Write up for programming exercise

SYDE 675

Programming Assignment 4

Zhiqi Bei

Wenhan Liu

Write up for exercise 1

For this Assignment, the train\_data was first loaded using numpy and its dimension and the dimension of each image were checked. Then, by using a for loop and reshape function, the image of each digit was printed. Next, by setting the range of x-axis from 0 to 9 and range of y-axis from 0 to 0.12, a normalized histogram was plotted with the use of the weight parameter in plt.hist function. Then, by finding the index of each digit in the dataset, the distance of each digit in dig\_list with every other digit in the dataset was calculated, and from the distance list, the datapoint with smallest distance was found and displayed along with the sample datapoint. Then, by finding the zeros and ones data in the dataset, the zero to zero, one to one, and zero to one distance were calculted and resulted in genuine and impostor distances, and a histogram about them was then plotted. Next, with the use of a for loop, the ROC Curve was plotted, and the two different error rates were shown. Then, a KNN was implemented, the distance between datapoint and every other data in the dataset was calculated, and the k smallest were chosen, next by equally voting, the final decision was made, and this was done by using 3 for loops. Finally, by separating the training\_data to train and label, and 3-fold cross validation was performed.

The accuracy along with the heatmap for the 3 trials are as follows:

Timeline

Description automatically generated

Timeline

Description automatically generated

Timeline, treemap chart

Description automatically generated

With each datapoint in the testing set, KNN was used to perform the prediction and the accuracy was calculted. From the heatmap that got plotted, we could see that the digit 5 is particularly tricky to classify.

And finally, the Kaggle results are as shown below:

Graphical user interface, text

Description automatically generated

Write up for exercise 2

**Loading data**

First is loading the data, we name the data from train.csv as training\_data, and the data from test.csv as testing\_data.

After loading the data, we can find that there are a lot of missing data in the training dataset.

表格

描述已自动生成

“Age” has 177 missing data, “Cabin” has 687 missing data, and “Embarked” has 2 missing data, so next we need to process this data.

**Process the missing “Age” data in the training set**

图表, 直方图

描述已自动生成

By visualizing the “Age” data, we can see from the above graph, the skewness is not 0, then using the median of age to fill the missing data is better than using the mean.

Therefore, we choose to use the median to fill in missing data, which is 28.

**Process the missing “Cabin” data in the training set**

Since the “cabin” is missing too much data, we decided to remove this feature

**Process the missing “Embarked” data in the training set**

图形用户界面, 文本, 应用程序

描述已自动生成

We observed that there are 3 boarding ports, and most of the people are from port S. Therefore, we choose “S” to fill the missing “Embarked” data.

**Check the final training dataset**

表格

描述已自动生成

After processing all the missing data, we check the new taring data to see if there still has missing data. There is no missing data, so we go to the next step.

**Process other features in training set**

Most people have different names and ticket number, so we decided to remove those data. And using one-hot encoding convert “Embarked” and “Sex” to number. Then the final training data set is:

应用程序

低可信度描述已自动生成

**Process the missing data in testing set**

There are also have missing data in the testing set.

表格

描述已自动生成

“Age” has 86 missing data, “Fare” has 1 missing data, and “Cabin” has 327 missing data. Similarly, as the training set, we use the median to fill the missing data in “Age” and remove the “Cabin” feature.

For the missing “Fare” data, we visualize the “Fare” data and observe that the skewness is not 0, so we choose the median of fare to fill the missing data.

图表, 直方图

描述已自动生成

At last, we also remove the “Name” and “Ticket” from the testing set. The final testing set is as below:

表格

描述已自动生成

**Logistic regression**

Then we can do the logistic regression, we first train the classifier using the training data, we split the training set into X\_train, X\_test, Y\_train, Y\_Test by 8:2, 8 is the training and 2 is the testing.

文本

描述已自动生成

The prediction accuracy is 0.79

**Train the classifier using all the training data**

Then we can use all the training data to train the classifier and test it using the testing data from the test.csv and write down the result to an csv file.

图形用户界面

低可信度描述已自动生成

**Submit to Kaggle**

At last, we submit this file to Kaggle, and below is the score.

图形用户界面, 应用程序

描述已自动生成