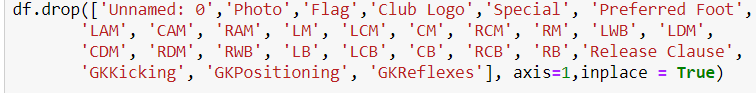
**FIFA project**

Basic classification and regression using Keras:

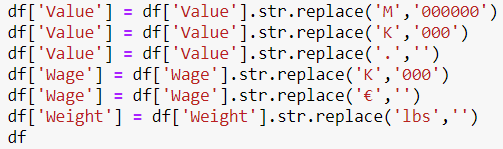
* Import and read the data set
* Drop unnecessary features (the positions we dropped are the player rating in each position. We decided to drop then and leave the player Overall rating).



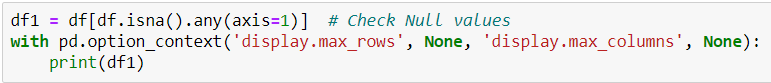
* Drop GK position (GK has different skills that set his value and wage).



* Change symbols and characters of DF



* Check and treatment null values



* For example, players without a club and players with 0 value





* Change values types



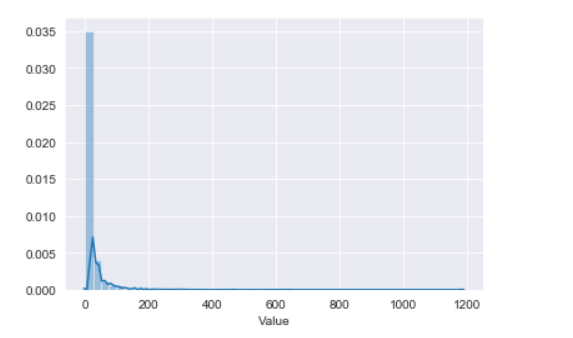
* Divide the value by million to decrease the values



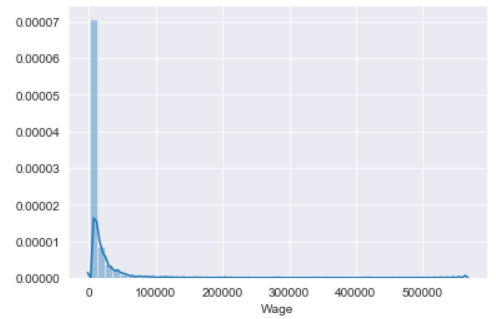
* Check if our features distribute normally – almost our DATA that involve skills distribute normally, our Y- the wage and the value doesn’t distribute normally.

Some of our features aren’t involve skills. For example, "International Reputation" is not a skill, the best players have the highest reputation (Like the Value and Wage).

* Value:

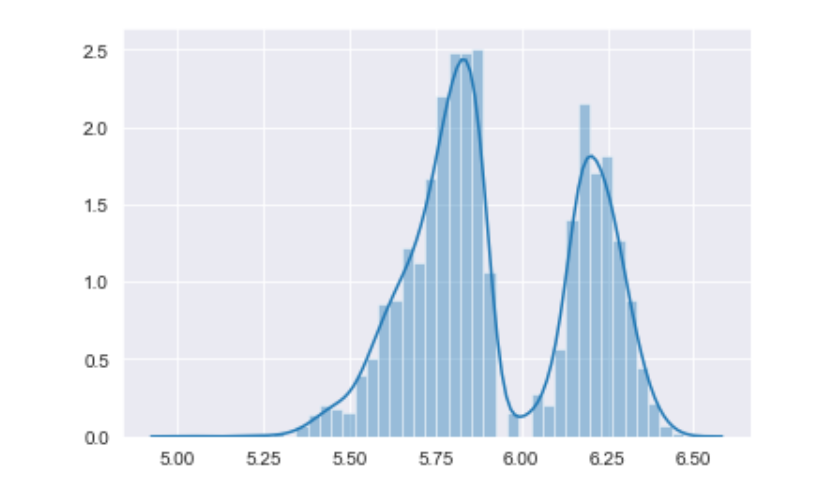


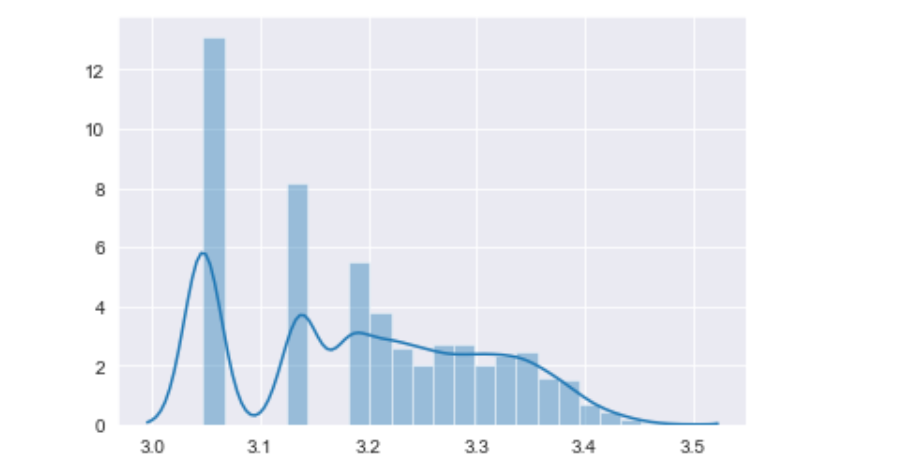
* Wage:



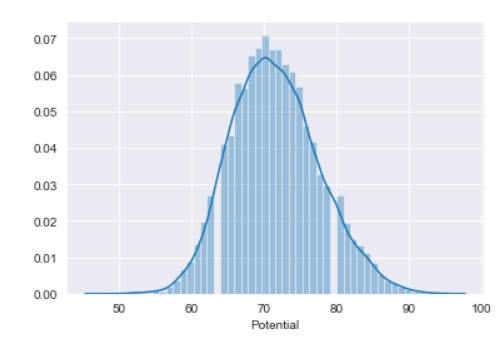
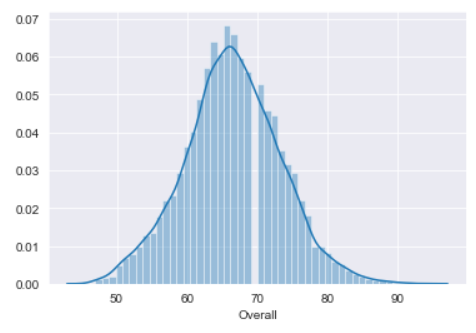
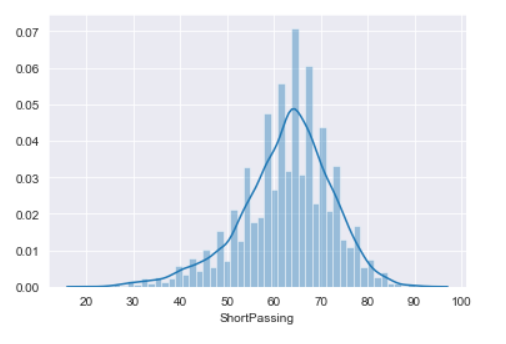
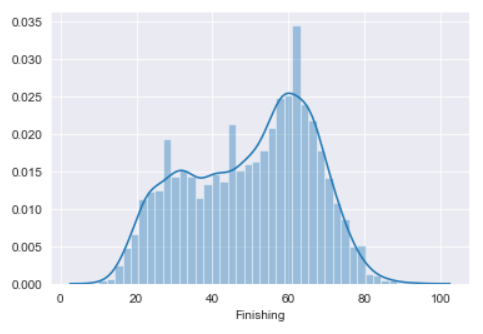
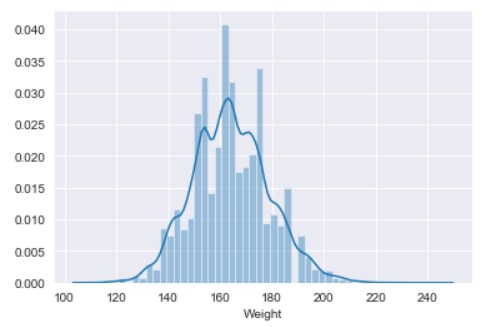
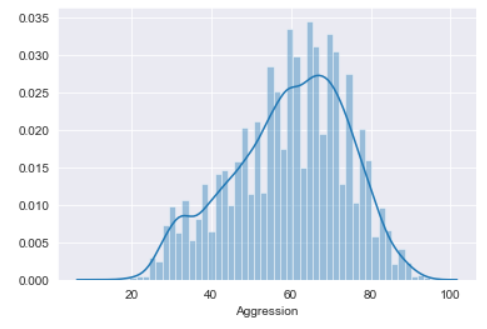
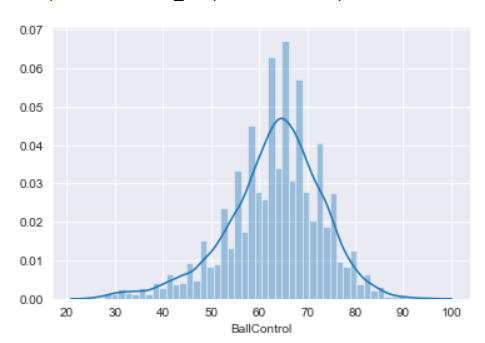
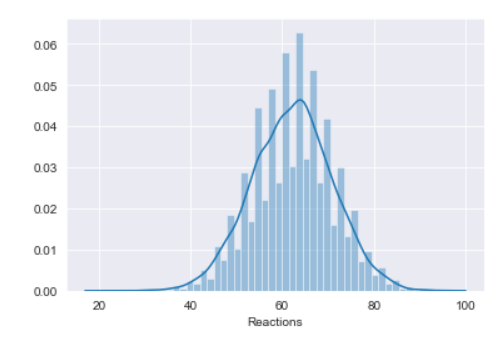
We tried to handle the skewed data, we found this website that helped us - [Top 3 Methods for Handling Skewed Data | by Dario Radečić | Towards Data Science](https://towardsdatascience.com/top-3-methods-for-handling-skewed-data-1334e0debf45), [Box Cox Transformation - Statistics How To](https://www.statisticshowto.com/box-cox-transformation/).

We applied box cox transformation to make the data behave like normalized data. That was the result – Value -

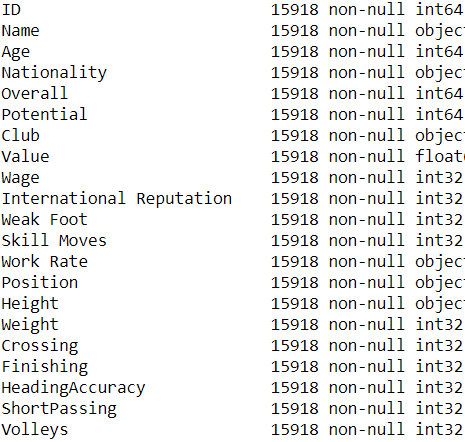


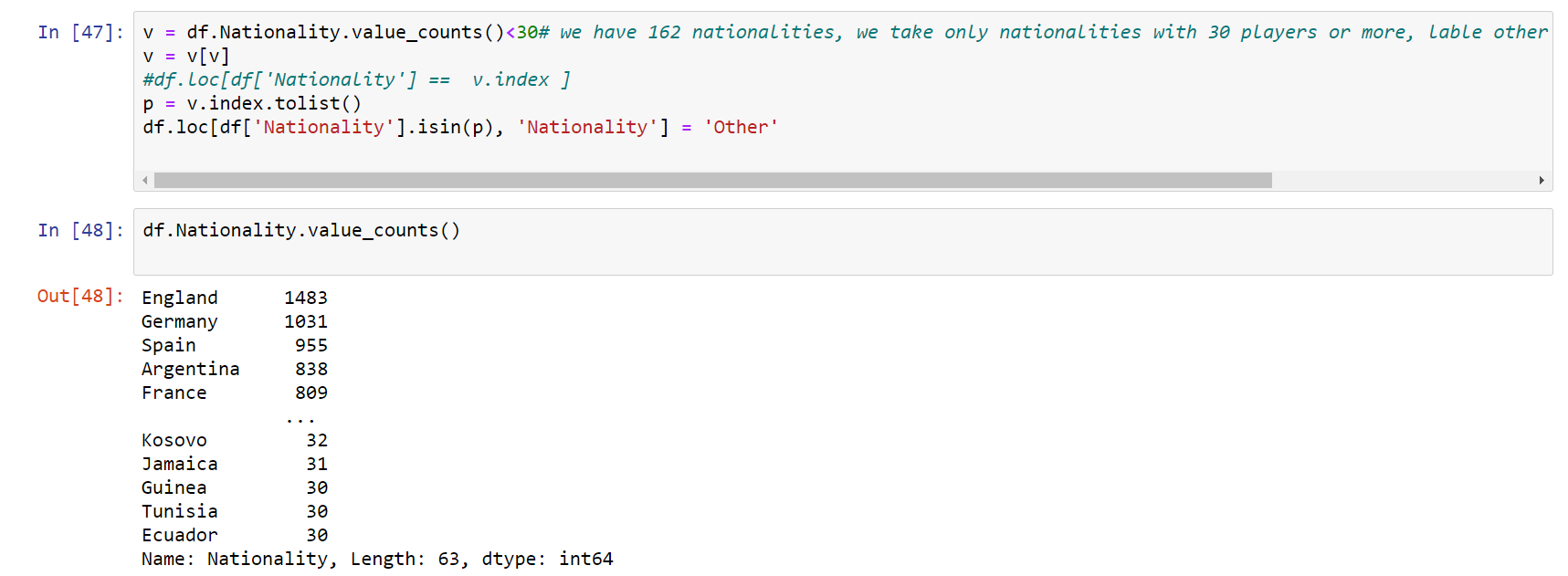
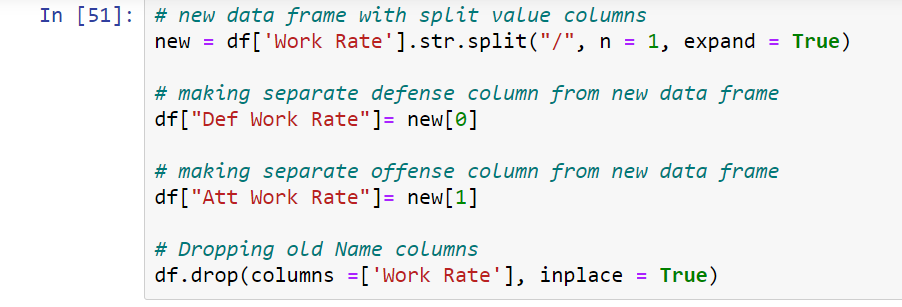
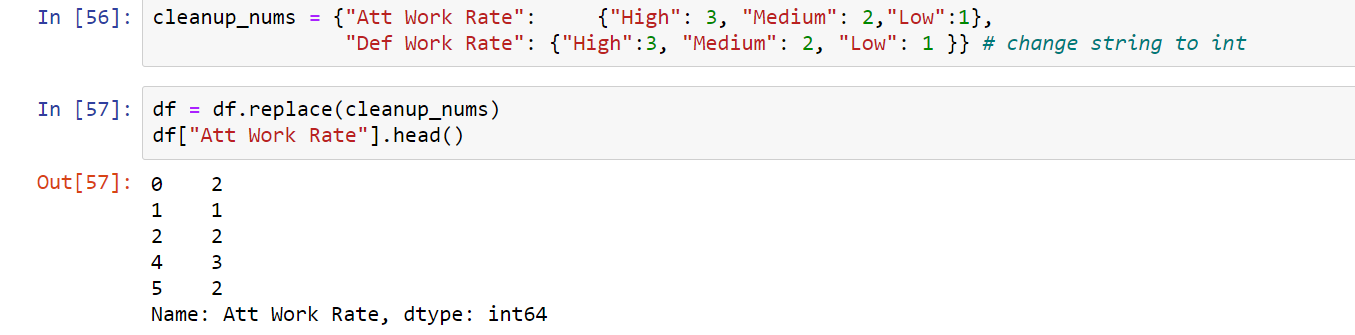
Wage - 

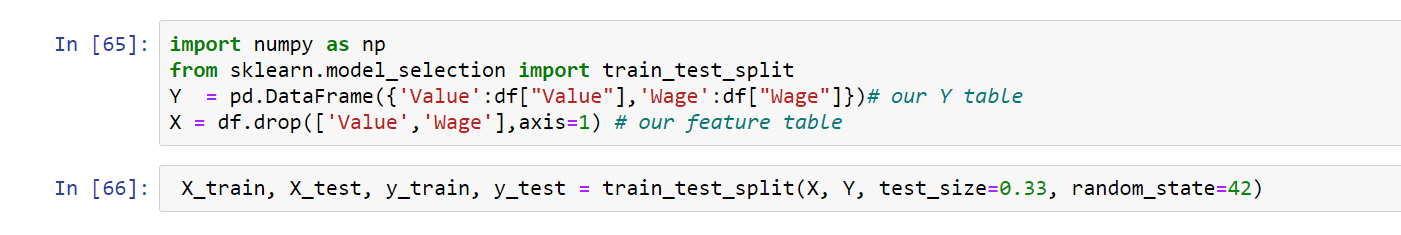
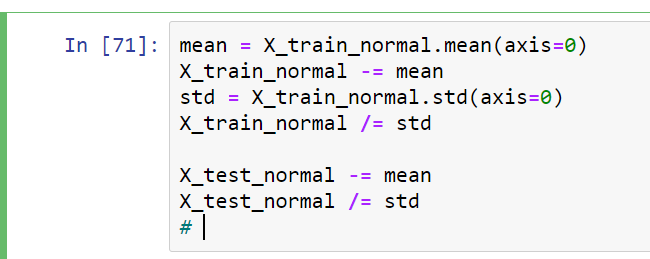
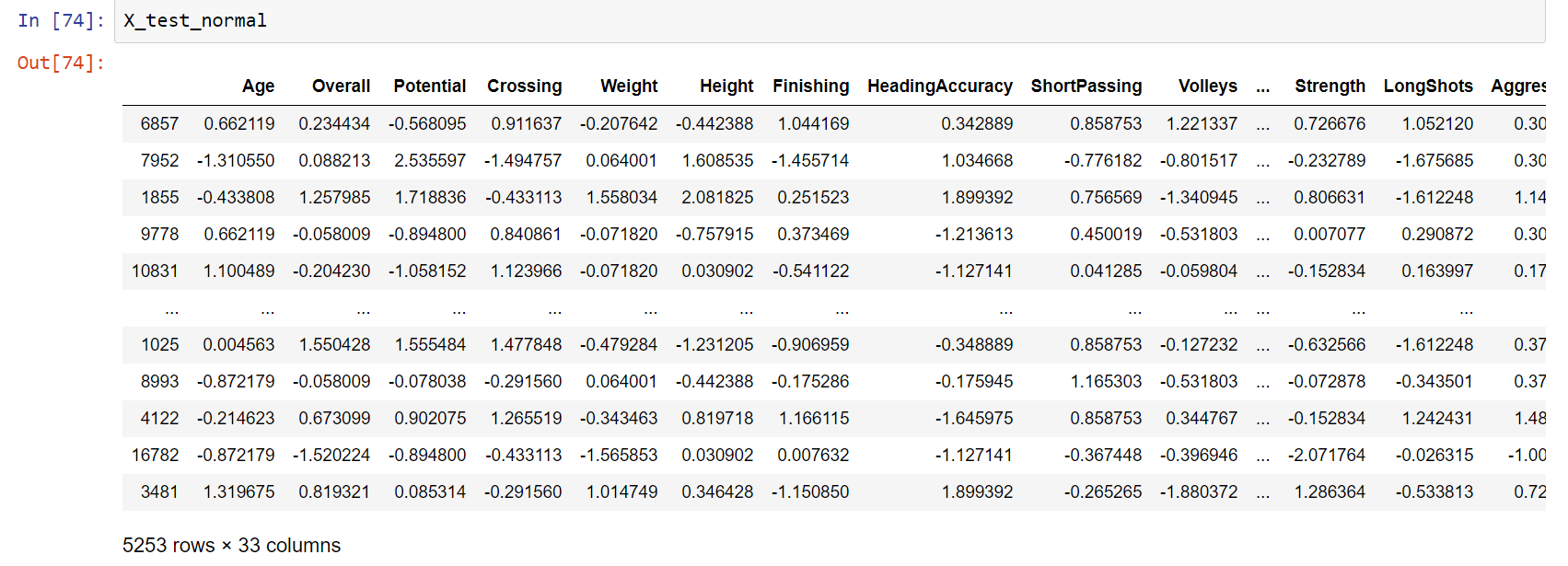
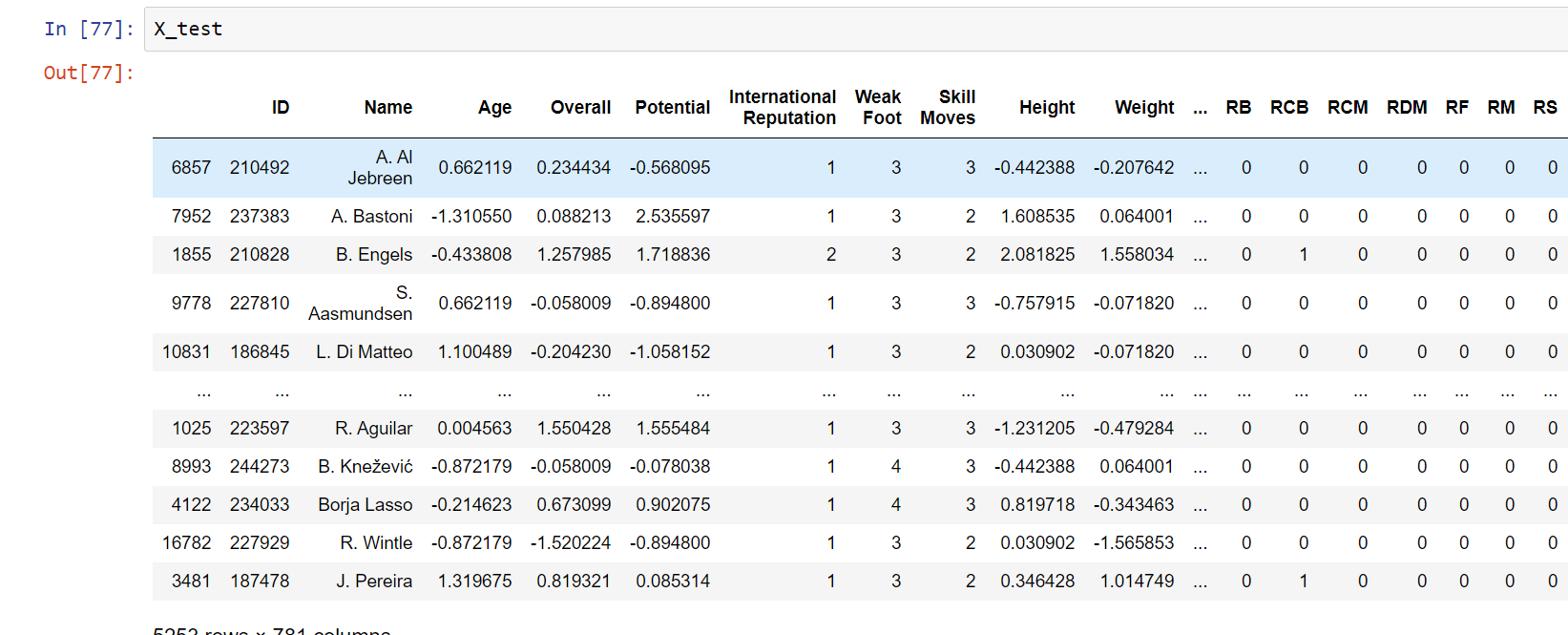
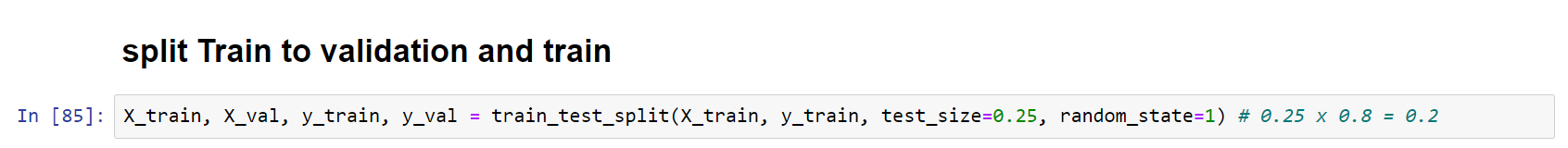
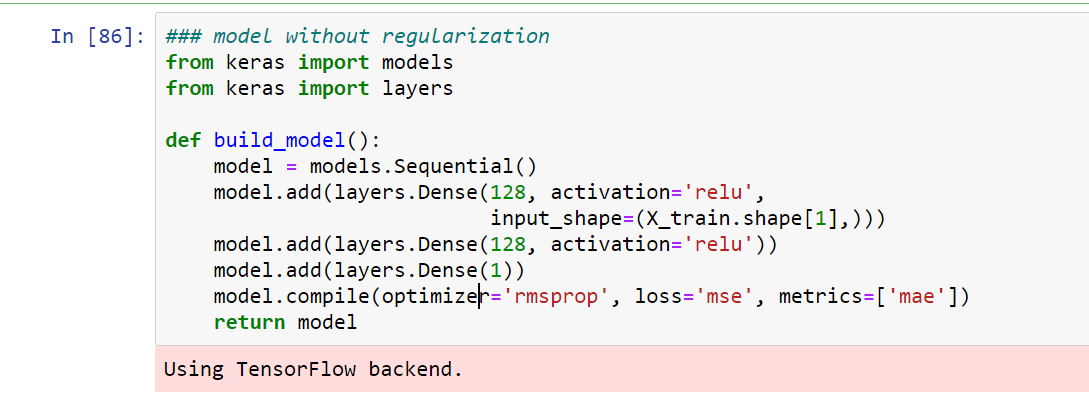
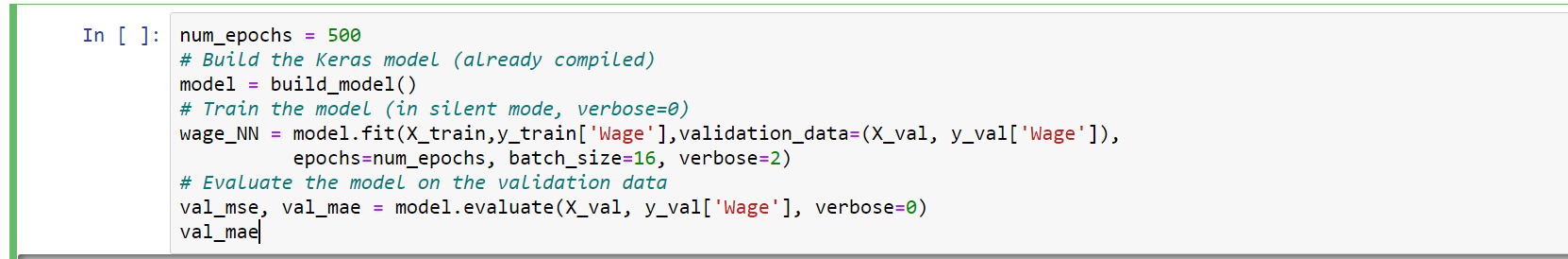
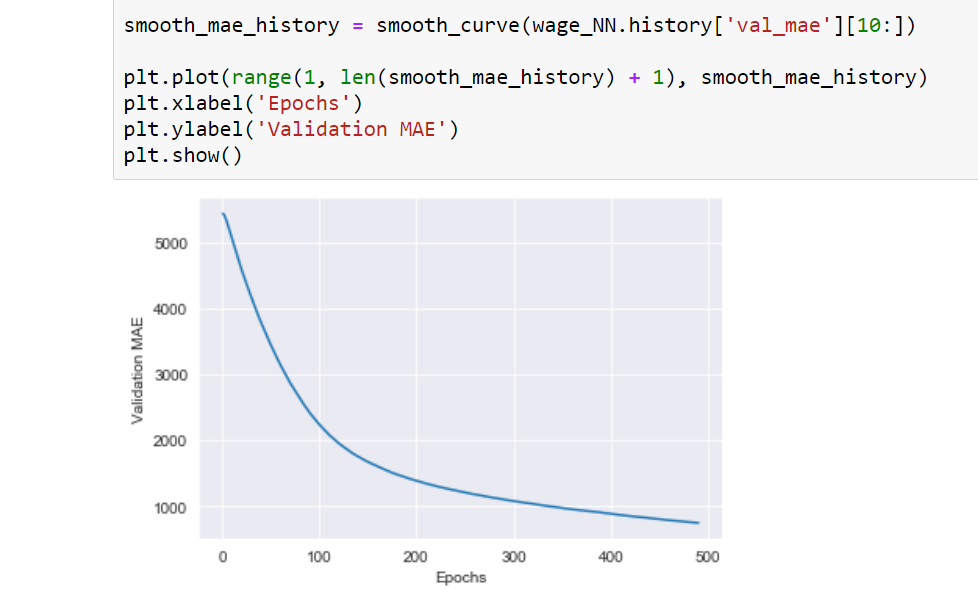
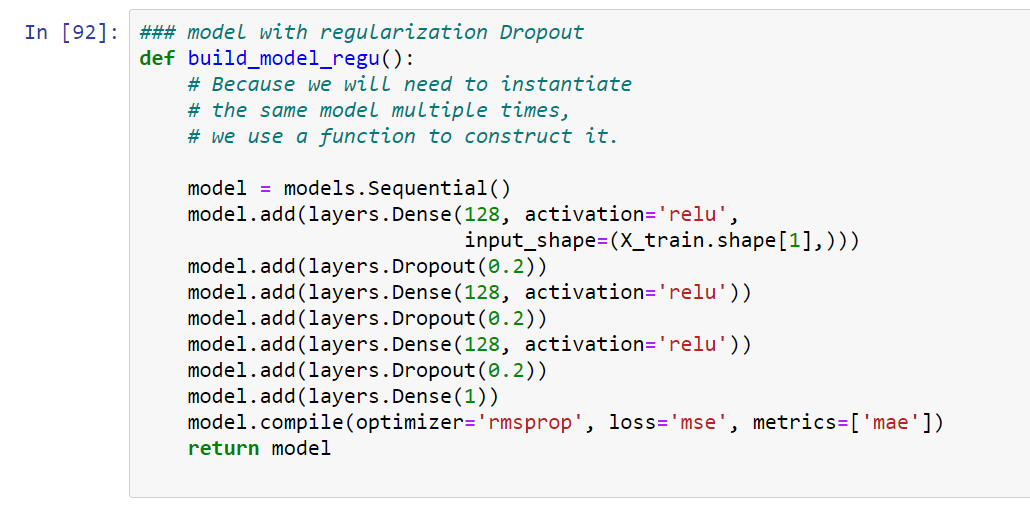
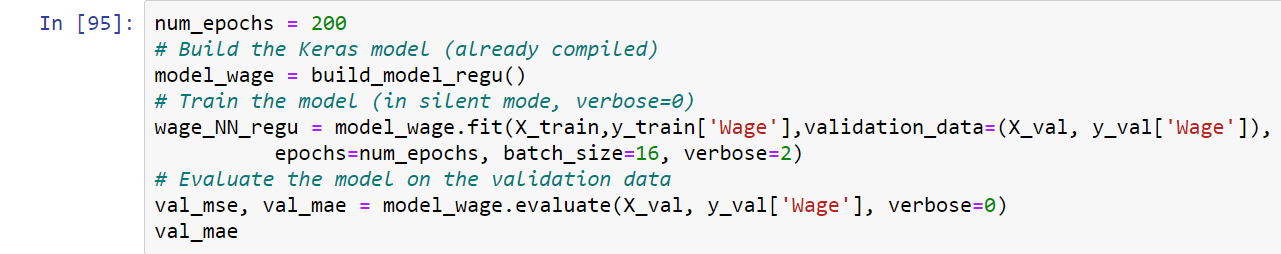
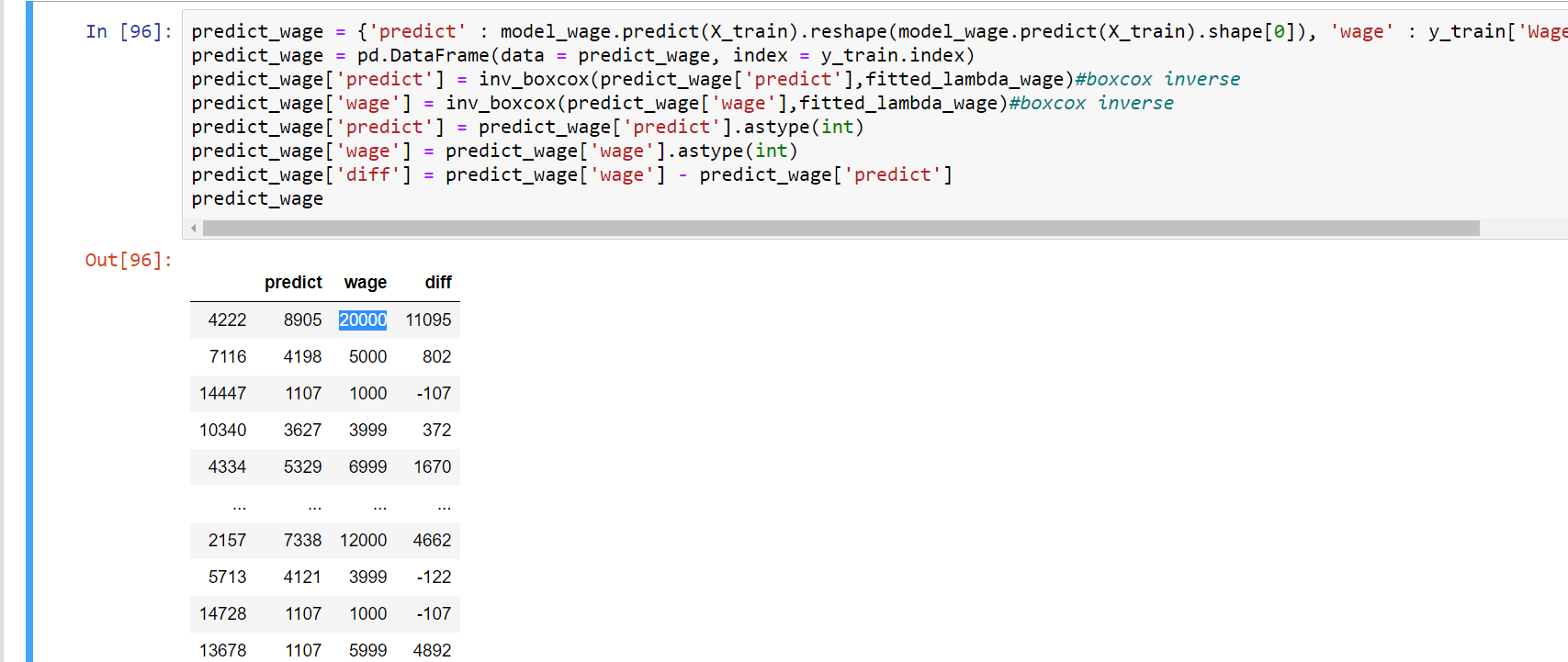
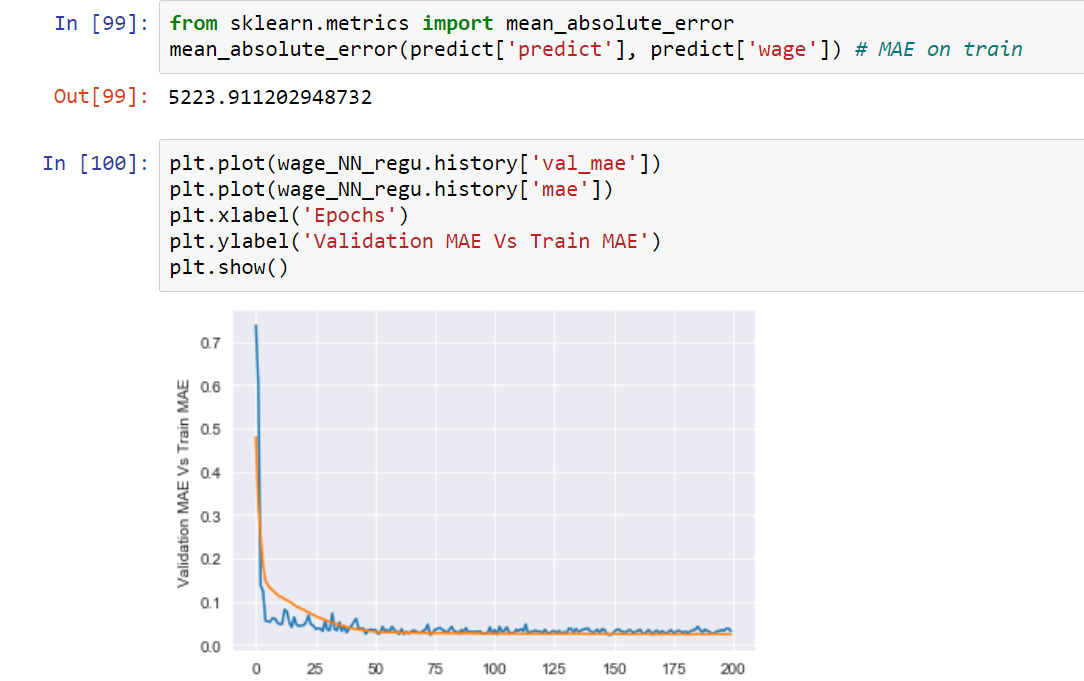
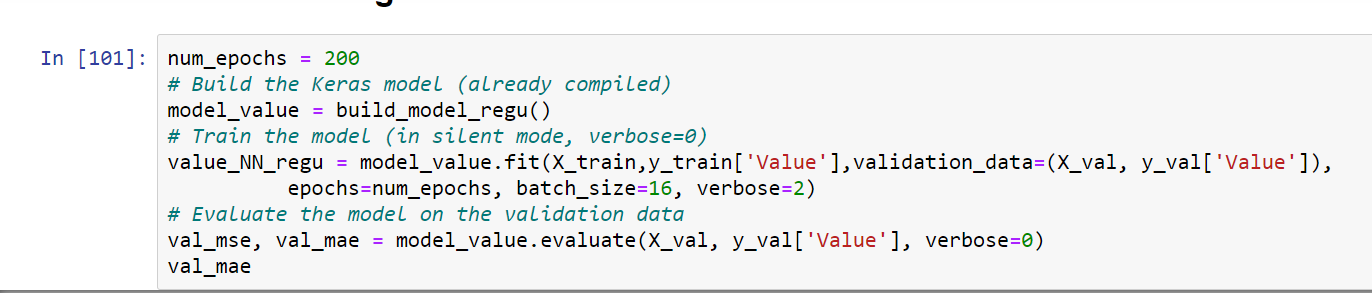
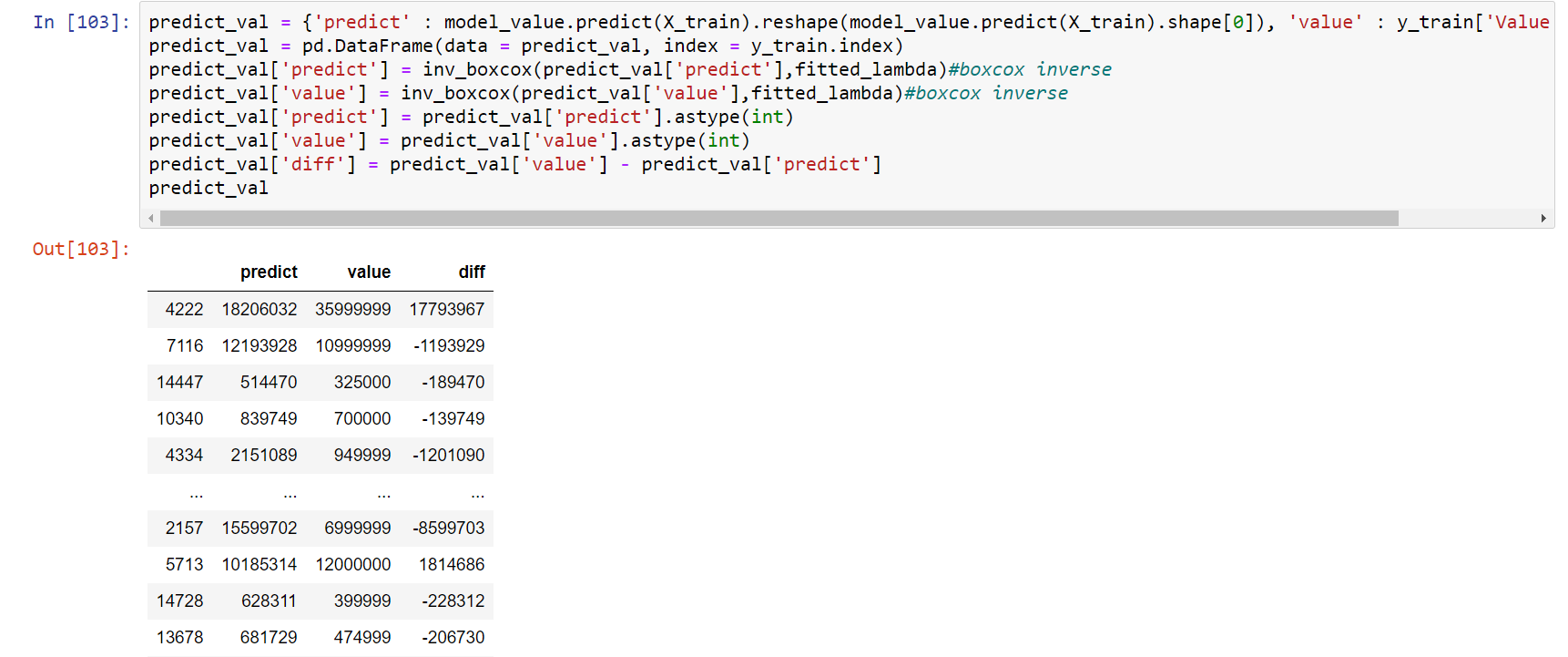
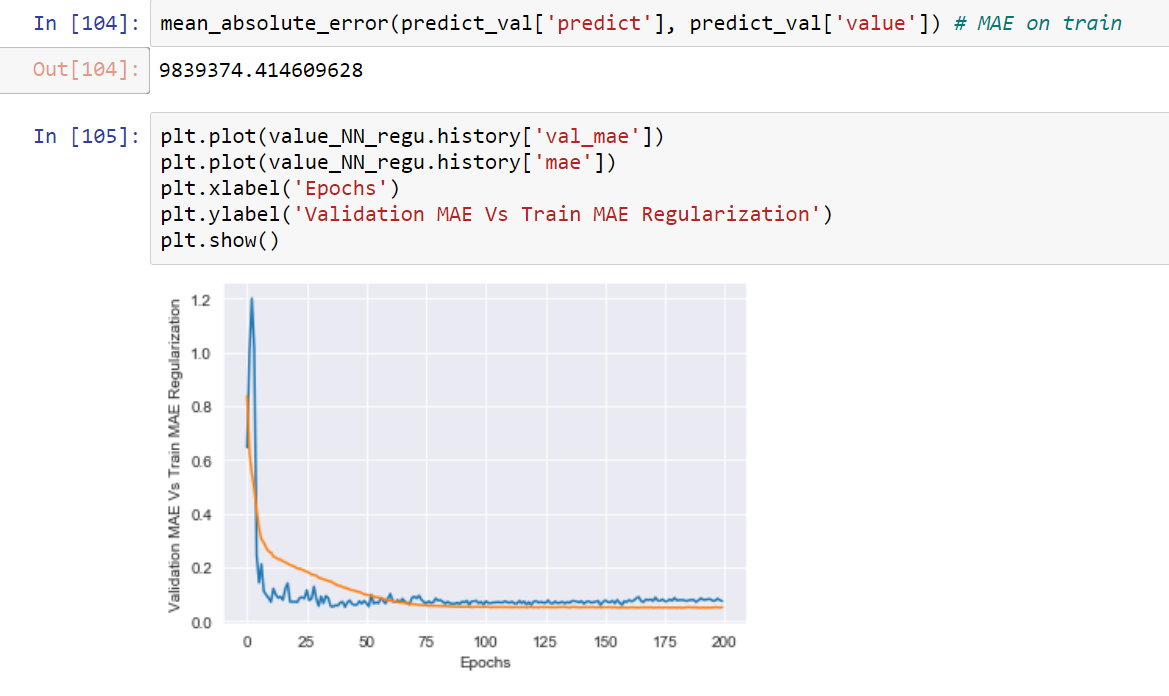
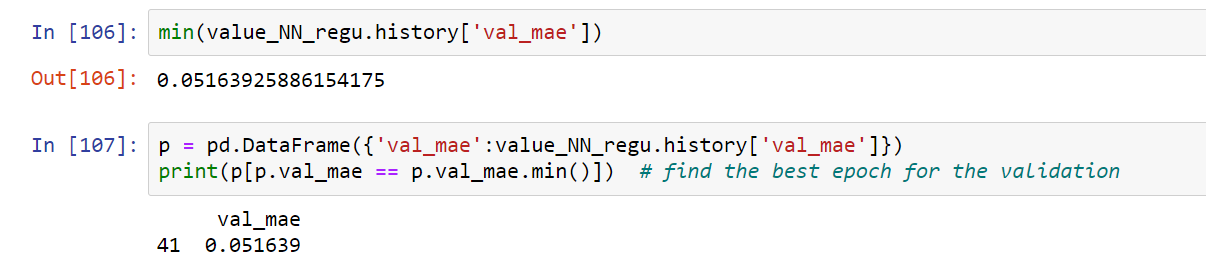
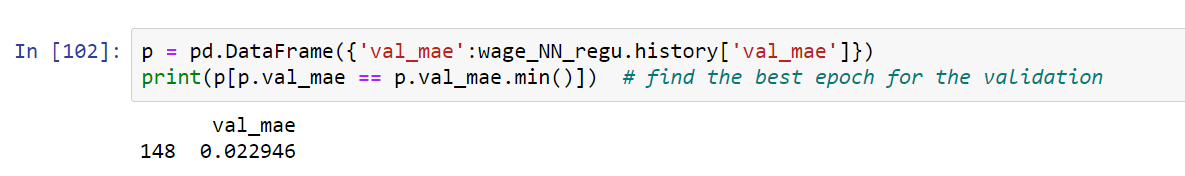
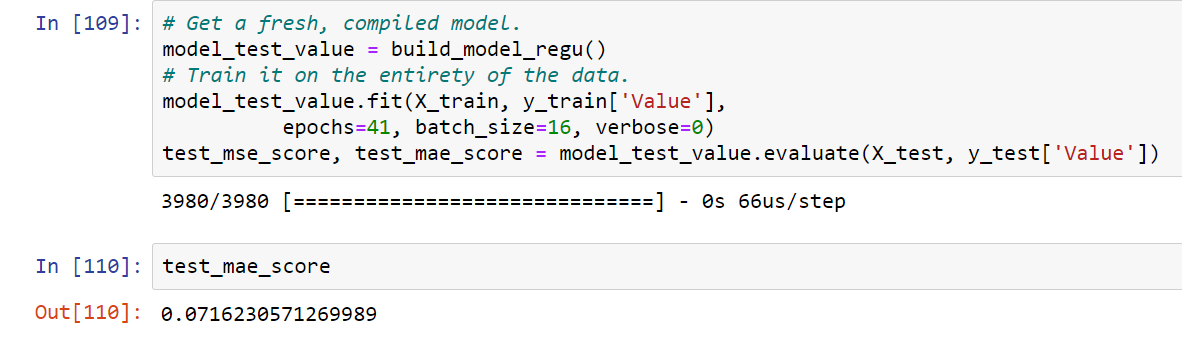
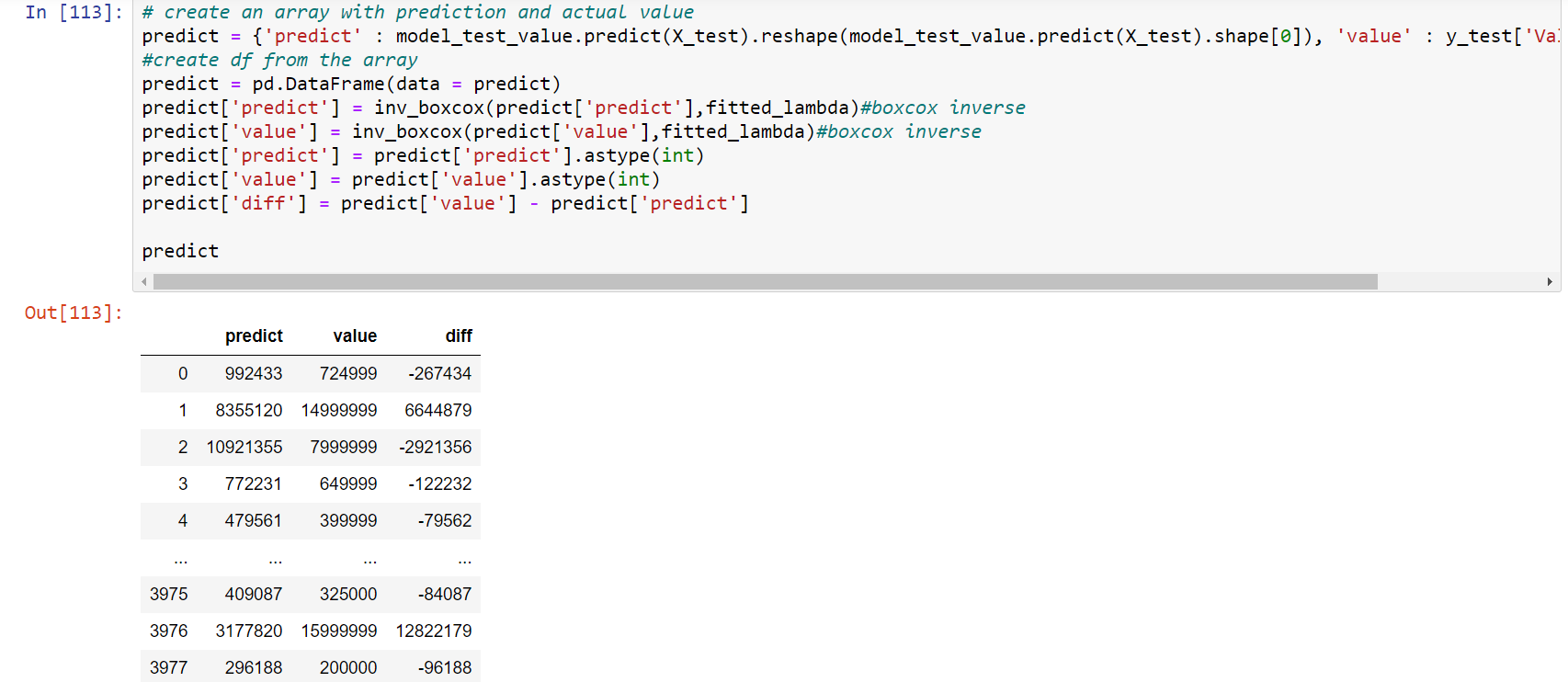
* More features:

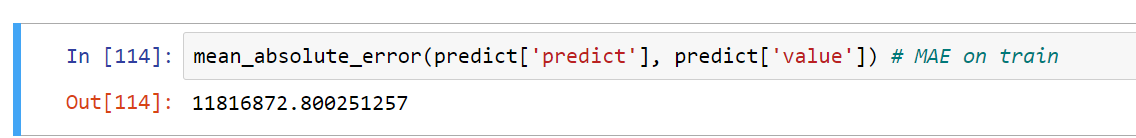
       

* Treatment object and categorical value:
* First, we check values types:

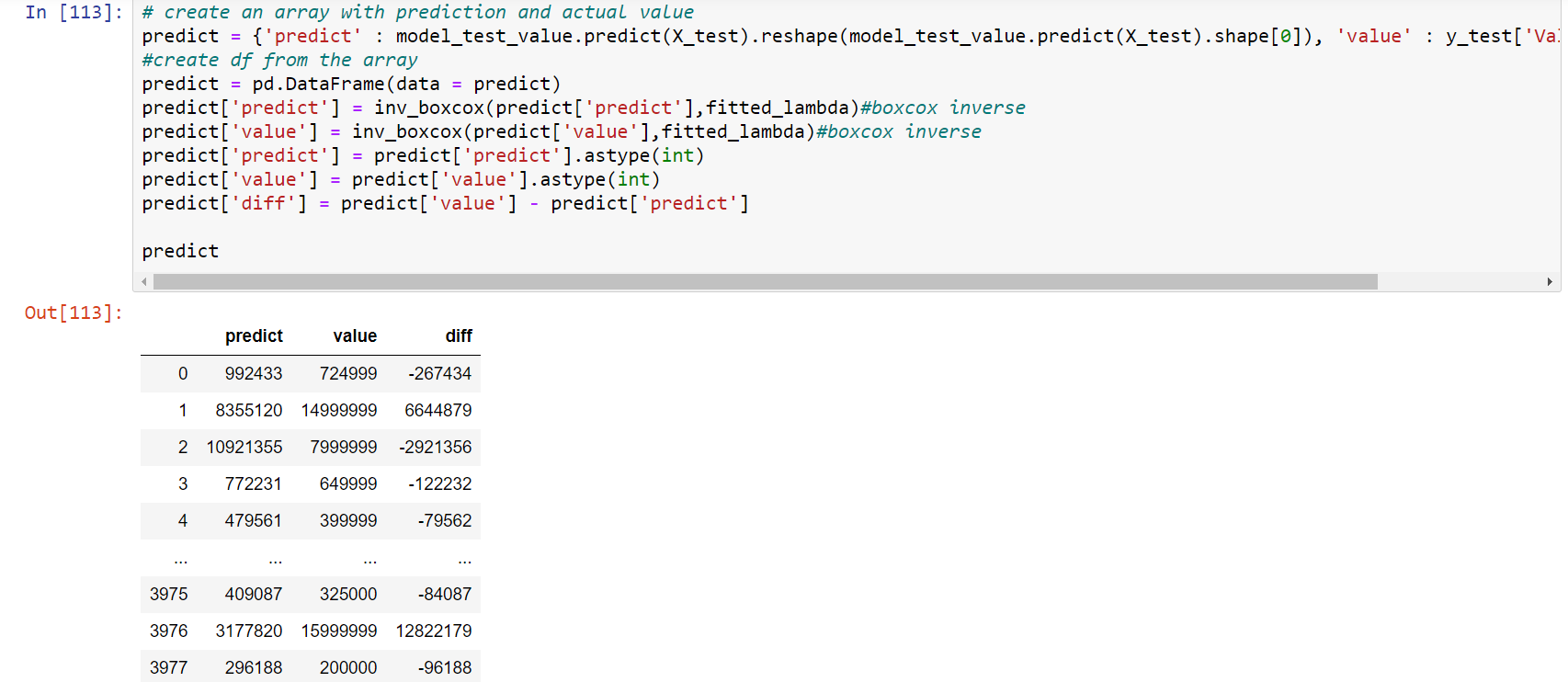


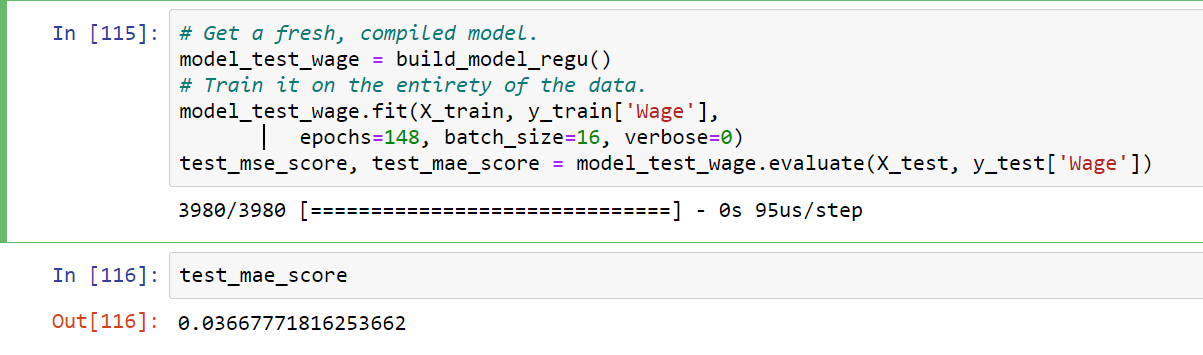
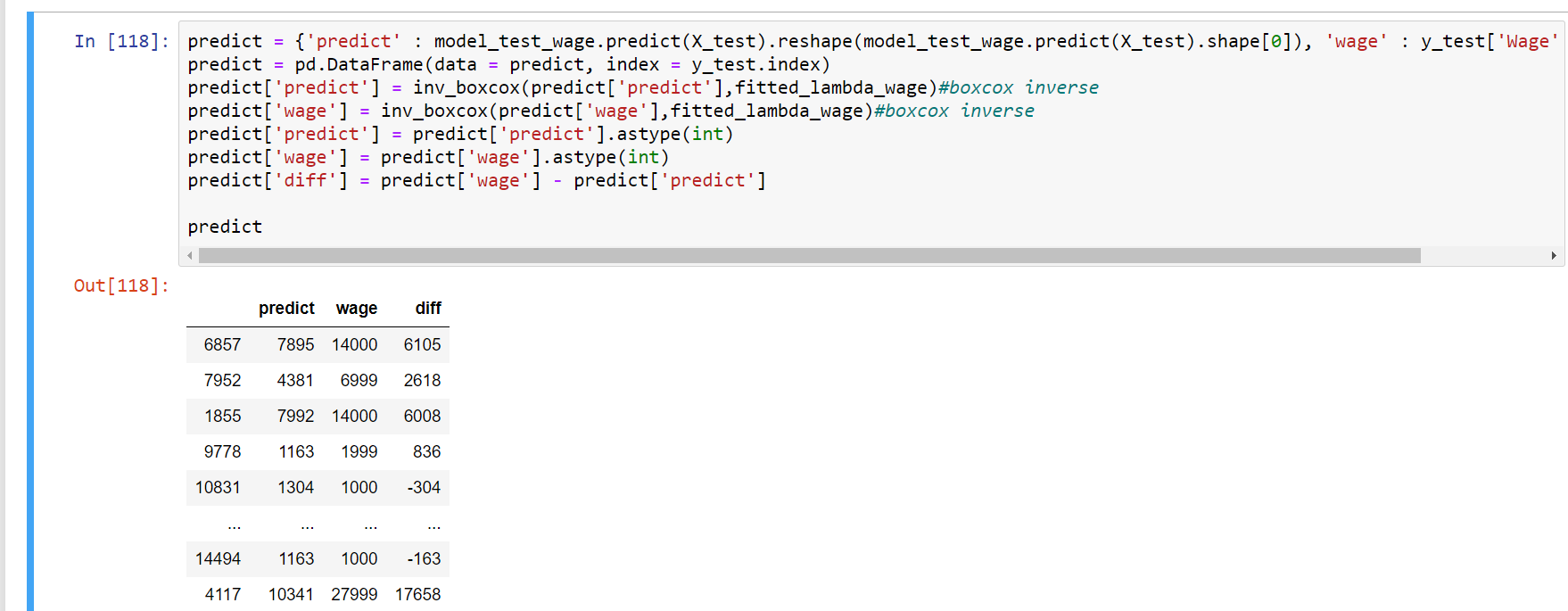
* Secondly, modifying relevant features:
* Nationality – we started with 162 nationalities. We took all the nationalities that had under 30 players in them. Then we made Nationality into dummy variable, and concatenate it into the table, then dropped the original Nationality values. 
* We created dummy variable for the clubs, and concatenate it into the table.
* We Split Work Rate category for attacking work rate and defensive work rate. It was all in one column – Work Rate. Also, we changed the values inside, from “High”, “Medium” ,”Low”, to 1,2,3 when 3 is “High”.  
* We did similar proces on the rest of our categorical values.

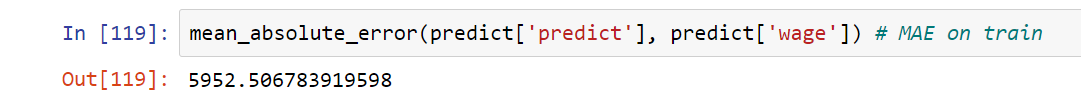
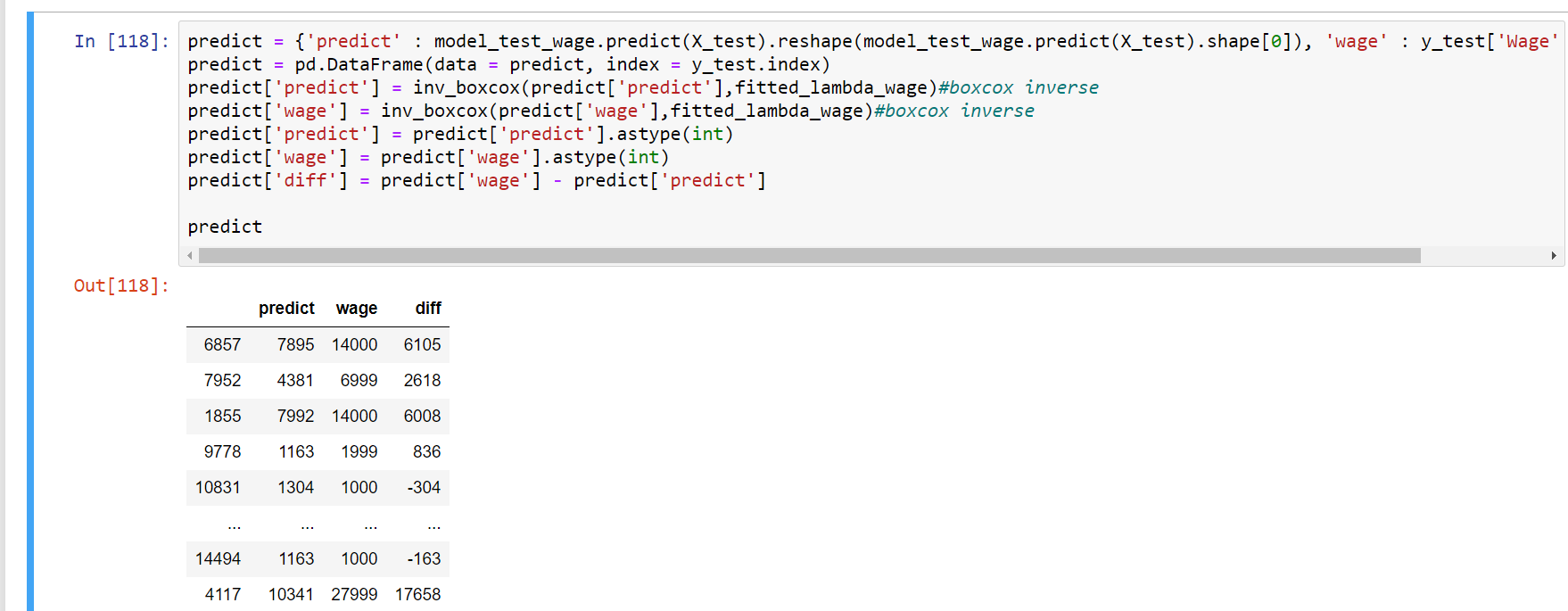
* Split the data to train and test   
  + 
* Normalization –
  + Normalizing the data – We transferred the continues features to different table, Normalized them, then copied them back to the X tables Normalized.  
      
      
      
    
* Split the Training data to train and validation -   
  
* Create a NN model without regularization -   
  
  + The model has 128 Neurons.
  + The model use relu as activation function.
  + Run the model on the train and validation Wage.
  + Evaluate the model - 
* Create a NN model with regularization
  + 
  + Run the model on the train and validation Wage. 
  + We reversed the data to try understanding how the model works - 
  + Evaluate the model – 
  + We did the same for the value - 
  +  
* Run the model on the test –
  + Find the minimum validation value and the epoch 
  + Run the model on the test - 
  + Create new data frame with the inverse value to evaluete the model - 



We can see that the model predicts well on the lower values, but on the high values, we can have big difference between the prediction and the real value.



* Run the model on the test - wage 
* On the wage, we had smaller MAE value, again, we had small difference when the wage was lower, and bigger ones when the wage was bigger.



Conclusions –

Our model predicts well for the lower values and has a big difference for the high ones. Even after we transformed the data into data that distribute close to normal distribution, we couldn’t decrease our MAE score to a sufficient level. Our conclusion is that we apparently have tried to split the data into two groups, one with high values and the other with lower values. If we had separate networks for each group, we might be able to predict better on the high values.