Project 1: Wine Review and Recommendation

Alex Kwan, NETID = akwan2

11/25/2019

Project Summary

Starting with a 14 columns data set using WineEnthusiast information, I developed several new features. I compared Linear Regression with LASSO (using glmnet) and kNN regression models (using caret) in predicting the points rated by WineEnthusiast on a scale of 1-100. The kNN regression model performed slightly better. I employed a similarity score to assess which wineries to recommend to a potential client based on the updated feature set. My recommendation interestinly enough included the Trump winery.

Project Description

The goals for this project are to (1) develop two distinct regression models to predict points awarded to various wines based on the data provided and any features that may be extracted from them and (2) provide a recommendation for potential wineries of interest to a client based on the wine features they prefer.

Data Processing

The data was processed in the wrangl_proj1.ipynb file. Knowing that several regression models work better with numeric data than categorical, I explored ways to create numeric features. From the description field, I was able to create a description_len feature which provides the number of characters in the description text. Using the Vader package (https://github.com/cjhutto/vaderSentiment (https://github.com/cjhutto /vaderSentiment)), I extracted a new feature, sentiment_score, which "is a lexicon and rule-based sentiment analysis tool that is specifically attuned to sentiments expressed in social media." I extracted a boolean fruity feature from the description which registers a 1 if the description includes the word fruit or any specific common fruit as identified by Wikipedia. From the title field, I extracted the year of the wine. I took the country, province, region_1 and region_2 information and used the Google Geocoding API (https://developers.google.com/maps/documentation /geocoding/start (https://developers.google.com/maps/documentation/geocoding/start)) to generate the numeric features lat and lon, which are the corresponding GPS coordinates (approximate in some cases) associated with the geographic textual information. I also paired down the variety and taster_name fields to account for their long tails and grouped the less frequent entries (the bottom 5%) into a larger 'other' group. I dropped some fields that I felt were duplicates or not pertinent like twitter handle and designation. My updated data set had 11 features and is found in the data_new.csv file.

Interesting Description Statistics

I won't go into exhaustive detail but a few descriptive statistics to note:

- There are over 20 different tasters. Some are prodigious like the top taster, Roger Voss, who accounts for ~20% of all review.
- There are some records with missing data but in the main there data quality is good with about 90% of rows with no missing data (once I assigned records with no taster to the 'other' category).
- The range of points is from 80 to 100.
- There are well over 700 varieties of wines. I chose to forgo the variety as a categorical feature used in my regression models.

```r
library(glmnet)
library(caret)
```

```r
data <- read.csv("data_new.csv", header=T)
data$fruity <- as.integer(as.logical(data$fruity))
head(data)
```

```
##   points price      lat         lon description_len sentiment_score fruity
## 1     87    NA 37.75100   14.993435             172          0.1531      1
## 2     87    15 41.50128   -5.512293             227          0.6486      0
## 3     87    14 44.94255 -122.933762             186         -0.1280      1
## 4     87    13 42.22087  -86.369469             199          0.3400      1
## 5     87    65 44.94255 -122.933762             249          0.8176      0
## 6     87    15 42.81698   -1.641765             261          0.1655      1
##   year              winery       varietyB       taster_nameB
## 1 2013             Nicosia    White Blend    Kerin Oâ\200\231Keefe
## 2 2011 Quinta dos Avidagos Portuguese Red         Roger Voss
## 3 2013           Rainstorm     Pinot Gris       Paul Gregutt
## 4 2013          St. Julian       Riesling Alexander Peartree
## 5 2012        Sweet Cheeks     Pinot Noir       Paul Gregutt
## 6 2011              Tandem          other  Michael Schachner
```

Method 1: Linear Regression with Lasso

```
l_data <- subset(data, select = -c(winery, varietyB))
l_data <- l_data[complete.cases(l_data), ]
```

```
ind<-sample(2,nrow(l_data),replace=T,prob = c(0.7,0.3))
ltrain<-l_data[ind==1,]
ltest<-l_data[ind==2,]

#convert to dummy variables
ltrain$taster_nameB <- model.matrix( ~ taster_nameB - 1, data=ltrain )
ltest$taster_nameB <- model.matrix( ~ taster_nameB - 1, data=ltest )

var_train<-as.matrix(ltrain[,2:9])
score_train<-as.matrix(ltrain$points)

var_test<-as.matrix(ltest[,2:9])
score_test<-as.matrix(ltest$points)
```

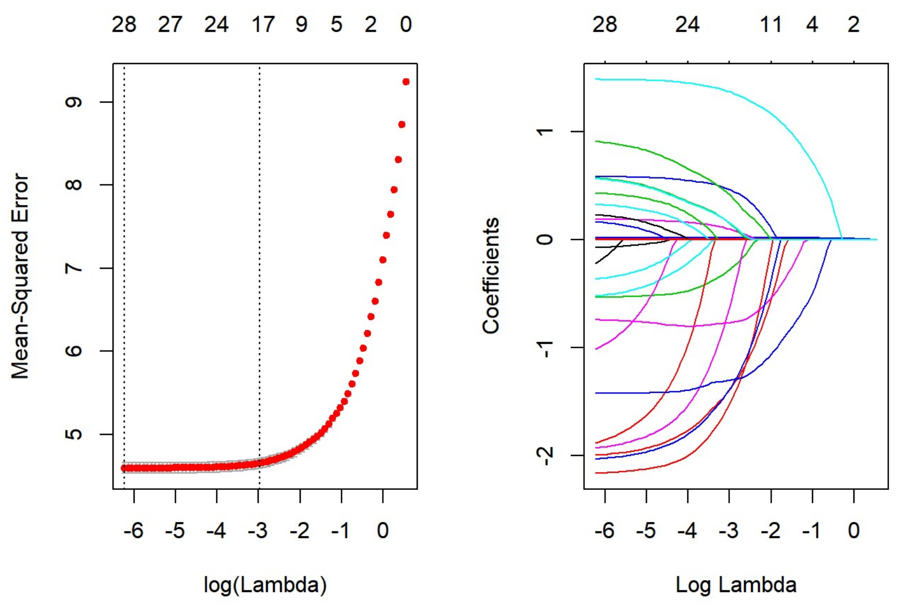
```
set.seed(3)
fit_lasso = cv.glmnet(var_train, score_train, nfolds = 10)
coef(fit_lasso, s = "lambda.min")
```

```
## 30 x 1 sparse Matrix of class "dgCMatrix"
##                                                          1
## (Intercept)                                   8.174937e+01
## price                                         1.885530e-02
## lat                                           4.327981e-03
## lon                                           3.674149e-03
## description_len                               2.155433e-02
## sentiment_score                               1.487394e+00
## fruity                                        1.923163e-01
## year                                          8.808229e-05
## taster_nameB.taster_nameBAlexander Peartree   -2.166482e+00
## taster_nameB.taster_nameBAnna Lee C. Iijima   -5.348889e-01
## taster_nameB.taster_nameBAnne KrebiehlÂ MW     5.879426e-01
## taster_nameB.taster_nameBAnne KrebiehlÃ‚Â MW   3.296130e-01
## taster_nameB.taster_nameBCarrie Dykes         -1.931109e+00
## taster_nameB.taster_nameBChristina Pickard    -2.203485e-01
## taster_nameB.taster_nameBFiona Adams          -1.882854e+00
## taster_nameB.taster_nameBJeff Jenssen          4.340096e-01
## taster_nameB.taster_nameBJim Gordon            1.684328e-01
## taster_nameB.taster_nameBJoe Czerwinski       -5.225589e-01
## taster_nameB.taster_nameBKerin OÃ¢â‚¬â„¢Keefe -1.012197e+00
## taster_nameB.taster_nameBKerin Oâ\200\231Keefe      -7.245166e-02
## taster_nameB.taster_nameBLauren Buzzeo        -1.999187e+00
## taster_nameB.taster_nameBMatt Kettmann         5.732175e-01
## taster_nameB.taster_nameBMichael Schachner    -1.426275e+00
## taster_nameB.taster_nameBMike DeSimone        -3.641324e-01
## taster_nameB.taster_nameBother                -7.428367e-01
## taster_nameB.taster_nameBPaul Gregutt          2.333221e-01
## taster_nameB.taster_nameBRoger Voss            .
## taster_nameB.taster_nameBSean P. Sullivan      9.111931e-01
## taster_nameB.taster_nameBSusan Kostrzewa      -2.033327e+00
## taster_nameB.taster_nameBVirginie Boone        5.665132e-01
```

```
coef(fit_lasso, s = "lambda.1se")
```

```
## 30 x 1 sparse Matrix of class "dgCMatrix"
##                                                            1
## (Intercept)                                    82.0625001328
## price                                           0.0182388664
## lat                                             0.0049283013
## lon                                             0.0006443021
## description_len                                 0.0210730375
## sentiment_score                                 1.3698123426
## fruity                                          0.0967751493
## year                                            .
## taster_nameB.taster_nameBAlexander Peartree   -1.5045334536
## taster_nameB.taster_nameBAnna Lee C. Iijima   -0.2688369031
## taster_nameB.taster_nameBAnne KrebiehlÂ MW      0.4605708962
## taster_nameB.taster_nameBAnne KrebiehlÃ,Â MW   .
## taster_nameB.taster_nameBCarrie Dykes         -0.5929072939
## taster_nameB.taster_nameBChristina Pickard     .
## taster_nameB.taster_nameBFiona Adams           .
## taster_nameB.taster_nameBJeff Jenssen          .
## taster_nameB.taster_nameBJim Gordon            .
## taster_nameB.taster_nameBJoe Czerwinski        .
## taster_nameB.taster_nameBKerin OÃ¢â,¬â„¢Keefe  .
## taster_nameB.taster_nameBKerin Oâ\200\231Keefe       .
## taster_nameB.taster_nameBLauren Buzzeo        -1.3794550858
## taster_nameB.taster_nameBMatt Kettmann         0.1456911474
## taster_nameB.taster_nameBMichael Schachner    -1.3013040599
## taster_nameB.taster_nameBMike DeSimone         .
## taster_nameB.taster_nameBother                -0.7672049643
## taster_nameB.taster_nameBPaul Gregutt          .
## taster_nameB.taster_nameBRoger Voss            .
## taster_nameB.taster_nameBSean P. Sullivan      0.4071903342
## taster_nameB.taster_nameBSusan Kostrzewa      -1.3726197738
## taster_nameB.taster_nameBVirginie Boone        0.1502679819
```

```
par(mfrow = c(1, 2))
plot(fit_lasso)
plot(fit_lasso$glmnet.fit, "lambda")
```



```
test_pred = predict(fit_lasso, s = "lambda.1se", newx=var_test)
RMSE(test_pred, score_test)
```

```
## [1] 2.136783
```

A few things to note:

- The lat features bears greater impact on the model than the lon feature. This makes sense because latitude impacts weather more than

longitude and weather likely impacts wine quality.
- The sentiment_score feature bears significant impact on the model. It appears that the VADER package is useful.
- The varying results for dummy variables for the different tasters indicates that some tasters are more biased than others.
- The year feature is not a useful one in building a model.
- The description_len bears as much impact on the model as the price and more than the latitude.

Method 2: kNN

For kNN, we need all numeric covariates. I will remove the categorical covariates: winery, varietyB and taster_nameB. I will also remove year (as the LASSO results show it is not meaningful). Also for kNN, to make it easy, I will drop all NAs.

```
library(tidyverse)
library(class)
```

```
set.seed(3)
k_data <- subset(data, select = -c(winery, varietyB, taster_nameB, year))
k_data <- k_data[complete.cases(k_data), ]

ind<-sample(2,nrow(k_data),replace=T,prob = c(0.7,0.3))
ktrain<-k_data[ind==1,]
ktest<-k_data[ind==2,]
```

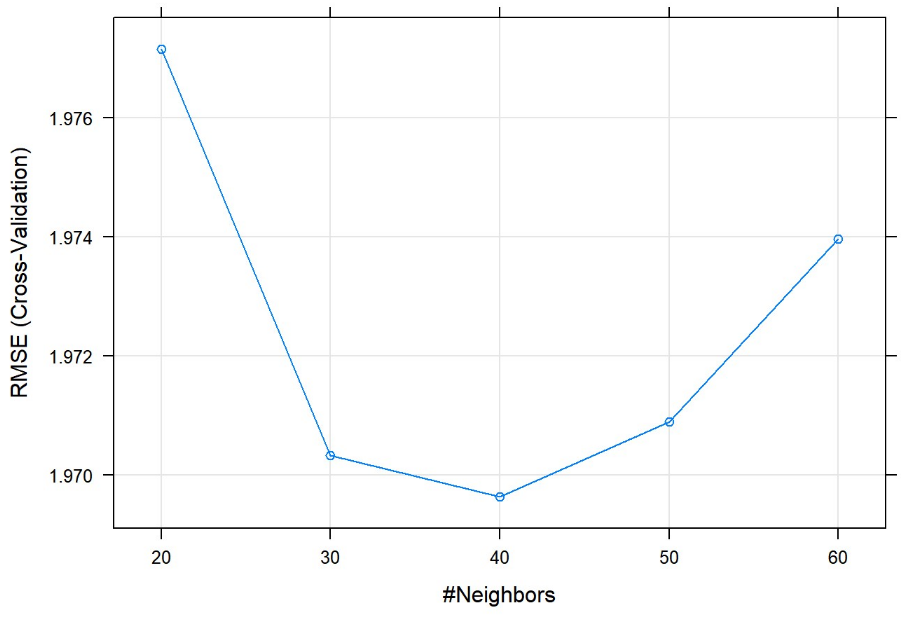
We'll use the following arguments in the function train():

- trControl, to set up 5-fold cross validation
- preProcess, to normalize the data
- tuneGrid, to specify the possible k values to evaluate

(below takes about 10 minutes)

```
model <- train(points ~ ., data = ktrain, method = "knn",
  trControl = trainControl("cv", number = 5),
  preProcess = c("center","scale"),
  tuneGrid = expand.grid(k = c(20, 30, 40, 50, 60))
)
```

```
# Plot model error RMSE vs different values of k
plot(model)
```



```
# Compute the prediction error RMSE
kpred <- model %>% predict(ktest[,2:7])
RMSE(kpred, ktest$points)
```

```
## [1] 1.950543
```

The best k value appears to be about 40 nearest neighbors. The RMSE value for the kNN model is lower than the linear regression with LASSO. The kNN regression model is likely the one.

Wine Recommendation

```
# We want pinot noir, with a price less than 20 dollars, and has a fruity taste.
target_wines <- data %>% filter(price < 20) %>% filter(varietyB == 'Pinot Noir') %>% filter(fruity == 1)

head(target_wines)
```

```
##   points price       lat        lon description_len sentiment_score fruity
## 1     85    13 -33.72899  -70.77780             193          0.3182      1
## 2     85    15  42.72382  -76.92972             415          0.7876      1
## 3     85    15 -34.67614  -71.09732             278          0.5346      1
## 4     87    17  38.25571 -122.32980             148          0.4215      1
## 5     85    13  43.80413 -120.55420             125          0.2960      1
## 6     85    12  37.59999   14.01536             285          0.6369      1
##   year         winery   varietyB      taster_nameB
## 1 2011  Tres Palacios Pinot Noir Michael Schachner
## 2 2006         Wagner Pinot Noir             other
## 3 2008       Cono Sur Pinot Noir Michael Schachner
## 4 2010 Michael Pozzan Pinot Noir             other
## 5 2014         Rascal Pinot Noir      Paul Gregutt
## 6 2008  Feudo Montoni Pinot Noir             other
```

Let's take the average of the target wines for points, price, lat, lon, description_len, and sentiment_score. The average will represent the type of wine the client might like. Then we will calculate the distance (similarity score) from each wine to this average representation and provide the five most represented wineries with similarity scores.

```
target_wines_vars <- subset(target_wines, select = -c(winery, varietyB, taster_nameB, fruity, year))
rep_wine_vals <- colSums(target_wines_vars) / nrow(target_wines_vars)
```

```
t_df <- subset(data, select = -c(varietyB, taster_nameB, fruity, year))
t_df <- t_df[complete.cases(t_df), ]
t_df_var <- subset(t_df, select = -c(winery))
```

```
normalize <- function(x) {
    return ((x - min(x)) / (max(x) - min(x)))
}

t_df_varn <- as.data.frame(lapply(t_df_var, normalize))

colmins <- apply(t_df_var,2,min)
colmaxs <- apply(t_df_var,2,max)

rep_wine_valn <- (rep_wine_vals - colmins) /  (colmaxs - colmins)
```

Now get distance for each wine to the normalized representative wine

```
euc_dist <- function(x1) {
  x1n <- as.numeric(x1)
  x2n <- rep_wine_valn
  edist <- sqrt(sum((x1n - x2n) ^ 2))
  return(edist)
}

t_df_dist <- apply(t_df_varn, 1, euc_dist)
t_df$distance <- t_df_dist
t_df_sort <- t_df[order(t_df$distance),]
```

We can then take the top 500 wines and look at that five most common wineries they come from to provide the recommendation.

```
t_df_sort_top <- t_df_sort[1:500,]
library(dplyr)
top_wineries <- t_df_sort_top %>% count(winery)
top_wineries_sorted <- top_wineries[with(top_wineries, order(-n)), ]
head(top_wineries_sorted, 5)
```

```
## # A tibble: 5 x 2
##   winery                       n
##   <fct>                    <int>
## 1 Veramar                     13
## 2 The Williamsburg Winery     12
## 3 Trump                        9
## 4 Tarara                       8
## 5 Page Springs                 7
```