The Price is Right? - Using Quality Measures to Determine the Price of Diamonds

Yi Tang Chen, Chi Ying Lin, Gargi Sundaram, Pradeep Kumar Thiagu

December 21, 2017

Abstract

Regression techniques were employed in this project to find the best effective way to predict the price of diamonds. Features of diamonds such as clarity, cut quality, color, weight, dimensions, etc., are the main predictors for evaluating the diamond's value. Various of popular regression methods have been attempted to achieve the best prediction and 5 of them including KNN, extreme gradient boosting, boosted tree, elastic net, and random forest were ideal approaches for this purpose. Within these methods, random forest provided the best prediction. Therefore, random forest is recommended for the diamond's price evaluation.

Introduction

Diamonds have been used in jewelry since the 15th century. In the 1930s, diamonds became the most popular gemstone for engagement rings when the De Beers corporation revived the American diamond market with the famous slogan "A diamond is forever." Since then, the diamond ring has become a culturally mandatory symbol of commitment between an engaged couple.

Today, buying a diamond is an investment that takes time, consideration, and especially information. Diamonds are priced based the four 'C's: Cut, Color, Clarity, and Carat. There is a standardized baseline price for loose diamonds based on size, color, and clarity that can be found on the weekly updated Rapaport Diamond Report. However, companies will often throw away the diamond's certificate if the quality is lower than expected and sell it at the price of a higher quality stone.

Most people find the best price by "shopping around", comparing prices of diamonds with similar qualities and hoping for the best outcome. But what if there were a way for the consumer to ensure that they were not being overcharged without consulting a certificate? Using the Diamonds' dataset, we can create a regression model that predicts a diamond's price based on objective size and quality measures. Using this as a benchmark, consumers can make sure they are getting the best deal for the best quality and avoid being scammed by the diamond industry.

Materials and Methods

Data

This diamond dataset was accessed from the Kaggle data repository, a site commonly used to access data sets and upload analyses. It contains information on the specs and selling price of over 50,000 diamonds.

The `diamonds` dataset has 53940 observations and 11 attributes, which are described below:

- `X` - Observation number
- `carat` - Carat weight (1 carat = 200 mg)
- `cut` - Cut quality (5 levels: Fair, Good, Very Good, Premium, Ideal)
- `color` - Color (7 levels: D (best) through J(worst))
- `clarity` - Clarity (8 levels, listed in order from worst to best: I1, SI1, SI2, VS2, VS1, VVS2, VVS1, IF)

- `depth` - Height of a diamond, measured from the culet to the table, divided by its average girdle diameter
- `table` - Width of the top of the diamond relative to its widest point
- `price` - Price in US dollars
- `x` - Length in mm
- `y` - Width in mm
- `z` - Depth in mm

As shown above, physical attributes and the price for each diamond provided in this dataset can allow us to predict the price of a diamond given its respective characteristics by fitting a regression model.

Variable Selection

Three variable selection methods are adopted in this project: the LASSO, GBM and stepwise selection based on AIC.

- LASSO

```
xx = model.matrix(price ~., trndata)[,-1]
yy = as.matrix(trndata$price)

fit_lasso_cv =
  cv.glmnet(
  x = xx,
  y = yy,
  nfolds = 5,
  alpha = 1
  )

plot(fit_lasso_cv)

c = coef(fit_lasso_cv, s = 'lambda.1se', exact = TRUE)
inds = which(c != 0)
variables1 = row.names(c)[inds]
variables1 = variables1[!variables1 %in% "(Intercept)"]
variables1 = selectedVar(variables1)
```

- GBM

```
set.seed(myseed)
cv_5 = trainControl(method = "cv", number = 5)
gbm_grid = expand.grid(
  interaction.depth = c(1, 2),
  n.trees = c(500, 1000),
  shrinkage = c(0.001, 0.01, 0.1),
  n.minobsinnode = 10
  )

diamond_gbm = train(
  price ~ .,
  data = trndata,
  method = "gbm",
  trControl = cv_5,
  verbose = FALSE,
  tuneGrid = gbm_grid
```

```
  )

diamond_imp = summary(diamond_gbm)
# consider variables which have total relative influence larger than 95
Nvariable = min(which(cumsum(diamond_imp$rel.inf) > 95))
variables2 = as.character(diamond_imp$var[1:Nvariable])
variables2 = selectedVar(variables2)
```

- Stepwise AIC

```
FS_step = step(
  object = glm(price ~ ., data = trndata),
  scope = ~ .,
  direction = "both",
  trace = FALSE,
  k = log(nrow(trndata))
  )
summary(FS_step)
slVarDF = as.data.frame(FS_step$coefficients)
variables3 = rownames(slVarDF)[2:nrow(slVarDF)]
variables3 = selectedVar(variables3)
```

Performance of four models with three variable selection methods are listed in Table 1. Noted that the test dataset for calculating the test rmse was splited from the original dataset by the same seed and the same portion as in the next section.

Table 1: Performance of four models with three variable selection methods

	All variables	LASSO	GBM	AIC
Random Forest by randomForest package	909.88	944.82	1017.70	1058.08
Stochastic Gradient Boosting	794.49	796.84	1055.37	788.59
eXtreme Gradient Boosting	682.21	679.92	1023.99	684.91
Support Vector Machine	1401.99	1505.16	996.95	1328.33

Based on the result above, we can see the variable selection is not effective for improving the model performance. The dataset has only 9 predictors, so the model complexity is not really an issue in this case. Therefore, in the following analysis, all 9 original predictors will be used.

Models

Before applying any models, we split our data into a testing and training set.

```
diamonds = diamonds[, -1]
set.seed(1)
diamond_index = createDataPartition(diamonds$price, p = 0.5, list = FALSE)
diamond_trn = diamonds[diamond_index, ]
diamond_tst = diamonds[-diamond_index, ]
```

We applied 5 approaches to our data including unscaled knn with pca, extreme gradient boosting, elastic net, boosted tree, and random forest. Then, we determined the best model that can most accurately predict a diamond's price based on physical attributes by comparing their respective root-mean-square-error(RMSE), which is calculated as followed.

$$\text{RMSE}(\hat{f}, \text{Data}) = \sqrt{\frac{1}{n}\sum_{i=1}^{n}\left(y_i - \hat{f}(\mathbf{x_i})\right)^2}$$

And, the models set-up are the following:

- Unscaled KNN with PCA

```
set.seed(1)
unscaled_knn_mod = train(price ~ . , data = diamond_trn, method = "knn",
  preProcess = "pca",
  trControl = trainControl(method = "cv", number = 5),
  tuneGrid = expand.grid(k = seq(1,11,2))
)
```

- Extreme Gradient Boosting

```
grid.xgb = expand.grid(
  eta = c(0.01, 0.1, 0.3),
  max_depth = c(1, 2, 3),
  colsample_bytree = seq(0.4, 0.8, 0.2),
  subsample = c(0.5, 0.7, 1),
  nrounds = c(150, 200),
  min_child_weight = 1,
  gamma = 0
  )
set.seed(1)
xgb_mod = train(price ~ ., data = diamond_trn, method = "xgbTree",
  tuneGrid = grid.xgb,
  trControl = trainControl(method = "cv", number = 5)
  )
```

- Elastic Net

```
set.seed(1)
elastic_mod = train(price ~ . ^ 2, data = diamond_trn, method = "glmnet",
  trControl = trainControl(method = "cv", number = 5),
  tuneLength = 10
)
```

- Boosted Tree

```
gbm_grid = expand.grid(interaction.depth = c(1, 2),
                       n.trees = c(500, 1000, 1500),
                       shrinkage = c(0.001, 0.01, 0.1),
                       n.minobsinnode = 10)
set.seed(1)
gbm_mod = train(price ~ ., data = diamond_trn, method = "gbm",
  trControl = trainControl(method = "cv", number = 5),
  tuneGrid = gbm_grid,
  verbose = FALSE
)
```

- Random Forest

```
rf_grid = expand.grid(mtry = 1:(ncol(diamond)-1))
set.seed(1)
rf_mod = train(price ~ ., data = diamond_trn, method = "rf",
```

```
  trControl = trainControl(method = "oob"),
  tuneGrid = rf_grid
)
```

Results

Again, the metric we used to assess the prediction accuracy of the model is root-mean-square-error(RMSE). The lower the RMSE is, the better a model predicting the price of a diamond. Therefore, we will choose random forest as our final model since it results the lowest RMSE as shown in Table 2.

Table 2: Comparisons of RMSE

Model	Resampled RMSE	Test RMSE
Unscaled knn with PCA	914.4455	888.9361
extreme gradient boosting	686.8950	682.2144
elastic net	721.6444	790.4597
Boosted Trees	779.8359	757.6824
Random Forest	669.6021	665.5588

As we can see, our final model, random forest, has the lowest resampled RMSE, which is 669.6020645.

Discussion

Based on our result demonstrated in Table 2, random forest is the best model among the five that can most accurately predict the price of a diamond due to its lowest resampled RMSE. Random forest is a non-parametric method since it does not assume parameters. And, it is also a discriminant and a non-linear method because it automatically accounts for non-linearity when modeling unobserved response conditioning on predictors. In contrast, elastic net is a parametric method, which assumes normal distribution, and it is a linear method because it does not automatically measure nonlinearity; yet, it is a discriminant method as well.

Regarding RMSE, the resampled RMSE value for random forest is 669.6020645, meaning that the margin of error for the predicted price of a diamond is 669.6020645. For instance, if we predicted the price of a diamond to be 1000 US dollars given its attributes, then the actual price of this diamond is in the range of 330.3979355 to 1669.6020645.

Conclusion

The purpose of this report is to make prediction on the price of a diamond given its characteristics in order to provide customers a guideline when shopping a diamond, so they will not be overcharged. Five methods were attempted and tested by comparing respective RMSE measures to determine which model is the best on predicting the price of a diamond. As the result, random forest is the model that has the lowest prediction error with resampled RMSE being 669.6020645.

Appendix

Diamond price samples were collected from Kaggle through the following link: - Diamonds - Analyze diamonds by their cut, color, clarity, price, and other attributes