Jacob Aguilard

Big Data Analytics Capstone

Assignment 3

1.1)

**Dataset:** Mushrooms: <http://archive.ics.uci.edu/ml/datasets/mushroom>

23 variables over 8,000 observations

**Question:** “I would like to classify whether mushrooms are poisonous or not using multiple methods”

**Methods Used:** *Decision Tree* – from the “Machine Learning with Tree-Based Models in Python” DataCamp course

Classification

All Code in Other Files in Folder

Results:

**Decision Tree with 70% / 30% training / testing split**

Cross Validation 10

Training results:

Accuracy Score: 1.0000

Classification Report:

precision recall f1-score support

0 1.00 1.00 1.00 2097

1 1.00 1.00 1.00 1965

avg / total 1.00 1.00 1.00 4062

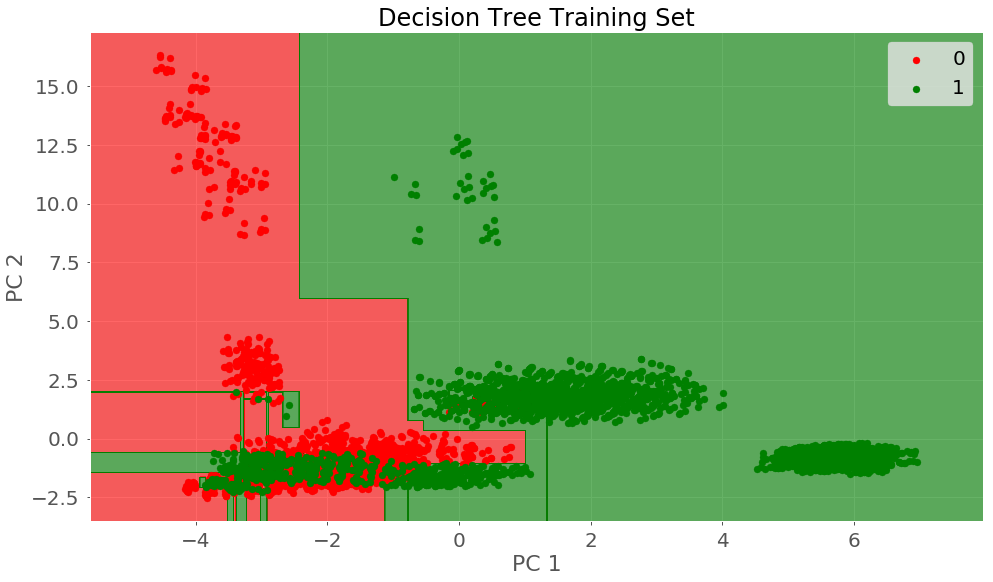
Confusion Matrix:

[[2097 0]

[ 0 1965]]

Average Accuracy: 1.0000

Standard Deviation: 0.0000



Test results:

Accuracy Score: 0.8970

Classification Report:

precision recall f1-score support

0 0.90 0.91 0.90 1257

1 0.90 0.89 0.89 1181

avg / total 0.90 0.90 0.90 2438

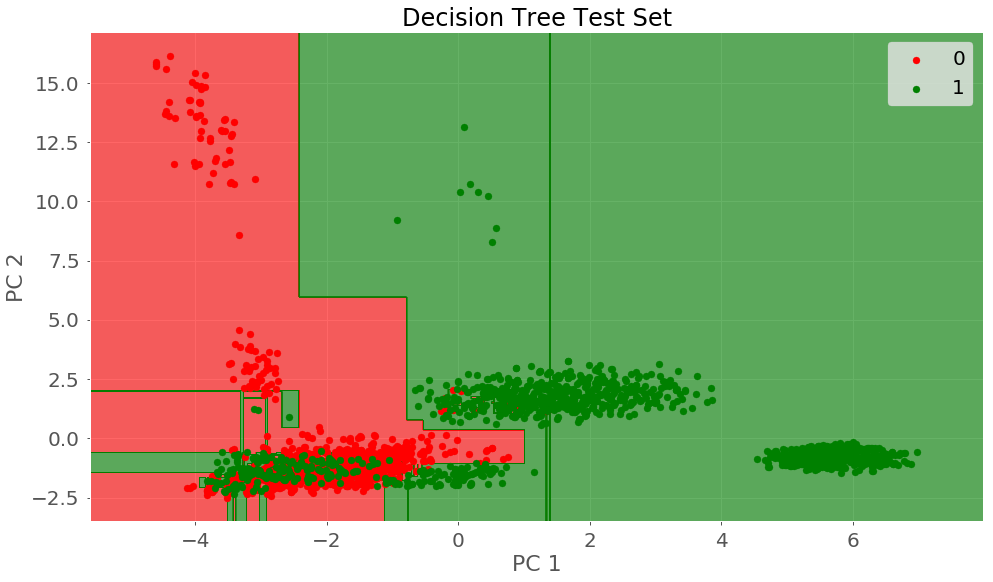
Confusion Matrix:

[[1139 118]

[ 133 1048]]

Average Accuracy: 0.8925

Standard Deviation: 0.0128



**Results with a 50/50 split for training and testing**

Training results:

Accuracy Score: 1.0000

Classification Report:

precision recall f1-score support

0 1.00 1.00 1.00 2097

1 1.00 1.00 1.00 1965

avg / total 1.00 1.00 1.00 4062

Confusion Matrix:

[[2097 0]

[ 0 1965]]

Test results:

Accuracy Score: 0.9990

Classification Report:

precision recall f1-score support

0 1.00 1.00 1.00 2111

1 1.00 1.00 1.00 1951

avg / total 1.00 1.00 1.00 4062

Confusion Matrix:

[[2111 0]

[ 4 1947]]

Average Accuracy: 1.0000

Standard Deviation: 0.0000

**Summary of Performance:** The 70/30 split of data had ~90% accuracy for classifying the mushrooms as poisonous with little difference in the cases where cross validation was 20 or 30 which likely means that 10 was enough or more than enough for this data and method. What was surprising was the performance for the 50/50 split of data. Having a smaller test class, I assumed it would perform worse and not nearly perfect. I wonder if this has anything to do with the data rather than the method. I was surprised through most of my efforts that splits with grater percentages of data as test class performed well or better. It’s possible the data is constructed in a way that overfitting comes easy to it.

1.2)

**Dataset:** Mushrooms: <http://archive.ics.uci.edu/ml/datasets/mushroom>

**Question:** “I would like to classify whether mushrooms are poisonous or not”

**Methods Used:** *K Nearest Neighbors*  –from the “Clustering Methods with SciPy” DataCamp course

Clustering Results and their performance measures –

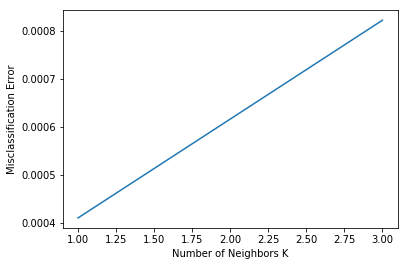
The first test was on **70/30 training / testing split with 10 cross validations**

K = 4; accuracy = ~0.998

K = 3; accuracy = ~0.999

K = 2; accuracy = ~0.999

K = 1; accuracy = ~1.0



Optimal k = 1

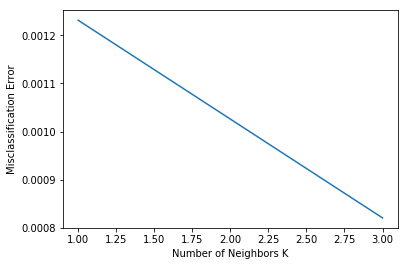
**70/30 training / testing split with 2 cross validations** (Note I originally did cv = 20 but the data mirrored the cv = 10 data and was uninteresting)

K = 4; accuracy = ~1.0

K = 3; accuracy = ~0.999

K = 2; accuracy = ~0.999

K = 1; accuracy = ~0.99



Optimal K = 3

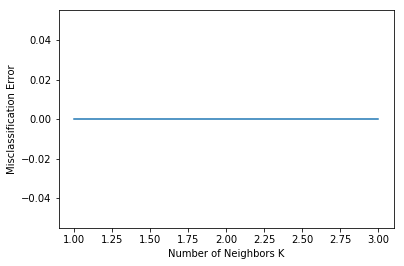
**50/50 training / testing split with 10 cross validations**

K = 4; accuracy = 1.0

K = 3; accuracy = 1.0

K = 2; accuracy = 1.0

K = 1; accuracy = 1.0



**50/50 training / testing split with 2 cross validations**

Same as the immediate above

**Summary of Performance:** Again here, we see that the 50/50 split is ‘perfect’ and that is suspicious. I know the data has few unique values per variable (up to about 10). It may be that the data is simply better for more even splits as there are certain characteristics common to poisonous mushrooms (see my doc with info-graphics). Using less cross validations, we saw that the accuracy of the model went ep with the value of k. 70/30 split with cv = 2 was the only model that had the most optimal performance above k = 1

1.3)

**Dataset:** Mushrooms: <http://archive.ics.uci.edu/ml/datasets/mushroom>

**Question:** “I would like to classify whether mushrooms are poisonous or not”

**Methods Used:** *Linear Regression (Logistic)* – should be from the “Machine Learning Toolbox” and “Unsupervised Learning in Python” DataCamp courses

NOTE: All of the data was converted to numerical type scaled outlined in the code provided

**Results:**

**On an 80/20 data split with Cross Validation = 2**

0 1

0 [[824, 21],

1 [ 23, 757]]

Accuracy: 0.9579

**On an 80/20 data split with Cross Validation = 10**

0 1

0 [[814, 31],

1 [ 37, 743]]

Accuracy: 0.9728

**On an 60/40 data split with Cross Validation = 2**

0 1

0 [1626, 65],

1 [ 61, 1498]

Accuracy: 0.9612

**On an 60/40 data split with Cross Validation = 10**

0 1

0 [1627, 64],

1 [ 60, 1499]

Accuracy: 0.9618

**Summary of the Performance:** Across all variations of the linear regression model performance was nearly the same. The accuracy ranged less than 2%. And although this time the greater sized test class was outperformed, it still did exceptionally well.

1.4)

**Dataset:** Mushrooms: <http://archive.ics.uci.edu/ml/datasets/mushroom>

**Question:** “”

**Methods Used:** ***Non Linear***–from the “Machine Learning with Tree-Based Models in Python” DataCamp course

Regression

**Summary of the Performance:**

1.5)

**Dataset:** Mushrooms: <http://archive.ics.uci.edu/ml/datasets/mushroom>

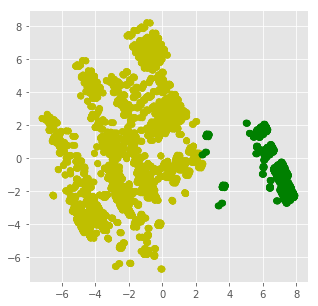
**Question:** “Can clustering be used to identify species of mushrooms”

**Methods Used:** *K Means Algorithm* – should be from the “Machine Learning with Tree-Based Models in Python” DataCamp course

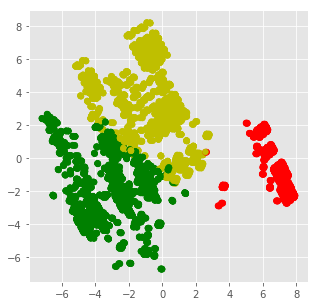
Clustering Results

I used visualizations to represent the clustering results

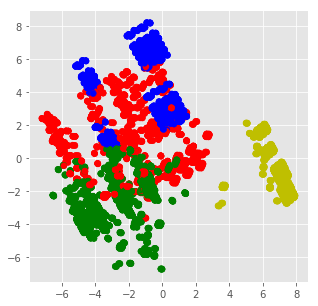
**K = 2 Clusters**



**K = 3 Clusters**

****

**K = 4 Clusters**

****

**Summary of the Performance:** The only surprise I found in the data was the split of the ‘blue’ data points, with more testing (and more clusters) it’s likely I could identify how many species were represented by the data. Species isn’t included in the data. Later on I could write a loop to test different k values like I did for K- Nearest neighbor.

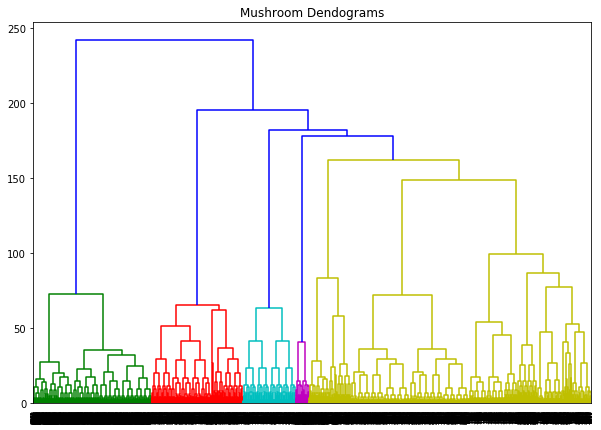
1.6)

**Dataset:** Mushrooms: <http://archive.ics.uci.edu/ml/datasets/mushroom>

**Question:** “Continuing the last problem, can this form of clustering identify species and trends”

**Methods Used:** *Hierarchical Clusters* – should be from the “Unsupervised Learning in Python” DataCamp course

Clustering Results

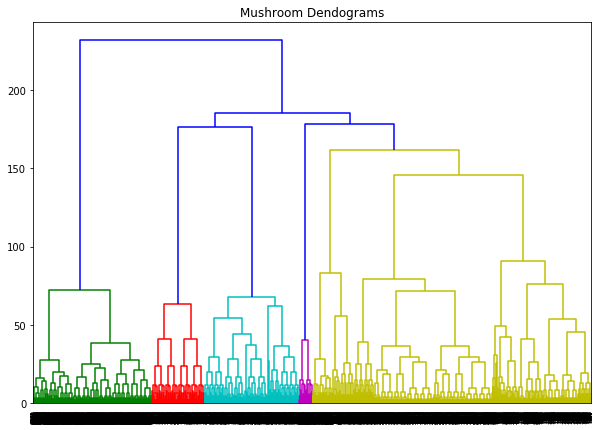


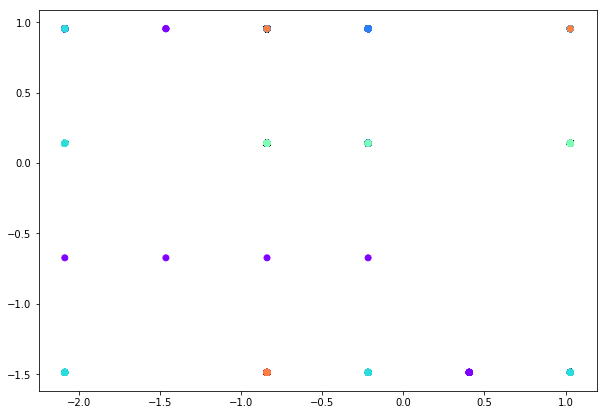
The Clustering Technique is a bottom up approach that split the data into 7 distinct clusters with more specific divides below. The dataset could be shortened so the dendrogram is explicit about the classification underneath



A scatter plot of the classes.

Next is the same methods without the poisonous variable





**Summary of the Performance:**

The scatterplot helps to visualize the variance in the clusters created by the dendrogram and shows that there are about 6 distinct classes. The dendrogram suggests that there are only 8 different groups at most. The model in which I left the poisonous trait showed a clear split in the scatter plot suggesting that the variable had a large impact on the clustering and this can be seen in the dendrogram having two major branches of unrelated clusters. This was also reflected in the dendrogram without that data point.

**Comparison with K-means:** The dendrogram is an easier way to provide insight into how many clusters may be optimal for the dataset versus k-means. The clustering itself creates a different scatterplot that shows well defined clusters though it may need some scaling or jitter added to show variance within the clusters better. The k-means scatterplots were more useful for a representation of haw the clusters are defined/related on the given variables. With fewer clusters however I suspect there were likely more clusters within and as I broke it down into more clusters, the different group and potential other distinct cultures became more obvious. Groups which the dendrogram had identified.

**Problem2 – Four more problems/methods**

**2.1 )Neural Network for Prediction of poisonous mushroom**

**Dataset: Mushroom**

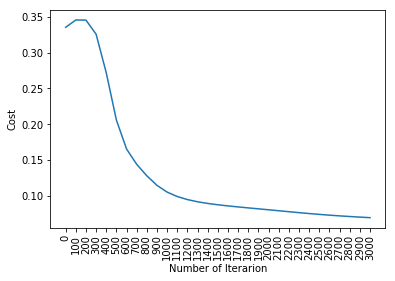
**Question:** “I would like to classify the mushrooms as poisonous or not using a neural network”

**Methods Used:** Neural Network from ‘Deep Learning’ Data Camp Course

**Results**

train accuracy: 96.26096322511155 %

test accuracy: 96.24615384615385 %

****

**Summary of Performance:** The Neural Network does the job incredibly well given it runs through enough iterations. Below one thousand iterations the cost function is a but higher than wanted and it continuously improves. With a task like determining if a mushroom is poisonous a false negative is dangerous so the most accurate model is wanted. Neural Network may be the optimal model.

I used an array to create a list of machine learning models to simplify the process for the next 3 methods and created a single visual and table to compare results.

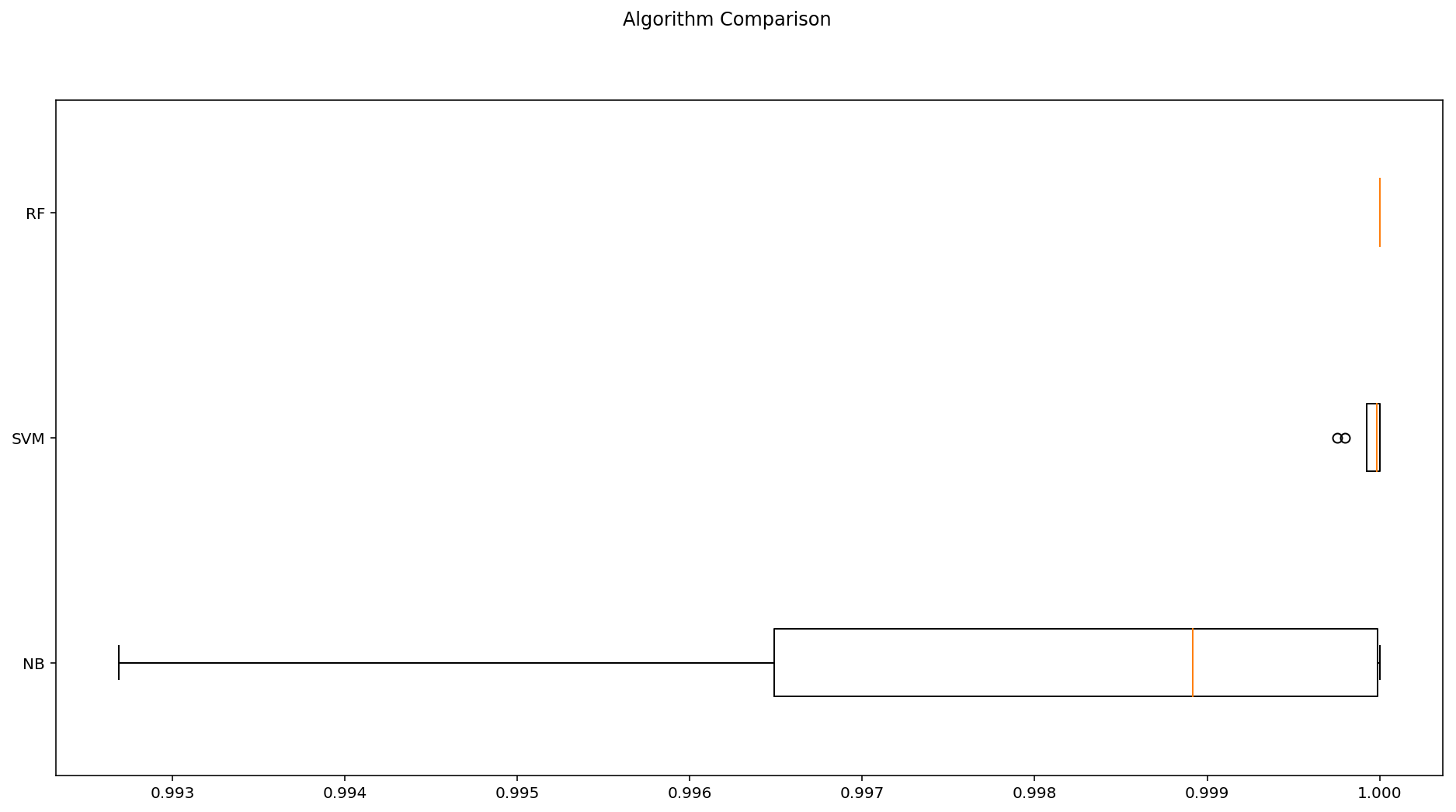
**2.2) Naïve Bayes for Prediction of poisonous mushroom**

**Dataset: Mushroom**

**Question:** “I would like to classify the mushrooms as poisonous or not using a Naïve Bayes”

**Methods Used:** Neural Network from ‘Intro to Data Science in Python’ Data Camp Course

**Results**

****

**Summary of Results:** Naïve Bayes had an average accuracy 99%. It performed well with the data using the default Gaussian Model.

**2.3 )Random Forest for Prediction of poisonous mushroom**

**Dataset: Mushroom**

**Question:** “I would like to classify the mushrooms as poisonous or not using a Random Forest”

**Methods Used:** Random Forest from ‘Deep Learning’ Data Camp Course

**Results**  100% accuracy. This makes sense as the decision tree is a great method for classification of plants and the random forest allows for a perfect model to be created from the results of many trees.

**Summary of Performance:** The Random Forest is possibly the best tool for this task.

**2.4 )Support Vector Machine for Prediction of poisonous mushroom**

**Dataset: Mushroom**

**Question:** “I would like to classify the mushrooms as poisonous or not using a Support Vector Machine”

**Methods Used:** Neural Network from ‘Linear Classifiers in Python’ Data Camp Course

**Results**

Another 100%

The table below is a matrix of correlation of the results of the above methods. It makes sense that the two 100% accurate models would have a 1-1 correlation. The Naïve Bayes method was the odd one out with less than 100% accuracy. The correlation isn’t the same as the accuracy here.

|  | **NB** | **SVM** | **RF** |
| --- | --- | --- | --- |
| **NB** | 1.000000 | 0.978058 | 0.978058 |
| **SVM** | 0.978058 | 1.000000 | 1.000000 |
| **RF** | 0.978058 | 1.000000 | 1.000000 |