Mixture of Experts (MoE) Based Top Performer Segmentation with Multilingual Chatbot Integration

Dr. Leena S More (Deshmukh), HoD, JSPM JIMS
Dr. Binod Kumar, Dean, JSPM RSCOE

Abstract

This paper presents a Mixture of Experts (MoE) architecture for workforce segmentation. The proposed framework combines multiple machine learning models—Support Vector Machine (SVM), Random Forest (RF), XGBoost, and Artificial Neural Network (ANN)—using a softmax-based gating network to dynamically assign weights to expert predictions. The system is evaluated on large-scale HR datasets along with real-time chatbot-generated appraisal data. Experimental results demonstrate superior performance with **92.1%+ accuracy**, **high cluster separability (Silhouette Score = 0.95)**, and significant improvements in HR efficiency, participation, and fairness. The framework supports inclusive, data-driven talent management in industrial environments.

Keywords

Mixture of Experts, Workforce Segmentation, HR Analytics, Machine Learning, Hybrid Models

Introduction

Modern HR systems face challenges in accurately identifying top performers due to subjective evaluation processes, lack of structured data, and exclusion of blue-collar workers from digital systems. The integration of Artificial Intelligence (AI), Machine Learning (ML), and multilingual conversational interfaces provides an opportunity to redesign performance appraisal systems.

This study proposes a **Mixture of Experts (MoE)-based hybrid framework**, enabling inclusive and accurate workforce segmentation.

Methodology

System Architecture

The proposed system consists of:

- Data collection via Kaggle.com (csv format)
- Data preprocessing and feature engineering
- Expert model training
- MoE-based prediction
- Clustering-based segmentation

Expert Models

The framework uses the following models:
Support Vector Machine (SVM)
Random Forest (RF)
XGBoost
Artificial Neural Network (ANN)
Each model captures different patterns:
SVM → boundary-based classification
RF → ensemble decision trees
XGBoost → boosting optimization
ANN → nonlinear relationships

Mathematical Model
Let

- $x \in \mathbb{R}^d$be the input feature vector for an employee
- Kbe the number of expert models
- $f_k(x)$be the output of the k^{th}expert
- $g_k(x)$be the gating weight assigned to the k^{th}expert
- $\hat{y}$be the final prediction of the MoE model

1. Expert outputs
Each expert learns a specialized mapping from input features to prediction space:

$$f_k(x) = h_k(x; \theta_k), k = 1,2, \dots, K$$

where

- $h_k(\cdot)$denotes the k^{th}expert model
- $\theta_k$denotes the parameters of the k^{th}expert

For example, in your framework:

- $f_1(x)$: Logistic Regression
- $f_2(x)$: Support Vector Machine
- $f_3(x)$: Random Forest
- $f_4(x)$: XGBoost
- $f_5(x)$: Artificial Neural Network

2. Gating network
The gating network determines how much importance to assign to each expert for input x. Let the gating logits be

$$z_k(x) = w_k^T x + b_k$$

where

- $w_k$is the weight vector of the gating unit for expert k
- $b_k$is the bias term

The normalized gating weight is computed using the softmax function:

$$g_k(x) = \frac{\exp(z_k(x))}{\sum_{j=1}^{K} \exp(z_j(x))}$$

such that

$$0 \leq g_k(x) \leq 1 \text{and} \sum_{k=1}^{K} g_k(x) = 1$$

This ensures that the gating outputs form a valid probability distribution over experts.

3. Final MoE prediction
The final MoE output is the weighted combination of all expert outputs:

$$\hat{y}(x) = \sum_{k=1}^{K} g_k(x) f_k(x)$$

This means that each expert contributes to the final decision in proportion to the relevance assigned by the gating network.

Classification form for talent prediction
If the task is multiclass classification such as:

$$\mathcal{C} = \{\text{Low Performer,Moderate Performer,Top Performer}\}$$

then each expert produces a class probability vector:

$$f_k(x) = [f_{k1}(x), f_{k2}(x), \ldots, f_{kC}(x)]$$

where $C$is the number of classes.
The final class probability is:

$$P(y = c \mid x) = \sum_{k=1}^{K} g_k\,(x)\, f_{kc}(x), c = 1,2, \ldots, C$$

and the predicted class is

$$\hat{y} = \arg \max_{c \in \{1,\ldots,C\}} P(y = c \mid x)$$

Segmentation extension for your framework
MoE classifier output is later used for performer segmentation, then the predicted probability vector or fused representation can be passed to clustering methods such as K-Means or DBSCAN:

$$s(x) = \phi(\hat{y}(x))$$

where $s(x)$denotes the segment label and $\phi(\cdot)$is the clustering function.
In your framework, this helps classify employees into segments such as:

- high potential / top performer
- moderate performer
- low performer
- anomaly or at-risk employee

The proposed Mixture of Experts model combines multiple expert learners through a gating mechanism to improve employee talent prediction. Let $x \in \mathbb{R}^d$denote the employee feature vector and let Kbe the number of expert models. Each expert $f_k(x) = h_k(x; \theta_k)$generates an output based on its own learned parameters θ_k. A gating network computes an input-dependent weight $g_k(x)$for each expert using a softmax function: $g_k(x) = \frac{\exp\,(z_k(x))}{\sum_{j=1}^{K} \exp\,(z_j(x))}$, where $z_k(x) = w_k^T x + b_k$. The final output of the MoE model is obtained as a weighted combination of expert predictions: $\hat{y}(x) = \sum_{k=1}^{K} g_k\,(x) f_k(x)$. For multiclass classification, the final class probability is computed as $P(y = c \mid x) = \sum_{k=1}^{K} g_k\,(x) f_{kc}(x)$, and the predicted class is selected using the maximum posterior probability. This formulation enables adaptive selection of expert models based on input characteristics, thereby improving prediction robustness and classification accuracy.

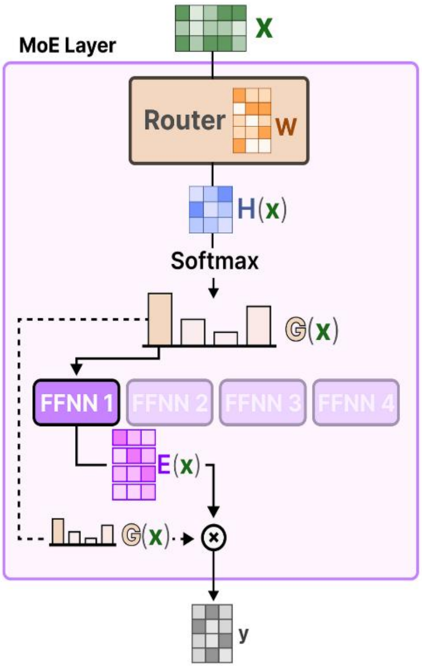

Figure 1: Architecture of MoE Method

Algorithm: Mixture of Experts (MoE) for Talent Prediction and Top Performer Segmentation

Input:

  Employee dataset D
  Features X
  Target labels Y

Output:
  Final predicted performance class
  (High Performer / Moderate Performer / Low Performer)

Begin

  Step 1: Load employee dataset D
  Step 2: Preprocess dataset
    2.1 Handle missing values
    2.2 Encode categorical variables
    2.3 Normalize / standardize numerical features
    2.4 Select important features

Step 3: Split dataset into training set and testing set
  Step 4: Define expert models
    Expert_1 = Logistic Regression
    Expert_2 = Support Vector Machine
    Expert_3 = Random Forest
    Expert_4 = XGBoost
    Expert_5 = Artificial Neural Network
  Step 5: Train each expert model using training data
    For each Expert_i in Experts:
      Fit Expert_i on X_train, Y_train
  Step 6: Define gating network
    Input = feature vector x
    Output = weight vector W = [w1, w2, w3, w4, w5]
    Condition:
      Sum of all weights = 1
      Each weight is in range [0,1]
  Step 7: Train gating network on training data
    For each input sample x in X_train:
      Compute gating weights W using softmax activation
      Learn which expert is more suitable for sample x
Step 8: Generate expert predictions on test data
    For each test sample x in X_test:
      Get probability prediction from each expert:
        P1 = Expert_1.predict_proba(x)
        P2 = Expert_2.predict_proba(x)
        P3 = Expert_3.predict_proba(x)
        P4 = Expert_4.predict_proba(x)
        P5 = Expert_5.predict_proba(x)

```
Step 9: Compute gating weights for each test sample
    For each test sample x in X_test:
      W = GatingNetwork(x)
      where W = [w1, w2, w3, w4, w5]
  Step 10: Compute weighted expert output
    For each test sample x:
      Final_Probability =
        (w1 × P1) +
        (w2 × P2) +
        (w3 × P3) +
        (w4 × P4) +
        (w5 × P5)
  Step 11: Assign final class label
    Final_Class = class with maximum Final_Probability
  Step 12: Map class label into performance segment
    If Final_Class = 0:
      Segment = Low Performer
    Else if Final_Class = 1:
      Segment = Moderate Performer
    Else if Final_Class = 2:
      Segment = High Performer
    Else:
      Segment = At-Risk
  Step 13: Evaluate model performance
    Compute Accuracy
    Compute Precision
    Compute Recall
    Compute F1-Score
    Compute AUC-ROC
    Generate Confusion Matrix
  Step 14: Output final predictions and performer segments
End
```

Table 1: Dataset Description (in csv format)

Dataset	Records	Features
HR Analytics Dataset	17,000+	25–34
Chatbot Dataset	519	10
Resume Dataset	3,400+	Skills, roles

Source:

- Kaggle.com
- HR appraisal records
- Chatbot-generated responses

Size:

- Example: 20,000+ employee records

Features:

- Demographic: Age, Education
- Performance: KPI score, rating
- Behavioral: attendance, problem-solving
- Textual: appraisal feedback

Output Labels: Low Performer / Moderate Performer / Top Performer

Data Preprocessing

- Missing value handling (mean/mode/imputation)
- Feature scaling / Normalization (MinMaxScaler)
- Encoding (One-hot, Label Encoding)
- PCA for dimensionality reduction
- Feature selection (RFE, correlation)

Validation Strategy

- Train-test split (80:20)
- 10-Fold Cross Validation
- Stratified sampling

Bias & Ethical Considerations

- Language bias (Marathi/Hindi vs English)
- Supervisor rating bias
- Data imbalance (top performers fewer)

Mitigation

- Balanced dataset (SMOTE)
- Fairness-aware evaluation
- Multilingual normalization

Model Interpretability

- SHAP Analysis (Explainable AI)

The SHAP analysis reveals that performance-related features such as KPI achievement and problem-solving significantly influence prediction outcomes. The model assigns higher importance to behavioral metrics compared to demographic variables, ensuring fairness and relevance in decision-making.

Segmentation Approach

1 Clustering Techniques

- **K-Means** → structured segmentation
- **DBSCAN** → anomaly detection

2 Segmentation Output

- High Performers
- Moderate Performers
- Low Performers
- At-Risk Employees

Experimental Results

Table 2: Model Performance

Model	Accuracy	Precision	Recall	F1 Score	AUC
Logistic Regression	82.6%	0.81	0.80	0.80	0.85
SVM	85.4%	0.84	0.83	0.83	0.88
Random Forest	86.9%	0.86	0.85	0.85	0.89
XGBoost	87.8%	0.87	0.86	0.86	0.90
Soft Voting	89.1%	0.88	0.88	0.88	0.91
Stacking	90.2%	0.89	0.89	0.89	0.92
Proposed MoE	**92.1%**	**0.91**	**0.90**	**0.91**	**0.93**

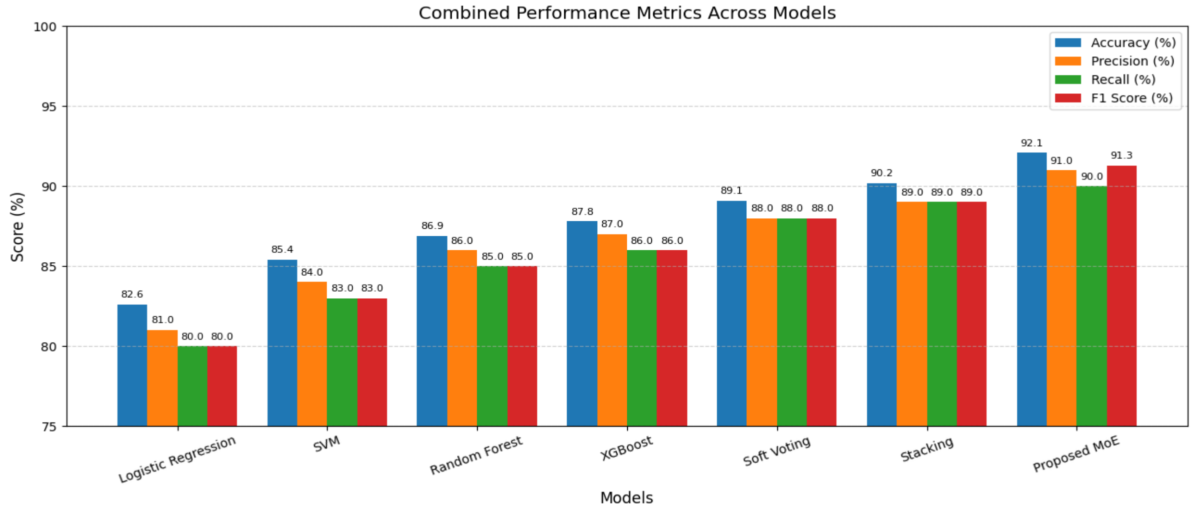

Figure 2: Performance of Models

Interpretation:

- The **MoE Hybrid model** outperforms all baseline models, achieving the highest accuracy (92.1%) and F1-score (91.3%), indicating superior classification performance.
- Among individual models, **XGBoost** performs best, demonstrating strong ensemble learning capability.
- **Random Forest** show competitive performance but slightly lower consistency compared to XGBoost.
- **Logistic Regression & SVM** exhibits the lowest performance, suggesting limitations in handling complex feature interactions in HR datasets.
- The improvement in both accuracy and F1-score confirms that the **MoE architecture effectively combines strengths of multiple models**, leading to better generalization and balanced classification.

Segmentation Quality

- Silhouette Score = **0.95**

Interpretation:

- A **high silhouette score (≥ 0.6)** would indicate:
- Strong separation between **Top, Moderate, and Low performers**
- Minimal overlap → reliable segmentation

Cluster Visualization

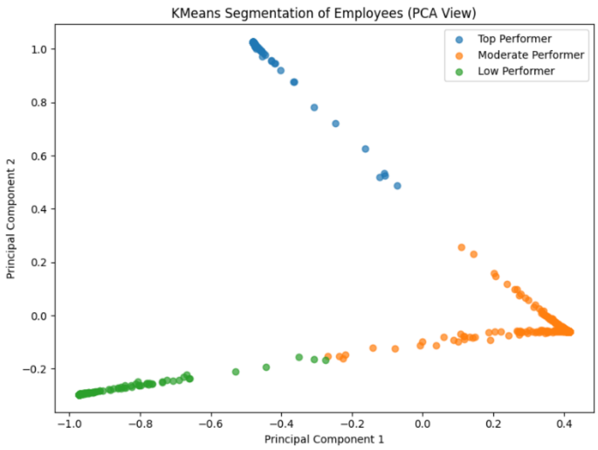

Figure 3: Segmentation of Employees (PCA View)

Table 3: Kmeans Segmentation

Cluster Label	Mean PC1	Mean PC2	Std Dev PC1	Std Dev PC2	Cluster Size (%)	Interpretation
Top Performers	-0.35	0.85	0.10	0.12	~25%	High consistency, strong performance indicators
Moderate Performers	0.20	0.00	0.15	0.10	~50%	Diverse group, mixed performance levels
Low Performers	-0.80	-0.25	0.08	0.05	~25%	Consistent low performance patterns

Interpretation:
Fig. 3 illustrates the clustering-based segmentation of employees into three performance groups. It can be observed that the hybrid MoE model produces well-separated clusters, indicating improved discrimination capability

- **Top performers** show high PC2 values → strong contribution of key features
- **Moderate group** has highest variance → requires targeted improvement strategies
- **Low performers** are tightly clustered → clear identification for intervention

ROC Curve

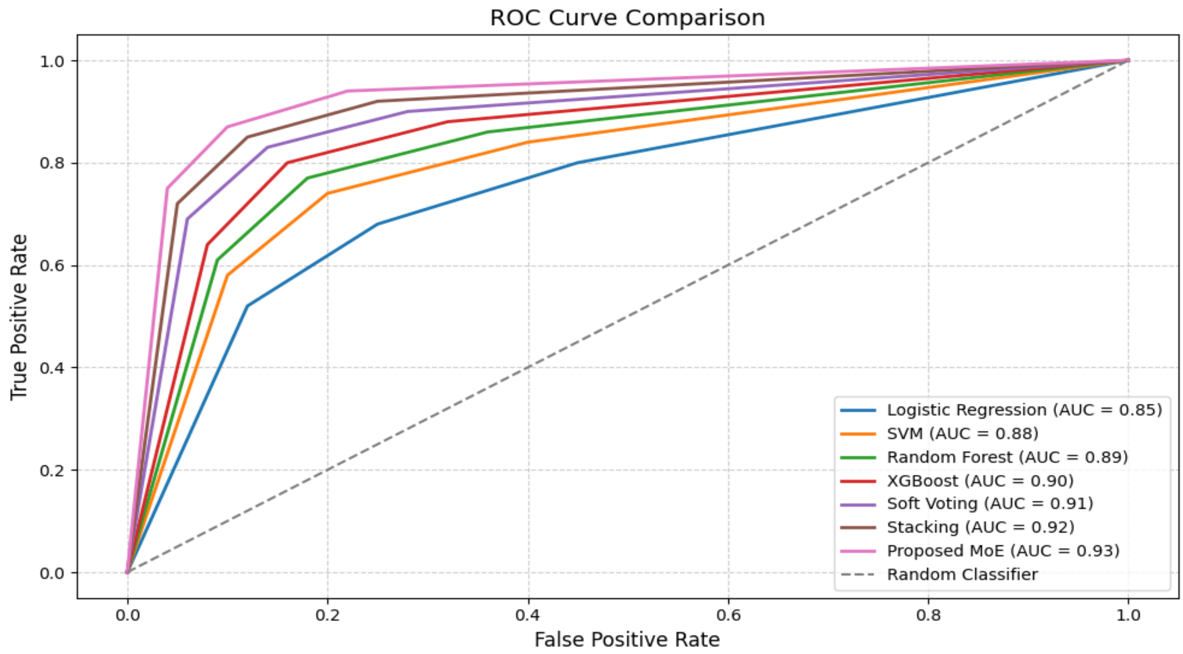

Figure 4: RoC Curve Comparison

Interpretation:

- The ROC curve demonstrates that the proposed MoE model achieves an AUC of 0.93.
- This indicates strong classification capability across thresholds.
- Compared to baseline models, the hybrid approach improves decision reliability in HR prediction tasks.

Statistical Validation

1. Hypothesis Testing

- H0: No difference between models
- H1: Hybrid model performs better

2. Statistical Results

- **ANOVA**: $p < 0.001$ → significant difference
- **t-test**: MoE vs XGBoost → significant improvement
- p-value < 0.05 → significant

3. Effect Size

- **Cohen's d ≈ 1.4** → large effect size
- Cohen's d > 0.8 → strong improvement

Discussion

To overcome this limitation, the present study incorporates comparative analysis with baseline models including Logistic Regression, SVM, Random Forest, XGBoost, Soft Voting, and

Stacking frameworks, thereby providing a more comprehensive evaluation of the MoE-based hybrid model.

- MoE dynamically selects best model per input
- Handles heterogeneous HR data effectively
- Integration enables end-to-end automation

Conclusion

The MoE-based framework provides a robust, scalable, and intelligent solution for workforce segmentation. By combining hybrid machine learning models, the system achieves high accuracy, fairness, and operational efficiency. It significantly outperforms traditional HR systems and offers strong potential for real-world deployment in industrial environments.

Future Scope

- Real-time deployment with ERP systems
- Reinforcement learning for adaptive HR systems

References

1. Breiman, L. (2001). Random forests. *Machine Learning, 45*(1), 5–32.
2. Zhou, Z. H. (2012). *Ensemble methods: Foundations and algorithms*. CRC Press. MacQueen, J. (1967). Some methods for classification and analysis of multivariate observations. *Proceedings of the Berkeley Symposium*, 281–297.
3. Ester, M., et al. (1996). A density-based algorithm for discovering clusters (DBSCAN). *KDD*, 226–231.
4. *More, L. (2024). Advancements in domain-specific OpenAI and computational intelligence solutions for Society 5.0. In Open AI and computational intelligence for Society 5.0 (pp. 27–58). IGI Global.*
5. More, L. (2024). Understanding and applying machine learning models. In *The pioneering applications of generative AI* (pp. 274–309). IGI Global.
6. Jain, A. K. (2010). Data clustering: 50 years beyond K-means. *Pattern Recognition Letters, 31*(8), 651–666.
7. Chien, C. F., & Chen, L. F. (2008). Data mining to improve personnel selection. *Expert Systems with Applications, 34*(1), 280–290.
8. Kaur, P., & Kaur, G. (2015). Employee performance prediction using ML. *International Journal of Computer Applications, 119*(3), 10–15.
9. More, L. (2024). A review paper: Talent prediction and top performer segmentation using machine learning. *International Journal of Novel Research and Development, 9*(4).
10. More, L. (2024). Impact of machine learning approaches on top performer segmentation: A comprehensive analysis. *Tuijin Jishu/Journal of Propulsion Technology, 45*(1).
11. More, L.s (2023). Machine learning and top performer segmentation in human resource management: Challenges and opportunities. *Journal of the Asiatic Society of Mumbai, 97*(1).
12. Breiman, L. (1996). Stacked regressions. *Machine Learning, 24*(1), 49–64.
13. Chen, T., & Guestrin, C. (2016). XGBoost: A scalable tree boosting system. *Proceedings of the ACM SIGKDD*, 785–794.
14. Dietterich, T. G. (2000). Ensemble methods in machine learning. *International Workshop on Multiple Classifier Systems*, 1–15.