ORIE 4741 Project Midterm Report:
Predict Trends of Gentrification by Restaurant-Related Data

Liye Zhong(lz246), Lu Li(ll565), Rong Tan(rt389)

THE PROBLEM

Gentrification is a process of changing the character of a neighborhood through the influx of more affluent residents and businesses. And it is a crucial step in the development of cities around the United States. Gentrification often involves an increase in housing prices, the number of evictions, and a change in the retail, consumer goods landscape in the region. Lives of the people, rich or poor, are affected by such a phenomenon. The process is a gradual change and often hidden in trivial aspects of people's daily life. We want to use big data to identify such trends.

The question that we ultimately want to answer is whether the change in the restaurants in the region can tell a story about gentrification.

THE DATA

We wanted to use three datasets from Kaggle. One is the New York City restaurant review data. These are the data collected from restaurant inspection at each restaurant for at least once a year(this data set will be referred to as "Restaurant Data". Relevant data points are the restaurant type in text data, e.g. "Italian", restaurant zip code as normal data, e.g. "10025", county name in-text "Albany", violations in text, e.g. "hot food not held above 140F" and a score given by the inspectors as normal data, e.g. "30", the lower the score it is, the better the situation is in restaurant. We believe that the score is a good indicator of the quality of the restaurants as better restaurants tend to have better management and stricter policies on their kitchen conditions and operations. An alternative is to use the "Grades" given by the inspectors the indicator for restaurant quality. However, we discover that there are too many "N/A" data in the "grade" category (about ⅔ of the data). For our analysis, we removed the "N/A" rows for "score" since there's no suitable replacement for missing data.

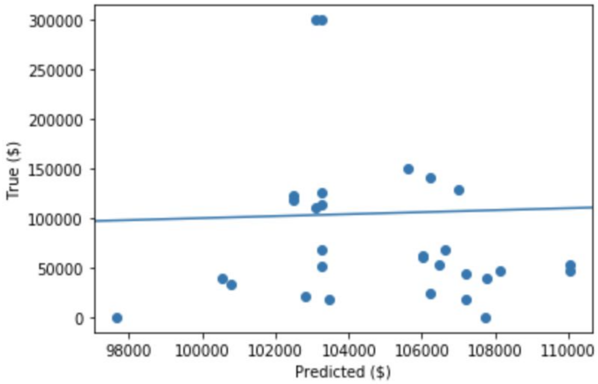
We also used another dataset from Kaggle, which is the New York City restaurant week review data. We tried to use the Yelp open data for NYC restaurants first, but the dataset's size is too small for NYC. So we used another dataset, the New York City restaurant week review data. The dataset contains detailed information for 347 restaurants around NYC, which contains useful columns of name, average_review, food_review service_review, ambience_review, value_review, price_range, postal_code.

The third data set that we are using is the Census Bureau Data on the local population income ("This data will be referred to as "Income Data"). This set of data illustrates the profile of citizens in each zip code district. The relevant data listed are: zip code, state code, the mean and median income (all in the form of normal data). For our analysis, we reformatted the data into the same format for later regression purposes.

ANALYSIS METHOD

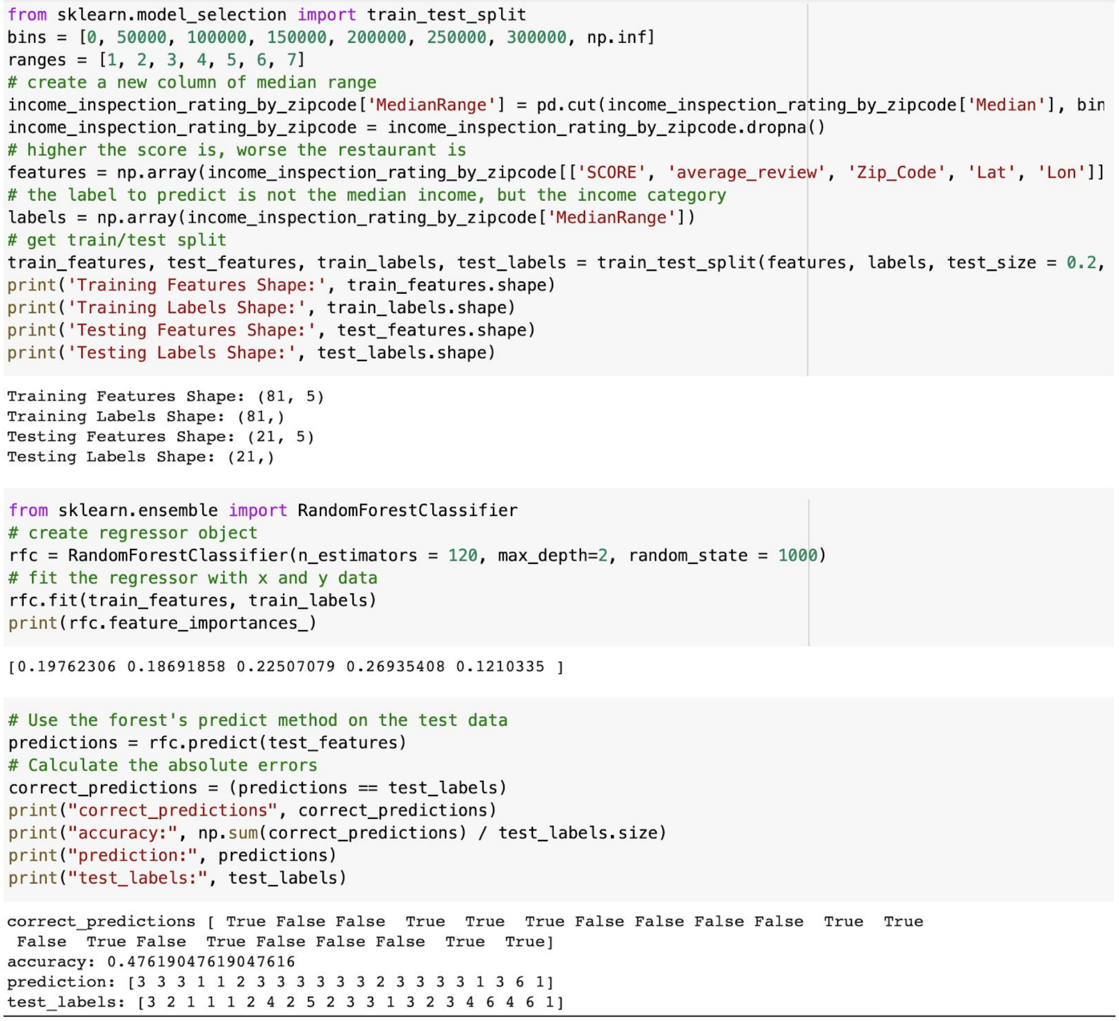
We first want to explore the relationship between restaurant quality and income levels in the area. We choose "Score" from Restaurant data as the feature (X) to predict the "Median" income (Y) for each zip code in New York City. The simplest regression model is $Y = \beta_0 + \beta_1 * X$. Such a test is important because it can show us how closely correlated are these two variables. If there's indeed a strong correlated relationship between X and Y, then we can observe the change in restaurant inspection in one area and predict how the income level in that area may change correspondingly.

In our first training model, we did an 80-20 train test data split. The results returned are β_0 = -1135.12 β_1 = 109272.98 This model predicts horribly as we've suspected since we are only using one feature to make predictions and here it shows that such feature is not a good predictor.



However, what if we remove outliers? Would the model predict better? Since there are not many restaurants in each zip code zone, the scores may be heavily skewed by one or two outliers:
The results returned are $\beta_0 = 6333.11344402$ $\beta_1 = -28415.50401049$

Part II: In order to improve the model, we decided to adopt restaurant week review data and choose another model using combined features with the previous dataset:



```
from sklearn.model_selection import train_test_split
bins = [0, 50000, 100000, 150000, 200000, 250000, 300000, np.inf]
ranges = [1, 2, 3, 4, 5, 6, 7]
# create a new column of median range
income_inspection_rating_by_zipcode['MedianRange'] = pd.cut(income_inspection_rating_by_zipcode['Median'], bin
income_inspection_rating_by_zipcode = income_inspection_rating_by_zipcode.dropna()
# higher the score is, worse the restaurant is
features = np.array(income_inspection_rating_by_zipcode[['SCORE', 'average_review', 'Zip_Code', 'Lat', 'Lon']]
# the label to predict is not the median income, but the income category
labels = np.array(income_inspection_rating_by_zipcode['MedianRange'])
# get train/test split
train_features, test_features, train_labels, test_labels = train_test_split(features, labels, test_size = 0.2,
print('Training Features Shape:', train_features.shape)
print('Training Labels Shape:', train_labels.shape)
print('Testing Features Shape:', test_features.shape)
print('Testing Labels Shape:', test_labels.shape)
```

```
Training Features Shape: (81, 5)
Training Labels Shape: (81,)
Testing Features Shape: (21, 5)
Testing Labels Shape: (21,)
```

```
from sklearn.ensemble import RandomForestClassifier
# create regressor object
rfc = RandomForestClassifier(n_estimators = 120, max_depth=2, random_state = 1000)
# fit the regressor with x and y data
rfc.fit(train_features, train_labels)
print(rfc.feature_importances_)
```

```
[0.19762306 0.18691858 0.22507079 0.26935408 0.1210335 ]
```

```
# Use the forest's predict method on the test data
predictions = rfc.predict(test_features)
# Calculate the absolute errors
correct_predictions = (predictions == test_labels)
print("correct_predictions", correct_predictions)
print("accuracy:", np.sum(correct_predictions) / test_labels.size)
print("prediction:", predictions)
print("test_labels:", test_labels)
```

```
correct_predictions [ True False False  True  True  True False False False False  True  True
 False  True False  True False False False  True  True]
accuracy: 0.47619047619047616
prediction: [3 3 3 1 1 2 3 3 3 3 3 3 2 3 3 3 3 1 3 6 1]
test_labels: [3 2 1 1 1 2 4 2 5 2 3 3 1 3 2 3 4 6 4 6 1]
```

The fourth method we tried is to use the random forest classifier, instead of a regression model.
In this model, we decided to transform the median incomes of each zip code area into median income ranges for mode simplicity and readability. For every median income of each zip code area, we classify each median income into seven classes [0-50000, 50000-100000, 100000-150000, 150000-200000, 250000-300000, 300000+]. And we trained our random forest classifier with features of restaurant score, restaurant review rating, zip code, latitude and longitude. After fitting the model by using the 80% of the dataset and predicting the rest 20%, we achieved an accuracy rate of 47.6%.
Interestingly, the feature importance of the 5 features in the random forest classifier is weighted evenly. In the model, all the features are equally important, and we may find more features to adopt in this random forest classifier to find more equally important features and boost our accuracy.

WORD OF CAUTION:
There are several items that subject our model to flaws:

1. The restaurant week data set is relatively small and the restaurants tend to concentrate in popular dining areas, though we do see the list expanding year over year and this should be less of a concern going forward.
2. We used inspection scores and local income levels rather than changes in inspection scores and changes in local income levels. The initial explorations of the relationship among different features but looking at the change in data can possibly help us make better predictions.
3. We are also aware that there might be a lag in housing prices. We are using income level as the y to predict here. But housing prices may change a few months or half a year later than the change in income level. We were not able to find a good dataset on rent in NYC. If we have more information on hand, building a model directly with rent prices within each neighborhood may improve our results responding to the question proposal.

NEXT STEP
Finally, this model, of course, can be and need to be improved. These seven features have similar coefficient weights in the model currently. This leads to intuitive suspicion of the results. We would want to take the next step of using feature engineering to find a better fit. An alternative method is to add a prior to the data before fitting the models. Furthermore, we divided the data by zip code in NYC. As known that NYC is divided into different neighborhoods, we can look into making predictions for each neighborhood rather than for each zip code. We can also improve the quality of data by digging further into the income level data. For example, in the initial models built, we used one year of income data. Last but not least, we have an extra set of data on NYC crime levels. People's preferences for living arrangements and willingness to pay higher rent are highly affected by the local crime levels. We plan on adding features from NYC crime data to make more accurate predictions.