2025

AI Resume Screeing for HR and Applicant Analysis

TEAM LOCKED IN
35361814 PRATEEK MISHRA
35017106 HANEEN MOHAMED RAFFIQUE
35096533 MAVERON TYRIEL VERGARA AGUARES

ICT304 ASSIGNMENT 1 PROPOSAL

Table of Contents

Executive Summary

The AI Resume Scanner is an intelligent system that automates and enhances the resume screening process and provides explainable analytics for HR departments and job seekers. The system employs an experimental AI architecture currently utilising ensemble machine learning methods while actively investigating advanced NLP techniques, including BERT-based skill matching. The system emphasises explainability and adaptability, allowing for methodological evolution as techniques are validated and optimised.

Unlike traditional systems like the Applicant Tracking Systems, which are widely used for resume screening currently, prioritise keyword matching, which is easily deceivable. Our system prioritises candidate quality and emphasises the explanability of the decision metrics as a fundamental requirement rather than it being an afterthought, making it a responsible AI solution that promotes fair hiring practices while streamlining operational efficiency.

System Overview

System Title

Intelligent Resume Screening and Analytics System (IRSAS)

Problem Statement

Traditional resume screening processes are time-consuming, subjective, and prone to human bias. Even the Applicant Tracking System used currently is prone to inherent bias and prioritises the right keywords instead of the right applicants. Our research into the recruitment domain reveals that organisations face critical bottlenecks in processing large volumes of applications efficiently while maintaining consistency in candidate evaluation and providing transparent, explainable hiring decisions.

The problem extends beyond mere efficiency concerns. Modern recruitment requires extracting meaningful insights from candidate profiles, understanding skill transferability across industries, and ensuring compliance with fair hiring practices. These challenges demand an AI system that can process natural language content with semantic understanding rather than simple pattern matching.

Requirements and Specifications of the System

The resume screening system utilises a dual architecture approach to address the requirements of the system.

Quantitative Assessment Layer

- **Ensemble Classification Framework**: Machine Learning ensemble providing robust numerical fit scoring with cross-validation and weighted voting mechanisms

- **Multi-dimensional Feature Engineering**: Statistical analysis of structured resume data, including experience years, skill counts, education levels, and career progression patterns
- **Traditional ML Explainability**: SHAP (SHapley Additive exPlanations) for global feature importance and LIME (Local Interpretable Model-agnostic Explanations) for individual prediction interpretability
- **Three-class Classification System**:
  - *Good Fit*: High probability match (>0.7 confidence)
  - *Potential Fit*: Moderate match requiring human review (0.4-0.7 confidence)
  - *Not Fit*: Low probability match (<0.4 confidence)
- **Confidence Scoring**: Probabilistic outputs with uncertainty quantification for risk assessment
- **Performance Benchmarking**: Continuous evaluation against traditional keyword-matching ATS systems

Qualitative Assessment Layer

- **Contextual Neural Processing**: Deep learning models for unstructured text analysis, capturing nuanced language patterns and implicit qualifications
- **Semantic Skill Matching**: Advanced NLP techniques (planned BERT integration) for understanding skill transferability and contextual relevance beyond exact keyword matches
- **Cultural Fit Analysis**: LLM-powered assessment of communication style, values alignment, and soft skill indicators from resume language and structure
- **Natural Language Explanation Generation**: Conversational AI system producing human-readable justifications for recommendations, translating statistical outputs into business-friendly insights
- **Dynamic Questioning Framework**: Adaptive follow-up question generation to clarify ambiguous qualifications or explore edge cases

- **Enhanced Explainability Pipeline**: Multi-modal explanation system combining statistical evidence with contextual reasoning for comprehensive decision transparency

Integration and Fusion Layer

- **Hybrid Scoring Algorithm**: Weighted combination of quantitative confidence scores and qualitative assessment ratings
- **Conflict Resolution System**: Protocols for handling disagreements between the ensemble, quantitative and qualitative layers
- **Adaptive Learning Mechanism**: A Feedback incorporation system allowing continuous improvement of both assessment layers based on hiring outcomes

We give precedence to interpretable explanations for every decision made by the system instead of the usual opaque scoring mechanisms. The transparency of the system supports fair hiring practices and aims to remove any form of bias in the hiring process.

Stakeholders

- **Primary Stakeholders**: Recruiters, HR managers, job seekers
- **Secondary Stakeholders**: Hiring managers, compliance officers, candidates

System Requirements and Concept Development

Functional Requirements

- **Resume Parsing**: Extract structured information from various resume types
- **Experimental AI Engine**: Hybrid neural-traditional integration
  - Modular architecture supporting multiple ML approaches
  - Current: Ensemble methods (Random Forest + Logistic Regression)
  - Under Development: BERT-based semantic skill matching
  - Planned: Hybrid neural-traditional integration

    - Framework designed for seamless technique substitution and A/B testing
- **Scoring and Ranking**: Provide quantitative candidate assessments
- **Explainability**: Generate interpretable results for stakeholders
- **Analytics Dashboard**: Visualise screening results and insights

Non-Functional Requirements

- **Performance**: Process 100+ resumes within 5 minutes
- **Accuracy**: Achieve 85 %+ precision in skill matching
- **Scalability**: Handle parallel processing of multiple job postings
- **Usability**: Intuitive interface for non-technical HR staff
- **Compliance**: Ensure bias reduction and fair hiring practices

Project Methodology

Following the NASA Systems Engineering Handbook, we've adopted a structured approach through the System Engineering Product Life Cycle. It ensures that we systematically develop the system from problem identification to deployment and maintenance. This assignment submission covers the first 4 phases of the system engineering lifecycle:

Problem Formulation: Stakeholder analysis, requirement gathering and feasibility analysis completed. We conducted exhaustive research into current recruitment challenges and gaps to identify pain points that our solution can tackle.

Concept Development: Solution architecture design, technology selection, and risk assessment completed. AI techniques that pertain to the use case were selected.

System Design: Detailed component specifications, interface definitions, and performance criteria have been confirmed. Architecture ensures scalability for future expansion while being integrable with existing HR systems.

Sub-System Prototyping: Backend development and model training are being carried out. Fitting model performance metrics to analyse feasibility is recorded.

System Architecture and Design

High-Level System Architecture

The system follows a modular architecture that is designed with scalability, maintainability and integrability in mind, with five layers each handling different aspects of system functionality.

1. **Data Ingestion Layer**
   - Resume upload and parsing module
   - Job description processing
   - Data validation and cleaning
2. **Experimental AI Engine**
   - Modular Architecture supporting multiple ML approaches
   - Current: Ensemble Methods
   - Under Development: BERT-base semantic skill matching
   - Planned: Hybrid neural-traditional integration
   - Framework designed for seamless technique substitution and A/B testing
3. **Analytics and Scoring Engine**
   - Candidate ranking algorithms
   - Explainability generation
   - Performance metrics calculation
4. **User Interface Layer**

- Web-based dashboard
- API endpoints for integration
- Reporting and visualisation tools

5. **Data Storage Layer**
   - Resume database
   - Model artifacts storage
   - Analytics and audit logs

Sub-Systems Identification

AI Sub-Systems

- **Resume Parser**: NLP-based text extraction and structuring. It can adapt to multiple formatting styles and improve performance by continually learning from different resume formats
- **Experimental AI Engine**: Dual-architecture three-class classifier for identifying good/potential/no fit using a combination of traditional ML models and NN/LLM-based models
- **Explainability Generator**: The Decision reasoning module produces interpretable metrics for the system recommendations. Promotes transparency in the hiring process.

Non-AI Sub-Systems

The File Management System handles document storage and retrieval with support for multiple file formats and version control.

The User Authentication module implements role-based access control, ensuring appropriate system access for different user types.

The Reporting Engine provides statistical analysis and visualisation capabilities for system performance monitoring and business intelligence.

Integration APIs enable connectivity with external HR systems and databases.

Prototype Development

Prototype Scope

The initial prototype focuses on the core AI sub-system: **Experimental AI Engine**

Prototype Components

1. **Quantitative Assessment Layer**: Traditional ML Layer using an ensemble method combining multiple ML models
2. **Qualitative Assessment Layer**: Use of NN/LLM-based systems to produce qualitative fit.
3. **Explainability Output**: Generate human-readable explanations

Technology Stack

- **Programming Language**: Python 3.9+
- **ML Libraries**: BERTopic, scikit-learn, transformers, spaCy
- **Data Processing**: pandas, numpy, NLTK
- **Document Processing**: PyPDF2, python-docx
- **Visualisation**: matplotlib, plotly
- **Development Environment**: Jupyter Notebook, VS Code

Prototype Implementation Status

- ✅ Environment setup and dependency management

- 🔄 Quantitative Layer
    - ✅ Random Forest Classifier
    - ⏳ Logistic Regression Classifier
    - ⏳ Ensemble Classifier
- ⏳Qualitative Layer
- ⏳ Explainability module implementation
- ⏳ Basic web interface development

AI Technique Investigation and Justification

The quantitative layer is composed of an ensemble model containing a random forest classifier and a logistic regression classifier. The following section details the experiments run for the random forest classifier and the Logistic Regression Classifier.

Random Forest was selected due to its inherent strengths in handling structured resume data and meeting HR industry requirements for transparency and reliability. The algorithm's ensemble nature provides robust classification performance while naturally generating feature importance rankings essential for explainable hiring decisions and regulatory compliance. Its computational efficiency enables real-time processing of high-volume applications, meeting our performance requirement of 100+ resumes within 5 minutes, while the probabilistic outputs facilitate seamless integration with our planned hybrid architecture combining quantitative and qualitative assessment layers.

Logistic Regression was selected as a complementary classifier in our ensemble approach due to its linear interpretability and probabilistic foundation that aligns perfectly with HR decision-making processes. The algorithm provides clear coefficient-based explanations showing how each resume feature influences the hiring decision, offering transparent mathematical relationships that HR professionals can easily understand and justify. Its probabilistic output naturally supports our three-class classification framework while providing well-calibrated confidence scores essential for risk assessment in candidate evaluation. Logistic Regression's computational efficiency, statistical rigour, and established performance in binary and multiclass problems make it an ideal partner to Random Forest, creating a robust ensemble that balances the strengths of both linear and tree-based approaches while maintaining the explainability and reliability requirements critical for responsible AI in recruitment.

Experiment files are available here: https://github.com/ashkree/Resume-Screener.git

Methodology

Data Processing

Models are trained using the resume-job-description-fit dataset by user "cnamuangtoun" from Huggingface (Muangtoun 2024). The dataset features columns for resume-text, job-description-text, and label (good/potential/no fit) and provides 6.24 thousand training samples and 1.76 thousand testing samples. 30% of the training samples were taken for validation.

As the text columns showed several anomalies, a text cleaner was developed that fixes spacing anomalies between words, numbers, and punctuation, standardises formatting inconsistencies, normalises technical terms and section headers, and corrects common OCR errors and text extraction artifacts that typically occur when processing PDF documents and various file formats.

All three data splits were undersampled to handle class imbalance. The final sample set includes 3000 training rows, 1200 validation rows and 300 rows for testing. After which, features were extracted using TF-IDF with 5000 features, resulting in a density of ~96% for each training split. The final features were exported for use in the experiments.

Model Training Setup

The training setup uses a custom ModelTrainer and ModelEvaluator to standardise all outputs. The ModelTrainer performs Bayesian Optimisation with K-Fold Cross Validation. K-fold cross-validation and optimization parameters are configurable, including scoring metrics. All experimental models are saved and can be loaded individually for testing and drop-in replacement for the AI engine.

Random Forest Experiments

Although random forests already possess built-in feature selection, features were further reduced to improve training time and computational efficiency. Several methods were tested, such as Chi2, Mutual Info, F-Score, and another RandomForestClassifier, with each method limiting the feature set from a range of 500 to 1500. Each method was tested using a baseline random forest model and compared on accuracy. After experimentation, Mutual Info, taking the top 500 features, performed the best with a validation accuracy of 62.33%.

Four models were trained and evaluated on the selected features. These models differed in parameter spaces. Each model detailed below was trained on 5 folds, 75 trials and an accuracy scoring metric.

1) Baseline Random Forest - default scikit-learn parameters with no optimisation
2) Anti-Overfit Parameter Space - comprehensive anti-overfitting strategies with constrained tree depth, high tree count, and regularisation techniques.
3) High-Performance Parameter Space - larger forests and deeper trees to boost accuracy
4) Balanced Parameter Space - A balance between generalisation and model complexity via moderate forest sizes and tree depth.

Logistic Regression

Before training models, a short experiment was performed to determine the effects of standardization of the data. The model used is a baseline logistic regression model with default scikit-learn parameters. The following table details the results of the experiment.

Scaling	CV score	Train	Val	Overfit Gap
Unscaled	0.5262 ± 0.0244	0.6430	0.5775	0.0655
Scaled	0.6506 ± 0.0322	0.9477	0.6967	0.2510

The results show that the scaling improved the validation accuracy of the baseline model by 0.1192. In order to improve the accuracy of models the scaled data are used for the experiment, with this in mind strong regularization parameters are prioritized.

The experiment replicates the setup of the random forest classifiers with 2 models.

1) Baseline Logistic Regression - A logistic regression model with default parameters
2) Anti Overfit Space - Parameters catered for strong regularization

Results

All experimental results are available on the respective notebooks for each model.

Data Management and Ground Truth Development

6.1 Data Sources

- **Public Resume Datasets**: Kaggle resume datasets, academic repositories
- **Synthetic Data Generation**: Generated resumes using templates and industry standards
- **Job Posting Data**: Scraped job descriptions from public job boards (with compliance)

6.2 Ground Truth Development

- **Skill Annotation**: Manual labelling of resume skills by HR experts
- **Topic Validation**: Expert review of BERTopic-generated topics
- **Matching Validation**: HR professional assessment of candidate-job fit

6.3 Data Privacy and Legal Considerations

- All personal identifiers removed from datasets
- Compliance with GDPR and local privacy regulations
- Synthetic data generation to supplement real data
- Explicit consent for any proprietary data usage

System Engineering Lifecycle Implementation

Performance Metrics Framework

Our evaluation framework incorporates multiple metrics to assess system performance comprehensively. Precision measures the accuracy of relevant candidate identification relative to the total identified candidates. Recall evaluates the system's ability to identify all relevant candidates within the applicant pool. F1-Score provides the harmonic mean of precision and recall, offering a balanced performance indicator. Processing time metrics ensure operational efficiency by measuring average analysis time per resume. User satisfaction scores capture stakeholder feedback on system utility and usability.

Multi-Dimensional Evaluation Approach

The evaluation strategy encompasses four distinct assessment dimensions:

Technical Evaluation focuses on algorithm performance using standardised test datasets, ensuring consistent and objective performance measurement. **User Evaluation** incorporates HR professional assessment of system utility in real-world scenarios, validating practical applicability. **Comparative Evaluation** benchmarks performance against traditional screening methods, demonstrating improvement over current practices. **Bias Evaluation** conducts fairness assessment across demographic groups, ensuring equitable treatment and compliance with fair hiring practices.

Success Criteria and Benchmarks

System success is defined through specific, measurable criteria: achieving 85% or higher accuracy in skill matching demonstrates technical competence, reducing resume screening time by 70% validates operational efficiency, maintaining 90% or higher user satisfaction rating ensures practical adoption, and demonstrating explainable decision-making capabilities supports transparent recruitment practices.

Tools and Project Management

Tools Implementation

Project Management

- **Tool**: Jira Scrum Board
- **Usage**: Sprint planning, task tracking, milestone management

Version Control

- **Tool**: Git with GitHub
- **Usage**: Code versioning, branch management, collaboration

Collaboration

- **Tool**: Microsoft Teams/GitHub
- **Usage**: Team communication, code reviews, documentation sharing

Communication

- **Tool**: Microsoft Teams + WhatsApp
- **Usage**: Stakeholder updates, progress reports, issue escalation

Development Methodology

The project follows Agile development principles with two-week sprints, enabling iterative progress and regular stakeholder feedback. Our testing strategy encompasses unit testing for individual components, integration testing for system interactions, and user acceptance testing for real-world validation. Quality assurance procedures include systematic code reviews, automated testing implementation, and continuous performance monitoring.

Evaluation Methodology

9.1 Performance Metrics

- **Precision**: Correctly identified relevant candidates / Total identified candidates
- **Recall**: Correctly identified relevant candidates / Total relevant candidates
- **F1-Score**: Harmonic mean of precision and recall
- **Processing Time**: Average time per resume analysis
- **User Satisfaction**: Stakeholder feedback scores

9.2 Evaluation Framework

1. **Technical Evaluation**: Algorithm performance on test datasets
2. **User Evaluation**: HR professional assessment of system utility
3. **Comparative Evaluation**: Performance vs. Traditional Screening Methods
4. **Bias Evaluation**: Fairness assessment across demographic groups

9.3 Success Criteria

- Achieve 85 %+ accuracy in skill matching
- Reduce resume screening time by 70%
- Maintain 90 %+ user satisfaction rating
- Demonstrate explainable decision-making capabilities

Timeline and Deliverables

Assignment Phase (Current)

- **Week 1-2**: System design and architecture documentation
- **Week 3-4**: Prototype AI sub-system development
- **Week 5**: Testing and documentation completion

Project Phase (Future)

- **Phase 1**: Complete system integration
- **Phase 2**: User interface development
- **Phase 3**: Testing and optimisation
- **Phase 4**: Deployment and validation

Conclusion

The Intelligent Resume Screening and Analytics System represents a significant advancement in HR technology, providing explainable and efficient candidate evaluation capabilities. The use of BERTopic for topic modeling ensures interpretable results while maintaining high performance. Prototype development focuses on core AI functionality, establishing a solid foundation for the complete system implementation.

The system addresses critical pain points in the recruitment process while ensuring transparency and fairness in decision-making. The modular architecture and comprehensive evaluation framework position the system for successful deployment and continuous improvement.

References

NASA Systems Engineering Handbook. (2016). NASA/SP-2016-6105 Rev2. National Aeronautics and Space Administration.

Grootendorst, M. (2022). BERTopic: Neural topic modeling with a class-based TF-IDF procedure. *arXiv preprint arXiv:2203.05794*.

Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics*, 4171-4186.

Honnibal, M., & Montani, I. (2017). spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing. *Computational Linguistics*, 7, 411-420.

Muangtoun, C. N. (2024). resume-job-description-fit. https://huggingface.co/datasets/cnamuangtoun/resume-job-description-fit

Zhang, Y., & Wallace, B. (2015). A sensitivity analysis of (and practitioners' guide to) convolutional neural networks for sentence classification. *arXiv preprint arXiv:1510.03820*.

European Union. (2018). General Data Protection Regulation (GDPR). *Official Journal of the European Union*, L119, 1-88.

IEEE Standards Association. (2017). IEEE Standard for Software Engineering Processes and Specifications. IEEE Std 1012-2016.