Jaeyoung Kim

Chungsang Tom Lam

ECON 9000

May 2, 2020

ECON 9000 Machine Learning and Data Scrapping (Spring2020) Final Exam

Purpose

The goal of this project is to predict the star rating in a review app, Yelp.

Prediction Model Selection

To predict the star rating, I use two supervised learning models: Linear regression and Random Forest.

(1) Why supervised model?

Throughout the semester, we learned different models of prediction including supervised models such as linear regression (OLS), Logit regression, decision tree, random forest, and unsupervised models such as support vector machine (SVM) and k-mean clustering.

First of all, I choose OLS as my main model and compare it with Random Forest model. The reason why I choose a supervised models is because I wanted to verify certain features of the store will affect the rating, rather than exploring data set initially. Furthermore, SVM is not appropriate when the data is huge. Given that the short project period and the huge data size, I concluded that it is not feasible to use unsupervised model. In general, random forests and decision are usually much faster to compute than SVMs. Lastly, SVM classifies the target by finding the line with the maximum amount of distance from groups, but it is hard to get the distance between these groups because our explanatory variables of the data sets are categorical variables.

(2) Why Linear Regression and Random Forest?

I hypothesize that treating star rating as continuous variable will give better prediction than classification of categorical dependent variable, which made me choose linear regression. Also, for comparison, the benefit of choosing these two models: linear regression and Random Forest is that we can compare the models' prediction performance when the target variable is continuous or categorical variables in two different models. In linear regression, my main model, I treat the dependent variable (i.e. star rating) as continuous variable and regress it on multiple categorical variables. But in Random Forest, I treat the dependent variable as a categorical variable (e.g., the classes include 1.0, 1.5, …, 4.5) and evaluate the classification performance.

I did not use logit regression because I was not interested in the probability of each star rating, and it requires 8 separate models for 8 classes within the star rating, which is harder to interpret when predicting one value for star rating. For the readers who are interested in the probabilities, logit regression is recommended because it solves the problem of negative probability of dummy dependent variable. Also, I don't use decision tree because using decision tree with many attributes

can cause overfitting problem due to too many layers of decision nodes. We can use random forest to solve this overfitting problem. Random forest does multiple decision trees separately and combine them.

Data cleaning

I use attributes of the business as main predictor of star rating. Thus, I parsed JSON file and made each attribute as categorical variable (True, False, and missing). However, there were too many missing values in the categorical variables. To account for the problem, I gave them a value 1, 0, and 0.5 to True, False, and missing values, respectively. I could not parse some attributes with nested JSON values, and it is a limitation of this project (e.g., 'BusinessParking', 'Ambience', 'GoodForMeal'). Also, I made another variable called 'stars_cat' which is a categorical variable of star rating. This categorical variable of star rating is used as a target variable of the random forests.

The remaining explanatory variables include

(1) ByAppointmentOnly
(2) BusinessAcceptsBitcoin
(3) BusinessAcceptsCreditCards
(4) RestaurantsDelivery
(5) RestaurantsGoodForGroups
(6) HasTV
(7) BikeParking
(8) OutdoorSeating
(9) RestaurantsTakeOut
(10) RestaurantsTableService
(11) Caters
(12) RestaurantsReservations
(13) GoodForKids.

And the target variables include

(1) Stars: star rating
(2) Stars_cat: categorical variable of star rating.

While there are 183,073 observations in the original data set, to accommodate the computational power of my computer, I limited the dataset into 5,000 observation. With more observations, my python kept crashing.

Prediction and Comparison Results

I used linear regression and random forest models for prediction and KFold validation model for validation. The attached file 'runme.py' trains the machine and predict the star rating using linear regression and random forest. Two models are compared in terms of R-squared. Accuracy scores and confusion matrices are computed for random forest as complementary metrics to compare it with other classification models in the future. The prediction uses the dataset 'business_no_stars.json' that has been cleaned as specified above. Appendix A shows the following results in command prompt.

(1) Linear Regression

Observation #	0	1	2	3	4	5	6	7	8	9
Predicted star rating	3.52	3.53	3.59	3.52	4.30	3.52	3.66	3.41	3.69	3.53
Rounded to the nearest possible star	3.5	3.5	3.5	3.5	4.5	3.5	3.5	3.5	3.5	3.5

(2) Random Forest Classifier

Observation #	0	1	2	3	4	5	6	7	8	9
Predicted star rating*	0	0	0	0	1	0	0	0	0	0

*There seems to be errors in coding, which produced incorrect prediction here.

(3) Comparison of two models

	R-squared from each KFold Split					
	Split 1	Split 1	Split 1	Split 1	Split 1	Mean
Linear Regression	0.0418	0.0519	0.0396	0.0297	0.0522	0.0431
Random Forest	-0.1549	-0.1806	-0.1341	-0.2078	-0.1615	-0.1678

	Accuracy Scores from each KFold Split					
	Split 1	Split 1	Split 1	Split 1	Split 1	Mean
Random Forest	0.856	0.863	0.857	0.846	0.856	0.8556

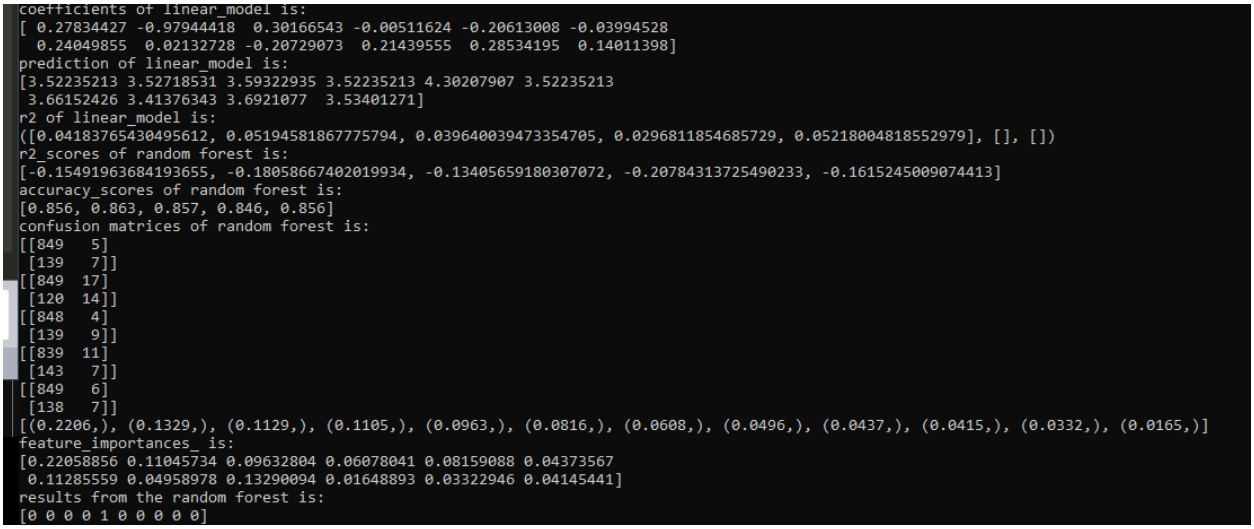
Confusion Matrices of Random Forest

[[1413 22]
[208 24]]
[[1410 19]
[209 29]]
[[1401 12]
[240 13]]

Conclusion

From the above results, we can observe that all values of R-squared in each KFold split and average R-squared are higher for linear regression model. It would be tempting to conclude that linear regression model is better to predict star rating. However, note that the R-squared from Random Forest model are negative, and the predicted star rating is out of range of the star rating in the training/test data set. It can be interpreted in two ways. First, the model uses too many explanatory variables, which led to overfitting problem. Second, there may have been some error in the code. The further correction for code and investigation could not be done due to the time limit. Future study should clarify the validity of this analysis by rechecking the data and codes.

Appendix A.



```
coefficients of linear_model is:
[ 0.27834427 -0.97944418  0.30166543 -0.00511624 -0.20613008 -0.03994528
  0.24049855  0.02132728 -0.20729073  0.21439555  0.28534195  0.14011398]
prediction of linear_model is:
[3.52235213 3.52718531 3.59322935 3.52235213 4.30207907 3.52235213
 3.66152426 3.41376343 3.6921077  3.53401271]
r2 of linear_model is:
([0.04183765430495612, 0.05194581867775794, 0.039640039473354705, 0.0296811854685729, 0.05218004818552979], [], [])
r2_scores of random forest is:
[-0.15491963684193655, -0.18058667402019934, -0.13405659180307072, -0.20784313725490233, -0.1615245009074413]
accuracy_scores of random forest is:
[0.856, 0.863, 0.857, 0.846, 0.856]
confusion matrices of random forest is:
[[849   5]
 [139   7]]
[[849  17]
 [120  14]]
[[848   4]
 [139   9]]
[[839  11]
 [143   7]]
[[849   6]
 [138   7]]
[(0.2206,), (0.1329,), (0.1129,), (0.1105,), (0.0963,), (0.0816,), (0.0608,), (0.0496,), (0.0437,), (0.0415,), (0.0332,), (0.0165,)]
feature_importances_ is:
[0.22058856 0.11045734 0.09632804 0.06078041 0.08159088 0.04373567
 0.11285559 0.04958978 0.13290094 0.01648893 0.03322946 0.04145441]
results from the random forest is:
[0 0 0 0 1 0 0 0 0 0]
```