My approach to identifying the relevant technical skills in the raw dataset was first doing some initial EDA. All of the given data was processed by creating a Jupyter Notebook and importing all of the data with Python. I looked through the Example_Technical_Skills.csv and found some repeated patterns: each of the words in the skill phrase were either capitalized, or spaced with a dash (-) or parentheses (( ). I also noticed a lot of the longer strings having camelcase formatting (ex. MySQL) and acronyms (ex. EAC). Looking through the raw dataset, I noticed almost all of the irrelevant/obviously incorrect technical skills had formatting errors, the most pertinent being the lack of capitalization when starting a new word in each of the respective phrases (ex. What ifs, code versioning tools).

I initially wanted to train a logistic regression model with scikit-learn to differentiate technical and non-technical skills, but the raw dataset didn't give any direct classification as to which was 100% right or wrong for the training set. So I decided for this assignment to keep it simple and focus on the major aspects: the case of non-capitalization.

I decided to create a function is_capital that would go through each of the skills in the raw dataset and check to see if there existed a word in the phrase that had its first letter lowercase. I created an initially empty array called indicator_arr. If the skill contained a word starting with a lowercase letter, I would append the value 0 to indicator_arr. If the skill did not contain a word starting with a lowercase, I would append the value 1 to indicator_arr.

I then combined the raw dataset and indivator_arr together, and filtered out the rows that contained the value 0 (the skills classified as non-technical). I then extracted the skills from the remaining rows to finally extract the technical skills. The number of skills from the raw dataset to the final list went from 34k to 13k.

This could have been done in a better way, such as including more features and making a more complex ML model. There are cases where this model is easily bypassed, such as random capitalized one letter words that don't actually correspond to any relevant technical software, API, etc. (ex. AWS vs Celery).