📌 Problem Statement

Recruiters and HR teams receive hundreds of resumes for a single job role, making manual screening time-consuming, prone to bias, and inconsistent. To streamline this process, we aim to **build an automated Resume Screening App** that:

1. **Ingests** resume text (uploaded as `.pdf` or `.docx`).
2. **Preprocesses** the text using Natural Language Processing (NLP).
3. **Classifies** the resume into a target job role (e.g., Data Scientist, Software Engineer, Web Developer, HR).
4. **Predicts Fit Score** (how well the candidate matches the target role).
5. Provides a **Streamlit Web App** interface for recruiters to upload resumes and instantly see results.
6. Deploys the application on **Streamlit Cloud / Hugging Face Spaces / AWS / Heroku** for easy access.

This project will demonstrate the practical use of NLP, classification models, and web deployment in solving a real-world HR challenge.

📂 Dataset

We will use an **open-source Resume Dataset** available on Kaggle:

🔗 **Dataset Link:** Kaggle - Resume Dataset

- **Description:** Contains labeled resumes categorized into multiple job domains such as **Data Science, HR, Advocates, Arts, Web Designing, Software Engineering, Business Development, Health, etc.**
- **Format:** Each entry includes raw resume text + job category label.
- **Size:** ~ 1000+ resumes.

⚙️ Machine Learning Model

Pipeline:

1. **Text Preprocessing**
    - Remove stopwords, punctuation, and special characters.
    - Tokenization and Lemmatization.
    - Vectorization (TF-IDF / Word2Vec / BERT embeddings).
2. **Feature Engineering**
    - Extract n-grams, keywords (skills, education, experience).
    - Skill-matching with job role requirements.
3. **Model Training**
   Possible ML models for classification:
    - **Logistic Regression** (baseline model).
    - **Naive Bayes** (works well for text classification).
    - **Random Forest** or **XGBoost** (better performance).
    - **Fine-tuned BERT / DistilBERT** (for advanced NLP).
4. ✅ Best trade-off for deployment: **TF-IDF + Logistic Regression / Naive Bayes** (fast, lightweight, interpretable).
5. **Evaluation Metrics**
    - Accuracy, Precision, Recall, F1-score.
    - Confusion Matrix per job category.

🌐 Deployment Plan

1. **Frontend:**
    - Streamlit interface:
        - File uploader (`st.file_uploader`) for `.pdf` / `.docx`.
        - Display extracted text.
        - Show predicted job category + fit score.
2. **Backend:**
    - Python ML model trained on the dataset.
    - Pickle / Joblib model saving and loading.
3. **Deployment Platforms:**

    Streamlit Cloud

📊 Example Output

- Uploaded Resume: `resume.pdf`
- Extracted Job Role: **Data Scientist**
- Fit Score: **87%**
- Key Matched Skills: `Python, Machine Learning, Pandas, Deep Learning`