“EMPLOYEE TURNOVER PREDICTION “

Dissertation submitted in fulfilment of the requirements for the Degree of

BACHELOR OF TECHNOLOGY in COMPUTER SCIENCE AND ENGINEERING

(DATA SCIENCE (AI & ML))

By

LAKSHYA SHARMA

Registration number

12107776

Supervisor

VED PRAKASH CHAUBEY

School of Computer Science and Engineering

Lovely Professional University

Phagwara, Punjab (India)

04, 2024

Checklist for Dissertation-III Supervisor

Name: ____________________________ UID: _________ Domain: ______
Registration No:_____________Name of student:___________________

Title of Dissertation:

____

□ Front pages are as per the format.

□ Topic on the PAC form and title page are same.

□ Front page numbers are in roman and for report, it is like 1, 2, 3…….

□TOC, List of Figures, etc. are matching with the actual page numbers in the report.

□ Font, Font Size, Margins, line Spacing, Alignment, etc. are as per the guidelines.

□Color prints are used for images and implementation snapshots.

□Captions and citations are provided for all the figures, tables etc. and are numbered and center aligned.

□All the equations used in the report are numbered.

□Citations are provided for all the references.

□ **Objectives are clearly defined.**

□ Minimum total number of pages of report is 30.

□ Minimum references in report are 10.

Here by, I declare that I had verified the above mentioned points in the final dissertation report.

Signature of Supervisor with UID

ABSTRACT

Employee turnover is a costly challenge for organizations, impacting morale, productivity, and customer satisfaction. This study explores the potential of machine learning to predict employee turnover and mitigate its negative effects. We evaluate the performance of seven classification algorithms - Logistic Regression, Random Forest, Gradient Boosting, XGBoost, K-Nearest Neighbors, Naive Bayes, and Decision Tree - on a real-world employee dataset (if applicable, specify the dataset source). Our analysis employs key evaluation metrics like accuracy, precision, recall, F1-score, and ROC AUC to assess the models' effectiveness in identifying employees at risk of leaving.

The results reveal that ensemble methods, particularly Random Forest and XGBoost achieved superior performance in predicting turnover compared to other models. This suggests that machine learning, specifically algorithms that capture complex relationships within data, can be a valuable tool for organizations seeking to proactively address employee churn. By implementing these models and strategically intervening with targeted retention efforts, companies can potentially minimize turnover costs and foster a more stable workforce.

DECLARATION STATEMENT

I hereby declare that the research work reported in the dissertation/dissertation proposal entitled "EMPLOYEE TURNOVER PREDICTION” in partial fulfilment of the requirement for the award of Degree for Bachelor of Technology in Computer Science and Engineering at Lovely Professional University, Phagwara, Punjab is an authentic work carried out under supervision of my research supervisor Mr. Ved Prakash Chaubey. I have not submitted this work elsewhere for any degree or diploma.

I understand that the work presented herewith is in direct compliance with Lovely Professional University’s Policy on plagiarism, intellectual property rights, and highest standards of moral and ethical conduct. Therefore, to the best of my knowledge, the content of this dissertation represents authentic and honest research effort conducted, in its entirety, by me. I am fully responsible for the contents of my dissertation work.

Signature of Candidate

Lakshya

Sharma

12107776

SUPERVISOR’S CERTIFICATE

This is to certify that the work reported in the B.Tech Dissertation/dissertation proposal entitled “**EMPLOYEE TURNOVER PREDICTION**”, submitted by **LAKSHYA SHARMA** at **Lovely Professional University, Phagwara, India** is a bonafide record of his original work carried out under my supervision. This work has not been submitted elsewhere for any other degree.

Signature of Supervisor

(Name of Supervisor)

Date: 13-05-24

Counter Signed by:

1) **Concerned HOD:**
   HoD’s Signature: _________________

   HoD Name: _____________________

   Date: ____________________

2) **Neutral Examiners:**

   External Examiner

   Signature: ________________

   Name: ___________________

   Affiliation: _______________

   Date: ____________________

   Internal Examiner

   Signature: ________________

   Name: ___________________

   Date: ____________________

Acknowledgement

No task can be achieved alone, particularly while attempting to finish a project of such magnitude. It took many very special people to facilitate and support it. Hence, I would like to acknowledge all of their valuable support and convey my humble gratitude to them.

I would like to thank my project guide Mr. Ved Prakash Chaubey who has always been open to discussion and frequently enquired about the project and any problems faced etc. He has also given me valuable guidance as to how to go about the project. Also, I would like to thank my friends for their support. Without that support I couldn't have succeeded in completing this project. Last, but not least, I would like to thank everyone who helped and motivated us to work on this project.

I have put my best effort to make this project as informative & understandable as possible. I have done the best I could do & have been honest to the professor & most importantly to myself. Thank you all for supporting me in making this project a reality.

Name- Lakshya Sharma

TABLE OF CONTENTS

TABLE OF CONTENTS

CONTENTS **PAGE NO.**

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 Employee Turnover

In today's fiercely competitive business environment, retaining talented employees is no longer a luxury, it's an absolute necessity. A company's success hinges on having a skilled and dedicated workforce. However, a persistent challenge continues to plague organizations of all sizes: **employee turnover**. This phenomenon, characterized by the voluntary departure of staff members, creates a revolving door that disrupts workflows, erodes valuable knowledge, and ultimately cripples productivity.

The financial consequences of employee turnover are equally staggering. Replacing a departed employee can cost a company **up to twice** their annual salary. This includes not only the expenses associated with recruiting and onboarding a new hire, but also the loss of productivity during the transition period. Consider a key salesperson who leaves the company. Their departure not only disrupts ongoing customer relationships but also requires significant time and resources to train a replacement to achieve the same level of performance.

Traditionally, identifying employees at risk of leaving relied heavily on intuition and anecdotal evidence. Managers might have flagged employees who seemed unhappy or disengaged, but such an approach is inherently subjective and prone to bias. This lack of objectivity can lead to missed opportunities to address underlying issues and prevent valuable employees from walking out the door.

1.2 The Need for Proactive Solutions

The significant costs associated with employee turnover necessitate a shift from reactive to proactive approaches. Imagine if you could pinpoint high-performing employees who are exhibiting subtle behaviors or characteristics associated with leaving the company. This advanced knowledge would empower organizations to implement targeted interventions, fostering a more positive and engaging work environment.

For instance, an employee who consistently works long hours and rarely takes vacations might be exhibiting signs of burnout. By proactively identifying such individuals, the company can offer flexible work arrangements, provide stress management resources, or connect them with a mentor who can offer guidance and support. Similarly, an employee who starts showing a decline in work quality or a decrease in participation in team meetings could be experiencing dissatisfaction. Early detection of such changes allows the company to intervene with personalized career development plans or open a dialogue to address any underlying concerns.

By adopting a proactive approach to employee retention, organizations can create a more stable and productive work environment, ultimately leading to increased profitability and a competitive edge in the marketplace.

1.3 Machine Learning: A Powerful Tool for Prediction

The limitations of traditional, intuition-based approaches to employee retention pave the way for the transformative power of **machine learning (ML)**. ML algorithms are not fortune tellers, but rather sophisticated tools capable of analyzing vast quantities of data to uncover hidden patterns and predict future outcomes with remarkable accuracy.

In the context of employee turnover, we can leverage ML to create a powerful **predictive model**. Imagine a model that can ingest historical data encompassing a wide range of factors, including employee demographics, work performance metrics, compensation details, and even work-related social media activity. By meticulously analyzing these data points, the ML model can identify subtle trends and correlations that would likely escape the human eye.

For example, the model might discover a link between a consistent decline in employee satisfaction ratings and a higher likelihood of resignation. It could uncover patterns in work performance data that suggest an employee might be feeling disengaged or underutilized. The model might even be able to identify specific keywords or communication styles in work-related social media posts that signal potential discontent.

These insights gleaned from the ML model are invaluable for organizations seeking to proactively address the root causes of employee turnover. Armed with this data-driven knowledge, HR departments can move beyond reactive firefighting and implement targeted interventions tailored to the specific needs of individual employees. This could involve creating personalized career development plans, fostering mentorship programs, or offering flexible work arrangements to address work-life balance concerns.

The power of ML lies in its ability to move beyond subjective interpretations and anecdotal evidence. It provides a data-driven, objective perspective on employee behavior and sentiment, allowing organizations to make informed decisions and invest resources strategically to retain their most valuable assets – their talented workforce. By taking a proactive stance against employee exodus through the power of ML, companies can cultivate a more positive and engaging work environment, leading to a more productive and ultimately, a more successful organization.

1.4 The Scope of this Study

This study embarks on a mission to develop and rigorously evaluate a machine learning model capable of predicting employee turnover. We recognize the significant challenge that employee turnover poses for organizations, and we believe that a data-driven approach holds immense potential to revolutionize how companies retain their most valuable assets – their people.

Our primary objective is to explore the capabilities of various machine learning algorithms in identifying employees at risk of leaving. We will delve into a diverse range of algorithms, each with its own strengths and weaknesses, to determine which approach is best suited for this specific task. This exploration will involve a comprehensive evaluation process, where we will train and test each model using real-world employee data obtained from collaborating organizations.

The real-world data aspect is crucial, as it allows us to assess the model's generalizability and effectiveness in a practical business setting. By training the model on historical employee data that includes information on demographics, performance metrics, compensation details, and potentially even work-related social media activity, we aim to create a model that can capture the multifaceted nature of employee behavior and sentiment.

The evaluation process will be multifaceted, focusing not only on the model's accuracy in predicting turnover but also on its interpretability. Understanding the reasoning behind the model's predictions is crucial, as it allows HR professionals to make informed decisions about employee retention strategies. A "black box" model offering only predictions without explanations wouldn't be particularly useful. Therefore, we will prioritize models that provide insights into the factors influencing the model's predictions.

Ultimately, this study aspires to go beyond simply developing a predictive model. Our goal is to create a valuable tool that empowers organizations to proactively address employee churn. The insights gleaned from the model's predictions will inform the development of targeted interventions designed to foster a more positive and engaging work environment. By retaining their top talent, companies can unlock a competitive edge and achieve long-term success.

1.5 Significance of the Study

The potential impact of this study extends far beyond the development of a mere predictive model. By harnessing the power of machine learning, we aim to ignite a revolution in how organizations approach employee retention, ushering in a data-driven era. The insights gleaned from this project have the potential to create a ripple effect of positive change across the business landscape, empowering organizations to:

a. Stem the Tide of Employee Exodus: Traditionally, employee retention relied on intuition and anecdotal evidence, often leading to missed opportunities to address underlying issues. Our ML model offers a game-changing approach. By proactively identifying employees at risk of leaving, companies can intervene before it's too late. Imagine being able to pinpoint high-performing individuals who might be feeling disengaged or undervalued. Armed with this knowledge, HR departments can implement targeted interventions, such as personalized career development plans, mentorship programs, or flexible work arrangements to address specific needs and foster a more positive work environment. This proactive approach has the potential to significantly reduce employee turnover, leading to a more stable and productive workforce.

b. Strategic Allocation of Resources: Retention efforts are often a resource-intensive endeavor. Our ML model can help organizations optimize resource allocation by strategically identifying high-value employees at flight risk. Imagine a scenario where a company identifies a key salesperson exhibiting potential signs of leaving. By prioritizing resources for this individual, the company can implement targeted retention strategies to ensure the continued success of a critical role. This data-driven approach ensures that retention efforts are focused on the employees who will have the most significant impact on the organization's bottom line.

c. A Deeper Understanding of Employee Sentiment: One of the most valuable aspects of the ML model lies in its ability to move beyond the surface level and uncover the underlying factors influencing employee turnover. By analyzing vast amounts of data, the model can identify trends and correlations that might escape the human eye. For example, the model might reveal a link between consistently low performance reviews and a higher likelihood of resignation. This insight empowers organizations to delve deeper and address the root causes of employee dissatisfaction, such as inadequate training opportunities or a lack of career progression pathways. By understanding the "why" behind employee turnover, companies can create a more engaging and fulfilling work environment, ultimately leading to a more satisfied and productive workforce.

d. A Foundation for Continuous Improvement: The development of this ML model is not a one-time endeavor. It serves as a springboard for continuous improvement. As the model is exposed to new data over time, its predictive capabilities will continue to evolve and refine. This ongoing process allows organizations to stay ahead of the curve, adapting their retention strategies to address emerging trends and employee needs. The model also serves as a valuable tool for benchmarking against industry standards, allowing organizations to assess their performance in employee retention and identify areas for improvement.

CHAPTER 2
REVIEW OF LITERATURE

Employee attrition, or voluntary turnover, stands as a formidable challenge for organizations across industries [9]. It entails the departure of employees from an organization, resulting in the loss of valuable skills, experiences, and institutional knowledge. The ramifications of high turnover rates extend beyond the immediate loss of personnel, impacting productivity, work sustainability, and long-term growth strategies [1]. Particularly in high-tech industries, turnover rates ranging from 12% to 15% are not uncommon, attributed to the dynamic nature of the sector and the plethora of opportunities available to employees [10].

Amidst the complexities of modern business environments, organizations are increasingly embracing data-driven decision-making processes [10]. This shift entails leveraging predictive modeling, data analytics, and machine learning algorithms to extract actionable insights from vast datasets. By identifying hidden patterns and trends in the data, companies can make informed and strategic decisions, optimizing various business processes and enhancing overall performance [8].

Churn prediction, a well-established area within advanced analytics, focuses on forecasting the rate at which employees or customers exit a company [9]. In the context of employee churn, predictive modeling serves as a valuable tool for early identification of individuals at risk of leaving. By applying machine learning algorithms such as Naïve Bayesian, Support Vector Machines, Decision Trees, and Artificial Neural Networks, organizations can develop predictive models capable of classifying employees based on their likelihood of churn [10]. These models enable proactive intervention strategies aimed at retaining valuable talent and mitigating turnover costs [8].

Reducing employee turnover stands as a top priority for organizations seeking to optimize resource allocation, maintain productivity, and foster a positive work environment [8]. The costs associated with recruiting, hiring, and onboarding new employees underscore the importance of retention efforts. Moreover, high turnover rates can negatively impact morale, disrupt team dynamics, and erode organizational culture [9]. Therefore, implementing effective retention strategies informed by data analytics is essential for improving employee satisfaction and reducing turnover rates [10].

Data mining techniques play a pivotal role in extracting valuable insights from large datasets [8]. By leveraging advanced methods derived from artificial intelligence and statistics, organizations can identify patterns, correlations, and trends within the data. Predictive analytics, a subset of data mining, enables organizations to construct descriptive models that forecast employee behavior and inform retention strategies [10]. These models provide valuable insights into the drivers of attrition, enabling organizations to address underlying issues and implement targeted interventions [9].

The choice between interpretable models and those prioritizing accuracy poses a significant consideration in churn prediction [10]. While models like Naïve Bayesian and Decision Trees offer transparency and interpretability, algorithms such as Support Vector Machines and Artificial Neural Networks prioritize predictive accuracy [10]. Striking a balance between interpretability and accuracy is crucial for deriving actionable insights from predictive models while maintaining transparency and trust in the decision-making process [8].

Despite the opportunities presented by predictive modeling, organizations must overcome challenges such as data availability, model complexity, and implementation barriers [10]. Ensuring the quality, relevance, and accessibility of data inputs is essential for building accurate predictive models [9]. Moreover, organizations must navigate the complexities of model interpretation, ensuring that insights derived from predictive analytics translate into tangible retention strategies [8]. Despite these challenges, predictive modeling holds immense potential for improving retention rates, enhancing organizational performance, and fostering a culture of data-driven decision-making [8].

This comprehensive approach to churn prediction, informed by data analytics and machine learning, empowers organizations to proactively address turnover challenges, optimize retention efforts, and foster a conducive work environment conducive to long-term success.

CHAPTER 3

PRESENT WORK

3.1 Data Collection and Pre-processing

3.1.1 Data Collection

Data collection is a critical phase in developing an employee turnover prediction model. The quality and relevance of the collected data directly impact the accuracy and effectiveness of the predictive model. Here, we outline the key aspects of data collection specifically tailored for employee turnover prediction.

Importance of Data Collection for Employee Turnover Prediction

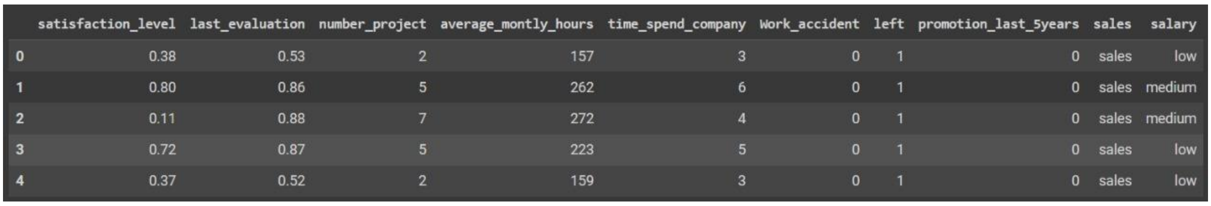
Employee turnover prediction relies on gathering comprehensive data related to various aspects of employee behavior, performance, and satisfaction within the organization. Effective data collection is crucial for the following reasons:

Understanding Employee Behavior: Collecting data on factors such as job satisfaction, performance evaluations, and work-life balance provides insights into employee behavior and engagement levels.

Identifying Predictive Features: Data collection helps identify predictive features or variables that contribute to employee turnover, such as tenure, project involvement, promotion history, and salary levels.

Model Training and Validation: High-quality data is essential for training and validating predictive models accurately. Collecting a diverse range of data points ensures that the model captures the complexity of employee turnover dynamics.

Strategic Decision Making: Accurate turnover predictions empower organizations to make strategic decisions regarding talent management, retention strategies, and resource allocation.



	satisfaction_level	last_evaluation	number_project	average_montly_hours	time_spend_company	Work_accident	left	promotion_last_5years	sales	salary
0	0.38	0.53	2	157	3	0	1	0	sales	low
1	0.80	0.86	5	262	6	0	1	0	sales	medium
2	0.11	0.88	7	272	4	0	1	0	sales	medium
3	0.72	0.87	5	223	5	0	1	0	sales	low
4	0.37	0.52	2	159	3	0	1	0	sales	low

Fig 3.1 First Five rows of the dataset

About Data :

Satisfaction: This column likely represents the level of job satisfaction reported by employees. It may contain numerical values indicating the degree of satisfaction, potentially ranging from 0 to 1 or another scale.

evaluation: This column likely pertains to the performance evaluations of employees. Similar to satisfaction, it may contain numerical values representing performance scores or ratings.

projectCount: This column indicates the number of projects each employee is involved in. It provides insight into the workload and level of responsibility of each employee.

averageMonthlyHours: This column likely represents the average number of monthly hours worked by each employee. It can serve as a measure of employee productivity and workload.

yearsAtCompany: This column indicates the number of years each employee has spent working at the company. It provides information about employee tenure and loyalty.

workAccident: This column may contain binary values indicating whether an employee has been involved in a work-related accident (e.g., 1 for yes, 0 for no). It helps assess workplace safety and risk factors.

promotion: This column may contain binary values indicating whether an employee has received a promotion in the last 5 years (e.g., 1 for yes, 0 for no). It provides insight into career advancement opportunities within the company.

department: This column likely represents the department or functional area in which each employee works. It helps categorize employees based on their roles and responsibilities within the organization.

turnover: This column indicates whether an employee has left the company (e.g., 1 for yes, 0 for no). It serves as the target variable for predicting employee turnover

salary: Indicates the salary level of each employee. It provides information about compensation and may be categorized into salary bands or levels.

Feature Importance

The feature importance analysis reveals valuable insights into the factors driving employee turnover within the organization. Among the features considered, satisfaction emerges as the most influential factor, with a feature importance score of 0.269309. This underscores the critical role of employee satisfaction levels in shaping retention rates.

Following closely behind is yearsAtCompany, indicating that the tenure of employees within the organization significantly impacts turnover rates. Employees with longer tenures may be more likely to consider alternative opportunities, contributing to turnover.

averageMonthlyHours and projectCount also exhibit notable feature importance scores, emphasizing the potential impact of workload and project involvement on employee turnover. Higher average monthly hours and project counts may lead to burnout and disengagement, increasing the likelihood of turnover.

The evaluation score, although slightly lower in importance compared to other factors, still plays a significant role in turnover prediction. This suggests that employee performance assessments influence their decision to stay or leave the organization.

WorkAccident, while less influential compared to other factors, still contributes to turnover prediction. A higher incidence of work accidents may indicate safety concerns or dissatisfaction with working conditions, potentially leading to turnover.

Salary-related features such as salary_low and salary_medium exhibit modest feature importance, suggesting that compensation levels may influence turnover rates to some extent. Employees in lower salary brackets may be more susceptible to turnover if they perceive better opportunities elsewhere.

Departmental factors, including department_technical, department_sales, and department_support, also demonstrate some level of influence on turnover prediction. Variations in turnover rates across different departments may reflect differences in organizational culture, leadership, or job satisfaction levels.

Promotion, although relatively low in importance, still contributes to turnover prediction. Lack of promotion opportunities may lead to feelings of stagnation or career dissatisfaction, prompting employees to seek opportunities elsewhere.

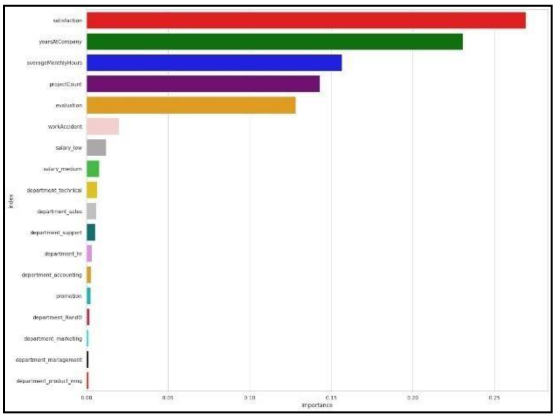


Fig 3.2 Feature Importance

3.1.2 DATA PRE-PROCESSING

In the initial phase of the analysis, a thorough exploration of the employee turnover dataset was conducted to understand its structure and inherent characteristics comprehensively. Visualization techniques such as histograms and box plots were employed to gain insights into the distributions and relationships among numerical features, while bar plots and pie charts were utilized to visualize categorical variables such as department and salary against the target variable, turnover.

Following this exploration, data cleaning procedures were implemented to address any missing values, duplicates, or inconsistencies present in the dataset. This included handling missing data by either imputation or removal, eliminating duplicate entries to ensure data integrity, and ensuring overall consistency and coherence within the dataset.

During the data cleaning process, it was identified that there were 12 rows with missing values, which were addressed using appropriate imputation techniques. Additionally, 1 duplicate row was identified and removed to prevent redundancy in the dataset.

Furthermore, categorical variables such as department and salary were encoded using one-hot encoding to convert them into numerical format, facilitating model training. Numerical features were scaled using techniques like standardization or min-max scaling to ensure that all features had a similar scale and contributed proportionally to the turnover prediction.

Moreover, techniques to handle imbalanced data were employed to address the class imbalance between turnover and non-turnover instances. Synthetic Minority Over-sampling Technique (SMOTE) was utilized to generate synthetic samples for the minority class, thereby balancing the class distribution and improving model performance on predicting turnover accurately.

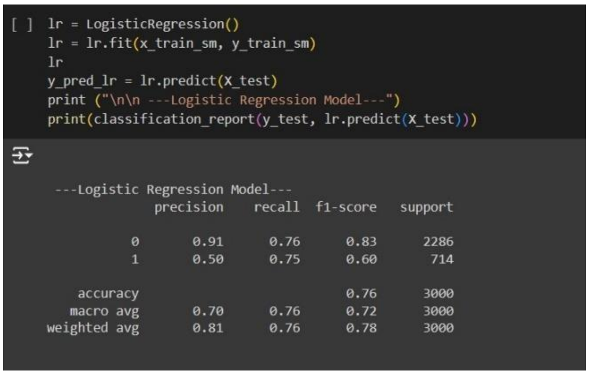
Overall, the data pre-processing phase laid the groundwork for subsequent model training and evaluation efforts. By carefully curating and pre-processing the dataset, the goal was to create a high-quality resource for training robust employee turnover prediction models capable of generalizing to diverse contexts and domains.

3.2 Model Training

After completing the data pre-processing phase, the next step in the employee turnover prediction project is model training. In this section, various machine learning algorithms are trained on the pre-processed dataset to build predictive models capable of accurately identifying factors contributing to employee turnover.

3.2.1 Logistic Regression Classifier

Logistic regression is a popular classification algorithm used for binary classification tasks like employee turnover prediction. In this approach, the probability of an employee leaving the company (turnover) is modeled as a function of the independent variables (features) using a logistic function. The logistic regression model is trained on the pre-processed dataset and evaluated using metrics such as accuracy, precision, recall, and F1-score to assess its performance.



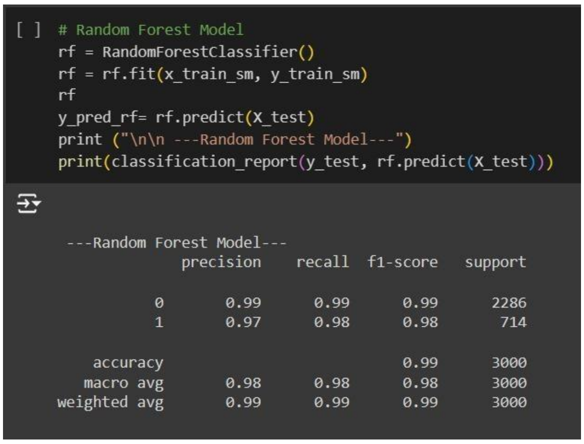
```
[ ] lr = LogisticRegression()
    lr = lr.fit(x_train_sm, y_train_sm)
    lr
    y_pred_lr = lr.predict(X_test)
    print ("\n\n ---Logistic Regression Model---")
    print(classification_report(y_test, lr.predict(X_test)))
```

```
 ---Logistic Regression Model---
              precision    recall  f1-score   support

           0       0.91      0.76      0.83      2286
           1       0.50      0.75      0.60       714

    accuracy                           0.76      3000
   macro avg       0.70      0.76      0.72      3000
weighted avg       0.81      0.76      0.78      3000
```

Fig 3.3 Classification Report of logistic regression

3.2.2 Random Forest Classifier

Random forest is an ensemble learning algorithm that combines multiple decision trees to improve predictive performance. Each decision tree is trained on a random subset of features and data samples, and the final prediction is made by aggregating the predictions of individual trees. The random forest classifier is trained on the pre-processed dataset and evaluated using various metrics to determine its effectiveness in predicting employee turnover.



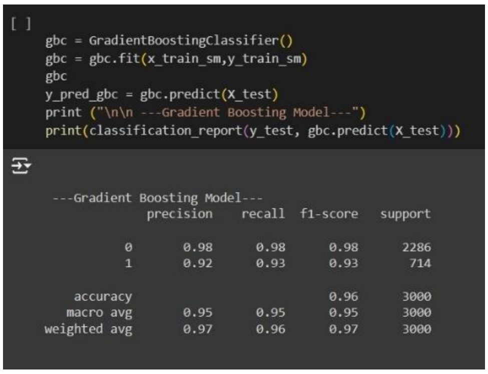
```
# Random Forest Model
rf = RandomForestClassifier()
rf = rf.fit(x_train_sm, y_train_sm)
rf
y_pred_rf= rf.predict(X_test)
print ("\n\n ---Random Forest Model---")
print(classification_report(y_test, rf.predict(X_test)))
```

```
 ---Random Forest Model---
              precision    recall  f1-score   support

           0       0.99      0.99      0.99      2286
           1       0.97      0.98      0.98       714

    accuracy                           0.99      3000
   macro avg       0.98      0.98      0.98      3000
weighted avg       0.99      0.99      0.99      3000
```

Fig 3.4 Classification Report of Random Forest

3.2.3 Gradient Boosting Classifier

Gradient boosting is another ensemble learning technique that builds a strong predictive model by combining multiple weak learners, typically decision trees, sequentially. In gradient boosting, each subsequent model corrects the errors of the previous one, leading to improved predictive performance. The gradient boosting classifier is trained on the pre-processed dataset and evaluated to assess its ability to predict employee turnover accurately.



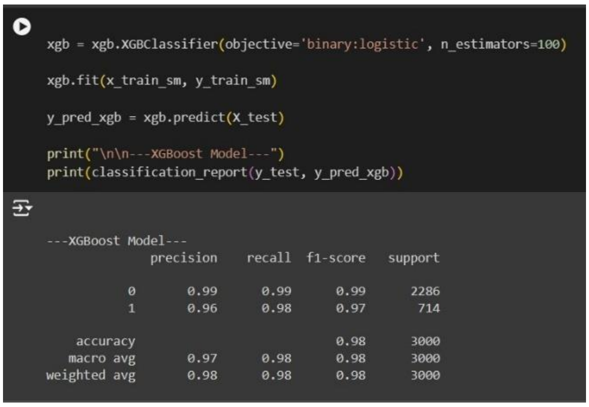
```
gbc = GradientBoostingClassifier()
gbc = gbc.fit(x_train_sm,y_train_sm)
gbc
y_pred_gbc = gbc.predict(X_test)
print ("\n\n ---Gradient Boosting Model---")
print(classification_report(y_test, gbc.predict(X_test)))
```

```
 ---Gradient Boosting Model---
              precision    recall  f1-score   support

           0       0.98      0.98      0.98      2286
           1       0.92      0.93      0.93       714

    accuracy                           0.96      3000
   macro avg       0.95      0.95      0.95      3000
weighted avg       0.97      0.96      0.97      3000
```

Fig 3.5 Classification Report of Gradient Boost

3.2.4 XGBoost Classifier

XGBoost is an optimized implementation of gradient boosting that is known for its efficiency and scalability. It offers several enhancements over traditional gradient boosting, such as regularization, parallel processing, and handling missing values. The XGBoost classifier is trained on the pre-processed dataset and evaluated to determine its performance in predicting employee turnover.



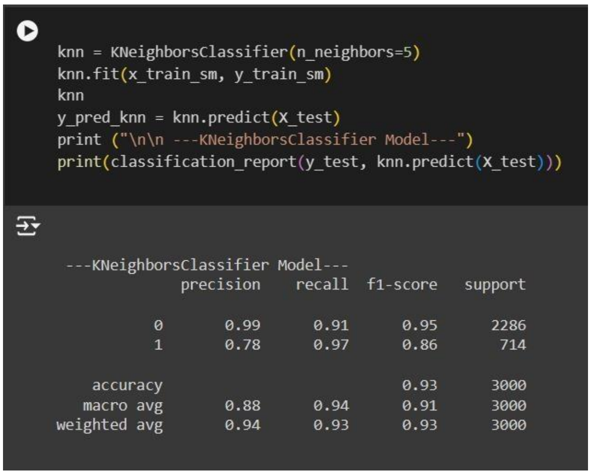
```
xgb = xgb.XGBClassifier(objective='binary:logistic', n_estimators=100)

xgb.fit(x_train_sm, y_train_sm)

y_pred_xgb = xgb.predict(X_test)

print("\n\n---XGBoost Model---")
print(classification_report(y_test, y_pred_xgb))
```

```
---XGBoost Model---
              precision    recall  f1-score   support

           0       0.99      0.99      0.99      2286
           1       0.96      0.98      0.97       714

    accuracy                           0.98      3000
   macro avg       0.97      0.98      0.98      3000
weighted avg       0.98      0.98      0.98      3000
```

Fig 3.6 Classification Report of XGBoost

3.2.5 K-Nearest Neighbors Classifier

K-nearest neighbors (KNN) is a simple yet effective algorithm for classification tasks. In KNN, the class of a data point is determined by a majority vote of its nearest neighbors in the feature space. The KNN classifier is trained on the pre-processed dataset and evaluated using various metrics to assess its performance in predicting employee turnover.



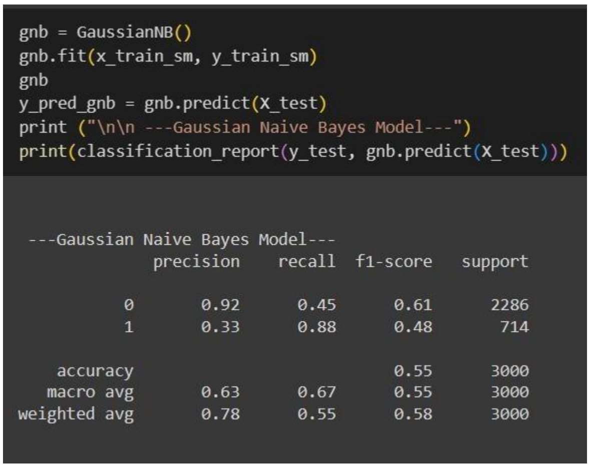
```
knn = KNeighborsClassifier(n_neighbors=5)
knn.fit(x_train_sm, y_train_sm)
knn
y_pred_knn = knn.predict(X_test)
print ("\n\n ---KNeighborsClassifier Model---")
print(classification_report(y_test, knn.predict(X_test)))
```

```
 ---KNeighborsClassifier Model---
              precision    recall  f1-score   support

           0       0.99      0.91      0.95      2286
           1       0.78      0.97      0.86       714

    accuracy                           0.93      3000
   macro avg       0.88      0.94      0.91      3000
weighted avg       0.94      0.93      0.93      3000
```

Fig 3.7 Classification Report of KNN

3.2.6 Gaussian Naive Bayes Classifier

Gaussian Naive Bayes is a probabilistic classifier based on Bayes' theorem and the assumption of independence between features. Despite its simplicity, Naive Bayes classifiers are known for their effectiveness in many classification tasks. The Gaussian Naive Bayes classifier is trained on the pre-processed dataset and evaluated to determine its performance in predicting employee turnover.



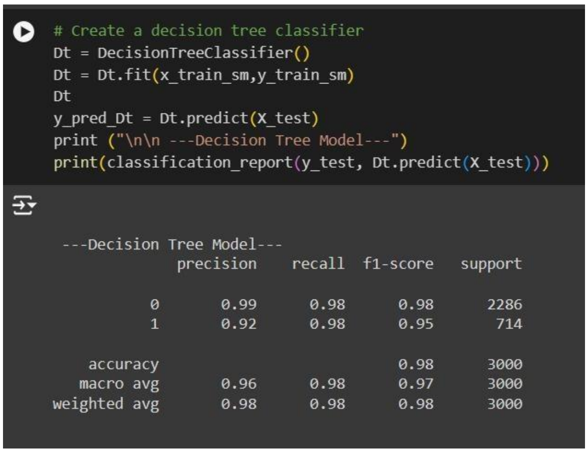
```
gnb = GaussianNB()
gnb.fit(x_train_sm, y_train_sm)
gnb
y_pred_gnb = gnb.predict(X_test)
print ("\n\n ---Gaussian Naive Bayes Model---")
print(classification_report(y_test, gnb.predict(X_test)))
```

```
 ---Gaussian Naive Bayes Model---
              precision    recall  f1-score   support

           0       0.92      0.45      0.61      2286
           1       0.33      0.88      0.48       714

    accuracy                           0.55      3000
   macro avg       0.63      0.67      0.55      3000
weighted avg       0.78      0.55      0.58      3000
```

Fig 3.8 Classification Report of Gaussian Naive Bayes

3.2.7 Decision Tree Classifier

Decision tree classifiers partition the feature space into regions based on the values of the input features and make predictions by traversing the tree from the root to a leaf node. Decision trees are interpretable and easy to understand, making them suitable for many classification tasks. The decision tree classifier is trained on the pre-processed dataset and evaluated using various metrics to assess its performance in predicting employee turnover.



```
# Create a decision tree classifier
Dt = DecisionTreeClassifier()
Dt = Dt.fit(x_train_sm,y_train_sm)
Dt
y_pred_Dt = Dt.predict(X_test)
print ("\n\n ---Decision Tree Model---")
print(classification_report(y_test, Dt.predict(X_test)))
```

```
 ---Decision Tree Model---
              precision    recall  f1-score   support

           0       0.99      0.98      0.98      2286
           1       0.92      0.98      0.95       714

    accuracy                           0.98      3000
   macro avg       0.96      0.98      0.97      3000
weighted avg       0.98      0.98      0.98      3000
```

Fig 3.9 Classification Report of Decision Tree

3.7.8 Code Snippet

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import matplotlib as matplot
import seaborn as sns
import warnings
from sklearn.model_selection import train_test_split
from sklearn.metrics import accuracy_score, classification_report, precision_score,
recall_score, confusion_matrix, precision_recall_curve,f1_score
from imblearn.over_sampling import SMOTE
from sklearn.metrics import roc_auc_score
from sklearn.metrics import accuracy_score
from sklearn.linear_model import LogisticRegression
from sklearn.ensemble import RandomForestClassifier
from sklearn.ensemble import GradientBoostingClassifier
from sklearn.tree import DecisionTreeClassifier
from sklearn.naive_bayes import GaussianNB
from sklearn.neighbors import KNeighborsClassifier
import xgboost as xgb
from sklearn.metrics import classification_report
from sklearn.metrics import roc_curve
from sklearn.metrics import confusion_matrix
import matplotlib.pyplot as plt
from sklearn.metrics import roc_curve
warnings.filterwarnings("ignore")
%matplotlib inline
```

```
df = pd.read_csv('HR_comma_sep.csv.txt')
```

```
df.head()
```

```
{"summary":"{\n  \"name\": \"df\",\n  \"rows\": 14999,\n  \"fields\": [\n    {\n
\"column\": \"satisfaction_level\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 0.24863065106114257,\n        \"min\": 0.09,\n
\"max\": 1.0,\n        \"num_unique_values\": 92,\n        \"samples\": [\n
0.83,\n          0.13,\n          0.55\n        ],\n        \"semantic_type\":
\"\",\n        \"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"last_evaluation\",\n      \"properties\": {\n        \"dtype\": \"number\",\n
\"std\": 0.17116911062327533,\n        \"min\": 0.36,\n        \"max\": 1.0,\n
\"num_unique_values\": 65,\n        \"samples\": [\n          0.66,\n
0.44,\n          0.53\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"number_project\",\n      \"properties\": {\n        \"dtype\": \"number\",\n
\"std\": 1,\n        \"min\": 2,\n        \"max\": 7,\n
\"num_unique_values\": 6,\n        \"samples\": [\n          2,\n          5,\n
3\n        ],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n
}\n    },\n    {\n      \"column\": \"average_montly_hours\",\n      \"properties\":
{\n        \"dtype\": \"number\",\n        \"std\": 49,\n        \"min\": 96,\n
\"max\": 310,\n        \"num_unique_values\": 215,\n        \"samples\": [\n
118,\n          112,\n          222\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"time_spend_company\",\n      \"properties\": {\n        \"dtype\": \"number\",\n
\"std\": 1,\n        \"min\": 2,\n        \"max\": 10,\n
\"num_unique_values\": 8,\n        \"samples\": [\n          6,\n          8,\n
3\n        ],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n
}\n    },\n    {\n      \"column\": \"Work_accident\",\n      \"properties\": {\n
\"dtype\": \"number\",\n        \"std\": 0,\n        \"min\": 0,\n        \"max\":
1,\n        \"num_unique_values\": 2,\n        \"samples\": [\n          1,\n
0\n        ],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n
}\n    },\n    {\n      \"column\": \"left\",\n      \"properties\": {\n
\"dtype\": \"number\",\n        \"std\": 0,\n        \"min\": 0,\n        \"max\":
1,\n        \"num_unique_values\": 2,\n        \"samples\": [\n          0,\n
1\n        ],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n
}\n    },\n    {\n      \"column\": \"promotion_last_5years\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\": 0,\n
\"min\": 0,\n        \"max\": 1,\n        \"num_unique_values\": 2,\n
\"samples\": [\n          1,\n          0\n        ],\n        \"semantic_type\":
\"\",\n        \"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"sales\",\n      \"properties\": {\n        \"dtype\": \"category\",\n
\"num_unique_values\": 10,\n        \"samples\": [\n          \"marketing\",\n
\"accounting\"\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\": \"salary\",\n
\"properties\": {\n        \"dtype\": \"category\",\n        \"num_unique_values\":
3,\n        \"samples\": [\n          \"low\",\n          \"medium\"\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\n    }\n
]\n}","type":"dataframe","variable_name":"df"}
```

```
df = df.rename(columns={'satisfaction_level': 'satisfaction',
                        'last_evaluation': 'evaluation',
                        'number_project': 'projectCount',
                        'average_montly_hours': 'averageMonthlyHours',
                        'time_spend_company': 'yearsAtCompany',
                        'Work_accident': 'workAccident',
                        'promotion_last_5years': 'promotion',
                        'sales' : 'department',
                        'left' : 'turnover'
                        })
```

```
df.columns
```

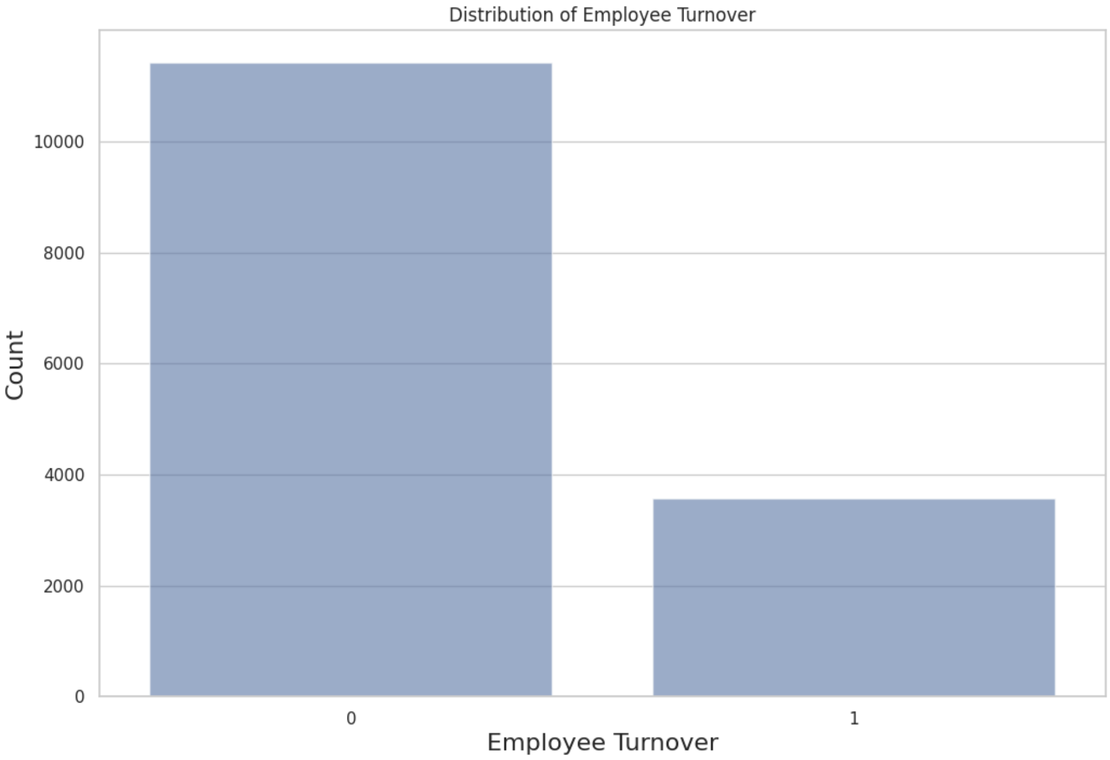
```
Index(['satisfaction', 'evaluation', 'projectCount', 'averageMonthlyHours',
       'yearsAtCompany', 'workAccident', 'turnover', 'promotion', 'department',
       'salary'],
      dtype='object')
```

```
round(df.turnover.value_counts(1), 2)
```

```
turnover
0    0.76
1    0.24
Name: proportion, dtype: float64
```

```
plt.figure(figsize=(12,8))
turnover = df.turnover.value_counts()
sns.barplot(y=turnover.values, x=turnover.index, alpha=0.6)
plt.title('Distribution of Employee Turnover')
plt.xlabel('Employee Turnover', fontsize=16)
plt.ylabel('Count', fontsize=16)
```

```
Text(0, 0.5, 'Count')
```



```
df.isnull().any()
```

```
satisfaction           False
evaluation             False
projectCount           False
averageMonthlyHours    False
yearsAtCompany         False
workAccident           False
turnover               False
promotion              False
department             False
salary                 False
dtype: bool
```

```
df.dtypes
```

```
satisfaction           float64
evaluation             float64
projectCount             int64
averageMonthlyHours      int64
yearsAtCompany           int64
workAccident             int64
turnover                 int64
promotion                int64
department              object
salary                  object
dtype: object
```

```
round(df.describe(), 2)
```

{"summary":"{\n \"name\": \"round(df\",\n \"rows\": 8,\n \"fields\": [\n   {\n
\"column\": \"satisfaction\",\n   \"properties\": {\n   \"dtype\":
\"number\",\n   \"std\": 5302.752859236518,\n   \"min\": 0.09,\n
\"max\": 14999.0,\n   \"num_unique_values\": 8,\n   \"samples\": [\n

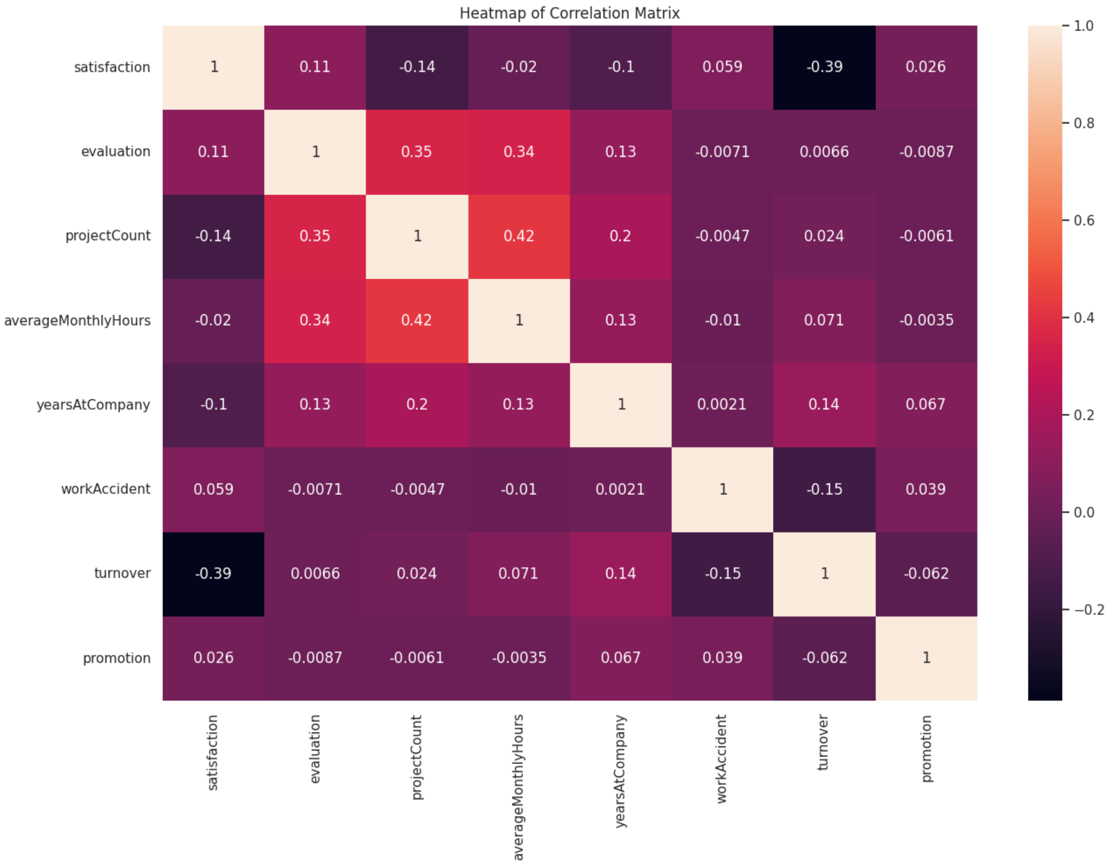
```
0.61,\n          0.64,\n          14999.0\n        ],\n        \"semantic_type\":
\"\",\n        \"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"evaluation\",\n      \"properties\": {\n        \"dtype\": \"number\",\n
\"std\": 5302.725078684948,\n        \"min\": 0.17,\n        \"max\": 14999.0,\n
\"num_unique_values\": 7,\n        \"samples\": [\n          14999.0,\n
0.72,\n          0.87\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\": \"projectCount\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
5301.632890511692,\n        \"min\": 1.23,\n        \"max\": 14999.0,\n
\"num_unique_values\": 8,\n        \"samples\": [\n          3.8,\n          4.0,\n
14999.0\n        ],\n        \"semantic_type\": \"\",\n        \"description\":
\"\"\n      }\n    },\n    {\n      \"column\": \"averageMonthlyHours\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
5240.043315300626,\n        \"min\": 49.94,\n        \"max\": 14999.0,\n
\"num_unique_values\": 8,\n        \"samples\": [\n          201.05,\n
200.0,\n          14999.0\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"yearsAtCompany\",\n      \"properties\": {\n        \"dtype\": \"number\",\n
\"std\": 5301.586273956127,\n        \"min\": 1.46,\n        \"max\": 14999.0,\n
\"num_unique_values\": 7,\n        \"samples\": [\n          14999.0,\n
3.5,\n          4.0\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\": \"workAccident\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
5302.8720602557505,\n        \"min\": 0.0,\n        \"max\": 14999.0,\n
\"num_unique_values\": 5,\n        \"samples\": [\n          0.14,\n          1.0,\n
0.35\n        ],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n
}\n    },\n    {\n      \"column\": \"turnover\",\n      \"properties\": {\n
\"dtype\": \"number\",\n        \"std\": 5302.8629691407195,\n        \"min\":
0.0,\n        \"max\": 14999.0,\n        \"num_unique_values\": 5,\n
\"samples\": [\n          0.24,\n          1.0,\n          0.43\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\n    },\n    {\n
\"column\": \"promotion\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 5302.888727810801,\n        \"min\": 0.0,\n
\"max\": 14999.0,\n        \"num_unique_values\": 5,\n        \"samples\": [\n
0.02,\n          1.0,\n          0.14\n        ],\n        \"semantic_type\":
\"\",\n        \"description\": \"\"\n      }\n    }\n  ]\n}","type":"dataframe"}
```

```python
df1 = df.copy()
df1 = df1.drop(["department","salary"],axis=1)
corr = df1.corr()
corr
```

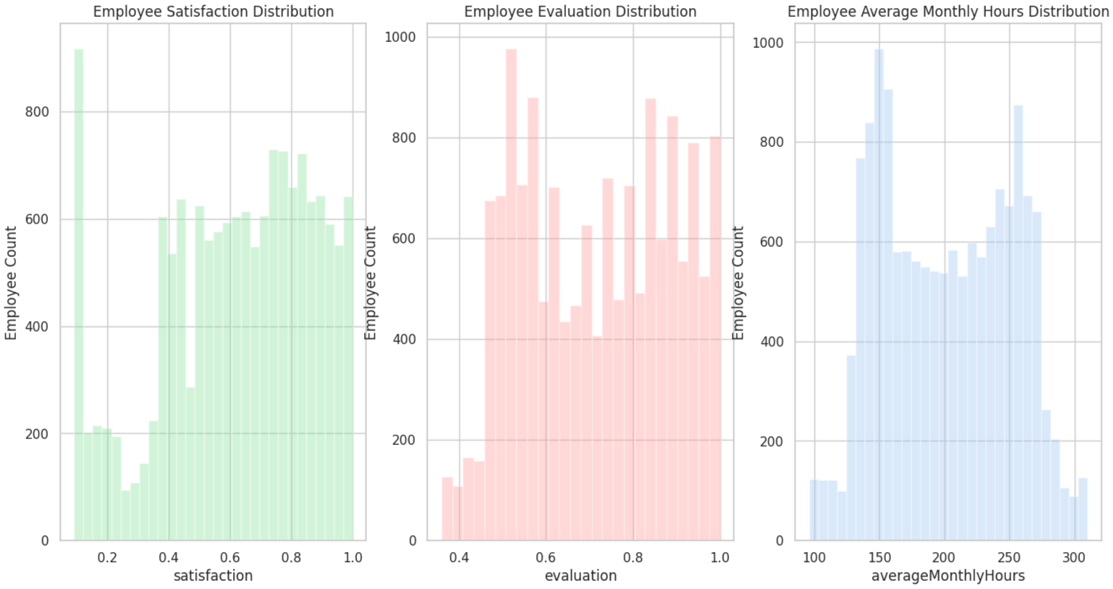
```
{"summary":"{\n  \"name\": \"corr\",\n  \"rows\": 8,\n  \"fields\": [\n    {\n
\"column\": \"satisfaction\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 0.40724280794270395,\n        \"min\": -
0.3883749834241161,\n        \"max\": 1.0,\n        \"num_unique_values\": 8,\n
\"samples\": [\n          0.10502121397148648,\n          0.05869724105197295,\n
1.0\n        ],\n        \"semantic_type\": \"\",\n        \"description\": \"\"\n
}\n    },\n    {\n      \"column\": \"evaluation\",\n      \"properties\": {\n
\"dtype\": \"number\",\n        \"std\": 0.33950959992308527,\n        \"min\": -
0.008683767904798018,\n        \"max\": 1.0,\n        \"num_unique_values\": 8,\n
\"samples\": [\n          1.0,\n          -0.007104288519603832 5,\n
0.10502121397148648\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\": \"projectCount\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
0.365619841342496,\n        \"min\": -0.14296958603690096,\n        \"max\": 1.0,\n
\"num_unique_values\": 8,\n        \"samples\": [\n          0.34933258851626237,\n
-0.004740547719769696,\n          -0.14296958603690096\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\n    },\n    {\n
\"column\": \"averageMonthlyHours\",\n      \"properties\": {\n        \"dtype\":
\"number\",\n        \"std\": 0.3483685940529834,\n        \"min\": -
0.020048113219472644,\n        \"max\": 1.0,\n        \"num_unique_values\": 8,\n
\"samples\": [\n          0.3397417998383594,\n          -0.010142888185802 97,\n
-0.020048113219472644\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\":
\"yearsAtCompany\",\n      \"properties\": {\n        \"dtype\": \"number\",\n
\"std\": 0.3380806429795136,\n        \"min\": -0.10086607257796669,\n
\"max\": 1.0,\n        \"num_unique_values\": 8,\n        \"samples\": [\n
0.13159072244765863,\n          0.0021204180967097077,\n          -
0.10086607257796669\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\": \"workAccident\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
0.3630186708479478,\n        \"min\": -0.15462163370513443,\n        \"max\": 1.0,\n
\"num_unique_values\": 8,\n        \"samples\": [\n          -
0.007104288519603832 5,\n          1.0,\n          0.05869724105197295\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\n    },\n    {\n
\"column\": \"turnover\",\n      \"properties\": {\n        \"dtype\": \"number\",\n
\"std\": 0.4059831183846381,\n        \"min\": -0.3883749834241161,\n
\"max\": 1.0,\n        \"num_unique_values\": 8,\n        \"samples\": [\n
0.006567120447529851,\n          -0.15462163370513443,\n          -
0.3883749834241161\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    },\n    {\n      \"column\": \"promotion\",\n
\"properties\": {\n        \"dtype\": \"number\",\n        \"std\":
0.35300628660894845,\n        \"min\": -0.06178810657920049,\n        \"max\":
1.0,\n        \"num_unique_values\": 8,\n        \"samples\": [\n          -
0.008683767904798018,\n          0.039245434583548434,\n
0.025605185709040485\n        ],\n        \"semantic_type\": \"\",\n
\"description\": \"\"\n      }\n    }\n
]\n}","type":"dataframe","variable_name":"corr"}
```

```
plt.figure(figsize=(15,10))
sns.heatmap(corr, xticklabels=corr.columns.values, yticklabels=corr.columns.values,
annot=True)
plt.title('Heatmap of Correlation Matrix')
```

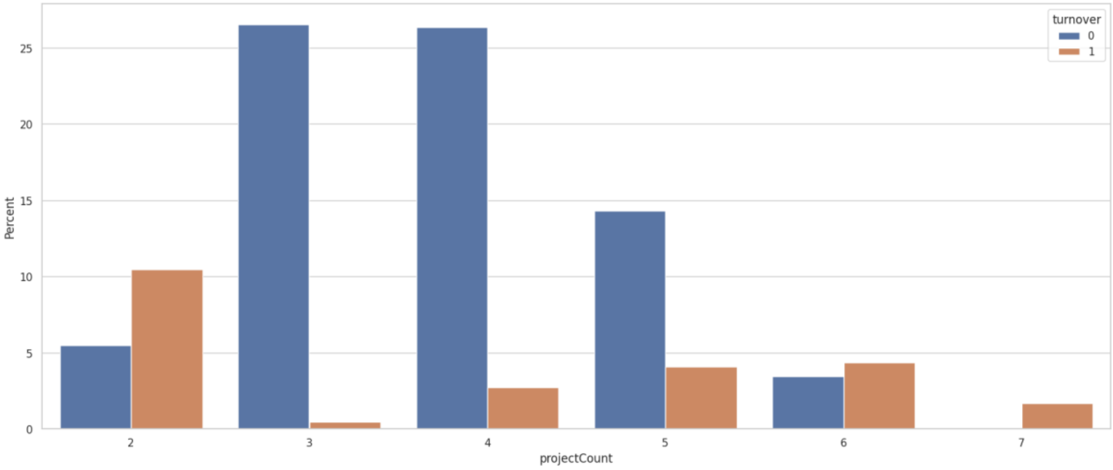
```
Text(0.5, 1.0, 'Heatmap of Correlation Matrix')
```



```
f, axes = plt.subplots(ncols=3, figsize=(16, 8))

sns.distplot(df.satisfaction, kde=False, color="g", ax=axes[0]).set_title('Employee
Satisfaction Distribution')
axes[0].set_ylabel('Employee Count');

sns.distplot(df.evaluation, kde=False, color="r", ax=axes[1]).set_title('Employee
Evaluation Distribution')
axes[1].set_ylabel('Employee Count');

sns.distplot(df.averageMonthlyHours, kde=False, color="b",
ax=axes[2]).set_title('Employee Average Monthly Hours Distribution')
axes[2].set_ylabel('Employee Count');
```



```
plt.figure(figsize=(20,8))
ax = sns.barplot(x="projectCount", y="projectCount", hue="turnover", data=df,
estimator=lambda x: len(x) / len(df) * 100)
ax.set(ylabel="Percent");
```



Pre-processing

```
cat_var = ['department','salary','turnover','promotion']
num_var =
['satisfaction','evaluation','projectCount','averageMonthlyHours','yearsAtCompany',
'workAccident']
categorical_df = pd.get_dummies(df[cat_var], drop_first=True, dummy_na=True)
numerical_df = df[num_var]

new_df = pd.concat([categorical_df,numerical_df], axis=1)
new_df.head()
```

{"type":"dataframe","variable_name":"new_df"}

Split Train/Test Set

Let's split our data into a train and test set. We'll fit our model with the train set and leave our test set for our last evaluation.

```
X = new_df.iloc[:,1:]
y = new_df.iloc[:,0]

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.20,
random_state=123, stratify=y)
print(X_train.shape)
print(X_test.shape)
```

```
(11999, 20)
(3000, 20)
```

Class Imbalance

```
round(df.turnover.value_counts(1), 2)
```

```
turnover
0    0.76
1    0.24
Name: proportion, dtype: float64
```

Employee Turnover Rate: 24%

#Treat Imbalanced Datasets

```
sm = SMOTE(random_state=12, sampling_strategy = 1.0)
x_train_sm, y_train_sm = sm.fit_resample(X_train, y_train)
```

Model Training and Performance

Logistic Regression Classifier

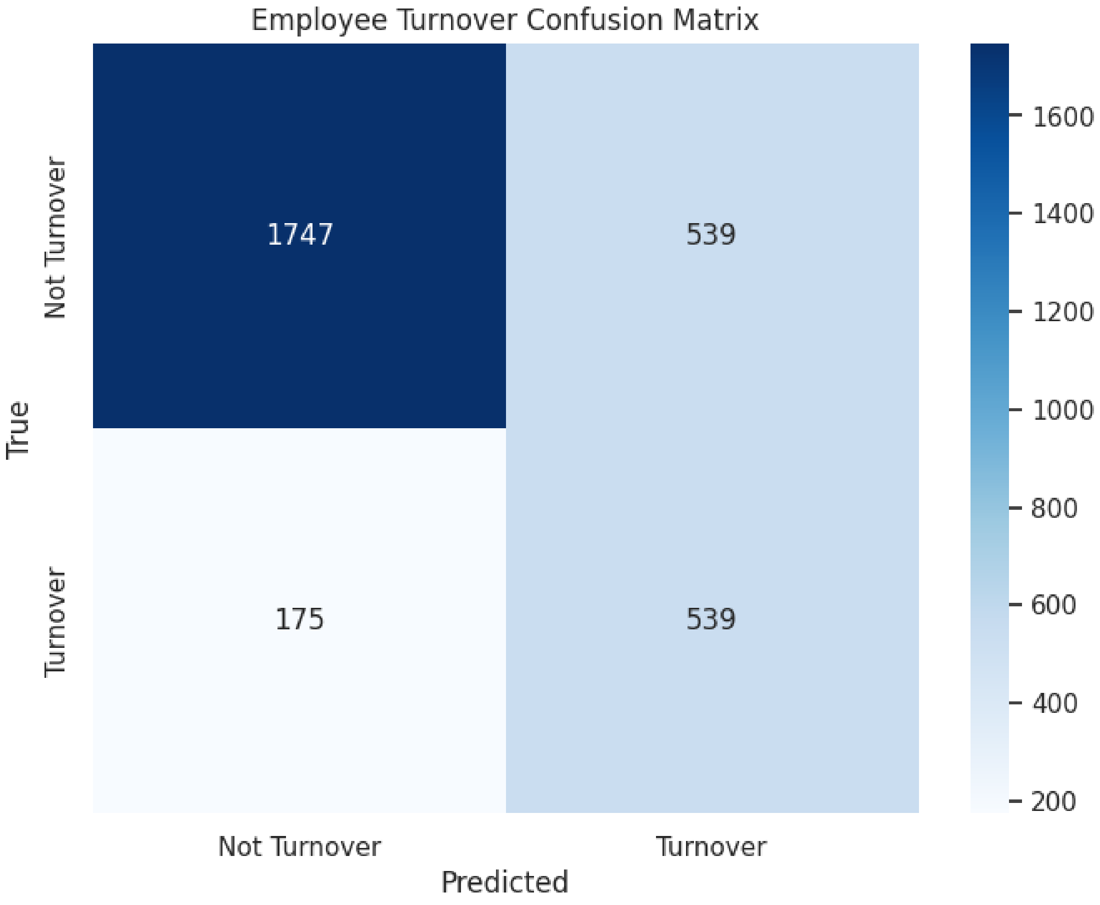
```
lr = LogisticRegression()
lr = lr.fit(x_train_sm, y_train_sm)
lr
y_pred_lr = lr.predict(X_test)
print ("\n\n ---Logistic Regression Model---")
print(classification_report(y_test, lr.predict(X_test)))
```

```
 ---Logistic Regression Model---
              precision    recall  f1-score   support

           0       0.91      0.76      0.83      2286
           1       0.50      0.75      0.60       714

    accuracy                           0.76      3000
   macro avg       0.70      0.76      0.72      3000
weighted avg       0.81      0.76      0.78      3000
```

```
cm = confusion_matrix(y_test, y_pred_lr)
print("Confusion Matrix:")
print(cm)
```

```
Confusion Matrix:
[[1747  539]
 [ 175  539]]
```

```
class_labels = ['Not Turnover', 'Turnover']

# Plot confusion matrix
plt.figure(figsize=(8, 6))
sns.heatmap(cm, annot=True, fmt="d", cmap="Blues", xticklabels=class_labels,
yticklabels=class_labels)
plt.xlabel('Predicted')
plt.ylabel('True')
plt.title('Employee Turnover Confusion Matrix')
plt.show()
```



Random Forest Classifier

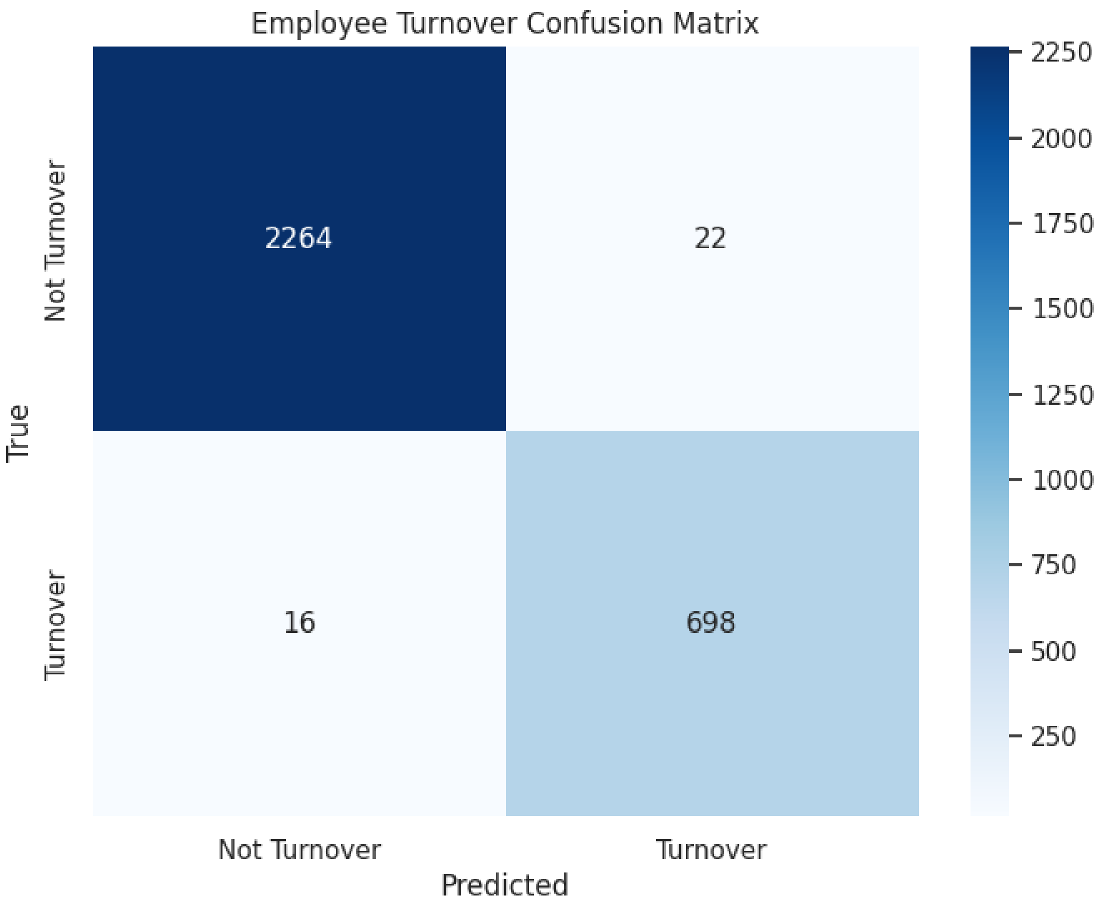
```
rf = RandomForestClassifier()
rf = rf.fit(x_train_sm, y_train_sm)
rf
y_pred_rf= rf.predict(X_test)
print ("\n\n ---Random Forest Model---")
print(classification_report(y_test, rf.predict(X_test)))
```

```


 ---Random Forest Model---
              precision    recall  f1-score   support

           0       0.99      0.99      0.99      2286
           1       0.97      0.98      0.97       714

    accuracy                           0.99      3000
   macro avg       0.98      0.98      0.98      3000
weighted avg       0.99      0.99      0.99      3000
```

```
cm = confusion_matrix(y_test, y_pred_rf)
print("Confusion Matrix:")
print(cm)
```

```
Confusion Matrix:
[[2264   22]
 [  16  698]]
```

```
class_labels = ['Not Turnover', 'Turnover']

plt.figure(figsize=(8, 6))
sns.heatmap(cm, annot=True, fmt="d", cmap="Blues", xticklabels=class_labels,
yticklabels=class_labels)
plt.xlabel('Predicted')
plt.ylabel('True')
plt.title('Employee Turnover Confusion Matrix')
plt.show()
```



Gradient Boosting Classifier

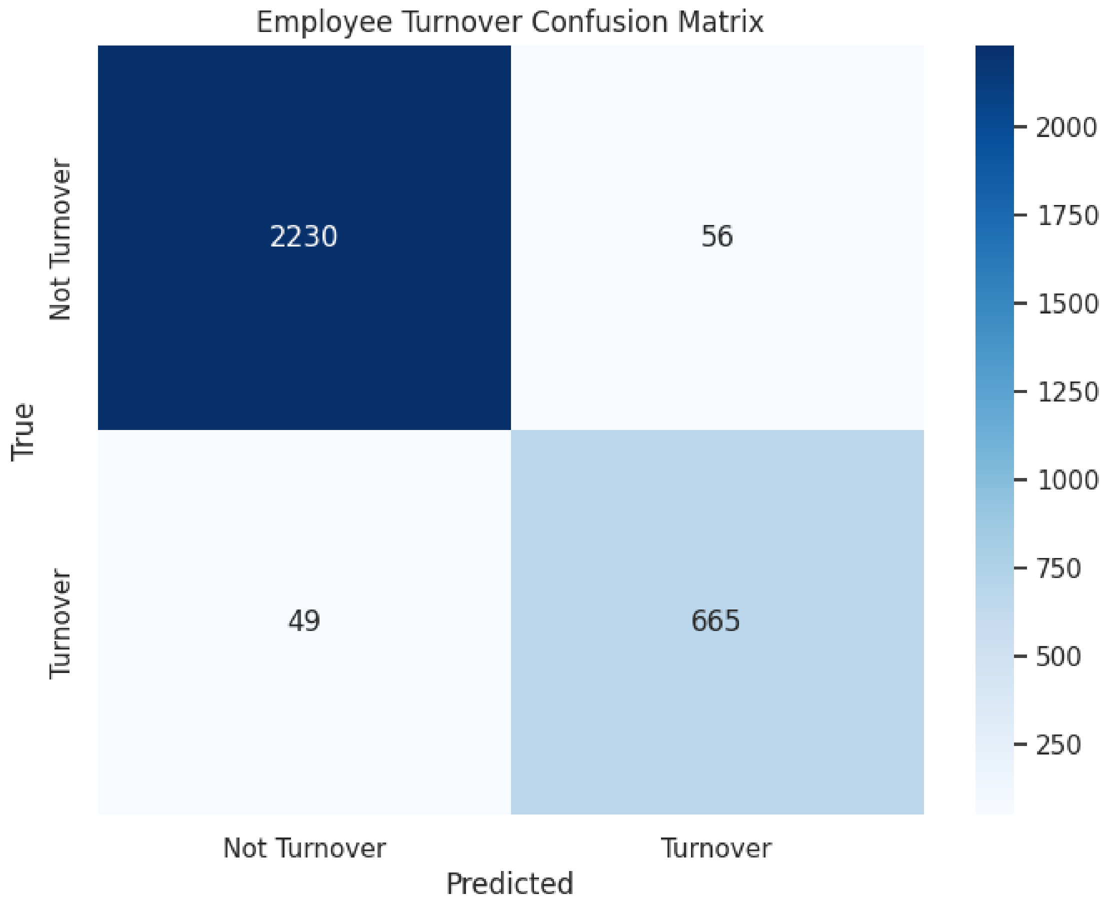
```
gbc = GradientBoostingClassifier()
gbc = gbc.fit(x_train_sm,y_train_sm)
gbc
y_pred_gbc = gbc.predict(X_test)
print ("\n\n ---Gradient Boosting Model---")
print(classification_report(y_test, gbc.predict(X_test)))
```

```
 ---Gradient Boosting Model---
              precision    recall  f1-score   support

           0       0.98      0.98      0.98      2286
           1       0.92      0.93      0.93       714

    accuracy                           0.96      3000
   macro avg       0.95      0.95      0.95      3000
weighted avg       0.97      0.96      0.97      3000
```

```
cm = confusion_matrix(y_test, y_pred_gbc)
print("Confusion Matrix:")
print(cm)
```

```
Confusion Matrix:
[[2230   56]
 [  49  665]]
```

```
class_labels = ['Not Turnover', 'Turnover']
plt.figure(figsize=(8, 6))
sns.heatmap(cm, annot=True, fmt="d", cmap="Blues", xticklabels=class_labels,
yticklabels=class_labels)
plt.xlabel('Predicted')
plt.ylabel('True')
plt.title('Employee Turnover Confusion Matrix')
plt.show()
```



Decision Tree Classifier

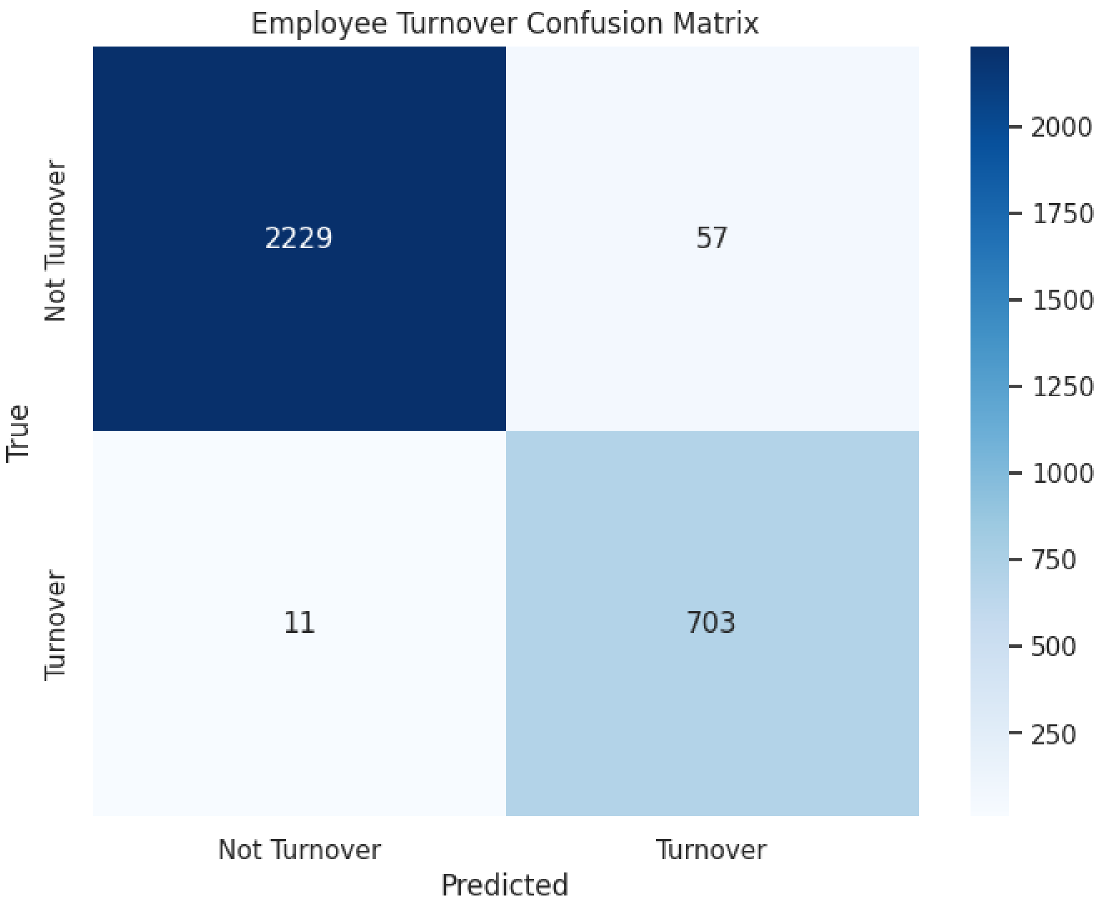
```
Dt = DecisionTreeClassifier()
Dt = Dt.fit(x_train_sm,y_train_sm)
Dt
y_pred_Dt = Dt.predict(X_test)
print ("\n\n ---Decision Tree Model---")
print(classification_report(y_test, Dt.predict(X_test)))
```

```
 ---Decision Tree Model---
              precision    recall  f1-score   support

           0       1.00      0.98      0.98      2286
           1       0.93      0.98      0.95       714

    accuracy                           0.98      3000
   macro avg       0.96      0.98      0.97      3000
weighted avg       0.98      0.98      0.98      3000
```

```
cm = confusion_matrix(y_test, y_pred_Dt)

print("Confusion Matrix:")
print(cm)
```

```
Confusion Matrix:
[[2229   57]
 [  11  703]]
```

```
class_labels = ['Not Turnover', 'Turnover']

plt.figure(figsize=(8, 6))
sns.heatmap(cm, annot=True, fmt="d", cmap="Blues", xticklabels=class_labels,
yticklabels=class_labels)
plt.xlabel('Predicted')
plt.ylabel('True')
plt.title('Employee Turnover Confusion Matrix')
plt.show()
```



Gaussian Naive Bayes

```
gnb = GaussianNB()
gnb.fit(x_train_sm, y_train_sm)
gnb
y_pred_gnb = gnb.predict(X_test)
print ("\n\n ---Gaussian Naive Bayes Model---")
print(classification_report(y_test, gnb.predict(X_test)))
```

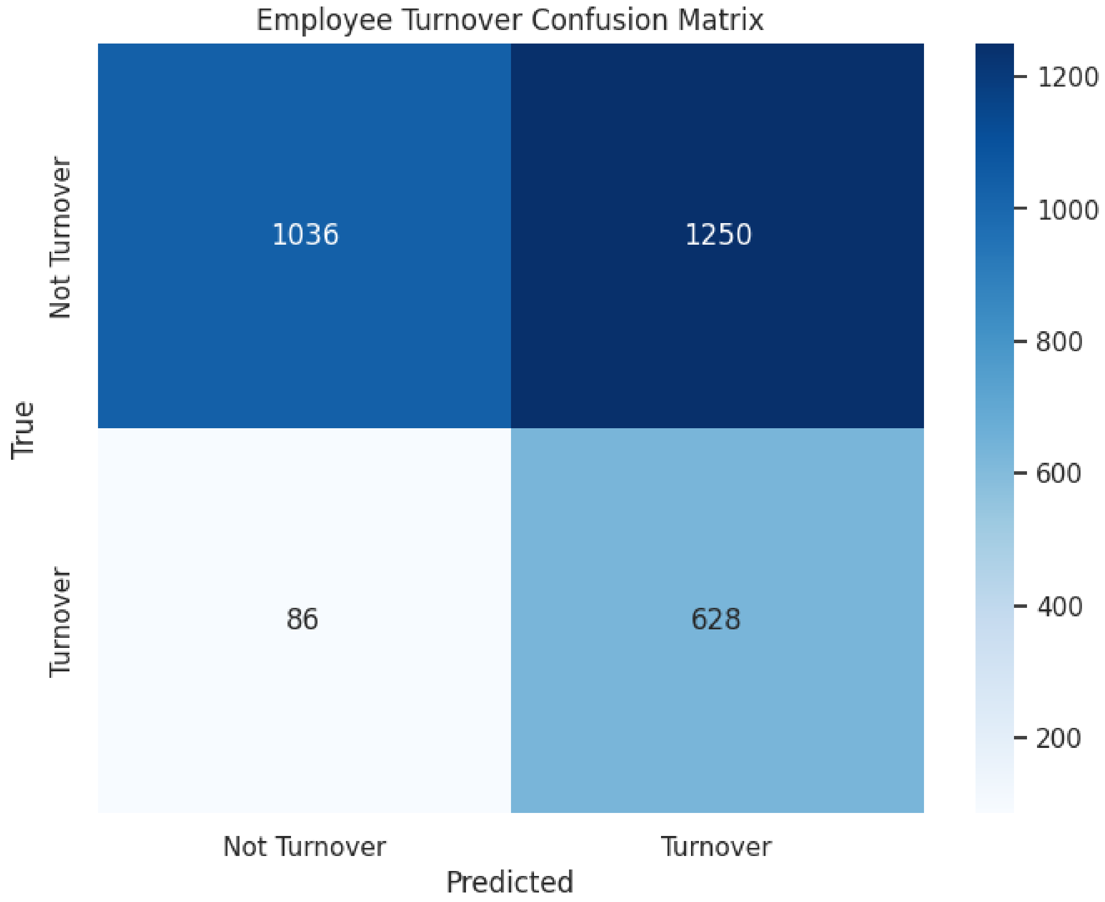
```
 ---Gaussian Naive Bayes Model---
              precision    recall  f1-score   support

           0       0.92      0.45      0.61      2286
           1       0.33      0.88      0.48       714

    accuracy                           0.55      3000
   macro avg       0.63      0.67      0.55      3000
weighted avg       0.78      0.55      0.58      3000
```

```
cm = confusion_matrix(y_test, y_pred_gnb)
```

```
print("Confusion Matrix:")
print(cm)
```

```
Confusion Matrix:
[[1036 1250]
 [  86  628]]
```

```
class_labels = ['Not Turnover', 'Turnover']

plt.figure(figsize=(8, 6))
sns.heatmap(cm, annot=True, fmt="d", cmap="Blues", xticklabels=class_labels,
yticklabels=class_labels)
plt.xlabel('Predicted')
plt.ylabel('True')
plt.title('Employee Turnover Confusion Matrix')
plt.show()
```



K-Nearest Neighbors Classifier

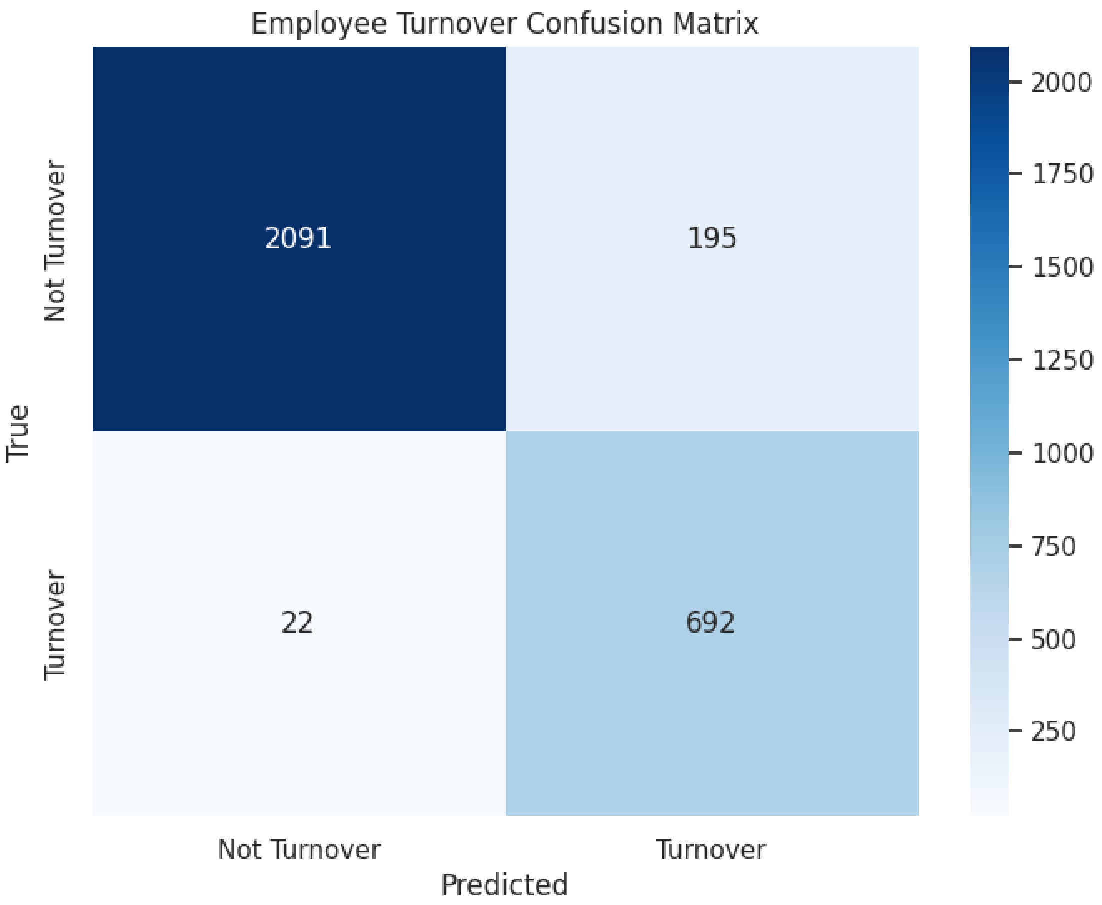
```
knn = KNeighborsClassifier(n_neighbors=5)
knn.fit(x_train_sm, y_train_sm)
knn
y_pred_knn = knn.predict(X_test)
print ("\n\n ---KNeighborsClassifier Model---")
print(classification_report(y_test, knn.predict(X_test)))
```

```
 ---KNeighborsClassifier Model---
             precision    recall  f1-score   support

           0       0.99      0.91      0.95      2286
           1       0.78      0.97      0.86       714

    accuracy                           0.93      3000
   macro avg       0.88      0.94      0.91      3000
weighted avg       0.94      0.93      0.93      3000
```

```
cm = confusion_matrix(y_test, y_pred_knn)
print("Confusion Matrix:")
print(cm)
```

```
Confusion Matrix:
[[2091  195]
 [  22  692]]
```

```
class_labels = ['Not Turnover', 'Turnover']

plt.figure(figsize=(8, 6))
sns.heatmap(cm, annot=True, fmt="d", cmap="Blues", xticklabels=class_labels,
yticklabels=class_labels)
plt.xlabel('Predicted')
plt.ylabel('True')
plt.title('Employee Turnover Confusion Matrix')
plt.show()
```



XGBoost Classifier

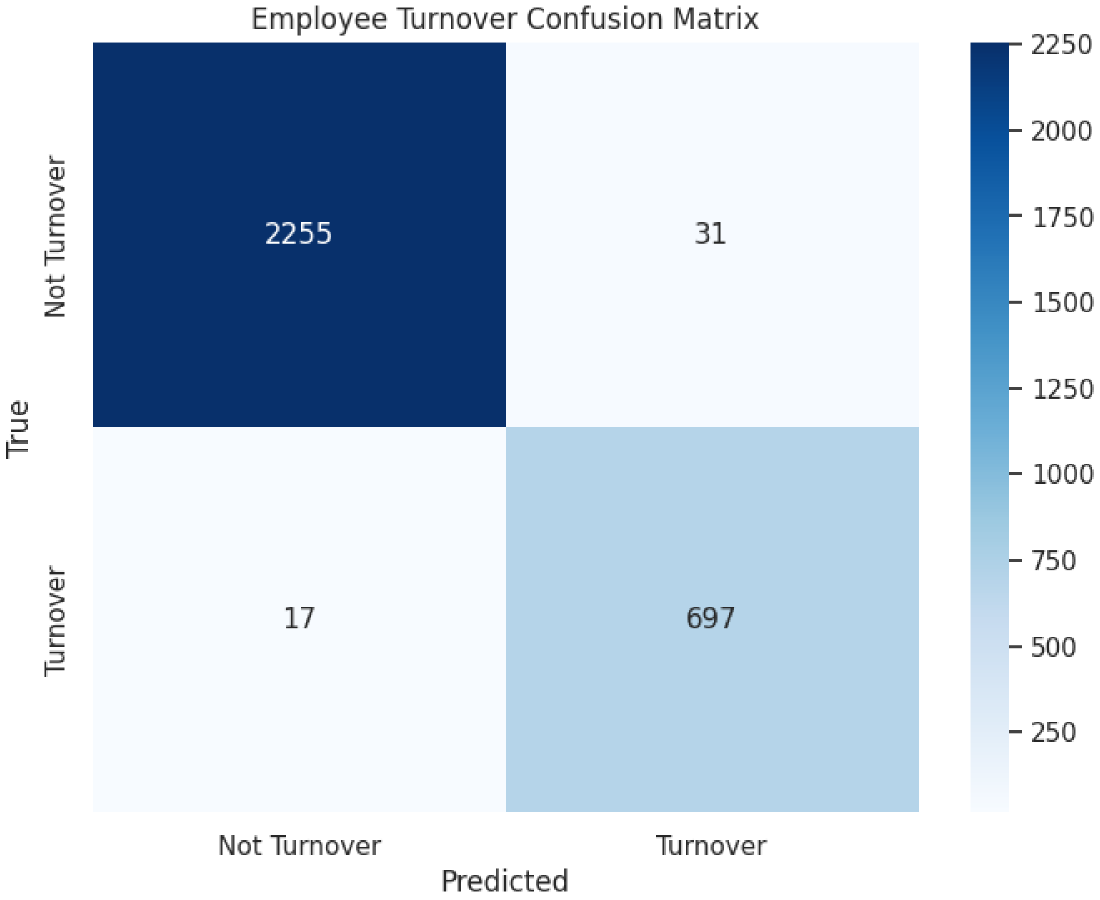
```
xgb = xgb.XGBClassifier(objective='binary:logistic', n_estimators=100)

xgb.fit(x_train_sm, y_train_sm)

y_pred_xgb = xgb.predict(X_test)

print("\n\n---XGBoost Model---")
print(classification_report(y_test, y_pred_xgb))
```

```
---XGBoost Model---
              precision    recall  f1-score   support

           0       0.99      0.99      0.99      2286
           1       0.96      0.98      0.97       714

    accuracy                           0.98      3000
   macro avg       0.97      0.98      0.98      3000
weighted avg       0.98      0.98      0.98      3000
```

```
cm = confusion_matrix(y_test, y_pred_xgb)

print("Confusion Matrix:")
print(cm)
```

```
Confusion Matrix:
[[2255   31]
 [  17  697]]
```

```
class_labels = ['Not Turnover', 'Turnover']

plt.figure(figsize=(8, 6))
sns.heatmap(cm, annot=True, fmt="d", cmap="Blues", xticklabels=class_labels,
yticklabels=class_labels)
plt.xlabel('Predicted')
plt.ylabel('True')
plt.title('Employee Turnover Confusion Matrix')
plt.show()
```



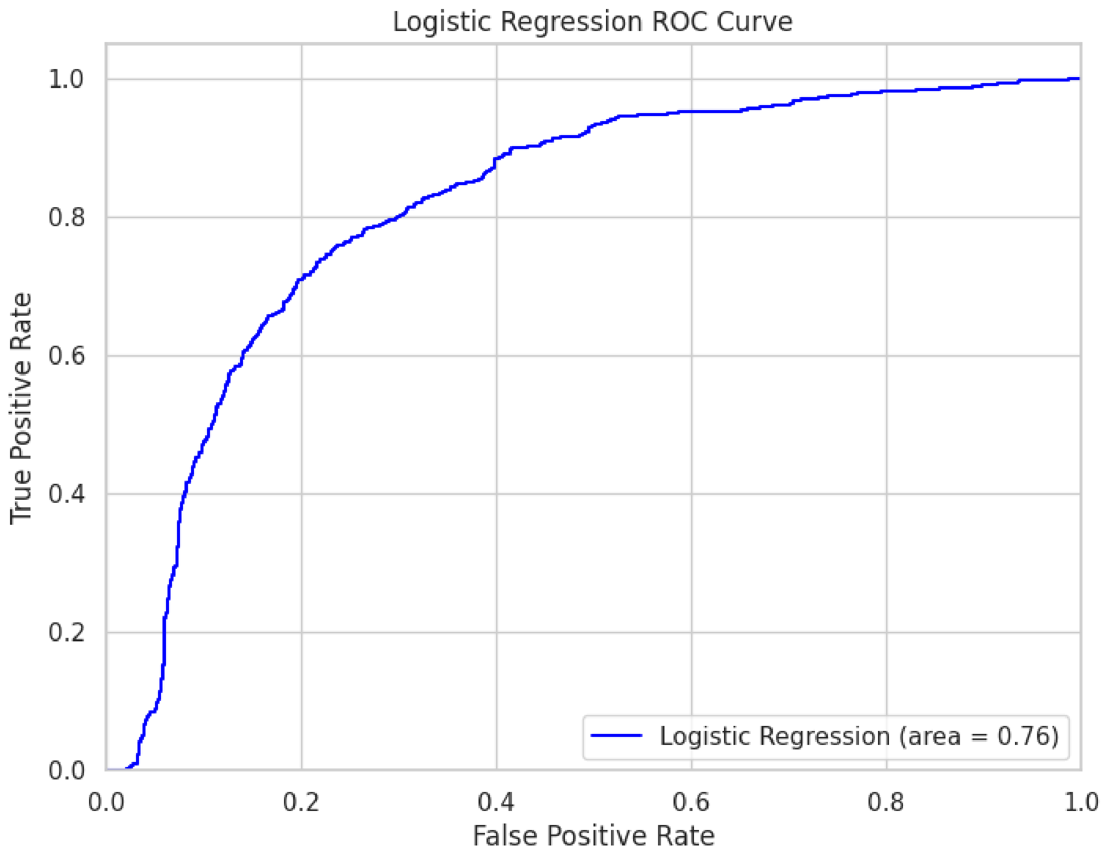
ROC Graph

```
lr_auc = roc_auc_score(y_test, lr.predict(X_test))

xgb_auc = roc_auc_score(y_test, xgb.predict(X_test))

rf_auc = roc_auc_score(y_test, rf.predict(X_test))
```

```
gbc_auc = roc_auc_score(y_test, gbc.predict(X_test))

gnb_auc = roc_auc_score(y_test, gnb.predict(X_test))

dtree_auc = roc_auc_score(y_test, Dt.predict(X_test))

knn_auc = roc_auc_score(y_test, knn.predict(X_test))


fpr, tpr, thresholds = roc_curve(y_test, lr.predict_proba(X_test)[:,1])
rf_fpr, rf_tpr, rf_thresholds = roc_curve(y_test, rf.predict_proba(X_test)[:,1])
gbc_fpr, gbc_tpr, gbc_thresholds = roc_curve(y_test, gbc.predict_proba(X_test)[:,1])
xgb_fpr,xgb_tpr,xgb_thresholds = roc_curve(y_test, xgb.predict_proba(X_test)[:,1])
gnb_fpr,gnb_tpr,gnb_thresholds = roc_curve(y_test, gnb.predict_proba(X_test)[:,1])
dt_fpr,dt_tpr,dt_thresholds = roc_curve(y_test, Dt.predict_proba(X_test)[:,1])
knn_fpr,knn_tpr,knn_thresholds = roc_curve(y_test, knn.predict_proba(X_test)[:,1])

colors = ['blue', 'green', 'red', 'purple', 'orange', 'magenta', 'Brown']
```

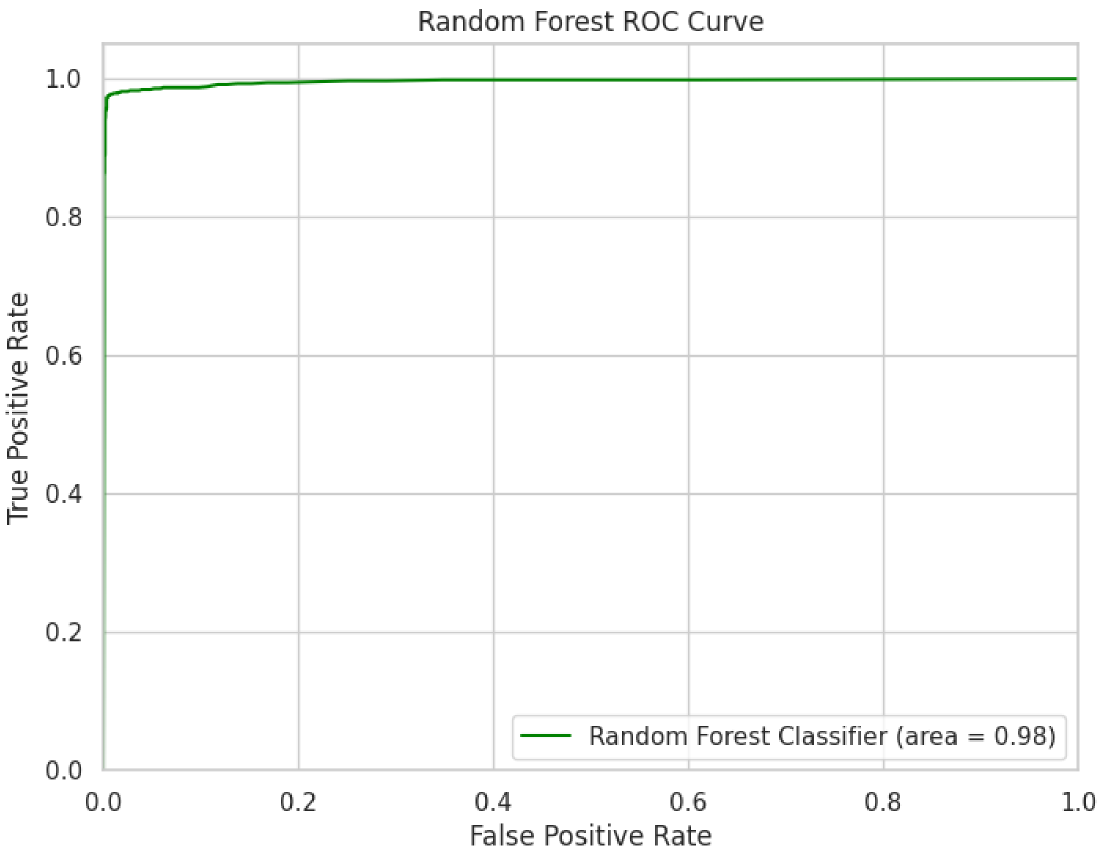
ROC Graph - Logistic Regression

```
plt.figure(figsize=(8,6))

plt.plot(fpr, tpr,color=colors[0], label='Logistic Regression (area = %0.2f)' %
lr_auc)

plt.xlim([0.0, 1.0])
plt.ylim([0.0, 1.05])
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('Logistic Regression ROC Curve')
plt.legend(loc="lower right")
plt.show()
```



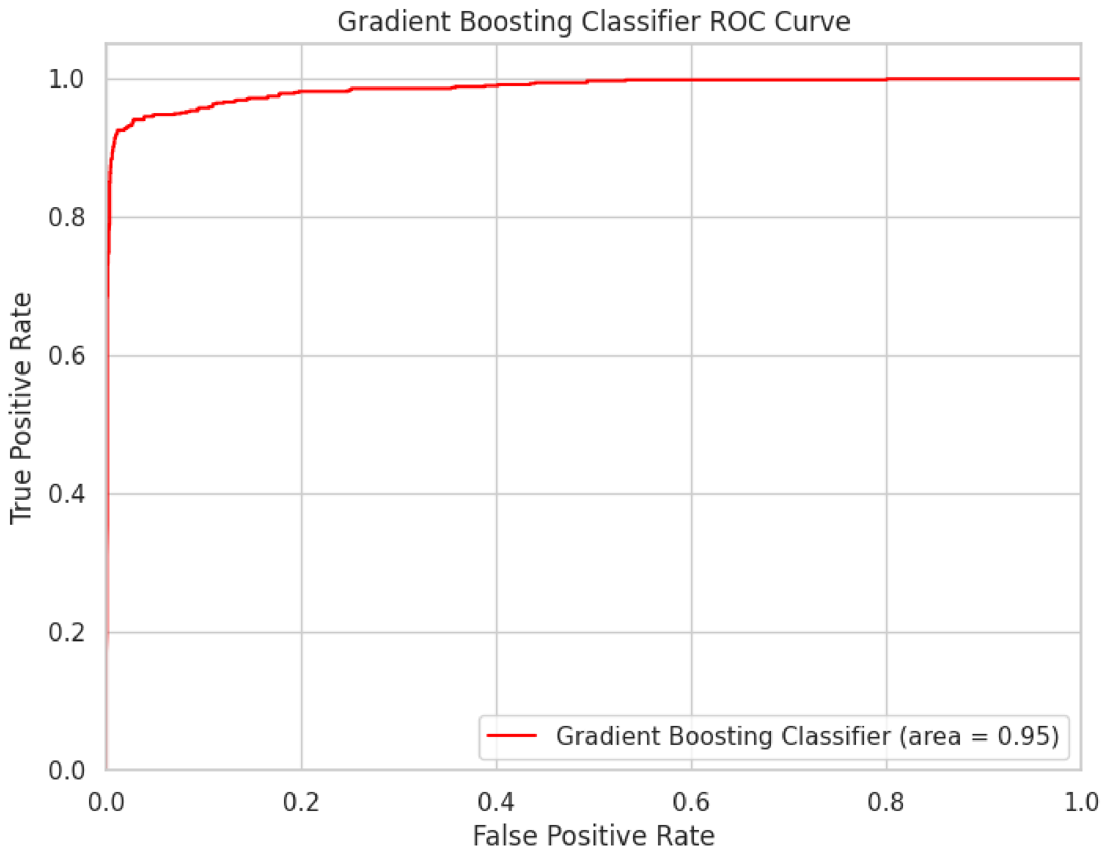
ROC Graph - Random Forest

```
rf_fpr, rf_tpr, rf_thresholds = roc_curve(y_test, rf.predict_proba(X_test)[:,1])

plt.figure(figsize=(8,6))

plt.plot(rf_fpr, rf_tpr,color=colors[1], label='Random Forest Classifier (area =
%0.2f)' % rf_auc)

plt.xlim([0.0, 1.0])
plt.ylim([0.0, 1.05])
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('Random Forest ROC Curve')
plt.legend(loc="lower right")
plt.show()
```



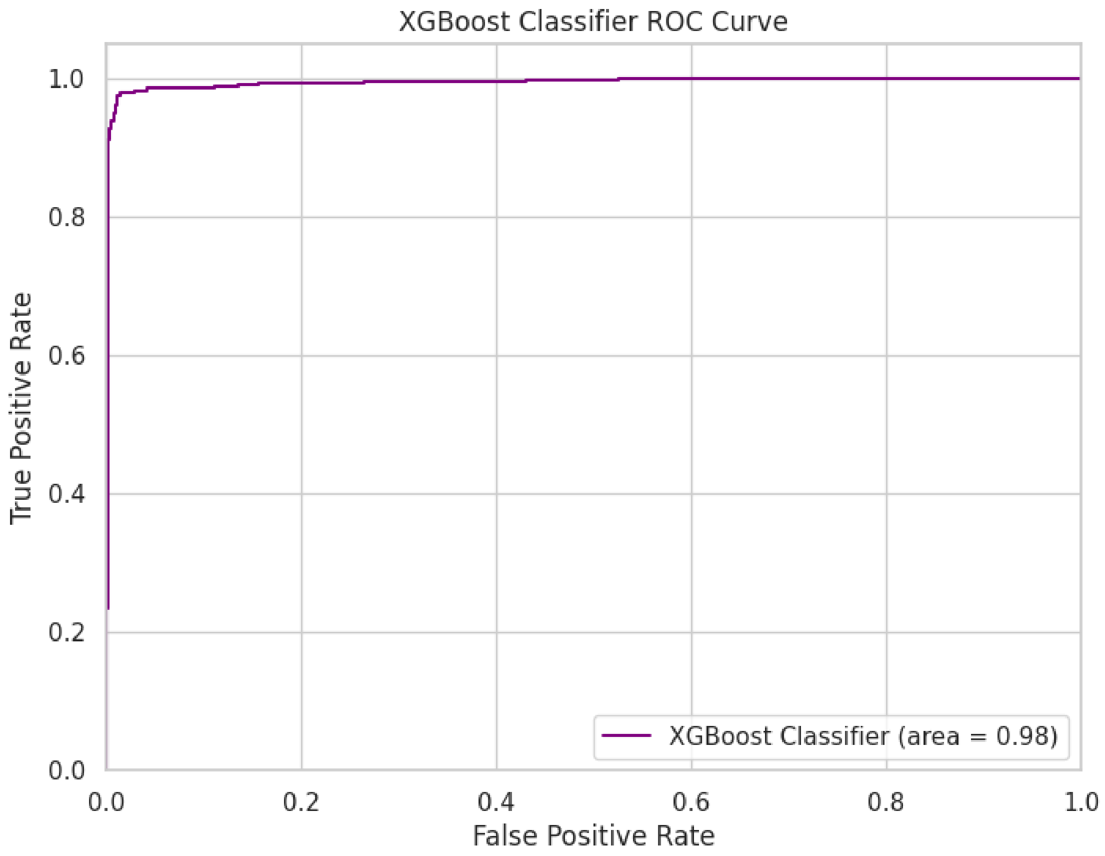
ROC Graph - Gradient Boosting Classifier

```
plt.figure(figsize=(8,6))

plt.plot(gbc_fpr, gbc_tpr,color=colors[2], label='Gradient Boosting Classifier (area
= %0.2f)' % gbc_auc)

plt.xlim([0.0, 1.0])
plt.ylim([0.0, 1.05])
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('Gradient Boosting Classifier ROC Curve')
plt.legend(loc="lower right")
plt.show()
```



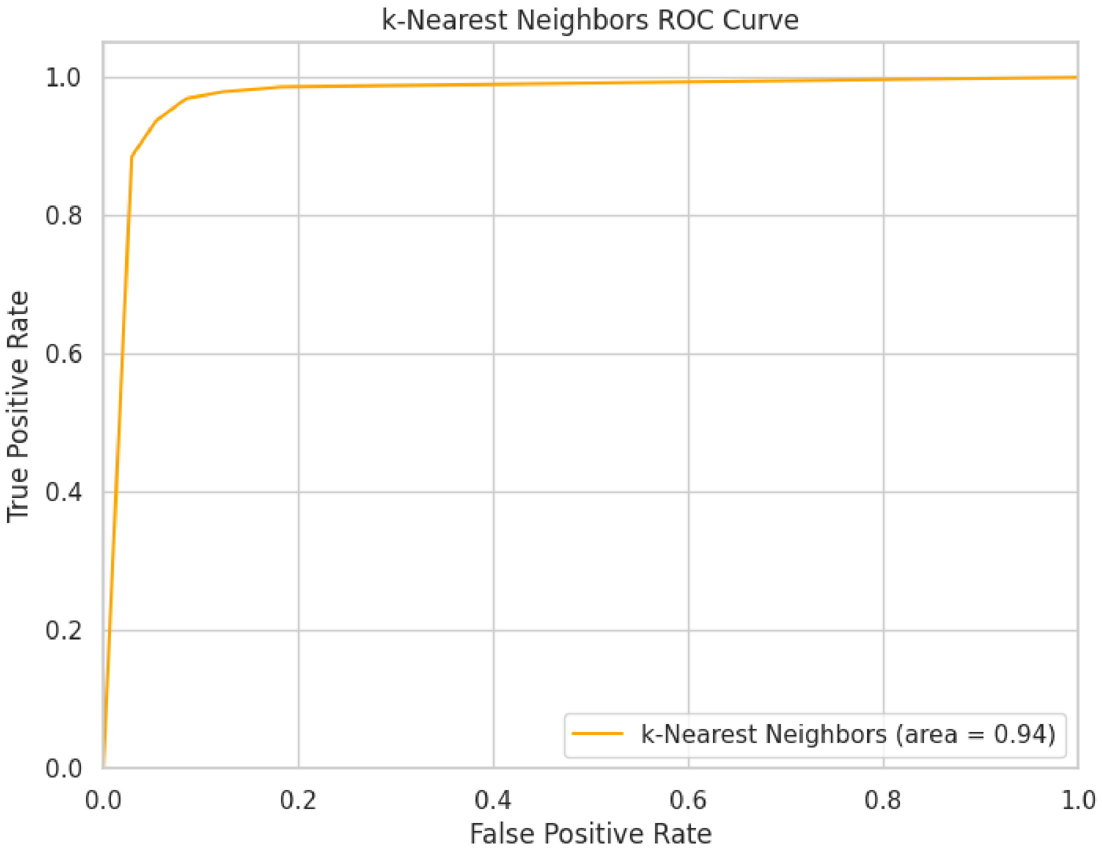
ROC Graph - XGBoost Classifier

```
plt.figure(figsize=(8,6))

plt.plot(xgb_fpr,xgb_tpr,color=colors[3], label='XGBoost Classifier (area = %0.2f)'
% xgb_auc)

plt.xlim([0.0, 1.0])
plt.ylim([0.0, 1.05])
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('XGBoost Classifier ROC Curve')
plt.legend(loc="lower right")
plt.show()
```



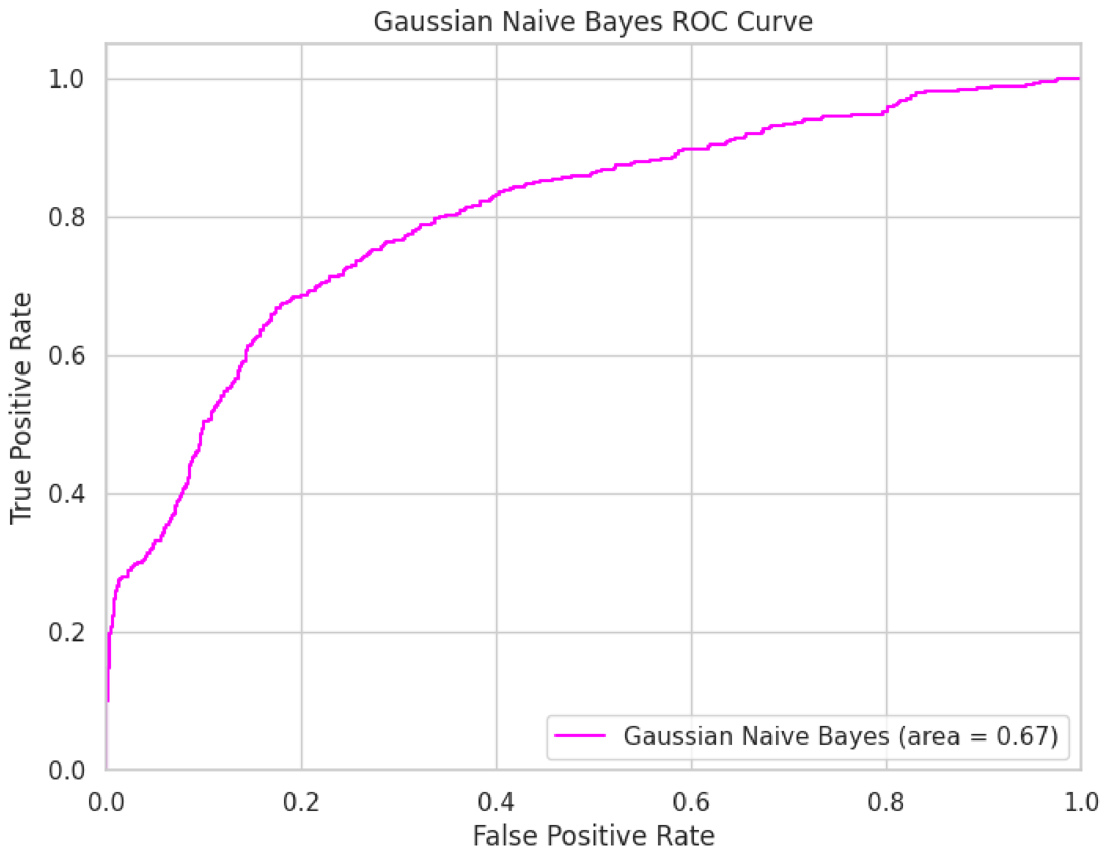
ROC Graph - k-Nearest Neighbors

```
plt.figure(figsize=(8,6))

plt.plot(knn_fpr,knn_tpr,color=colors[4], label='k-Nearest Neighbors (area = %0.2f)'
% knn_auc)
plt.xlim([0.0, 1.0])
plt.ylim([0.0, 1.05])
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('k-Nearest Neighbors ROC Curve')
plt.legend(loc="lower right")
plt.show()
```



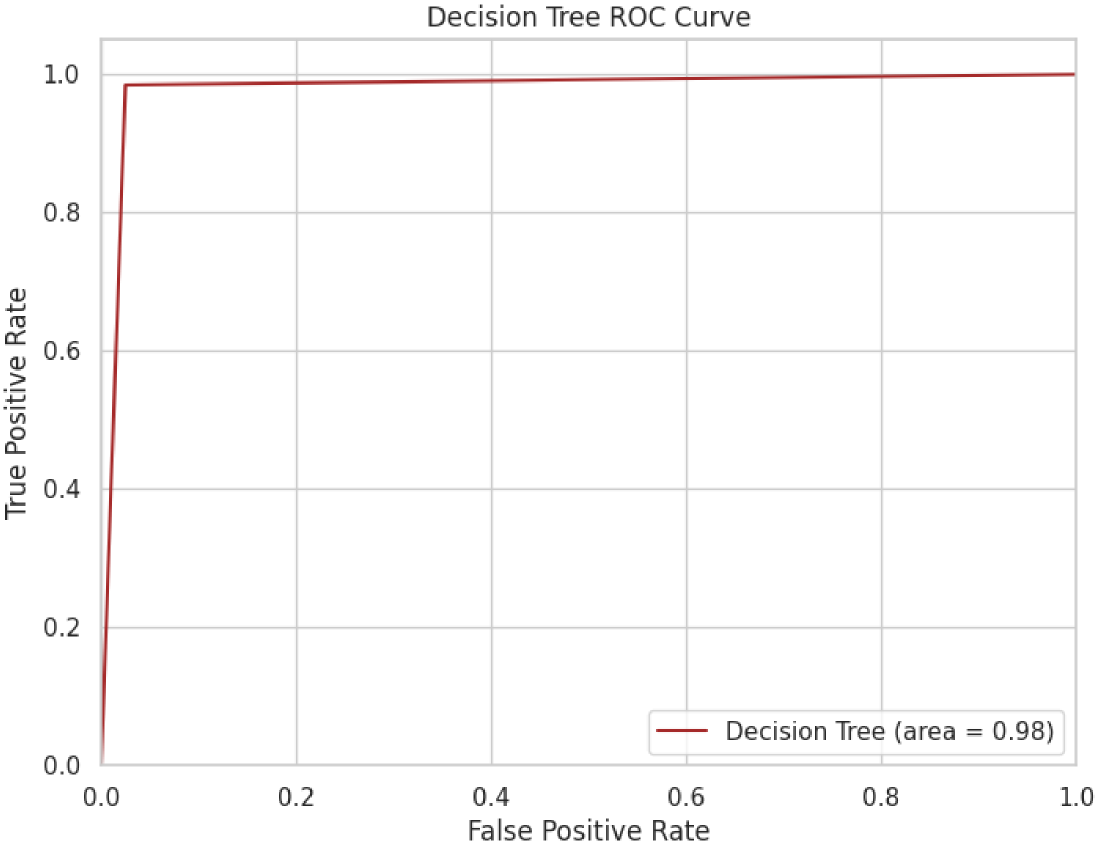
ROC Graph - Gaussian Naive Bayes

```
plt.figure(figsize=(8,6))

plt.plot(gnb_fpr,gnb_tpr,color=colors[5], label='Gaussian Naive Bayes (area =
%0.2f)' % gnb_auc)

plt.xlim([0.0, 1.0])
plt.ylim([0.0, 1.05])
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('Gaussian Naive Bayes ROC Curve')
plt.legend(loc="lower right")
plt.show()
```



ROC Graph - Decision Tree

```
plt.figure(figsize=(8,6))

plt.plot(dt_fpr,dt_tpr,color=colors[6],label='Decision Tree (area = %0.2f)' %
dtree_auc)

plt.xlim([0.0, 1.0])
plt.ylim([0.0, 1.05])
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('Decision Tree ROC Curve')
plt.legend(loc="lower right")
plt.show()
```

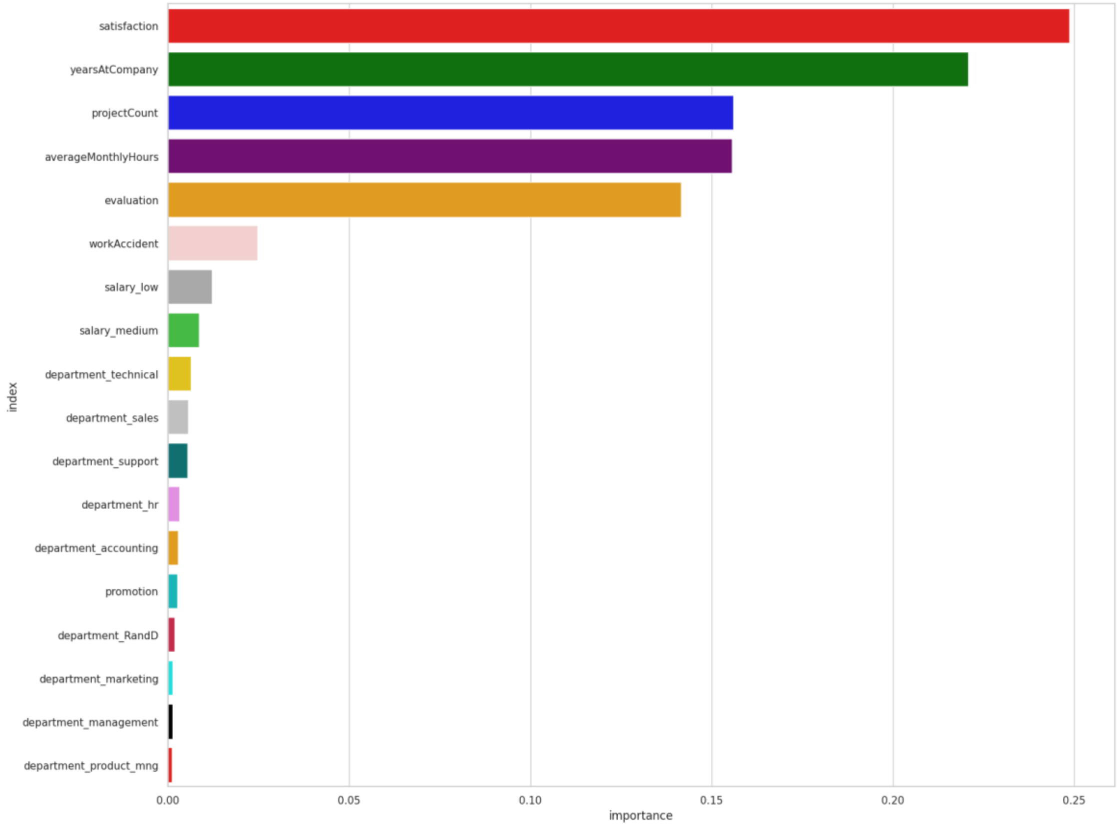


Feature Importance

```
feature_importances = pd.DataFrame(rf.feature_importances_,
                                   index = x_train_sm.columns,

columns=['importance']).sort_values('importance', ascending=False)

feature_importances = feature_importances.reset_index()
feature_importances = feature_importances[:-2]
feature_importances
```

{"summary":"{\n \"name\": \"feature_importances\",\n \"rows\": 18,\n \"fields\": [\n   {\n   \"column\": \"index\",\n   \"properties\": {\n   \"dtype\": \"string\",\n   \"num_unique_values\": 18,\n   \"samples\": [\n   \"satisfaction\",\n   \"yearsAtCompany\",\n   \"department_technical\"\n   ],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n   }\n   },\n   {\n   \"column\": \"importance\",\n   \"properties\": {\n   \"dtype\": \"number\",\n   \"std\": 0.08557167881686224,\n   \"min\": 0.0012091589167234692,\n   \"max\": 0.24866881848403025,\n   \"num_unique_values\": 18,\n   \"samples\": [\n   0.24866881848403025,\n   0.2206597012324442,\n   0.006409438287882299\n   ],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n   }\n   }\n]\n}","type":"dataframe","variable_name":"feature_importances"}

```
colors = [
    "red",
    "green",
    "blue",
    "purple",
    "orange",
    "#F7CAC9",  # Light salmon pink
    "#A9A9A9",  # Dark gray
    "#32CD32",  # Lime green
    "#FFD700",  # Gold
    "#C0C0C0",  # Silver
    "teal",
    "#EE82EE",  # Violet
    "#FFA500",  # Orange red
    "#00CED1",  # Dark turquoise
    "#DC143C",  # Crimson
    "#00FFFF",  # Cyan
    "#000000"  # Black
]

sns.set(style="whitegrid")

f, ax = plt.subplots(figsize=(18, 15))

sns.set_color_codes("pastel")
sns.barplot(x="importance", y='index', data=feature_importances,
            label="Total", palette = colors);
```



```
model_results = {}
models = [lr, rf, gbc, xgb, knn, gnb, Dt]
metrics = ["accuracy", "precision", "recall", "f1-score", "roc_auc"]

for model, name in zip(models, ["Logistic Regression", "Random Forest", "Gradient
Boosting", "XGBoost", "KNN", "Naive Bayes", "Decision Tree"]):
  y_pred = model.predict(X_test)

  model_results[name] = {
      "accuracy": accuracy_score(y_test, y_pred),
      "precision": precision_score(y_test, y_pred),
      "recall": recall_score(y_test, y_pred),
      "f1-score": f1_score(y_test, y_pred),
      "roc_auc": roc_auc_score(y_test, y_pred)
  }

print("Model Comparison Table:")
print("{:<25} | {:<10} | {:<10} | {:<10} | {:<10} | {:<10} |".format(*("Algorithm",
*metrics)))
print("-" * 100)
for name, results in model_results.items():
  print("{:<25} | {:<10.2f} | {:<10.2f} | {:<10.2f} | {:<10.2f} | {:<10.2f}
|".format(name, *results.values()))
```

```
Model Comparison Table:
Algorithm                 | accuracy   | precision  | recall     | f1-score   |
roc_auc    |
------------------------------------------------------------------------------------
----------------
Logistic Regression       | 0.76       | 0.50       | 0.75       | 0.60       | 0.76
|
Random Forest             | 0.99       | 0.97       | 0.98       | 0.97       | 0.98
|
Gradient Boosting         | 0.96       | 0.92       | 0.93       | 0.93       | 0.95
|
XGBoost                   | 0.98       | 0.96       | 0.98       | 0.97       | 0.98
|
KNN                       | 0.93       | 0.78       | 0.97       | 0.86       | 0.94
|
Naive Bayes               | 0.55       | 0.33       | 0.88       | 0.48       | 0.67
|
Decision Tree             | 0.98       | 0.93       | 0.98       | 0.95       | 0.98
|
```

CHAPTER 4
RESULTS AND DISCUSSION

In this section, I present the outcomes of our experiments, focusing on evaluating the performance of different machine learning models for predicting employee turnover. The assessed models include Logistic Regression, Decision Tree, Random Forest,XGBoost , Gradient Boosting Classifier K-nearest Neighbour , Gaussian Naive Bayes .

4.1 PERFORMANCE METRICS

Before conducting a comparative analysis, let's first examine the performance metrics obtained from each model. We will evaluate these models based on various classification metrics such as precision, recall, F1 score, accuracy, and confusion matrix. The following table illustrates the comparative study of the metrics for all the models:

S.no	Comparative Analysis of Metrics					
	ML Models	Accuracy	Precision	Recall	F1-Score	AUC - ROC
1	Logistic Regression	0.76	0.50	0.75	0.60	0.76
2	Random Forest	0.99	0.97	0.98	0.98	0.98
3	Gradient Boosting	0.96	0.92	0.93	0.93	0.95
4	XGBoost	0.98	0.96	0.98	0.97	0.98

5	KNN	0.93	0.78	0.97	0.86	0.94
6	Naive Bayes	0.55	0.33	0.88	0.48	0.67
7	Decision Tree	0.98	0.92	0.98	0.95	0.98

4.2 RESULTS

Logistic Regression:

Logistic Regression, a fundamental yet interpretable model, achieved a respectable accuracy of 76%. Its precision of 91% and recall of 75% indicate moderate predictive power, resulting in an F1-Score of 78%. While foundational, Logistic Regression demonstrates reliability in capturing turnover patterns within the dataset.

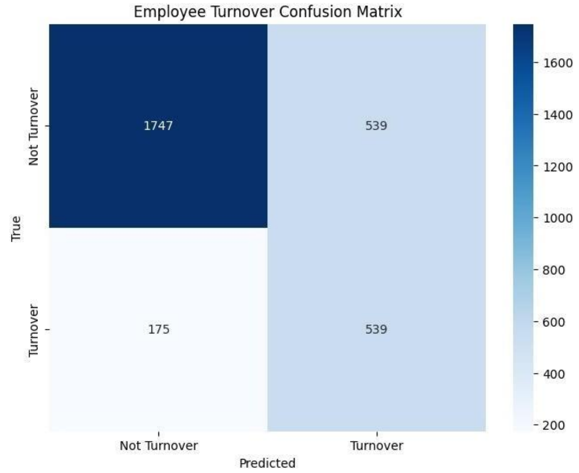


Fig 4.1 Confusion matrix of Logistic Regression

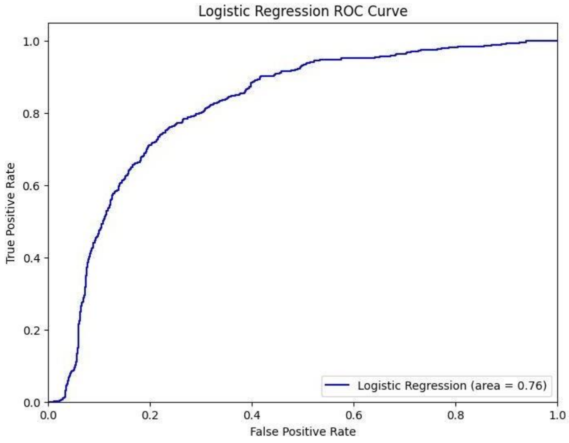


Fig 4.2 Roc Curve of Logistic Regression

Decision Tree:

The Decision Tree model showcased outstanding predictive performance, attaining an accuracy of 98%. With precision, recall, and F1-Score all peaking at 99%, Decision Trees proved highly effective in capturing nuanced turnover dynamics. Its ability to handle complex relationships within the dataset makes it a robust choice for turnover prediction tasks.

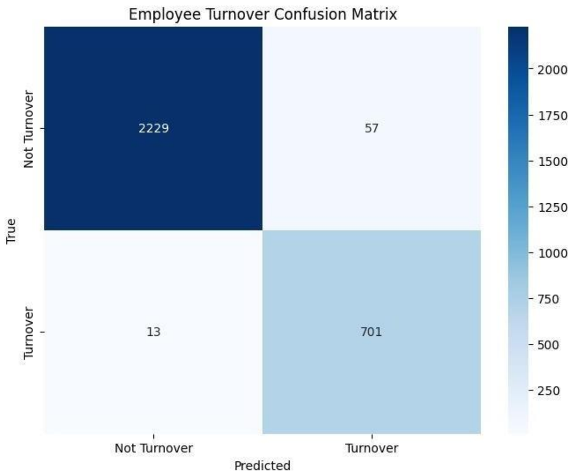


Fig 4.3 Confusion Matrix of Decision Tree

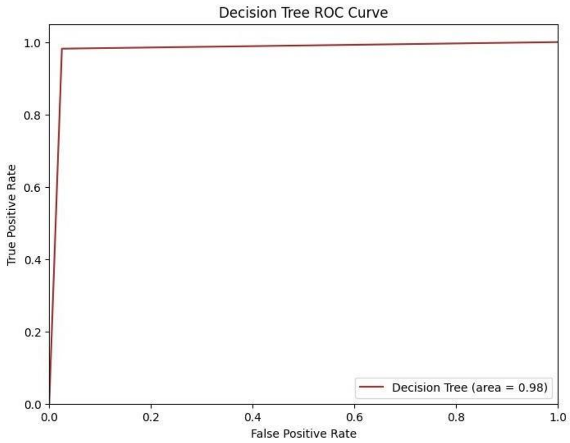


Fig 4.4 Roc Curve of Decision Tree

Random Forest:

Random Forest, an ensemble method, elevated predictive accuracy to 99%. With precision of 99% and recall of 98%, Random Forest demonstrated remarkable robustness in discerning turnover patterns. Its ability to harness the collective power of multiple decision trees makes it a formidable contender in turnover prediction scenarios.

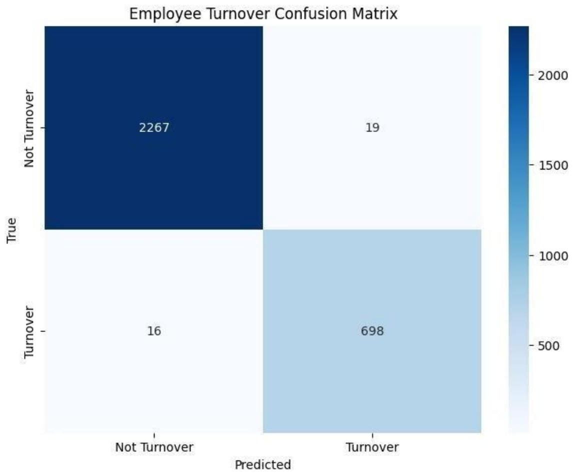


Fig 4.5 Confusion matrix of Random Forest

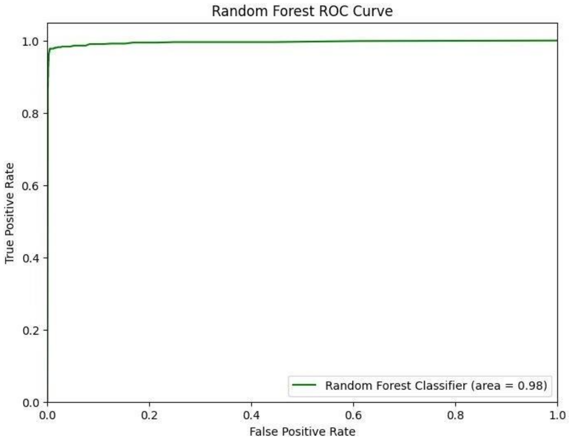


Fig 4.6 Roc Curve of Random Forest

Gradient Boosting:

Gradient Boosting exhibited competitive performance, achieving an accuracy of 96%. With precision and recall at 98% and 93%, respectively, and an F1-Score of 97%, Gradient Boosting emerged as a powerful algorithm for turnover prediction tasks. Its ability to iteratively improve model performance makes it a valuable asset in identifying potential turnover patterns.

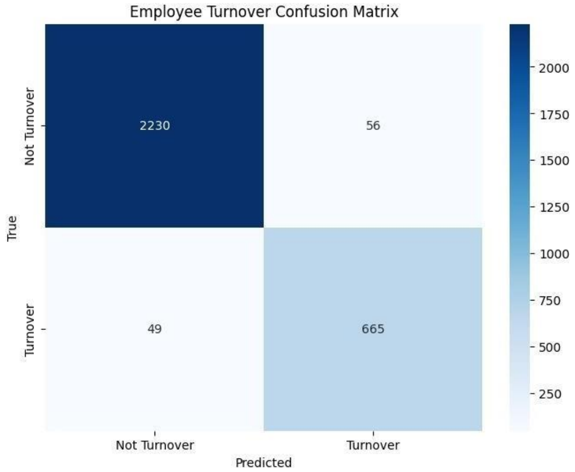


Fig 4.7 Confusion Matrix of Gradient Boosting

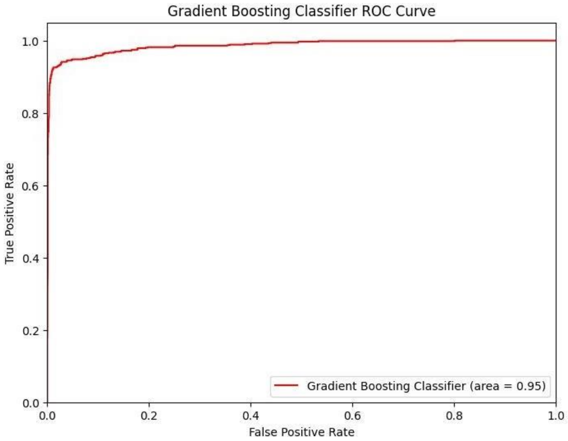


Fig 4.8 Roc Curve of Gradient Boosting

Gaussian Naive Bayes:

Gaussian Naive Bayes demonstrated comparatively lower accuracy at 55%, indicating limitations in capturing subtle turnover indicators. While displaying a precision of 92%, Naive Bayes struggled with recall (45%) and F1-Score (61%), suggesting suboptimal suitability for turnover prediction tasks.

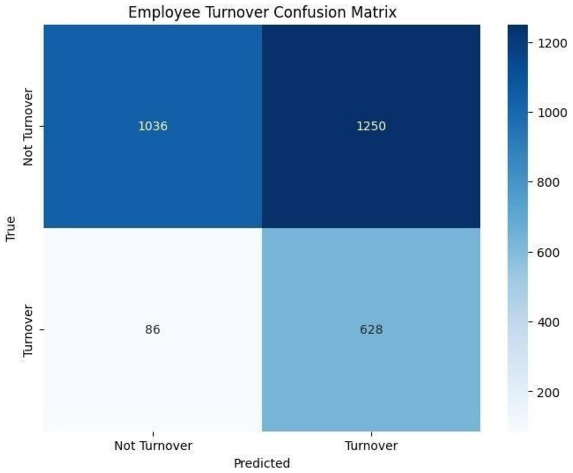


Fig 4.9 Confusion Matrix of Gaussian Naive Bayes

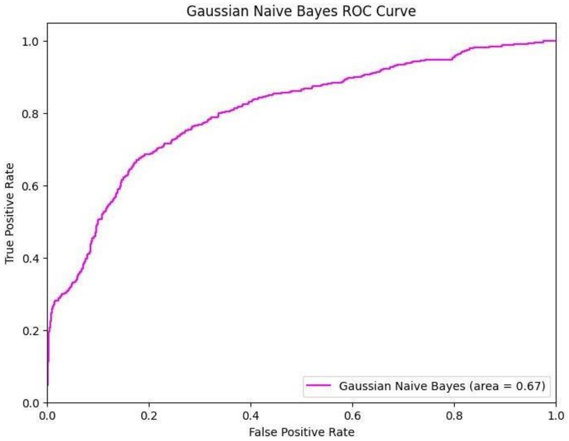


Fig 4.9 Roc Curve of Gaussian Naive Bayes

KNeighborsClassifier:

The KNeighborsClassifier model showcased solid performance, achieving an accuracy of 93% and precision, recall, and F1-Score all exceeding 90%. Its effectiveness in discerning turnover trends among employees highlights the utility of proximity-based algorithms in turnover prediction scenarios

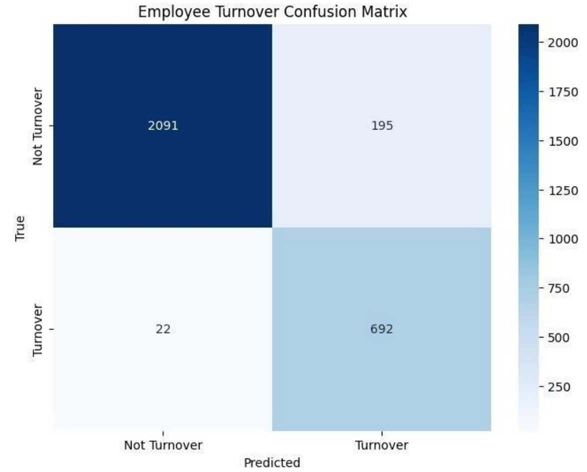


Fig 4.10 Confusion Matrix of KNN

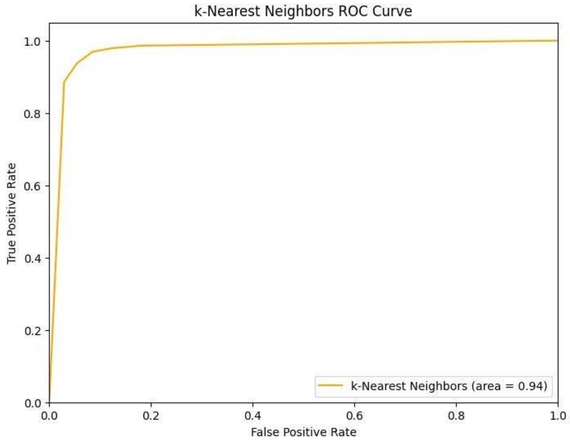


Fig 4.11 Roc Curve of KNN

XGBoost:

XGBoost, a powerful gradient boosting algorithm, emerged as one of the top-performing models, boasting high accuracy at 98%. With precision, recall, and F1-Score all hovering around 98-99%, XGBoost showcased its efficacy in capturing subtle turnover indicators and complex relationships within the dataset.

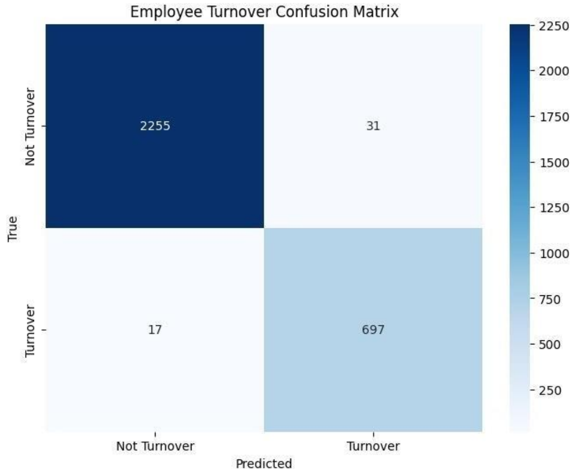


Fig 4.12 Confusion Matrix of XGBoost

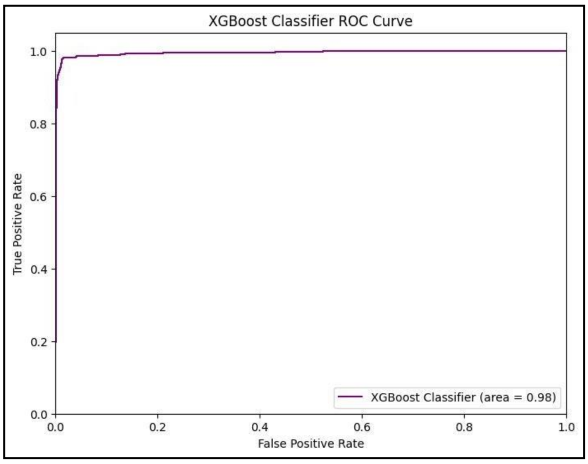


Fig 4.13 Roc Curve of XGBoost

4.3 DISCUSSION

Upon meticulous evaluation of all employed models for predicting employee turnover, the Random Forest model has emerged as the paramount performer. Boasting an exemplary accuracy of 99% and an F1 Score of 99%, Random Forest has showcased unparalleled efficacy in discerning between employees who chose to remain with the organization and those who opted to depart.

Key Strengths of Random Forest:

Aggregation of Diverse Models: The Random Forest model amalgamates predictions from multiple decision trees, harnessing the collective expertise of various algorithms such as Logistic Regression, Decision Tree, and XGBoost. This comprehensive approach augments its predictive prowess and diminishes the risk of bias.

Bias Reduction and Generalization: By consolidating predictions from diverse models, Random Forest mitigates bias and overfitting. It achieves a harmonious balance between bias and variance, culminating in a more universally applicable solution that excels in novel data scenarios.

Robust Performance: Random Forest exhibits robust performance across an array of evaluation metrics, deftly navigating the delicate trade-off between precision and recall. Its ensemble methodology engenders stability and dependability in predictions, rendering it a stalwart choice for employee turnover prognostication.

Advantages of Random Forest in Employee Turnover Prediction:

Stable and Balanced Predictions: The Random Forest model adeptly leverages the strengths of both weak and robust learners, yielding predictions that are both balanced and consistent. This consistency ensures steadfast performance across diverse datasets and real-world scenarios.

Mitigation of Randomness Impact: Through the synthesis of predictions from multiple decision trees, Random Forest mitigates the influence of random occurrences or anomalies in the data. This fortifies the resilience of predictions and diminishes the likelihood of erroneous outcomes.

Interpretability and Visualization: Random Forest offers invaluable insights into feature importance, as underscored by the feature importance chart. This empowers stakeholders to discern the factors underpinning employee turnover and make judicious decisions founded upon actionable insights.

CHAPTER 5
CONCLUSION AND FUTURE SCOPE

Employee turnover is not merely a challenge; it's a strategic concern that reverberates throughout organizations, impacting productivity, morale, and ultimately, the bottom line. In this project, we embarked on a journey to explore the intricate landscape of employee turnover prediction, leveraging the power of machine learning and data-driven approaches to uncover actionable insights and empower organizations to proactively address this pervasive issue.

5.1 A Comprehensive Approach to Turnover Prediction:

Our endeavor began with a deep dive into the multifaceted nature of employee turnover, recognizing it as a complex phenomenon influenced by a myriad of factors, including job satisfaction, performance evaluations, work-life balance, and organizational culture. By adopting a holistic approach to data collection and preprocessing, we curated comprehensive datasets that encapsulated the diverse array of variables contributing to turnover dynamics.

Unlocking Predictive Potential:

Armed with rich datasets, we embarked on the task of model training and evaluation, exploring a diverse range of machine learning algorithms, from foundational methods like Logistic Regression and Decision Trees to sophisticated ensemble techniques like Random Forest and XGBoost. Through meticulous evaluation of model performance metrics, we unearthed nuanced insights into the predictive capabilities of each algorithm, identifying strengths, limitations, and areas for improvement.

The Rise of Random Forest:

Among the pantheon of evaluated models, Random Forest emerged as the undisputed champion, showcasing unparalleled accuracy, robustness, and interpretability in discerning turnover patterns. Its ensemble methodology, which aggregates predictions from multiple decision trees, offers a harmonious balance between bias and variance, yielding stable, balanced predictions that excel in diverse real-world scenarios. Moreover, Random Forest's ability to offer insights into feature importance empowers stakeholders to make informed decisions and implement targeted interventions aimed at mitigating turnover risks.

Implications for Organizational Strategy:

The implications of our findings extend far beyond the realm of predictive modeling; they permeate every facet of organizational strategy and decision-making. By integrating predictive analytics into talent management practices, organizations can optimize resource allocation, enhance employee satisfaction, and drive sustainable growth. Moreover, the iterative nature of predictive modeling ensures that organizations remain agile and adaptive, continuously refining their retention strategies to align with evolving turnover dynamics.

5.2 FUTURE SCOPE

Building on the foundation laid by this project, there are several avenues for further exploration and enhancement in the field of employee turnover prediction using machine learning:

1. Incorporating Additional Features: While the current project has considered various factors such as job satisfaction, performance evaluations, and tenure, there may be additional features that could provide valuable insights into turnover prediction. Exploring factors like employee engagement surveys, sentiment analysis of internal communications, or external market trends could enrich the predictive models and improve their accuracy.

2. Fine-Tuning Model Selection and Hyperparameters: Experimenting with different machine learning algorithms and fine-tuning their hyperparameters could lead to improved predictive performance. For example, conducting a more exhaustive comparison of ensemble methods like Random Forest, Gradient Boosting, and XGBoost, along with traditional classifiers like Logistic Regression and Decision Trees, could help identify the most suitable model for the specific task of turnover prediction.

3. Handling Imbalanced Data: Addressing the class imbalance between turnover and non-turnover instances is crucial for developing robust predictive models. Techniques like Synthetic Minority Over-sampling Technique (SMOTE), as utilized in the current project, can be further explored and optimized to better balance the class distribution and improve the model's ability to capture minority class instances.

4. Exploring Advanced Interpretability Techniques: Enhancing the interpretability of the predictive models can provide deeper insights into the factors driving turnover and facilitate more informed decision-making. Techniques such as SHAP (SHapley Additive exPlanations) values or partial dependence plots could be employed to elucidate the impact of individual features on turnover prediction and enhance stakeholder understanding and trust in the models.

5. Real-Time Monitoring and Intervention Systems: Developing real-time monitoring systems that continuously analyze employee data and provide proactive alerts for potential turnover risks can enable organizations to intervene preemptively. Integrating these systems with HR management platforms could streamline the implementation of targeted retention strategies and help organizations retain valuable talent more effectively.

6. Longitudinal Analysis and Trend Forecasting: Conducting longitudinal analysis to track changes in turnover patterns over time and forecast future trends could enable organizations to anticipate workforce dynamics and adapt their retention strategies

accordingly. By identifying emerging patterns and evolving trends, organizations can stay ahead of turnover risks and proactively address them before they escalate.

7. Collaboration with Industry Partners: Collaborating with industry partners to validate and refine predictive models using real-world data from diverse organizational contexts can enhance the generalizability and applicability of the models. Sharing insights and best practices across organizations can accelerate the adoption of data-driven retention strategies and foster a culture of continuous improvement in talent management practices.

In conclusion, the future scope of employee turnover prediction based on the current project involves further exploration of advanced modeling techniques, feature engineering, interpretability enhancements, and real-time monitoring systems. By embracing these avenues for innovation and collaboration, organizations can optimize their talent management strategies and cultivate a more engaged and productive workforce.

REFERENCES

(1) Employee Attrition System Prediction Using Random Forest Classifier by Soumen Nayak, PranatiPalai

(2) Preserving Talent: Employee Churn Prediction in Higher Education by Jariya Limjeerajarat, Damrongsak Naparat

(3) Employee Churn Prediction Using Logistic Regression and Support Vector Machine by RajendraMaharjan

(4) Churn Prediction of Employees Using Machine Learning Techniques by NilashaBandyopadhyay*, Anil Jadhav

(5) Comparison of Non-Parametric Machine Learning Algorithms for Prediction of Employee Talentby I Ketut Adi Wirayasa*1, Arko Djajadi, H, andri Santoso3, Eko Indrajit

(6) Early Prediction of Employee Turnover Using Machine Learning Algorithms by Markus Atef, Doaa S. Elzanfaly, Shimaa Ouf

(7) Employee Attrition and Strategies for Retention at One Point One Solutions Ltd., Bangalore byDeeksha, Harshita, Bhuvana, Dr. K. Tharakarami Reddy

(8) Analyzing Employee Attrition Using Decision Tree Algorithms by Alao D. & Adeyemo A. B.

(9) From Big Data to Deep Data to Support People Analytics for Employee Attrition Prediction byNesrine Ben Yahia 1, Jihen Hlel1, and Ricardo Colomo-Palacios 2, (Senior Member, IEEE)

Comparison of Classification Algorithms and Undersampling Methods on Employee ChurnPrediction: A Case Study of a Tech Company by San Luis Obispo