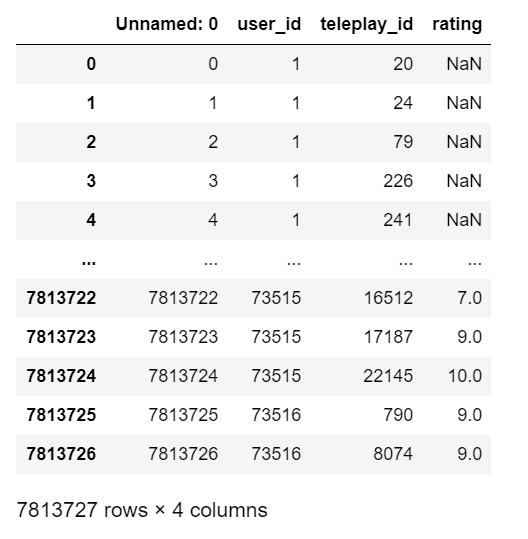
**COMP4434: Project Report**

**Sweta Das**

**Student ID: 18080395d**

**Task1:**

1. Details of Data analytics
   1. Purpose: The aim is to predict ratings of a teleplay using neural network and find average ratings by new users as well, using map reduce.
2. Details of model design and implementation
   1. Pre-processing of data:
      1. Unwanted values: The pre-processing of data is a preliminary step which is taken before analysing the data. This helps to organise the data well into a desired structure, so that while using the data for analysis we do not have to write the coding to ignore the unwanted values, hence being saved from including unwanted values while calculation. Also, if we can convert the string values into numerical values, it will be easier to access the data.
      2. Encoding: For Teleplay.csv, the column ‘type’ includes the type of teleplays into the following categories- long, short, medium, special, music, ONA and Nan. These values have been encoded into numerical values using label encoding method, into the following- [3,4,6,2,0,1,5] respectively. We will ignore the number 1 and 5 as it is ONA and Nan respectively. Converting string values to numbers will enable to make programs easier as we can access the labels easily. Hence, a new column which corresponds to the label of the teleplay is added, having numerical values of the labels.
      3. Invalid ratings: For Rating.csv, the rating for each teleplay (represented by teleplay\_id) is provided by users (represented by user\_id). However, some ratings are mentioned as ‘-1’, representing that it has not been rated by a particular user. Therefore, wherever ‘-1’ appears, those are replaced by NaN.



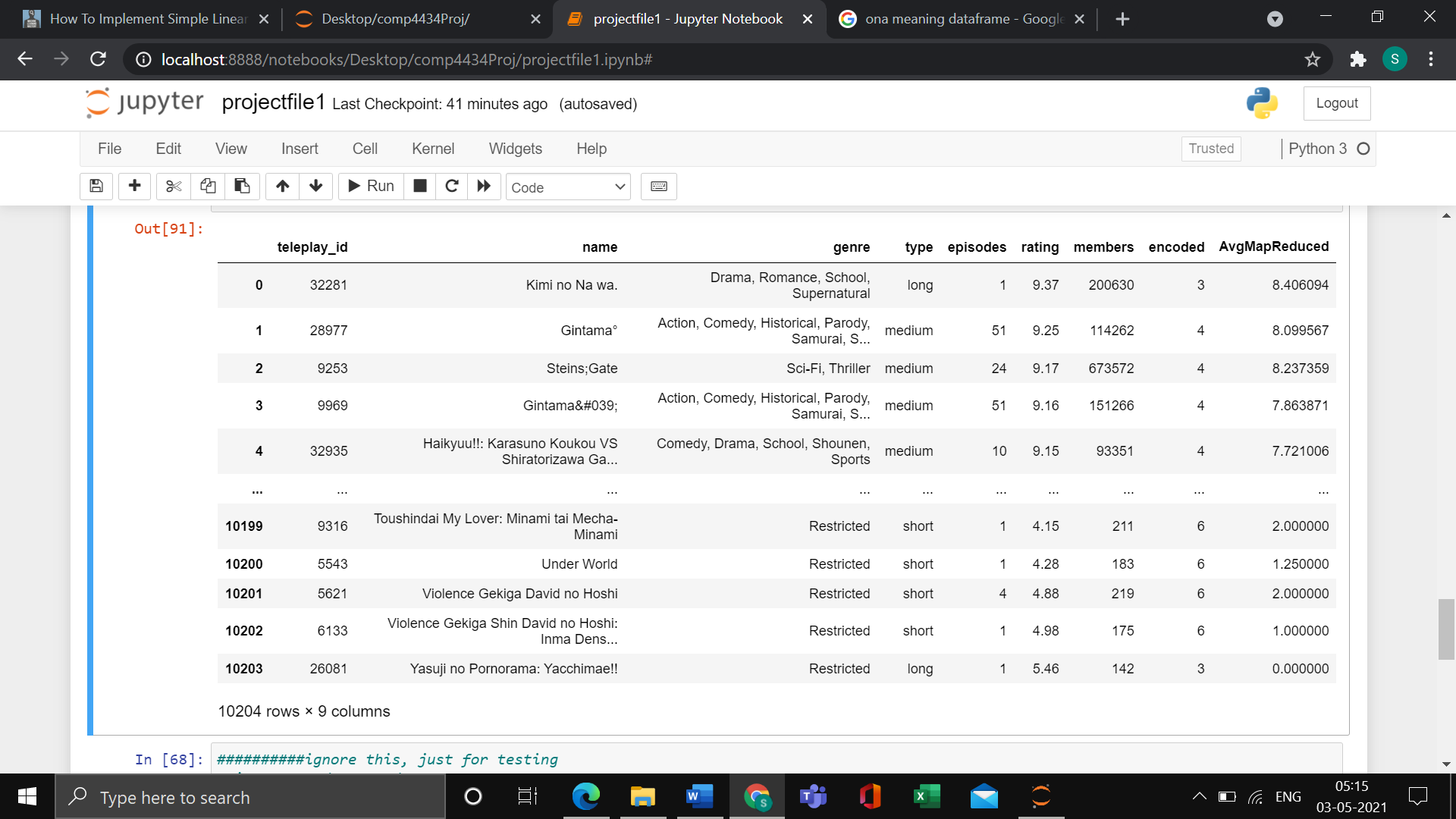
1. Map-reduce:
   * 1. Dictionary to store ratings: This method is used to find the average rating of a teleplay, rated by the users in the Rating.csv file. The aim to find the average rating is achieved by using a dictionary. The ID of the teleplay, mentioned under the ‘teleplay\_id’ column, is taken as the key. The values assigned to the key are the list of ratings given by users. As some ratings are represented by ‘-1’, such value is replaced by 0 when assigning to the list of ratings of a teleplay. The dictionary will appear like-

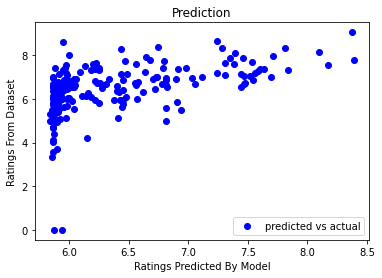
{20: [0,9,8, 10, ….], 1000: [0,0,0,1,2,8….] ………}

* + 1. Dictionary to store average of ratings: After creating a dictionary, another dictionary is created where the key is the ID of the teleplay and value is the average value of the rating of the teleplay. For example, the value of the key ‘20’ will be the average of the numbers in the list:[0,9,8,10,…]. The new dictionary will look like-

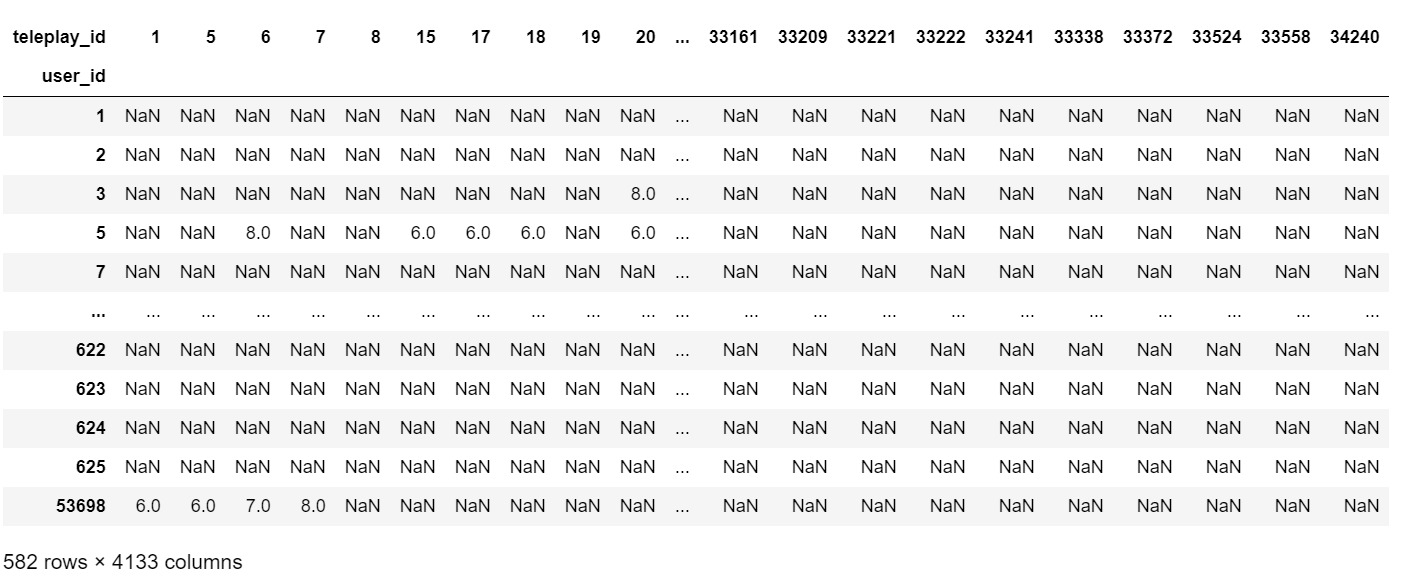
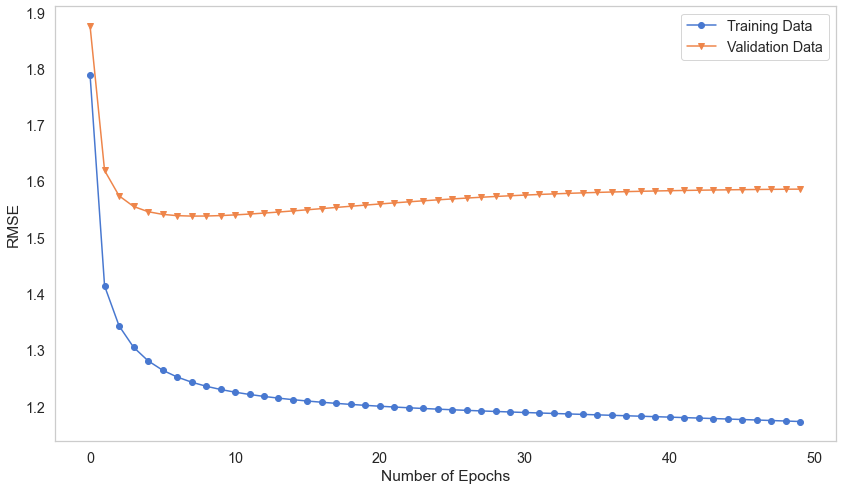
{20:6.8, 1000: 9, 39090: 3, ……}

* + 1. Adding a new column: Hence, a new column ‘**AvgMapReduced**’ is added to the data frame for Teleplay.csv for the average ratings produced by map-reduce. Here is how the data frame will appear:

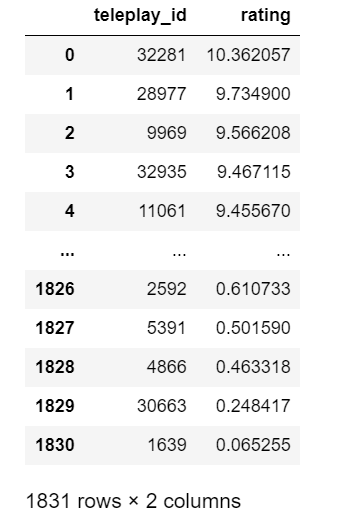


1. Primary Result
   1. Features of neural network: We use fully connected Neural network for the prediction of ratings of the movie. Our neural network has 2 input features- encoded, which is the encoded labels as integers, as the numbers are needed in neural network. Another feature is members, which is the number of members who have given their ratings for that teleplay. The input features represent the independent variables, and the rating represents the dependent variable.
   2. Structure of neural network: The neural network takes the 2 input features, which are the 2 input neurons. There are 3 hidden layers, and one output layer. The first hidden layer includes 32 neurons, second input neuron has 16 neurons, and third hidden layer has 4 neurons.
   3. Other measures of the network: The optimiser function used is Stochastic Gradient Descent, and the loss function used to represent loss is Mean Square Error. We perform 10 epochs on the neural network. Performing too many epochs will result in overfitting.
   4. Result: The MSE value is approximately 1. This seems reasonable and not too large. Just for comparison purpose, we tested with a linear regression model, which has MSE of approximately 1.6. Hence, as a proof, the neural network model is better functioning. Below is the plot of actual versus predicted values: 
2. Summary of discoveries and future work
   1. Improvement in accuracy: The accuracy will improve if we encode the type using one-hot encoding. This will not consider the weights in different values unlike in label encoding. To be more precise, the genre can be encoded instead of type.
   2. Use of other models: For better comparison, we can use other models like decision trees, which is used famously as classifier and for regression.

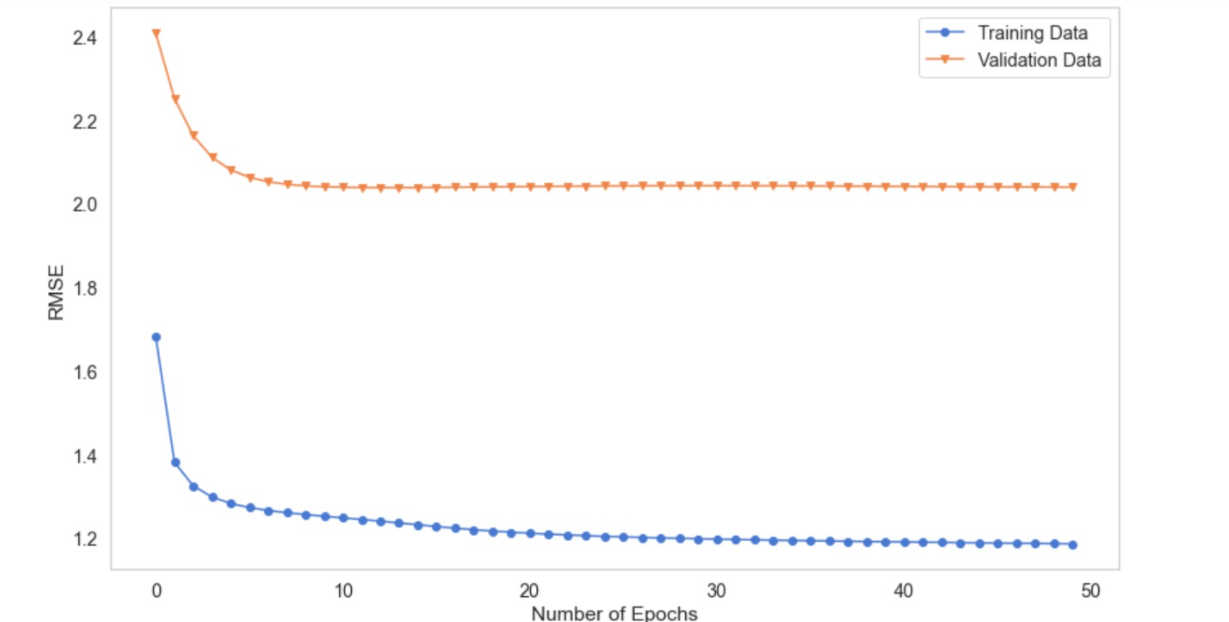
**Task 2:**

1. Problem definition
   1. Purpose: We use Collaborative filtering recommender system to predict movie ratings to the user having user\_id= 53698
2. Model design and analysis
   1. Idea: The collaborative filtering, or CF is used to predict movie ratings for a user which is based on the ratings and preferences of the other users. If a user has similar tastes as the other user, then the movies that the other user has rated good will be recommended to the current user.
3. Solutions and implementation details
   1. Specifications: The gradient descent algorithm for collaborative filtering is used. It updates the parameters each time it is run. The other features which are the inputs are also updated, like rating.
   2. Pre-Processing: The data points used are initial 60000 data points from both Rating.csv and Teleplay.csv. However, later we add other rows related to the user 53698. This will not let us ignore the data points related to this user and help us in better prediction. The values which were having ratings as -1 are replaced by NaN. A data frame rating\_df is defined. Pivot table is used with index value as user\_id, columns as teleplay\_id, and values as Rating of the teleplays. This is the concerned data frame ratings\_df-However, ratings data frame contains the modified version of this dataframedata frame where NaN values are replaced by 0, so that during running the program, we have only integer values.
   3. Other functions: RMSE function is used to calculate the root mean square values. The class Recommender calculates the updated parameters and the updated features. The function test\_train\_splits separates data into training and testing, for training the model. We finally fit the data into the trained model and represent it by the following plot- 

The function create\_artist\_ratings helps to fit in our original data for predicting ratings for any given user in the dataset. In our case, it is user 53698. These are the predicted ratings for this user for each teleplay-



1. Performance evaluation and discussion
   1. The Graph prediction: The graph shows that the RMSE for training set is decreasing constantly as the epoch value increases. But for validation dataset, the values decrease sharply and becomes almost constant. The values of the test and training data are almost the same initially, but gap increases as epoch value increases. But when we increase the value of MIN\_USER\_RATINGS and DELETE\_USER\_COUNT, the gap between test and training data increases. Proof is the following graph which has the MIN\_USER\_RATINGS and DELETE\_USER\_COUNT values as 1000-



1. Summary of discoveries and future work
   1. Improvement- If more fields are taken into consideration, the accuracy will be better. And the RMSE will be lower. Also, if we include the whole dataset and run on a more powerful computer, performance will improve.
2. References-

##Video uploaded on YouTube- <https://youtu.be/cA6Zo3-P8zg>

Without the following references, the project would have been incomplete-

* Labs and assignment codes of COMP4434 at The Hong Kong Polytechnic University.
* <https://stackoverflow.com/questions/43288550/iopub-data-rate-exceeded-in-jupyter-notebook-when-viewing-image>
* <https://stackoverflow.com/questions/45785745/python-how-to-close-my-csv-input-and-output-files>
* <https://support.microsoft.com/en-us/office/filter-data-in-a-range-or-table-7fbe34f4-8382-431d-942e-41e9a88f6a96>
* <https://stackoverflow.com/questions/7353743/make-dictionary-from-csv-file-columns>
* <https://www.cs.cornell.edu/courses/cs1110/2016sp/lectures/03-17-16/13A.ListsOfNumbers.pdf>
* <https://www.calculatorsoup.com/calculators/statistics/mean-median-mode.php>
* <https://www.cs.cornell.edu/courses/cs1110/2016sp/lectures/03-17-16/13A.ListsOfNumbers.pdf>
* <https://stackoverflow.com/questions/7353743/make-dictionary-from-csv-file-columns>
* <https://stackoverflow.com/questions/9039961/finding-the-average-of-a-list>
* <https://stackoverflow.com/questions/33686747/save-a-list-to-a-txt-file>
* <https://www.journaldev.com/32742/python-remove-duplicates-from-list>
* <https://www.foxinfotech.in/2018/09/python-read-csv-columns-into-list.html>
* <https://www.javaer101.com/en/article/18702605.html>
* <https://stackoverflow.com/questions/43288550/iopub-data-rate-exceeded-in-jupyter-notebook-when-viewing-image>
* <https://machinelearningmastery.com/implement-simple-linear-regression-scratch-python/>
* <https://github.com/cugzj/Simple-Linear-Regression/blob/master/LR_example.py>
* <https://towardsdatascience.com/linear-regression-from-scratch-cd0dee067f72>
* <https://www.freecodecamp.org/news/building-a-3-layer-neural-network-from-scratch-99239c4af5d3/>
* <https://www.theanalysisfactor.com/outliers-to-drop-or-not-to-drop/>
* <https://docs.anaconda.com/anaconda/user-guide/tasks/tensorflow/>
* <https://stackoverflow.com/questions/43030363/valueerror-could-not-convert-string-to-float>
* <https://towardsdatascience.com/gradient-descent-from-scratch-e8b75fa986cc>
* <https://machinelearningmastery.com/gradient-descent-optimization-from-scratch/>
* <https://github.com/k8xu/movie-rating-prediction/blob/master/Prediction_of_Movie_Ratings_Among_Young_Adults.ipynb>
* <https://stackoverflow.com/questions/15891038/change-column-type-in-pandas#:~:text=The%20best%20way%20to%20convert%20one%20or%20more%20columns%20of,floating%20point%20numbers%20as%20appropriate>.
* <https://www.interviewqs.com/ddi-code-snippets/rows-cols-python>
* <https://stackoverflow.com/questions/45271344/importerror-no-module-named-keras>
* <https://stackoverflow.com/questions/49008074/how-to-create-a-neural-network-for-regression>
* <https://stackoverflow.com/questions/42379138/keras-valueerror-no-data-provided-for-input-1-need-data-for-each-key/50666476#50666476>
* <https://stackoverflow.com/questions/61980349/tensorflow-typeerror-cannot-unpack-non-iterable-float-object>
* <https://stats.stackexchange.com/questions/259263/keras-numpy-error-why-does-accuracy-go-to-zero#:~:text=It%20shouldn't%20go%20to,%2C%20but%20it%20doesn't>.
* <https://stackoverflow.com/questions/12236566/setting-different-color-for-each-series-in-scatter-plot-on-matplotlib>
* <https://stats.stackexchange.com/questions/521719/dummy-variables-in-regression-analysis>
* <https://github.com/k8xu/movie-rating-prediction/blob/master/Prediction_of_Movie_Ratings_Among_Young_Adults.ipynb>
* <https://machinelearningmastery.com/regression-tutorial-keras-deep-learning-library-python/>
* <https://machinelearningmastery.com/one-hot-encoding-for-categorical-data/#:~:text=For%20example%2C%20in%20the%20case,be%20calculated%20using%20linear%20algebra>.
* <https://towardsdatascience.com/one-hot-encoding-multicollinearity-and-the-dummy-variable-trap-b5840be3c41a>
* <https://github.com/keras-team/keras/issues/2645#:~:text=model.add(Dense(12,in%20the%20FIRST%20hidden%20layer>.
* <https://machinelearningmastery.com/tutorial-first-neural-network-python-keras/>
* <https://keras.io/examples/structured_data/collaborative_filtering_movielens/>
* <https://pythonprogramminglanguage.com/introduction-to-neural-networks/>
* <https://gist.github.com/maunashjani/922e5a2a60130367dc58f1ee1fd6da36>
* <https://www.ritchieng.com/machinelearning-one-hot-encoding/>
* <https://stackoverflow.com/questions/43490495/how-to-set-notebookapp-iopub-data-rate-limit-and-others-notebookapp-settings-in>