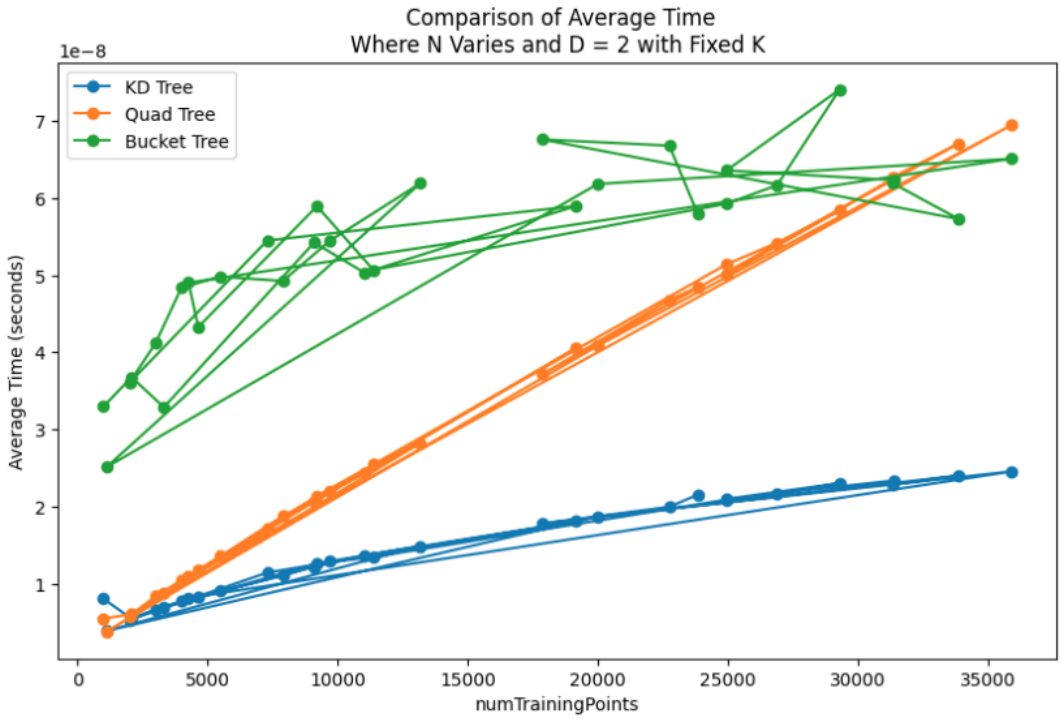
Gloria Dukuzeyesu

University of Utah
MSD Program, Class of 2023

Spatial Partitioning Analysis

5 Data set Points were plotted.

1. DATA SET1: Comparison of Average Time where N varies and D = 2 with a fixed K



Observation:

The first plot focused on comparing the average time taken by three data structures (KD Tree, Quad Tree, and bucket tree) when varying the number of training points (N) while keeping the dimension (D) fixed to 2 and the value of K fixed at 10.

int[] N_values = [1000,2084,3324,9098,11039,19168,7296,2000,3000,4000, 5500, 7891,9715, 13171, 1121,20000,35890,4274,4645,9209,11374,24948,26893,29294, 24950, 31369, 31345, 33868,17860, 22753,23841].

The following observations were made from DataSet 1.

- **KD Tree:** KD Tree demonstrated the fastest performance among the three. It consistently outperformed both Quad tree and Bucket tree in terms of finding the closest neighbors in 2D space. The average time taken by the KD tree to search for the nearest neighbors remained relatively low throughout the tested range of test points.
- **Quad tree**: The Quad tree showed a slight lower performance compared to the KD tree(when the testing points were smaller) but was faster than the Bucket Tree. It demonstrated a reasonable efficiency in finding the closest neighbors. However, when the testing points grew large enough, the Quad tree demonstrated a significantly slower performance compared to KD Tree. In other words, as the number of training points increased, the average search time also increased gradually.
- **Bucket tree:** The bucket tree exhibited the slowest performance among the three data structures when searching for the K nearest neighbors. It consistently took more time more time compared to both the KD tree and Quadtree. The search time in the bucket Tree varies based on the distribution of the K values within the grid. Consequently, depending on the location of the k values, the time taken to find the nearest neighbors varies significantly.

Conclusion

Based on the observation above, it can be concluded that when the number of training points (N) varies and the D is fixed at 2, the KD tree data structure outperforms both the Quad tree and Bucket tree in terms of speed in finding the K nearest neighbors. The Quad tree demonstrates reasonably good performance, but slightly slower than the KD tree. And the Bucket tree proves to be slower in comparison, varying search times depending on the distribution of the K values within the grid.

Linear regression Results

KD_Model Summary

- R^2 = 0.964
- P value = 0.000
- Coef = 5.787e-07

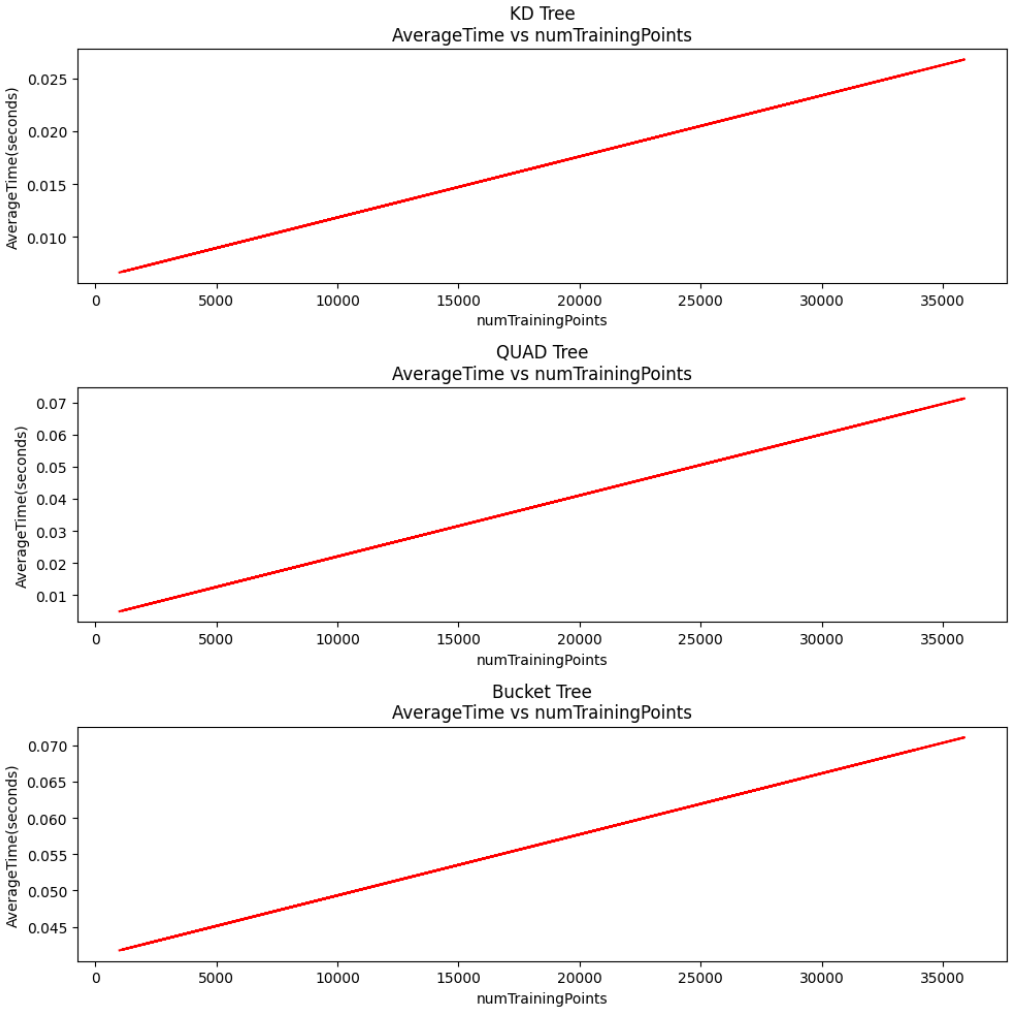
QUAD_Mode summary:

- R^2 = 0.999
- P value = 0.000
- Coef = 1.902e-06

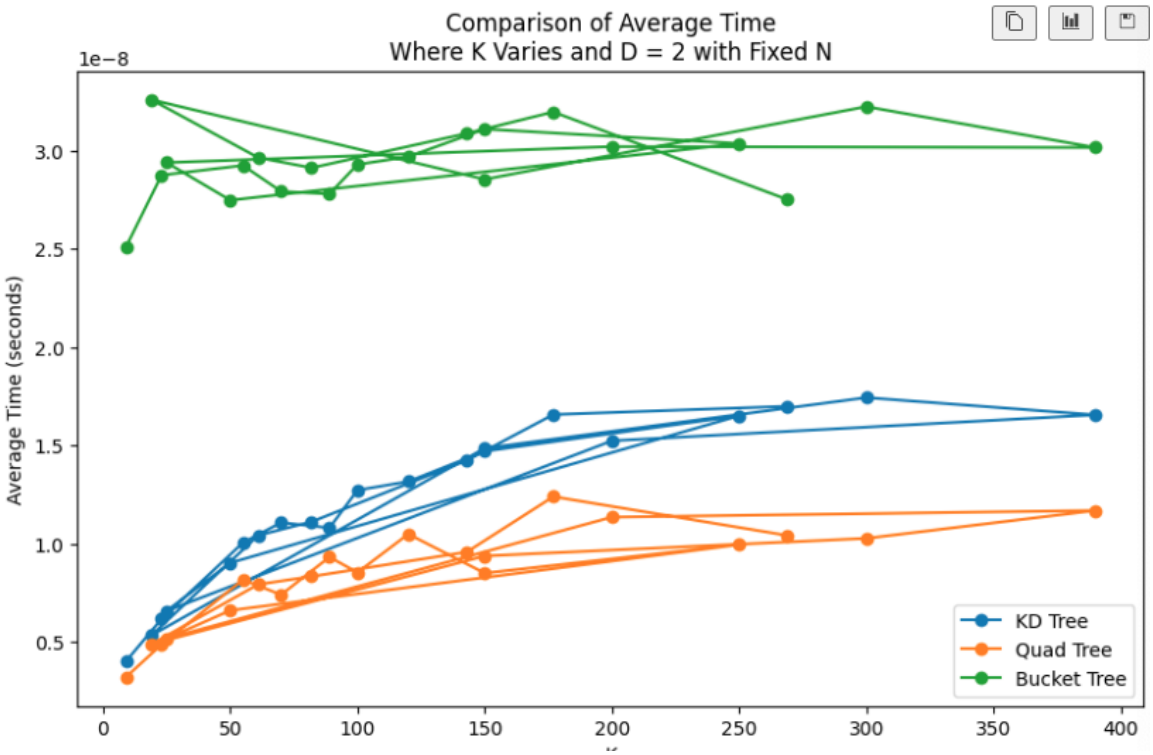
Bucket_model summary

- R^2 = 0.641
- P value = 0.000
- Coef = 8.407e-07

Based on the model's summary we can conclude that there is statistical evidence that the num training points have a significant impact on the average time in all three data structures and the coefficients initiated the direction and magnitude of this relationship. The higher R squared values observed in all modes (96.4 % in KD Model, 99.9% in Quad model and 64.1% in bucket model) indicates that the models fit the data well and can explain a substantial portion of the variability in the average time.



2. DATA SET2: Comparison of Average Time where K varies and D = 2 with a fixed N.



Observation

The DataSet 2 focused on comparing the average time taken by KD Tree, Quad tree and bucket tree when varying the K values while Keeping the D fixed to 2 and the Value of Num training points fixed at 1000.
Here's a list of K values used.
int[] K_values = [9,23,55,70,89,100,120, 150, 250, 50, 25, 200,390, 300, 150, 19, 61, 82, 143, 177, 269]

The following observations were made from DataSet 2:

- **Quad Tree:** The Quad Tree consistently outperformed both the KD Tree and Bucket Tree in terms of finding the closest neighbors in 2D space. It exhibited a relatively low average time throughout the tested range of K values. Quad Tree demonstrated efficient performance and proved to be the fastest among the three data structures in this scenario.

- **KD Tree:**The KD Tree demonstrated slightly lower performance compared to the Quad Tree when the K values were relatively smaller. However, it still performed faster than the Bucket Tree. The KD Tree showed reasonably good performance in finding the K nearest neighbors, although it was slightly slower than the Quad Tree.
- **Bucket Tree:**The Bucket Tree exhibited the slowest performance among the three data structures when searching for the K nearest neighbors. It consistently used a similar amount of time to find the K nearest values, with slight fluctuations. The performance of the Bucket Tree was influenced by the location of the K closest neighbors within the grid array. The overhead caused by empty buckets contributed to the overall slower performance of the Bucket Tree.

Conclusion

Based on the observations, it can be concluded that when the number of training points (N) is fixed, the dimension (D) is fixed at 2, and the value of K varies, the Quad Tree data structure outperforms both the KD Tree and Bucket Tree in terms of speed and efficiency in finding the K nearest neighbors. The KD Tree demonstrates reasonably good performance, although slightly slower than the Quad Tree. On the other hand, the Bucket Tree exhibits slower search times due to the influence of empty buckets and the distribution of K values within the grid.

Linear regression

KD_Summary

- R^2 = 0.764 : The R-squared value of 0.764 indicates that the KD tree model explains approximately 76.4% of the variance in the data.
- P value = 0.000 : Statistically significant
- Coef = 3.489e-05: The positive coefficient suggests a positive relationship between the number of K values and the average search time.

Quad_Summary

- R^2 = 0.628
- P value = 0.000
- Coef = 1.878e-05

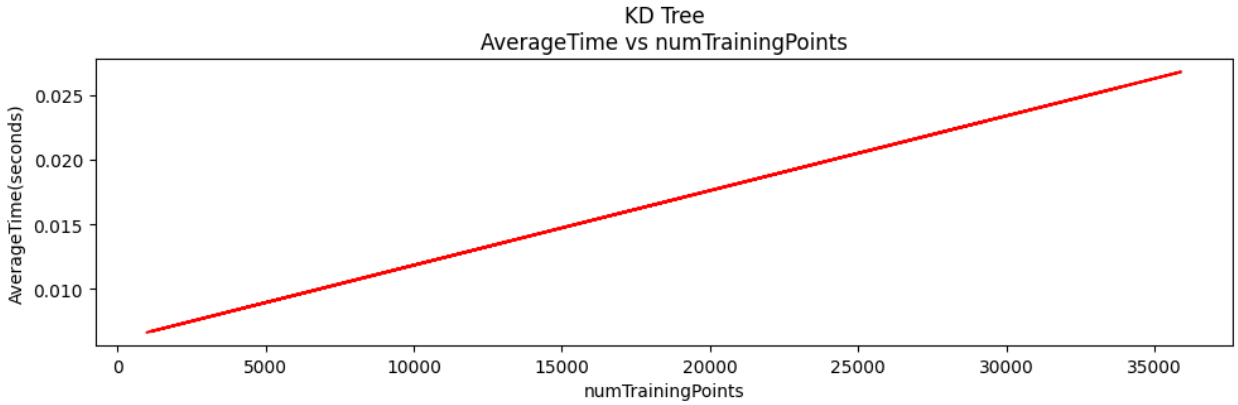
The Quad tree model explains approximately 62.8% of the variance in the data. This is consistent with the report's observation that Quad tree generally outperforms both KD tree and Bucket tree in terms of finding the closest neighbors in 2D space.

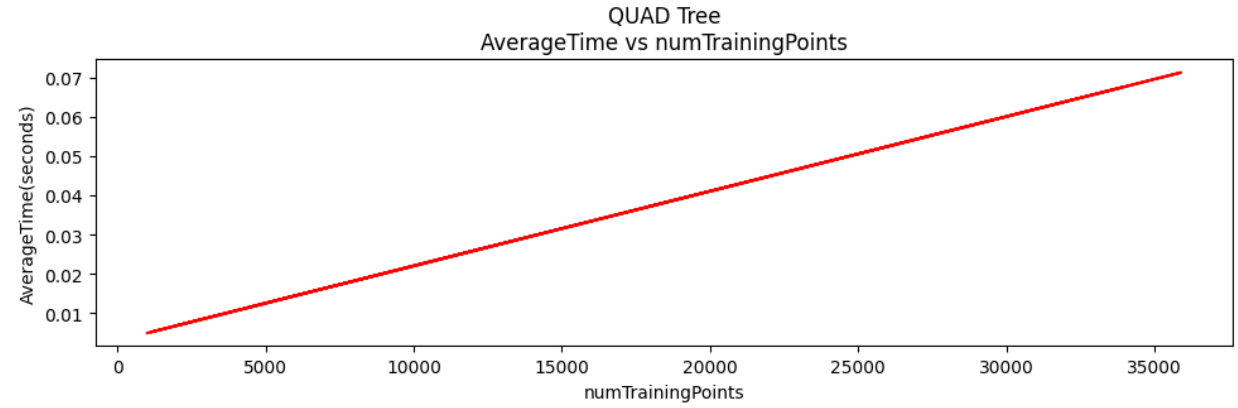
Bucket_Summary

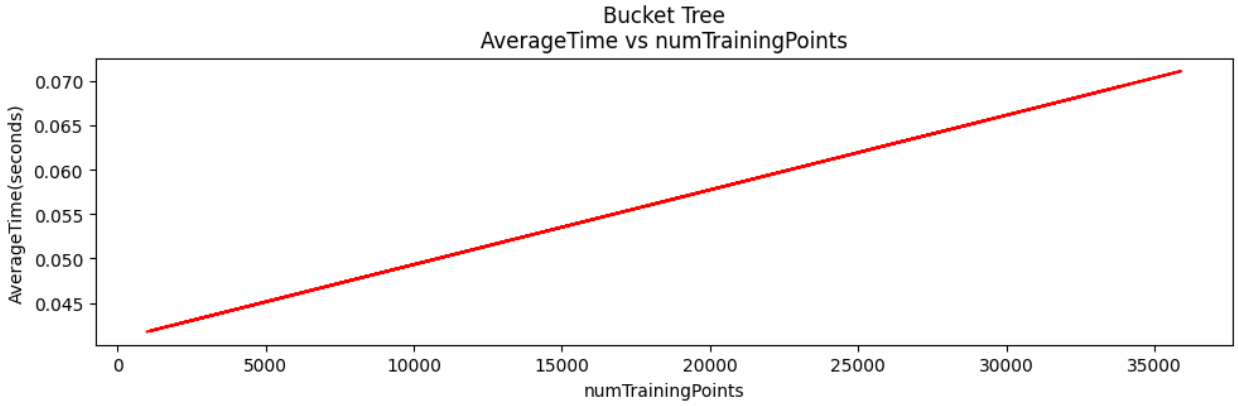
- R^2 = 0.124
- P value = 0.117
- Coef = 6.104e-06

The Bucket tree model explains only 6.104e-06 of the variance in the data, and the positive coefficient of 0.0287 suggests a positive relationship between the K and the average search time.

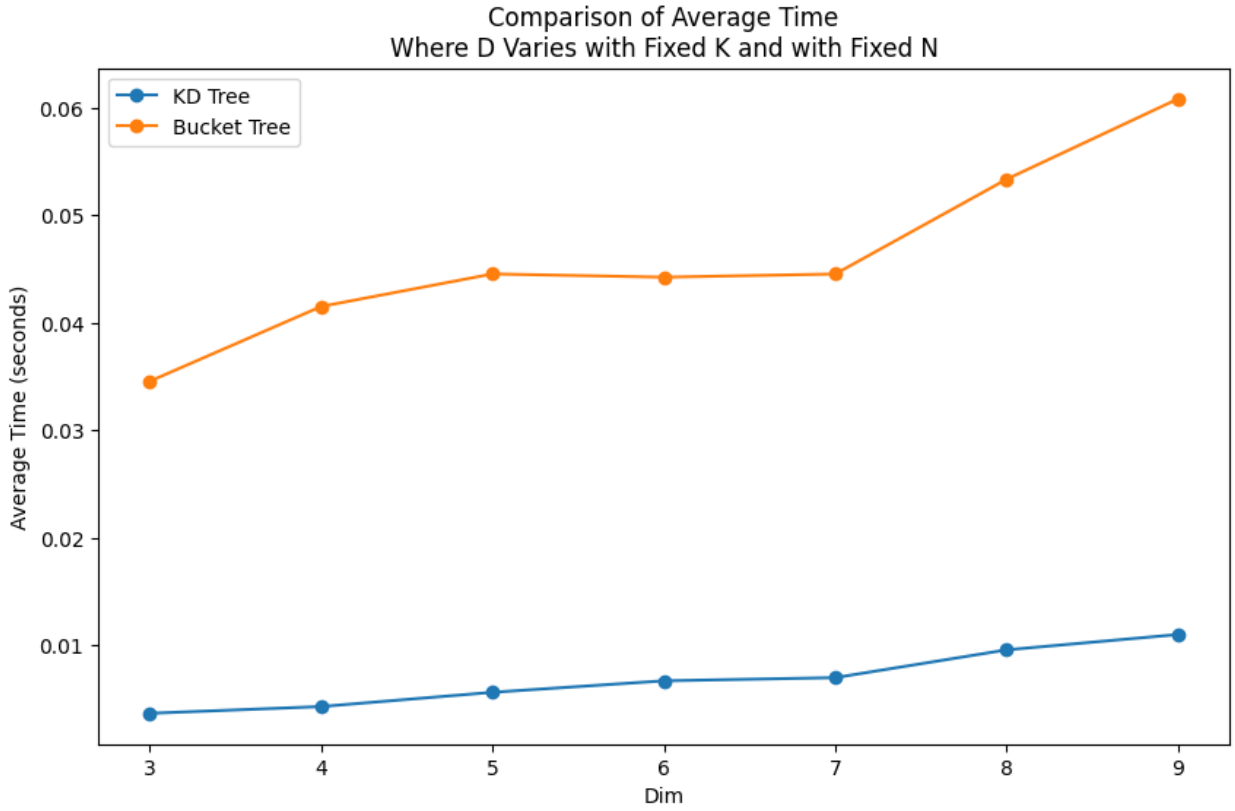
From the Linear regression summary and the plots below, we can conclude that the regression generally support the observations made in the observation above.







3. DATA SET3: Comparison of Average Time where D varies and with fixed K and Fixed N. Only KD tree and Bucket Tree were used



Observation:

The DataSet 3 analysis focused on comparing the average time taken by KD Tree and Bucket Tree when varying the dimension (D) while keeping the value of K fixed and the number of training points (N) fixed. Quad Tree was not used in this analysis as it only works with a dimension of 2. The dimensions used in this analysis ranged from 3 to 10. The following observations were made:

- **KD Tree**. The KD Tree consistently outperformed the Bucket Tree in terms of speed when searching for the K nearest neighbors. As the dimension increased, the time taken by the KD Tree to search for K values also increased, but it remained faster than the Bucket Tree in all tested dimensions. The KD Tree demonstrated efficient performance across different dimensions, maintaining a faster search time compared to the Bucket Tree.

- **Bucket Tree:** The Bucket Tree exhibited relatively higher search times compared to the KD Tree in different dimensions. In lower dimensions, the Bucket Tree showed relatively lower search times, but as the dimension increased, the search time in the Bucket Tree increased significantly. The Bucket Tree's performance was affected by the dimensionality of the data, with higher dimensions resulting in longer search times.

Conclusion

Based on the observations, it can be concluded that when the dimension (D) varies with a fixed value of K and N, the KD Tree outperforms the Bucket Tree in terms of speed and efficiency when searching for the K nearest neighbors. The KD Tree consistently demonstrated faster search times across different dimensions, while the Bucket Tree exhibited higher search times, especially as the dimensionality of the data increased.

Linear regression

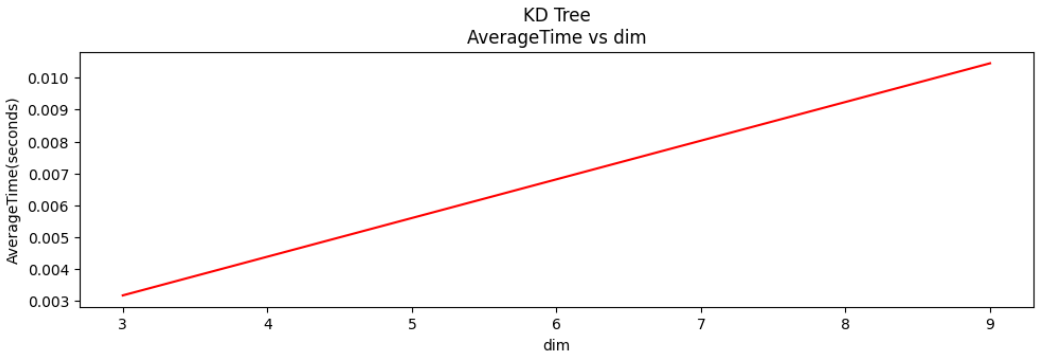
KD_Summary

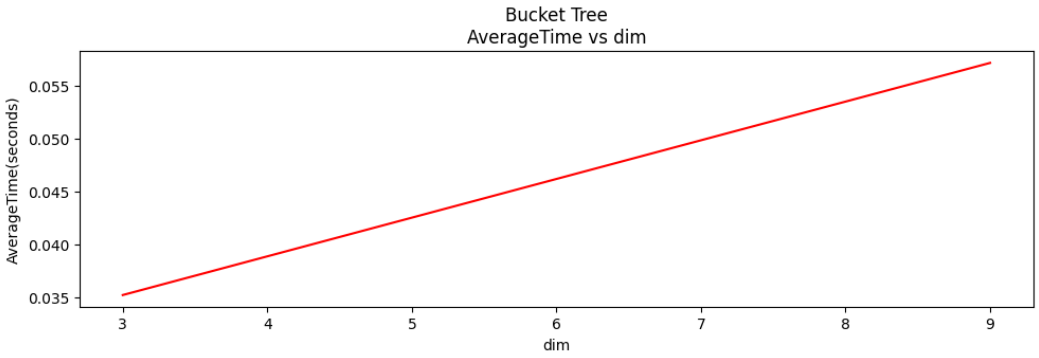
- R^2 = 0.959 : 95% R-squared value indicates that there is a strong correlation between the dim and average time.
- P value = 0.000
- Coef = 0.0012 (a less increase in search time compared to Bucket Tree)

Bucket_Summary

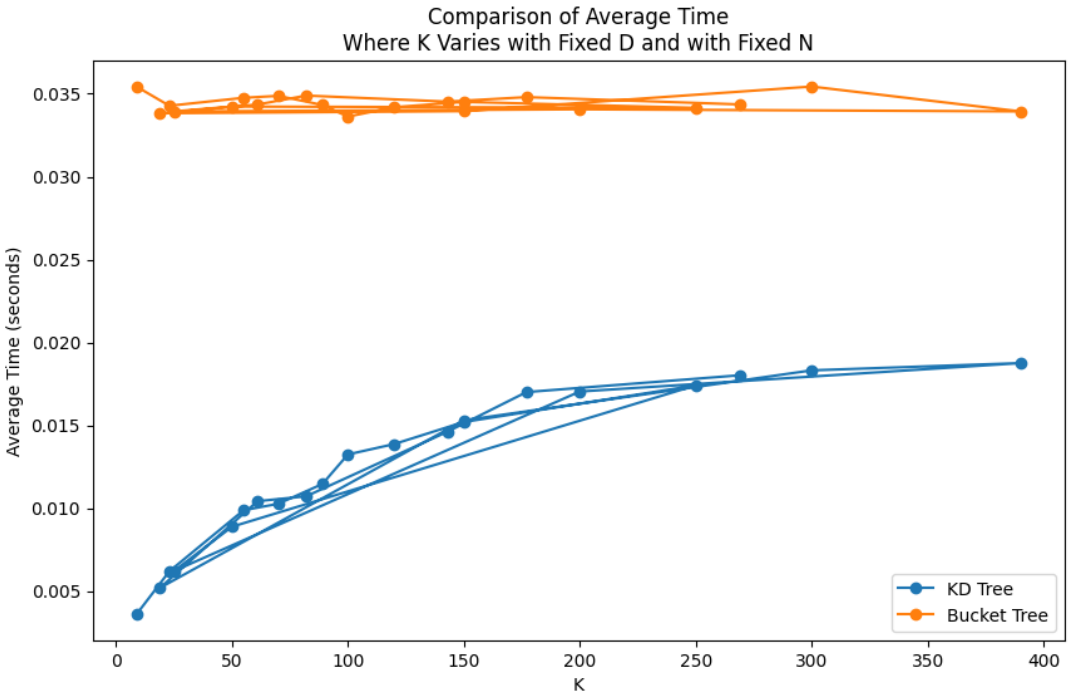
- R^2 = 0.868 : There is a moderate correlation (86.8%) indicating a moderate correlation between the dim and the average time.
- P value = 0.002
- Coef =0.0037 : (a more significant increase in search time compared to KD tree)

The linear regression results align well with the observation made above. KD Tree demonstrates a relatively smaller increase in search time with increasing dimensions, while the Bucket Tree shows a more noticeable and faster increase in search time as the dimensionality grows.





4. DATA SET4: Comparison of Average Time where K varies with a fixed D and With fixed N.



Observation

The Dataset analysis focused on comparing the average time taken by KD Tree and Bucket Tree when varying the value of K while keeping the dimension (D) fixed at 3 and the number of training points (N) fixed at 1000. The following observations were made:

Here is the observation:

- **KD Tree:** The KD Tree consistently outperformed the Bucket Tree in terms of speed when searching for different values of K within the fixed dataset. As the number of K values to search increased, the time taken by the KD Tree also increased, but it remained faster than the Bucket Tree in all tested scenarios. The KD Tree demonstrated efficient and faster search times compared to the Bucket Tree.
- **Bucket Tree:** The Bucket Tree exhibited relatively slower performance compared to the KD Tree. It showed relatively similar and constant search times for different K values, with minimal variations in some instances. However, the search time in the Bucket Tree was generally slower compared to the KD Tree across the tested range of K values.

Conclusion

Based on the observations, it can be concluded that when varying the value of K with a fixed dimension (D) and a fixed number of training points (N), the KD Tree outperforms the Bucket Tree in terms of speed and efficiency when searching for the nearest neighbors. The KD Tree consistently demonstrates faster search times, while the Bucket Tree exhibits slower performance in comparison.These findings highlight the advantages of using the KD Tree data structure when searching for nearest neighbors with varying K values in a fixed dimension

Linear regression Results

KD_model summarY

- R^2 = 0.818
- P-value = 0.000
- Coeff = 4.089e-05

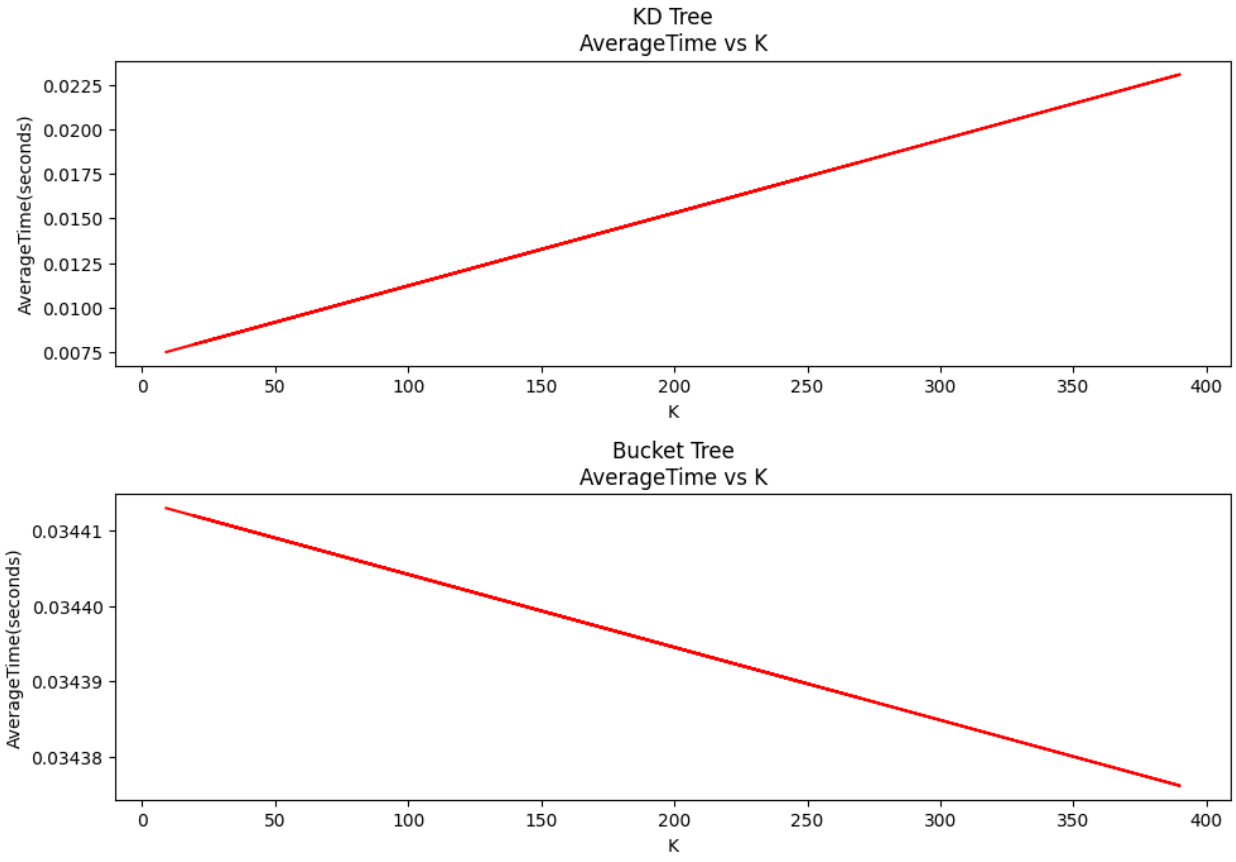
The linear regression analysis for the KD Tree resulted in an R^2 value of 0.818, indicating a moderate correlation between the varying K values and the average search time. The coefficient (Coeff) of 4.089e-05 suggests a very small positive relationship between K values and search time.

Bucket model summary

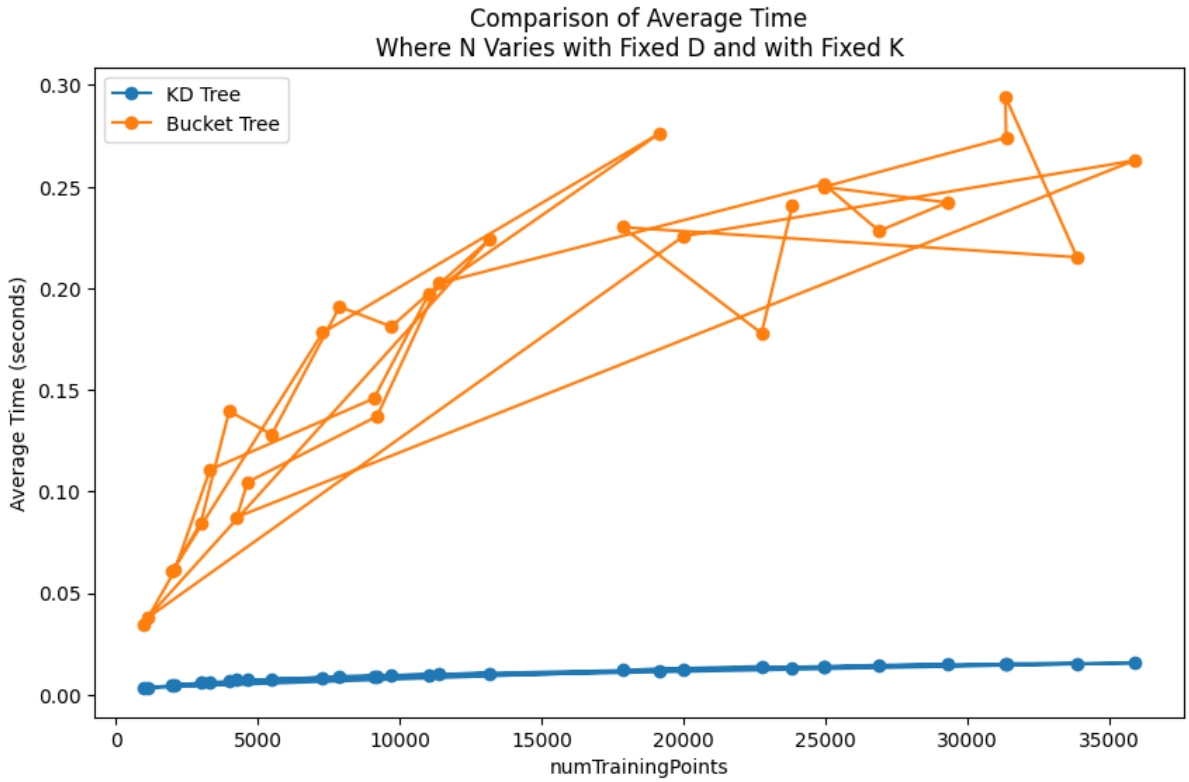
- R^2 = 0.000
- P_value = 0.930
- Coef = -9.648e-08

The linear regression analysis for the Bucket Tree yielded an R^2 value of 0.000, indicating no significant correlation between the varying K values and the average search time. The coefficient (Coeff) of -9.648e-08 suggests no noticeable relationship between K values and search time.

The linear regression results partially align with the observations made in the observation for Data Set 4. The KD Tree shows a positive but weak relationship between K values and search time, supporting the conclusion that it outperforms the Bucket Tree. However, the Bucket Tree's lack of correlation between K values and search time contradicts the report's findings.



5. DATA SET5: Comparison of Average Time where N varies with Fixed D and Fixed D and Fixed K



Observation

The DataSet 5 focused on comparing the average time that it takes the KD tree and Bucket Tree to search for a fixed K value (k = 10) in a fixed Dimension (D= 2) . While varying the number of training points.

The following observation were made:

- **KD Tree:** The KD Tree demonstrated higher speed and efficiency in searching for the K nearest neighbors across different numbers of training points. The search time remained relatively consistent with minor fluctuations throughout the tested range of N values. The KD Tree consistently provided fast search times, regardless of the number of training points.

- **Bucket Tree:** The Bucket Tree exhibited slower performance compared to the KD Tree. As the number of training points increased, the average time taken to search for the K value also increased, although not consistently. The search time showed fluctuations and varied based on the location of the K value within the grid. However, overall, the Bucket Tree took significantly more time to search for the K nearest neighbors as the number of training points increased.

Conclusion

Based on the observations, it can be concluded that when varying the number of training points (N) with a fixed dimension (D) and a fixed K value, the KD Tree outperforms the Bucket Tree in terms of speed and efficiency. The KD Tree consistently demonstrates faster search times, while the Bucket Tree exhibits slower performance, with the search time increasing as the number of training points increases.

Linear regression results

KD_Model summary

- R^2 = 0.941
- P_value = 0.000
- Coeff = 3.341e-07

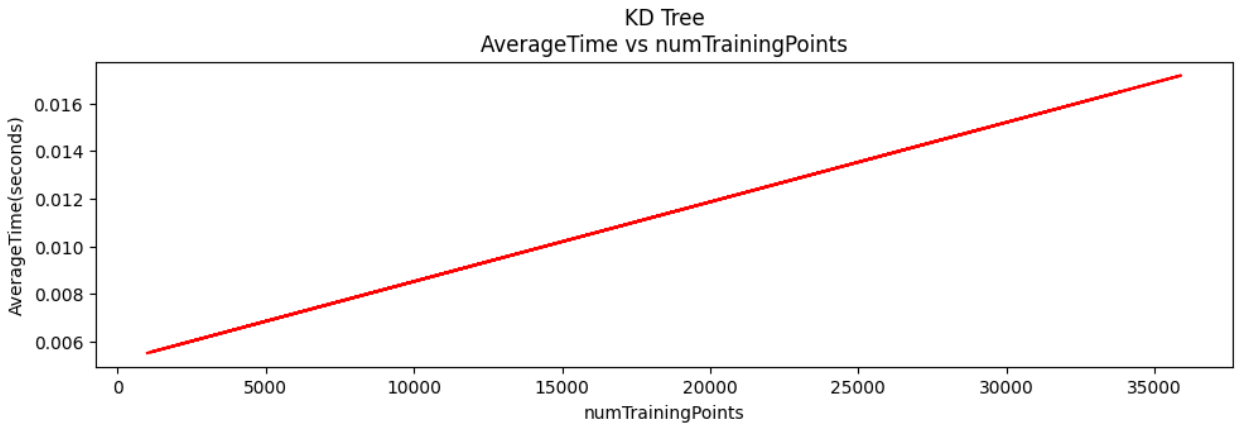
R^2 = 94.1%, indicating a strong correlation between the number of training points (N) and the average search time. The positive coefficient suggests that as the number of training points increases, the search time also increases, albeit at a relatively slower rate.

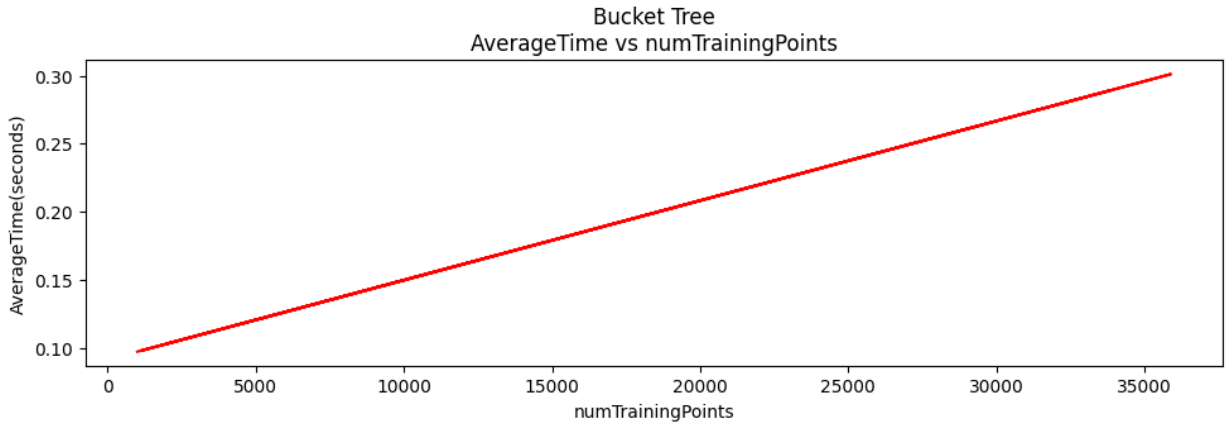
Bucket_Model Summary

- R^2 = 0.745
- P_value = 0.000
- Coeff = 5.854e-06

R^2 = 74.5%, suggests a moderate correlation between the number of training points (N) and the average search time. The coefficient indicates that the search time increases more significantly with larger N values compared to the KD Tree.

The linear regression results align with the observations made for Data Set 5. The KD Tree demonstrates a slower but more consistent increase in search time as N increases, while the Bucket Tree shows a relatively higher increase in search time with larger N values. Therefore, the KD Tree outperforms the Bucket Tree in terms of scalability with increasing N values while maintaining a reasonable search time.





Big O Analysis:

- **KD Tree** : Olog(N) where N is the number of training points.
- **Quad tree** : O(N) where N is the number of Training Points.
- **Bucket tree:** Depends on the distribution of the points. In the worst case the run time complexity is O(N) where N is the number of Training points and all points fall into the same bucket requiring linear searching into the same bucket. However, in the average case with well balanced distribution, the run time complexity is O(1).

Aspects of the data that seem unusual

- Dataset 4: The R^2 value for the Bucket Tree is close to zero, indicating a very weak correlation between the number of training points and the average search time. This unexpected result could be attributed to the specific characteristics of the data set or the implementation of the Bucket Tree in this scenario.