Reproducing results of the paper “Machine Learning for Fluid Property Correlations: Classroom Examples with MATLAB” published by Lisa Jobs and Erich A. Muller in 2019 at American Chemical Society Publications

Dan Ni Lin, dan_lin@usp.br

MOTIVATION

As motivation for reproducing the results of Lisa Jobs and Erich A. Muller paper was because I wanted to learn about machine learning algorithms and applications in order to apply it on my ungraduate scientific project “Unraveling fluid-fluid interfaces through molecular simulations and machine learning” supervised by Caetano Rodrigues Miranda and Alexsandro Kirch.

INTRODUCTION

The paper published by Lisa Jobs and Enrich A. Muller is a good starting point for beginners in the Machine Learning area. It introduces about the ML uses, types, potential in the physical, chemistry science and engineering fields and last, it brings up a hands-on ML exercise which was initially made in MATLAB, but it was also possible to do it in Python.

According to the text, the uses of ML algorithms just became popular recently because of the availability of big amount of data (Big Data) and the exponential growth of computational power. These two main facts led to applications such as the extraction of intermolecular force-field from quantum mechanical calculations, the use of AI in process system modelling and the design and development of catalyst are but some examples of upcoming trends. Then the paper writes about the two types of ML algorithms: Classification and Regression problems. Classification problems are about identifying in which category each data of our dataset fits and Regression problems states in finding an analytical solution which relates several parameters amongst themselves. By introducing briefly about the uses and types of ML, it shows the potential of ML in science by writing about the inherent difficulties (and sometimes an impossibility) of performing experiments for fluids in high ranges of temperature, pressure and volume conditions. In these situations, we have ML and equation of states as an alternative to substitute the experimental process to acquire information in such tough conditions (The paper "Generating a Machine-learned Equation of State for Fluid Properties" written by Kezheng Zhu and Erich A. Muller can give more details about it).

To finish, the paper proposes a hands-on ML exercise which is about predicting the boiling point temperature (Tb) of fluids by giving a large dataset with 6000 points of experimental fluids data. The exercise is composed of 3 different types of problems: 1)Linear regression problem - Predicts Tb by giving molecular weigh (mw) as input; 2)Multivariate Regression problem - Predicts the ratio between boiling point and critical temperature (Tb/Tc) by giving mw and accentric factor (ac) as inputs; 3)Artificial Networks problem - Predicts (Tb/Tc) by giving mw and ac as inputs. Explaining about why in the second and third part we are going to use the ratio between boiling point and critical temperature (Tb/Tc) is because as we have added one more feature to our model, more numerical process will be required and by doing that step our values will be between 0 and 1 and this regularizes the data, making easier to deal with numerical operations. Another advantage is related to the solid theoretical basis in Statistical Mechanics and recognizes that there is a relationship in the general physical behavior of all pure fluid, if an appropriate scale is used, their behavior can be mapped into a single ‘ universal’ conformal trend (The paper "Generating a Machine-learned Equation of State for Fluid Properties" written by Kezheng Zhu and Erich A. Muller has examples about it).

METHOD

Initially the exercise proposed by the paper was made in MATLAB, but as Python is the most preferred programming language in ML, in this work the results were reproduced using Python. The code in python can be found here.

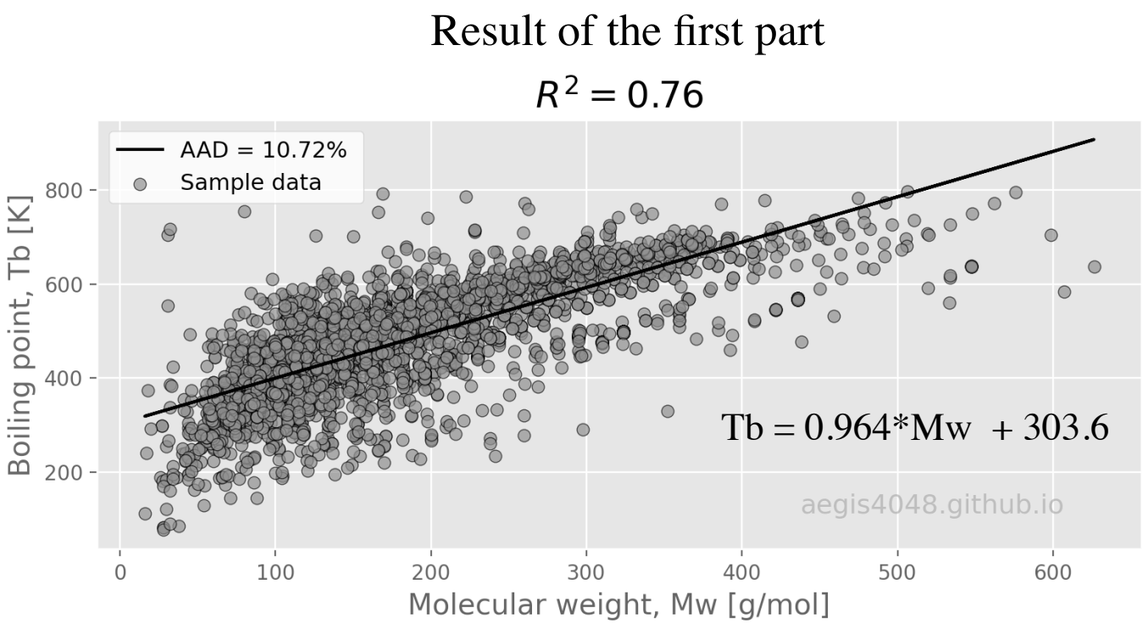
Description of the reproduction setting / implementation	
Module used	What was used for
Matplotlib	Plot 2D graph for the linear regression and Artificial Neural Networks problems.
Numpy	Round values, create arrays and matrixs to build the linear and multiple regression models.
Mpl_toolkits.mplot3d	Plot 3D graph for the multiple regression problem.
Scipy	Solve the linear regression and multiple regression math systems problems to find the parameters for each model.
Sklearn	Caculates R2.
Keras	Build the Artificial Neural Networks model.

RESULTS

During organic chemistry classes, we learned that the boiling of point of a substance depends on the molecular weight (mw) of the chemical compound: the heavier the compound the higher boiling temperature it will have and vice-versa. But what about methane (16g/mol) and water (18g/mol), they have almost the same molecular weight so is expected both have a similar boiling temperatures? Experiments shows that in this case, although they are roughly comparable molecular weight, methane’s boiling temperature (111.6K) is lower than water’s (373.15K) and that is because they have different interactions between molecules.

In these exercises, we are going to first confirm the lesson learned from those classes by obtaining a linear equation to predict the boiling temperature depending on the molecular weight of the fluids and in the last two exercise we are going add the accentric factor which is a dimensionless number that helps to quantify the relative strength of the intermolecular interactions in order to improve our model and finally verify if the experimental observations can be seen in our last two ML models.

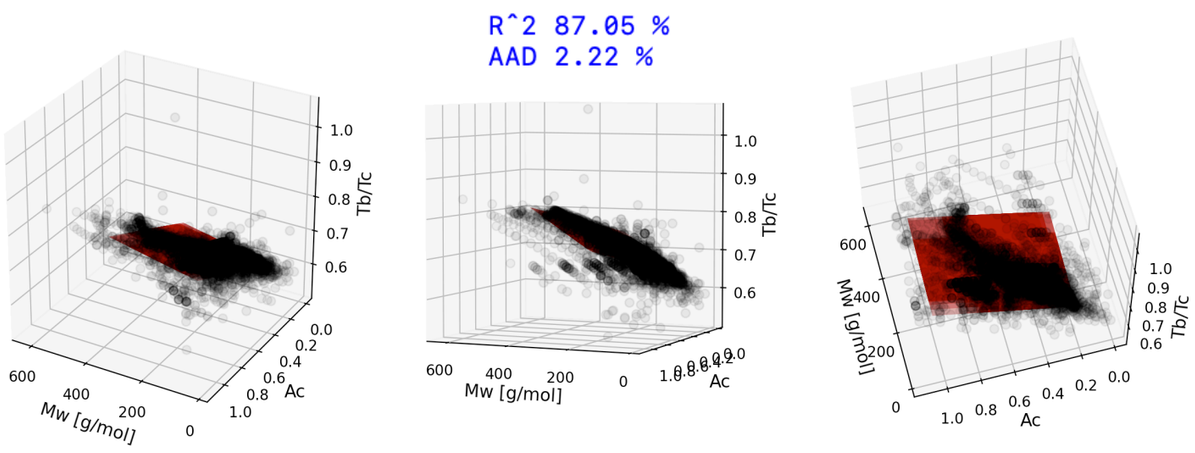
1) Linear regression problem - Predicts Tb by giving molecular weigh (mw) as input.



The result of the first part is represented in the figure on the left and it shows that the straight line obtained from the linear regression does confirm what we have learned in our organic chemistry lessons and also we have obtained a good correlation coefficient $R^2 = 0.76$ which means molecular weight and boiling point are strongly correlated. Another observation to point out is about the average absolute deviation value (AAD = 10.72%) which can be considered also a good result because this regression model was obtained from a large dataset with 6000 compound and by looking at the results only 10.72% (approximately 643 compounds) is far from the line. A more in detailed analysis would indicate that the prediction of boiling point temperature will be better for molecular weight value in the range of 50 to 400 which is the part with more data points concentration.

2) Multivariate Regression problem - Predicts the ratio between boiling point and critical temperature (Tb/Tc) by giving mw and accentric factor (ac) as inputs.

Result of the second part

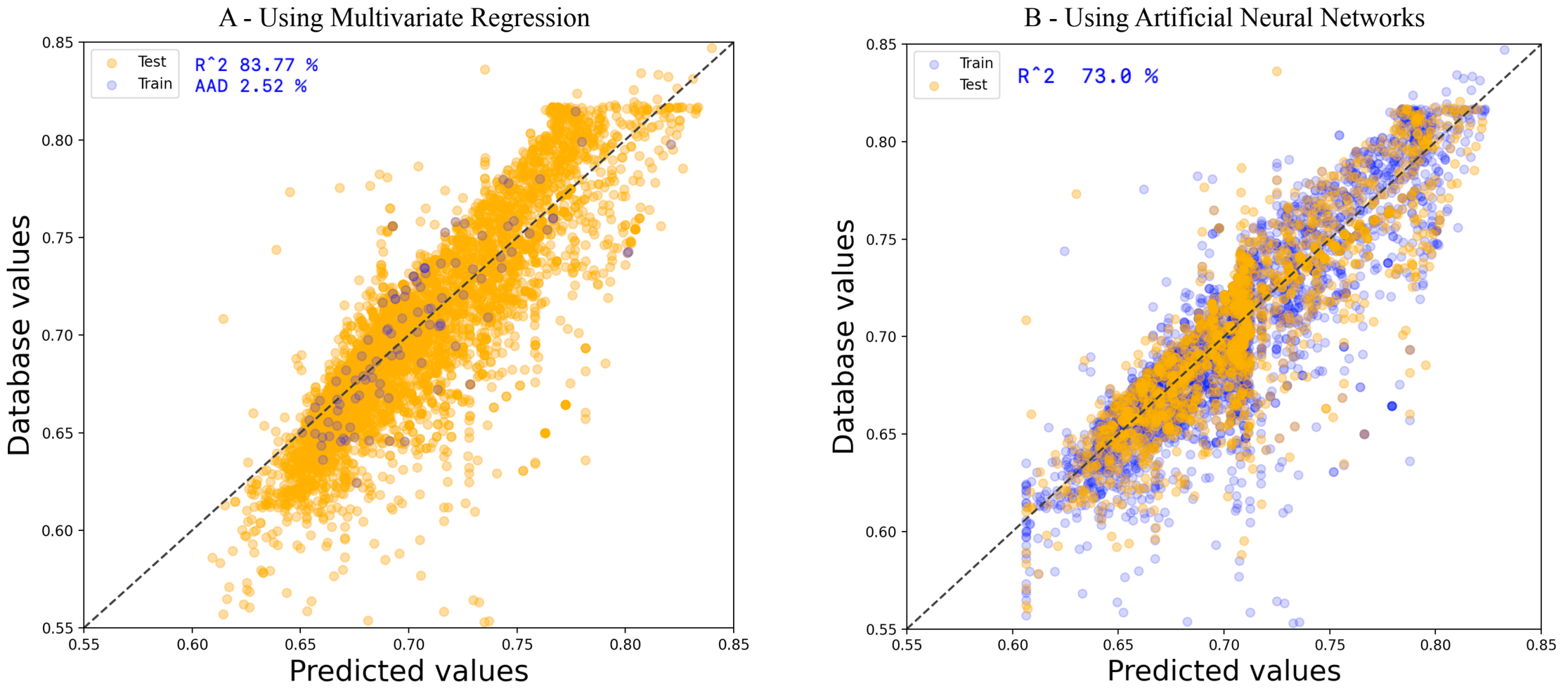


$$Tb/Tc = 0.6 + 1.7x10^{-4}\ Mw + 1.7x10^{-1}\ Ac$$

In the case for the second part, we only used 100 random data from our dataset to obtain the parameters of the plane because, according to the paper, using all the data to train the model it didn't contribute to obtain higher value of correlation coefficient and using fewer number of data besides the fact that will require less computational power and then speeding up the process, also we can got a good value of $R^2 = 87\%$. Another analysis to mention is that by adding one more feature to our model based on the theories insight that we have previously discussed, even with many outliers which can be seen in our graphics, it still improved our model in the $R^2 = 87\%$ and the AAD = 2.22% values by comparing the results with the linear regression model.

3) Artificial Neural Networks problem - Predicts (Tb/Tc) by giving mw and ac as inputs.

Results of the third part



For the last part, it was expected to obtain a better correlation coefficient by using artificial neural networks model, because this model is well suited for processing very large data sets where there is very little understanding of the correlations between them. This may caused by the model of artificial neural network used wasn't the same of the paper so the results obtained were different, but if you analyzed it without comparing with other models we also got good results with $R^2 = 73\%$.

4) Artificial Neural Networks Model Details

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense (Dense)                (None, 4)                 12
_________________________________________________________________
dense_1 (Dense)              (None, 8)                 40
_________________________________________________________________
dense_2 (Dense)              (None, 16)                144
_________________________________________________________________
dense_3 (Dense)              (None, 1)                 17
=================================================================
Total params: 213
Trainable params: 213
Non-trainable params: 0
_________________________________________________________________
Start training for 200 epochs ...
```

```
 History head:       loss  mean_squared_error  val_loss  val_mean_squared_error  epoch
0  0.846992          0.846992  0.068550              0.068550      0
1  0.065159          0.065159  0.054846              0.054846      1
2  0.048528          0.048528  0.033835              0.033835      2
3  0.033231          0.033231  0.023958              0.023958      3
4  0.021592          0.021592  0.015518              0.015518      4

 History tail:        loss  mean_squared_error  val_loss  val_mean_squared_error  epoch
195  0.000642          0.000642  0.000874              0.000874    195
196  0.000652          0.000652  0.000862              0.000862    196
197  0.000641          0.000641  0.000878              0.000878    197
198  0.000658          0.000658  0.000908              0.000908    198
199  0.000655          0.000655  0.000857              0.000857    199

 Train RMSE: 0.025934128803426677

 Test RMSE: 0.02390493204055573
```

Taking a look into our artificial neural network model details, it has 213 parameters with 200 epochs and is observable that our model is improving for each epoch. It started with high values of loss and mean squared erros and in the last step ended up with small values. Also we can see that the root mean squared error is small and almost the same for the training and test datasets which means that our model is not over fitted or under fitted.

CONCLUSIONS

The results reproduction of the first and second parts were not difficult, but the third problem it took me more time to solve it, also I asked help to my supervisor and even that I didn't get the same results. In fact, not getting the same results it doesn't means that my results were wrong, it also allows to get new insight and learn better about it, because once we get wrong something, our attention just increases leading us to go after more knowledge and study more about it. An interesting fact to notice is that comparing the 3 ML techniques, the multivariate regression model used less data and the linear regression model which doesn't require to much numerical operations, both got a better R^2 value than the artificial neural network which used 70% of the data to train and 30% to test and also requires big computational power. It seems that for this case simpler model fits better than complex ones.

REFERENCES

[1] Machine Learning for Fluid Property Correlations: Classroom Examples with MATLAB" published by Lisa Jobs and Erich A. Muller in 2019 at American Chemical Society Publications.