AI driven resume screening

Manik Jain, Praveen Kumar, Logan McGovern

Problem statement ↘

Hiring the right candidate for a job in 2026 is extremely black box, time consuming, and requires multiple rounds of candidate reachout, ATS parsing, manual resume shortlist, and an influx of AI generated “perfect” resumes. Applicants are also aware of how to “cheat” the system, and they are using all sorts of techniques to increase keyword frequency which confuses the system whether is actually capable. Hiring managers spend nearly 10 seconds screening a resume or rely on outdated ATS systems which are inheriting a lot of training data from older hiring strategies and using limited and stale data sets for training and testing, leading to high false-negatives; thereby rejecting a potential strong candidate. This results in a waste of crucial resources - time and money for the technical hiring teams.

Problems

Organizations - Private/Public; Applicants

Organizations are unable to target the potential candidates, and applicants are irritated because of multiple rejections

Organizations are over reliant on old processes, stale data, out-of-date Machine Learning models

Project objectives

01

Design and deploy an efficient, learning based resume screening model to significantly lower the rate of false-negatives

02

Provide a transparent, bias-free alternative to manual resume screening in an AI-saturated job market.

03

Reduce false-negative rate by $X\%$ (to be determined during testing).

04

Help hiring teams find strong, relevant candidates without wasting time or unfairly rejecting people

Resources

Data

- Label Resumes and job descriptions
- Signals of good fit
- examples of AI generated or keyword stuffed resumes

Modelling and NLP capabilities

- OCR to understand job descriptions and resumes
- Model to score candidate role fit instead of just matching keywords

Fairness, Bias, Evaluation

- Bias Check to ensure model doesn't replicate old, unfair hiring patterns
- Metrics for False-positives and False-Negatives

System and Interface

- Web API for recruiters to interact and feed decisions to train model
- Inbuilt Observability and Analytics

Approach

01	Goal	Predict potential candidates for a given job description based on their YOE, Education, Project,, Open source contributions
02	Gather Data	Use linkedin profile data, github contributions, other quantifiable skills
03	Prepare Data	Imputate, categorize, and select the most relevant features without bias and outliers
04	Train Model	Build baseline and advanced regressors
05	Measure Results	Check Target variable accuracy
06	Deploy	Create and deploy in the cloud a Web2.0 application using PyCaret and Streamlit.