Hiring With Artificial Intelligence

Noah Minard, Subash Adusumilli, Joel Chaple, Lewi Anamo

Introduction

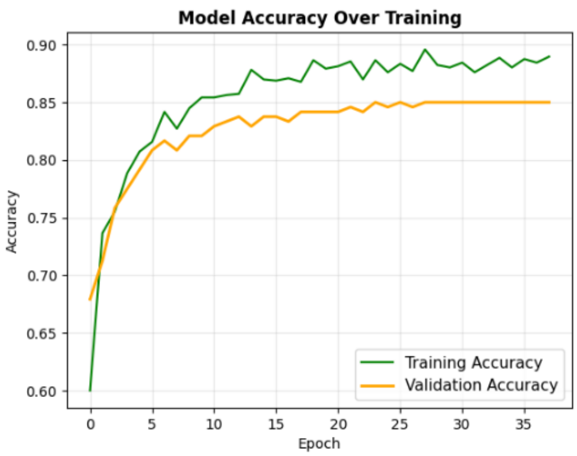
In a country where work defines the lifestyle of nearly every American, securing a job that fits the needs of an individual is crucial. That's why millions of applications cross desks every year, as candidates are evaluated for whether they match the job requirements. The wave of applicants for almost every available position makes hiring a difficult process for many companies to handle, which has incited a search for machine-based solutions. Machine learning promises fast and precise application review but has detrimental implications if it were to mess up or produce biased results. The question we are asking is: how well can AI select the best job candidate(s) from applications and interviews? What models are the most accurate in doing so?

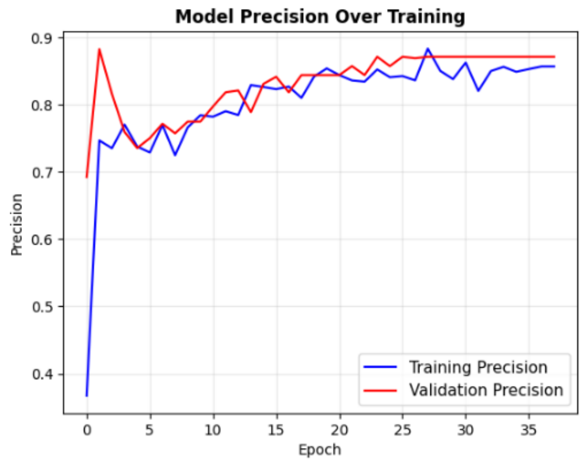
Methods

To answer these questions, we used a recruitment dataset that held comprehensive data, totaling fifteen hundred entries with eleven features. Some of the data would be ethical to use in the hiring process, and some would be unethical, giving us the opportunity to evaluate whether the model or data was biased based on the results. The Python tools used throughout the course of the investigation were Pandas, NumPy, Scikit-learn, Matplotlib, while using singular functions from packages like TensorFlow and Seaborn. We cleaned the data and configured it with an 80/20 train-test split to make it easier for our models to process, and tested four different models against the dataset. The two models that we wrote the code for were the logistic regression model and the neural network model. When testing popular Large Language Models (LLMs) against our data, ChatGPT used a random forest model, and Google Gemini used a gradient boosting classifier to come up with the most accurate results.

Results

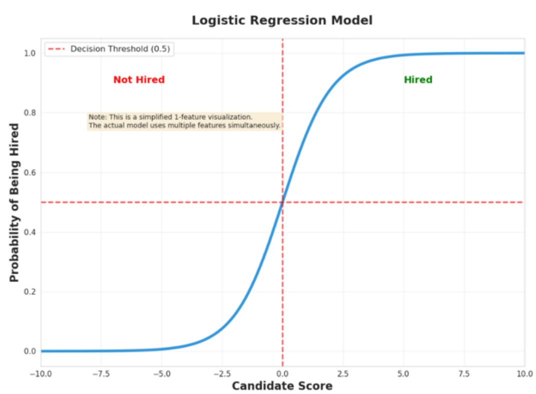
The first model we tested was the neural network model. Its architecture consisted of a rectified linear unit activation on the hidden layers, a sigmoid activation on the output layers, and dropout regularization to prevent overfitting. The result was an accuracy score of 87.33%, with a precision score of 84.81%, as shown in the figures below.

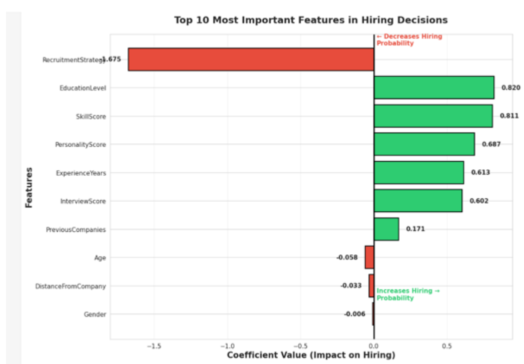




The figure on the left shows the accuracy progression after each epoch for the neural network model, and the figure on the right shows the progression of the model's precision after each epoch.

The next model evaluated was the logistic regression model. Using a standard Scikit-learn function, we were able to derive an accuracy score with the dataset of 86.7%. This test determined that the most indicative features of success in hiring were the education level, skill score, and personality score. These results are reflected in the figures below.





The figure on the left shows the probability of being hired against the candidate score calculate by the modely. The figure on the right shows the list of the features and how impactful they were on the model.

The final models that we tested were the popular LLMs, which were represented by ChatGPT and Google Gemini. ChatGPT used a random forest model that yielded an accuracy score of 93%, with a precision score of 93%. Google Gemini used a gradient boosting classifier that achieved an accuracy score of 93%, with a precision score of 94%. The results they displayed after their analysis can be seen in the figures below.

Classification Report

markdown

```
              precision    recall  f1-score   support

    ... 300
weighted avg       0.93      0.93      0.93       300
```

This indicates high performance across both classes with strong precision and recall.

Classification Report

Here is the performance breakdown for the Gradient Boosting model:

Class	Precision	Recall	F1-Score	Support
Not Hired (0)	0.92	0.98	0.95	207
Hired (1)	0.94	0.82	0.87	93
Accuracy			0.93	300

The figure on the left shows the results from ChatGPT's random forest model, while the figure on the right shows the results for Google Gemini's gradient boosting classifier.

Conclusion

Overall, the models we tested proved to be quite accurate and precise. The question this inevitably raises is this: how accurate should a model be before it is implemented in actual hiring? There is a moral dilemma lying underneath every attempt to truncate the recruitment process using machine learning. What if the models get it wrong? What if they are trained with biased data? Though we did not encounter any bias in our data or models, that is simply because the data we used for training was relatively free from prejudiced hiring practices. Though these models were effective in their predictions, we would still say that a 7% margin for error is too high for it to be implemented in hiring practices. There is still plenty of work to be done in machine learning before it can be trusted with a process that affects people's livelihoods so deeply.