

Employee Resignation Prediction

*A project report submitted to ICT Academy of Kerala
in partial fulfillment of the requirements
for the certification of*

CERTIFIED SPECIALIST IN DATA SCIENCE & ANALYTICS

submitted by

Adita Anil

Niranjana Krishna B J

S Muhammed



**ICT ACADEMY OF KERALA
THIRUVANANTHAPURAM, KERALA, INDIA
Dec 2024**

List of Figures

Sl. No.	Name	Page no.
1	Dataset	10
2	Confusion matrix	16
3	Visualization	20
4	Architecture of Random Forest	22
5	Imbalanced Target Column	24
6	Balanced Target data using SMOTE	24
7	Comparison of all the algorithms used	25
8	Home Page	31
9	Resignation prediction	32

List of Abbreviations

EDA	Exploratory Data Analysis
KNN	K-Nearest Neighbor
RF	Random Forest
LR	Logistic Regression
DT	Decision Tree
XGB	XG Boost
SMOTE	Synthetic Minority Oversampling Technique
XGBoost	Extreme Gradient Boosting

Table of Contents

Sl. no.	Type	Page no.
1	Abstract	5
2	Problem Definition	6
3	Introduction	8
4	Literature Survey	10
5	Materials and Methods	12
6	Methodology	23
7	Future Scope	27
8	Limitations	29
9	Result	31
10	Conclusion	33
11	Reference	34

Abstract

Employee retention is a critical challenge for organizations striving to maintain efficiency and minimize costs. This study leverages a comprehensive dataset of 100,000 employees, encompassing key factors such as demographics, job performance metrics, and satisfaction scores, to develop a predictive model for employee resignation.

High resignation rates disrupt workflows, increase recruitment expenses, and lower overall productivity. This project leverages machine learning techniques to predict employee resignations by analysing key factors like satisfaction levels, job roles, salaries, and work environments. The insights derived from this predictive model aim to assist HR departments in identifying at-risk employees and implementing proactive retention strategies. A user-friendly interface, developed using Flask and Streamlit, ensures seamless interaction with the model, enabling data-driven decision-making to enhance employee satisfaction and reduce turnover rates.

1. Problem Definition

1.1 Overview

1.1.1 Project Purpose

The purpose of this project is to design a machine learning-based system that predicts the likelihood of employee resignations. By analyzing historical HR data, the system identifies factors contributing to resignations, enabling the HR department to implement effective interventions. This approach empowers organizations to reduce turnover rates, improve employee satisfaction, and optimize resource utilization.

1.1.2 Business Need

Employee resignations have a profound impact on organizational performance and finances. High turnover rates disrupt team dynamics, increase recruitment and training costs, and affect employee morale. HR departments need a robust data-driven tool to predict resignations and address underlying issues proactively.

1.1.3 Objectives

- Develop a predictive model to forecast employee resignations.
- Identify key factors contributing to resignations for actionable insights.
- Provide a scalable solution for HR departments to implement retention strategies.

1.1.4 Timeline

The timeline given for us is one month (30/12/2024 -17/12/2024)

- (0-1 weeks): Data collection, cleaning, and exploratory data analysis (EDA).
- (2nd week): Model building, feature engineering, and model selection.
- (3rd week): Hyperparameter tuning, model validation, testing, deployment, and report generation, presentation

1.1.5 Resources

- Google Colab, Jupyter Notebook, VS Code
- Python (Pandas, NumPy, scikit-learn, Flask, Streamlit).
- Data visualization tools like Matplotlib, Seaborn.

1.2 Problem Statement

High employee turnover leads to operational inefficiencies and increased costs. HR teams face challenges in identifying at-risk employees and addressing their concerns effectively. This project seeks to address the following questions:

1. What factors are most predictive of employee resignations?
2. How can data-driven insights help HR departments improve employee retention strategies?
3. Can a predictive model provide actionable outputs for timely interventions?

By addressing these challenges, the project aims to equip HR departments with a tool to proactively manage workforce stability.

2. Introduction

2.1 Overview

Employee retention is an ongoing challenge for organizations across industries. With the growing demand for skilled professionals, retaining talent is as crucial as attracting it. Dissatisfaction with factors like compensation, work-life balance, and career growth are common reasons for resignations. This project applies data science to analyze these factors and predict the likelihood of resignations, offering actionable insights for HR teams.

2.2 Needs of HR Department

1. **Proactive Retention Strategies:** Predicting resignations enables HR to address dissatisfaction before employees decide to leave.
2. **Optimized Resource Allocation:** Insights into resignation trends help prioritize resource allocation for training and retention programs.
3. **Informed Policy Development:** Data-driven insights support the formulation of employee-friendly policies, such as flexible work arrangements or improved career development plans.
4. **Cost Savings:** Reducing turnover rates decreases recruitment and training costs, ensuring long-term organizational stability.

2.3 Detailed Approach

2.3.1 Dataset Overview

The dataset includes features such as:

- Employee demographics (e.g., age, gender, education level).

- Job-related factors (e.g., job title, department, salary).
- Performance metrics (e.g., satisfaction score, projects handled).
- Behavioural indicators (e.g., overtime hours, sick days, promotions).

2.3.2 General EDA steps

1. **Data Cleaning:** Handling missing or inconsistent data entries.
2. **Visualization:** Creating graphs to understand feature distributions (e.g., satisfaction levels) and correlations (e.g., between salary and resignations).
3. **Insights:** Identifying trends, such as whether dissatisfaction with salary correlates with resignations.

3. Literature Survey

1. El-Rayes et al. (2020) explored the use of tree-based models, such as decision tree, random forest, and gradient boosted tree, for predicting employee attrition. The study used a dataset from Glassdoor, focusing on features like employee satisfaction, performance score, monthly salary, and number of promotions as key predictors. The random forest model performed best, offering insights that could assist human resources in understanding the drivers of attrition.

The advantages of this proposed model are the strong interpretability and clear identification of key factors, customizable for various organizational contexts.

Disadvantages are dependent on data quality and completeness, also it may not capture complex interactions as well as other techniques

2. Sufian and Varadarajan (2020) explored employee attrition prediction by combining machine learning, econometric, and statistical methods. Their hybrid approach uses historical data to identify trends and predict the likelihood of employee turnover. This model helps organizations understand key factors contributing to attrition and supports evidence-based human resource management strategies.

The advantages of this model is the integration of multiple methodologies, which enhances the robustness of the analysis and allows for the prediction of attrition trends with greater accuracy. However, the approach also has disadvantages, including its complexity in implementation due to the need to align different methodologies. Additionally, data collection challenges and the difficulty in harmonizing econometric models with machine learning outputs pose limitations to its practical application.

3. Kumar et al. on 2021 investigated the use of XGBoost for predicting employee resignation, integrating structured data with employee feedback for improved predictions. Their model demonstrated the importance of features such as training hours, promotions, and team size in influencing employee decisions.

The advantage high performance with imbalanced datasets and ability to handle missing data effectively. The disadvantage is dependence on well-tuned hyperparameters and computational complexity.

4. Smith and Lee on 2018 utilized Gradient Boosting Machines (GBMs) to predict employee attrition in a multinational company. Their model highlighted key factors such as lack of promotions, low satisfaction scores, and work-life balance issues as critical predictors of turnover. The study demonstrated the effectiveness of ensemble methods in handling complex HR datasets.

The advantages of their model are strong predictive performance and the ability to handle missing values. The disadvantages include susceptibility to overfitting and reliance on extensive hyperparameter tuning.

5.Chen and Lee on 2019 focused on applying Neural Networks for employee turnover prediction in large organizations. Their model considered a wide range of features such as employee age, department, satisfaction, and job promotion history. They found that Neural Networks performed exceptionally well in capturing complex patterns and interactions between variables.

The advantages of this approach include its ability to learn from large datasets and model intricate patterns. However, the disadvantages include its black-box nature, which makes it difficult to interpret the results, as well as the computational resources required for training the model on large datasets.

4. Materials and methods

4.1 Dataset

4.1.1 Employee Resignation Dataset

The dataset taken is from a private US based company. Below are the Columns in the dataset:

```
[31]: df.columns  
[31]: Index(['Employee_ID', 'Department', 'Gender', 'Age', 'Job_Title',  
          'Years_At_Company', 'Education_Level', 'Performance_Score',  
          'Monthly_Salary', 'Work_Hours_Per_Week', 'Projects_Handled',  
          'Overtime_Hours', 'Sick_Days', 'Remote_Work_Frequency', 'Team_Size',  
          'Training_Hours', 'Promotions', 'Employee_Satisfaction_Score',  
          'Resigned'],  
          dtype='object')
```

fig 1. Dataset

4.1.2 Barriers and Risks

- **Imbalanced Target Data:**

The dataset had an imbalanced target variable, with a significantly higher number of instances for employees who stayed compared to those who resigned. This imbalance posed challenges for model training, as the model tended to favor the majority class, reducing its ability to correctly predict resignations. To address this, techniques such as oversampling, undersampling, or using algorithms designed to handle imbalanced data (e.g., SMOTE or class-weighted models) were implemented.

- **Equal Distribution of Attributes:**

Many features in the dataset had equal distributions across their values, with minimal skewness or variance. This made it difficult for the model to distinguish between classes effectively, as the features provided limited discriminative power. Feature engineering and selection became critical in addressing this limitation to improve model performance.

- **Limited Feature Importance:**

Some attributes had weak or negligible correlations with the target variable. This required careful analysis to identify and retain only the most relevant features while excluding noise that could degrade model accuracy.

- **Risk of Overfitting:**

While addressing class imbalance and performing hyperparameter tuning, there was a risk of overfitting the model to the training data. Overfitting would have resulted in poor generalization to unseen data.

- **Data Quality Issues:**

Potential for missing values or erroneous data entries in features such as job satisfaction or performance scores, which could skew the analysis.

Inconsistent formatting of categorical variables (e.g., differences in job titles or department names) required extensive preprocessing to ensure uniformity.

- **Computational Complexity:**

Handling imbalanced data and performing techniques like oversampling or ensemble learning (e.g., Random Forest, XGBoost) increased computational requirements, making the process resource-intensive and time-consuming.

- **Interpretability Challenges:**

Advanced machine learning models like XGBoost and Random Forest, while accurate, can be challenging to interpret for non-technical stakeholders. Translating the results into actionable insights for HR teams required additional effort in generating explainable visualizations (e.g., SHAP plots).

- **Bias in the Dataset:**

The dataset may inherently reflect organizational biases (e.g., gender, role-specific trends), which could affect the model's fairness and reliability. Ensuring ethical model outcomes required careful evaluation of feature importance and predictions.

- **Dataset Context**

The dataset was sourced from a US-based company, with salaries represented in **USD**. Adapting this data to an Indian context presented challenges due to differences in salary ranges, economic conditions, and industry standards. For instance, salary figures in USD may not directly correlate with job satisfaction levels or resignation tendencies in an Indian company, where cost-of-living differences and cultural factors also play a significant role.

4.1.3 Materials and Methods

Tools and Technologies

- **Programming Language:** Python (NumPy, Pandas, scikit-learn, Streamlit).
- **Visualization Libraries:** Matplotlib, Seaborn.
- **Deployment:** Streamlit for web interface.
- **Modelling:** Machine learning algorithms such as Logistic Regression, Random Forest, Decision Tree and XGBoost.

Methodology

- **Data Preprocessing:**

Data preprocessing is the process of preparing raw data for further analysis by cleaning, transforming, and formatting the data. It is a critical step in data mining, machine learning, and other data-intensive applications, as it ensures the quality, consistency, and suitability of the data for the specific analytical task. The main objectives of data preprocessing include handling missing or incomplete data, removing outliers and noise, resolving inconsistencies, transforming data into a suitable format (e.g., scaling, encoding categorical variables), integrating data from multiple sources, and reducing the dimensionality of the data when necessary. Techniques such as imputation, normalization, encoding, Principal Component Analysis (PCA), and feature selection are commonly

used in data preprocessing. Proper data preprocessing improves the accuracy and reliability of subsequent analyses, eliminates potential sources of bias or errors, and enhances the overall quality of the data, making it more suitable for the chosen analytical methods

- o Encoding categorical variables (e.g., job titles, education level, gender etc)
 - ✓ Gender – OneHotEncoding
 - ✓ Education level – Ordinal
 - ✓ Job Title – Ordinal
 - ✓ Department – Frequency Encoding
- o Scaling numerical features to ensure uniformity. (MinMaxScaler)
- **Model Development:**
 - o Training various machine learning models like Logistic Regression, KNN, Decision Tree, Random Forest and XGBoost.
 - o Tuning hyperparameters for DT, RF AND XGB to optimize model performance.
- **Evaluation Metrics:**
 - o Accuracy, precision, recall, and F1 score were used to evaluate the models.
 - o RF selected as the model.

4.1.4 Confusion Matrix

A confusion matrix is a table that provides a detailed evaluation of a classification model's performance by comparing its predicted outcomes with the actual or true labels from a test dataset. It is particularly valuable for assessing both binary and multiclass classification problems. In this matrix, the rows represent the actual classes, while the columns represent the predicted classes. The diagonal entries indicate the instances that were correctly classified, whereas the off-diagonal entries represent the misclassifications. The confusion matrix consists of four key components:

- **True Positives (TP):** Cases correctly identified as belonging to the positive class.
- **True Negatives (TN):** Cases correctly identified as not belonging to the positive class.
- **False Positives (FP):** Cases incorrectly identified as positive when they actually belong to the negative class (Type I error).
- **False Negatives (FN):** Cases incorrectly identified as negative when they actually belong to the positive class (Type II error).

		True Class	
		Positive	Negative
Predicted Class	Positive	TP	FP
	Negative	FN	TN

fig 2. Confusion matrix

Analyzing the values in the confusion matrix provides valuable insights into a model's performance and helps identify potential issues or biases. For instance, a high number of false positives could indicate that the model is overly sensitive, incorrectly classifying many instances as positive, while a high number of false negatives might suggest that the model is failing to capture true positive instances. The confusion matrix also serves as the basis for calculating key performance metrics, such as accuracy, precision, recall, and F1-score. These metrics offer different perspectives on the model's effectiveness and are essential for selecting the most suitable model for a specific task or application.

From the confusion matrix, you can calculate various metrics that provide deeper insights into model performance:

1.Accuracy

Accuracy is a fundamental metric in evaluating the performance of classification models,

providing insight into their effectiveness in predicting the correct class labels for given instances. In multi-class classification scenarios, where instances can belong to one of several possible classes, calculating accuracy involves assessing the model's ability to correctly classify instances across all classes. The process of determining accuracy begins with making predictions for each instance in the dataset. For every instance, the model assigns a predicted class label based on the features it observes and the learned patterns from the training data. The accuracy of these predictions is then assessed by comparing them to the true class labels of the instances. To calculate overall accuracy in multi-class classification, we sum up the number of correct predictions across all classes and divide it by the total number of instances in the dataset. This approach ensures that each correctly predicted instance contributes equally to the overall accuracy, regardless of its class.

$$\text{Accuracy} = (TP + TN) / (TP + TN + FP + FN)$$

2. Recall

Recall, also known as sensitivity or true positive rate, is a crucial metric in evaluating the performance of classification models, particularly in scenarios where correctly identifying positive instances is of utmost importance. It measures the ability of a model to correctly identify all relevant instances from a given class, out of all instances that truly belong to that class. In multi-class classification, calculating recall for each class involves treating that class as the "true" class and all other classes as "false" values. This means that for a specific class, the recall indicates the proportion of instances from that class that the model correctly identifies, compared to the total number of instances that actually belong to that class. To compute recall for a particular class in a multi-class setting, we focus on the instances where that class is the true positive (TP) and classify all other instances as either false negatives (FN) or true negatives (TN) depending on whether they belong to the same class or not.

$$\text{Recall} = TP / (TP + FN)$$

3.Precision

Precision is a metric used in machine learning and information retrieval to evaluate the performance of a classification or prediction model. It measures the fraction of correctly identified positive instances (true positives) among all instances that the model predicted as positive (true positives and false positives). A high precision value means that the model is making confident positive predictions, and most of the instances it identifies as positive are indeed positive. A low precision value indicates that the model is incorrectly classifying many negative instances as positive. Precision is often used in conjunction with recall (sensitivity) and F1-score to provide a comprehensive evaluation of a model's performance. It is particularly important in applications where false positives are costly or undesirable, such as spam detection, fraud detection, or disease diagnosis.

$$\text{Precision} = \text{TP} / (\text{TP} + \text{FP})$$

4.F1 Score

The F1 score is a measure of a model's performance that combines precision and recall into a single metric. It is the harmonic mean of precision and recall. The F1 score ranges from 0 to 1, with 1 being the best possible value. A higher F1 score indicates that the model has a good balance between precision and recall. The main motivation behind the F1 score is that optimizing for precision or recall alone is usually not enough. A model with high precision but low recall will miss a lot of positive instances, while one with high recall but low precision will classify too many negative instances as positive. The F1 score attempts to find the best compromise between precision and recall based on their harmonic mean. As the name suggests, the F1 score weights recall and precision equally. However, it is also possible to calculate a generalized F-beta score that weights one metric over the other. The F1 score is widely used in classification problems, especially when there is a class imbalance or when the costs of false positives and false negatives are different. It provides a single summary statistic of a model's performance across multiple classes.

$$\text{F1score} = (2 \times \text{Precision} \times \text{Recall}) / (\text{Precision} + \text{Recall})$$

How to Interpret It

- A **perfect model** would have only diagonal entries (TP and TN), with no off-diagonal elements (FP or FN).
- A model with a **high number of FP or FN** indicates it is either over-predicting or under-predicting certain classes.
- For imbalanced datasets, **precision, recall, and F1 score** are more informative than accuracy.

4.2 Visualization Tools

4.2.1 Matplotlib

Matplotlib is a versatile, low-level library used for creating static, interactive, and animated visualizations in Python. It serves as the foundation for many other visualization libraries.

- Supports a wide variety of plots: line charts, bar charts, scatter plots, histograms, etc.
- Highly customizable: You can control every aspect of a plot, from axis labels to colors and line styles.
- Exports high-quality visualizations in multiple formats (e.g., PNG, PDF, SVG).
- Includes both functional (pyplot) and object-oriented interfaces for greater flexibility.
- Simple visualizations for quick insights (e.g., `plt.plot()`).
- Customizing plots for research papers or presentations.
- Creating subplots and advanced layouts.

4.2.2 Seaborn

Seaborn is built on top of Matplotlib and provides a high-level interface for creating aesthetically pleasing and informative statistical graphics.

- Simplifies complex visualizations with minimal code.
- Built-in themes and styles for attractive plots (e.g., darkgrid, whitegrid).
- Specializes in statistical plots: distribution plots, heatmaps, pair plots, etc.
- Easily integrates with Pandas DataFrames for data visualization.

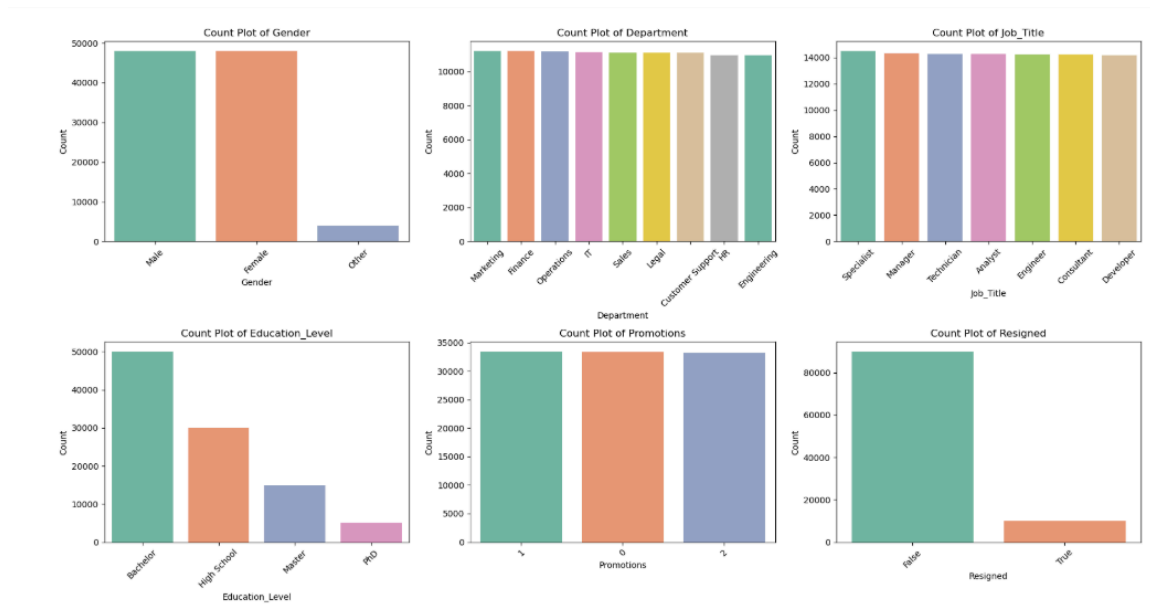


fig 3: Univariate analysis using Matplotlib

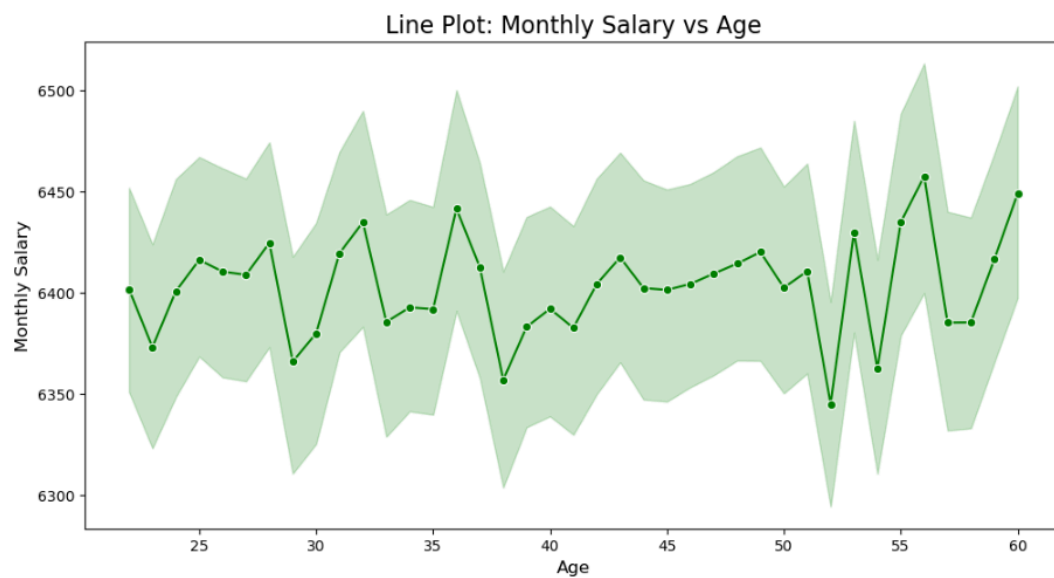


fig 4: Line plot using Seaborn

4.3 Algorithms

Initially, we evaluated multiple algorithms, including Logistic Regression, K-Nearest Neighbors (KNN), Naive Bayes, Decision Tree, Random Forest, and XGBoost, to identify the most suitable model for our problem. Among these, Decision Tree, Random Forest, and XGBoost demonstrated the highest accuracy. To further improve their performance, we applied hyperparameter tuning. While this process slightly reduced the accuracy of the Decision Tree model, it significantly enhanced the accuracy of both Random Forest and XGBoost. Ultimately, Random Forest emerged as the best-performing model, delivering the highest accuracy among all the algorithms evaluated. Consequently, we selected Random Forest for our final implementation.

4.3.1 Random Forest

The **Random Forest algorithm** is a powerful ensemble learning technique widely used for both classification and regression tasks. It operates by building multiple decision trees on random subsets of the training data through a method called bootstrapping (random sampling with replacement). Each decision tree is trained on a random sample of data, and at each split, only a random subset of features is considered, introducing additional randomness. The final prediction is made by aggregating the results of all trees—through majority voting for classification tasks or averaging for regression tasks—making the algorithm robust and highly accurate.

One of the key strengths of Random Forest is its ability to handle overfitting, which is a common issue in single decision trees. By averaging predictions from multiple trees, it ensures better generalization to unseen data. Additionally, Random Forest provides feature importance rankings, helping in identifying the most significant predictors in a dataset. It is also versatile, capable of handling high-dimensional datasets, missing data, and a variety of applications, including employee attrition prediction, fraud detection, medical diagnostics, and market analysis.

However, the algorithm comes with some trade-offs. It can be computationally intensive due to the large number of trees, making it slower for large datasets or real-time predictions. Additionally, while it offers superior performance, it is less interpretable compared to a single decision tree, as the collective decision-making process of hundreds of trees is complex. Despite

these limitations, Random Forest remains one of the most reliable and widely used algorithms due to its accuracy, scalability, and ability to handle diverse data structures effectively.

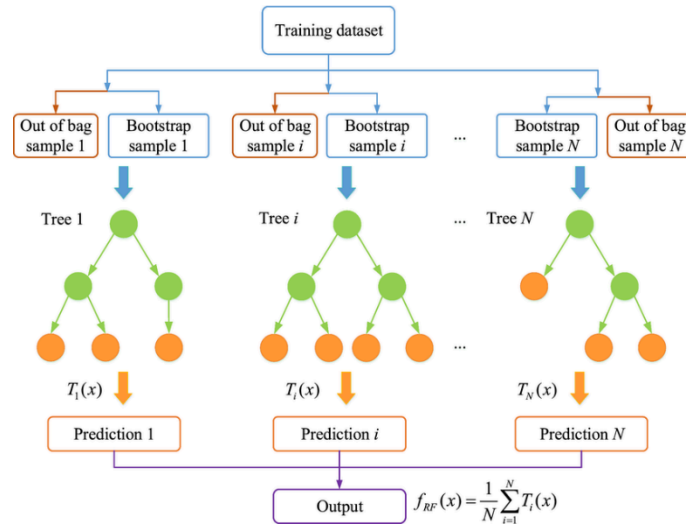


fig 5. Architecture of Random Fores

5. Methodology

Methodology serves as the foundation of any project, offering a structured framework for planning, execution, and evaluation. Without a well-defined methodology, projects are prone to inefficiencies, mismanagement, and failure. A robust methodology is essential for ensuring project success, as it helps manage risks, streamline processes, and maximize the chances of achieving desired outcomes. Organizations that adopt strong methodologies are better equipped to meet stakeholder expectations and deliver successful results. In prediction-based projects, key steps typically include defining an appropriate classification system, selecting training samples, extracting relevant features, choosing suitable classification techniques, conducting post-classification processing, and evaluating the model's accuracy. These steps form a cohesive workflow that supports reliable and effective project execution.

Data Collection and Preprocessing

- Dataset was collected from Kaggle, the dataset was carried information of a US based company.
- The dataset underwent preprocessing steps like handling missing values, encoding categorical variables using OneHotEncoding, Frequency Encoding, and Ordinal Encoding, and normalizing numerical features.
- **Exploratory Data Analysis (EDA)**
 - Descriptive statistics and visualizations were used to understand the distribution of features and identify correlations.
- **Handling Imbalanced dataset**
 - Since the target of our dataset carried imbalanced data proportion [ie, 90%of the dataset showed “not resigned” while only 10%of the data showed “resigned”]
 - So, there was a need to handle our dataset using SMOTE (Synthetic Minority Oversampling TEchnique), where the minority [resigned] was oversampled synthetically.

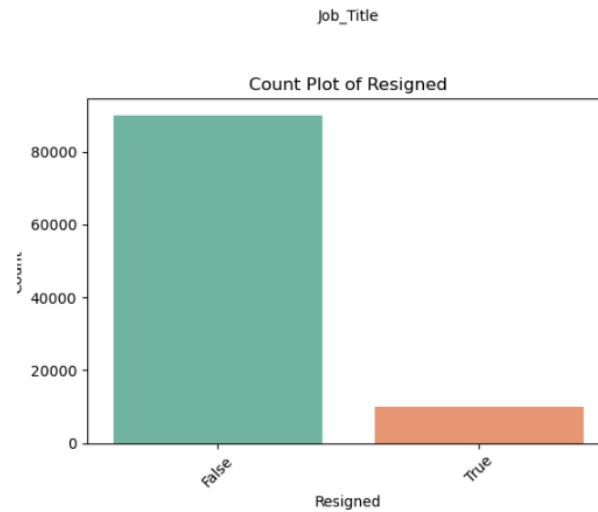


Fig 6: Imbalanced Target Column



fig 7: Balanced Target data using SMOTE

- **Employee Resignation Prediction**

- The prediction models focused on identifying whether an employee resign or not using various parameters including employee satisfaction score, performance score, promotions, training hours, sick days etc.

- Classification algorithms were compared (LR, KNN, NB, DT, RF, XGB)
- **Hyperparameter Tuning**
 - Hyperparameter tuning is the process of optimizing the parameters that control the behavior of a machine learning algorithm, such as the number of trees in a Random Forest or the learning rate in XGBoost. It involves systematically searching for the best combination of these parameters to enhance the model's performance on unseen data.
 - We checked hyperparameter tuning of DT, RF and XGB
 - Random Forest was selected as the final model due to its superior accuracy and robustness.
- **Evaluation**

Evaluation metrics such as accuracy, precision, recall, and F1-score were used to assess the models.

Model Name	Accuracy	Precision	Recall	F1	After Hyperparameter Tuning
Logistic Regression	0.483	0.816	0.483	0.579	-
KNN	0.596	0.819	0.596	0.676	-
Naive Bayes	0.545	0.820	0.545	0.634	-
Decision Tree	0.896	0.815	0.896	0.850	0.809
XGBoost	0.898	0.833	0.898	0.851	0.898333
Random Forest	0.804	0.818	0.804	0.811	0.8989666

Fig 8: Comparison of all the algorithms used

- **Deployment**

The deployment of the model was carried out using Flask, a lightweight and versatile web framework in Python. Flask was chosen for its simplicity and efficiency in creating interactive web applications, enabling seamless integration of the trained machine learning model into a user-friendly interface for real-time predictions

6. Future Scope of the Prediction Model

1. **Integration with Real-Time Data:**

Enhancing the model by integrating real-time data, such as live employee feedback, performance metrics, and attendance records, can improve prediction accuracy and enable dynamic monitoring of resignation risks.

2. **Scalability to Diverse Industries:**

While the current model is tailored to a specific dataset, it can be adapted and generalized for use in various industries and regions, accounting for diverse workforce dynamics and organizational structures.

3. **Advanced Feature Incorporation:**

Incorporating additional features, such as behavioural patterns, employee engagement survey results, and external economic factors, can offer deeper insights into resignation tendencies and enhance the model's predictive capabilities.

4. **Predictive Analytics Dashboards:**

Developing comprehensive dashboards for HR departments can provide visualized insights into employee retention trends, high-risk groups, and actionable strategies to reduce turnover rates.

5. **Integration with HR Management Systems:**

Embedding the model into existing HR software solutions can streamline decision-making by providing automated predictions and suggestions directly within the HR workflow.

6. **Customizable Alerts and Interventions:**

The model can be extended to generate automated alerts for at-risk employees and recommend personalized retention strategies, such as career growth plans, training programs, or incentives.

7. **Cross-Cultural and Regional Analysis:**

By adapting the model for datasets from different countries or regions, organizations can address cultural and economic differences that influence employee behaviour, ensuring more targeted retention strategies.

8. **AI-Driven Workforce Planning:**

Expanding the model to include predictive analytics for workforce planning can help organizations optimize hiring, training, and succession planning based on anticipated turnover trends.

7. Limitations

1. Dependency on Data Quality:

The accuracy of the model heavily relies on the quality and completeness of the dataset. Missing, inconsistent, or inaccurate data, such as outdated employee information or incorrect performance metrics, can impact predictions and reduce reliability.

2. Limited Generalizability:

The model is trained on a dataset from a specific organization and geographic region. Its predictions may not generalize well to other organizations or regions with different workforce dynamics, cultural influences, and economic conditions without retraining on relevant data.

3. Imbalanced Dataset:

The model faced challenges due to the imbalance in the target variable (resignation vs. retention). While techniques like oversampling and SMOTE were applied, the inherent bias in the data may still affect prediction accuracy for the minority class.

4. Exclusion of External Factors:

The model does not account for external factors such as economic downturns, industry trends, or competitor actions, which can significantly influence employee resignations.

5. Limited Interpretability:

While algorithms like Random Forest and XGBoost provide high accuracy, they lack transparency in decision-making compared to simpler models, making it harder for HR teams to fully understand the reasons behind specific predictions.

6. Static Predictions:

The model generates predictions based on historical data and cannot automatically adapt to real-time changes in employee behavior or organizational policies without periodic retraining.

7. Feature Limitations:

Certain important factors, such as employee relationships with managers, workplace culture, or job satisfaction trends over time, are not included in the dataset, potentially limiting the model's ability to capture all aspects of resignation behavior.

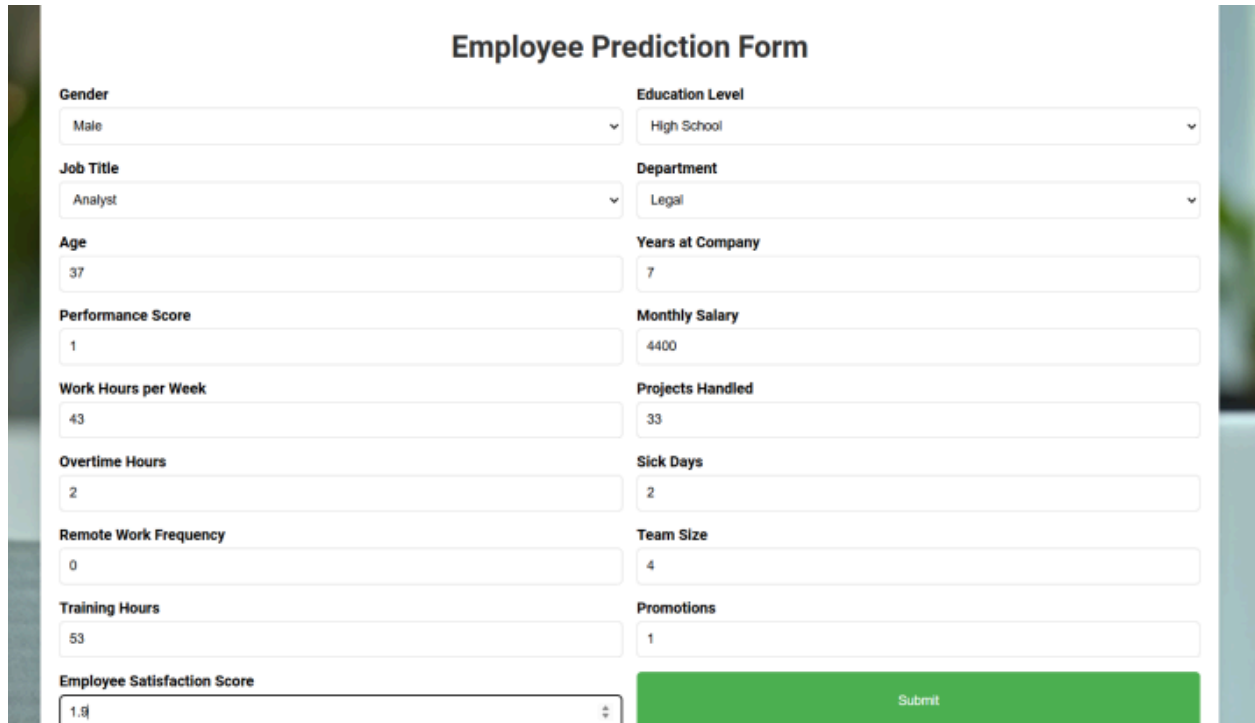
8. Resource-Intensive:

Advanced algorithms like Random Forest and XGBoost require significant computational power for training, which can pose challenges for organizations with limited resources or infrastructure.

9. Ethical and Privacy Concerns:

Using sensitive employee data for predictions raises ethical considerations and requires strict adherence to data privacy and security standards to prevent misuse of personal information.

8. Result



The image shows a web form titled "Employee Prediction Form". It contains 16 input fields arranged in two columns. The left column includes dropdowns for Gender (Male), Job Title (Analyst), Age (37), Performance Score (1), Work Hours per Week (43), Overtime Hours (2), Remote Work Frequency (0), Training Hours (53), and a slider for Employee Satisfaction Score (1.5). The right column includes dropdowns for Education Level (High School), Department (Legal), Years at Company (7), Monthly Salary (4400), Projects Handled (33), Sick Days (2), Team Size (4), and Promotions (1). A green "Submit" button is at the bottom right.

Field	Value
Gender	Male
Education Level	High School
Job Title	Analyst
Department	Legal
Age	37
Years at Company	7
Performance Score	1
Monthly Salary	4400
Work Hours per Week	43
Projects Handled	33
Overtime Hours	2
Sick Days	2
Remote Work Frequency	0
Team Size	4
Training Hours	53
Promotions	1
Employee Satisfaction Score	1.5

fig 9. Home Page

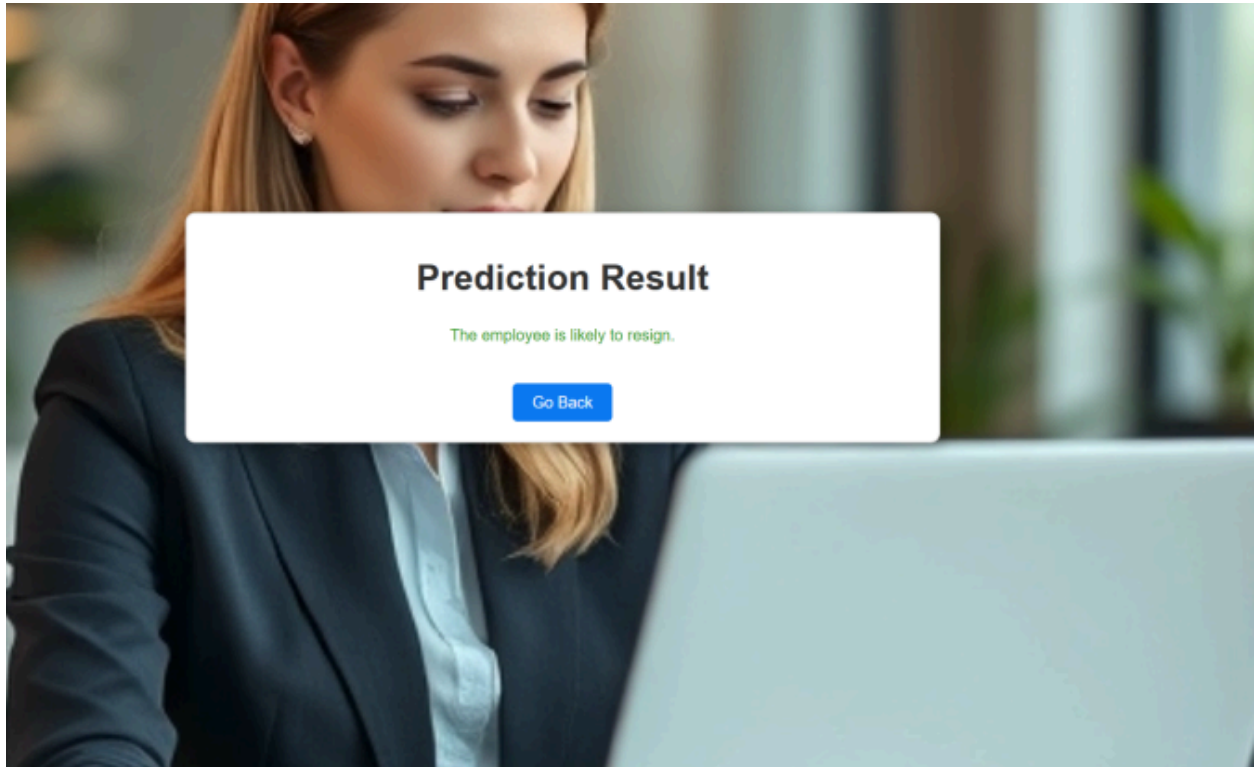


fig 10. Resignation prediction

9. Conclusion

Employee retention is a critical aspect of organizational success, as high resignation rates can disrupt workflows, increase costs, and impact overall productivity. This project utilized a data-driven approach to predict employee resignations, leveraging advanced machine learning algorithms such as Decision Tree, Random Forest, and XGBoost. Through comprehensive evaluation and hyperparameter tuning, Random Forest emerged as the most effective model, delivering the highest accuracy and actionable insights.

By analyzing key factors such as satisfaction levels, job roles, and salaries, the model provides HR departments with valuable predictions to identify at-risk employees and implement proactive retention strategies. While the model demonstrates strong performance, limitations such as dependency on data quality, static predictions, and generalizability challenges highlight areas for future improvement.

Overall, this project showcases the potential of predictive analytics in workforce management. With further enhancements, such as real-time data integration and expanded feature inclusion, the model can become a robust tool for organizations to reduce turnover, optimize resource allocation, and foster a more engaged and stable workforce.

References

- [1] <http://www.google.co.in>
- [2] [Employee Performance Analysis and Resignation P...](#)
- [3] [Hyperparameter tuning - GeeksforGeeks](#)
- [4] [XGBoost Documentation — xgboost 2.1.3 documentation](#)
- [5] [Feature Encoding Techniques - Machine Learning - GeeksforGeeks](#)
- [6] [Stack Overflow - Where Developers Learn, Share, & Build Careers](#)