**Assignment 2 Report**

Team: Sruti Paku(sxp131731)

Maringanty Krishna Kavya(kxm152630)

1. **Summary of Dataset:**

The Iris dataset is chosen to perform the classification task. The dataset is multivariate.

-Number of Instances and attributes: Instances:150, Attributes: 4

-Number of classes in the predicted variable:3

-There are no missing or null values in the dataset.

-There is no preprocessing needed.

1. **Summary of the Research Paper:**

“THE USE OF MULTIPLE MEASUREMENTS IN TAXONOMIC PROBLEMS”

* **How was the Data Obtained:** The measurements of fifty Plants have been taken from same colony and measured by Dr. E. Anderson.
* **What did the Author use the Dataset for:** to find in what manner the specific difference may be tested for significance, so as to allow for a variate can be chosen so as to maximize the distinctiveness in species.
* **What type of experiments were done on the dataset:** A sequential Arithmetic procedure was performed on the dataset which involved finding the Observed means for the species (versicolor, Setosa ), sum of squares and products of measurements of attributes and finding the variation of the coefficients.
* **Summary of the author’s results:** Given, multiple populations having several characters measured to a certain measurement, populations are best discriminated.

1. **Pre-process the dataset:**

For Pre-processing of data we have checked for any duplicates in the data, performed scaling of data, check for N/A values.

Our chosen dataset does not have any NA values and does not contain duplicates.

**Code for Preprocessing:**

# load libraries

library(caret)

# load the dataset

data(iris)

# summarize data

summary(iris[,1:5])

print(NAValues<- is.na(iris))

print(duplicated(iris))

# calculate the pre-process parameters from the dataset

preprocessParams <- preProcess(iris[,1:5], method=c("scale"))

# summarize transform parameters

print(preprocessParams)

# transform the dataset using the parameters

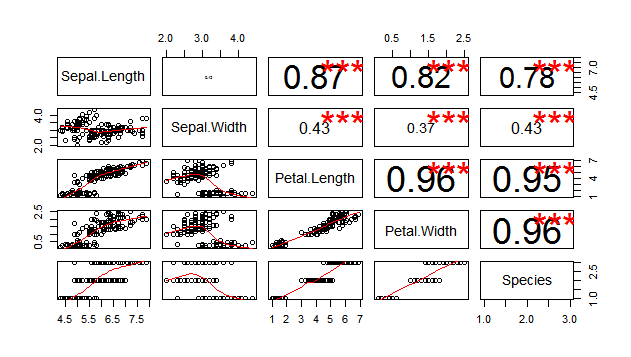
transformed <- predict(preprocessParams, iris[,1:4])

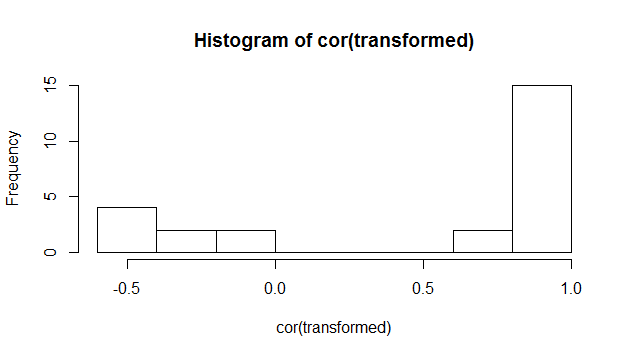
# summarize the transformed dataset

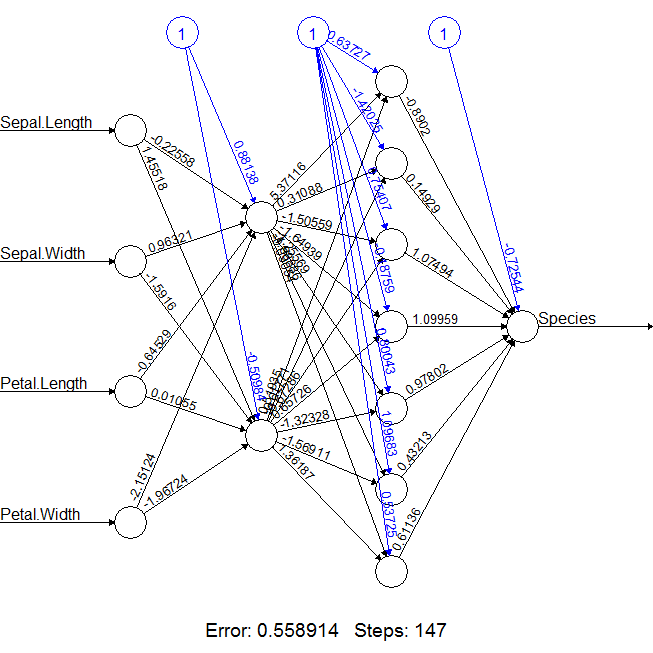
summary(transformed)

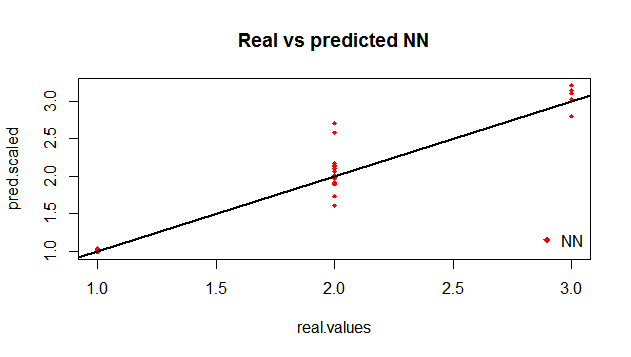
transformed is the preprocessed data set i.e scaled dataset.

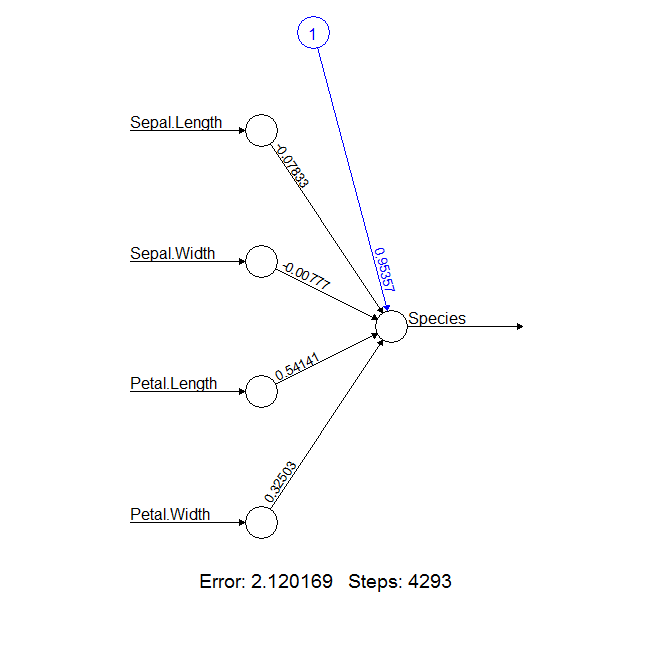
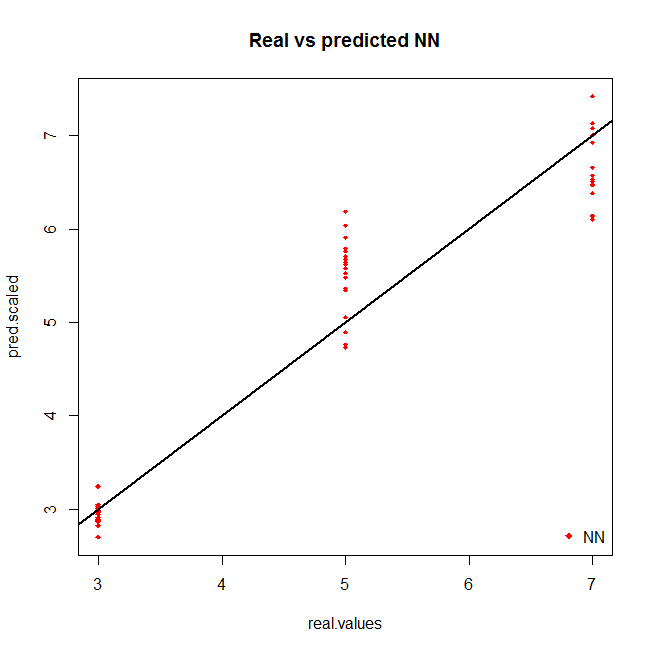
Below are the plot for correlation between various variables and output:

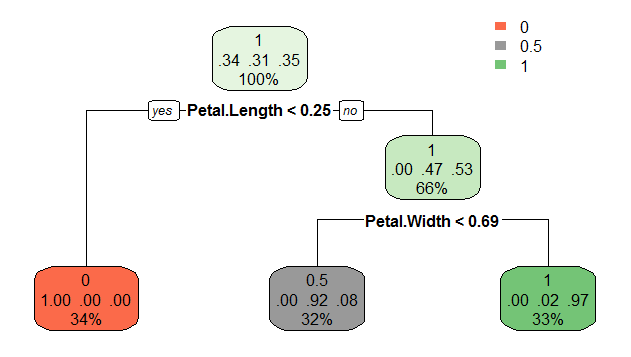




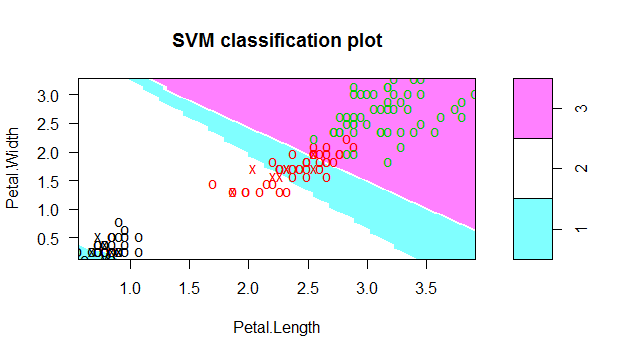
**Neural Net:** ****

****

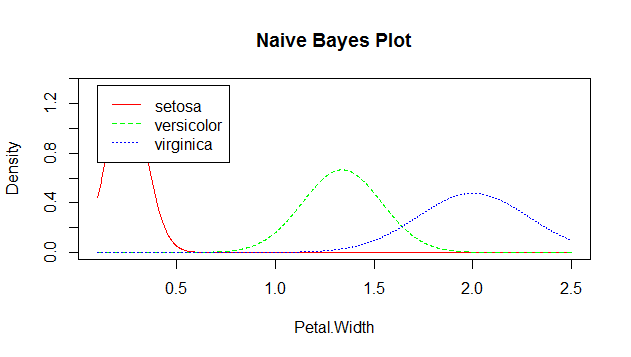
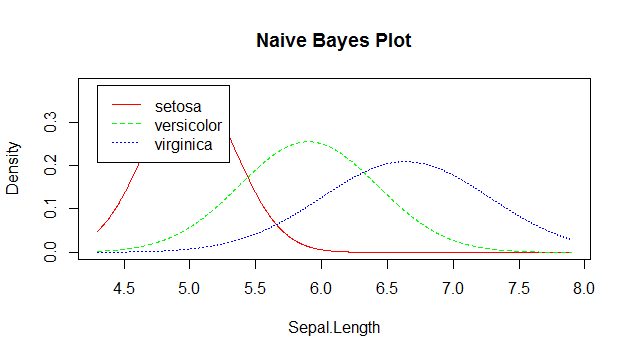
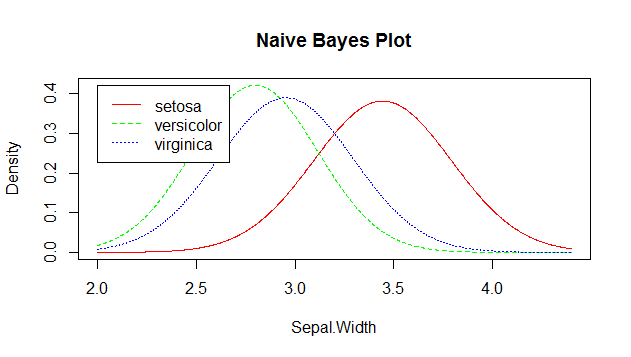
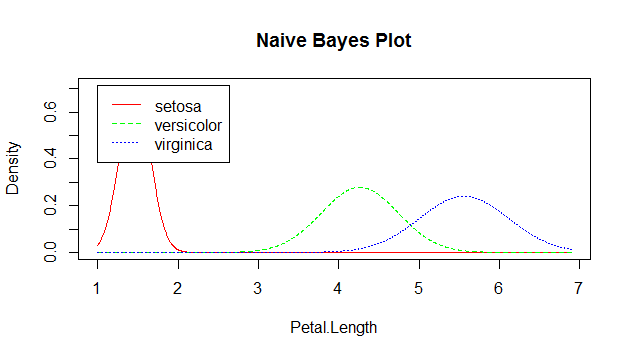
**Perceptron** ****

**Decision Tree** 

**SVM:**

****

**Naïve Bayes:**

****

1. **Implement classifiers on the dataset:**

B. Classifier parameters

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Experiment** | **# Classifier** | **Train/Test Ratio …** | **Parameter1** | **Parameter2** | **Parameter 3** | **Error/Accuracy** |
| 1 | Neural net | 80/20 | Hidden=1 | - | - | 0.04291217334 |
| 2 | Neural net | 80/20 | Hidden=c(5,3) | Threshold =0.01 | Rep=1 | 0.04278103565 |
| 3 | Neural net | 80/20 | Hidden=c(7,2) | Threshold=0.1 | Rep=2 | 0.04622902595 |
| **4** | **Neural net** | **80/20** | **Hidden=c(2,7)** | **Threshold=0.05** | **Rep=5** | **0.04131280543** |
| 5 | Perceptron | 80/20 | Hidden=0 | Threshold=0.01 | - | 0.04644461988 |
| 6 | Perceptron | 80/20 | Hidden=0 | Threshold=0.001 | - | 0.04609874476 |
| **7** | **Perceptron** | **80/20** | **Hidden=0** | **Threshold=0.0001** | **-** | **0.04607376703** |
| 8 | Perceptron | 80/20 | Hidden=0 | Threshold=0.5 | - | 0.05970884446 |
| **9** | **Decision Tree** | **80/20** | **Split on Information gain** | **Without pruning** | **-** | **Root node Error: 0.63333333** |
| 10 | Decision Tree | 80/20 | Split on Information Gain | With Pruning | - | Root node Error: 0.63333333 |
| **11** | **SVM** | **80/20** | **Cost=4** | **Gamma=0.5** | **Kernel=’Linear’** | **Error= 0.0115510340** |
| 12 | SVM | 80/20 | Cost=8 | Gamma=1.0 | Kernel=’Linear’ | Error: 0.01536860130 |
| 13 | SVM | 80/20 | Cost=16 | Gamma=2.0 | Kernel=’Linear’ | Error: 0.02166086466 |
| **14** | **NaiveBayes** | **80/20** | **-** | **-** | **-** | **Accuracy : 0.9666667** |

1. **Results:**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Sampling/Accuracy** | **Train/Test** | **Neural** | **Perceptron** | **Decision** | **SVM** | **Naive** |
| 1 | 90/10 | 99.94% | 99.94% | 86% | 100% | 93% |
| 2 | 80/20 | 99.95 | 99.94% | 80% | 93% | 93% |
| 3 | 70/30 | 99.98% | 99.95% | 100% | 100% | 100% |
| 4 | 60/40 | 99.95% | 99.94% | 86% | 100% | 93% |
| 5 | 50/50 | 99.93% | 99.93% | 73% | 100% | 86% |
|  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |

1. **Analysis:**

Which method performs best and why do you think it performs the best?

According to our analysis SVM method works the best because it gave 100% accuracy in most of the scenarios.

Which method is worst and why?

Decision Tree method is the worst performer as its accuracy levels fluctuate a lot when the train and test data split percentage is changed. The decision tree accuracy varies with a large amount.

What would you change to obtain better accuracy?

To obtain better accuracy we can perform more test to get the best parameter values for the classifiers and also increase the input data.