”The Zillow Prize Challenge”: Analysis and Improvement

Dibyajyoti Pati, Mohaddeseh Bastan

Stony Brook University

Abstract

In this investigation, a detailed analysis was made to understand if there exists any uncovered latent relation of property with any external factors. The machine learning pipeline comprised of aggregating of data from various **external sources** that is relevant to property and its cost, the features derived in the above step was used to score and rank the properties to understand the **likeability** of a property. This step was quite vital as it gives us an understanding of how far are we from the ground truth in understanding the property as a function of given variables. The ground truth here comes from the actual cost of the similar properties in & around the location of the ranked property. This was followed by an exploratory analysis of the features to determine the similarity of features using **K-Means Clustering**. The distance metric used in this step was the normal **Euclidian Distance** between features. And the next stage of the pipeline involved model creation using the original features and features derived in above steps & Ultimately validating the genuineness and significance of the model using **Permutation test**. During this course, we used k-fold cross validation using k=200 on an Extra Tree regressor. The use of k-fold validation method brought significant strides in the model. All codes and the comments in the notes including important notes are in the notebook 'HW3.ipynb' as markdown text . Other files in the git account are the output of preprocessing, making model, merging with another dataset

Keywords: Data imputation, Xtra Tree Regressor, k-fold validation, K-Means, Clustering, One Hot Encoding, P-test

Task 1: The Scoring Function

The scoring and ranking of a property for likeability was initially determined using few of the actual features from the dataset. The features that were included comprised of the area, number of rooms, number of bathrooms, year of build etc. The decision for using these features was based on our previous knowledge of property and its cost. But in this process, it was determined that a better and accurate ranking can be determined if we consider the most vital factor included in the dataset that is the location of the property (data like coordinates, fips, region id, county etc.). This decision lead to exploring of external features that would provide the value of location. And as a result, the task of ranking was performed after including the external data of School location, crime count at a location and minimum distance to the crime (details of these feature in the external data section).

The ranking function used for this purpose is:

Ranking Function(R)= InverseCrimeCount^4 + MinimumSchoolDistance^4 + Build Quality + Total Finished Area

The ranking function was kept simple albeit effective. The ranking considers the Inverse of Crime Count in the nearest crime zone of the property, Minimum School Distance from the property, Build Quality and total finished area. The equation comes from the fact that it was assumed that the most important factor for buying a property is the how far is it from the nearest crime hot zone and how many incidents of crime are recorded in the neighborhood. At the same time, it is vital to living in a close proximity from the schools which is empirically proven to be a driving cost for property and hence the likeability of that. We considered that the likeability score would depend on the forth degree of these two factors and at the same time depend linearly on factors like the area of the property and Build Quality. Here we made some simple assumptions for filling out the missing data, the consideration was newer houses are generally of better build quality than older houses, defining a linear relationship between quality of house and year it was built.

After completion of this task, we did a check to understand how far our estimation of likeability coincides with the actual likeability(Price) of the property. So, we went ahead and found the average cost (from Zillow website) of the property in the neighborhood of our top 10 best properties. And indeed, they were some of the costliest real estates of the country. Our ranks also included some the posh commercial properties.

Task 2: Distance function

For this task, we define a distance function which would be best for the features we are considering for clustering. For calculating the distance between properties, we have different approaches. As there are some categorical and some numerical features, we need to process the categorical to be able to use in calculating the distance function. A simple way is to add a binary variable for each class/category as a new feature and if a property value for that categorical feature is of that type, the corresponding binary column would be 1 otherwise that would be zero. But this makes the dimensional of the dataset big at the same time it is sparse. So, the better work is to decide whether it is necessary to add that categorical type as a new feature or not. To do so, first for each categorical feature we count how many of each type is in the dataset. Then by looking at the numbers, we decide whether to add that type as a new binary feature or not. For instance, 'airconditioningtypeid' feature there are 744291 properties which have a value of 1,58487 have a value of 13, 8795 have a value of 5, 1818 have a value of 11, 59 have a value of 12,7 have a value of 3 and 2169855 have a value of NaN. So, we decided to make 4 new binary features for this categorical variable (considering only the classes that have the highest representation). First is for 'airconditioningtypeid' equal to 1, second for 'airconditioningtypeid' equal to 13, airconditioningtypeid' equal to everything else except NaN & finally airconditioningtypeid' equal to NaN. Instead of 'heatingorsystemtypeid' we created 6 binary features, one for the row which has a value of 2 for 'heatingorsystemtypeid’, one for 7, one for 6, one for 24, one for other non-NaN values and one for NaN values. Instead of 'propertylandusetypeid' we create 8 different new features, one for row has value of 269, one for

value of 261, one for value of 266, one for value of 246, one for value of 247, one for value of 248, one for value of 263 and one for other values. And because a number of Nan's are too high, we disregard other features. We did the same for other categorical features. If a feature is sparse we disregarded it.

We defined a pairwise distance function as follows:

1. For numerical feature calculate Euclidean distance
2. For binary features calculate inner multiplication.

For example, if two records have 10 binary features as same, their distance is 1/10 By normalizing the numerical distance and adding the binary distance to it we defined the whole distance function. Which is not only simple but also considers all features and is very precise.

Task 3: Clustering:

Initially, we used fcluster with our predefined distance function and clustered a small sample of our dataset hierarchically. This function helps us to use our distance function (other clustering function does not let us define our distance function) and decides what is the best number for a number of clusters. This method inherently came with its own drawback, like huge resource requirement & slow convergence. So, we modified our Distance function to use Euclidian Distance and used K-Means algorithm instead.

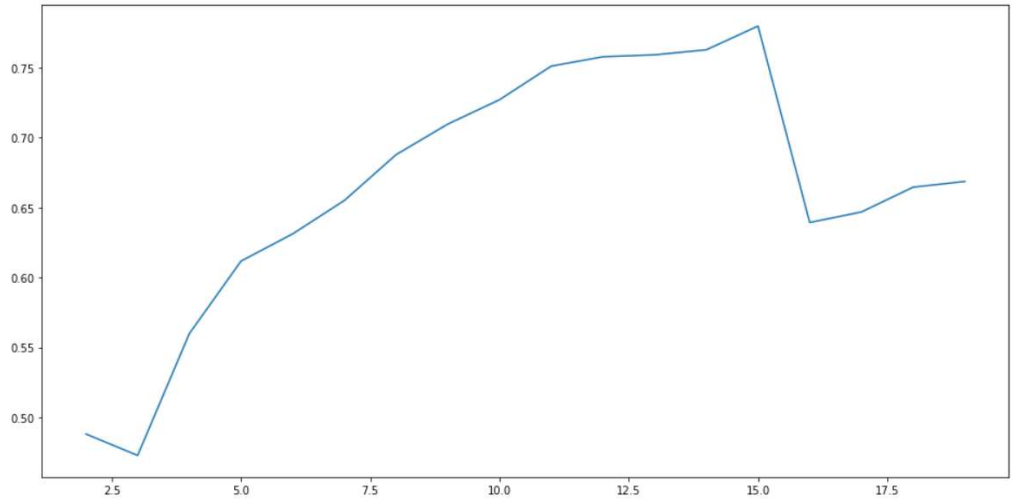
We used k-means clustering on the following features:

1. Derived: minimum distance from school & crime and maximum crime at the location
2. Few of the property type formed to binary variable where the property was Single Family Residential, Condominium, Inferred Single Family Residential, Duplex, Planned Unit Development; The purpose of this being they have maximum number of non-zero values, i.e. most of the properties belong to any of these categories
3. Some of the numerical like features finished square feet, room count, tax value assessed and the tax amount collected for the property.

For finding the best number for clusters we used the Silhouette score. Silhouette refers to a method of interpretation and validation of consistency within clusters of data. The silhouette value is a measure of how similar an object is to its own cluster (cohesion) compared to other clusters (separation). The silhouette ranges from −1 to +1, where a high value indicates that the object is well matched to its own cluster and poorly matched to neighboring clusters. If most objects have a high value, then the clustering configuration is appropriate. If many points have a low or negative value, then the clustering configuration may have too many or too few clusters.

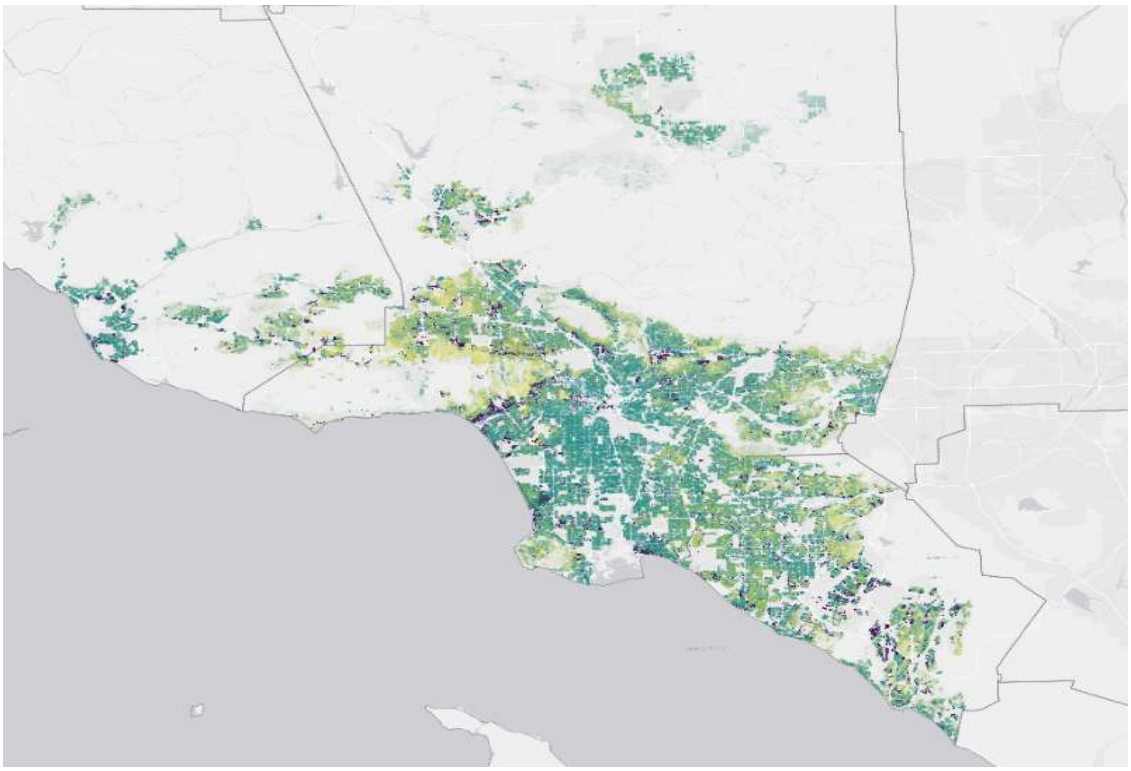
Silhouette Score Analysis: Silhouette score analysis the clusters based on how similar they picked clusters are with respect to other clusters i.e. A silhouette score of -1 indicates that the picked sample belongs to another cluster and a value of plus 1 indicates that the samples are not strongly related to cluster, whereas a value close to 0 indicates that the samples in the cluster are strongly related to each other and have a very low likelihood of shifting to another cluster on starting at another init.

Number of Clusters Vs Silhouette Score



A number of clusters were analyzed till n=20, the two driving factors for this analysis were:

1. Understand the silhouette score
2. Maximize the distribution and prevent skewing of clusters.



Silhouette score has a minimum value at cluster=3

There is a distinct separation of three counties based on the cluster even though the clustering involved no parameters related to a position directly. The result of clustering based on features is in the notebook.

Task 4: The External Variable:

Integrating school location data:

In this section, we tried to integrate data pertaining to the locations of schools with respect to the property. There was no segregation of property on the property type, rather the distance was determined for all parcel ids. The candidates for data integration were the distance from top Private schools and top public schools. The ranking(top) of a school was determined based on many enrolled students as the average SAT score for the selected school wasn't available for ranking purpose. The data used for this purpose was assembled from California dept of education and Eric Hurst's gist (git link) which in turn was collected and composed from US Census data. In this attempt, we have integrated the database of private schools only (with the assumption that private schools are more sorted after than a public school and thereby its location would have more impact on the desirability factor). The data from the private school database was sorted and filtered with county code filter and the top 15 schools from each county were selected. The selected data was then mapped to the zip code to Lat/Lon mapping file by performing an inner join on the zip code (which is available for both). This code is submitted as part 'SchoolLocations' notebook uploaded in git. This resultant of this operation was a database with

schools in each county and their respective coordinates (the assumption being, as these are densely populated regions the average error between the actual coordinate of the school and the average coordinate of the zip code would be relatively small) The database obtained in the above step was used to determine the distance between the property and each of the fifteen top schools in each county. The haversine method or flat earth method was used for this purpose. The distance computation was done at county level i.e. properties belonging to LA were matched with schools in LA to avoid humongous computations. The resultant of this was a distance parameter for each property representing the nearest private school distance. The resultant dataframe was saved with parcel id, minimum school distance, and coordinates for later consumption by performing join operation over the parcel id

Integration of Crime data

A dataset from Open justice was used to arrange and integrate the crime statistics. The Crime statistics record came with two caveats, it had recorded crime for all the years from 1985-2016 overall crime codes/NCIC Codes, they had to be filtered by year and a summation was applied to all crimes at a given location for a given year. This dataset was not directly mapped to a zip code or coordinate, but a rather to a region. Which in turn had to be mapped to a zip code and zip code to coordinates. The purpose of this dataset was to collect two types of statistics

1. What is the crime rate near a given property
2. How far is the nearest crime reported from the property

For simplification If a region had multiple zip codes, then average crime of all zip codes for that region was assigned to that region instead of maintaining multiple zip codes for a single region. Following a similar approach of joining and mapping, we finally reached a database with each property's parcel id, the recorded number of crimes in the nearest crime location. This data provides us with a very sophisticated method for analyzing a property based on its location which in fact is the most valuable attribute of a property.

Task 5: The Better Model:

We used different methods for this task. The first one was a linear regression that we used in previous homework. It is simple, But the result was not very good. So, for this, we decided to make use of our new features and use a random forest to see whether the model gets better or not. Also, the dataset is different from the previous task. The main difference between 2017 and 2016 datasets is that the first one contains the information about the properties which has been sold in 2017. Therefore, the log error and transaction dates are very different. But some other small features are also different. After analyzing the datasets, we saw that some tax features have changed too. The value of R^2 (the score of our model) was completely different from the value for 2016 dataset. The 2016 dataset has a range of 0.009 and this dataset has ranged around 0.14 and initially, the thought was the model is better but after uploading at Kaggle we saw that our score in test sets decreased. So, we decided to use other methods for solving this problem and predicting the log error. The other attempt was to use our new features instead of the usual one.

We changed the categorical features into binary and used them for calculating distance function. We decided to use these new features and random forest but the result didn't get better. The other **interesting thing** we found is that when tuning the parameters of random forest model, we found that increasing the depth of the tree is not improving the model, however, it increases the complexity of the program. It may be because of the types of the data we have that the depth has the effect to some degree. Also, we used another method for solving this part, we used a neural network. We created a simple MLP and tried to tune the hyperparameters (number of layers, number of node in each layer, activation function, solver method and alpha, beta parameters in adadelta) but none of them improved the model.

Next, we have explored other ensemble models in an attempt to improve the performance. After trying out multiple methods, we picked out extra tree regressor for this problem. The reason being the best score in validation set. Extra tree regressor stands for extremely randomized trees and in a way like a random forest but uses optimal cut points instead of bootstrapping for introducing the variance. The performance of this model was further improved by trying out multiple values of tuning parameters like, number of estimators: which is the number of trees to use to make the prediction (the output being an average of these values) and the maximum depth of the tree. We tried our hands-on k-fold cross-validation method, and it certainly improved the model performance. K-fold cross-validation divides the train and test set to k- different subsets and uses k-1 sets in each iteration for training and just 1 set for validation purpose. This randomizes the train and test set in total of k- iterations. We observed that increasing the number of folds improved the **MAE** score to an extent, after which the increase in k- has no impact.

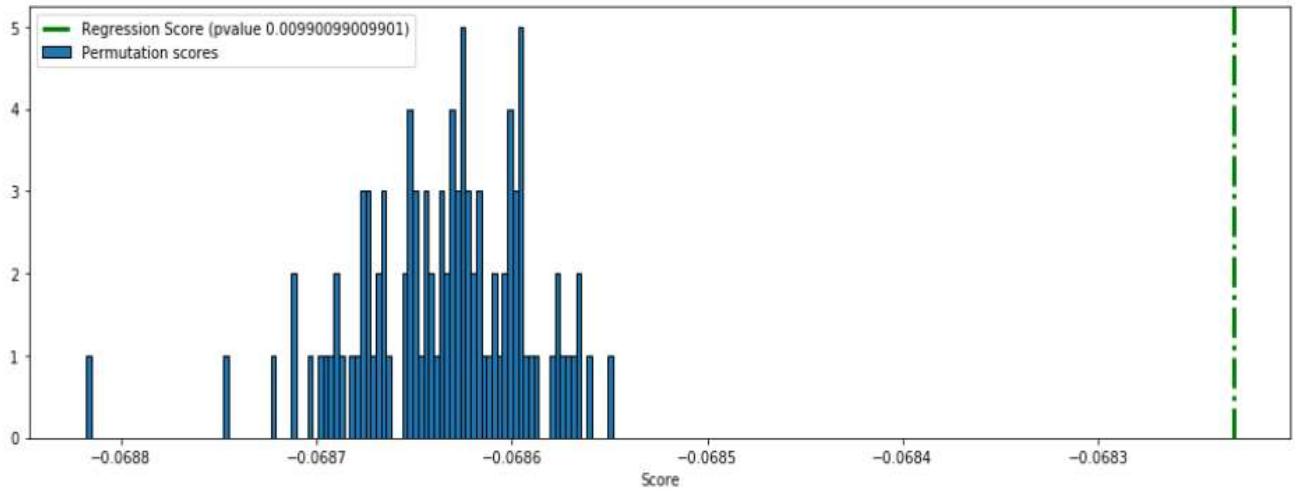
The values of k were tried from 20-500, and the optimal value was found to be at 200

Task 6: P-Test:

Evaluation of model on p-test: we used sklearn' s permutation package to evaluate the effectiveness of the model. This used 100 permutations tests to determine the significance of the model, the model used in this case was the best model derived by extra tree regressor in the steps above it, to break the tasks, the model was pickled to file and read later.

We used a metric of 'Negative Mean Absolute Error (MAE)'(As MAE is not available in the p-test library) which is same as the metric used while evaluating the performance of our model in the above steps. We observed that for all 100 different combinations the MAE was less than the MAE when ran on the original set.

The outcome of this result was plotted as a histogram the magic number 0.00990 for the significance value comes from the fact that the input model was better on actual target than all the permutation inputs than provided to it



As seen the model prediction is seen to have outperformed all other random permutation tests performed, its affinity to the target variable is evident from the fact that the average mean absolute score reduces when applied on random shuffled target.

Task 7: The Kaggle score:

With the new model of Extra Tree Regressor and maintaining two different models for 2016 and 2017, we could make our model better and the final score was 0.0648558 and the final ranking was 2377.