**Decision Tree Quiz**

In this quiz, you'll be given the following sample dataset, and your goal is to define a model that gives 100% accuracy on it.

Chart, scatter chart

Description automatically generated

The data file can be found under the "data.csv" tab in the quiz below. It includes three columns, the first 2 comprising of the coordinates of the points, and the third one of the label.

The data will be loaded for you, and split into features X and labels y.

**You'll need to complete each of the following steps:**

**1. Build a decision tree model**

* Create a decision tree classification model using scikit-learn's [DecisionTreeClassifier](http://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeClassifier.html" \t "_blank) and assign it to the variablemodel.

**2. Fit the model to the data**

* You won't need to specify any of the hyperparameters, since the default ones will yield a model that perfectly classifies the training data. However, we encourage you to play with hyperparameters such as max\_depth and min\_samples\_leaf to try to find the *simplest possible* model.

**3. Predict using the model**

* Predict the labels for the training set, and assign this list to the variable y\_pred.

**4. Calculate the accuracy of the model**

* For this, use the function sklearn function [accuracy\_score](http://scikit-learn.org/stable/modules/generated/sklearn.metrics.accuracy_score.html" \t "_blank). A model's **accuracy** is the fraction of all data points that it correctly classified.

When you hit **Test Run**, you'll be able to see the boundary region of your model, which will help you tune the correct parameters, in case you need them.

***Note:*** This quiz requires you to find an accuracy of 100% on the training set. This is like memorizing the training data! A model designed to have 100% accuracy on training data is unlikely to generalize well to new data. If you pick very large values for your parameters, the model will fit the training set very well, but may not generalize well. Try to find the smallest possible parameters that do the job—then the model will be more likely to generalize well.