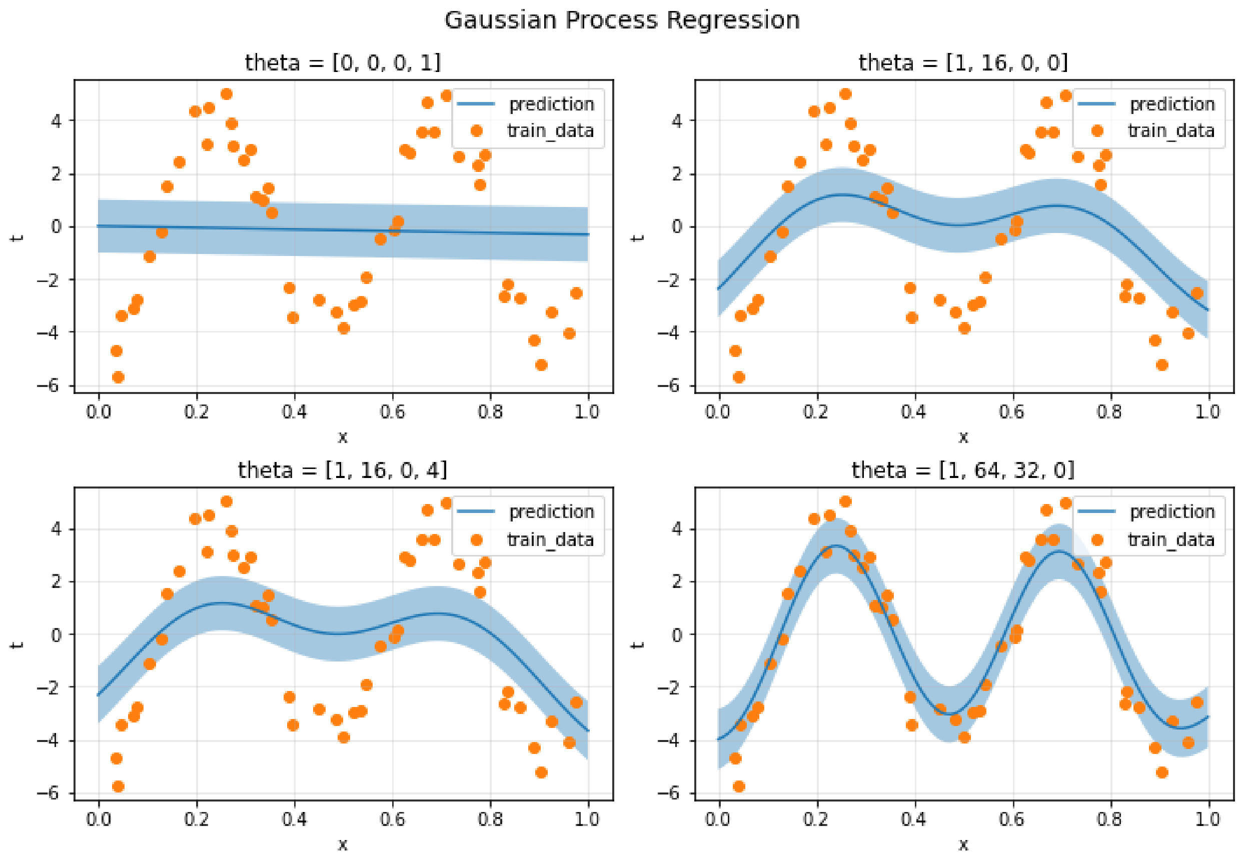
Machine Learning

Homework 3

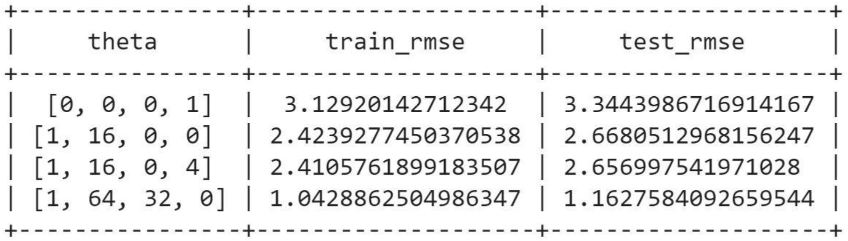
309551059 羅文慧

- ## Gaussian Process for Regression

1. **Plot the prediction result like Figure 6.8 of textbook for training set but one standard deviation instead of two and without the green curve.**



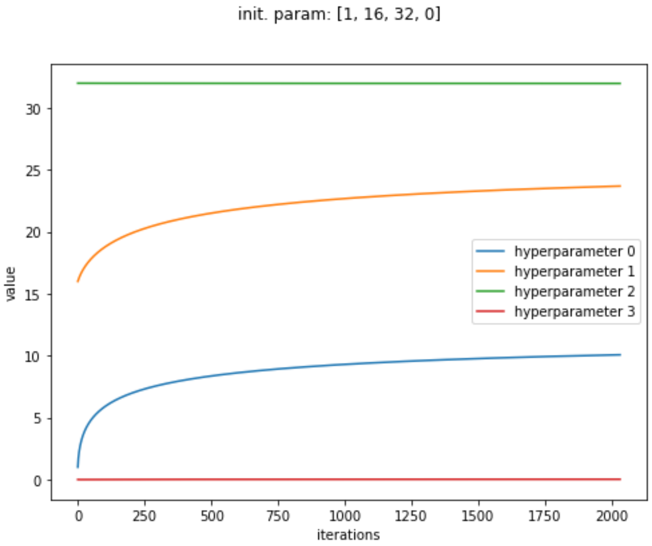
2. **Show the corresponding root-mean-square errors for both training and test sets with respect to the four kernels.**



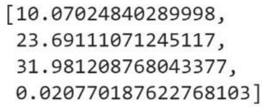
```
+----------------+--------------------+--------------------+
|     theta      |     train_rmse     |     test_rmse      |
+----------------+--------------------+--------------------+
|  [0, 0, 0, 1]  |  3.12920142712342  | 3.3443986716914167 |
| [1, 16, 0, 0]  | 2.4239277450370538 | 2.6680512968156247 |
| [1, 16, 0, 4]  | 2.4105761899183507 | 2.656997541971028  |
| [1, 64, 32, 0] | 1.0428862504986347 | 1.1627584092659544 |
+----------------+--------------------+--------------------+
```

3. **Try to tune the hyperparameters by yourself to find the best combination for the dataset. You can tune the hyperparameters by trial and error or use automatic relevance determination (ARD) in Chapter 6.4.4 of textbook.**

- *Tune the hyperparameters by ARD:*

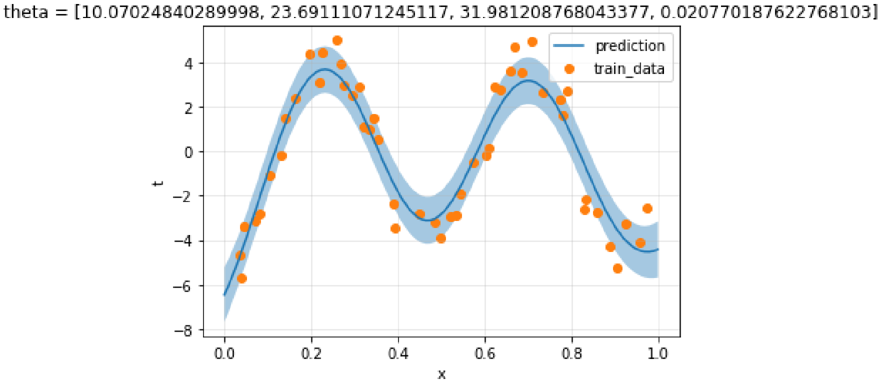


- *The best combination for the dataset:*

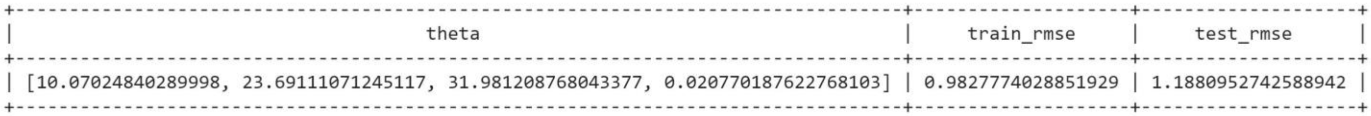


```
[10.07024840289998,
 23.69111071245117,
 31.981208768043377,
 0.020770187622768103]
```

- *The prediction result under the best combination:*



- *The corresponding root-mean-square errors:*



```
+---------------------------------------------------------------------------+--------------------+--------------------+
|                                   theta                                   |     train_rmse     |     test_rmse      |
+---------------------------------------------------------------------------+--------------------+--------------------+
| [10.07024840289998, 23.69111071245117, 31.981208768043377, 0.020770187622768103] | 0.9827774028851929 | 1.1880952742588942 |
+---------------------------------------------------------------------------+--------------------+--------------------+
```

4. Explain your findings and make some discussion.

The rmse is higher than other combinations when theta = [0, 0, 0, 1], since it's a linear kernel $x_n^T x_m$. And it's underfitting on both training and testing data.

The result is similar between theta = [1, 16, 0, 0] and [1, 16, 0, 4], rmse under theta = [1, 16, 0, 4] is slightly lower than another since it adds term $x_n^T x_m$, which make it more flexible to the dataset.

From the figure 1, we can see the prediction curve with theta = [1, 16, 32, 0] is more fit and rmse is less than the others.

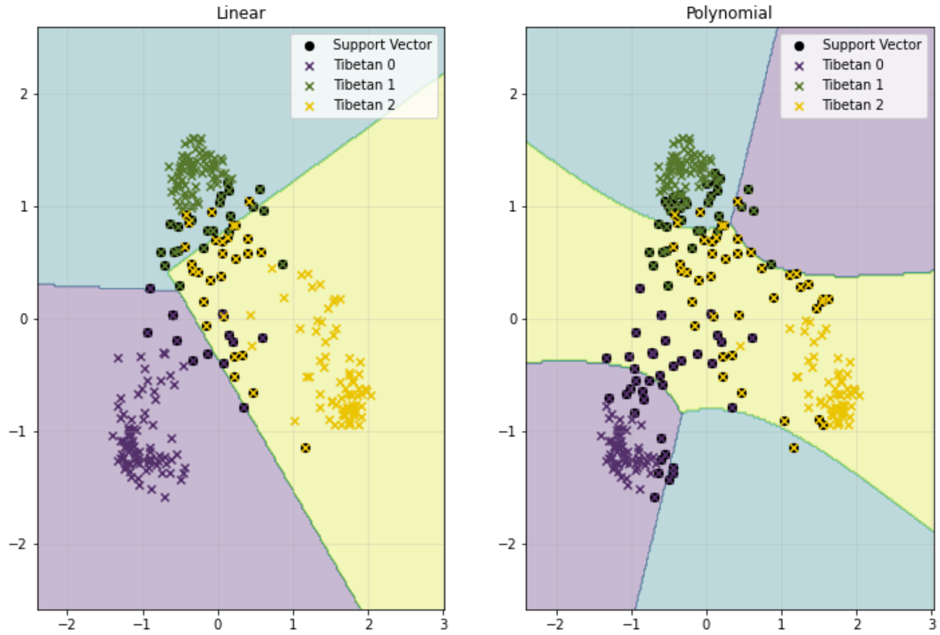
- ## Support Vector Machine

1. **Analyze the difference between two decision approaches (one-versus-the-rest and one-versus-one). Decide which one you want to use and explain why you choose this approach.**

   The one-versus-the-rest (ovr) strategy splits a multi-class classification into one binary classification problem per class. The one-versus-one (ovo) strategy splits a multi-class classification into one binary classification problem per each pair of classes.

   A possible drawback of ovr is that it requires one model to be created for each class, and if the training dataset is 1:M, it would exist bias. So, I choose ***ovo*** *as my decision approach in the following sections.*

2. **Use the dataset to build a SVM with linear kernel to do multi-class classification. Then plot the corresponding decision boundary and support vectors.**
3. **Repeat (2) with polynomial kernel (degree = 2).**



4. **Discuss the difference between (2) and (3).**

   Training data has been normalized, and using one-versus-one as decision approach. Use sklearn to fit data and get coefficients, and distribute multipliers to their corresponding classifiers. The prediction is based on the classifier's weight and bias. Finally, the result is voted by classifiers.

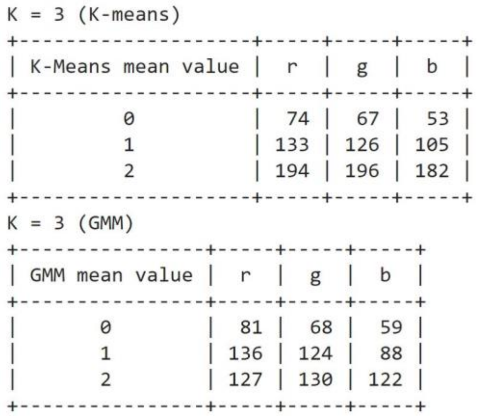
   The figure shows linear kernel can classify the dataset pretty well, and all boundaries are straight lines.

   Polynomial kernel performs not well compared to linear one, and boundaries are smoother curves, which along with Tibetan's distribution.

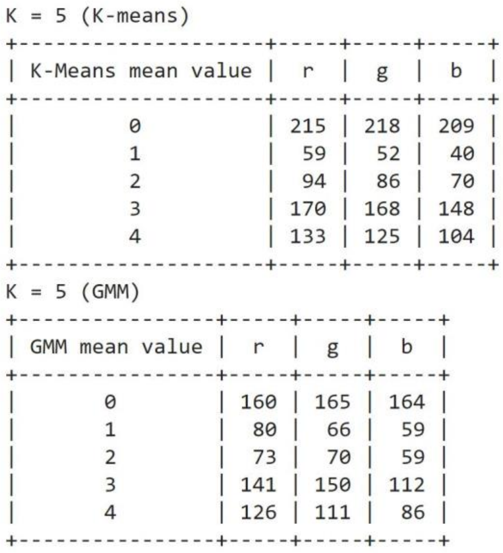
   The reason of polynomial kernel's misclassification may be it projects data to feature space, but can't classify data clearly.

- ## Gaussian Mixture Model

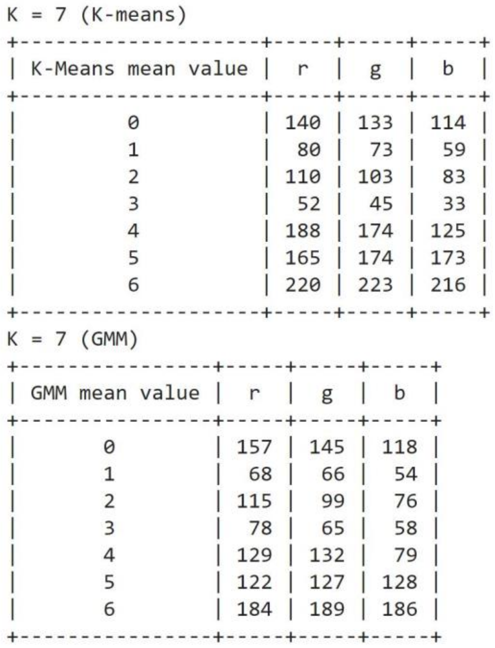
1. **Please build a K-means model and show the table of estimated {μk}.**



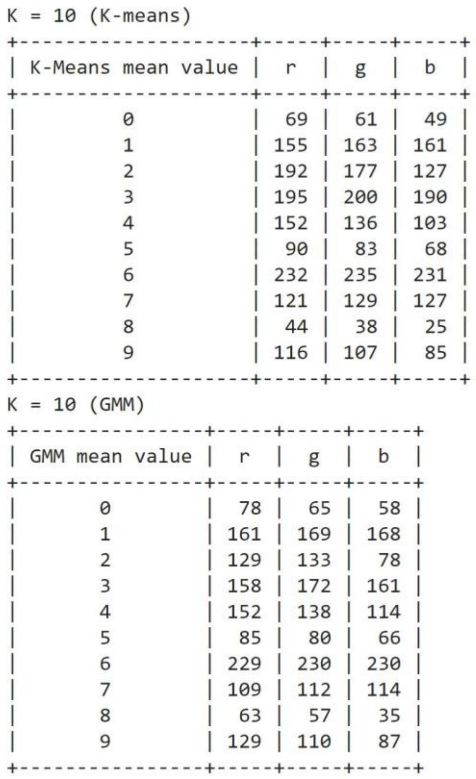
```
K = 3 (K-means)
+--------------------+-----+-----+-----+
| K-Means mean value |  r  |  g  |  b  |
+--------------------+-----+-----+-----+
|         0          |  74 |  67 |  53 |
|         1          | 133 | 126 | 105 |
|         2          | 194 | 196 | 182 |
+--------------------+-----+-----+-----+
K = 3 (GMM)
+----------------+-----+-----+-----+
| GMM mean value |  r  |  g  |  b  |
+----------------+-----+-----+-----+
|       0        |  81 |  68 |  59 |
|       1        | 136 | 124 |  88 |
|       2        | 127 | 130 | 122 |
+----------------+-----+-----+-----+
```



```
K = 5 (K-means)
+--------------------+-----+-----+-----+
| K-Means mean value |  r  |  g  |  b  |
+--------------------+-----+-----+-----+
|         0          | 215 | 218 | 209 |
|         1          |  59 |  52 |  40 |
|         2          |  94 |  86 |  70 |
|         3          | 170 | 168 | 148 |
|         4          | 133 | 125 | 104 |
+--------------------+-----+-----+-----+
K = 5 (GMM)
+----------------+-----+-----+-----+
| GMM mean value |  r  |  g  |  b  |
+----------------+-----+-----+-----+
|       0        | 160 | 165 | 164 |
|       1        |  80 |  66 |  59 |
|       2        |  73 |  70 |  59 |
|       3        | 141 | 150 | 112 |
|       4        | 126 | 111 |  86 |
+----------------+-----+-----+-----+
```

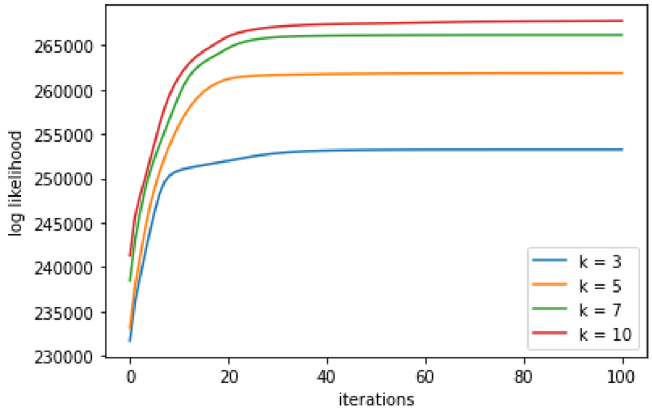


```
K = 7 (K-means)
+--------------------+-----+-----+-----+
| K-Means mean value |  r  |  g  |  b  |
+--------------------+-----+-----+-----+
|         0          | 140 | 133 | 114 |
|         1          |  80 |  73 |  59 |
|         2          | 110 | 103 |  83 |
|         3          |  52 |  45 |  33 |
|         4          | 188 | 174 | 125 |
|         5          | 165 | 174 | 173 |
|         6          | 220 | 223 | 216 |
+--------------------+-----+-----+-----+
K = 7 (GMM)
+----------------+-----+-----+-----+
| GMM mean value |  r  |  g  |  b  |
+----------------+-----+-----+-----+
|       0        | 157 | 145 | 118 |
|       1        |  68 |  66 |  54 |
|       2        | 115 |  99 |  76 |
|       3        |  78 |  65 |  58 |
|       4        | 129 | 132 |  79 |
|       5        | 122 | 127 | 128 |
|       6        | 184 | 189 | 186 |
+----------------+-----+-----+-----+
```



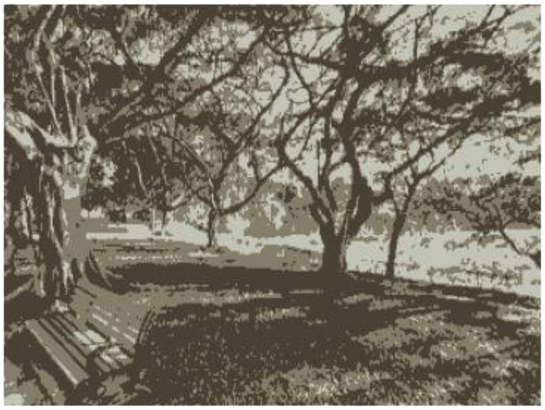
```
K = 10 (K-means)
+--------------------+-----+-----+-----+
| K-Means mean value |  r  |  g  |  b  |
+--------------------+-----+-----+-----+
|         0          |  69 |  61 |  49 |
|         1          | 155 | 163 | 161 |
|         2          | 192 | 177 | 127 |
|         3          | 195 | 200 | 190 |
|         4          | 152 | 136 | 103 |
|         5          |  90 |  83 |  68 |
|         6          | 232 | 235 | 231 |
|         7          | 121 | 129 | 127 |
|         8          |  44 |  38 |  25 |
|         9          | 116 | 107 |  85 |
+--------------------+-----+-----+-----+
K = 10 (GMM)
+----------------+-----+-----+-----+
| GMM mean value |  r  |  g  |  b  |
+----------------+-----+-----+-----+
|       0        |  78 |  65 |  58 |
|       1        | 161 | 169 | 168 |
|       2        | 129 | 133 |  78 |
|       3        | 158 | 172 | 161 |
|       4        | 152 | 138 | 114 |
|       5        |  85 |  80 |  66 |
|       6        | 229 | 230 | 230 |
|       7        | 109 | 112 | 114 |
|       8        |  63 |  57 |  35 |
|       9        | 129 | 110 |  87 |
+----------------+-----+-----+-----+
```

2. **Use {μk} calculated by the K-means model as means, and calculate the corresponding variances and mixing coefficient πk for initialization of GMM. Optimize the model by maximizing the log likelihood function log p(x|π, μ, σ2) through EM algorithm. Please show the graph of Log likelihood at different iterations for K = 3, 5, 7, 10 respectively.**

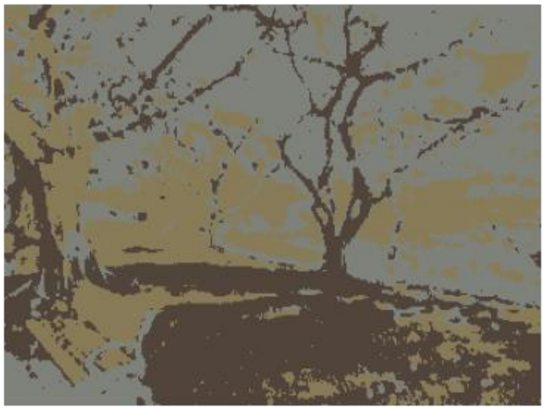


3. Repeat step (1) and (2) for K = 3, 5, 7, and 10. Please show the resulting images in your report.

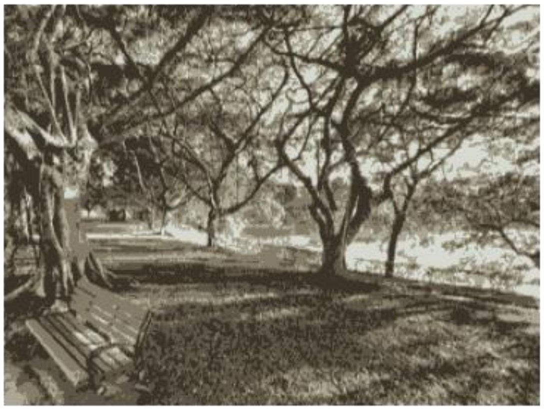
K-means (K = 3)



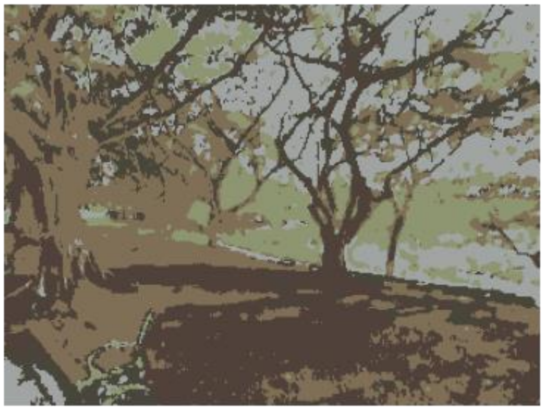
GMM (K = 3)



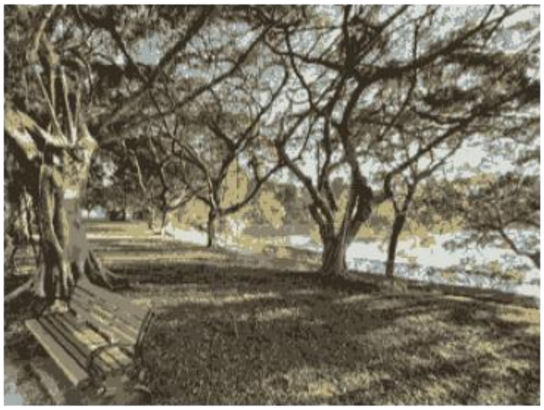
K-means (K = 5)



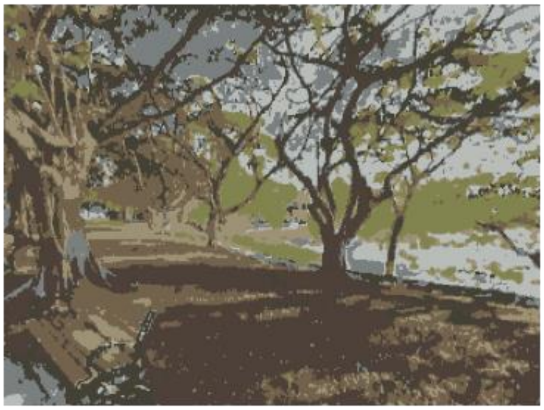
GMM (K = 5)



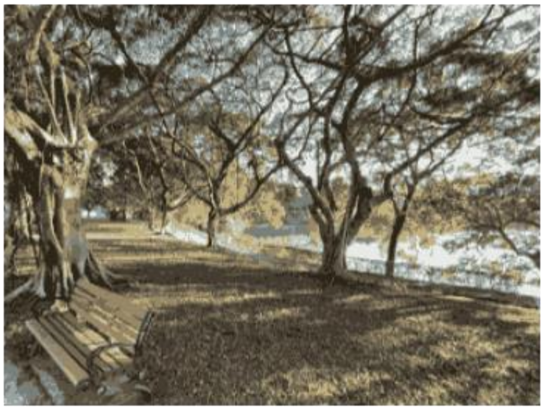
K-means (K = 7)



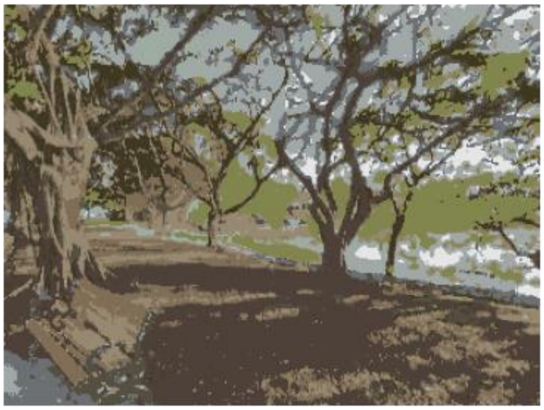
GMM (K = 7)



K-means (K = 10)



GMM (K = 10)



4. **You can make some discussion about what is crucial factor to affect the output image and explain the reason?**

   From figures above, K-means can present images that color and contour are close to original one. Images produced by GMM from lower K to higher K is like a painting process, which draw the contour of trees and branches first, then add the detailed items later.

   When K = 10, there're many color blocks in GMM image, it may due to K-means puts similar color to the same class, however GMM starts from probability.