Data Science Project 2018

TEAM 16

Analysis and Interpretation of Data

Edgar Meireles, ist424755 (83451)
Francisco Centeno, ist424765 (83461)
Paulo Revés, ist424840 (83537)

CONTENTS

1. INTRODUCTION

For this project we were expect to analyze and understand the information from two different datasets, using techniques and algorithms learned on the course.

The first one is about Colposcopies this dataset explores the subjective quality assessment of digital colposcopies, studied at Hospital Universitario de Caracas. The dataset has three modalities (i.e. Hinselmann, Green and Schiller) each with 69 attributes (62 predictive attributes, 7 target variables). The Green dataset has 98 records, Hinselmann 97 records and Schiller 92 records.

The second dataset is about Air Pressure system (APS) which generates pressurized air that is utilized in heavy trucks. The attributes from this dataset were anonymized for proprietary reasons. The dataset is composed by 60000 records each with 171 attributes. After analyzing the information we found that the target variable was the class, composed of two nominal values “neg” if the failures for the components are not related to the APS and “pos” if they are.

2. PRE-PROCESSING

2.1 Non-Supervised Mining

2.1.1 Problem 1

With the Colposcopies datasets we applied the KMeans algorithm choosing the two predominant attributes based on the decision trees Figures 36, 37 and 38, and then applied the algorithm doing this for each of the datasets.

2.1.2 Problem 2

For this problem we started by analyzing the Decision Tree created on the classification part, based on this we chose the three predominant attributes and verified the relations between them. We did this because the number of attributes was to large, so to analyze efficiently and reach conclusions we opted for this approach, using the KMeans algorithm.

2.2 Classification

2.2.1 Problem 1

With the Colposcopies datasets we started by separating the “consensus” target class from the rest of the information, then split into training and test sets, balancing the training set. We applied this for each of the datasets.

2.2.2 Problem 2

With the APS Failure at Scania Trucks dataset we started by transforming the attribute class from nominal to binary doing this to both the training and test set. We then balanced the training set data increasing the lowest count class to have the same value as the major one, being “pos” the lowest one with 1000 records and “neg” the biggest one with 59000. But we still have missing values in some attributes of the dataset; we fill them with different values based on the attribute to observe which deliver the best results (0, min, max, mean).

3. EXPLORATION

3.1 Non-Supervised Mining

3.1.1 Problem 1

3.1.1.1 Methods and Parametrization

The goal of KMeans Clustering is to find groups in the data, where the number of groups is given by the K variable. The algorithm assigns each data point to one of the K groups based on the distance to the cluster center. To discover the K variable, we used the Elbow Curve Method, assigning different values to K and using the Euclidean distance between each data point with the cluster center. Plotting a graph with the variation of K and the distance, we can notice that the distance decreases as K gets larger. This happens because when you have more clusters, they will be smaller, so the distance between the data points and the cluster center will also be smaller. After doing this we determined that the best number of clusters would be 5, Figure 1.

3.1.1.2 Results

By comparing the Figures 3, 5 and 7 present in the Attachments section, we can observe that there is a bigger difference between the centroids coordinates on Figure 3, which indicates that the attributes *means rgb_cervix_b_mean_minus_std* and *rgb_cervix_g_std* are the ones that best aggregate the data in subsets for later analysis.

3.2 Problem 2

3.2.1.1 Methods and Parametrization

As we did in the problem 1, we started by using the Elbow Curve Method to identify which K would be the most suitable to use in the KMeans method. Using the various values to replace the missing values, as explained on section 2.2.2, we plot the lines in the graph, shown on Figure 2. As we can see, there is no significant differences on the lines, and we can notice that the lines start to stabilize in k=5, so that's the number of clusters we will find.

3.2.1.2 Results

As we can see in the Figures 2 and 4, there are not a lot of conclusions we can take, because there isn't a clear difference between the cluster centers. That lead us to admit that any of the chosen attributes are good to aggregate data in the subsets, and since we have 171 attributes, we wouldn't be able to make all the combinations between two attributes.

3.3 Classification

3.3.1 Problem 1

3.3.1.1 Methods and Parametrization

The algorithms used are KNN, Naïve Bayes, CART (Decision Trees) and Random Forest.

3.3.1.2 Results

Green	KNN = 5	Naïve Bayes	CART	Random Forest
Confusion matrix	[[7 2] [8 13]]	[[5 4] [1 20]]	[[6 3] [0 21]]	[[7 2] [5 16]]
Accuracy	0.6666(6)	0.8333(3)	**0.90000**	0.7666(6)
Error Rate	0.3333(3)	0.1666(6)	**0.10000**	0.2333(3)
Precision	**0.7777(7)**	0.5555(5)	0.6666(6)	**0.7777(7)**
Specificity	0.61905	**0.95238**	1.00000	0.76190
FP rate	0.1333(3)	0.1666(6)	0.12500	**0.1111(1)**
TP rate	**0.35000**	0.20000	0.2222(2)	0.30435

Hinselmann	KNN = 5	Naïve Bayes	CART	Random Forest
Confusion matrix	[[1 4] [10 15]]	[[2 3] [5 20]]	[[2 3] [4 21]]	[[2 3] [2 23]]
Accuracy	0.5333(3)	0.7333(3)	0.7666(6)	**0.8333(3)**
Error Rate	0.4666(6)	0.2666(6)	0.2333(3)	**0.1666(6)**
Precision	0.20000	**0.40000**	**0.40000**	**0.40000**
Specificity	0.60000	0.80000	0.84000	**0.92000**
FP rate	0.21053	0.13043	0.12500	**0.11538**
TP rate	0.06250	**0.0909(09)**	0.08696	0.08000

Schiller	KNN = 5	Naïve Bayes	CART	Random Forest
Confusion matrix	[[5 3] [3 17]]	[[7 1] [1 19]]	[[8 0] [3 17]]	[[7 1] [4 16]]
Accuracy	0.78571	**0.92857**	0.89286	0.82143
Error Rate	0.21429	**0.07143**	0.10714	0.17857
Precision	0.62500	0.87500	**1.00000**	0.87500
Specificity	0.85000	**0.95000**	0.85000	0.80000
FP rate	0.15000	**0.05000**	0.00000	0.05882
TP rate	0.227(27)	0.26923	**0.32000**	0.30435

3.3.2 Problem 2

3.3.2.1 Methods and Parametrization

The algorithms used are KNN, Naïve Bayes, CART (Decision Trees) and Random Forest. We run them with different datasets depending on the values filled for the missing values. Changing some of the parameters for KNN and CART based on the accuracy depending on the number of neighbors and accuracy depending on the depth, figure 8 and x respectively.

3.3.2.2 Results

NaN value(0)	KNN = 5	Naïve Bayes	CART	Random Forest
Confusion matrix	[[15425 200] [92 283]]	[[15136 489] [36 339]]	[[15532 93] [156 219]]	[[15599 26] [148 227]]
Accuracy	0.98175	0.96719	0.98444	**0.98913**
Error Rate	0.01825	0.03281	0.01556	**0.01088**
Precision	0.98720	0.96870	0.99405	**0.99834**
Specificity	0.7546(6)	**0.90400**	0.58400	0.6053(3)
FP rate	0.41408	0.59058	0.29808	**0.10277**
TP rate	0.98198	0.97809	**0.98610**	0.98566

NaN value(min)	KNN = 5	Naïve Bayes	CART	Random Forest
Confusion matrix	[[15425 200] [92 283]]	[[15136 489] [36 339]]	[[15543 82] [156 219]]	[[15593 32] [136 239]]
Accuracy	0.98175	0.96719	0.98513	**0.98950**
Error Rate	0.01825	0.03281	0.01488	**0.01050**

Precision	0.98720	0.96870	0.99475	**0.99800**
Specificity	0.7546(6)	**0.90400**	0.58400	0.6373(3)
FP rate	0.41408	0.59058	0.27243	**0.11808(11808)**
TP rate	0.98198	0.97809	0.98611	**0.98490**

NaN value(max)	KNN = 5	Naïve Bayes	CART	Random Forest
Confusion matrix	[[15251 374] [229 146]]	[[14398 1227] [98 277]]	[[15558 67] [146 229]]	[[15601 24] [141 234]]
Accuracy	0.96231	0.91719	0.98669	**0.98969**
Error Rate	0.03769	0.08281	0.01331	**0.01031**
Precision	0.97606	0.92147	0.99571	**0.99846**
Specificity	0.3893(3)	**0.7386(6)**	0.6106(6)	0.62400
FP rate	0.71923	**0.81582**	0.226351	0.09302
TP rate	**0.99052**	0.98112	0.985494	0.98522

NaN value(mean)	KNN = 5	Naïve Bayes	CART	Random Forest
Confusion matrix	[[15092 533] [88 287]]	[[15172 453] [41 334]]	[[15522 103] [197 178]]	[[15604 21] [192 183]]
Accuracy	0.96119	0.96913	0.98125	**0.98669**
Error Rate	0.03881	0.03088	0.01875	**0.01331**
Precision	0.96589	0.97101	0.99341	**0.99866**
Specificity	0.7653(3)	**0.8906(6)**	0.4746(6)	0.48800
FP rate	0.65000	0.57560	0.36655	**0.10294**
TP rate	0.98134	0.97846	**0.98866**	0.98841

4. CRITICAL ANALYSIS

As we can see in the section 3.3.1.2, given the information present on the Green set, the best algorithm depends on the goal of the user, since CART algorithm gives us better accuracy, but KNN gives us better precision. Relatively to the Hinselmann set, we can clearly see that the Random Forest algorithm is the best. For last, the Naïve Bayes algorithm is the best choice for the Schiller set.

Looking at the section 3.3.2.2, we can compare the different results that we achieved by replacing the missing values with different values, but we can conclude that the Random Forest algorithm is the best choice, independently of the value chosen to replace the missing values. This happens because Random Forest builds multiple decision trees and merges them together in order to have a more accurate and stable prediction.

Below as attachments are the Roc chart graphs for each Algorithm with different missing values, in conformity with the results given for each dataset.

As the report had a limit of 10 pages for contents and for the sake of being able to properly analyze it, we also deliver with the code an extra folder with some bigger attachments, showing the Decision Tree for the CART algorithm. The Trees were made for both problems, with an extra for the second problem with only the five most predominant attributes.

5. CONCLUSIONS

We can conclude that the classification process is best suited for datasets with lots of attributes. In contrast, with the non-supervised process which requires binary associations between attributes, this makes datasets with lots of attributes hard to study because the number of relations would be too large to compute

and analyze, which can be prove by the explanation given in section 3.2.1.2, where we explain why the KMeans didn't work very well in a dataset with 171 attributes.

REFERENCES

[1] (2018, August 5). *Working with missing data*. https://pandas.pydata.org/pandas-docs/stable/missing_data.html

[2] D'Souza, Jocelyn (2018, March 15). *Let's learn about AUC ROC Curve!* https://medium.com/greyatom/lets-learn-about-auc-roc-curve-4a94b4d88152?fbclid=IwAR0ua-JrL5IMb91zO0Ic2H1JH8kiKhynBlQrIRstSL88dHnCNff0oPJDsEk

[3] (2018, February 2). *Algoritmo K-means: Aprenda essa Técnica Essêncial aatravés de Exemplos PAsso a Passocom Python*. http://minerandodados.com.br/index.php/2018/02/02/algoritmo-k-means-python-passo-passo/

[4] (2013, December 6). *How can we choose a "good" K for K-means clustering?* https://www.quora.com/How-can-we-choose-a-good-K-for-K-means-clustering

ATTACHEMENTS

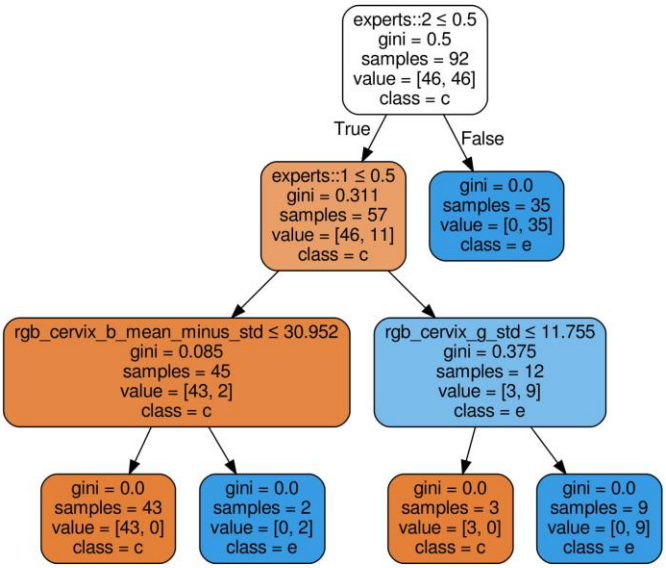

Figure 37 – Decision Tree (Green)

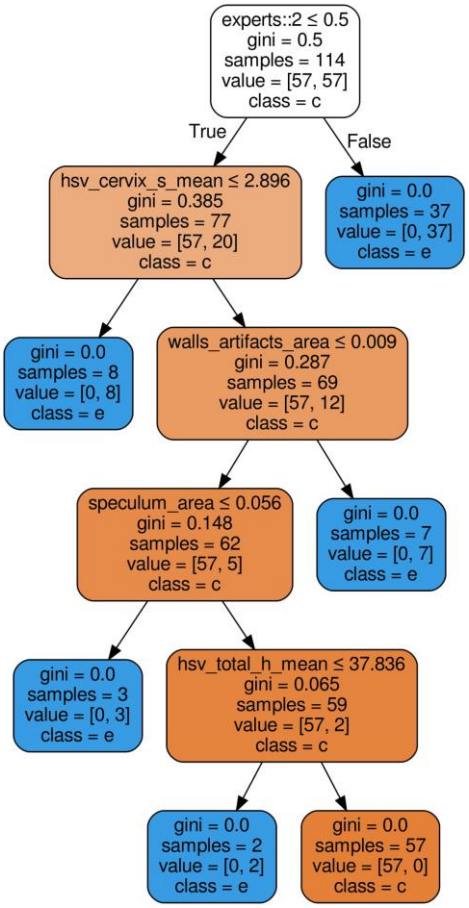

Figure 38 – Decision Tree (Hinselmann)

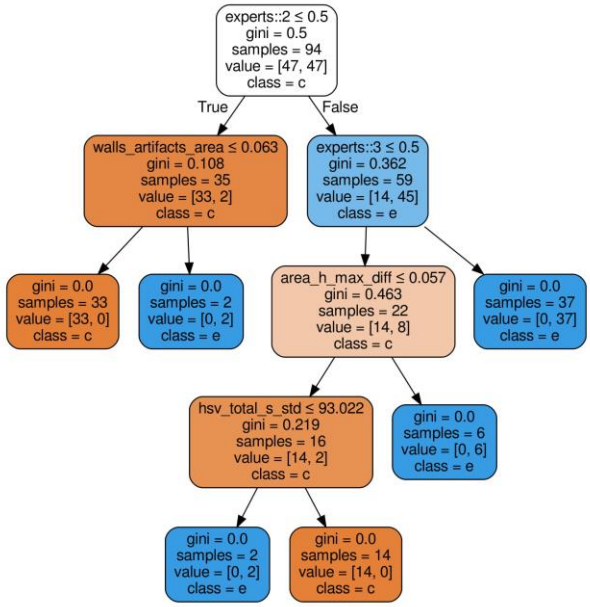

Figure 39 – Decision Tree (Schiller)

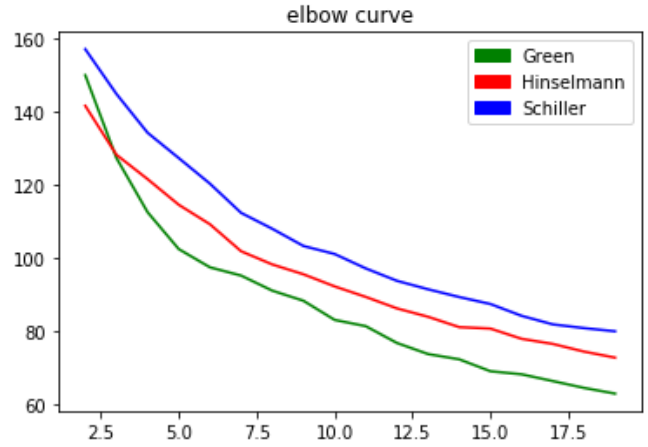

Figure 1 – Elbow Curve First Problem

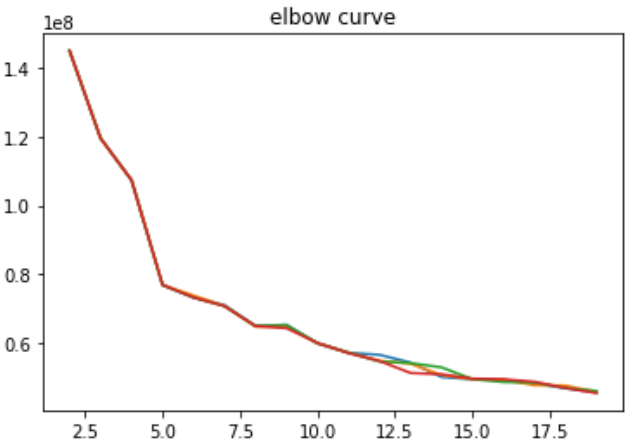

Figure 2 – Elbow Curve Second Problem (NaN -> min)

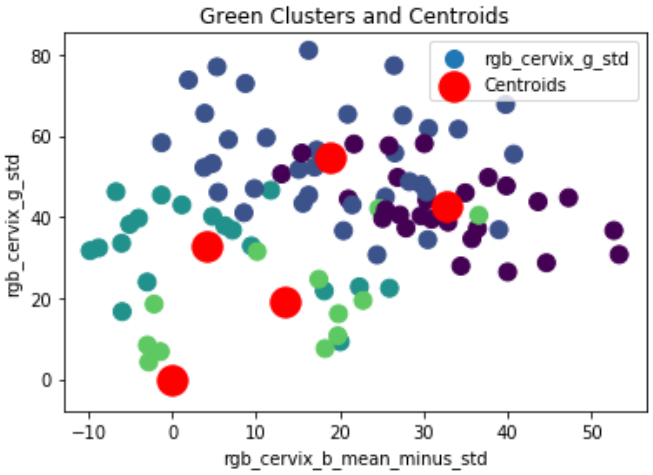

Figure 3 – KMeans First Problem (Green)

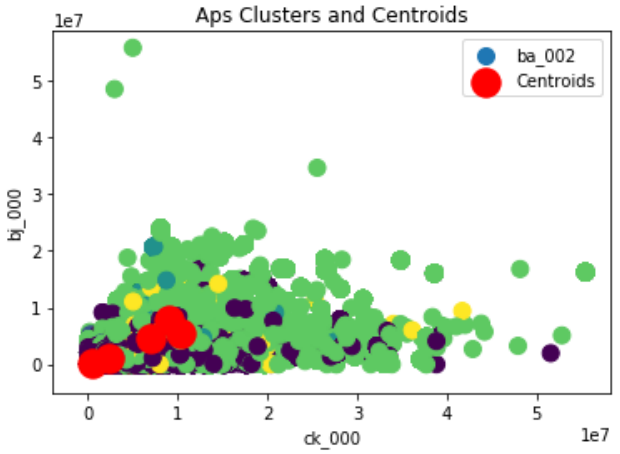

Figure 4 – KMeans Second Problem

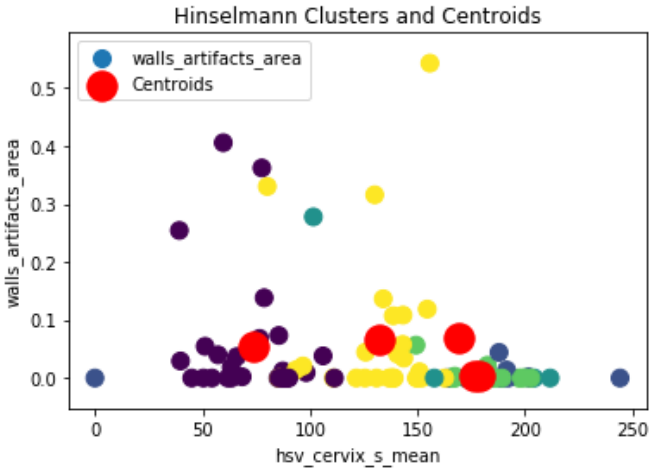

Figure 5 – KMeans First Problem (Hinselmann)

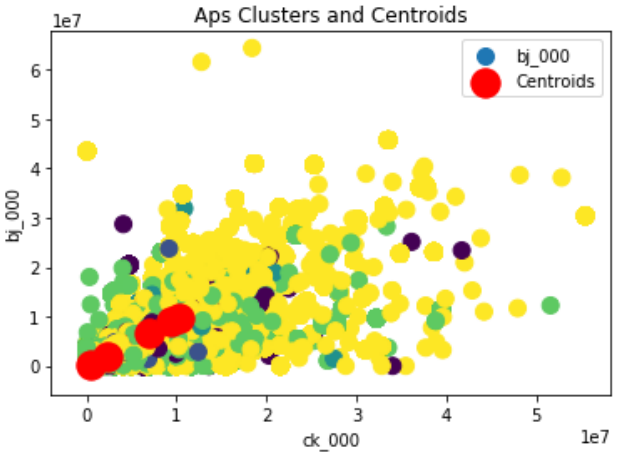

Figure 6 – KMeans Second Problem

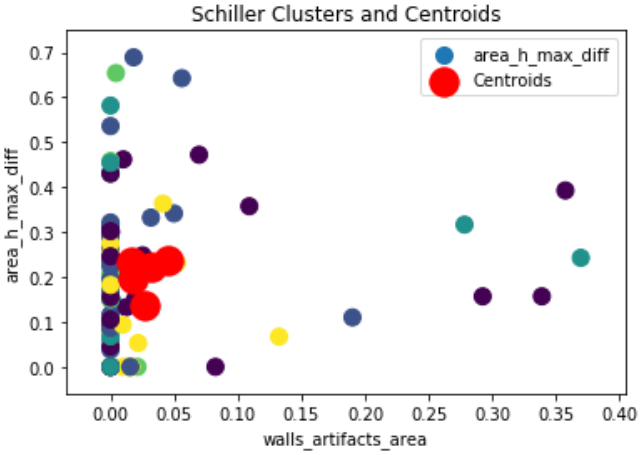

Figure 7 – KMeans First Problem (Schiller)

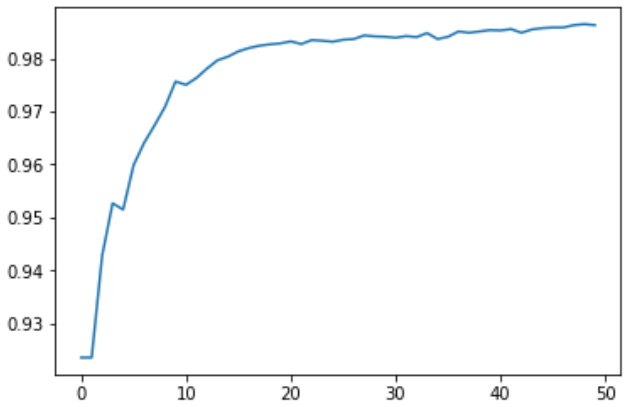

Figure 8 – Accuracy Second Problem

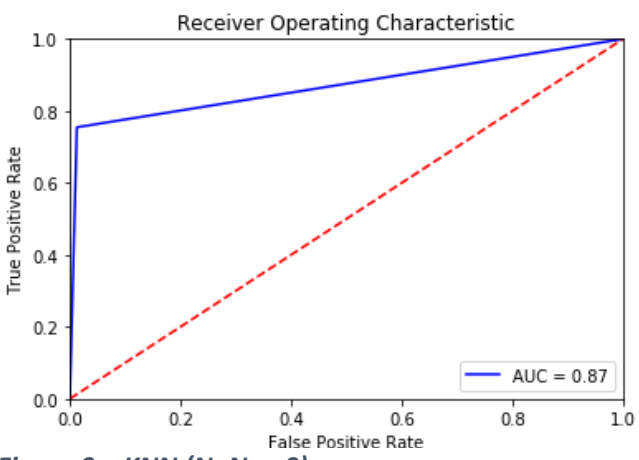

Figure 9 – KNN (NaN -> 0)

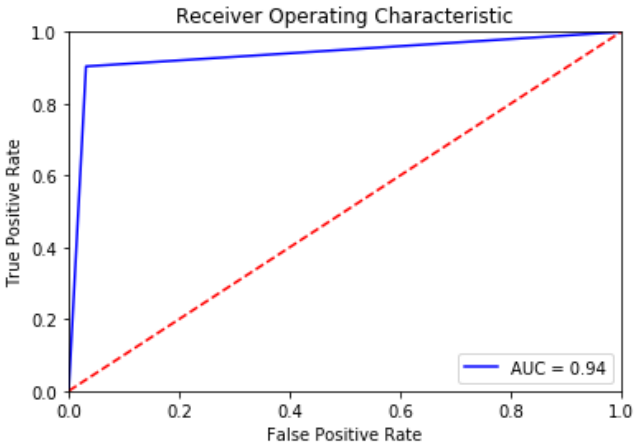

Figure 10 – Naïve Bayes (NaN -> 0)

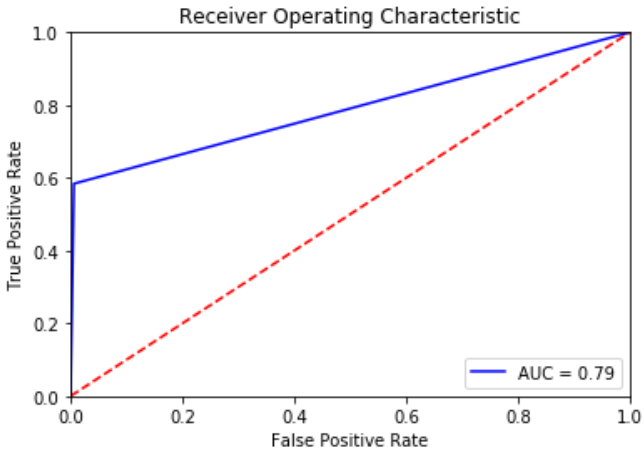

Figure 11 – CART (NaN -> 0)

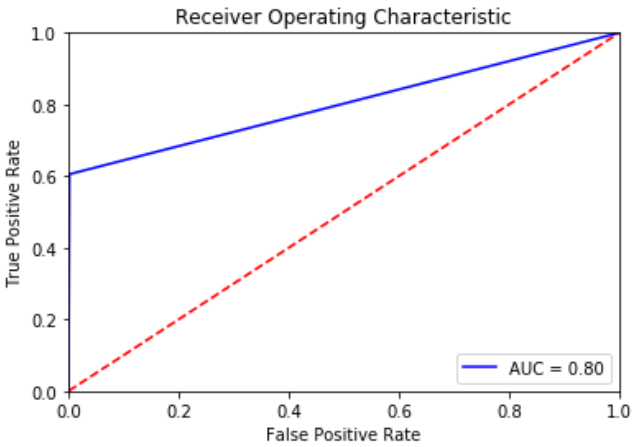

Figure 12 – Random Forest (NaN -> 0)

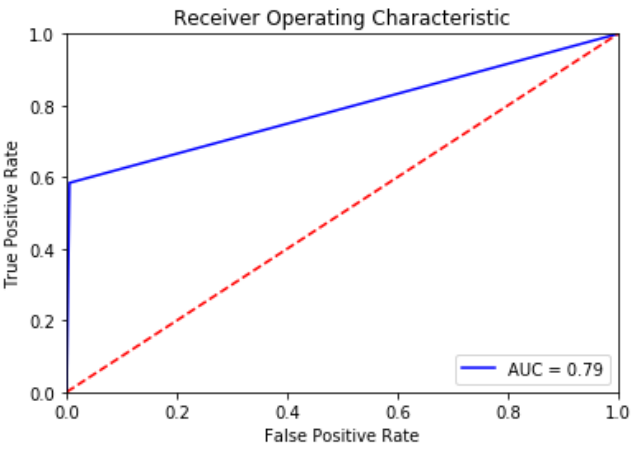

Figure 13 – KNN (NaN -> Min)

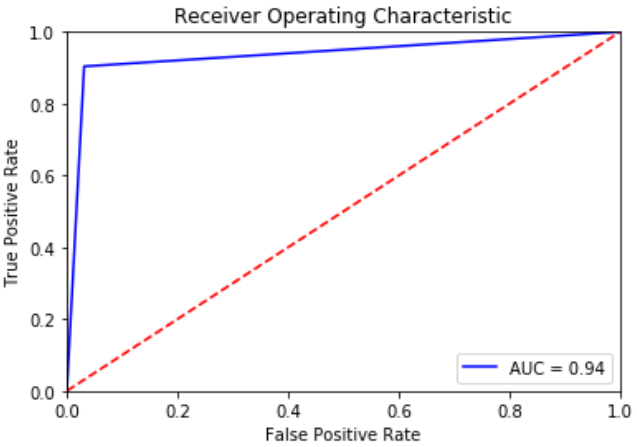

Figure 14 – Naïve Bayes (NaN -> Min)

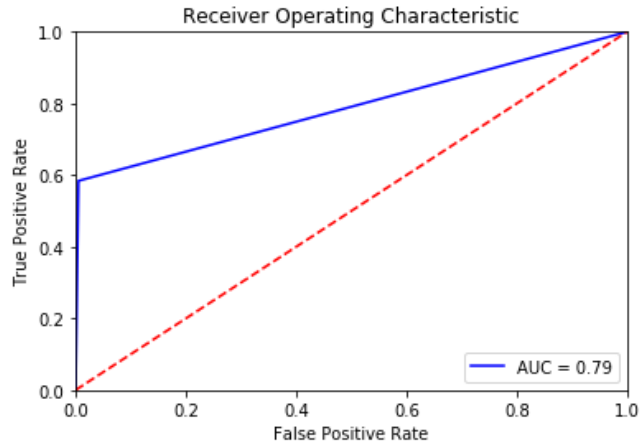

Figure 15 – CART (NaN -> Min)

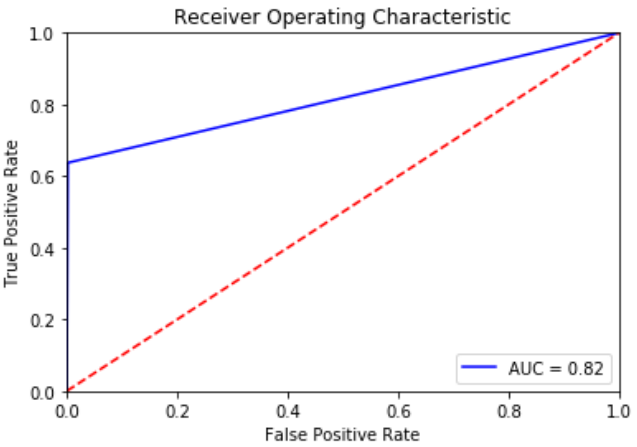

Figure 16 – Random Forest (NaN -> Min)

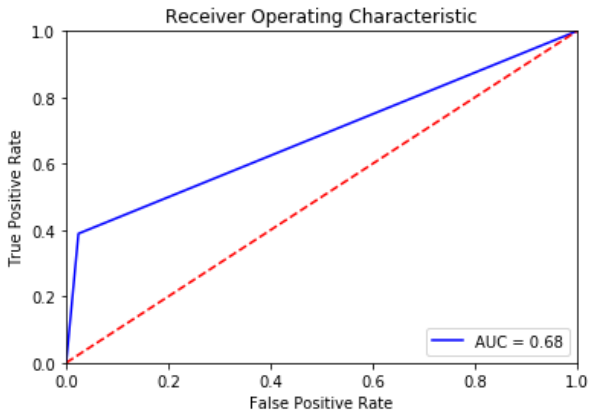

Figure 17 – KNN (NaN -> Max)

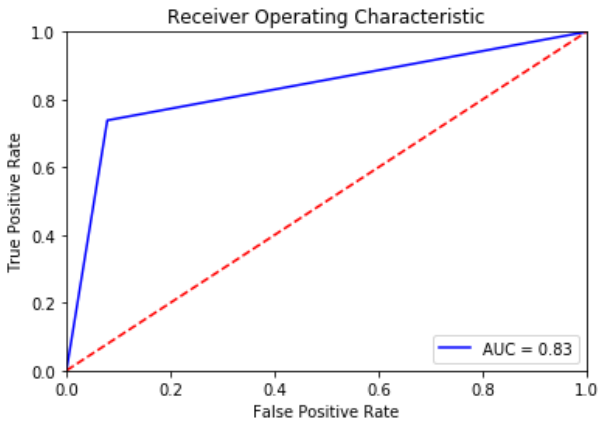

Figure 18 – Naïve Bayes (NaN -> Max)

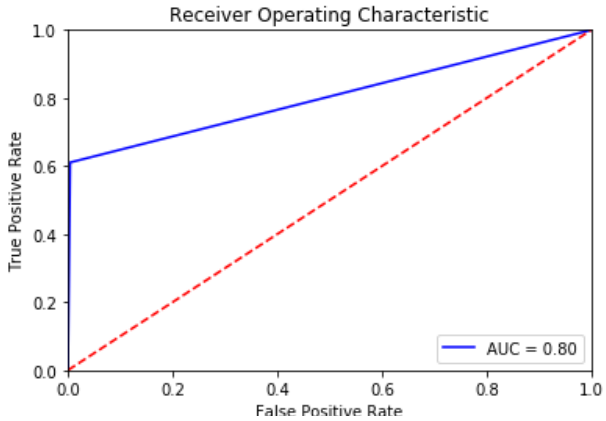

Figure 19 – CART (NaN -> Max)

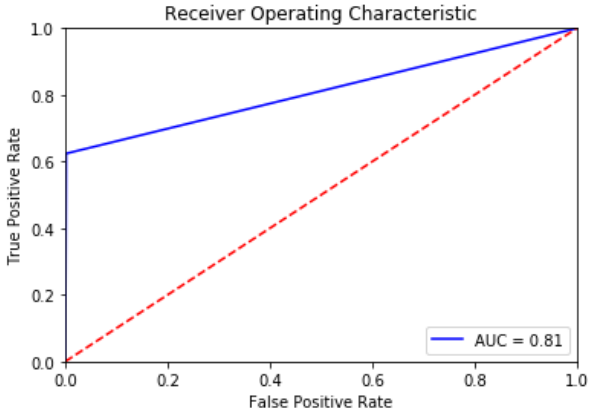

Figure 20 – Random Forest (NaN -> Max)

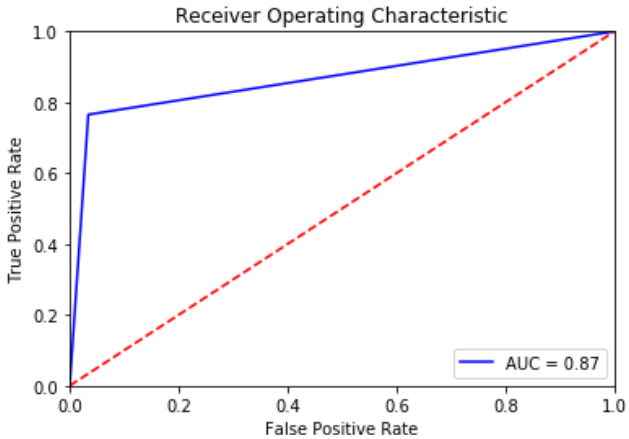

Figure 21 – KNN (NaN -> Mean)

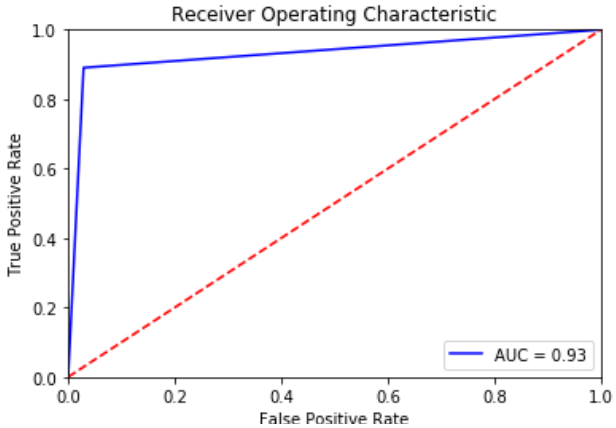

Figure 22 – Naïve Bayes (NaN -> Mean)

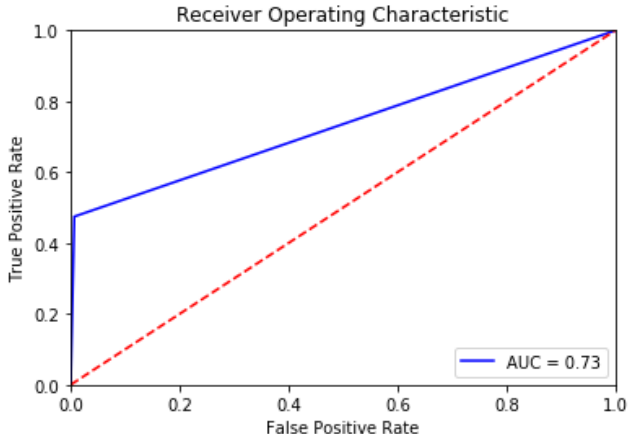

Figure 23 – CART (NaN -> Mean)

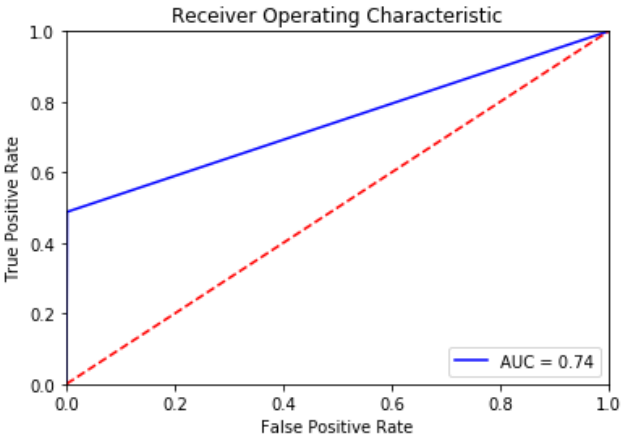

Figure 24 – Random Forest (NaN -> Mean)

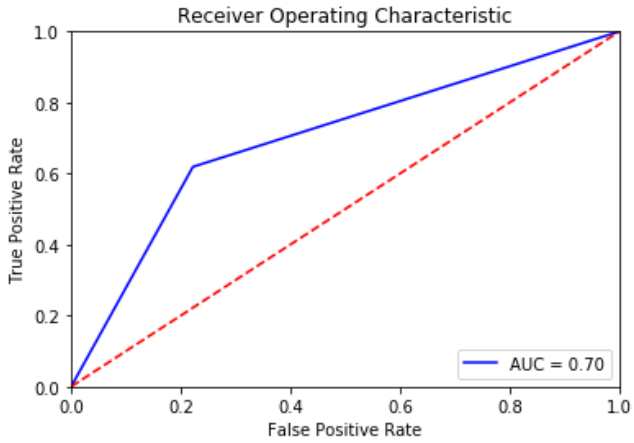

Figure 25 – KNN (Green)

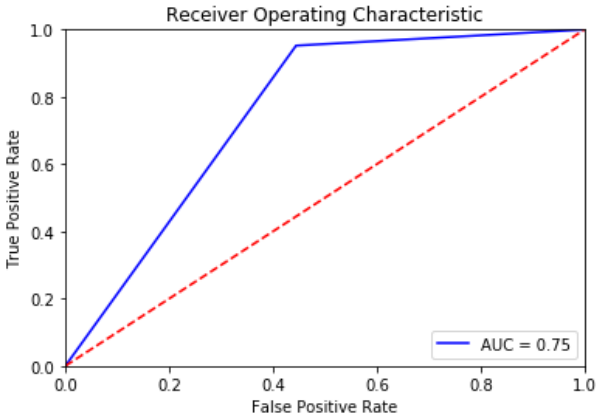

Figure 26 - Naïve Bayes (Green)

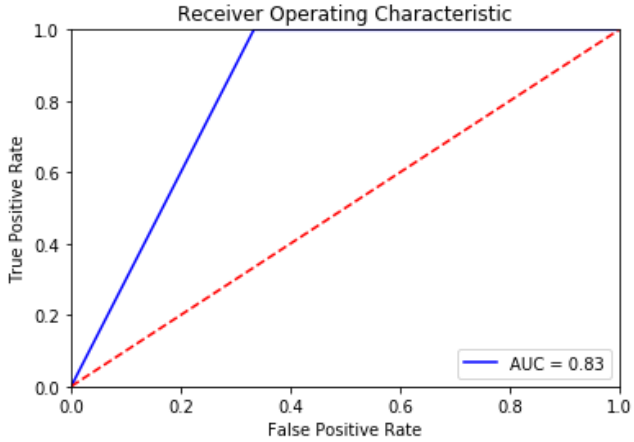

Figure 27 – CART (Green)

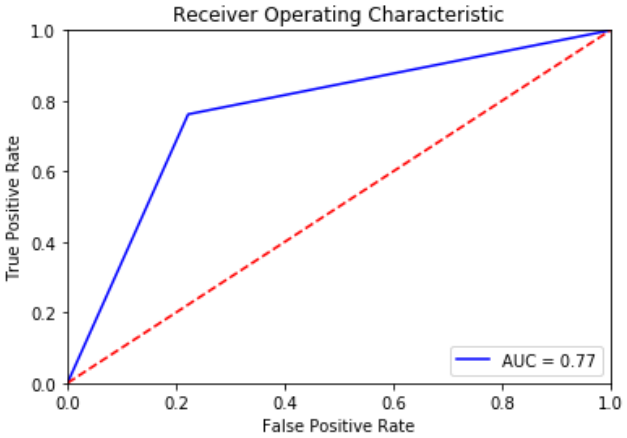

Figure 28 – Random Forest (Green)

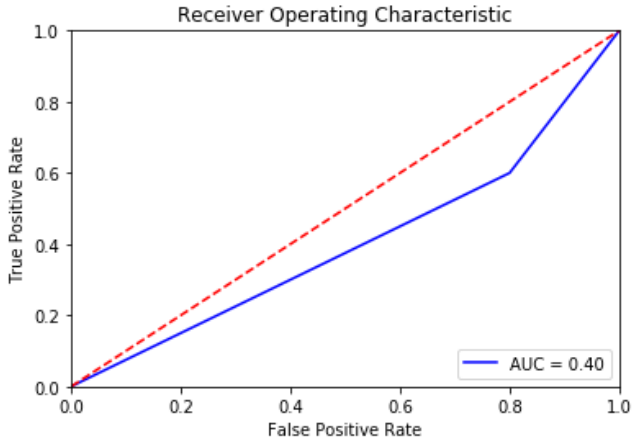

Figure 29 – KNN (Hinselmann)

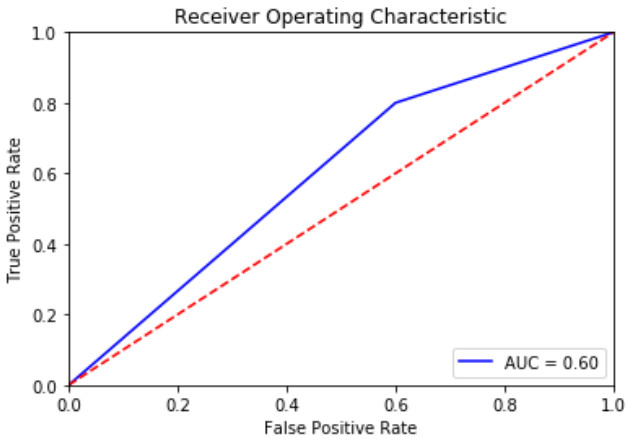

Figure 30 – Naïve Bayes (Hinselmann)

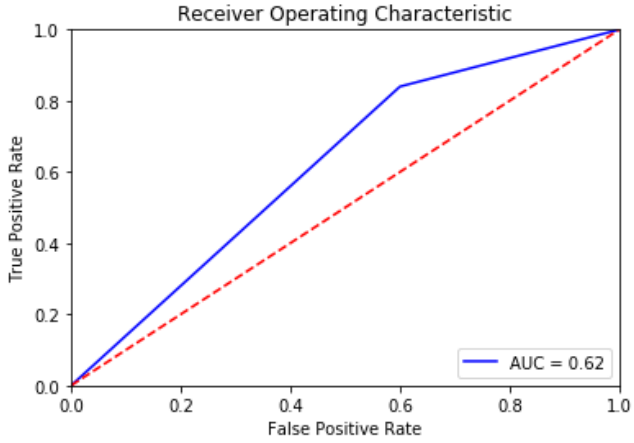

Figure 31 – CART (Hinselmann)

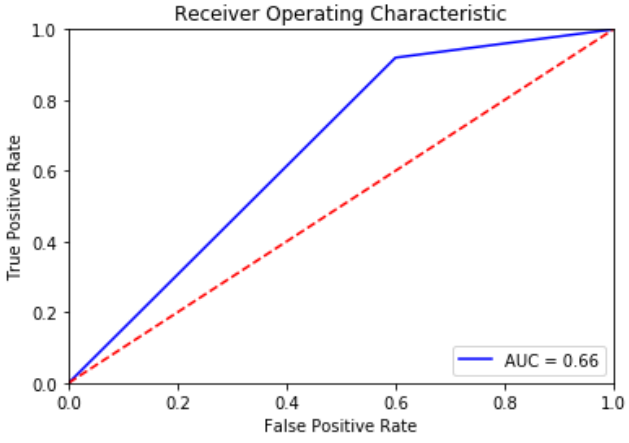

Figure 32 – Random Forest (Hinselmann)

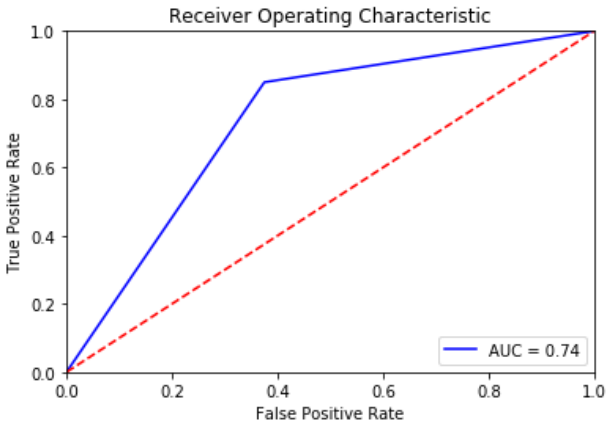

Figure 33 – KNN (schiller)

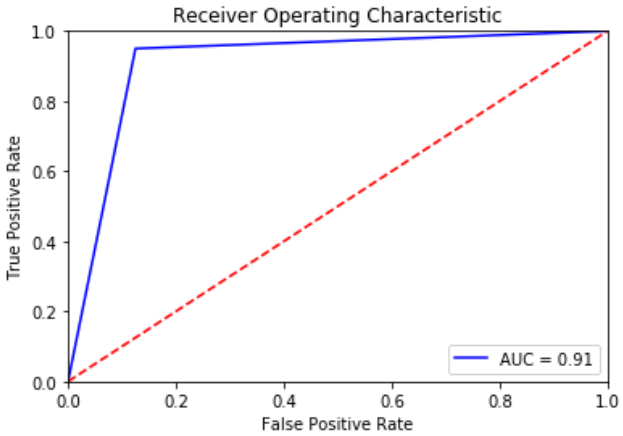

Figure 34 – Naïve Bayes (schiller)

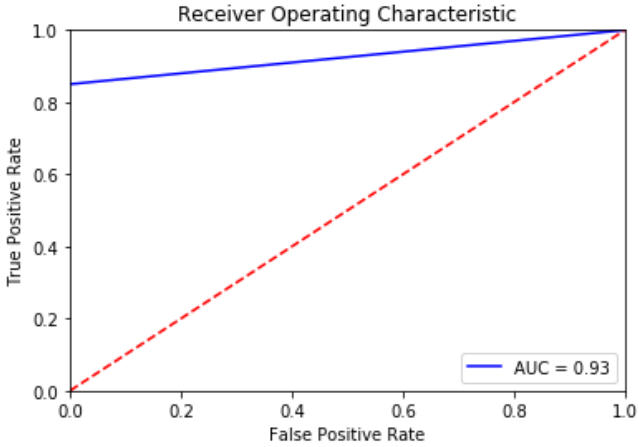

Figure 35 – CART (schiller)

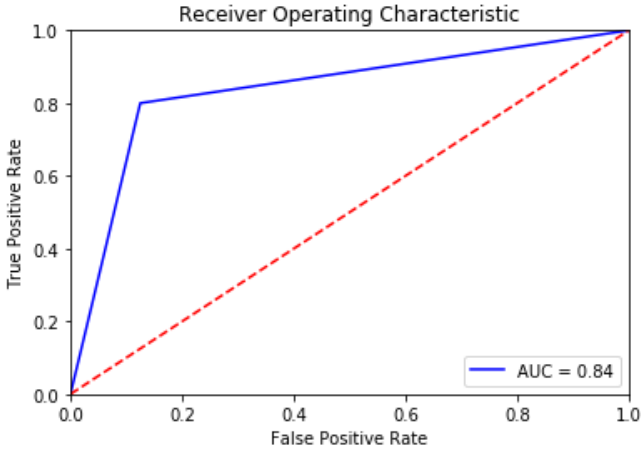

Figure 36 – Random Forest (schiller)