|  |  |
| --- | --- |
| Algorithm | Results |
| Original:  svm\_model <- svm(Class ~ ., data = trainData\_SMOTE[, c(selected\_features\_rfe, "Class")],  kernel = "radial",  cost = 1,  probability = TRUE) |  |
| svm\_model <- svm(Class ~ ., data = trainData\_SMOTE[, c(selected\_features\_rfe, "Class")], kernel = "radial", cost = 1, degree = 3, type = "C-classification", tolerance = 0.0005, probability = TRUE) |  |
| svm\_model <- svm(Class ~ ., data = trainData\_SMOTE[, c(selected\_features\_rfe, "Class")],  kernel = "radial",  cost = c(0.1, 1, 10, 100),  gamma = c(0.01, 0.1, 1, 10),  degree = 3,  type = "C-classification",  tolerance = 0.0005,  probability = TRUE) |  |
| svm\_model <- svm(Class ~ ., data = trainData\_SMOTE[, c(selected\_features\_rfe, "Class")],  kernel = "radial",  cost = c(0.1, 1, 10, 100,150),  gamma = c(0.001,0.01, 0.1, 1, 10),  degree = 3,  type = "C-classification",  tolerance = 0.0005,  probability = TRUE)  print(svm\_model) |  |
| svm\_model <- svm(Class ~ ., data = trainData\_SMOTE[, c(selected\_features\_rfe, "Class")],  kernel = "radial",  cost = tune\_result$best.parameters$cost,  gamma = tune\_result$best.parameters$gamma,  type = "C-classification",  class.weights = c("0"=1, "1"=1.5),  probability = TRUE) |  |
| **svm\_model** <- svm(Class ~ ., data = trainData\_SMOTE[, c(selected\_features\_rfe, "Class")],  kernel = "radial",  cost = c(0.1, 1, 10, 100,150),  gamma = c(0.001,0.01, 0.1, 1, 10),  degree = 3,  type = "C-classification",  tolerance = 0.0005,  class.weights = c("0"=1, "1"=1.5),  probability = TRUE) |  |
| svm\_model <- svm(Class ~ ., data = trainData\_SMOTE[, c(selected\_features\_rfe, "Class")],  kernel = "radial",  cost = c(0.1, 1, 10, 100,150),  gamma = c(0.001,0.01, 0.1, 1, 10),  degree = 3,  type = "C-classification",  tolerance = 0.0005,  class.weights = c("0"=1, "1"=2),  probability = TRUE) |  |
| svm\_model <- svm(Class ~ ., data = trainData\_SMOTE[, c(selected\_features\_rfe, "Class")],  kernel = "radial",  cost = c(0.1, 1, 10, 100,150),  gamma = c(0.001,0.01, 0.1, 1, 10),  degree = 3,  type = "C-classification",  tolerance = 0.0001,  class.weights = c("0"=1, "1"=1.5),  probability = TRUE) |  |

|  |  |
| --- | --- |
| svm\_model <- svm(Class ~ ., data = trainData\_SMOTE[, c(selected\_features\_rfe, "Class")],  kernel = "radial",  gamma = c(0.001,0.01, 0.1),  cost = c(0.1, 1, 10, 100,150),  class.weights = c("0"=1, "1"=1.1),  probability = TRUE)  print(svm\_model) |  |
| With rose included:  svm\_model <- svm(Class ~ ., data = trainData\_SMOTE[, c(selected\_features\_rfe, "Class")],  kernel = "radial",  gamma = c(0.001,0.01, 0.1),  class.weights = c("0"=1, "1"=1.1),  probability = TRUE)  print(svm\_model) |  |
|  |  |

Graph the data for each class

Add the tunes for RFE in smote

And try the new data for SVM, Logistic Regression,

How does he want to show the negative and positive class represented in the results.

How do we calculate the Kappa with different classes

Does scaling depend on the model being used? Do you want the same approach for all of the models