**Comparison: Classification algorithms**

**Describe the similarities and differences between logistic regression, SVM and decision trees from different aspects.**

Logistic regression, Supported Vector Machine (SVM), and Decision Trees, are three different classification algorithms for supervised learning.

All of them are useful tools to tackle classification problems with large databases. Depending on the specific issue, the number of features and the number of samples, one will be better than the other. For example, supported vector machine is well-recognised for having a good performance with high-dimensional data. And capable of using numerical and categorical data. It doesn’t rely on the entire data. On the contrary, it is a method that uses lots of memory and CPU time. That means that it is inefficient with a large number of samples. Also, it is not easy to find appropriate kernel sometimes.

On the other side, logistic regression allows us to have good performances with diagonal decision boundaries, that’s similar to SVM but not with decision trees which don’t perform well in diagonal decision boundaries. Logistic regression has a low variance, and it usually has a high bias in decision boundaries that are not diagonal. Logistic regression, as SVM, can be used with kernel methods. And it is widely spread on industry to perform their solutions because it usually gives us convenient probability scores for our observations.

On the other hand, logistic regression relies on the entire data unlike SVM and decision trees and that could be a handicap to find the best solution when we must work with lots of outliers or lack of values. Also, it doesn’t perform well when feature space is too large, unlike SVM.

Finally, decision trees are easy to interpret and explain visually, are not affected by outliers and can deal with a reasonable amount of missing values. They are fast at classifying unknown records and are inexpensive to construct. Although, decision trees are highly biased to training set (to solve this we can use Random Forests), and are prone to overfitting like SVM.