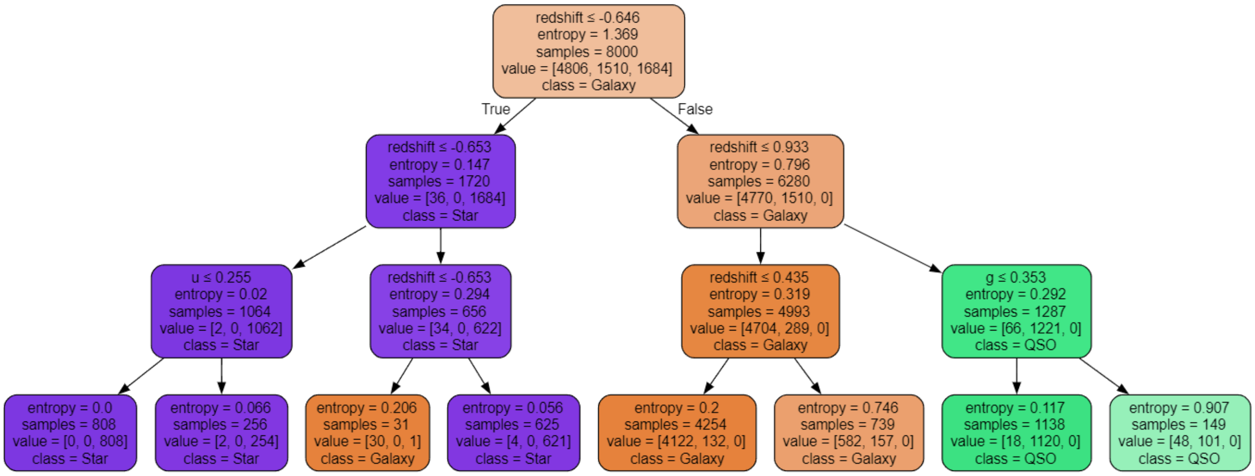
Celestial Body Classification Insights (My Dataset)

Decision Tree



Redshift shows up very frequently in this tree suggesting it is the most powerful indicator in determining what type of celestial body the data point is. The other two are u and g suggesting ultraviolet and gamma values are also quite predictive.

Churn Classification (Provided Dataset)

Removing Useless Features

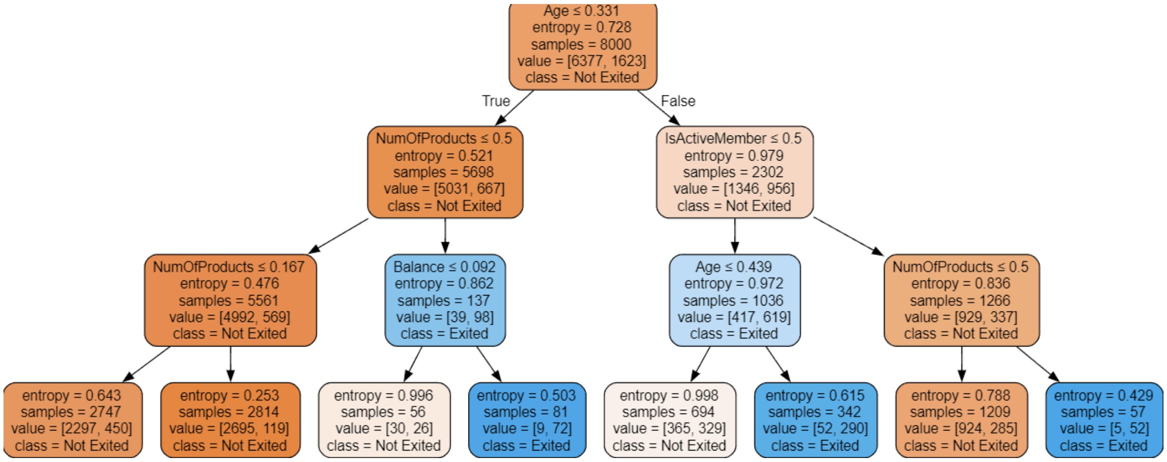
I decided to remove RowNumber, CustomerId, and Surname. These are all identifiers in a way and therefore are useless to our predictions. I also eliminated features based on low correlation to the exited feature. This led to a sizable increase in some models such as KNN from 82% to 86%.

Train Test Split

After testing many train test splits they all yielded relatively similar results with 1-2% variation in the most extreme cases. After trying them the optimal appeared to be the 80:20 split keeping the models on the higher end of the accuracy variation.

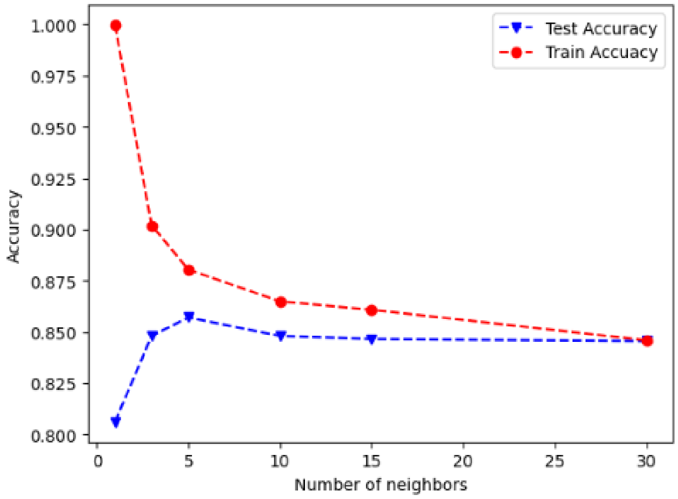
Decision Tree

I found that my decision tree model maintained an 83%-84% accuracy across all train test splits. The highest being 84% at 80:20.



Age was found to be the root feature suggesting it plays a significant role in predicting those who will cancel their subscription in a given amount of time. NumOfProducts, IsActiveMember, and Balance are the only other features found on the tree suggesting they are very important.

KNN

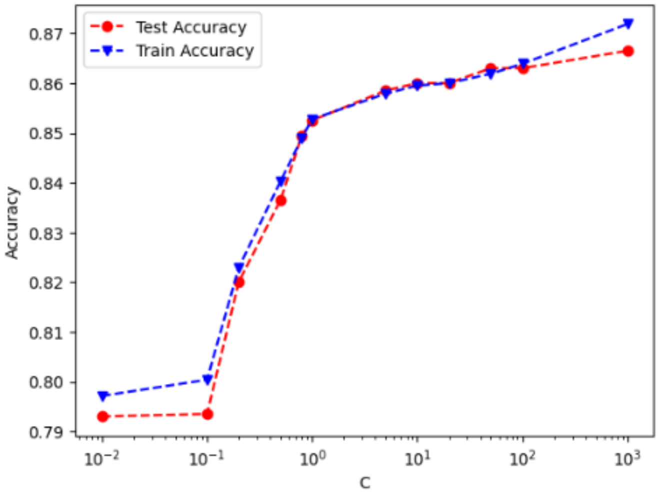


KNN at k=5 has around 82% accuracy. We can see the peak is at k=5 with sharp incline leading up to it and gradual decrease as k increases indicating k=5 is optimal. This indicates that customer behavior is quite granular because as k increases past 5 the neighbors become less and less predictive of the individual data point.

Naive Bayes

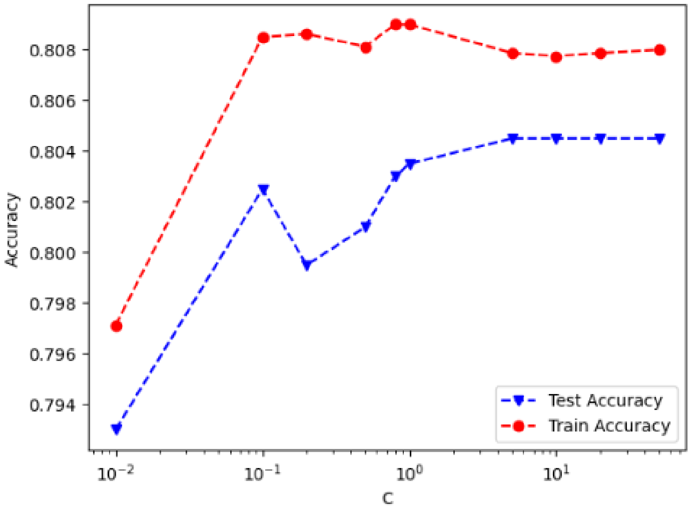
I found that my naive bayes model maintained a consistent 82% accuracy regardless of the train test split. The splits tried were 90:10, 80:20, 70:30, and 60:40.

SVM



Initially I was using kernel=linear for SVC however I was getting static results as in accuracy would not change as C did. Once I set kernel=rbf I got the graph above. This indicates that the relationship is better represented non linearly. As we can see from the red line the optimal C param is very large with an accuracy of up to 86.5%. The model could potentially benefit from even higher C values however this results in extremely long compute time and the model . This indicates that the train and test data behave similarly in SVM as C is high.

Logit



The optimal C param is C>=10 with an 80% accuracy. After C= 10 the graph plateaus indicating higher C values will not help the model. The high C value also indicated that the datasets both train and test behave relatively the same in the context of logistic regression.