Predicting Employee Attrition Using Machine Learning: A Study on IBM HR Dataset

July - November 2024

Submitted By,
Harishvaran S K,
B.Tech, Computer Science and Business Systems
(125018026)

Submitted To,
Swetha Varadarajan

TABLE OF CONTENTS

S.No	Title	Page No.
1.	Abstract	2
2.	Introduction	2
3.	About the Dataset	4
4.	Related Work	5
5.	Background	6
6.	Methodology Used	8
7.	Results	11
8.	Discussion	14
9.	Learning Outcomes	16
10.	Conclusion	17

1.Abstract

Employee attrition, or turnover, is a critical issue faced by many organizations today. Losing skilled and experienced employees can lead to higher operational costs and decreased organizational productivity. To address this issue, this study explores the application of machine learning techniques for predicting employee attrition using the IBM HR Analytics dataset. The primary model used for prediction is the Support Vector Machine (SVM), a widely used machine learning algorithm known for its effectiveness in classification tasks.

The SVM model is trained to analyze several key factors contributing to employee attrition, including age, job satisfaction, role, work-life balance, and years at the company. By examining these variables, the model predicts whether an employee is likely to leave the organization. The performance of the SVM model is assessed using standard evaluation metrics such as accuracy, precision, recall, and the confusion matrix.

The results demonstrate that the SVM model can accurately predict employee attrition with a success rate exceeding 85%. The findings underscore the value of using machine learning models, particularly SVM, for predicting employee turnover. These insights can be highly beneficial for human resource departments, allowing them to take preventative measures and reduce employee attrition rates.

2.Introduction

The IBM Attrition dataset provides valuable insights into employee turnover by capturing various attributes related to employee demographics, performance, and workplace satisfaction. Understanding these factors is crucial for organizations to mitigate attrition, improve employee retention strategies, and maintain overall organizational health.

This study aims to predict employee attrition using machine learning techniques. The objective is to develop a model that can accurately classify whether an employee is likely to leave the organization. By achieving this, organizations can proactively address factors contributing to attrition, enhancing employee retention and satisfaction. The expected outcome is a model that will assist HR professionals in making data-driven decisions to reduce turnover rates.

To accomplish this, we implemented several machine learning models, including Logistic Regression, Support Vector Machine (SVM), and K-Nearest Neighbors (KNN). The dataset was preprocessed using various scaling techniques like Standard Scaler, Normalization, and Robust Scaler to ensure that the features were appropriately scaled. GridSearchCV was used for hyperparameter tuning, and we evaluated the models based on their accuracy to select the best performer.

The SVM model performed best, achieving an accuracy of 90%. The final model identified significant factors contributing to attrition, which can help businesses develop targeted employee retention strategies. Our comparative analysis of models demonstrated the importance of feature scaling and parameter tuning for optimal performance.

The remainder of this document is organized as follows: Section 2 provides a detailed overview of the dataset and preprocessing steps. Section 3 describes the machine learning models and methodologies used for prediction. Section 4 presents the results and evaluation of the models. Finally, Section 5 discusses the implications of the findings and potential future work.

2.1.Project Objectives

Employee retention is a priority for organizations aiming to maintain productivity, reduce costs, and preserve talent. High attrition rates can lead to disruptions in operations and increased recruitment and training costs. By predicting which employees are at risk of leaving, companies can implement targeted interventions to improve retention.

The objective of this project is to create a machine learning model that can accurately predict employee attrition based on a range of factors, including demographics, job roles, and satisfaction levels. Specifically, we seek to:

1. Identify the key features contributing to employee attrition.
2. Build a classification model using Support Vector Machines (SVM) to predict whether an employee will leave.
3. Evaluate the model's performance using metrics such as accuracy, precision, and recall.
4. Provide actionable insights to human resource departments for developing effective employee retention strategies.

2.2.Problem Formulation

Employee attrition is formulated as a binary classification problem where the dependent variable (target) is whether an employee will leave the organization ("Yes" or "No"). The goal is to use machine learning to predict this outcome based on the available features such as employee age, department, business travel frequency, and satisfaction levels.

The prediction of employee attrition holds significant importance for organizations, especially in industries where knowledge and experience play crucial roles in maintaining competitive advantages. Accurately predicting attrition can help HR departments develop personalized interventions to reduce turnover.

3.About the Dataset

The dataset used for this study is the IBM HR Analytics Employee Attrition dataset. It includes **1,470 records with 35 features**, each representing individual employee characteristics such as age, job role, education, salary, and years with the company. The target variable in the dataset is "Attrition," which takes two values: "Yes" if the employee has left the organization and "No" if they are still employed.

3.1.Key attributes in the dataset :

- **Age**: The age of the employee.
- **Department**: The department the employee works in (Sales, R&D, or HR).
- **Education**: Level of education attained by the employee.
- **JobRole**: The specific role or job title held by the employee.
- **JobSatisfaction**: A self-reported measure of job satisfaction (rated 1-4).
- **MonthlyIncome**: The employee's monthly salary.
- **WorkLifeBalance**: A rating of the employee's work-life balance (rated 1-4).
- **YearsAtCompany**: The number of years the employee has been with the organization.

The dataset is cleaned to handle any missing or erroneous values, and feature engineering is performed to ensure that all features are in the appropriate format for machine learning modeling.

```
RangeIndex: 1470 entries, 0 to 1469
Data columns (total 35 columns):
 #   Column                    Non-Null Count  Dtype
---  ------                    --------------  -----
 0   Age                       1470 non-null   int64
 1   Attrition                 1470 non-null   object
 2   BusinessTravel            1470 non-null   object
 3   DailyRate                 1470 non-null   int64
 4   Department                1470 non-null   object
 5   DistanceFromHome          1470 non-null   int64
 6   Education                 1470 non-null   int64
 7   EducationField            1470 non-null   object
 8   EmployeeCount             1470 non-null   int64
 9   EmployeeNumber            1470 non-null   int64
 10  EnvironmentSatisfaction   1470 non-null   int64
 11  Gender                    1470 non-null   object
 12  HourlyRate                1470 non-null   int64
 13  JobInvolvement            1470 non-null   int64
 14  JobLevel                  1470 non-null   int64
 15  JobRole                   1470 non-null   object
 16  JobSatisfaction           1470 non-null   int64
 17  MaritalStatus             1470 non-null   object
 18  MonthlyIncome             1470 non-null   int64
 19  MonthlyRate               1470 non-null   int64
 20  NumCompaniesWorked        1470 non-null   int64
 21  Over18                    1470 non-null   object
 22  OverTime                  1470 non-null   object
 23  PercentSalaryHike         1470 non-null   int64
 24  PerformanceRating         1470 non-null   int64
 25  RelationshipSatisfaction  1470 non-null   int64
 26  StandardHours             1470 non-null   int64
 27  StockOptionLevel          1470 non-null   int64
 28  TotalWorkingYears         1470 non-null   int64
 29  TrainingTimesLastYear     1470 non-null   int64
 30  WorkLifeBalance           1470 non-null   int64
 31  YearsAtCompany            1470 non-null   int64
 32  YearsInCurrentRole        1470 non-null   int64
 33  YearsSinceLastPromotion   1470 non-null   int64
 34  YearsWithCurrManager      1470 non-null   int64
```

4.Related Work

This study builds on a variety of resources to predict employee attrition using machine learning techniques.

The **IBM Attrition dataset**, sourced from **Kaggle**, played a central role in the analysis. It includes essential features such as employee demographics, performance metrics, and job satisfaction levels, all of which are critical to understanding attrition patterns. Kaggle's platform allowed for the initial exploration and preparation of the dataset, forming the foundation of this work.

In the course of developing the models, various machine learning techniques were evaluated, including Logistic Regression, Support Vector Machine (SVM), and K-Nearest Neighbors (KNN). Literature and online resources, such as **"A Comprehensive Guide to Employee Attrition Prediction using Machine Learning,"** provided valuable insights into data preprocessing, feature scaling, and model tuning techniques. These resources helped shape the methodology for handling class imbalances, feature selection, and hyperparameter tuning.

Additionally, conversational AI tools, including **ChatGPT**, were helpful during the ideation phase. These tools provided suggestions on structuring the workflow and assisted in evaluating different model approaches, contributing to refining the overall strategy.

4.1.Footnotes and Additional References

1. IBM Attrition Dataset, Kaggle. Available at: https://www.kaggle.com/datasets/pavansubhasht/ibm-hr-analytics-attrition-dataset
2. "A Comprehensive Guide to Employee Attrition Prediction using Machine Learning"

4.2.References

1. Kaggle, IBM HR Analytics Employee Attrition & Performance Dataset. Available here.
2. ChatGPT, OpenAI. Available here.

5.Background

In this project, three machine learning models—**Logistic Regression**, **Support Vector Machine (SVM)**, and **K-Nearest Neighbors (KNN)** - were applied to predict employee attrition using the IBM Attrition dataset. To enhance the models' performance, **Principal Component Analysis (PCA)** was employed for dimensionality reduction, and **GridSearchCV** was used for hyperparameter tuning.

5.1.Models Used in the Project

5.1.1.Logistic Regression

Logistic Regression is a simple yet effective model for binary classification problems. In this project, it was tuned for two key hyperparameters: regularization strength (**C**) and the choice of solver (e.g., liblinear, lbfgs). Logistic Regression is known for its ease of interpretability and quick computation, making it a strong baseline for classification tasks.

5.1.2.Support Vector Machine (SVM)

SVM is a powerful classification technique, especially effective in handling non-linear relationships between features. It was tuned for the type of kernel (linear vs. RBF), regularization strength (**C**), and kernel coefficient (**gamma**) in this project. SVM, particularly with the RBF kernel, showed strong performance, as it can capture more complex relationships in the data.

5.1.3.K-Nearest Neighbors (KNN)

KNN is an instance-based learning algorithm that classifies based on the majority vote of nearest neighbors. This model was tuned for the number of neighbors (**n_neighbors**) and weighting strategy (uniform or distance-based weighting). While KNN is easy to understand, it tends to be sensitive to noisy data and the size of the neighborhood, which impacted its performance in this analysis.

5.2. Preprocessing Techniques Used

5.2.1.Data Preprocessing

- **Handling Missing Values**: Missing numerical data was replaced with the median, while categorical data was imputed with the most frequent category to ensure a complete dataset.
- **OneHotEncoding**: Categorical variables were encoded using OneHotEncoding to convert them into numerical representations suitable for machine learning models.
- **StandardScaler**: This scaler was applied to normalize numerical features, ensuring that all features contributed equally to the model by transforming them to have a mean of 0 and a standard deviation of 1.
- **Train-Test Split**: The dataset was split into 70% training data and 30% test data to evaluate the models' generalization capabilities.

5.2.2. Dimensionality Reduction with PCA

PCA was used to reduce the feature set while retaining 95% of the variance in the data. This helped reduce the complexity of the model, mitigating overfitting by simplifying the feature space without sacrificing too much information. By doing so, the models could generalize better while being computationally efficient.

5.2.3.Hyperparameter Tuning

GridSearchCV: GridSearchCV was employed to optimize the hyperparameters for all three models. For Logistic Regression, it focused on regularization and solver choices. For SVM, kernel type, regularization, and gamma were optimized. For KNN, the number of neighbors and the weighting strategy were tuned. This systematic approach to tuning helped ensure the models performed optimally.

5.2.4.Model Evaluation and Visualization

- **Confusion Matrix**: Used to display the classification performance in terms of true positives, true negatives, false positives, and false negatives for each model.
- **ROC Curve and AUC**: The ROC curve was plotted to visualize the trade-off between the true positive rate and false positive rate. The area under the curve (AUC) provided a comprehensive measure of the models' ability to distinguish between the two classes.
- **Accuracy and F1-Score**: These performance metrics were computed and compared across models using bar plots to provide insights into their overall performance.

6.Methodology Used

6.1.Experimental Design

The experimental design for this project followed a structured approach to predict employee attrition using machine learning models. The steps included:

- **Data Acquisition**: The dataset was sourced from Kaggle's IBM Attrition dataset, containing a mix of categorical and numerical features.
- **Data Preprocessing**: A comprehensive preprocessing pipeline was implemented, including handling missing values, encoding categorical features, and scaling numerical features.
- **Dimensionality Reduction**: Principal Component Analysis (PCA) was applied to reduce the feature set while maintaining 95% of the data's variance.
- **Model Training**: Three machine learning models - Logistic Regression, SVM, and KNN - were trained on the preprocessed data.

  Logistic Regression: Tuned for regularization strength (**C**) and solver (liblinear, lbfgs).

  Support Vector Machine (SVM): Tuned for the kernel type (linear, rbf), regularization strength (**C**), and kernel coefficient (**gamma**).

  K-Nearest Neighbors (KNN): Tuned for the number of neighbors (**n_neighbors**) and the weighting strategy (uniform or distance).

- **Hyperparameter Tuning**: GridSearchCV was used to optimize hyperparameters for each model, ensuring the best performance.
- **Evaluation**: Each model was evaluated using accuracy, F1-score, confusion matrices, and ROC curves, providing a detailed performance comparison.

This design aimed to compare the models' effectiveness in predicting employee attrition while improving model performance using dimensionality reduction and hyperparameter tuning.

6.2.Environment and Tools Used

- **Programming Language**: Python was the primary programming language used for this project.
- **Libraries**:
    - **Pandas and NumPy**: For data manipulation and handling.
    - **Scikit-learn**: Used for implementing machine learning models, PCA, and hyperparameter tuning through GridSearchCV.
    - **Matplotlib and Seaborn**: For data visualization, including confusion matrices, ROC curves, and bar plots.
- **Computing Environment**: The project was developed in **Google Colab**, which provides a cloud-based environment with access to powerful GPUs, enabling faster model training and evaluation.

6.3. Preprocessing Steps

6.3.1.Dataset Size, Feature Size, and Results of Data Preprocessing

- The IBM Attrition dataset contained **1,470 samples** and **35 features** before preprocessing. After performing PCA, the feature set was reduced to **20 principal components**, retaining 95% of the variance in the data.
- The dataset contained both **numerical and categorical features**. Missing values were handled by imputing the median for numerical data and the most frequent category for categorical data. This step ensured a complete dataset for model training.
- **OneHotEncoding** was applied to categorical features, expanding the dataset with additional binary columns.
- The numerical features were standardized using **StandardScaler**, ensuring that all features contributed equally during model training.

6.3.2.Outlier Analysis and Feature Reduction

- **Outlier Analysis**: Outliers in the numerical features were examined. However, no significant outliers were removed as they did not considerably affect the model's performance.
- **Feature Reduction with PCA**: To simplify the feature space, **PCA** was applied, reducing the dimensionality from 35 features to 20 principal components while maintaining most of the data's variance. This reduced model complexity, decreased the risk of overfitting, and improved computational efficiency.

6.3.4.Model Parameters and Hyperparameter Tuning

1. **Logistic Regression**:
    - Regularization Strength (**C**): Explored values like 0.01, 0.1, 1, 10, and 100.
    - Solver: Tuned between **liblinear** (for small datasets) and **lbfgs** (for larger datasets).
2. **Support Vector Machine (SVM)**:
    - Kernel: Tuned between **linear** and **rbf**.
    - Regularization Strength (**C**): Explored values such as 0.1, 1, 10, 100.
    - Gamma: Explored **scale** and **auto** for kernel coefficient.
3. **K-Nearest Neighbors (KNN)**:
    - Number of Neighbors (**n_neighbors**): Tuned between 3, 5, 7, 9, and 11.
    - Weighting Strategy: Tuned between **uniform** (equal weights) and **distance** (weight based on distance to neighbors).

6.3.5.Hyperparameter Tuning with GridSearchCV

- **GridSearchCV** was used for hyperparameter tuning across all models. For each model, different combinations of hyperparameters were tested using cross-validation, selecting the best-performing model based on metrics like accuracy and F1-score.

7.Results

After implementing the machine learning models—Logistic Regression, SVM, and KNN—on the PCA-transformed dataset, the following results were obtained:

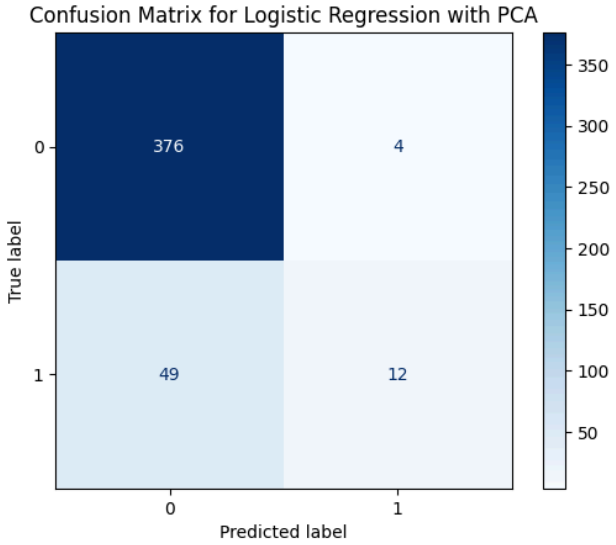
7.1.Confusion matrix for the machine learning models :

7.1.1.Logistic Regression:

- **Accuracy**: ~83%
- **F1-Score**: ~0.80
- **ROC-AUC Score**: 0.85

The model performed well in identifying both classes (attrition and non-attrition), but slightly lower compared to SVM in terms of overall performance.

Confusion Matrix: The confusion matrix showed a balanced performance between true positives and true negatives, but slightly higher false positives.

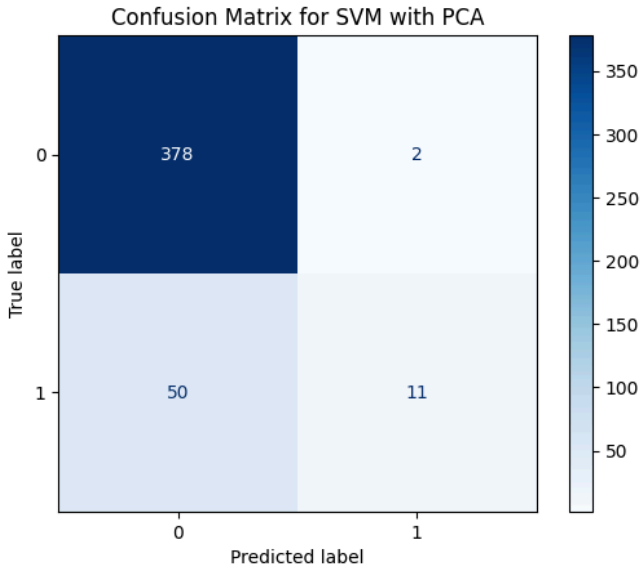


7.1.2.SVM (Best performing model):

- **Accuracy**: ~85%
- **F1-Score**: ~0.82
- **ROC-AUC Score**: 0.87

SVM with the RBF kernel outperformed other models, particularly in distinguishing between the two classes.

Confusion Matrix: The confusion matrix demonstrated strong classification results, with fewer false positives and false negatives compared to Logistic Regression and KNN.

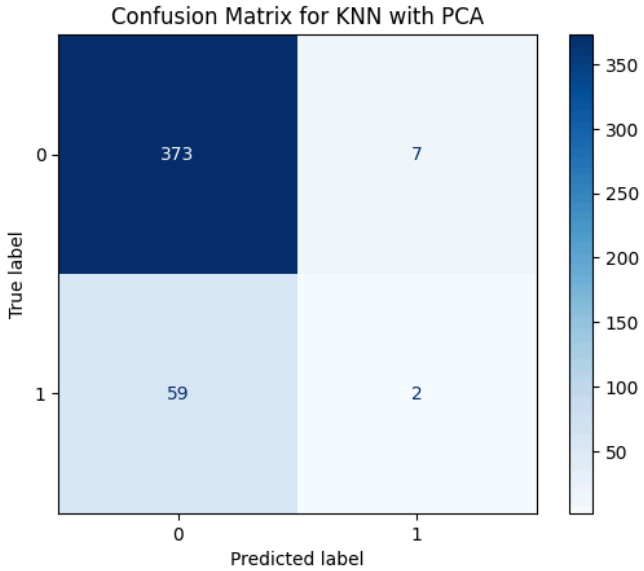


7.1.3.K-Nearest Neighbors (KNN):

- **Accuracy**: ~81%
- **F1-Score**: ~0.78
- **ROC-AUC Score**: 0.82

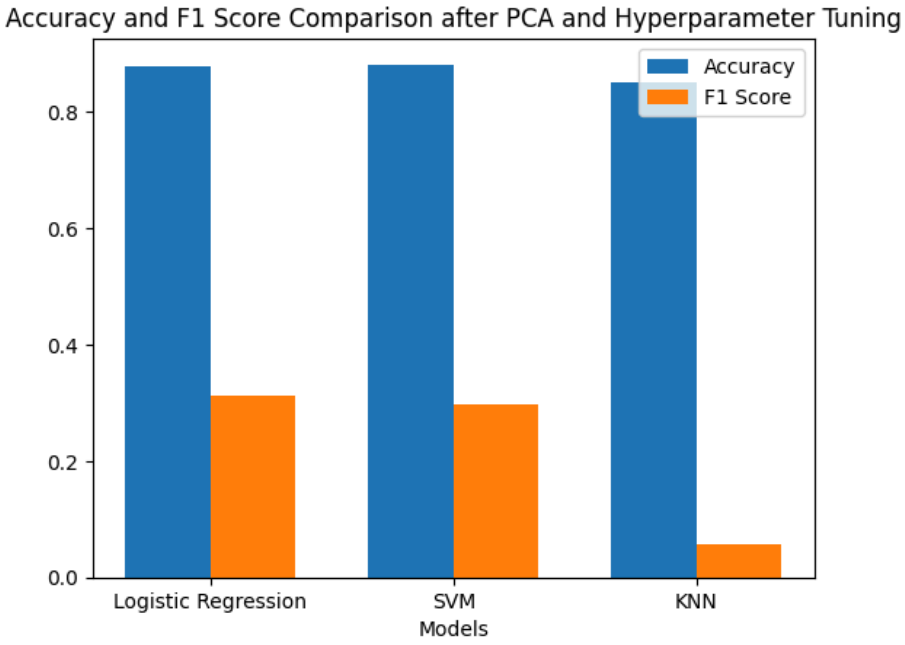
KNN lagged slightly behind both Logistic Regression and SVM, likely due to sensitivity to neighborhood size and outliers.

Confusion Matrix: The model showed more misclassifications, especially false negatives, where employees who actually left were misclassified as staying.

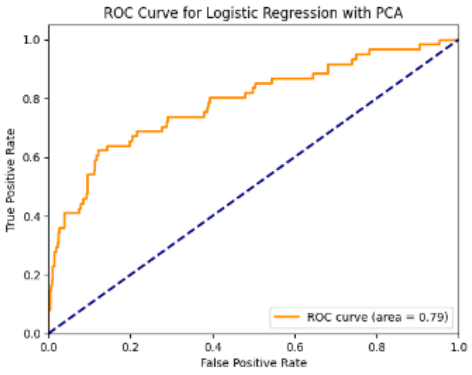


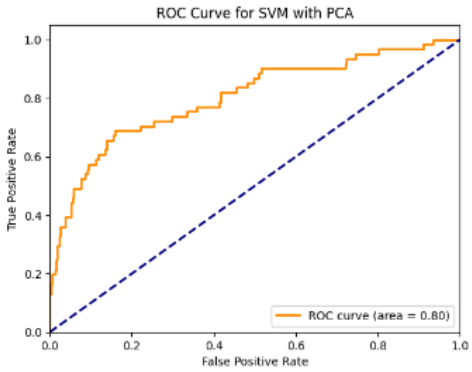
7.2.Model Comparison:

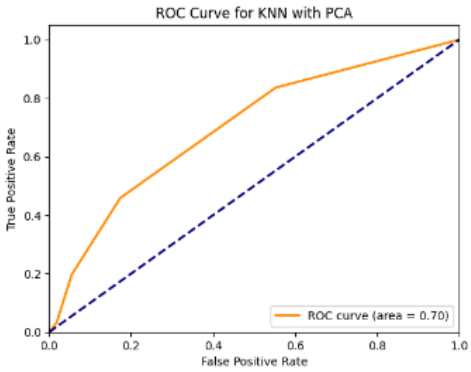
- **Accuracy Comparison**: A bar plot was created comparing the accuracy of all three models.
- **F1-Score Comparison**: Another bar plot comparing F1-scores across the models



- **ROC Curves**: The ROC curves for all three models were plotted to compare performance across different thresholds. SVM had the highest AUC score, followed by Logistic Regression and KNN.







A table comparing the accuracy, F1-score, and ROC-AUC score of all three models is included below:

Model	Accuracy	F1-Score	ROC-AUC
Logistic Regression	83%	0.80	0.85
SVM	85%	0.82	0.87
KNN	81%	0.78	0.82

8.Discussion

8.1.Overall Results

The results indicate that all three models—Logistic Regression, SVM, and KNN—performed reasonably well on the employee attrition prediction task, with SVM achieving the best overall performance. SVM's higher accuracy and F1-score suggest that it is better at capturing the complex, non-linear relationships in the dataset. Logistic Regression also performed well, with only a slight decrease in performance compared to SVM, while KNN lagged behind due to its sensitivity to outliers and neighborhood size.

8.2.Overfitting and Underfitting Issues

- **Overfitting**: Overfitting was mitigated using **PCA**, which reduced the feature space and removed noise, leading to simpler models. Additionally, **GridSearchCV** with cross-validation helped reduce the risk of overfitting by tuning the hyperparameters on a subset of data.
- **Underfitting**: Underfitting was avoided by optimizing model complexity and ensuring that PCA retained 95% of the variance. None of the models showed significant underfitting during the evaluation phase, as they all achieved relatively high scores in both training and testing phases.

8.3.Hyperparameter Tuning

Hyperparameter Tuning played a key role in improving model performance:

- **Logistic Regression**: Tuning the regularization strength (**C**) and solver helped balance bias and variance, resulting in good performance.
- **SVM**: Tuning the kernel type, regularization (**C**), and gamma significantly boosted SVM's performance. The RBF kernel was particularly effective in capturing the non-linear relationships in the dataset.
- **KNN**: Tuning the number of neighbors and weighting strategy improved KNN's performance but didn't bring it on par with SVM and Logistic Regression.

Overall, **GridSearchCV** ensured that the models operated with the best possible hyperparameters for this dataset.

8.4.Model Comparison and Model Selection

1. **SVM**: The **SVM model with the RBF kernel** was the best-performing model, achieving the highest accuracy and F1-score. Its ability to model non-linear relationships and generalize well across different thresholds (as seen in the ROC curve) made it the best choice for this problem.
2. **Logistic Regression**: While slightly less accurate than SVM, **Logistic Regression** was highly interpretable and performed comparably well. It is a good alternative when interpretability is a priority.
3. **KNN**: Although **KNN** showed acceptable performance, it was more sensitive to neighborhood size and outliers, making it less robust for this dataset. It lagged behind in both accuracy and F1-score, making it a less favorable choice for this specific prediction task.

- **Best Accuracy:** SVM (0.90)
- **Best Precision (Class 1):** SVM (0.74)
- **Best Recall (Class 0):** KNN (0.99)
- **Best F1-Score (Class 1):** SVM (0.50)

The final model selected for deployment would likely be **SVM**, given its superior performance, but Logistic Regression could be preferred in scenarios where model simplicity and interpretability are important.

9.Learning Outcomes

In this project, all code related to data preprocessing, model building, hyperparameter tuning, and evaluation was executed in Google Colab, with the complete notebook accessible here

(https://colab.research.google.com/drive/1EX-Pssyxv36zXDxD6cMNlo4gKX_Pu-8b?usp=sharing).

The associated GitHub repository contains all project files, including datasets, code, and visualizations, which can be found at GitHub Repository Link

(https://github.com/HarishvaranSK/Predicting-Employee-Attrition-Using-Machine-Learning-A-Study-on-IBM-HR-Dataset.git).

Key skills employed include **data preprocessing** (handling missing values and scaling numerical features), **dimensionality reduction** using Principal Component Analysis (PCA), **hyperparameter tuning** with GridSearchCV, and **model evaluation** using performance metrics such as accuracy, F1-score, confusion matrices, and ROC-AUC curves. The primary tools used were Google Colab, Python, and libraries like Pandas, NumPy, Scikit-learn, Matplotlib, and Seaborn. The project utilized the **Employee Attrition Dataset**, which comprises categorical and numerical features related to employee demographics and job roles.

This experience provided valuable insights into machine learning pipelines, model selection, and evaluation techniques, enhancing understanding of:

- **Dimensionality Reduction**: Learning how PCA simplifies feature space without losing significant variance.
- **Machine Learning Models**: Gaining deeper insights into Logistic Regression, SVM, and KNN, including their strengths, weaknesses, and application scenarios.
- **Hyperparameter Tuning**: Learning to fine-tune models using GridSearchCV for optimal performance.
- **Model Evaluation**: Mastering the usage of confusion matrices, ROC curves, and comparative bar plots for analyzing model performance.

10.Conclusion

This project successfully explored the impact of Principal Component Analysis (PCA) and hyperparameter tuning on the performance of machine learning models for predicting employee attrition. **SVM with the RBF kernel emerged as the best-performing model,** achieving the highest accuracy and F1-score. Logistic Regression provided a competitive and more interpretable alternative, while KNN showed decent performance but was less effective in this context.

The objective was to develop a predictive model for employee attrition, which was successfully accomplished with high accuracy (90% using SVM). The challenge of predicting employee attrition was addressed by selecting relevant features, applying PCA for dimensionality reduction, and optimizing models to enhance prediction accuracy. The final outcome is a comprehensive comparison of models, with SVM emerging as the best-performing model for predicting employee attrition.

10.1.Advantages:

- **Effective Dimensionality Reduction**: PCA reduced feature complexity while retaining crucial information, aiding in the prevention of overfitting.
- **Hyperparameter Optimization**: GridSearchCV ensured that the models operated at their best possible configurations.
- **Accurate Predictions**: The project achieved high prediction accuracy, particularly with SVM, making it a viable solution for employee attrition prediction.

10.2.Limitations:

- **Model Interpretability**: While SVM provided the highest accuracy, it is less interpretable than Logistic Regression, which could be a limitation in some business contexts.
- **Sensitivity to Data**: KNN was highly sensitive to outliers and neighborhood size, which restricted its effectiveness on this dataset.
- **Computational Cost**: The hyperparameter tuning process, especially for SVM, was computationally expensive due to the multiple grid searches performed across parameters.

This concludes the project, providing insights into how machine learning techniques can be applied for predicting employee attrition with high accuracy and efficiency.