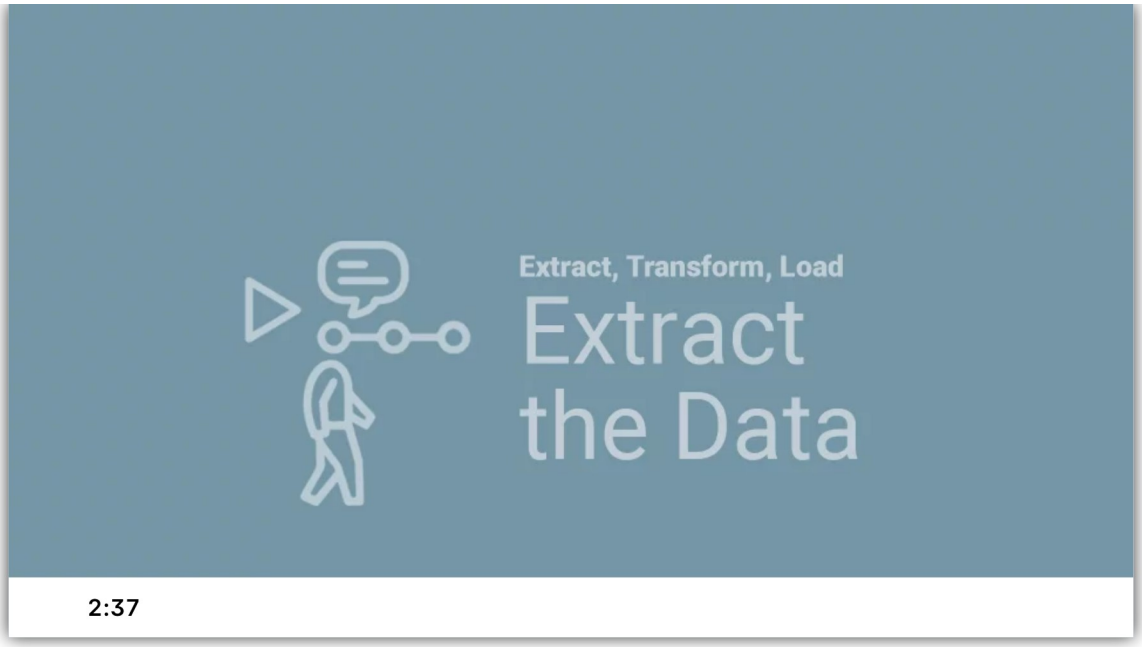
8.2.2 Extract the Kaggle Data

Now that we've loaded the Wikipedia scrape, Britta wants us to include ratings data. However, she knows that her employer, Amazing Prime, won't want to give out their proprietary ratings data to all the hackathon teams. Luckily, she found a dataset on Kaggle that contains ratings data from MovieLens, a site run by the GroupLens research team, which has over 20 million ratings.

Before extracting the Kaggle data, watch the following video to get more acquainted with the Extract step.



MovieLens is a website run by the GroupLens research group at the University of Minnesota. The Kaggle dataset pulls from the MovieLens dataset of over 20 million reviews and contains a metadata file with details about the movies from **The Movie Database (TMDb) (https://www.themoviedb.org/)** . Download the **zip file from Kaggle (https://www.kaggle.com/rounakbanik/the-movies-dataset/download)** , extract it to your class folder, and decompress the CSV files. We're interested in the `movies_metadata.csv` and `ratings.csv` files.

Since the Kaggle data is already in flat-file formats, we'll just pull them into Pandas DataFrames directly with the following code.

```
kaggle_metadata = pd.read_csv(f'{file_dir}movies_metadata.csv', low_memory=F
ratings = pd.read_csv(f'{file_dir}ratings.csv')
```

Inspect the two DataFrames using the `head()`, `tail()`, and `sample()` methods to make sure that everything seems to be loaded in correctly. (We'll do a deeper dive in the Transform step.)

SKILL DRILL

When creating a new DataFrame, you've probably made a habit of using the `head()` method to get a sense of the data and make sure it's imported correctly, and then using the `tail()` method to ensure the data at the end is imported correctly. However, errors can still occur in the middle of the file, so the best practice is to sample a handful of rows randomly using the `sample()` method. For a DataFrame called `df`, `df.sample(n=5)` will show five random rows from the dataset. We'll cover this in more detail later; for now, just focus on getting a rough sense of the data.

Congratulations! You've just completed the Extract step in ETL. Now we're ready to tackle Transform. And when we get to Load, we're going to use many of the same ideas we just used to extract the data.