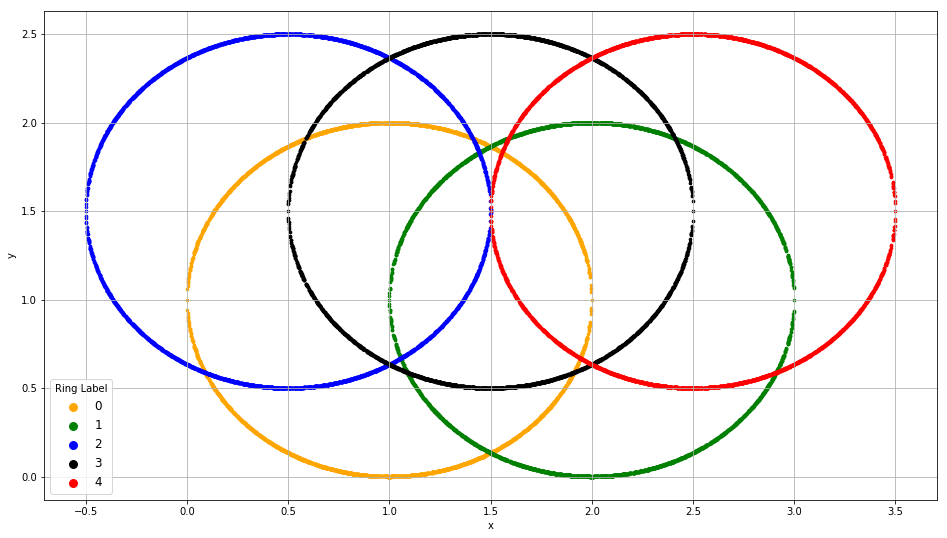
CS 584-04: Machine Learning

Spring 2019 Assignment 5

# The Five Rings Data

1. You will analyze the FiveRing.csv data for all the questions.
2. This data has 20,010 observations and three numeric fields, namely, x, y, and ring.
3. The fields x and y are the x-coordinate and the y-coordinate of the rings respectively.
4. The field ring indicates to which ring the coordinates belong.
5. The rings are labelled 0, 1, 2, 3, and 4.
6. The graph below shows the five rings.



# Misclassification Rate

Let be the predicted probability that the *i*-th observation will belong to the *j*-th ring. The predicted ring for the *i*-th observation is the smallest ring label which has the highest predicted probability. The following examples illustrate how the predicted ring is determined.

* Suppose , then the predicted ring is 3 because is the highest value among the five probabilities.
* Suppose , then the predicted ring is 0. Although are tied for the highest probability, the smallest ring label is 0.

An observation is *misclassified* if the predicted ring label is different from the observed ring label. The Misclassification Rate is the proportion of all the observations which are misclassified.

# Root Average Squared Error (RASE)

The Root Average Squared Error is

where

* is the number of observations.
* if the ring label of the *i*-th observation is *j*. Otherwise, .

# Question 1 (100 points)

You will build the multinomial logistic model according to the specifications below. You will use the Misclassification Rate and the Root Average Squared Error to assess the performance of your model.

* The nominal target variable is ring
* The predictors are x and y.
* The model will have the Intercept terms.
* The maximum number of iterations is 1000.

Build and assess the multinomial logistic model using all 20,010 observations **without** bagging and answer the following questions.

1. **(10 points). List the parameter estimates (round to four decimal places) in a table. The rows are the Intercept, the predictor x, and the predictor y. The columns are the ring labels.**

**Ans:**

Model Parameter Estimates:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **0** | **1** | **2** | **3** |
| **const** | -4.435535 | 0.187691 | -2.701434 | -8.702577 |
| **x** | 2.953259 | -1.467861 | 1.455439 | 4.438213 |
| **y** | 0.000669 | 0.720070 | 0.694733 | 0.719375 |

1. **(10 points). What is the Misclassification Rate?**

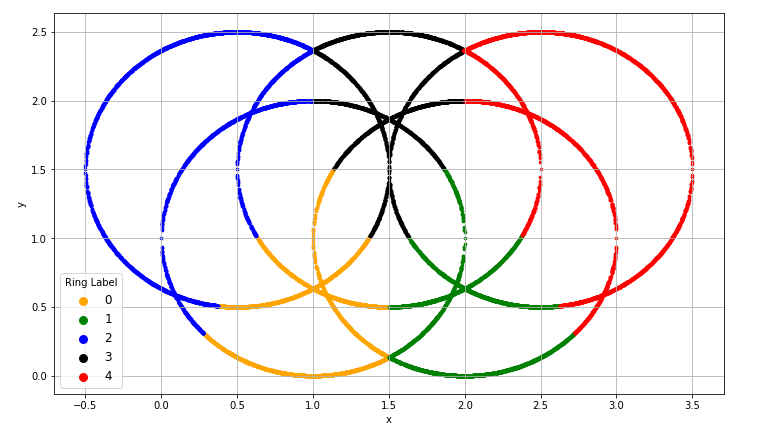
**Ans:** Misclassification Rate = 0.588006

1. **(10 points). What is the Root Average Squared Error?**

**Ans:** Root Average Squared Error = 0.558195

1. **(10 points). Redraw the above picture (i.e., the field y on the vertical axis and the field x on the horizontal axis), however, use the predicted ring label for coloring. The coloring scheme is 0 = orange, 1 = green, 2 = blue, 3 = black and 4 = red.**

**Ans:**



Apply the Bagging technique, build and assess the multinomial logistic model using all 20,010 observations. The initial random seed is 20190430. Try number of bootstraps equals to 10, 20, 30, 40, 50, 60, 70, 80, 90 and 100.

**Note: you are not allowed to use any functions (e.g., BaggingClassifier) in the sklearn.ensemble module to perform Bagging. Instead, you must write your Python codes to implement the Bagging algorithm**.

1. **(40 points). List the Misclassification Rate and the Root Average Squared Error of the bootstrap results. The columns are the two metrics. The rows are the number of bootstraps. Also, include the no-bootstrap (i.e., zero number of bootstrap) metrics.**

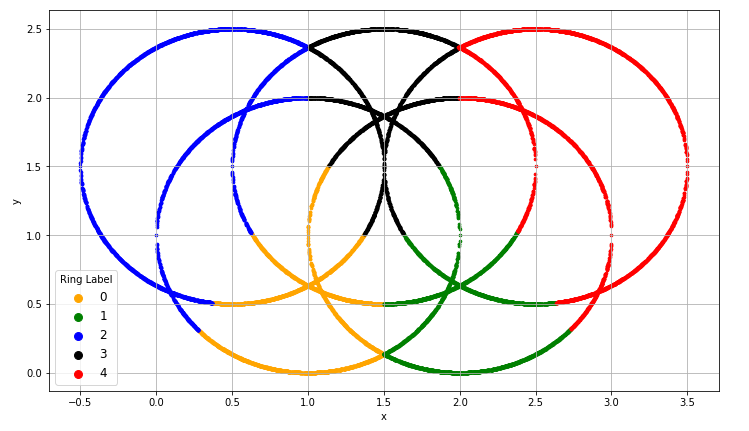
**Ans:**

|  |  |  |
| --- | --- | --- |
| **Maximum Iterations** | **Misclassification Rate** | **Root Average Squared Error** |
| 0 (No Bagging) | 0.588006 | 0.558195 |
| 10 | 0.588156 | 0.558191 |
| 20 | 0.588256 | 0.558152 |
| 30 | 0.588156 | 0.558136 |
| 40 | 0.588156 | 0.558139 |
| 50 | 0.588106 | 0.558143 |
| 60 | 0.588056 | 0.558147 |
| 70 | 0.587906 | 0.558149 |
| 80 | 0.588106 | 0.558152 |
| 90 | 0.588056 | 0.558152 |
| 100 | 0.588056 | 0.558151 |

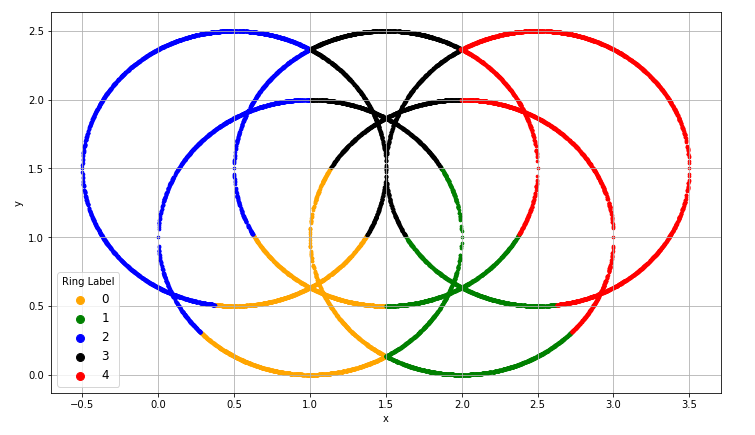
1. **(10 points). Redraw the above picture (i.e., the field y on the vertical axis and the field x on the horizontal axis), however, use the predicted ring label for coloring. The coloring scheme is 0 = orange, 1 = green, 2 = blue, 3 = black and 4 = red. There should be ten pictures, one for each set of bootstraps.**

**Ans:**

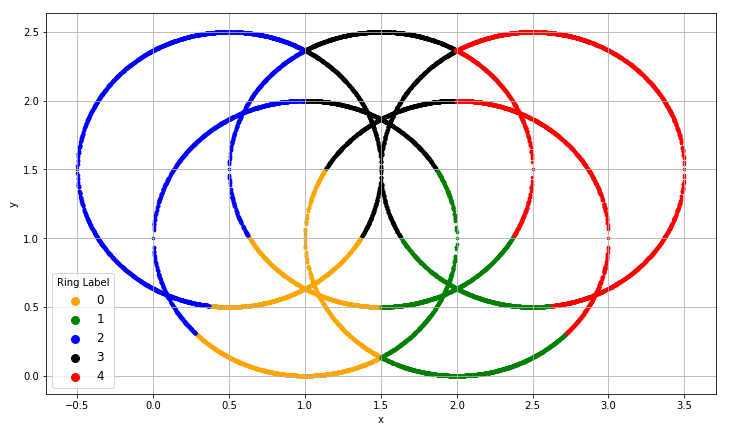
**10 iterations:**



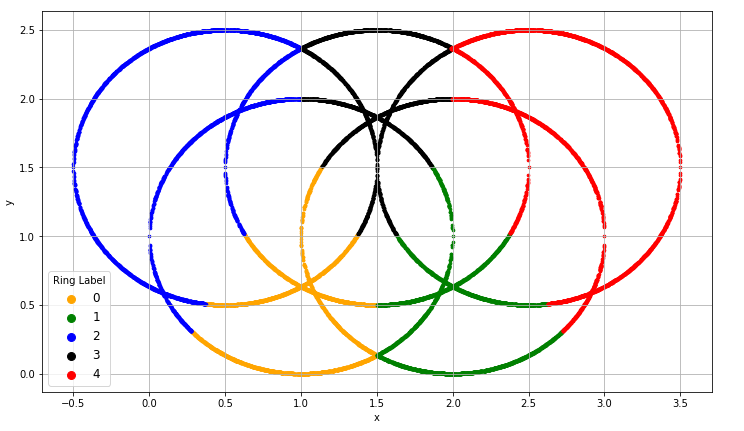
**20 iterations:**



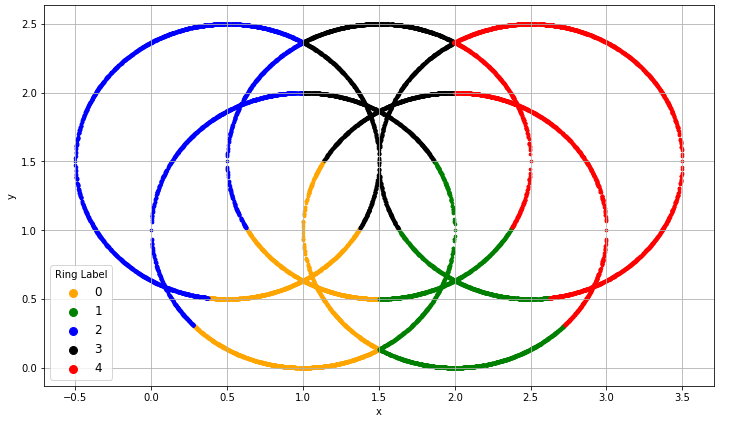
**30 iterations:**



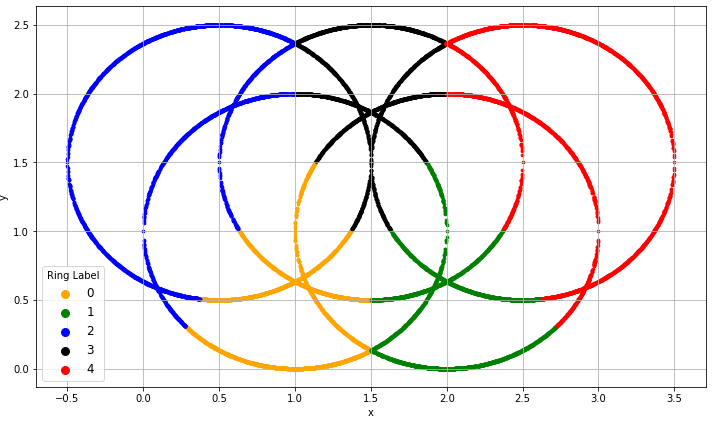
**40 iterations:**



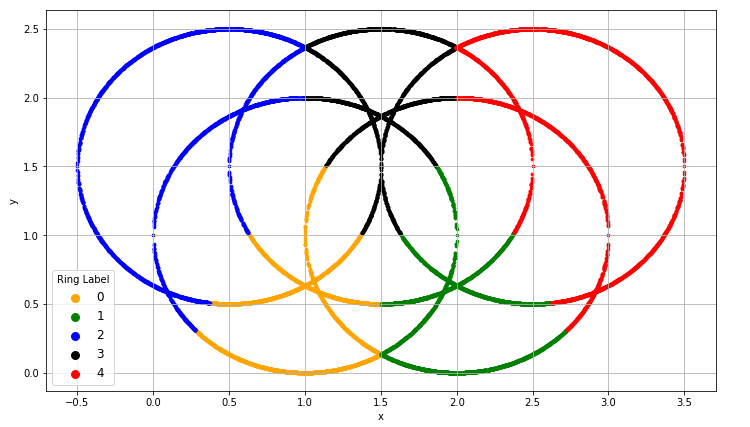
**50 iterations:**



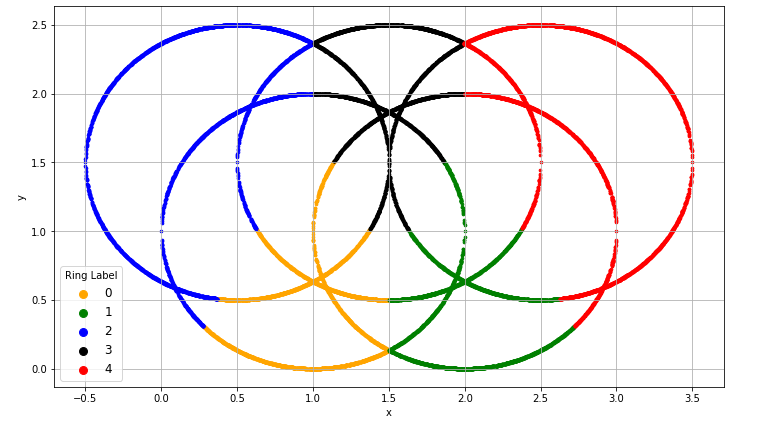
**60 iterations:**



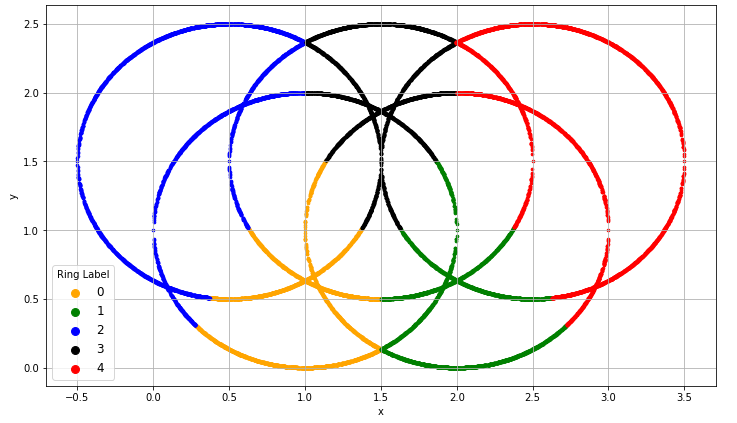
**70 iterations:**



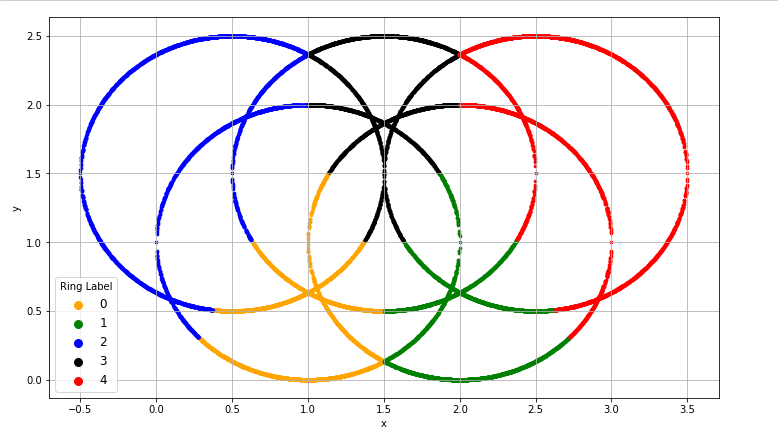
**80 iterations:**



**90 iterations:**



**100 iterations:**



1. **(10 points). Compare the results between the bagging results and the non-bagging results. Briefly comment on the comparison.**

**Ans:** Basically, bagging method is used to reduce error and it is not much known for getting more accurate results. So, that is what is seen here also the error is decreasing but not much significant difference is observed in the predictions

For bagging in this assignment, bootstraps created with the given number of iterations and the probability from the model was calculated each time and then the average was taken of probability obtained from each iteration. Thereafter, the classes were predicted to calculate metrics

So, it can be seen through the table and graphs above that misclassification rate for bagging increases a little bit from non-bagging results, but then it decreases after some iterations and remains almost constant. For Root Average Squared Error, it is seen to be decreasing from non-bagging results to bagging for 100 iterations. There is although no significant change in the graphs as bagging provides no significant amount of change in predictions

Thus, Bagging is helping with reducing the error obtained through model with no – bagging results in this assignment

# Question 2 (100 points)

You will build the classification tree model and then apply the Adaptive Boosting technique. You will use the Misclassification Rate and the Root Average Squared Error to assess the performance of your model. The classification tree model should be built according to the specifications below.

* The nominal target variable is Ring
* The predictors are x and y.
* The splitting criterion is Entropy
* The maximum depth is 2.
* The random state value is 20190415

Build and assess the classification tree model using all 20,010 observations **without** boosting and answer the following questions.

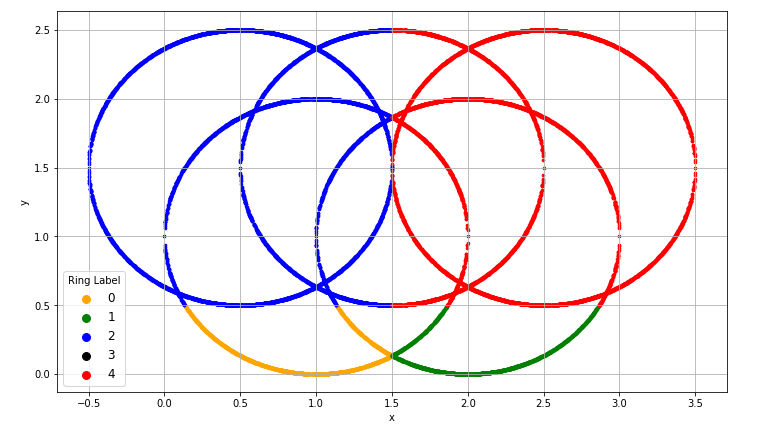
1. **(10 points). What is the Misclassification Rate?**

**Ans:** Misclassification Rate = 0.464118

1. **(10 points). What is the Root Average Squared Error?**

**Ans:** Root Average Squared Error = 0.550792

1. **(10 points). Redraw the above picture (i.e., the field y on the vertical axis and the field x on the horizontal axis), however, use the predicted ring label for coloring. The coloring scheme is 0 = orange, 1 = green, 2 = blue, 3 = black and 4 = red.**

**Ans:** 

Build and assess the classification tree model using all 20,010 observations **with** boosting with initial random seed 20190430. Try the maximum number of iterations equals to 100, 200, 300, 400, 500, 600, 700, 800, 900, and 1000. The case weights are determined as follows.

* The case weights are initialized to 1.
* After each iteration, the case weight is the for a misclassified observation. The case weight is for a correctly classified observation.

The iteration stops if either the Misclassification Rate is zero or the maximum number of iterations is reached. The aggregated predicted probabilities from a set of boosting are the weighted mean of the predicted probabilities of the iterations. The weights are the accuracy (i.e., one minus the misclassification rate) of the iterations.

**Note: you are not allowed to use any functions (e.g., AdaBoostClassifier) in the sklearn.ensemble module to perform boosting. Instead, you must write your Python codes to implement the Boosting algorithm**.

1. **(50 points). List the Misclassification Rate and the Root Average Squared Error of the aggregated boosting results. The columns are the number of iterations performed and the two metrics. The rows are the maximum number of iterations. Also, include the no-boosting metrics.**

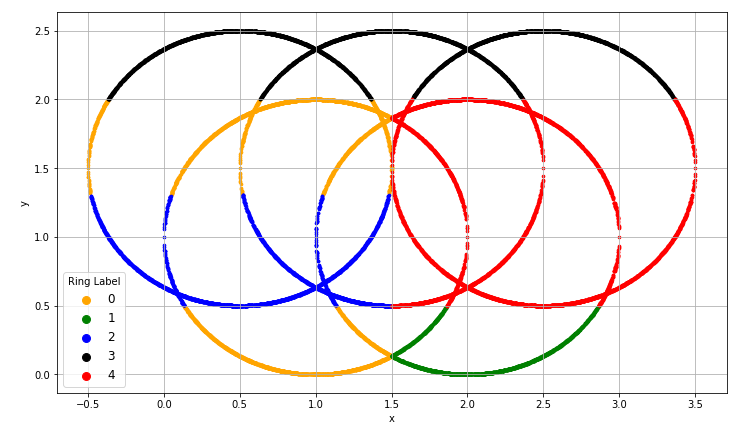
**Ans:**

|  |  |  |
| --- | --- | --- |
| **Maximum Iterations** | **Misclassification Rate** | **Root Average Squared Error** |
| 0 | 0.464118 | 0.550792 |
| 10 | 0.493453 | 0.552826 |
| 100 | 0.549325 | 0.550271 |
| 200 | 0.549325 | 0.550166 |
| 300 | 0.549325 | 0.550131 |
| 400 | 0.549325 | 0.550114 |
| 500 | 0.549325 | 0.550104 |
| 600 | 0.549325 | 0.550098 |
| 700 | 0.549325 | 0.550093 |
| 800 | 0.549325 | 0.550089 |
| 900 | 0.549325 | 0.550086 |
| 1000 | 0.549325 | 0.550084 |

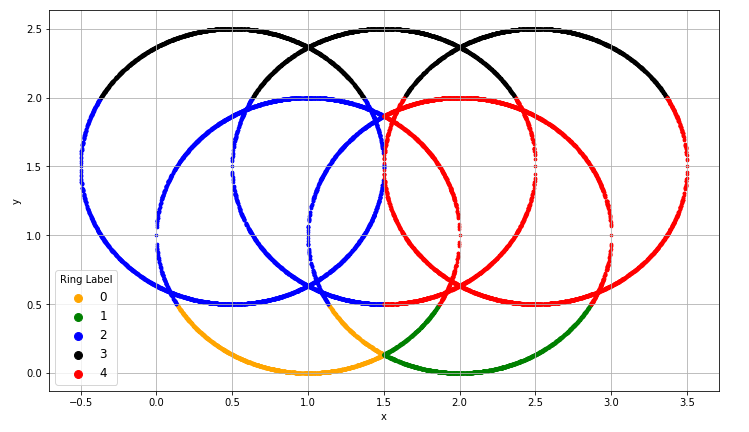
1. **(10 points). Redraw the above picture (i.e., the field y on the vertical axis and the field x on the horizontal axis), however, use the predicted ring label for coloring. The coloring scheme is 0 = orange, 1 = green, 2 = blue, 3 = black and 4 = red. There should be ten pictures, one for each set of boosting.**

**Ans:** I have included the figure of 10 iterations also to show how the colors of the rings are changing and after 50 iterations, it is kind of constant but if I go to 5000 iterations or more there might be further changes and will get better results of boosting

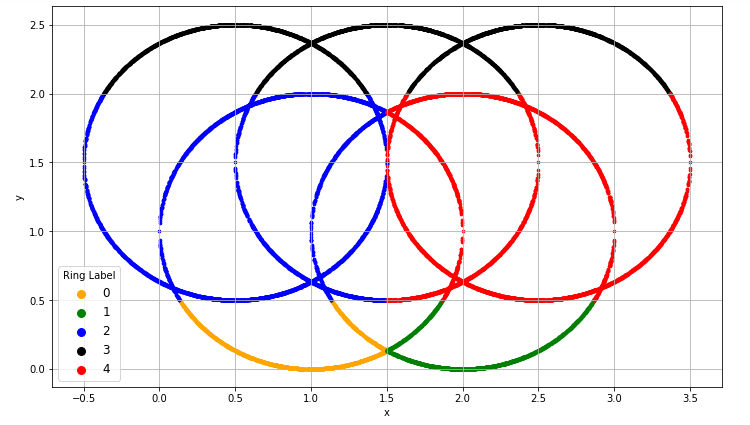
**10 iterations:**



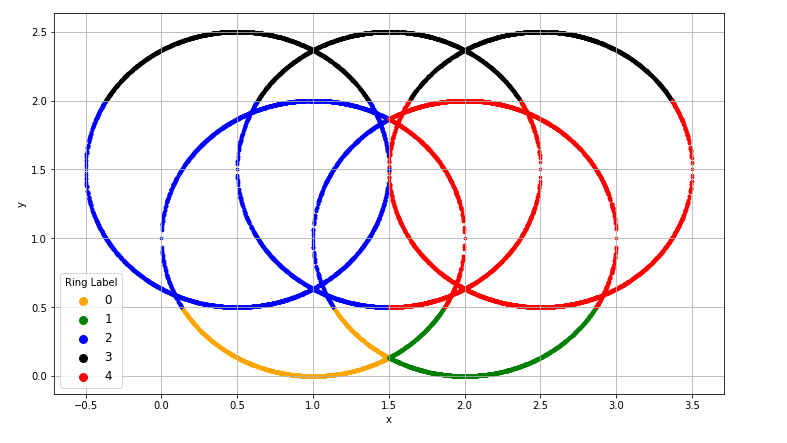
**100 iterations:**



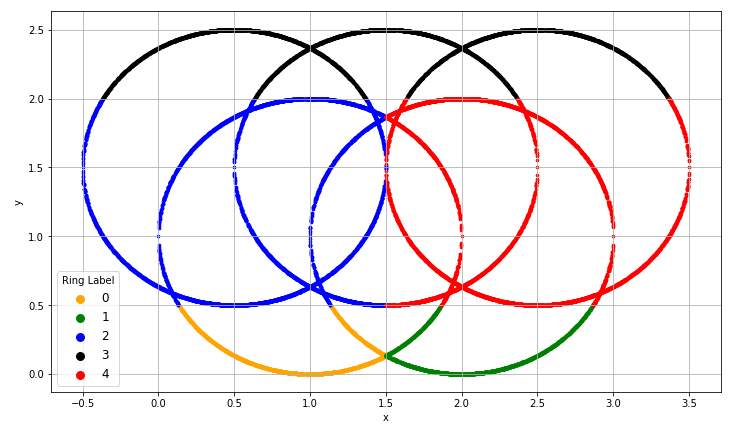
**200 iterations:**



**300 iterations:**



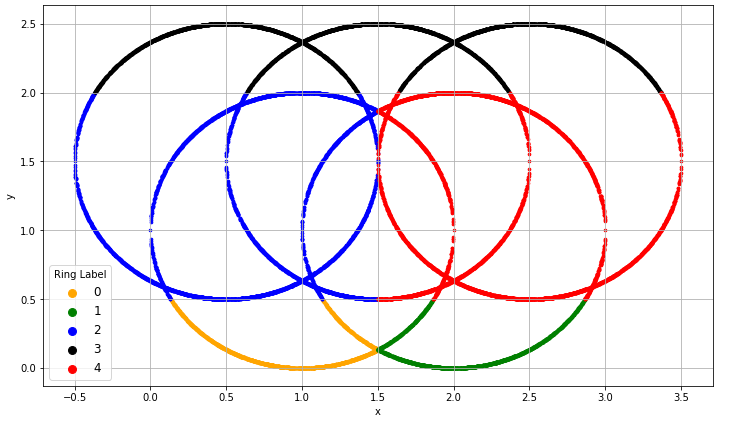
**400 iterations:**



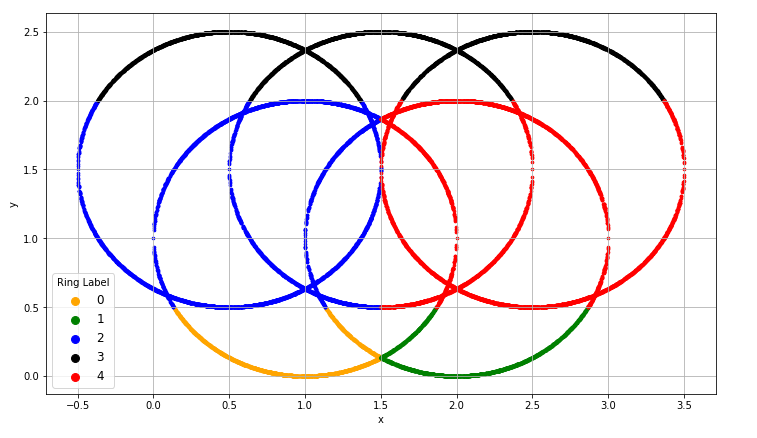
**500 iterations:**



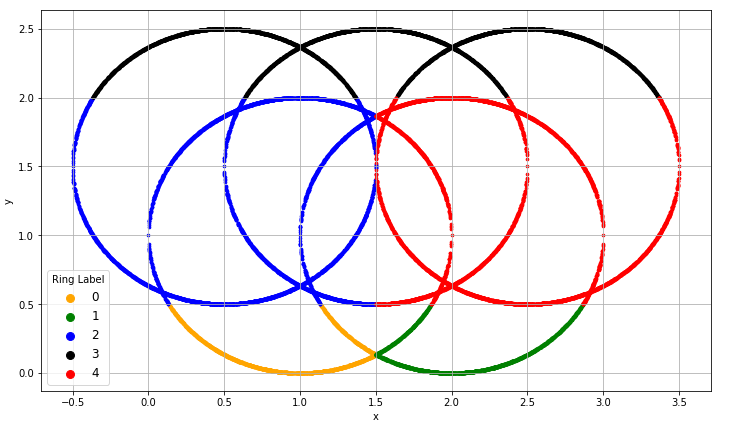
**600 iterations:**



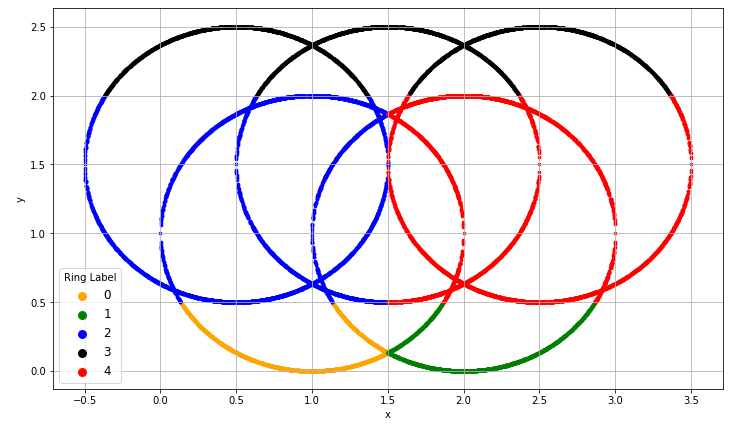
**700 iterations:**



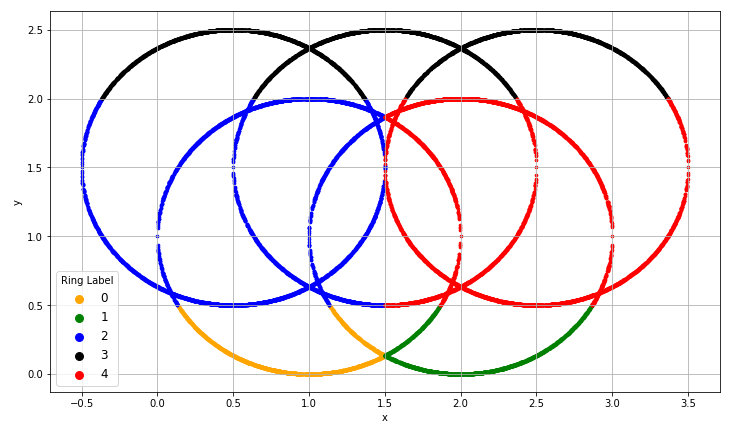
**800 iterations:**



**900 iterations:**



**1000 iterations:**



As it can be seen from the graphs above that the we started seeing the black ring after boosting of model which was not seen in original model

1. **(10 points). Compare the results between the boosting results and the non-boosting results. Briefly comment on the comparison.**

**Ans:** Boosting is generally used to get more accurate results and for making the model better that is we can get better prediction results when we apply Boosting to the model. It can be seen that although the misclassification rate increases here but we start seeing the black ring in boosting results which was not present in non-boosting result.

For boosting, we take equal weights (that is one) at first, then we calculate the error and then start assigning higher weights to the points in the data which are misclassified. The results of each iteration were aggregated on the basis of accuracy into single model

Thus, for non-boosting result the Misclassification Rate is low, Root Average Squared Error is high, and we cannot see all the five rings in graph with the original model. After performing boosting, the Misclassification Rate increases and remains constant till certain iterations, Root Average Squared Error is decreasing, and we start seeing all the colors of ring in the graph as we go higher in maximum number of iterations. As the size of the data is very large, we need more iterations (like 5000 and more) to get better results and see the rings more clearly