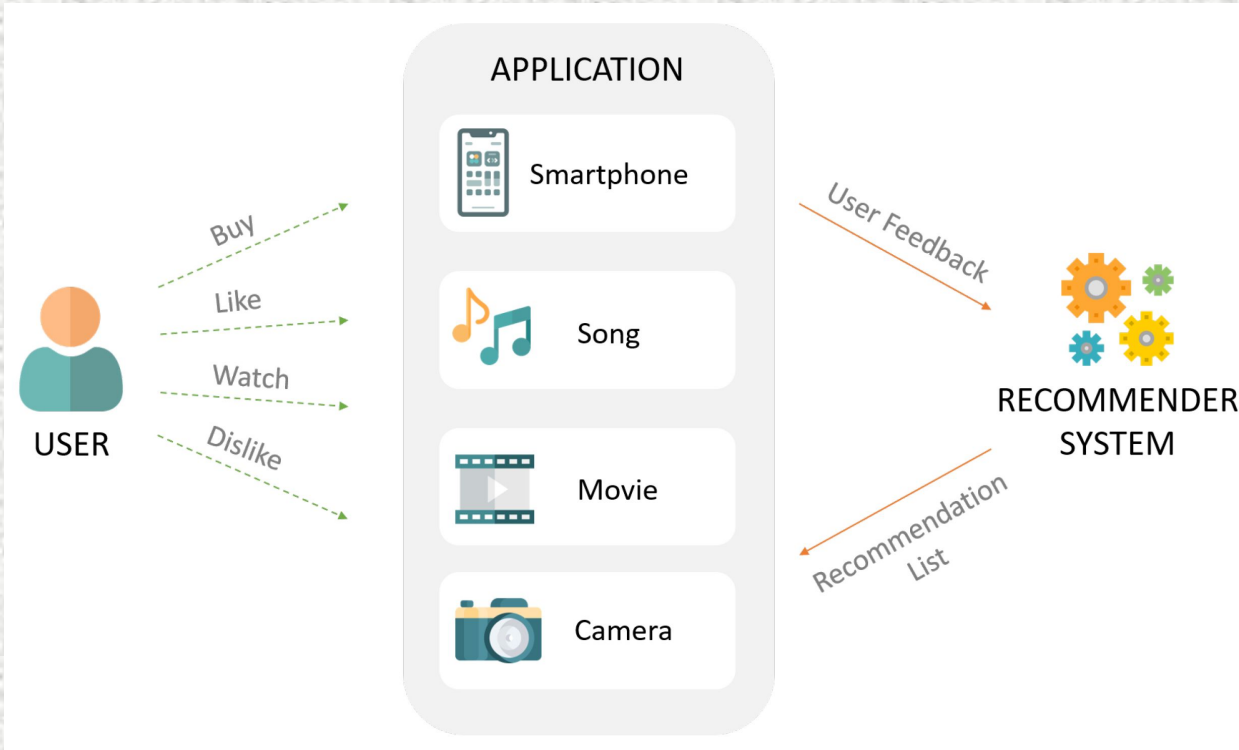


Recommender System with TensorFlow

Osama Mohammed AL-Atraqchi



uses Big Data to suggest or recommend additional items to users. These can be based on various criteria, including past information, search history, demographic information, and other factors.

Recommender systems are highly useful as they help users discover products and services they might otherwise have not found on their own.

Importance

- 40% of app installs on Google Play
- 60% of watch time on YouTube
- 35% of purchase on Amazon*
- 75% of movie watches on Netflix*

Collaborative Filtering

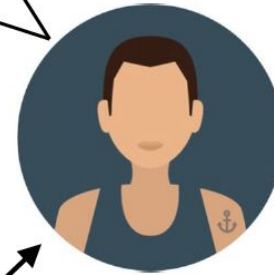
watched by both users



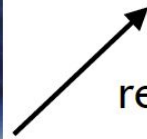
watched
by her



similar users



recommended
to him



Content-based Filtering

watched by user



Genres:

- Comedy
- Romance

Cast:

- Tom Hanks
- Meg Ryan

similar
movies

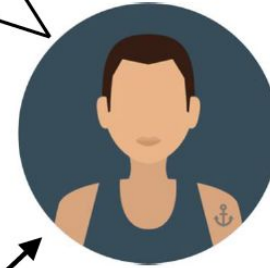


Genres:

- Comedy
- Romance

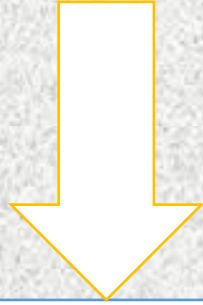
Cast:

- Tom Hanks
- Meg Ryan

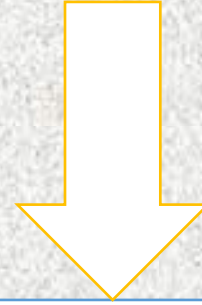


recommended
to him

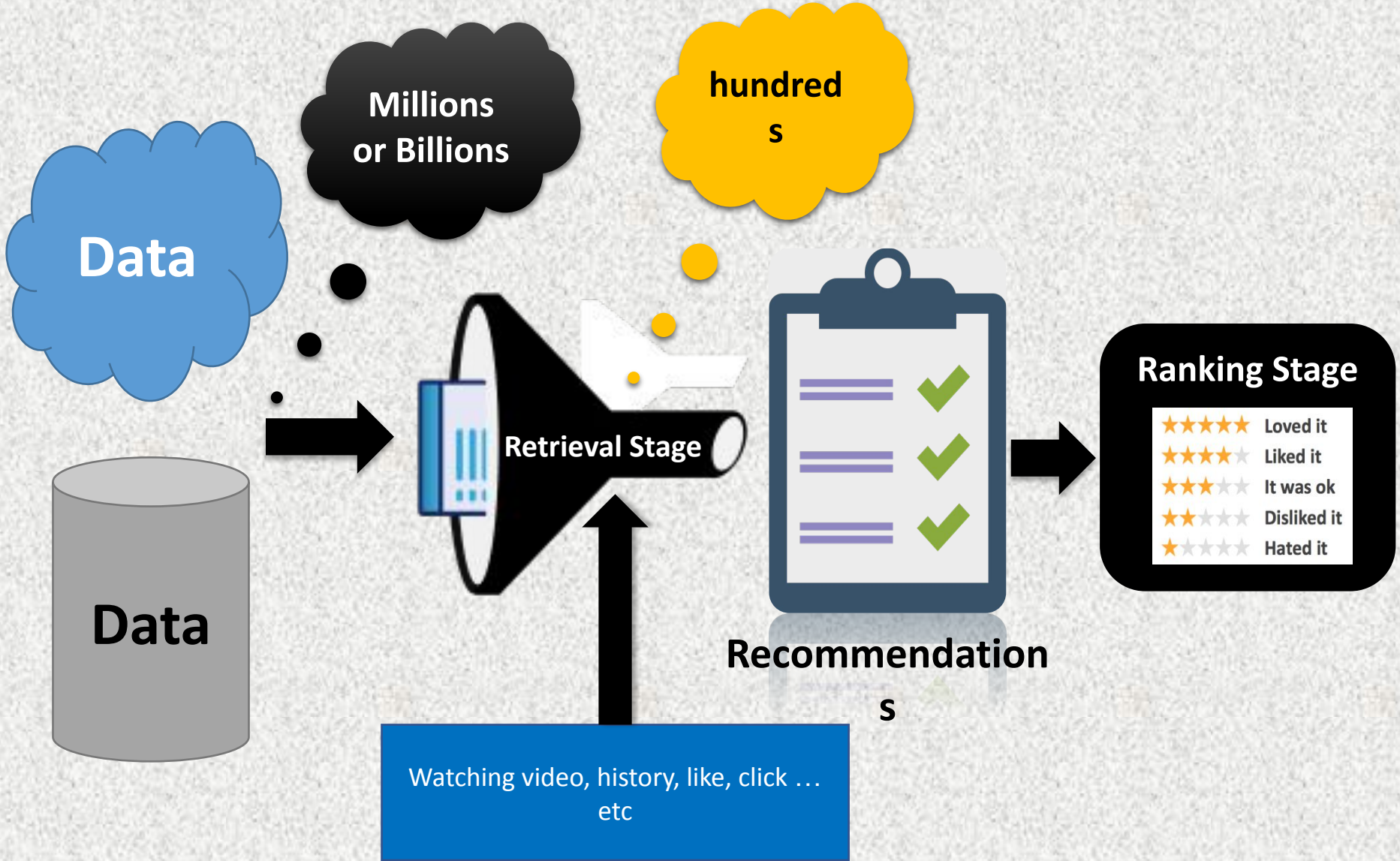
Recommendation System Stages



The **retrieval stage** is responsible for selecting an initial set of hundreds of candidates from all possible candidates (Millions or billions).



The **ranking stage** takes the outputs of the retrieval model and fine-tunes them to select the best possible recommendations (short list recommendations) and ranks them.



TensorFlow Recommender

TensorFlow Recommenders (TFRS) is a library for building recommender system models.

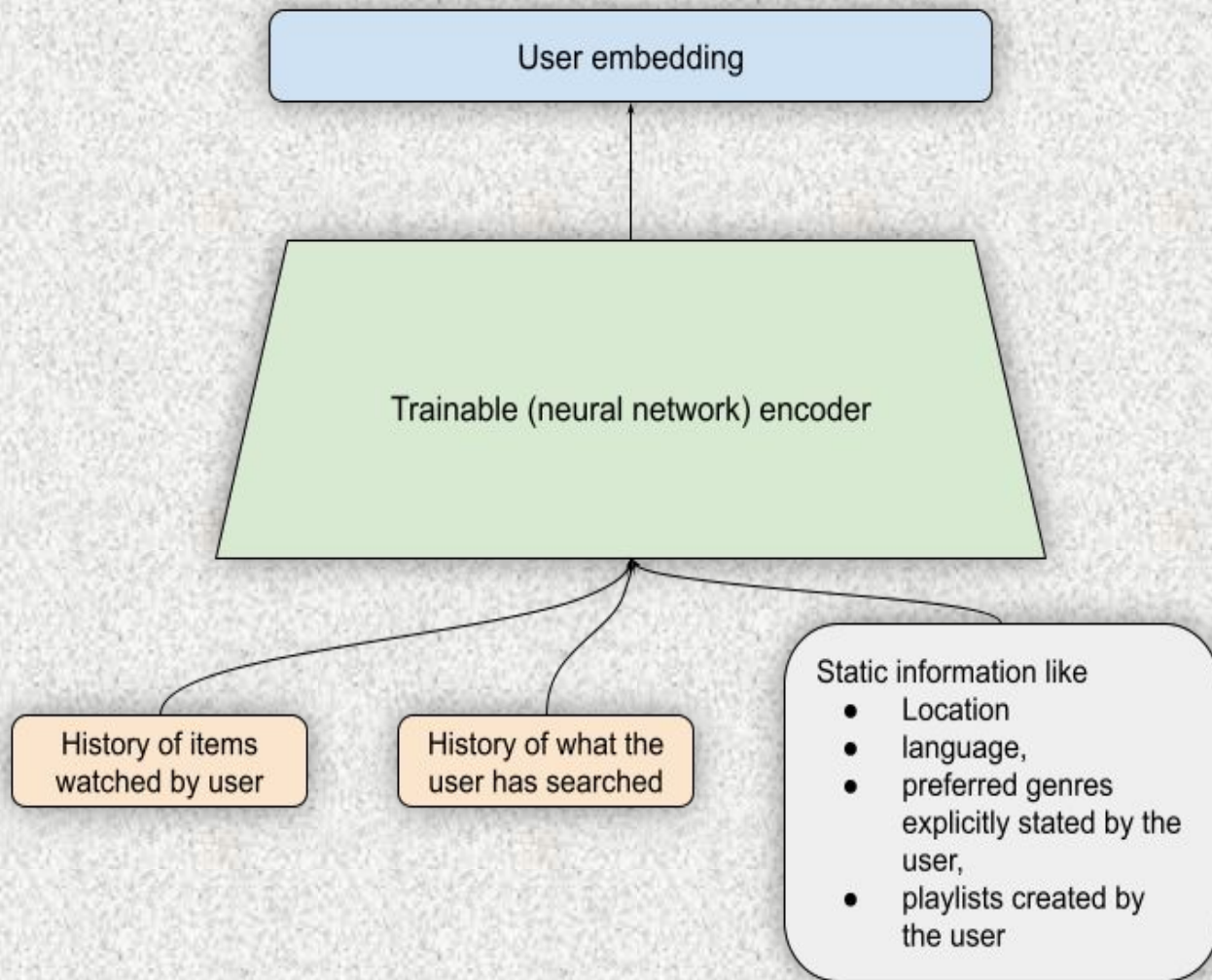
It helps with the full workflow of building a recommender system: data preparation, model formulation, training, evaluation, and deployment.

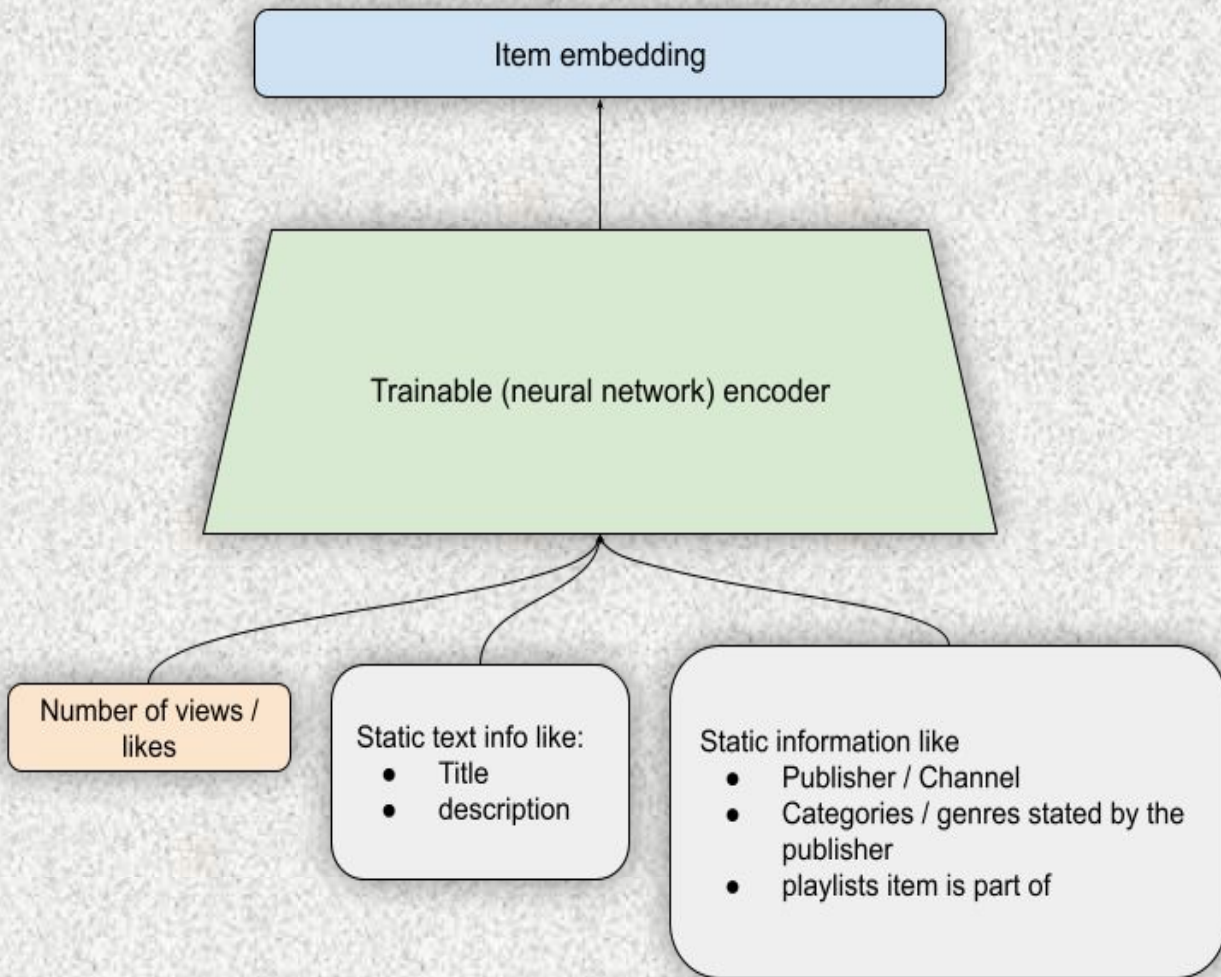
It's built on Keras and aims to have a gentle learning curve while still giving you the flexibility to build complex models.

TensorFlow Recommender

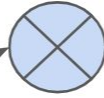
Retrieval Stage

two-tower model: a neural network with two sub-models that learn representations for users and candidates separately. The score of a given user-candidate pair is simply the dot product of the outputs of these two towers.





Sigmoid,
dot product



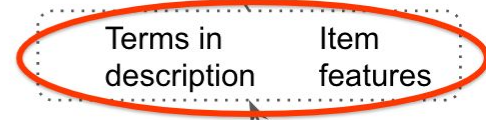
User Embeddings

Item Embeddings

Info about the user



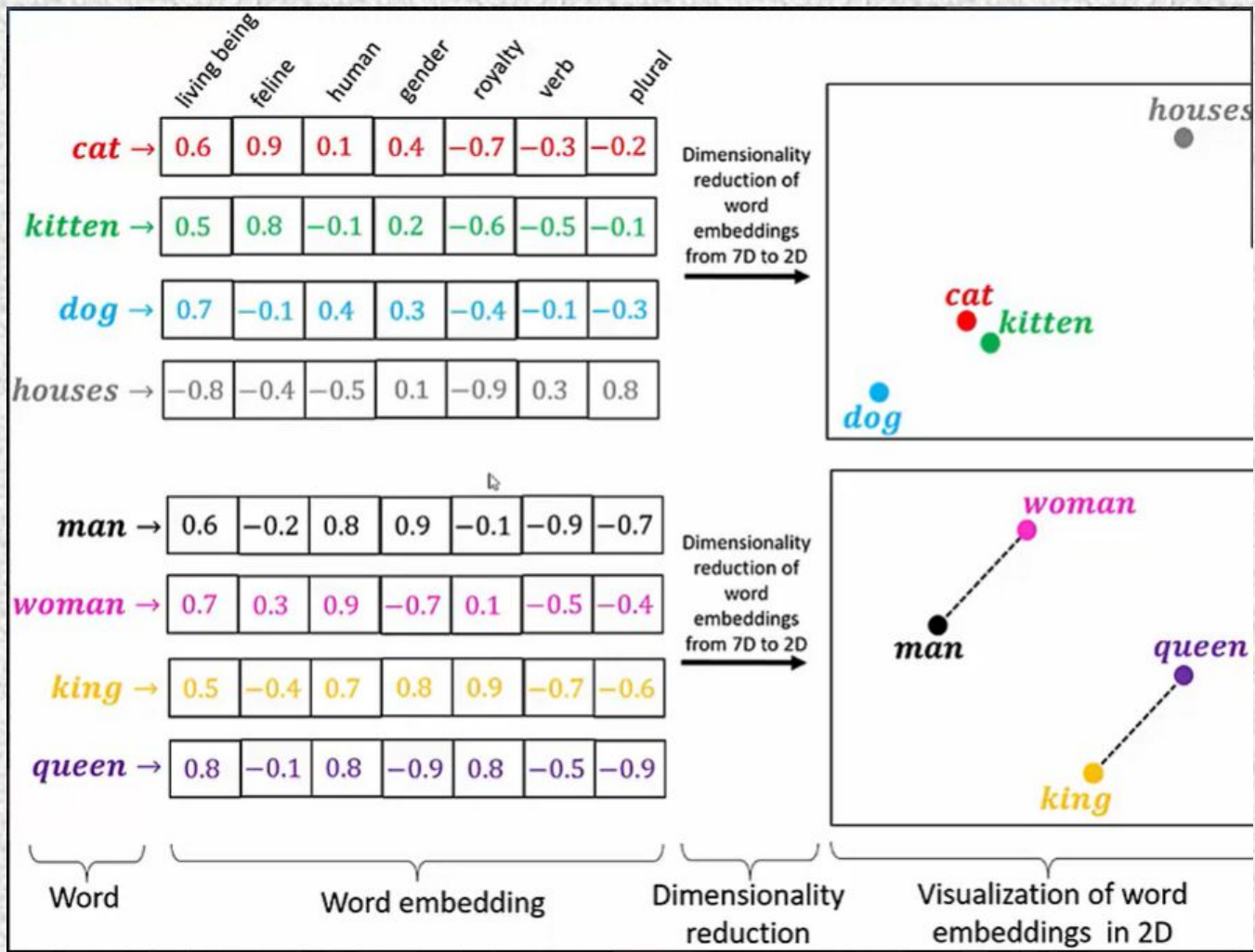
Info about the item



Embeddings
of items in
user history

Embeddings of
terms in documents
in user history, or
terms searched by
the user.

Embeddings
of terms in
the document



```
import tensorflow_recommenders as tfrs

# Load data on movie ratings.
ratings = tfds.load("movielens/100k-ratings", split="train")
movies = tfds.load("movielens/100k-movies", split="train")

# Build flexible representation models.
user_model = tf.keras.Sequential([...])
movie_model = tf.keras.Sequential([...])

# Define your objectives.
task = tfrs.tasks.Retrieval(metrics=tfrs.metrics.FactorizedTopK(
    movies.batch(128).map(movie_model)
))

# Create a retrieval model.
model = MovielensModel(user_model, movie_model, task)
model.compile(optimizer=tf.keras.optimizers.Adagrad(0.5))

# Train.
model.fit(ratings.batch(4096), epochs=3)

# Set up retrieval using trained representations.
index = tfrs.layers.ann.BruteForce(model.user_model)
index.index_from_dataset(
    movies.batch(100).map(lambda title: (title, model.movie_model(title)))
)

# Get recommendations.
_, titles = index(np.array(["42"]))
print(f"Recommendations for user 42: {titles[0, :3]}")
```

Recommender System - Jupyter
Notebook