**Building a recommendation engine**

**Machine Learning pipeline:**

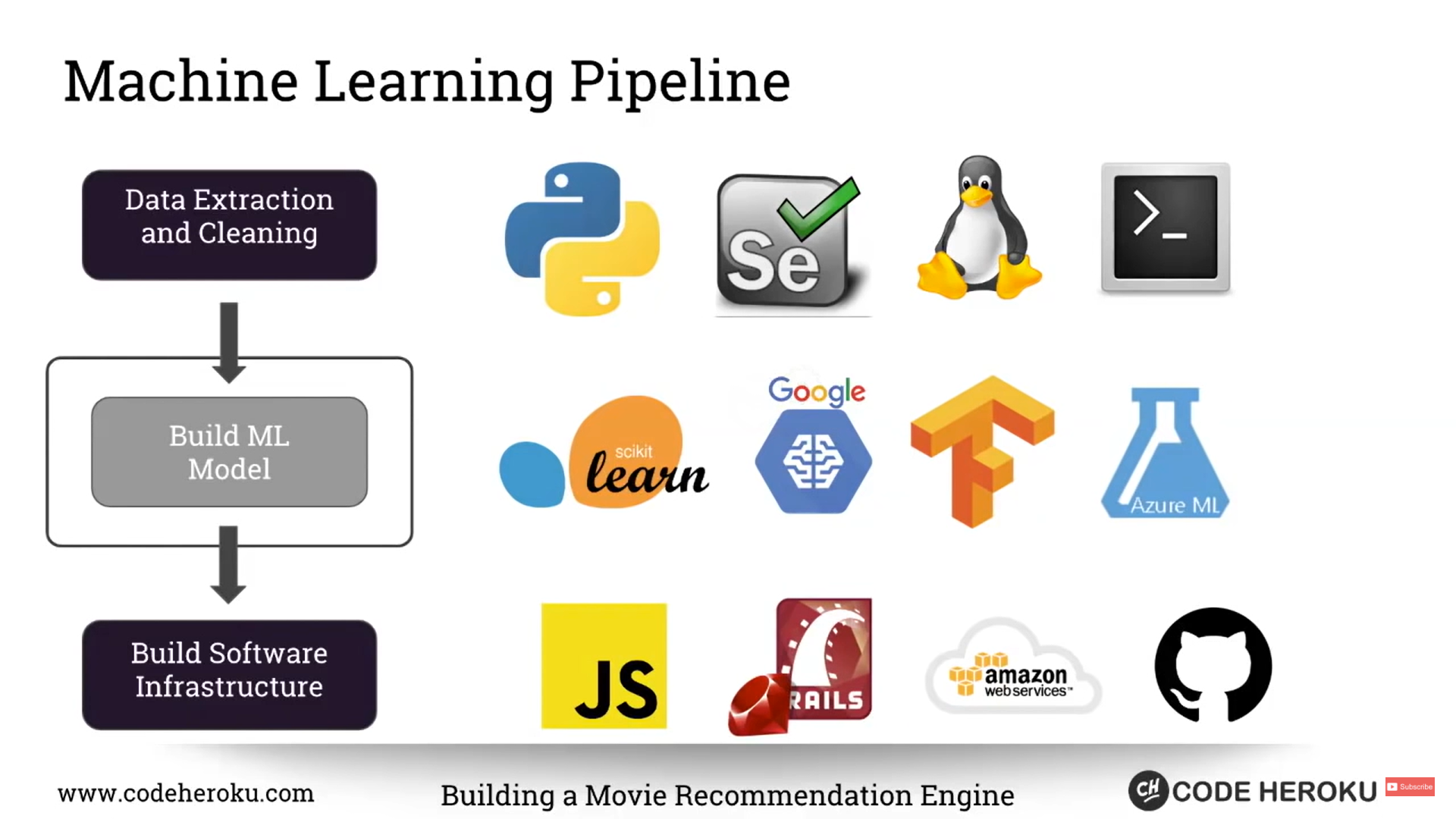
**Extracting and cleaning data:** Real life data is dirty(too many null values, duplicates etc) therefore it needs to be clean.

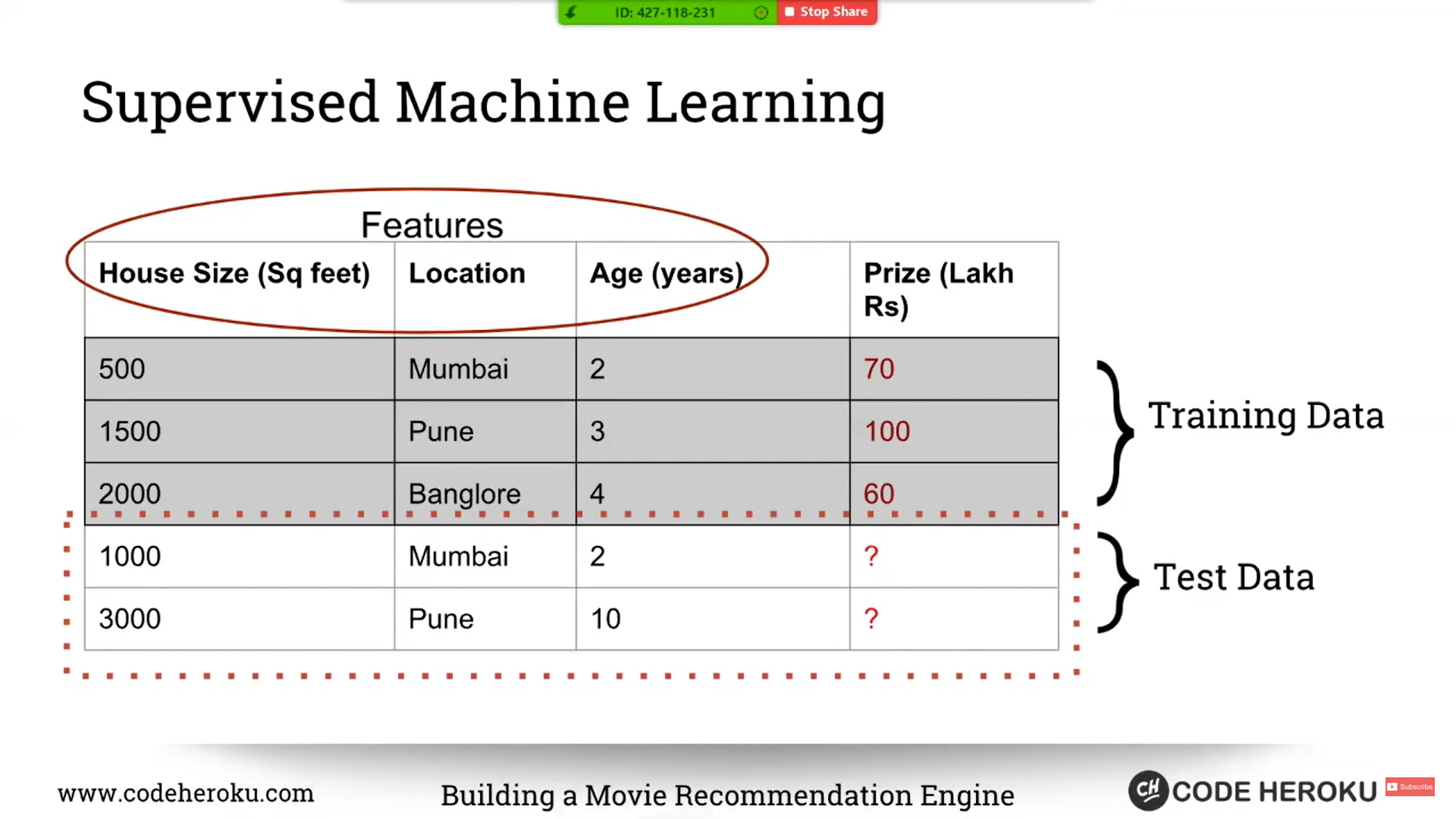
**Tools used**: Python, Shell etc

**Build ML model:** To train the data

**Tools used:** scikit learn, azure ML, tensor flows(build your own model)

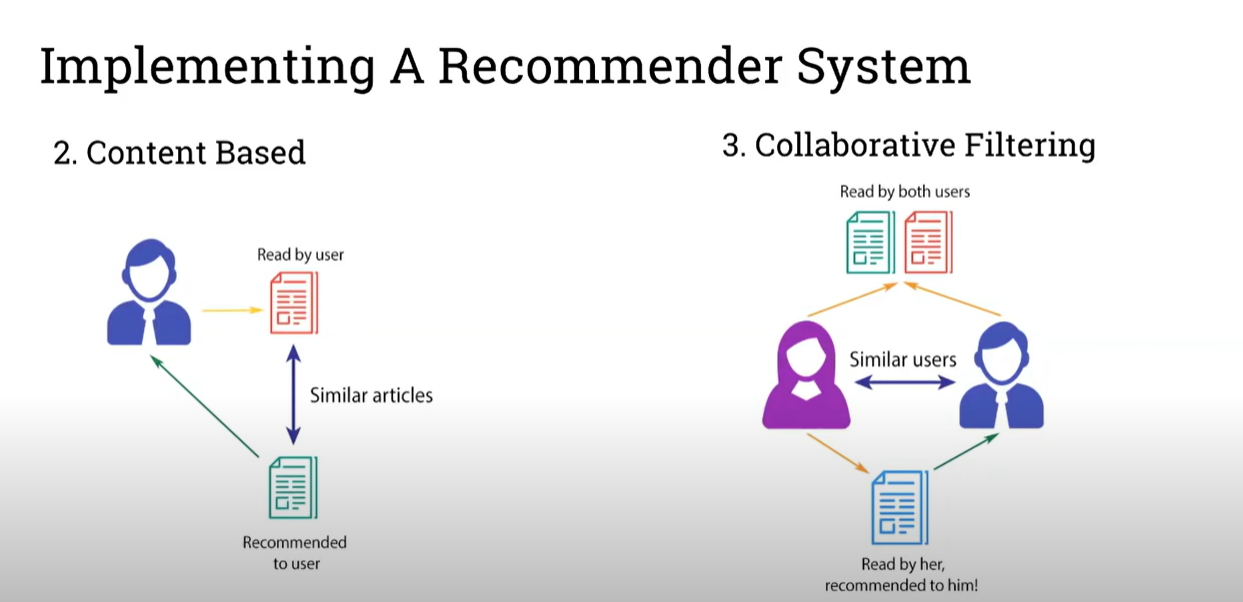
**Build Software infrastructure:** Deploy it on the cloud or create an application for the users.





**Types of recommendation system:**

1. **Popularity based/trending lists**: Based upon the number of users
2. **Content based:** User has read one book. Find similar books with that book.
3. **Collaborative filtering:** 2 users read the same book. Now if user A reads a new book it will be recommended to user B(“you might also like this”) and vice versa. Netflix uses this algorithm.



In order to find similarity we use either

1. Euclidean distance
2. Cosine distance

The user needs to provide the movie for which he needs recommendations similar to that movie. I used the publicly available dataset of IMDB5000. Recommendations are provided based upon a cosine similarity index. Certain features of the movies such as Cast, Director, Genres are merged into one row. The frequency of each and every word in one record is calculated using the”CountVectorizer” which is a class imported from the Scikit-learn library. The cosine similarity index is calculated with every other movie using the cosine\_similarity\_index method also imported from the Scikit-learn library. Finally the indexes with the highest similarity index are matched with the movie and the titles of those movies are displayed.