FINAL SUBMISSION 1

- Rushat Chopra (20854690)

Algorithm 1)
Training MSE - 0.2922281
Training R^2 - 0.6439893
Using random forest and regression trees to get the value.
I have compared the results of both, however, the regression tree gave a better value in the end. The random forest gave a close result and I have included it in my code in order to get the variable importance which is due in submission 2.

Importing the dataset and then making a few changes::

```
1   ### importing dataset :
2   data <- read.csv("Desktop/training_small_final_pc.csv", header = TRUE)
3
4   library(tree)
5   library(randomForest)
6   set.seed(100)
7   train<- sample(1:nrow(data),500)
8   data$sellingprice<- log(data$sellingprice)
9
10  data.test = data[-train, ]
11  sp.test<- data[-train,"sellingprice"]
12
```

Implementing a random forest:

```
13  ### Using random forest and bagging in order to see the MSE and R^2
14  ### to compare the final value. RF helps gives a better variable importance.
15  rf.sp <- randomForest(sellingprice~.-mmr, data = data, subset = train,|
16                        importance = TRUE,na.action=na.omit,mtry=6)
17  yhat.rf <- predict(rf.sp,newdata = data.test)
18  yhat.rf <- na.roughfix(yhat.rf)
19  mse <- mean((yhat.rf - sp.test)^2)
20  mse
21  rss<- mean((yhat.rf - sp.test)^2)
22  tss <- mean((data$sellingprice-mean(sp.test))^2)
23  rsq <- 1 - rss/tss
24  rsq
25  importance(rf.sp)
26
27  varImpPlot(rf.sp)
```

Values from the random forest (as received):

```
> mse <- mean((yhat.rf - sp.test)^2)
> mse
[1] 0.3131778
> rss<- mean((yhat.rf - sp.test)^2)
> tss <- mean((data$sellingprice-mean(sp.test))^2)
> rsq <- 1 - rss/tss
> rsq
[1] 0.618467
```

The above is low accuracy

Thus, implementing a regression tree :

```
29  ## To get a better MSE and R^2, using regression trees which is giving a better value
30  tree.sp <- tree(sellingprice~.-mmr, data = data , subset = train)
31  plot(tree.sp)
32  text(tree.sp, pretty = 0)
33  yhat.tree <-  predict(tree.sp, newdata = data.test)
34  prune.sp<- prune.tree(tree.sp)
35  mse <- mean((yhat.tree-sp.test)^2)
36  mse
37  rss<- mean((yhat.tree-sp.test)^2)
38  tss <- mean((data$sellingprice-mean(sp.test))^2)
39  rsq<- 1 - rss/tss
40  rsq
41
42
```

Value received from the regression tree:

```
> mse <- mean((yhat.tree-sp.test)^2)
> mse
[1] 0.2922281
> rss<- mean((yhat.tree-sp.test)^2)
> tss <- mean((data$sellingprice-mean(sp.test))^2)
> rsq<- 1 - rss/tss
> rsq
[1] 0.6439893
```

Giving us a higher level of accuracy than the random forest.

Algorithm 2)

Training MSE - 0.01444225
Training R^2 - 0.7427869
Used boosting for this algorithm.

Importing the data and making a few changes :

```
1   ### importing dataset :
2   data <- read.csv("Desktop/training_small_final_pc.csv", header = TRUE)
3   library(gbm)
4   library(randomForest)
5   set.seed(100)
6   train<- sample(1:nrow(data),500)
7   data$sellingprice<- log(data$sellingprice)
8   data$mmr <- log(data$mmr)
9
10  ### new column for the difference between the selling price and mmr
11  data$difference <- data$sellingprice - data$mmr
```

Using a random forest to check the importance of all the variables and how they correlate to the difference :

```
14  ### most important features when predicting the diff. :::
15  bag.diff <- randomForest(difference~., data = data, subset = train,
16                           importance = TRUE,na.action=na.omit, ntree = 25)
17  importance(bag.diff)
18  ### we get that mmr and sleeing price have the highest influence .
19
```

```
> importance(bag.diff)
                %IncMSE IncNodePurity
year          2.68564776    0.9811430
make          1.22490973    0.8714997
model         0.01278847    0.5412341
trim          3.16206564    1.0339857
body         -0.80993032    0.4965083
transmission  0.49400740    0.1072895
vin          -0.57339560    1.6599613
state        -1.96065397    0.7080283
condition     1.31753097    2.3651459
odometer      1.96040117    1.5877221
color        -1.77041153    0.5156934
interior     -0.86621311    0.3140183
seller        0.62614086    0.5916099
mmr           4.00485306    2.9087377
sellingprice  4.54952052    4.6211607
saledate      0.70689472    0.8054446
```

As we can see, and confirm, the most important variables in the difference are obviously the selling price and the mmr.

Using boosting to get a prediction:

```
21  ### using boosting in order to predict the difference best using the most
22  ## influential features:
23
24  boost.diff <- gbm(difference~mmr+sellingprice, data = data[train,],
25                    distribution = "gaussian",n.trees = 5000, interaction.depth = 4)
26  summary(boost.diff) ## sellingprice has a little higher influence than mmr.
27  yhat.boost<- predict(boost.diff, newdata= data[-train,], n.trees=5000)
28  diff.test<- data[-train,"difference"]
29  mse <- mean((yhat.boost-diff.test)^2)
30  mse
31  rss<- mean((yhat.boost-diff.test)^2)
32  tss <- mean(((data$difference-mean(diff.test))^2))
33  rsq <- 1 - rss/tss
34  rsq
35
36
```

Results ::

```
> summary(boost.diff) ## sellingprice has a little higher influence than mmr.
                       var   rel.inf
sellingprice sellingprice 57.87155
mmr                   mmr 42.12845
> yhat.boost<- predict(boost.diff, newdata= data[-train,], n.trees=5000)
> diff.test<- data[-train,"difference"]
> mse <- mean((yhat.boost-diff.test)^2)
> mse
[1] 0.01444225
> rss<- mean((yhat.boost-diff.test)^2)
> tss <- mean(((data$difference-mean(diff.test))^2))
> rsq <- 1 - rss/tss
> rsq
[1] 0.7427869
```