Introduction

For the final project we were tasked to use what we have learned on a dataset of our choice. Since we both watch Youtube, we decided to work with a Youtube data set from the Kaggle collection. This dataset contains multiple sets of Youtube metadata from many countries around the world. Each country has the same columns but a wide variety of different Youtube videos and categories. In this project we focused on the USA data set since that had the most familiar videos. From the Youtube dataset our goal was to extract useful information about the trending videos; what videos are trending and how fast do they trend? Using this information we wanted to use a decision tree to predict how long it will take for a video to go trending. There is a lot of information in these datasets that we are not using but would not aid the decision tree in its training or prediction. Overall, it was interesting challenge for the decision tree algorithm to predict when a video will go trending.

Implementation

We want to build a decision tree to help us to predic the date based on our input data. For decision tree, our input should be integer instead of text, so we can not input our data directly. It means we need to preprocess our data into the right format. At first, we choose these attributes that are helpful for our target. For attributes that are text formats we just convert them into category type because of the need of integer type input of decision tree. And we have two different time in raw data attributes, we can summrize them by the duration. Because our target is the time it takes from published time. So we can work out this duration and replace other dates. There are also some data that might have some errors and they are already marked. We also need to clean these data. Then we split our data set into training and testing part.

Results

After parsing our data and skimming it down to just the right columns we still struggled to make the decision tree preform well. We were barely even getting 10 percent accuracy from the tree. Obviously, this not ideal and we ultimately would want

Conclusion