Activity: Exercises with Classifiers using Orange, Weka, and RSES

This is a preview of the published version of the quiz

Started: May 21 at 8:14pm

Quiz Instructions

Exercises with Classifiers

Using Orange, Weka, and RSES

In our last activity, we got some hands-on experience with Orange and Weka. We will not get some more practice with both generating and modeling various other classifiers. We will also introduce the RSES tool.

A reminder on:

The Data Science Pipeline

In the previous activity, we got some experience with the overall **data science pipeline,** and will gain additional experience with this activity. Through this "pipeline," we are entering data with the goal of obtaining insights. The steps of this pipeline include:

1. Obtaining our data
2. Scrubbing or cleaning our data (or "data preprocessing")
3. Exploring, visualizing, and gaining a better understanding of our data (e.g., finding patterns)
4. Modeling our data (e.g., creating decision trees or finding classification rules)
5. Interpreting our data

Part 1

Comparing Different Models in Orange

In our last activity, we got a first look at the overall data science pipeline and got some practice with generating decision trees and classification rules. When generating models, it is also important to **evaluate how well those models do.**

How to Evaluate Models

Some ways to evaluate models is through accuracy, precision, and recall. **Accuracy** is *how good a model is at predicting the correct category.* This is good to use if our data set is balanced, but that's not always the case.

$$Accuracy = \frac{True\ Positives + True\ Negatives}{All\ samples}$$

Let's say we have a dataset to predict or classify the presence of heart disease in a patient. Let's say our model correctly predicted 45 patients as having heart disease, and correctly predicted 30 patients as not having heart disease. It also had incorrect predictions for 35 patients - 15 of those the model predicted as having heart disease

when they really didn't, and the remaining 20 were predicted to not have heart disease when they really did.

		Actual values	
		Positives	Negatives
Predicted values	Positives	45	15
	Negatives	20	35

In this table above, the columns are the actual values while the rows are the predicted values. We see that the 45 are our "true positives," the 35 are our "true negatives," the 20 are our "false negatives" and the 15 are our "false positives."

To calculate accuracy, we would have:

$$Accuracy = \frac{45 + 35}{45 + 15 + 20 + 35} = \frac{80}{115} = 0.6957$$

Other ways to evaluate a model's performance is precision and recall. **Precision** is the ratio of what our model predicted correctly to *what our model predicted*. **Recall**, on the other hand, is the ratio of what our model predicted correctly to *what the actual labels are*.

$$Precision = \frac{True\ Positives}{True\ Positives + False\ Positives\ (or,\ the\ total\ predicted\ positive}$$

$$Recall = \frac{True\ Positives}{True\ Positive + False\ Negatives\ (or,\ total\ actual\ positives)}$$

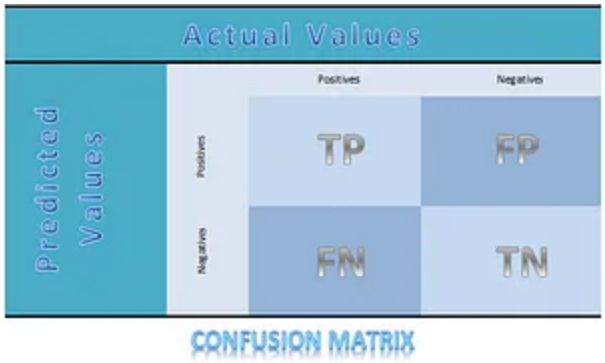
The **F1-Score** is a method which combines both precision and recall. It is the precision and recall averaged into a single metric:

$$F1\ Score = \frac{2 \cdot (Precision \cdot Recall)}{Precision + Recall}$$

Additional reading: [https://medium.com/swlh/explaining-accuracy-precision-recall-and-f1-score-f29d370caaa8 (https://medium.com/swlh/explaining-accuracy-precision-recall-and-f1-score-f29d370caaa8)](https://medium.com/swlh/explaining-accuracy-precision-recall-and-f1-score-f29d370caaa8)

Another way to evaluate our models is through a **confusion matrix**, "one of the most powerful analytical tools in machine learning and data science." This is a tool which displays and compares the values a model predicted versus the actual values they should have been. It

looks like what you see below (or the table we used earlier to show heart disease predictions!).



Resource: https://www.unite.ai/what-is-a-confusion-matrix/ [(https://www.unite.ai/what-is-a-confusion-matrix/)](https://www.unite.ai/what-is-a-confusion-matrix/)

Question 1 **0 pts**

Now that we have discussed the different parts of a confusion matrix (true positive, true negative, false positive, and false negative), let's say we are trying to predict the presence of heart disease in a patient - which one may be more dangerous, false positives or false negatives? Why? (Discuss this with your groups also!)

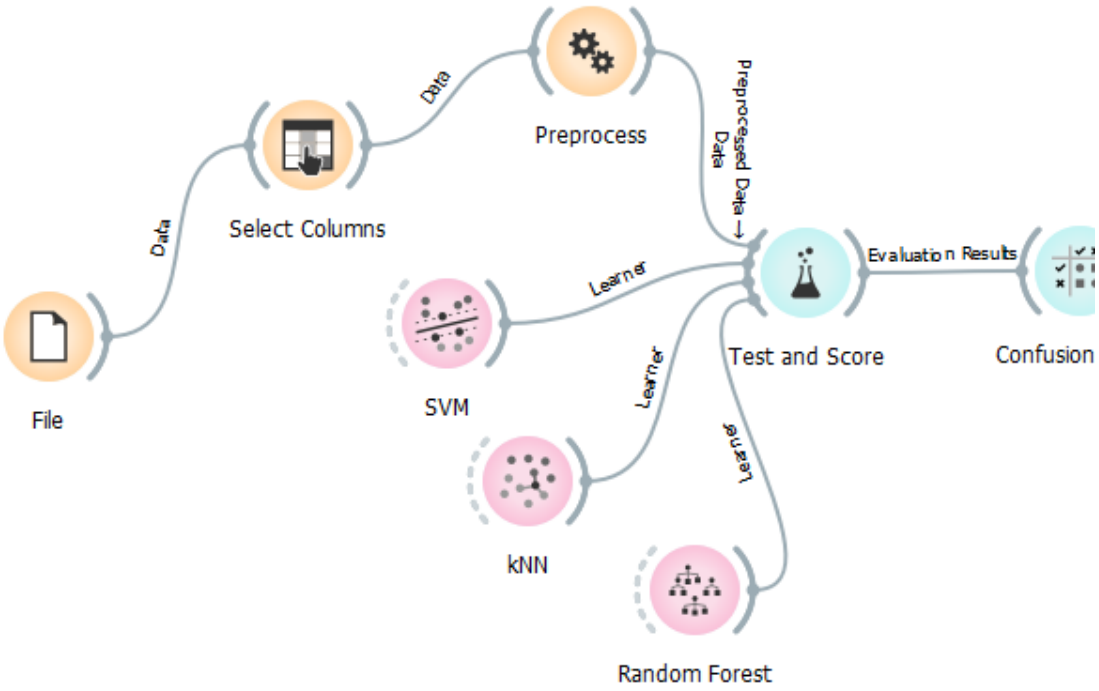
Edit View Insert Format Tools Table

12pt Paragraph

p 0 words

For this activity, we will use the data on heart disease here: [https://www.kaggle.com/nareshbhat/health-care-data-set-on-heart-attack-possibility (https://www.kaggle.com/nareshbhat/health-care-data-set-on-heart-attack-possibility)](https://www.kaggle.com/nareshbhat/health-care-data-set-on-heart-attack-possibility)

Once you have downloaded the data, create the following pipeline in Orange:



We discussed many of these widgets in the last activity: **File** (to load in our data), **Select Columns** (to select our features and target), and **Preprocess** (to, for example, remove null values). Now, we will use the **Test and Score** widget to evaluate multiple types of models. Here, we will experiment with the **SVM**, **kNN**, and **Random Forest** models to compare them. We will also look at a final **Confusion Matrix** for the different models.

Once you have experimented with the different models and taken a look at the results, answer the following questions.

Question 2 — 0 pts

Under the **Test and Score** widget, take a look at the accuracy, precision, and recall when "Cross Validation" is selected as our sampling method. Then take a look at "Random sampling" and "Test on train data."

Note that when evaluating models, we usually split the dataset into a "train" and "test" set. We train the model on the first "train" subset of data, and then we test on the "test" subset (meaning, we take the model and then make predictions on the remaining data to see if they are correct or not).

Out of these different sampling methods, which has the highest performance? Why might that be so?

Edit View Insert Format Tools Table

12pt Paragraph

p 0 words

The "Test on train data" should have been highest!

This is because in this sampling method, we are not splitting between a train and test subset - we are simply training *and* testing on the same sample of data. Therefore, our model should have learned the existing patterns fairly well (hence the high accuracy). However, this may be a problem for trying to generalize our model to new data - it is possible for our model to have gotten very good at making predictions with our current data, but has "overfitted" or is now very specified to just the patterns of the initial dataset.

This is why we usually look at sampling methods such as cross validation or random sampling when trying to evaluate our model, as it is a more accurate representation of the model's performance to *new data.*

Question 3 **0 pts**

Looking through both the results in the **Test and Score** and **Confusion Matrix** widgets, which model had the best performance? How did you tell which had the best performance?

Edit View Insert Format Tools Table

12pt Paragraph

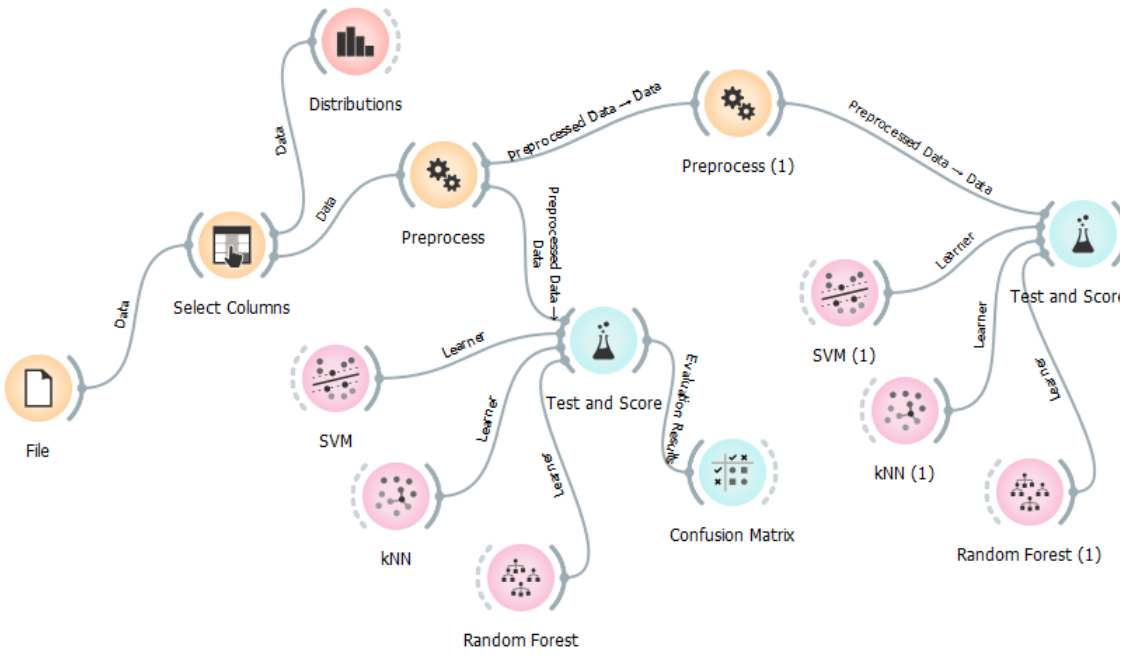
p 0 words

Question 4 0 pts

Oftentimes with Data Mining, it is more of an **iterative process** of discovery, evaluating models, exploring new ways to improve the models, and then evaluating them again.

Now that we've looked at our first set of models and results, let's try adding some more to our preprocessing step!

Instead of simply editing our original preprocessing step, we want a way to easily compare our new results to our original ones, so let's set up our pipeline to look something like this:



(Note, the added **Distributions** widget at the beginning was just to take a further look at the data.)

We will add that *second Preprocess* widget, and then copy and paste our various classifiers with the test and score.

In our second preprocessing widget, let's add a step to **Normalize** our data. Click on the **Normalize Features** under Preprocess. Let's leave the default selection, and click Apply. Then compare the results from our models with the new preprocessing step added to that of our original ones. What differences do you see?

Edit View Insert Format Tools Table

12pt Paragraph

p 0 words

Question 5 **0 pts**

Part 2

Comparing Different Models in Weka

Now we will do the same in Weka!

First of all, we need to add some of our algorithms to Weka to use.

1. Click **Tools** > **Package Manager**
2. Scroll down to find **LibLINEAR,** then click **Install** (this will give us our algorithm for the SVM model)

The others for Random Forest and Knn are already included.

For this one, we will use the banking data here: **https://www.kaggle.com/kidoen/bank-customers-data (https://www.kaggle.com/kidoen/bank-customers-data)**

In Weka, we would have needed to discretize our data if continuing with the heart dataset. It would have been good to do this in Orange as well, but the software was able to automatically discretize for us, so it wasn't a problem.

Once again, we will click to use the **Explorer.**

Select **Open file** to load in the data. Don't forget to also change the File type to .csv so you can view your data.

Under the **Classify** tab, we will now select our classifiers. We will go through the following types of classifiers, one at a time:

1. KNN: Select lazy > IBk
2. Random Forest: Select trees > RandomForest
3. SVM: Select functions > LibLINEAR

Once you select one classifier, also select "Percentage split" under our **Test options.** This will help save us some time as compared to using Cross-validation (for the sake of the activity). Also check that "(Nom) term_deposit" is selected as our target. Then click "Start." Once it is running, we should see information in the **Status** at the bottom, such as "Building model on training data..." Once it has finished running, take a look at the **Classifier output.** Here, we can view a summary, including the accuracy ("Correctly Classified Instances"), and more detailed information, such as our Precision, Recall, and F-Measure.

Now run this process on each of the classifiers listed. Don't worry about memorizing the results, as we will be able to click back to view them again at any time.

Once you have run through all three, take a look at all the results. Which one seemed to perform the best and why?

Edit View Insert Format Tools Table

12pt Paragraph

p 0 words

Question 6 **0 pts**

Part 3

Comparing Different Models in RSES

Dr. Ras has provided the download for RSES under our class Google Drive folder here: **https://drive.google.com/drive/folders/1mQV6XmiQLCKcOmpmLoy485IMAiW32OeD (https://drive.google.com/drive/folders/1mQV6XmiQLCKcOmpmLoy485IMAiW32OeD)**

Click RSES2_2_inst.exe and click download. Once it has downloaded, open it to install.

Also, note that this requires Java JDK or JRE to work, so if you don't have that already, you can download it here: **https://www.oracle.com/java/technologies/javase-downloads.html (https://www.oracle.com/java/technologies/javase-downloads.html)** Once on this page, click the **JDK download** and then select the proper download for your system.

For this one, we will simply work with kNN to get an initial feel for the tool.

Click **New Project** (the paper icon in the left toolbar). Then click **Insert Table.** Instead of loading in our own data, we will explore one of the already provided datasets. Right click the icon for Insert table and then click **Load.** We should already be in the folder for RSES2, so now select the "DATA" folder, and click "heart_disease.tab" (this should look very familiar to the heart dataset we used earlier). Now that it is loaded in, we should also be able to double click the Insert Table icon to view the table of data.

To start generating our kNN model, right click our icon for Insert Table. Then select Classify > Test table using k-NN. Let's leave all the default values for now, and click OK. Once it has finished, we can double click the new icon to view our results. How do they look?

Edit View Insert Format Tools Table

12pt Paragraph

p 0 words

Were you skeptical at the 100% accuracy?

We didn't split our dataset at all! We can right click our table icon again, and then select Classify > Cross-validation method to add this. Again, let's leave the default values and click OK. Once that has completed, take another look at your confusion matrix.

Question 7 **0 pts**

Reflect on your learning

Briefly reflect on what you learned today (evaluating and comparing different algorithms/models). Think about questions like how this could help you with your future goals, anything specific that was interesting (any "aha!" moments?), or perhaps fuzzy areas you may need to review again.

Reflection is an important part of learning as it is the step that helps us bring together our experiences to make sense of and grow from them.

Also, please let me know if there were any parts of this activity that were confusing or could be improved upon! :)

Edit View Insert Format Tools Table

12pt Paragraph

p 0 words

Not saved Submit Quiz