# Pros and Cons of LR, Decision Trees and SVM

## 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.

## Cons

1. **Logistic regression**
   1. It does not perform well when the features space is too large.
   2. It does not perform well when there are a lot of categorical variables in the data.
   3. The nonlinear features have to be transformed to linear features in order to efficiently use them for a logistic model.
   4. It relies on entire data i.e. if there is even a small change in the data, the logistic model can change significantly.
2. **Decision trees**
   1. Trees are highly biased towards the training set and overfit it more often than not.
   2. There is no meaningful probability score as the output.
3. **Support vector machines**
   1. SVMs are not efficient in terms of computational cost when the number of observations is large.
   2. It is tricky and time-consuming to find the appropriate kernel for a given data.

**Pros and Cons of Different Machine Learning Models**

Say you work for a large e-commerce company such as Amazon and need to build a classification model to classify a user as likely to buy / unlikely to buy. You have a large number of features and observations, and have to deploy the model in real time. Compare the pros and cons of logistic regression, decision trees and support vector machines in such a case. Write your arguments and the final choice/approach in the box below.

**Suggested Answer**

***Cons: 1.*** *Logistic regression might not perform as well as other algorithms in terms of accuracy and other such performance metrics because of the potential nonlinearity in the dataset.* ***2.*** *Decision trees are prone to overfit the data by creating complex rules which mug up the whole data.* ***3.*** *Support vector machines might not be appropriate for this task since it requires the model to be deployed in real time, and as discussed earlier, SVMs are resource hungry and slow as compared to other machine learning models.* ***Pros:*** *1. Since the project is to be deployed in real time, logistic regression and decision trees will be the right choice since they are faster to build than support vector machines. 2. In general, support vector machines give a really good performance as compared to logistic regression or decision trees when the number of features is large. In the end, you have to test and compare all the models in terms of the following - 1. Predictive power (accuracy, sensitivity and specificity, AUC etc.), and 2. Computational cost After analysing the above, you have to choose the model that gives a right balance of both the goals.*

## CART and CHAID Trees

So far, you studied a specific type of tree: **CART (Classification and Regression Trees)**. There is one more tree that is used widely. It is called **CHAID (Chi-square Automatic Interaction Detection)**. Both of these trees have different applications. A **chi-square test** is a statistical hypothesis test where the test statistic is chi-squared distribution. This test is used to compare the interaction of independent variables with the dependent variable.

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**.

**CHAID:**

1. It uses multiway splits by default
2. It is intended to work with categorical or discretized target
3. It uses truncation by making a node split only if a significance criterion is fulfilled
4. It tries to prevent overfitting right from the start, because at every node it checks the multiway splits is significance or not

If the parameters are binomial then we go for CART whereas if the parameters are more like multi categorical we go for CHAID. There is a difference between driver analysis v/s model forcasting/prediction. Driver analysis is to find more prominent parameters for which we are arriving such result, for those kind of analysis CHAID is better choice where as to predict or forcast CART is better.

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.

Suppose you are asked to build a decision tree model for a classification problem, by your manager. The aim of the organisation is to understand the driver KPIs (Key Performance Indicators), i.e. features that play an important role in the problem at hand. Which type of tree would be more appropriate for this task?

Top of Form

CART (Classification and Regression Tree)

CHAID (Chi-square Automatic Interaction Detection)

**Feedback :***CHAID trees are suitable when you need to understand the driver KPIs, instead of predicting the class.*

Bottom of Form

**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.