Introduction and Research Objectives

In recent years, artificial intelligence has grown at an rapid pace and has transformed industries and economies worldwide, fueled by breakthroughs in machine learning and the rise of generative models such as ChatGPT. This growth is evident in expanding scope of AI applications from natural language processing and computer vision to predictive analytics and robotics. For instance, a study by the McKinsey Global Institute suggests that "up to 800 million jobs worldwide could be displaced by automation by 2030, with approximately 375 million requiring significant retraining" (Tiwari (2023)). Similarly, the World Economic Forum projects that "AI and machine learning will create 133 million new roles by 2022 while displacing around 75 million jobs", underscoring the rapid transformation AI brings to the global workforce(World Economic Forum (2020)).

The job market today experiences AI's impact in dual ways. On one hand, automation is enhancing efficiency by taking over routine and low-skilled tasks. A survey found that nearly half (48%) of surveyed companies report that AI solutions have already replaced some worker functions to streamline operations across sectors (Manyika and Sneader (2018)). On the other hand, AI is generating new opportunities in areas such as AI development, data analytics, and digital transformation, demanding skills that align with emerging technologies. This duality underscores the importance of addressing automation risks. If left unchecked, these shifts could deepen inequality and disrupt livelihoods, particularly for workers unprepared for change. Policymakers and organizations thus face a critical imperative to invest in reskilling programs, robust policy measures, and strategic human capital development to ensure that AI-driven productivity gains benefit workers at all levels.

To explore these dynamics, this study draws on the "AI-Powered Job Market Insights" dataset from Kaggle, which includes 500 synthetic job listings designed to mirror real-world employment trends (Tharmalingam (2024)). Each listing provides detailed attributes, including Job_Title, Industry, Company_Size, Location, AI_Adoption_Level, Automation_Risk, Required_Skills, Salary_USD, Remote_Friendly, and Job_Growth_Projection. These features enable a comprehensive analysis of AI's influence on modern employment, from salary patterns and skill demands to job stability risks. With categorical indicators for AI adoption and automation risk, the dataset shows a wide view of job titles across different fields. It reveals how various roles connect with AI-driven automation.

It also allows exploration of how factors like wages, company size, and remote work policies intersect with AI use. While synthetic, the dataset reflects patterns seen in fields like technology, healthcare, and finance, making it a valuable tool for research, modeling, and scenario planning rather than direct decision-making.

This report aims to examine how AI shapes job vulnerability to automation by addressing three core questions: (1) Which roles are most likely to be automated in the near future? (2) What

specific skills or gaps in skills contribute to this likelihood? (3) How do organizational factors, such as industry type, company size, and AI adoption levels, influence this risk? This study seeks to identify where automation pressures are most intense and which workforce capabilities could reduce these risks. The goal is to inform both organizational strategy and policy decisions, helping stakeholders anticipate and adapt to a rapidly evolving employment landscape.

To achieve these objectives, the analysis employs two machine learning techniques: Random Forest and Gradient Boosting. Random Forest excels at handling diverse data types such as numerical and categorical, while minimizing overfitting through variance reduction. Gradient Boosting, meanwhile, is adept at capturing complex interactions, making it well suited for classification tasks. The report begins by introducing these methods and their strengths, followed by an explanation of how the models are trained and validated, using techniques like cross-validation and parameter tuning to optimize performance. Next, it presents the results, comparing model accuracy, feature importance, and robustness. Finally, it concludes with a discussion of each approach's benefits and limitations, offering recommendations for future applications. Through this structured approach, the study seeks to illuminate AI's transformative role in the job market and guide efforts to navigate its challenges and opportunities.

Data Understanding and Preparation

The AI-Powered Job Market Insights dataset is synthetic, consisting of 500 observations and 10 columns with no missing values. Nine of these variables are categorical, while only Salary_USD is numeric. For this study, Automation_Risk serves as the target variable in a multi-class classification task. Table 1 below is an overview of each feature, including its type, number of unique values, and example values for each column.

Variable	Type	# of Unique Values	Example Values
Job_Title	object	10	HR Manager, AI Researcher, etc.
Industry	object	10	Entertainment, Technology, Retail, etc.
Company_Size	object	3	Small, Medium, Large
Location	object	10	Dubai, Singapore, Berlin, Tokyo, etc.
AI_Adoption_Level	object	3	Low, Medium, High
Required_Skills	object	10	UX/UI Design, Marketing, etc.
Salary_USD	float64	N/A	Max: 155k, Mean: 91k, Min: 30k
Remote_Friendly	object	2	Yes, No
Job_Growth_Projection	object	3	Growth, Decline, Stable

Table 1: Summary of Variables in the Dataset

To gain a deeper understanding of the dataset, the study first plotted the distributions of all 10 variables using histograms with kernel density estimation overlays. The distribution plot shows that the target variable, Automation_Risk, is balanced among its three categories—Low, Medium, and High—which benefits this multi-class classification task. This balance ensures a fair spread for training the model across the different risk levels. The Job_Title distribution reveals a relatively balanced spread across the 10 roles, with counts ranging between 40 and 60 per role. No single role dominates, which helps avoid bias toward a specific job type in the analysis. Similarly, the Industry distribution displays an even representation across the 10 sectors, with approximately 40 to 60 occurrences per sector. This balanced distribution provides a fair comparison across industries. All other categorical variables also exhibit nearly even distributions. The only numerical variable, Salary_USD, has a range from 31K to 155K USD, a mean of 91K USD, and most salaries fall between 50K and 100K USD, with a close-to-normal distribution. Overall, the well-balanced dataset across categorical variables, along with the distribution of Salary_USD, supports a fair analysis of automation risk across different roles, skills, and organizational factors.

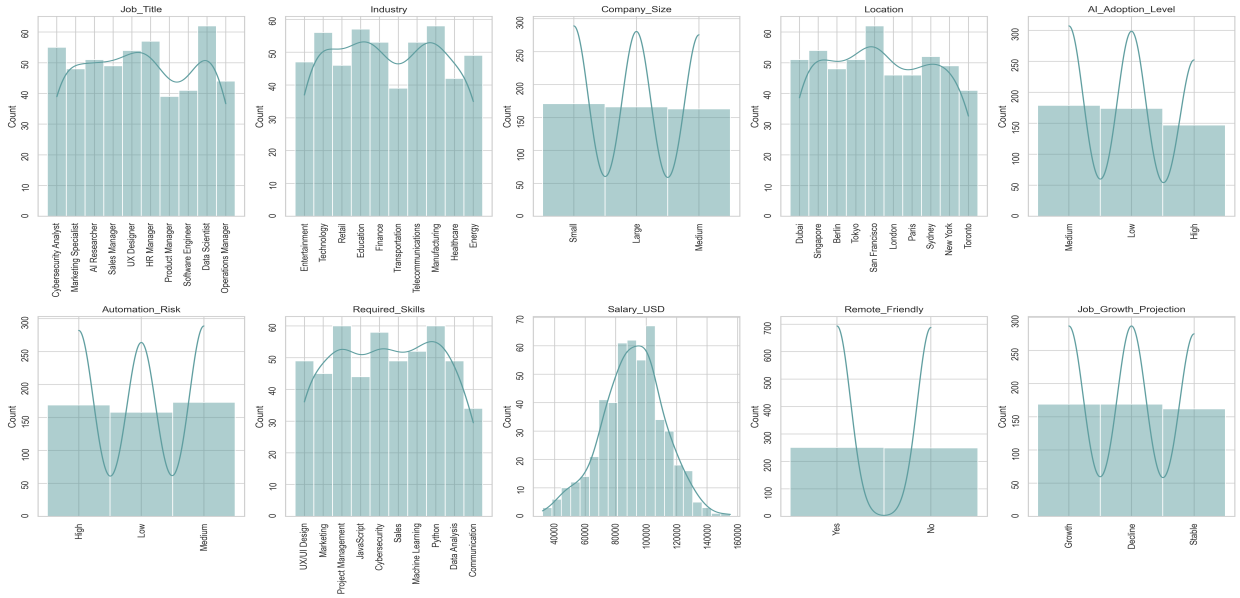

Figure 1: Distribution of all Variables

To further explore the relationship between AI adoption levels and industries, the proportion of Medium or High AI adoption for each industry was calculated by summing the counts for these two categories and dividing by the total number of jobs in that industry. The Figure 2 shows that Healthcare has the highest proportion of AI usage level, suggesting a strong emphasis on AI-based solutions for diagnostics, patient care, or administrative tasks. Retail and Technology also have high AI usage. This might come from competition to use data-driven strategies. Telecommunications has the lowest AI adoption. This may indicate slower integration or different operational

priorities. This early analysis spots where AI is most common in certain industries.

Additionally, the same imputation was applied to reveal the relationship between AI adoption level and job types. Figure 3 shows that Marketing Specialists have the highest percentage of AI-driven tasks. HR Managers, Sales Managers, and Product Managers also show relatively high AI usage, most likely reflecting the growing need for data-driven analysis in hiring, sales forecasting, and product optimization. UX Designers have the lowest proportion of Medium or High AI adoption. This may be because design roles rely heavily on artistic intuition and creativity, which can slow the integration of AI technologies into their workflow.

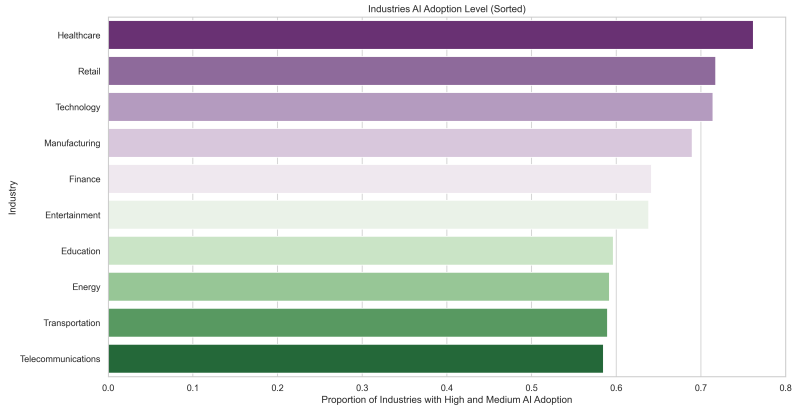

Figure 2: AI Adoption vs. Industry

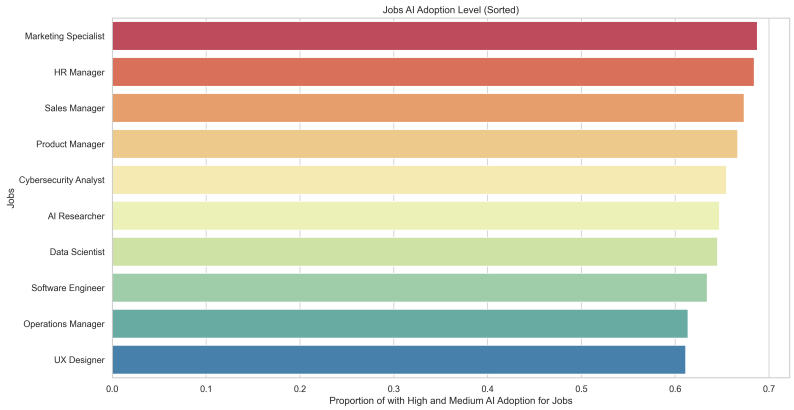

Figure 3: AI Adoption vs. Job Types

Moving forward, the relationship between automation risk and job titles was investigated to provide an intuitive view of how automation affects various roles. The analysis calculated the average salary for each job title alongside the proportion of jobs with high or medium automation risk. As shown in Figure 4, job titles are sorted by their automation risk proportion (displayed above each bar) while each job's average salary is shown on the vertical axis. It is observed that

the Operations Manager role has the highest average salary but the lowest automation risk. In contrast, more specialized roles such as UX Designer, Marketing Specialist, and HR Manager exhibit the highest automation risk. This analysis suggests that roles requiring integrated, multi-faceted skills, such as those needed to optimize processes, manage resources, and lead teams, are less likely to be replaced by AI automation. Conversely, roles focused on a narrower set of tasks appear to be comparatively easier to replace with AI automation.

Finally, an analysis was conducted to examine how skills are influenced by both automation risk and AI adoption levels. Based on Figure 5, discrepancies were observed in the influence on skills regarding automation risk and AI adoption. For example, communication skills exhibit the lowest automation risk despite showing a relatively high level of AI adoption. This difference might be due to the dataset being synthetic, each job listing contains only a single value for Required_Skills, and some listings with the same Job_Title have different skill requirements. In real-world scenarios, the same job would most likely require a similar set of skills. Nonetheless, this analysis provides insights into which skills are most vulnerable to automation, guiding further investigation into skill gaps related to automation risk.

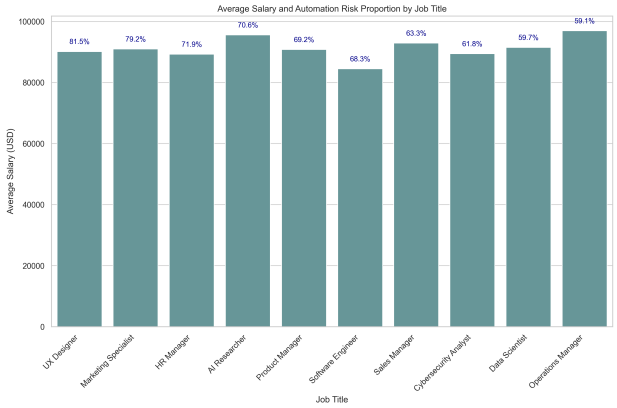

Figure 4: Job Types vs. Automation Risk

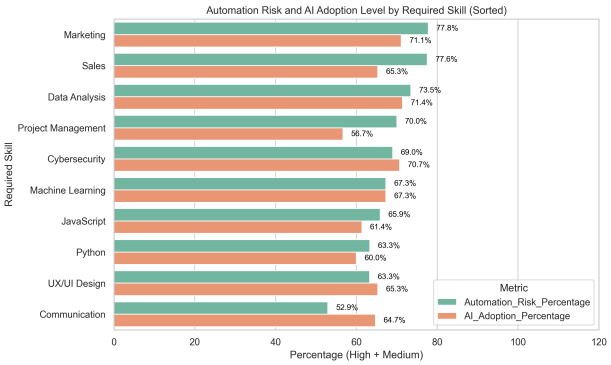

Figure 5: Skills vs. Automation Risk & AI Adoption

I began data preprocessing by focusing on the categorical variables. For features with unordered categories, "Job_Title," "Industry," "Required_Skills," and "Job_Growth_Projection", I applied one-hot encoding using pd.get_dummies with drop_first=False. For example, "Job_Title" contains 10 roles (such as Cybersecurity Analyst and Marketing Specialist), "Industry" includes 10 sectors (like Technology and Healthcare), "Required_Skills" lists 10 skills (such as UX/UI Design and Machine Learning), and "Job_Growth_Projection" offers three options (Growth, Decline, and Stable). This process created separate binary columns for each category, allowing the model to treat each one independently without implying any order. Next, I addressed features that have a natural order. For the "Company_Size" column, which contains the categories, Small, Medium, and Large, I used ordinal encoding by mapping Small to 1, Medium to 2, and Large to 3. This

approach retains the inherent order in a single numerical column, helping the model understand the progression in company size. I applied the same method to the "AI_Adoption_Level" column, mapping Low to 1, Medium to 2, and High to 3, which represents the increasing levels of AI adoption. This encoding is important for identifying patterns where higher AI adoption might lead to increased automation risk. For the "Remote_Friendly" column, I used binary encoding, mapping "No" to 0 and "Yes" to 1. This conversion turns the feature into a numerical format that the model can easily interpret, helping to determine if remote work influences automation risk. Since the analysis does not focus on geographic factors, I dropped the "Location" column. Although "Location" includes 10 different values (for example, Dubai, Singapore, and New York), it could add unnecessary complexity and noise. Removing it reduces the number of features and keeps the focus on job and company factors. Finally, I processed the target variable, "Automation_Risk," which has three categories: Low, Medium, and High. I applied label encoding by mapping Low to 0, Medium to 1, and High to 2. This conversion prepares the target variable for a multi-class classification task, allowing Random Forest and Gradient Boosting models to treat these values as separate classes. Overall, these preprocessing steps, dropping irrelevant columns, applying one-hot encoding to nominal variables, using ordinal and binary encoding for ordered and binary features, and label encoding the target, prepare the dataset effectively for modeling. The resulting df_encoded dataset has 38 columns: 37 predictors and 1 target (Automation_Risk).

Methodology

(i) **Random Forest**

Random Forest is an ensemble method built on Decision Trees. Each tree is trained on a different bootstrap sample. A bootstrap sample is created by sampling with replacement from the original dataset to form a new dataset of the same size. This adds randomness to each tree and reduces the chance that all trees will make the same mistakes.

At each split in a Decision Tree, Random Forest chooses a random subset of the total features. A common choice is $m = \sqrt{p}$ for classification tasks, where p is the total number of features. The algorithm then picks the feature and split point among those m features that best separate the data. The quality of a split is measured by minimizing Gini impurity or maximizing information gain in classification.

Splitting is a crucial step in building a classification tree. When deciding a split, the algorithm chooses which feature to split on and which threshold (or category) to use. It repeats this process until a stopping criterion is met, such as a maximum depth, a minimum number of samples, or no further improvement. Because each tree sees different subsets of features,

the trees become less correlated. This lowers the overall variance and makes the model more robust.

Mathematically, a Decision Tree partitions the training data into nodes. Each node corresponds to a subset of the data, denoted by R_m. Here, N_m is the number of training samples (observations) in region R_m. The proportion of observations from class k in node m is:

$$\hat{p}_{mk} = \frac{1}{N_m} \sum_{x_i \in R_m} \mathbf{1}(y_i = k).$$

In this expression, $\hat{p}_{mk}$ is the observed probability of class k in node m. The indicator function $\mathbf{1}(y_i = k)$ equals 1 if y_i is class k, and 0 otherwise. Summing this indicator over all observations in R_m counts how many samples in node m belong to class k. Dividing by N_m gives the fraction of samples that are in class k. The class predicted by node m is:

$$\text{class } k^* = \arg\max_k \hat{p}_{mk}.$$

This means the tree assigns all observations in R_m to the class with the highest observed proportion $\hat{p}_{mk}$. Overall, adding randomness in the features selected at each split is the key to Random Forest's power. It reduces correlation among the trees, lowers variance, and leads to stronger performance (King-Yu (2025b)).

(ii) **Gradient Boosting**

In boosting, many weak models are added sequentially, each focusing on correcting errors made by the previous models. By "learning slowly," boosting avoids aggressively fitting one large, complex model that might overfit. Instead, it gradually improves the model by iterating through multiple rounds of training (King-Yu (2025a)).

First, the algorithm starts with a empty model $f_0(x) = 0$ that predicts 0 for every input. The initial residual r_i is the actual value y_i, since no model has been fitted yet. Second, for each iteration $b = 1, 2, \ldots, B$, to fit a small regression tree $f^b(x)$ with d splits to the current residuals at first. Then, to update the model by the learning rate: $f(x) \leftarrow f(x) + \lambda\, f^b(x)$. Finally, to update the residuals by subtracting the scaled predictions: $r_i \leftarrow r_i - \lambda\, f^b(x_i)$.

The final Output of the boosted model is the sum of all the individual trees $f^1, f^2, \ldots, f^B$.

$$f_B(x) = \sum_{b=1}^{B} f^b(x).$$

Each tree is small (a weak learner), but together they form a strong predictive model (James et al. (2021)).

In conclusion, the boosting method builds a strong model by combining many small trees. Each new tree learns from the residuals, which are the mistakes the current model makes. The learning rate λ controls how fast the model adjusts, playing a key role in its performance. The trees stay small, with a depth (d) usually between 1 and 4, to avoid overfitting at each step. Since each tree focuses on the current residuals, boosting reduces errors step by step, leading to a highly accurate combined model.

(iii) Parameter Optimization

Grid search 5-fold cross validation was used to find the best set of parameters for both Random Forest and Gradient Boosting models. To make the comparison fair, I used the same parameter grid for all models. For instance, I tested the number of trees with values ([100, 200, 300, 500]) and the maximum depth with values ([3, 5, 7, 10]). This uniform approach ensure to assess each model under the same settings and compare their results directly.

Model	**Parameter Grid**	**Optimal Parameters**
Gradient Boosting	'n_estimators': [100, 200, 300, 500]	'n_estimators': [500]
	'learning_rate': [0.01, 0.05, 0.1, 0.2]	'learning_rate': [0.05]
	'max_depth': [3, 5, 7, 10]	'max_depth': [7]
	'min_samples_split': [2, 5, 10, 20]	'min_samples_split': [2]
	'min_samples_leaf': [1, 2, 4, 8]	'min_samples_leaf': [2]
	'subsample': [0.8, 0.9, 1.0]	'subsample': [0.8]
Random Forest	'n_estimators': [100, 200, 300, 500]	'n_estimators': [100]
	'max_depth': [None, 10, 20, 30]	'max_depth': [None]
	'min_samples_split': [2, 5, 10, 20]	'min_samples_split': [2]
	'min_samples_leaf': [1, 2, 4, 8]	'min_samples_leaf': [8]
	'max_features': ['log2', 'sqrt']	'max_features': ['log2']
	'max_leaf_nodes':[None, 10, 50]	'max_leaf_nodes':[10]

Table 2: Hyperparameter Grid and Optimal Parameters across Models

(iv) Feature Selection

Two methods were employed to select features based on their importance. The first method leverages the feature importance scores calculated by the best-performing Random Forest model. When a Random Forest is trained, it evaluates how much each feature contributes to

reducing the model's error. The top 15 features were selected by this method. The second method, Recursive Feature Elimination (RFE), is more iterative. It repeatedly removes the least important features until an optimal subset remains. Finally, the intersection of these two sets of selected features is used to refine the model's performance.

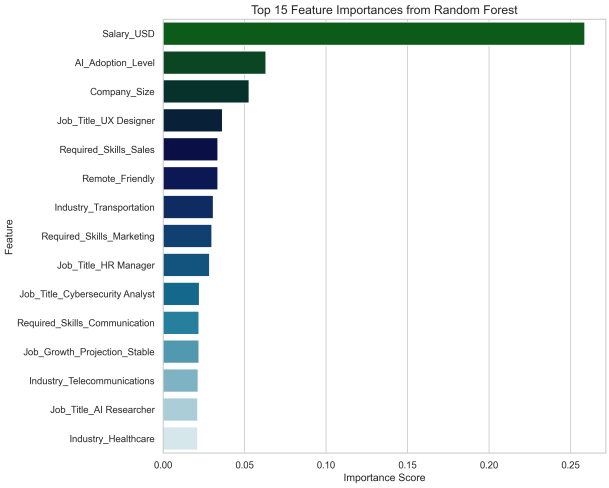

Figure 6: Feature Selection by Improtance Score

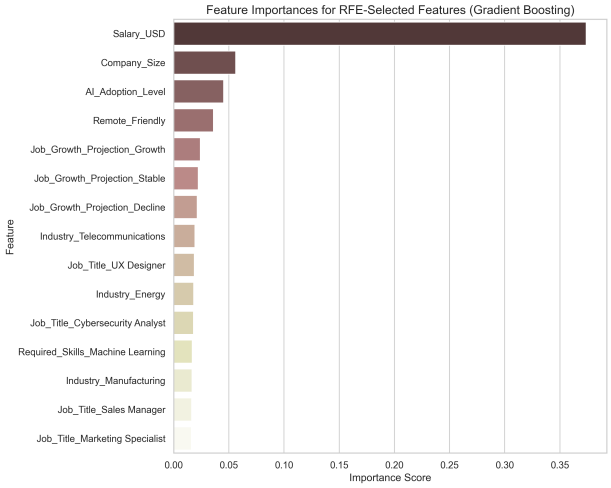

Figure 7: Feature Selection by RFE

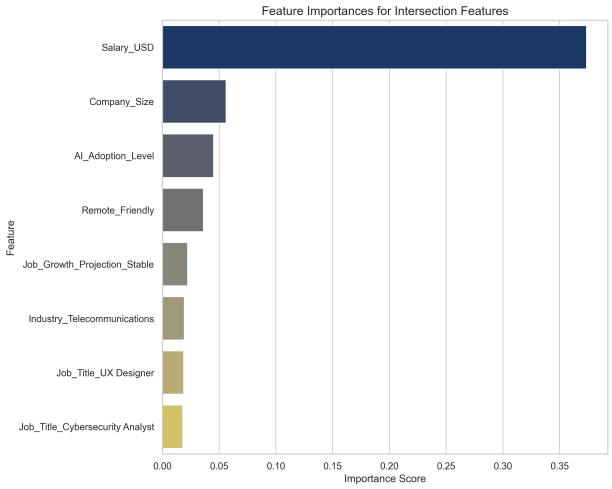

Figure 8: Intercection of Selected Features

These three plots above show how two feature selection methods, Random Forest–based importance and Recursive Feature Elimination (RFE) with Gradient Boosting, rank the predictors of automation risk. In all three plots, Salary_USD stands out as the most important factor. This suggests that a job's pay level might be closely tied to its chance of being automated. Company_Size and AI_Adoption_Level also rank high in all plots. This shows that

company factors, like the size of the firm and how much AI it uses, play a big role in automation risk. Other factors, such as Remote_Friendly, Job_Growth_Projection categories, and some Job_Title or Industry labels, have middle to lower importance scores. This means they have a smaller effect on automation risk. By using the intersection of features chosen by both methods, the study keeps only the factors that both agree are important. This approach aims to make the model stronger. It ensures the final set of features includes the most dependable predictors of automation risk.

Results

The performance of the four models, RF_base, RF_feature, GB_base, and GB_feature, was assessed using three metrics: accuracy, Adjusted Rand Index (ARI), and macro F1-score. Figure 9 and Table 3 present these results, comparing baseline models (base) with versions refined through feature selection (feature) for both Random Forest (RF) and Gradient Boosting (GB). The analysis aims to identify the best-performing model and determine if feature selection improves performance.

Accuracy results show GB_feature with the highest score at 0.40, followed by RF_feature at 0.39. Both RF_base and GB_base have an accuracy of 0.38. This indicates that feature selection slightly boosts accuracy for both RF and GB models. However, all models' accuracy remains low, just above random guessing (0.33 for a 3-class problem). This suggests that noise in the synthetic dataset may still limit performance. Next, ARI scores measure how well predicted labels match true labels. RF_feature achieves the highest ARI at 0.222. In contrast, GB_feature has the lowest ARI at -0.008, meaning its predictions are worse than random guessing. For macro F1-scores, which balance precision and recall across all classes, RF_feature leads with 0.379.

When comparing all models, RF_feature performs the best overall. Although GB_feature has the highest accuracy, its negative ARI score suggests it predicts some classes well but misclassifies others more often. The low performance across all models (accuracy ranging from 0.38 to 0.40) indicates that noise in the synthetic dataset, as noted earlier, restricts predictive power. Still, the improvement from feature selection shows that focusing on key predictors like Salary_USD, AI_Adoption_Level, and Company_Size slightly enhances performance.

Model	**Accuracy**	**ARI**	**F1**
RF_base	0.38	0.020	0.371
RF_feature	0.39	0.022	0.379
GB_base	0.38	0.020	0.371
GB_feature	0.40	-0.008	0.395

Table 3: Comparison of different models and their performance metrics.

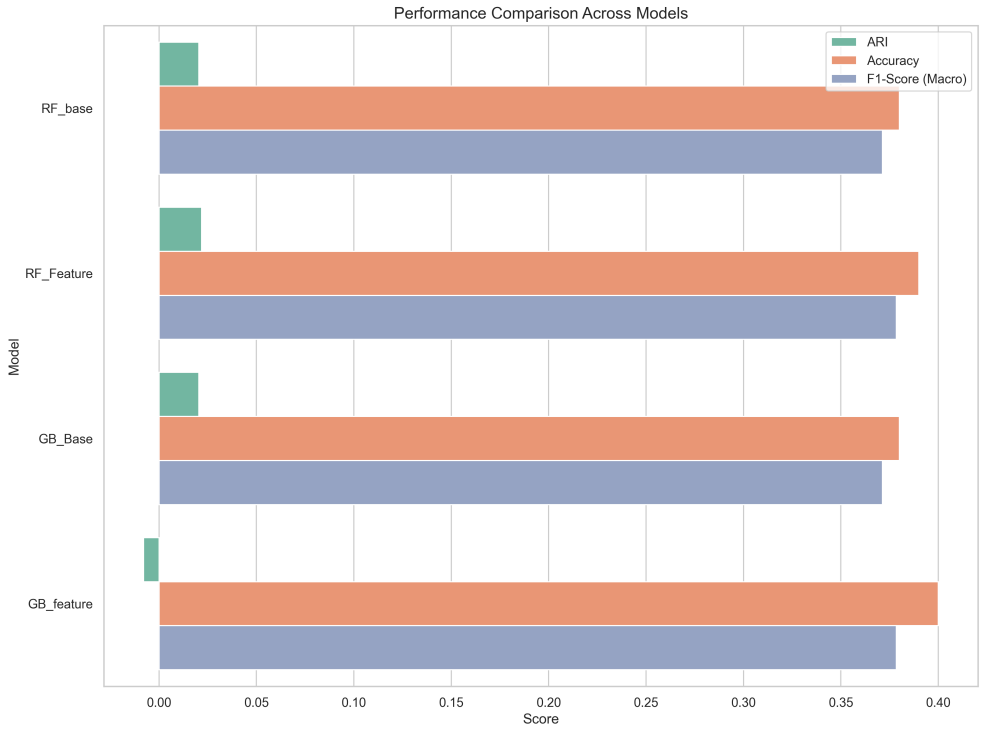

Figure 9: Performance Comparison across All Models

Conclusion and Future Work

By carefully analyzing this synthetic dataset, the study aimed to find out which areas are most affected by automation and to identify the skills that can help lower these risks. This information is meant to guide organizations and policymakers as they plan for a fast-changing job market.

The analysis answers three main questions: (1) Which roles are most likely to be automated soon? (2) What specific skills or skill gaps make roles more likely to be automated? (3) How do factors like industry type, company size, and AI adoption levels affect this risk?

For the first question, the analysis found that roles such as UX Designer and Cybersecurity Analyst are among those most likely to be automated. These roles consistently appeared in the intersection of selected features with moderate importance scores in both the Random Forest and Gradient Boosting models. This agrees with earlier findings that show a high automation risk for UX Designers (81.5%), suggesting that jobs with routine or data-driven tasks are at higher risk.

For the second question, certain skills like Marketing and Sales emerged as important predictors of automation risk. In the Random Forest model, these skills had importance scores around 0.08 to 0.10. This suggests that jobs requiring these skills face higher automation pressure because they involve repetitive, data-driven tasks.

For the third question, organizational factors such as AI adoption level, company size, and

salary were consistently the top predictors across all models. This indicates that companies with higher AI adoption, larger sizes, and higher salary levels are more likely to automate roles.

The refined models, which used selected features, showed a slight improvement over the baseline models. The refined Random Forest model achieved the highest Adjusted Rand Index (ARI) of 0.222 and a good F1-score of 0.379. The refined Gradient Boosting model had the highest accuracy of 0.40 and an F1-score of 0.395, although its negative ARI (-0.008) suggests some issues with how well the predicted labels match the true labels. Despite these improvements, the overall performance remained low, possibly due to noise in the synthetic dataset that limited the models' predictive power.

In conclusion, both Random Forest and Gradient Boosting models performed poorly with this synthetic dataset, even after careful data preprocessing, hyperparameter tuning, cross-validation, and feature selection. However, the intersection of features selected by the RF and RFE methods provided useful insights. The study found that roles in companies with high salaries, large sizes, and advanced AI adoption face intense automation pressure, while skills like communication may be automated slowly.

There are several limitations and opportunities for future research. First, the synthetic dataset introduced noise that affected model performance (with accuracy between 0.38 and 0.40). Future studies should use real-world data to validate these findings and improve accuracy. Second, further feature engineering such as creating interaction terms between AI adoption level and specific skills could improve model performance. Third, exploring additional models like XGBoost or neural networks might yield better results, especially with larger datasets. Finally, extending the analysis to specific industries or regions may reveal more detailed patterns, helping policymakers and organizations tailor their strategies to address automation challenges effectively. These steps would build on the current findings and offer deeper insights into how AI influences job vulnerability.

References

James, G., Witten, D., Hastie, T., and Tibshirani, R. (2021). *An Introduction to Statistical Learning*. Springer, 2nd edition.

King-Yu, S. (2025a). Statistical learning lecture 6: Ensemble learning part i – boosting. Lecture Slides, STATS/CSE 790.

King-Yu, S. (2025b). Statistical learning lecture 8: Ensemble learning part iii – random forests. Lecture Slides, STATS/CSE 790.

Manyika, J. et al. (2017). Jobs lost, jobs gained: What the future of work will mean for jobs, skills, and wages. Technical report, McKinsey Global Institute.

Manyika, J. and Sneader, K. (2018). AI, Automation, and the Future of Work: Ten Things to Solve For. Technical report, McKinsey Global Institute. Executive Briefing.

Tharmalingam, L. (2024). Ai powered job market insights. `https://www.kaggle.com/datasets/uom190346a/ai-powered-job-market-insights/data`.

Tiwari, R. (2023). The impact of ai and machine learning on job displacement and employment opportunities. `https://www.researchgate.net/publication/367254440_The_Impact_of_AI_and_Machine_Learning_on_Job_Displacement_and_Employment_Opportunities`.

World Economic Forum (2020). The future of jobs report 2020. Technical report, World Economic Forum. WEF.