Technical Report

Martin Scheffauer, 51917931

Dominik Geschwinde, 12108977

Gruppe : 01

Einleitung

In diesem Technical Report beschreiben wir unsere Implementierung von Supervised Machine Learning Algorithmen. Es wurden sowohl das Heart Disease als auch das Parkinson Sound Recording Datenset benutzt.

Code Design

Der Code für die Experiments wurde in mehreren Schleifen implementiert. Zum einen die Hauptschleife welche über beide verwendeten Datensets iteriert. Die Daten jedes Datensets werden mit der selbst implementierten Methode split_data der Klasse Dataset in Trainings und Testset aufgeteilt und werden danach mittels dem MinMaxScaler auf den Bereich 0 bis 1 skaliert. Danach folgt die Schleife über alle verwendeten Classifier um Metrics jedes Classifiers zu erhalten.

Die Ergebnisse wurden je in eine Liste aufgenommen um diese Listen wiederum mit Schleifen zu plotten. Eine Schleife für die Confusion Matritzen und eine für die Barcharts. Die Bar Charts wurden hintereinander für jeden Score ausgegeben und die Confusion Matritzen in einem 3 Reihen mal 2 Zeilen Plot. Der Code wurde möglichst so implementiert, dass die Anzahl der Classifier, Datensets sowie der Metrics leicht erhöht oder verringert werden kann da die Schleifen dynamisch alle Elemente durchgehen.

Zusätzlich wurden die Klassen SimpleBaselineClassifier und kNN mit einer abgeleiteten erweiterten Klasse versehen die zusätzlich zur Basisfunktion die Methode get_params() enthält. Diese gibt in einem Dictionary die Parameter der Classifier aus und wird für den cross validation score benötigt, da sonst ein Laufzeitfehler auftritt. Anstatt der echten Klasse wird in den Experiments die abgeleitete Klasse verwendet.

Würde mehr Zeit zur Verfügung stehen würden wir die Wahl der Classifier noch genauer überdenken sowie einzelne Anpassungen im Code vornehmen:

- Implementierung der Methode get_params in eigener Klasse kNN und SimpleBaselineClassifier
- Bessere Darstellung der Plots sowie besseres Handling falls mehr oder weniger Classifier verwendet werden.

Experimentelles Setup

Folgende Classifier mit folgenden Parametern wurden verwendet:

- SimpleBaselineClassifier (random_state = 42)
- Gaussian Naive Bayes
- kNN
- Logistic Regression (random_state = 42)
- Linear Support Vector Classifier (random_state = 42, dual = False)
- Decision Tree Classifier (random_state = 42)

Ein Testdatenset mit der Größe von 0.33 für Parkinson und 0.25 für Heart Disease wurde gewählt.

Für Heart Disease wurden die Features age, sex, trestbps, restecg, slope ausgewählt da diese die höchste Aussagekraft versprachen.

Für das Parkinson Dataset wurden alle Features verwendet, da ein starker Zusammenhang von einzelnen Features nicht erkannt werden konnte.

Fehlende Werte wurden für beide Datensets mit der impute_strategy „mean“ imputed. Diese wurde in der Methode „split_data“ implementiert.

Als Baseline wurde der selbst erstellte Classifier SimpleBaselineClassifier verwendet.

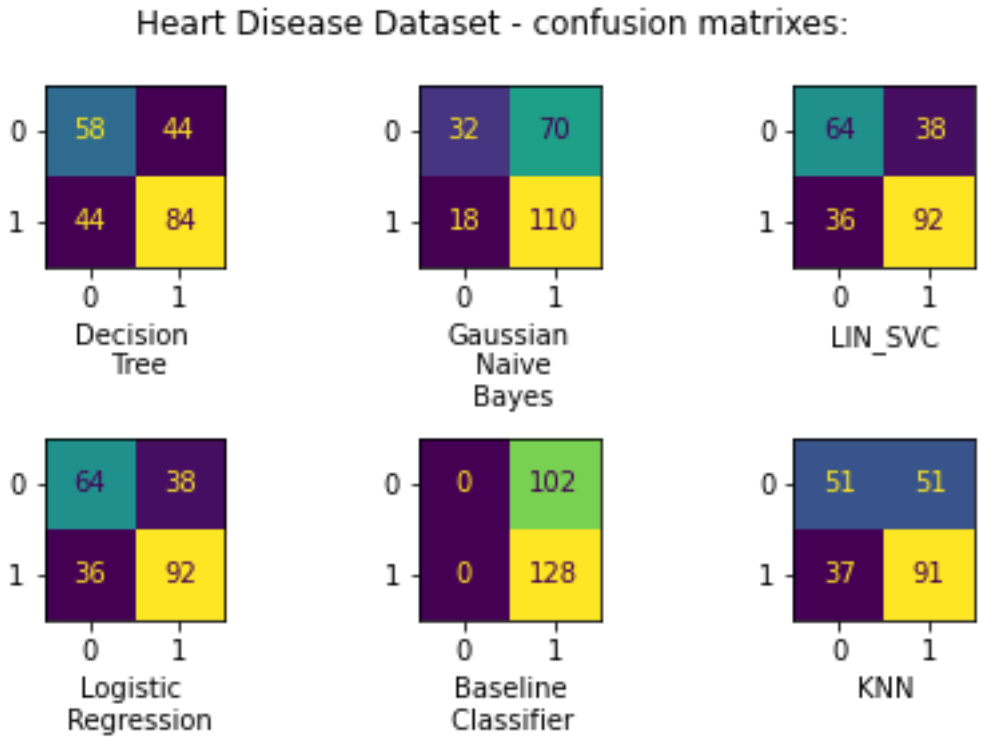
Folgende Metriken wurden für ein Datenset erstellt:

- **Cross validation accuracy score** (n_splits = 10, shuffle = True, random_state = 42)
    - Scoring: accuracy. Hier wurde der „mean“ Wert als Anzeigewert gewählt
    - Mittels Methoden aus sklearn: „KFold“ und „cross_val_score“
- **Accuracy score:** mittels sklearn Methode „accuracy_score“
- **Recall score:** mittels sklearn Methode „recall_score“
- **Confusion matrix:** mittels sklearn Methode „confusion_matrix“

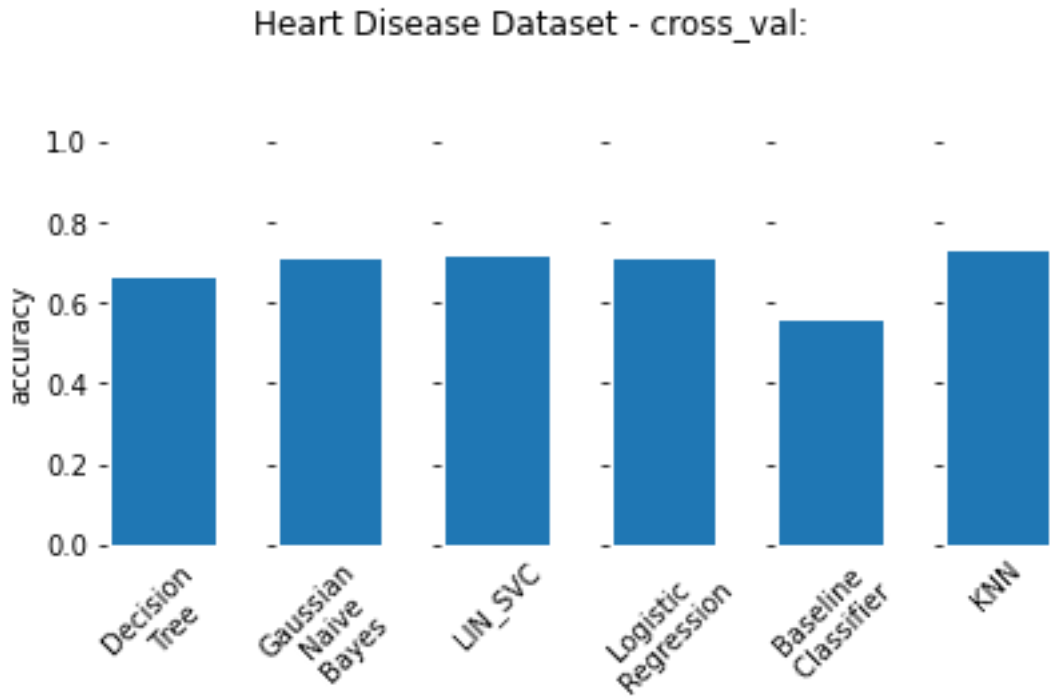
Ergebnisse und Diskussion

Ergebnisse für das Heart Disease Datenset:

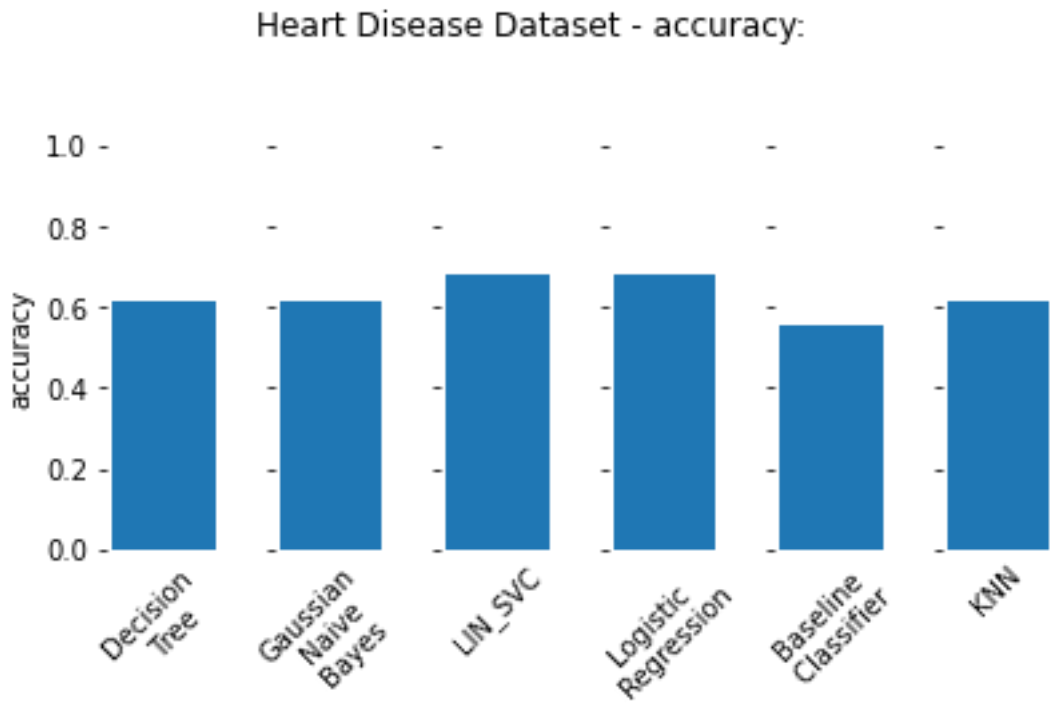
Am schlechtesten schneidet bei der Confusion Matrix die Baseline ab wobei alle Classifier die Baseline schlagen. Am besten schneiden sowohl die SVC als auch die Logistic Regression ab, da die höchsten Werte auf der Hauptdiagonalen erreicht wurden.



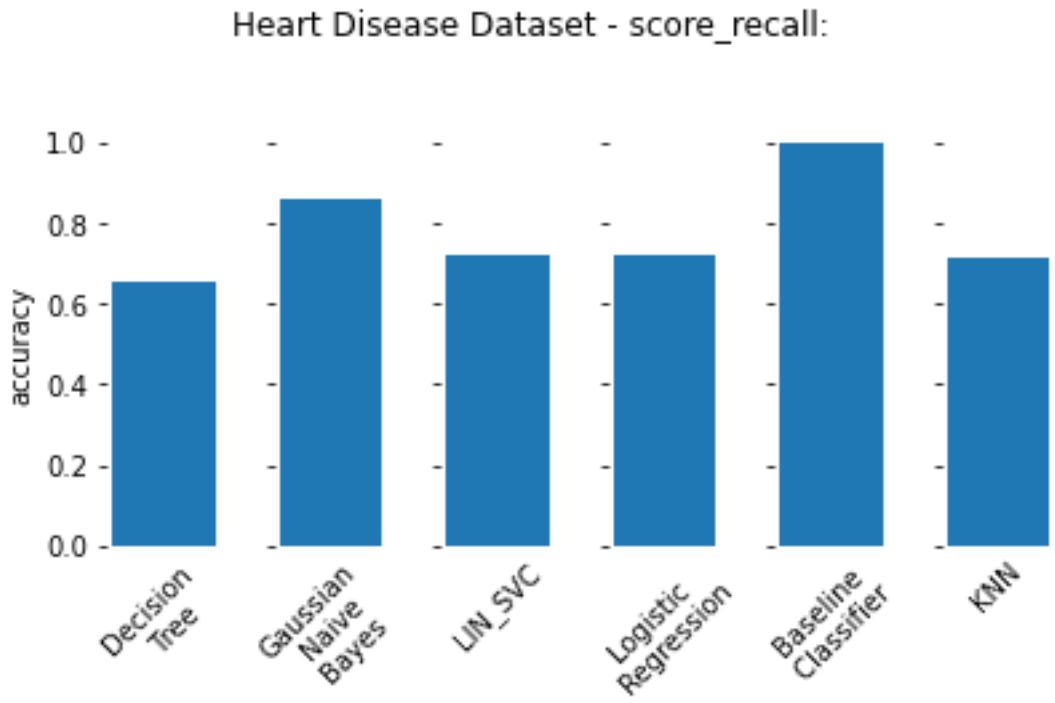
Am besten schneidet für die cross validation der KNN Classifier ab. Alle überbieten die Baseline.



Am Besten schneiden in accuracy SVC und Logistic Regression ab und am schlechtesten die Baseline:

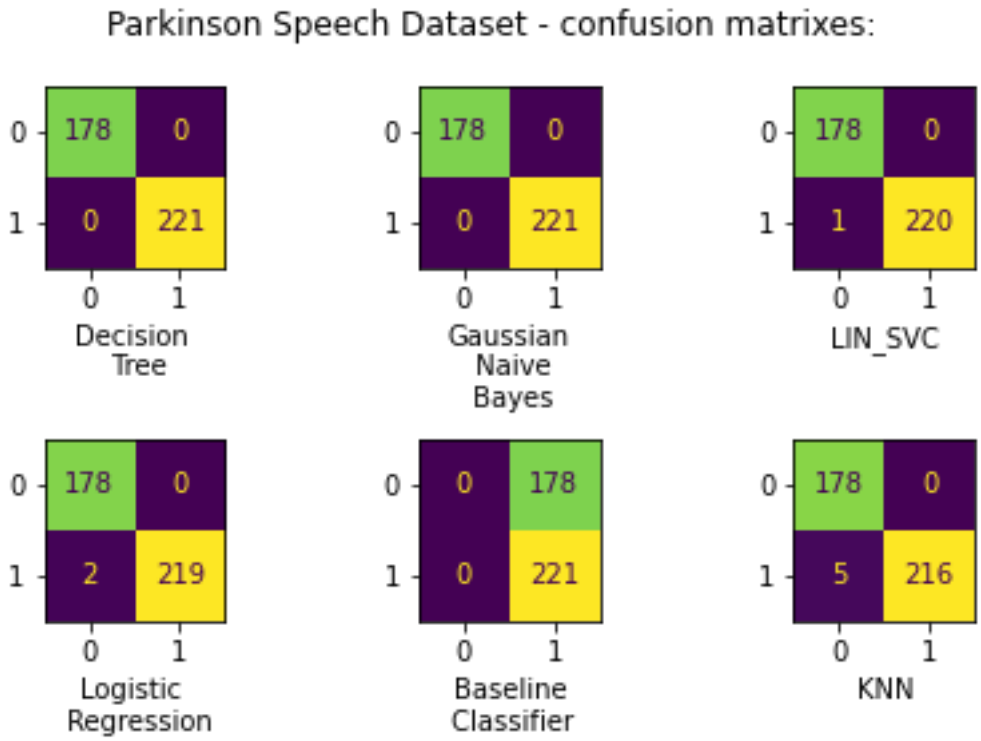


Hier sieht man den Recall Score. Interessant zu sehen ist dass neben der Baseline welche den höchsten Wert aufweist der Gaussian Naive Bayes den zweitbesten Wert aufweist, obwohl dieser bei den anderen Metriken nicht besonders gut war. Am schlechtesten schneidet der Decision Tree ab:

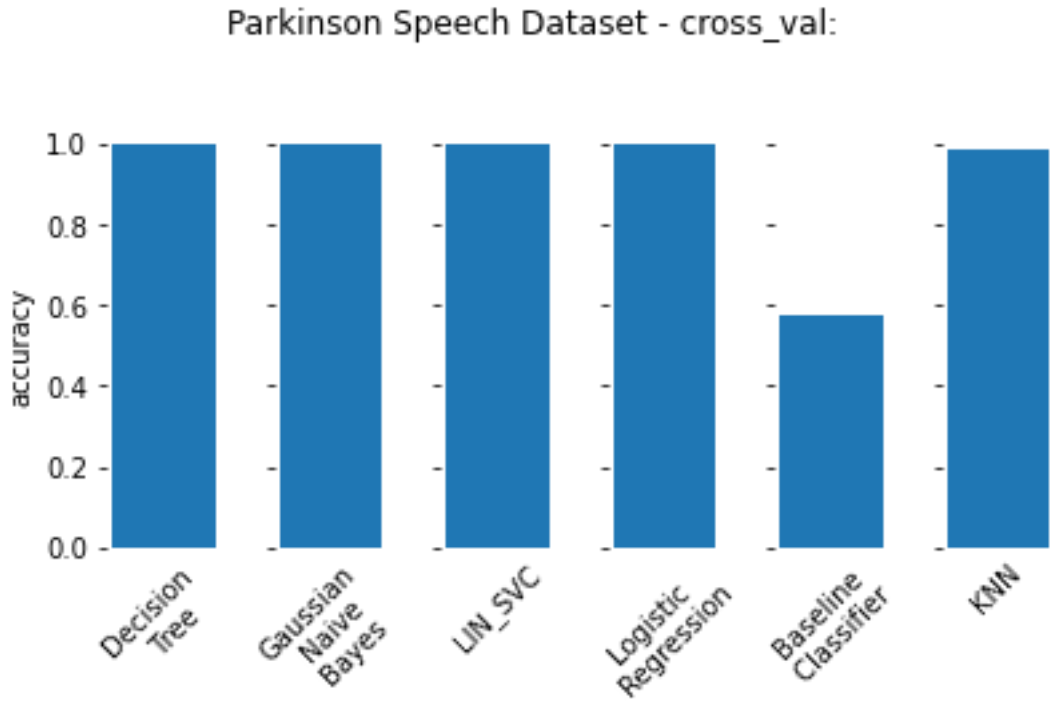


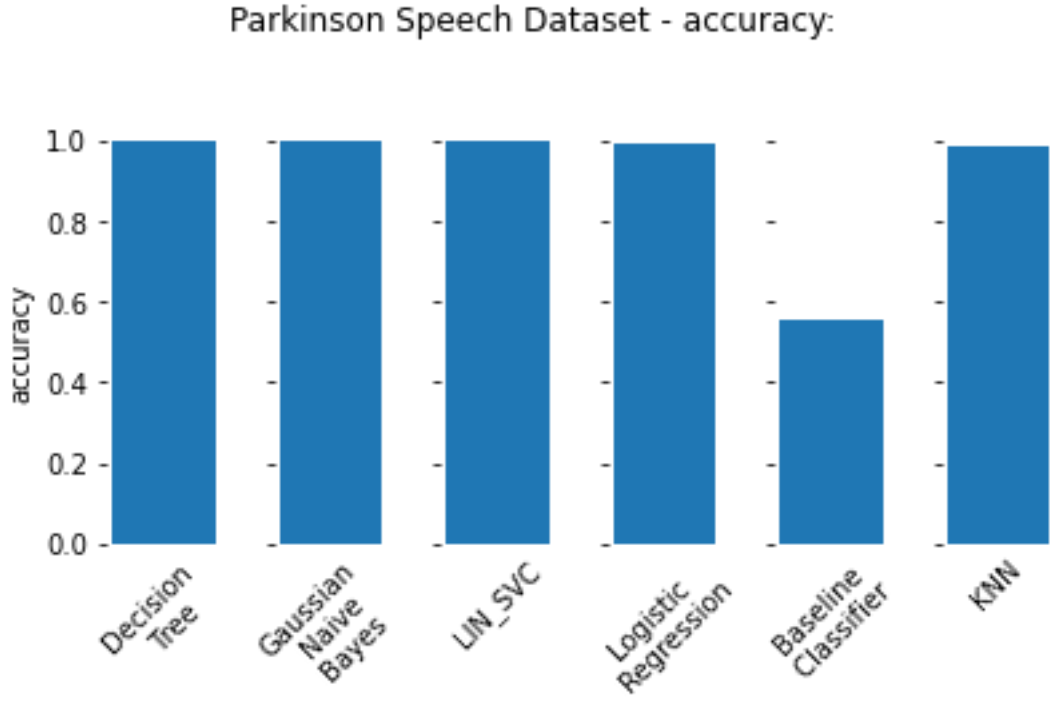
Ergebnisse für das Parkinson Datenset.

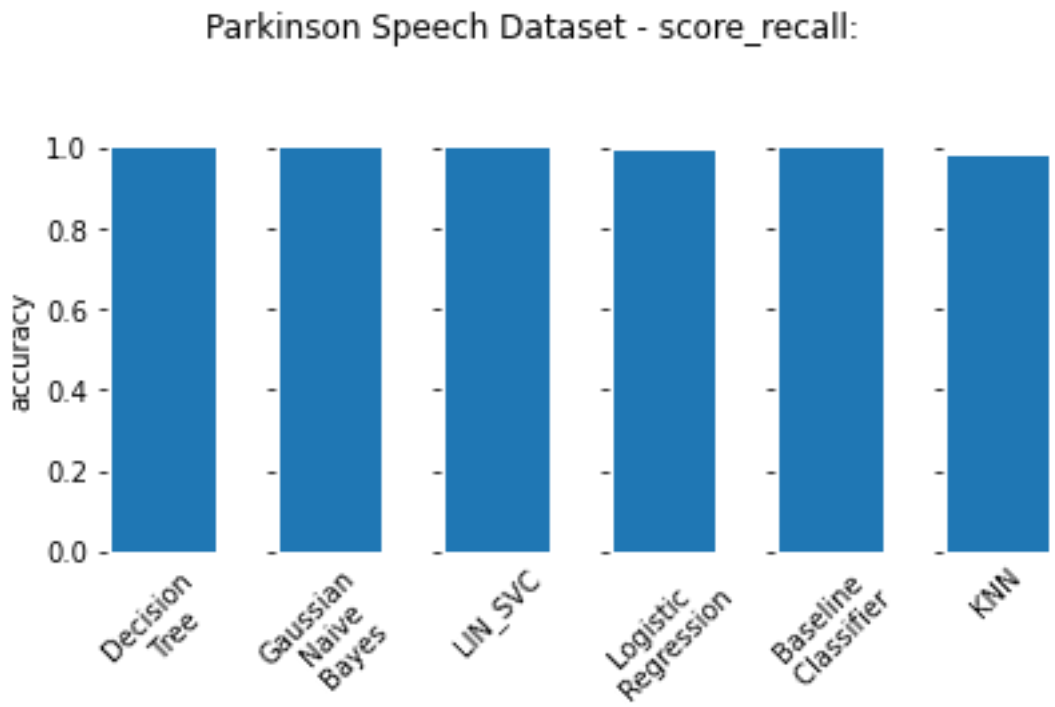
Die Ergebnisse sind bei allen Classifiern nahezu ident und sehr gut da die Hauptdiagonalen am größten sind. Nur die Baseline ist erwartungsgemäß nicht aussagekräftig.



Alle anderen Metriken sind sehr gut bei allen Classifiern außer der Baseline. Dies kann daran liegen dass die Datensätze eine hohe Korrelation aufweisen, das heißt dass bestimmte Messwerte einfach bestimmten Ergebniswerten zugeordnet werden (zB ein hoher Wert in Feature 1 führt immer zu hoher Wahrscheinlichkeit dass die Krankheit auftritt)


Parkinson Speech Dataset - cross_val:
accuracy
1.0
0.8
0.6
0.4
0.2
0.0
Decision Tree
Gaussian Naive Bayes
LIN_SVC
Logistic Regression
Baseline Classifier
KNN


Parkinson Speech Dataset - accuracy:
accuracy
1.0
0.8
0.6
0.4
0.2
0.0
Decision Tree
Gaussian Naive Bayes
LIN_SVC
Logistic Regression
Baseline Classifier
KNN



Fazit

Zusammenfassend kann man erkennen, dass verschiedene Metriken mit verschiedenen Classifiern maximiert werden können und je nachdem ob beispielsweise ein hoher Recall – Wert gewünscht ist kann dieser bei Heart Disease am besten mit Gaussian Naive Bayes erreicht werden. Die Auswahl der verwendeten Features spielt eine große Rolle was sich auch im Vergleich Heart Disease zu Parkinson gut zeigt. Dies sollte jedenfalls genauer untersucht werden und die passenden Features für das Parkinson Datenset gefunden werden.