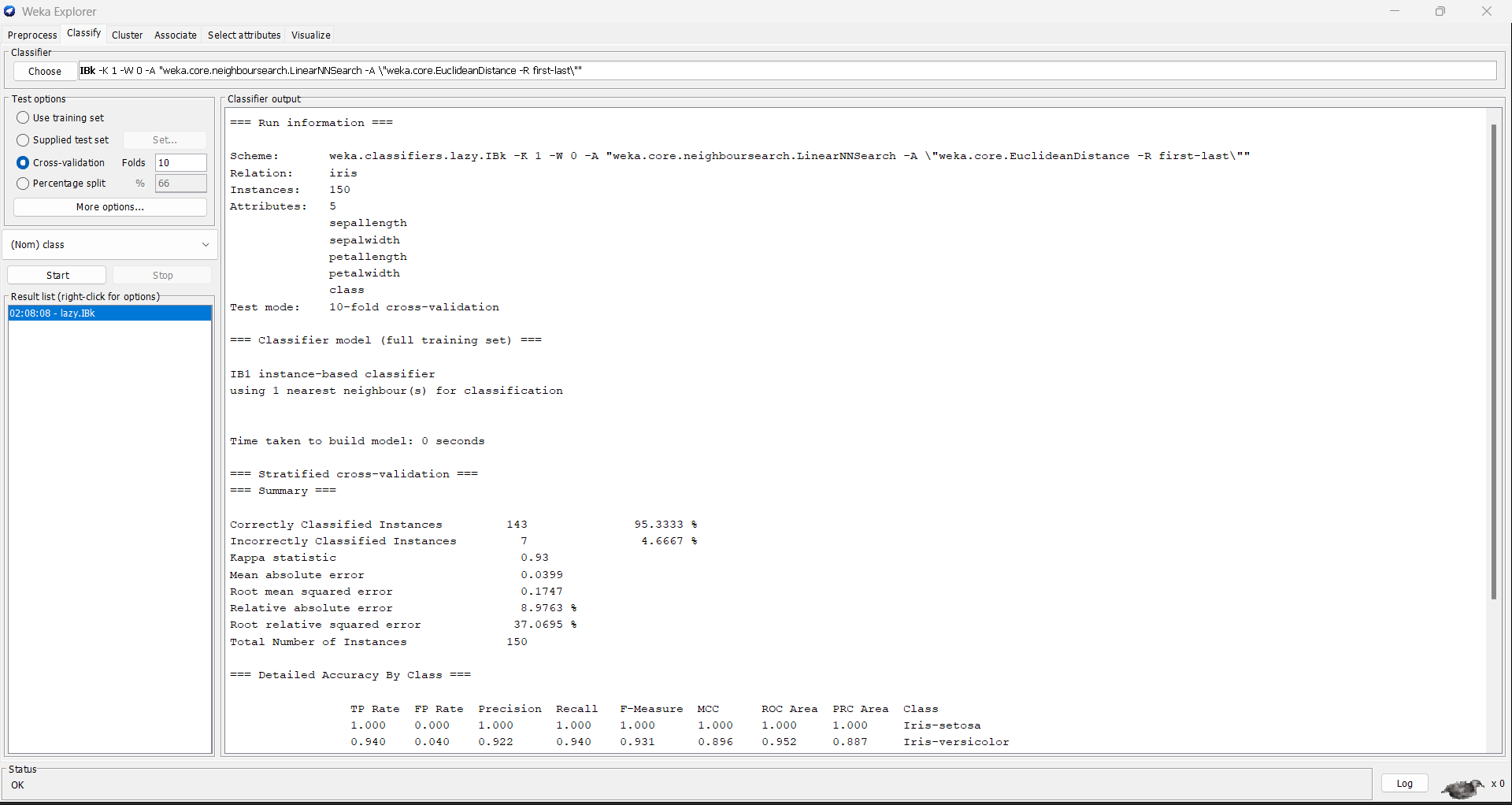
**Part 1 (50 pts)**

**1. Using Sequential Backward Selection (aka backwards elimination)**

Graphical user interface

Description automatically generated with low confidence

**Classification Accuracy**



**Using Sequential Backward Selection (aka backwards elimination)** Graphical user interface

Description automatically generated with medium confidence

**Classification Accuracy**

Text

Description automatically generated

**Using Sequential Backward Selection (aka backwards elimination)**

Graphical user interface, application, Word

Description automatically generated

**Classification Accuracy**

Text

Description automatically generated

**Subset size 3**

**Attributes in “best” subset ALL**

**Using Sequential Backward Selection (aka backwards elimination)**

Graphical user interface, application

Description automatically generated

**Classification Accuracy**

Text

Description automatically generated

**Subset size 1**

**Attributes in “best” subset ALL**

**Using Sequential Backward Selection (aka backwards elimination)**

Graphical user interface, application

Description automatically generated

**Classification Accuracy**

Text

Description automatically generated with medium confidence

**Subset size 5**

Chart

Description automatically generated with medium confidence

**Subset size 3**

Graphical user interface, application

Description automatically generated

**Subset size 2**

Graphical user interface, application

Description automatically generated

**Subset size 1**

Graphical user interface

Description automatically generated

**Given that you did not explore all possible subset of features, is this "backwards elimination" approach guaranteed to find the optimal set of features?**

**No, because it only analyses a tiny portion of all potential feature combinations, the "backwards elimination" strategy is not guaranteed to uncover the best collection of features.**

|  |  |  |
| --- | --- | --- |
| **Subset Size** | **Attributes in “best” subset** | **Classification Accuracy** |
| **4** | **ALL** | **95.3333%** |
| **3** | **Sepallength,petellength,peterwidth** | **94.667%** |
| **2** | **Sepalength,petellength** | **96.0%** |
| **1** | **peterwidth** | **33.33%** |

**In most cases, exploring all potential feature combinations is impractical since the number of potential combinations increases exponentially with the number of characteristics. For example, if we have ten features, there are 210 (1024) potential feature combinations. If we have 100 features, we have 2100 (1.2676506e+30) potential feature combinations. This renders a computationally infeasible, if not impossible, exhaustive search of all potential combinations.**

**To avoid this issue, we may use a technique known as Sequential Backward Selection (SBS). SBS is a feature selection method that chooses a subset of features by deleting the least relevant feature repeatedly at each step. This method is capable of reducing the number of features to a reasonable amount while preserving a high level of ability to forecast.**

**Part 2 (50 pts)**

**Using Principal Components Analysis (aka PCA)**

**Subset size 3**

**Graphical user interface, application

Description automatically generated**

**Report the Accuracy by populating**

**A picture containing text

Description automatically generated**

**Using Principal Components Analysis (aka PCA)**

**Subset size 2**

**A picture containing graphical user interface

Description automatically generated**

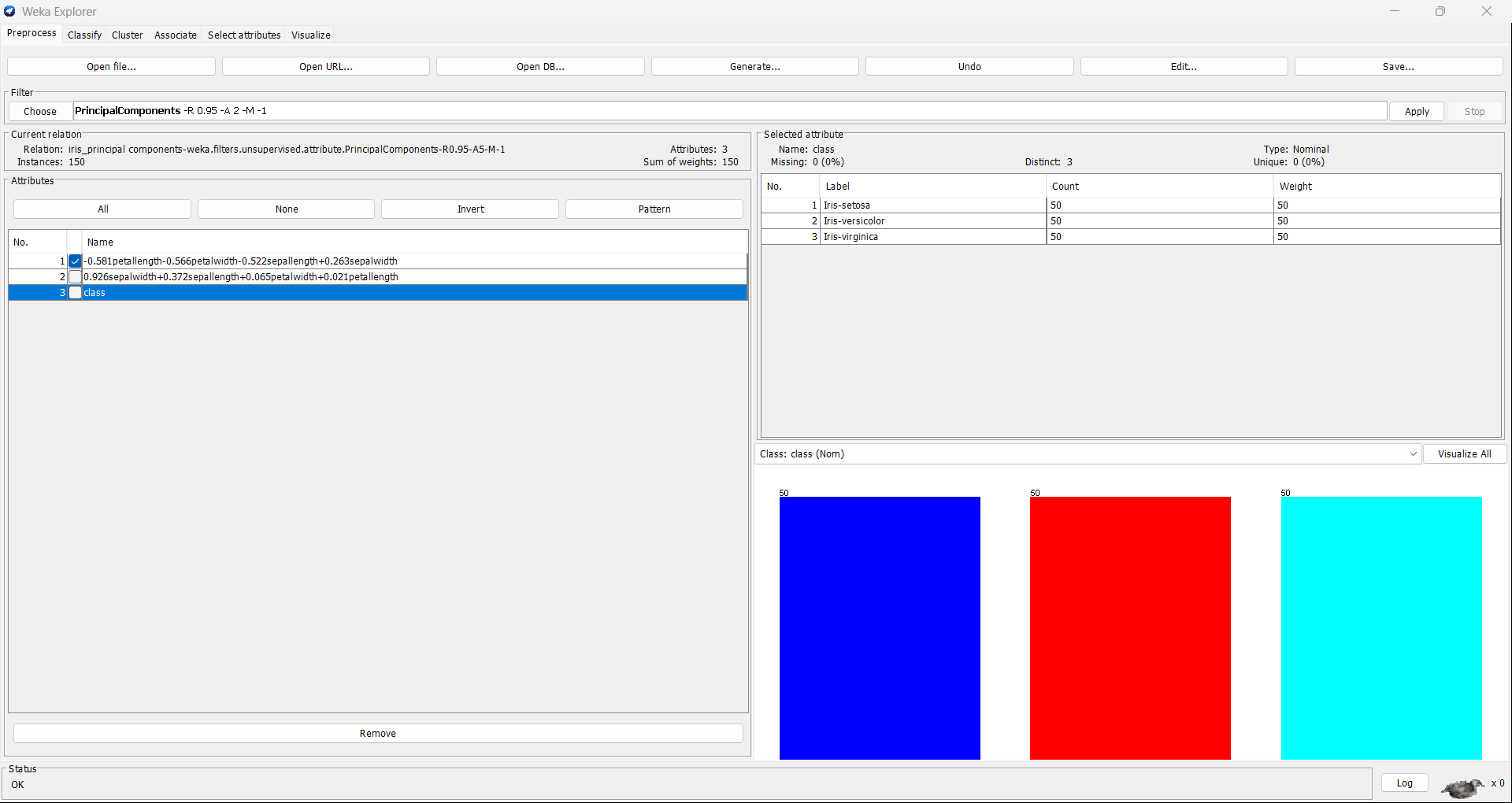
**Report the Accuracy by populating**

**Text

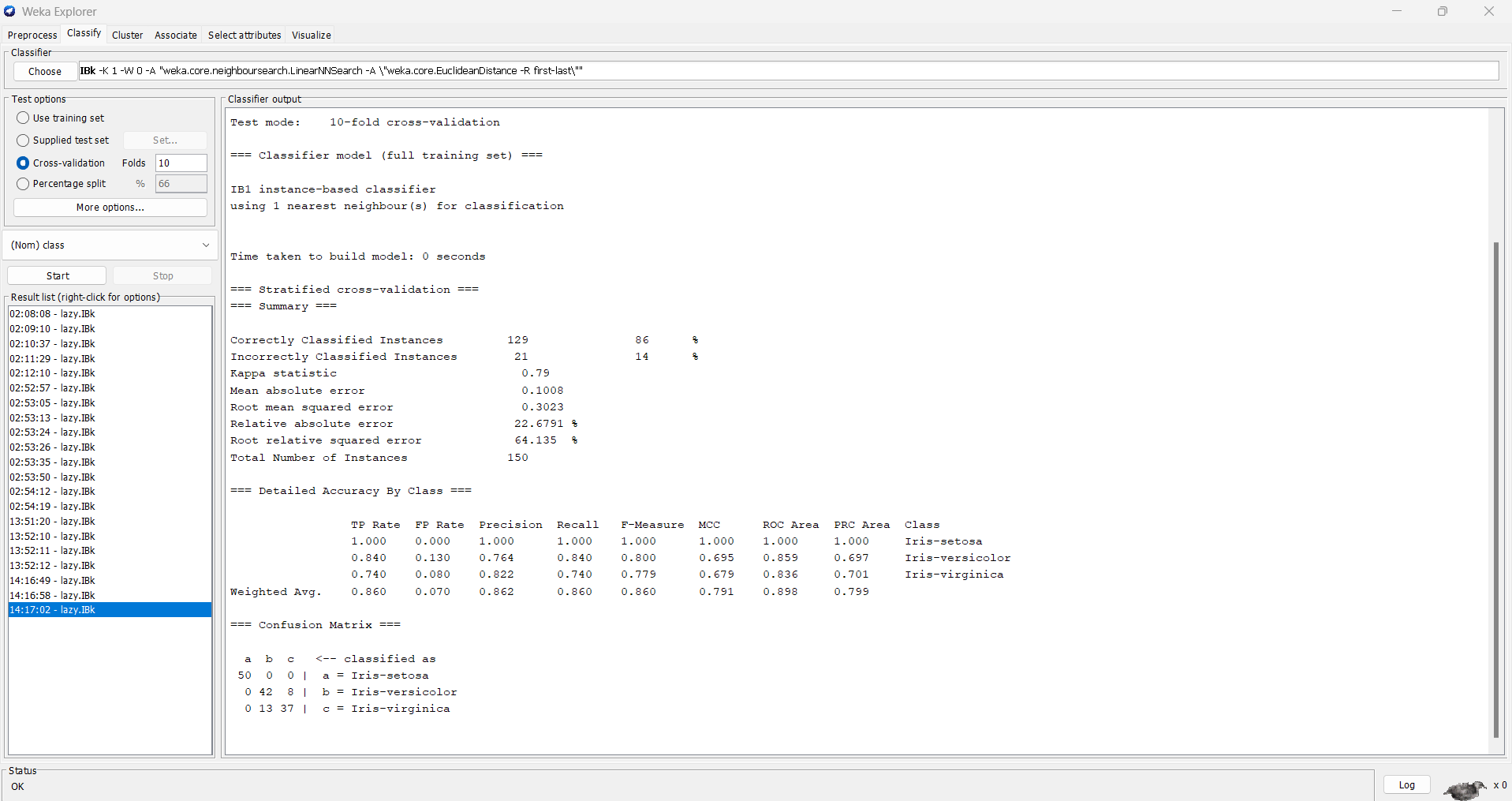
Description automatically generated**

**Using Principal Components Analysis (aka PCA)**

**Subset size 1**

****

**Report the Accuracy by populating**

****

|  |  |  |
| --- | --- | --- |
| **Subset Size** | **Attributes in “best” subset** | **Classification Accuracy** |
| **3** | **All the three** | **86.0%** |
| **2** | **Except Class Subset** | **91.0%** |
| **1** | **First Subset** | **48.0%** |

**02:53:35: Started weka.classifiers.lazy.IBk**

**02:53:35: Command: weka.classifiers.lazy.IBk -K 1 -W 0 -A "weka.core.neighboursearch.LinearNNSearch -A \"weka.core.EuclideanDistance -R first-last\""**

**02:53:35: Finished weka.classifiers.lazy.IBk**

**02:53:50: Started weka.classifiers.lazy.IBk**

**02:53:50: Command: weka.classifiers.lazy.IBk -K 1 -W 0 -A "weka.core.neighboursearch.LinearNNSearch -A \"weka.core.EuclideanDistance -R first-last\""**

**02:53:50: Finished weka.classifiers.lazy.IBk**

**02:54:12: Started weka.classifiers.lazy.IBk**

**02:54:12: Command: weka.classifiers.lazy.IBk -K 1 -W 0 -A "weka.core.neighboursearch.LinearNNSearch -A \"weka.core.EuclideanDistance -R first-last\""**

**02:54:12: Finished weka.classifiers.lazy.IBk**

**02:54:19: Started weka.classifiers.lazy.IBk**

**02:54:19: Command: weka.classifiers.lazy.IBk -K 1 -W 0 -A "weka.core.neighboursearch.LinearNNSearch -A \"weka.core.EuclideanDistance -R first-last\""**

**02:54:19: Finished weka.classifiers.lazy.IBk**

**13:51:20: Started weka.classifiers.lazy.IBk**

**13:51:20: Command: weka.classifiers.lazy.IBk -K 1 -W 0 -A "weka.core.neighboursearch.LinearNNSearch -A \"weka.core.EuclideanDistance -R first-last\""**

**13:51:20: Finished weka.classifiers.lazy.IBk**

**13:52:10: Started weka.classifiers.lazy.IBk**

**13:52:10: Command: weka.classifiers.lazy.IBk -K 1 -W 0 -A "weka.core.neighboursearch.LinearNNSearch -A \"weka.core.EuclideanDistance -R first-last\""**

**13:52:10: Finished weka.classifiers.lazy.IBk**

**13:52:11: Started weka.classifiers.lazy.IBk**

**13:52:11: Command: weka.classifiers.lazy.IBk -K 1 -W 0 -A "weka.core.neighboursearch.LinearNNSearch -A \"weka.core.EuclideanDistance -R first-last\""**

**13:52:11: Finished weka.classifiers.lazy.IBk**

**13:52:12: Started weka.classifiers.lazy.IBk**

**13:52:12: Command: weka.classifiers.lazy.IBk -K 1 -W 0 -A "weka.core.neighboursearch.LinearNNSearch -A \"weka.core.EuclideanDistance -R first-last\""**

**13:52:12: Finished weka.classifiers.lazy.IBk**

**14:16:49: Started weka.classifiers.lazy.IBk**

**14:16:49: Command: weka.classifiers.lazy.IBk -K 1 -W 0 -A "weka.core.neighboursearch.LinearNNSearch -A \"weka.core.EuclideanDistance -R first-last\""**

**14:16:49: Finished weka.classifiers.lazy.IBk**

**14:16:58: Started weka.classifiers.lazy.IBk**

**14:16:58: Command: weka.classifiers.lazy.IBk -K 1 -W 0 -A "weka.core.neighboursearch.LinearNNSearch -A \"weka.core.EuclideanDistance -R first-last\""**

**14:16:58: Finished weka.classifiers.lazy.IBk**