Word2vec MovieLens (implemented): <https://github.com/SarangDeshmukh7/Recommendation-Engine>

AutoEncoder (implemented): <https://github.com/RaptorMai/Deep-AutoEncoder-Recommendation>

References:

<https://irjmets.com/rootaccess/forms/uploads/IRJMETS640440.pdf>

More about training Word2Vec:

https://stackoverflow.com/questions/60852962/training-time-of-gensim-word2vec

Word2Vec for a different dataset, for reference: <https://www.analyticsvidhya.com/blog/2019/07/how-to-build-recommendation-system-word2vec-python/>

Autoencoder for Movielens 1M, tensorflow, collaborative filtering:

* <https://github.com/vitobellini/tfautorec>
* <https://github.com/RaptorMai/Deep-AutoEncoder-Recommendation/blob/master/DeepAE_Rec.ipynb>
* <https://github.com/tyutjxs/AutoEncoder-for-recommend-system>
* <https://github.com/smashound/Movie-Recommendation-VAE/blob/master/finalproject.ipynb>
* <https://github.com/artem-oppermann/Deep-Autoencoders-For-Collaborative-Filtering>
* https://github.com/mkfilipiuk/VAE-CF

Deep autoencoders movielens 100k, pytorch: - <https://github.com/Chinmayrane16/DeepRecommender/blob/master/Training_Deep_AE.ipynb>

Decoding BST:

Embedding explanation:

<https://stats.stackexchange.com/questions/270546/how-does-keras-embedding-layer-work>

BST structure/overview:

<https://hzhaoaf.github.io/data/bst-poster.pdf>

Similar paper: Temporal Collaborative Ranking Via Personalized Transformer (SSE-PT):

https://arxiv.org/abs/1908.05435

SSE-PT Github: Transformer example (own class):

<https://github.com/wuliwei9278/SSE-PT>

Tensorflow tutorial of transformer with positional embedding with generated text

(language translation example - includes decoder):

https://www.tensorflow.org/tutorials/text/transformer