Customer Classification

Clayton Daly, Jesse Dugan, Holly Hammons, Luke Logan

A. Data Processing

This project applies clustering and classification techniques to a group of hypothetical utility customers. The "building net" data from the RSF dataset provided was converted to customer data as follows. Data for each day of the year was used to represent a single customer, resulting in 365 time series for analysis, which represents daily load curves. Each time series was then converted to a 9-dimensional data instance consisting of the attributes shown below. The final dataset is a 365x9 matrix to represent the hypothetical customers.

- Peak demand [W]
- Time of peak demand [h]
- Minimum demand [W]
- Time of minimum demand [h]
- Total energy [Wh]
- Load factor
- Maximum energy ramping [W]
- Hour of the day when the maximum energy ramp occurs [h]
- Demand variance [W^2]

Peak demand and the hour of peak demand are helpful attributes to explain a load curve, and are important for utilities to note, as their grid infrastructure must be large enough to supply power at peak levels. Peak demand will vary throughout the year and differ between the weekdays and weekends, so these attributes will help to distinguish between different groups of hypothetical customers.

Similarly, the minimum daily demand and hour of minimum demand are important for these time series, as some of the "building net" values are negative, which represent when a customer supplies more PV power to the grid than they use on site [1]. These attributes will show high PV production days and the time of maximum production. They will be helpful to distinguish between different customers due to variations in weather and cloud cover throughout the year.

The total energy used by the customer is calculated from the area under the load curve and is important for utilities to know how much total daily electricity they need to supply. Like peak and minimum demand, this will change throughout the year, between weekends and weekdays.

Load factor is a measure of efficiency of electricity utilization and is defined in Equation 1 as,

$$\text{Load Factor} = \frac{\text{total daily energy use [Wh]}}{\text{peak demand [kW]} \cdot 24 \text{ hours}}. \quad (1)$$

The load factor varies from 0 to 1, with higher values pointing toward more efficient electricity utilization [2]. A high load factor indicates a relatively constant load curve, while a low load factor results from a sharp peak during only a few hours.

The maximum rate of change allows utilities to know how much electricity demand can change at given points in time to prevent excessive stress on distribution lines. The maximum rate of change of power demand signals how much stability is present in the electricity network, where a lower maximum rate of change signals a more stable network. For example, some generating plants need more time from startup to synchronize with the load, which must be accounted for when the demand is rapidly changing [3]. Additionally, there is inertia in electrical generators, which means they operate most efficiently when they stay at the same generation rates [4].

The hour before the maximum rate of change is important for utilities because it serves as an indicator of when the electric grid will see higher levels of stress to accommodate changing demand. Awareness of rapid demand change is important so that utilities can anticipate when more maintenance issues and problems will arise. For example, some generating plants need more time from startup to synchronize with the load. So, knowing the time when the load will ramp up will help utilities better plan for rapid changes in demand [3].

The variance of the consumption for each day is important as it can indicate where a consumer may have unusual use patterns. The variance can also be used to identify consumers who are more predictable with their power usage per day.

B. Customer Clustering

Using the dataset developed in section A, customers were grouped into clusters based on their energy consumption patterns. Prior to clustering, the data was normalized to ensure no attribute dominated the others. Clustering is unsupervised, so it is best to test numerous

models, using multiple performance metrics, to determine the best one. K-means, k-medoids, and hierarchical clustering algorithms were evaluated using the average sum of within-cluster distances, silhouette plots, and intra-cluster to inter-cluster distance ratios.

The first approaches tested were k-means and k-medoids. K-medoids was highly considered because it is more interpretable than k-means due to its representatives belonging to the original dataset [5]. The k-medoids algorithm can be especially useful for datasets that contain values that are very different from one another, which is helpful in instances such as this where the data describes utility consumption. For both options, models were developed with various distance functions and several number of clusters, k, ranging from 1 through 20. The within cluster sum of point-to-centroid distances were calculated and the average value was plotted against the number of clusters, shown in Figure 6 in Appendix A. The optimal value of k for each model was found using the elbow method, which describes the point after which the slope of the plot levels out [6]. Using this method, it was determined that the optimal number of clusters is 4. Silhouette plots were generated for numerous values of k and different distance metrics for both k-means and k-medoids models, shown in Appendix A. These silhouette plots were used to narrow down the best performing models. The k-medoids models, with correlation and cosine distance functions and k = 4, were shown to perform the best.

Next, hierarchical clustering was applied. For this approach, dendrograms and silhouette plots were generated for different combinations of linkage functions, using different distance metrics, shown in Appendix A. Many of these models resulted in silhouette plots that contained low silhouette values and very uneven clusters. In addition, these silhouette plots showed large negative values, which indicated a large amount of cluster overlap. It was determined that using the cosine distance metric, along with the centroid and complete linkage functions, produced the best silhouette plots for the hierarchical models.

The intra-cluster to inter-cluster distance ratios were found, which are summarized in Table I below, for the models that produced the best silhouette plots. These distance ratios were developed from varying model types, distance metrics, and number of clusters 'k', and were averaged between the clusters. It was found that, as the number of clusters increased, the distance ratios decreased. However, because the silhouette plots for k = 4 were found to contain the fewest negative values and more even clusters, it was decided to choose a model with four clusters that contained the lowest intra-cluster to inter-cluster distance ratio. Ultimately, it was found that the model with the best intra-cluster to inter-cluster distance ratio for k = 4 is the k-medoids model with a correlation distance function.

Model	Distance	Number 'k'	Intra:Inter Ratio Average
K-Medoids	Cosine	4	0.2233
K-Medoids	Correlation	4	0.2044
K-Medoids	Correlation	3	0.2326
K-Medoids	Correlation	5	0.1932
K-Means	Correlation	4	0.2108
K-Means	Cosine	4	0.2276
K-Means	Correlation	3	0.2296
K-Means	Correlation	5	0.1937
Hierarchical- Complete Linkage	Cosine	4	0.242
Hierarchical- Centroid Linkage	Cosine	4	0.2273
Hierarchical- Complete Linkage	Correlation	4	0.2552
Hierarchical- Centroid Linkage	Correlation	4	0.1932

Table I: Intra-Cluster to Inter-Cluster Distance Ratios for Best Performing Models

The final model chosen is a k-medoids model with 4 clusters that employs a correlation distance function. A visual representation of this model is shown in the silhouette plot in Figure 1. Figure 1 confirms that the four clusters are evenly distributed. While many of the clusters contain lower silhouette values, they contain a very small number of negative values when compared to other models, shown in Appendix A, which indicates less overlap between the clusters than other models tested. In addition, this model proved to have the lowest intra-cluster to inter-cluster distance ratio when compared to other well performing models with four clusters.

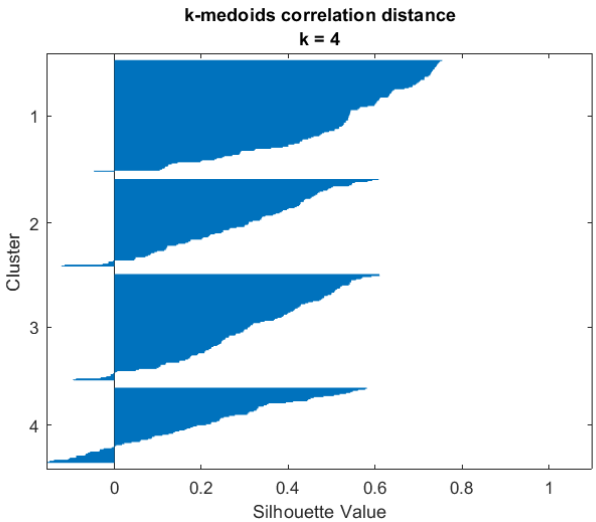

Fig. 1: Final Model Silhouette Plot

C. Customer Classification

Using the results from clustering the data, multiple classification models were developed to predict the class of a customer, based on available attribute data. The models compared include K-nearest neighbor (KNN),

multinomial logistic regression, linear discriminant analysis (LDA), quadratic discriminant analysis (QDA), and a classification tree. To develop each classifier, the 365 instances of customer data were separated into 334 instances of training data and 31 instances of testing data.

The metrics chosen to evaluate the performance of each classifier include the correct rate, the sensitivity, and the specificity. Each metric is weighted based on its importance and summed to form a performance score. Since classifying a customer correctly based on the attribute data is the most important outcome of the classifier, the correct rate is weighted the most at 40%. Considering that identifying true positive classes of energy consumption is the main goal, and there isn't a high cost to false positives, sensitivity is rated higher than specificity. So, given the importance of true positives over true negatives, sensitivity is weighted at 40% and specificity is rated at 20%. Combining these metrics, the formula for the performance score of each classifier is shown in Equation 2 below.

$$\text{score} = 0.4 \times \text{correct rate} + 0.4 \times \text{sensitivity} + 0.2 \times \text{specificity} \quad (2)$$

MATLAB functions were used to create a model for each classifier based on the training data. Then, the testing data was used from the month of December to evaluate the performance of each classifier. The attributes from the testing data were passed into each model and the output from each class was compared to the true classes in the testing data.

In order to evaluate the performance, a confusion matrix was found for each model, based on comparing the ground truth from the testing data and the output of the model. The confusion matrix was then used to solve for the sensitivity and specificity. Since sensitivity and specificity are based on which classes are considered positive and negative, an algorithm was developed that considers the specificity and sensitivity of each class. The algorithm works by first calculating the specificity and sensitivity where one class is considered positive and the rest are considered negative. Each combination of one positive class and all other classes being negative is then evaluated for sensitivity and specificity. Then, for every combination, the sensitivity and specificity are summed together. Finally, these totals are normalized to reflect the total sensitivity and specificity of the model. Additionally, the correct rate was found using a built-in MATLAB function for classifier performance. The performance score was found using Equation 2 above, where a score of 1 is the highest possible. For the KNN classifier, values of k from 1 to 20 were compared based on their performance score using the training data, where the optimal value of k was found to be 6, shown in Appendix B. Table II below shows the performance score of each classifier after training and testing.

Classifier Model	Performance Score
KNN	0.785
Multinomial Logistic Regression	0.948
LDA	1.000
QDA	0.817
Decision Tree	0.809

Table II: Classification Performance Outcomes

The results demonstrate that multinomial logistic regression and LDA were the highest performing classifiers based on the testing performance. While LDA has the highest score, the testing data is only a small section of the data, which doesn't reflect every classification scenario. So, the small difference in scores between LDA and the logistic classifier warranted a further comparison of both approaches.

The logistic classifier predicts the probability of each class based on a linear combination of each predictor [7]. LDA predicts the probability of each class based on approximating linear boundaries between each class [7]. While both probabilistic classifiers are similar, LDA has advantages over logistic regression because it is easier to use with a larger number of classes and is faster to compute. As a result, LDA was chosen as the best classification model for energy consumption classes. The confusion matrix for LDA is shown in Figure 2 below, where the classifier perfectly predicted every class of the testing data. Thus, the sensitivity, specificity, and correct rate had a value of 1. While a larger testing set would likely lead to a decrease in each of these metrics, the excellent performance of LDA with the given testing set suggests that this model is a very accurate predictor of the customer's energy consumption class.

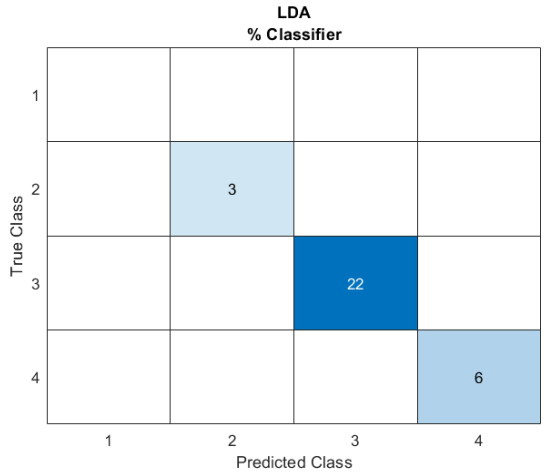

Fig. 2: LDA Confusion Matrix

REFERENCES

[1] *Electricity demand patterns matter for valuing electricity supply resources*, en. [Online]. Available: https://www.eia.gov/todayinenergy/detail.php?id=12711 (visited on 05/01/2022).

[2] "What is load factor and why is it important?" Iowa Lakes Electric Cooperative. (), [Online]. Available: https://www.ilec.coop/news-and-events/news/what-is-load-factor-and-why-is-it-important (visited on 05/06/2022).

[3] Y. Wan, "Analysis of Wind Power Ramping Behavior in ERCOT," en, *Renewable Energy*, p. 23, 2011.

[4] P. Denholm, T. Mai, R. Kenyon, B. Kroposki, and M. O'Malley, "Inertia and the Power Grid: A Guide Without the Spin," en, Tech. Rep. NREL/TP-6A20-73856, 1659820, MainId:6231, May 2020, NREL/TP–6A20–73 856, 1659820, MainId:6231. DOI: 10.2172/1659820. [Online]. Available: https://www.osti.gov/servlets/purl/1659820/ (visited on 05/01/2022).

[5] S. Mohagheghi, *EENG 415 - Data Science for Electrical Engineers: Lecture 10 - Clustering*, Mar. 2022.

[6] *Using the elbow method to determine the optimal number of clusters for k-means clustering*. [Online]. Available: https://bl.ocks.org/rpgove/0060ff3b656618e9136b (visited on 05/01/2022).

[7] S. Mohagheghi, *EENG 415 - Data Science for Electrical Engineers: Lecture 19 - Classification (2)*, Mar. 2022.

APPENDIX A
CUSTOMER CLUSTERING

1) K-Means Models:

Figure 3 below show the silhouette plots for the k-means models for a variety of parameters that were tested. Due to the nature of the algorithm, the results for k-means change with each run. The initial selection of centroid is random, and thus the output and resulting silhouette plot will not be the same. The plot demonstrates that k-means is not an appropriate approach for our dataset. Cityblock distance in Figure 3 showed more uniform clusters compared to the squared Euclidean distance metric. However, both metrics resulted in significant negative silhouette values, which indicate significant overlap between clusters.

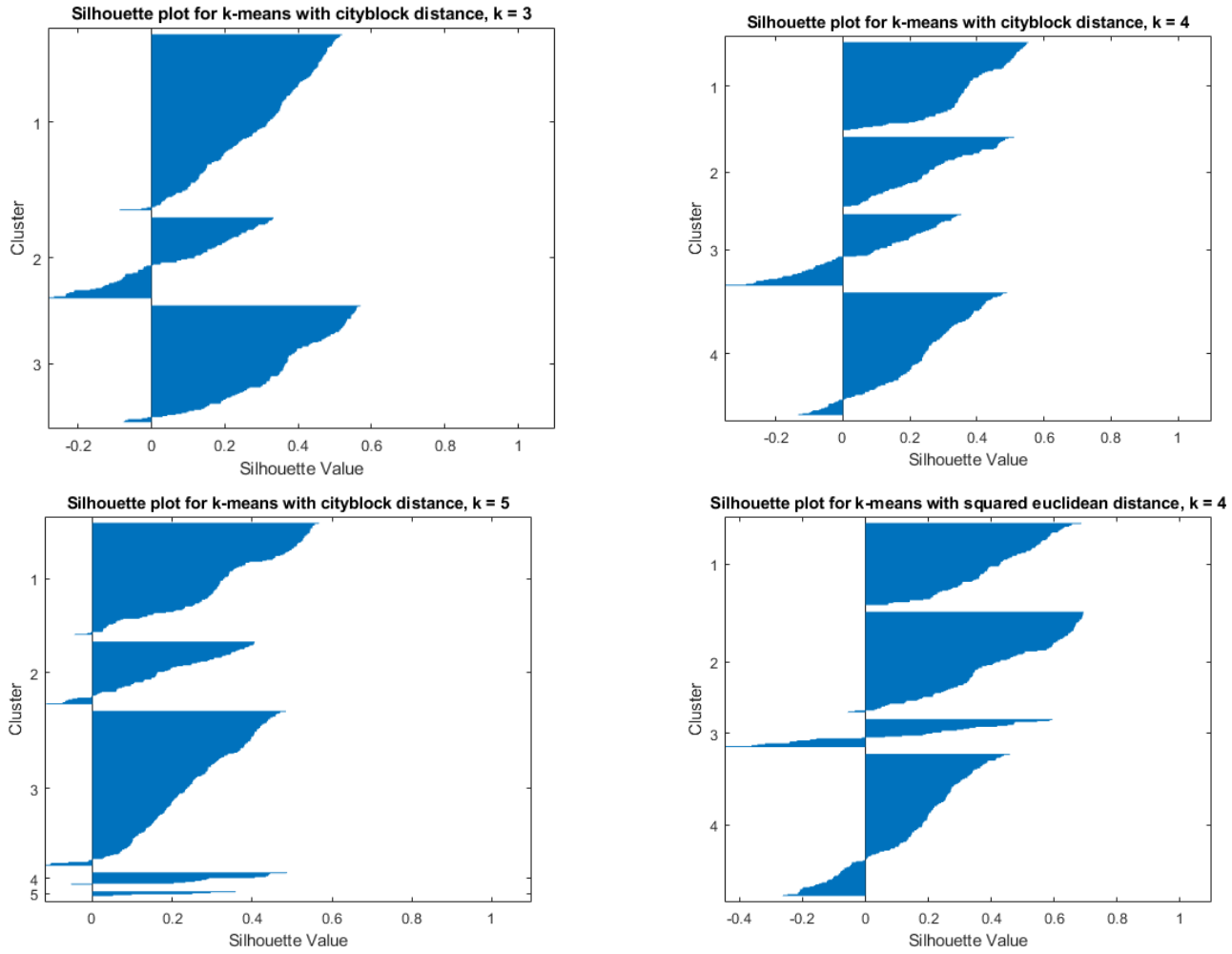

Fig. 3: K-Means Silhouette Plots for Testing

2) K-Medoids Models:

Figure 4 shows the silhouette plots for the k-medoids models that were tested. The more uniform silhouette plots confirm the results that cosine and correlation distances result in better clusters than Euclidean distance, which is the third-best performing distance function in Figure 4. The cosine and correlation distances generate fewer negative numbers, and the largest silhouette values within the clusters of all the models analyzed.

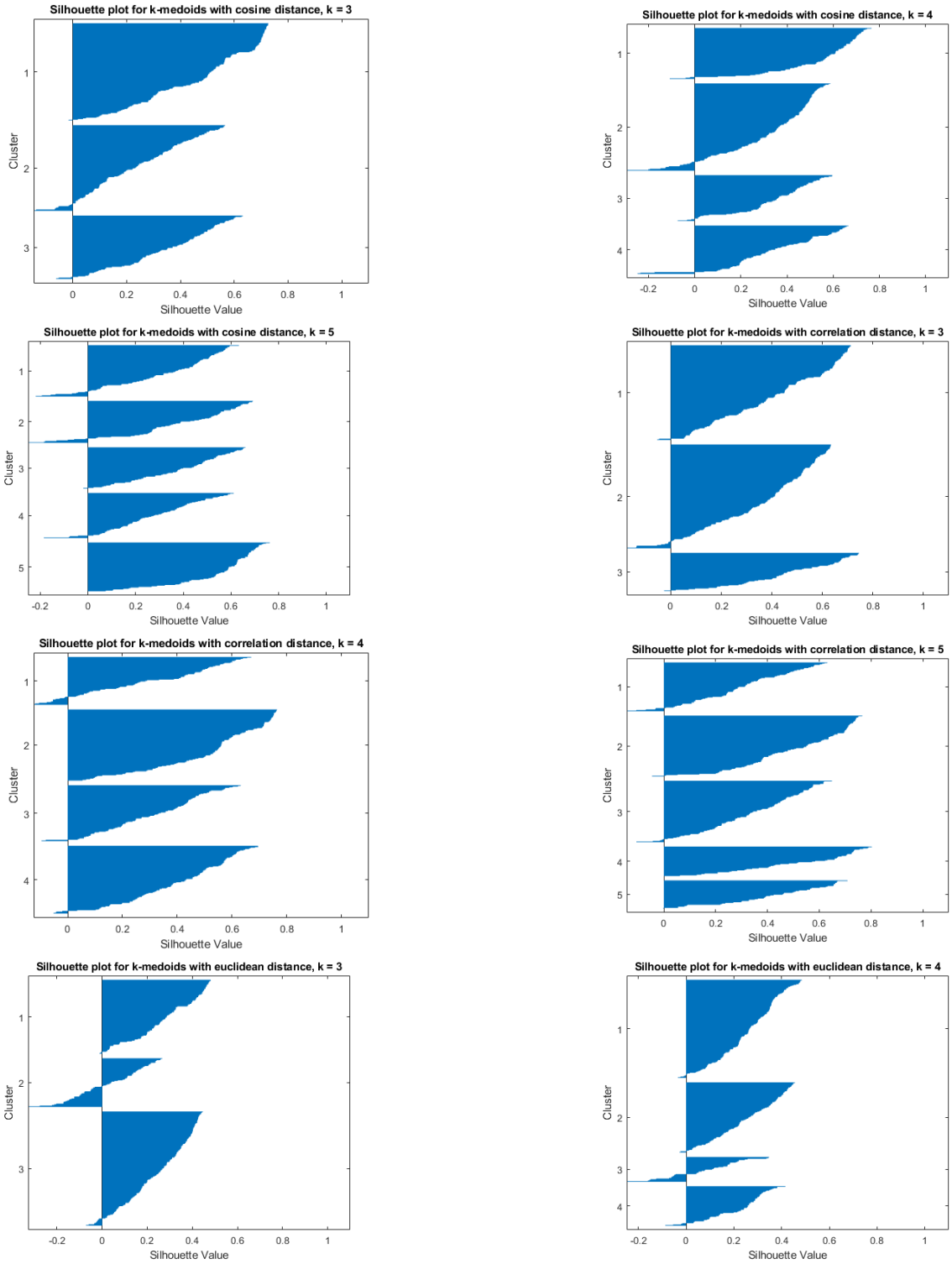

Fig. 4: K-Medoids Silhouette Plots for Testing

3) Hierarchical Models:

Figure 5 below show the silhouette plots for the hierarchical models, and their corresponding dendrograms, for a variety of parameters that were tested. While several maximum clusters, k, were tested, these figures depict a constant parameter of k = 4 to better analyze the results of changing distance metrics. These plots show how the hierarchical models performed poorly in comparison to the k-means and k-medoids models. The size of the clusters, and their silhouette values, are very inconsistent depending on the distance metrics used. Although the model that used the cosine distance metric appear to have more even and meaningful clusters, it contains larger negative values, indicating greater overlap between the clusters.

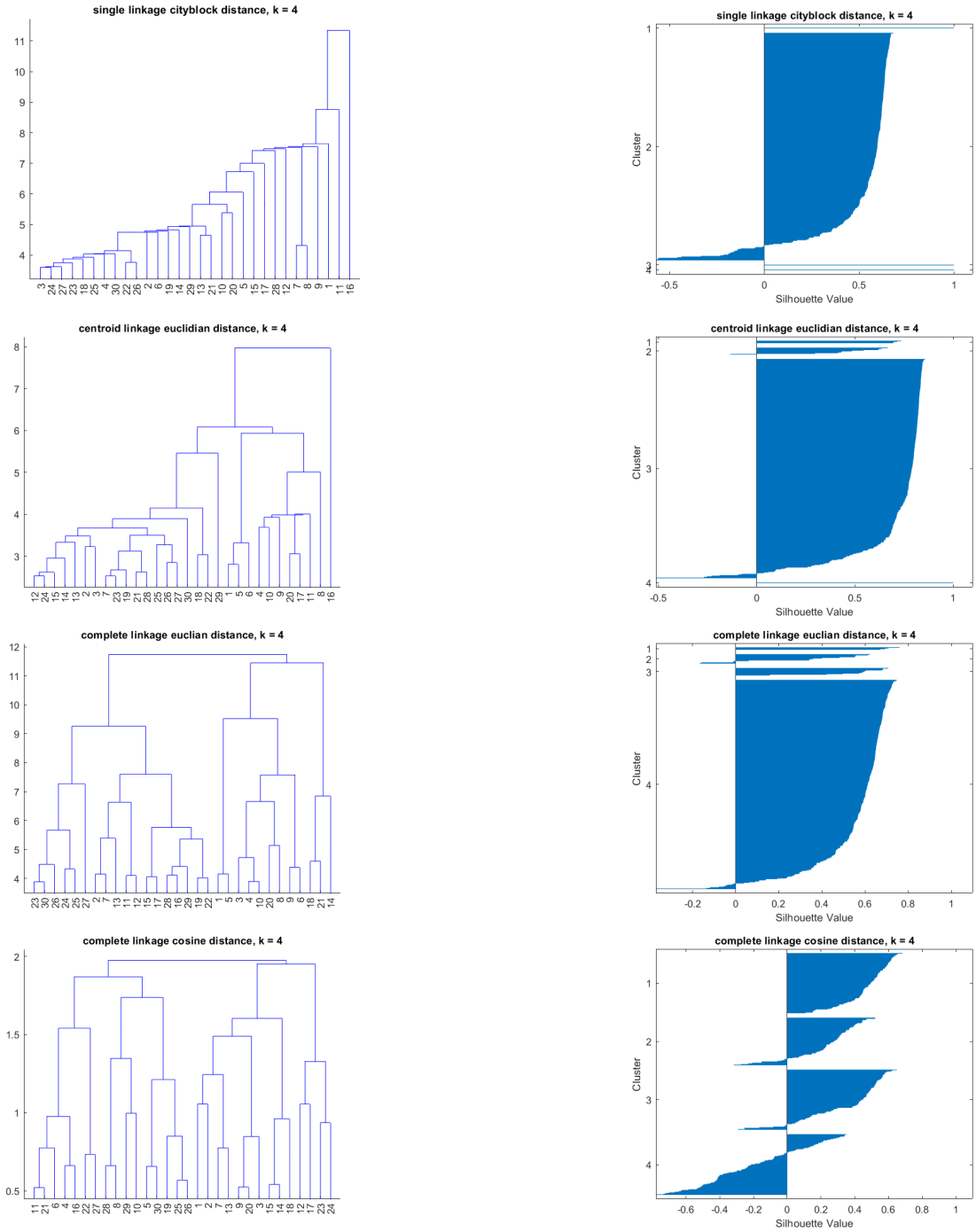

Fig. 5: Hierarchical Silhouette Plots for Testing

4) Elbow Method for Finding Optimum k:

Figure 6 below shows the graph that was generated to determine the optimal number of clusters, k. The elbow method [6] was used to determine that the optimal value of k for all of the models is about k = 4.

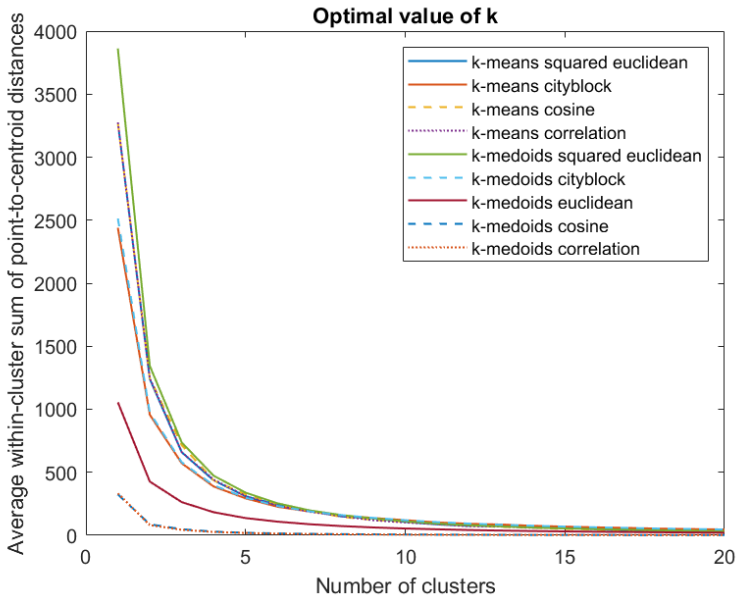

Fig. 6: Determining the Optimal Number of Clusters

APPENDIX B
CUSTOMER CLASSIFICATION

1) Confusion Matrices:

Figure 7 shows the confusion matrices for each classifier that was not chosen for the final classification model including KNN (k = 6), multinomial logistic regression, QDA, and a decision tree.

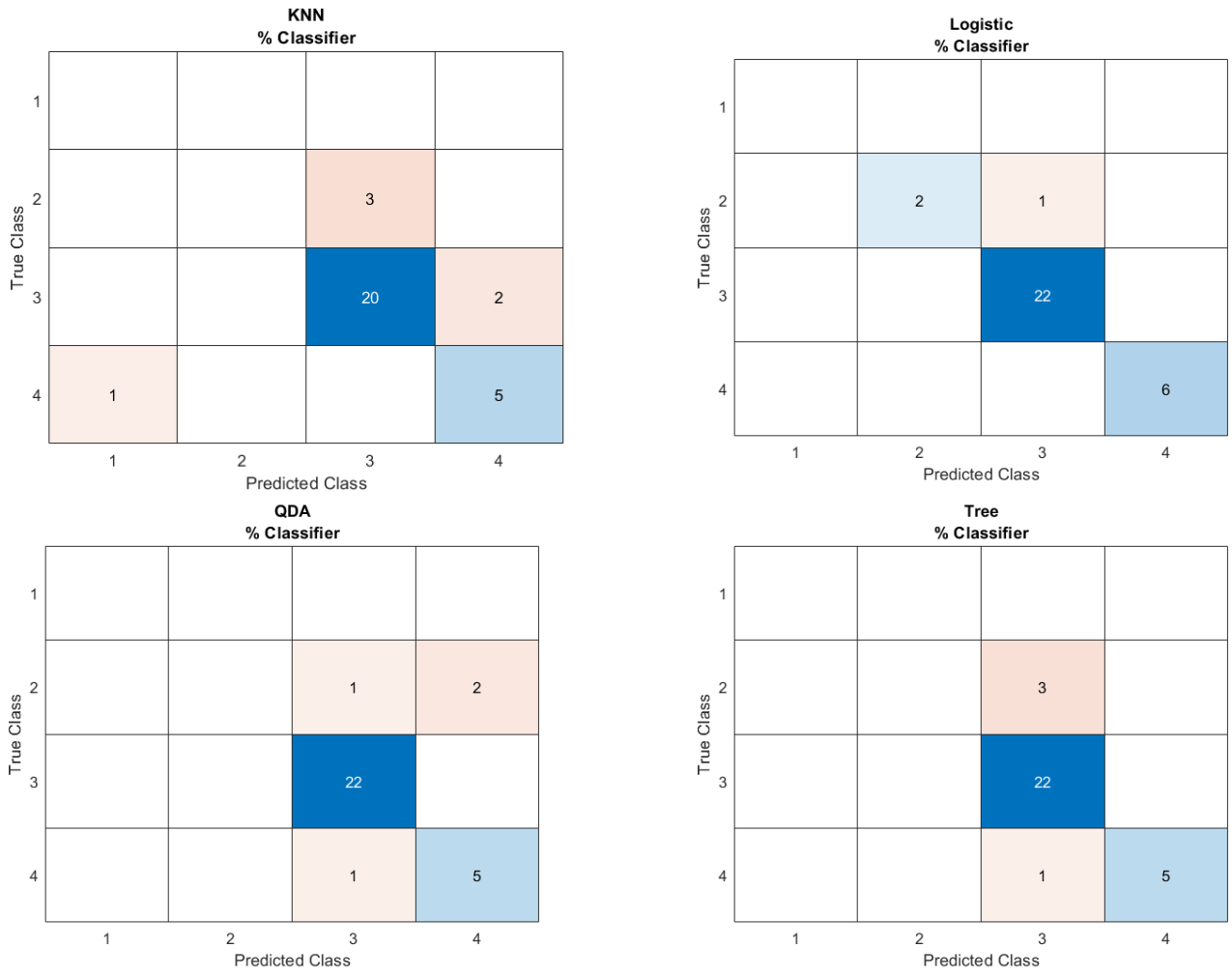

Fig. 7: Confusion Matrices for Classifiers Not Chosen

2) Classifier Performance Table:

Table III below shows the sensitivity, specificity, correct rate, and performance score for each of the classification models evaluated.

Model	Sensitivity	Specificity	Correct Rate	Score
KNN	0.6856	0.8886	0.8333	0.785
Multinomial Logistic Regression	0.9167	0.9722	0.9677	0.948
LDA	1.0000	1.0000	1.0000	1.000
QDA	0.7083	0.9244	0.8710	0.817
Decision Tree	0.7083	0.8889	0.8710	0.809

Table III: Classifier Performance Table

3) KNN Finding Best k:

Table IV below shows the sensitivity, specificity, correct rate, and performance score for the KNN classification model for values of k ranging from 1-20.

K	Sensitivity	Specificity	Correct Rate	Score
1	0.7667	0.6629	0.8697	0.7458
2	0.7586	0.5909	0.8438	0.7086
3	0.7667	0.6629	0.8707	0.7460
4	0.7667	0.6326	0.8508	0.7299
5	0.7667	0.6326	0.8508	0.7299
6	**0.8333**	**0.6856**	**0.8886**	**0.7853**
7	0.7667	0.6326	0.8508	0.7299
8	0.8000	0.6439	0.8608	0.7497
9	0.7333	0.6212	0.8408	0.7100
10	0.8000	0.6439	0.8608	0.7497
11	0.7333	0.6212	0.8408	0.7100
12	0.7667	0.6326	0.8508	0.7299
13	0.7241	0.6098	0.8505	0.7037
14	0.7667	0.6629	0.8686	0.7455
15	0.7241	0.6098	0.8505	0.7037
16	0.7241	0.6098	0.8505	0.7037
17	0.7586	0.6212	0.8605	0.7240
18	0.7241	0.6098	0.8505	0.7037
19	0.7143	0.5682	0.8425	0.6815
20	0.7143	0.5682	0.8425	0.6815

Table IV: Evaluation of Number of k

4) Receiver Operating Characteristics :

Figure 8 shows the Receiver Operating Characteristics (ROC) curves for each of the classification models developed. The curves were obtained using a "one versus all" approach to simplify a multiclass problem into several binary classification problems. One class is considered positive, with all other classes considered negative. This was done for each class except for class 1, as it did not appear in the test results from the clustering algorithm. The area under the curves are also given in the figures, which corresponds to the performance of the classification model, with a value of 1 indicating a perfect model.

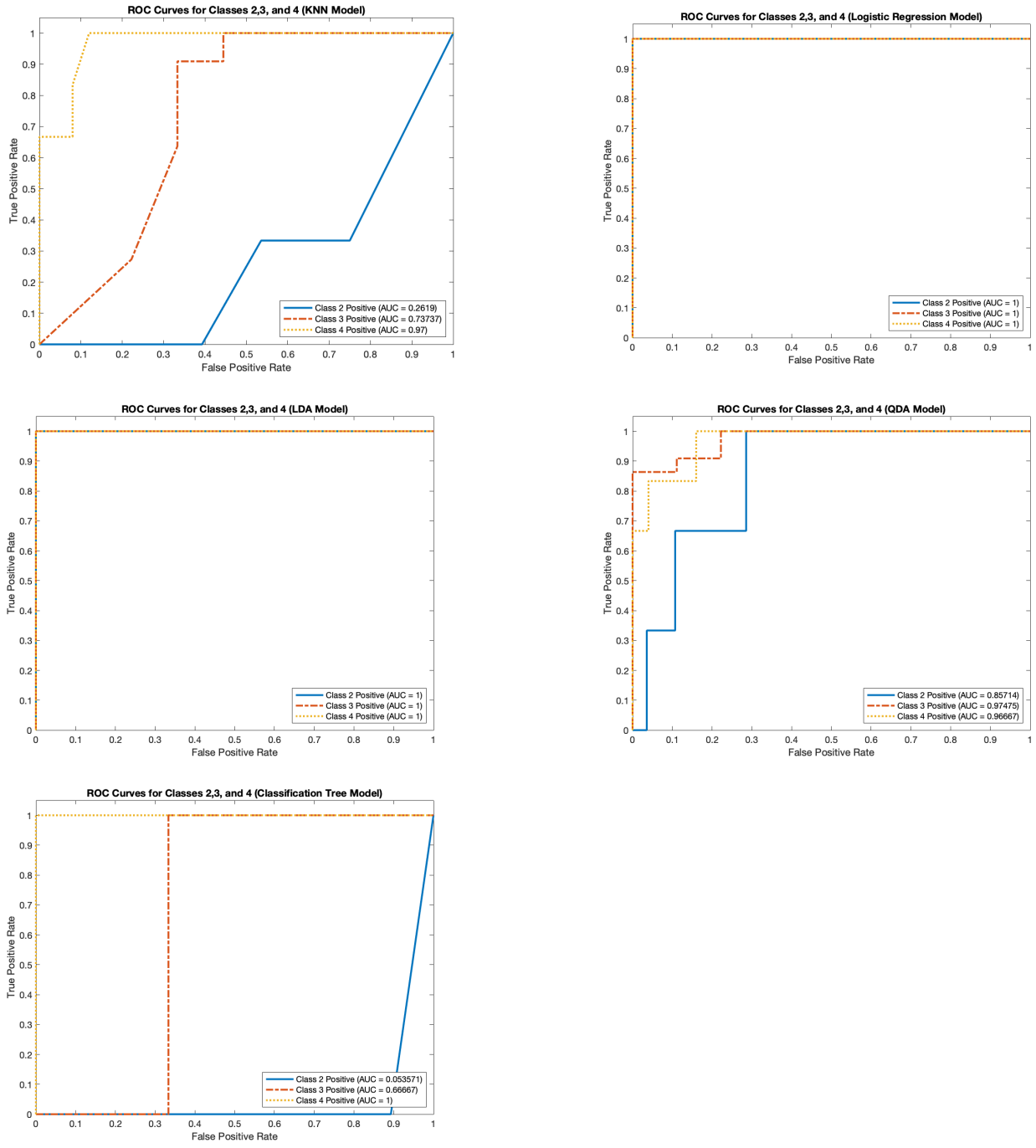

Fig. 8: ROC Curves for Each Model Developed