**Image Segmentation**

University of Central Missouri

Data Mining CS4630

Group Members

Ravali Anthati(700688033)

Rajyalakshmi Linga(700689106)

Kishan Polekar(700663012)

**Individual Contribution to the Project**

We started working on this project together by choosing the Image Segmentation dataset from UCI Machine Learning Repository with 2310 Instances and 19 attributes, once the dataset was decided then we began to work on the classification Techniques like Naïve Bayes Classification, Decision tree, Hold-out, Bagging, Random-Forest Classification and SVM using different kernels with different costs and gamma and degree respectively.

**Ravali Anthati** – She worked on preparing the slides for Introduction part continuing with the R-code of Naïve Bayes Classification.

**Rajyalakshmi Linga** – She started working on the R-code for Decision Tree with Hold-out and Bagging Random Forest Classification and preparing slides according to it.

**Kishan Polekar** – He Started to work on the R-code for Support Vector Machines using different kernels(Linear, Radial, Polynomial) with different costs and gamma and Degree and preparing slides for that part.

So Finally by combining the R-code and Slides all together and executing them using R by removing errors occurred in code and obtaining the output, we successfully gave our presentation in class within the time limit.

**Introduction**

In simple terms, image segmentation is the process of partitioning or separating an image into various segments so that it is easier to analyze.

More precisely, image segmentation is the process of assigning a label to every pixel in an image such that pixels with the same label share certain characteristics.

Segmentation has two objectives.

The first objective is to decompose the image into parts for further analysis. In simple cases, the environment might be well enough controlled so that the segmentation process reliably extracts only the parts that need to be analyzed further.

The second objective of segmentation is to perform a change of representation. The pixels of the image must be organized into higher-level units that are either more meaningful or more efficient for further analysis (or both).

The Main Abstract of the project is to analyze image data described by high-level numeric-valued attributes, 7 classes namely Brick face, Cement, Foliage, Grass, Path, Sky, Window.

We have taken first column as Class Attribute and gave a name of **Type.**

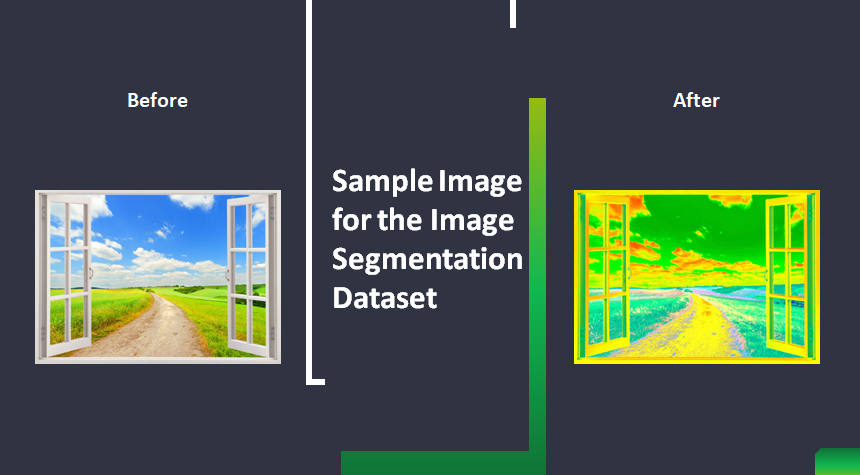
**Attributes**

The Following are the Attributes in our project. One of them is Nominal and remaining are the Numericals.

1. region-centroid-col  
2. region-centroid-row  
3. region-pixel-count  
4. short-line-density-5   
5. short-line-density-2   
6. vedge-mean   
7. vegde-sd  
8. hedge-mean   
9. hedge-sd  
10. intensity-mean

11. rawred-mean    
12. rawblue-mean   
13. rawgreen-mean   
14. exred-mean   
15. exblue-mean   
16. exgreen-mean   
17. value-mean   
18. saturation-mean  
19. hue-mean

By implementing the Image Segmentation dataset on an Image, it looks like below picture because in our dataset we are having three colors: Red, Blue and Green.



**Results**

When we consider random observations comprising of 2100 rows and 19 attributes. To import the data, we use the following code

library(e1071)

setwd("C:/Users/Maruthi/Desktop/UCMDocuments/Spring2019/Datamining/Project")

#Read data from test file

data=read.csv("segmentation test.csv", row.names=NULL)

#Set the first row name to Type

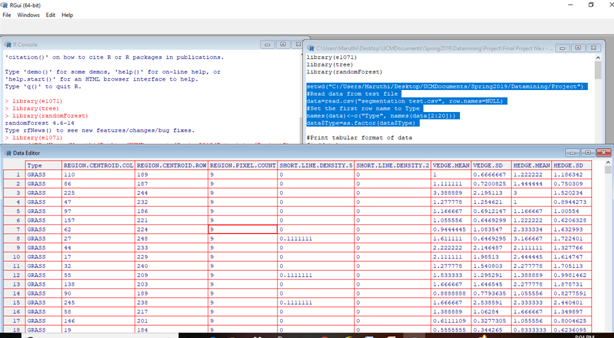
names(data)<-c("Type", names(data[2:20]))

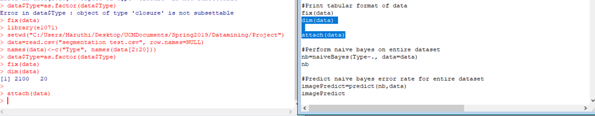
data$Type=as.factor(data$Type)

fix(data)

dim(data)

attach(data)



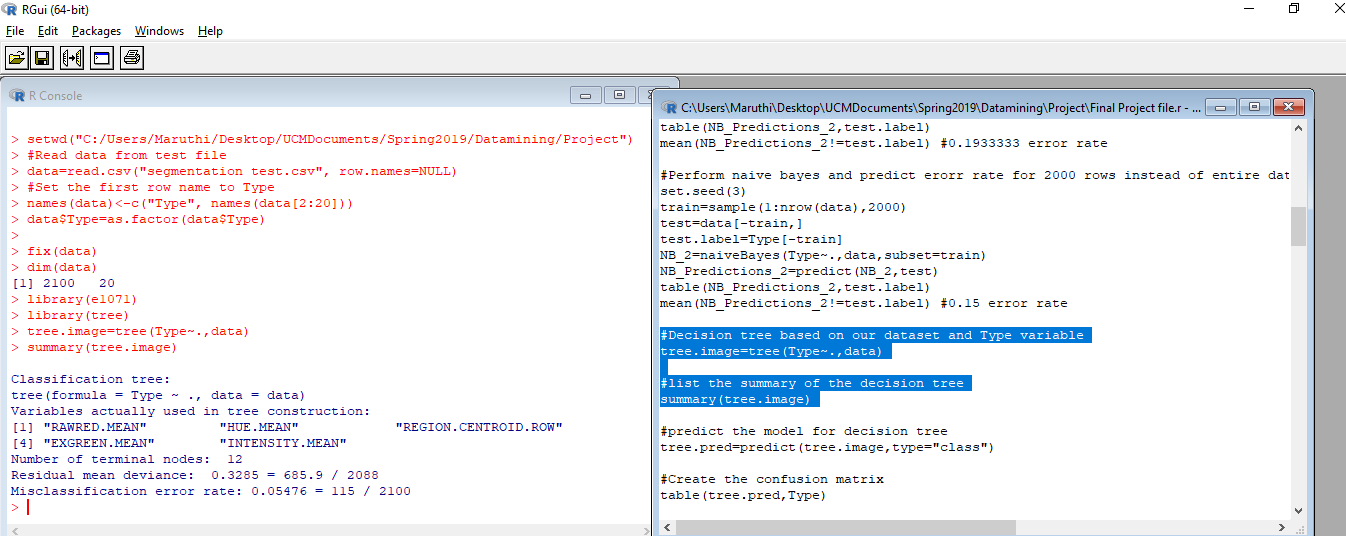


**Decision Tree**

It is one of the Base classifier and a simple representation for classifying examples.

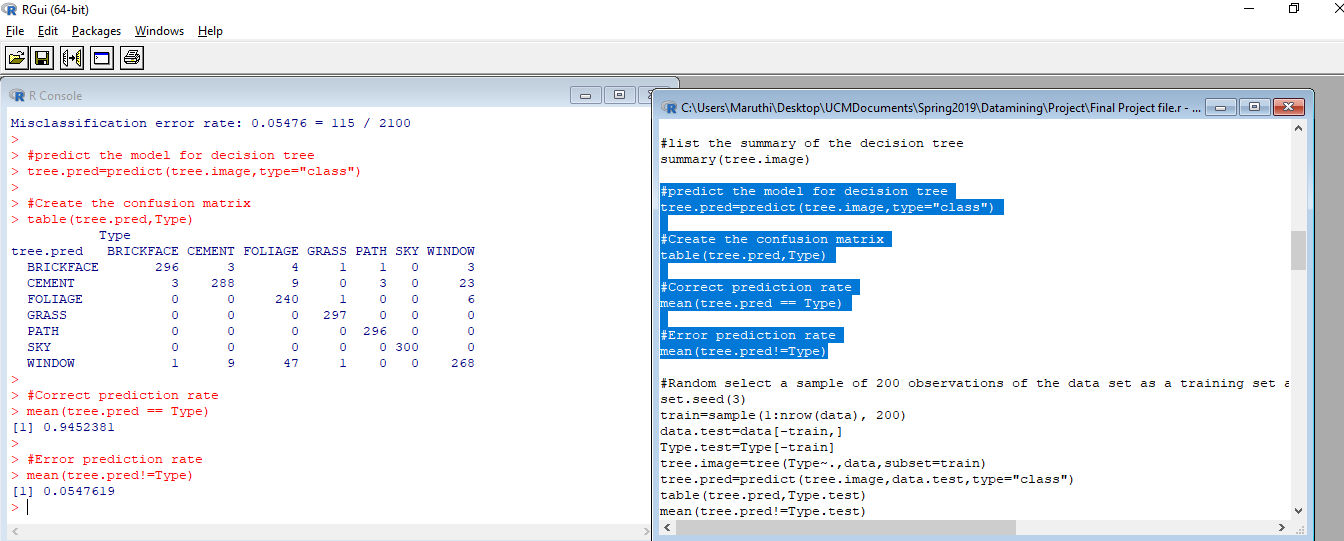
A decision tree is a structure that includes a root node, branches, and leaf nodes. Each internal node denotes a test on an attribute, each branch denotes the outcome of a test, and each leaf node holds a class label. The topmost node in the tree is the root node.

The library(tree) is the tree library for making decision tree classification.



**Confusion Matrix**

A **confusion matrix** is a table that is often used to describe the performance of a classification model (or "classifier") on a set of test data for which the true values are known. The **confusion matrix** itself is relatively simple to understand, but the related terminology can be confusing.

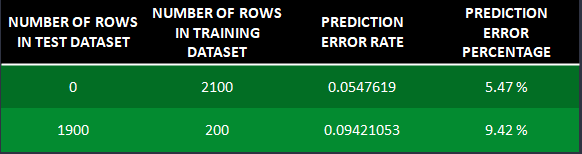


**Hold-out Method**

In the Hold-out method, the data is randomly partitioned into two independent sets: Training set and Test set.

Holdout sample is a sample of data that not used in fitting a model but used to assess the performance of that model.

**Results for Hold-out**

****

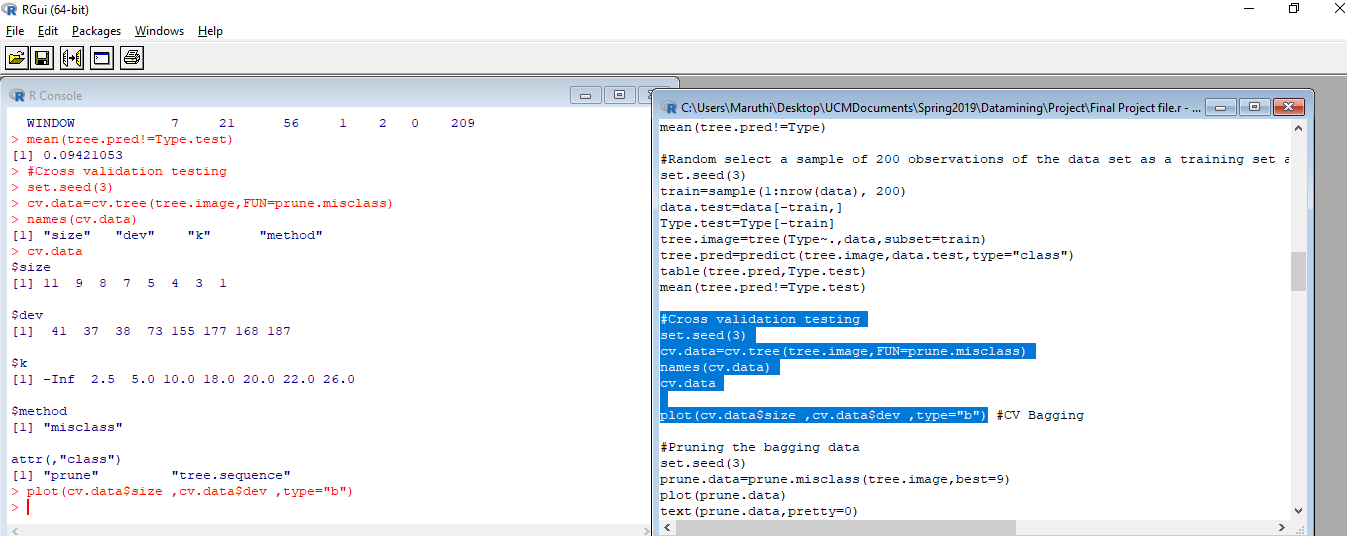
Out of the two cases, we will choose the one which is having low prediction error rate 5.47% and high Accuracy rate 94.53%.

**Cross-Validation**

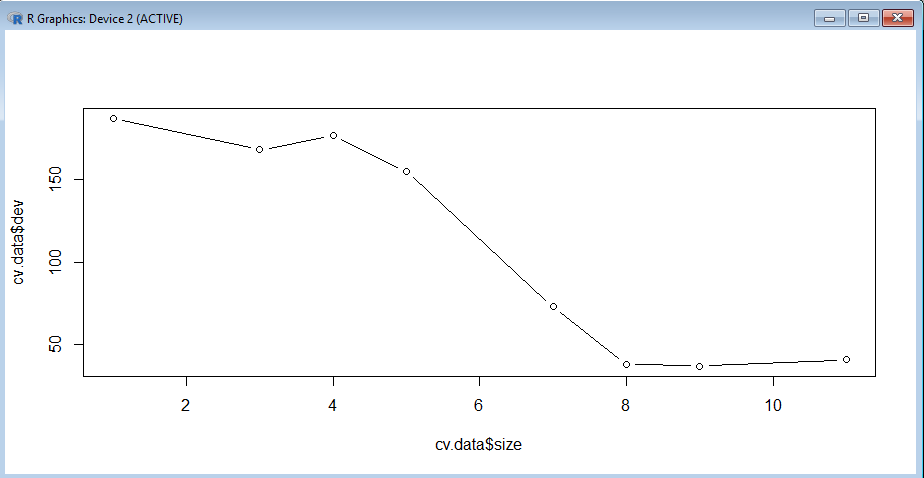
Randomly partition the data into k mutually exclusive subsets, each approximately equal size.

Cross-validation is a technique that is used for the assessment of how the results of statistical analysis generalize to an independent data set.

Cross-validation is largely used in settings where the target is prediction and it is necessary to estimate the accuracy of the performance of a predictive model. The prime reason for the use of cross-validation rather than conventional validation is that there is not enough data available for partitioning them into separate training and test sets. This results in a loss of testing and modeling capability.



plot(cv.data$size ,cv.data$dev ,type="b")



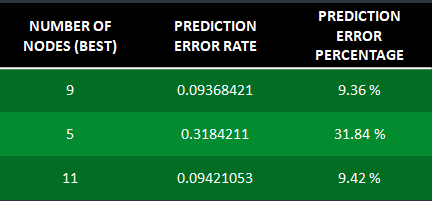
**Bagging**

Bagging is one of the ensemble classifier and also called as Bootstrap Aggregating which is designed to improve the stability and accuracy of algorithms used in statistical classification.

Bagging is simply a special case of Random Forest.

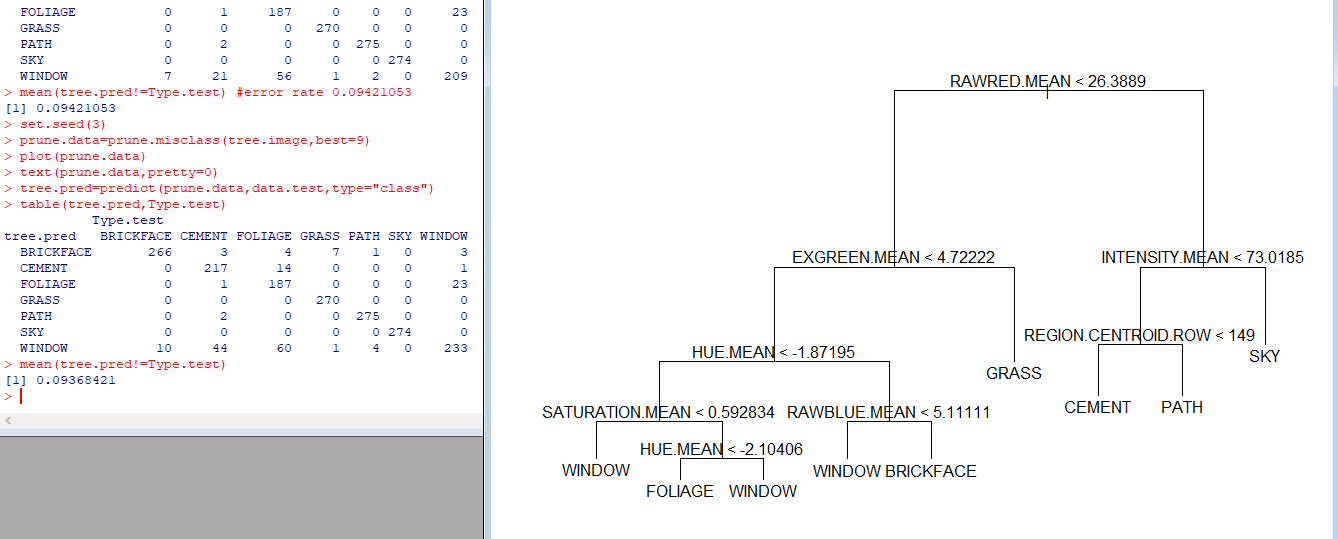
In this bagging we will prune the data which means it will reduce the size of decision tree by removing parts of the tree that do not provide power to classify instance.

So, we will find the error rate by taking the number of the nodes for pruning dataset. Below are the three cases we have taken by three different number of nodes and found the best case among them.



The best case by taking 9.36% low error rate and with accuracy rate of 90.64% is when number of nodes is 9.

The Output when pruning dataset with number of nodes is 9

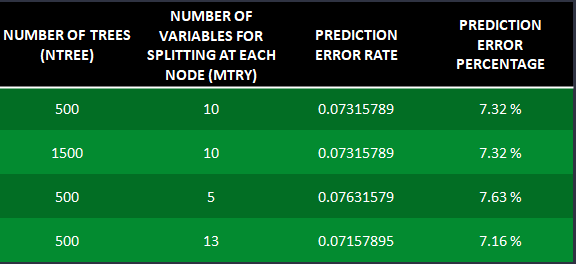


**Random Forest Classification**

Random forest is a trademark term for an ensemble classifier that consists of many decision trees. Each decision tree is constructed by using a random subset of the training data. Random forests are collections of trees, all slightly different.

Random forest() – both bagging and Random forest

Here in Random forest we use mainly two things like **mtry** (number of variables randomly sampled as candidates at each spilt) and **ntree**(number of trees to grow). We have taken four cases by assigning different mtry and ntree values for each case.



So the best case out of four is the fourth case with low predicted error rate of 7.16% and Accuracy rate of 92.84%.

**Naïve Bayes Classifier**

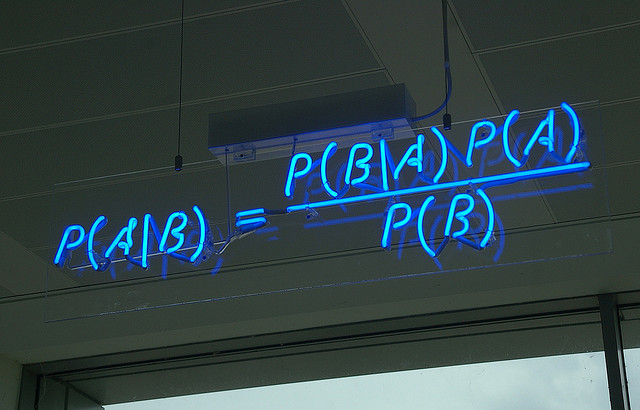
Naïve Bayes is a classification technique based on Bayes theorem with an assumption of independence among predictors. In simple terms, a **Naive Bayes classifier** assumes that the presence of a particular feature in a class is unrelated to the presence of any other feature.

This model is easy to build and particularly useful for very large datasets.

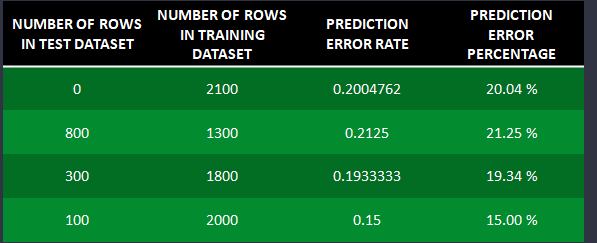
Now, before moving to the formula for Naive Bayes, it is important to know about Bayes’ theorem.

**Bayes’ Theorem**

Bayes’ Theorem finds the probability of an event occurring given the probability of another event that has already occurred. Bayes’ theorem is stated mathematically as the following equation:



For evaluating data mining models, we separate data into training and testing sets.  
Most of the data is assigned to training data set which is collection of records in given dataset and small portion of data is for testing purpose. According to it, we have taken four cases and calculated Error rate and Accuracy of Naïve Bayes Classification.



The best case here is fourth one which is low error rate of 15% and good accuracy of 85% compared to the remaining three in the above cases.

**Support Vector Machine Classification**

**Support Vector Machines** (SVMs) are supervised learning methods used for classification and regression tasks that originated from statistical learning theory. As a classification method, **SVM** is a global classification model that generates non-overlapping partitions and usually employs all attributes.

A Support Vector Machine (SVM) performs classification by finding the hyperplane that maximizes the margin between the two classes. The vectors (cases) that define the hyperplane are the support vectors.

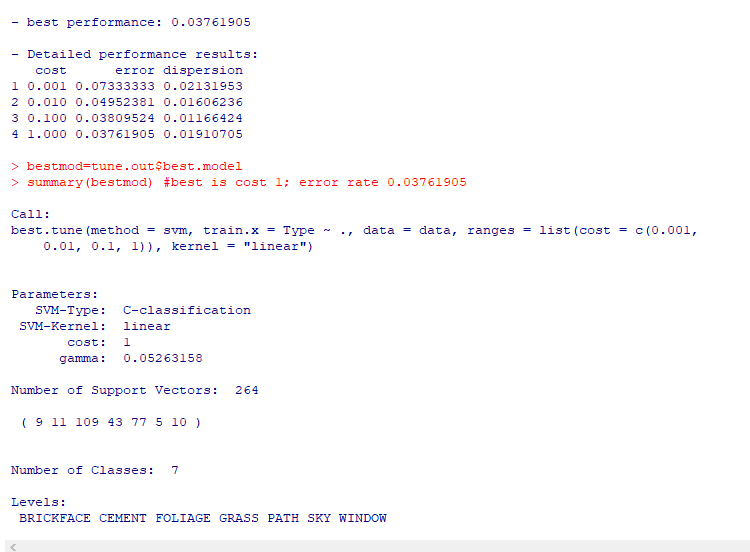
SVM or Support Vector Machine is a linear model for classification and regression problems. It can solve linear and non-linear problems and work well for many practical problems. The idea of SVM is simple: The algorithm creates a line or a hyperplane which separates the data into classes.

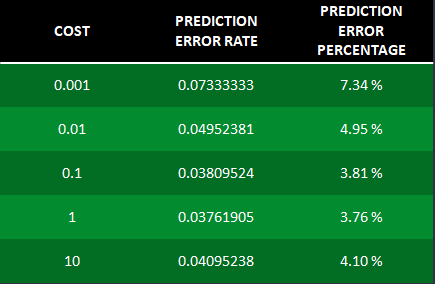
 It uses a technique called the kernel trick to transform your data and then based on these transformations it finds an optimal boundary between the possible outputs. Simply put, it does some extremely complex data transformations, then figures out how to separate your data based on the labels or outputs you've defined.

We have already discussed about it. Here, we have various options available with kernel like, “linear”, “Radial”,” Polynomial”.  Here “Radial” and “Polynomial” are useful for non-linear hyper-plane.

**Results of SVM using Linear Kernel with different costs**

Now we can see how the linear kernel works with different costs to achieve to accuracy and low error rate.

****

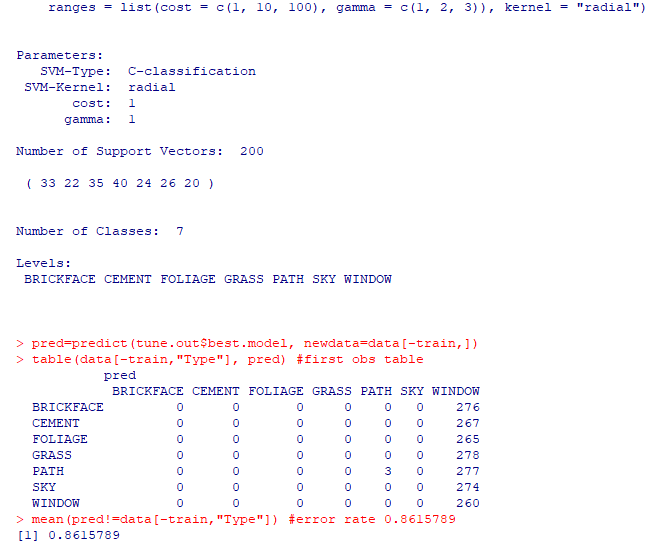
****

We have taken five different costs for linear kernel to check which one gives low error rate and high accuracy to our project. Out of these, we chose the cost value having the lowest error rate of 3.76% and good accuracy rate of 96.24%.

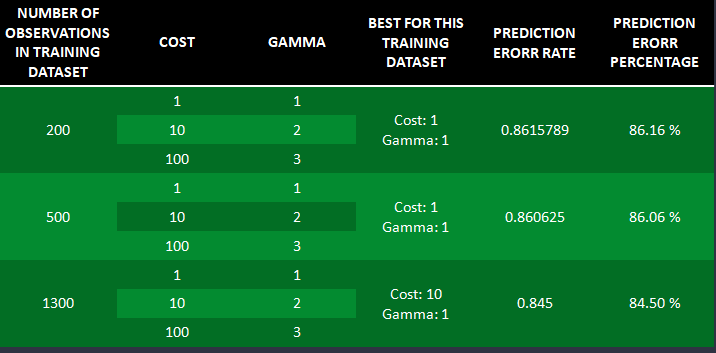
**Results of SVM using Radial Kernel with different Gammas**

In the Support Vector Machine of non-linear radial kernel, gamma is the parameter. If gamma is large, then variance is small implying the support vector does not have wide-spread influence. Technically speaking, large gamma leads to high bias and low variance models, and vice-versa.

In support vector machine of non-linear radial kernel, if we take 200 observations the error rate very high with low accuracy which does not fit for our project.



When we take three different set of observations with different costs and gamma values, the results are as follows:

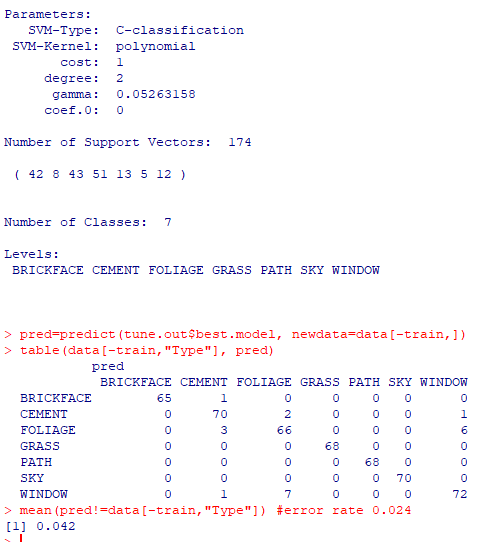


Among all of them the set of 1300 observations is giving two percent less error rate of 84.50%, as compared to other set of observations which is also not an accurate one to consider for a project.

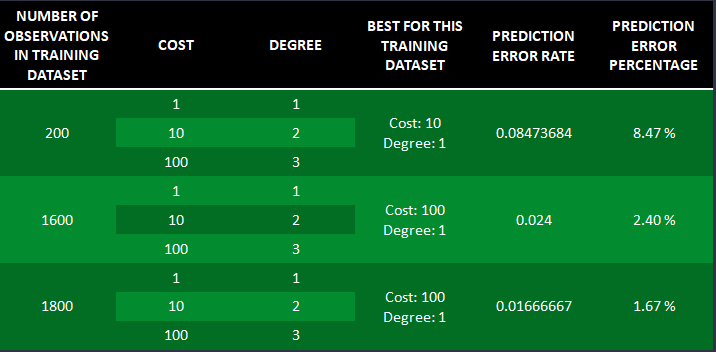
**Results of SVM using Polynomial Kernel with different Degree**

In this non-linear polynomial kernel of support vector machine, Degree is the parameter. Based on that we can decide the error rate and can justify whether it will fit for a project or not.

When we take 1600 observations, we can check how the error and accuracy rates are coming whenever cost and degree are given.



When we want to see for different iterations, the following gives the brief description of two three set of observations for taking different cost values for degree.

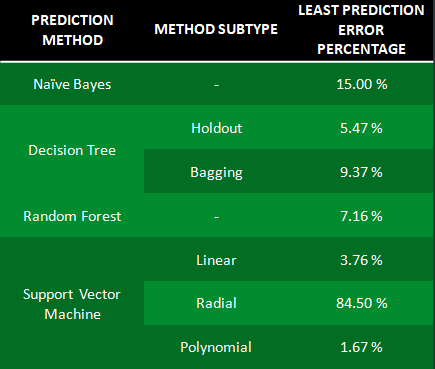


Among them all the very lowest error rate of 1.67% is when we take 1800 observations we can get the good accuracy rate of almost 98% which is the best fit for the project. Finally we can conclude that when there is low error rate there will be high accuracy which can be recommended for any project.

**Comparison of Multiple Classification Techniques**

The classification techniques we used in our project are Naïve Bayes, Decision Tree, Random forest, Bagging and Support Vector Machines for both Linear and Non-Linear Kernel. To compare them in our project the one with highest error rate is in Support Vector Machine of non-linear Radial kernel 84.50% and with the lowest error rate is also in Support Vector Machine of non-linear Polynomial kernel 1.67%.

We can have a clear picture of comparison of all classification techniques is as below:



**Potential performance issues and possible future study**

**Conclusion of the project**

Although all methods have their pros and cons, there has to be a best way to represent data. There can be several factors affecting the results like the dataset, attributes, etc. but our project which focused on Image Segmentation was best predicted using the Support Vector Machine (SVM) Polynomial Method. It had the least prediction error rate when compared to others.

**Performance Issues**

There could have been many factors which led to the results obtained. We might have had better results if those factors were not in place. For instance, the dataset size could be a factor. If we had a larger dataset, the results would have been more accurate and perhaps less faulty. Another factor could have been the size of training and test dataset that we took. We tried many different values, but we cannot try all 2100. At some point, there might have been a least error rate for the image segmentation dataset which we might have missed. The random data selection out of the dataset may also affect the results.

**Possible future study**

Image segmentation is a vital part of future technology. Self-driving cars, smart phone cameras, and many more technologies use this technique. Mining data from this image segmentation sector can be a huge thing in the future. We can try to gather as much data as we can and analyze it to help technology advance and in turn help our community grow.

**References:**

[**https://archive.ics.uci.edu/ml/datasets/Image+Segmentation**](https://archive.ics.uci.edu/ml/datasets/Image+Segmentation)

[**https://towardsdatascience.com/paper-summary-recent-progress-in-semantic-image-segmentation-d7b93ee1b705**](https://towardsdatascience.com/paper-summary-recent-progress-in-semantic-image-segmentation-d7b93ee1b705)