**Regression problem on**

**Online news popularity**

**Data introduction**

This dataset was provided by UCI machine learning repository, original acquired by Mashable in the period of 2 years. The goal of this dataset was to predict number of shares in social network after their article was published.

In this dataset there are total of 39,797 records with 61 attributes. These attributes contain 58 predictive attributes, 2 non-predictive attributes and 1 goal field (shares: Number of shares (target)). Dataset does not contain any missing values. Supervise learning technique is used to approach this problem, since target I given in this dataset.

Further details of each attributes reference can be found in attached [*OnlineNewsPopularity.names*](OnlineNewsPopularity.names) document.

**Data cleaning**

On preliminary inspection, retraining only predictive attributes, dataset will still give a relatively large attributes space. Through a linear regression as baseline, prediction produce undesired result. Which is cost by substantial number of attributes in dataset. To improve model ability to generalize, dataset number of attributes need to be reduce. The followings are removed as they are irrelevant on contributing to train neural network model.

Reference material on why these attributes are removed can be found in[*online news popularity.html*](ref%20materials/online%20news%20popularity.html)file.

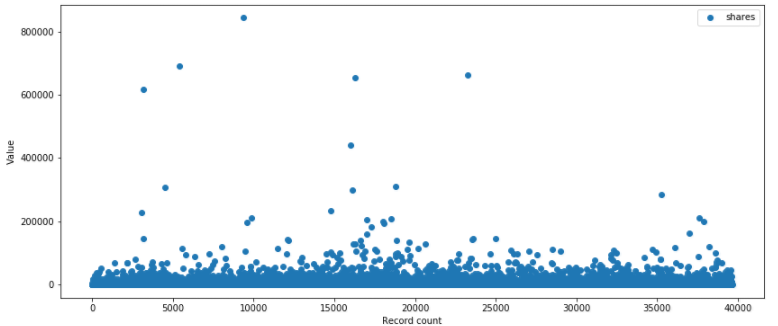
Following variables are omitted:

|  |  |
| --- | --- |
| **Attributes** | **Remarks** |
| **url** | URL of the article. (non-predictive) |
| **timedelta** | Days between the article publication and the dataset acquisition. (non-predictive) |
| **LDA\_01**  **LDA\_02**  **LDA\_03**  **LDA\_04**  **LDA\_05** | Latent Dirichlet allocation variables, derive field from URL on topics. |
| **is\_weekend** | Since it seems to be duplicating days of week. |
| **kw\_min\_min**  **kw\_avg\_min**  **kw\_min\_avg** | Contain mainly of zero or negative values, incomplete dataset. |
| **Weekday\_is\_monday**  **weekday\_is\_tuesday**  **weekday\_is\_wednesday**  **weekday\_is\_thursday**  **weekday\_is\_friday**  **weekday\_is\_saturday**  **weekday\_is\_sunday** | Binary fields, will be converted to categorical variable to reduce the numbers of attributes in dataset. |
| **data\_channel\_is\_lifestyle**  **data\_channel\_is\_entertainment**  **data\_channel\_is\_bus**  **data\_channel\_is\_socmed**  **data\_channel\_is\_tech**  **data\_channel\_is\_world** | Binary fields, will be converted to categorical variable to reduce the numbers of attributes in dataset. |

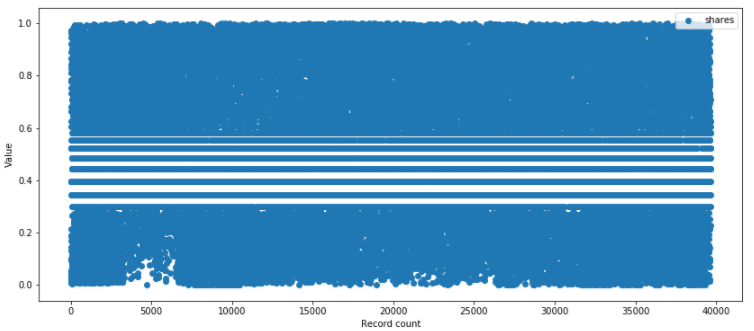
After omission dataset left 38 predictive attributes and 1 goal field. Following shows the remaining attributes that will be used for training.

|  |  |
| --- | --- |
| **Retain attributes** | |
| n\_tokens\_title  n\_tokens\_content  n\_unique\_tokens  n\_non\_stop\_words  n\_non\_stop\_unique\_tokens  num\_hrefs  num\_self\_hrefs  num\_imgs  average\_token\_length  num\_keywords  kw\_max\_min  kw\_min\_max  kw\_max\_max  kw\_avg\_max  kw\_max\_avg  kw\_avg\_avg  self\_reference\_min\_shares  self\_reference\_max\_shares  self\_reference\_avg\_sharess | global\_subjectivity  global\_sentiment\_polarity  global\_rate\_positive\_words  global\_rate\_negative\_words  rate\_positive\_words  rate\_negative\_words  avg\_positive\_polarity  min\_positive\_polarity  max\_positive\_polarity  avg\_negative\_polarity  min\_negative\_polarity  max\_negative\_polarity  title\_subjectivity  title\_sentiment\_polarity  abs\_title\_subjectivity  abs\_title\_sentiment\_polarity  weekday  data\_channel  Shares (Target) |

Objective of this project is to estimate the exact shares of a given article. When visualizing shares, there are many outliers in this field. These outliers are removed by using Quantile Transformer, after transformation resulting shares will fall within 0 and 1.



**Before normalization**



**After normalization**

Training dataset is split into 80% training set 20% test set and seeded shuttle is applied when generate of these datasets. Summary of training and testing dataset look at follows.

Dataset shape: (39644, 38), Labels: (39644, 1)

x\_train: (31715, 38), y\_train: (31715, 1)

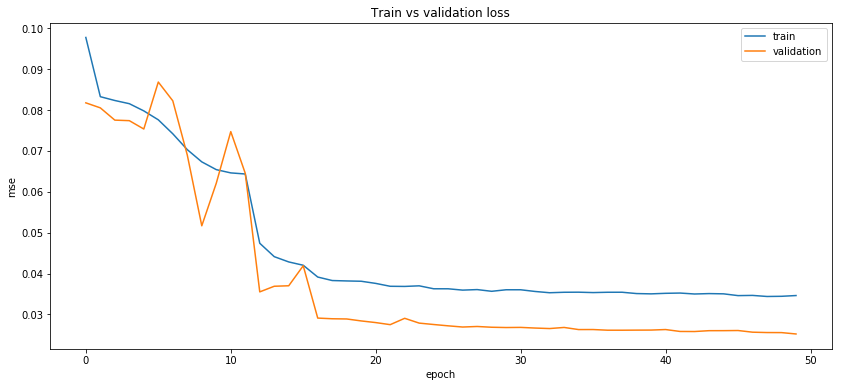
x\_test: (7929, 38), y\_test: (7929, 1)

[**Multilayer Perceptron**](regression_mlp%20-%20shares.ipynb)

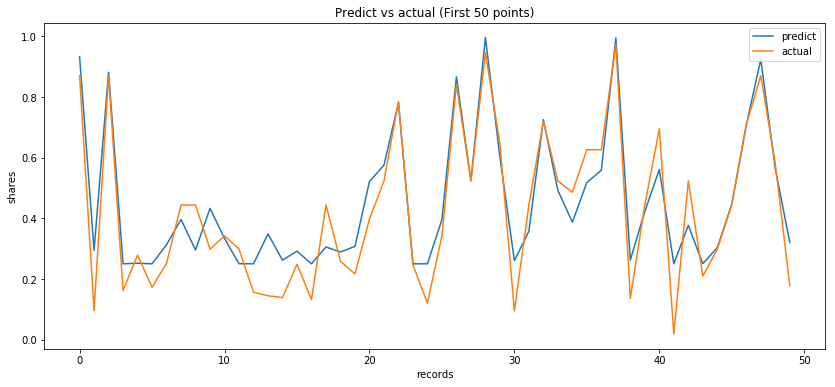
Layer (type) Output Shape Param #   
=================================================================  
dense\_11 (Dense) (None, 512) 20480   
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
dropout\_9 (Dropout) (None, 512) 0   
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
dense\_12 (Dense) (None, 256) 131328   
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
dropout\_10 (Dropout) (None, 256) 0   
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
dense\_13 (Dense) (None, 256) 65792   
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
dropout\_11 (Dropout) (None, 256) 0   
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
dense\_14 (Dense) (None, 128) 32896   
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
dropout\_12 (Dropout) (None, 128) 0   
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
dense\_15 (Dense) (None, 1) 129   
=================================================================  
Total params: 250,625  
Trainable params: 250,625  
Non-trainable params: 0  
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

MLP model consist of 5 layers, each layer is a fully connected layer. To prevent vanishing gradients problem, hidden layers use ReLu for activations. And to avoid overfitting during training, L2 regularization and dropout are used. This model uses mean square error to calculate losses and uses RMSprop as optimizer. In summary, model will accept the 39 attributes (after pre-data preparation) as input and output a single number between 0 and 1. All nodes in model are trainable.

**Evaluating training results**

****

Train and validate loss before 5th epoch shows signs of under fitting. After 10 epoch, loss started to plateau and shows sign of converge for this model. Training loss does not lower beyond validate loss shows model have yet to over fit after 30 epoch.

****

MLP model shows predicted value fluctuate closely to actual value. This model has an issue as it fail to predict value lower than 0.2.

[**Radial Basis Function Network**](regression_rbf%20-%20shares.ipynb)

RBF network is an architecture that has fully connected input layer to a single hidden layer. This hidden layer is then fully connected to the output layer, each hidden node provides a radial basis function of input variables.

In order to improve RBF network performance, a simple MLP network is use to further extract important attributes from the dataset. This extraction network will output 20 nodes that will act as input for the RBF network.

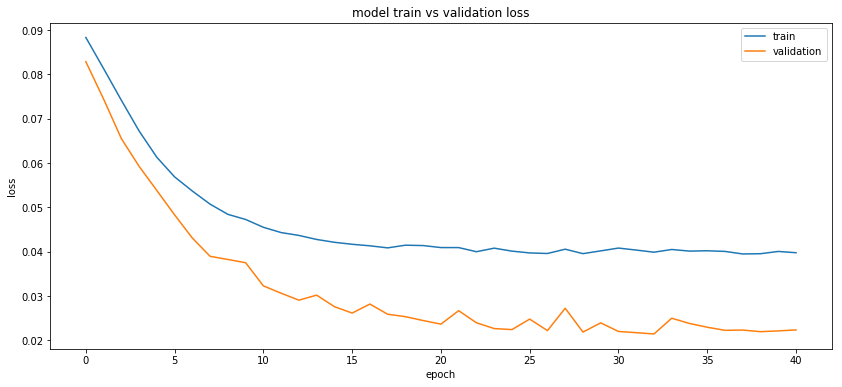
Extraction MLP network before slicing

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
Layer (type) Output Shape Param #   
=================================================================  
dense\_17 (Dense) (None, 128) 5120   
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
dropout\_9 (Dropout) (None, 128) 0   
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
dense\_18 (Dense) (None, 20) 2580   
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
dropout\_10 (Dropout) (None, 20) 0   
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
dense\_19 (Dense) (None, 1) 21   
=================================================================  
Total params: 7,721  
Trainable params: 7,721  
Non-trainable params: 0  
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

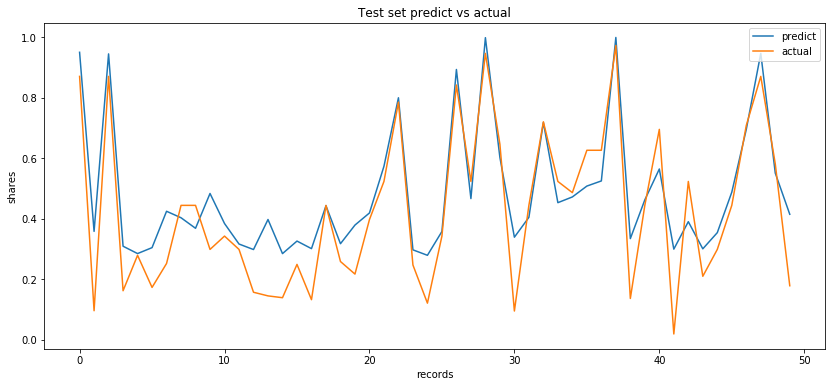
After training, all layers of extraction networking will be freezing to prevent it from learning during the training of RBF network. To output 20 nodes to RBF network the final layer of the extraction network will be removed. After which an RBF network will be attached as final layer to slice extraction network. This network will then go through another round of training with the same dataset.

RBF network attached to freeze extraction MLP network

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
Layer (type) Output Shape Param #   
=================================================================  
dense\_17 (Dense) (None, 128) 5120   
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
dropout\_9 (Dropout) (None, 128) 0   
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
dense\_18 (Dense) (None, 20) 2580   
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
dropout\_10 (Dropout) (None, 20) 0   
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
rbf\_layer\_5 (RBFLayer) (None, 20) 420   
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
dense\_20 (Dense) (None, 1) 21   
=================================================================  
Total params: 8,141  
Trainable params: 441  
Non-trainable params: 7,700  
\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

****

Train and validate loss before 10th epoch shows signs of under fitting. After 15 epoch, loss started to plateau and shows sign of converge for this model. Training loss does not lower beyond validate loss shows model have yet to over fit after 30 epoch. Compare to MLP training and validation curve is much smoother this might be the result of features extraction model apply before RBF training.



MLP model shows predicted value fluctuate closely to actual value. This model has an issue as it fail to predict value lower than 0.2. Similar to MLP model, RBF model suffer from the same problem of not able to learn prediction below 0.2.

**Results**

|  |  |  |  |
| --- | --- | --- | --- |
|  | [**Linear Regression**](data_visualization.ipynb)  **(Baseline)** | [**MLP**](regression_mlp%20-%20shares.ipynb) | [**RBF**](regression_rbf%20-%20shares.ipynb) |
| **Loss** | - | 0.022189 | 0.022356 |
| **MSE (mean-square error)** | 0.065264 | 0.01104 | 0.014396 |
| **RMSE (root-mean-square error)** | 0.255469 | 0.105073 | 0.119984 |
| **Explained Variance Score** | -4.512672 | 0.817475 | 0.762068 |
| **Mean Absolute Error** | 0.216581 | 0.07962 | 0.09119 |
| **Mean Squared Log Error** | 0.032551 | 0.006801 | 0.009118 |
| **Median Absolute Error** | 0.2093 | 0.056895 | 0.069093 |
| **R2** | -4.675613 | 0.810248 | 0.72928 |

|  |  |  |
| --- | --- | --- |
|  | **MLP** | **RBF** |
| **5 KFold Loss** | 0.109718 | 0.109575 |
| **5 KFold MSE** | 0.053234 | 0.069591 |
| **5 KFold R2** | 0.828304 | 0.750599 |
| **5 KFold RMSE** | 0.102846 | 0.117929 |

|  |  |  |
| --- | --- | --- |
|  | **Average ensemble** | **Weighted ensemble** |
| **MSE (mean-square error)** | 0.010654 | 0.00989 |
| **RMSE (root-mean-square error)** | 0.103218 | 0.09945 |
| **Explained Variance Score** | 0.817691 | 0.836669 |
| **Mean Absolute Error** | 0.080815 | 0.075827 |
| **Mean Squared Log Error** | 0.00657 | 0.006131 |
| **Median Absolute Error** | 0.06289 | 0.055935 |
| **R2** | 0.815295 | 0.832654 |

Overall MLP perform slightly better than RBF by looking at 5 kfold results. When using ensemble on both models, weighted ensemble result in better performance compare to average ensemble. And also weighted ensemble provides better result compare to MLP model.