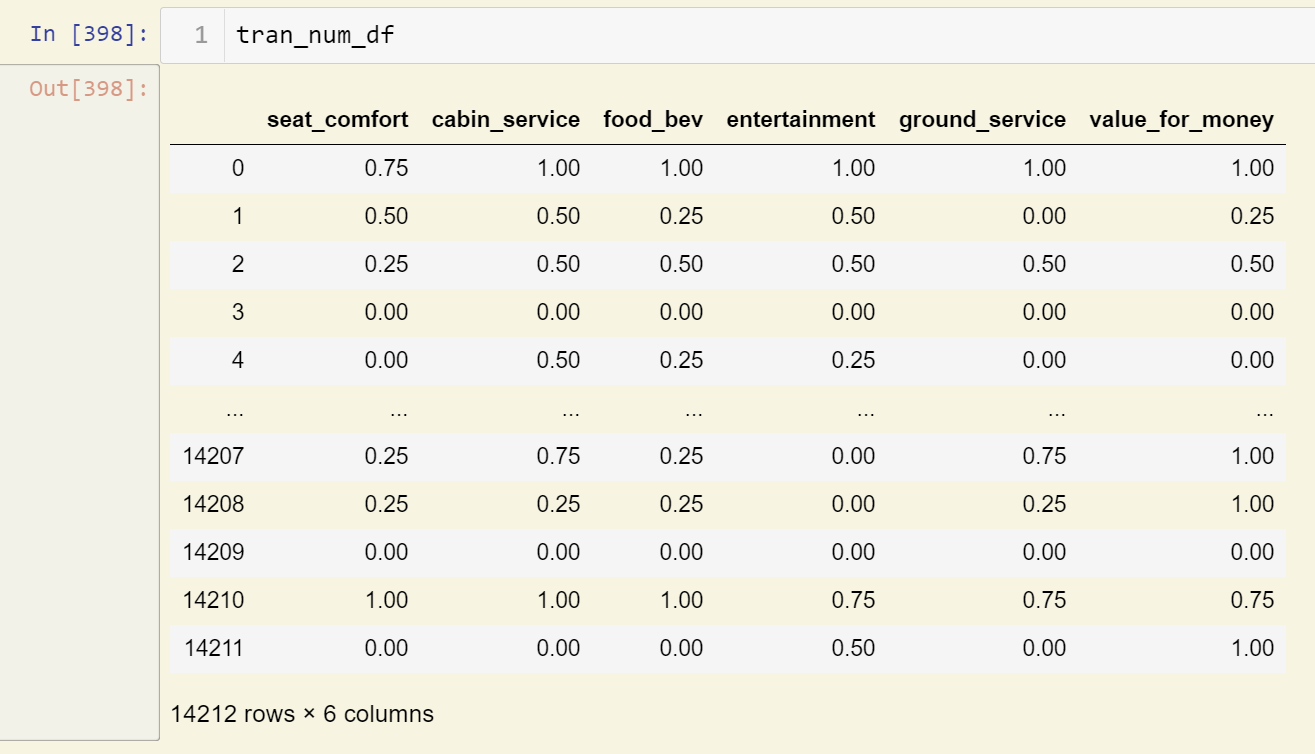
**ASSIGNMENT 2-3**

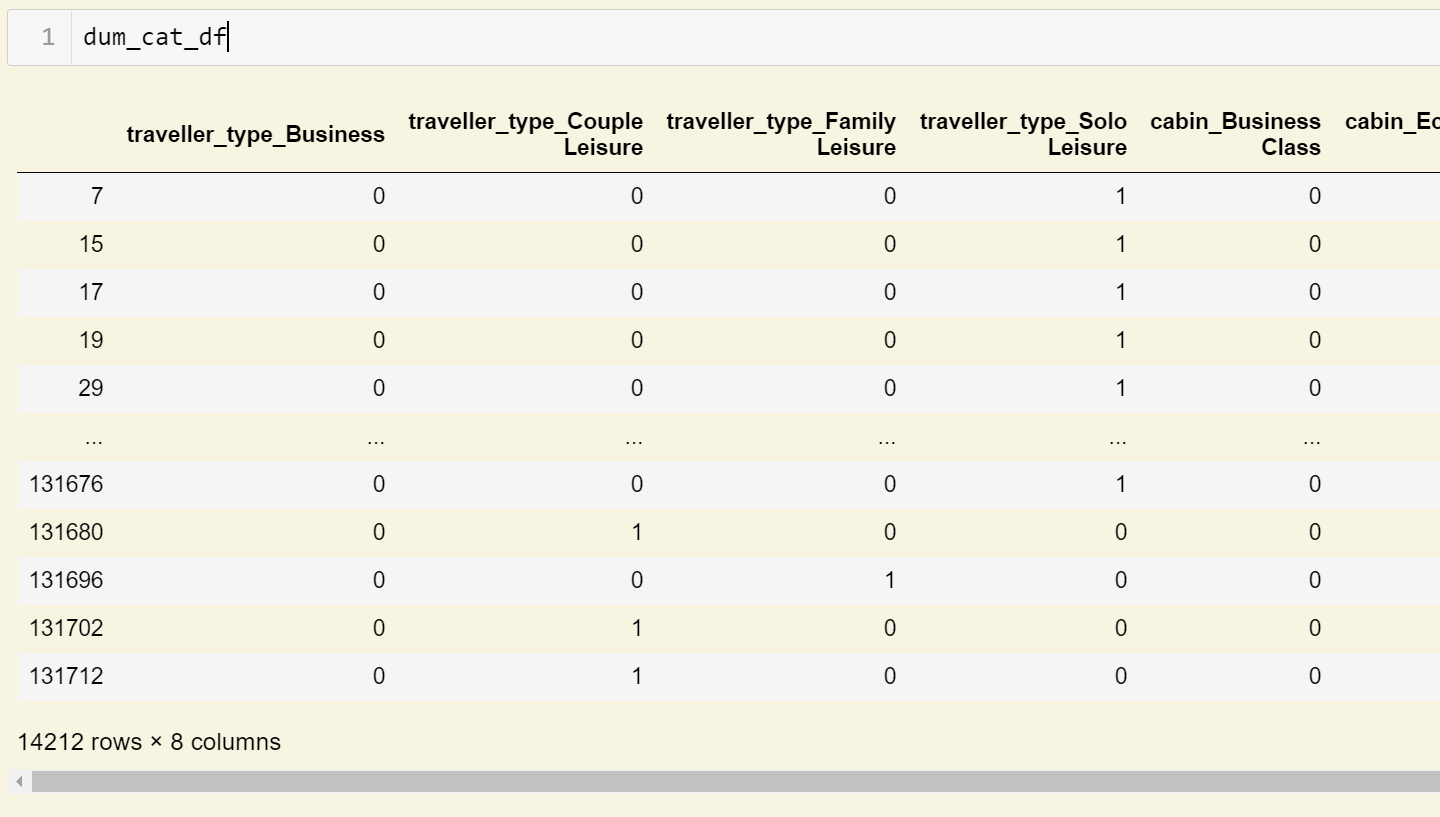
TUNG LAM TRINH - 300267601

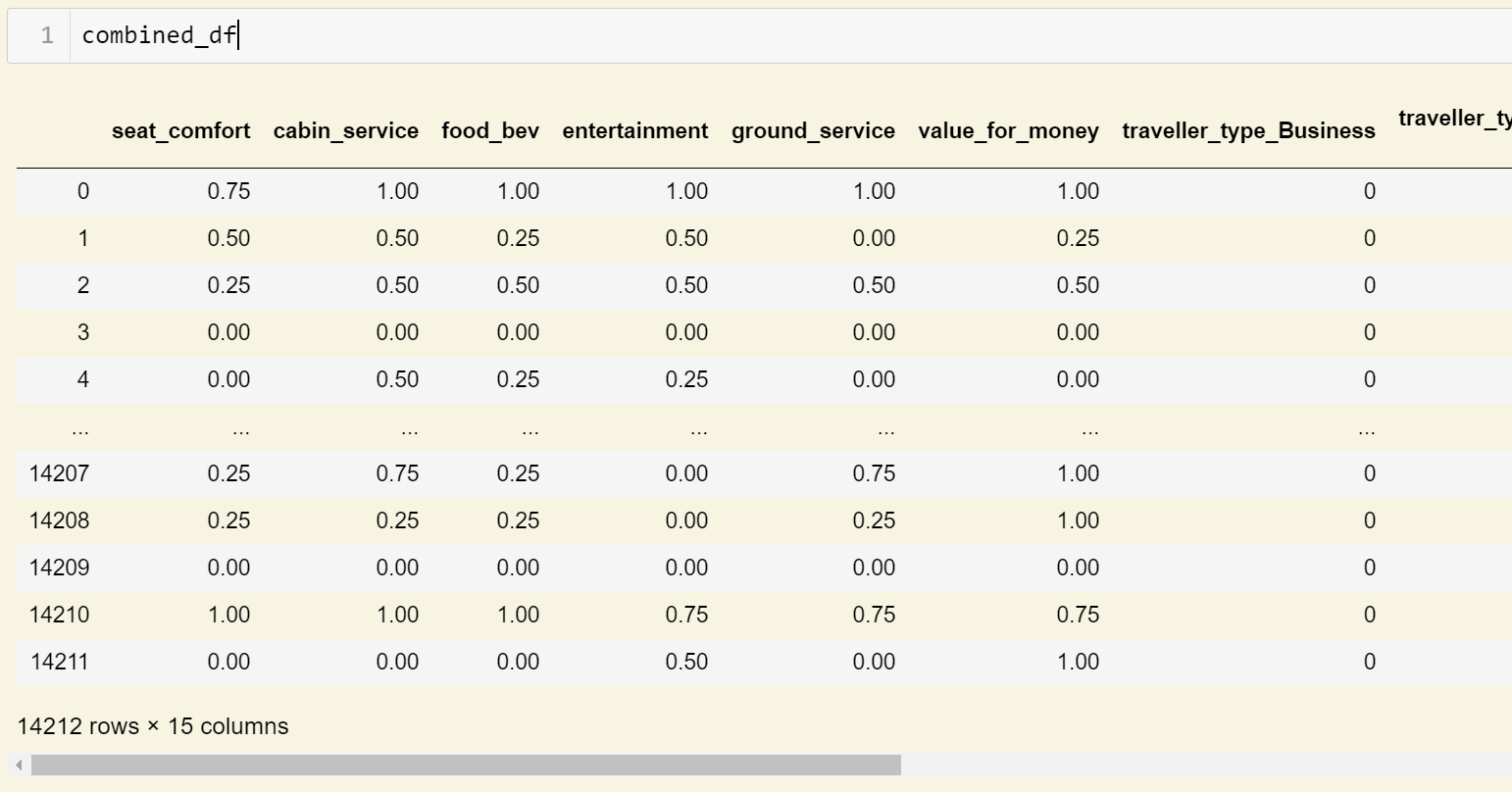
**ANSWER:**

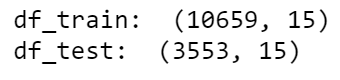
**PART 6: DISCUSSION OF THE RESULTS AND THEIR MANAGERIAL IMPLICATIONS**

1. **Part 2 - Baseline Model 1 Using Numeric and Categorical Features**

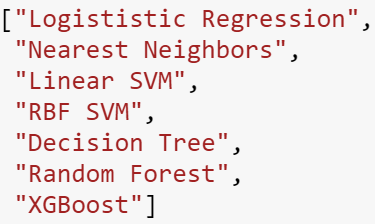
* We combine the scaled numerical dataframe and dummy categorical features data frame (from part 1) into one dataframe.
* Scaled numerical dataframe:
* Numerical dataframe:

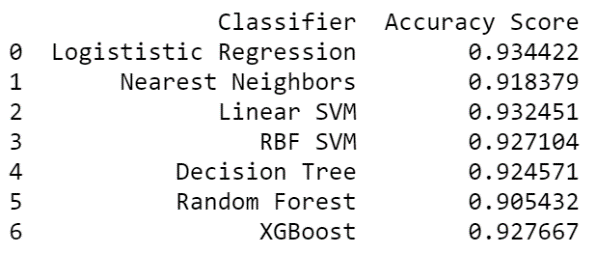


* Combined numerical and categorical dataframe:
* After that, we split the combined data into 75% training sample (10659 rows of data) and 25% test sample (3553 rows of data), and we get the train and test data shape:



* We will run the models pipeline based on train and test data above. There are 7 models:



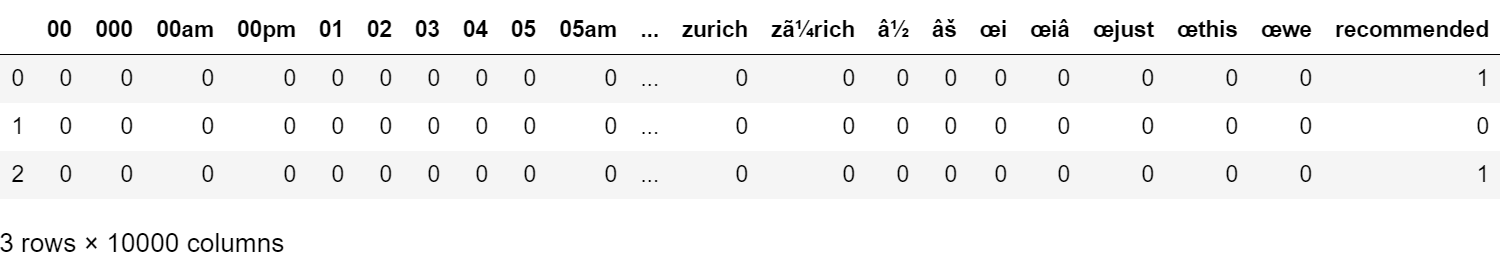
* After running models, we get the model results which are the Accuracy Score of each model from the loss-function:

As we could see, the accuracy score of all models are very high, which means the models we used are fit and relevant to our study: predicting the ‘Recommended’ feature.

And we might want to choose the highest score model (Logistic Regression model) to predict output from input of the future practical data.

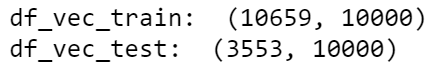
1. **Part 3 - Baseline Model 2 Using Text Features**

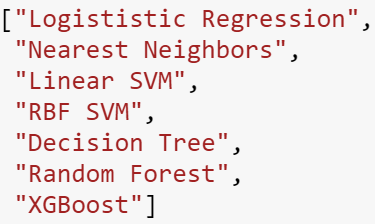
* Instead of using combined numerical-categorical data, we will use text data from customer reviews to predict the outcome ‘Recommended’ feature.

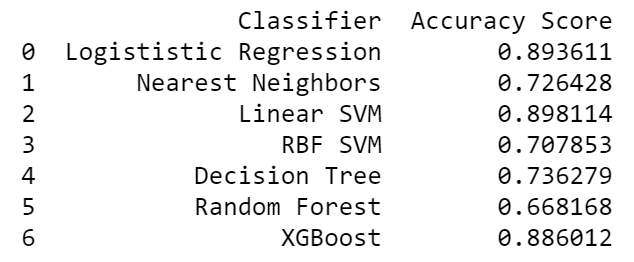
Count-Vectorizer data will be used to run the models:

The last column is ‘Recommended’ output feature transformed to dummy variable (YES is 1 and NO is 0)

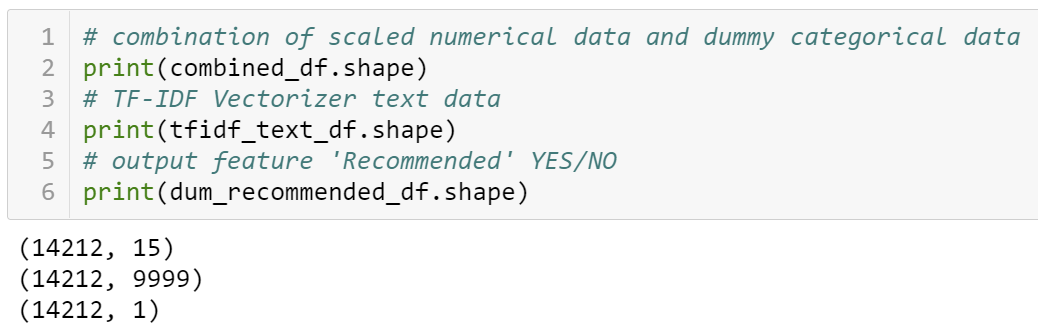
* There are also 7 models will be applied to the train and test data which are splitted from Count-Vectorizer data (75% train data and 25% test data)

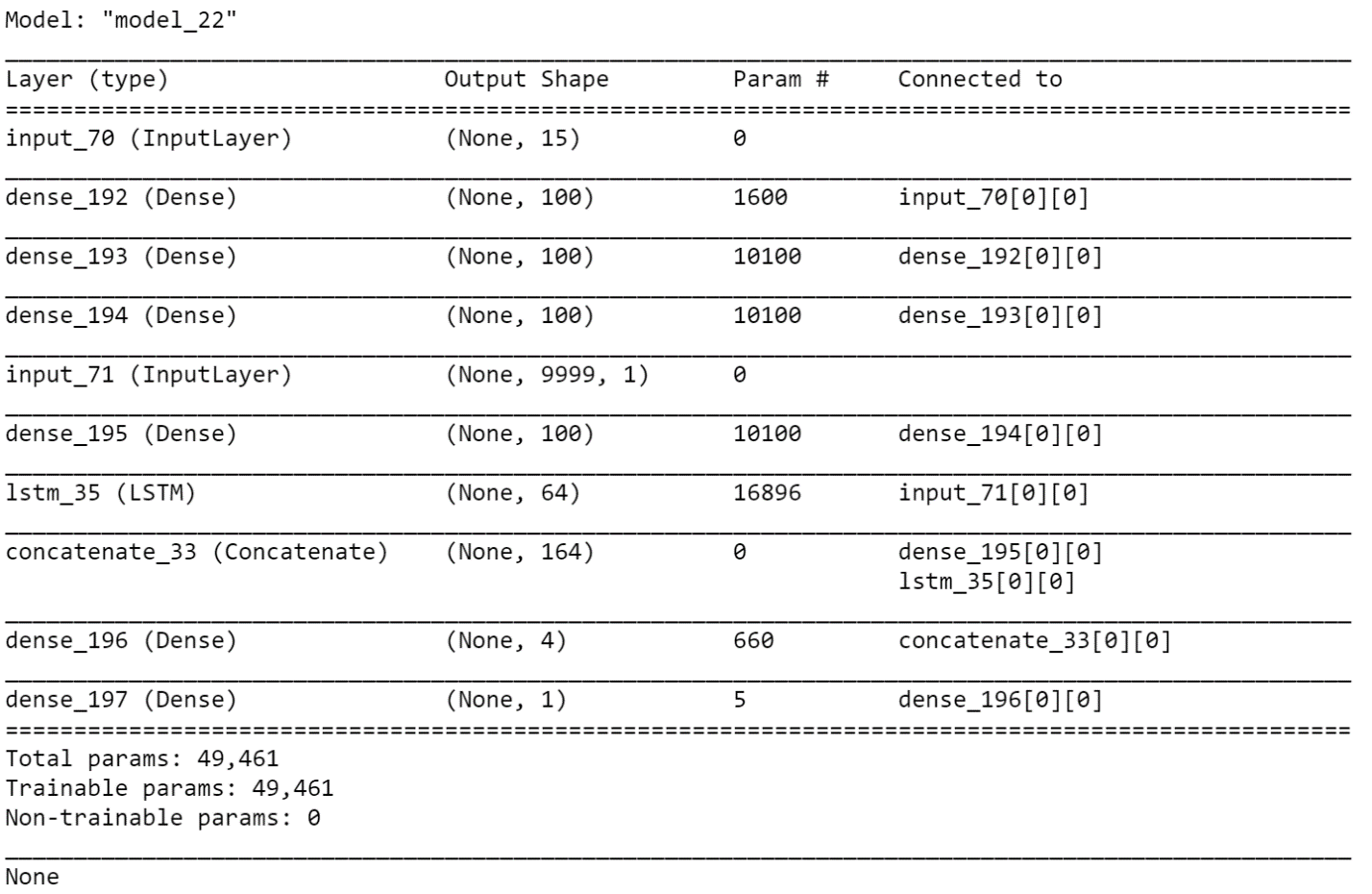




* After running models, we get the model results which are the Accuracy Score of each model from the loss-function:
* Model’s scores are quite high, however as we could see, they are lower than model’s scores of the numerical data. That means using combined numeric and categorical data would have more accuracy.

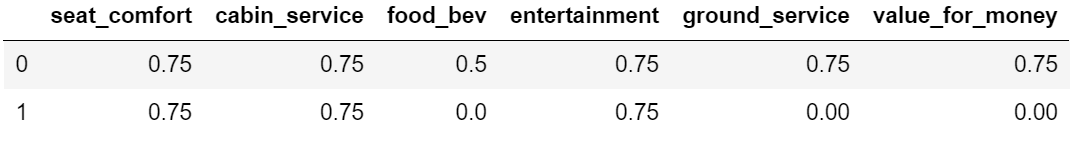
1. **Part 4 - Mixed Input Deep Learning Model**

* From previous parts, we are going to use 3 dataframes for the Mixed Input Deep Learning Model:
* Splitting 3 dataframes to 75% train data and 25% test data for our running Mixed Input Deep Learning Model purpose.
* After compiling the sequence process of the model, the result is shown as below:

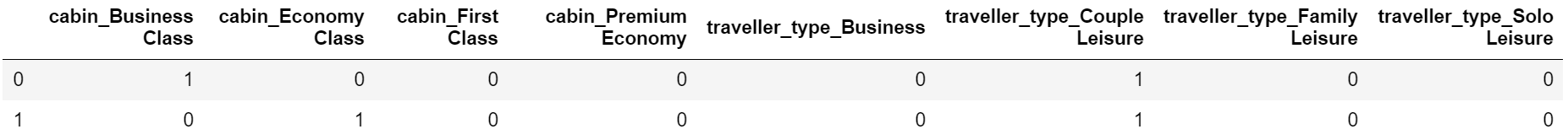


1. **Part 5 - Making Predictions (out-of-sample) with the Model**
2. Scaled numerical sample

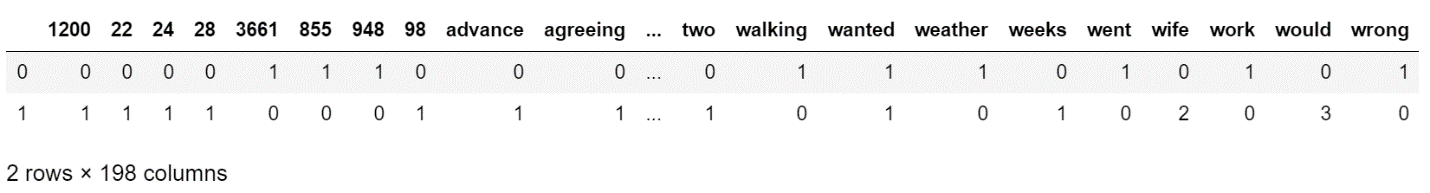
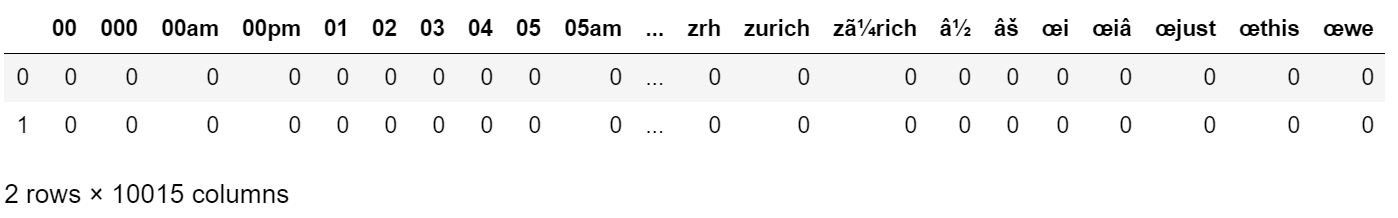
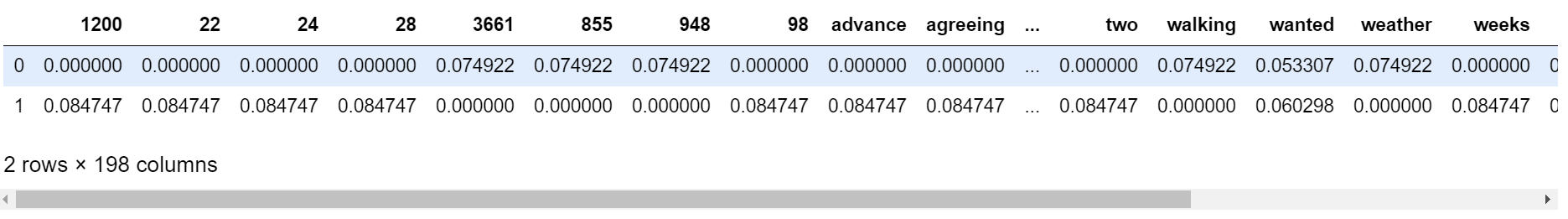
* We scale the numerical data using MinMaxScaler() and based on the scaled data from part 1 (we add sample data to original numerical data (part 1), and scale this data, and extract 2 rows of scaled sample data)
* The result of scaled sample data:



1. Dummy categorical sample

* We convert categorical sample data to dummy categorical data following the original dummy cat data (part 1)
* The result of dummy cat data:

1. Text review sample

* Using python-docx library to read the docx file and extract the reviews of 2 objects
* After removing the unnecessary words from 2 reviews, we have the dataframe
* We need to convert them to CountVectorizer or TfidfTransformer for further studies such as the ‘Recommended’ output prediction.
* The CountVectorizer data and TfidfTransformer data following the original CountVectorizer and TfidfTransformer data columns
* CountVectorizer sample
* TfidfTransformer sample