```
package first;

//package first;
import java.io.BufferedReader;
import java.io.FileReader;
import java.util.Arrays;
import java.util.Random;

import weka.attributeSelection.BestFirst;
import weka.attributeSelection.CfsSubsetEval;
import weka.attributeSelection.InfoGainAttributeEva
l;
import weka.attributeSelection.PrincipalComponents;

import weka.attributeSelection.Ranker;
import weka.core.Instances;
import weka.core.stemmers.IteratedLovinsStemmer;
import weka.core.tokenizers.NGramTokenizer;
import weka.classifiers.Classifier;
import weka.classifiers.Evaluation;
import weka.classifiers.lazy.IBk;
import weka.filters.Filter;
import weka.filters.unsupervised.attribute.StringTo
WordVector;
import weka.filters.supervised.attribute.AttributeS
election;
import weka.core.OptionHandler;
public class demo {



    public static void main(String args[]) throws E
xception{
    // load data

    Instances data = new Instances(new BufferedRead
er(new
    FileReader("C:/Users/user/Desktop/out.arff")));


    // specifying k value
    String[] options = weka.core.Utils.splitOptions
```

```
("-K 3");
    data.setClassIndex(data.numAttributes() - 1);

    //build model
    StringToWordVector filter = new StringToWordVec
tor();
    filter.setInputFormat(data);
    Instances newTrain = Filter.useFilter(data, fil
ter);
    AttributeSelection selector=new AttributeSelect
ion();
    BestFirst ser=new BestFirst();
    CfsSubsetEval pc=new CfsSubsetEval();
    selector.setEvaluator(pc);
    selector.setSearch(ser);
    selector.setInputFormat(newTrain);
    Instances newData = Filter.useFilter(newTrain,
selector);
    Classifier ibk = new IBk();
    ibk.setOptions(options);
    ibk.buildClassifier(newData);
    Evaluation eval = new Evaluation(newData);
    eval.crossValidateModel(ibk, newData, 10, new R
andom(1));
    System.out.println(ibk);
    System.out.println(eval.toSummaryString("\nResu
lts\n===\n", false));
     System.out.println(eval.toMatrixString());

    }
    }
```