Hiring with Artificial Intelligence

Group 10: Noah Minard, Subash Adusumilli, Joel Chaple, Lewi Anamo

Driving Questions:

"How well can AI select the best job candidate(s) from applications and interviews?"

"What models are the most accurate in doing so?"

Motivation:

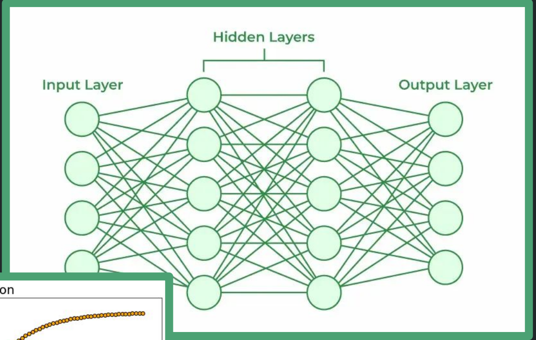
- Millions of job applications are handled every year in the U.S.
- Candidate assessment processes are often time + resource consuming
- AI/ML tools promise fast and consistent hiring decisions
- Can these tools make accurate and unbiased choices in hiring?

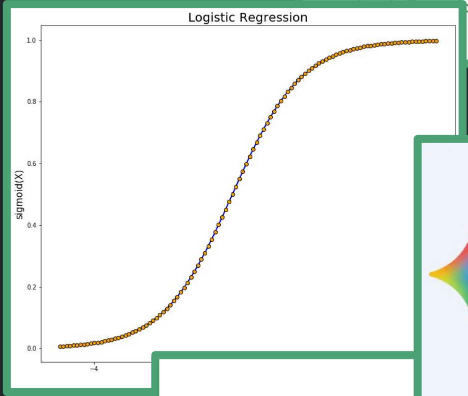
Methods:

Clean Data *Test Models* *Analyze Results*

Models evaluated include:

- Neural Network Model
- Logistic Regression Model
- Popular LLMs





Basic Dataset Information

Strengths:

- ❖ Dataset from Kaggle
- ❖ 11 comprehensive features and over 1500 entries
- ❖ Binary hiring decision feature to assist our model training
- ❖ Contains data that would be both ethical and unethical to use in hiring

Weaknesses:

- ❖ Origins/motivations of data are not clear
- ❖ Uses scores that may not be practical for real companies to use

	Age	Gender	EducationLevel	ExperienceYears	PreviousCompanies	DistanceFromCompany
0	26	1	2	0	3	26.783828
1	39	1	4	12	3	25.862694
2	48	0	2	3	2	9.920805
3	34	1	2	5	2	6.407751
4	30	0	1	6	1	43.105343
5	27	0	3	14	4	31.706659
6	48	0	2	6	1	17.291229
7	40	0	4	13	3	10.586811
8	26	1	3	6	5	28.774864
9	45	1	2	2	5	30.195964

InterviewScore	SkillScore	PersonalityScore	RecruitmentStrategy	HiringDecision
48	78	91	1	1
35	68	80	2	1
20	67	13	2	0
36	27	70	3	0
23	52	85	2	0
54	50	50	1	1
24	52	64	3	0
6	3	92	3	0
80	78	51	1	1
92	16	94	3	0

Data Splitting + Cleaning

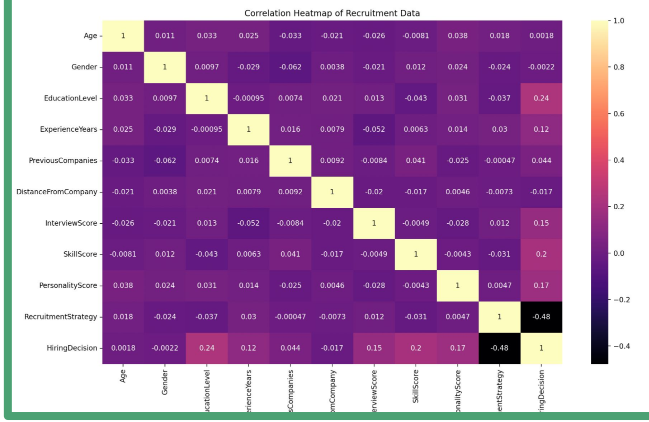
Python Tools Used:

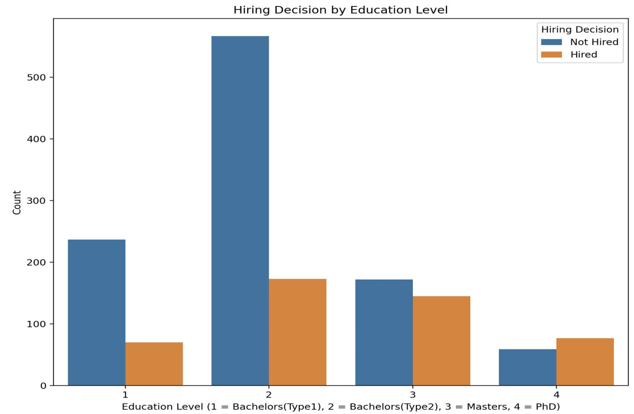
❖ Pandas, train_test_split, accuracy_score, matplotlib

Key Results:

❖ **Functional Datasets:** The split data is being properly used and is being currently improved to best aid models in learning and prediction accuracy

❖ **Visualization:** We see that education level is one of the most important features for predicting a hiring decision





Neural Network Model Methods

Model Architecture:

- Deep Neural Network: Input -> 64 -> 32 -> 16 neurons -> Output
- ReLU activation (hidden layers), Sigmoid activation (output layer) (output gives us a probability between 0 to 1)
- Dropout regularization (30%, 30%, 20%) to prevent overfitting

Training Configuration

- Adam optimizer with binary cross entropy loss (Automatically adjusts learning rate for each parameter)
- 80/20 train-test split with 20% validation set (To test for overfitting during training)
- Early stopping: Stops training when validation loss stops improving for a certain number of epochs
- Reduces learning rate for faster and efficient convergence when learning is not improving
- Batch size: 32, Max epochs: 100

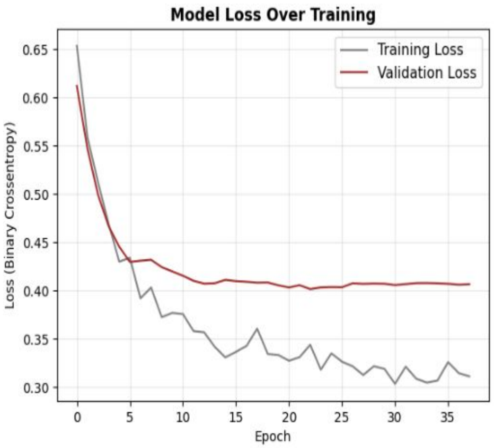
Layer (type)	Output Shape	Param #
hidden_layer_1 (Dense)	(None, 64)	704
dropout_1 (Dropout)	(None, 64)	0
hidden_layer_2 (Dense)	(None, 32)	2,080
dropout_2 (Dropout)	(None, 32)	0
hidden_layer_3 (Dense)	(None, 16)	528
dropout_3 (Dropout)	(None, 16)	0
output_layer (Dense)	(None, 1)	17

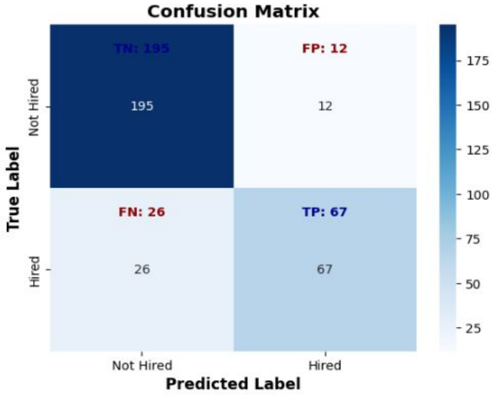
Total params: 3,329 (13.00 KB)

Trainable params: 3,329 (13.00 KB)

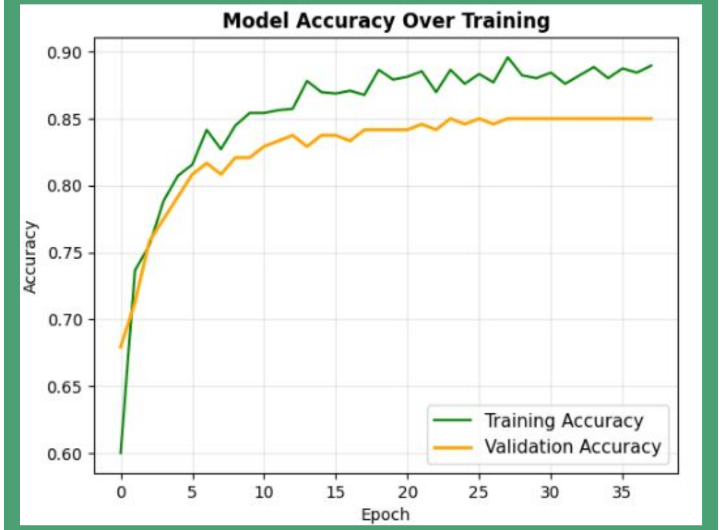
Non-trainable params: 0 (0.00 B)

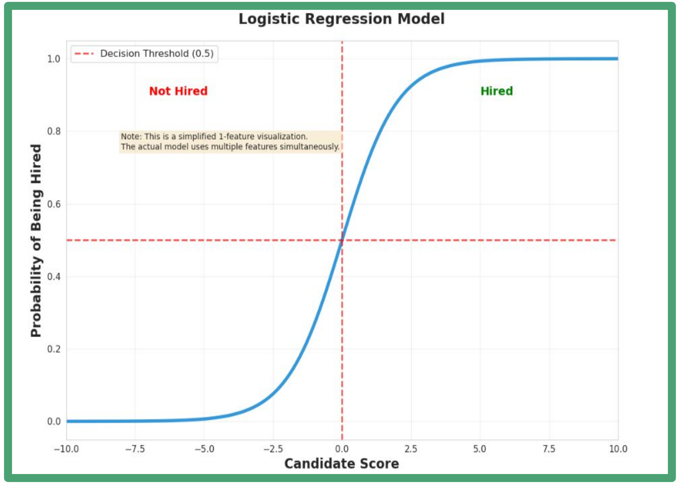
Neural Network Model Results

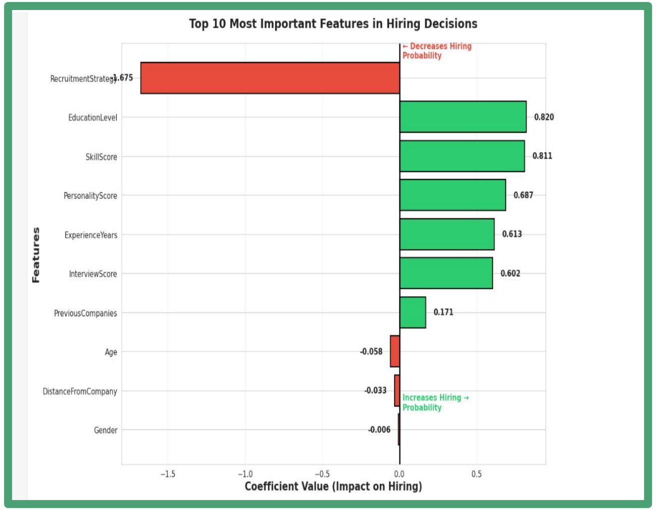




- ❖ Test Accuracy: 87.33%
- ❖ Precision: 84.81%
- ❖ Recall: 72.04%
- ❖ F1-Score: 0.7790
- ❖ ROC AUC: 0.8949
- ❖ Strong generalization with minimal overfitting







Logistic Regression Model

Brief Description:

❖ Logistic regression predicts hiring outcomes (hired/not hired) by calculating probabilities from candidate features using a sigmoid function.

Python Tools Used:

❖ pandas, numpy, scikit-learn, matplotlib

Key Results:

❖ Accuracy: **86.7%** (AUC = 0.905 - excellent discrimination)

❖ Top Predictors: EducationLevel (+0.82), SkillScore (+0.81), InterviewScore (+0.60)

ChatGPT

- Model used: Random Forest Model
- Split training and testing data for quality analysis
- Prompts discouraged use of logistic regression or neural network models

Classification Report

markdown   Copy code

```
             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.

Gemini

- Model used: Gradient Boosting Classifier
- Also attempted random forest model, but had lower accuracy than ChatGPT
- Prompts discouraged use of logistic regression or neural network models

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

Discussion of Results

Models ranked in order of accuracy (based on accuracy score):

1. Google Gemini's *Gradient Boosting Classifier* (**93%**, higher precision)

2. ChatGPT's *Random Forest Model* (**93%** , lower precision)

3. *Neural Network Model* (**87.3%**)

4. *Logistic Regression Model* (**86.7%**)

Exploring Further

- ❖ According to the results, the data appears to be free from bias. Would any of these models be acceptable if it were trained with biased data?
- ❖ What level of accuracy should a model have before being implemented in real hiring?
- ❖ Which characteristics of a person's application should be taken into account by machine learning?
- ❖ How can we hold models accountable to make sure they make accurate predictions and smart decisions?