Wine Review and Recommendation

Hao Tang

Contents

I. Project Description and Summary

a) Project Summary

Wine Enthusiast is a leading multichannel marketer to growing wine products. At 1988, Wine Enthusiast founded Wine Enthusiast Magazine which brings consumers vital information on the world of wine. Their wine trend, rating and review data to diversified wine is well analyzed in multiple data science forum. This project is to explore the regression models to the review points from some famous and professional wine tasters. I am interested in the points prediction based on the producing location, price, winery information. Here I will use random forest and boosting model in the project. Simulataneously, I'll investigate Pinor Noir with specific contraints so that customer can find the most approriate choice to this wine with more convinience in the variable market.

b) Dataset

The dataset being analyzed in this project is zackthoutt's `winemag-data-130k-v2` updated on Nov 24th, 2017. It has **129971** reviews with detailed comment from the taster and their twitter handle. Here is the **14** columns in the dataset: **id**(This column name is added by me for efficient tracking), **country**, **description**, **designation**(the grape vineyard), **points**, **price**, **province**, **regoin_1**, **region_2**(region_1 and region_2 are the detailed infomation to the producin location, the latter column can be treated as extension), **taster_name**, **taster_twitter_handle**, **title**(closely related to the wine variety and vintage), **variety**, **winery**.

II. Data Processing

After loading the dataset, **id** is assigned to the first column. There are quite a few of NA value in the **price** column so I replace the 'NA' with '0' but this replacement is just for indication and the zero value shouldn't be part of the model processing. Meanwhile, some of the review is lack of the taster's name, thus the consistent 'others' will be assigned to **taster_name** column accordingly. In the following model processing, I excluded the following column: **id**, **country**, **description**, **designation**, **region_1**, **region_2**, **taster_twitter_handle** and **title**. Apparently **id** will not help in our procedure. **description** shall be useful in data mining but it's not a efficient predictor to the response **points** in our models and it has quite a few of exactly duplicated value. **designation** is with `37465` empty elements and it has `37980` different values so it's not a good choice to our models. **country**, **province**, **region_1** and **region_2** are highly geographically related to each other. Considering 21247 empty elements in **region_1** and 79460 empty elements in **region_2**, our choice can be focus on **country** and **province**. **province** is more detailed than **country** especial to the major wine country, such the U.S., which has multiple producing province nationwide. Thus, only **province** is applied in these four columns in our model analysis. **taster_twitter_handle** is not informative and **title** is too specific to the wine itself so they're not included. **winery** has `16757` values so it won't be included but it will be applied in our Pinot Noir analysis.

Once we have the targeted columns, we can start to browse their value distribution. The quantity of values to the interesting column are display in Table 1. There are too many levels to most of these columns thus the model based on all of these levels will be inefficient and very unstable. After more investegation, I found small set of the values of each column have composed the great portion of the whole reviews based on the elements quantity. Table 2 shows the top values in each column and the elements percentage with these values. Please notice there are `~20%` elements in **taster_name** is missing value. The following model analysis will be based on the interesting columns with these values' factor form.

Table 1: Values Quantity of interesting Columns

	taster_name	province	variety
Qty of Columns	19	426	708

Table 2: Values Percentage of All Reviews

	taster_name_top10	province_top25	variety_top25
Percentage of All Reviews	0.7350101	0.7964777	0.7711951

III. Summaries of Data

Figure 1. Wine Percentage on All Countries in Final Dataset

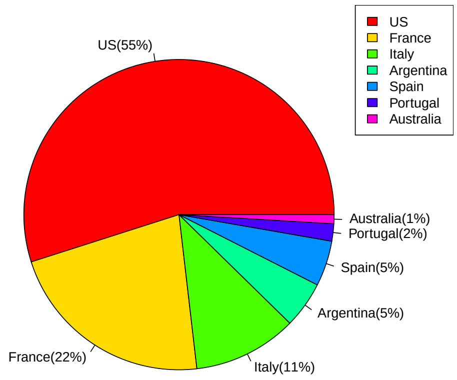
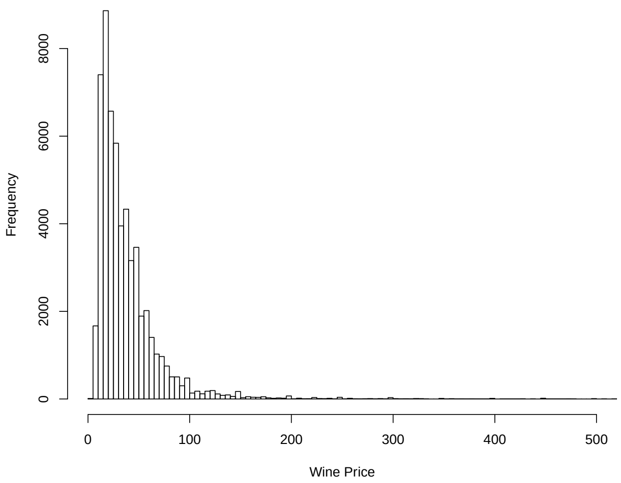


Figure 2. Histogram for the Wine Price in Final Dataset



Let's investigate our processed data before starting the regression model analysis. Remember we have replace the NA value with `0` in **price** column, our final dataset will exclude the zero price review accordingly since these reviews will become noise to our data. Also, the final model will be generated only by the following predictors: **price**, **province**, **taster_name**, **vareity**.The **points** is response. As stated in last part, the criteria includes top 10 tasters, top 25 wine produsing province and top 25 wine variety. Our regression model will be based on all `57247` reviews in this dataset. Comparing to the average point(`88.45`) in the original dataset, the final dataset's average points (`88.89`) is very similar and their quantile barely have difference. This is good because we want the model constructed on the processed dataset can somehow closely related to the orginal dataset. In the final dataset, all the reviews are from `7` countries' wine (Figure 1) and all these countries are also the top 10 produsing countries in the orginal dataset. That being said, wine industry is pretty concentrated in specific area in the world. And Figure 2 shows that the price distribution. It is similar to that of orginal datast too, less than `1%` of all the wine is with the price higher than `200`.

In the four predictors selected in the model analysis, three of them are character variables with 10 to 25 levels, which means regular linear regression, polynomial regression, logistic regression or kernel regression wouldn't be a good fit to the model candidate here. The tree models should be more efficient to our final dataset after transforming all character variables to factor variable. For another, I also want to lower the variance while construct the models. So random forest and boosting could be the reasonable choice here. One may also brings up ridge/lasso regression that can also deal with character variable by matrix transformation. I will also present the result from the lasso regression after the random forest and boosting model for comparison.

IV. Model Analysis

Random Forest

Random Forest is efficient model to multiple type of dataset and it should be a reasonable choice in our mixed variable in the project. This model apply bagging or bootstrap aggregating idea and construct many decision trees (forest) from bagging. When we make prediction on a datapoint/observation, the regression result will be generated by the trees' result averaging. The most important advantage of random forest is the correlation reduction design in the model fitting process. In each node when we want to grow the tree, only some of the variables (**mtry** as below analysis) will be selected as candidates randomly. We then chooce the best split value in these candidates as the criteria for the next level's tree. This process will be processed recursively in each level of the tree until we hit the maximum depth (**max.depth** as below anlaysis) or minimum node size of assigned observations. For another, we can always increase the depth of each tree to decrease the bias which will improve the final prediction accuracy just like ordinary decision tree. But with random forest, it's relatively unlikely to overfit the model like the ordinary dicision tree model.

In this project, I choose different values of **mtry** and **max.depth** for comparison and we will see the prediction accuracy evolution with different combination of the parameters. One of the important feature of random forest model is its out-of-bag (OOB) error estimate. This OOB error estimate is pretty close to N-fold cross-validaion performed during model construction. I will also compare the OOB prediction mean squred error(MSE) to the generated testing set.

The final dataset from pre-processing part above is seperated into training set and testing set(80% and 20%). The random forest models with differetn paramter combinations are constructed on the training set. We can see the OOB prediction MSE from the random forest model is pretty close to the Testing set prediction MSE in Figure 3. This matches the OOB error generation mechanism and it apply to all different parameters' combination. Meanwhile, the prediction error/MSE decreases as the max depth increase because the bias will decrease as well. However, higher tree depth means more intensive computation. Also, as the max.depth increase, the prediction accuracy improvement will getting lower and lower, and finally we will see there is barely any improvement even though we dramatically increase the max.depth of the tree. This trend can be observed in Figure 3 too. We can see the MSE difference from all lines in max.depth = 4 is larger than the difference when max.depth is 8. Considering efficiency and performance, I choose max.depth = 6 in final random forest model. About the mtry, we can see plot with mtry = 1 has high MSE comparing to mtry = 2 and mtry = 3. With more candidate variable, it's more likely to fit a tree with high performance but it will also brings up the correlation between trees and the model variance. So we need to balance these effects and choose an optimal mtry to the final random forest model. From Figure 3, the plot with mtry = 2 and mtry = 3 are almost identical, both of two values seems to be reasonable choice. In fact, the default number of variables in most random forest model is **p/3** where **p** is the number of all predictors. In our dataset, p is equal to 4 and **p/3** ~ **2**. mtry = 2 will be the parameters applied to the random forest model. Other parameter will follow the default setting in the random forest (ranger) package.

Figure 3. Random Forest MSE with Tuning Paramters

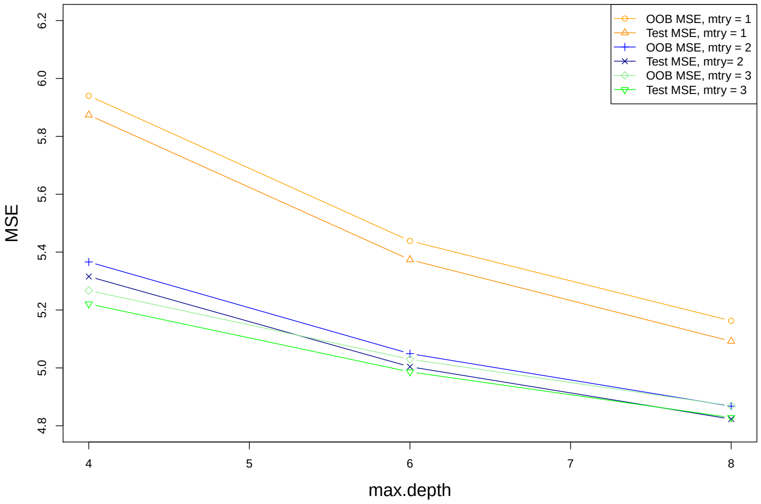


Table 3 is the final model parameters and its test prediction MSE and OOB MSE. In fact, the prediction performance can be futhur improved by select specific level in some columns or less levels in these columns. Increasing the tree depth can provide better performance as well. The goal to this project is to explore certain effective models to analyze our dataset. We will come back a bit later to compare this with next model.

Boosting

Another powerful model that will work effectively with our mixed dataset is Boosting. In the process of model construction, many weak learners/trees will be generated sequentially to form an additive model. Each tree will learn the result according to the correctness of the prevous tree. Finally each tree will provide weighted vote to generate the final result for any input data points. Different from random forest or bagging method, boosting will use all the training data for each tree but the data points' importance can be different.

Generally we will tune three parameters in boosting method: **a)** the number of trees B, **b)** the shrinkage paramter λ and **c)** the max.depth of each tree d. In order to compare the regression models in this project, I will set the similar max.depth parameters to each tree in boosting model. Together with the shrinkage parameter, I will present the model MSE according to different combinations of parameters in the following section. As for the number of tree, its effect is closely related to the shrinkage parameter so I didn't include it into our analysis. Other than these parameter, all factors will follow the default setting in the `gbm` package in R.

Table 3: Final Random Forest Model

	Package	mtry	max.depth	num.trees	Test MSE	OOB MSE
Final RF Model	ranger	2	6	500	5	5.05

Table 4: Final Boostingt Model

	Package	shrinkage	interaction.depth (max.depth)	n.trees(num.trees)	Test MSE	CV MSE
Final Boosting Model	gbm	0.6	6	200	4.66	4.71

Figure 4 display the MSE from gbm function crocess validation and testing set prediction. The training set and testing set I used here is same as their counterparts in Random Forest section. In random forest, we have OOB prediction error that's similar to cross validation error. Here in boosting, gbm function apply cross validation when it constructs the boosting models. So We can conveniently extract the CV MSE from the model. The CV MSE will be very close to the testing set prediction MSE. From Figure 4, we can see the MSE do not always do down when the max.depth increase, the trend also depends on the shrinkage/learning rate, λ. For another, when the learning rate gets smaller, it's likely to generate lower MSE in certain max.depth. This makes sense because smaller learning rate can motivate the algorithm to look for the optimal point slowly but effectively. The algorithm probably will not skip the optimal value for the large step. As we further lower the λ and choose appropriate max.depth, we may get even lower MSE but it will dramatically the numbers of trees before we achieve the optimal point. Considering balance between performance and efficiency, I choose max.depth = 6 and shrinkage = 0.6 for the final boosting model. Table 4 show the parameter setting to the final boosting model.

Figure 4. Boosting MSE with Tuning Paramters

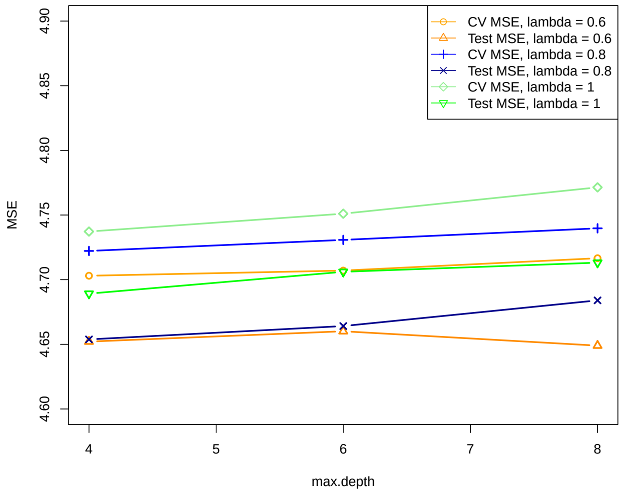


Figure 5. MSE to CV and Training Prediction

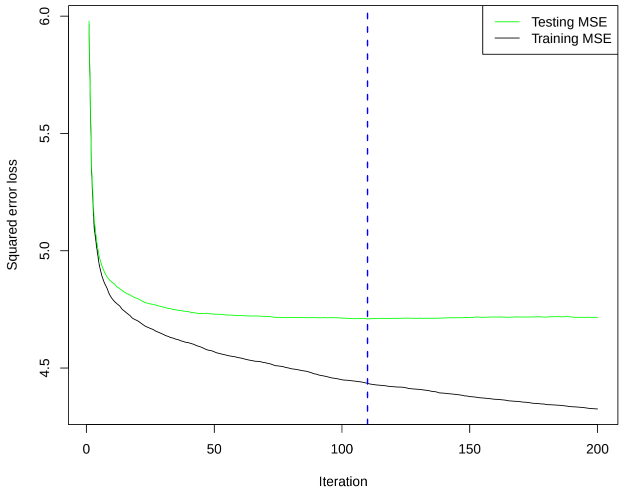


Figure 6. Variable Importance in Boosting Model

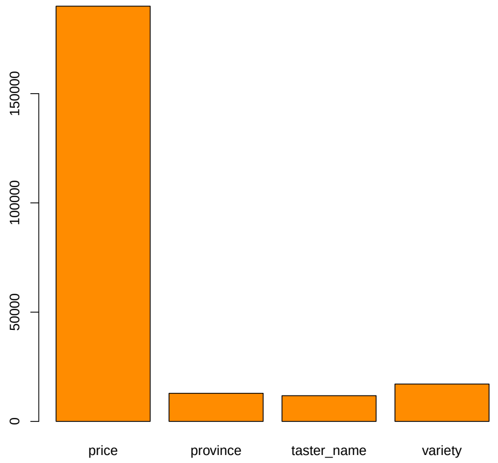
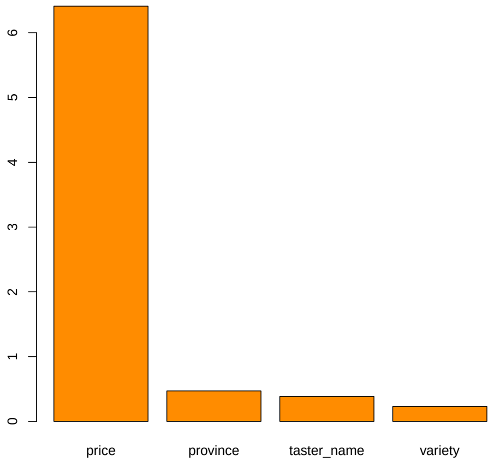


Figure 7. Variable Importance in RF Model

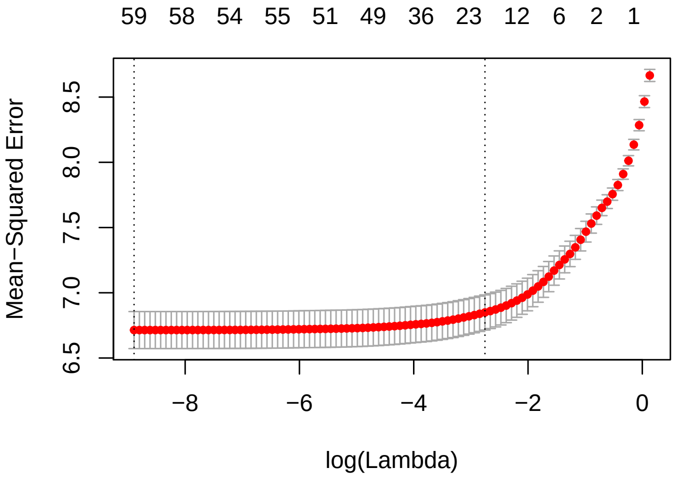


In the final boosting model, we actually choose the tree from 110 iteration for testing prediction. Its value is x-coordinate of the vertical blue dash line in Figure 5. More over, we can see the training MSE will continously decrese as iteration increase but the CV error will reach a minimal value and increase again after the minimal value. With the same depth in each tree, Boosting model perform better than Random Forest model on the MSE according to Table 3 and Table 4. Now let's check their variable importance comparison in Figure 6 and Figure 7. The variable importance calculation are different in two models however we can still indentify the **price** is apparently more important than any other variables in both models.

Comparison to Lasso Model

Lasso Regression can also work with mixed variable dataset but we need to transform the dataset into one-hot matrix first. Figure 8 is the plot for MSE vs. logarithmic λ (A tuning paramter to omit/minimize predictors in our final effective model). The lasso regression testing MSE is 6.3147952 with minimal λ in the function. You can also find this value on the intersection between the left grey dashed line and the red dot line. One of the advanced feature to lasso regression is it can subset the coefficients to a get a simple model. However, with the same training set here, the lasso MSE is apparently higher than the MSE of Random Forest and Boosting model.

Figrue 8. MSE vs. log(lambda) in Lasso Regression



V. Winerires Recommending

If you are looking for a pinot noir with reasonable price, this review dataset should be the great start since pinot noir is the most popular wine in all its reviews (There are 13272 review about Pinot Noir). Of course, we don't want to read a catalog with the thousand of reviews to make the decision. Let's transform our data to meet your requirement.

Figure 9. Winery Comparison on Fruity Pinot Noir

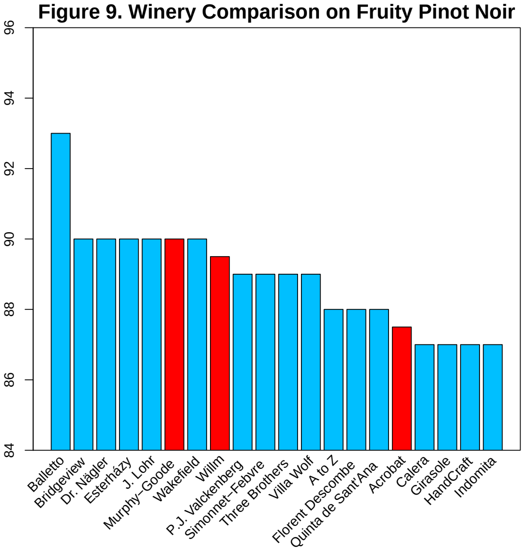
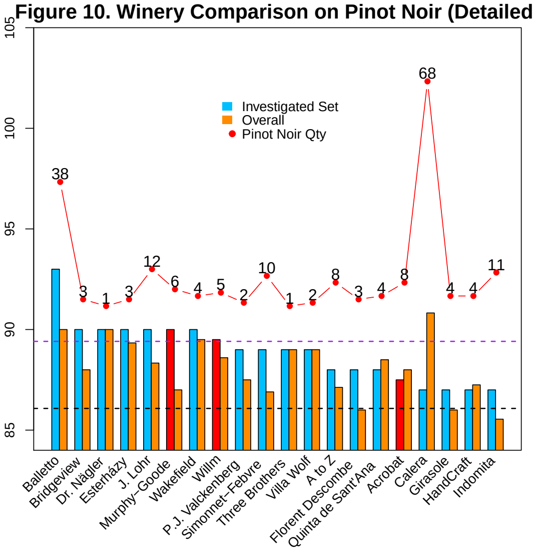


Figure 10. Winery Comparison on Pinot Noir (Detailed)



I subset this review dataset so only the review with less than 20-dollar price and fruity comment will be considered in the folloiwng section and investigated set. Figure 9 shows the top 20 wineries with highest average points in our subset. Since the mean score to all the Pinot Noir is `86.0786517`, it seems like it can't be wrong to pick five wineries from these top Pinot Noir wineries, doesn't it? Let's go a bit deeper into our subsetting data. I found most of the wineries only have one review about their Pinot Noir which means we can't be confident to our estimate based on such a small dataset. Of course, we do have some wineries with more than one Pinot Noir and these Pinot Noir are all with great score. e.g. Murphy-Goode, Willm, Acrobat, their bars are highlighted in Figure 9 and Figure 10 with red color. They may be good candidates to us. Generally we need more information to their overall Pinot Noir quantity score. Except for the Pinot Noir information in our sebsetting dataset, I also added the Pinot Noir score and Quantity to all the wineries based on the orginal dataset. We can see the detail from Figure 10. The blue and red bars are from our subsetting dataset wineries with top 20 scores. The orange bars are the overall Pinot Noir score to these wineries. The red curve on top display the Pinot Noir quantity produced by these wineries. The black dashed line and purple dashed line is added as reference. They are the average scores to our subsetting dataset and all Pinot Noir records. Below are two criteria that may be helpful to your decision.

1) Winery experience on Pinot Noir. According to the overall Pinot Noir product to all these wineries, the top five wineries are `Balletto`, `J. Lohr`, `Simonnet-Febvre`, `Calera`, and `A to Z` or `Acrobat`.

2) Winery scores based on subsetting dataset and overall Pinot Noir products review. The top five wineries are `Balletto`, `Dr. Nägler`, `Esterházy`, `Wakefield`, `Calera`. I didn't recommand `Three Brothers` and `Villa Wolf` just because their Pinot Noir sample is small. If you like Anna Lee C. Iijima's recommandation, you can definitely try them.

You may have noticed the we have overlap in these two criteria. `Balletto` and `Calera` show up in both my criteria. `Balletto`'s score are perfect based on our customer's requirement, so as it's overall Pinot Noir score. `Calera` are not that good in customer's requirement but their overall score is unbeatable. If we are very confident to our taster's judgement, these two wineries should be the ideal choice to you. The only thing we may want to pay attention is both wineries' product are mostly judged by specific tasters, Virginie Boone and Matt Kettmann. They happened to be the tasters who graded the wine with highest average score (Table 5). There may be some relationship between taster's grading style and wine's score.

Table 5: Taster's Average Score in all reviews

	taster_name	points	taster_name	points	taster_name	points
A	Matt Kettmann	91.03121	Mike DeSimone	89.00000	Alexander Peartree	87.00000
B	Virginie Boone	90.15389	Others	88.63359	Christina Pickard	87.00000
C	Jim Gordon	90.10357	Anna Lee C. Iijima	88.20398	Susan Kostrzewa	86.90909
D	Roger Voss	89.86538	Joe Czerwinski	88.07862	Lauren Buzzeo	86.50820
E	Anne Krebiehl MW	89.85882	Jeff Jenssen	87.86667	Michael Schachner	85.97137
F	Paul Gregutt	89.52738	Sean P. Sullivan	87.58929	NA	NA

Before we close this section, I would like to recommand `Bridgeview` to you if you would like to look for buried treasure. All their Pinot Noir are graded by strict taster. This is why their overall score is not remarkable. However, the top score of their Pinot Noir is from the most strict taster Michael Schachner who is strict to all kinds of wine! If there is no serious mistake, Bridgeview's top scored Pinot Noir, the one in our subsetting dataset, definitely worth a try.

VI. Conclusion

In this project, we investigate the wine review dataset from Wine Enthusiast on the wine score and the relationship between score and other variables in the dataset. By constructing two ensemble models, I estimate the performance of these regression model. Simultaneously, the parameter tuning procedure is presented in IV part to both regression models. Generally their MSE performance are much better than ridge regression, which is from standard linear regression. Between the two models, boosting did slightly better than random forest. I also compare the variable importance from these models. The price viable play relatively important role in the models while predicting the points. As data scientist to this project, I also provide the most reasonable criteria for choosing the best Pinot Noir wineries with my familiarity to the dataset. Based on my discoveries, I strongly recommand the oustanding wine and its winery to our customer according what the data tell. Let's keep going.