CSCE 411/811 Assignment 3 - Data Analysis

Dat Nguyen
UNL CSCE 411
datnguyen5653@huskers.unl.edu

Shivani Tamkiya
UNL CSCE 411
shivanitamkiya@gmail.com

Shea Winkler
UNL CSCE 811
sheawinkler@gmail.com

Abstract*—This paper discusses the third assignment of CSCE 411/811 in which a variety of data analysis methods are tested and discussed. Data analysis approaches including K-Means and DBSCAN cluster-based analyses, and outlier detection utilizing Local Outlier Factor, Isolation Forest, and Gaussian Mixture Model approaches. In the review of these approaches for data analysis, we develop a DBSCAN algorithm from scratch that is index-based to improve the time-complexity. Throughout the rest of this paper we investigate these data analysis tools using a variety of tools from the reputable Sci-Kit Learn Python module.

I. Introduction (*Heading 1*)

Data analysis is a crucial aspect of the data management process that allows users to gain insights into their data and potentially recognize unexpected patterns. Businesses can act on these patterns through selective advertising or creating new revenue streams.

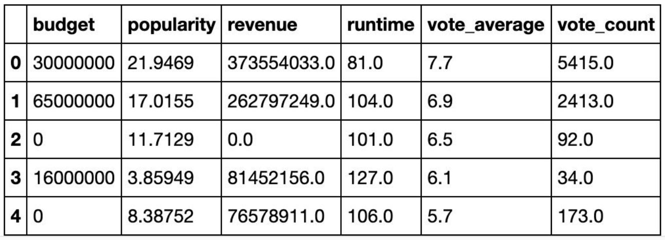
To start we will describe the data sets used during our tests of the data analysis clustering and outlier tools that we explore. Then we will go further into detail for each analysis tool we tested and review their results.

II. Data sets & Pre-Processing

A. *Movies_metadata.csv*

From the original dataset we chose numerical columns that would enable easy normalization and analysis. The following preprocessing techniques were applied:

1. Removes row with null or N/A values
2. Filter out only movies with more than 30 votes
3. Standardize the dataset using sklearn.preprocessing.StandardScaler



	budget	popularity	revenue	runtime	vote_average	vote_count
0	30000000	21.9469	373554033.0	81.0	7.7	5415.0
1	65000000	17.0155	262797249.0	104.0	6.9	2413.0
2	0	11.7129	0.0	101.0	6.5	92.0
3	16000000	3.85949	81452156.0	127.0	6.1	34.0
4	0	8.38752	76578911.0	106.0	5.7	173.0

Figure 1: Subset of movies dataset

B. *NHL-2003-2004.csv*

The dataset is retrieved from nhl.com. To simplify the process and cover up the limitation of ISO on curse of dimensionality problem, we split the data into 2 feature set:

1. Feature set #1
   - Game played (GP)
   - Goals scored (G)
   - Shooting percentage (S%)
2. Feature set #2
   - POint scored (P)
   - plus-minus statistic (+/-)
   - penalty-minutes(PIM)

We remove non-numerics values(e.g "--") as part of the preprocessing step.

III. Evaluation metrics

In this section of the paper we will detail the various data analysis methods that were tested and measured.

A. *Silhouette score*

An instance's silhouette coefficient is equal to (b-a)/max(a,b) where a is the mean distance to other instances in the same cluster and b is the mean nearest-cluster distance, that is the mean distance to the instances of the next closest cluster.

- A coefficient close to +1 means that the instance is well inside its own cluster and far from other clusters.
- A coefficient close to 0 means that it is close to a cluster boundary.
- A coefficient close to -1 means that the instance may have been assigned to the wrong cluster.

B. *K-Means*

The K-Means algorithm is $argmin_c \sum_{j=1}^{k} \sum_{x \in c_j} d(x, \mu_j)$, where k is the number of clusters, c_j is the set of points that belong to cluster j and μ_j is the centroid of the class represented by c_j .

Two evaluation metrics were used for determining the number of clusters for our K-Means algorithm to find. The elbow method for finding optimal number of clusters using the inertia metric of a fitted K-Means model. This method finds the number of clusters before which the change in the sum of squared differences is at its maximum. On the graph it was obvious that after 2 clusters the inertia begins dropping at a slower rate. The number of clusters was confirmed by our second evaluation metric, the silhouette score, as described above. We used this as a way to check our elbow analysis because, especially because the inertia rate stayed relatively high until about 5 clusters. After trying out the different metrics we saw that all of their results agreed with 2 clusters being the optimal number.

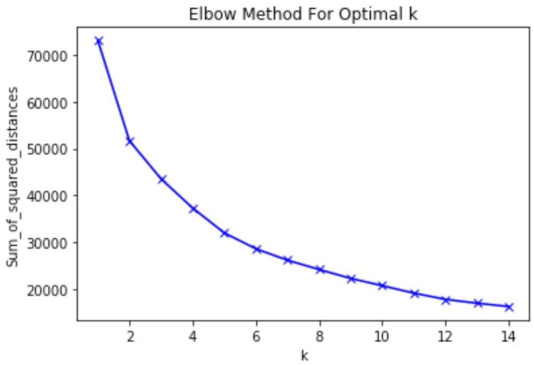


Figure 2: Elbow method visualization for movies dataset

C. DBSCAN

Parameters searching

The knee method was first applied to find optimal eps. We also applied parameters searching to find optimal values on following parameters range

eps = [0.1, 1.0, 2.0, 3.5, 5.0, 10.0]
min_samples = [4, 9, 15, 30, 50]
distance_metric = ['euclidean', 'manhattan']

Silhouette score was the metric to evaluate the quality of the cluster.

Using the Silhouette score to find optimal value among a range of parameters ended up providing a better result.

C. Local Outlier Factor (LOF)

Choosing optimal k value from a range of k.

Intuitively, the range of k is found by determining the minimum number of outliers that one datapoint need to not become an outlier and the maximum point can stay in close proximity as outliers without being considered as a cluster of normal points.

Ideally, we use range of k to compute the LOF of data points and take the maximum LOF value from the range. We do this because we don't usually have the label of the data. the following method was used to precompute the negative LOF matrix

```
def compute_negative_lof_mat(X, k_list):
    negative_lof_matrix = np.ones((X.shape[0],
len(k_list))) * 99999

    for i in range(len(k_list)):
        lof =
LocalOutlierFactor(n_neighbors=k_list[i],
algorithm='auto', contamination='auto', n_jobs=-1)
        lof.fit(X)
        # store the negative LOF values in the
matrix
        for j in
range(len(lof.negative_outlier_factor_)):
            negative_lof_matrix[j, i] =
lof.negative_outlier_factor_[j]
    return negative_lof_matrix
```

Once the LOF matrix is found, a threshold is selected by observing the min LOF value of each row.

D. Isolation Forest

ISO Scores

Intuitively, ISO anomaly score is determined by two to the power of the normalized average height. As a result, negative score' point is classified outlier and positives are classified as inlier

E. Gaussian Mixture Model

Find optimal k

Optimal k is determined by analyzing the range of Bayesian Information criterion (BIC) and Akaike information criterion (AIC). The optimal k is determined by point with minimum BIC values.

Log-likelihood

To determine whether a data point is an anomaly we need to compute the log-likelihood of the given data.. We use the "score" method of GMM to compute the per-sample average log-likelihood of the data. Then, compare the likelihood values with the density threshold. we identify the outliers using the first percentile lowest density as the threshold. I.e., approximately 1% of the instances will be flagged as anomalies.

IV. Results & Answers to Questions from handout

Brief discussion about the different insights we get from the different data analysis tools.

Our answers to specific assignment questions are listed below:

A. Part B - Q9

We used 'l1' for the Silhouette score distance metric because it gave us the highest correlation for n_clusters = 2, and all valid normalizations wer*e* in agreement with n_clusters = 2.

B. Part B - Q10

Cluster size is related to features in a very simple sense. Because n_clusters = 2, we have two possible clusters 0 and 1. Cluster 0 contained ~95% of the movies in the dataset while cluster 1 contained ~5% of the data. Cluster 1 contained movies that were more likely to have correlations between two or more variables, while Cluster 0's relationships showed less correlation between variables.

C. Part B - Q13

The number of cluster in this dataset is highly dependent on the 4 data cluster lie on the correlated line between vote_count-revenue, budget-revenue, vote_count-budget and vote_count-popularity as shown in pairplot below

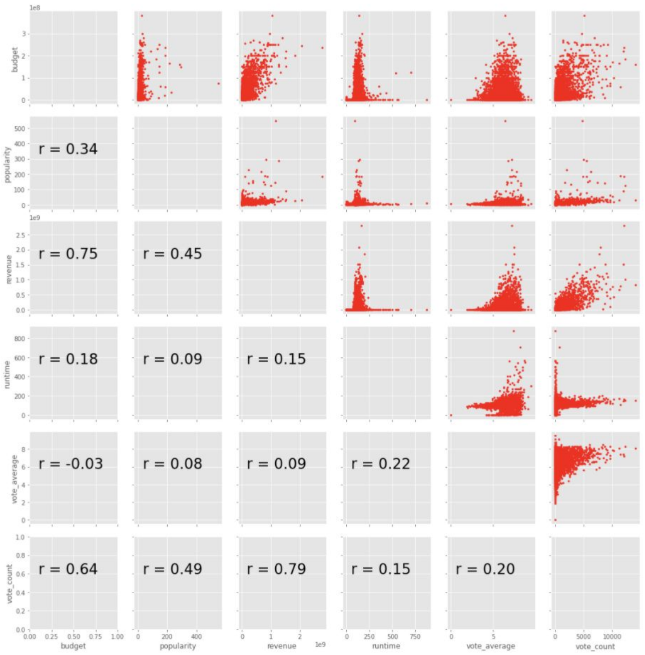
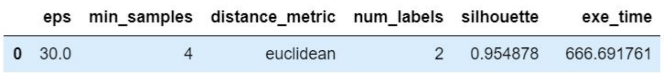


Figure 3: Pairplot of movies dataset

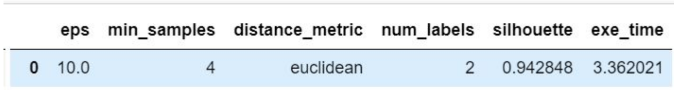
D. Part B - Q15

Overall sklearn our DBSCAN performed as well as sklearn in terms of silhouette scores but considering time, sklearn's DBSCAN is way faster as seen in the figure below.



	eps	min_samples	distance_metric	num_labels	silhouette	exe_time
0	30.0	4	euclidean	2	0.954878	666.691761

Figure a: parameters of our DBSCAN



	eps	min_samples	distance_metric	num_labels	silhouette	exe_time
0	10.0	4	euclidean	2	0.942848	3.362021

Figure b: parameters of sklearn DBSCAN

The silhouette score on our DBSCAN is 95 whereas on sklearn it is 3.36.

E. Part B - Q16

Clusters from sklearn KMeans, our DBSCAN and sklearn DBSCAN came out to be 2. All three gave the same number of clusters results. In terms of common metric i.e silhouette score; our DBSCAN gave the best results but time is a big tradeoff.

F. Part B - Q17

	Brute-Force	**Optimized**
Walltime	11mins 11 secs	9mins 54 secs

Table 1: Wall time of DBSCAN

Optimized DBSCAN performed better than brute-force DBSCAN model in terms of time complexity. Optimized DBSCAN run time is 9mins and 54 secs and brute-force is 11mins and 11 secs as shown in table 1. The total number of datapoints is 12177. This significant difference in time is due to memoization of euclidean distances which reduced repeated calculations for distances. The worst time complexity of brute-force DBSCAN approach is $O(n^2)$ since for each datapoint, distance is calculated against all the data points. On the other hand the overall average runtime complexity for optimized DBSCAN is $O(nlogn)$ but optimized DBSCAN approach has a space complexity of $O(n^2)$ since all the distances from each datapoint against all other data points are stored.

G. Part C - Q6

Feature set #1

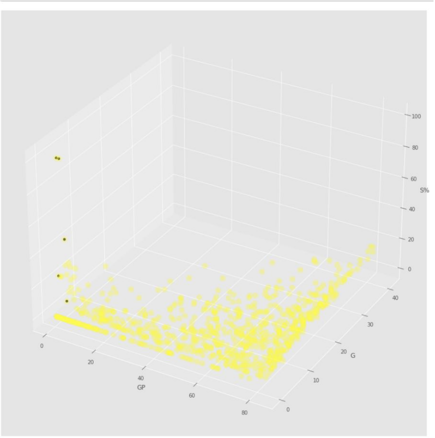


Figure 4: Feature set #1 with highlighted anomalies

Optimal k range on X1 dataset = [43, 44, 45, 46, 47, 48, 49, 50, 51]

According to heuristic for selecting the range for k, min k is selected by counting minimal cluster size which consist of similar behaving points. Max k is selected by considering the maximum number of objects that we want to be outliers if clustered together.

In general, LOF > 1 is used to identify outliers but since the X1 dataset is sparse, varying density, with many local fluctuations specific to that local cluster, we conclude that datapoint with LOF value > 3. would consider to be an outlier. After filtering out all the LOF values of all data points based on the anomaly LOF values, we notice that when k is less than 43 or greater than 51, all data points have similar LOG values. when k is in between 43 and 51, the distance points (738th, 742th, 745th) in X1 have a much smaller LOF value. Thus, the LOF algorithm successfully detected 3 outliers for k in between 43 and 51.

As shown in experiment below, datapoint 712th, 738th, 742th, 743th and 745th are detected by LOF algorithm as anomaly with three top worth highlighting anomaly 712th, 738th, and 742th. The same characteristic that.listed anomaly has in common is extremely low gameplayed, goals scored values and high shooting percentage. Intuitively, this is expected as in most hockey leagues, the number of gameplayed should be proportional to shoot percentage indicates that the player plays well and are given many opportunities to make a strike. The 3 anomaly datapoint went against with gameplayed-shooting percentage proportionality logic thus, the algorithm detected it in the low density area.

Datapoint #743, #745 is also classified as an anomaly due to its extremely low gameplayed and goal score values. In real life, this incident rarely happens because professional leagues often optimize their budget and hire well played hockey players to play often throughout the league. These 2 incidents indicate that there number of games played, shooting percentage and goal scored is the same. This can be interpreted as luck.

Feature set #2

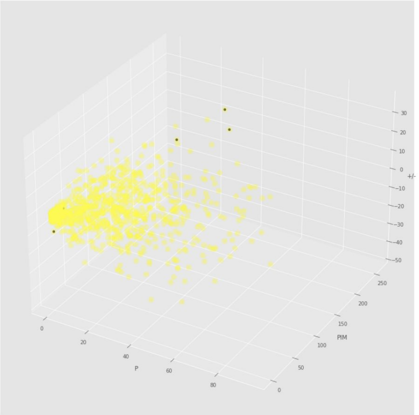
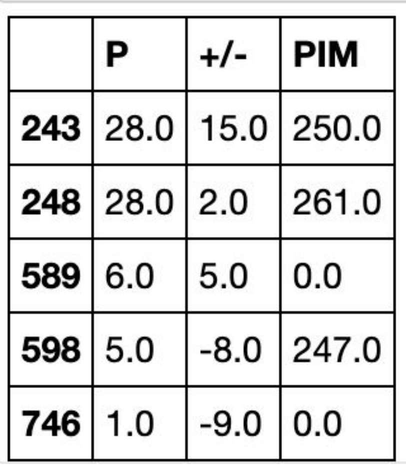


Figure 5: Feature set #1 with highlighted anomalies



	P	+/-	PIM
243	28.0	15.0	250.0
248	28.0	2.0	261.0
589	6.0	5.0	0.0
598	5.0	-8.0	247.0
746	1.0	-9.0	0.0

Figure 6: Display list of anomaly in feature set #2 using LOF

Optimal k range on X2 dataset = [51, 52, 53, 54, 55]

Similarly, the same procedure is applied to find optimal k value on the X2 dataset. when k is in between 51 and 55, LOF is varied between 2.5 and 3.0. Since the dataset is tight, clean and uniform, the LOF values of around 2.5 were picked as a threshold to determine optimal k.

The algorithm classified datapoint as an anomaly based on 2 categories of logic.

1. moderate point scored, moderate plus-minus statistic, and high penalty-minute values (243th, 248th, and 598)
2. moderate point scored, moderate plus-minus statistic, and low penalty-minute values (589th, 746th)

Intuitively, this is expected as moderated point-scored, moderate plus-minus(game impact) players are well trained and know the rule well so they should be expected to have a small number of penalties in a game. Since well played players often have a high number of games played, they are expected to have more conflict thus the number of penalties should be moderate.

Note: Moderate means near expectation value and within 1 std.

H. Part C - Q8 + Q9

Visually, isolation forest model is more sensitive to global outlier as shown in picture showing anomaly in feature set #2. anomalies lie in the low density with extreme feature values. Technically speaking, this is expected as the algorithm try to create multiple decision branches based on common mean values therefore, exceptionally low and high values will be detected and classified in the sparse branch region.

In case of LOF, the algorithm has advantage due to the local aspect of LOF, meaning that it only compares the score of anomalous of one sample with the score of its neighbors.

Depend on the application, isolation forest can be chosen over LOF where global anomaly is more interested over local

I. Part C - Q12

GMM is a Gaussian based model meaning each datapoint is evaluated among others to find its likelihood of belonging to a cluster. Since each datapoint is called within a smoothed Gaussian distribution, therefore, instances located in low-density regions will be considered as anomalies. Visually, datapoints 3, 22, 50, 64, ... 712 lie in the low density region. Datapoint 738th and 742th, who were classified as anomaly by LFO and Isolation Forest, were not detected by GMM as anomaly. This is expected because lots of other data points have low GP and G values which balance out the density thus, 738th and 742th were not classified as anomalies. Same argument can be held for feature set #2

J. Part C - Q13

GMM takes a different approach as LOF and Isolation Forest. GMM is a generative model, aiming to learn the probability distribution governing the dataset, while the latter is an pure outlier detection model, which rather than finding the clusters in the dataset, detects outlying points. Since GMM is governed by a probability distribution, datapoint in a low threshold is classified as anomaly. If a dataset has a high dimension then value in one dimension can balance out the other thus, GMM can fail to detect such points. LOF and isolation forest can be used as alternatives.

It is challenging to call 1 model perform better than another as depending on the dataset one can be prefered over the other. For applications like detecting anomaly players in sport, LOF and isolation forest can be used as those players with exceptionally high and low in a score can possibly be anomaly. Overall, GMM requires significantly longer time to train, excluding the required time for optimal k selection.

V. Conclusions & summary

In summary, KMean and GMM is a partially unsupervised learning model as we need to know the number of clusters ahead of time, thus, requiring prior knowledge of the dataset. Kmean can be very sensitive to outliers and can only form spherical clusters only. Whereas, GMM can help resolve those issues.

In terms of outlier detection model LOF, ISO and GMM, each provides its own advance over others in different dataset contexts. Having little prior knowledge to determine the best fit outlier detection model is still required.