Week 7 Decision Trees Vs Random Forest

When would you prefer a decision tree over a random forest and vice versa?

When the objective is to comprehend the underlying structure of the data and interpretability is crucial, decision tree is better. When using ensemble methods to increase the precision and stability of forecasts, a random forest is better choice. Because the random forest can lower variance by mixing numerous trees, it is more appropriate when the data is noisy. On the other hand, if the tree is excessively deep and complicated, decision trees can easily overfit, in which case a random forest may be a preferable option.

Week 8 Ada Boost

Let's talk about Adaptive Boosting here. Based on your understanding, when/why would you want to use adaptive boosting for your modeling purposes? Feel free to use any websites here.

AdaBoost can be a powerful tool for improving the performance of a model, particularly when the base model has high bias and low variance. It is also relatively simple to implement and can be applied to a wide range of models and problems such as decision trees, neural networks, and SVM.

Some other reasons you would want to use Ada boosting is to to improve the performance of a weak classifier or reduce overfitting or be more computationally efficient and handle large datasets. And lastly to remove noise in the data

Week 9: Support Vector Machine

From your perspective, what are some of the advantages of using Support Vector Machine?

Some of the advantages of SVM are that they can be used for both Regression and Classification problems. SVM can handle high dimensional data as they can efficiently perform linear or non-linear classification in high-dimensional space. Furthermore, SVM handles noise in data well. SVM can be used to handle imbalanced data sets as well as non-linear separable data as it can find the maximum-margin hyperplane, which can separate non-linearly separable data. And finally, SVM can be used for both linear and non-linear problems and multi-class classification.

Week 10: Unsupervised learning

Let's talk about unsupervised learnings. Unsupervised learning methods are extremely useful in real world situations, when data are often unreasonable messy, unlabeled. These techniques allow us to find unknowns subgroups and patterns as well as conduct dimension reduction, which might be necessary in very wide datasets. Based on your understanding, and for the industry domain you are in, when do you think you could be using unsupervised methods? What would be some of the advantages of using unsupervised methods? Feel free to cite websites.

I think a great way to use unsupervised learning is finding bottlenecks in today’s Data Engineering Stack. Many times, we don’t have the structural data to get analysis on where something may not be performing as expected or there is a bottle neck in the pipeline. By applying Unsupervised learnings once can look for trends, groups, patterns, and conduct analysis on where there is a bottle neck and the impact of fixing that bottleneck. It would be tough to find this with supervised data as there might not be structure data which can track all this lineage, interaction transferring of data through a modern data stack.

Week 11: Unsupervised learning: K-means clustering and hierarchical clustering

Let's discuss the difference between K-means clustering and hierarchical clustering.

K-Mean clustering and Hierarchical clustering are both clustering algorithms. K mean is distance based algorithm based on closest distance to the centroid. It works by partitioning the data into K clusters, where K is a user-specified parameter. Hierarchical clustering is a more flexible technique that does not require the user to specify the number of clusters in advance. It works by creating a hierarchical tree also called dendrograms of the data, where each leaf node represents a single observation, and each branch represents a cluster of observations. -means is computationally faster than hierarchical clustering and is better suited for large datasets, while hierarchical clustering is better for smaller datasets and can be used for creating a more informative representation of the clusters.

Week 12: Dimension reduction

When we are working with a dataset with a large number of variables (or, columns), perhaps in the hundreds or thousands, we might want to think about dimensionality reduction. We could either choose a subset of variables, or extract information from the variables/features to create new feature subspace. Let's discuss how Principal Component Analysis could help us reduce dimension.

PCA can help to reduce dimensionality through Data Compression, Visualization, and Improved performance. It chooses to retain only most important principal components thus it reduce the size of the data while retaining most of the information. It has highly interpretable plots which makes it easier to visualize patterns and relationships in the data and lastly by reducing the number of variables, PCA can improve the performance of machine learning algorithms, such as K-Means or XGBoost, which can be computationally expensive when dealing with large numbers of variables.

Week 13: Neural Network

Let's take a look at this article and discuss!

Its interesting how the university was responsible for major breakthroughs in deep neural network technology, including backpropagation. However, the articles states how deep neural nets may not use the same algorithm as real brains, as backpropagation is not compatible with the brain's anatomy and physiology. As a result, Researchers are exploring more biologically plausible learning mechanisms to understand the algorithms used by the brain. Once there is more discovery on the human brain This could have implications for both AI and understanding the brain itself.