STA521 Report

Team Omega

December 13, 2017

1. Introduction:

Summary of Problem:

We are given a dataset which provides information about auctions of paintings which were sold between years 1764 and 1780. The data tells us about different attributes of paintings, information about auction, information about the artist, information about the buyer and the price at which it was sold. There are total 59 columns (variables) in data including 'price' and 'log(price)'. Our task is to find out the relation between variables and the price so that we can predict the price of a paintings. We should also interpret the results as which are the most/least influential factor in determining the price of the painting.

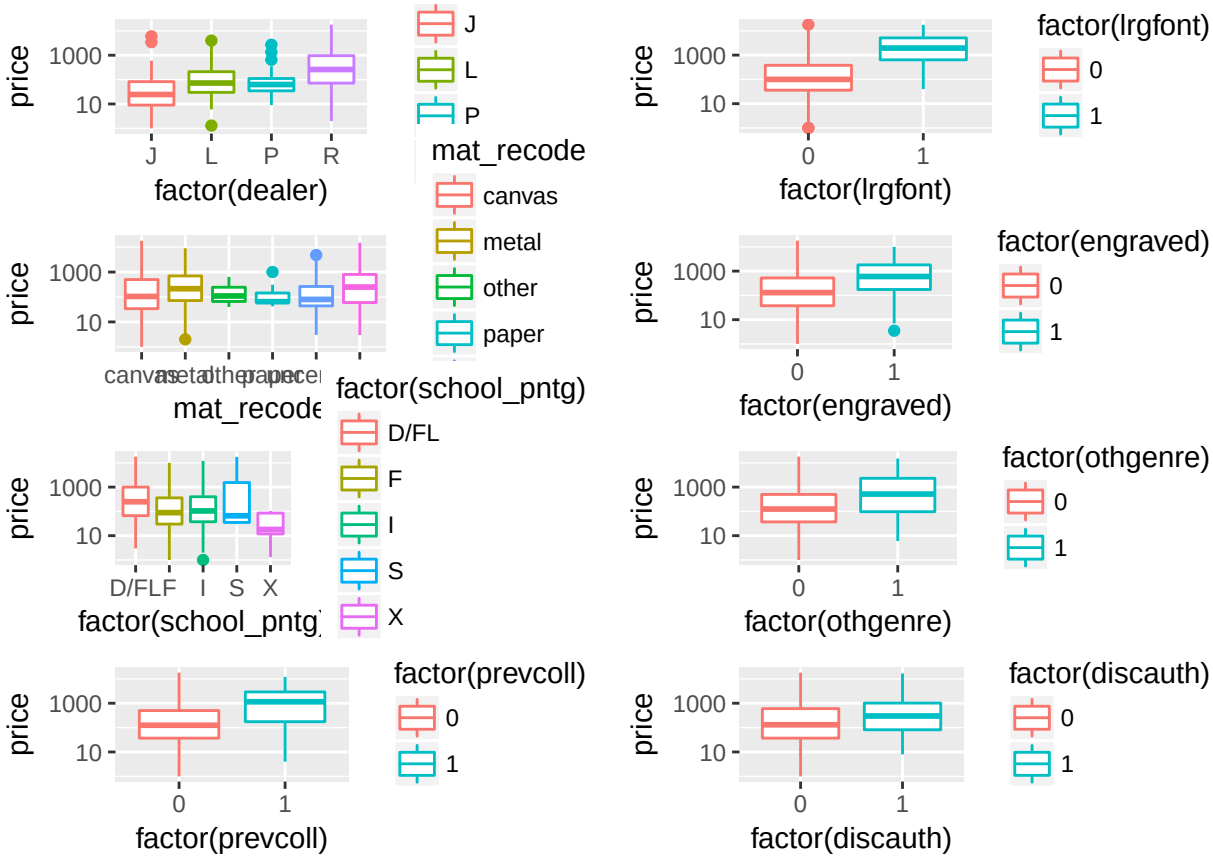
Objectives:

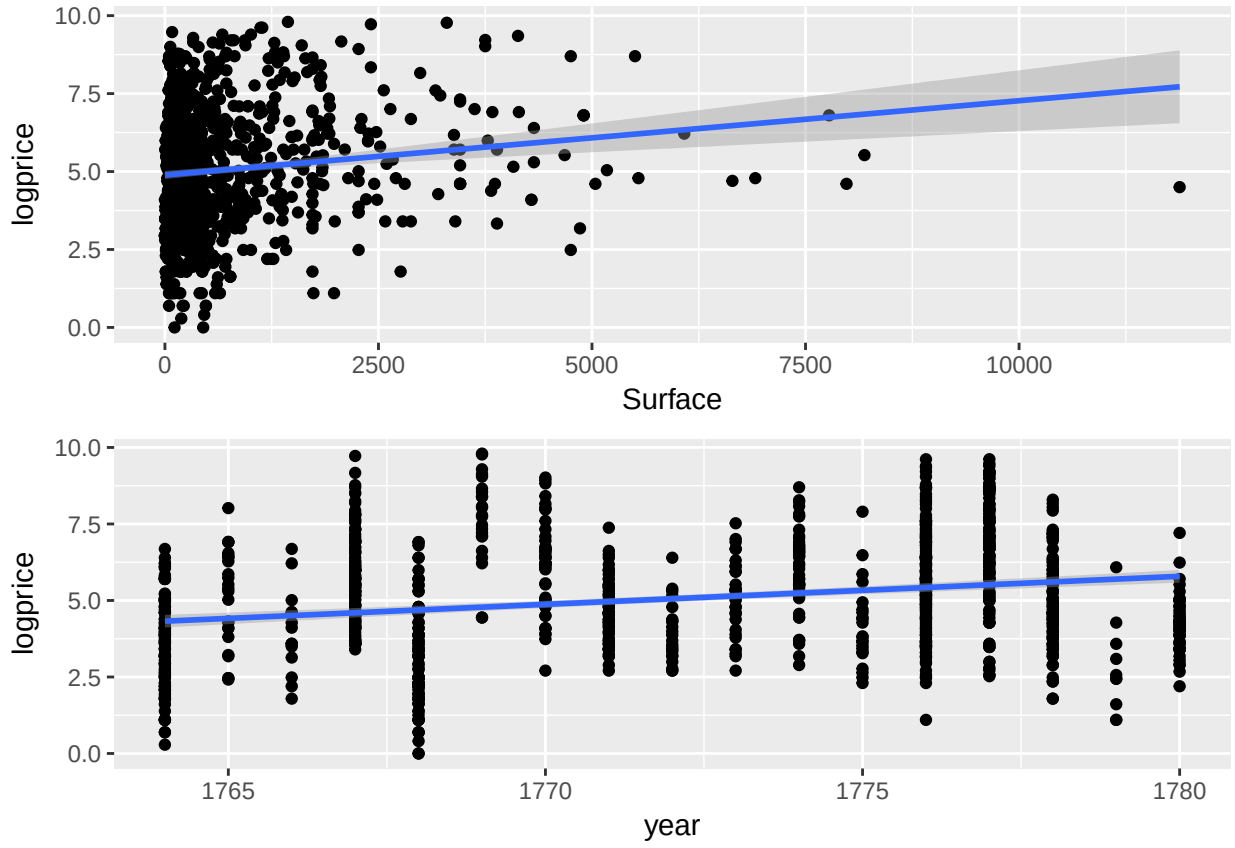
1) Exploratory data analysis
2) Create new variables : Adding to previous submission, we created some new variables based on the observationsa and included those variables in the final model.
3) Find out interaction
4) Impute missing values if neccessary
5) Create a best-fit model influential variables and interpret results
6) Test the model with test data and interpret the results

2. Exploratory Data Analysis

Some Graphs and analysis

The following is a group of boxplots that show the effect of different variables on price. Based on these boxplots, we observe that the mean of price differ between layers of categorical variables; the discrepancies are later on substantiated by hypothesis testing to be statistically significant. As a preliminary analysis, we decide to put them into our raw model before BIC analysis. It turns out that all but mat_recode are included in our final model. Even though mat_recode is not significant on its own, it proves to be a very interesting variable in interaction term.





New Variables that are added

1) dealer_author - Based on the groupby operations we observed that, if dealer is 'R' and 'origin_author' is 'D/FL' or 'S' mean price of painting is significantly higher than other ones. So we created a binary variable with value equal to 1 if the above condition is satified and 0, if otherwise

```
## # A tibble: 23 x 3
##    dealer origin_author mean_price
##     <chr>         <chr>      <dbl>
##  1      R             S  5253.6250
##  2      R          D/FL  1711.0676
##  3      L             G   888.0000
##  4      R             F   747.2678
##  5      R             G   746.8889
##  6      J             I   658.0455
##  7      R             I   452.2500
##  8      L          D/FL   364.5538
##  9      P          D/FL   270.5208
## 10      L             F   227.5833
## # ... with 13 more rows
```

2) dealer_school - Based on the groupby operations we observed that, if dealer is 'R' and 'school_pntg' is 'D/FL' or 'S' mean price of painting is significantly higher than other ones. So we created a binary variable with value equal to 1 if the above condition is satified and 0, if otherwise

```
## # A tibble: 18 x 3
```

```
##    dealer school_pntg mean_price
##     <chr>       <chr>      <dbl>
##  1      R           S 9539.50000
##  2      R        D/FL 1624.08485
##  3      R           F  748.13557
##  4      R           I  596.36607
##  5      J           I  433.97059
##  6      L        D/FL  359.74648
##  7      P        D/FL  261.62000
##  8      L           F  221.09677
##  9      J        D/FL  148.98387
## 10      L           I  119.64706
## 11      P           F   78.48148
## 12      P           S   66.00000
## 13      P           I   55.00000
## 14      R           X   48.41818
## 15      J           F   42.48163
## 16      L           S   32.50000
## 17      P           X   14.00000
## 18      L           X    1.30000
```

3) endbuyer_paired - Based on the groupby operations we observed that, if endbuyer is 'B' or 'C' and paired is 0, mean price of painting is significantly higher than other ones. So we created a binary variable with value equal to 1 if the above condition is satified and 0, if otherwise

```
## # A tibble: 12 x 3
##    endbuyer paired mean_price
##       <chr>  <int>      <dbl>
##  1        C      0 2620.56471
##  2        B      0 1642.84615
##  3        D      0 1021.85437
##  4        C      1  816.35895
##  5        D      1  592.06045
##  6        B      1  437.16667
##  7        E      0  430.34762
##  8        U      0  213.93590
##  9                0  192.44718
## 10        E      1  130.15385
## 11        U      1  117.69767
## 12                1   98.54627
```

4) endbuyer_history -Based on the groupby operations we observed that, if endbuyer is 'C' and 'history' is 1, mean price of painting is significantly higher than other ones. So we created a binary variable with value equal to 1 if the above condition is satified and 0, if otherwise

```
## # A tibble: 12 x 3
##    endbuyer history mean_price
##       <chr>   <int>      <dbl>
##  1        C       1  6800.0000
##  2        C       0  1909.7186
##  3        B       0  1462.3667
##  4        D       0   857.0979
##  5        B       1   733.0000
##  6        E       0   378.6443
##  7        U       1   344.0000
##  8        U       0   178.3667
```

```
##  9               0   169.4608
## 10        D     1    69.5000
## 11        E     1    58.3000
## 12              1    35.0000
```

Imputing missing values:

Missing values in Surface variable were imputed using 'mice' package. Code is shown below

```
suppressWarnings(library(mice))

###loading dataset
load("paintings_train.Rdata")
train <- paintings_train

temptrain <- mice(train,m=5, maxit=100 ,meth='pmm',seed=500)

completeData <- complete(temptrain, 1)

train$Surface <- completeData$Surface
```

Interactions

(1). Surface:mat_recode

Surface area and mat_recode (recoding of material) should have some interaction as different material is used fo paintings for different sizes of paintings. This is further supported as we observe significantly different slopes from layer to layer.

(2). school_pntg:Surface

School of painting and surface area of painting will definitely have some interaction as different style of paintings will use different figures and landscapes in them which will have diffeent sizes. For example, a portrait will usually have smaller surface area than a landscape which includes mountains and rivers.

(3). winningbiddertype:prevcoll

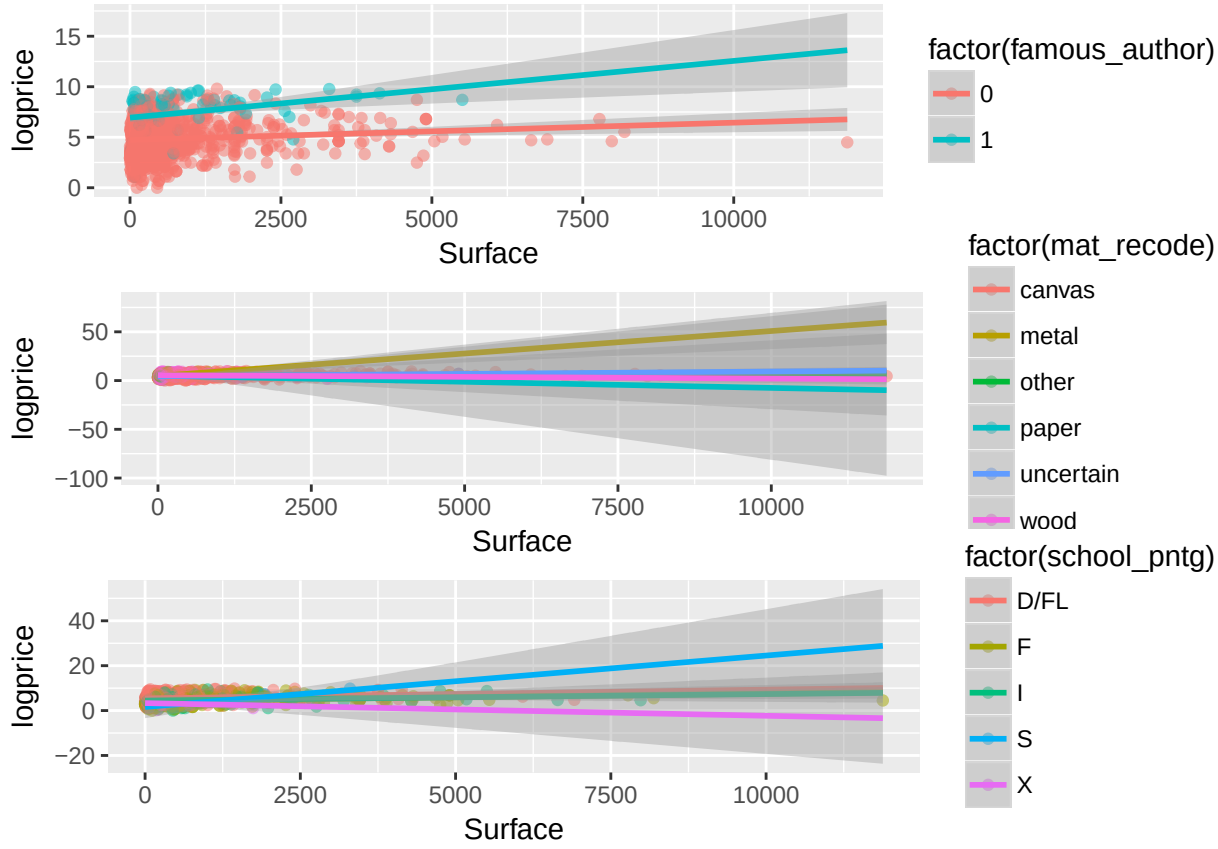
One of the motivating factors of purchasing a painting could be the name of previous owner of that painting. If bidders know that the previous owner of the painting was a well-known person, then they may be more willing to pay for a higher price. So we figure that 'winningbiddertype' and 'prevcoll' may have some interaction.

(4). famous_author and Surface We created anothe variable called 'famous_author' using following criteria: We calculated the mean price for each author using 'groupby' function.

```
## # A tibble: 423 x 2
##                                 authorstandard mean_price
##                                          <chr>      <dbl>
##  1                         Le Sueur, Eustache   8775.000
##  2                      Mieris I (F), Frans van   8100.000
##  3                  Velde I (W), Willem van de   6830.500
##  4 "Gell\u008ee (Lorrain), Claude; Miel, Jan"   5952.000
##  5          "Murillo, Bartolom\u008e Esteban"   5882.714
##  6   Both (J), Jan; Poelenberch, Cornelis van   5601.000
##  7              Berchem, Nicolaes Pieterszoon   4978.000
##  8                   Weenix (JB), Jan Baptist   4659.500
##  9                                Dou, Gerrit   4578.675
## 10                                Raoux, Jean   3906.250
```

```
## # ... with 413 more rows
```

We decided that we will call all those authors 'famous_authors' whose mean price is greater than 3000. Once we had 'famous'author' variable, we plotted the following graph to figure out that it is interacting with 'Surface' variable.



3. Discussion of preliminary model Part I: We formew=d a linear model in part 1, which we decided to improve. We started adding our new variables and a new interaction. After observing the results from leaderboard, we observed that the following linear model performs best in terms of RMSE.

With 3 newly created variables, RMSE increases a bit but coverage improve.

Different complex models that we tried

BART

We used Bayesian Additive Regression Trees as our 1st complex model to do predictions. We tried several predictors and figured out that BART with all the predictors gives us really good coverage (123) on testing data set but RMSE is in range of 1500. If we reduce some of the predictors with multiple factors in it, RMSE is improved (1441) but coverage goes down to 171. The code is as shown:

```
X = subset(newTrain, select = -c(price))
y = subset(newTrain, select = c(price))
#change price into numeric
y <- as.numeric(gsub(",","",newTrain$price))
y <- log(y)
X = subset(X, select = -c(other, pastorale, allegory, singlefig, peasant, fig_mention, artist_living_no

bart_machine = bartMachine(X,y, num_trees = 65, num_burn_in = 500, num_iterations_after_burn_in = 1000)

Xnew = subset(newTest, select =-c(other, pastorale, allegory, singlefig, peasant, fig_mention, artist_li
pred_int = calc_prediction_intervals(bart_machine, Xnew)
predictions1 = predict(bart_machine, Xnew)
y1 <- exp(pred_int)
y2 <- exp(predictions1)
predictions <- cbind(y2, y1)
colnames(predictions) <- c("fit","lwr","upr")
save(predictions, file="predict-test.Rdata")
summary(bart_machine)
```

Elastic Net

Our group also considers elastic net as one of the possible models. First, we create a grid of α and λ, and train our dataset using 5-fold cross validation. After obtaining the optimal α and λ on the grid, we proceed to make prediction with "glmnet" model using the optimal α and λ. The prediction result is encouraging, which managed to reach 1214 on the testing dataset. However, one of the main drawbacks of elastic net methods is that there is currently no consensus on how to obtain prediction interval from it. As a result, our group has to make do with an estimate, generated from running boostrap 1000 times, which still performs poorly by covering merely 29%. Nevertheless, despite its low coverage, we still believe that elastic net is one of our better performing models in prediction (if only coverage is not one of the major factors in evaluation).

```
tr_control=trainControl(method="cv",number = 5)
tunegrid=expand.grid(lambda=10^seq(-10,10,0.5),alpha=seq(0,1,0.005))
fml <- log(price) ~ dealer + year + school_pntg + diff_origin +
        artistliving + winningbiddertype + Surface + engraved +
        prevcoll + paired + finished + lrgfont + othgenre +
        discauth + winningbiddertype:prevcoll +
        Surface:mat_recode + famous_author:Surface
```

```
some_model=train(fml,tuneGrid=tunegrid,data=newTrain,
                 trControl=tr_control,method="glmnet")
some_model$bestTune
```

```
B=1000
#just training the model
tr_control=trainControl(method="none")
n=dim(newTrain)[1]
preds=matrix(0,nrow=B,ncol=dim(newTest)[1])
for (i in 1:B){
  bootsample=sample(1:n,size=n,replace=TRUE)
  trainset=newTrain[bootsample,]
  mod_temp=train(fml,tuneGrid=some_model$bestTune,data=trainset,
                 trControl=tr_control,method="glmnet")
  preds[i,]=predict(mod_temp,newTest)
  if (i %% 100 ==0){
    print(i)
  }
}
mod=train(fml,tuneGrid=some_model$bestTune,data=newTrain,
          trControl=tr_control,method="glmnet")
mean_b=predict(mod,newTest)
sd_b=apply(preds,2,sd)
lower_b=mean_b-qnorm(0.975)*sd_b
upper_b=mean_b+qnorm(0.975)*sd_b
Yhat=exp(cbind(mean_b,lower_b,upper_b))
colnames(Yhat)=c("fit","lwr","upr")
predictions_glmnet = as.data.frame(Yhat)
```

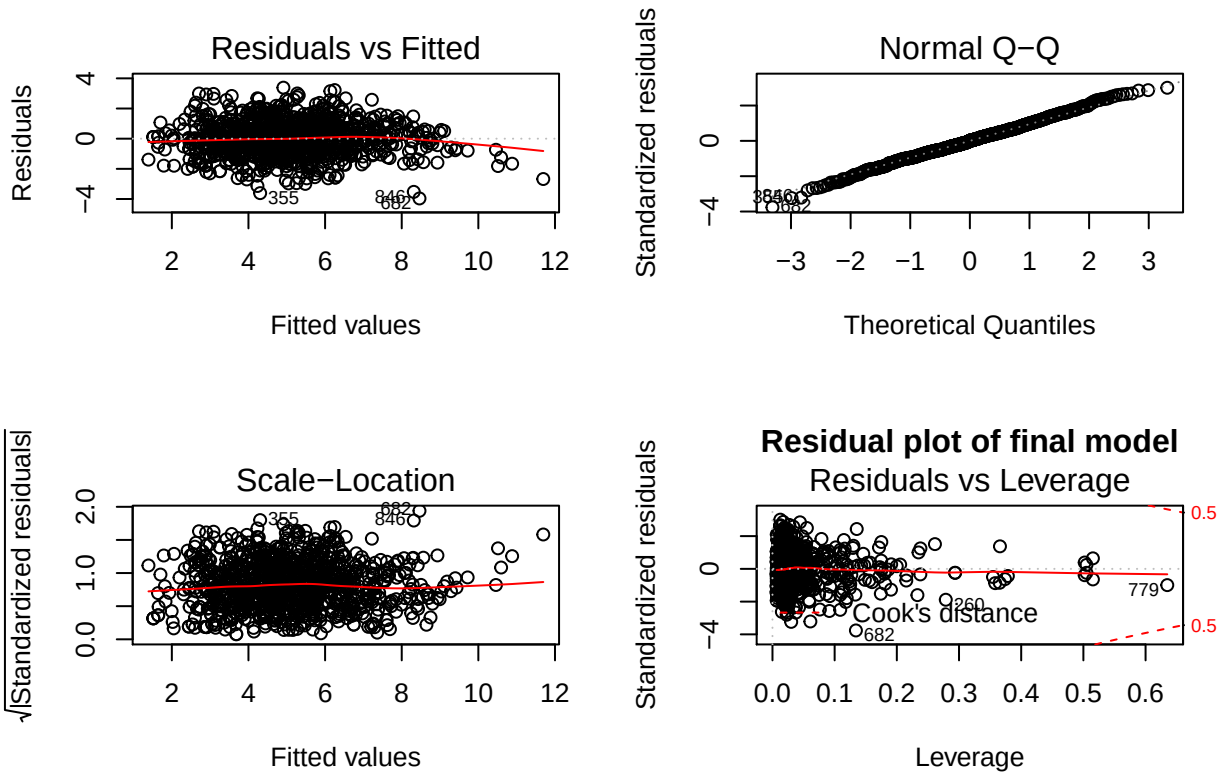
Conclusion

In conclusion, our group choose linear model over other complex model, not only because our linear model has better performance over our complex models (based on the training set, our linear model has lower RMSE, and high Coverage, while slightly sacrificing performance on Bias); but also its simple model interpretation as compared to other complex ones.

Despite the variety of models we have experimented on, our modified linear model still outperforms them in terms of RSME (1204), coverage(0.93) and bias (236.5). This final model differs from the previous linear model in part I by a new interaction variable that we discovered between famous_author and Surface. The rationale behind this new interaction is pretty intuitive, that the more well known the author of the painting is, the more valuable his or her works become. On top of that, the larger the painting is, it tends to be regarded as more valuable. As it turned out, the coefficient of this interaction variable is positive and has a very small p-value (2.29 * 10^-7). This indicated that the price of the painting is positively correlated with these two factors in addition to the ones we have discussed previously in part I.

```
## Warning: not plotting observations with leverage one:
##   694, 1037
```

```
## Warning: not plotting observations with leverage one:
##   694, 1037
```




```
##
## Call:
## lm(formula = log(price) ~ dealer + year + school_pntg + diff_origin +
##     artistliving + winningbiddertype + Surface + engraved + prevcoll +
##     paired + finished + lrgfont + othgenre + discauth + winningbiddertype:prevcoll +
##     endbuyer_history + endbuyer_paired + dealer_author + Surface:mat_recode +
##     famous_author:Surface, data = newTrain)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -3.9640 -0.7146  0.0032  0.7099  3.3819
##
## Coefficients: (6 not defined because of singularities)
##                                 Estimate Std. Error t value Pr(>|t|)
## (Intercept)                   -1.456e+02  1.542e+01  -9.443  < 2e-16 ***
## dealerL                        1.585e+00  1.498e-01  10.580  < 2e-16 ***
## dealerP                        4.132e-01  1.925e-01   2.146 0.032078 *
## dealerR                        1.635e+00  1.384e-01  11.815  < 2e-16 ***
## year                           8.376e-02  8.699e-03   9.629  < 2e-16 ***
## school_pntgF                  -5.331e-01  1.079e-01  -4.940 9.09e-07 ***
## school_pntgI                  -5.541e-01  1.275e-01  -4.346 1.53e-05 ***
## school_pntgOTHER              -1.185e+00  5.219e-01  -2.271 0.023331 *
## school_pntgX                  -9.054e-01  3.372e-01  -2.685 0.007370 **
## diff_origin                   -6.510e-01  9.480e-02  -6.867 1.13e-11 ***
## artistliving                   3.999e-01  1.233e-01   3.243 0.001222 **
## winningbiddertypeB             7.514e-01  3.759e-01   1.999 0.045862 *
```

```
## winningbiddertypeBC              1.030e+00  3.989e-01   2.581 0.009988 **
## winningbiddertypeC               5.746e-01  1.851e-01   3.104 0.001962 **
## winningbiddertypeD               7.789e-01  1.272e-01   6.124 1.29e-09 ***
## winningbiddertypeDB              1.324e+00  8.315e-01   1.592 0.111707
## winningbiddertypeDC              1.770e+00  2.103e-01   8.420  < 2e-16 ***
## winningbiddertypeDD              1.654e+00  4.875e-01   3.393 0.000717 ***
## winningbiddertypeE               3.824e-01  1.622e-01   2.357 0.018606 *
## winningbiddertypeEB             -2.201e-01  8.296e-01  -0.265 0.790830
## winningbiddertypeOTHER           1.865e+00  8.410e-01   2.217 0.026812 *
## winningbiddertypeEC              1.046e+00  2.748e-01   3.806 0.000149 ***
## winningbiddertypeU               4.905e-01  1.456e-01   3.369 0.000784 ***
## Surface                          3.342e-04  3.651e-05   9.154  < 2e-16 ***
## engraved                         5.832e-01  1.646e-01   3.544 0.000411 ***
## prevcoll                         7.617e-01  3.749e-01   2.031 0.042464 *
## paired                          -1.453e-01  8.777e-02  -1.656 0.098115 .
## finished                         6.191e-01  1.040e-01   5.955 3.56e-09 ***
## lrgfont                          7.778e-01  1.310e-01   5.935 4.00e-09 ***
## othgenre                         4.147e-01  1.222e-01   3.394 0.000715 ***
## discauth                         5.221e-01  1.511e-01   3.456 0.000571 ***
## endbuyer_history                 2.504e+00  1.158e+00   2.163 0.030787 *
## endbuyer_paired                  5.098e-01  1.731e-01   2.946 0.003294 **
## dealer_author                    2.953e-01  1.263e-01   2.337 0.019618 *
## winningbiddertypeB:prevcoll             NA         NA      NA       NA
## winningbiddertypeBC:prevcoll            NA         NA      NA       NA
## winningbiddertypeC:prevcoll     -1.376e-01  5.649e-01  -0.244 0.807658
## winningbiddertypeD:prevcoll     -4.084e-02  4.427e-01  -0.092 0.926516
## winningbiddertypeDB:prevcoll            NA         NA      NA       NA
## winningbiddertypeDC:prevcoll    -9.938e-01  5.449e-01  -1.824 0.068457 .
## winningbiddertypeDD:prevcoll            NA         NA      NA       NA
## winningbiddertypeE:prevcoll      3.473e-01  6.119e-01   0.568 0.570466
## winningbiddertypeEB:prevcoll            NA         NA      NA       NA
## winningbiddertypeOTHER:prevcoll         NA         NA      NA       NA
## winningbiddertypeEC:prevcoll    -1.420e-01  7.944e-01  -0.179 0.858147
## winningbiddertypeU:prevcoll     -2.035e+00  1.204e+00  -1.691 0.091222 .
## Surface:mat_recodemetal          1.492e-03  5.608e-04   2.661 0.007911 **
## Surface:mat_recodeother          6.168e-04  1.364e-03   0.452 0.651162
## Surface:mat_recodepaper          2.587e-04  1.533e-03   0.169 0.866001
## Surface:mat_recodeuncertain      1.190e-05  1.296e-04   0.092 0.926865
## Surface:mat_recodewood           3.461e-05  1.321e-04   0.262 0.793412
## Surface:famous_author            4.817e-04  1.004e-04   4.797 1.85e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 1.133 on 1030 degrees of freedom
## Multiple R-squared:  0.6576, Adjusted R-squared:  0.6426
## F-statistic: 43.96 on 45 and 1030 DF,  p-value: < 2.2e-16
```

If we were given more time, our group is planning to do a complex Ensemble of models using the package "caretEnsemble". Unfortunately, we have learnt this package at a very late stage, so we do not have enough time to experiment with it. Given more time, however, we should be able to use its intrinsic function to create "caretList" and "caretStack", which can allow use to better tune model, for example by assigning different weights to different models in "caretList".