8.1.5
Correctly Classified Instances 655 —> 65.56%
Incorrectly Classified Instances 316 —> 31.6%

Das Modell funktioniert eher so mittelmäßig. Der Kappa statistic Wert und der Root mean squared error haben nur 0.34 . In den einzelnen Klassen ist die Genauigkeit das Modells fast gleich gut. Die Recall werte sind einmal 0,66 (Klasse 0) & 0,68 (Klasse 1)

8.1.6
Berechnen von Precision & Recall aus der Confusion Matrix:

=== Confusion Matrix ===

```
   a   b   <-- classified as
 324 162 |   a = 0
 154 331 |   b = 1
```

Klasse 0:

TP: 324 FP: 162 FN: 154

Precision = TP/(TP+FP) = 0.6667
Recall = TP/(TP+FN) = 0.678

Klasse 1:

TP: 331 FP: 162 FN: 154

Precision = TP/(TP+FP) = 0.671
Recall = TP/(TP+FN) = 0.682

Formel für F- measure:
F_measure = 2*(Precision * Recall)/ (Precision + Recall)

Klasse 0 : 2* (0.6667*0.678)/ (0.6667+0.678) = 0.672
Klasse 1 : 2* (0.671*0.682)/ (0.671+0.682) = 0.677

Die Werte stimmen mit den Werten von WEKA überein.

8.2
Bei meinem Modell hat sich trotz anpassen und anwenden des filters nichts an dem F-Measure score geändert.

In meinem WEKA finde ich kein simpleNaiveBayes

Random Forest:

Das erstellen des Modells dauert um einiges länger als für einen Naive Bayes Modell. Jedoch ist der F-Measure Wert besser als zuvor. Der Random Forest Classifier kombiniert verschiedene Teilbäume, die individuell nicht die beste Genauigkeit aufweisen aber in Form eines Ensemble eine bessere Aussage treffen können.

Simple Logistic:
Das Modell braucht wieder länger als Naive Bayes aber nur halb so lange wie der Random Forrest Classifier. Die Ergebnisse liegen auch etwas höher als der Naive Bayes Classifier, jedoch hat Random Forest am besten abgeschnitten.
Das Simple Logistic Modell modelliert die Wahrscheinlichkeit eines binären Ereignisses (z. B. Ja/Nein, Erfolg/Misserfolg) basierend auf einer oder mehreren unabhängigen Variablen. Sie verwendet eine logistische Funktion, auch Sigmoid-Funktion genannt, um die Wahrscheinlichkeit zu berechnen. Der Algorithmus passt die Koeffizienten an, indem er eine Maximierung der Likelihood-Funktion durchführt, um die bestmögliche Anpassung an die Trainingsdaten zu erreichen. Das resultierende Modell kann dann verwendet werden, um Vorhersagen für neue Daten zu treffen, indem die Eingabevariablen in die logistische Funktion eingesetzt werden, um die Wahrscheinlichkeit des Ereignisses zu berechnen.

8.3

Die angegebenen Wörter wurden alle durch _NOT in der .arff file ersetzt (so etwas wie “isn’t” usw. auch).

Das sind die Ergebnisse von naive bias mit den veränderten Daten:

Correctly Classified Instances	651	65.1652 %
Incorrectly Classified Instances	320	32.032 %

Eine Verbesserung ist nicht zu sehen (eine minimale Verschlechterung hingegen schon).

Mit Random Forest das gleiche Ergebnis:

Ergebnisse mit unveränderter .arff Datei (random forest):

Correctly Classified Instances	778	77.8779 %
Incorrectly Classified Instances	221	22.1221 %

Ergebnisse mit veränderter .arff Datei (random forest):

Correctly Classified Instances	769	76.977 %
Incorrectly Classified Instances	230	23.023 %