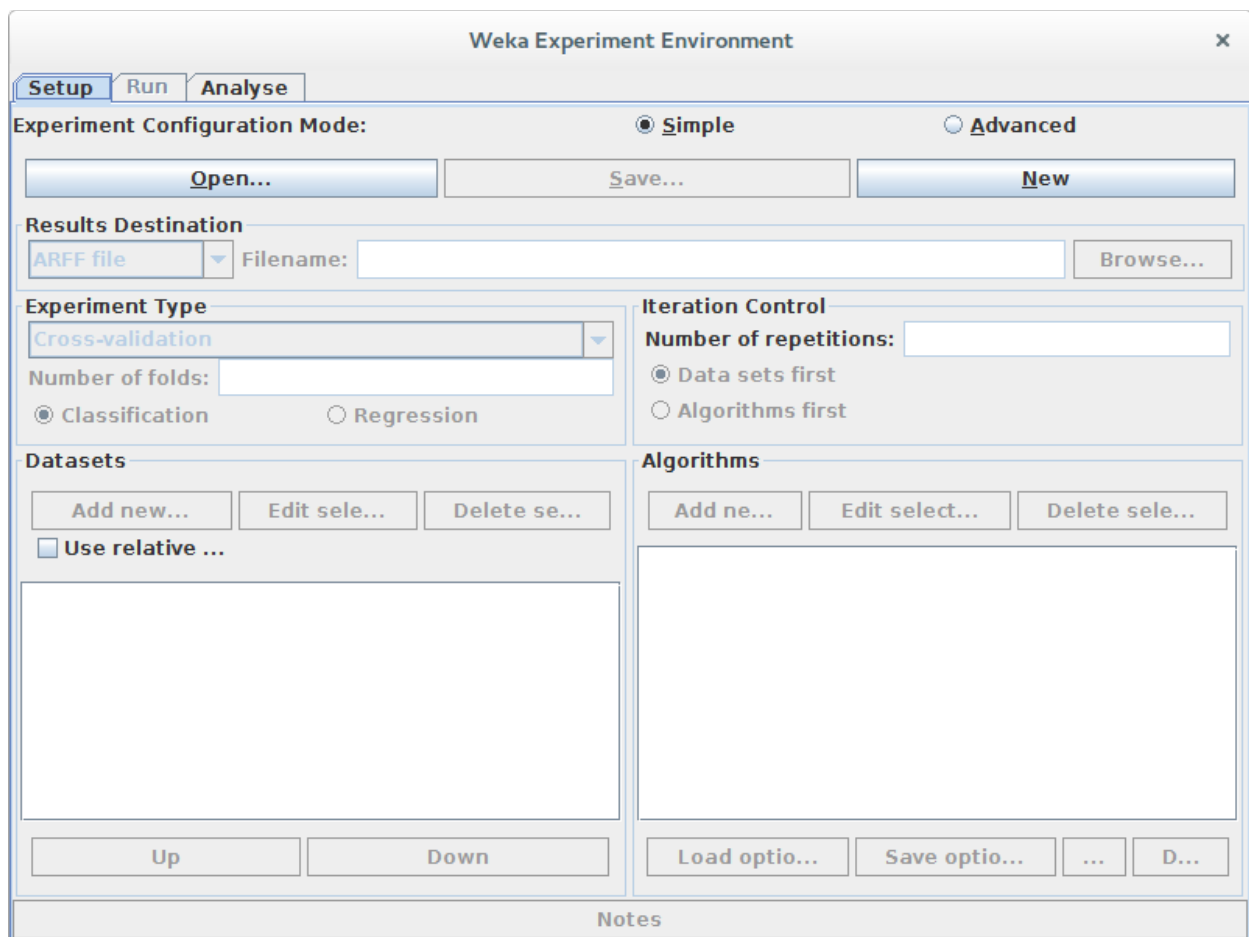**Step1:** Click the "*Experimenter*" button to launch the Weka Experimenter.

The Weka Experimenter allows you to design your own experiments of running algorithms on datasets, run the experiments and analyze the results. It's a powerful tool.

Click the "*New*" button to create a new experiment configuration.



**Test Options**

The experimenter configures the test options for you with sensible defaults. The experiment is configured to use Cross Validation with 10 folds. It is a "Classification" type problem and each algorithm + dataset combination is run 10 times (iteration control).

**Iris flower Dataset**

Let's start out by selecting the dataset.

1. In the "*Datasets*" select click the "Add new…" button.
2. Open the "*data*"directory and choose the "*iris.arff*" dataset.

The Iris flower dataset is a famous dataset from statistics and is heavily borrowed by researchers in machine learning. It contains 150 instances (rows) and 4 attributes (columns) and a class attribute for the species of iris flower (one of setosa, versicolor, virginica).

Let's choose 3 algorithms to run our dataset.

**ZeroR**

1. Click "***Add new…***" in the "Algorithms" section.
2. Click the "***Choose***" button.
3. Click "***ZeroR***" under the "***rules***" selection.

ZeroR is the simplest algorithm we can run. It picks the class value that is the majority in the dataset and gives that for all predictions. Given that all three class values have an equal share (50 instances), it picks the first class value "setosa" and gives that as the answer for all predictions. Just off the top of our head, we know that the best result ZeroR can give is 33.33% (50/150). This is good to have as a baseline that we demand algorithms to outperform.

**OneR**

1. Click "***Add new…***" in the "Algorithms" section.
2. Click the "***Choose***" button.
3. Click "***OneR***" under the "***rules***" selection.

OneR is like our second simplest algorithm. It picks one attribute that best correlates with the class value and splits it up to get the best prediction accuracy it can. Like the ZeroR algorithm, the algorithm is so simple that you could implement it by hand and we would expect that more sophisticated algorithms out perform it.

**J48**

1. Click "***Add new…***" in the "Algorithms" section.
2. Click the "***Choose***" button.
3. Click "***J48***" under the "***trees***" selection.

J48 is decision tree algorithm. It is an implementation of the C4.8 algorithm in Java ("J" for Java and 48 for C4.8). The C4.8 algorithm is a minor extension to the famous C4.5 algorithm and is a very powerful prediction algorithm.
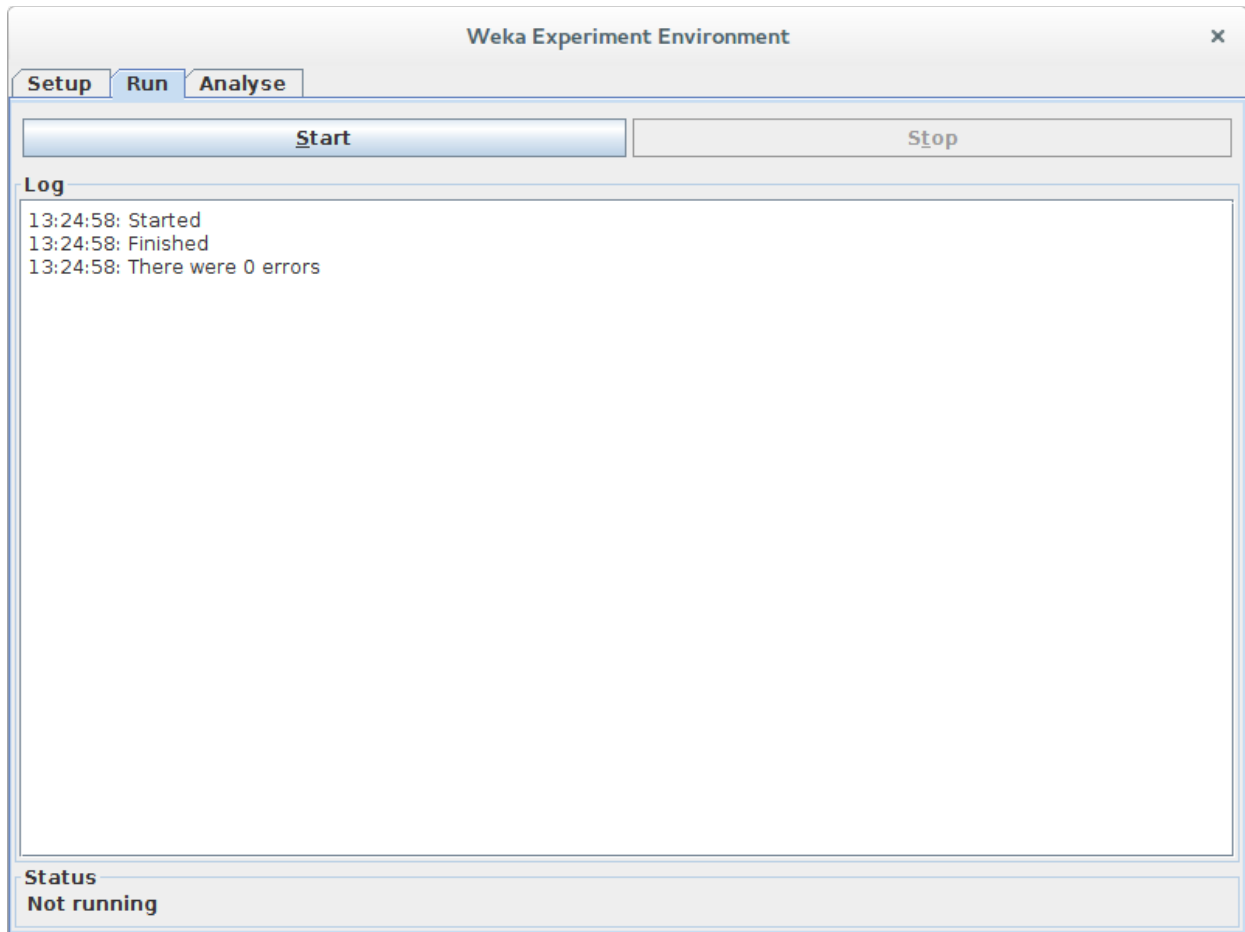
We are ready to run our experiment.

**Step2. Run Experiment**

Click the "***Run***" tab at the top of the screen.

This tab is the control panel for running the currently configured experiment.

Click the big "***Start***" button to start the experiment and watch the "Log" and "Status" sections to keep an eye on how it is doing.



Given that the dataset is small and the algorithms are fast, the experiment should complete in seconds.

**Step3: Review Results**

Click the "*Analyse*" tab at the top of the screen. Click the "**Experiment**" button in the "**Source**" section to load the results from the current experiment.
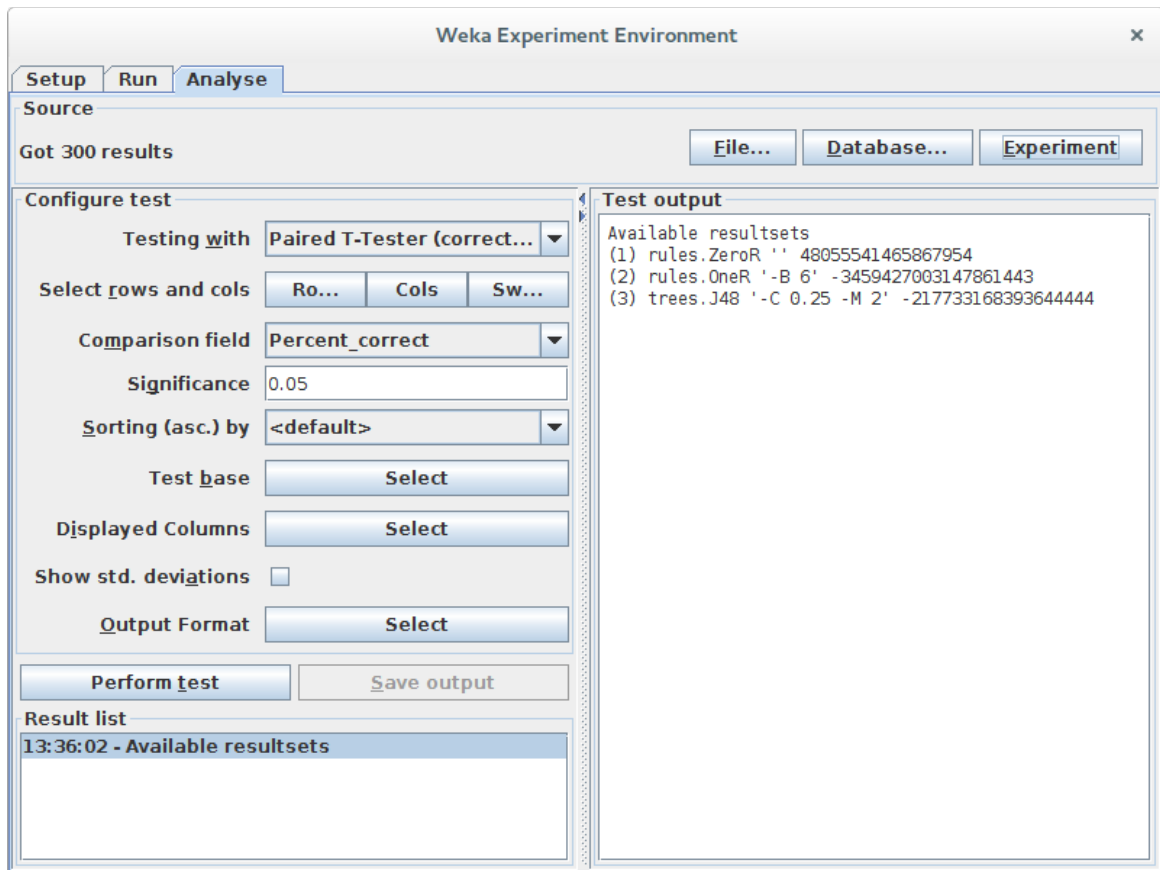


**Algorithm Rank**

The first thing we want to know is which algorithm was the best. We can do that by ranking the algorithms by the number of times a given algorithm beat the other algorithms.

1. Click the "*Select*" button for the "*Test base*" and choose "*Ranking*".
2. Now Click the "*Perform test*" button.
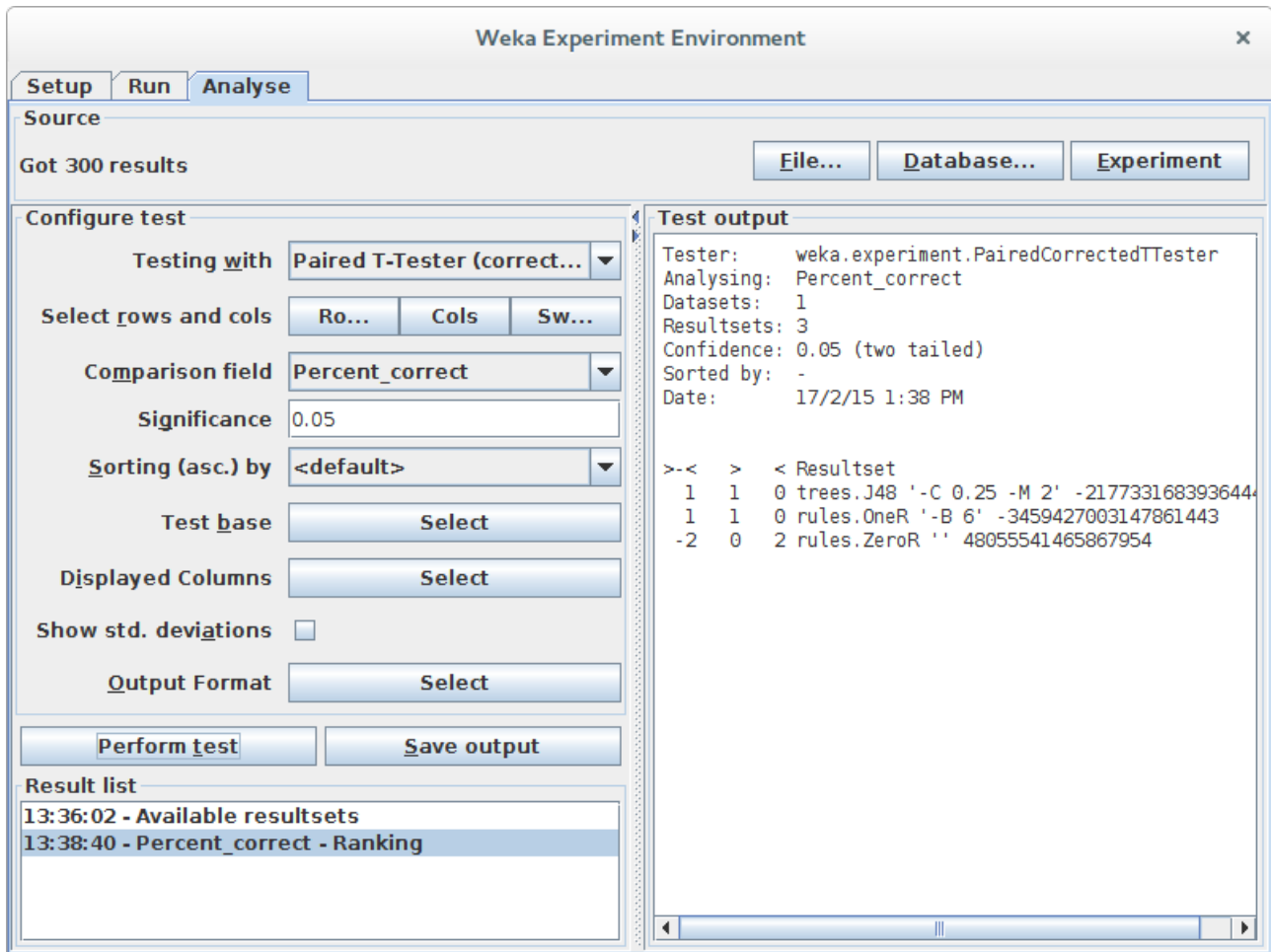
The ranking table shows the number of statistically significant wins each algorithm has had against all other algorithms on the dataset. A win, means an accuracy that is better than the accuracy of another algorithm and that the difference was statistically significant.

We can see that both J48 and OneR have one win each and that ZeroR has two losses. This is good, it means that OneR and J48 are both potentially contenders outperforming out baseline of ZeroR.

**Algorithm Accuracy**

Next we want to know what scores the algorithms achieved.

1. Click the "*Select*" button for the "*Test base*" and choose the "*ZeroR*" algorithm in the list and click the "*Select*" button.
2. Click the check-box next to "*Show std. deviations*".
3. Now click the "*Perform test*" button.
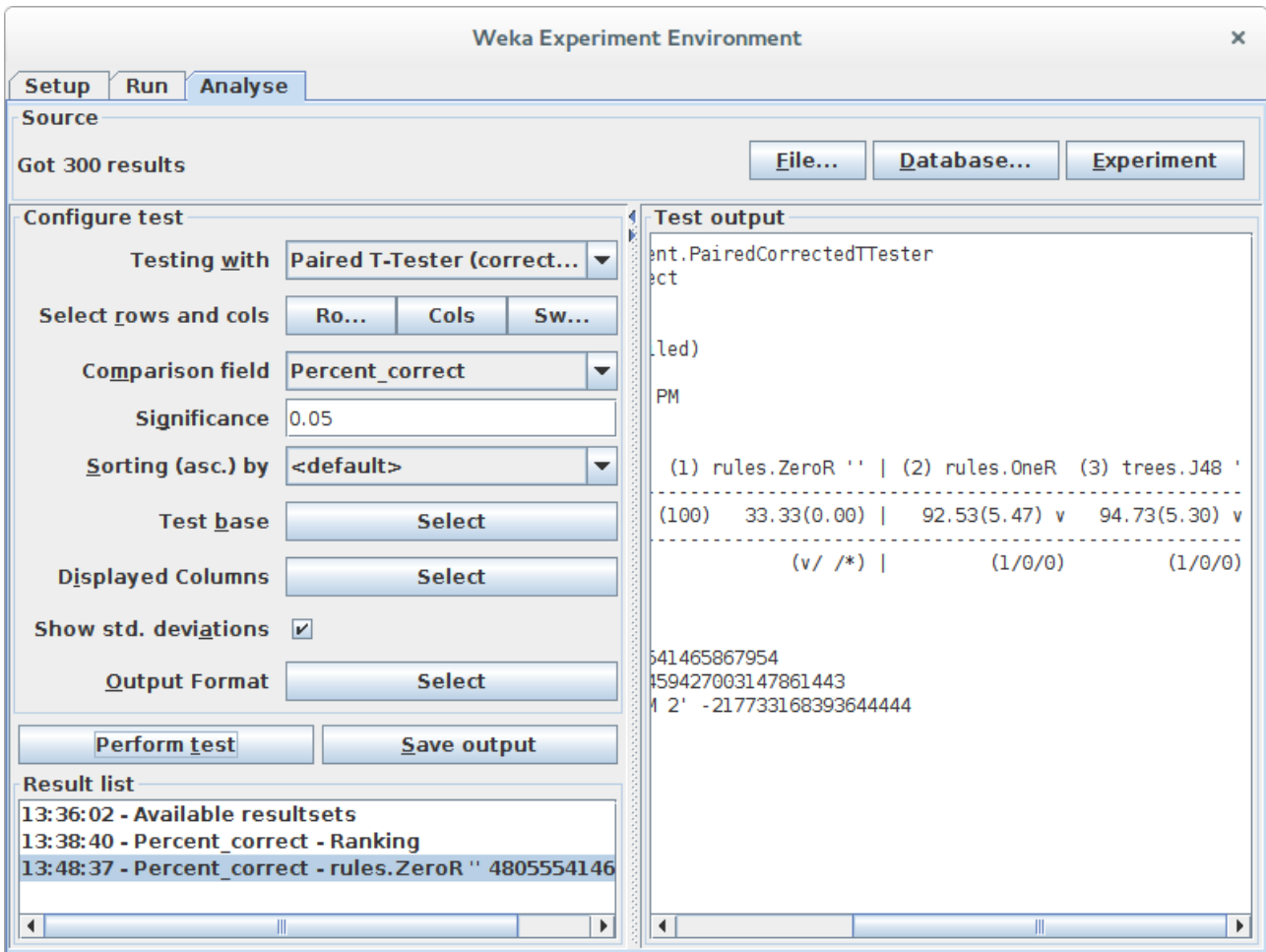
In the "Test output" we can see a table with the results for 3 algorithms. Each algorithm was run 10 times on the dataset and the accuracy reported is the mean and the standard deviation in rackets of those 10 runs.

We can see that both the OneR and J48 algorithms have a little "v" next to their results. This means that the difference in the accuracy for these algorithms compared to ZeroR is statistically significant. We can also see that the accuracy for these algorithms compared to ZeroR is high, so we can say that these two algorithms achieved a statistically significantly better result than the ZeroR baseline.

The score for J48 is higher than the score for OneR, so next we want to see if the difference between these two accuracy scores is significant.

1. Click the "*Select*" button for the "*Test base*" and choose the "*J48*" algorithm in the list and click the "*Select*" button.
2. Now click the "*Perform test*" button.
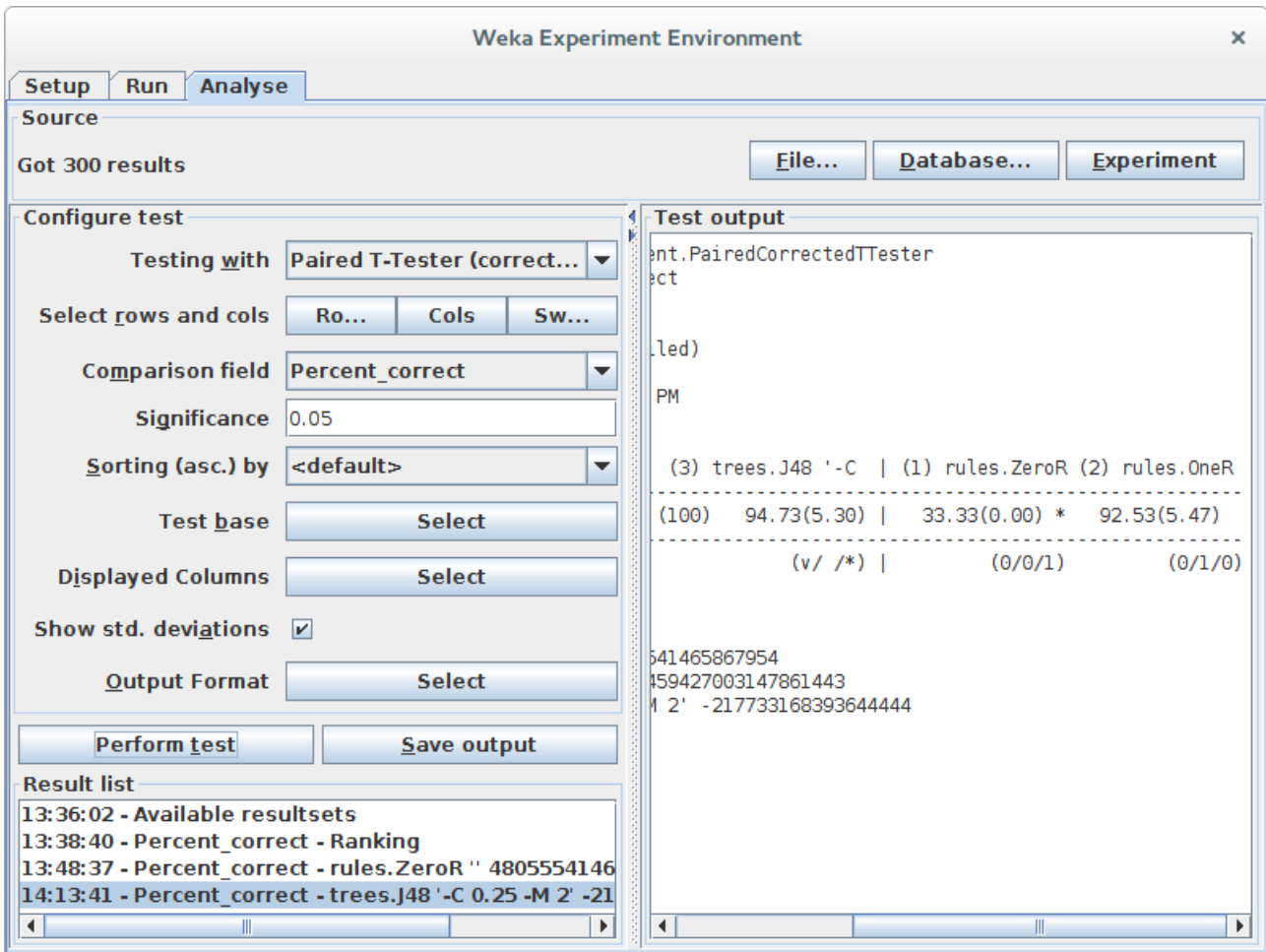
We can see that the ZeroR has a "*" next to its results, indicating that its results compared to the J48 are statistically different. But we already knew this. We do not see a "*" next to the results for the OneR algorithm. This tells us that although the mean accuracy between J48 and OneR is different, the differences is not statistically significant.

All things being equal, we would choose the OneR algorithm to make predictions on this problem because it is the simpler of the two algorithms.

If we wanted to report the results, we would say that the OneR algorithm achieved a classification accuracy of 92.53% (+/- 5.47%) which is statistically significantly better than ZeroR at 33.33% (+/- 5.47%).