hw5

Qianhui Yang

10/28/2020

Q1

The accuracy is 0.918, is much better than the last homework GBM result which is only about 0.5

```
## # A tibble: 6 x 3
## # Groups:   Gender [2]
##   Gender exp_group     n
##   <fct>  <chr>     <int>
## 1 Female test       1666
## 2 Female train      1667
## 3 Female validate   1667
## 4 Male   test       1666
## 5 Male   train      1667
## 6 Male   validate   1667
```

```
## [1] 0.9211158
```

Q2

1. there are some missing data represented by power=0 and total=5, which are omitted, the omitted data has 434 rows of observations.
2. we need two components to get 85% of variation in the data set
3. Yes. Because the Durability has a range between 0-120, while the rest of the variables have 0-100. The normalization will make sure that each variable weight the same
4. Yes, the "total" column really is the total as the values in the other columns
5. If we include the total column in the PCA, the largest principle components PC1 has Total column correspond the largest proportion (0.52) 6.PCA can't classify the alignment of superhero, PCA function on the linear correlation, indicate that in each group of alignment, there are little linear correlation between each other.May be because the fact that the alignment of superhero does not relate to their ability, it is randomly assigned by the writer.

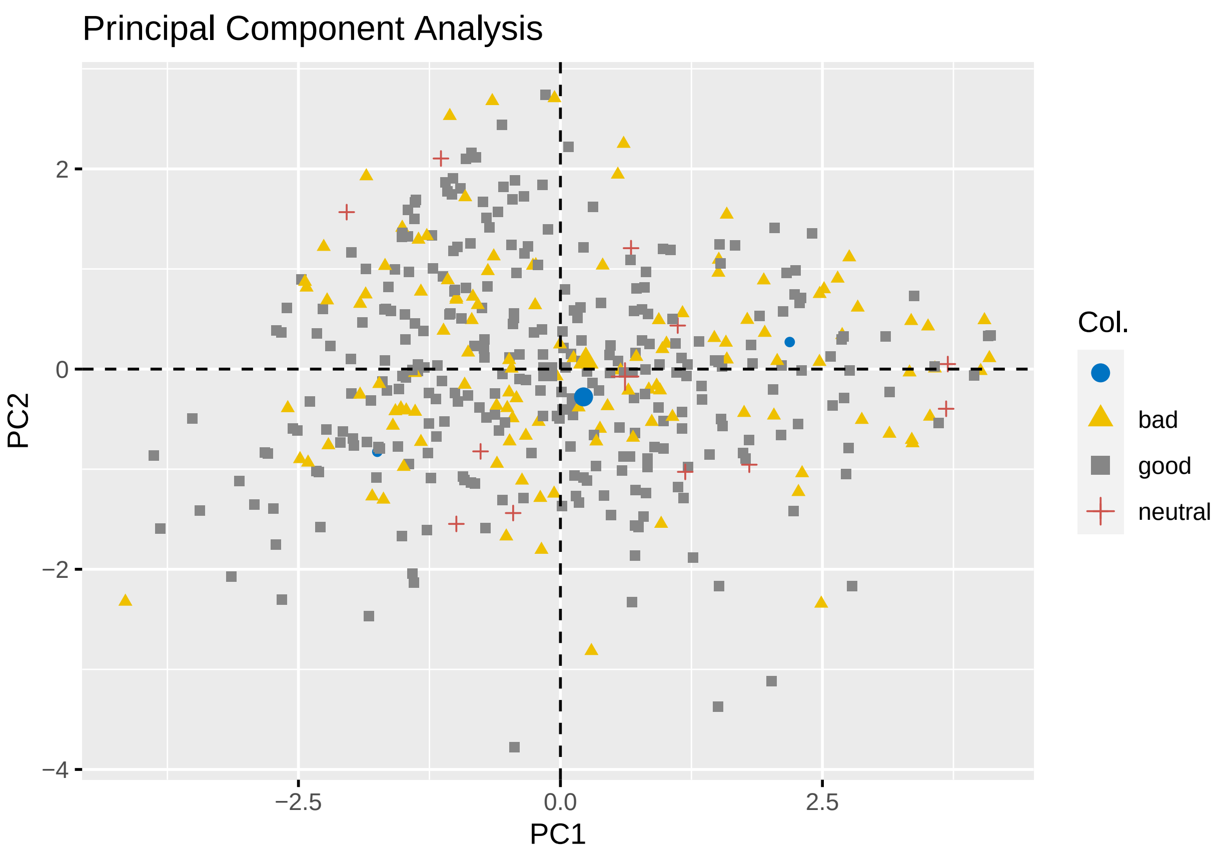
```
## Importance of components:
##                           PC1     PC2     PC3      PC4      PC5      PC6
## Standard deviation     46.664 23.6134 22.8884 18.88294 17.74412 17.02230
## Proportion of Variance  0.516  0.1321  0.1241  0.08449  0.07461  0.06866
## Cumulative Proportion   0.516  0.6481  0.7722  0.85673  0.93134  1.00000
```

```
## Importance of components:
##                           PC1    PC2    PC3    PC4     PC5     PC6
## Standard deviation     1.6412 1.0353 0.8695 0.7831 0.74653 0.55485
## Proportion of Variance 0.4489 0.1787 0.1260 0.1022 0.09289 0.05131
## Cumulative Proportion  0.4489 0.6276 0.7536 0.8558 0.94869 1.00000
```

```
## Warning: In prcomp.default(total_super, scale. = T, graph = FALSE) :
##  extra argument 'graph' will be disregarded
```

```
## Importance of components:
##                           PC1    PC2    PC3     PC4     PC5     PC6       PC7
## Standard deviation     1.9211 1.0366 0.8695 0.78316 0.74661 0.55491 2.856e-16
## Proportion of Variance 0.5272 0.1535 0.1080 0.08762 0.07963 0.04399 0.000e+00
## Cumulative Proportion  0.5272 0.6807 0.7888 0.87638 0.95601 1.00000 1.000e+00
```

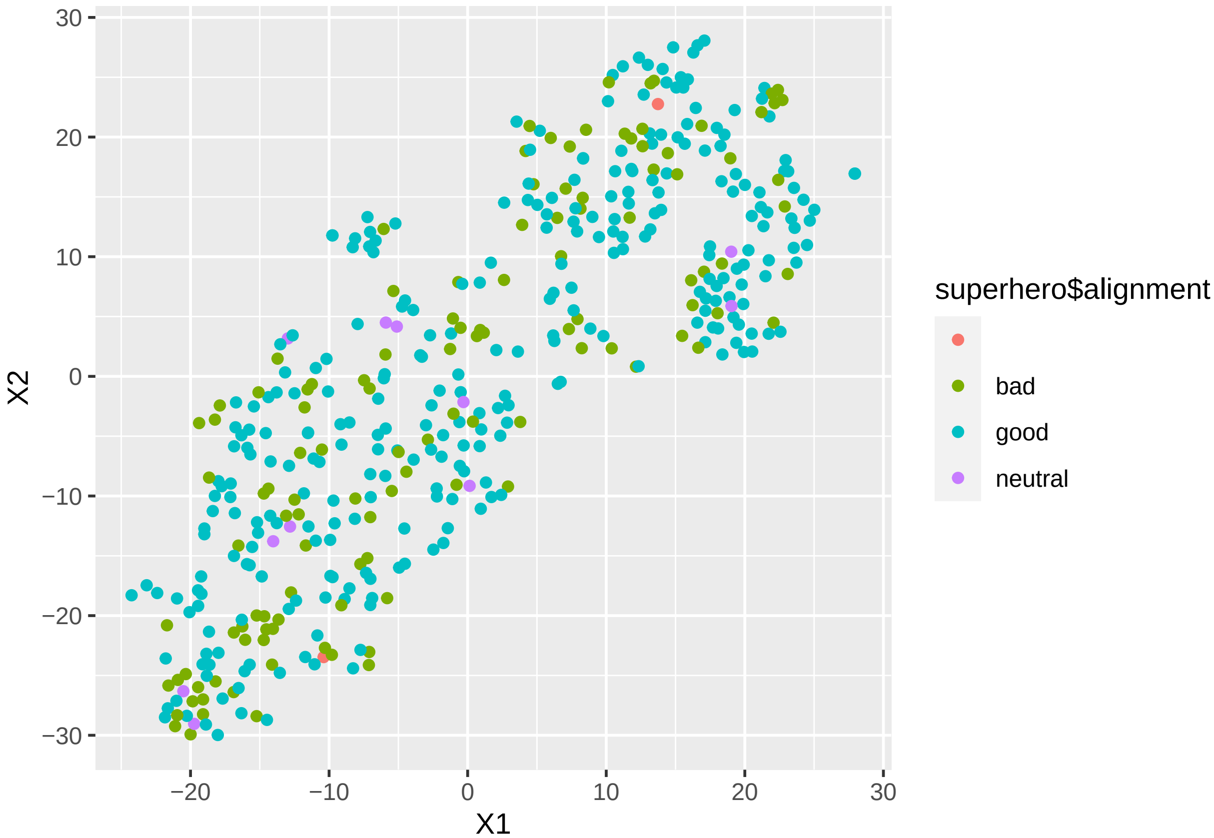
```
##                    PC1         PC2         PC3         PC4         PC5
## Intelligence 0.2496939 -0.60774304  0.60332047 -0.01526458  0.41765874
## Strength     0.4213660  0.16406255 -0.28736310  0.26366604  0.36562901
## Speed        0.3469851  0.28532893  0.03652766 -0.87143286  0.04583000
## Durability   0.4280366  0.20751276 -0.20342168  0.31444116  0.18853211
## Power        0.3578820  0.20627640  0.48732635  0.23723409 -0.68592601
## Combat       0.2407513 -0.65969518 -0.52266011 -0.12482382 -0.42847391
## Total        0.5200471 -0.03885949 -0.00718631  0.01085131 -0.01214364
##                       PC6        PC7
## Intelligence  0.036805516 -0.1681645
## Strength     -0.665084678 -0.2596432
## Speed         0.026818310 -0.1861172
## Durability    0.733783483 -0.2438995
## Power        -0.130336944 -0.2193658
## Combat        0.003081813 -0.1857071
## Total        -0.012372817  0.8529779
```



Q3

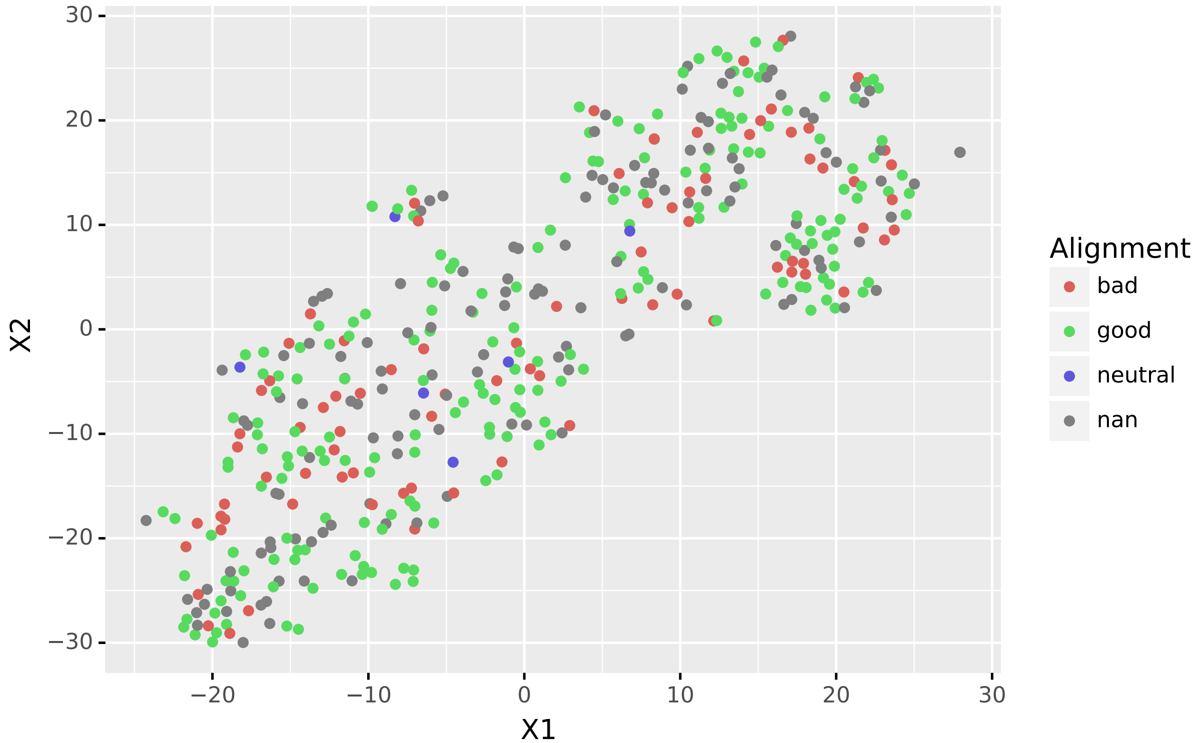
Similar to PCA, TSNE did not do show a classification of superhero alignment. t-SNE (t-Distributed Stochastic Neighbor Embedding) is nonlinear dimensionality reduction technique in which interrelated high dimensional data (usually hundreds or thousands of variables) is mapped into low-dimensional data (like

2 or 3 variables) while preserving the significant structure (relationship among the data points in different variables) of original high dimensional data. The result shows no nonlinear correlation in the superhero alignment group.




```
## List of 1
##  $ legend.position: chr "right"
##  - attr(*, "class")= chr [1:2] "theme" "gg"
##  - attr(*, "complete")= logi FALSE
##  - attr(*, "validate")= logi TRUE
```

Q4



The python codes are hided in the report but can be found in the Rmarkdown code.

#Q5 The best accuracy is 0.71.

```
## # A tibble: 6 x 3
## # Groups:   alignment [2]
##   alignment exp_group     n
##   <chr>     <chr>     <int>
## 1 bad       test         40
## 2 bad       train        41
## 3 bad       validate     40
## 4 good      test         99
## 5 good      train       100
## 6 good      validate     99
```

```
## Stochastic Gradient Boosting
##
## 141 samples
##   6 predictor
##   2 classes: 'bad', 'good'
##
## No pre-processing
## Resampling: Cross-Validated (10 fold, repeated 10 times)
## Summary of sample sizes: 126, 127, 127, 127, 127, 127, ...
## Resampling results across tuning parameters:
##
##   interaction.depth  n.trees  Accuracy   Kappa
##   1                   50      0.7021905  0.099797248
```

```
##   1                    100      0.6838095  0.090196111
##   1                    150      0.6653333  0.063186900
##   2                     50      0.6732857  0.063298446
##   2                    100      0.6433333  0.021550071
##   2                    150      0.6263333  0.003287784
##   3                     50      0.6687619  0.068863089
##   3                    100      0.6305714  0.015934602
##   3                    150      0.6403333  0.059661551
##
## Tuning parameter 'shrinkage' was held constant at a value of 0.1
##
## Tuning parameter 'n.minobsinnode' was held constant at a value of 10
## Accuracy was used to select the optimal model using the largest value.
## The final values used for the model were n.trees = 50, interaction.depth =
##  1, shrinkage = 0.1 and n.minobsinnode = 10.
```

```
## [1] 0.705036
```

Q6

A conceptual question: why do we need to characterize our models using strategies like k-fold cross validation? Why can't we just report a single number for the accuracy of our model?

No. Because it is possible that we have selected data that can't represent our data set. It can not predict the data which is known as overfiting. To prevent this happens and make sure we can repeat the result, we use cross validation to lower the bias.

Q7

Describe in words the process of recursive feature elimination.

First, after the initial set of feature training, the importance of each feature which is calculated by the coefficient and feature importance attribute. Then, the least important feature will be eliminated and from the current model and result as a less featured model. The process in repeated until the feature is eliminated to the numbers we want.