encoding Technique with Machine Learning for Employee Turnover Prediction

Ahya Radiatul Kamila[1], Johanes Fernandes Andry[2], Francka Sakti Lee[3] Felliks F. Tampinongkol[4]

[1,4]Data Science Department, Bunda Mulia University, Jakarta, Indonesia
[2,3]Information System Department, Bunda Mulia University, Jakarta, Indonesia
Email: [1]akamila@bundamulia.ac.id

Abstract

Employee turnover refers to the replacement of employees within an organization, which can lead to losses such as recruitment costs and decreased productivity. Predicting turnover is crucial for companies to anticipate and take appropriate actions to retain potential employees. This study aims to optimize the employee turnover prediction model by integrating hash encoding techniques and machine learning. The dataset used in this study is an open-source dataset obtained from Kaggle dataset. It consists of 14,994 rows and 10 columns (features) representing employee-related information such as satisfaction level, evaluation score, number of projects, average monthly hours, and whether the employee left the company. Among these features, some are of object data type. Since machine learning algorithms generally cannot work directly with object-type features, the use of hash encoding is proposed. This technique converts object-type data into numerical data. It is part of the preprocessing stage, aiming to reduce memory usage, speed up data preprocessing, and improve model performance. After preprocessing is completed, the prediction model is trained using the Random Forest algorithm to predict employee turnover. The evaluation is conducted using accuracy, recall, precision, and F1-score metrics, which yielded results of 0.988, 0.961, 0.988, and 0.974, respectively. These results indicate that the integration of hash encoding techniques and machine learning can produce a well-performing model for predicting employee turnover.

Keywords: Hash Encoding, Machine learning, Turnover Prediction, Random Forest

1. INTRODUCTION

Turnover is a term used in human resource management to describe the replacement of employees within an organization [1]. The turnover rate can be measured by the number of employees who voluntarily leave a company within a certain period [2]. A high turnover rate is generally considered problematic, as it can lead to significant losses for the company [3]. With a high turnover rate, the company is required to find replacement employees to ensure smooth operations and maintain workflow continuity. Moreover, high turnover may result in a loss of

knowledge and skills that are difficult for new employees to immediately replace. Consequently, work processes and team performance can be disrupted, ultimately affecting the company's overall productivity [4]. Therefore, predicting employee turnover is crucial, enabling companies to anticipate spikes in turnover and take appropriate steps to retain valuable employees.

Number of studies have been conducted to control employee turnover rates through analytical approaches using machine learning algorithms. One notable study in this context [5] compared the performance of several machine learning algorithms and found that Logistic Regression achieved the highest accuracy of 87.71%. A similar study was conducted by [6], who compared various machine learning algorithms for the same objective, with the best result obtained from the K-Nearest Neighbor algorithm, which achieved 84% accuracy after preprocessing with One-Hot Encoding. Additionally, a Principal Component Analysis (PCA)-based approach for feature dimensionality reduction, followed by training using a Support Vector Machine (SVM), was applied by [7], achieving a model accuracy of 95.1%.

On the other hand, an innovative approach by [8] converted tabular data into knowledge graphs and attained an accuracy of 92.5%. Meanwhile, [9] presented an advanced predictive approach to assessing turnover intentions among new employees using Logistic Regression, K-Nearest Neighbor, and Extreme Gradient Boosting (XGB) algorithms, based on data from university graduates in South Korea. They found that XGB delivered the highest accuracy of 78.5% and revealed that traditional factors such as workload and major-field fit were no longer significant predictors, while job security emerged as the most important factor influencing turnover intention. While these studies demonstrate the potential of various machine learning algorithms in predicting turnover, most of them rely on traditional encoding techniques such as One-Hot Encoding or Label Encoding, which may not perform efficiently when handling categorical features with high cardinality.

These approaches often lead to increased dimensionality, resulting in higher memory consumption and potential model overfitting. Moreover, although some studies have employed dimensionality reduction methods like PCA, they may lose interpretability and fail to preserve essential relationships within the data. Furthermore, limited research has explored the effectiveness of hash encoding—an efficient method for handling categorical features with large numbers of unique values in the context of turnover prediction. This indicates a gap in the existing literature regarding preprocessing strategies that balance model performance, efficiency, and scalability.

p-ISSN: **2656-5935** http://journal-isi.org/index.php/isi e-ISSN: **2656-4882**

This study aims to implement a supervised learning machine learning algorithm to detect early signs of potential employee turnover based on data with specific characteristics. Supervised learning in machine learning algorithms is a method that enables machines to learn by identifying data patterns to predict future events [10]. This makes machine learning a reliable method for identifying turnover potential, allowing companies to be more proactive in taking strategic steps to retain employees. However, machine learning algorithms cannot directly work with object-type data, so an encoding process is required to convert such data into numerical types before modelling can be performed. In this study, the encoding process is carried out using the hashing method to convert object-type data into numerical data. This method is chosen due to its ability to efficiently handle data with many unique categories without adding excessive feature load to the model. After the encoding process, the data is trained using an ensemble learning algorithm, which is known for its capability to combine predictions from multiple base models to improve prediction accuracy and stability [11]. This approach is expected to capture complex relationships between features while also reducing the risk of overfitting in high-dimensional datasets.

2. METHODS

The approach used in this study is classification-based, as the target variable in the dataset is discrete, specifically, the employee turnover status (yes or no). The objective of this research is to predict whether an employee will leave the company or remain employed based on relevant features. One of the challenges in this process is the presence of categorical features in the dataset, which cannot be directly used in machine learning algorithms. To address this issue, this study proposes the use of hash encoding as a technique for encoding categorical features. Hash encoding is expected to improve prediction accuracy by creating more efficient numerical representations and preserving the informational integrity between categories. This approach is chosen because hash encoding can handle datasets with diverse categories without significantly increasing the dataset's dimensionality, as often occurs with one-hot encoding [12]. Furthermore, hash encoding offers faster computation and helps prevent overfitting, enabling the model to be trained more efficiently and yield more accurate results.

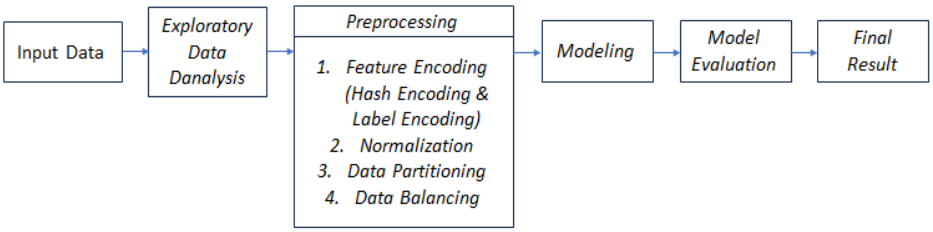

Figure 1. Block Diagram of Predictive Analysis Model

2.1. Exploratory Data Analysis and Preprocessing

Exploratory Data Analysis (EDA) is the initial process in data analysis aimed at understanding the structure, patterns, and characteristics of the data before proceeding to the modeling or further analysis stage [13]. Through EDA, anomalies, hidden patterns, relationships between variables, and potential issues within the data can be identified. Following this exploratory stage, the process continues with preprocessing, which consists of a series of steps to prepare and clean the data for use in analysis or model development. The purpose of preprocessing is to improve data quality and ensure that the data can be optimally processed by machine learning algorithms [14].

2.1.1. Feature Encoding

In machine learning model development, most algorithms cannot work directly with categorical data, as they are designed to process numerical inputs. Therefore, a feature encoding process is performed to convert categorical data into numerical representations that can be interpreted by the model [15]. This step is crucial to ensure that the learning algorithms can effectively extract patterns and relationships from the data during the training phase. In this study, there are two categorical columns, namely the "Salary" and "Department" columns, each containing several unique categories. To address this, an encoding approach is applied that takes into account the meaning and the number of classes of each categorical feature. One such technique employed is hash encoding, a method that uses a hash function to map unique categories into fixed-size numerical representations in the form of vectors [16]. Unlike one-hot encoding, which can significantly increase the dimensionality of a dataset when the number of categories is large, hash encoding limits the number of encoded columns to a specified size, known as the "hashing space." This technique is particularly useful when working with datasets that contain a large number of unique categories, as hash encoding can handle them without drastically increasing the data's dimensionality.

The hashing process works by taking a categorical value and applying it to a hash function, which then produces a numerical value based on the defined hashing space [17]. Although hash encoding is more efficient in terms of computation and storage, it does have the potential for "collisions," where two different categories are mapped to the same hash value. However, with a sufficiently large hashing space, the impact of these collisions is generally negligible. Hash encoding is especially valuable in scenarios involving large categorical datasets or continuously growing data [18]. Fundamentally, the main goal of hashing is to generate a unique numerical value for each element. This process involves bit manipulation, modulus division, and other techniques to ensure a relatively unique and fast hash result with two main characteristics: determinism and uniform distribution. Determinism

refers to the property of hash encoding that always produces the same output for the same input. On the other hand, uniform distribution ensures that the values generated from different inputs are evenly spread across the hash space.

$$y = x \oplus k \tag{1}$$

Where $\oplus$ represents the XOR operation. This operation is commonly used in many hashing algorithms to distribute data more evenly.

$$h(x) = (\sum_{i=1}^{n} ASCII(x_i).p^{i-1}) \bmod N \tag{2}$$

Where :
ASCII : ASCII value from x_i character
p : Basic for Polynomial
N : The modulus value is used to constrain the hash values within a specific range.

2.1.2. Normalization

Normalization is a data preprocessing process aimed at transforming feature values in a dataset into the same scale or a specific range, such as 0 to 1. The goal is to ensure that each feature carries equal weight during the modeling process, especially for machine learning algorithms that are sensitive to scale differences between features [19]. With normalization, the model can more easily learn from the data, as features with large scales will not dominate those with smaller scales. Normalization can also improve model performance and convergence speed, as well as prevent bias in predictions [20]. One of the most popular normalization methods is the Min-Max Scaler, which transforms feature values into a range between 0 and 1.

$$\hat{x} = \frac{x - x_{min}}{x_{max} - x_{min}} \tag{3}$$

2.1.3. Data Partitioning

Data Partitioning is a critical step in the machine learning model development process to ensure that the resulting model performs well and can generalize to new, unseen data. In this study, the dataset is divided into three subsets: 80% for training, 10% for validation, and 10% for testing. The training set is used to train the model by allowing it to learn patterns and relationships from the data. The validation set is utilized during the model development phase to fine-tune hyperparameters, compare different models, and monitor performance, helping to prevent overfitting and ensure that the model generalizes well. The testing set, on

the other hand, is kept completely separate and is only used for the final evaluation of the model. It provides an unbiased estimate of the model's real-world performance on previously unseen data [21]. This three-way data partitioning strategy ensures that the model is trained effectively, tuned accurately, and evaluated fairly, leading to more robust and reliable predictive performance.

2.2. Modeling

Modeling is a stage in the machine learning process where the model is built, trained, and evaluated using a previously processed dataset [22]. In this stage, machine learning algorithms are applied to learn patterns in the data, with the goal of generating predictions or making decisions based on new data. This stage involves several steps, including selecting an appropriate algorithm based on the type of data and problem at hand, as well as training the model with hyperparameter optimization.

This study employs the Random Forest algorithm, which is an ensemble of weak learners—specifically, decision trees. The algorithm works by constructing multiple decision trees during the training process and combining their outputs to improve prediction accuracy and reduce the risk of overfitting [23]. A large number of decision trees are built using subsets of the data that are randomly sampled with replacement through the bootstrap sampling technique. Each decision tree is trained on a different subset of the data and, at each splitting step, is influenced only by a random subset of the available features [24]. In classification tasks, each decision tree in the random forest outputs a class prediction. The final prediction for class C for a given dataset X is determined by majority vote from all the individual decision trees [25].

$$\hat{y}=majority\_vote(\sum_{t=1}^{T} h_t(X)) \quad (4)$$

3. RESULTS AND DISCUSSION

This section will explain the results and discussion divided into three main parts. The first part discusses the results and discussion of the Exploratory Data Analysis (EDA) stage, the second part covers the results and discussion of the data preprocessing stage, and the final part addresses the results and discussion of the model performance after training and evaluation.

Model evaluation aims to measure the model's performance in making predictions on new data. This evaluation stage is conducted after the model has been trained using training data and tested on separate testing or validation data. Model evaluation is essential to ensure that the developed model has good generalization

ability and does not overfit the training data. Various evaluation metrics are used, depending on the objectives and the type of problem, to objectively assess the model's performance.

3.1. Exploratory Data Analysis

In this study, bivariate and multivariate analyses were conducted to explore the relationships between variables and to understand the underlying patterns in the data. The bivariate analysis used violin plots to examine the relationship between numerical features and the target variable. This approach allows us to understand the distribution patterns and variability within each target category, providing insights into whether the feature distributions differ significantly across the target categories. Figure 2 illustrate Bivariate Analysis Visualization.

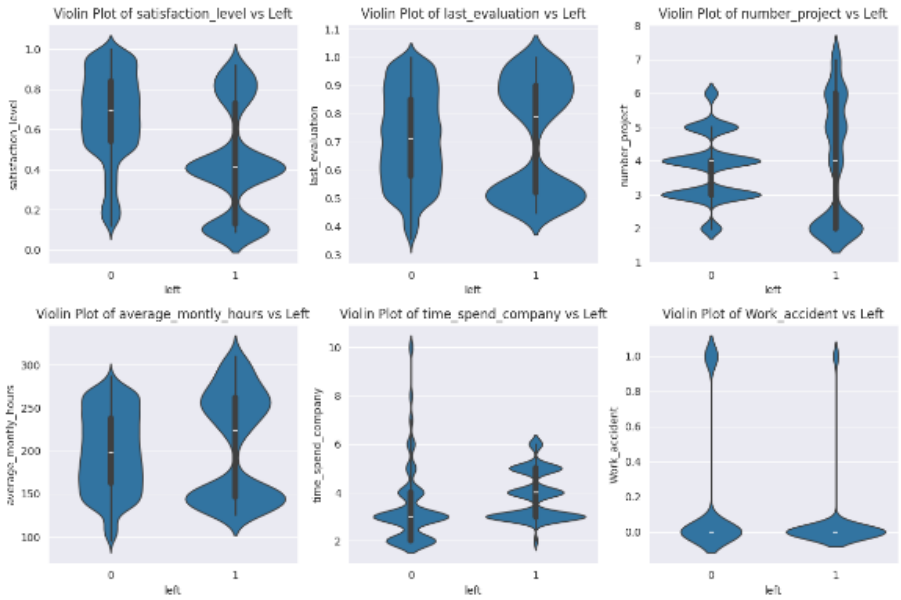

Figure 2. Bivariate Analysis Visualization

On the other hand, for the multivariate analysis, this study uses a correlation heatmap to visualize the relationships among numerical features in the dataset. This allows us to identify which features have a high correlation with the target variable or with other features. A high correlation between two features may indicate a strong linear relationship, while a low or zero correlation suggests no significant relationship. The strength and direction of the linear relationship

between two numerical variables X and y is typically quantified using the Pearson correlation coefficient r.

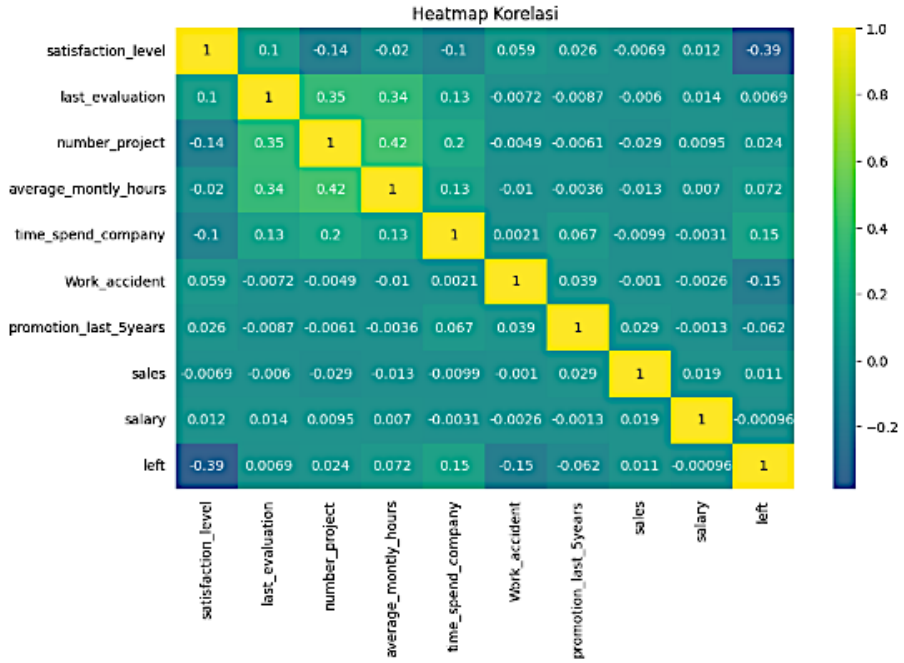

Figure 3. Multivariate Analysis Visualization

$$r_{XY} = \frac{\sum_{i=1}^{n}(X_i-\bar{X})(Y_i-\bar{Y})}{\sqrt{\sum_{i=1}^{n}(X_i-\bar{X})^2}\sqrt{\sum_{i=1}^{n}(Y_i-\bar{Y})^2}} \quad (5)$$

By visualizing these coefficients in a heatmap, multicollinearity issues among independent variables can be quickly detected, and predictors that have the strongest linear association with the target variable can be identified, which is particularly useful for feature selection and dimensionality reduction. In addition, a bar plot was used to analyse the class balance of the target variable. This analysis helps to determine the distribution of sample counts across each target class. The results of this analysis are then used to design an optimal data preprocessing strategy, such as handling multicollinearity, feature selection, or variable transformation, in order to improve the performance of the employee turnover prediction model.

p-ISSN: **2656-5935** http://journal-isi.org/index.php/isi e-ISSN: **2656-4882**

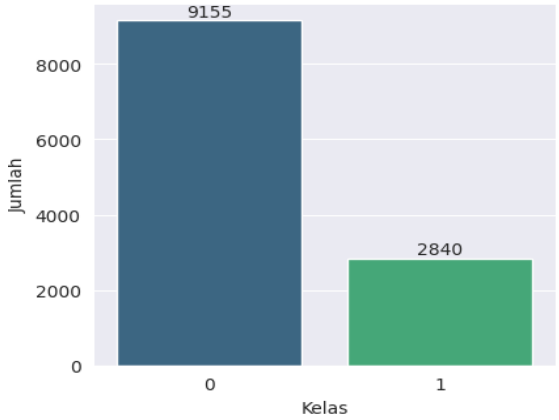

Figure 4. Class Distribution of the Target Feature

3.2. Preprocessing

In this study, four preprocessing steps were carried out: feature encoding, normalization, data partitioning, and data balancing. Each approach was tailored to suit the dataset and the objectives of the study. For feature encoding, hash encoding was applied to several categorical features such as "Gender", "MaritalStatus", "Travelling", "Vertical", "EducationField", "Role", and "OverTime", all of which originally contained string or object-type values. This technique converts each category into an integer within a predefined range using Python's hash() function combined with modulus division, specifically through the operation hash(x) % 100, to constrain the values within the range of 0–99. Hash encoding is particularly effective in managing high-cardinality categorical variables and in preventing the dimensional explosion commonly caused by One Hot Encoding. While this method introduces the possibility of collisions—where different categories may be mapped to the same value—the risk can be minimized by selecting a sufficiently large hash space. In return, it offers considerable benefits in memory efficiency and computational speed. Moreover, hash encoding is deterministic, ensuring that the same input value will always yield the same encoded output across different phases of model development, including training and inference. In addition, Label Encoding was used for the "Salary" feature, which comprises ordinal categories: "Low", "Medium", and "High". This method preserves the intrinsic order of the categories, allowing the algorithm to recognize that "High" ranks above "Medium", and "Medium" ranks above "Low". Such encoding is especially suitable for tree-based models or algorithms that can exploit ordinal relationships to improve prediction accuracy.

An additional categorical feature, "Department", which consists of 10 unique categories, was also encoded using hash encoding. Compared to One Hot Encoding, which would have resulted in 10 additional binary columns, hash

p-ISSN: **2656-5935** http://journal-isi.org/index.php/isi e-ISSN: **2656-4882**

encoding reduced feature dimensionality, minimized the risk of the curse of dimensionality, and enhanced computational efficiency. The transformed features were then visualized using kernel density plots to examine how each encoding method affected the distribution of the data. This step was important to assess whether encoding introduced distortions or preserved meaningful patterns for model training. The hash-encoded variables showed reasonably uniform distributions across their hash space, confirming their readiness for modeling without significant bias.

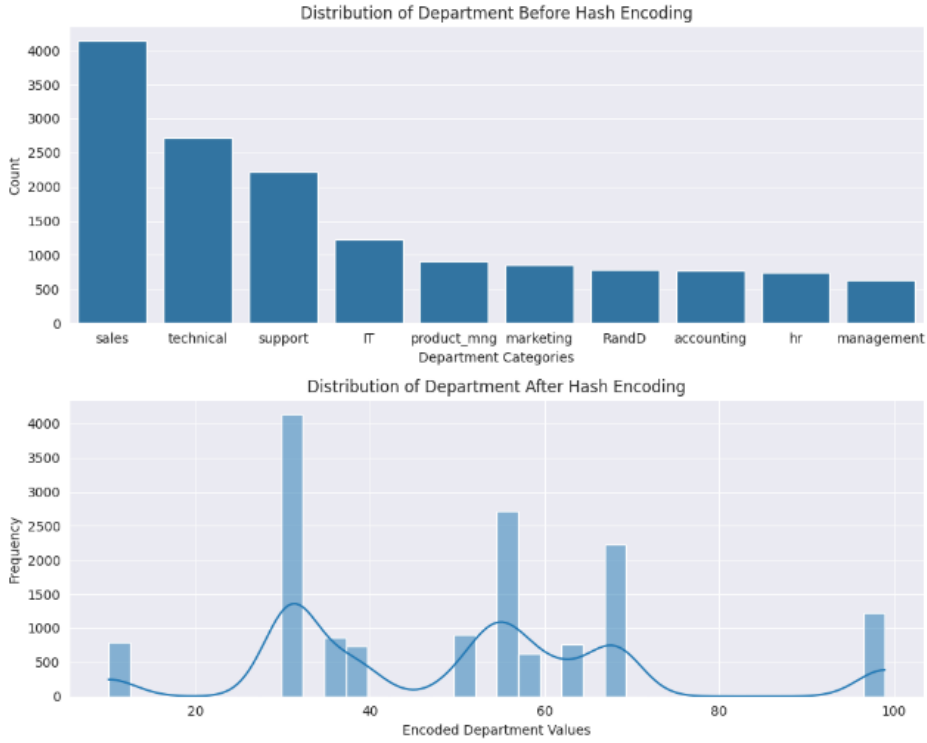

Figure 5. Data Distribution After Encoding the "Department" Feature

In this study, the number of features before and after encoding remained the same for both encoding techniques used. This indicates that the encoding process was applied correctly, without causing any changes to the number of columns in the dataset. In other words, the information was preserved and no feature representations were lost after encoding, which is a crucial step before proceeding to the modeling stage.

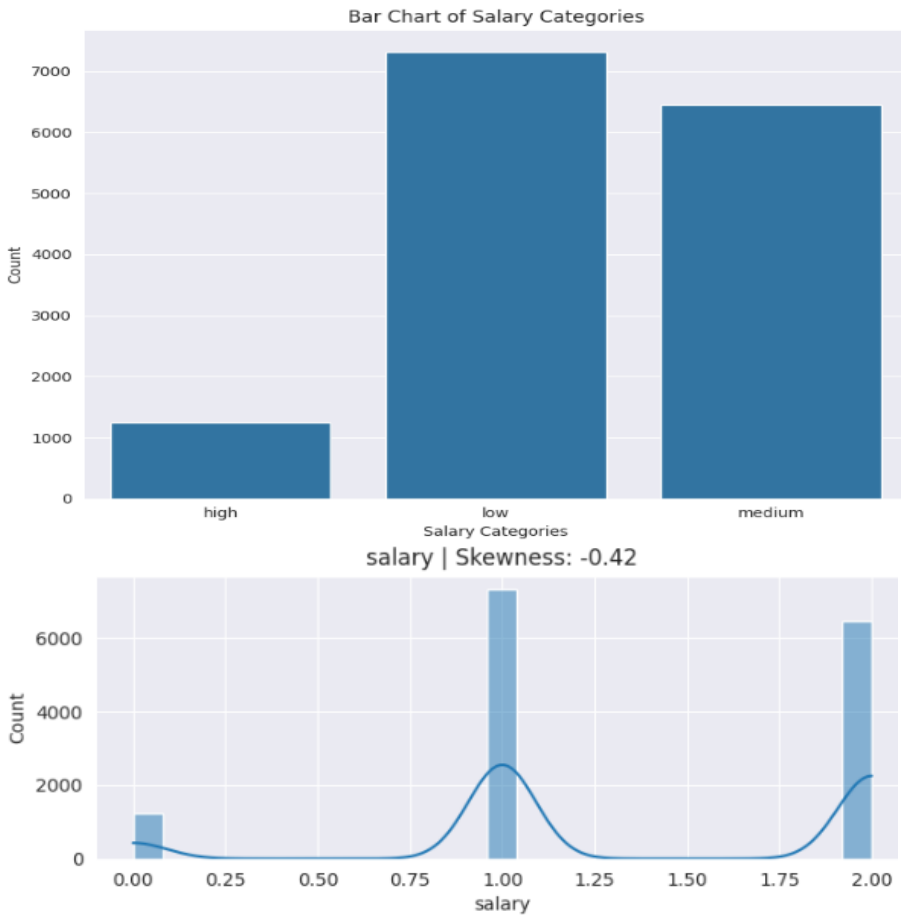

Figure 6. Data Distribution After Encoding the "Salary" Feature

Following the encoding process, addressing the class distribution imbalance became the next important step. This imbalance can affect the performance of machine learning models, especially in recognizing the minority class. To overcome this issue, a data balancing process was carried out using the oversampling technique. This technique works by increasing the number of samples in the minority class until it reaches a balanced proportion with the majority class. The goal is to enable the model to learn fairly from both classes, thereby improving predictive performance and reducing bias toward the majority class.

p-ISSN: **2656-5935** http://journal-isi.org/index.php/isi e-ISSN: **2656-4882**

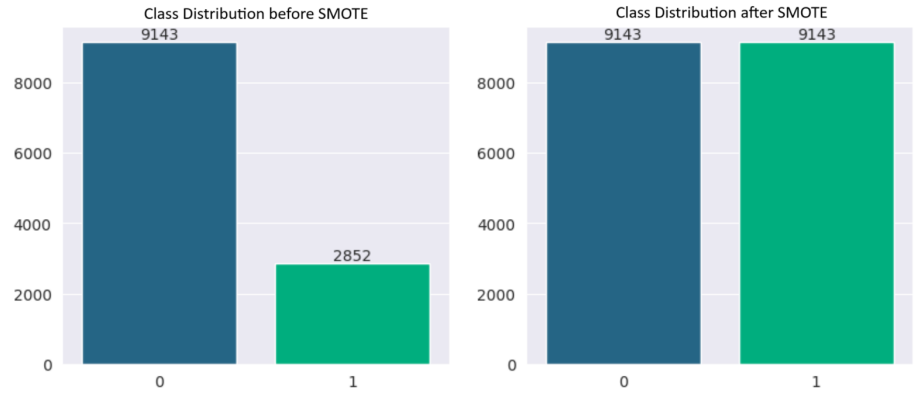

Figure 7. Visualization of Class Distribution Before and After Applying the SMOTE Approach

The visualization above represents the successful balancing of data using the SMOTE algorithm by increasing the number of samples in the minority class. Initially, the minority class had 2,852 samples, which were then synthetically augmented to 9,143 samples. This synthetic data generation was performed after the data partitioning step, and only on the training data. Once the samples in the target class were balanced, the dataset was considered ready to be modeled using machine learning algorithms. A dataset with a balanced target class is expected to produce a model with good generalization ability by reducing the likelihood of bias caused by the model underearning from the minority class.

3.3. Model Performance

The model performance in this study was evaluated using classification performance metrics. Four performance metrics were calculated: accuracy, recall, precision, and F1-score. Among these classification metrics, two main metrics were selected as key indicators of the model's success: accuracy and recall. These two metrics were chosen due to their relevance to the study's objectives. The other performance metrics were used to assess the balance between positive and negative classes and to provide a more comprehensive overview of the overall model quality.

Accuracy measures the percentage of correct predictions out of the total number of predictions made by the model. On the other hand, recall focuses on minimizing false negatives. This metric is suitable for predicting employee turnover because false negatives have a significant impact in this context. A false negative occurs when the model incorrectly predicts that an employee will not leave, while in reality, the employee does leave. Such a prediction prevents the company from taking preventive measures to retain employees at risk of leaving.

p-ISSN: **2656-5935** http://journal-isi.org/index.php/isi e-ISSN: **2656-4882**

Tabel 1. Model Peerformance

Metrics	Model Performance
Accuracy	0.988
Recall	0.961
Precision	0.988
F1-Score	0.974

The results in Table 1 demonstrate the robustness of the model, which integrated the hash encoding technique with a machine learning classifier. The model achieved a high accuracy of 98.8%, indicating strong overall performance. More importantly, the recall score of 96.1% highlights the model's ability to effectively identify employees who are likely to leave. Furthermore, the precision score of 98.8% suggests that the employees predicted to leave by the model are indeed likely to leave, minimizing false positives. This balance between recall and precision is summarized by the F1-score of 97.4%, indicating a high level of consistency between the two and confirming that the model performs reliably across both positive and negative classes.

3.4. Discussion

This study aimed to develop a robust machine learning model for predicting employee turnover using a well-structured approach comprising Exploratory Data Analysis (EDA), comprehensive data preprocessing, and rigorous model evaluation. The discussion section integrates findings from these three stages to demonstrate how each contributed to the model's high performance, providing valuable insights for future applications and organizational decision-making.

1) Exploratory Data Analysis (EDA): Uncovering Hidden Patterns

The EDA phase was instrumental in laying the foundation for subsequent preprocessing and modeling. Bivariate analysis using violin plots revealed meaningful differences in the distributions of several numerical features across target categories. This visualization method effectively highlighted how specific features like monthly income, years at company, and age could influence employee turnover tendencies. The patterns observed indicated a skew in certain variables among employees who left, suggesting these features hold predictive power. For instance, employees who had either just joined or had stayed for many years showed different turnover trends, possibly pointing to dissatisfaction at early or late career stages.

Multivariate analysis, through correlation heatmaps, added another layer of understanding by identifying inter-feature relationships. The Pearson correlation coefficient made it possible to detect multicollinearity—an issue that can dilute the

predictive power of models if not addressed. Highly correlated features such as "TotalWorkingYears" and "YearsAtCompany" required careful handling to avoid redundancy. Additionally, the correlation of certain features with the target variable offered insights into which predictors were most influential for turnover, guiding feature selection in the modeling phase.

One important takeaway from EDA was the imbalance in the target class distribution. Figure 4 clearly showed a disproportionate number of retained versus departed employees, a common occurrence in employee attrition datasets. Recognizing this early allowed for strategic planning in the data preprocessing stage to ensure the model would not be biased towards the majority class.

2) Preprocessing: Transforming Data for Effective Modeling

The preprocessing phase tackled multiple challenges using efficient and tailored solutions. The adoption of hash encoding for high-cardinality categorical features proved to be a pragmatic choice. Unlike traditional one-hot encoding, hash encoding significantly reduced dimensionality and preserved memory and computational efficiency without compromising the interpretability of the model. Despite the possibility of hash collisions, selecting a sufficiently large hash space mitigated the risks, ensuring reliable feature representation across model training and inference stages.

The label encoding of ordinal features such as "Salary" was another strategic move. It preserved the intrinsic order among categories—crucial for models that consider the ordinal nature of features during split decisions or weight calculations. This encoding ensured that higher salaries were recognized as superior to medium or low ones, thereby maintaining semantic integrity. Moreover, visualizations of the encoded features using kernel density plots confirmed that the transformations preserved data distribution. This step, often overlooked, validated the reliability of the encoded inputs and confirmed that preprocessing had not introduced distortions, which can significantly affect model outcomes.

Addressing class imbalance was perhaps the most impactful preprocessing step. Using SMOTE (Synthetic Minority Oversampling Technique), the study synthetically increased the representation of the minority class to ensure balanced learning. This approach eliminated the skew in class representation, allowing the model to learn equally from both classes and significantly improving its generalization capability. The visualization of class distribution before and after applying SMOTE (Figure 7) clearly demonstrated the success of this approach, with the minority class growing from 2,852 to 9,143 samples. Importantly, this synthetic generation was applied post-split and only to the training data, ensuring that model evaluation remained unbiased.

p-ISSN: **2656-5935** http://journal-isi.org/index.php/isi e-ISSN: **2656-4882**

3) Model Performance: Measuring Predictive Success

The final model, powered by the preprocessing strategies and trained on a balanced dataset, exhibited exceptional performance across all key metrics. Achieving an accuracy of 98.8% reflects the model's ability to make correct predictions in almost all cases. However, the emphasis on recall (96.1%) is particularly important in the context of employee turnover. Recall measures the model's ability to correctly identify employees who are likely to leave—a critical aspect for human resource departments aiming to implement retention strategies. Missing a potential leaver (false negative) could mean a lost opportunity to intervene and retain valuable talent.

High precision (98.8%) complements the strong recall, indicating that the employees flagged as potential leavers by the model are highly likely to actually leave. This minimizes false positives, ensuring that retention efforts are focused on the right individuals without unnecessary resource expenditure. The F1-score of 97.4%, a harmonic mean of precision and recall, underscores the model's balance and consistency.

The effectiveness of the model can be attributed to the synergy between thoughtful preprocessing and model selection. By transforming categorical variables appropriately, managing dimensionality, and addressing class imbalance, the study ensured that the model could learn effectively from the data. Additionally, the use of robust evaluation metrics ensured that the model's performance was thoroughly validated, not just in terms of accuracy, but also in terms of meaningful predictions that support actionable insights.

4. CONCLUSION

This study successfully demonstrates that the integration of hash encoding techniques with machine learning algorithms, particularly Random Forest, can be effectively used to predict employee turnover. Hash encoding efficiently addresses the limitations of handling categorical features in machine learning algorithms without significantly increasing data dimensionality, as often occurs with one-hot encoding. The findings indicate that the use of hash encoding not only accelerates the data preprocessing stage and conserves memory, but also enhances the predictive model's performance in the context of human resource management, especially in anticipating potential turnover. Therefore, this approach is worth considering as an alternative solution for developing a reliable and efficient employee prediction system.

REFERENCES

[1] K. S. Andrews and T. Mohammed, "Strategies for Reducing Employee Turnover in Small- and Medium-Sized Enterprises," *Westcliff International Journal of Applied Research*, vol. 4, no. 1, pp. 57–71, Nov. 2020, doi: 10.47670/wuwijar202041katm.

[2] A. F. Lestari, Y. M. Fauzi, A. I. Wazdi, and A. M. Sarusu, "Pengaruh Komitmen Organisasi dan Stres Kerja terhadap Turnover Intention Karyawan di PT BPRS HIK Parahyangan Bandung," *Jurnal Dimamu*, vol. 1, no. 1, pp. 23–36, 2021, doi: 10.32627.

[3] A. Wijaya, Tannia, Handoko, J. Matthew Karsten, and S. J. Salim, "The Effect Of Authentic Leadership On Turnover Intention In Service Sector With Work Engagement As Mediator," *Jurnal Muara Ilmu Ekonomi dan Bisnis*, vol. 8, no. 1, pp. 75–86, Apr. 2024, doi: 10.24912/jmieb.v8i1.28150.

[4] D. Ningsih, Maftukhin, I. D. Mulyani, A. Niasari, A. Sholeha, "Pengaruh Turnover dan Inventory Turnover terhadap Perubahan Laba pada Perusahaan Pertambangan Turnover and Inventory Turnover on Profit Changes in Mining Companies", *Journal of Accounting and Finance,* vol.1, no.1, 2019.

[5] P. Kumar, S. B. Gaikwad, S. T. Ramya, T. Tiwari, M. Tiwari, and B. Kumar, "Predicting Employee Turnover: A Systematic Machine Learning Approach for Resource Conservation and Workforce Stability †," *Engineering Proceedings*, vol. 59, no. 1, 2023, doi: 10.3390/engproc2023059117.

[6] M. Atef, D. S. Elzanfaly, and S. Ouf, "Early Prediction of Employee Turnover Using Machine Learning Algorithms 135 Original Scientific Paper", *International Journal of Electrical and Computer Engineering Systems,* vol.13, no.2, 2022.

[7] Y. Zhang, Z. Cai, and H. Fei, "Predicting Employee Turnover in High-Tech Enterprises Using Machine Learning: Based on the Psychological Contract Perspective", *Atlantis Press*, pp. 341–352, 2024, doi: 10.2991/978-94-6463-488-4_38.

[8] M. Al Akasheh, O. Hujran, E. Faisal Malik, and N. Zaki, "Enhancing the Prediction of Employee Turnover with Knowledge Graphs and Explainable AI," *IEEE Access*, vol. 12, pp. 77041–77053, 2024, doi: 10.1109/ACCESS.2024.3404829.

[9] J. Park, Y. Feng, and S. P. Jeong, "Developing an advanced prediction model for new employee turnover intention utilizing machine learning techniques," *Sci Rep*, vol. 14, no. 1, Dec. 2024, doi: 10.1038/s41598-023-50593-4.

[10] G. Obaido *et al.*, "Supervised machine learning in drug discovery and development: Algorithms, applications, challenges, and prospects," *Machine Learning with Applications*, vol. 17, p. 100576, Sep. 2024, doi: 10.1016/j.mlwa.2024.100576.

[11] X. Huang, H. Chen, and Z. Zhang, "Design and Application of Deep Hash Embedding Algorithm with Fusion Entity Attribute Information," *Entropy*, vol. 25, no. 2, Feb. 2023, doi: 10.3390/e25020361.

[12] P. Cerda and G. Varoquaux, "Encoding High-Cardinality String Categorical Variables," *IEEE Trans Knowl Data Eng*, vol. 34, no. 3, pp. 1164–1176, Mar. 2022, doi: 10.1109/TKDE.2020.2992529.

[13] K. R. Putra and M. A. Rachman, "Perbandingan Metode Content-based, Collaborative dan Hybrid Filtering pada Sistem Rekomendasi Lagu," *MIND Journal*, vol. 9, no. 2, pp. 179–193, Dec. 2024, doi: 10.26760/mindjournal.v9i2.179-193.

[14] L. N. Aina, V. R. S. Nastiti, C. S. K. Aditya, "Implementasi Extra Trees Classifier dengan Optimasi Grid Search CV pada Prediksi Tingkat Adaptasi", *MIND (Multimedia Artificial Intelligent Networking Database)*", 2024, doi: 10.26760/mindjournal.v9i1.78-88.

[15] D. Breskuvien and G. Dzemyda, "Categorical Feature Encoding Techniques for Improved Classifier Performance when Dealing with Imbalanced Data of Fraudulent Transactions," *International Journal of Computers, Communications and Control*, vol. 18, no. 3, 2023, doi: 10.15837/ijccc.2023.3.5433.

[16] M. Andrecut, "Additive Feature Hashing," 2021, doi: 10.48550/arXiv.2102.03943.

[17] A. Zheng and A. Casari, "Feature engineering for machine learning : principles and techniques for data scientists". *O'Reilly Media*, 2018.

[18] C. García-Vicente *et al.*, "Evaluation of Synthetic Categorical Data Generation Techniques for Predicting Cardiovascular Diseases and Post-Hoc Interpretability of the Risk Factors," *Applied Sciences (Switzerland)*, vol. 13, no. 7, Apr. 2023, doi: 10.3390/app13074119.

[19] I. Moura, A. Teles, D. Viana, J. Marques, L. Coutinho, and F. Silva, "Digital Phenotyping of Mental Health using multimodal sensing of multiple situations of interest: A Systematic Literature Review," Feb. 01, 2023, *Academic Press Inc.* doi: 10.1016/j.jbi.2022.104278.

[20] A. R. Kamila, J. F. Andry, A. W. C. Kusuma, E. W. Prasetyo, and G. H. Derhass, "Analysis Comparison of K-Nearest Neighbor, Multi-Layer Perceptron, and Decision Tree Algorithms in Diamond Price Prediction," *COGITO Smart Journal*, vol. 10, no. 2, 2024.

[21] J. Park, Y. Feng, and S. P. Jeong, "Developing an advanced prediction model for new employee turnover intention utilizing machine learning techniques," *Sci Rep*, vol. 14, no. 1, 2024, doi: 10.1038/s41598-023-50593-4.

[22] M. Cabanillas-Carbonell and J. Zapata-Paulini, "Evaluation of machine learning models for the prediction of Alzheimer's: In search of the best performance," *Brain Behav Immun Health*, vol. 44, Mar. 2025, doi: 10.1016/j.bbih.2025.100957.

[23] A. A. Khan, O. Chaudhari, and R. Chandra, "A review of ensemble learning and data augmentation models for class imbalanced problems: Combination, implementation and evaluation," Jun. 15, 2024, *Elsevier Ltd.* doi: 10.1016/j.eswa.2023.122778.

[24] A. R. Kamila, F. Adikara, C. Herdian, and Sutrisno, "Pengaruh Penambahan Fitur dengan Perbandingan Algoritma berbasis Bagging dan Boosting pada Deteksi Phishing Link", *JEPIN (Jurnal Edukasi dan Penelitian Informatika),* vol.10, no.3, 2024.

[25] J. Brabec and L. Machlica, "Decision-Forest Voting Scheme for Classification of Rare Classes in Network Intrusion Detection", *IEEE International Conference on Systems, Man, and Cybernetics,* pp. 3325–3330, 2018.