A cover of a report

AI-generated content may be incorrect.

1. ***Main Idea of the Project: Smart Career Guidance & Recruitment System***

The core idea of this project is to build an AI-powered matchmaker between people and careers.

Imagine a system that acts like a highly intelligent career counselor and a efficient recruitment assistant combined into one. Instead of a person manually reading through a resume (CV) and trying to guess the best job matches, this system uses Data Science and Artificial Intelligence to do it automatically, accurately, and instantly.

**It works in three main steps:**

1. Understand the Person: The system analyzes a student's or job seeker's CV. It doesn't just read the text; it uses Natural Language Processing (NLP) to intelligently identify and extract all the skills mentioned (e.g., Python, Project Management, Communication).
2. Understand the Jobs: The system has already analyzed thousands of job descriptions. It knows what skills are required for different career paths like "Data Scientist," "Software Engineer," or "Product Manager."
3. Make the Match: Using a Machine Learning model, the system compares the person's skills against the skills required for various careers. It then provides a personalized career recommendation, showing the top suitable job roles and even identifying any key skills the person might be missing.
4. ***Data Cleaning and Preparation: Main Idea***

The primary goal of the cleaning script is to transform raw, messy text from job descriptions and CVs into a structured, standardized list of skills that a machine learning model can understand and process effectively. Think of it as translating human-written resumes into a consistent "machine language" of skills.

The process can be broken down into five key phases:

**1. Initial Extraction and Structuring:**

* **Input:** The raw data starts as a single, combined text field for the job description (e.g., "Data Scientist needed with experience in Python, SQL, and Machine Learning").
* **Action:** The script acts like a smart parser. It searches for keywords to split this single string into two logical parts:
  + specialization**:** The job title (e.g., "Data Scientist").
  + experience\_needed**:** The string of required skills (e.g., "Python, SQL, and Machine Learning").
* **Why?** This creates distinct, meaningful categories from a block of text, which is the first step toward organization.

**2. Text Normalization and Noise Removal:**

* **Action:** The script prepares the text for analysis by:
  + Converting everything to lowercase to ensure "Python" and "python" are treated as the same skill.
  + Removing punctuation, numbers, and common but meaningless "stop words" (e.g., "the," "and," "in").
  + Identifying and removing "filler words" that are not actual technical or soft skills (e.g., "experienced," "professional," "strong knowledge of"). This is crucial to avoid polluting the skill list with vague terms.
* **Why?** This eliminates variations and clutter, ensuring the analysis focuses only on the valuable content—the skills themselves.

**3. Skill Standardization and Lemmatization:**

* **Action:** This is the core intellectual step. The script uses a pre-defined "skill dictionary" (skill\_mapping) to standardize thousands of different skill variations into a consistent set of canonical names.
  + For example: "py", "python", "python3" all get mapped to the standard term "Python". "sql", "Structured Query Language", and "mysql" might be mapped to "Structured Query Language".
* It also uses **lemmatization**, an advanced technique to find the root form of a word (e.g., "running" -> "run", "better" -> "good"), ensuring different grammatical forms of a skill are consolidated.
* **Why?** Without this step, the model would see "Python," "python," and "Py" as three completely different skills, leading to terrible accuracy. Standardization creates a clean, consistent vocabulary for the model to learn from.

**4. Handling Multi-Word Concepts:**

* **Action:** The script is smart enough to recognize and preserve phrases that represent a single skill concept. It temporarily combines terms like **"machine learning"** or **"power bi"** into single tokens (e.g., "machine\_learning") during processing to prevent them from being split up and misinterpreted.
* **Why?** Treating "machine" and "learning" as separate skills would lose the specific and valuable meaning of the combined term.

**5. Final Filtering and Deduplication:**

* **Action:** As a final cleanup, the script filters out any remaining nonsense words that might have slipped through earlier steps. It then removes duplicate skills from each list, ensuring a candidate's proficiency in "Python" is only counted once.
* **Output:** The final result is two clean, structured columns for each record:
  + resume\_skills\_mapped: A standardized list of skills found in the CV.
  + experience\_needed\_skills\_mapped: A standardized list of skills required by the job.
* **Why?** This creates the pristine, high-quality dataset needed to train an accurate machine learning model. The model can now reliably compare a candidate's skills against job requirements because both are spoken in the same, clean language.

1. ***Explanation of Data Visualizations***

The visualizations in this project are designed to provide clear, actionable insights into the job market data and the performance of the recommendation system. They help us understand trends, validate our approach, and present findings in an easily digestible format.

A graph with blue dots

AI-generated content may be incorrect.**1. Match Score vs. Skill Overlap**

* What it is: A scatter plot showing the relationship between the number of skills a candidate has that match a job's requirements (skill\_overlap) and the overall match\_score they received.
* What it shows: Each dot represents a candidate-job pair. The plot shows a strong positive correlation.
* Key Insight: As the number of matching skills increases, the match score also increases. This validates the core logic of our system: skill overlap is a primary driver of a good job match. It proves our model's reasoning is grounded in a real, observable trend within the data.

A graph of multicolored lines

AI-generated content may be incorrect.**2. Top 10 Resume Skills**

* What it is: A bar chart displaying the most frequently occurring skills across all candidate resumes in the dataset.
* What it shows: The height of each bar represents how many times a particular skill was listed.
* Key Insight: This reveals the most common skills in the current candidate pool. For example, if "Python" and "SQL" are at the top, it shows these are baseline expectations for many roles. It helps job seekers understand what skills are in high supply and allows educators to see what is being taught.

**3. Top 10 Experience Skills**

* A graph of different colored bars

  AI-generated content may be incorrect.What it is: A bar chart displaying the most frequently demanded skills in the job descriptions analyzed.
* What it shows: The skills that employers are looking for the most.
* Key Insight: This reveals employer demand. Comparing this chart with the "Top Resume Skills" is incredibly valuable. For instance, if "Cloud Computing" is a top experience skill but *not* a top resume skill, it identifies a significant skills gap in the market. This is crucial guidance for both job seekers (what to learn) and companies (what skills are scarce).

A graph showing different colored rectangular shapes

AI-generated content may be incorrect.**4. Specialization Counts**

* What it is: A bar chart showing the distribution of different job titles (specializations) in the dataset.
* What it shows: How many resumes or job postings belong to each category (e.g., Data Analyst, Software Engineer, Product Manager).
* Key Insight: This illustrates the popularity and availability of different career paths in the data. A high count for "Data Analyst" suggests it's a common role. This also helps check for class imbalance, which is important for ensuring our machine learning model is trained on a representative sample of all specializations.

A graph of different colored bars

AI-generated content may be incorrect.**5. Distribution of Match Score**

* What it is: A histogram or count plot showing how often each possible match score (e.g., 1 to 5) appears in the dataset.
* What it shows: The frequency of different match quality levels.
* Key Insight: This helps us understand the overall quality of matches in the market. Is the dataset full of highly qualified candidates (scores skewed high)? Or are most candidates a mediocre fit (scores clustered in the middle)? A normal distribution might indicate a healthy and varied job market.

**6. Career Recommendation Output**

* A graph showing different colored rectangular shapes

  AI-generated content may be incorrect.What it is: The final output of the web application, showing a ranked list of specializations with confidence percentages after a user uploads their CV.
* Key Insight: This is the ultimate application of the entire system. It moves from general market trends to a personalized, actionable recommendation. The probabilities give the user a clear sense of how strong their fit is for each role, adding transparency and trust to the system's suggestion.

**Summary of Why Visualizations are Important**

* Build Trust: They provide evidence that the model's recommendations are based on real data patterns (e.g., the skill overlap vs. match score plot).
* Reveal Trends: They uncover macro-level insights about the job market that are not obvious from raw data (e.g., identifying skills gaps).
* Provide Guidance: They offer valuable insights for students, job seekers, and trainers on what skills are in demand.
* Communicate Clearly: They transform complex data analysis into simple, understandable charts that can be easily presented to any audience, technical or non-technical.

1. ***Machien\_learning***

Its main job is to **learn from historical data** (a dataset of CVs and their corresponding job specializations) and then **predict the best career path for a new person based on their skills.**

It does this by training a Machine Learning model to find patterns between sets of skills and job titles.

**What It Does & Why: Step-by-Step**

**1. Loads and Prepares the Data:**

* **What:** It reads a pre-cleaned CSV file containing lists of skills from resumes and the known job specializations for those resumes (e.g., "Data Scientist," "Software Engineer").
* **Why:** You can't train a model on raw text. This data has already been processed into a structured format that the algorithm can understand.

**2. Encodes the Data for the Machine:**

* **What:** This is a crucial step. It converts the text-based information into numbers.
  + LabelEncoder**:** Converts the job specializations (like "Data Scientist") into numbers (like 0, 1, 2). This is needed because machine learning models work with numbers, not text labels.
  + MultiLabelBinarizer**:** This is the most important part. It looks at all possible skills in the dataset (e.g., "Python," "SQL," "Communication") and creates a giant "checklist" for each resume. It creates a list of 0s and 1s where a 1 means "this person has this skill" and a 0 means "they don't."
  + StandardScaler**:** It adjusts the numerical data so that no single skill unfairly dominates the model just because it's more common. It puts all the features on the same scale.
* **Why:** Machine learning models are mathematical. They require all input to be in a numerical, standardized format to perform calculations effectively.

**3. Trains and Selects the Best Model:**

* **What:** The code tests different machine learning algorithms (like LogisticRegression) to see which one is best at finding the pattern between a person's skills (the input) and their job title (the output). It uses a robust method called **Stratified K-Fold Cross-Validation**.
* **Why:** This ensures the chosen model is accurate and reliable, not just lucky on a specific subset of data. It picks the best model to be the core of the recommendation system.

**4. Makes Predictions and Shows Clusters (Optional Analysis):**

* **What:**
  + It tests the final model on unseen data to show how well it performs.
  + It also uses **clustering algorithms** (KMeans, AgglomerativeClustering) to group similar resumes together *without* using the job title label. It then visualizes these groups.
* **Why:**
  + **Testing** proves the model's real-world usefulness.
  + **Clustering** is an exploratory technique to see if natural groupings exist in the data (e.g., a cluster of "web developers" and a cluster of "data people") which can provide additional insights beyond simple classification.

**5. The Recommendation Function (The Key Output):**

* **What:** This function (recommend\_top\_specializations) is the ultimate goal. When given a new list of skills, it:
  1. Cleans and standardizes them.
  2. Encodes them into the same "checklist" format used in training.
  3. Asks the trained model to predict the probability of each possible job specialization.
  4. Returns the top 5 most likely careers with their confidence scores.
* **Why:** This function is what powers the web application. It takes a user's input and delivers an intelligent, data-driven recommendation.

**6. Saves the Engine for Later Use:**

* **What:** The code saves the trained model and all the encoders to files (.pkl files).
* **Why:** This is essential for **deployment**. The web app (built with Streamlit) can load these files and use the model to make predictions without having to retrain it every time. It's like saving the brain so another program can use it.

**The Main Benefits**

1. **Automation:** It automates the task of career matching, which is typically done manually by humans, making it faster and scalable.
2. **Data-Driven Decisions:** Recommendations are not based on gut feeling but on patterns learned from real-world data.
3. **Personalization:** Provides personalized career advice tailored to an individual's unique skill set.
4. **Foundation for Deployment:** The saved model files are a ready-to-use engine that can be easily plugged into a website or application for real-world use.
5. ***Deployment Interface Explanation***

**The web application is designed to be incredibly simple and user-friendly, guiding the user through a three-step process to get their personalized career advice.**

**1. Upload Your CV Section**

* Purpose: This is where the user provides their information to the system.
* How it Works:
  + The user sees a clear area labeled "Upload your CV (TXT file)".
  + They can either drag and drop a file from their computer directly onto this area, or click to browse and select a file manually.
  + The instruction "Limit 200MB per file • TXT" informs them of the technical requirements. For this initial version, the system works with simple text (.txt) files to avoid the complexity of parsing different formats like PDF or Word. The user can easily create a .txt file by copying and pasting their resume text.
* **User Feedback: Once a file is successfully uploaded, the system confirms it by displaying the filename, its size, and a success message: "✅ CV uploaded successfully!". This immediate feedback is crucial for assuring the user that the process is working.**

**2. Recommendation Trigger Button**

* Purpose: This button gives the user control to start the analysis when they are ready.
* How it Works:
  + After uploading their file, the user sees a button labeled "Get Career Recommendations" or similar.
  + The user must click this button to proceed. This separates the "upload" step from the "analysis" step, making the process feel intentional and controlled.
  + A screenshot of a computer

    AI-generated content may be incorrect.Clicking this button is the trigger that tells the backend AI system to start processing the uploaded CV.

**3. Results Display Section**

* Purpose: This is the most important part—it delivers the final, personalized result.
* How it Works:
  + After a brief moment of processing, the results appear in a clean, easy-to-read format under a header like "Top 5 recommended specializations for this CV:".
  + The results are presented as a ranked list of job titles, from most to least suitable.
* **Key Features of the Results:**
  + Ranked List: The career path deemed to be the best fit is shown at the top.
  + Confidence Scores: Each recommendation includes a percentage (e.g., 57.30%). This is the AI model's confidence level in its prediction. A higher percentage means the system is more certain that the user's skills are a strong match for that specific role.
  + **A black background with white text

    AI-generated content may be incorrect.**Multiple Options: Providing a top 5 list is much more valuable than a single result. It shows the user their strongest potential path and also reveals other viable alternatives they might want to explore.