Exploring Factors of a Restaurant Success in Florence Via Predictive Modelling

Iacopo Ghinassi

Applied Data Science Capstone by IBM/Coursera

1. Introduction: Business Problem

In this project we will try to find what makes a resturant succesful in a major Italian touristic city. Specifically, this report will be targeted to stakeholders interested in opening a *restaurant* in **Florence**, Italy.

We are particularly interested in understanding which factors might correlate with success of a food venue in this town and, as such, we can try to predict some measure of success of already existing venues via a set of features and machine learning techniques. By looking at the performance obtained and the relative importance of each feature in the model, we can then see what factor might determine the success of a restaurant. In addition, a prediction of how a restaurant will perform based on those same features can be obtained as well.

The outcome of this analysis might then inform the decision of stakeholders willing to open a restaurant in Florence, while providing a foundation for future research in similar town and/or with additional/different features.

2. Data

Based on definition of our problem, we will need to find the following:

- information about existent restaurants in Florence, including their characteristics and a measure of their performance.

As a ready database containing such information could not be found, we will proceed in creating one from scratch using different data sources. In this process, candidate features and target variables will be individuated.

Following data sources will be needed to extract/generate the required information:

- Name of neighborhoods of Florence and some general demographic information of their macro-areas will be scraped from the table contained in the following Wikipedia page: https://it.wikipedia.org/wiki/Quartieri_di_Firenze.
- coordinates of the neighborhoods will be obtained via geopy, a python open-source library allowing to send requests to **Nominatim's API**.
- restaurants information, including their type and location in every neighborhood will be obtained using **Foursquare API**
- ratings of restaurants in Florence will be scraped from the travel website TripAdvisor, at the following address https://www.tripadvisor.co.uk//Restaurants-g187895-oa30-Florence_Tuscany.html#EATERY_LIST_CONTENTS

Once obtained, the different data can be variously combined in a form that is both theoretically sound and effective to reach our main goal of assessing the importance of different features in a restaurant success, as well as predicting the success of a new venue based on such features.

2.1 Obtaining Neighborhoods of Florence names and geolocation

As mentioned, the name and some characteristics of the neighborhoods of Florence will be obtained by scraping an existent table on Wikipedia (https://it.wikipedia.org/wiki/Quartieri_di_Firenze). Such a table contains the name of the neighborhoods grouped inside the *quartieri* or boroughs of Florence, representing a higher-level division of the city. Demographic information at the borough level (*n.b.* not per neighborough) are included as well in the table. Such information seem already valuable predictors for the future prediction task, so we will keep them in mind but remembering that they pertain to boroughs and, as such, are shared by neighborhoods in the same borough.

From the scraped Wikipedia page, a series of transformations will allow to obtain a dataframe having each neighborhood as a row. At this stage, both the Borough level and neighborhood level dataframes are augmented with geolocation data (i.e. latitude and longitude) obtained via geopy: these data will be later used to obtain venues' information per neighborhood, while allowing us to show visually the position of Boroughs and Neighborhoods in Florence (Figure 1 and 2).

Figure 1: Boroughs of Florence.

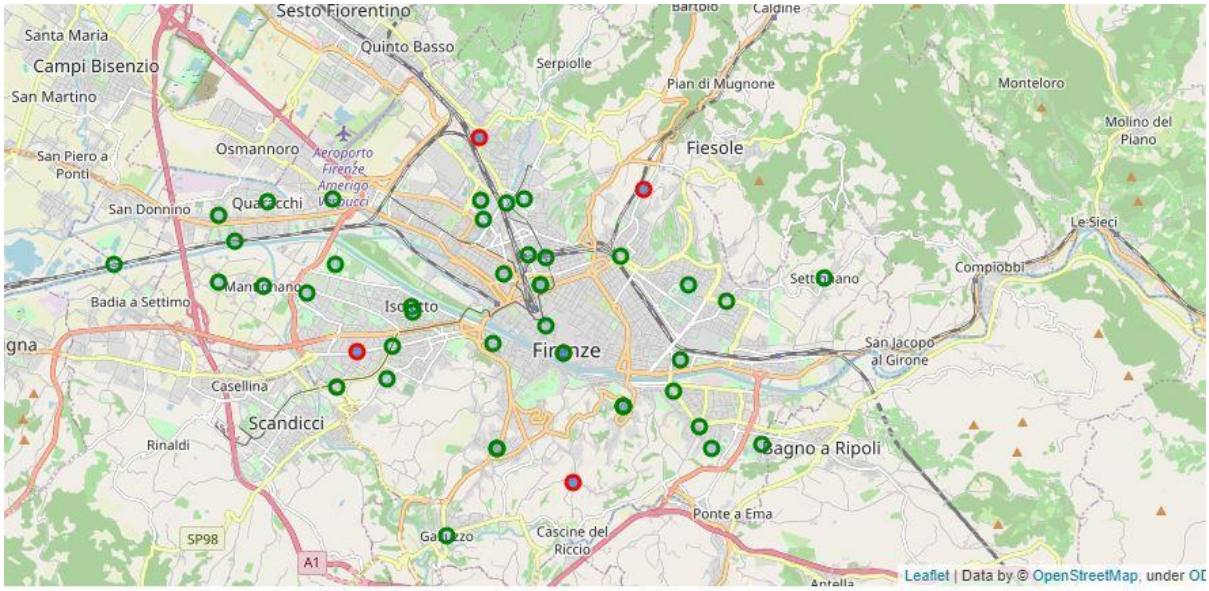

Figure 2: Florence with its Boroughs (in red) and its Neighborhoods (green).

2.2 Obtaining Neighborhoods' Restaurants' names, category and location via Foursquare API

At this point, we have the coordinates for each neighborhood. By sending an "explore" request to Foursquare's API [1] with the coordinates of each neighborhood, we can obtain a list of venues that are contained in Foursquare's database and that are within a predefined radius of the given location (in our case set to be 500 meters). These venues can then be filterd by category so that we save just the restaurants, finally creating a dataframe of Florentine restaurants with associated geolocation data and restaurant category, in addition to the neighborhood of Florence they belong to. Figure 3 shows some rows of the final table thus obtained.

	Neighborhood	Neighborhood Latitude	Neighborhood Longitude	Venue	Venue Latitude	Venue Longitude	Venue Category
0	Centro Storico	43.769263	11.254822	Mangia Pizza Firenze	43.768940	11.253940	Pizza Place
1	Centro Storico	43.769263	11.254822	Osteria Vini Vecchi Sapori	43.770068	11.256763	Italian Restaurant
2	Centro Storico	43.769263	11.254822	La Bussola	43.770250	11.252494	Italian Restaurant
3	Centro Storico	43.769263	11.254822	La Petite	43.770619	11.253610	Restaurant
4	Centro Storico	43.769263	11.254822	Trattoria Dall'Oste Signoria	43.771240	11.256172	Italian Restaurant

Figure 3: Table containing Restaurants of Florence retrieved by Foursquare. The column names show what information are associated with each restaurant at this stage, including location and restaurant category.

2.3 Obtaining Ratings from TripAdvisor and combining dataframes

Having a list of restaurants and some of their information, we can start thinking about what features can be used to build a predictive model out of the dataframe. As a first thing for such a model, we need some objective defining the performance of our restaurants. Data directly showing the economic performance of these venues are not publicly available, but, luckily, we have a very popular proxy for restaurants' success represented by customer generated ratings widely available on the Internet via travel websites. One such a website, and a very popular one, is TripAdvisor. We can obtain the ratings for all the reviewed restaurants in Florence on TripAdvisor [2] by scraping the page with the webscraper package BeautifulSoup [3]. As we will see, at this point we will also have to account for a potential flaw in these ratings, represented by the fact that certain restaurants might have very high rating but from very few reviewers, while other restaurants might have an higher absolute number of ratings but a lower average as more users rated it. To account for this, we will have to normalise the ratings, taking into consideration also how many ratings were available on TripAdvisor for each restaurant. Before doing so, however, we also need to join the dataframe obtained by scraping TripAdvisor and the one previously obtained via the Foursquare API. In particular, we want to perform a join, such that all observations from the Foursquare dataframe are included, while excluding the restaurants found of TripAdvisor that do not have a correspondence in the Foursquare dataframe. This is because we will need the location data and category for the restaurants and we can't approximate them just with the data obtained on TripAdvisor, while we can well approximate the ratings on the basis of the ones of restaurants in the same neighborhood. Figure 4 and 5 show the table obtained by scraping TripAdvisor and the final joint table respectively.

	name	rating	num_ratings
0	Gustarium	5.0	1
1	Degusteria Italiana agli Uffizi	5.0	697
2	Pizzagnolo - Pizza e Sfizi	5.0	92
3	Mangiapepe	5.0	352
4	Pitti Express	5.0	86

Figure 4: By scraping the results for Restaurant in Florence found on TripAdvisor we have obtained a table of Florentine restaurants with associated names, ratings and number of reviews.

	Neighborhood	Neighborhood Latitude	Neighborhood Longitude	Venue	Venue Latitude	Venue Longitude	Venue Category	rating	num_ratings
0	Centro Storico	43.769263	11.254822	Mangia Pizza Firenze	43.768940	11.253940	Pizza Place	4.5	1.0
1	Centro Storico	43.769263	11.254822	Osteria Vini Vecchi Sapori	43.770068	11.256763	Italian Restaurant	NaN	NaN
2	Centro Storico	43.769263	11.254822	La Bussola	43.770250	11.252494	Italian Restaurant	4.5	2.0
3	Centro Storico	43.769263	11.254822	La Petite	43.770619	11.253610	Restaurant	4.0	168.0
4	Centro Storico	43.769263	11.254822	Trattoria Dall'Oste Signoria	43.771240	11.256172	Italian Restaurant	NaN	NaN

Figure 5: We joined the results of scraping TripAdvisor and the restaurants with location data from Foursquare's API. Note that where a restaurant in the Foursquare dataframe is not matched by a restaurant with the same name on the ratings' dataframe NaN values are introduced in the rating and number of reviews field for the restaurant.

2.4 Feature Engineering: preprocessing and data transformation to obtain dependent and independent variables

At this stage, we will obtain both target variables and predictors via a series of transformation/further elaboration on the data we already gathered. This step can be considered as being part of preprocessing the data, as we still are trying to distil the information that will enable us later to build a successful model.

We previously noticed how the ratings might be biased by the differences in number of reviews per restaurant: as such we can employ the number of reviews to normalise the ratings. In doing so, we can also substitute the missing values in our table with appropriate value.

For both these tasks we can assume a Bayesian perspective. Under this perspective, the value of each rating is assumed to be close to the average of the population the restaurant belongs to, i.e. the average of the restaurants in the same neighborhood. The more number of reviews we have, the more we can be sure that the average rating we have recorded for that particular restaurant is accurate, the less number of reviews the more we will tend to back-off to our prior assumption, i.e. that the actual rating of that restaurant is close to the average rating of restaurants in that neighborhood. If we do not have any rating for a particular restaurant, we just rely entirely on such a prior, by assigning the prior to that restaurant. To normalise the ratings by their number of ratings, we will first normalise the number of ratings per neighborhood so that they range between 0 and 1. There are several ways of doing it, here we will do this by using an implementation of the MinMax function, such that $x=(x-min(X))/(max(X)-min(X))$ whereas lower case x represents the number of reviews for the current restaurant, while capital X represents all the number of reviews

for each restaurant in the same neighborhood as x. The new weighted ratings are finally obtained via the following equation:

$WeightedRating = WeightingFactor * ActualRating + (1 - WeightingFactor) * Prior\ Belief$, where WeightingFactor is the normalised number of users previously obtained for each observation, the ActualRating is the value of rating for the current observation and PriorBelief is the average rating per neighborhood. To explain it in simple terms, this equation makes observations with few reviews be more similar to the average rating of the restaurants in the neighborhood, being it our best guess of what the rating for that particular restaurant should be. Obervations with a lot of reviews, instead, are assumed to have a trustable rating associated with them, such that the value remains similar to itself. Figure 6 shows the final table that we obtained storing the new weighted ratings computed as just described.

	Neighborhood	Neighborhood Latitude	Neighborhood Longitude	Venue	Venue Latitude	Venue Longitude	Venue Category	rating	num_ratings	normalised_number_ratings	weighted_rating
0	Centro Storico	43.769263	11.254822	Mangia Pizza Firenze	43.768940	11.253940	Pizza Place	4.50	1.0	0.000000	4.250000
1	Centro Storico	43.769263	11.254822	Osteria Vini Vecchi Sapori	43.770068	11.256763	Italian Restaurant	4.25	NaN	NaN	4.250000
2	Centro Storico	43.769263	11.254822	La Bussola	43.770250	11.252494	Italian Restaurant	4.50	2.0	0.001672	4.250418
3	Centro Storico	43.769263	11.254822	La Petite	43.770619	11.253610	Restaurant	4.00	168.0	0.279264	4.180184
4	Centro Storico	43.769263	11.254822	Trattoria Dall'Oste Signoria	43.771240	11.256172	Italian Restaurant	4.25	NaN	NaN	4.250000

Figure 6: Weighted rating's column include the new rating for each restaurant that will be used as target variable to build our predictive model.

2.5 Obtaining Predictive Features

After having our target variables in place, we need some independent variables that could predict them. Such features should represent characteristics of the venue or of the neighborhood is in that we think might be a factor of success/failure for the venue itself. As a first thing, we already had some demographic information for the boroughs of Florence, we can add them to our rated_restaurants dataframe. An important factor in a relatively small touristic city seems also to be the distance from the city centre and, as such, we can calculate a form of distance from city centre (in our case the euclidean distance) as a feature for each venue. As there is a possibility that we are missing other shared characteristics between neighborhoods when ignoring venues other than restaurants, we will also cluster all the neighborhoods in a fixed numbers of clusters via k-mean using all the venues' categories and, then, we will use the clusters' labels assigned to each neighborhood as features as well. Finally, we also have indicators of what the restaurant is serving, via the category field obtained via Foursquare. Both the neighborhood's cluster and the category of restaurant associated with each observation can be represented as variables via the so called one-hot encoding, i.e. by creating vectors of length equal to the number of categorical values and filling them with 0 except for a 1 in the position of the categorical value associated with the current observation. With all these features, we will create seven different training sets to test whether adding additional features actually will help getting better results. The final training sets, then, will be the following:

- **X_0**: including just the scaled numerical features (distance from centre, inhabitants, density and area of borough).
- **category**: including just the one-hot encoded venues' categories.
- **clusters**: including just the one-hot encoded neighborhoods' clusters.
- **X_category**: including both X_0 and the one-hot encoded venues' categories.
- **X_clusters**: including both X_0 and the one-hot encoded neighborhoods' clusters.

- **cluster_category**: including just the categorical features (just X_0 missing).
- **X_all**: including all features.

3. Methodology

This part of the report sets up the methodology we will use in performing our analysis. First, we are going to explore the data previously obtained via some visualisations and basic descriptive statistics. Then, keeping our obective in mind, we will try different machine learning model with the intention of predicting the rating (i.e. the success) of the restaurant base on the set of features we defined. The machine learning model that we want should be easily explorable, so that the various factors of success could be individuated and communicated to the relevant stakeholders. At the same time, the model should perform well enough to make the results reliable. Specifically, we will use a multiple linear regression and a regression tree model, both having the advantage of being easily explorable in terms of the effect that each feature has on the output.

4. Analysis

4.1 Exploratory Data Analysis

As a first step, we will explore the data obtained in the previous section in more details. Even if the Data section did explore the data under some respect, additional visualisations and descriptive statistics on the final training set and target variables at hand might give us a better idea of what kind of patterns could be found between the two. We will start with some histograms, so as to get an understanding of the distribution of our categorical variables. After that, we can eventually plot each numerical feature against the target variable with a regression plot to see if some linear relationship can be individuated. A general summary of the numerical data will be generated as well. Figure 7 and 8 show the distributions of restaurants' categories and restaurants' neighborhood clusters respectively. Figure 9 combines the two and show the distribution of restaurant categories inside each neighborhood cluster.

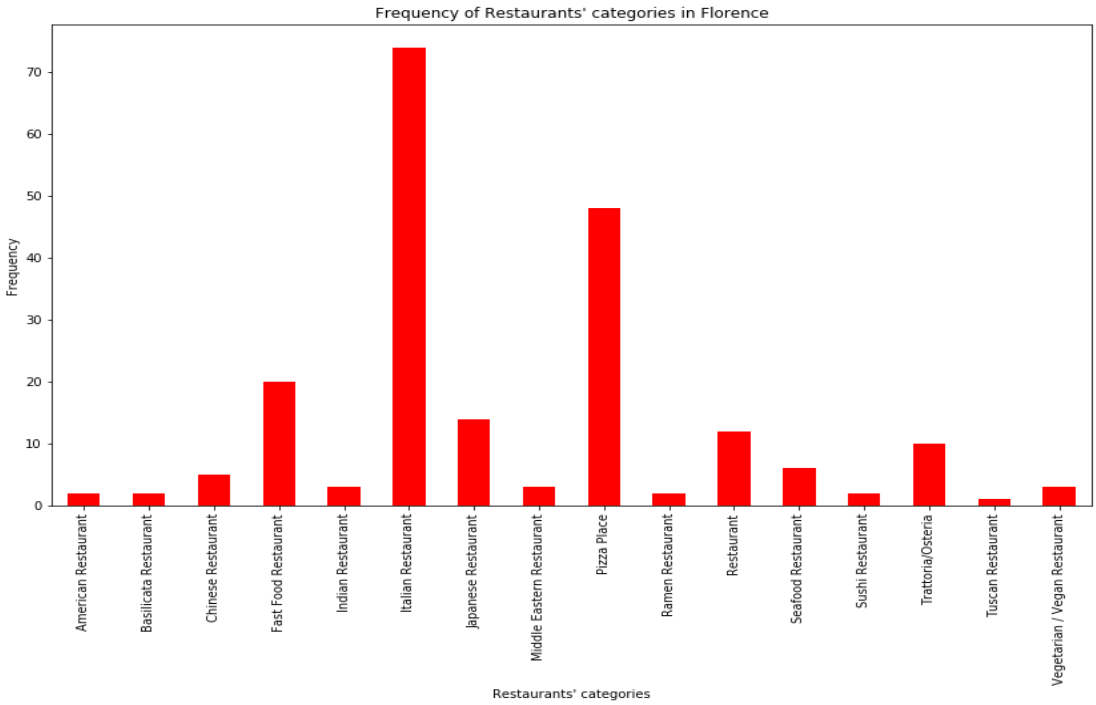

Figure 7: Distribution of restaurants' categories

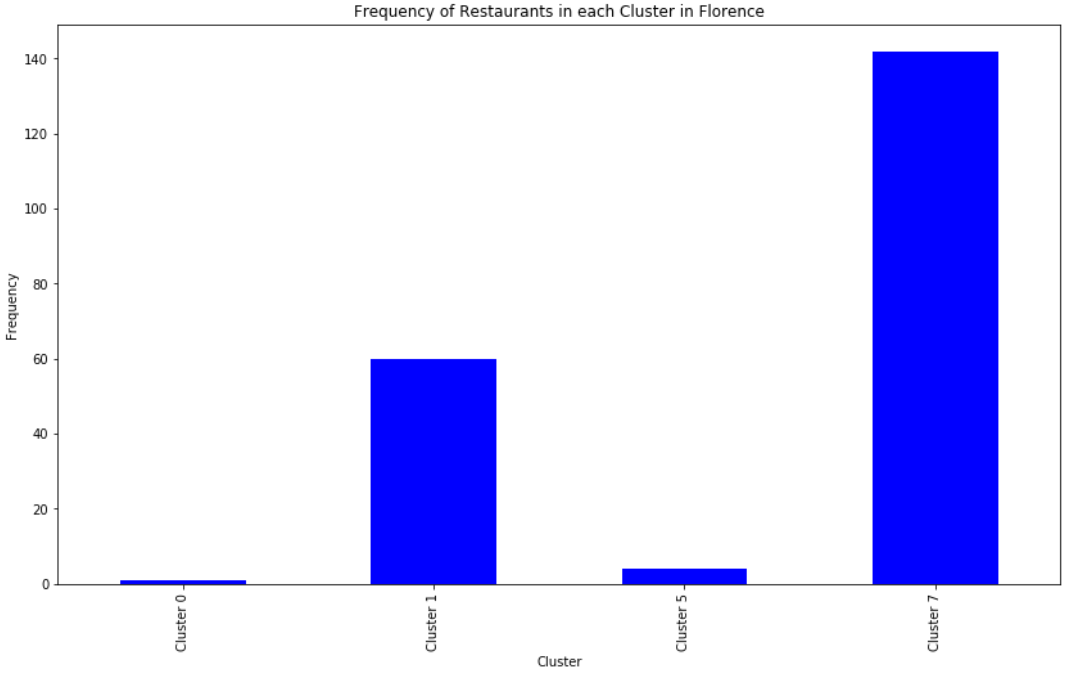

Figure 8: Distribution of neighborhood clusters (according to how many restaurants are in each cluster)

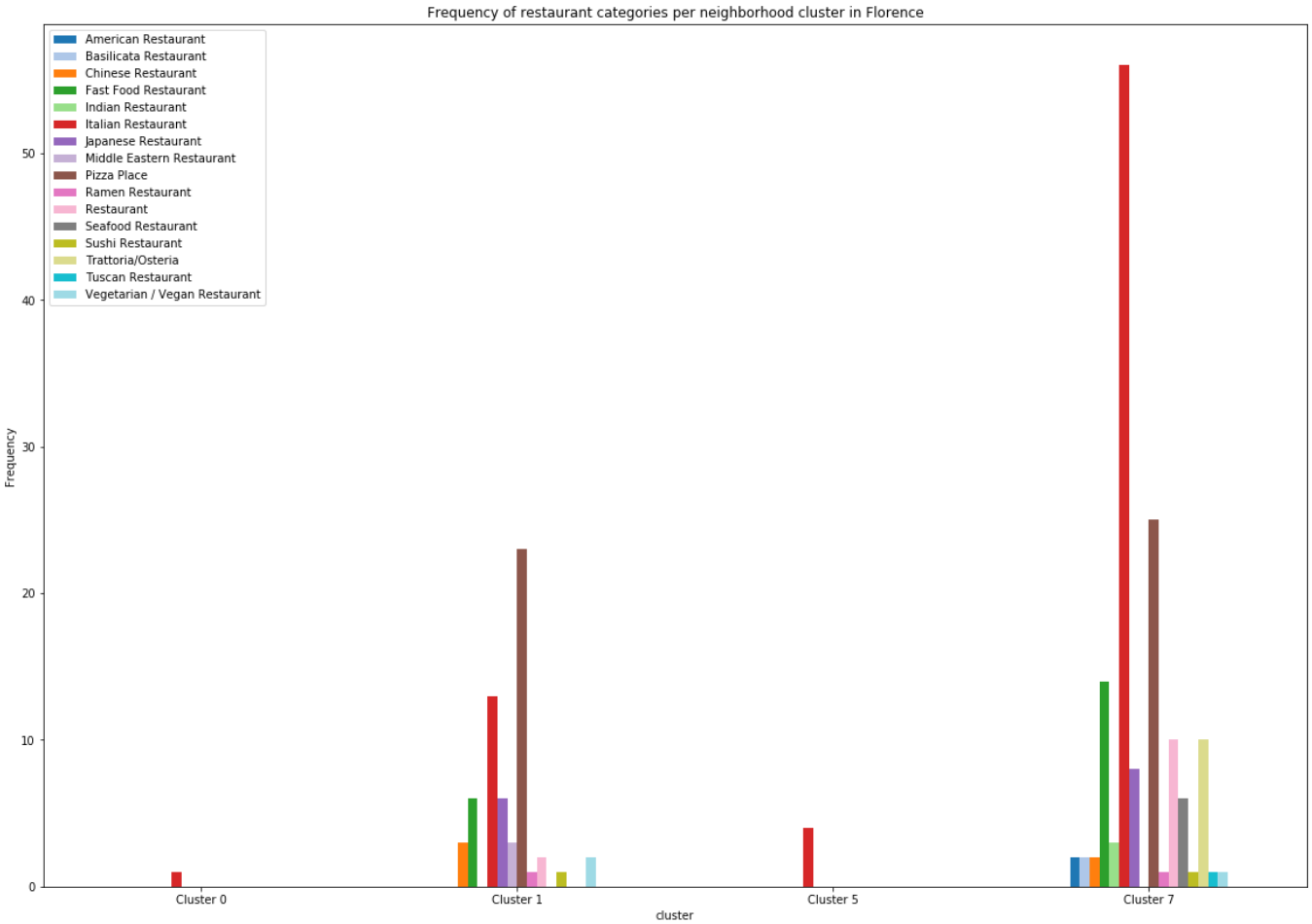

Figure 9: Distribution of restaurant categories in each cluster

From figure 7, it looks like the generic *Italian Restaurant* category is by far the most represented in the dataframe, while many other categories have very few examples. This evidence needs to be kept in mind when interpreting the results, as any model we build with categories as features will risk to overfit given the few number of examples available for certain categories. Figure 8 shows that the majority of restaurants are contained in cluster 1 and 7. This, again, will mean that using cluster 5 and 0 as features might result in overfitting, given the small number of examples in those clusters. In our case, we will use all the variables anyway, but it is important to keep in mind this evidence for later, when we will acknowledge the limits of our analysis and suggest solutions. Figure 9, finally, seems to confirm that what previously observed by plotting the clusters' frequency is true also for the restaurants' categories in the minority clusters: cluster 0 and cluster 5, in fact, contains just italian restaurants, while cluster 1 and 7 seems to have the biggest variety of restaurants' categories. This means that the use of cluster 0 and cluster 5 in a model will most likely have just the same effect of the italian restaurant category and, as such, the features should be redundant. For the time being we leave them there, but, if the results of the analysis will prove unsatisfactory, we can try going back and delete the rows pertaining to the two minority clusters.

At this point, we can use the library seaborn and its convenient function *regplot* to see if we can find any linear relationship between our numerical features and the target variable. If such relation is observed for one or more of the features it might mean that we could use a linear model for our purpose (e.g. linear regression). Figure 10 include such visualisations.

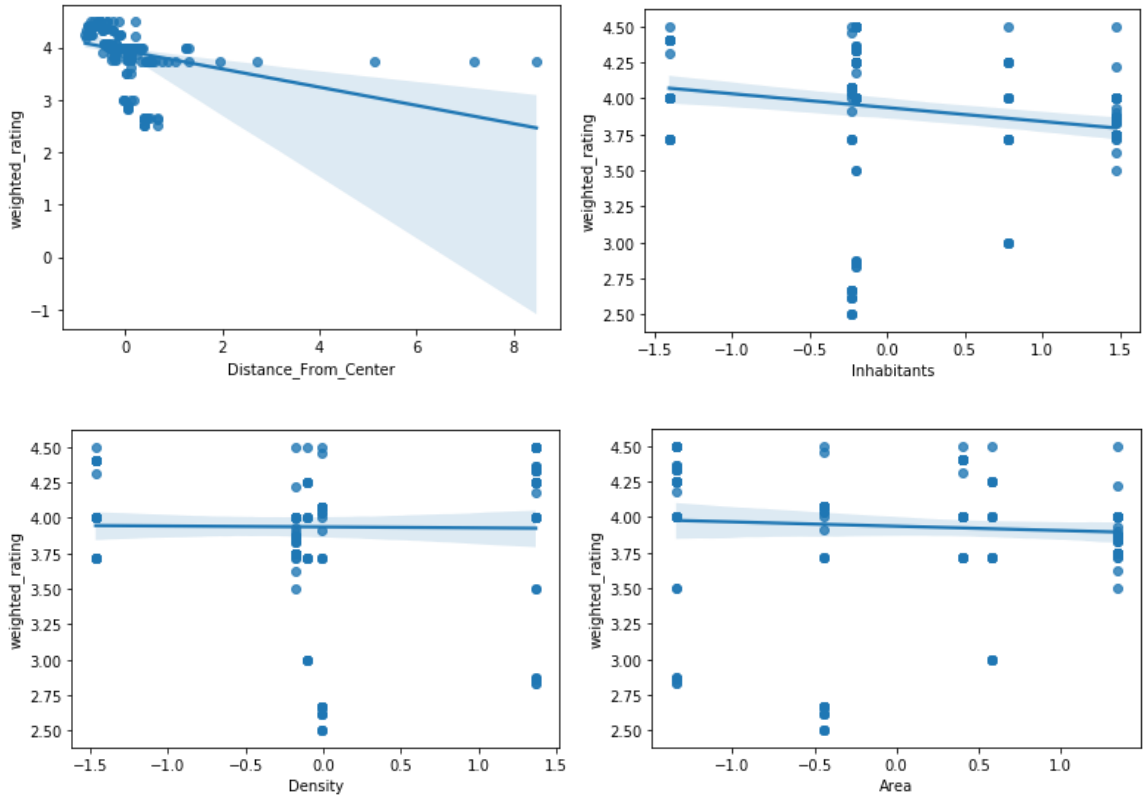

Figure 10: Figure 10: Regression plots highlighting the relationship between each one of the numerical features and the weighted rating of each restaurant.

Just distance from centre and inhabitants seem to have some linear relationship with the target variable, even though the relationship seem all but perfect, as many observations do not follow the regression line. Density and area of the boroughs do not seem to have a clear relationship with

ratings, as the regression line is almost flat. In general, we can remember that the demographic information included as features pertain to the Boroughs, not to the restaurants or even to their neighborhood, therefore such information might be not fine-grained enough to effectively describe some trend.

4.2 Multiple Linear Regression

As a first attempt of building a model, we will build a simple multiple linear regression model. Suc h a model has the advantage of being interpretable via the analysis of the weights of each feature , while the relative speed with which we can build them will allow for us to see how different sets o f independent variables behave in the prediction task. We used the R squared score in conjunctio n with cross validation to evaluate the results for each set of training features. Cross validation wa s built and evaluated different models for a predefined number of times with different subsets of t he data (in our case 4 subsets) while testing on the left out samples: this way we were also be ab le to have an idea of the generalisation power of our models. After cross validating, we also re-tra ined with the entire training set for each separate set of features and report the R squared adjuste d by the number of predictors as an additional parameter to see what training set works the best when using multiple linear regression to predict restaurants' ratings. Full results are reported in th e accompanying notebook. Here it will suffice saying that the X_all training set is the one that see med to work the best for this model with an adjusted R squared of `0.80`, while the cross-validatio n yielded a mean R squared of 0.43.

4.3 Regression Tree

We saw that a multiple linear regression model could perform quite well and gave us interpretable weights to assess the importance of different factors in a restaurant success in Florence. Can we do better? If we think back at the non-linear nature of the relationship between certain features and the ratings, then we might consider also a non-linear model that is easily interpretable as well. Such a model exists in the form of regression trees, a generalisation of decision tree to predict continuous values. Such models can be easily built via sklearn and also have the advantage of given interpretable rules leading to the obtained results: seems exactly what we might want! The algorithm was run for all the different training sets, the same way we did for the multiple linear regression model. Again, the full results are reported in the notebook, while the best model was built once again by including all the features, obtaining an adjusted R squared of 0.99 and a mean R squared after cross-validation of 0.65. Both results seem much more promising than the previous model as well, confirming the assumption we made about the efficacy of this model.

Results and Discussion

By performing all of the above analysis, we were able to create two different models giving good results in the rating prediction task. Starting by those two models we can then interpret their rationale and, consequently, understand the factors of a restaurant success. We did this by performing the following operations:

- create a dataframe of the weights associated with the predictors in the multiple regression model and visualise them: the bigger the weight, the stronger the effect of the variable on the rating (positive or negative according to the sign of the weight).
- create a visualisation showing the decision making process of the regression tree.

Such visualisations are displayed in figure 11 and 12 below.

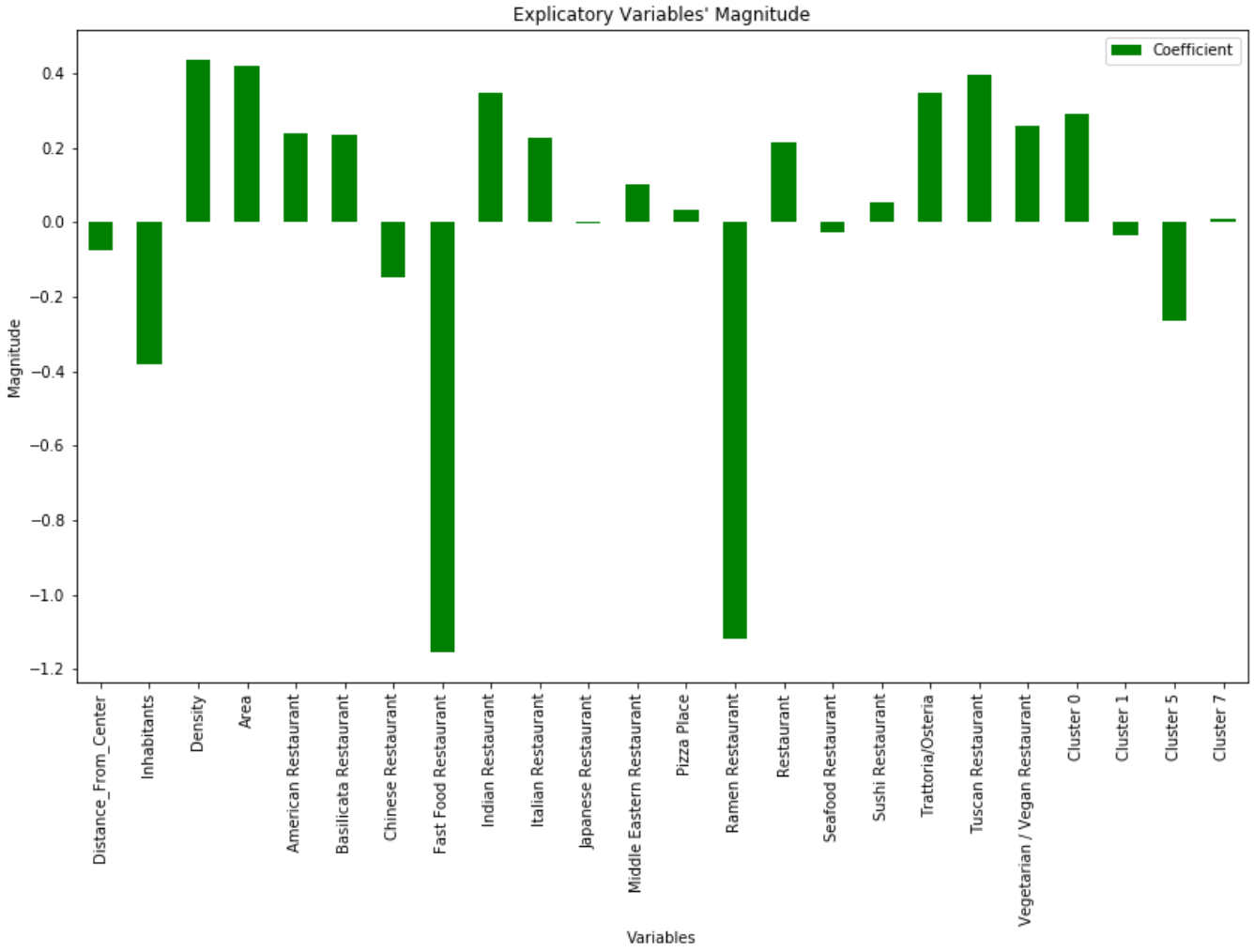

Figure 11: weights' magnitude (named after the variable they refer to) in the multiple linear regression model.

From the above picture we can see the important factors for a success of a restaurant in Florence, according to the Multiple Linear Regression Model. Important highlights include:

- Fast-food restaurants do not seem to be particularly appreaciated in Florence, nor Ramen Restaurants.
- Staying in a densely populated and large enough area might help your restaurant having success, probably because many people are available as potential customer but the competition is more distributed.
- Indian, Italian, Tuscanian Restaurants and Trattorie (typical restaurants) seem to be appreciated type of restaurants.

These results must be interpreted under the light of the performance of the multiple linear regression model, as shown in the previous section, but, in general, can give a direction to people who wants to invest in restoration in Florence.

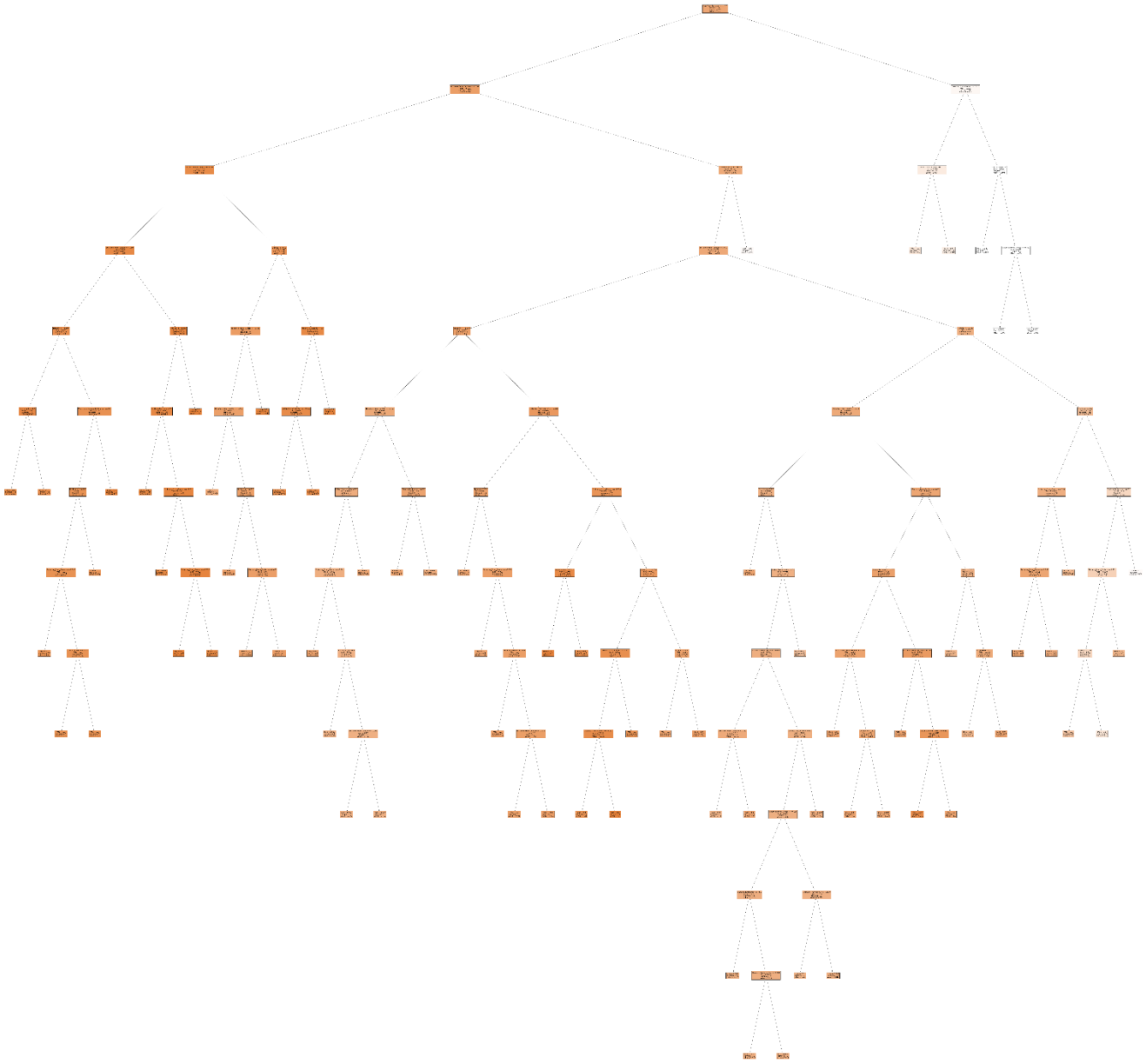

Figure 12: Graph showing the decision making of the regression tree.

The tree is very big and, as such, reading in the various boxes is very hard. By downloading the above image and zoom into it, however, the various factor leading to a prediction can be individuated. In particular, it can be seen how the decisions over dummy variables are represented as *category*<=0.5, meaning *if is not that category*. The results of the regression tree seems to confirm the ones from the multiple regression model in the importance it's given to the various features. In particular, the first split highlight once more and even more strongly that to have success in ratings in Florence you should not open a fast-food restaurant and the same applies to Ramen restaurants.

Conclusion

In this report I have used geolocation data, demographic information and restaurants' categories to build predictive models to predict rating of Florentine restaurant as appearing on TripAdvisor. This way I could open up the models themselves and see what features might determine a restaurant success in Florence, while having a mean of predicting the eventual success (always in terms of rating on TripAdvisor) of restaurant to be opened in the city.

In general, the overall project might benefit from additional data, as it could be seen that certain neighborhoods were under-represented, while demographic data was not fine-grained enough to make it very relevant in predicting a restaurant's rating. The pipeline, however, gave reasonable results and, as such, it could be applied to other cities or replicated for the same city as more data become available, therefore giving increasing value over time to stakeholders.

References

[1] Foursquare API, retrieved at **https://developer.foursquare.com** on November, 11, 2020

[2] TripAdvisor: Best Restaurant in Florence, retrieved at https://www.tripadvisor.co.uk/Tourism-g187895-Florence_Tuscany-Vacations.html on November, 17, 2020

[3] BeautifulSoup, retrieved at https://pypi.org/project/beautifulsoup4/ on November 19, 2020