Experimentaufbau:

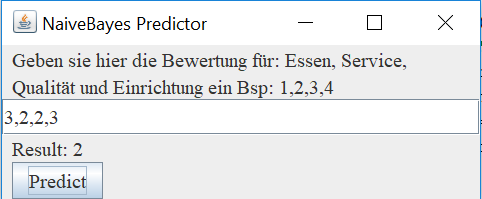
Wir haben uns für das Experiment dazu entschieden einen Naive Bayes Classifier, der die Gesamtbewertung vorhersagt, mit Hilfe von Java zu entwickeln. Dazu verwendeten wir die Weka- Libarary der University of Waikato.

Die Trainingsdaten für den Naive Bayes Classifier der Restaurantbewertungen wurden auf der Internetseite Tripadvisor gesammelt. Da die Tripadvisor API leider für Studenten nicht zugänglich war wurden die Trainingsdaten per Hand in das für Weka spezifische Arff-File geschrieben.

Das Arff- File ist folgendermaßen aufgebaut: Am Anfang der Datei werden die Attribute gesetzt, was in unserem Fall Essen, Service, Qualität, Einrichtung und Gesamtbewertung war. Das letzte Attribut wird im Arff-Format automatisch zum Classifier, was somit die Gesamtbewertung gewesen ist.

Die Trainingsdaten bestehen aus ca. 80 verschiedensten Restaurantbewertungen, die von 5-Sterne Bewertungen bis hin zu 1-Sterne Bewertungen reichen.

Um die Usability zu erhöhen wurde ein kleines User Interface integriert. Dieses wurde mit Java der Java Swing-Libarary integriert und besteht aus einem Beschreibungslabel, einem Texteingabefeld, einem Ergebnisausgabefeld und einem Button, um die Berechnung der Gesamtbewertung zu starten. In der UI-Klasse wir neben der Erstellung des UI auch die Eingabe des Inputfeldes verarbeitet. Die Eingabe des Inputfelds wird in ein Arff- File als Datensatz eingeschrieben und ein neues Testdaten Dokument entsteht. Zudem wird im Eingabefeld überprüft, ob die Eingabe valide ist.



Das User Interface der Anwendung

Für die Berechnung existiert die NaiveBayes-Klasse. In dieser werden zuerst zwei Arff- Dokumente, die aus Trainingsdaten und Testdaten bestehen, mit einem Loader eingelesen.

Danach wird der Classifier auf Gesamtbewertung gesetzt werden. Daraufhin wird ein Naive Bayes Classifier-Model mit Hilfe der Trainingsdaten gebildet, damit das Programm von selbst die Vorhersage der Gesamtbewertung lernen kann. Dieses liest die Trainingsdaten einzeln ein bis das Ende der Datei erreicht ist und gibt das fertige Model zurück.

Die Funktion „nb.classifyInstance(test.instance(0));“ berechnet durch das zuvor erstellte Modell die Gesamtbewertung der Testdaten aus. Danach wird noch der Classifier der TestDaten.arff auf den soeben berechneten Wert gesetzt. Da der Classifier den Index 4 im ersten Testdatensatz besitzt wir dieses als Ergebnis gespeichert und kann von der UI ausgegeben werden.