STA363_Client_Challenge2

Alex Zhang

2024-02-09

Part 1

I will recommend linear regression approach.

If we want to see how different features relate to the response variable, a linear regression is often used. In this case, it will provide a formal mathematical equation with all parameters explaining the relation between different features and how it will affect the number of times student used ChatGPT for their homework. We are able to tell, for example, whether increasing students' age will increase the number of times student used ChatGPT. If so, we can tell by how much from the regression model.

KNN is not helpful if we are trying to find out relation between features and response variable. KNN computes the similarity between different rows of the data, and predicts the value based on some closest rows. We could not tell whether a feature has a positive or negative relation to the number of times for using ChatGPT. We can only get predicted values.

Part 2

I will not recommend proceeding with the plan the colleague has suggested.

The first potential problem is that the training size has been reduced a lot. Reminds from handling the missing data part, missing about 11% of the data is already unacceptable. With this approach, we reduced our training size about 20%. Losing this amount of data may lead to a failure of capturing all possible patterns given that this dataset is not big.

The second potential problem is the way of choosing training set and validation set. We cannot just move the first 520 rows into training set because we haven't checked how data is arranged. It is possible that this data is sorted based on whether a student has a computer at home. This may cause the training set to only contain student who has a computer at home and validation set only contains the student who do not. In this case, the predictive ability may not be really good because training set has missed a pattern.

Even if each row in the data is randomly placed, there is still a problem for this method. The predictive ability of our model will change if we move first 130 rows into validation set and the rest into training set. Though this time the numbers of rows in two sets are exactly the same as the colleague's plan, the predictive power will change since we do not have the same rows in training and validation set. we cannot determine whether this linear regression approach is good or not based on various predicting performances.

Part 3

```
# Create data frame to store the value
KNNRMSE <- data.frame("K" = rep(NA, 23), "valRMSE" = rep(NA, 23))
KNNRMSE$K <- as.integer(KNNRMSE$K)
```

```
for(i in 1:23){
  # Perform KNN with training and testing set given k from 3 to 25
  knnPred <- knnGower(train[,-c(1:2)],test[,-c(1:2)], train[,2], k = (i+2))

  # Store each k values and corresponding RMSE values
  KNNRMSE[i,"K"] <- (i+2);
  KNNRMSE[i, "valRMSE"] <- compute_RMSE(test$price, knnPred)

}
# Change data type for K
KNNRMSE$K <- as.integer(KNNRMSE$K)
```

```
# Create a table to show all values
knitr::kable(KNNRMSE)
```

K	valRMSE
3	160.8978
4	153.9546
5	151.0481
6	149.3054
7	150.3088
8	236.5467
9	147.1044
10	147.3770
11	146.9821
12	148.0015
13	145.8796
14	145.7187
15	145.9623
16	146.7323
17	147.6556
18	147.7817
19	148.2547
20	147.7595
21	148.3158
22	148.5848
23	149.0222
24	148.8837
25	148.8884

Based on the result we get from the table, we should choose our K = 14. Because we have the smallest RMSE value this time. This means on average our predicted values are closest to the true test values.