**Resume Classification System – Veridia.io Internship**

**Submitted by:** Atharva Gaikwad  
**Date:** October 20, 2025

**1. Introduction**

Recruitment in large organizations involves screening hundreds or thousands of resumes for various job roles. Automating resume classification speeds up shortlisting and improves fairness. This project delivers an end-to-end, AI-powered solution using deep learning to classify candidate resumes into 24 professional job categories, leveraging a fine-tuned BERT model and an interactive Streamlit web app.

**2. Problem Statement**

Build a resume classification pipeline capable of:

* Preprocessing and cleaning resume data
* Training a high-performance deep learning model for classifying resumes into correct job categories
* Deploying the model via a user-friendly web app
* Delivering at least 85% classification accuracy (Actual: 87.15%)

**3. Dataset Used**

* **Source:** [Kaggle Resume Dataset](https://www.kaggle.com/datasets/snehaanbhawal/resume-dataset)
* **Data Points:** 2484 resumes with job category labels (24 classes)
* **Formats:** Mostly text, diverse structure, multiple professions
* **Preprocessing:** Data cleaning to remove emails, URLs, extra formatting; basic normalization

**4. Technical Stack**

* **Language:** Python 3.10+
* **Deep Learning:** PyTorch 2.0+, Hugging Face Transformers (BERT)
* **ML Libraries:** scikit-learn, pandas, numpy
* **Web App:** Streamlit
* **Visualization:** matplotlib, seaborn
* **Serving:** Streamlit + ngrok for tunneling
* **Development:** Google Colab (GPU T4)

**5. Methodology**

**5.1 Data Preprocessing & Augmentation**

* Resumes cleaned to remove emails, URLs, extra spaces, and special characters
* Min length enforced (≥50 chars) for data quality
* Applied BERT WordPiece tokenization (max length 512 tokens)

**5.2 Model Architecture**

* Used bert-base-uncased with sequence classification head (24 class outputs)
* Label encoding for class mapping

**5.3 Training Approach**

* Train/Val/Test Split: 80% / 10% / 10%, stratified
* Optimizer: AdamW, lr=3e-5
* Scheduler: Cosine with 10% warmup
* Batch size: 16 (gradient accumulation)
* Loss: Cross-entropy
* Early stopping on F1-Score (patience=3)
* Mixed precision training (torch.cuda.amp)
* Ran for 6 epochs; best F1 at 4th epoch

**5.4 Evaluation Metrics**

* Overall Accuracy
* Weighted F1-Score (main metric)
* Precision and Recall per class
* Confusion Matrix (visualized)
* Training Loss and Accuracy Curves

**5.5 Deployment**

* Streamlit web app for human-in-the-loop inference
* File upload, PDF/DOCX/TXT parsing, and real-time predictions
* Top-5 categories + confidence

**6. Results**

**6.1 Final Model Quality**

* **Test Accuracy:** 87.15%
* **Weighted F1-Score:** 86.40%
* **Precision:** 86.10%
* **Recall:** 87.00%

**6.2 Notable Category Scores**

| **Category** | **F1-Score (%)** |
| --- | --- |
| Information Technology | 92 |
| Data Science | 89 |
| HR | 87 |
| Sales | 85 |

**6.3 Visualizations**

* Category distribution bar plot (category\_distribution.png)
* Confusion matrix heatmap (confusion\_matrix.png)
* Training/validation loss and accuracy (training\_history.png)

**6.4 Error Analysis**

* Most confusion between similar roles (e.g., HR <-> Admin, Web Dev <-> Java Dev)
* Some rare categories underrepresented (future: augment data)

**7. How to Use**

1. **Clone or download the project folder.**
2. Install requirements:

text

pip install -r code/requirements.txt

1. Ensure model/resume\_classifier\_best/ folder is present.
2. Launch Streamlit:

text

streamlit run code/app.py

1. Use the browser interface – upload resumes and see predictions instantly.

**8. Limitations and Future Work**

* **Data Limitation:** Some job categories are very under-represented; more data will improve performance.
* **Permanent Hosting:** For now, the Streamlit ngrok link is ephemeral; future ambition is to deploy on AWS, Azure, or Heroku.
* **Multi-language Support:** Current model is English-only.
* **Advanced Extraction:** In production, use deeper NLP for parsing work experience, skills, etc.

**9. Conclusion**

This project demonstrates a robust, accurate, and scalable solution for resume classification using modern deep learning techniques. The model achieves strong accuracy on challenging, real-world data and is accessible via a streamlined web app interface.

**Contact:**  
Atharva Gaikwad  
Veridia.io AI/ML Internship – October 2025