

ASSIGNMENT NO 3

Report on the analysis of each task

Shruti Nair SBUID: 111481332
Rohan Vaish SBUID: 111447435
Kiranmayi Kasarapu SBUID: 111447596

Task 1

Scoring Function

Here we have ranked all the properties based on the scores calculated for each one of them and then found the top 10 most desirable properties and top 10 least desirable properties. The assumptions made/factors considered for the same are as follows:

Note:

1. All the features values have been Z-normalized. Power (square or unsquared) signifies the degree of impact that feature will have on the final score. Weight (negative or positive) signifies the directional impact on the score

2. Positive weight will add up to the score to increase the rank of a property while a feature with negative weight sabotages the desirability of the property, hence decreases the score

1. Unsquared Features with weight ≥ 1
 - a. Bedroom count $w=1$
 - b. Bathroom count $w=2$
 - c. Garage car count (capacity) $w=1$
 - d. Number of stories $w=1$
 - e. Unit count (Duplex/triplex..) $w=2$
 - f. Year Built $w=1$
 - g. Finished total area $w=1.5$
 - h. Assessed value of the built structure on the parcel $w=1$
 - i. Assessed value of the land area of the parcel $w=1$
 - j. Tax amount for the assessment year $w=1$
 - k. Building Quality $w=2$
 - l. Building Type $w=1$
2. Unsquared Features with weight < 1
 - a. Fireplace count $w=0.5$
 - b. Heating system type $w=0.5$

- 3. Squared Features
 - a. Crime counts $w=-1$

Task 2

Designing a pairwise distance function

Euclidean

Standard Euclidean function which find the sum of squares of distance between 2 feature vectors

Custom Pairwise Distance function

In this function, we calculated various parameters that could be used:

1. Physical distance between 2 houses using latitude and longitude
2. Difference between zip codes, tells us how close the houses are physically.
3. We calculated the difference in the square feet between 2 houses. If a house has more than 1 unit and many stories, the finished square feet count will be a factor of the units and stories, since the total area covered on ground will be same for 1 storey or many storey building.
4. absolute difference between the years when both the houses are built
5. Tax amount is calculated per square feet area. Large houses with more square feet area would definitely pay higher taxes. But if a particular place has same tax for per square feet, then the houses are likely to be similar in my opinion.
6. The number of crime rates between the places where both the houses are located.
7. Finally, we gave some weights for each of these factors according to which will weigh more and calculated the similarity/dissimilarity between the houses. If the distance returned is more than the houses are dissimilar. If there are close to 0, then they are similar.

Compute Euclidean distance and cluster based on them

Here we use KMeans clustering to cluster the houses into 90 clusters. The KMeans algorithm by default uses Euclidean distance for computing the similarity between the houses. Means algorithm works as follows:

1. It is a machine learning algorithm which clusters the data points into various clusters(classes)
2. Clustering is done based on similarity function which we call distance function
3. KMeans by default uses Euclidean distance to measure the similarity between 2 data points
4. Data points are clustered based on the feature similarity

5. The number of clusters is decided by the user itself, defined as "k"
6. KMeans is an iterative algorithm which assigns each data point one cluster based on its similarity with the other data points in the cluster
7. Finally, it provides with a centroid for each cluster, basically a vector of features, whose distance with all the data points is as minimum as possible.
8. The centroid of each cluster can be used to identify the cluster for a new data point

Task 3

Plot the clusters on a map based on location

We can see the following observations from the below plot:

1. Most of the houses got clustered based on their physical distance. The houses that are close physically or belong to the same location are clustered together.
2. Since Euclidian distance, considers all the features that are provided to it in calculating the similarity, we can say that probably the tax amount that is paid by each of the houses in a given cluster could be same. We can assume this because tax amount paid by individuals belonging to the same location is usually same.
3. Since the houses in the same location are clustered together, we can say that their ZIP codes are not very much different which is usually the case.
4. Towards the extreme right we can see that houses far apart are also clustered into the same cluster (Pink and Dark green). This tells us that, even though location of the houses is not same, they are other factors that might be similar with respect to those houses. Probably all these houses have the same room/bathroom/unitent which is playing important role. They may have been built in the same and have the same square feet area.
5. We can see a clear distinction between each cluster and not many outliers being present.
6. In my opinion KMeans has done good clustering for this dataset for the features we used

Task 4

For Integration of new dataset, we used crime data

We obtained the dataset from the following website: <https://data.lacity.org/A-Safe-City/Crime-Data-from-2010-to-Present/y8tr-7khq> The data is specific to Los Angeles. It has data of all the crimes that have occurred in a particular locality from 2010 to present data. It is updated regularly

The data has location in the form of (latitude, longitude). Here again we have converted latitude, longitude to ZIP code and created a new column in the dataset with ZIP code

We calculated the number of crimes that have occurred at a particular zip code from 2010 to present. We used this information and integrated it with our housing dataset. Since both the datasets have ZIP code available, we created a new feature in the housing dataset with crime counts and populated it with the data from crime data

The process followed for Data Integration of 'properties_2017' and 'crime_data_with_zip' occurs as follows:

Step 1 - Collection of Crime Data California from Data.gov.

Step 2 - Unmasking the zip code using the 'latitude' and 'longitude' variables. The API used for this purpose is uszipcode.

Step 3 - Once the zip code is un masked, it is merged with the crime data using zip code as the identity key. If the zip code is not present for a certain city, it is replaced with NAN value.

Step 4 - The last stage includes writing the zip code for all the property values. It is achieved using filling the NaN values using mode.

Crime occurring in a particular region helps in determining whether the place is suitable for constructing houses or not. If there are more reporting's of theft/kidnapping/burglary in a particular zip, it indicates that there are more expensive properties concentrated in that area. Hence, it indicates high prices of properties for that specific area.

The above graph represents how crime is related to the age of the victim. It is clearly seen here that, the victims here are majorly between 10-30 years of age. This can be interpreted as:

1. The crime rate occurring area also has more no of educational schools, given the victims are mostly children.
2. The possible crimes include kidnapping and burglary.

3. The segment of retired people living in that area is very few, thus we can interpret that, the properties are mainly focused in lush cities with great apartment costs.
4. More reporting's of crime on an area therefore means rich estates with great apartment cost

Task 5

KMeans clustering

We have applied KMeans clustering on the integrated data which contains crime_counts information.

We clustered for sample set of 1000 entries to see verify how it is integrated. We were able to create the data points on a map and listed the features for each house. Upon hovering on the house, we can see the features for each house. The map is somehow not visible in the GitHub, but if you compile it, it will be visible in the notebook. For the same reason, I have attached a screenshot of the map. Also, I have attached the html map with this homework. Please take a look at files datamap_custom.html and datamap_eucledian.html to see the feature similarity between houses belonging to the same cluster

I tried to gather features of sample of houses belonging to a single cluster:

Cluster A

1. ID:17276347.0 Stories:1.0 Year:1960.0 Tax:1269.22 SqrFt:3280.0
2. ID:17214819.0 Stories:2.0 Year:1987.0 Tax:4477.28 SqrFt:2754.86923259
3. ID:17226677.0 Stories:2.0 Year:1972.0 Tax:1826.76 SqrFt:2754.86923259
4. ID:17202298.0 Stories:2.0 Year:1997.0 Tax:3484.22 SqrFt:2754.86923259

Cluster B

1. ID:17057038.0 Stories:1.0 Year:1954.0 Tax:1610.66 SqrFt:2754.86923259
2. ID:17061147.0 Stories:1.0 Year:1955.0 Tax:6142.48 SqrFt:2754.86923259
3. ID:17101878.0 Stories:1.0 Year:1955.0 Tax:5092.26 SqrFt:2754.86923259
4. ID:17078624.0 Stories:1.0 Year:1927.0 Tax:4316.0 SqrFt:2754.86923259

We can clearly see that cluster B has houses that built before 1960 and most of them have square feet 2754, Cluster A has somewhat recent houses and have number of storeys as 2 in most of them and so on.

Compute pairwise distance using the custom function defined and use the distance matrix to compute clusters

Here again I use the KMeans clustering algorithm, but the distance matrix is precomputed. The distance matrix is computed using the custom function I have defined above. The function seems to have worked good. I have sampled for 1000 data points. We can see that few of the clusters are

distinctly separated based on their location. Some clusters are spread across different locations. This could happen for many reasons. The following lists them:

1. Houses belonging to the same location may have different stores and units.
2. As some weight is given to the year in which the house is built, it is possible that there are houses which are old and new in a particular region.

Sample houses properties from 3 clusters are shown below:

Cluster A

1. ID:17161984.0 Stories:1.0 Year:1961.0 Tax:1183.08 SqrFt:2754.86923259
2. ID:17158741.0 Stories:1.0 Year:1987.0 Tax:6580.52 SqrFt:2754.86923259
3. ID:17076853.0 Stories:1.0 Year:1926.0 Tax:871.86 SqrFt:2754.86923259
4. ID:17092083.0 Stories:2.0 Year:1986.0 Tax:6384.54 SqrFt:2754.86923259

Cluster B

1. ID:11732547.0 Stories:1.0 Year:1925.0 Tax:3695.84 SqrFt:4320.0
2. ID:11829547.0 Stories:1.0 Year:1959.0 Tax:5716.15 SqrFt:2156.0
3. ID:12366747.0 Stories:1.0 Year:1939.0 Tax:6546.14 SqrFt:2029.0
4. ID:10933547.0 Stories:1.0 Year:1955.0 Tax:6773.34 SqrFt:2754.86923259

Cluster C

1. ID:11229347.0 Stories:1.0 Year:1955.0 Tax:174.21 SqrFt:2754.86923259
2. ID:13119747.0 Stories:1.0 Year:1977.0 Tax:305.16 SqrFt:2754.86923259
3. ID:13119347.0 Stories:1.0 Year:1982.0 Tax:169.02 SqrFt:2754.86923259
4. ID:13118947.0 Stories:1.0 Year:1984.0 Tax:202.01 SqrFt:2754.86923259

Building a Model for the dataset

I have used the following procedure for building my model

1. Create 60 clusters using KMeans clustering for all the data points in the dataset.
2. Extract the houses for each cluster.
3. Extract the data points for each cluster for which log error is given. Divide it into train and test sets.
4. Build a model for each cluster.
5. I have built 2 models for each cluster, Linear Regression and K-Nearest Neighbors
6. Train the model with the data for which log error is provided.
7. Whichever model is giving the least mean-square error in a given cluster, that model is used for predicting log errors for that cluster.
8. Finally collect the predicted values for each cluster and calculate the mean-square error.
9. Using this model, we have got a Mean square error of 0.0210088855757.

Why this model works.

1. Since clustering is already done, we know that, each cluster has houses which have similar features.
2. Since the sale price for the houses is not provided, we can assume that the prediction of the prices belonging to a single cluster might also be similar.
3. Going by the above argument, it is possible that the actual sale of the houses belonging to a single cluster might also be the same.
4. So, the Zestimate i.e. log of the difference of the actual and the predicted price, will also be similar for all the houses belong to a single cluster.
5. Therefore, using different model for each cluster will probably yield better prediction of logerror values.
6. This model has worked better than the previous models that we have built in Homework2 in terms of mean-square errors.

Task 6

Permutation test







For this test, we have done the following

1. We used the above described model for predicting log errors.
2. We train the data for given set of X_{train} and respective y_{train} (log errors)
3. We predict log errors for X_{test} data.
4. Now we permute the y_{test} values and calculate the mean square errors, between the predicted and the permuted values.
5. We do these 500 times and keep all the mean square values.
6. We plotted a histogram, showing the range of mean square error and the number of permutations that resulted in those mean-square errors.
7. The vertical line at 0.021, shows the actual mean square for the correct values.
8. We can clearly see that for a very minute fraction of the permutations, the mean-square error has surprisingly come out to be lower than the actual one.
9. But for a large number of permutations, the error was sufficiently larger than the actual value.
10. We can safely say that our model is simply not guessing any random values, but is actually predicting the log errors to some extent
11. For 3 permutations, out of 500 we got mean square error less than our actual model

Task 7

Screenshot of Kaggle Submission

55 submissions for Team Sea Wolves		Sort by	Most recent ▼
All Successful Selected			
Submission and Description		Public Score	Use for Final Score
predicted_dataframe10.csv 2 minutes ago by Kiranmayi add submission details		0.0652741	<input type="checkbox"/>
predicted_dataframe9.csv 27 minutes ago by Kiranmayi add submission details		0.0653513	<input type="checkbox"/>
predicted_dataframe8.csv 39 minutes ago by Kiranmayi add submission details		0.0654314	<input type="checkbox"/>

2641	▼ 113	flysoon		0.0649972	2	11d
2642	▼ 113	malugina		0.0649975	4	2mo
2643	▼ 113	Team Sea Wolves	  	0.0649976	42	2h
Your Best Entry ↑ Your submission scored 0.0880324, which is not an improvement of your best score. Keep trying!						
2644	▼ 113	ChiYuan		0.0650004	3	3mo