

# PACE Framework - Recruitment Success Prediction

---

## P - PLAN

### Define the Problem:

- Build a binary classification model to predict job applicant hiring outcomes (Employed: Yes/No)
- Deploy as a user-friendly web application for HR teams
- Target:  $\geq 85\%$  accuracy with balanced precision/recall

### Project Scope:

- **In Scope:** 12-feature predictive model, multiple algorithm comparison, bias analysis, web deployment
- **Out Scope:** ATS integration, resume NLP, mobile apps

### Key Stakeholders:

- Primary: HR recruitment teams, hiring managers
- End Users: HR personnel (varying technical skills)

### Resources & Timeline:

- 10 weeks total
- Tools: Python, scikit-learn, XGBoost, Streamlit
- Free cloud deployment platform

### Risk Mitigation:

- Class imbalance  $\rightarrow$  SMOTE/class weighting
  - Model bias  $\rightarrow$  Fairness analysis across demographics
  - Deployment issues  $\rightarrow$  Local testing first
- 

## A - ANALYZE

### Exploratory Data Analysis:

- **Data Quality:** Check missing values, outliers, duplicates
- **Class Balance:** Distribution of Employed (Yes/No)
- **Feature Distributions:** Univariate analysis of all 9 features
- **Relationships:** Correlations between features and target variable
- **Bias Check:** Hiring rates across Age and Gender groups

### Key Insights to Identify:

- Which features strongly predict hiring?
- Is there class imbalance?
- Any demographic bias patterns?
- Salary and experience thresholds?

### Data Preprocessing:

- **Missing Values:** Median (numeric), mode (categorical)
- **Encoding:** One-hot for Gender/MainBranch, ordinal for EdLevel
- **Scaling:** StandardScaler for distance-based models
- **Imbalance:** SMOTE or class weights if needed
- **Split:** 70% train, 15% validation, 15% test (stratified)

### Feature Engineering (Optional):

- Experience ratio: YearsCodePro / YearsCode
  - Salary per skill: PreviousSalary / ComputerSkills
  - Only implement if improves performance
- 

## C - CONSTRUCT

### Model Development:

#### Baseline Models (Week 3):

- Logistic Regression
- Decision Tree
- Target:  $\geq 75\%$  accuracy baseline

#### Advanced Models (Week 4):

- Random Forest
- Gradient Boosting
- XGBoost

#### Optimization (Week 5):

- Hyperparameter tuning (GridSearchCV/RandomizedSearchCV)
- 5-fold cross-validation
- Feature importance analysis

### Evaluation Metrics:

- Accuracy, Precision, Recall, F1-Score, ROC-AUC
- Confusion matrix
- Cross-validation consistency (std dev  $< 5\%$ )

### Model Selection (Week 6-7):

- Compare all models on validation set
  - Select best based on F1-Score and ROC-AUC
  - Conduct fairness analysis (Gender, Age)
  - Retrain on train+validation, evaluate on test set
  - Serialize final model pipeline (preprocessing + model)
- 

## E - EXECUTE

### Application Development (Week 8):

#### UI Components:

- Input fields for all 12 features (dropdowns, number inputs)
- "Predict" button
- Output: Prediction result + confidence score
- Input validation and error handling

#### Technology Stack:

- **Framework:** Streamlit (recommended for simplicity)
- **Alternative:** Gradio or Flask

### Deployment (Week 9):

- Push code to GitHub repository
- Deploy to Streamlit Cloud (or Hugging Face Spaces/Render)
- Test deployed application thoroughly
- Obtain public URL

### Documentation (Week 10):

- **README:** Project overview, model performance, usage guide, technical details
- **User Guide:** How to use the application with screenshots
- **Technical Docs:** Model specs, preprocessing steps, fairness analysis

### Final Presentation:

1. Problem statement & objectives (2 min)
2. EDA insights with key visualizations (2 min)
3. Model comparison & results (3 min)
4. Feature importance & bias analysis (2 min)
5. **Live demo** of deployed application (3 min)
6. Conclusions & future work (2 min)
7. Q&A

### Final Deliverables:

- Jupyter notebooks (EDA + modeling)
  - Serialized model pipeline
  - Deployed web application with public URL
  - Complete documentation (README, user guide)
  - Presentation slides
  - GitHub repository
-