Multiple transformer blocks example:

<https://keras.io/examples/vision/image_classification_with_vision_transformer/>

Saving and loading models in TF:

<https://www.tensorflow.org/tutorials/keras/save_and_load>

Transformer library:

* <https://towardsdatascience.com/tensorflow-and-transformers-df6fceaf57cc>

Skip-gram:

* <https://github.com/leaprovenzano/movie2vec/blob/master/notebook.ipynb>
* - <https://towardsdatascience.com/word2vec-to-transformers-caf5a3daa08a>

Bag of words - content-based:

* https://github.com/emmagrimaldi/Content\_based\_movie\_recommender/blob/master/EG\_project5.ipynb

Autoencoder, content-based:

* <https://medium.com/analytics-vidhya/movie-recommender-system-using-content-based-and-collaborative-filtering-84a98b9bd98e>

Recommendation engine references:

* https://github.com/zishansami102/Recommendation-Engine

Explanations of different recommendation system: https://blog.codecentric.de/en/2019/07/recommender-system-movie-lens-dataset/

From meeting with Boqing:

* <https://keras.io/examples/structured_data/movielens_recommendations_transformers/>
* <https://towardsdatascience.com/anime2vec-a-sequence-recommender-1e0a3e558c44>
* set target words and context words
* NLP domain
  + - Get a central target user and context movie
  + - Get a central target user and context user if they like the same movie they are in the same window
  + - Vector embedding for each user and movie and input in transformer
  + - Set embed dimension to lower may not work from least to most popular, learn embedding in that way, make the user be the first thing in the sequence,
  + - Define by frequency that they give ratings the movie, use the domain knowledge, draw histograms of users like the same movies, how many users could be connected to each other according to this rule to see if its reasoning
  + - The sliding window
  + - Positional relationship
  + -
  + - Why it doesn’t work well, use good scientific reasoning, explain the choices we made
  + - Focus Encoder and preprocessing

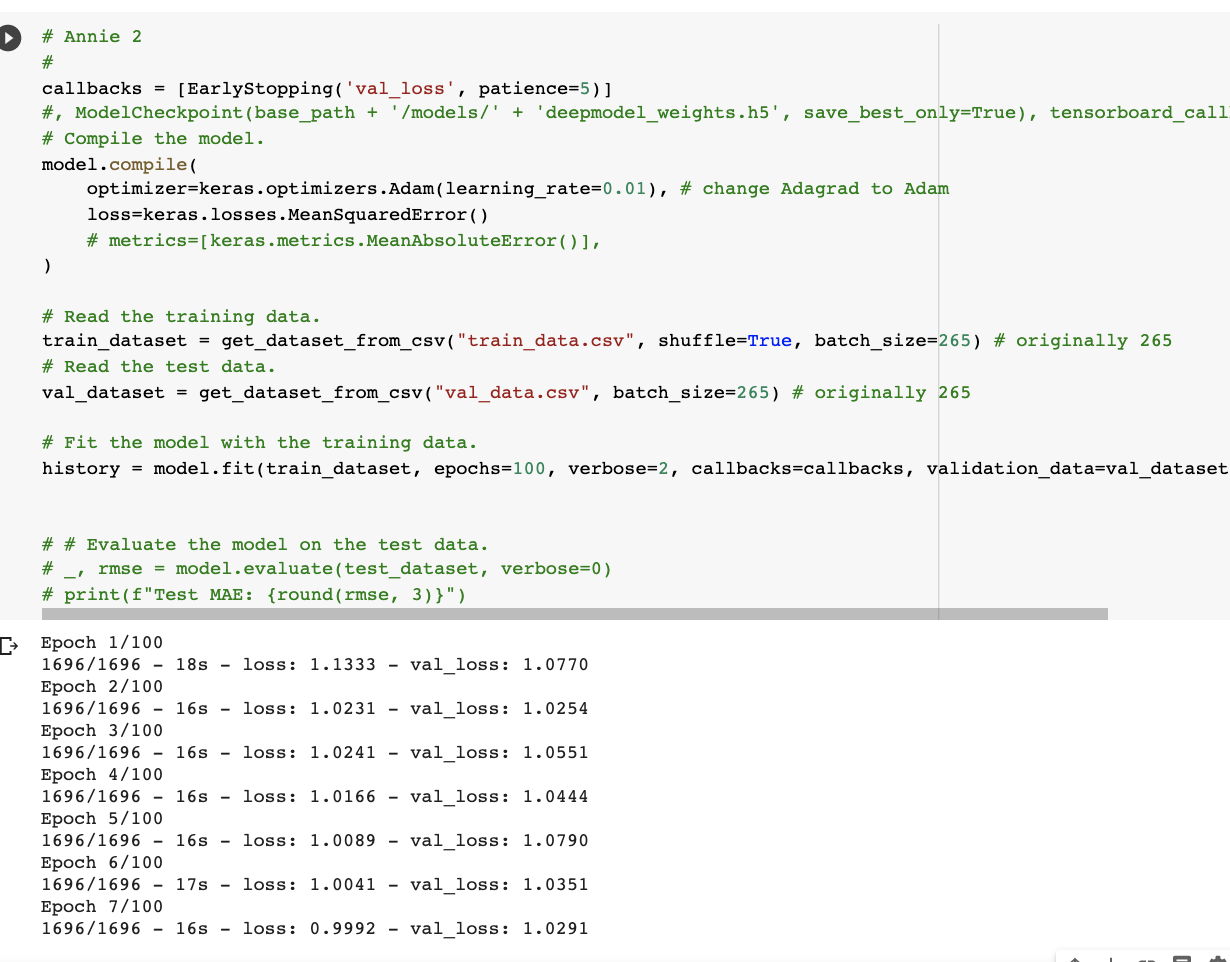
Further ideas::

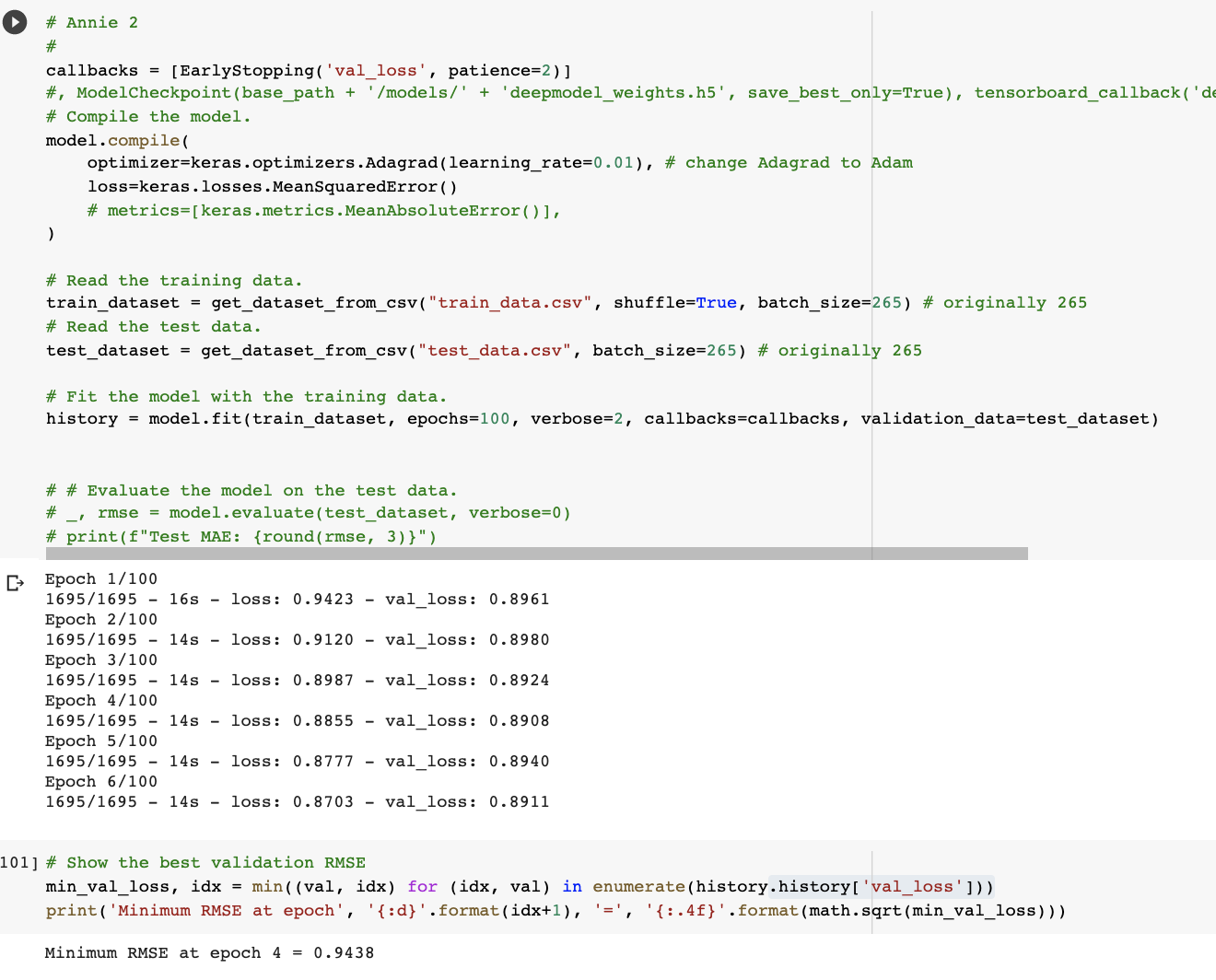
* + increasing the input sequence length
  + training the model for a larger number of epochs
  + include other features like movie release year and customer zipcode
  + including cross features like ‘sex X genre’.
  + From Keras page <https://keras.io/examples/structured_data/movielens_recommendations_transformers/>

Others:

* <http://tech.octopus.energy/timeserio/examples/MovieLens100k.html>
* <https://www.kaggle.com/terminate9298/movie-recommendation-system-for-deployment>
* <https://www.tensorflow.org/tutorials/text/transformer>

Annie’s try

* Changed the splitting to 0.9
* Adam, 3 heads, only 2 dense inside create\_mode, dropout rate = 0.1
* 
* Adam, 5 heads, dense inside create\_model add one more 64 units dense, dropout rate = 0.1:



* hidden\_units = [256, 128,64], dropout\_rate = 0.2, num\_heads = 5