In-Class Kaggle Competition

CompSci 671

Jingxian Huang

NetID: jh654

1 Exploratory Analysis

The provided data is the feature for posts that have different number of likes in a Social Media. For the labels, there are three categories: 0, 1, 2. Where 0 denotes the 'few people like it', 1 denotes 'many people like it' and 2 denotes 'it goes rival'. The task is that given a post of 36 columns of feature and predict the corresponding label of it.

Before digging into the task, I first want to see how the data is distributed, whether there is any noise and how the features are related to the label.

First of all, in order to get the distribution of the dataset to see if there are any abnormity, I used a boxplot to visualize each columns of feature in the training set, because the boxplot contains the information of minimum, first quartile (Q1), median, third quartile (Q3), and maximum. We can also tell the outliers and values of the data or any special discovery like symmetric, skewed, grouped etc. The boxplot of training data is shown as Figure (1). From Figure (1), we can see that all the features are distributed between 0 and , which means that we don't need to apply the normalization to the data during the experiment because they have already been normalized. Besides, there are also some data that are either 0 or 1, these maybe some

boolean data. The dataset is clean and does not have abnormal data points or distributions.

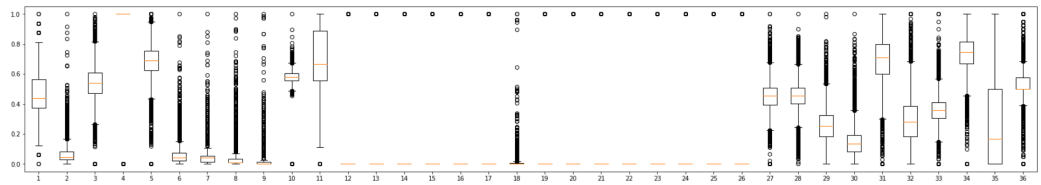

Figure (1) Boxplot of training set.

Another important thing we want to check is that whether the feature of the test dataset is similar to that of the training set. If they are, we know that the model trained by the training set can be applied to the test set, otherwise, the model is meaningless to the test dataset. The boxplot generated by the feature of the test set is shown in Figure (2). By comparing Figure (2) with Figure(1), we find that the test set has very similar distribution as the training set. Therefore, we can apply our training model on the test data.

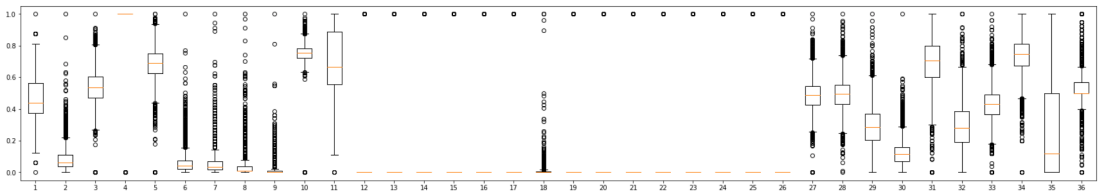

Figure (2) Boxplot of test set.

In order to know whether all features play a role in deciding the label, I used a heatmap to show the correlation of label with other feature. The result is shown as in Figure (3). In this figure, the larger the correlation of two elements, the ‘greener’ is the color of the square; the smaller the correlation is, the ‘redder’ the square is. From the heatmap, we can see the correlation of label with all the feature (the last row that is highlighted by the red circle) and find that all features have very similar colors,

which means that all features play almost equivalent role in determine the label of the post. And therefore, feature selection is not really necessary in this task.

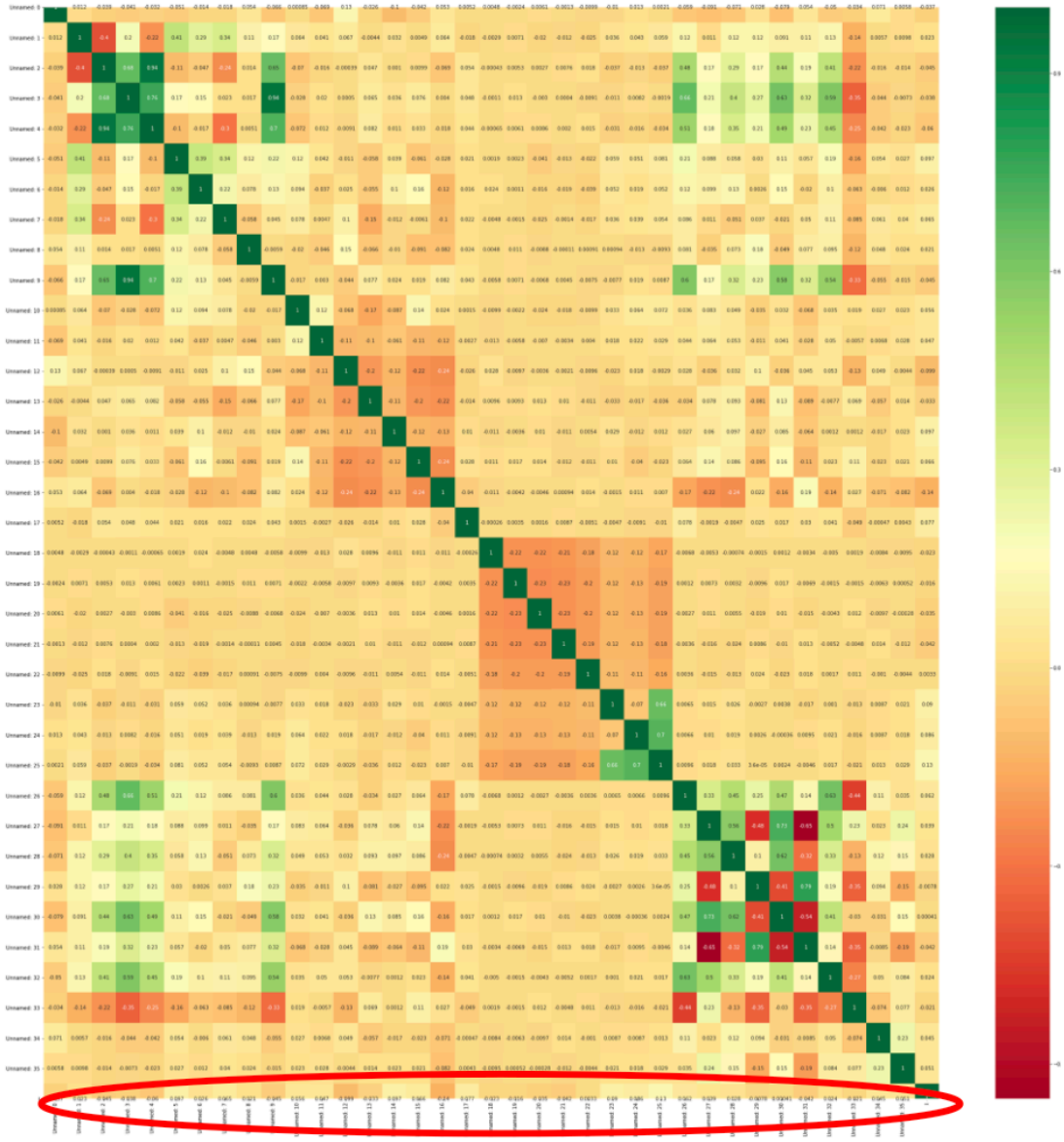

Figure (3) Heatmap of the correlation. The circled part is the correlation of different features corresponding to the label.

2 Models

I choose decision tree (and decision forest) and neural networks to predict the label.

2.1 Decision Tree and decision forest

Decision Tree and decision forest are very similar algorithms, so I count them as one algorithm. They use tree-like structure in which each tree node test on one of the feature and the outcome is represented by the branches. By checking each feature, it can make a decision which category the post is in. Decision trees are also easy to understand and visualize [1]. Therefore, decision tree algorithm is suitable for supervised classification problems.

For decision tree, there are many algorithms to build and I choose CART algorithm (uses Gini Index) and ID3 algorithm (uses Information Gain) [2].

Decision forest is constructed of many decision trees and choose on the label that is vote most by the decision trees.

2.2 Neural Network

Neural network is another model for supervised classification problems. By updating parameters along the propagation process, the neural network learns the best model to classification problems. In my model, I used different layers of network with different activation functions, different learning rate and see how these changes influences the prediction.

3 Training

3.1 Environment and component

Component	Version	Usage
Python	3.6.9	The main language
Jupyter Notebook	6.0.1	Serves as the primary platform
Numpy	1.17.4	Performs basic data manipulation

Pandas	0.25.3	Used to read the data
Sklearn	0.22	Mainly used for decision tree and decision forest
Pytorch	1.3.1	Mainly used for neural network
Matplotlib	3.1.1	Used for data visualization
Seaborn	0.9.0	Used for data visualization

Table (1) Environment and component.

3.2 Dataset

The training set contains 15000 rows of data and each have 36 columns of features.

The test dataset contains 5000 rows of data and each have 36 columns of features.

3.3 Decision Tree and Random Forest

For decision tree, I used Sklearn to generate the model. During the training process, I applied 'Entropy' criterion and max_depth is 5.

For random forest, I choose the number of trees to be 10, the criterion to be 'Entropy' and max_depth be 5.

3.4 Neural Network

I used mainly PyTorch to build my neural network model.

In neural network, I used a structure of with four layers (one input layer, two hidden layers and one output layer) and the function of the network is as follows:

$$h_1 = \sigma(W_1 x + b_1)$$

$$h_2 = \sigma(W_2 h_1 + b_2)$$

$$o = W_3 h_2 + b_3$$

Where x is the input and o is the output. h_1 and h_2 are both hidden layers. W_1, W_2, W_3 are of size [36,24], [24,12], [12,3]; b_1, b_2, b_3 are of size [24,1], [12,1], [3,1]. σ is the Relu activation function. The learning rate is 0.01.

During training, I ran 100 iterations with batch size 10 and the optimizer is SGD. The loss is calculated with cross entropy.

4 Hyperparameter Selection

4.1 Decision Tree and Decision Forest

4.1.1 Decision Tree

In decision tree, I tried two criterion: 'Gini' and 'Entropy' while fix max_depth as 5. And use the metrice in Sklearn to compute accuracy. The result is as follows:

Criterion	Accuracy
Gini	0.568
Entropy	0.58

Table (2) Accuracy of decision tree with different criterion.

It can be seen from Table (2) that 'Entropy' criterion has higher accuracy.

Then I fix the criterion to be 'Entropy' and tried different max_depth, ranging in (1, 3, 5, 8, 10, 100, 1000) and the result is shown in Figure (4). We can see that the Accuracy reaches the maximum when max_depth equals to 5.

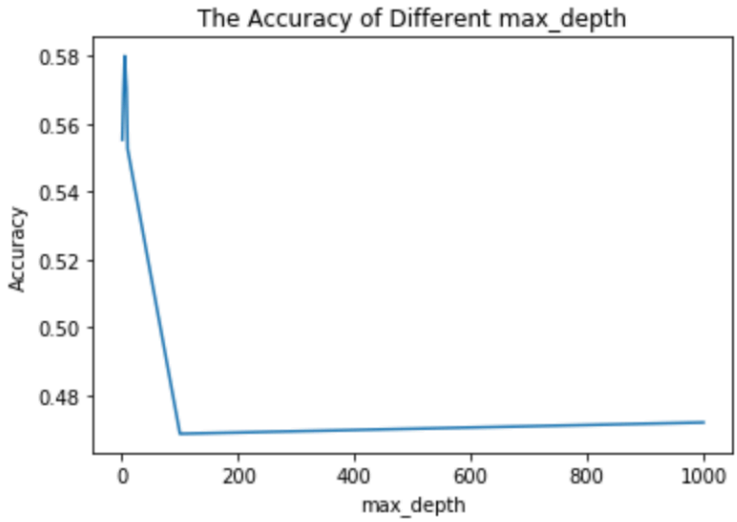

Figure (4) The accuracy of decision tree with different max_depth

4.1.2 Decision Forest

For decision forest, I fixed max_depth to be 5, criterion to be 'Entropy' and try different number of trees ranging in (1, 10, 100, 1000).

The result is shown in Figure (5). We can see that the accuracy reaches maximum when there are 10 trees.

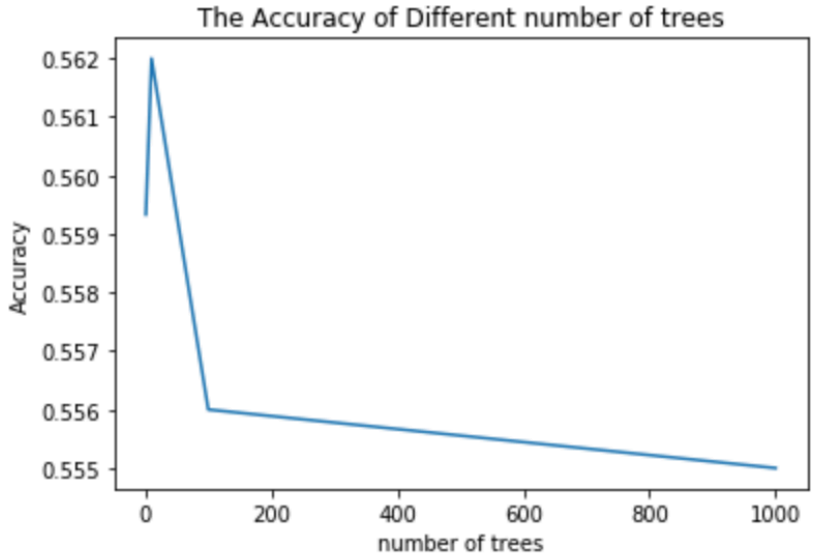

Figure (5) The accuracy of decision forests with different number of trees

4.2 Neural Network

In this network, I use a network structure with four layers and fix the optimizer to be SGD. Since the loss converge fast, I set the epoch to be 100. I also fix batch size to be 10. The hyperparameter I want to tune is the size of the hidden layers, different activation functions and different learning rate.

4.2.1 Hidden Layer Size

I first tried different size of hidden layer, fix the activation function to be sigmoid and the learning rate to be 0.001. I trained the networks on the same training set and validation set. The result is in Table (3).

Size of $W_1,\ W_2,\ W_3$	Accuracy
[36,24], [24,12], [12,3]	0.5736666666666667
[36,36], [36,36], [36,3]	0.5726666666666667
[36,48], [48,64], [64, 3]	0.567

Table (3) The accuracy of different neural network with different hidden layer sizes

It can be seen that [36, 24], [24, 12], [12, 3] performs best. [36,36], [36,36], [36,3] is not as good as the first one but the performance is not too different. However, if we expand the size of the network, the performance obviously drops.

4.2.2 Activation Function

Then I fix the size of the hidden layers to be [36, 24], [24, 12], [12, 3] and fix the learning rate to be 0.001. I tried sigmoid and Relu as activation functions. The result is in Table (4).

Activation function	Accuracy

sigmoid	0.5736666666666667
Relu	0.5756666666666667

Table (4) The accuracy of different neural network with different activation functions

We can see from Table (4) that the accuracy of two activation functions do not differ much and Relu performs slightly better.

4.2.3 Learning Rate

Then we fix the hidden layer size to be [36,24], [24,12], [12,3] and the activation function to be Relu, and train the network on learning rate ranging in [0.001, 0.01, 0.1, 1]. The result is shown in Figure (6) and we can see that the model achieves the best result when learning rate is 0.01.

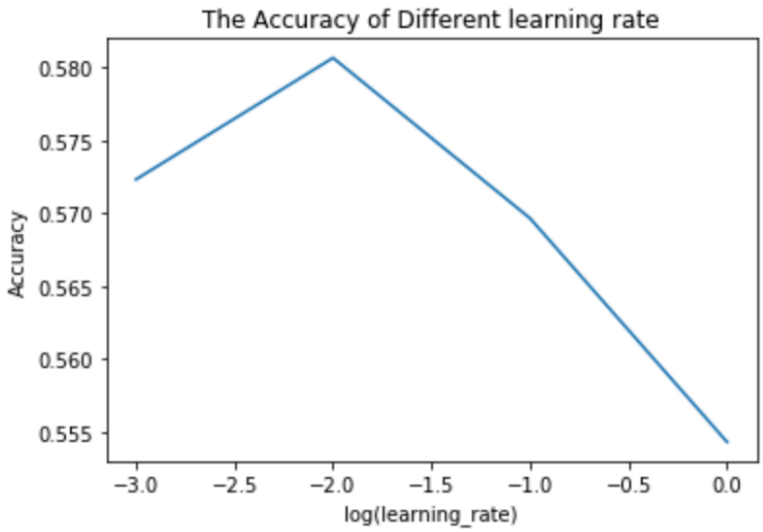

Figure (6) The accuracy of different neural network with different learning rate

5 Data Splits

In order to do cross validation, I split the training data into five subsets and each time, I use one of them as the validation set and the rest as the training set. After train on

the model(learning rate 0.01; three hidden layers of size [36,24], [24,12], [12,3]; Relu activation function; SGD optimizer) on each training set and apply them on the corresponding validation set and select the model that performs best and apply it to the test set.

To test whether it overfits, I apply the model to the corresponding training set and see that the accuracy is 0.5785833333333333 while the accuracy on the validation set is 0.587, which are very similar. Therefore, it is not overfitted.

6 Errors and Mistakes

Since I calculated the correlation of each feature with regard to the label, I try to check whether selection of feature will help increase the accuracy. Therefore, in the decision tree part, I selected 16 features that has the highest correlation with the label. However, I find that the accuracy does not increase. This may because that the correlation of each feature with regard to label is very similar, therefore, all features play relative equal roles.

7 Predictive Accuracy

I submitted my result on test dataset to the Leaderboard with username: **Jingxian Huang** and achieve score of 0.42466.

The accuracy on different models on validation set is in Table (5).

Model	Accuracy
Decision Tree - Max_depth: 5 - Criterion: Entropy	0.58
Decision Forest - Max_depth: 5 - Criterion: Entropy	0.5593333333333333

- Number of trees: 10	
Neural Network - Hidden Layer: 3 - Hidden Layer size: [36,24], [24,12], [12,3] - Activation function: Relu - Learning rate: 0.01 - epoch: 100 - batch size: 10	0.587

Table (5). Accuracy of different model on validation set

8 Code

The code is pasted in the following link:

https://github.com/huangjingxian/Social-Media-Data/blob/master/classification.ipynb

Reference

[1]. "Introduction to Decision Trees" [Online]. Available: https://medium.com/greyatom/decision-trees-a-simple-way-to-visualize-a-decision-dc506a403aeb. [Accessed: 14-Dec-2019]

[2]. "Chapter 4: Decision Trees Algorithms" [Online]. Available: https://medium.com/deep-math-machine-learning-ai/chapter-4-decision-trees-algorithms-b93975f7a1f1. [Accessed: 14-Dec-2019]