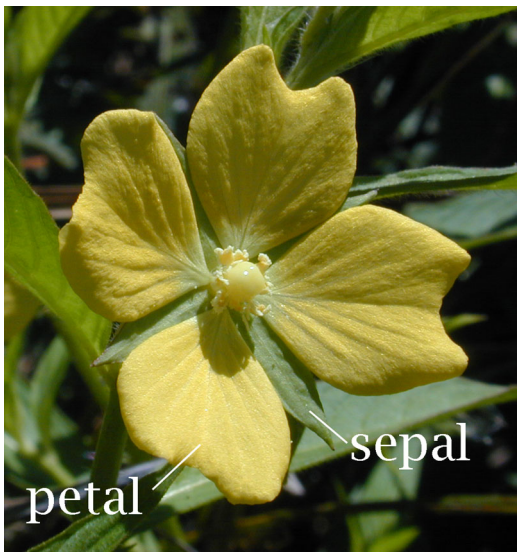
IoT Lab-5 (Week-11 Session-3)

Topic: IoT data analytics

Objective: Data classification using Weka on an Android embedded platform

Follow these steps to design an Android App running on Udoo and use the Weka k-NN classifier to classify the Iris flower species (*Iris setosa*, *Iris virginica* and *Iris versicolor*). The 4 features (attributes) are the length and the width of the sepals and petals, in centimeters.



1. Download and install Weka

URL: http://sourceforge.net/projects/weka/

In the downloaded package, you will find an application and a folder:

- weka-3-6-13-oracle-jvm.app
- weka-3-6-13

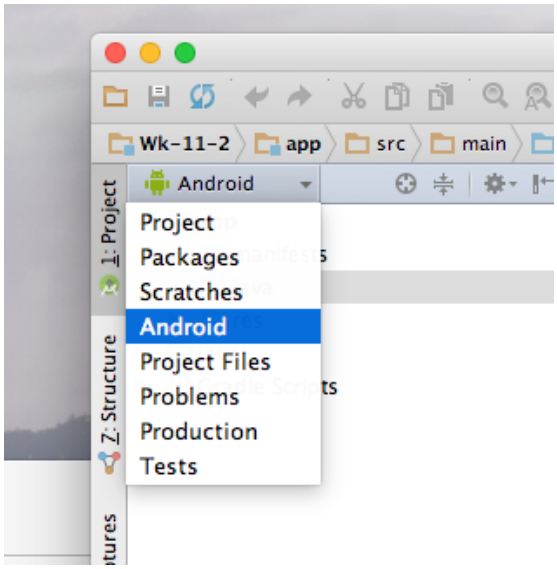
Inside '*weka-3-6-13*', look for '*weka.jar*'.

JAR is a Java archive, packages many Java class files and other resources into one file to distribute as a library.

2. Import Weka library into an Android App

Create an Android App for Udoo.

Switch from *Android* to *Project* view.



Identify your '`app/libs`' folder, copy '`weka.jar`' to the folder (using Finder or Windows Explorer).

In AS, right click it and select '*Add as library*'.

Ensure that '`compile files('libs/weka.jar')`' is inside your '`app/build.gradle`' file.

Then, do a clean re-build of the entire app in *AS: Build -> Clean Project*.

3. Copy training dataset

Copy the training dataset '`iris_train.arff`' onto Udoo '`Download`' folder. You may use Android File Transfer or other tools (Windows Explorer may also work).

(You may want to copy '`iris_test.arff`' onto Udoo '`Download`' folder also. The test dataset will be used later.)

You may need to ensure following to access to Udoo file system:

1. Settings -> Develop Options:
    a. 'USB debugging' is selected
    b. 'External OTG port enabled' is selected

2. Drag down the notification, check 'Connected as a media device', choose 'Media device (MTP)'.

4. Open the training dataset file in Android

In '`onCreate`' (or a callback from button, up to your app design), open the training dataset file as a `BufferedReader`:

```
File root =
Environment.getExternalStoragePublicDirectory(Environment.DIRECTORY_DOWN
LOADS);

File f = new File(root, "iris_train.arff");

BufferedReader inputReader;

inputReader = readFile(f);
// you need to code the readFile() that takes in a File object and
returns a BufferedReader
```

We use '`Environment.getExternalStoragePublicDirectory`' to find '`Download`' folder on Udoo; do not hard-code that.

You also need to add access permission to your app. In "`app/src/main/AndroidManifest.xml`", add following:

```
<uses-permission android:name="android.permission.READ_EXTERNAL_STORAGE"/>
```

5. Build the KNN model

```
// inputReader is the BufferedReader that you obtained in the previous step

Instances data = new Instances(inputReader);
data.setClassIndex(data.numAttributes() - 1);


Classifier ibk = new IBk();
ibk.buildClassifier(data);
```

6. Load the test dataset, perform classification

```
f = new File(root, "iris_test.arff");
inputReader = readFile(f);

Instances test = new Instances(inputReader);
test.setClassIndex(test.numAttributes() - 1);

int correct = 0;
int incorrect = 0;
for (int i = 0; i < test.numInstances(); i++) {
    double pred = ibk.classifyInstance(test.instance(i));
    double act = test.instance(i).classValue();

    if (pred == act)
        correct ++;
    else
        incorrect++;
}
```

Check off: Display the classification results in your App to obtain a check-off.

Reference

ARFF file format: An ARFF (Attribute-Relation File Format) file is an ASCII text file that describes a list of instances sharing a set of attributes.

https://weka.wikispaces.com/ARFF+%28book+version%29