Predicting the Interference of Mental Health Illness at Work Productivity in the Technology Industry using Machine Learning Methods

Samuel de Jesús Rodríguez Agudelo

Dissertation written under the supervision of Professor Pedro Afonso Fernandes

Dissertation submitted in partial fulfillment of requirements for the MSc in Business Analytics, at the Universidade Católica Portuguesa, December 2023.

Predicting the Interference of Mental Health Illness at Work Productivity in the Technology Industry using Machine Learning Methods

Samuel de Jesús Rodríguez Agudelo

Resumo

Os avanços nas tecnologias de aprendizagem automática estão a proporcionar progressivamente maiores benefícios, especialmente na realização de previsões precisas aplicáveis a diversos domínios. Um desses cenários de interesse crítico é a saúde mental e o seu impacto na produtividade no local de trabalho, particularmente na indústria tecnológica. Esta tese tem como objetivo prever a interferência no trabalho decorrente de problemas de saúde mental no sector tecnológico utilizando técnicas de aprendizagem automática.

Sete modelos de classificação de aprendizagem automática cuidadosamente seleccionados foram aplicados a um conjunto de dados provenientes da organização não governamental conhecida como Open Sourcing Mental Illness. A base de dados foi submetida a um processamento prévio, garantindo a retenção de todas as variáveis relevantes necessárias para satisfazer os requisitos de cada modelo e facilitar uma aplicação bem sucedida. Subsequentemente, os modelos foram rigorosamente avaliados utilizando várias métricas, incluindo Exatidão e Precisão, entre outras.

A investigação identificou o 'Gradient Boosting Classifier' como o modelo mais eficaz, apresentando um desempenho superior na maioria das medidas de previsão, incluindo uma precisão de 83,2%. Esta investigação também revelou limitações semelhantes às observadas em estudos anteriores de aprendizagem automática relacionados com a saúde mental, tal como referido na revisão da literatura. No entanto, os resultados contribuem com informações valiosas para a aplicação da aprendizagem automática na previsão da interferência no trabalho devido a doenças mentais, particularmente no panorama dinâmico da indústria tecnológica.

Palavras Chave: Aprendizado de Máquina, Modelos de Classificação, Gradient Boosting Classifier, Indústria Tecnológica, Saúde Mental.

Predicting the Interference of Mental Health Illness at Work Productivity in the Technology Industry using Machine Learning Methods

Samuel de Jesús Rodríguez Agudelo

Abstract

Advancements in Machine Learning technologies are progressively providing enhanced benefits, especially in the domain of making accurate predictions applicable to diverse scenarios. One such scenario of critical interest is mental health and its impact on workplace productivity, particularly within the technological industry. This thesis aims to predict work interference arising from mental health issues in the technology sector using Machine Learning techniques. Seven carefully selected Machine Learning classification models were applied to a dataset sourced from the non-governmental organization known as Open Sourcing Mental Illness. The dataset underwent strategic cleaning and preparation, ensuring the retention of all relevant variables necessary to meet the requirements of each model and facilitate successful deployment. Subsequently, the models were rigorously evaluated using various measures of prediction, including Accuracy and Precision, among others.

The research identified the 'Gradient Boosting Classifier' as the most effective model, exhibiting superior performance across the majority of prediction measures, including an 83.2% accuracy. This investigation also uncovered similar limitations to those observed in prior machine learning studies related to mental health, as discussed in the Literature Review. However, the findings contribute valuable insights into the application of Machine Learning in predicting work interference due to mental health illness, particularly within the dynamic landscape of the technology industry.

Keywords: Machine Learning, Classification Models, Gradient Boosting Classifier, Tech Industry, Mental Health.

Acknowledgements

I would like to take the time to express my gratitude for the immense opportunity to study this master's program in Lisbon, Portugal. Special thanks to my thesis supervisor, Pedro Afonso Fernandes; his guidance, patience, and support have been invaluable in presenting this thesis.

I extend my thanks to my family, who has always trusted in me. Thanks to Jannett (my mother), Jorge (my father), and Salomón (my brother). Your support, even from a distance, has been essential in this personal, academic, and professional journey. La vida es mejor con ustedes.

I also want to acknowledge and express my gratitude to my partner, Susana Ortíz, for sharing this challenging and beautiful experience with me. Great things are waiting for us.

Finally, I want to extend my thanks to all the people in Portugal, because this experience has been amazing, and I feel very thankful with the country for opening me the doors to study here. Appreciation to all the professors who have shared knowledge and wisdom with me, to Miguel Godinho, an amazing master's director, and to all the staff at the Universidade Católica Portuguesa.

Thank you.

Samuel Rodríguez Agudelo

CONTENTS

LIST OF FIGURES

LIST OF TABLES

1 INTRODUCTION

The contemporary work environment poses escalating challenges for both companies and individuals, with an increasing demand for employees to perceive their professional endeavors as contributing to their life's purpose. Simultaneously, organizations seek a workforce that demonstrates unwavering commitment to their responsibilities. However, a crucial issue that significantly impacts both spheres is the growing importance of mental health and its consequential influence on daily activities, particularly its interference in the domain of work. This research aims to investigate the application of machine learning tools in predicting the extent to which mental health disorders impede work performance within the technology industry. The study is grounded in a dataset from a non-governmental organization (NGO) dedicated to fostering awareness about the paramount significance of mental health in the context of work. This emphasis is particularly pertinent in the technology sector, characterized by its dynamic evolution and the continuous emergence of unexplored domains, which pose distinct challenges for the workforce.

Why Is Important to Study the Interference of Mental Health Illness at Work Using Machine Learning Methods?

The investigation of mental health illness interference in the workplace through machine learning methods is crucial for multifaceted reasons. Firstly, machine learning facilitates early detection and intervention by identifying patterns indicative of mental health challenges, thereby mitigating potential adverse effects on individual well-being and overall productivity. Additionally, the customization of support strategies, based on individualized risk factors discerned through machine learning algorithms, may contribute to a more inclusive and supportive work environment. The objectivity inherent in machine learning assessments ensures accurate evaluations, overcoming potential biases associated with subjective appraisals. Efficient resource allocation, tailored interventions, and the cultivation of an organizational culture prioritizing employee well-being further underscore the importance of employing machine learning in this context. Moreover, the destigmatization of mental health issues is fostered through data-driven approaches, encouraging open dialogue, and reducing reluctance to seek support. Finally, machine learning's adaptability proves especially beneficial in dynamic work environments, such as the technology sector, providing insights into emerging challenges and enabling proactive strategies to address mental health concerns.

Research Question and Hypothesis

At the core of this study lies the research question: Does the mental health of employees interfere with their work productivity in technology companies, and can this relationship be predicted using machine learning techniques? This question serves as the foundation for our exploration into the subtle interplay between mental health and work interference.
The following hypothesis is formulated to guide the research: Machine learning models can effectively predict the interference of mental health-related variables on work productivity at tech companies.
The objective is to investigate this hypothesis and illuminate the intricate connections between mental health and work productivity. Through a combination of data collection, statistical analysis, and machine learning techniques, the study aims to understand and extract valuable insights into which machine learning methods can more accurately predict the work interference of mental health illnesses at the workplace. Ultimately, the findings are anticipated to contribute to enhancing the overall well-being and productivity of employees in technology companies.

Motivation

The journey into understanding mental health within professional contexts initiates with a personal connection to this interesting subject. As the author of this dissertation, the individual has been fortunate to grow up in an environment deeply immersed in the field of mental health science. The author's father, a distinguished psychologist with over two decades of experience, has devoted his life to comprehending, advocating for, and fostering awareness about mental health in South America. His work has illuminated the relationship between mental health and work performance.
This personal connection has been intertwined with another personal passion: artificial intelligence and its role in facilitating enhanced predictions and decision-making across various domains, with a particular emphasis on the technology industry. This sector, where the inception of new technologies occurs, is distinguished by its rapid pace, high demands, and the relentless pursuit of innovation, thereby posing unique challenges for its employees, and makes it a unique space to make this study. The confluence of these two spheres has ignited the motivation behind this research endeavor.

In the subsequent sections of the thesis, a comprehensive exploration will be undertaken into the literature review, the data collection process, the methodologies employed along with their respective descriptions, the principal findings derived from the research, the limitations inherent in the analysis, and the overarching conclusions drawn from the study.

2 LITERATURE REVIEW

Mental health is an indispensable aspect of individual well-being with substantial implications for economic growth and societal development. This literature review investigates the intricate relationship between mental health and work, particularly in technology companies, employing machine learning techniques for prediction. We begin by examining the overarching importance of mental health, as established by Fit Mind, Fit Job (2015), in both personal and economic contexts.

2.1 The Concept of Mental Health

The foundational work by Fit Mind, Fit Job (2015) underscores the role of mental health as a fundamental variable in individuals' lives, asserting its tight link with well-being and quality of life. This citation emphasizes that when mental health deteriorates, it significantly impacts various life domains, including education, employability, and work performance. Notably, mild-to-moderate forms of mental ill-health affect a substantial portion, up to 20%, of the working-age population, emphasizing the economic relevance of this issue. This evidence suggests that mental health transcends the boundaries of the healthcare sector and extends its implications into broader labor market and social policies within OECD countries.

Differential experiences of mental health and their consequences are highlighted by Bubonya et al., (2017). The research underscores that women generally face a higher prevalence of internalizing problems, mental health conditions, and overall health issues compared to men. These gender disparities in mental health experiences are associated with increased absenteeism from work Vandenheuvel & Wooden (1995). These findings are vital for understanding how mental health might influence productivity in technology companies, which often grapple with gender diversity issues.

Chisholm et al., (2016)bring the economic perspective into focus by emphasizing the substantial economic losses associated with depression and anxiety disorders. These prevalent mental health conditions result not only in human suffering but also in considerable lost economic output, accentuating the profound financial implications of mental health challenges.

The interconnectedness of mental health with various other health conditions is emphasized by Prince et al., (2007)who argue that there can be no health without mental health. This interconnectedness extends to both communicable and non-communicable diseases and

contributes to unintentional and intentional injuries. Such comprehensive insight underscores the holistic nature of mental health and its impact on overall health.
Gustavsson et al., (2011) delve into a wide array of mental and neurological disorders, ranging from addictive disorders to neurologic disorders. This comprehensive categorization serves as a reminder of the diverse range of conditions that can influence work productivity in technology companies, necessitating sophisticated machine learning models for prediction.
Addressing the treatment gap in mental health services, De silva et al., (2014) highlight the importance of scaling up mental health services. Such scaling requires a robust understanding of program coverage and changes over time, which is central to effectively addressing mental health challenges in technology companies.
Patel et al., (2016) shed light on the global burden of mental, neurological, and substance use disorders, which have witnessed a significant increase over time. This substantial burden poses a direct challenge to the interference at work for people. The authors advocate for interventions that target both social causation and social drift, suggesting that alleviating poverty and providing prevention and treatment can have significant economic benefits. Intergenerational transmission of poor health and poverty, as well as the impact of severe disorders on economic performance, are noteworthy examples of the far-reaching consequences of mental health conditions.
Stigma and discrimination associated with mental illness, as highlighted by Evans-Lacko et al., (2014), can lead individuals to avoid seeking treatment and deter them from engaging in social activities. This, in turn, may exacerbate the negative impact of mental health at work in the technology industry.

2.2 Hidden Cost of Mental Health Illness for Companies

The impact of mental health on work productivity and the substantial costs it incurs for employers and society are central themes within the literature. Fit Mind, Fit Job (2015) underscores that, although most individuals with mental health problems are employed, they often struggle to perform well, resulting in significant costs for employers. Presenteeism, wherein employees attend work despite their mental ill-health, is prevalent, with Eurobarometer 2010 revealing that three in four such workers report accomplishing less than desired. This phenomenon is consistent across European OECD countries (*Fit Mind, Fit Job*, 2015) .
The high prevalence of mild to moderate mental ill health significantly contributes to these costs, affecting one-fifth of the working-age population at any given time and half of the

population over their lifetime. Chisholm et al., (2016) provide global context, estimating that depression and anxiety disorders alone result in over 12 billion days of lost productivity annually across 36 countries, costing approximately US$925 billion. The interference of these disorders at work extend to reduced labor participation, diminished tax revenue, and increased health and welfare expenditures, emphasizing the societal and economic challenges they pose (Chisholm et al., 2016).

Gustavsson et al., (2011) draw attention to the direct and indirect costs associated with mental health disorders. Their analysis reveals that indirect costs, particularly those linked to presenteeism, significantly contribute to the total economic burden. In their study, direct healthcare costs constituted 37%, direct non-medical costs accounted for 23%, and indirect costs related to patients' production losses (work interference) represented 40% of the total cost. The impact on society of depressive disorders is highlighted by Schoenbaum et al., (2002). These disorders, which are highly prevalent and leading causes of disability and reduced quality of life, significantly reduce employment rates and productivity. Annual social costs associated with affective disorders in the United States were estimated at $44 billion in 1990 (Schoenbaum et al., 2002). Nieuwenhuijsen et al., (2014) reinforce this point by revealing that depressive disorders, with a 6% 12-month prevalence in a US worker population, lead to an estimated annual human capital loss of about USD 36 billion.

Considering the information on mental health in the labor market, the Organisation for Economic Co-operation and Development has prioritized mental health as a new and critical concern. Bubonya et al., (2017) emphasize the need to redesign employment policies and practices to support the inclusion and productivity of individuals experiencing mental illness. They find that absence rates are approximately 5% higher among workers reporting poor mental health.

Presenteeism, absenteeism, direct healthcare costs, and indirect costs related to the interference of mental health illness at work contribute to the economic significance. These references underscore the urgency of addressing mental health challenges in the workplace and the potential benefits of doing so for both employers and society.

2.3 Calculation of the Annual Cost of Lost Productive Time (LPT) due to Absenteeism and Presenteeism

The annual cost of Lost Productive Time is associated with the interference of mental health

illness at work. The calculation of the cost due to absenteeism and presenteeism follows a defined process. For absenteeism, the number of missed workdays due to health problems is multiplied by 8 hours per day, with partial days weighted at 4 hours per day. The annual cost is then projected by considering total hours missed over a year, each multiplied by the individual's hourly wage. This permits a precise estimation of the economic load related with absenteeism, particularly in cases of mental health disorders like Major Depressive Disorder (MDD).
To quantify the cost of lost productive time due to presenteeism, the actual work hours are multiplied by the reduced performance level, assessed based on productivity ratings in the past four weeks, and then divided by 10. The annual cost is calculated by estimating the total hours lost due to reduced performance over a year, each multiplied by the individual's hourly wage. To assess the overall cost of lost productive time attributed to mental disorders, both the cost of absenteeism and presenteeism are combined. These calculation methods provide a broad framework for evaluating the financial impact of mental health disorders, accounting for both missed workdays and reduced productivity when employees are present (Woo et al., 2011).

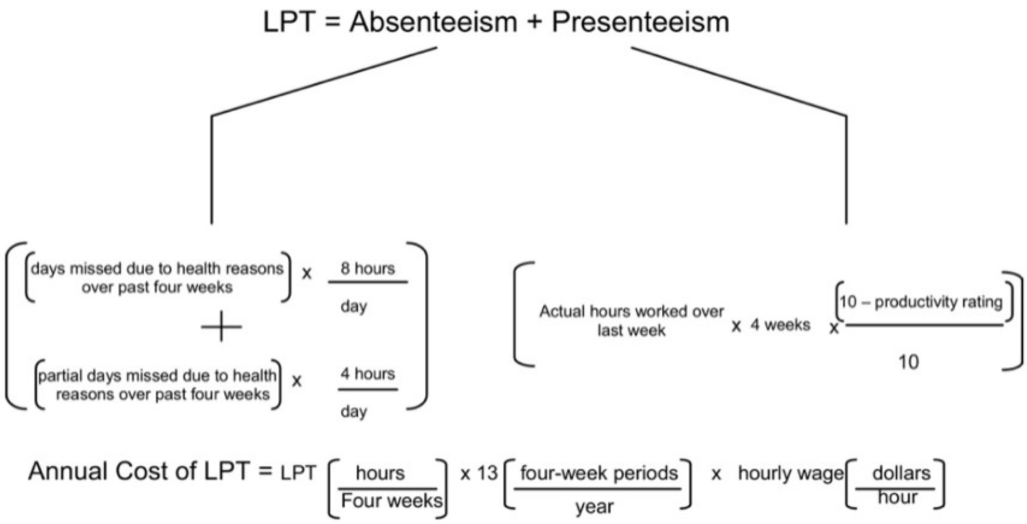


Figure 1: Calculation of the Cost of Lost Productive Time (LPT)

2.4 Machine Learning and Mental Health

Machine learning (ML) has emerged as a powerful tool in addressing mental health questions, as highlighted byGraham et al., (2019). It offers various learning approaches, including supervised, unsupervised, and deep learning, to extract valuable insights from mental health data. Supervised learning, a key category in Machine Learning, focuses on learning a function that maps data to labels, often provided by a set of training samples. In the context of mental health, these labels often relate to user self-reports. Training samples consist of data instances and their corresponding labels, comparable to a teacher supervising a learning process. Once the model learns this mapping function through training, it can predict labels for new data instances, which, in mental health applications, may involve classifying categorical values or conducting regression for continuous data (Mohr et al., 2017).

The integration of modern artificial intelligence and machine learning into mental health care is notable, as indicated by D'Alfonso, (2020). The digital transformation of mental health, facilitated by data-driven Artificial Intelligence (AI) methods, has led to the development of prediction, detection, and treatment solutions. This technological revolution has established digital mental health as a significant field, with a focus on Artificial Intelligence-driven approaches to enhance mental health care. The accessibility of abundant data streams in modern times has further fueled the development of Artificial Intelligence-driven prediction and detection models for mental health conditions, positioning Artificial Intelligence as a valuable resource for improving mental health care outcomes (D'Alfonso, 2020).

Artificial Intelligence technology, though prevalent in physical health, has been slower to gain traction in mental health due to the field's patient-centered, qualitative nature. Mental health practice, reliant on interpersonal skills and subjective patient data, can significantly benefit from Artificial Intelligence, promising to redefine mental illness diagnosis and understanding (Graham et al., 2019).

2.5 Predictive Analytics

Predictive analysis is a fundamental task in Machine Learning, crucial to numerous organizational activities. Across various industries, organizations rely on predictions for capacity planning, efficient resource allocation, and performance measurement against predefined benchmarks. Notably, the challenge of generating high-quality predictions is one that poses difficulties for both machines and human analysts.

Furthermore, Januschowski et al., (2022) highlight the potential of tree-based methods as blackbox learners in predicting analytics. This observation underscores the versatility and robustness of such methods in generating predictions. Tree-based approaches offer a data-driven means to tackle analytics challenges, and their ability to handle complex relationships and patterns within data makes them valuable tools for predictive tasks, particularly when dealing with large and complex datasets.

2.6 Limitations of AI on Mental Health Studies

The implementation of AI in mental health studies is not without limitations, as highlighted by Graham et al., (2019). One key constraint is the size and quality of available data, which can significantly impact the performance of the algorithms. Small sample sizes increase the risk of overfitting, thereby limiting the generalizability of machine learning models. Furthermore, the predictive capabilities of these models are constrained by the specific features (e.g., clinical data, demographics, biomarkers) used as inputs, which may not encompass the full spectrum of clinical factors relevant to mental health or pose ethical concerns. Additionally, the practical significance and applicability of the performance metrics generated by these algorithms are not always explicitly clarified, making it essential to critically assess their efficacy.

Another significant challenge in AI-based mental health studies relates to modeling critical events or illnesses, such as suicide ideation. These instances often result in imbalanced datasets, where the event of interest occurs infrequently or affects a small portion of the population. In such cases, classifiers tend to predict outcomes based on the majority class, potentially missing scarce events like the mentioned above. While techniques like under-sampling, over-sampling, and ensemble learning methods have been employed to address this issue, it is noteworthy that only a limited number of studies report the use of these techniques, underscoring the need for more robust strategies in managing imbalanced datasets (Graham et al., 2019).

3 DATASET

3.1 Data Source

The dataset under consideration for this study was gathered from the non-profit organization, Open Sourcing Mental Illness (OSMI), which is committed to advancing awareness, imparting education, and furnishing resources to bolster mental well-being within the technology industry and other communities.

The dataset in question is derived from a survey conducted in 2014. The respondents represent diverse geographical locations, various companies, and include both employees and self-employed individuals engaged in the technology sector. Notably, the survey targeted individuals who self-identify as suffering from mental health disorders, irrespective of whether these conditions have been formally diagnosed by medical professionals or not.

3.2 Data Understanding

This dataset consists of 1254 observations and 27 feature variables, including both independent variables and the target variable, 'work_interfere,' which indicates the extent to which an individual's mental health interferes with their work. To enhance the analysis, dummy variables were created (1 - yes, 0 – no) to represent categorical variables, ensuring a complete representation of the data. Descriptive statistics, such as mean, median, and standard deviation, were quantified to characterize the central tendency and dispersion of the features, offering a detailed understanding of the dataset’s key attributes.

The list below presents each variable with a small description of it:

- **Age**
- **anonymity:** Is your anonymity protected if you choose to take advantage of mental health or substance abuse treatment resources?
- **benefits:** Does your employer provide mental health benefits?
- **care_options:** Do you know the options for mental health as part of an employee wellness program?
- **Gender**
- **family_history:** Do you have a family history of mental illness?
- **leave:** Is it easy for you to take medical leave for a mental health condition?

- **mental_health_consequence:** Do you think that discussing a mental health issue with your employer would have negative consequences?
- **mental_health_interview:** Would you bring a mental health issue with a potential employer in an interview?
- **mental_vs_physical:** Do you feel that your employer takes mental health as seriously as physical health?
- **no_employees:** How many employees does your company or organization have?
- **obs_consequence:** Have you heard of or observed negative consequences for coworkers with mental health conditions in your workplace?
- **remote_work:** Do you work remotely (outside of an office) at least 50% of the time?
- **seek_help:** Does your employer provide resources to learn more about mental health issues and how to seek help?
- **self_employed:** Are you self-employed?
- **supervisor:** Would you be willing to discuss a mental health issue with your direct supervisor(s)?
- **treatment:** Have you sought treatment for a mental health condition?
- **wellness_program:** Has your employer ever discussed mental health as part of an employee wellness program?
- **work_interfere:** If you have a mental health condition, do you feel that it interferes with your work?

3.3 Descriptive Statistics

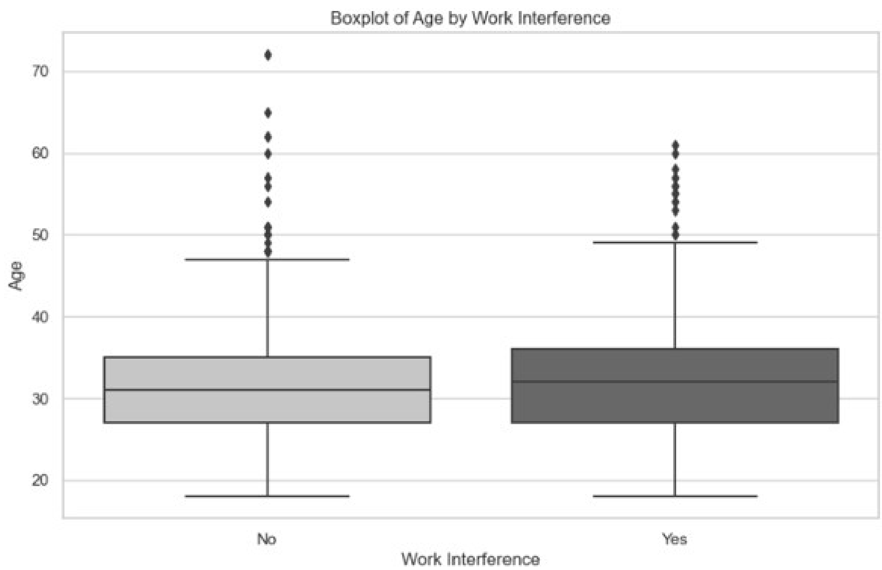
In *Table 1* are presented the descriptive statistics for the most relevant numerical variables in the study, providing valuable insights into the characteristics of the surveyed individuals. For example, the average age of participants is approximately 32 years, with a minimum age of 18 and a maximum age of 72. Regarding the 'Work Interference' (work_interfere) column, the mean value of 0.62 indicates that a significant percentage of the interviewees experience interference with their work due to mental health illness. Furthermore, the statistics for the 'treatment' variable reveal that about half of the participants have sought treatment for mental health conditions.

Table 1: Descriptive Statistics

Column	Age	work_interfere	treatment	family_history
count	1251	1251	1251	1251
mean	32.077	0.621	0.505	0.391
std	7.288	0.485	0.5	0.488
min	18	0	0	0
25%	27	0	0	0
50%	31	1	1	0
75%	36	1	1	1
max	72	1	1	1
Column	care_options	obs_consequence	benefits	remote_work
count	1251	1251	1251	1251
mean	0.952	0.145	1.053	0.297
std	0.865	0.352	0.837	0.457
min	0	0	0	0
25%	0	0	0	0
50%	1	0	1	0
75%	2	0	2	1
max	2	1	2	1

Additionally, examining the variables related to work environment and mental health, it is observed that the majority of participants have access to mental health benefits ('benefits' column), indicating a positive trend in employer-provided support. The 'care_options' column demonstrates variability in participants' knowledge about the mental health care options offered by their employers. The 'remote_work' column suggests that around 30% of individuals work remotely. And the low mean and standard deviation in the 'obs_consequence' column indicate that negative consequences for coworkers with mental health conditions are infrequently observed.

Figure 2:Boxplot of Age by Work Interference



In *Figure 2*, it is observable that the median age (early 30s) is similar for respondents who reported work interference due to mental health illness and those who did not. The first quartile values are almost the same for both groups, but the third quartile is slightly higher for those who answered ‘yes’. Additionally, both answers present outliers, having the ‘no’ group with a greater presence of them.

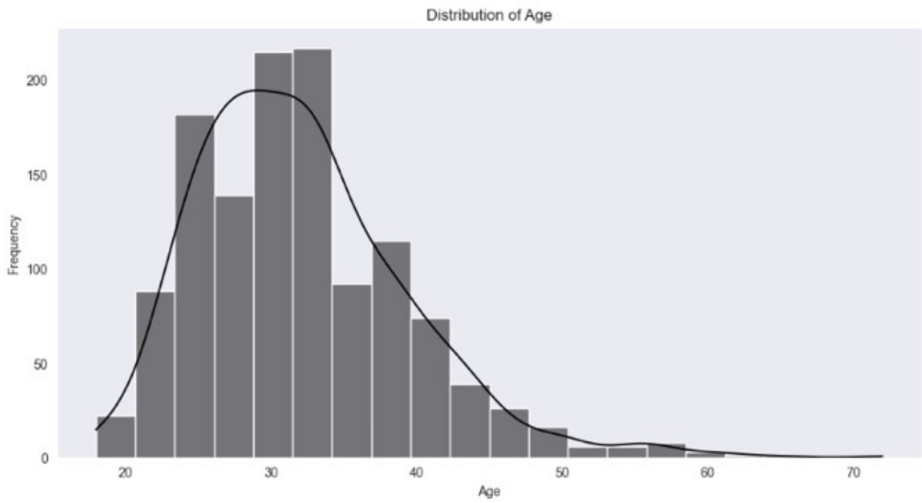


Figure 3: Distribution of Age

3.4 Data Preparation

Data preparation involved managing missing values, dropping meaningless variables, encoding categorical variables, and standardizing or normalizing numerical features. Imputation techniques, like mean or median replacement, where missing values are replaced with the most frequently occurring category within the respective variable, were employed to address missing data, ensuring the dataset's completeness. Categorical variables were encoded to numerical format, facilitating machine learning models to process them effectively. Standardization or normalization of numerical features ensured uniformity in scales.

It is important to note that correlation coefficients play a central role in the data exploration by providing a quantitative measure of both the strength and direction of linear relationships between pairs of variables.

Figure 5 serves as a valuable tool for data understanding, offering a concise summary of the degree to which variables move in tandem or in opposition.

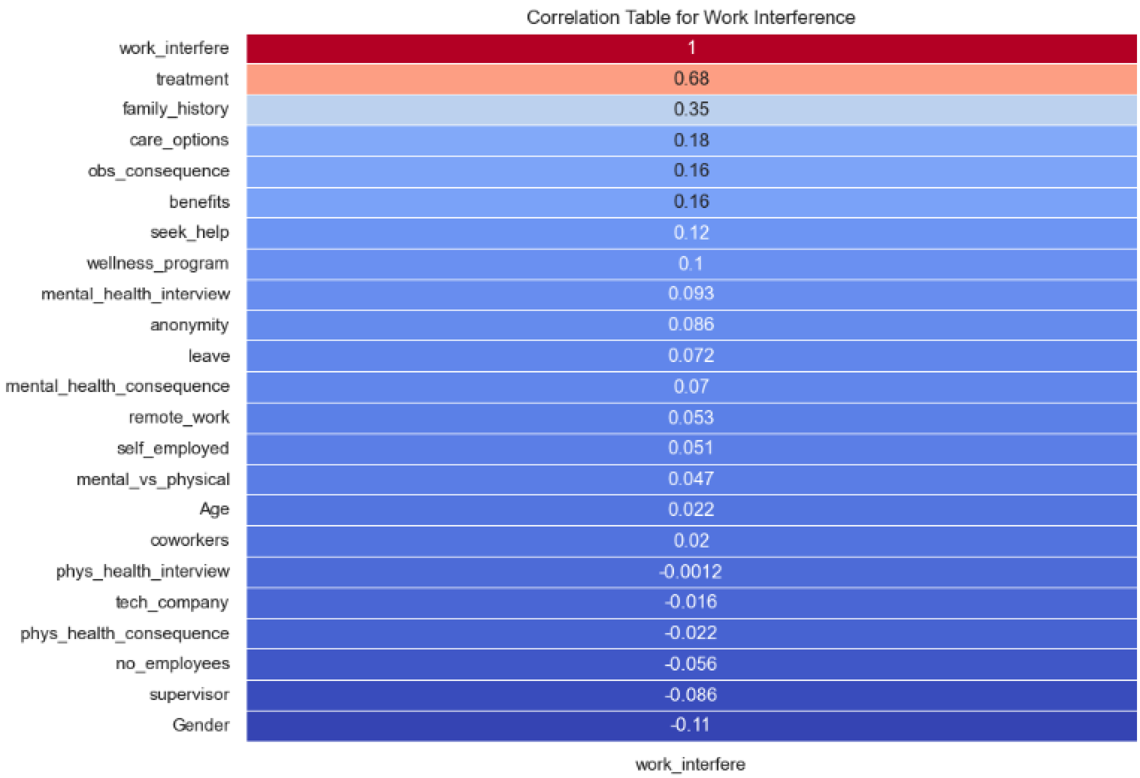


Figure 4: Correlation Table for the Response Variable 'Work Interference'

Figure 4 illustrates the correlation between the response variable (work_interfere) and the feature variables. 'Treatment' shows a correlation of 0.68 with 'Work Interference', followed by 'family_history' displaying 0.35 of correlation, implying a moderate relation between a family history of mental illness and the interference with work. The variables 'care_options', 'obs_consequence', and 'benefits' exhibit a positive correlation at 0.18, 0.16 and 0.16, respectively.

3.5 Copyright

The dataset utilized in this research is sourced from Open Sourcing Mental Illness (OSMI) and is governed by the Creative Commons Attribution-ShareAlike 4.0 International License (CC BY-SA 4.0). According to the terms of this license, researchers are granted the freedom to share, copy, and redistribute the material in any medium or format. Additionally, the license permits the adaptation, remixing, transformation, and building upon the dataset for any purpose.
However, under the CC BY-SA 4.0 license, certain conditions must be met. That is why in this research appropriate credit is given to the original data source. Importantly, no additional legal terms or technological measures may be applied to restrict others from exercising the freedoms granted by the license.

4 METHODOLOGY

In order to develop a structured methodology, a diverse set of seven machine learning models were incorporated: 'LogisticRegression,' 'KNeighborsClassifier,' 'DecisionTreeClassifier,' 'RandomForestClassifier,' 'GradientBoostingClassifier,' 'AdaBoostClassifier,' and 'XGBClassifier.' This selection aimed to explore various algorithmic approaches, spanning from non-linear models like Logistic Regression to ensemble methods such as Random Forest and Gradient Boosting (decision trees models). The goal was to thoroughly investigate their predictive performance across classification tasks, recognizing the variability in their capacities. For each model was recognized its algorithmic underpinnings and potential contributions to classification tasks. This approach provides insights into the strengths and weaknesses of each model, contributing to a precise knowledge of their predictive capabilities.

Furthermore, the methodology integrates four evaluation metrics, including traditional measures like accuracy and precision, as well as sophisticated indicators such as the area under the receiver operating characteristic curve (AUC-ROC), along with the Confusion Matrix. These selected metrics and tools form a robust framework for assessing the models' performance in detail.

Additionally, each model underwent systematic parameterization, and the dataset was prepared to align with the requirements of the models. The following section outlines the chosen machine learning models and provides explanations for each. It also delves into the evaluation metrics selected for this research, offering detailed explanations of their relevance and application.

4.1 Machine Learning Models

4.1.1 Classification

The utilization of classification methods is employed to predict qualitative responses for observations, involving the assignment of observations to specific categories or classes. These methods operate by predicting the probability of an observation belonging to each category of a qualitative variable, thereby laying the foundation for the subsequent classification (James et al., 2021). The use of classification techniques is motivated by their dual functionality, facilitating both class assignment and regression-like predictions, in the case of this study, the prediction was a classification one because the target variable ‘work interference (work_interfere)’ is binary. This approach enhances the ability to discern qualitative relationships within the dataset, contributing to a comprehensive understanding of predictive dynamics.

4.1.2 Logistic Regression

This strategy was very useful for the research as logistic regression models the probability that a dependent variable (denoted as γ) belongs to a specific category (James et al., 2021). Within the framework of binary classification, where the response variable is either 0 or 1, the logistic regression is particularly appropriate for categorizing the outcome as "yes" or "no". This binary characteristic fits excellently with the dataset used for this research (Taddy, 2019).
Through the utilization of logistic regression, this research not only makes predictions regarding the probability of specific outcomes but also enables the impact of diverse predictor variables on the likelihood of the occurrence.

$$\log \frac{P}{1-P} = \beta_0 + \beta_1 X$$

Where:

- P represents the probability of having the outcome.
- $\frac{\mathrm{P}}{1-\mathrm{P}}$ denotes the odds of the outcome.
- Increasing X by one unit results in β_1 change in the log odds.

4.1.3 K-Nearest Neighbors (KNN)

This method commences the process of model fitting and subsequent prediction, allowing streamlined prediction outputs through a single command (James et al., 2021). K-Nearest Neighbors (KNN) is a non-parametric and instance-based supervised learning approach broadly utilized in classification and regression tasks. The KNN algorithm seeks for the most frequent observation in the proximity of x for prediction purposes (Taddy, 2019). In other words, it identifies the k nearest data points to a given input x within the feature space, determining the most prevalent class within them for the prediction of x.
The parameter k represents the number of neighbors considered, influencing the model's sensitivity to local patterns. This methos is particularly beneficial in situations where the data distribution is not clearly defined.
There are numerous distance functions that play a vital role in determining results. Euclidean distance is employed for numerical attributes. Hamming distance is utilized for categorical

variables, which is more appropriate for this study. There are other alternative distance metrics available, offering flexibility and adaptability for a wide type of data and scenarios. Below the K Nearest Neighbors algorithm:

$$d\left(x_i, x_f\right) = \sqrt{\sum_{j=1}^{p}\left(x_{ij} - x_{fj}\right)^2}$$

4.1.4 Decision Tree Classifier

Decision tree serves as a logical framework for mapping inputs to outcomes. Trees are hierarchical, it means, a series of organized decisions are needed to come to a completion. The decision-making process implies a split on the dataset used and the final decisions are based on predictions conditional upon the splits. The structure of the decision tree nodes follows a parent – child hierarchy (Taddy, 2019).

This model predicts that each observation pertains to the most frequently occurring class observation within its corresponding training region. When interpreting Decision Tree Classifier results, it is important to consider both the class prediction associated with a specific terminal node region and the proportions of classes among the training observations covered by that region (James et al., 2021).

Impurity measures:

- **Classification Error**

  Represents the fraction of training observations in a specific node that do not belong to the most common class within that node:

$$E = 1 - \max_k \left(\hat{p}_{mk}\right)$$

- **Gini Index**

  A measure of node purity that quantifies the probability of misclassifying a randomly chosen observation within a node. A small Gini Index value indicates a purer node (containing predominantly observations from a single class):

$$G = \sum_{k=1}^{k} \hat{p}_{mk}\left(1 - \hat{p}_{mk}\right)$$

- **Entropy**

  Similar to the Gini Index, Entropy quantifies the uncertainty or disorder within a node:

$$D = - \sum_{k=1}^{k} \hat{p}_{mk} \log \hat{p}_{mk}$$

4.1.5 Random Forest Classifier

The use of Random Forests builds a significant progression beyond bagged trees. The introduction of randomness helps to reduce the correlation between individual trees, improving the global strength and predictive efficacy of the model.

Through the construction of each decision tree, 'a randm sample of m predictors' is chosen as split candidates from the set of 'p predictors', and only one predictor from this sample is allowed for each split (James et al., 2021). This strategic adaptation boosts not only to the diversification of the ensemble but also to increased predictive performance, rendering Random Forests a methodological improvement worthy of strict consideration within the dissertation's analytical context.

4.1.6 Extreme Gradient Boosting (XGBoostClassifier)

The incorporation of the XGBoost algorithm into the research methodology is justified by its well-documented capacity for high accuracy (Lim & Chi, 2019). XGBoost excels in mitigating the risks of overfitting, a characteristic attributed to the effectiveness of the employed boosting algorithm. The method prioritizes precision in predictions while ensuring commands related to model generalizability and resilience against overfitting. The XGBoostClassifier's adaptability to diverse datasets, efficiency in handling missing data, and ability to capture intricate patterns makes it a powerful classification tool to be included in this thesis.

4.1.7 Gradient Boosting Classifier

The main objective of Gradient Boosting is to estimate a target function by reducing the loss function, this algorithm handles large datasets excellently and, also, uses gradient descent optimization for efficiency.

To prevent overfitting, uses techniques like shrinkage, and limits the complexity of the models, using tree depth for example. Furthermore, it iteratively adjusts its models based on the pseudo-residuals calculated from the difference between the predicted and actual values. Some of the regularization parameters include learning rate, maximum tree depth, subsampling rate, the number of features considered, and the minimum number of samples required to split a node.

The parameters' importance lies in the role of preventing overfitting, improving generalization, and optimizing the performance of the algorithm when predicting (Bentéjac et al., 2021).

4.1.8 Adaptive Boosting (AdaBoostClassifier)

AdaBoostClassifier is a powerful tool for classification tasks, especially when there is a need for a robust and adaptive algorithm that can handle a variety of data patterns. By combining the predictions of multiple weak learners, AdaBoost creates a robust model that often outperforms individual weak learners, providing better generalization to new data (Zhao et al., 2016).

4.1 Confusion Matrix

The Confusion Matrix is a fundamental tool in assessing the performance of classification models. It provides a detailed breakdown of the model's predictions, distinguishing between true positive (TP), true negative (TN), false positive (FP), and false negative (FN) instances. In the context of this research, where the prediction of work interference due to mental health is of paramount importance, the Confusion Matrix aids in elucidating the model's ability to correctly classify instances and discerning potential areas of improvement.

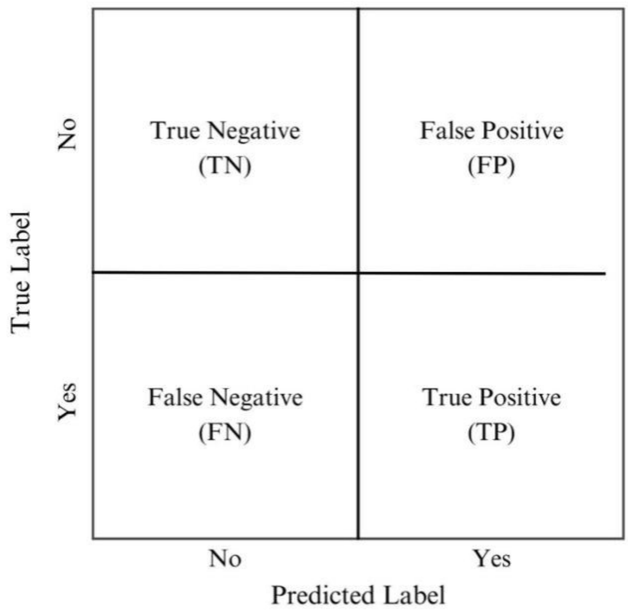


Figure 5: Confusion Matrix

4.2 Evaluation Metrics

To achieve efficiently the work interference in the work due to mental health issues, there various evaluation metrics picked for this aim. These metrics ensure a thorough and clear-eyed evaluation of how well the models are working in different aspects, giving the research a robust and reliable foundation.

4.2.1 Accuracy

This metric serves as a fundamental measure of the overall correctness of the predictive models. It is calculated as the ratio of correctly predicted instances to the total number of instances in the dataset. In the context of this thesis, accuracy provides a holistic view of the model's ability to correctly classify both positive and negative instances of work interference due to mental health issues.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

Where:

- TP is the number of true positive predictions.
- TN is the number of true negative predictions.
- FP is the number of false positive predictions.
- FN is the number of false negative predictions.

4.2.2 Recall (Sensitivity)

Recall, also known as true positive rate, quantifies the model's ability to correctly identify instances of work interference due to mental health issues among all the actual positive instances.

$$Recall = \frac{TP}{TP + FN}$$

4.2.3 Precision

This metric measures the accuracy of positive predictions made by the model and is particularly relevant in scenarios where misclassifying positive instances has significant consequences.

Precision evaluates the model's capacity to accurately identify instances of work interference due to mental health without introducing a substantial number of false positives.

$$Precision = \frac{TP}{TP + FP}$$

4.2.4 F1 Score

The F1 score is a harmonic mean of precision and recall, providing a balanced assessment of a model's overall performance. In this research, the F1 score offers a consolidated measure of the model's ability to simultaneously optimize precision and recall.

$$F_1 = \frac{2 \cdot (Precision \cdot Recall)}{Precision + Recall}$$

4.3 AUC-ROC

The Area Under the Receiver Operating Characteristic Curve (AUC-ROC) is particularly useful as it provides assessment of the model's ability to rank instances of work interference due to mental health issues.

AUC-ROC is a widely recognized metric for evaluating the discriminative ability of a binary classification model. The ROC curve plots the true positive rate (sensitivity or recall) against the false positive rate at various classification thresholds. A high AUC-ROC score signifies a model that effectively distinguishes between individuals experiencing work interference and those who are not, irrespective of the chosen classification threshold (James et al., 2021).

The combination of the Confusion Matrix and AUC-ROC enables a thorough evaluation of the predictive models, shedding light on their classification accuracy, sensitivity, specificity, and overall discriminatory power. Together, these metrics offer a robust framework for assessing the models' efficacy in predicting work interference due to mental health within the dynamic environment of tech companies.

5 FINDINGS

5.1 Overall Results

In the evaluation of the seven utilized models, six demonstrated an ‘Accuracy’ and ‘Recall’ exceeding 70%. These same six models exhibited a ‘Precision’ surpassing 80% and an ‘F1 Score’ exceeding 75%. Notably, the model with the least favorable performance across all evaluation metrics was the ‘KNeighborsClassifier’. Conversely, both the ‘LogisticRegression’ and ‘GradientBoostingClassifier’ models demonstrated superior performance among the seven models, being the last mentioned emerging as the most effective overall.

In the process of model prediction, data cleansing procedures were implemented, addressing issues such as null values, outliers, and encoding. Subsequently, the dataset was partitioned into training data (X_Train and Y_Train) and test data (X_Test and Y_Test) to facilitate the evaluation of predictions made by the models.

The meticulous assessment of model performance, considering metrics such as ‘Accuracy’, ‘Recall’, ‘Precision’, and ‘F1 Score’, revealed distinct variations in their predictive capabilities. The findings underscore the significance of careful model selection and data preprocessing steps in achieving optimal predictive outcomes within the context of this study.

The ‘GradientBoostingClassifier’ emerged as the most robust performer, showcasing superior predictive accuracy, and demonstrating its efficacy in addressing the specified research objectives as shown in the table below:

Table 2: Models' Performance

index	Model	Accuracy	Precision	Recall	F1 Score
0	LogisticRegression	0.824	0.916	0.791	0.849
1	KNeighborsClassifier	0.58	0.784	0.449	0.571
2	DecisionTreeClassifier	0.742	0.831	0.735	0.78
3	RandomForestClassifier	0.814	0.894	0.795	0.842
4	GradientBoostingClassifier	0.832	0.925	0.795	0.855
5	AdaBoostClassifier	0.816	0.891	0.803	0.845
6	XGBClassifier	0.814	0.88	0.812	0.844

5.2 Gradient Boosting Classifier Results

The Gradient Boosting Classifier showcased a notable performance in both training and test data. On the training set, it achieves an Accuracy of 89.1%, meaning it gets it right most of the time when making the predictions. When it predicts a positive outcome (Precision), it proves to be highly accurate at 93.4%. Moreover, it shows a great ability to capture the actual positive instances, scoring an 88.8% on Recall. The balance between Precision and Recall is reflected in the F1 Score with a solid 91.0% again on the training data.

Model	Training Accuracy	Training Precision	Training Recall	Training F1 Score	Test Accuracy	Test Precision	Test Recall	Test F1 Score
GradientBoostingClassifier	0.891	0.934	0.888	0.91	0.832	0.925	0.795	0.855

Table 3: Gradient Boosting Classifier - Train & Test Results

Moving beyond the training set, this technique maintains a consistence performance on the test data, achieving an Accuracy of 83.2% and a 92.5% on Precision, being this evaluation metric the most consistent within the two datasets for the Gradient Boosting Classifier technique. The model gets a Recall of 79.5%, indicating its effectiveness in identifying most of the actual positive cases, and the F1 Score on the test set remains strong at 85.5%, showcasing the model's consistency between minimizing false positives and false negatives.

To conduct a more detailed analysis of the results derived from the 'GradientBoostingClassifier' algorithm, it is presented both the Confusion Matrix and the Area Under the Receiver Operating Characteristic Curve (AUC-ROC) for a better understanding.

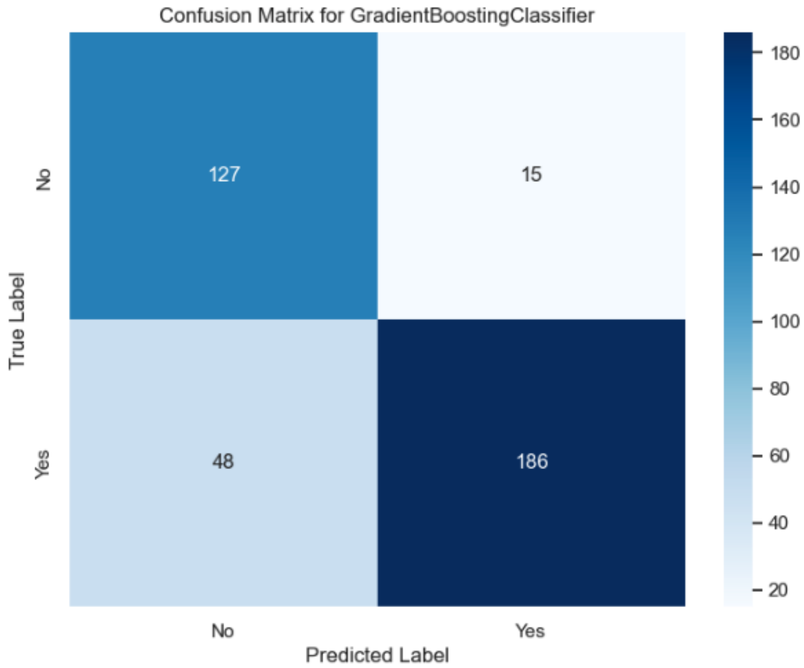


Figure 6: Confusion Matrix for 'GradientBoostingClassifier'

The Confusion Matrix serves to clarify the distribution of predicted and actual outcomes, while the AUC-ROC curve offers an assessment of the model's capability to distinguish between different classes.

After analyzing the *Figure 6: Confusion Matrix for 'GradientBoostingClassifier'*, it's visible that the model demonstrated proficiency in correctly identifying instances that do not belong to the predicted class, achieving a count of 127 true negatives (TN). Furthermore, it demonstrated accuracy in identifying instances that are part of the predicted work interference, with 186 true positives (TP). However, the model did misclassify 15 instances as positive when they were actually negative, represented as false positives (FP), and 48 instances as negative when they were actually positive, called as false negatives (FN).

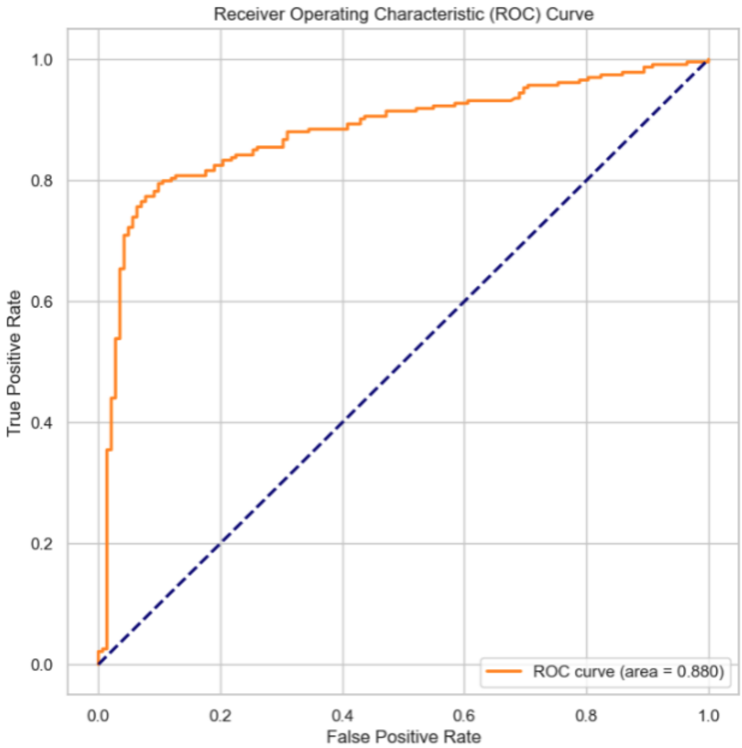


Figure 7: AUC-ROC for 'GradientBoostingClassifier'

Figure 7 holds significant implications, particularly in the context of predicting the interference of diverse mental health variables at work within the tech industry, which is the objective of this research. The AUC-ROC serves as a very important metric, reflecting the model's ability to discriminate between instances where there is interference due to mental health issues and instances where there is not.

An AUC-ROC value of 0.880 indicates a strong discriminatory performance of the ‘GradientBoostingClassifier’, asserting its proficiency in distinguishing between the complexities of mental health variables in a workplace setting.

The model's effectiveness in differentiating between classes aligns with the primary goal of predicting interference. It implies that the ‘GradientBoostingClassifier’ can reliably identify instances where mental health variables apply an influence on work dynamics and where they do not.

Feature	Importance
treatment	0.458827
Age	0.084526
leave	0.062043
no_employees	0.03672
care_options	0.035148
wellness_program	0.034446
supervisor	0.034412
mental_health_consequence	0.030741
remote_work	0.030733
seek_help	0.025584
phys_health_interview	0.024971
mental_vs_physical	0.024842
coworkers	0.017494
family_history	0.016134
benefits	0.015035
self_employed	0.014551
anonymity	0.012776

Table 4: Features Importance in Decision Tree

After reviewing and analyzing the findings, it is also relevant to address the factors influencing the work interference due to mental health illness in technology companies. An insightful interpretation of the decision tree model's feature importance could help to enhance the understanding. In *Table 4,* shows clearly that the most influential factor in predicting work interference in mental health is whether an individual is undergoing treatment (importance: 0.459) with a huge gap over the other features. This highlights the significant impact of treatment treatment on the likelihood of work interference. Additionally, age, leave policies, and the size of the company (no_employees) also contribute to the model's decision-making process, indicating that these factors play identifiable roles in predicting work interference.

6 DISCUSSION

To enrich and provide a broader and more comprehensive understanding of the results obtained in this dissertation, this section will discuss the efficacy of Machine Learning methods in predicting Mental Health Illness events. With a significant focus, it will also compare the performance of ‘Gradient Boosting Classifier’ against other methods and algorithms. The findings obtained in other similar studies will serve as a point of reference to contrast the outcomes got in this research.

To begin, the ‘Boosting’ techniques are gaining popularity for their interesting proposals and other characteristics previously mentioned in the methodology section. In this research, they were the models that better classified and correctly identified instances of work interference due to mental health illness (Recall), achieving a percentage greater than 79% by all the employed ‘Boosting’ techniques ('AdaBoostClassifier', 'Gradient Boosting Classifier', and 'XGBoostClassifier'). Moreover, the 'Gradient Boosting Classifier' was the model that achieved better Precision among all seven models considered in this thesis with a 92,5%.

In comparison with the research presented by Bentéjac et al., (2021), titled ‘A comparative Analysis of Gradient Boosting Algorithms’, where it was studied the ‘family’ of gradient boosting techniques with a big focus on speed and accuracy. Their empirical analysis showed that ‘CatBoost’ was, on average, the most accurate classifier. However, the difference with other gradient boosting techniques, like ‘Gradient Boosting Classifier’ and ‘XGBoost’, was not found to be significant. It is important to mention that in this research, techniques as ‘CatBoost’ and ‘LightGBM’ were not considered. Therefore, it could be a mistake to claim that the performance of the ‘Gradient Boosting Classifier’ technique is the best in its ‘family’ for its performance in this research, as all these types of techniques were not considered in this study.

However, how did these ‘Boosting’ methods perform compared to other techniques when predicting and identifying mental illness? As presented by Graham et al., (2019) in the study 'Artificial Intelligence for Mental Health and Mental Illnesses'. Supervised Machine Learning (SML) techniques were the most common method of Artificial Intelligence within their work (23 out of 28 studies). The data analyzed with these methods came from survey data, demographics, and social media posts, aligning with this study where survey data was the source of the information. Regarding the results, the accuracy in their studies ranged from the low 60s (62%) and scale up until high 90s (98%) using Supervised Machine Learning methods, showing heterogeneity in their findings. The wide range in their results was mainly explained by the type of data they handled for making the predictions. In their work, they achieved therir lowest accuracy (63%) when predicting from social media posts but their highest (98%) when predicting from sociodemographic. It is challenging to determine if their results are closely resembled or significantly differ from those obtained in this research. Nevertheless, it is important to highlight that the findings got in this investigation using Supervised Machine Learning techniques align with the range showed in the referenced study.

Additionally, according to the paper 'Machine Learning in Mental Health: A Scoping Review of Methods and Applications' (Shatte et al., 2019), published by Cambridge University Press, the primary applications of machine learning in mental health related topics were: (1) detection and diagnosis, (2) prognosis, (3) public health, and (4) research and clinical administration. These applications were identified from the analysis of three hundred papers made in that study. The main application of machine learning methods in this research, which it was to predict the interference of mental health illness at work in the tech industry, represents a marked difference with the mentioned study. While this thesis might be related to the prognosis application, the main emphasis in this study is on mental health illness and work, an aspect not explicitly highlighted in the previously mentioned applications.

To close this part, numerous limitations presented in the work by Graham et al., (2019) are similar to the ones found in this research. Noted by Graham, the main limitations when incorporating artificial intelligence and mental health studies are the size of the samples, because when the sample is small, it usually represents a challenge for the study as overfitting is very likely and it limits the generalizability of the results. Also, it mentions that the predicted ability of the studies is usually restricted by the features used for Machine Learning methods, another similarity found with this research. And that to predict 'rare events' which is what it has been done in this study, it is very challenging due to the imbalance of the datasets.

7 CONCLUSIONS

7.1 Main Insights

In the prediction of Work Interference due to Mental Health Illness in tech companies, seven distinct machine learning classification models and their predictive capabilities were implemented. Essential metrics such as 'Accuracy,' 'Recall,' 'Precision,' and 'F1 Score' were implemented as well. And the best predicting technique among all these models was the 'GradientBoostingClassifier,' showcasing superior accuracy, precision, and recall. In parallel, ‘Boosting’ techniques and the 'LogisticRegression' model demonstrated remarkable performance.

The careful preparation of the data, addressing issues like null values and encoding, had a significant impact on the success of the predictive models. By carefully dividing the dataset into training and test sets, the evaluation of each model's predictive capabilities was possible. This emphasizes the close connection between thorough data preparation and the best possible performance of the models, highlighting the significance relationship between data quality and predictive outcomes.

The 'GradientBoostingClassifier' was the technique with a bigger emphasis due to its predictions, a close look was done using tools like the Confusion Matrix and the Area Under the Receiver Operating Characteristic Curve (AUC-ROC). The Confusion Matrix provided insights into the model's proficiency in correctly identifying instances of both positive and negative classes. While excelling in true negatives and true positives, the model exhibited constraints with false positives and false negatives. This analysis provides a deeper understanding of the model's strengths and potential areas for enhancement, giving a richer comprehension of its predictive performance in the realm of work interference due to mental health illness in technology companies.

7.2 Limitations and Future Research

Examining the limitations of this work, few factors have come to the forefront. One key challenge is the quality of the data, with a specific focus on structural validity. The presence of potential inaccuracies or inconsistencies within the dataset introduces uncertainties that might have impacted the overall findings and interpretations. Another notable limitation involves the need for extensive datasets, as the outcomes and model performance may be sensitive to dataset size. The emphasis on binary classification, while providing clarity, oversimplifies the complexities inherent in the studied topic, Mental Health Illness and their interference at work in technology industries is a complex topic and this binary classification could represent bias to the study. Imbalances in dataset distribution present challenges, influencing the model's predictive accuracy for less prevalent outcomes. Additionally, the reliance on selected features and potential data biases underscores the necessity of addressing these aspects to enhance model reliability. Looking ahead, future research directions could explore alternative techniques and delve into ethical considerations, also open the study to other industries, not only technology sector, ensuring a more complete understanding of machine learning applications in mental health at the workplace.

8 BIBLIOGRAPHY

Bentéjac, C., Csörgő, A., & Martínez-Muñoz, G. (2021). A comparative analysis of gradient boosting algorithms. *Artificial Intelligence Review*, *54*(3), 1937–1967. https://doi.org/10.1007/s10462-020-09896-5

Bubonya, M., Cobb-Clark, D. A., & Wooden, M. (2017). Mental health and productivity at work: Does what you do matter? *Labour Economics*, *46*, 150–165. https://doi.org/10.1016/j.labeco.2017.05.001

Chisholm, D., Sweeny, K., Sheehan, P., Rasmussen, B., Smit, F., Cuijpers, P., & Saxena, S. (2016). Scaling-up treatment of depression and anxiety: a global return on investment analysis. *The Lancet Psychiatry*, *3*(5), 415–424. https://doi.org/10.1016/S2215-0366(16)30024-4

D'Alfonso, S. (2020). AI in mental health. *Current Opinion in Psychology*, *36*, 112–117. https://doi.org/10.1016/j.copsyc.2020.04.005

De silva, M. J., Lee, L., Fuhr, D. C., Rathod, S., Chisholm, D., Schellenberg, J., & Patel, V. (2014). Estimating the coverage of mental health programmes: A systematic review. *International Journal of Epidemiology*, *43*(2), 341–353. https://doi.org/10.1093/ije/dyt191

Evans-Lacko, S., Clement, S., Corker, E., Brohan, E., Dockery, L., Farrelly, S., Hamilton, S., Pinfold, V., Rose, D., Henderson, C., Thornicroft, G., & McCrone, P. (2014). How much does mental health discrimination cost: Valuing experienced discrimination in relation to healthcare care costs and community participation. *Epidemiology and Psychiatric Sciences*, *24*(5), 423–434. https://doi.org/10.1017/S2045796014000377

Fit Mind, Fit Job. (2015). OECD. https://doi.org/10.1787/9789264228283-en

Graham, S., Depp, C., Lee, E. E., Nebeker, C., Tu, X., Kim, H. C., & Jeste, D. V. (2019). Artificial Intelligence for Mental Health and Mental Illnesses: an Overview. In *Current Psychiatry Reports* (Vol. 21, Issue 11). Current Medicine Group LLC 1. https://doi.org/10.1007/s11920-019-1094-0

Gustavsson, A., Svensson, M., Jacobi, F., Allgulander, C., Alonso, J., Beghi, E., Dodel, R., Ekman, M., Faravelli, C., Fratiglioni, L., Gannon, B., Jones, D. H., Jennum, P., Jordanova, A., Jönsson, L., Karampampa, K., Knapp, M., Kobelt, G., Kurth, T., … Olesen, J. (2011). Cost of disorders of the brain in Europe 2010. *European Neuropsychopharmacology*, *21*(10), 718–779. https://doi.org/10.1016/j.euroneuro.2011.08.008

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2021). *An Introduction to Statistical Learning with Applications in R Second Edition*.

Januschowski, T., Wang, Y., Torkkola, K., Erkkilä, T., Hasson, H., & Gasthaus, J. (2022). Forecasting with trees. *International Journal of Forecasting*, *38*(4), 1473–1481. https://doi.org/10.1016/j.ijforecast.2021.10.004

Lim, S., & Chi, S. (2019). Xgboost application on bridge management systems for proactive damage estimation. *Advanced Engineering Informatics*, *41*, 100922. https://doi.org/10.1016/j.aei.2019.100922

Mohr, D. C., Zhang, M., & Schueller, S. M. (2017). Personal Sensing: Understanding Mental Health Using Ubiquitous Sensors and Machine Learning. *Annu. Rev. Clin. Psychol*, *13*, 23–47. https://doi.org/10.1146/annurev-clinpsy-032816

Nieuwenhuijsen, K., Faber, B., Verbeek, J. H., Neumeyer-Gromen, A., Hees, H. L., Verhoeven, A. C., van der Feltz-Cornelis, C. M., & Bültmann, U. (2014). Interventions to improve return to work in depressed people. In *Cochrane Database of Systematic Reviews* (Vol. 2014, Issue 12). John Wiley and Sons Ltd. https://doi.org/10.1002/14651858.CD006237.pub3

Patel, V., Chisholm, D., Parikh, R., Charlson, F., Degenhardt, L., Dua, T., Ferrari, A., Hyman, S., Laxminarayan, R., Levin, C., Lund, C., Mora, M. M., Petersen, I., Scott, J., Shidhaye, R., Vijayakumar, L., Thornicroft, G., Whiteford, H., & DCP MNS Author Group. (2016). Republished: Addressing the burden of mental, neurological, and substance use disorders: key messages from Disease Control Priorities, 3 rd edition . *Indian Journal of Social Psychiatry*, *32*(3), 196. https://doi.org/10.4103/0971-9962.193189

Prince, M., Patel, V., Saxena, S., Maj, M., Maselko, J., Phillips, M. R., & Rahman, A. (2007). No health without mental health. *The Lancet*, *370*(9590), 859–877. https://doi.org/10.1016/S0140-6736(07)61238-0

Schoenbaum, M., Unützer, J., McCaffrey, D., Duan, N., Sherbourne, C., & Wells, K. B. (2002). The Effects of Primary Care Depression Treatment on Patients' Clinical Status and Employment. *Health Services Research*, *37*(5), 1145–1158. https://doi.org/10.1111/1475-6773.01086

Shatte, A. B. R., Hutchinson, D. M., & Teague, S. J. (2019). Machine learning in mental health: a scoping review of methods and applications. *Psychological Medicine*, *49*(09), 1426–1448. https://doi.org/10.1017/S0033291719000151

Taddy, Matt. (2019). *Business Data Science: Combining Machine Learning and Economics to Optimize, Automate, and Accelerate Business Decisions.*

Vandenheuvel, A., & Wooden, M. (1995). Do Explanations of Absenteeism Differ for Men and Women? *Human Relations*, *48*(11), 1309–1329. https://doi.org/10.1177/001872679504801104

Woo, J. M., Kim, W., Hwang, T. Y., Frick, K. D., Choi, B. H., Seo, Y. J., Kang, E. H., Kim, S. J., Ham, B. J., Lee, J. S., & Park, Y. L. (2011). Impact of depression on work productivity and its improvement after outpatient treatment with antidepressants. *Value in Health*, *14*(4), 475–482. https://doi.org/10.1016/j.jval.2010.11.006

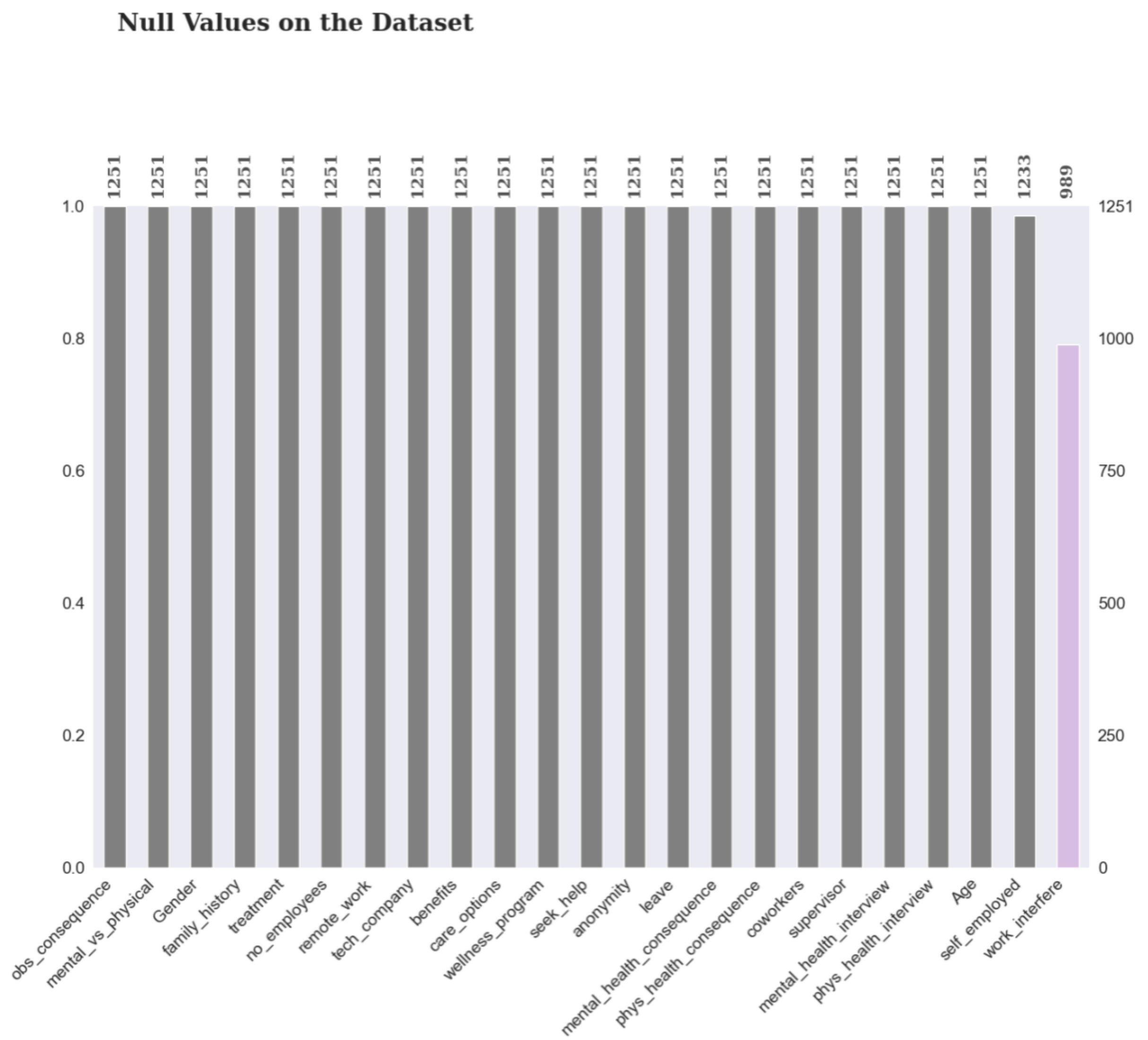
Zhao, Y., Gong, L., Zhou, B., Huang, Y., & Liu, C. (2016). Detecting tomatoes in greenhouse scenes by combining AdaBoost classifier and colour analysis. *Biosystems Engineering*, *148*, 127–137. https://doi.org/10.1016/j.biosystemseng.2016.05.001

9 APPENDIX

9.1 Code

Checking NULL VALUES

```
sns.set_style('dark')
color = ['grey','grey','grey','grey','grey','grey','grey','grey','grey','grey' ,'grey','grey',
'grey','grey','grey','grey','grey','grey', 'grey','grey','grey','grey','#D7BDE2']
msno.bar(df,fontsize =14, color = color, sort = 'descending', figsize = (15,10))

plt.text(0.05,1.265,'Null Values on the Dataset', {'font':'serif', 'size':20, 'weight':'bold'})
plt.xticks( rotation = 90,

**{'font':'serif','size':14,'weight':'bold','horizontalalignment':
'center'},alpha = 0.8)

plt.show()
```



Handling Missing Values

```
#Blank values will be replaced with 'Don't Know' for work_interfere column
df['work_interfere'] = df['work_interfere'].fillna('Don\'t know' )

#Blank values will be replaced with 'No' for self_employed column

df['self_employed'] = df['self_employed'].fillna('No')
df.isnull().sum()
```

Age 0
Gender 0
self_employed 0
family_history 0
treatment 0

```
work_interfere 0

no_employees 0

remote_work 0

tech_company 0

benefits 0

care_options 0

wellness_program            0

seek_help 0

anonymity 0

leave 0

mental_health_consequence    0
phys_health_consequence      0

coworkers 0

supervisor 0

mental_health_interview      0
phys_health_interview        0
mental_vs_physical           0
obs_consequence              0

dtype: int64

# Mapping values in 'work_interfere' column to binary categories

df['work_interfere'] = df['work_interfere'].map({
    'Often': "yes",
    'Rarely': "yes",
    'Sometimes': "yes",
    'Never': "no",
    "Don't know": "no"})
```

```
print(df['work_interfere'])

0                    yes

1                    yes

2                    yes

3                    yes

4 no



1254 no

...



1255 yes



1256 yes



1257 no



1258 yes



Name: work_interfere, Length: 1251, dtype: object

#Columns and unique values
list_col = ['Age', 'Gender', 'self_employed', 'family_history',
'treatment',
           'work_interfere', 'no_employees', 'remote_work',
'tech_company',
           'benefits', 'care_options', 'wellness_program',
'seek_help',
           'anonymity', 'leave', 'mental_health_consequence',
           'phys_health_consequence', 'coworkers', 'supervisor',
           'mental_health_interview', 'phys_health_interview',
           'mental_vs_physical', 'obs_consequence']
for col in list_col:
   print('{}: {} '.format(col.upper(), df[col].unique()))

AGE: [37 44 32 31 33 35 39 42 23 29 36 27 46 41 34 30 40 38 50 24 18
```

```
28 26 22

19 25 45 21 43 56 60 54 55 48 20 57 58 47 62 51 65 49 53 61 72]

GENDER: ['Female' 'Male' 'Other']
SELF_EMPLOYED: ['No' 'Yes']
FAMILY_HISTORY: ['No' 'Yes']
TREATMENT: ['Yes' 'No']
WORK_INTERFERE: ['yes' 'no']

NO_EMPLOYEES: ['Jun-25' 'More than 1000' '26-100' '100-500' '01-May'

'500-1000']

REMOTE_WORK: ['No' 'Yes']
TECH_COMPANY: ['Yes' 'No']

BENEFITS: ['Yes' "Don't know" 'No']

CARE_OPTIONS: ['Not sure' 'No' 'Yes']

WELLNESS_PROGRAM: ['No' "Don't know" 'Yes']

SEEK_HELP: ['Yes' "Don't know" 'No']

ANONYMITY: ['Yes' "Don't know" 'No']

LEAVE: ['Somewhat easy' "Don't know" 'Somewhat difficult' 'Very

difficult'

'Very easy']

MENTAL_HEALTH_CONSEQUENCE: ['No' 'Maybe' 'Yes']
PHYS_HEALTH_CONSEQUENCE: ['No' 'Yes' 'Maybe']

COWORKERS: ['Some of them' 'No' 'Yes']

SUPERVISOR: ['Yes' 'No' 'Some of them']

MENTAL_HEALTH_INTERVIEW: ['No' 'Yes' 'Maybe']
PHYS_HEALTH_INTERVIEW: ['Maybe' 'No' 'Yes']

MENTAL_VS_PHYSICAL: ['Yes' "Don't know" 'No']

OBS_CONSEQUENCE: ['No' 'Yes']

# Label Encoding the categorical variables

from sklearn.preprocessing import LabelEncoder
object_cols = ['Gender', 'self_employed', 'family_history', 'treatment',

    'work_interfere', 'no_employees', 'remote_work',
'tech_company',
    'benefits', 'care_options', 'wellness_program', 'seek_help',
    'anonymity', 'leave', 'mental_health_consequence',
```

```
       'phys_health_consequence', 'coworkers', 'supervisor',
       'mental_health_interview', 'phys_health_interview',
       'mental_vs_physical', 'obs_consequence']
label_encoder = LabelEncoder()
for col in object_cols:
    label_encoder.fit(df[col])
    df[col] = label_encoder.transform(df[col])
```

Descriptive Statistics Getting the Descriptive Statistics

```
# Main descriptive statistics
desc_stats = df.describe(include='all')
# Display descriptive statistics
print("Descriptive Statistics:")
print(desc_stats)
```

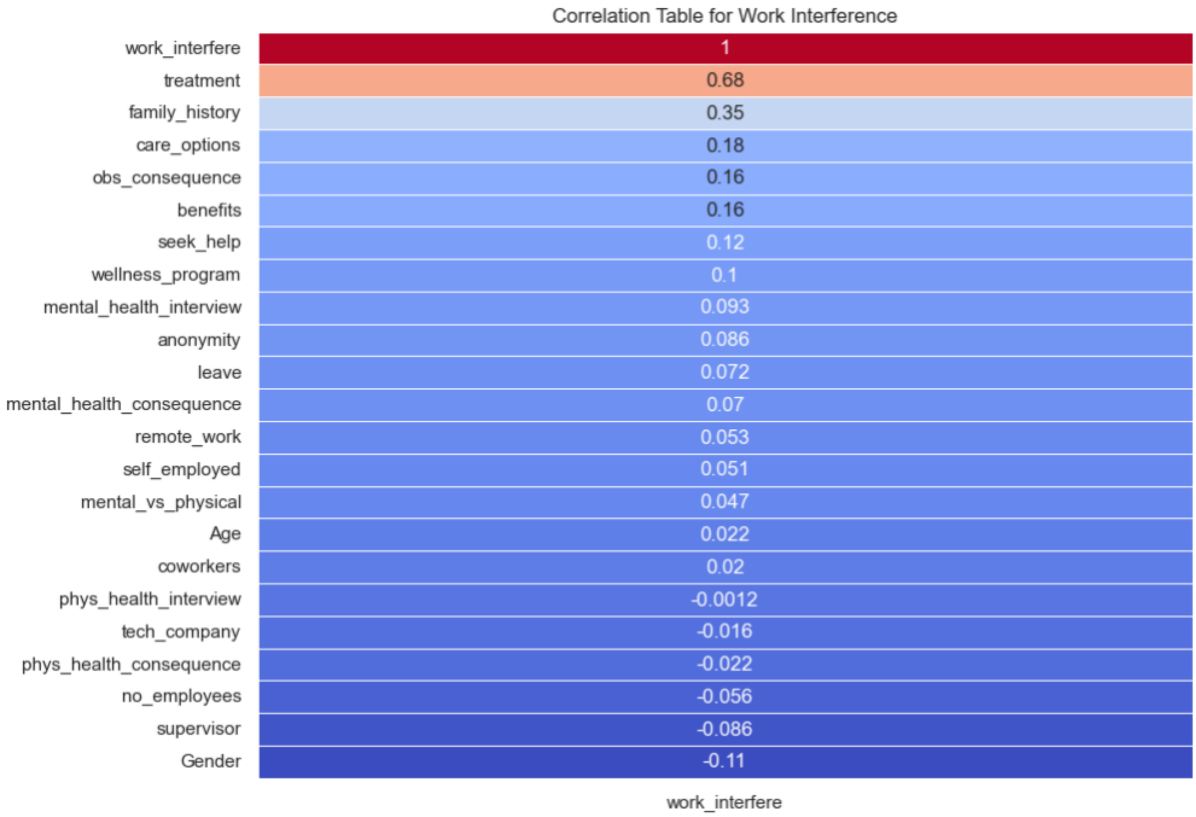
```
# List of the most important numerical columns

important_numerical_cols = ['Age', 'work_interfere', 'treatment',
'family_history']

# Generate descriptive statistics for the selected columns

descriptive_stats = df[important_numerical_cols].describe()

# Print the descriptive statistics table

print("Descriptive Statistics for Important Columns:")
print(descriptive_stats)

Descriptive Statistics for Important Columns:

Age  work_interfere    treatment  family_history

count 1251.000000 1251.000000 1251.000000 1251.000000

mean 32.076739 0.621103 0.505196 0.390887

std 7.288272 0.485306 0.500173 0.488144

min 18.000000 0.000000 0.000000 0.000000

25% 27.000000 0.000000 0.000000 0.000000

50% 31.000000 1.000000 1.000000 0.000000

75% 36.000000 1.000000 1.000000 1.000000

max 72.000000 1.000000 1.000000 1.000000

# List of the most important numerical columns

important_numerical_cols_2 = ['care_options', 'obs_consequence',
'benefits', 'remote_work']
```

```
# Generate descriptive statistics for the selected columns

descriptive_stats_2 = df[important_numerical_cols_2].describe()

# Print the descriptive statistics table

print("Descriptive Statistics for Important Columns:")
print(descriptive_stats_2)

Descriptive Statistics for Important Columns:

care_options obs_consequence benefits remote_work

count 1251.000000 1251.000000 1251.000000 1251.000000

mean 0.952038 0.144684 1.052758 0.296563

std 0.864926 0.351923 0.837385 0.456925

min 0.000000 0.000000 0.000000 0.000000

25% 0.000000 0.000000 0.000000 0.000000

50% 1.000000 0.000000 1.000000 0.000000

75% 2.000000 0.000000 2.000000 1.000000

max 2.000000 1.000000 2.000000 1.000000

#Distribution of Age
plt.figure(figsize=(12, 6))
sns.histplot(df['Age'], bins=20, kde=True, palette='Greys',
color='black')
plt.title('Distribution of Age')
plt.xlabel('Age')
plt.ylabel('Frequency')
plt.show()
```

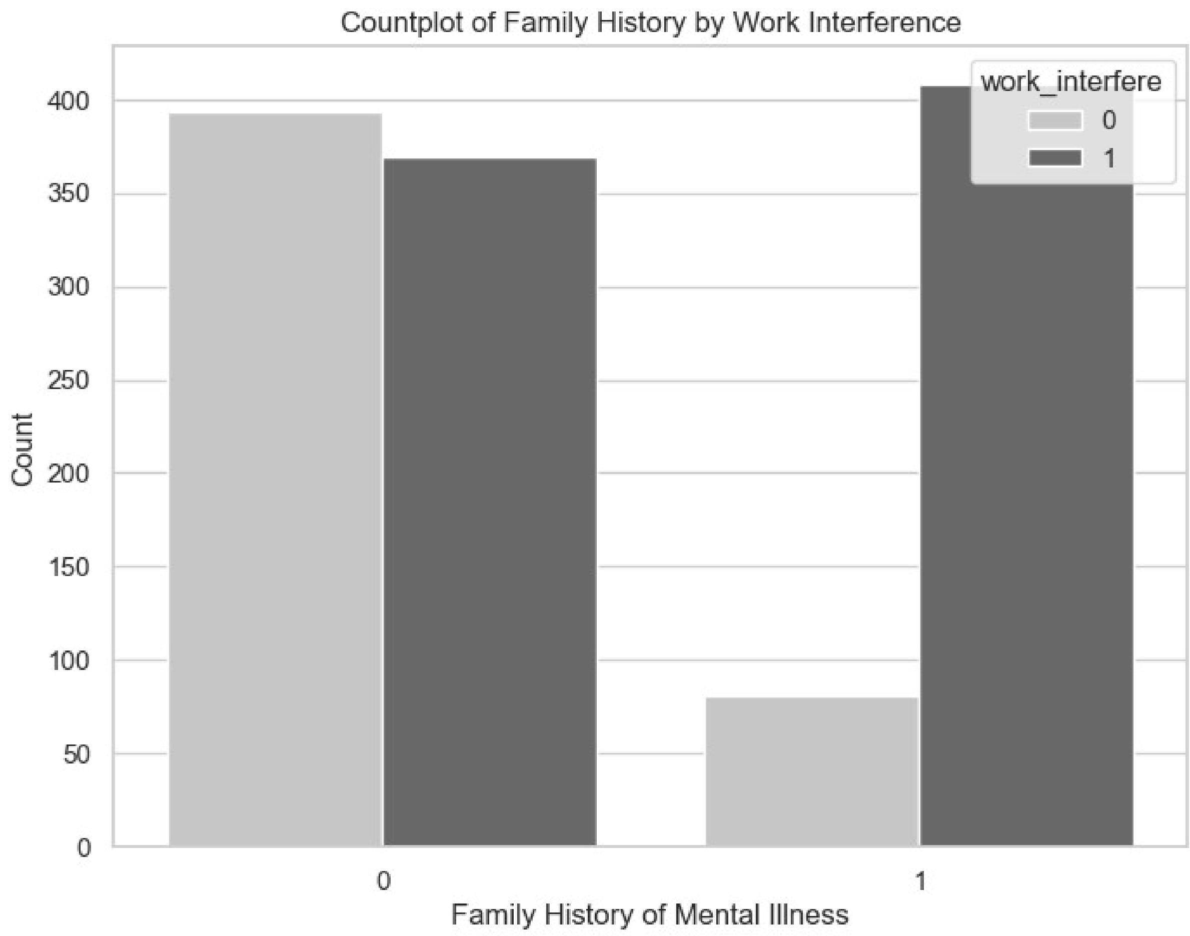

```
#Boxplot of Age by Work Interference
plt.figure(figsize=(10, 6))
sns.boxplot(x='work_interfere', y='Age', data=df, palette='Greys')
plt.title('Boxplot of Age by Work Interference')
plt.xlabel('Work Interference')
plt.ylabel('Age')
# Set x-axis tick labels
plt.xticks([0, 1], ['No', 'Yes'])
plt.show()
```


```
#CORRELATION TABLE FOR 'WORK INTERFERENCE'
sns.set(style="whitegrid")
correlation_work_interfere = df.corr()
['work_interfere'].sort_values(ascending=False)
```

```
plt.figure(figsize=(10, 8))
sns.heatmap(correlation_work_interfere.to_frame(), annot=True,
cmap='coolwarm', linewidths=.5, cbar=False)
plt.title('Correlation Table for Work Interference')
plt.show()
```




```
#Countplot of 'family_history' by 'work_interfere'

plt.figure(figsize=(8, 6))
sns.countplot(x='family_history', hue='work_interfere', data=df, palette='Greys')
plt.title('Countplot of Family History by Work Interference') plt.xlabel('Family History of Mental Illness')
plt.ylabel('Count')
plt.show()
```

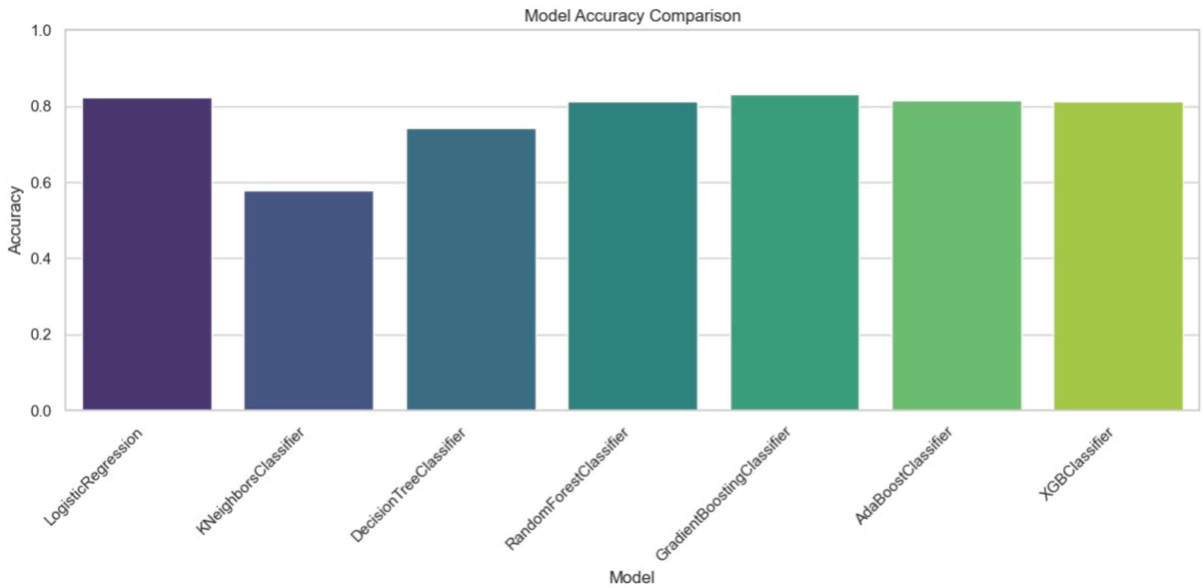


Predicting

```
# Setting 'work_interfere' as the target variable
target_variable = 'work_interfere'
# Splitting the data
X = df.drop(target_variable, axis=1)
y = df[target_variable]

X_train, X_test, y_train, y_test = train_test_split(X, y, stratify=y, test_size=0.3, random_state=101)

# Creating the models to be evaluated
key = ['LogisticRegression', 'KNeighborsClassifier',
'DecisionTreeClassifier', 'RandomForestClassifier',
'GradientBoostingClassifier', 'AdaBoostClassifier', 'XGBClassifier']
value = [
    LogisticRegression(),
    KNeighborsClassifier(n_neighbors=2, weights='uniform'),
    DecisionTreeClassifier(random_state=10),
    RandomForestClassifier(n_estimators=60, random_state=0),
    GradientBoostingClassifier(random_state=20),
    AdaBoostClassifier(),
    xgb.XGBClassifier(random_state=0, booster="gbtree")
]
models = dict(zip(key, value))
#Predicting just on the Test Data
predicted = []
results = []
```

```
for name, algo in models.items():
    model = algo
    model.fit(X_train, y_train)
    predict = model.predict(X_test)
    acc = accuracy_score(y_test, predict)
    predicted.append(acc)
    results.append({'Model': name, 'Accuracy': acc})
    print(f"{name} Accuracy: {acc:.4f}")

#Comparing performance in the Train and Test Dataset

train_predicted = []
train_results = []
test_predicted = []
test_results = []
for name, algo in models.items():
    model = algo
    model.fit(X_train, y_train)
    # Predict on training set
    train_predict = model.predict(X_train)
    train_acc = accuracy_score(y_train, train_predict)
    train_predicted.append(train_acc)
    train_results.append({'Model': name, 'Training Accuracy':

train_acc})

    # Predict on test set
    test_predict = model.predict(X_test)

test_acc = accuracy_score(y_test, test_predict) # Corrected from 'predict' to 'test_predict'

    test_predicted.append(test_acc)
    test_results.append({'Model': name, 'Test Accuracy': test_acc})
LogisticRegression Accuracy: 0.8245
KNeighborsClassifier Accuracy: 0.5798
DecisionTreeClassifier Accuracy: 0.7420
RandomForestClassifier Accuracy: 0.8138
GradientBoostingClassifier Accuracy: 0.8324
AdaBoostClassifier Accuracy: 0.8165
XGBClassifier Accuracy: 0.8138

print(f"{name} Training Accuracy: {train_acc:.4f}, Test Accuracy: {test_acc:.4f}")

LogisticRegression Training Accuracy: 0.8469, Test Accuracy: 0.8245

KNeighborsClassifier Training Accuracy: 0.8046, Test Accuracy: 0.5798

DecisionTreeClassifier Training Accuracy: 1.0000, Test Accuracy:

0.7420

RandomForestClassifier Training Accuracy: 1.0000, Test Accuracy:

0.8138

GradientBoostingClassifier Training Accuracy: 0.8914, Test Accuracy:
```

```
0.8324

AdaBoostClassifier Training Accuracy: 0.8469, Test Accuracy: 0.8165

XGBClassifier Training Accuracy: 1.0000, Test Accuracy: 0.8138

#Visualizing the Accuracy
results_df = pd.DataFrame(results)

plt.figure(figsize=(12, 6))
sns.barplot(x='Model', y='Accuracy', data=results_df, palette='viridis')
plt.title('Model Accuracy Comparison')
plt.xlabel('Model')
plt.ylabel('Accuracy')
plt.ylim(0, 1) # Set y-axis limits to better visualize differences plt.xticks(rotation=45, ha='right') # Rotate x-axis labels for better readability
plt.tight_layout() # Adjust layout to prevent label overlap plt.show()
```




```
results_df = pd.DataFrame(results)
# Display the results table using tabulate
print("Model Accuracy Comparison:")
print(tabulate(results_df, headers='keys', tablefmt='fancy_grid',
showindex=False))
Model Accuracy Comparison:
```

Model	Accuracy
LogisticRegression	0.824468
KNeighborsClassifier	0.579787
DecisionTreeClassifier	0.742021
RandomForestClassifier	0.81383
GradientBoostingClassifier	0.832447

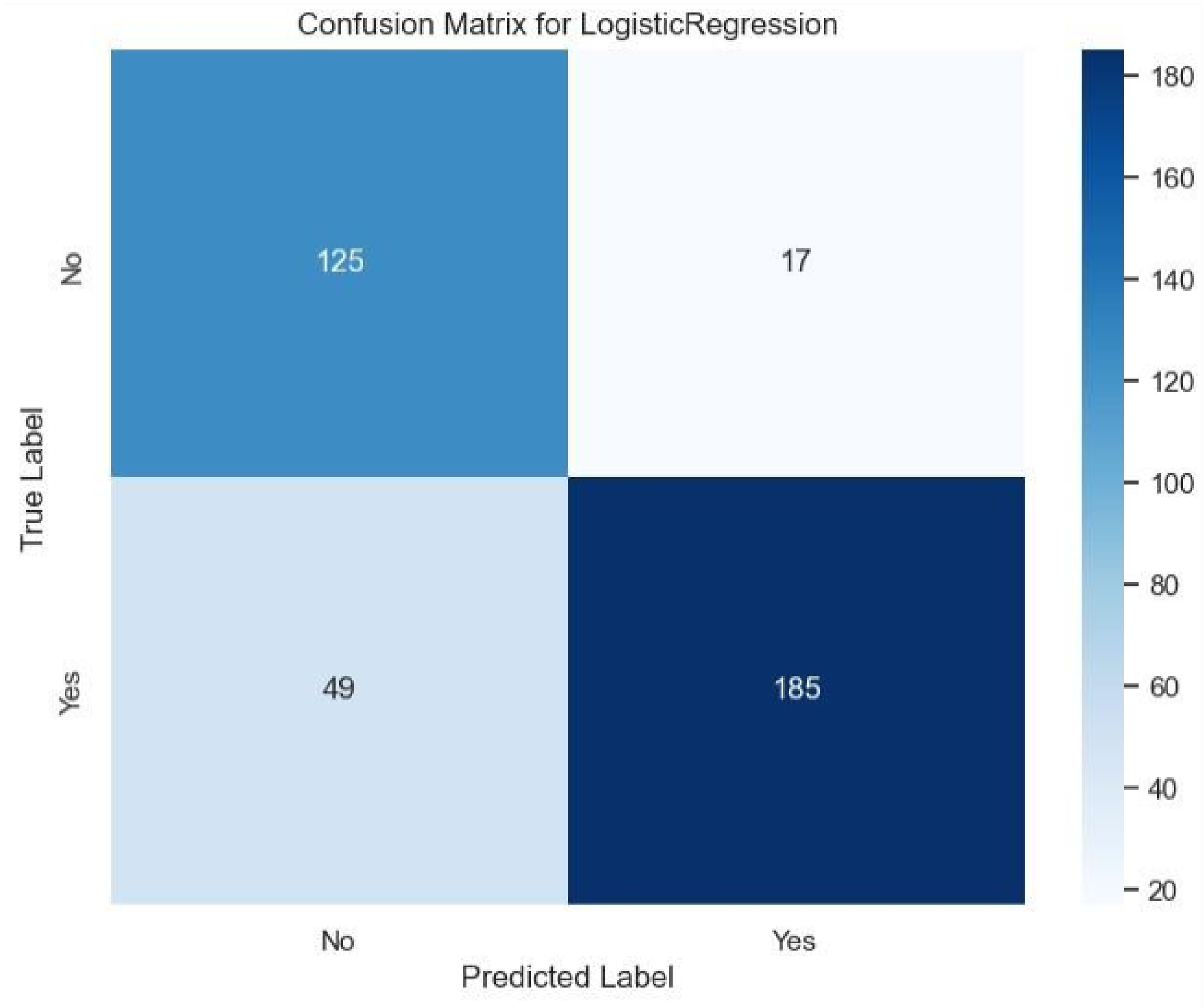
AdaBoostClassifier	0.816489
XGBClassifier	0.81383

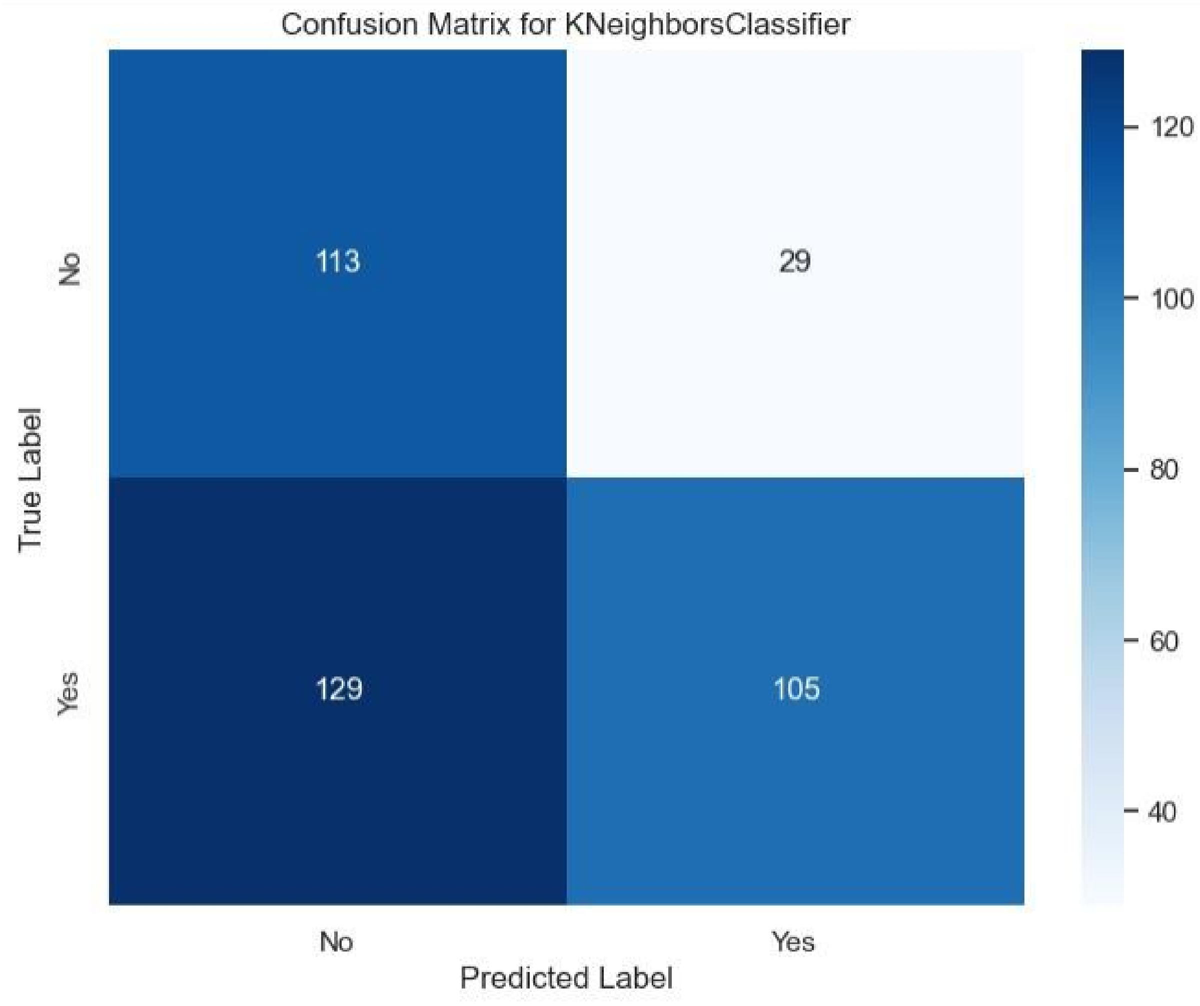
```
#Confusion Matrix

results = {'Model': [], 'Accuracy': [], 'Precision': [], 'Recall': [], 'F1 Score': []}

for name, algo in models.items():
    model = algo
    model.fit(X_train, y_train)
    predict = model.predict(X_test)
    acc = accuracy_score(y_test, predict)
    precision = precision_score(y_test, predict)
    recall = recall_score(y_test, predict)
    f1 = f1_score(y_test, predict)

#3 decimals

acc = round(acc, 3)
precision = round(precision, 3) recall = round(recall, 3)
f1 = round(f1, 3)

    results['Model'].append(name)
    results['Accuracy'].append(acc)
    results['Precision'].append(precision)
    results['Recall'].append(recall)
    results['F1 Score'].append(f1)
    #Display confusion matrix
    cm = confusion_matrix(y_test, predict)
    print(f"\nConfusion Matrix for {name}:\n{cm}")

#Storing the results in a DataFrame for plotting

results_df = pd.DataFrame(results)

#Displaying the results in a table

print("\nModel Performance Metrics:")
print(results_df)
Confusion Matrix for LogisticRegression:


[[125 17]



[ 49 185]]


Confusion Matrix for KNeighborsClassifier:


[[113 29]
```

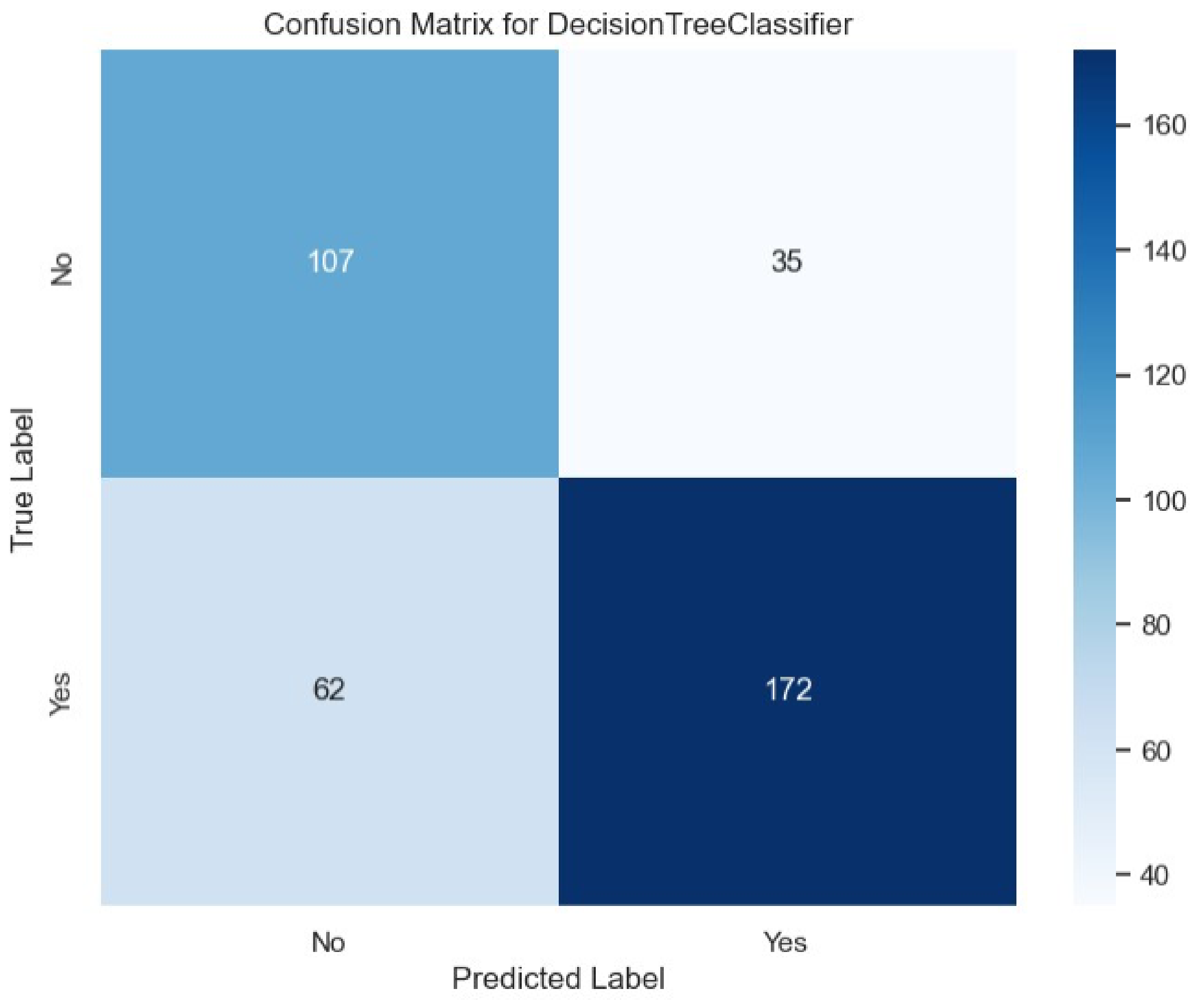
```
[129 105]]

Confusion Matrix for DecisionTreeClassifier:

[[107 35]

[ 62 172]]

Confusion Matrix for RandomForestClassifier:

[[120 22]

[ 48 186]]

Confusion Matrix for GradientBoostingClassifier:

[[127 15]

[ 48 186]]

Confusion Matrix for AdaBoostClassifier:

[[119 23]

[ 46 188]]

Confusion Matrix for XGBClassifier:

[[116 26]

[ 44 190]]

Model Performance Metrics:
```

```
Model Accuracy Precision Recall F1 Score

0 LogisticRegression 0.824 0.916 0.791 0.849

1 KNeighborsClassifier 0.580 0.784 0.449 0.571

2 DecisionTreeClassifier 0.742 0.831 0.735 0.780

3 RandomForestClassifier 0.814 0.894 0.795 0.842

4 GradientBoostingClassifier 0.832 0.925 0.795 0.855

5 AdaBoostClassifier 0.816 0.891 0.803 0.845

6 XGBClassifier 0.814 0.880 0.812 0.844

print("Confussion Matrix Results:")
print(tabulate(results_df.reset_index(), headers='keys',
tablefmt='fancy_grid', showindex=False))
Confussion Matrix Results:
╒══════════╤════════════════════════════════════════╤═════════════════╤══════════════
═══╤═══
═══════════╤═════════════════╕

│ index │ Model │ Accuracy │ Precision │

Recall │   F1 Score │
╞══════════╪════════════════════════════════════════╪═════════════════╪══════════════
═══╪═══
═══════════╪═════════════════╡

│ 0 │ LogisticRegression │ 0.824 │ 0.916 │

0.791 │      0.849 │
├──────────┼────────────────────────────────────────┼─────────────────┼──────────────
──┼───
───────────┼─────────────────┤

│ 1 │ KNeighborsClassifier │ 0.58 │ 0.784 │

0.449 │      0.571 │
├──────────┼────────────────────────────────────────┼─────────────────┼──────────────
──┼───
───────────┼─────────────────┤

│ 2 │ DecisionTreeClassifier │ 0.742 │ 0.831 │

0.735│ 0.78 │

├──────────┼────────────────────────────────────────┼─────────────────┼──────────────
──┼───
───────────┼─────────────────┤

│ 3 │ RandomForestClassifier │ 0.814 │ 0.894 │

0.795 │      0.842 │
```

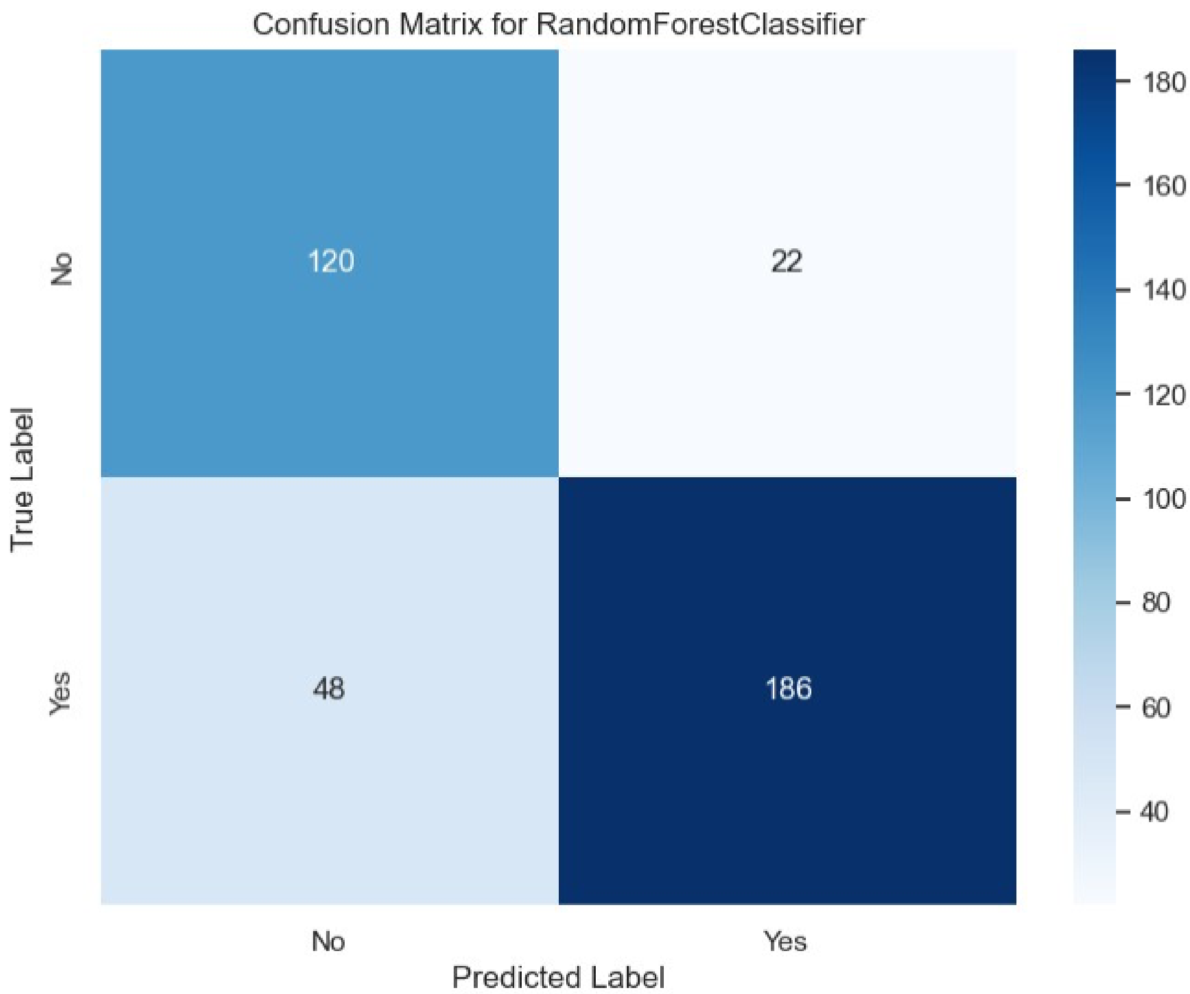
```
├──────────┼─────────────────────────────┼─────────────┼───────────
──┼─────
───────┼─────────────┤

│ 4 │ GradientBoostingClassifier │ 0.832 │ 0.925 │

0.795 │    0.855 │
├──────────┼─────────────────────────────┼─────────────┼───────────
──┼─────
───────┼─────────────┤

│ 5 │ AdaBoostClassifier │ 0.816 │ 0.891 │

0.803 │    0.845 │
├──────────┼─────────────────────────────┼─────────────┼───────────
──┼─────
───────┼─────────────┤

│ 6 │ XGBClassifier │ 0.814 │ 0.88 │

0.812 │    0.844 │
╘══════════╧═════════════════════════════╧═════════════╧═══════════
══╧═════
═══════╧═════════════╛

#Visualizing Confusion Matrix for each model

for name, algo in models.items():
   model = algo
   model.fit(X_train, y_train)
   predict = model.predict(X_test)
   cm = confusion_matrix(y_test, predict)
   plt.figure(figsize=(8, 6))
   sns.heatmap(cm, annot=True, fmt='d', cmap='Blues',
xticklabels=['No', 'Yes'], yticklabels=['No', 'Yes'])
   plt.title(f"Confusion Matrix for {name}")
   plt.xlabel("Predicted Label")
   plt.ylabel("True Label")

plt.show()
```

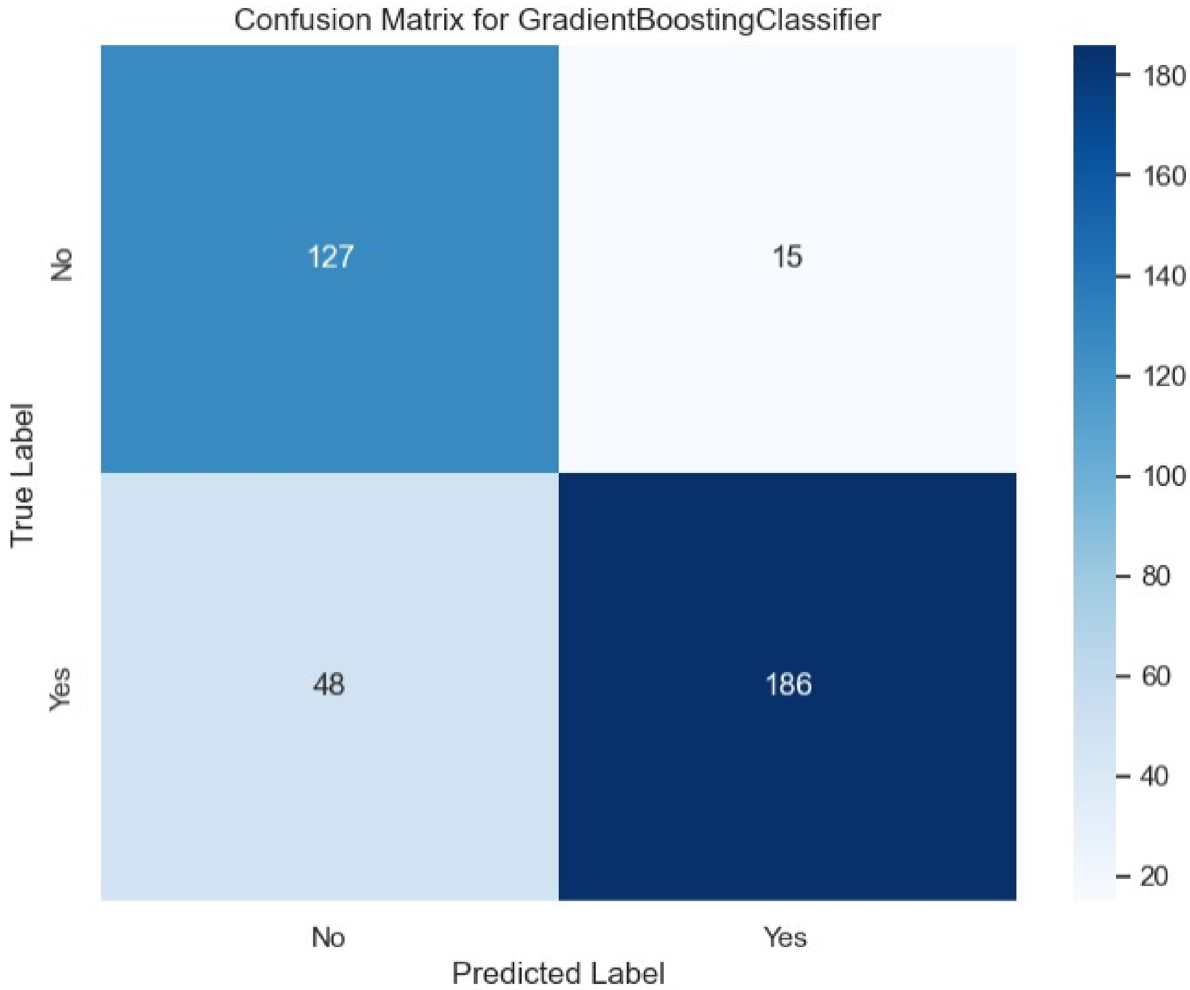

Confusion Matrix for LogisticRegression
True Label
No
Yes
125
17
49
185
Predicted Label
No
Yes
180
160
140
120
100
80
60
40
20

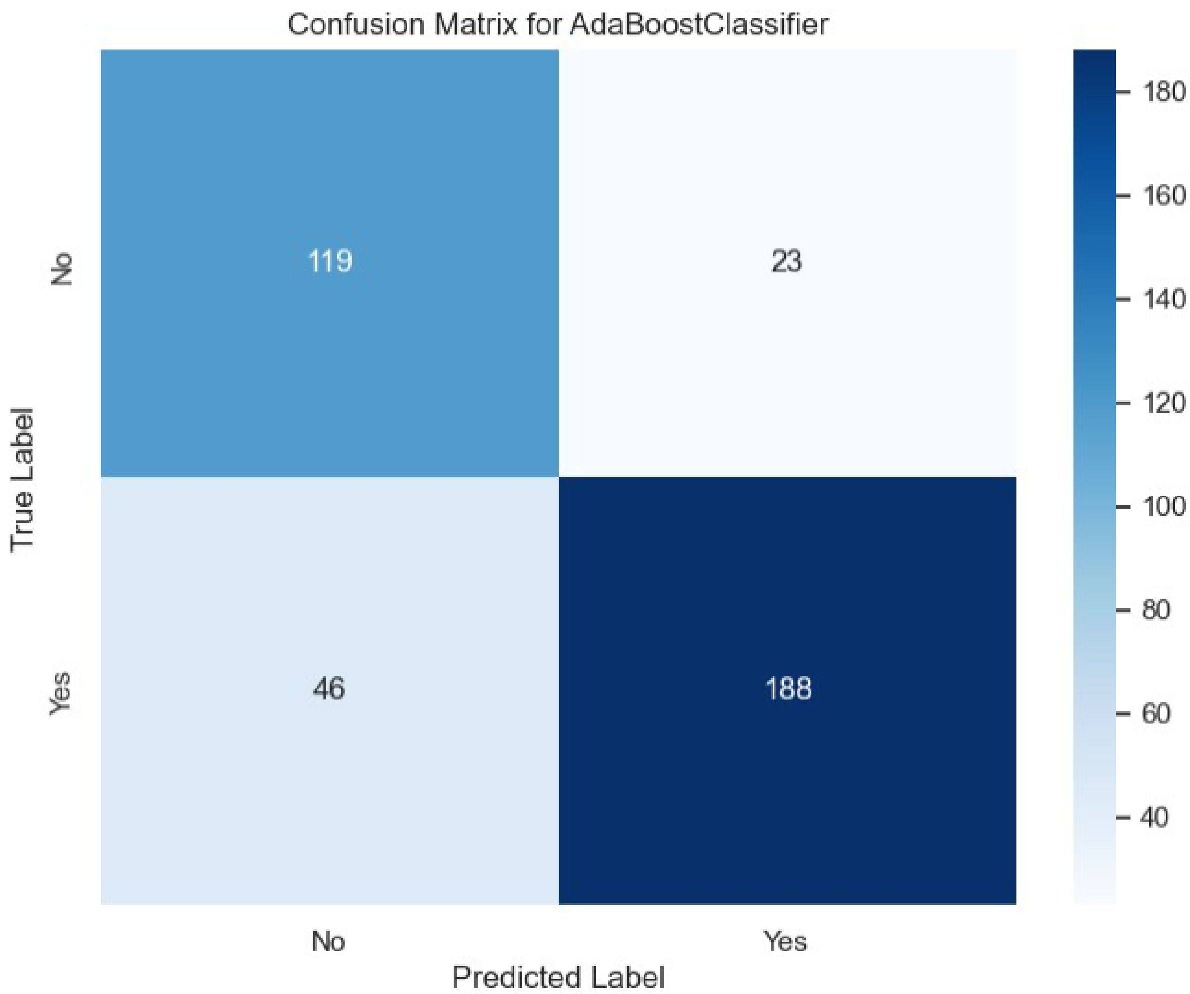

Confusion Matrix for KNeighborsClassifier
113
29
129
105
No
Yes
No
Yes
True Label
Predicted Label
120
100
80
60
40

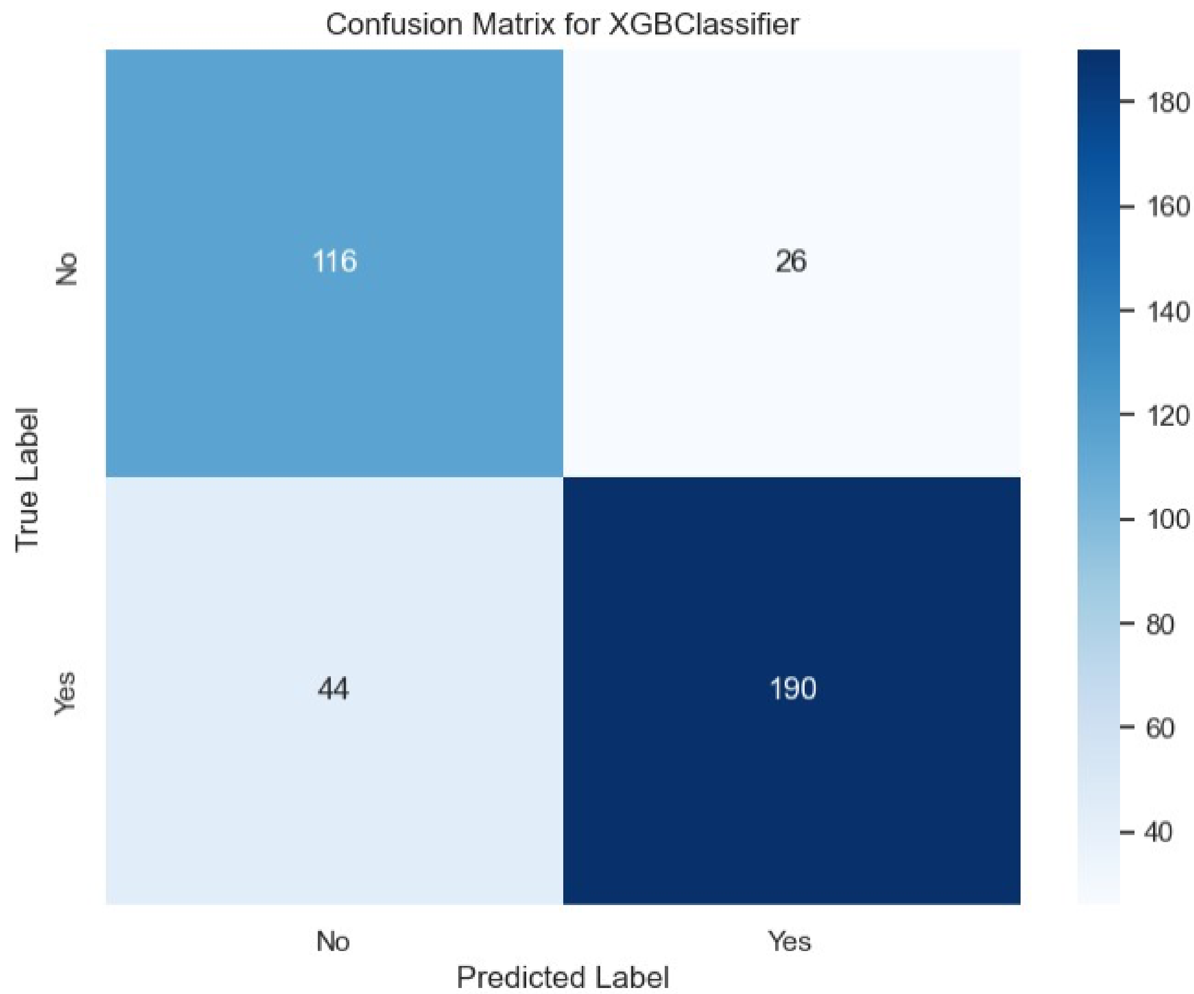


Confusion Matrix for DecisionTreeClassifier
True Label
No
Yes
107
35
62
172
160
140
120
100
80
60
40
No
Yes
Predicted Label

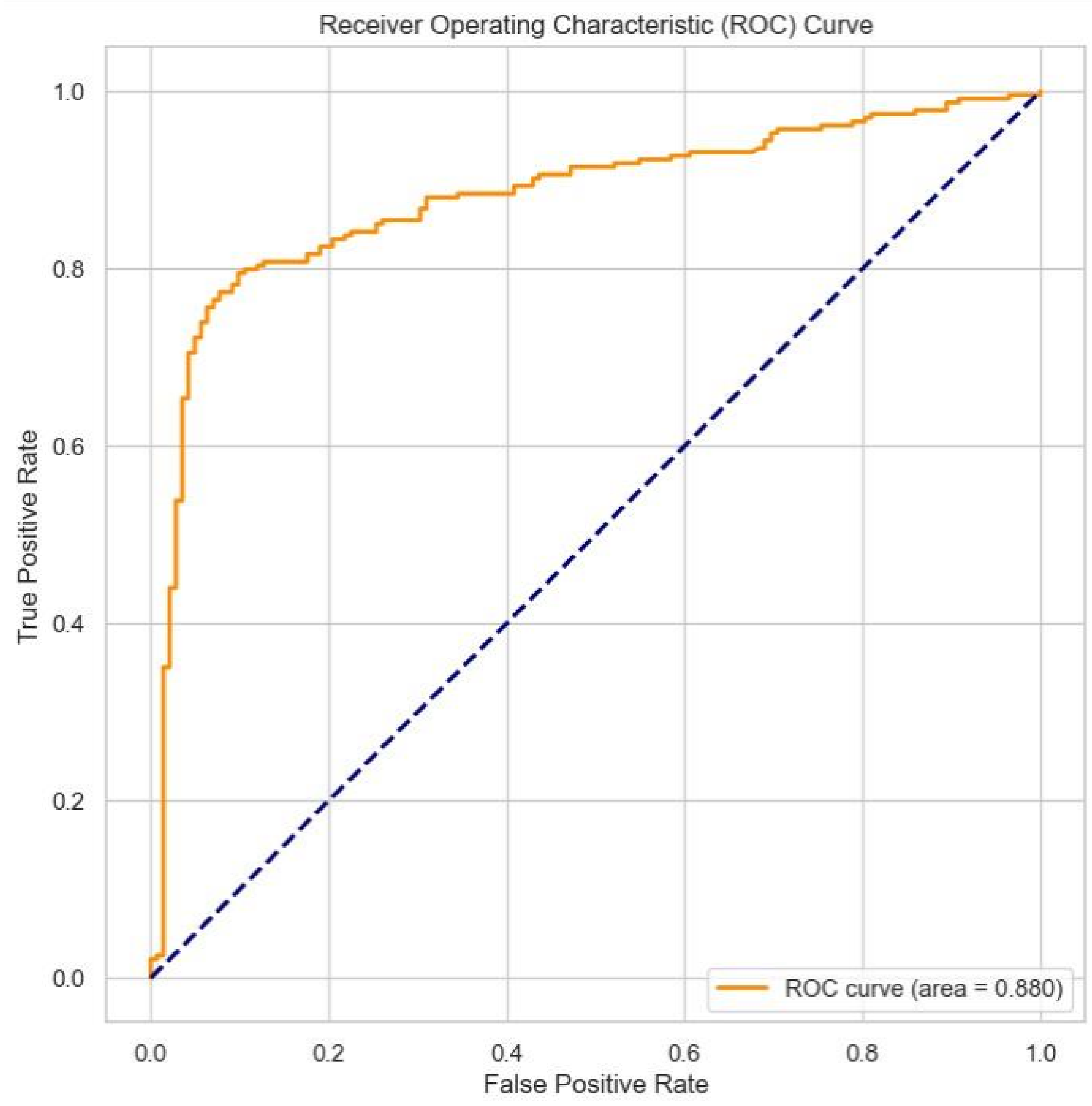

Confusion Matrix for RandomForestClassifier
True Label
No
Yes
120
22
48
186
No
Yes
Predicted Label
180
160
140
120
100
80
60
40


Confusion Matrix for GradientBoostingClassifier
127
15
48
186
No
Yes
True Label
No
Yes
Predicted Label
180
160
140
120
100
80
60
40
20


Confusion Matrix for AdaBoostClassifier
True Label
No
Yes
119
23
46
188
Predicted Label
No
Yes
180
160
140
120
100
80
60
40




```
#ROC Curve for GradientBoostingClassifier
model = GradientBoostingClassifier()
model.fit(X_train, y_train)

#Get predicted probabilities for the positive class

y_prob = model.predict_proba(X_test)[:, 1]

#Computing ROC curve and AUC

fpr, tpr, thresholds = roc_curve(y_test, y_prob) roc_auc = auc(fpr, tpr)

#Plotting ROC curve
plt.figure(figsize=(8, 8))
plt.plot(fpr, tpr, color='darkorange', lw=2, label='ROC curve (area =
{:.3f})'.format(roc_auc))
plt.plot([0, 1], [0, 1], color='navy', lw=2, linestyle='--')
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('Receiver Operating Characteristic (ROC) Curve')
plt.legend(loc='lower right')
plt.show()
```




```
#Predicting on the Training and Test Dataset for All the Evaluation Metrics

train_results = {'Model': [], 'Training Accuracy': [], 'Training Precision': [], 'Training Recall': [], 'Training F1
Score': []} test_results = {'Model': [], 'Test Accuracy': [], 'Test Precision':

[], 'Test Recall': [], 'Test F1 Score': []}
for name, algo in models.items():
    model = algo
    model.fit(X_train, y_train)
    #Predict on training set
    train_predict = model.predict(X_train)
    train_acc = accuracy_score(y_train, train_predict)
    train_precision = precision_score(y_train, train_predict)
    train_recall = recall_score(y_train, train_predict)
    train_f1 = f1_score(y_train, train_predict)
    #Round the values to 3 decimals
```

```
train_acc = round(train_acc, 3) train_precision = round(train_precision, 3) train_recall = round(train_recall, 3)
train_f1 = round(train_f1, 3)

    train_results['Model'].append(name)
    train_results['Training Accuracy'].append(train_acc)
    train_results['Training Precision'].append(train_precision)
    train_results['Training Recall'].append(train_recall)
    train_results['Training F1 Score'].append(train_f1)
    #Predict on test set

test_predict = model.predict(X_test)
test_acc = accuracy_score(y_test, test_predict) test_precision = precision_score(y_test, test_predict) test_recall =
recall_score(y_test, test_predict) test_f1 = f1_score(y_test, test_predict)

    #Round the values to 3 decimals

test_acc = round(test_acc, 3) test_precision = round(test_precision, 3) test_recall = round(test_recall, 3) test_f1 =
round(test_f1, 3)

    test_results['Model'].append(name)
    test_results['Test Accuracy'].append(test_acc)
    test_results['Test Precision'].append(test_precision)
    test_results['Test Recall'].append(test_recall)
    test_results['Test F1 Score'].append(test_f1)

print(f"{name} Training Accuracy: {train_acc:.4f}, Test Accuracy: {test_acc:.4f}")

#Store the results in DataFrames
train_results_df = pd.DataFrame(train_results)
test_results_df = pd.DataFrame(test_results)
#Display the results tables

print("\nTraining Model Performance Metrics:") print(train_results_df)

print("\nTest Model Performance Metrics:")
print(test_results_df)

LogisticRegression Training Accuracy: 0.8470, Test Accuracy: 0.8240

KNeighborsClassifier Training Accuracy: 0.8050, Test Accuracy: 0.5800

DecisionTreeClassifier Training Accuracy: 1.0000, Test Accuracy:

0.7420

RandomForestClassifier Training Accuracy: 1.0000, Test Accuracy:

0.8140

GradientBoostingClassifier Training Accuracy: 0.8910, Test Accuracy:

0.8320

AdaBoostClassifier Training Accuracy: 0.8470, Test Accuracy: 0.8160

XGBClassifier Training Accuracy: 1.0000, Test Accuracy: 0.8140
```

Training Model Performance Metrics:

Precision \

1 2 3 4 5 6

KNeighborsClassifier
DecisionTreeClassifier
RandomForestClassifier
GradientBoostingClassifier
AdaBoostClassifier
XGBClassifier

0.805	1.000
1.000	1.000
1.000	1.000
0.891	0.934
0.847	0.897
1.000	1.000

Model Training Accuracy Training

0 LogisticRegression 0.847 0.911

Training Recall Training F1 Score
0 0.834 0.871
1 0.685 0.813
2 1.000 1.000
3 1.000 1.000
4 0.888 0.910
5 0.851 0.873
6 1.000 1.000

Test Model Performance Metrics:

Recall \

Model Test Accuracy Test Precision Test

0 LogisticRegression 0.824 0.916

0.791

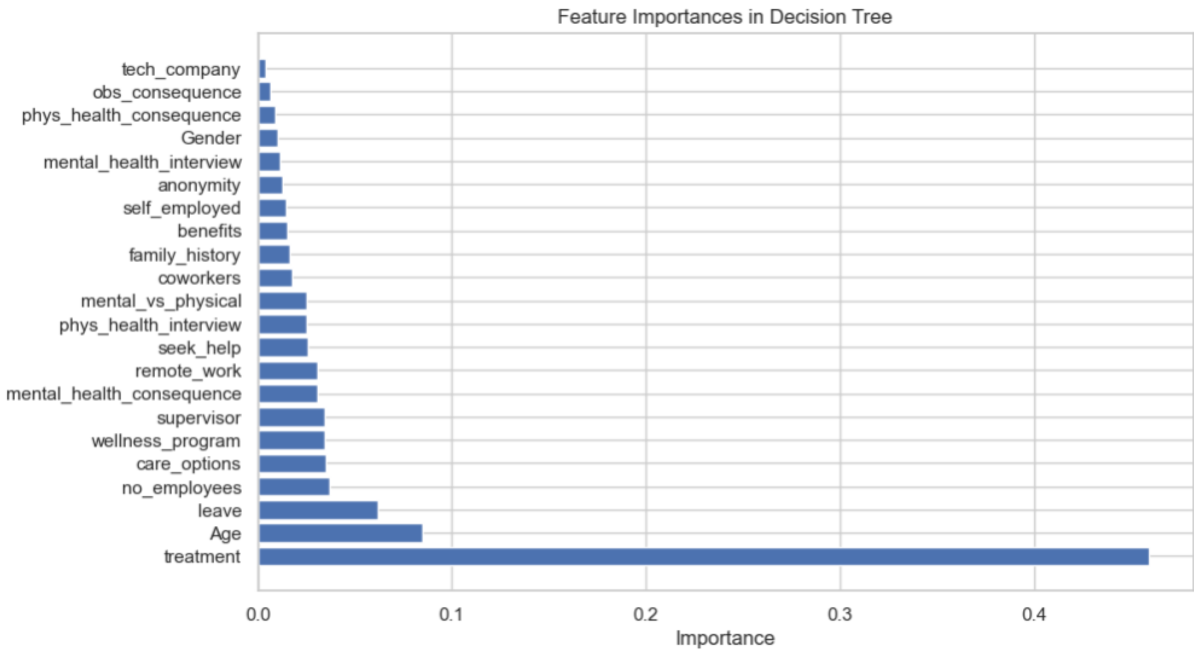
```
1 KNeighborsClassifier 0.580 0.784

0.449

2 DecisionTreeClassifier 0.742 0.831

0.735

3 RandomForestClassifier 0.814 0.894

0.795

4 GradientBoostingClassifier 0.832 0.925

0.795

5 AdaBoostClassifier 0.816 0.891

0.803

6 XGBClassifier 0.814 0.880

0.812
```

Test F1 Score
0 0.849
1 0.571
2 0.780
3 0.842
4 0.855
5 0.845
6 0.844

```
#Visualizing only GradientBoostingClassifier Results

gb_train_results = train_results_df[train_results_df['Model'] ==
'GradientBoostingClassifier']
gb_test_results = test_results_df[test_results_df['Model'] ==
'GradientBoostingClassifier']

#Combine train and test results for Gradient Boosting Classifier

gb_results_comparison = pd.concat([gb_train_results, gb_test_results],
keys=['Train', 'Test'])
# Convert the DataFrame to a dictionary
gb_results_dict = gb_results_comparison.to_dict(orient='records')
# Display the comparison table

print("\nPerformance Metrics Comparison for Gradient Boosting Classifier:")
for row in gb_results_dict:

    print("\n" + tabulate([row], headers='keys',
tablefmt='fancy_grid'))

Performance Metrics Comparison for Gradient Boosting Classifier:
```

```
╒════════════════════════════╤═════════════════════╤════════════
══════
════╤═══════════════════╤═════════════════════╤═════════════════
╤══════
═════════════╤═══════════════╤═════════════════╕

│ Model │ Training Accuracy │ Training

Precision │ Training Recall │ Training F1 Score │ Test Accuracy

│ Test Precision │ Test Recall │ Test F1 Score │

╞════════════════════════════╪═════════════════════╪════════════
══════
════╪═══════════════════╪═════════════════════╪═════════════════
╪══════
═════════════╪═══════════════╪═════════════════╡
│ GradientBoostingClassifier │         0.891 │

0.934 │ 0.888 │ 0.91 │ nan │

nan │       nan │        nan │
╘════════════════════════════╧═════════════════════╧════════════
══════
════╧═══════════════════╧═════════════════════╧═════════════════
╧══════
═════════════╧═══════════════╧═════════════════╛
╒════════════════════════════╤═════════════════════╤════════════
══════
════╤═══════════════════╤═════════════════════╤═════════════════
╤══════
═════════════╤═══════════════╤═════════════════╕

│ Model │ Training Accuracy │ Training

Precision │ Training Recall │ Training F1 Score │ Test Accuracy

│ Test Precision │ Test Recall │ Test F1 Score │

╞════════════════════════════╪═════════════════════╪════════════
══════
════╪═══════════════════╪═════════════════════╪═════════════════
╪══════
═════════════╪═══════════════╪═════════════════╡
│ GradientBoostingClassifier │           nan │

nan │ nan │ nan │ 0.832 │

0.925 │ 0.795 │ 0.855 │

╘════════════════════════════╧═════════════════════╧════════════
══════
════╧═══════════════════╧═════════════════════╧═════════════════
╧══════
═════════════╧═══════════════╧═════════════════╛
#DECISION TREE
#Fit the decision tree model
dt_model = DecisionTreeClassifier(random_state=10)
```

```
dt_model.fit(X_train, y_train)
#Extract feature importances
feature_importances = dt_model.feature_importances_

#Create a DataFrame to display feature importances

importance_df = pd.DataFrame({'Feature': X_train.columns,
'Importance': feature_importances})

#Sort the DataFrame by importance in descending order

importance_df = importance_df.sort_values(by='Importance',
ascending=False)
#Plot the feature importances

plt.figure(figsize=(10, 6))
plt.barh(importance_df['Feature'], importance_df['Importance']) plt.xlabel('Importance')
plt.title('Feature Importances in Decision Tree')

plt.show()

#Display the sorted importance DataFrame

print("Feature Importances:")
print(importance_df)
```




```
Feature Importances:
Feature  Importance
4                  treatment    0.458827

0 Age 0.084526

13 leave 0.062043

5              no_employees    0.036720
9              care_options    0.035148
10          wellness_program    0.034446
```

```
17                supervisor    0.034412
14  mental_health_consequence    0.030741
6                remote_work    0.030733
11                 seek_help    0.025584
19      phys_health_interview    0.024971
20         mental_vs_physical    0.024842
16                 coworkers    0.017494
3             family_history    0.016134
8                   benefits    0.015035
2              self_employed    0.014551
12                 anonymity    0.012776
18    mental_health_interview    0.011196

1 Gender 0.010125

15    phys_health_consequence    0.009099
```