Skill Categorization Assessment Report

By: Arnav Chopra
arnavchopra2610@gmail.com
9889987111

Requirements:

- After observing and studying the problem statement carefully I listed down my requirements on priority basis from top to bottom.
- Firstly, I needed a dataset that satisfies my needs and the project's requirements.
- Secondly, I had to define a tech stack on the basis of which the model will be trained and evaluated.
- At last I wanted to finalise a deployment technique. (I have used both Fast API and Python Script in this case)

Approach:

- Data Collection: As I could not find a dataset that consisted both Skills and Category at one place I synthetically generated my own dataset.
- Text-Preprocessing: Lowercasing, removing symbols like commas and brackets, transforming skills like ML -> Machine Learning, C++ -> C Plus Plus, C# -> C Hash, Handling stop words and other such EDA, and at last Tokenization.
- Training ML Model: I trained and tested my data on various models and chose the one which gave the best accuracy (SVM : 96.23%) Other model choices were: Regression: 95.37, Random Forest: 91.12%, Naïve Bayes: 94.59, kNN: 88.03% Also, observed misclassifcations using Confusion Matrix.
- Hyperparameter Tuning: Applied hyperparameter tuning on my model to improve accuracy.
- For categorization I formed three functions, categorize_skills_dynamic, predict_category and assign_unknown_category_name

- Loaded the models into an API using Fast API for deployment.

Challenges Faced and How I Solved Them:

- First challenge that I faced was finding the dataset. It took me around 3 days to decide over my dataset. I searched all over the web, on every open source like Kaggle, Hugging Face, Google Dataset, O*Net. Also tried AI tools like DeepSeek. I even searched through research papers related to skill categorizations and resume analysis using NLP but had no luck there. In the end I decided that I had a deadline to take care of and cannot waste anymore time over this as Training and Preprocessing also need time, hence started to design a synthetic dataset and worked on it.

- The second challenge I faced was that the amount of data that I had generated synthetically was of around only 250-300 rows which is not at all enough to train a model like this. Thus I applied Data Augmentation Techniques to my dataset which increased my dataset size and it was ready for further action.

- Moving forward the next challenge was creating a new dynamic column everytime the model could not predict the skill's category. I was able to solve this via some help from github and GPT tools.

- After some evaluation of the model I saw that sometimes the skills that my model cant predict, it is by default categorizing as Technical Skills, I solved this using Probabilty Functions that if a probability of a skill to lie in the columns is less than a gradient then it should form its own column.

- Last issue that I faced was during API forming and Deployment. Being a college student I have worked considerably less in deployment and publishing hence faced some issues. One that I still couldn't solve due to lack of expertise is that some functions of my model lack behind when I use it in API but works just perfectly on code and IDE's

Evaluation Results:

- Below is the classification report of the SVM model:
  1. F1-score: 96.2%   2. Precision: 96%   3. Accuracy: 96.23

Conclusion: I have tried to make this report concise yet descriptive about my thought process and the issues that I faced. I certify all details mentioned in this report are legit.