Final Project

Jingwei BAO | JBL1457 Jingyao YU | JYB4658

1. Topic

Our task is to predict the rating of Apps based on attributes of the 12 terms, includes App, Category, Reviews, Size, Installs, Type, Price, Content Rating, Genres, Last Updated, Current Ver and Android Ver.

2. Methods

We choose KNN and Decision tree to train our data and use LOOCV to test our Machine Learning model.

For training:

We implement KNN first to train the model and it shows that when finding nearest 33 neighbors, we could get the most accuracy which is only 14%, so we feel it would not be a good idea so we implement Decision tree to train the raw data and the result shows we are right.

For testing and evaluation:

We implement LOOCV by dividing the data into 10 chunks which 9 set of data used to training and 1 left used for testing.

3. Dataset

We are using data set "GoogleplayStore", which contains 13 attributes, includes

App: String
Category: String
Rating: Number
Reviews: Number
Size: Number
Installs: String
Type: String
Price: Number
Content Rating: String
Genres: String
Last Updated: String
Current Ver: Number
Android Ver: Number

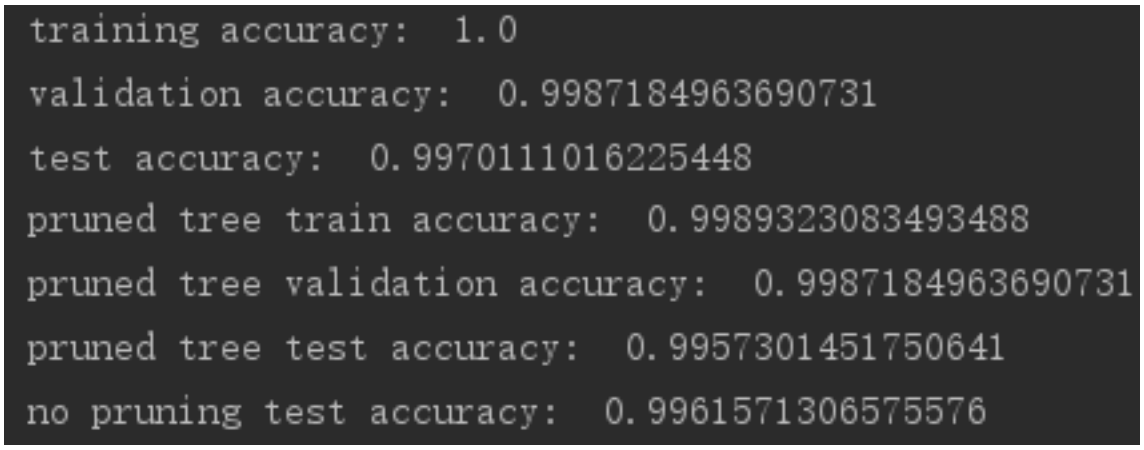
We first change the parse function to change the format of data from Google Play Store Apps (from www.kaggle.com) into "UTF-8" format.

For installs: we delete "," and "+" in the numbers of Installs and change it back to number type;
For size: we transfer all "… MB" and "… KB" into the same format which is "… MB" based on binary system.
For Android Ver: we use NAN to represent default value.
For Price: we delete the symbol "$" so that it could change to number type which makes sense when training.

For category, reviews and genres: we use Labelencoder to transfer all String type into int type which could be used in training after transformation.

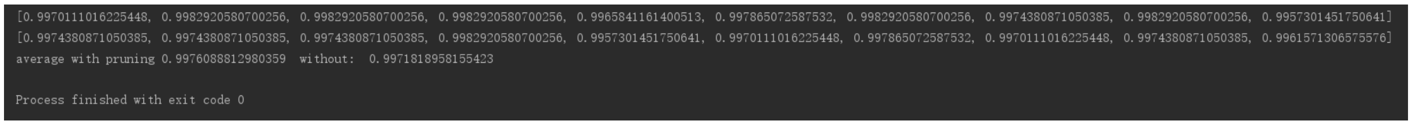
4. Results

This is a random result which we run it once.



```
training accuracy:  1.0
validation accuracy:  0.9987184963690731
test accuracy:  0.997011016225448
pruned tree train accuracy:  0.9989323083493488
pruned tree validation accuracy:  0.9987184963690731
pruned tree test accuracy:  0.9957301451750641
no pruning test accuracy:  0.996157130657557б
```

There are all the 10 results and average results.

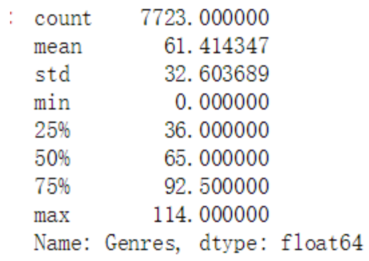


```
[0.997011016225448, 0.9982920580700256, 0.9982920580700256, 0.9982920580700256, 0.9965841161400513, 0.997865072587532, 0.9982920580700256, 0.9974380871050385, 0.9982920580700256, 0.9957301451750641]
[0.9974380871050385, 0.9974380871050385, 0.9974380871050385, 0.9982920580700256, 0.9957301451750641, 0.997011016225448, 0.997865072587532, 0.997011016225448, 0.9974380871050385, 0.9961571306575576]
average with pruning 0.9976088812980359  without:  0.9971818958155423

Process finished with exit code 0
```

5. Analysis and suggestions for future work

For KNN training, if we change rating to the integers (rather than float), the accuracy was about 61%. The best result using KNN shows when raw data is float and evaluation data is integer. If all changed to integer or float, it will be lower. It may because the data is not comparatively equal, which means some sample is far more than other samples and the graph is shown below.



```
: count    7723.000000
  mean       61.414347
  std        32.603689
  min         0.000000
  25%        36.000000
  50%        65.000000
  75%        92.500000
  max       114.000000
  Name: Genres, dtype: float64
```

For Decision tree, the accuracy is about 98% which shows that will be a good Machine Learning model for this dataset.