MD REVIEW DATA ANALYSIS

Project2

Author

Jenny Huang

AI Launch Lab

2023.4.29

Contents

1 Abstract

In this project, the data set is information containing 'provider name (provider-name__lnk)', 'provider detail specialty (provider-details__specialty)', 'grades (sr-only)', 'number of ratings (star-rating__reviews)', 'office locations (location-info__office-loc)', 'location distance (location-info__distance)', 'feats (feats-of-strength__feat-title)' and 'rating website (star-rating href)' from patients and medical centers. The raw dataset has some empty rows and noisy messages. Firstly, I use **pandas** and **numpy** to clean the dataset within 7 Excel sheets, remove empty rows, reset headers, convert some of the string columns into numerical data type, and create 8 CSV files for data exploration. Then I did data exploration and found that besides leaving 0 reviews, people intend to leave positive reviews like rating 5 or 4.5 out of 5. There's no significant evidence showing that the grading behavior has anything to do with the provider specialty. I checked the data inside each sheet, which shows that Missouri and Wisconsin's datasets contain a lot of data outside of the areas, only around 35% of the data are from the labeled area. Indiana and Kansas datasets contain around 60-70% of data from the labeled area and the rest sheets contain fairly well data from the labeled area. I combined all sheets with columns 'sr-only', 'location-info__office-loc 2', 'location-info__distance', 'review_number', 'provider_detail', 'feats', 'feats1' and 'feats2' in one CSV file called 'combine' ready for model. During creating the model, I use **Decision Tree Classifier** for output variables which are 'sr-only' and 'review_number' separately and **Decision Tree Regression** for two output variables which are 'sr-only' and 'review_number' together.

2 Proposition

Proposition 2.1. This report is based on a proposition that the distribution and the pattern do not vary in different states so I used the overall data sets combined with all states.

Proposition 2.2. I treat duplicate names as different people and ignore the effect of different doctors' impact on the star rating score.

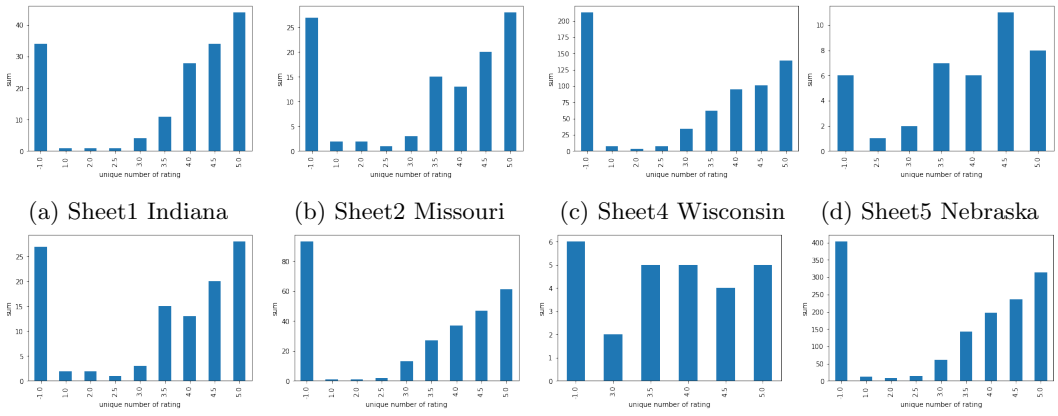
3 Data Cleaning and Data Exploration

There are 7 sheets in the raw data set: Indiana, Missouri, Lowa, Wisconsin, Nebraska, Kansas, and Illinois. I removed empty rows in each sheet and found that there are actually two parts inside of sheet 6 which is the Kansas data set so I split the sheet into two called Kansas and Kansas_2. I checked that there's no duplicate provider name in each sheet, however, there are 546 duplicate names among all values combined. Among those 546 duplications, there are 60 names appeared in different office locations but in the same state, the other 3 names which are 'Dr. Adam Cohen, MD', 'Dr. Matthew Marr, MD', 'Dr. Rao Chundury, MD' appeared in different states (take these 3 as different doctors with the same name). I checked the sr-only value of those 63 duplicate names and found out that all duplicate names have the same star rating except for one, 'Dr. Adam Cohen, MD', also

appeared in two different states and got different ratings in different states, in Indiana, he got a rating of 5 out of 5 but in Illinois, this name got no rating. In this report, I treat all those duplicate names as different people and exclude the name variable for the final model.

3.1 Rating (sr-only) overview

We can notice that among all the variables, only the star rating, star rating review number and location distance info are numerical numbers and others are categorical numbers. So firstly, I took a look at the star rating number and found that most people aren't willing to leave a review information and those who leave a review, they tend to leave a relatively positive review. *Figure 1* shows the results of each sheet's sr-only results distribution (from (a) to (g)) and also the sr-only result of all the sheets combined(h). According to other research, products reviews tend to be bimodal distributions (eg. J- or U-shaped)[1] and the reason causing this kind of distribution is that people tend to write reviews only when they are either extremely satisfied or extremely unsatisfied, people who feel the product is average might not be bothered to write a review according to Nan Hu et al.[2] I categorize the sr-only column as '1-2', '2-3', '3-4' and '4-5'.



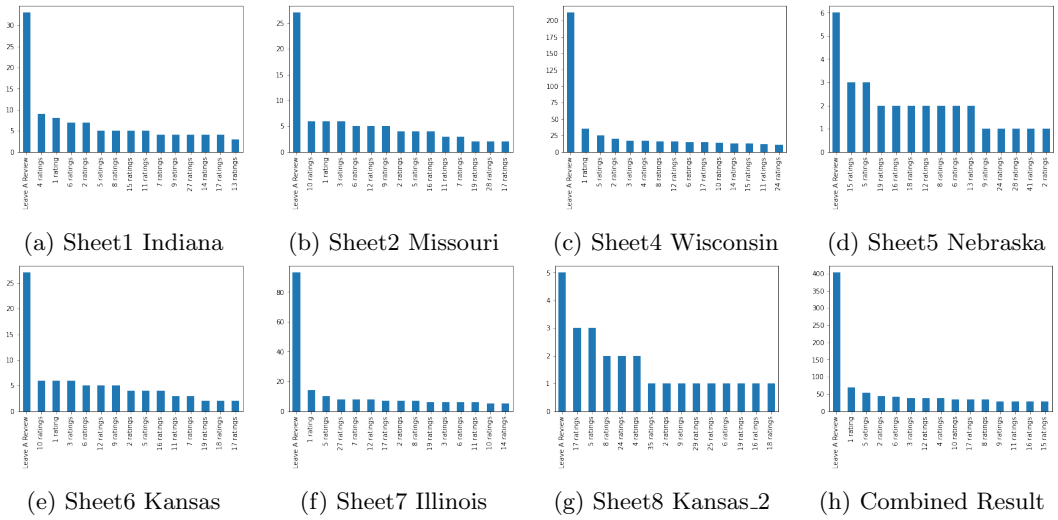
(a) Sheet1 Indiana (b) Sheet2 Missouri (c) Sheet4 Wisconsin (d) Sheet5 Nebraska

(e) Sheet6 Kansas (f) Sheet7 Illinois (g) Sheet8 Kansas_2 (h) Combined Result

Figure 1: sr-only overview

3.2 Review number

Next, I explore the review number value pattern. It turns out that people tend not to leave a countable review. I can't see any pattern of each sheet's review number seperately except for 'Leave A Review' value which means that the review number is null (*Figure 2(a-g)*). We can see in the combined result (*Figure 2(h)*) that first three value of review number are 'Leave A Review', '1 rating' and '5 ratings' which is pretty low.

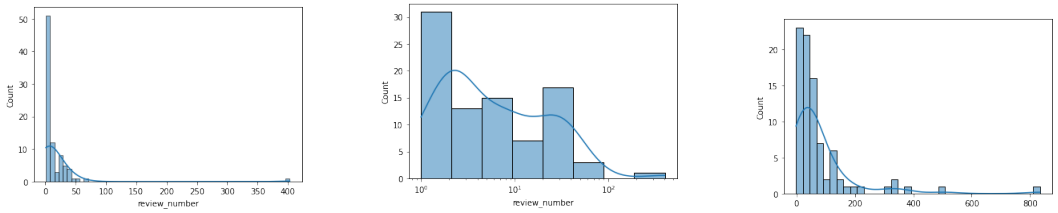


(a) Sheet1 Indiana (b) Sheet2 Missouri (c) Sheet4 Wisconsin (d) Sheet5 Nebraska

(e) Sheet6 Kansas (f) Sheet7 Illinois (g) Sheet8 Kansas_2 (h) Combined Result

Figure 2: review number overview

And I found that the overall review number's distribution is similiar to normal distribution which is shown in *Figure 3(a)*. Because the review number is too decentralized, I try the log scale for review number value, such as *Figure 3(b)*. Besides, *Figure 3(c)* shows that unique review number's distribution is also similiar to normal distribution.



(a) Raw Value (b) Log Scale for x Axis (c) Unique Review Number

Figure 3: Overall Review Number Distribution

As we can see in *Figure 3(c)*, most of the unique review numbers are distributed below 200, so I zoom in to see the detail distribution below 200 which is shown in *Figure 4*. I categorized the distance values into '0-25', '25-50', '50-75', '75-100', '100-125', '125-150', '150-175', '175-400' and '>400' according to the distribution.

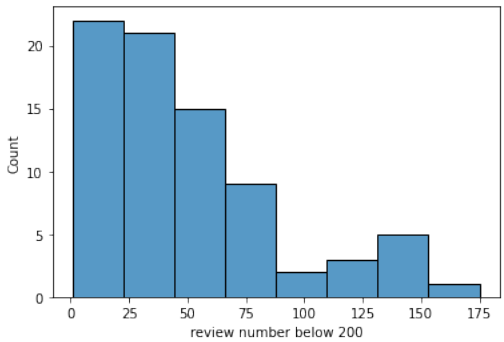
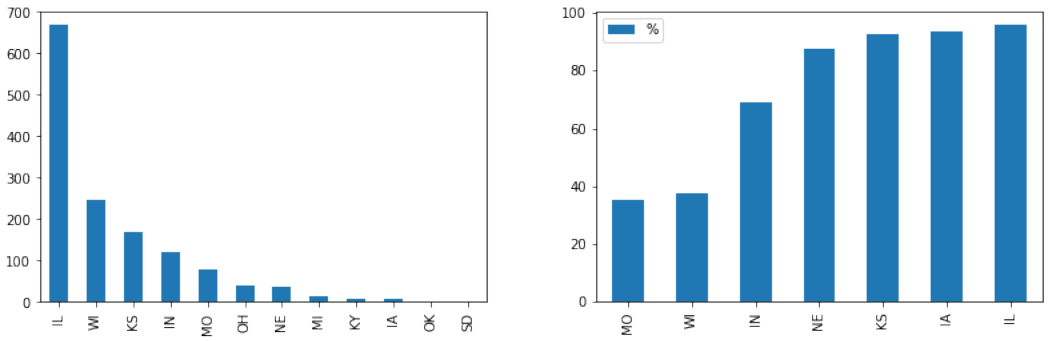


Figure 4: Unique Distance Value Below 200

3.3 States overview

As for states static overview, not as the original sheets only show 7 states, there exist 12 states overall. The top three states that have the most records are Illinois, Wisconsin, and Kansas as shown in *Figure 5(a)*. *Figure 5(b)* shows the percentage of the states within each sheet is actually as the sheet name declares.



(a) State's records count

(b) States Percentage

Figure 5: States overview

3.4 Provider details specialty

As for provider details specialty, I collected all the values from each sheet and found out the overall unique value contains 30 different specialties. I mapped each value into categorical numbers from 0 to 29. I mapped provider details specialty to categorical number as follows: {nan: 0, 'General Surgery': 1, 'Nursing (Nurse Practitioner)': 2, 'Mohs Micrographic Surgery': 3, 'Urogynecology & Reconstructive Pelvic Surgery': 4, 'Oculoplastic Surgery': 5, 'Ophthalmology': 6, 'Physician Assistant (PA)': 7, 'Dermatology (Nurse Practitioner)': 8, 'Orthopedic Hand Surgery': 9, 'Family Medicine': 10, 'Oral & Maxillofacial Surgery': 11, 'Ear, Nose, and Throat': 12, 'Dermatology': 13, 'Urology': 14, 'Bariatric Surgery': 15, 'Phlebology': 16, 'Cosmetic Medicine': 17, 'Emergency Medicine': 18, 'Hair Transplant Surgery': 19, 'General Hand Surgery': 20, 'Dentistry': 21, 'Speech-Language Pathology': 22, 'Breast Surgery': 23, 'Cosmetic, Plastic & Reconstructive Surgery': 24, 'Pediatric Plastic Surgery': 25, 'Dermatologic Surgery': 26, 'Obstetrics & Gynecology': 27, 'Dermatopathology': 28, 'Head & Neck Surgical Oncology': 29}.

For each state, the most common specialty is 'Cosmetic Medicine': 17. For 4 provider details specialties, the star ratings are 5.0 on average, see *Table 1* for detailed info. For 14 provider details specialties, the maximum star rating is 5.0, see *Table 2* for detailed info. For each provider detail specialty, I calculated the sum of each review number and the top five provider detail specialties that got the most reviews are 'Cosmetic Medicine', 'Physician Assistant (PA)', 'Urology', 'Family Medicine', and 'Dermatology (Nurse Practitioner)', see *Table 3* for detailed info.

Table 1: Top rating on avg

provider_detail	sr-only
Oral & Maxillofacial Surgery	5.0
Bariatric Surgery	5.0
Dermatologic Surgery	5.0
Dermatopathology	5.0

Table 2: Max rating is 5.0

provider_detail	sr-only
General Surgery	5.0
Mohs Micrographic Surgery	5.0
Urogynecology & Reconstructive Pelvic Surgery	5.0
Oculoplastic Surgery	5.0
Ophthalmology	5.0
Physician Assistant (PA)	5.0
Family Medicine	5.0
Oral & Maxillofacial Surgery	5.0
Bariatric Surgery	5.0
Phlebology	5.0
Cosmetic Medicine	5.0
Emergency Medicine	5.0
Dermatologic Surgery	5.0
Dermatopathology	5.0

Table 3: Top review number

provider_detail	review_number
Cosmetic Medicine	19312
Physician Assistant (PA)	1318
Urology	934
Family Medicine	241
Dermatology (Nurse Practitioner)	194

3.5 Location information distance

As for 'location-info__distance' column, I extract the pure numbers, remove the unit mi and replace the original column. *Figure 6* shows the distribution of the overall distance from the medical center which combined all sheets. The distance is separated on both sides, which is either close to the medical center or rather far away from the medical center.

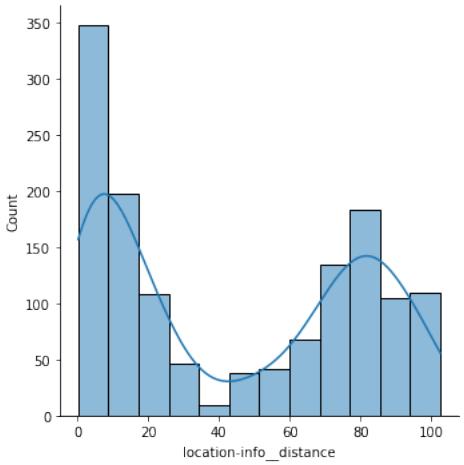


Figure 6: Overall Distance Info

The location information distance will be treated as a categorical variable so it will be an error for not seeing the values of the testing data set in the training data set if not categorized it. So to better categorize the information, I checked the unique value of this column and find out that most of the values are within 100 which is shown in *Figure 7* so I categorize them as '0-20', '20-40', '40-60', '60-80', '80-100' and '>100'.

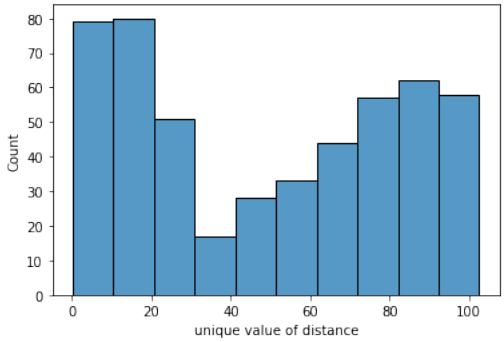


Figure 7: Unique Value of Distance

3.6 Features

As for features of the strength of each provider, which are 'feats-of-strength__feat-title' columns, I collect all values of each column which is 'feats-of-strength__feat-title' & 'feats-of-strength__feat-title 2' & 'feats-of-strength__feat-title 3' of all sheets and combine each column's values separately and find the overall unique values of each column and map them to categorical number. I appended at the end of each sheet 'feats', 'feats 1', and 'feats 2' columns.

For 'feats-of-strength__feat-title', I got the unique value: {'NaN': 0, 'Employs friendly staff': 1, 'Explains conditions well': 2, 'Offers Telehealth': 3, 'Low wait times': 4, 'Easy scheduling': 5, 'Patients found trustworthy': 6}.

For 'feats-of-strength__feat-title 2', I got the unique value: {'NaN': 0, 'Employs friendly staff': 1, 'Explains conditions well': 2, 'Low wait times': 3, 'Easy scheduling': 4, 'Patients found trustworthy': 5}.

For 'feats-of-strength__feat-title 3', I got the unique value: {'Employs friendly staff': 0, 'NaN': 1, 'Explains conditions well': 2, 'Low wait times': 3, 'Patients found trustworthy': 4}.

Table 4 shows the most common features of each location and we can see that the most common feature of 'feats-of-strength__feat-title', 'feats-of-strength__feat-title 2', and 'feats-of-strength__feat-title 3' are 'Easy scheduling', 'Employs friendly staff', and 'Explains conditions well' in order. *Table 5* shows the average star rating values of each feature from high to low, we can see that 'Easy scheduling' got the highest star rating among all features except for 'feats2' which has no value called 'Easy scheduling', instead, feats2 shows that 'Low wait times' got the highest rating score which is the lowest in both 'feats' and 'feats1'.

Table 4: Most common features of each location

location	feats	feats1	feats2
IA	Easy scheduling	Employs friendly staff	Employs friendly staff
IL	Easy scheduling	Employs friendly staff	Employs friendly staff
IN	Easy scheduling	Employs friendly staff	Explains conditions well
KS	Easy scheduling	Employs friendly staff	Explains conditions well
KY	Easy scheduling	Easy scheduling	Patients found trustworthy
MI	Easy scheduling	NaN	NaN
MO	Easy scheduling	Employs friendly staff	Employs friendly staff
NE	Easy scheduling	Employs friendly staff	Explains conditions well
OH	Easy scheduling	Employs friendly staff	Explains conditions well
OK	Easy scheduling	NaN	NaN
SD	Easy scheduling	Employs friendly staff	Low wait times
WI	NaN	NaN	NaN

Table 5: Features' avg sr-only value

feats	sr-only
Easy scheduling	4.443131
Patients found trustworthy	4.000000
Offers Telehealth	3.758721
Explains conditions well	3.611111
Employs friendly staff	3.468085
Low wait times	3.000000
NaN	-0.220251
feats1	sr-only
Easy scheduling	4.527132
Patients found trustworthy	4.500000
Employs friendly staff	4.472435
Explains conditions well	3.979167
Low wait times	3.333333
NaN	0.233507
feats2	sr-only
Low wait times	4.784722
Employs friendly staff	4.585366
Explains conditions well	4.524904
Patients found trustworthy	4.042857
NaN	0.550235

4 Model

The data set that I used for the model includes 'sr-only', 'location-info__office-loc 2', 'location-info__distance', 'review_number', 'provider_detail', 'feats', 'feats1' and 'feats2' 8 columns overall. The shape of the data set is 1389 rows $\times$ 8 columns.

4.1 Classification Method

I used 'sr-only' and 'review_number' as dependent variables separately one by one and for each case, use all the other variables as independent variables. Because the data set is not large, I split the data set into two using **train_test_split** from package **sklearn.model_selection** which split the data set randomly, 70% for training and 30% for testing. I prepare input data using **OrdinalEncoder** from **scikit-learn** to encode each input variable to numerical values and use **LabelEncoder** to encode each output variable to numerical labels. So for training, the size of the input variables is (972, 7), the size of the output variables is (972,) for testing, the size of the input variables is (417, 7), size of the output variables is (417,). Firstly, I checked the correlation within the training variables after encoding and found that the relationship between review number and sr-only is negative, shown in *Figure 8*.

	review_number	sr-only	location-info__office-loc 2	location-info__distance	provider_detail	feats	feats1	feats2
review_number	1.00	-0.79	0.02	-0.03	-0.02	-0.60	-0.38	-0.34
sr-only	-0.79	1.00	-0.11	0.04	0.05	0.83	0.57	0.41
location-info__office-loc 2	0.02	-0.11	1.00	-0.08	-0.05	-0.07	-0.09	-0.02
location-info__distance	-0.03	0.04	-0.08	1.00	0.04	0.02	0.05	-0.01
provider_detail	-0.02	0.05	-0.05	0.04	1.00	-0.01	0.06	-0.05
feats	-0.60	0.83	-0.07	0.02	-0.01	1.00	0.41	0.55
feats1	-0.38	0.57	-0.09	0.05	0.06	0.41	1.00	-0.15
feats2	-0.34	0.41	-0.02	-0.01	-0.05	0.55	-0.15	1.00

Figure 8: Correlation within the training vars

I use **SelectKBest** in **sklearn.feature_selection** for feature selection and chose Chi-squared stats of non-negative features for scores and set k='all' to include all features. Just for reference, the whole data set's features mapping is: {0: 'sr-only', 1: 'location-info__office-loc 2', 2: 'location-info__distance', 3: 'review_number', 4: 'provider_detail', 5: 'feats', 6: 'feats1', 7: 'feats2'}. When review_number is dependent variable, the features mapping will be: {0: 'sr-only', 1: 'location-info__office-loc 2', 2: 'location-info__distance', 3: 'provider_detail', 4: 'feats', 5: 'feats1', 6: 'feats2'}. When sr-only is dependent variable, the features mapping will be: { 0: 'location-info__office-loc 2', 1: 'location-info__distance', 2: 'review_number', 3: 'provider_detail', 4: 'feats', 5: 'feats1', 6: 'feats2'}

When review_number is dependent variable, the best feature is sr-only and the feature selection results are as follows:

Table 6: Feature selection for review_number

Feature name	Score
Feature 0: sr-only	1032.803508
Feature 4: feats	1002.285728
Feature 5: feats1	371.851265
Feature 6: feats2	61.007442
Feature 3: provider_detail	39.189656
Feature 1: location-info__office-loc 2	33.323659
Feature 2: location-info__distance	7.047352

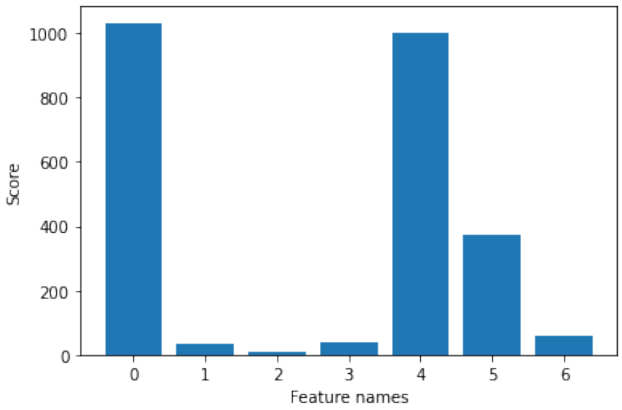


Figure 9: Feature selection result for dependent variable review_number

When sr-only is dependent variable, the best feature is review_number and the feature selection results are as follows:

Table 7: Feature selection results for sr-only

Feature name	Score
Feature 2: review_number	2779.373843
Feature 4: feats	1330.800413
Feature 5: feats1	534.199561
Feature 6: feats2	93.874274
Feature 0: location-info__office-loc 2	18.927337
Feature 3: provider_detail	17.756385
Feature 1: location-info__distance	3.200193

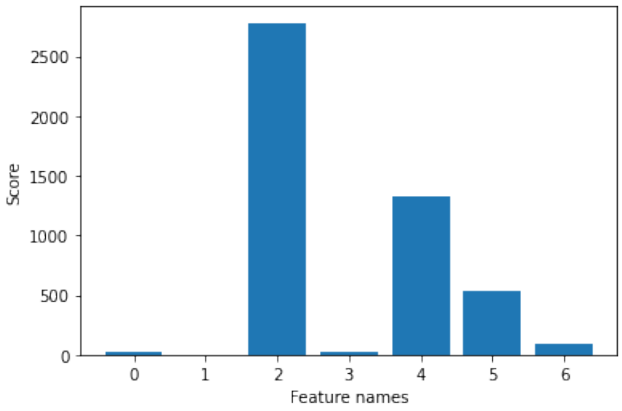


Figure 10: Feature selection result for dependent variable sr-only

I used **DecisionTreeClassifier** to fit the data and get the accuracy result: 77.7% for review_number as dependent variable and 90.65% for sr-only as dependent variable.

Besides, I tried **RFE** (Recursive Feature Elemination) for both cases, set features to select

as 3 and estimator is still **DecisionTreeClassifier**, split the data sets into two folders and repeat 3 times for cross-validate. The accuracy result for dependent variable review_number is 80.2%, the standard deviation is 0.011, the mean absolute error is 0.1 and the standard deviation is 0.016. The accuracy result for dependent variable sr-only is 91.6%, the standard deviation is 0.015, the mean absolute error is 0.102 and the standard deviation is 0.017.

Table 8: Feature selection for dependent variable review_number

Column	Whether selected	Rank
Column: 0 (sr-only)	Selected True	Rank: 1.000
Column: 1 (location-info__office-loc 2)	Selected True	Rank: 1.000
Column: 2 (location-info__distance)	Selected True	Rank: 1.000
Column: 3 (provider_detail)	Selected False	Rank: 3.000
Column: 4 (feats)	Selected False	Rank: 4.000
Column: 5 (feats1)	Selected False	Rank: 5.000
Column: 6 (feats2)	Selected False	Rank: 2.000

Table 9: Feature selection for dependent variable sr-only

Column	Whether selected	Rank
Column: 0 (location-info__office-loc 2)	Selected False	Rank: 2.000
Column: 1 (location-info__distance)	Selected False	Rank: 4.000
Column: 2 (review_number)	Selected True	Rank: 1.000
Column: 3 (provider_detail)	Selected False	Rank: 5.000
Column: 4 (feats)	Selected False	Rank: 3.000
Column: 5 (feats1)	Selected True	Rank: 1.000
Column: 6 (feats2)	Selected True	Rank: 1.000

4.2 Regression Model

I split the data set into two parts, first is a dependent variable which includes 'sr-only' and 'review_number', and next is independent variables whose mapping is: {0: 'location-info__office-loc 2', 1: 'location-info__distance', 2: 'provider_detail', 3: 'feats', 4: 'feats1', 5: 'feats2'}.

Similarly, I pass **DecisionTreeRegressor** as the estimator and set 3 features to be selected to **RFE** and used **DecisionTreeRegressor** as the model. The three features that be selected are 'location-info__office-loc 2', 'location-info__distance', and 'provider_detail'. Then use **RepeatedKFold** to split the data sets into 10 folders and repeat 3 times for cross-validate. I got the mean absolute error of 1.3 from cross-validation and the standard deviation of the scores is 0.132.

Table 10: Feature selection for regression model

Column	Whether selected	Rank
Column: 0 (location-info__office-loc 2)	Selected True	Rank: 1.000
Column: 1 (location-info__distance)	Selected True	Rank: 1.000
Column: 2 (provider_detail)	Selected False	Rank: 2.000
Column: 3 (feats)	Selected True	Rank: 1.000
Column: 4 (feats1)	Selected False	Rank: 3.000
Column: 5 (feats2)	Selected False	Rank: 4.000

4.3 Model evaluation

A regression model can only predict values that are lower or higher than the actual value. As a result, the only way to determine the model's accuracy is through residuals, in this case, I use mean absolute value to measure and compare the models' accuracy. As we can see in *Table 11*, where I simply use a decision tree classification model without any feature selection on 'review_number' and 'sr-only' as dependent variable individually, the result shows that when 'sr-only' be the dependent variable will get a better result (accuracy: 90.65%). As we can see in *Table 12*, where I use a classification model with feature selection (select top 3 features). Similarly, the result shows that when using sr-only as the dependent variable will get a better result (accuracy: 91.6%). Apparently, when feature selection is included, the model will get a better result. For comparison with the regression model's result, I also calculate the MAE results. The MAE results are aligned with the accuracy results that it will get a better result when sr-only is the dependent variable. *Table 13* shows the MAE result of the regression model which is 1.3. Compared to the classification model, the regression model's result is worse.

Table 11: Classification method without feature selection

Dependent var	review_number	sr-only
Accuracy	77.70%	90.65%
MAE	1.105	0.166

Table 12: Classification method with feature selection

Dependent var	review_number	sr-only
Accuracy	80.2%	91.6%
Std for Accuracy	0.011	0.015
MAE	0.1	0.016
Std for MAE	0.102	0.017

Table 13: Regression method with feature selection

Dependent var	review_number & sr-only
MAE	1.3
Std	0.132

5 Result

To sum up, since the model results are better when sr-only is the dependent variable, sr-only is more relevant to other variables. In other words, it's more accurate to predict sr-only with the other information. The most relative value to sr-only is the review number, according to the correlation in *Figure 8*, we know that those two variables are negatively affected, that is the higher the review number is, the lower the sr-only will be. The second relative value to sr-only is feats which are positively related to each other. The third relative value is feats1, which is positively related to sr-only. The fourth one is feats2 which is positively related to sr-only. The fifth one is location-info__office-loc 2 which is negatively related to sr-only. Lastly, the sixth one is location-info__distance which is positively related to sr-only.

According to data exploration, we can learn that the distribution of sr-only is typically J- or U-shaped, and the unique review number is distributed similarly to a normal distribution, and the review number declines gradually as the rating results increase gradually except for the null value. As for features, I explore three feature columns separately and found that 'Easy scheduling' got the highest sr-only score for 'feats' and 'feats1' columns, 'Patients found trustworthy' is the second highest value for 'feats' and 'feats1' column, 'Offers Telehealth' got the third highest sr-only score which does not appear in the other two feature columns. As 'feats' is the most relative one for sr-only, people tend to value 'Easy scheduling', 'Patients found trustworthy', and 'Offers Telehealth'. As for provider detail specialty, the data shows that 'Cosmetic Medicine' has the largest review number, and then 'Physician Assistant (PA)', 'Urology', 'Family Medicine', and 'Dermatology (Nurse Practitioner)' in sequence. As we don't have the total patients' data, we cannot find out whether the number of 'Cosmetic Medicine' patients is the most certain but according to the distribution, we can presume that the reason for the high review number is because large patients number. As for the state information, we can see that the sr-only distribution for each state is similar according to *Figure 1* so there's no big difference between different states which align to the model's result that the sr-only is not so affected by states. And similarly, the location distance is also not very effective for sr-only.

References

[1] J. Yu, D. Landy, and R. Goldstone, "How do people use star rating distributions?" *Proceedings of the Annual Meeting of the Cognitive Science Society*, vol. 44. Retrieved from https://escholarship.org/uc/item/6fb9821g.

[2] N. Hu, N. Hu, P. A. Pavlou, and J. J. Zhang, "Overcoming the j-shaped distribution of product reviews (october 18, 2009)," *Communications of the ACM, October 2009*, vol. 52, No. 10, Available at SSRN: https://ssrn.com/abstract=2369332.

A Personal Note

During the data cleaning process, I first transit 3 feats attributes into numeric numbers. I wanted to transmit the provider details to a numeric data type as well. Still, after I did that I realized that if I transited all those data types into the same scale of numeric, there would be problems for models and feature selection methods to recognize the features as they might be highly relevant since the numbers are the same according to Python. So I decided to try the raw data for feature selection first.

For feature selection, I need to find out which distribution most fits the output (target) variable: sr-only (rating result). I learned from this website that I could use Fitter library to find the fittest distribution. Use sheet1's 'sr-only' header, the result is **wrapcauchy** distribution and the sumsquare_error is 32.405518 which is too big and the p-value is still less than 0.05 which means that the distribution is still not good enough.

When I did data exploration, I found that there are two data sets inside Kansas data, so I moved back to clean this data set again by splitting this sheet into two sheets and named Kansas_2 and save all of those sheets separately to 7 different csv files use Path within pathlib package.

During feature selection, KFold is a cross-validator that divides the dataset into k folds. Stratified (StratifiedKFold) is to ensure that each fold of dataset has the same proportion of observations with a given label. Refer to stackoverflow link for a detailed explanation. When use cross_val_score (cross validation) to evaluate the model, the 'scoring' parameter can refer to model evaluation documentation for details.