Employability Prediction Using Different Machine Learning Methods

Ginny Dang

The University of Texas at Austin

A I 395T

Dr. Junfeng Jiao

December 2, 2024

Abstract

This research investigates the application of machine learning (ML) models to predict employability based on data from over 70,000 job applicants. The goal is to compare the performance of various ML algorithms in predicting employability outcomes, focusing on accuracy and the ability to distinguish between classes. Specifically, logistic regression, decision tree, random forest, XG boost, lightGBM, shallow neural network, and stack ensemble are employed and evaluated. The results provide insights into each approach's comparative accuracy and ability to distinguish between classes, shedding light on their suitability for real-world recruitment scenarios. The findings aim to assist recruiters and organizations in making more informed decisions while optimizing the hiring process.

Keywords: Employability, binary classification, logistic regression, decision tree, random forest, XG boost, lightGBM, shallow neural network, stack ensemble

Employability Prediction Using Different Machine Learning Methods

In an increasingly competitive job market, organizations and recruiters face the challenge of identifying the most suitable candidates from an ever-expanding pool of applicants. Traditional recruitment procedures, often reliant on subjective evaluations and limited data, can be time-consuming, inefficient, and prone to bias. In response, machine learning (ML) in recruitment has gained significant traction; it helps automate a large part of the recruitment process (Rąb-Kettler and Lehnervp, 2019), evaluate job applicants (Faliagka et al., 2012), and offers a data-driven approach to predict efficient candidates joining (Reddy et al., 2020). Using historical data and predictive algorithms, ML models have the potential to streamline candidate evaluation, uncover valuable information, and support fair and inclusive hiring practices (Kulkarni-Masurkar, n.d.).

This study investigates the prediction of employability in the data set *Employability Classification of Over 70,000 Job Applicants* found in Kaggle (Tankha, n.d.). The primary objective is to compare the performance of various ML algorithms in employability prediction, focusing on accuracy and the ability to distinguish between classes. The models and methods include logistic regression, decision tree, random forest, XG boost, lightGBM, shallow neural network, and stacking ensemble. Analyzing the performance of these approaches aims to provide recruiters with insight into selecting appropriate models based on their specific needs and constraints.

Research Background

Recruitment is critical for organizations seeking to attract and retain top talent. However, traditional recruitment methods often struggle with inefficiencies and subjective decision-making. In this context, machine learning (ML) offers a promising solution that enables predictive modeling and data-driven insights, making hiring more efficient and objective (Kulkarni-Masurkar, n.d.). A dynamic bias mitigation framework for AI-powered recruitment systems was developed to improve fairness, identify and address real-time biases, and promote fairness between various demographic groups (Yanamala, 2022). In addition, a job matching ML algorithm has also been developed to prevent (as much as possible) the pitfall of unfairness and discrimination (Delecraz et al., 2022). Past research has also shown the potential of machine learning to transform recruitment processes. Researchers have proposed a data analytics approach that serves as a decision support tool for recruiters in real-world settings by combining a local prediction model, which evaluates recruitment success for individual candidates and job types, with a global optimization model that streamlines the recruitment process to align organizational needs with candidate profiles (Pessach et al., 2020). Recently, researchers have used human resource documents, such as resume data, to streamline the employee selection process using machine learning techniques, including latent semantic analysis (LSA), bidirectional encoder representation transformer (BERT), and support vector machines (SVM) (Tian et al., 2023).

Despite significant advancements in ML applications in recruitment, limited research has focused on systematically comparing different ML models for employability prediction. Furthermore, existing studies often need to pay more attention to integrating diverse applicant attributes, such as coding experience, previous salary, and technical skills, which are critical to robust predictions.

Dataset

Overview

This research analysis utilizes the *Kaggle* dataset, *Employability Classification of Over 70,000 Job Applicants*, which provides a unique opportunity to analyze various characteristics of the applicants such as age, level of education, gender, coding experience, previous salary and computer skills, as shown in **Figure 1**. The data set, curated from various sources such as job portals, career fairs, and online applications, encompasses various attributes relevant to job applicants (Tankha, n.d.).

Figure 1

"stackoverflow_full.csv" from Kaggle's "Employability Classification of Over 70,000 Job Applicants" dataset

	Unnamed: 0	Age	Accessibility	EdLevel	Employment	Gender	MentalHealth	MainBranch	YearsCode	YearsCodePro	Country	PreviousSalary	HaveWorkedWith	ComputerSkills	Employed
0	0	<35	No	Master	1	Man	No	Dev	7	4	Sweden	51552.0	C++;Python;Git;PostgreSQL	4	0
1	1	<35	No	Undergraduate	1	Man	No	Dev	12	5	Spain	46482.0	Bash/Shell;HTML/CSS;JavaScript;Node.js;SQL;Typ...	12	1
2	2	<35	No	Master	1	Man	No	Dev	15	6	Germany	77290.0	C;C++;Java;Perl;Ruby;Git;Ruby on Rails	7	0
3	3	<35	No	Undergraduate	1	Man	No	Dev	9	6	Canada	46135.0	Bash/Shell;HTML/CSS;JavaScript;PHP;Ruby;SQL;Gi...	13	0
4	4	>35	No	PhD	0	Man	No	NotDev	40	30	Singapore	160932.0	C++;Python	2	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
73457	73457	<35	No	Undergraduate	1	Man	No	Dev	7	2	Germany	41058.0	C#;HTML/CSS;JavaScript;TypeScript;Docker;Kuber...	13	1
73458	73458	>35	No	Undergraduate	1	Man	No	Dev	21	16	United States of America	115000.0	C#;HTML/CSS;Java;JavaScript;npm;ASP.NET Core ;...	11	1
73459	73459	<35	No	Undergraduate	1	Man	No	Dev	4	3	Nigeria	57720.0	HTML/CSS;JavaScript;TypeScript;Docker;Express;...	12	1
73460	73460	<35	Yes	Undergraduate	1	Man	Yes	Dev	5	1	United States of America	70000.0	C#;HTML/CSS;JavaScript;SQL;TypeScript;npm;Yarn...	15	1
73461	73461	<35	No	Master	1	NonBinary	No	Dev	10	3	United Kingdom of Great Britain and Northern I...	75384.0	Python;Docker	2	0

73462 rows × 15 columns

Data Cleaning

The *Unnamed* column in the data set does not contain any information, since it is only used to number the data points, so it is safe to drop that column during preprocessing. Moreover, the *HaveWorkedWith* column contains many *null* values, making it impractical for a meaningful analysis; we also drop it to streamline the data set and focus on the most relevant features for our analysis. **Figure 2** demonstrates the structure of the dataset after data cleaning.

Figure 2

"stackoverflow_full.csv" from Kaggle's "Employability Classification of Over 70,000 Job Applicants" dataset after data cleaning

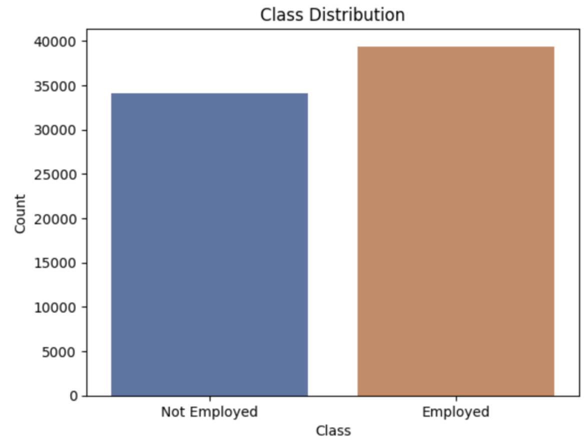
	Age	Accessibility	EdLevel	Employment	Gender	MentalHealth	MainBranch	YearsCode	YearsCodePro	Country	PreviousSalary	ComputerSkills	Employed
0	<35	No	Master	1	Man	No	Dev	7	4	Sweden	51552.0	4	0
1	<35	No	Undergraduate	1	Man	No	Dev	12	5	Spain	46482.0	12	1
2	<35	No	Master	1	Man	No	Dev	15	6	Germany	77290.0	7	0
3	<35	No	Undergraduate	1	Man	No	Dev	9	6	Canada	46135.0	13	0
4	>35	No	PhD	0	Man	No	NotDev	40	30	Singapore	160932.0	2	0

Exploratory Data Analysis

Class Imbalance Analysis. The data set appears relatively balanced, as the counts for the two classes (0 - *Not Employed* and 1 - *Employed*) are close, with neither class significantly outweighing the other, as shown in **Figure 3**.

Figure 3

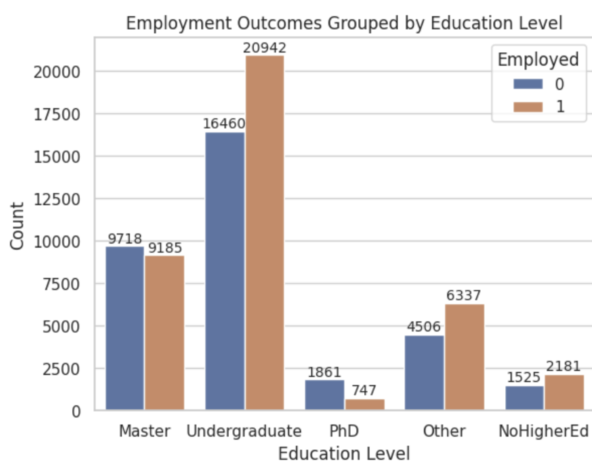
Target class distribution



Bivariate Analysis. To examine how the dependent variable is influenced or explained by the independent variable, we perform a bivariate analysis (Bertani et al., 2018). When looking at *EdLevel*, which is Education Level, we find an intriguing observation from this data set: advanced degrees, that is, master's and Ph.D., do not correlate with higher employability (**Figure 4**).

Figure 4

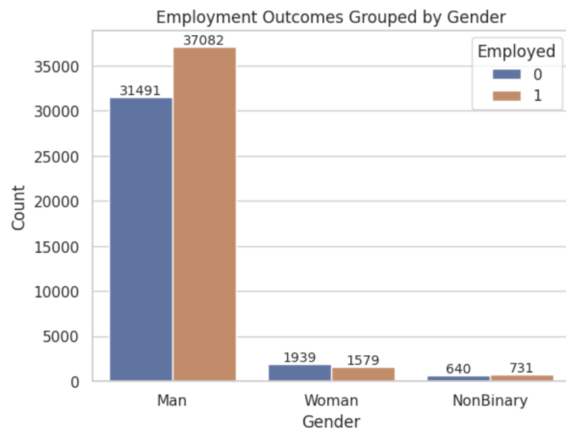
Employability by "Education Level"



Bivariate analysis also reveals significant gender bias, a problem described by Njoto et al. (2022). *Men* dominate both representation and employment results, while the under-representation of *Women* and *NonBinary* individuals highlights potential systemic barriers to recruitment (**Figure 5**). Addressing these disparities requires promoting fairness and ethical design in machine learning algorithms (Dhabliya et al., 2024).

Figure 5

Employability by "Gender"



Factors such as *Previous Salary* and *Total Years of Coding Experience* show no clear distinction between employed and unemployed individuals (**Figure 6**). However, *Total Years of Professional Coding Experience* and *Number of Computer Skills* play a more substantial role in employability, with higher values corresponding to better employment outcomes (**Figure 7**). This analysis highlights the importance of skill diversity and professional experience in predicting employment success.

Figure 6

Employability by "Previous Salary" and Employability by "Total Years of Coding Experience"

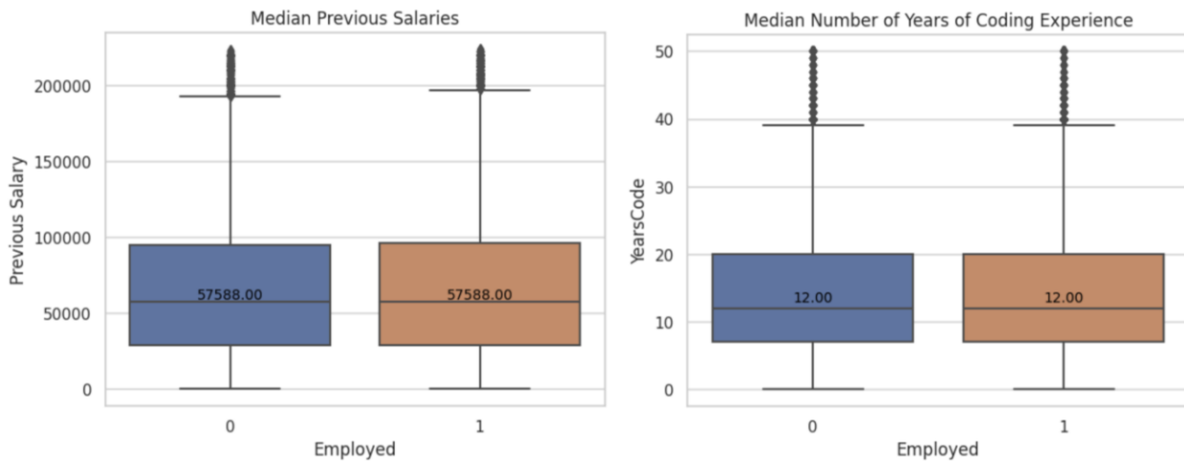
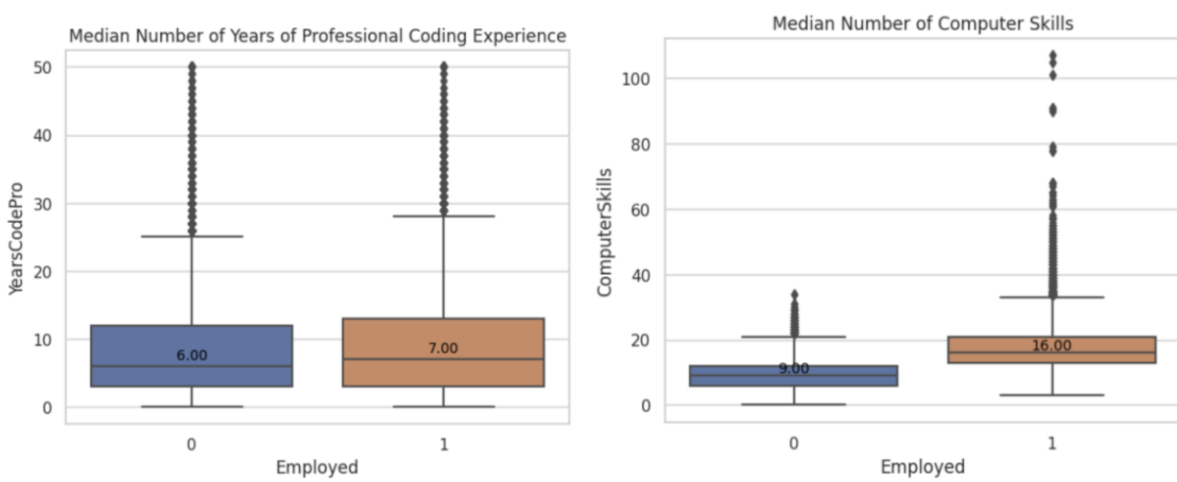


Figure 7

Employability by "Total Years of Professional Coding Experience" and Employability by "Number of Computer Skills"



The original data set includes 172 unique countries, we group them into six continents for simplified analysis: *North America*, *South America*, *Europe*, *Asia*, *Australia*, and *Others*. According to the distribution, *Europe* has the largest share of applicants (38.4%), followed by *North America* (24.8%), with the remaining continents contributing smaller proportions (**Figure 8**). Despite significant variations in the number of candidates across continents, the ratio of employed to unemployed remains relatively consistent, indicating a stable class balance globally (**Figure 9**).

Figure 8

Applicant distribution by continent

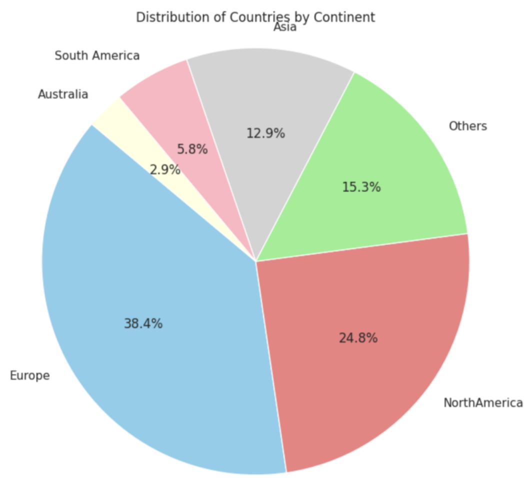
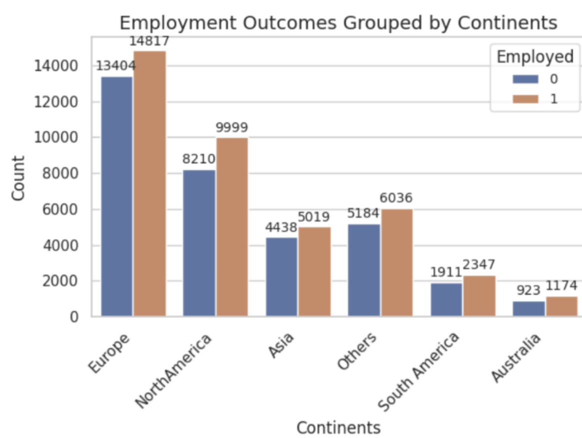


Figure 9

Employability by "Continent"



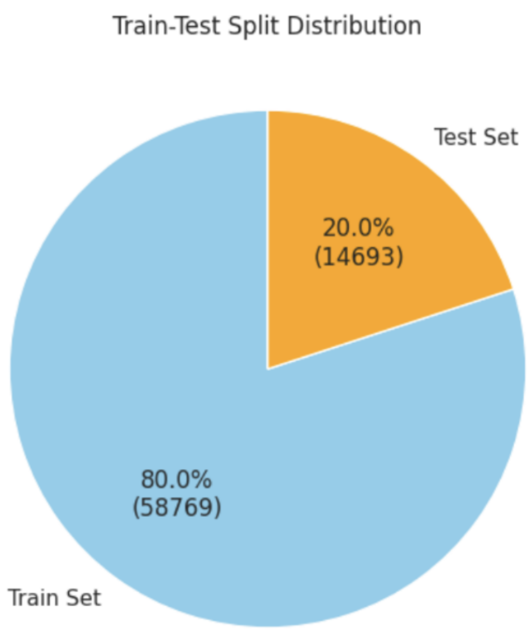
Methods

Validation

Splitting data is crucial in machine learning to promote effective model generalization (Kebonye, 2021). To ensure that the model performs effectively on unseen data and avoids bias, we need to use data that are not part of the training process (Sree, n.d.). A typical approach involves dividing the original data set into a training set to build the classifier and an independent test set to assess its performance (Dobbin and Simon, 2011). In this study, the data set is divided into training and testing subsets using an 80-20 split strategy (**Figure 10**), a common approach where 80% (58769 samples) of the data is allocated for training and 20% (14693 samples) for testing (Ahmed, n.d.).

Figure 10

Train-Test split distribution



Methods and Algorithms

Metrics. Various algorithms with different complexity and parameter sizes are used to evaluate the effectiveness of ML models in predicting the target variable. For performance assessment, we use the accuracy score and the area under the curve (AUC) as primary metrics, and the confusion matrix and the receiver operating characteristic (ROC) curve as diagnostic tools to provide deeper insights into model performance.

The AUC of the ROC curve, initially used in medical diagnosis since the 1970s, has become a widely adopted single number metric to assess the predictive performance of machine learning algorithms (Huang and Ling, 2005). The AUC-ROC curve measures the classification performance across thresholds, with ROC as the probability curve and AUC indicating class separability. A higher AUC indicates that the model is more effective at correctly classifying 0s as 0 and 1s as 1 (Narkhede, 2018).

In this study, we utilize the `metrics` module from **Scikit-learn** to calculate the accuracy scores and the AUC scores and plot the ROC curves and the confusion matrices, following the guidelines in the Scikit-learn documentation (Scikit-learn, n.d.).

Logistic Regression. Logistic regression models are used to examine the effects of predictor variables on categorical outcomes, typically binary ones such as the presence or absence of a condition (e.g. non-Hodgkin lymphoma), in which case they are called binary logistic regression models (Nick and Campbell, 2007). This study uses the `LogisticRegression` model from **Scikit-learn**'s `linear_model` module to predict employability outcomes based on factors like education, coding experience, technical skills, etc.

Decision Tree Classifier. A binary decision tree comprises the root, internal, and leaf nodes. Data flow from the root node, branching into two paths at each step. This process continues through the internal nodes until it reaches the leaf nodes, where each leaf is assigned a class label as the output (Hassan and Bermak, 2014). Here, we leverage the `DecisionTreeClassifier` from **Scikit-learn**'s `tree` module, which is capable of binary and multiclass classification, to predict employment outcomes in our data set.

Random Forest. The random forest, first introduced by Leo Breiman of the University of California in 2001, consists of multiple independent decision tree classifiers. When a test sample is input, the class label is determined by aggregating the votes of all individual classifiers (Parmar et al., 2019). We employ the `RandomForestClassifier` from **Scikit-learn**'s `ensemble` module to predict employability outcomes on our dataset. In the configuration, our classifier has 100 trees.

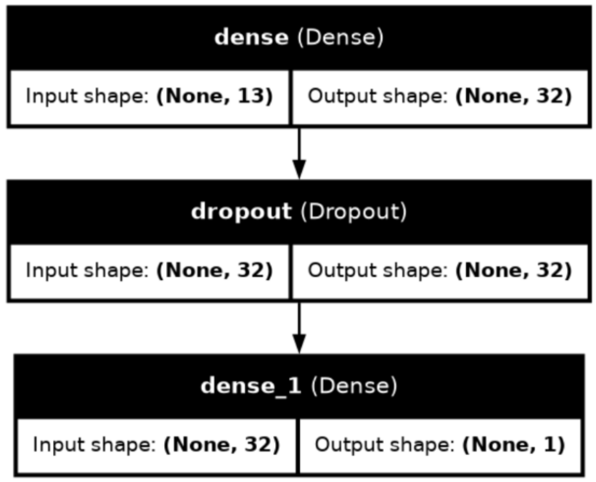
XG Boost. XG boost, or eXtreme Gradient Boosting, is a scalable ML method with tree boosting to mitigate overfitting (Chen and Guestrin, 2016). XG boost is well suited for binary classification tasks that involve predicting one of two possible classes. It models the output using a logistic function to provide probabilities for class membership (XGBoost, n.d.). For our problem, we import `XGBClassifier` from **xgboost** . Our classifier has 100 boosting rounds and a maximum tree depth of 5.

LightGBM. Another classifier used in this study is the lightGBM or light gradient boost machine. LightGBM is a gradient-boosting framework that leverages decision trees to enhance performance while reducing memory consumption. Its unique features include Gradient-Based One-Side Sampling (GOSS) and Exclusive Feature Bundling (EFB), which define its core characteristics (Goswami et al., 2024). Our experiment uses the `LGBMClassifier` implementation from **lightgbm** (LightGBM, n.d.) with 100 boosting rounds, a maximum tree depth of 5, and 31 leaves.

Shallow Neural Network. Shallow neural networks are made up of an input layer, one or two hidden layers, and a final output layer. The input layer receives the data processed by the hidden layers to generate the final output. Each layer comprises neurons that perform basic computations by applying weights and biases to their input. The results are then passed through an activation function to introduce non-linearity which allows the network to learn complex patterns in the data (Sandu, n.d.). In our experiment, we use **TensorFlow** (TensorFlow, n.d.) to implement a shallow neural network consisting of an input layer with 32 neurons and ReLU activation, a dropout layer for regularization, and an output layer with a single neuron and sigmoid activation for binary classification tasks (**Figure 11**). Also, when compiling the model, the Adam optimizer (Ioffe and Szegedy, 2015) and the Binary Cross-entropy loss function (Pardede et al., 2023) are used.

Figure 11

Shallow Neural Network architecture



Stack Ensemble Classifier. The Stack Ensemble approach combines multiple models to minimize bias and variance (Snehi and Bhandari, 2022). Using a stacking approach, our experiment combined three base models: decision tree classifier, random forest classifier, and XG boost classifier. The implementation is done with `StackingClassifier` imported from the **Scikit-learn**'s `ensemble` module.

Results

Accuracy and AUC scores

The table below shows the model performance metrics for the seven algorithms considered.

Table 1

Model Performance

Algorithm	Accuracy	AUC
Logistic Regression	0.7376	0.8190
Decision Tree	0.7135	0.7127
Random Forest	0.7766	0.8603
XG Boost	0.7911	0.8753
LightGBM	0.7925	0.8750
Shallow Neural Network	0.7914	0.8736
Stack Ensemble	0.7874	0.8735

Confusion Matrix

A confusion matrix summarizes the performance of a classification model by comparing predicted and actual labels. It includes the counts of true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN) from the model predictions. (Evidently AI, n.d.). **Figures 12-15** show the confusion matrices of all models.

Figure 12

Confusion matrices of Logistic Regression and Decision Tree

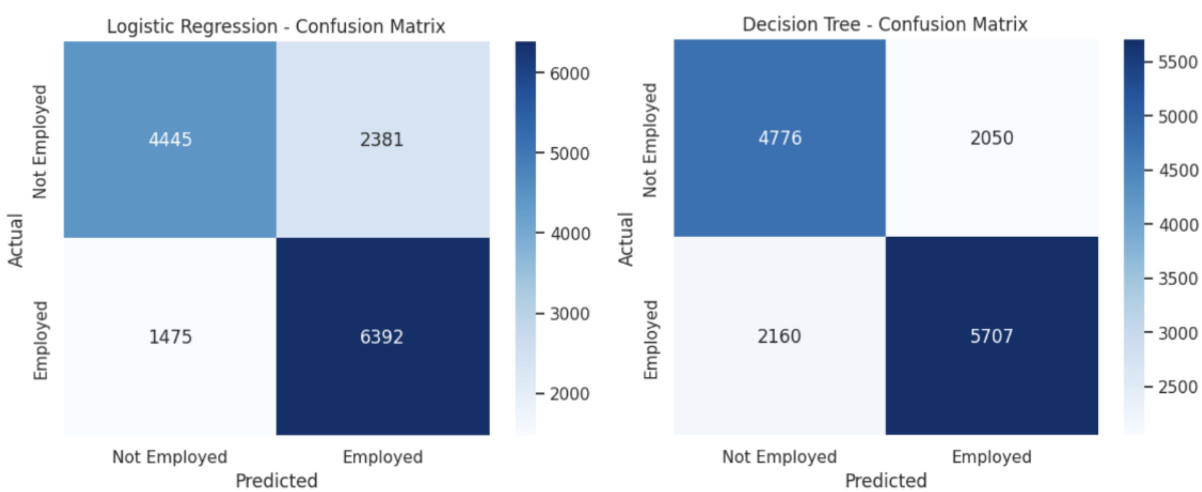


Figure 13

Confusion matrices of Random Forest and XG Boost

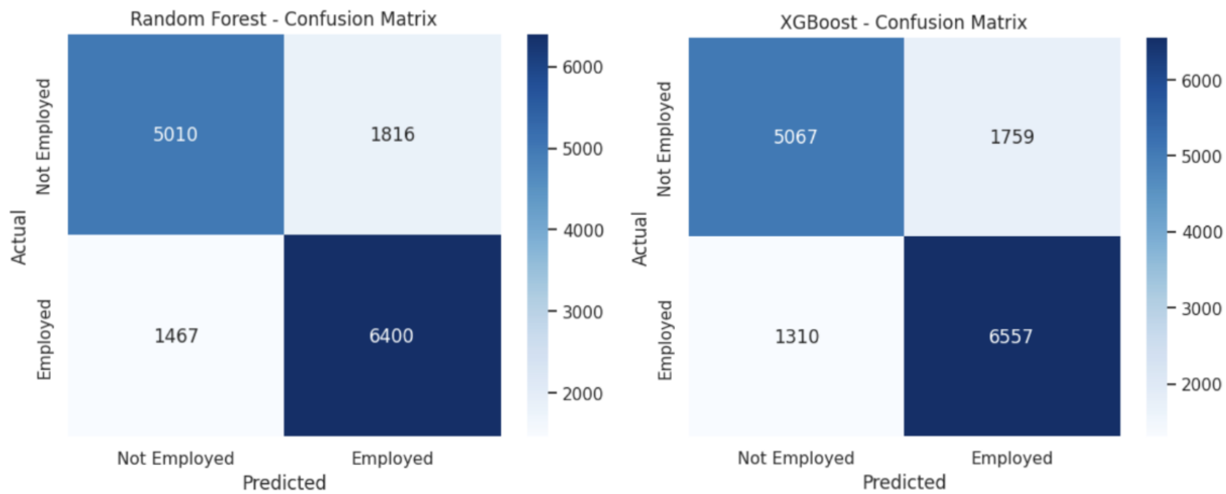


Figure 14

Confusion matrices of LightGBM and Shallow Neural Network

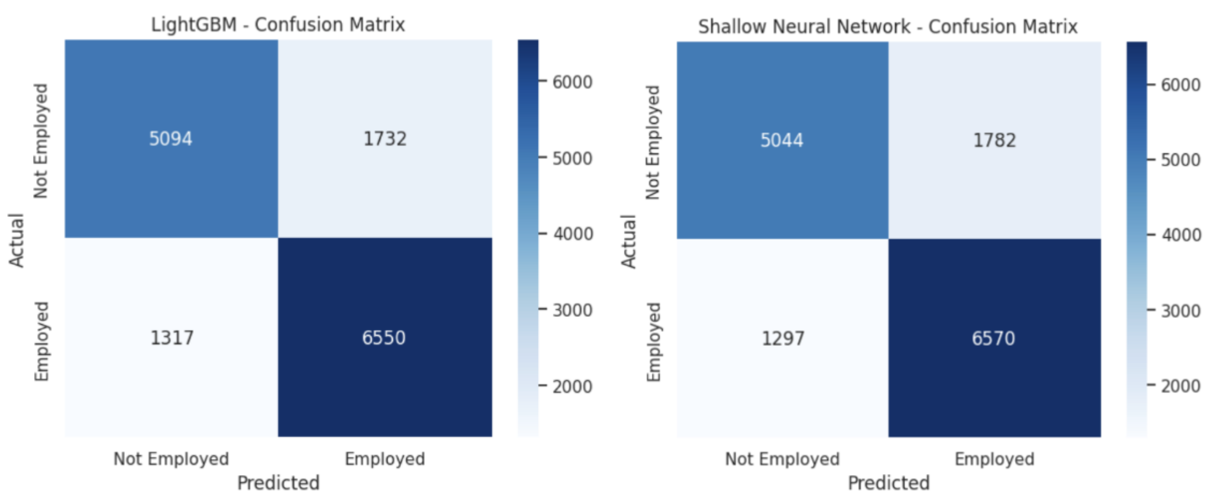
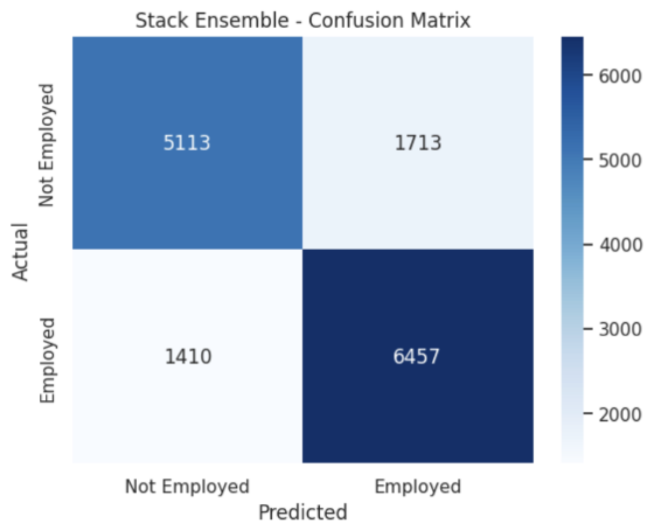


Figure 15

Confusion matrix of Stack Ensemble



ROC curve

A ROC curve that aligns closely with the diagonal line $y = x$ indicates that the false positive rate (FP) is equal to the true positive rate (TP). Therefore, a diagnostic test with acceptable accuracy is expected to have an ROC curve in the upper left triangle above the reference line $y = x$ (Hoo et al., 2017). All models have ROC curves above the line $y = x$, demonstrating their ability to distinguish between the two classes (**Figures 16 - 19**).

Figure 16

ROC Curves of Logistic Regression and Decision Tree

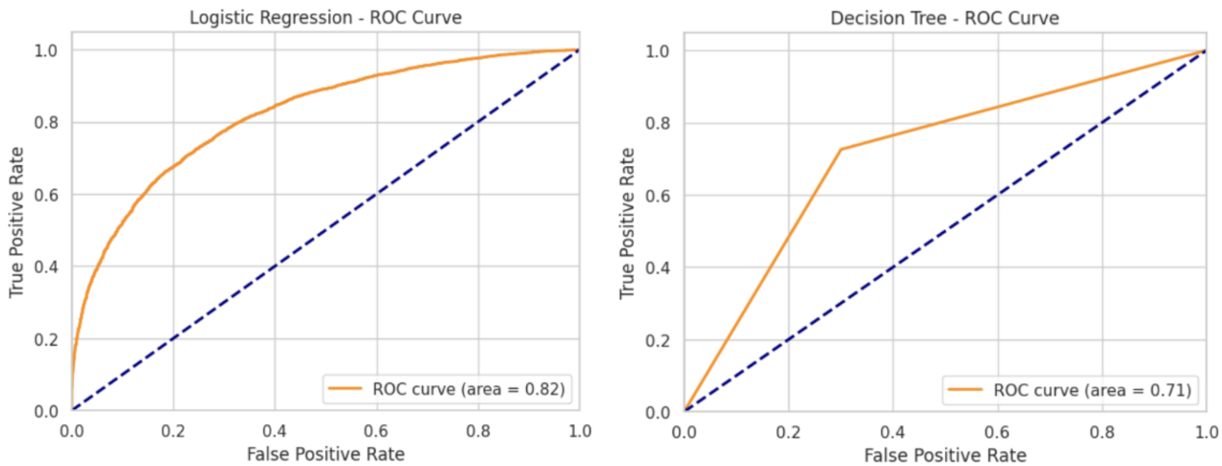


Figure 17

ROC Curves of Random Forest and XGBoost

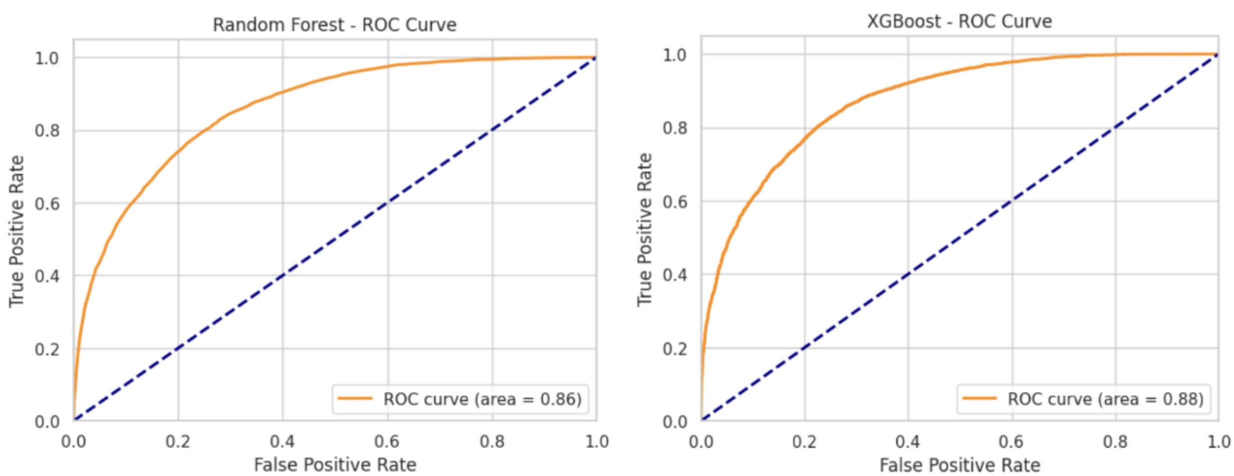


Figure 18

ROC Curves of LightGBM and Shallow Neural Network

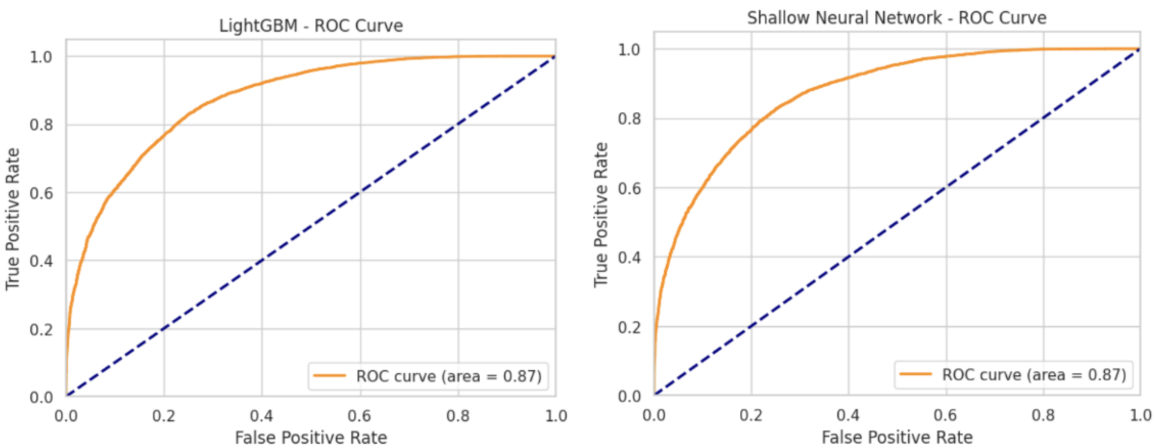
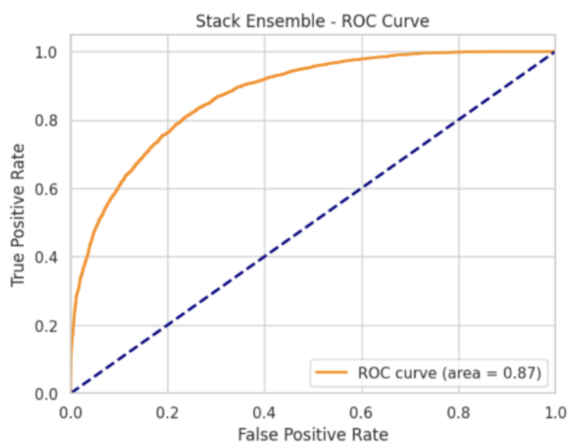


Figure 19

ROC Curve of Stack Ensemble

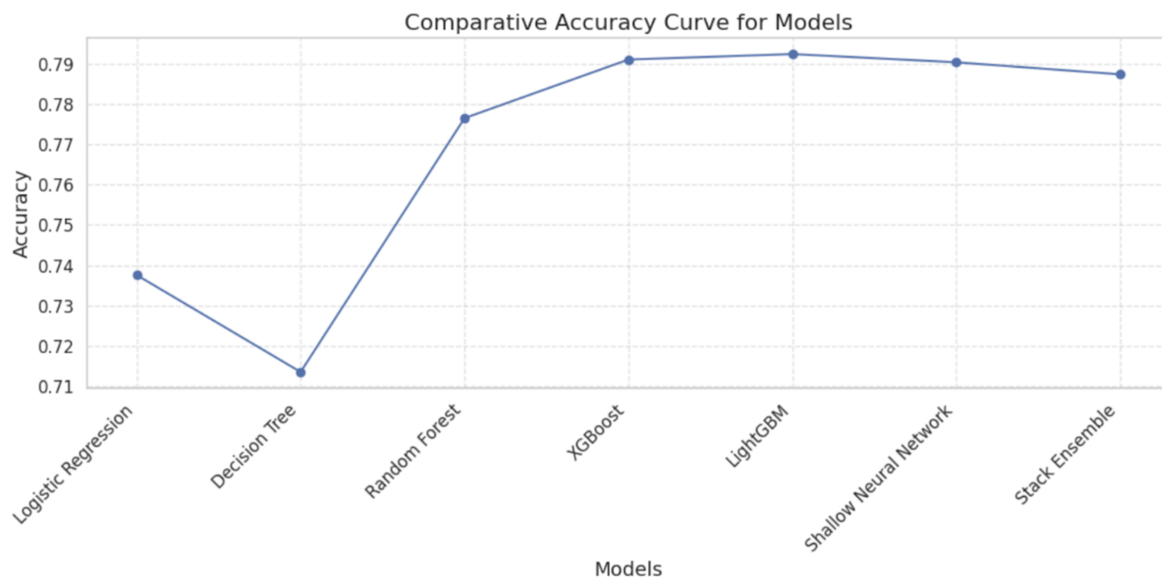


Comparative Analysis

Regarding the accuracy score (**Figure 20**), lightGBM achieves the highest accuracy (0.7925), demonstrating its strong capability to handle the complexity of the data set and effectively capture patterns in the data. The shallow neural network (0.7914) and the XG boost (0.7911) closely follow, showcasing their effectiveness as powerful models for this binary classification task. The stack ensemble (0.7874) shows solid performance. However, it does not surpass lightGBM or other standalone advanced models like shallow neural network and XG boost, suggesting that the ensemble may have yet to fully leverage its base models' strengths. The random forest (0.7766) has decent accuracy but lags behind more sophisticated models, highlighting its limitations compared to gradient-boosting and neural networks. Simpler models such as logistic regression (0.7376) and decision tree (0.7135) exhibit the lowest accuracy scores, making them less suitable for capturing complex relationships in the data set.

Figure 20

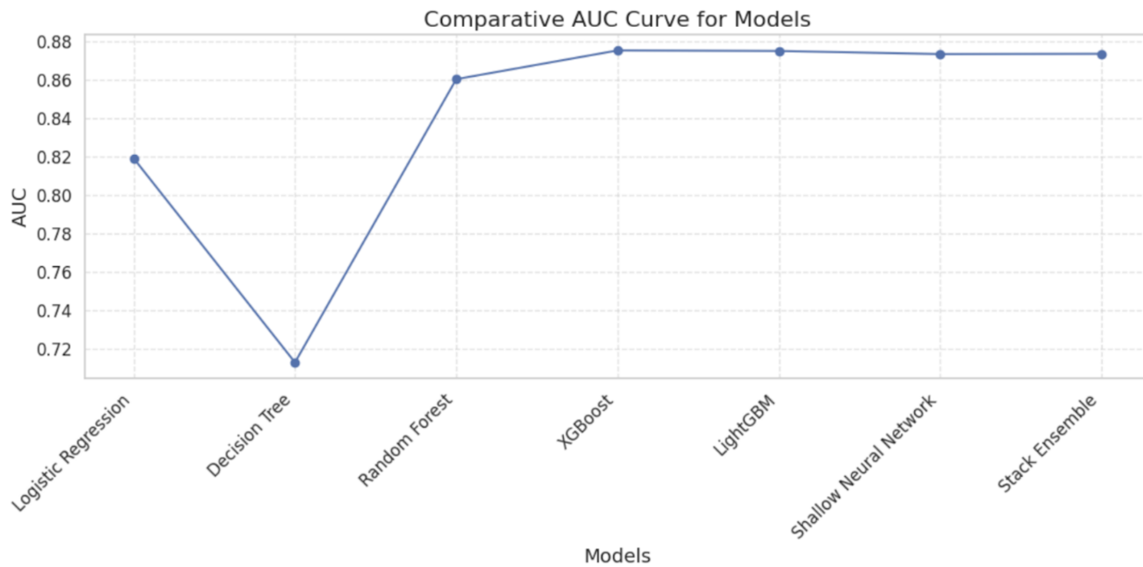
Comparative analysis of different models in terms of accuracy



Taking into account the AUC score (**Figure 21**), XG boost achieves the highest score (0.8753), demonstrating its excellent ability to distinguish between the two classes and its robustness for this classification task. LightGBM (0.8750) and the shallow neural network (0.8736) closely follow, reflecting their strong discrimination capabilities and effectiveness in capturing complex patterns in the data set. The stack ensemble (0.8735) performs nearly as well as the top three models but does not surpass them, suggesting that it combines the base models effectively, but may not fully leverage their strengths to achieve superior discrimination. Random forest (0.8603) delivers a respectable AUC score, indicating its ability to separate classes adequately but less effectively than advanced boosting or neural network models. Logistic regression (0.8190) shows an acceptable AUC score, reflecting its ability as a baseline model for binary classification, but falls short compared to more advanced algorithms. Finally, the decision tree (0.7127) ranks the lowest, highlighting its limited ability to differentiate between classes, likely due to its simplicity and tendency to overfit.

Figure 21

Comparative analysis of different models in terms of AUC



Conclusion

This research evaluates the effectiveness of various ML models in predicting employability outcomes. Using various algorithms: logistic regression, decision tree, random forest, XG boost, lightGBM, shallow neural network, and stack ensemble, we analyze their performance using Accuracy, AUC, Confusion Matrices, and ROC curves.

The results reveal that advanced models like lightGBM, XG boost, and shallow neural network consistently outperformed simpler models in accuracy and AUC metrics, highlighting their ability to capture the dataset's complex patterns. Among these, lightGBM has the highest accuracy, while XG boost achieves the best AUC, making them the most effective models for employability prediction. Although slightly behind in performance, the stack ensemble shows competitive results, indicating the potential of the ensemble methods to stabilize predictions across varied datasets. However, simpler models such as logistic regression and the decision tree are less effective, serving primarily as benchmarks for comparison.

This study emphasizes the importance of selecting advanced ML models for predictive tasks involving complex data sets. Future work could further explore hyperparameter tuning, additional ensemble methods, and feature engineering to improve predictive performance. In addition, addressing bias and fairness in these models remains crucial in ensuring equitable outcomes in real-world recruitment scenarios. The findings of this research provide a solid foundation for leveraging machine learning to optimize hiring decisions and enhance recruitment processes.

References

Ahmed, N. (n.d.). The motivation for train-test split [Retrieved from Medium]. https://medium.com/@nahmed3536/the-motivation-for-train-test-split-2b1837f596c3

Bertani, A., Di Paola, G., Russo, E., & Tuzzolino, F. (2018). How to describe bivariate data. *Journal of Thoracic Disease*, *10*(2), 1133.

Chen, T., & Guestrin, C. (2016). Xgboost: A scalable tree boosting system. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 785–794.

Delecraz, S., Eltarr, L., Becuwe, M., Bouxin, H., Boutin, N., & Oullier, O. (2022). Making recruitment more inclusive: Unfairness monitoring with a job matching machine-learning algorithm. *Proceedings of the 2nd International Workshop on Equitable Data and Technology*, 34–41.

Dhabliya, D., Dari, S. S., Dhablia, A., Akhila, N., Kachhoria, R., & Khetani, V. (2024). Addressing bias in machine learning algorithms: Promoting fairness and ethical design. *E3S Web of Conferences*, *491*, 02040.

Dobbin, K. K., & Simon, R. M. (2011). Optimally splitting cases for training and testing high dimensional classifiers. *BMC Medical Genomics*, *4*, 1–8.

Evidently AI. (n.d.). Confusion matrix [Retrieved November 25, 2024]. https://www.evidentlyai.com/classification-metrics/confusion-matrix

Faliagka, E., Ramantas, K., Tsakalidis, A., & Tzimas, G. (2012). Application of machine learning algorithms to an online recruitment system. *Proc. International Conference on Internet and Web Applications and Services*, 215–220.

Goswami, B., Bhuyan, M. K., Alfarhood, S., & Safran, M. (2024). Classification of oral cancer into pre-cancerous stages from white light images using lightgbm algorithm. *IEEE Access*, *12*, 31626–31639.

Hassan, M., & Bermak, A. (2014). Gas classification using binary decision tree classifier. *2014 IEEE International Symposium on Circuits and Systems (ISCAS)*, 2579–2582.

Hoo, Z. H., Candlish, J., & Teare, D. (2017). What is an roc curve? *Emergency Medicine Journal*, *34*(6), 357–359.

Huang, J., & Ling, C. X. (2005). Using auc and accuracy in evaluating learning algorithms. *IEEE Transactions on Knowledge and Data Engineering*, *17*(3), 299–310.

Ioffe, S., & Szegedy, C. (2015). Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1412.6980*. https://arxiv.org/pdf/1412.6980

Kebonye, N. M. (2021). Exploring the novel support points-based split method on a soil dataset. *Measurement*, *186*, 110131.

Kulkarni-Masurkar, S. (n.d.). Predictive analytics in recruitment: How models transform talent acquisition [Retrieved December 1, 2024]. https://www.linkedin.com/pulse/predictive-analytics-recruitment-how-models-talent-kulkarni-masurkar-qqwtc/

LightGBM. (n.d.). Lgbmclassifier — lightgbm 3.3.5.99 documentation [Retrieved November 24, 2024]. https://lightgbm.readthedocs.io/en/latest/pythonapi/lightgbm.LGBMClassifier.html

Narkhede, S. (2018). Understanding auc - roc curve [Retrieved November 25, 2024]. https://towardsdatascience.com/understanding-auc-roc-curve-68b2303cc9c5

Nick, T. G., & Campbell, K. M. (2007). Logistic regression. In *Topics in biostatistics* (pp. 273–301). Springer.

Njoto, S., Cheong, M., Lederman, R., McLoughney, A., Ruppanner, L., & Wirth, A. (2022). Gender bias in ai recruitment systems: A sociological-and data science-based case study. *2022 IEEE International Symposium on Technology and Society (ISTAS)*, *1*, 1–7.

Pardede, H. F., Adhi, P., Zilvan, V., Yuliani, A. R., & Arisal, A. (2023). A generalization of sigmoid loss function using tsallis statistics for binary classification. *Neural Processing Letters*, *55*(4), 5193–5214.

Parmar, A., Katariya, R., & Patel, V. (2019). A review on random forest: An ensemble classifier. *International Conference on Intelligent Data Communication Technologies and Internet of Things (ICICI) 2018*, 758–763.

Pessach, D., Singer, G., Avrahami, D., Ben-Gal, H. C., Shmueli, E., & Ben-Gal, I. (2020). Employees recruitment: A prescriptive analytics approach via machine learning and mathematical programming. *Decision Support Systems*, *134*, 113290.

Rąb-Kettler, K., & Lehnervp, B. (2019). Recruitment in the times of machine learning. *Management Systems in Production Engineering*, *27*(2), 105–109.

Reddy, D. J. M., Regella, S., & Seelam, S. R. (2020). Recruitment prediction using machine learning. *2020 5th International Conference on Computing, Communication and Security (ICCCS)*, 1–4.

Sandu, C. (n.d.). Shallow neural networks [Retrieved November 25, 2024]. https://medium.com/@calin.sandu/shallow-neural-networks-9b7a7c4ae6e8

Scikit-learn. (n.d.). Metrics and scoring: Quantifying the quality of predictions [Retrieved November 24, 2024]. https://scikit-learn.org/stable/api/sklearn.metrics.html

Snehi, M., & Bhandari, A. (2022). A novel distributed stack ensembled meta-learning-based optimized classification framework for real-time prolific iot traffic streams. *Arabian Journal for Science and Engineering*, *47*(8), 9907–9930.

Sree, K. (n.d.). Train test split and its importance [Retrieved from Medium]. https://medium.com/@kavyasree42/train-test-split-and-its-importance-f2022472382d

Tankha, A. (n.d.). 70k+ job applicants data (human resource) - unleashing employability insights: Analyzing 70k job applicants for optimal hiring [data set] [Retrieved from Kaggle]. https://www.kaggle.com/datasets/ayushtankha/70k-job-applicants-data-human-resource

TensorFlow. (n.d.). Tf.keras.sequential [Retrieved November 25, 2024]. https://www.tensorflow.org/api_docs/python/tf/keras/Sequential

Tian, X., Pavur, R., Han, H., & Zhang, L. (2023). A machine learning-based human resources recruitment system for business process management: Using lsa, bert and svm. *Business Process Management Journal*, *29*(1), 202–222.

XGBoost. (n.d.). Xgboost for binary classification [Retrieved November 24, 2024]. https://xgboosting.com/xgboost-for-binary-classification/

Yanamala, K. K. R. (2022). Dynamic bias mitigation for multimodal ai in recruitment ensuring fairness and equity in hiring practices. *Journal of Artificial Intelligence and Machine Learning in Management*, *6*(2), 51–61.