

# **Employee360: Predicting Promotions in the Workplace**



## Project Objective:

To build a machine learning system that can predict:

- Whether an employee is likely to be promoted in the upcoming cycle



## Tools & Tech Stack:

- Python (Jupyter Notebook )
- pandas, numpy, seaborn, matplotlib
- scikit-learn, xgboost, joblib
- imbalanced-learn tensorflow



# Dataset Overview:

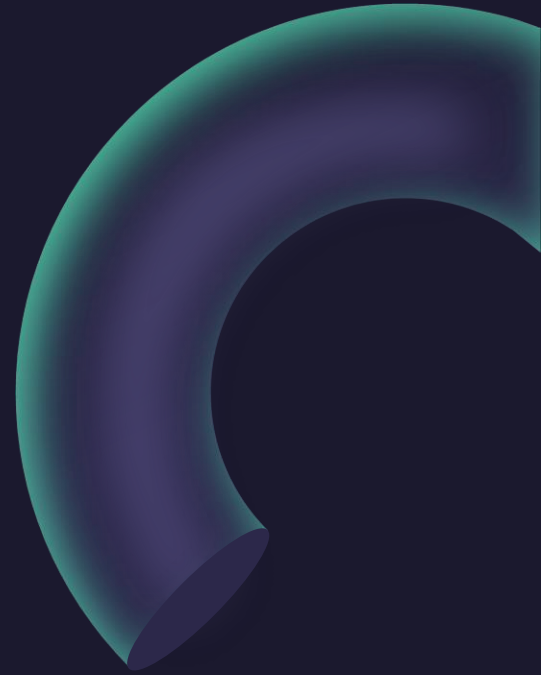
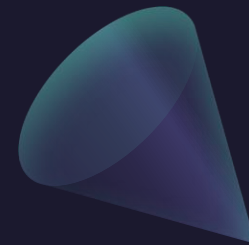
- **Entries:** 54,809
- **Columns:** 14
  - employee\_id
  - department
  - region
  - education
  - gender
  - recruitment\_channel
  - no\_of\_trainings
  - age
  - previous\_year\_rating
  - length\_of\_service
  - KPIs\_met >80%
  - awards\_won?
  - avg\_training\_score
  - is\_promoted (*label*)





# Project Workflow:

- Imports
- Load Dataset
- Data Preprocessing
- Feature Engineering
- Feature Selection
- Data Splitting and Balancing
- Scaling
- Model Training and Evaluation



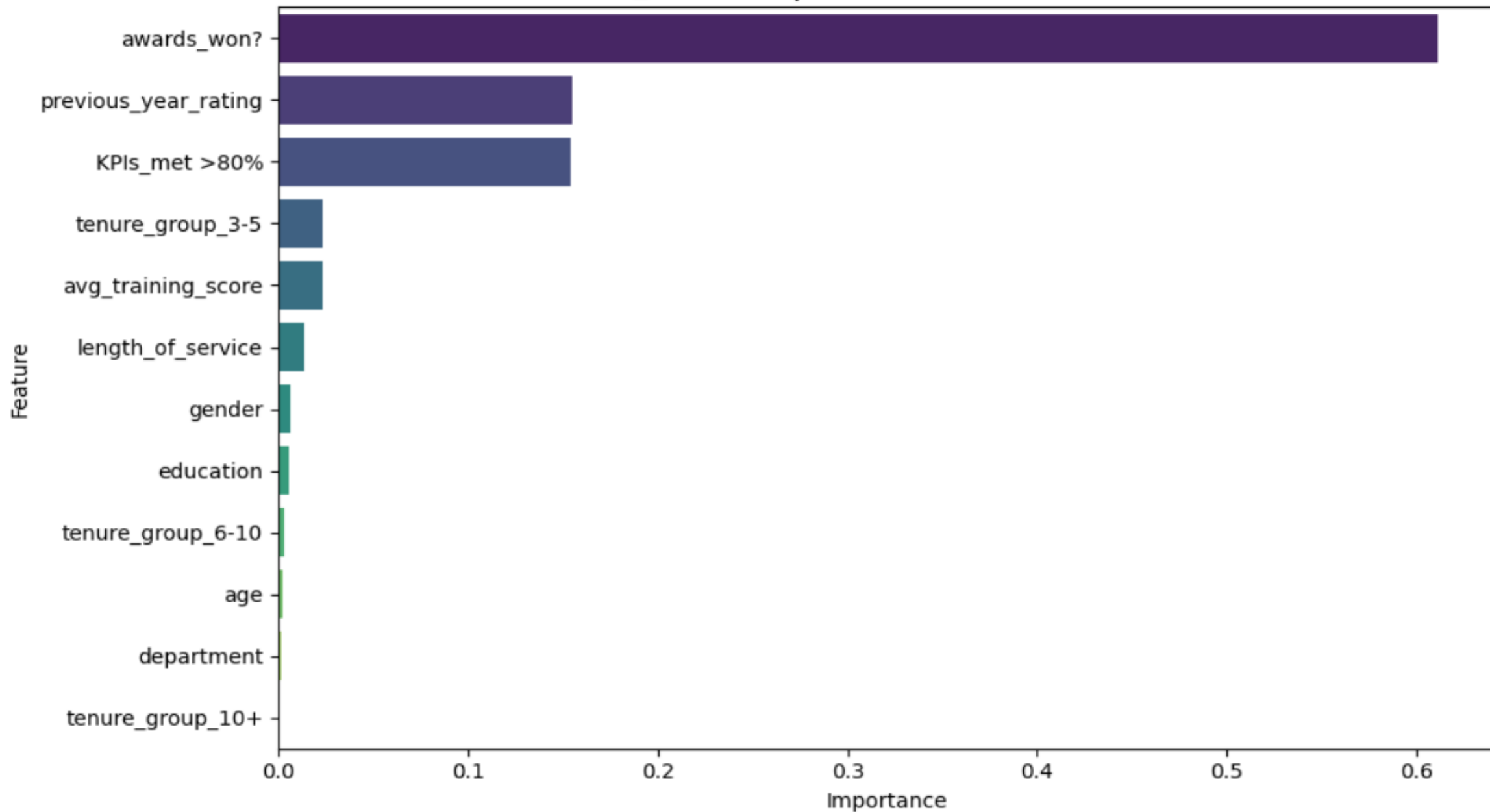


# Models:

- Random Forest
- XGBoost
- DNN Dropout
- DNN BatchNorm

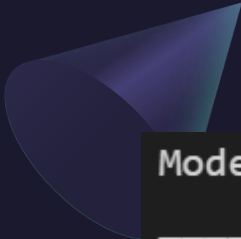


Feature Importances (Random Forest)





# DNN Dropout Summary




```
Model: "sequential"
```

Layer (type)	Output Shape	Param #
=====		
dense (Dense)	(None, 64)	832
-----		
dropout (Dropout)	(None, 64)	0
-----		
dense_1 (Dense)	(None, 32)	2080
-----		
dropout_1 (Dropout)	(None, 32)	0
-----		
dense_2 (Dense)	(None, 1)	33
=====		

```
Total params: 2,945
```

```
Trainable params: 2,945
```

```
Non-trainable params: 0
```







# DNN

## BatchNorm

## Summary

Model: "sequential\_1"

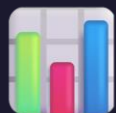
Layer (type)	Output Shape	Param #
=====		
dense_3 (Dense)	(None, 128)	1664
-----		
batch_normalization (Batch Normalization)	(None, 128)	512
-----		
re_lu (ReLU)	(None, 128)	0
-----		
dropout_2 (Dropout)	(None, 128)	0
-----		
dense_4 (Dense)	(None, 64)	8256
-----		
batch_normalization_1 (Batch Normalization)	(None, 64)	256
-----		
re_lu_1 (ReLU)	(None, 64)	0
-----		
dropout_3 (Dropout)	(None, 64)	0
-----		
dense_5 (Dense)	(None, 1)	65
=====		

Total params: 10,753

Trainable params: 10,369

Non-trainable params: 384





# Evaluation Metrics:

- . Precision
- . Recall
- . F1 Score
- . AUC-ROC



# Random Forest Report

	precision	recall	f1-score	support
0	0.95	0.93	0.94	10197
1	0.93	0.95	0.94	9859
accuracy			0.94	20056
macro avg	0.94	0.94	0.94	20056
weighted avg	0.94	0.94	0.94	20056

# XGBoost Report

	precision	recall	f1-score	support
0	0.91	0.89	0.90	10197
1	0.89	0.90	0.90	9859
accuracy			0.90	20056
macro avg	0.90	0.90	0.90	20056
weighted avg	0.90	0.90	0.90	20056

# DNN Dropout Report

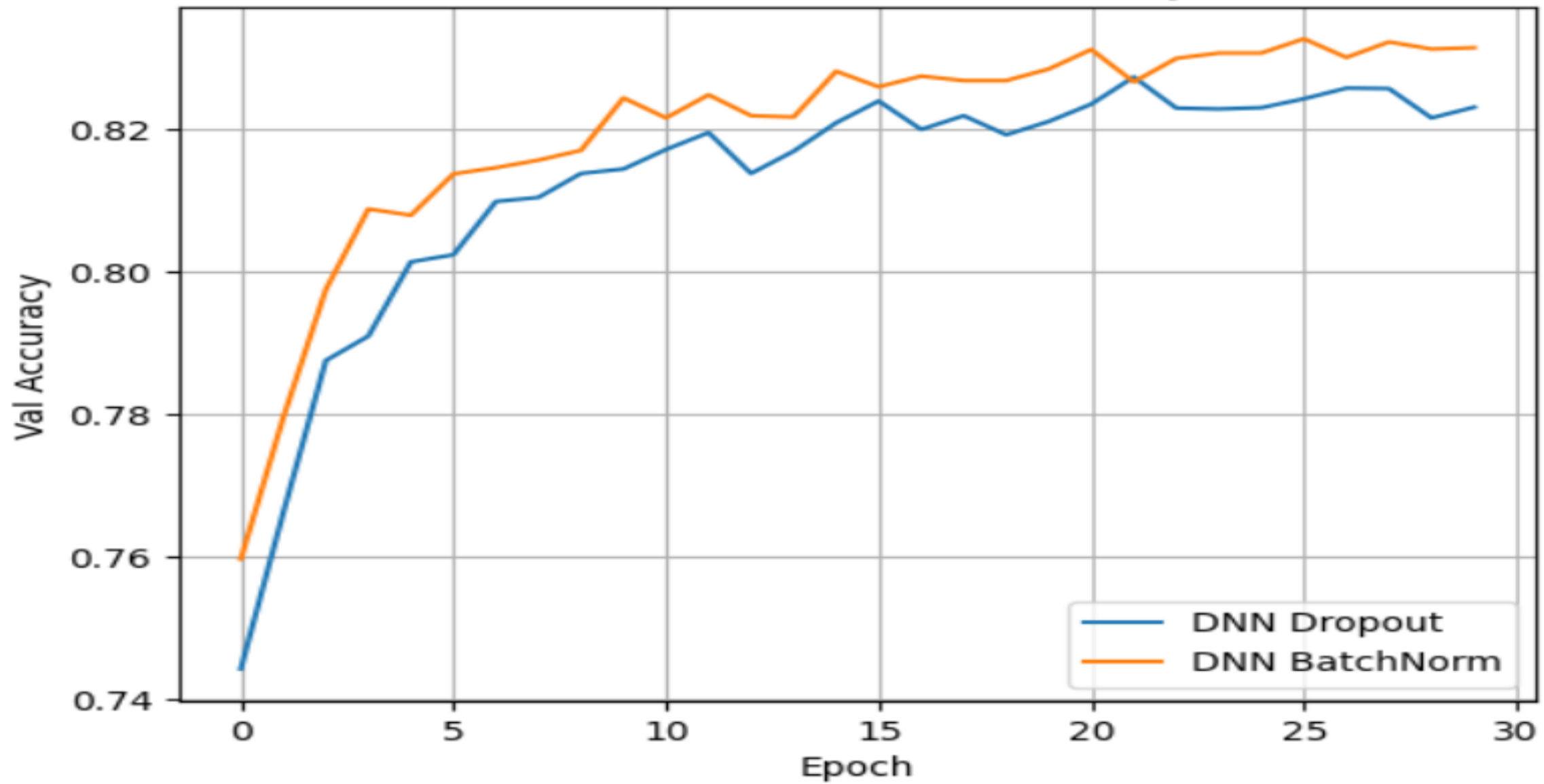
	precision	recall	f1-score	Support
0	0.92	0.71	0.80	10197
1	0.76	0.94	0.84	9859
accuracy			0.82	20056
macro avg	0.84	0.82	0.82	20056
weighted avg	0.84	0.82	0.82	20056

# DNN BatchNorm Report

	precision	Recall	f1-score	support
0	0.89	0.75	0.82	10197
1	0.78	0.90	0.84	9859
accuracy			0.83	20056
macro avg	0.83	0.83	0.83	20056
weighted avg	0.84	0.83	0.83	20056



DL Model Validation Accuracy



# Model Comparison:

Random Forest	Accuracy: 0.9432	ROC AUC: 0.9434
XGBoost	Accuracy: 0.8979	ROC AUC: 0.8980
DNN Dropout	Accuracy: 0.8193	ROC AUC: 0.8212
DNN BatchNorm	Accuracy: 0.8272	ROC AUC: 0.8284

# Code and Data

- Code: <https://github.com/pratik-a/Employee-Promotion-Prediction/blob/main/Employee360.ipynb>
- Data: [HR Analytics: Employee Promotion Datahttps://www.kaggle.com/datasets/arashnic/hr-ana?select=train.csv](https://www.kaggle.com/datasets/arashnic/hr-ana?select=train.csv)





# Future Improvements:

- Hyperparameter Tunning
- Deployment
- Attrition Prediction





# References:

- Kaggle
- sklearn
- TensorFlow



# Thank you

Pratik Awasthi

[awasthipratik.98@gmail.com](mailto:awasthipratik.98@gmail.com)

