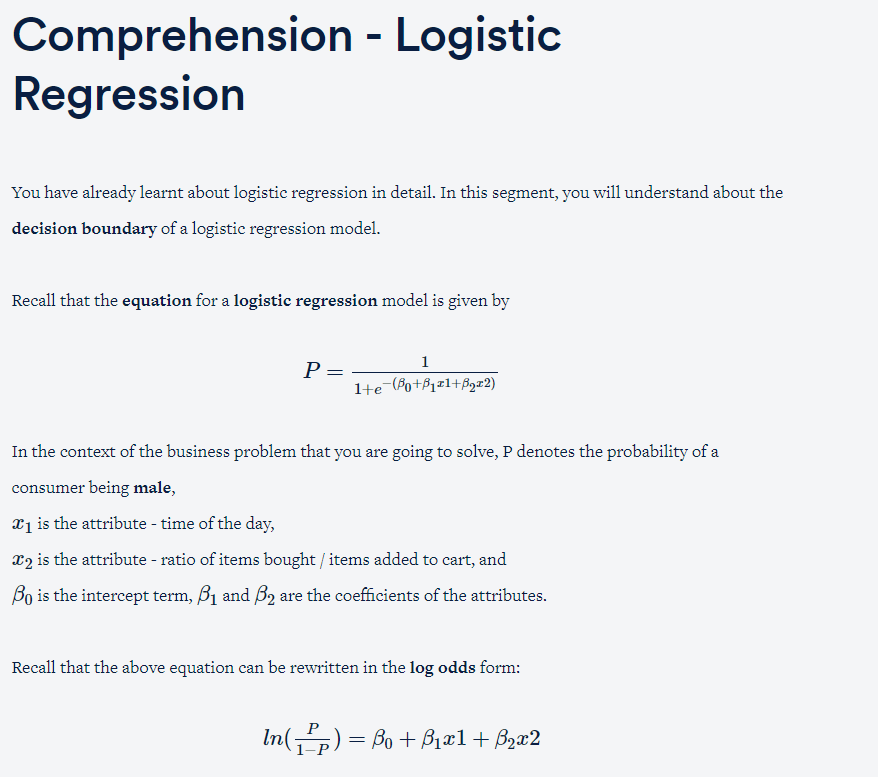
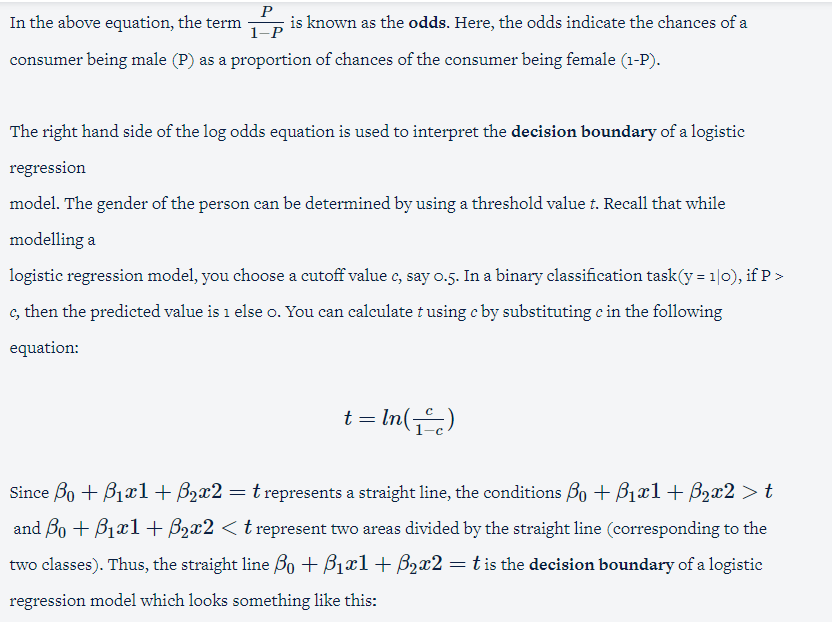
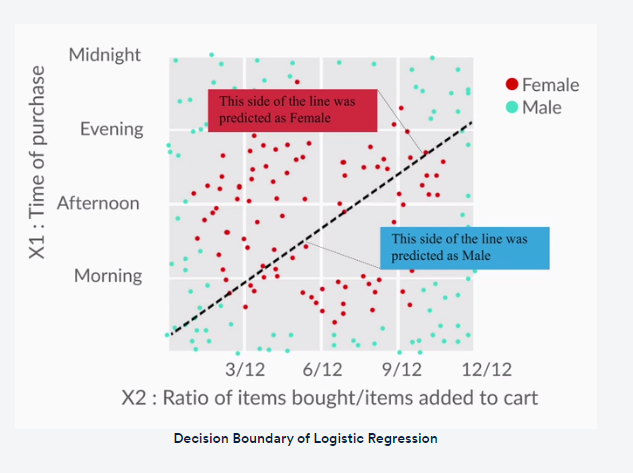
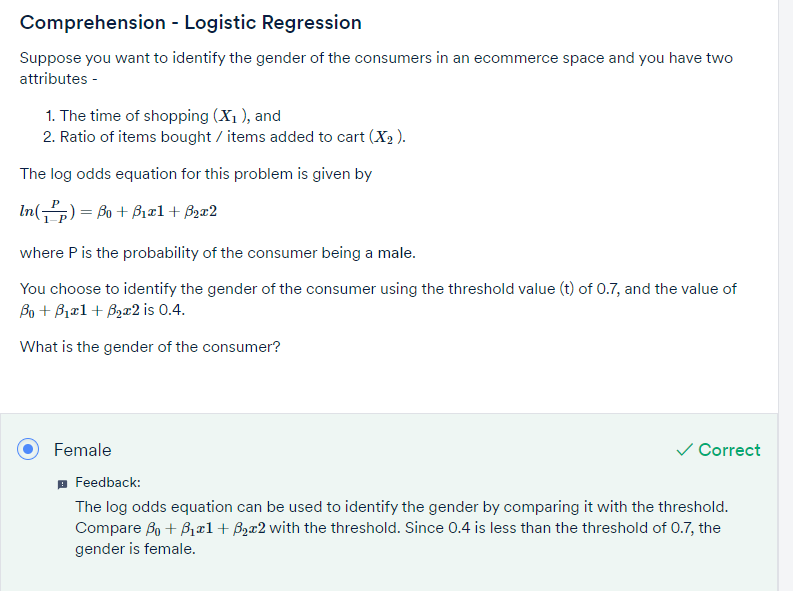
In this session, you will learn about the following:

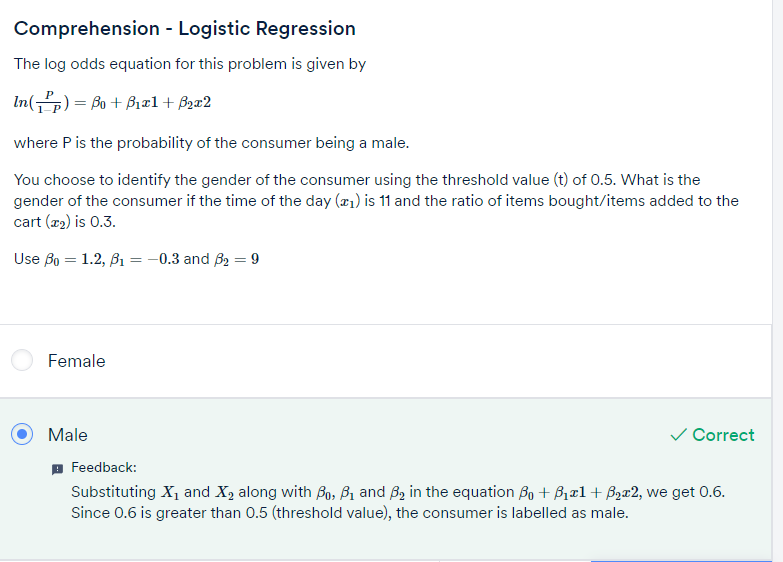
* Building and comparing different machine learning models
* Pros and Cons of different machine learning models
* CART and CHAID trees

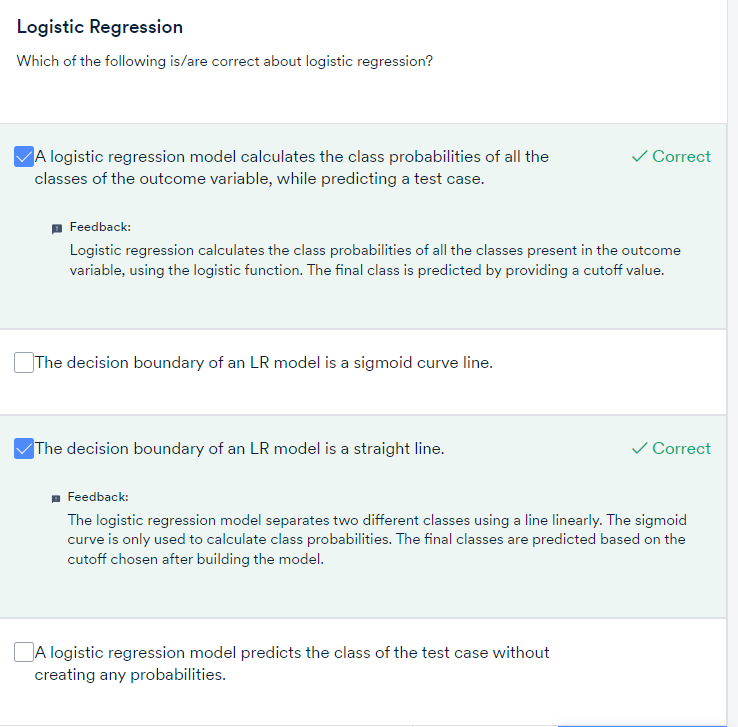




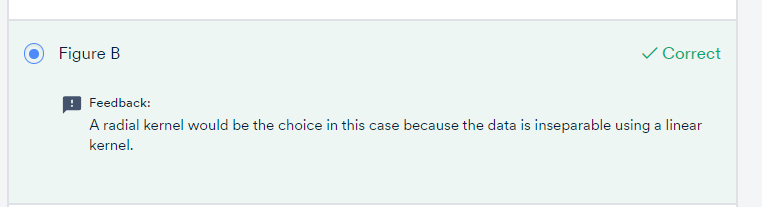






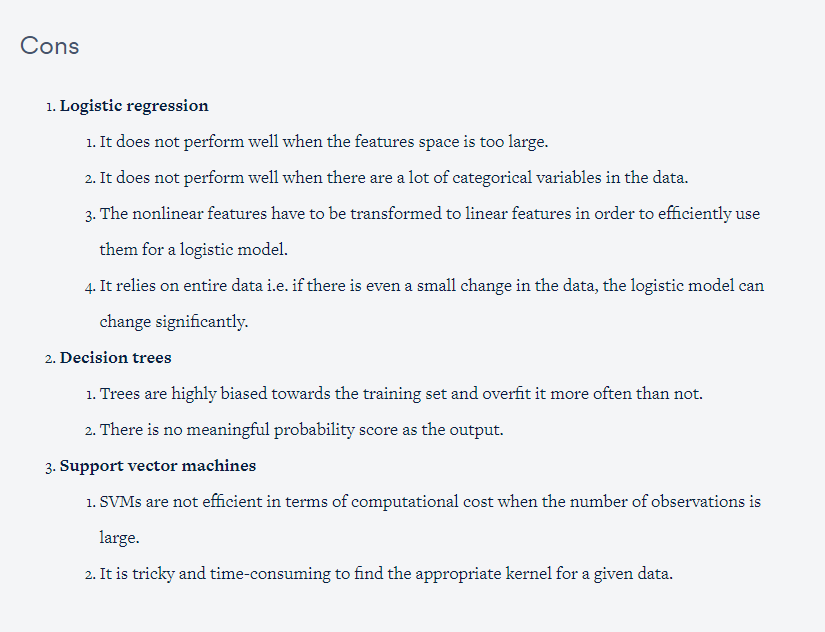


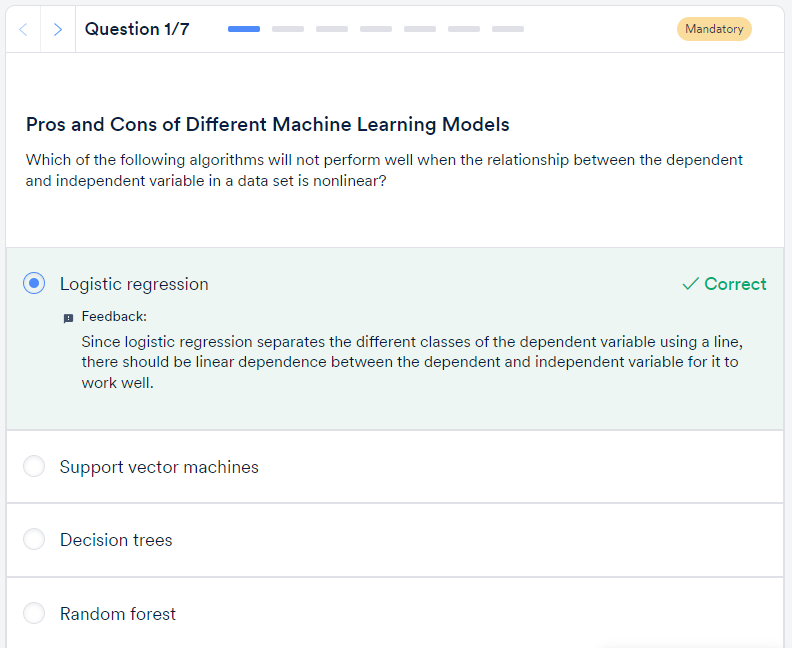


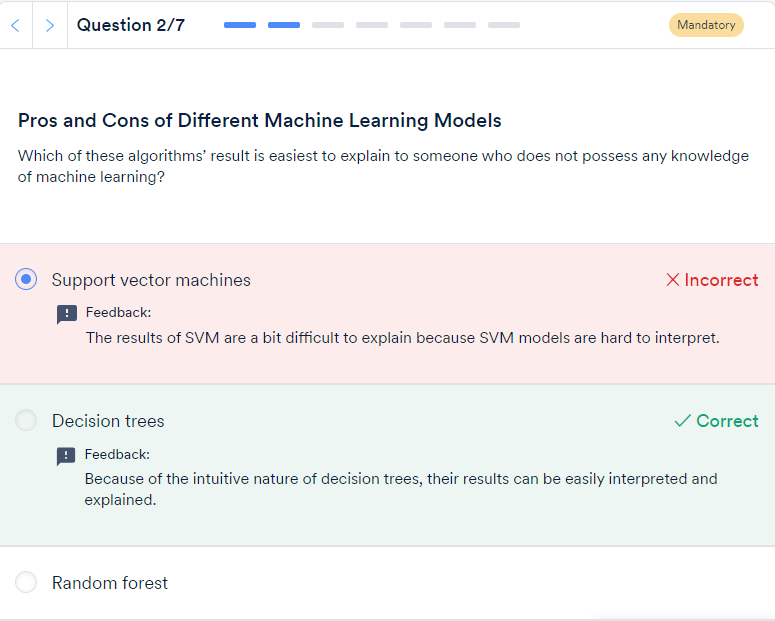


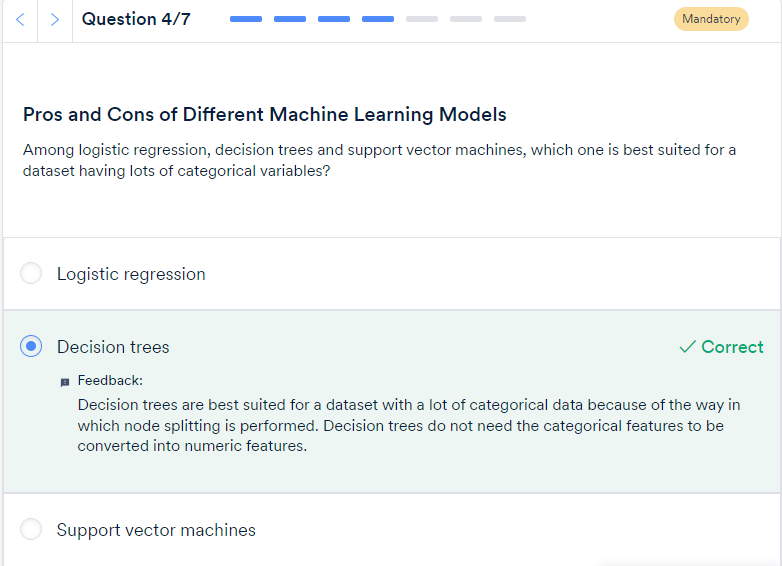
Pros

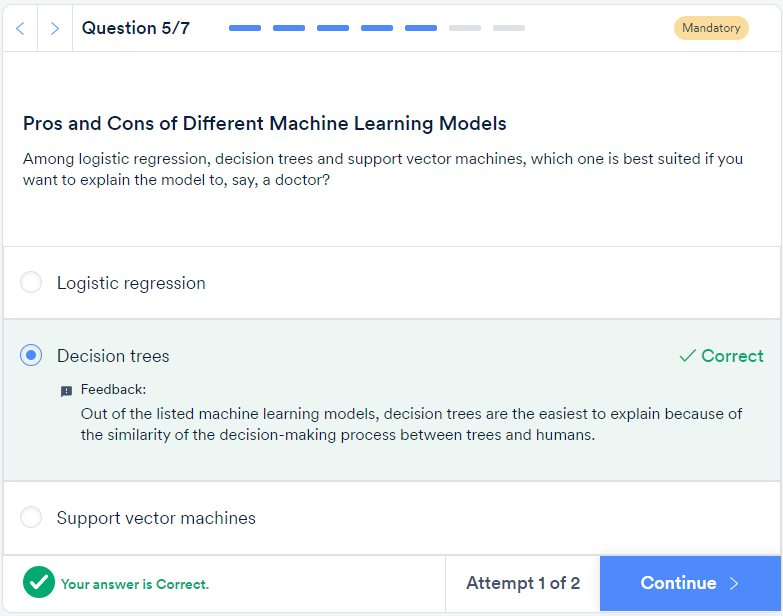
1. **Logistic regression**
   1. It is convenient for generating probability scores.
   2. Efficient implementation is available across different tools.
   3. The issue of multicollinearity can be countered with regularisation.
   4. It has widespread industry use.
2. **Decision trees**
   1. Intuitive decision rules make it easy to interpret.
   2. Trees handle nonlinear features well.
   3. The variable interaction is taken into account.
3. **Support vector machines**
   1. SVMs can handle large feature space.
   2. These can handle nonlinear feature interaction.
   3. They do not rely on the entire dimensionality of the data for the transformation.

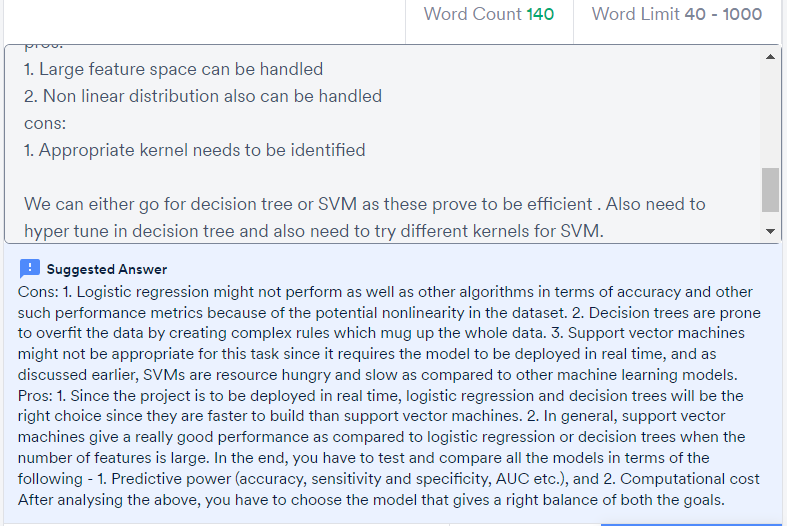


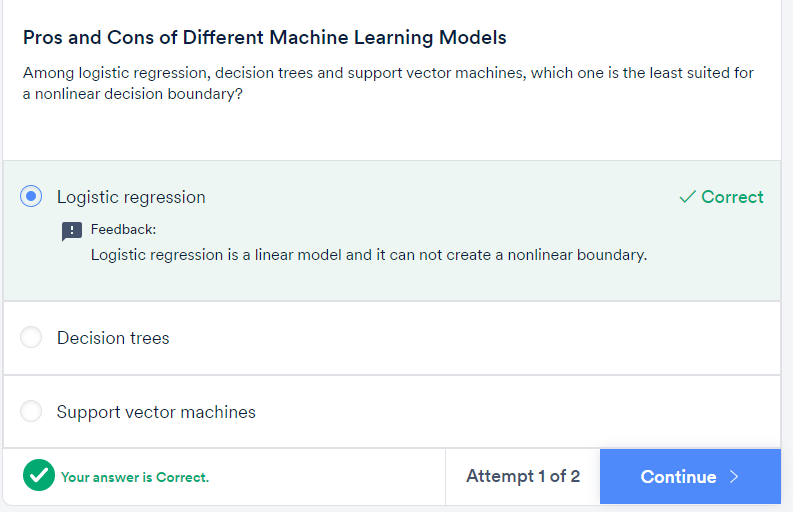












You could get overwhelmed by the choice of algorithms available for classification. To summarise—

1. **Start with logistic regression**. Using a logistic regression model serves two purposes: 1) It acts as a **baseline** (benchmark) model. 2) It gives you an idea about the important variables.
2. Then, go for **decision trees** and compare their performance with the logistic regression model. If there is no significant improvement in their performance, then just use the important variables drawn from the logistic regression model.

Finally, if you still do not meet the performance requirements, use **support vector machines**. But, keep in mind the **time and resource constraints**, because it takes time to find an appropriate kernel for SVM. Also, they are computationally expensive.

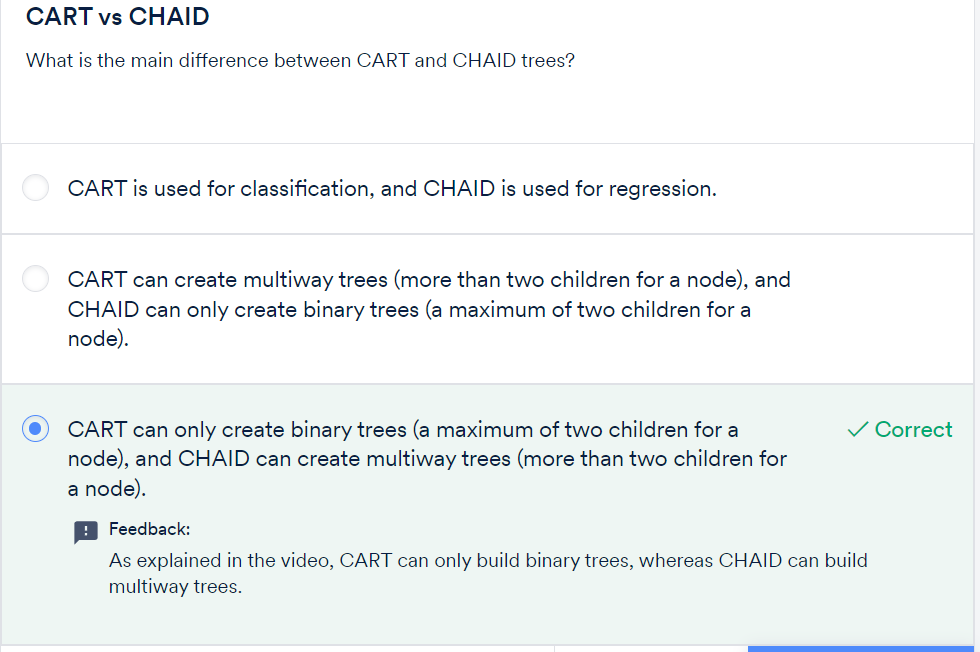
# CART and CHAID Trees:

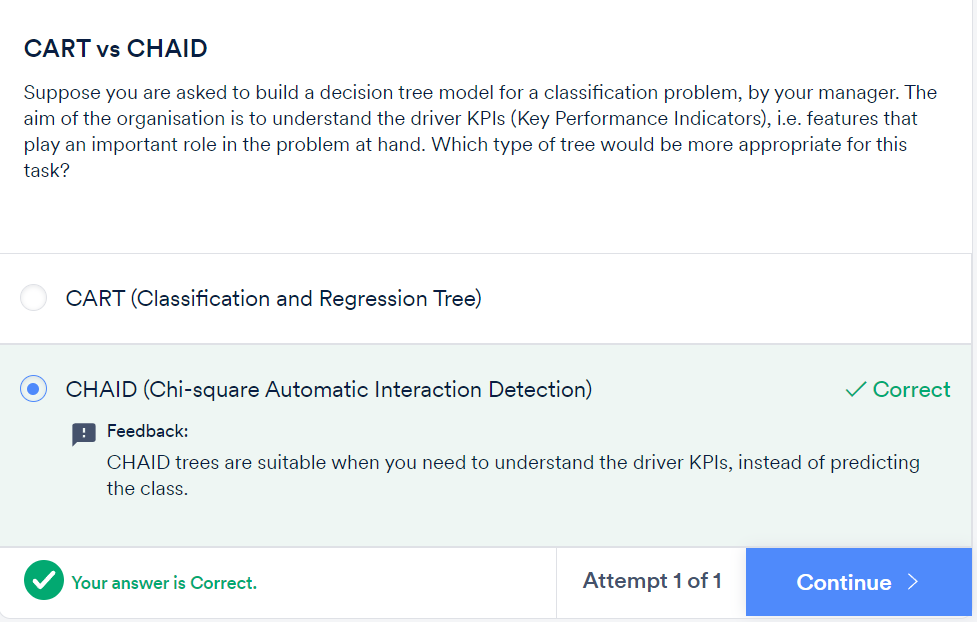
**CART (Classification and Regression Trees)**.

**CHAID (Chi-square Automatic Interaction Detection)**

You are already familiar with **CART**, which creates a **binary tree-**a tree with a maximum of two child nodes for any node in the tree. Sometimes CART is not appropriate to visualise the important features in a dataset because binary trees tend to be much **deeper** and more **complex** than a **non-binary tree-** a tree which can have more than two child nodes for any node in the tree. This is where **CHAID** comes in. CHAID can create non-binary trees which tend to be shallower than the binary trees. This makes CHAID trees easier to look at and understand the important drivers (features) in a business problem. The process of finding out important features is also referred to as **driver analysis**.

You looked at the different applications of CART and CHAID trees. To put them in the form of an analogy, suppose you are working with the Indian cricket team, and you want to **predict** whether the team will win a particular tournament or not. In this case, **CART** would be more preferable because it is more suitable for prediction tasks. Whereas, if you want to look at the **factors** that are going to influence the win/loss of the team, then a **CHAID** tree would be more preferable.



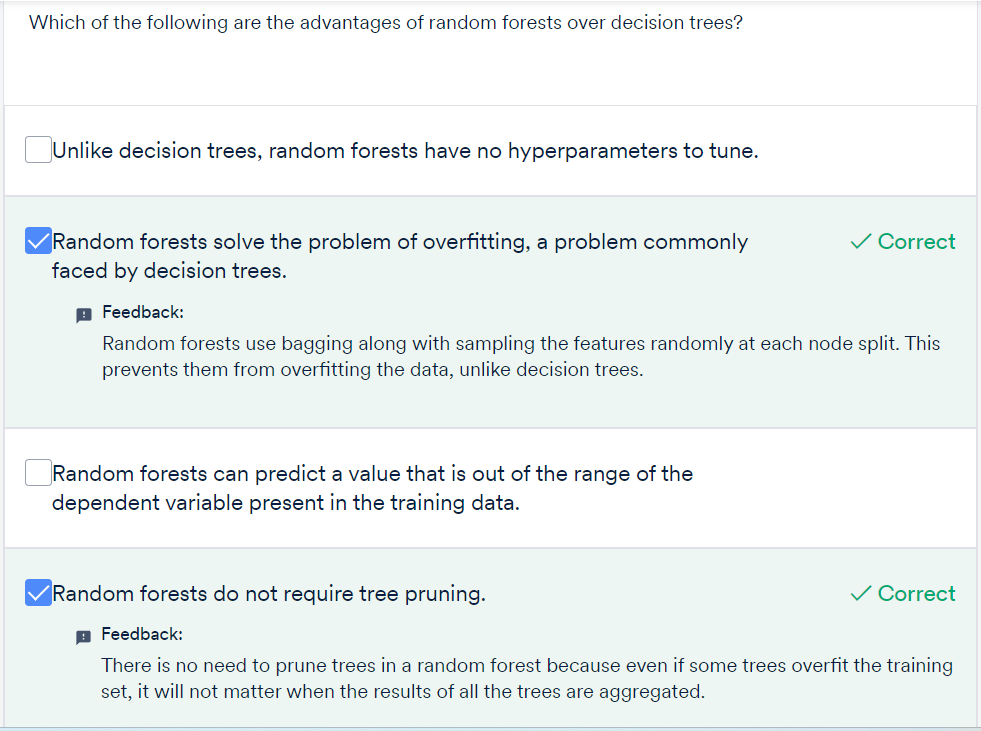


**Disadvantages of decision trees:**

1. Trees have a tendency to **overfit** the training data.
2. Splitting with **multiple linear decision boundaries that are perpendicular to the feature space** is not always efficient.
3. It is not possible to **predict beyond the range** of the response variable in the training data in a regression problem. Suppose you want to predict house prices using a decision tree and the range of the the house price (response variable) is $5000 to $35000. While predicting, the output of the decision tree will always be within that range.

**Advantages of random forests:**

1. No need to **prune** the trees of a forest.
2. The **OOB error** can be calculated from the training data itself which gives a good estimate of the model performance on unseen data.
3. It is hard for a random forest to **overfit**the training data.
4. A random forest is not affected by **outliers** as much because of the aggregation strategy.



The limitations of a random forest are:

1. Owing to their origin to decision trees, random forests have the same problem of **not predicting beyond the range of the response variable** in the training set.
2. The **extreme values are often not predicted** because of the aggregation strategy. To illustrate this, let’s take the house prices example, where the response variable is the price of a house. Suppose the range of the price variable is between $5000 and $35000. You train the random forest and then make predictions. While making predictions for an expensive house, there will be some trees in the forest which predict the price of the house as $35000, but there will be other trees in the same forest with values close to $35000 but not exactly $35000. In the end, when the final price is decided by aggregating using the mean of all the predictions of the trees of the forest, the predicted value will be close to the extreme value of $35000 but not exactly $35000. Unless all the trees of the forest predict the house price to be $35000, this extreme value will not be predicted.

