South Dakota Mines
Spring 2022

CSC 454 - Data Mining Theory

Stellar Data Mining

Stellar Data Mining Final Deliverable Document

Mathew Clutter, Chami Senarath

Contents

1 Introduction

There are many different types of stars in the universe, from Protostars to Red Supergiants. They can be categorized according to their mass, and temperature. Stars are also classified by their spectra (the elements that they absorb), along with their brightness (apparent magnitude). The spectral class of a star can tell astronomers a lot about it. There are seven main types of stars. In order of decreasing temperature, O, B, A, F, G, K, and M. Although there are scientific reasons why stars are different colors and sizes, everyone can enjoy this reality by simply looking up at the night sky.

In this paper, we are using a dataset consisting of many different stars to evaluate different classifiers abilities to accurately classify stars according to their spectral class. We explore the ability of the Decision Tree, Rule-Based, k-Nearest Neighbor, and Naïve Bayesian Classifiers. We also make some notes about Neural Network and Random Forest. We arrived at a hypothesis that given how the data in our dataset behaves (explained in the next section), Decision Tree would be the best classifier. Then we present accuracy results for above mentioned classifiers using different flavors of sampling and cross validation methods. These results are then evaluated to determine and validate our hypothesis.

This analysis was be developed by Mathew Clutter and Chami Senarath. Their email addresses can be found below.

- Mathew Clutter: Mathew.Clutter@mines.sdsmt.edu
- Chami Senarath: Chamaka.Senarath@mines.sdsmt.edu

As both Mathew and Chami are outer-space enthusiasts, they are excited to work on a project related to stellar classification and expand their knowledge on such subjects.

2 Information about the Data Set

This dataset was obtained from a French website from the University of Strasbourg. This website allowed us to query a large database of stars and obtain attributes that were desired to perform the classifications. A link to the website that our data was obtained from follows: `https://vizier.u-strasbg.fr/viz-bin/VizieR-3?-source=V/137D/XHIP&-out.max=50&-out.form=HTML%20Table&-out.add=_r&-out.add=_RAJ,_DEJ&-sort=_r&-oc.form=sexa`. This website makes queries to the larger Extended Hipparcos Compilation database. From this database, the following attributes were selected for our dataset for this study:

- Name: The common name of the star, if known (meta attribute)
- Right Ascension: The right ascension of the star in the sky (component of the position of the star). This attribute is not used in the classification.
- Declination: Declination of the star in the sky (another component of the position of the star in the sky). This attribute is not used in the classification.
- Spectral Type (Full): The spectral type (Morgan-Keenan System) that the star is a member of. This is the full spectral type, that contains both the class and subclass, as well as a luminosity class. (For example, G2V)
- Spectral Type: The spectral type from the Morgan-Keenan System, with the class and subclass. (For example, G8). This is the target attribute that will be classified.
- Temperature Class Codified (Tc): The temperature of the star, with values scaled to a range 0-147, with a median of 50, and mean of 48.9. The temperature range will allow comparisons of temperatures between stars.
- Luminosity Class Codified (Lc): The luminosity of the star, with values scaled to a range 0-126, with median 4, and mean 4.1. Luminosity class allows for comparisons of luminosity between stars.
- Iron Abundance: The amount of iron present in the star, if known. Compared to the amount of iron that is present in the sun.
- Age: The age of the star, in billions of years.

- Number of Exoplanets: The number of exoplanets that the star has, if known.
- B-V Color Index: Describes numerically the color of the star, according to the Johnson Color Index.
- Absolute Magnitude: The absolute magnitude of the star.
- Stellar Luminosity: The luminosity of the star, compared to the luminosity of the sun.

This database and website allowed us to obtain a large dataset with 117955 stars listed to train and evaluate classifiers.

As this is a large dataset, and not every star has every property recorded, there are some missing values for certain attributes. In particular, there is a lot of data missing for the number of exoplanets, age, and iron content.

Finally, some additional information regarding this dataset can be found in the appendix, and more information about how we processed and analyzed this dataset before classifying can be found in the preproccesing section.

3 Our Hypothesis

Before our analysis began, we anticipated that the decision tree classifier would produce the most accurate model. This was anticipated for a number of reasons. Primarily, the decision tree classifier can handle missing and irrelevant values with relative ease, compared to other classifiers, such as rule based and Naïve Bayesian classifiers. This dataset has a large number of missing values, and potentially some irrelevant attributes. Thus, the decision tree classifier is well suited to handle such a dataset. The relative robustness of the decision tree classifier led us to anticipate good performance from the decision tree classifier.

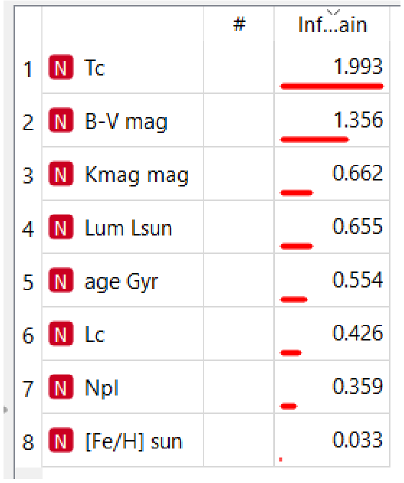
In regards to evaluation, it was hypothesized that the method of evaluation for the classifiers would not have much impact on the calculated classification accuracy. We anticipated that the difference in classification accuracy between stratified vs not stratified, and cross-validation vs random sampling would be minimal, suggesting that the method used to evaluate the classifier's performance has little importance.

4 How the Study was Conducted

4.1 Preprocessing

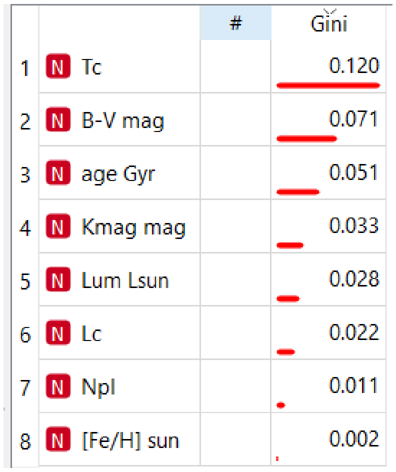
The process to make this data usable in Orange was fairly lengthy. Firstly, the data downloaded from the website was in a semicolon separated file, as opposed to a comma separated file. This was an easy fix, by simply replacing the semicolons present with commas. A second issue came with some improper values for certain attributes, that made Orange unable to interpret the data properly. For example, there were occasional values within the luminosity and temperature class that consisted of characters, as opposed to integers. This meant that Orange was unable to properly import the data without first cleaning up all of the excess characters present. After the garbage in the dataset was cleaned up, a new column was needed to use as a target value to predict. The original spectral type attribute in the dataset consisted of 2011 different spectral types. This was too many attributes for Orange to properly handle. In order to cut down the number of categories we sought to classify data into, only the class and first subclass of the spectral type was kept. This reduced the number of categories from 2011 to 91. This was much more manageable number of categories to classify the data into. Overall, preparing the data, and manipulating it into a usable form for Orange was not a trivial task.

In an effort to better understand the dataset, the information gain and Gini decrease was calculated for each attribute, using the Rank widget provided in Orange. While both measures agree that the most significant attribute to help with the classification is the temperature, other attributes have slightly different orderings between the two measures. The output of the Rank widget can be seen below:



		#	Inf...ain
1	N Tc		1.993
2	N B-V mag		1.356
3	N Kmag mag		0.662
4	N Lum Lsun		0.655
5	N age Gyr		0.554
6	N Lc		0.426
7	N Npl		0.359
8	N [Fe/H] sun		0.033

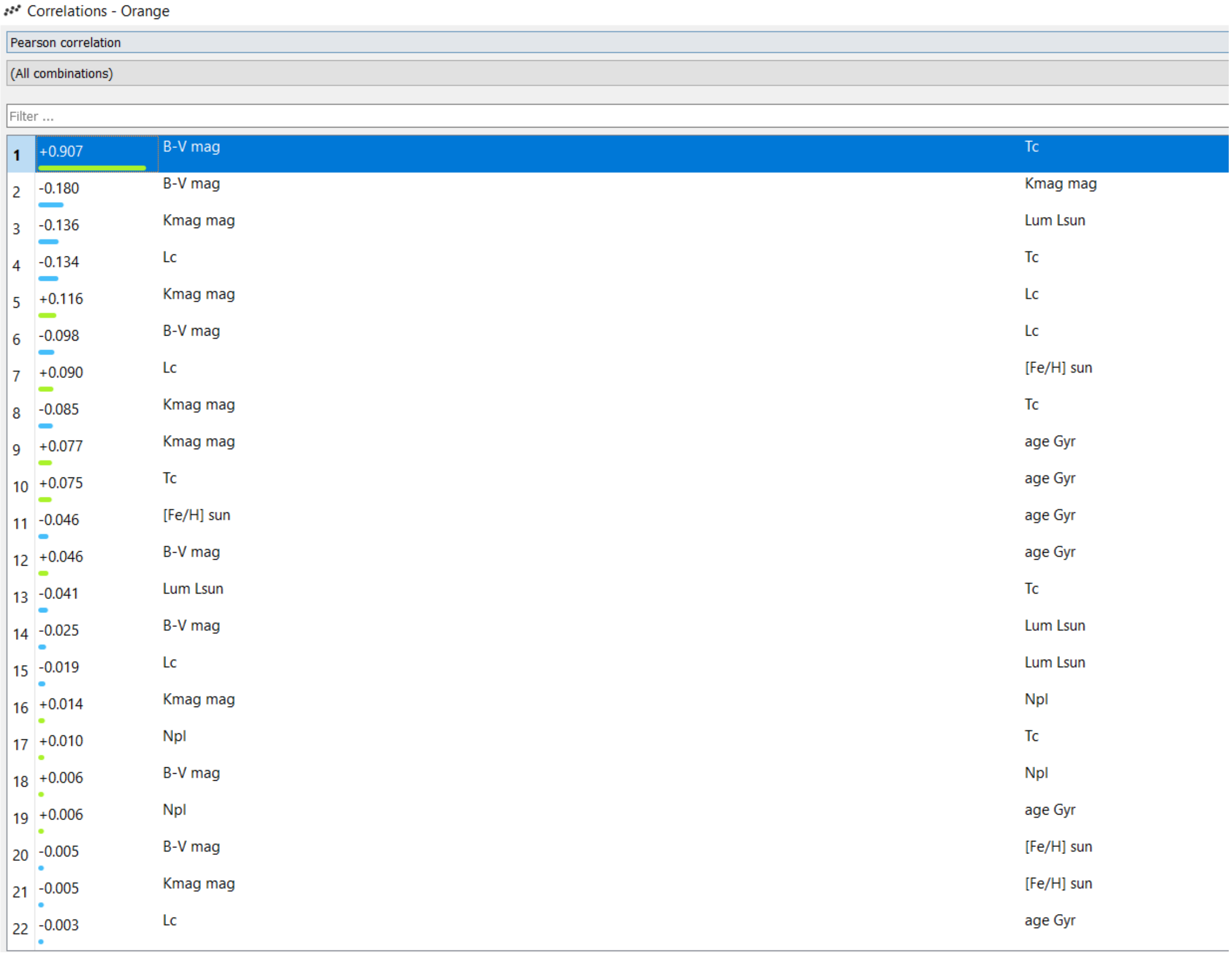
Figure 1: Information Gain Ranks



		#	Gini
1	N Tc		0.120
2	N B-V mag		0.071
3	N age Gyr		0.051
4	N Kmag mag		0.033
5	N Lum Lsun		0.028
6	N Lc		0.022
7	N Npl		0.011
8	N [Fe/H] sun		0.002

Figure 2: Gini Decrease Ranks

Additionally, there are some strong correlations between different attributes in this data set. Using the correlation widget in Orange, the following correlations were calculated for both Pearson and Spearman correlations.



Correlations - Orange

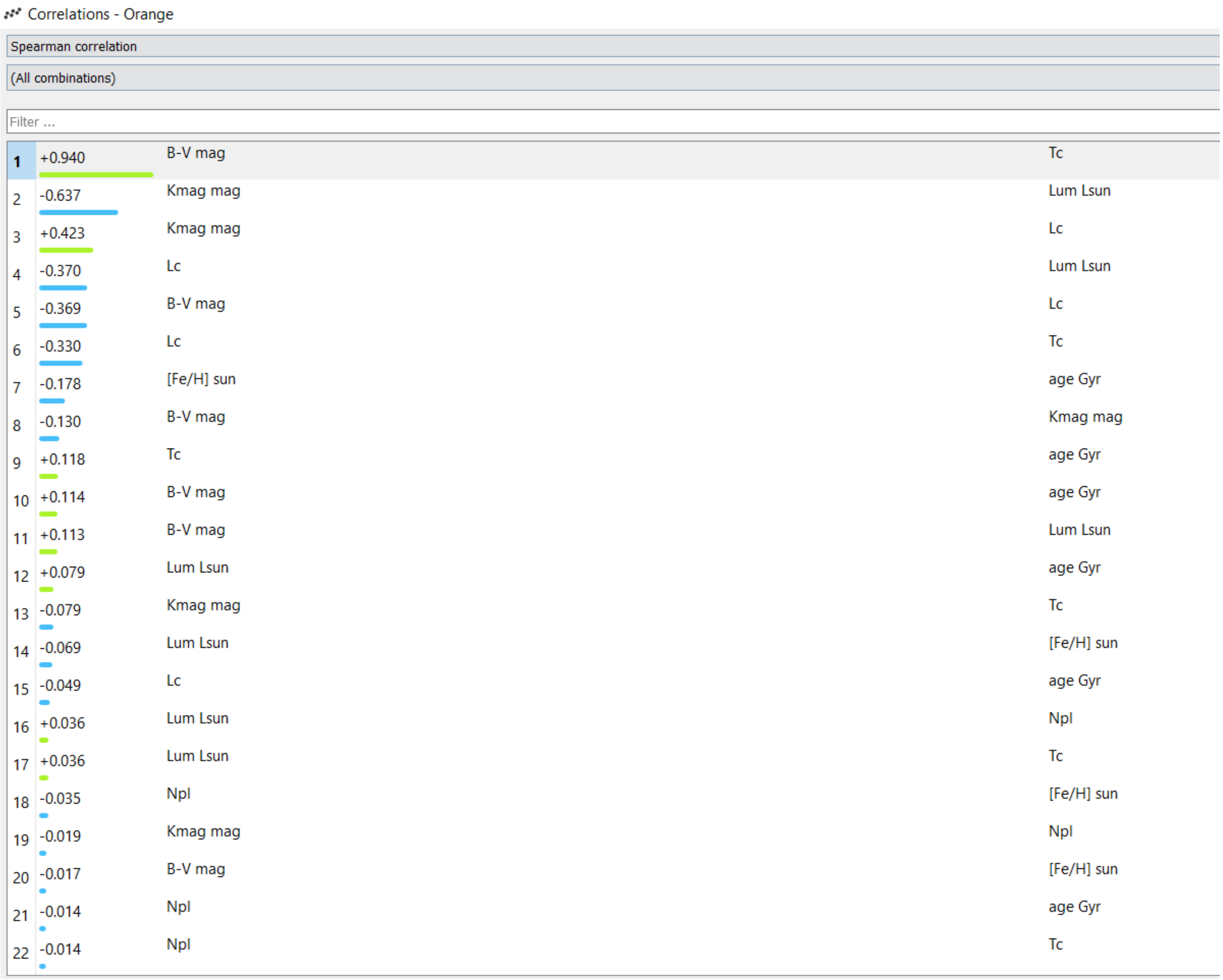
Pearson correlation

(All combinations)

Filter ...

1	+0.907	B-V mag	Tc
2	-0.180	B-V mag	Kmag mag
3	-0.136	Kmag mag	Lum Lsun
4	-0.134	Lc	Tc
5	+0.116	Kmag mag	Lc
6	-0.098	B-V mag	Lc
7	+0.090	Lc	[Fe/H] sun
8	-0.085	Kmag mag	Tc
9	+0.077	Kmag mag	age Gyr
10	+0.075	Tc	age Gyr
11	-0.046	[Fe/H] sun	age Gyr
12	+0.046	B-V mag	age Gyr
13	-0.041	Lum Lsun	Tc
14	-0.025	B-V mag	Lum Lsun
15	-0.019	Lc	Lum Lsun
16	+0.014	Kmag mag	Npl
17	+0.010	Npl	Tc
18	+0.006	B-V mag	Npl
19	+0.006	Npl	age Gyr
20	-0.005	B-V mag	[Fe/H] sun
21	-0.005	Kmag mag	[Fe/H] sun
22	-0.003	Lc	age Gyr

Figure 3: Pearson Correlations



Correlations - Orange

Spearman correlation

(All combinations)

Filter ...

1	+0.940	B-V mag	Tc
2	-0.637	Kmag mag	Lum Lsun
3	+0.423	Kmag mag	Lc
4	-0.370	Lc	Lum Lsun
5	-0.369	B-V mag	Lc
6	-0.330	Lc	Tc
7	-0.178	[Fe/H] sun	age Gyr
8	-0.130	B-V mag	Kmag mag
9	+0.118	Tc	age Gyr
10	+0.114	B-V mag	age Gyr
11	+0.113	B-V mag	Lum Lsun
12	+0.079	Lum Lsun	age Gyr
13	-0.079	Kmag mag	Tc
14	-0.069	Lum Lsun	[Fe/H] sun
15	-0.049	Lc	age Gyr
16	+0.036	Lum Lsun	Npl
17	+0.036	Lum Lsun	Tc
18	-0.035	Npl	[Fe/H] sun
19	-0.019	Kmag mag	Npl
20	-0.017	B-V mag	[Fe/H] sun
21	-0.014	Npl	age Gyr
22	-0.014	Npl	Tc

Figure 4: Spearman Correlations

Before the models were created, the data was split into testing and training sets. In total, there are 117955 stars in this dataset. We conducted a random 70/30 split into training and testing data respectively. This resulted in a training set with 82594 stars, and a testing set with 35361 stars.

Once the data was manipulated into a usable format for Orange, and our understanding of the structure of the data was complete, the data mining process could begin.

4.2 Data Mining

Using Orange, we built models for Decision Tree, Rule-Based Classifier, k-Nearest Neighbor, Naïve Bayesian Classifier, Neural Network, and Random Forest.

Each classifier was then evaluated using the following classification accuracy measurements with both training and testing data:

- Cross Validation, 2 folds, stratified
- Cross Validation, 2 folds, not stratified
- Cross Validation, 10 folds
- Cross Validation, 10 folds, stratified

- Cross Validation, 20 folds
- Cross Validation, 20 folds, stratified
- Random Sampling, 2 repeat, 66% sample size
- Random Sampling, 10 repeat, 5% sample size
- Random Sampling, 10 repeat, 66% sample size

4.3 Postprocessing

Once the models were created, we attempted to revise the models using both imputing of missing data (by filling missing values with the average) and removing outliers (using local outlier factor and euclidean metric) from the dataset. We also performed some further testing by classifying with only the Tc class (which has the largest information gain and Gini decrease), and classifying without the Tc class.

5 Different Classifications

5.1 Decision Tree

The decision tree classifier creates a tree that splits based on certain attributes, to divide the dataset into the appropriate classes. This is perhaps the most basic of classifiers, as it's structure is quite intuitive. The decision tree classifier performed quite well with this data set. Trees created with different settings all produced accuracies evaluated at over 97%.

The first tree was constructed from the training data, with the default settings of a minimum of 3 instances in each leaf, not splitting subtrees smaller than 5, and limiting the maximal tree depth to 100. These settings created the following tree with 171 nodes and 86 leaves:

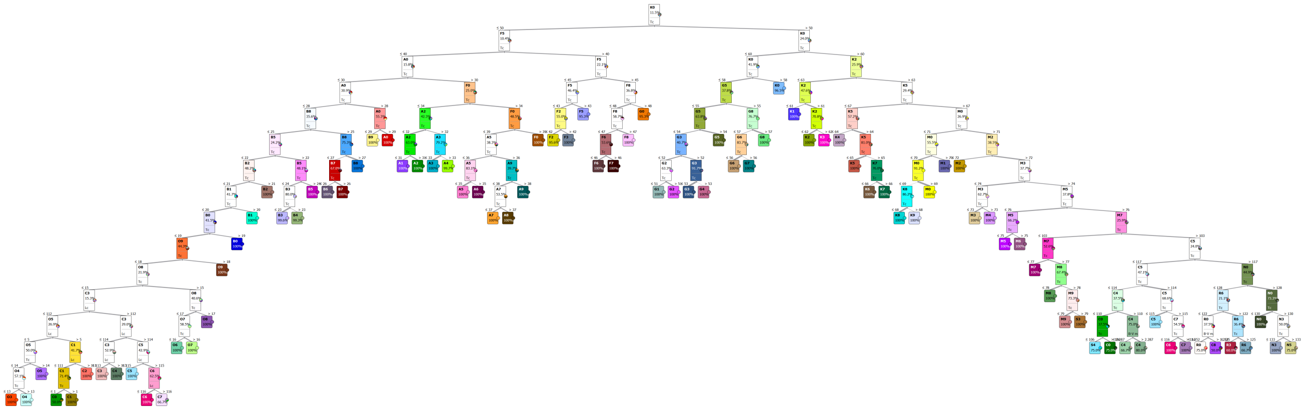


Figure 5: Basic Decision Tree

Using the training data, the following evaluations were obtained:

- Cross Validation, 2 folds, stratified: 0.988
- Cross Validation, 2 folds, not stratified: 0.988
- Cross Validation, 10 folds: 0.988
- Cross Validation, 10 folds, stratified: 0.988
- Cross Validation, 20 folds: 0.988
- Cross Validation, 20 folds, stratified: 0.988

- Random Sampling, 2 repeat, 66% sample size: 0.987
- Random Sampling, 10 repeat, 5% sample size: 0.989
- Random Sampling, 10 repeat, 66% sample size: 0.989

Using the testing data, the following evaluations were obtained:

- Cross Validation, 2 folds, stratified: 0.994
- Cross Validation, 2 folds, not stratified: 0.992
- Cross Validation, 10 folds: 0.994
- Cross Validation, 10 folds, stratified: 0.994
- Cross Validation, 20 folds: 0.995
- Cross Validation, 20 folds, stratified: 0.995
- Random Sampling, 2 repeat, 66% sample size, stratified: 0.995
- Random Sampling, 2 repeat, 66% sample size: 0.992
- Random Sampling, 10 repeat, 5% sample size, stratified: 0.986
- Random Sampling, 10 repeat, 5% sample size: 0.983
- Random Sampling, 10 repeat, 66% sample size, stratified: 0.995
- Random Sampling, 10 repeat, 66% sample size: 0.993

A second tree was created, and this tree was allowed to be a bit more complex. This was done by allowing the minimum number of instances in leaves to be 1, and not allowing splits of subsets smaller than 2. This did make the tree a bit more complex, with 191 nodes and 96 leaves.

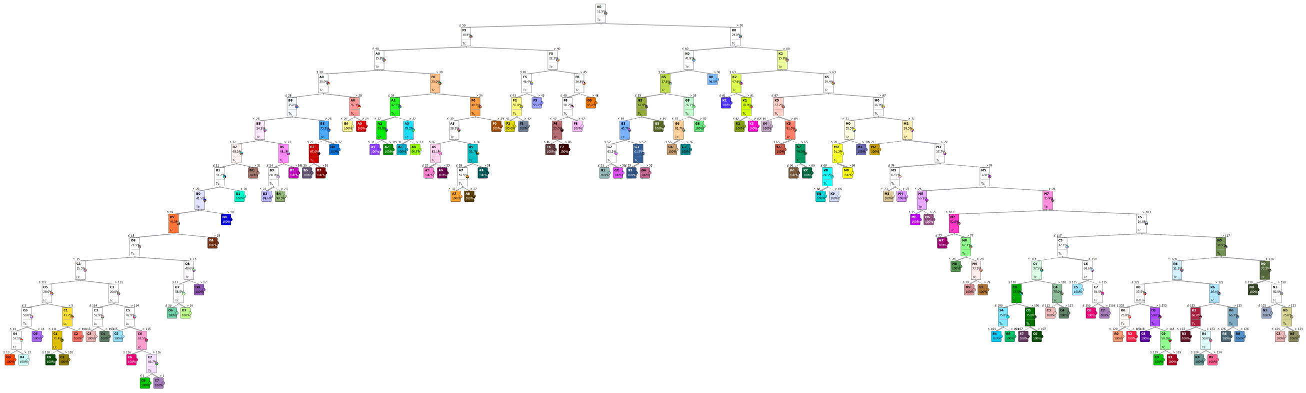


Figure 6: More Complex Decision Tree

Using the training data, the following evaluations were obtained:

- Cross Validation, 2 folds, stratified: 0.988
- Cross Validation, 2 folds, not stratified: 0.988
- Cross Validation, 10 folds: 0.989
- Cross Validation, 10 folds, stratified: 0.989

- Cross Validation, 20 folds: 0.989
- Cross Validation, 20 folds, stratified: 0.989
- Random Sampling, 2 repeat, 66% sample size: 0.988
- Random Sampling, 10 repeat, 5% sample size: 0.991
- Random Sampling, 10 repeat, 66% sample size: 0.989

Using the testing data, the following evaluations were obtained:

- Cross Validation, 2 folds, stratified: 0.995
- Cross Validation, 2 folds, not stratified: 0.992
- Cross Validation, 10 folds: 0.995
- Cross Validation, 10 folds, stratified: 0.995
- Cross Validation, 20 folds: 0.995
- Cross Validation, 20 folds, stratified: 0.995
- Random Sampling, 2 repeat, 66% sample size, stratified: 0.996
- Random Sampling, 2 repeat, 66% sample size: 0.993
- Random Sampling, 10 repeat, 5% sample size, stratified: 0.990
- Random Sampling, 10 repeat, 5% sample size: 0.989
- Random Sampling, 10 repeat, 66% sample size, stratified: 0.995
- Random Sampling, 10 repeat, 66% sample size: 0.994

A third decision tree was also created, with conditions of the tree designed to be a bit simpler. This tree was created with the minimum number of instances in leaves to be 6, and not allowing splits of subsets smaller than 10. This created a tree with 149 nodes and 75 leaves.

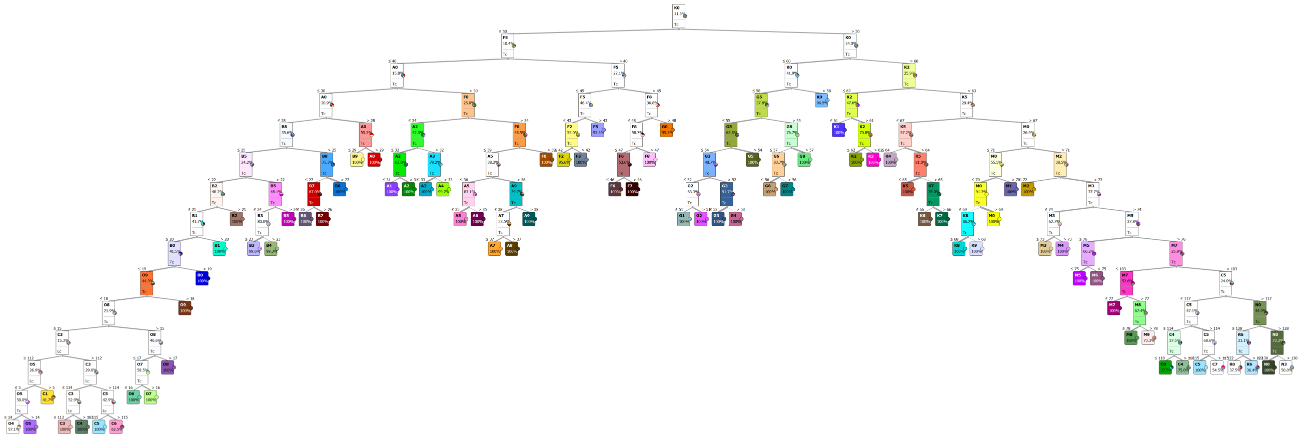


Figure 7: Simpler Decision Tree

Using the training data, the following evaluations were found:

- Cross Validation, 2 folds, stratified: 0.987

- Cross Validation, 2 folds, not stratified: 0.987
- Cross Validation, 10 folds: 0.988
- Cross Validation, 10 folds, stratified: 0.988
- Cross Validation, 20 folds: 0.988
- Cross Validation, 20 folds, stratified: 0.988
- Random Sampling, 2 repeat, 66% sample size: 0.987
- Random Sampling, 10 repeat, 5% sample size: 0.985
- Random Sampling, 10 repeat, 66% sample size: 0.988

Using the testing data, the following evaluations were found:

- Cross Validation, 2 folds, stratified: 0.994
- Cross Validation, 2 folds, not stratified: 0.991
- Cross Validation, 10 folds: 0.994
- Cross Validation, 10 folds, stratified: 0.994
- Cross Validation, 20 folds: 0.994
- Cross Validation, 20 folds, stratified: 0.994
- Random Sampling, 2 repeat, 66% sample size, stratified: 0.994
- Random Sampling, 2 repeat, 66% sample size: 0.992
- Random Sampling, 10 repeat, 5% sample size, stratified: 0.971
- Random Sampling, 10 repeat, 5% sample size: 0.971
- Random Sampling, 10 repeat, 66% sample size, stratified: 0.994
- Random Sampling, 10 repeat, 66% sample size: 0.992

A few miscellaneous observations regarding the decision tree classifier:

- The majority of misclassifications are close to the correct class. From inspecting the confusion matrices, most of the missclassified items have the wrong number, but the correct letter in their spectral type. This implies that the missclassifications are close to the correct class.
- Removing the temperature class from the data does not have a notable negative impact on the classification accuracies. However, it does lead to a much more complex decision tree. By removing the temperature data from the tree building algorithm, a tree with 14751 nodes and 7376 leaves is created.
- Similarly, creating the tree with only the temperature class creates a tree with 163 nodes and 82 leaves. This tree also does not see a significant decrease in its classification accuracy.
- Removing outliers from the training data did not result in a notable change in classification accuracies. It did result in a slightly simpler tree, with 143 nodes and 72 leaves.
- Imputing missing values from the training data also did not result in a notable change in classification accuracies. Imputing missing values resulted in a slightly more complex tree however, with 179 nodes, 90 leaves. [1]

In summary, the decision tree classifier is very good at classifying stars into their proper spectral type. The analysis suggests a worst case scenario of a 97% classification accuracy rate.

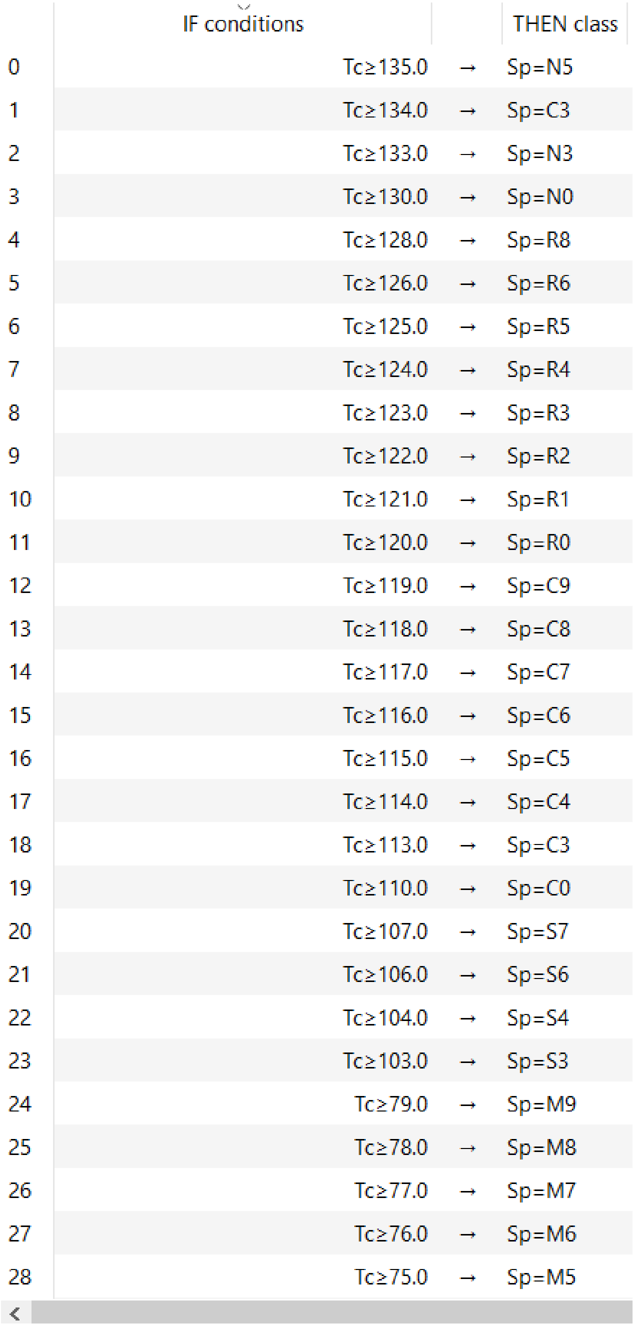
[1] Some of these additional trees are present in the appendix.

5.2 Rule-Based

A rule-based classifier functions by generating a set of rules that determine the target, based on the value of attributes. Items are then classified based on the rule set generated during the creation of the model. These rules consist of a collection of if-then statements connecting the attributes with the target.

The rule-based classifier had strong evaluation results, ranging from .95 to over .99 percent accuracy. However, due to computational limitations, there were only a couple of different methods to create the rule based classifier that were feasible to run. These included two different evaluation measures, entropy and Laplace accuracy. Using entropy to create the rule set resulted in slightly better results than using Laplace accuracy. The following classification accuracies resulted from an ordered, exclusive rule set.

A snippet of the rules generated using entropy is shown below:



	IF conditions		THEN class
0	Tc≥135.0	→	Sp=N5
1	Tc≥134.0	→	Sp=C3
2	Tc≥133.0	→	Sp=N3
3	Tc≥130.0	→	Sp=N0
4	Tc≥128.0	→	Sp=R8
5	Tc≥126.0	→	Sp=R6
6	Tc≥125.0	→	Sp=R5
7	Tc≥124.0	→	Sp=R4
8	Tc≥123.0	→	Sp=R3
9	Tc≥122.0	→	Sp=R2
10	Tc≥121.0	→	Sp=R1
11	Tc≥120.0	→	Sp=R0
12	Tc≥119.0	→	Sp=C9
13	Tc≥118.0	→	Sp=C8
14	Tc≥117.0	→	Sp=C7
15	Tc≥116.0	→	Sp=C6
16	Tc≥115.0	→	Sp=C5
17	Tc≥114.0	→	Sp=C4
18	Tc≥113.0	→	Sp=C3
19	Tc≥110.0	→	Sp=C0
20	Tc≥107.0	→	Sp=S7
21	Tc≥106.0	→	Sp=S6
22	Tc≥104.0	→	Sp=S4
23	Tc≥103.0	→	Sp=S3
24	Tc≥79.0	→	Sp=M9
25	Tc≥78.0	→	Sp=M8
26	Tc≥77.0	→	Sp=M7
27	Tc≥76.0	→	Sp=M6
28	Tc≥75.0	→	Sp=M5

Figure 8: Entropy Ruleset

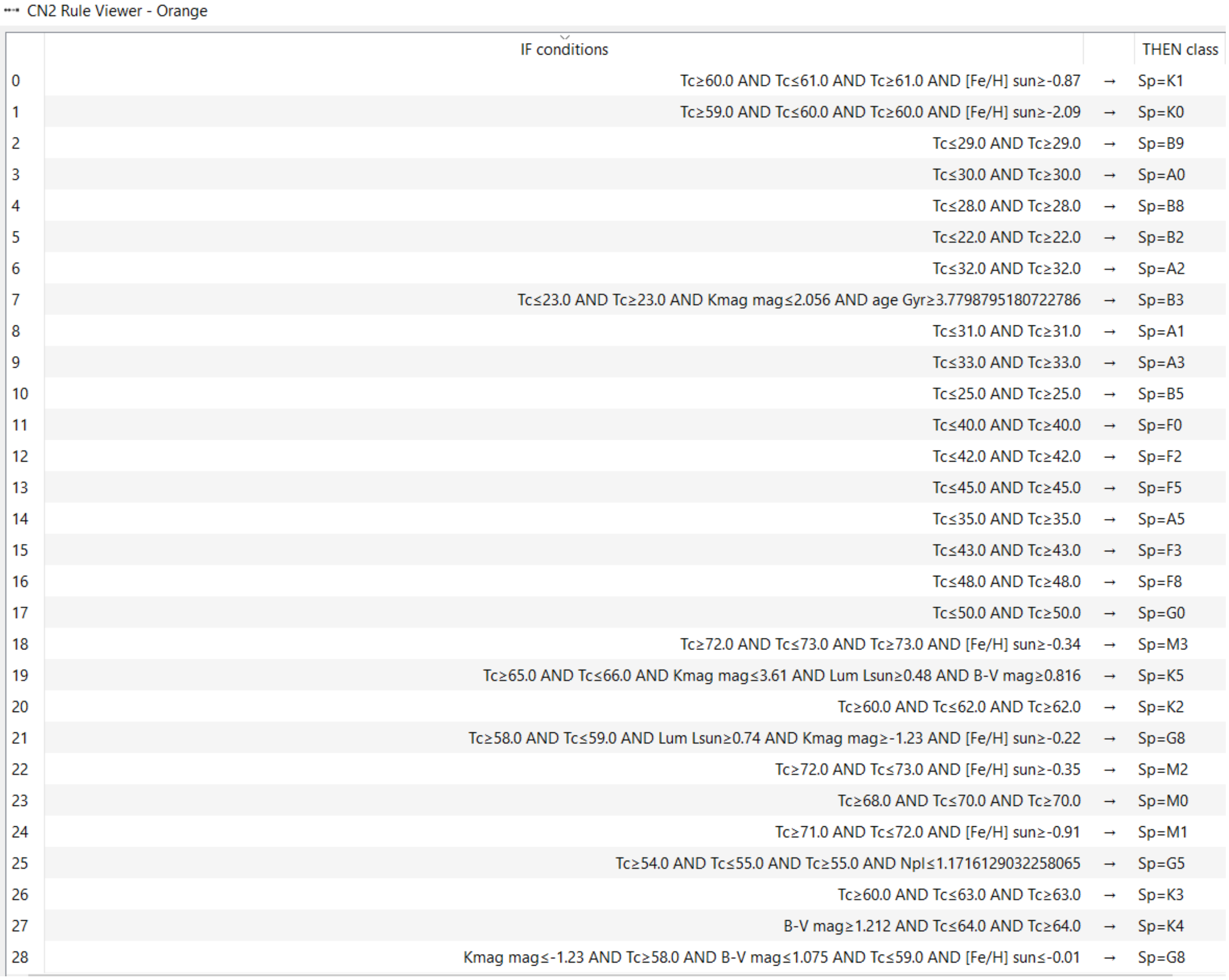
The evaluation results for entropy with the training set were:

- Cross Validation, 2 folds, stratified: 0.999
- Cross Validation, 2 folds, not stratified: 0.999
- Cross Validation, 10 folds: 0.999
- Cross Validation, 10 folds, stratified: 0.999
- Cross Validation, 20 folds: 0.999
- Cross Validation, 20 folds, stratified: 0.999
- Random Sampling, 2 repeat, 66% sample size: 0.999
- Random Sampling, 10 repeat, 5% sample size: 0.990
- Random Sampling, 10 repeat, 66% sample size: 0.999

The evaluation results for entropy with the testing set were:

- Cross Validation, 2 folds, stratified: 0.997
- Cross Validation, 2 folds, not stratified: 0.999
- Cross Validation, 10 folds: 0.999
- Cross Validation, 10 folds, stratified: 0.999
- Cross Validation, 20 folds: 0.999
- Cross Validation, 20 folds, stratified: 0.999
- Random Sampling, 2 repeat, 66% sample size, stratified: 0.998
- Random Sampling, 2 repeat, 66% sample size: 0.999
- Random Sampling, 10 repeat, 5% sample size, stratified: 0.986
- Random Sampling, 10 repeat, 5% sample size: 0.979
- Random Sampling, 10 repeat, 66% sample size, stratified: 0.998
- Random Sampling, 10 repeat, 66% sample size: 0.998

A snippet of the rules generated using Laplace is shown below:



	IF conditions		THEN class
0	Tc≥60.0 AND Tc≤61.0 AND Tc≥61.0 AND [Fe/H] sun≥-0.87	→	Sp=K1
1	Tc≥59.0 AND Tc≤60.0 AND Tc≥60.0 AND [Fe/H] sun≥-2.09	→	Sp=K0
2	Tc≤29.0 AND Tc≥29.0	→	Sp=B9
3	Tc≤30.0 AND Tc≥30.0	→	Sp=A0
4	Tc≤28.0 AND Tc≥28.0	→	Sp=B8
5	Tc≤22.0 AND Tc≥22.0	→	Sp=B2
6	Tc≤32.0 AND Tc≥32.0	→	Sp=A2
7	Tc≤23.0 AND Tc≥23.0 AND Kmag mag≤2.056 AND age Gyr≥3.7798795180722786	→	Sp=B3
8	Tc≤31.0 AND Tc≥31.0	→	Sp=A1
9	Tc≤33.0 AND Tc≥33.0	→	Sp=A3
10	Tc≤25.0 AND Tc≥25.0	→	Sp=B5
11	Tc≤40.0 AND Tc≥40.0	→	Sp=F0
12	Tc≤42.0 AND Tc≥42.0	→	Sp=F2
13	Tc≤45.0 AND Tc≥45.0	→	Sp=F5
14	Tc≤35.0 AND Tc≥35.0	→	Sp=A5
15	Tc≤43.0 AND Tc≥43.0	→	Sp=F3
16	Tc≤48.0 AND Tc≥48.0	→	Sp=F8
17	Tc≤50.0 AND Tc≥50.0	→	Sp=G0
18	Tc≥72.0 AND Tc≤73.0 AND Tc≥73.0 AND [Fe/H] sun≥-0.34	→	Sp=M3
19	Tc≥65.0 AND Tc≤66.0 AND Kmag mag≤3.61 AND Lum Lsun≥0.48 AND B-V mag≥0.816	→	Sp=K5
20	Tc≥60.0 AND Tc≤62.0 AND Tc≥62.0	→	Sp=K2
21	Tc≥58.0 AND Tc≤59.0 AND Lum Lsun≥0.74 AND Kmag mag≥-1.23 AND [Fe/H] sun≥-0.22	→	Sp=G8
22	Tc≥72.0 AND Tc≤73.0 AND [Fe/H] sun≥-0.35	→	Sp=M2
23	Tc≥68.0 AND Tc≤70.0 AND Tc≥70.0	→	Sp=M0
24	Tc≥71.0 AND Tc≤72.0 AND [Fe/H] sun≥-0.91	→	Sp=M1
25	Tc≥54.0 AND Tc≤55.0 AND Tc≥55.0 AND Npl≤1.1716129032258065	→	Sp=G5
26	Tc≥60.0 AND Tc≤63.0 AND Tc≥63.0	→	Sp=K3
27	B-V mag≥1.212 AND Tc≤64.0 AND Tc≥64.0	→	Sp=K4
28	Kmag mag≤-1.23 AND Tc≥58.0 AND B-V mag≤1.075 AND Tc≤59.0 AND [Fe/H] sun≤-0.01	→	Sp=G8

Figure 9: Laplace Ruleset

The evaluation results using Laplace accuracy with the training set were:

- Cross Validation, 2 folds, stratified: 0.993
- Cross Validation, 2 folds, not stratified: 0.993
- Cross Validation, 10 folds: 0.995
- Cross Validation, 10 folds, stratified: 0.995
- Cross Validation, 20 folds: 0.995
- Cross Validation, 20 folds, stratified: 0.995
- Random Sampling, 2 repeat, 66% sample size: 0.993
- Random Sampling, 10 repeat, 5% sample size: 0.975
- Random Sampling, 10 repeat, 66% sample size: 0.994

The evaluation results using Laplace accuracy with the testing set were:

- Cross Validation, 2 folds, stratified: 0.992
- Cross Validation, 2 folds, not stratified: 0.954
- Cross Validation, 10 folds: 0.992
- Cross Validation, 20 folds: 0.992
- Random Sampling, 2 repeat, 66%, stratified: 0.993
- Random Sampling, 2 repeat, 66%: 0.991
- Random Sampling, 10 repeat, 5%, stratified: 0.970
- Random Sampling, 10 repeat, 5%: 0.954
- Random Sampling, 10 repeat, 66%, stratified: 0.993
- Random Sampling, 10 repeat, 66%: 0.992

The rule set created using entropy performed slightly better than the rule set created using Laplace accuracy. One notable difference in the evaluation was the differences between stratified and not stratified evaluations. Generally, the non-stratified evaluations showed a slightly lower classification accuracy compared to their stratified counterparts. This was especially noticeable for the Laplace rule set. Additionally, the Laplace rules utilized more of the attributes present in the data set to perform the classification. The entropy rule set primarily uses only the Tc attribute to make its classifications, while the Laplace rules often use a combination of Tc and other attributes.

Overall, the rule based classifier performed quite well. Unfortunately, due to processing limitations and the computational complexity of calculating a rule set with many possible classifications and a large number of instances, it was not feasible to create an unordered, or weighted rule set. In order to create rule sets of this fashion, it may be possible to take a smaller sample of the data to create a rule set. However, this may negatively impact the classification accuracy. Another note is that neither imputing missing values, nor removing outliers had a notable effect on the performance of the rule based classifier. In addition, creating the rule sets using only the temperature class did not lead to a decrease in classification accuracy. Once again, due to the computational complexity in creating a full rule set, Orange was unable to compute a rule set without the Tc class with entropy based rules. However, it was able to compute a rule set using Laplace accuracy. This rule set did not see a notable decrease in classification accuracy when compared with Laplace rules using all of the attributes, or only the Tc class. A snippet of these rules in included in the appendix.

In closing, the rule based classifier performed well according to the evaluation (consistently above 95% classification accuracy). However, there were some limitations in the creation of different types of rule sets, due to the computational complexity of generating the rule sets.

5.3 Nearest-Neighbor Classifier

Decision tree and rule-based classifiers are examples of eager learners because they are designed to learn a model that maps the input attributes to the class label as soon as the training data becomes available. An opposite strategy would be to delay the process of modeling the training data until it is needed to classify the test examples. Techniques that employ this strategy are known as lazy learner. One way to make this approach more flexible is to find all the training examples that are relatively similar to the attributes of the test example. These examples, which are known as nearest neighbors, can be used to determine the class label of the test example.

For evaluating our dataset, we set the number of neighbours to 5, the metric to calculate distance to Euclidean, and the weight to uniform. The evaluation results with training data are as follows:

- Cross Validation, 2 folds, stratified: 0.929
- Cross Validation, 2 folds, not stratified: 0.929
- Cross Validation, 10 folds: 0.945
- Cross Validation, 10 folds, stratified: 0.945

- Cross Validation, 20 folds: 0.946
- Cross Validation, 20 folds, stratified: 0.946
- Random Sampling, 2 repeat, 66% sample size: 0.938
- Random Sampling, 10 repeat, 5% sample size: 0.812
- Random Sampling, 10 repeat, 66% sample size: 0.938

The evaluation results with testing data are as follows:

- Cross Validation, 2 folds, stratified: 0.893
- Cross Validation, 2 folds, not stratified: 0.893
- Cross Validation, 10 folds: 0.919
- Cross Validation, 10 folds, stratified: 0.919
- Cross Validation, 20 folds: 0.920
- Cross Validation, 20 folds, stratified: 0.920
- Random Sampling, 2 repeat, 66% sample size, stratified: 0.908
- Random Sampling, 2 repeat, 66% sample size: 0.910
- Random Sampling, 10 repeat, 5% sample size, stratified: 0.745
- Random Sampling, 10 repeat, 5% sample size: 0.74
- Random Sampling, 10 repeat, 66% sample size, stratified: 0.908
- Random Sampling, 10 repeat, 66% sample size: 0.908

The Nearest Neighbor classifier was not notably impacted by removing outliers or imputing missing values. Additionally, using only the Tc class, and removing the Tc class did not impact the classification accuracies.

5.4 Naïve Bayes Classifier

A naive Bayes classifier estimates the class-conditional probability by assuming that the attributes are conditionally independent, given the class label g. The conditional independence assumption can be formally stated as follows:

$$P(\mathbf{X}|Y=y) = \sum_{i=1}^{d} P(X_i|Y=y)$$

where each attribute set $\mathbf{X} = X_1, X_2, ..., X_d$ consists of d attributes.

The evaluation results with training data are as follows:

- Cross Validation, 2 folds, stratified: 0.000
- Cross Validation, 2 folds, not stratified: 0.000
- Cross Validation, 10 folds: 0.001
- Cross Validation, 10 folds, stratified: 0.001
- Cross Validation, 20 folds: 0.001
- Cross Validation, 20 folds, stratified: 0.001
- Random Sampling, 2 repeat, 66% sample size: 0.001
- Random Sampling, 10 repeat, 5% sample size: 0.003

- Random Sampling, 10 repeat, 66% sample size: 0.001

The evaluation results with testing data are as follows:

- Cross Validation, 2 folds, stratified: 0.000
- Cross Validation, 2 folds, not stratified: 0.000
- Cross Validation, 10 folds: 0.001
- Cross Validation, 10 folds, stratified: 0.001
- Cross Validation, 20 folds: 0.003
- Cross Validation, 20 folds, stratified: 0.003
- Random Sampling, 2 repeat, 66% sample size, stratified: 0.000
- Random Sampling, 2 repeat, 66% sample size: 0.001
- Random Sampling, 10 repeat, 5% sample size, stratified: 0.005
- Random Sampling, 10 repeat, 5% sample size: 0.006
- Random Sampling, 10 repeat, 66% sample size, stratified: 0.000
- Random Sampling, 10 repeat, 66% sample size: 0.001

The Naïve Bayes classifier did a remarkably poor job at classifying this dataset. This may be due to the the heavily correlated nature of this dataset. Many of the attributes are closely related, and thus the assumption of conditional independence does not hold. However, it is also notable that classifying with only the Tc class, and classifying without the Tc class also leads to similarly poor results. Removing outliers and imputing missing values also showed no increase in classification accuracy. Overall, the Naïve Bayes classifier did incredibly poorly.

5.5 Other Classifiers

5.5.1 Neural Network

A neural net seeks to replicate the behavior of the human brain in order to classify data. An artificial neural net consists of many nodes that are connected together with weighted links. Training a neural net consists of adapting the weights of the links to produce an accurate classification. [2]

A neural net was trained to classify this data. The default Orange neural net settings of 100 neurons in hidden layers, ReLu activation, and Adam solver were used to train the network.

Using the training data, the following results were found:

- Cross Validation, 2 folds, stratified: 0.980
- Cross Validation, 2 folds, not stratified: 0.980
- Cross Validation, 10 folds: 0.989
- Cross Validation, 10 folds, stratified: 0.989
- Cross Validation, 20 folds: 0.990
- Cross Validation, 20 folds, stratified: 0.990
- Random Sampling, 2 repeat, 66% sample size: 0.985
- Random Sampling, 10 repeat, 5% sample size: 0.692
- Random Sampling, 10 repeat, 66% sample size: 0.985

[2] The nodes are analogous to neurons, and the links are analogous to the synaptic connection between neurons.

Using the testing data, the following results were found:

- Cross Validation, 2 folds, stratified: 0.936
- Cross Validation, 2 folds, not stratified: 0.929
- Cross Validation, 10 folds: 0.972
- Cross Validation, 10 folds, stratified: 0.972
- Cross Validation, 20 folds: 0.974
- Cross Validation, 20 folds, stratified: 0.974
- Random Sampling, 2 repeat, 66% sample size, stratified: 0.957
- Random Sampling, 2 repeat, 66% sample size: 0.952
- Random Sampling, 10 repeat, 5% sample size, stratified: 0.460
- Random Sampling, 10 repeat, 5% sample size: 0.463
- Random Sampling, 10 repeat, 66% sample size, stratified: 0.956
- Random Sampling, 10 repeat, 66% sample size: 0.952

One item to note regarding the neural net classifier was that there was a significant decrease in classification accuracy with random sampling using 5% of the testing data.

There is no significant difference between the classification accuracies when outliers are removed, or missing values are imputed. Additionally, there is no significant change when classifying with only the Tc class, or when removing the Tc class entirely.

There is a fairly significant drop in classification accuracy between the training and testing data for the neural net. This may suggest some overfitting to the training data, to the detriment of the testing set accuracy.

5.5.2 Random Forest

Random forest is a class of ensemble methods specifically designed for decision tree classifiers. It combines the predictions made by multiple decision trees, where each tree is generated based on the values of an independent set of random vectors.

A representation of the Random Forest classifier using the Pythagorean Forest widget is shown below:

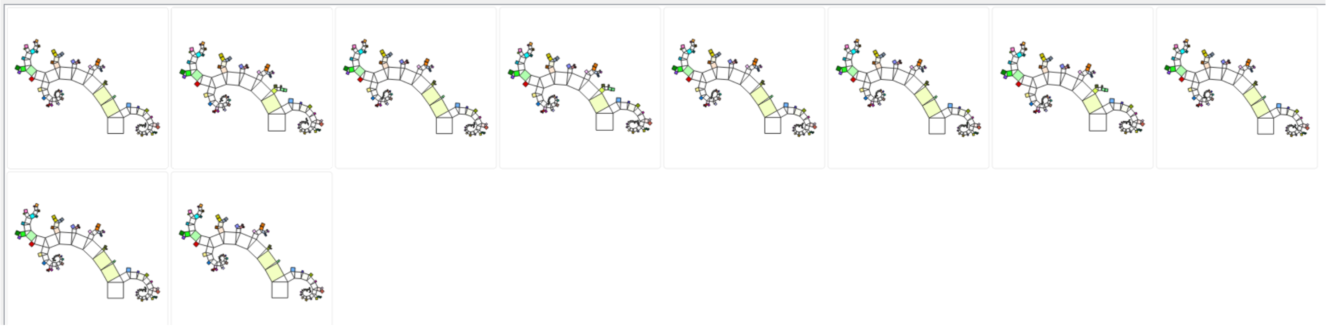


Figure 10: Random Forest Representation

We performed some tests on Random Forest to compare its performance against others. The results for training is as follows:

- Cross Validation, 2 folds, stratified: 0.952
- Cross Validation, 2 folds, not stratified: 0.954

- Cross Validation, 10 folds: 0.968
- Cross Validation, 10 folds, stratified: 0.969
- Cross Validation, 20 folds: 0.969
- Cross Validation, 20 folds, stratified: 0.969
- Random Sampling, 2 repeat, 66% sample size: 0.966
- Random Sampling, 10 repeat, 5% sample size: 0.857
- Random Sampling, 10 repeat, 66% sample size: 0.963

The results for testing is as follow:

- Cross Validation, 2 folds, stratified: 0.925
- Cross Validation, 2 folds, not stratified: 0.925
- Cross Validation, 10 folds: 0.946
- Cross Validation, 10 folds, stratified: 0.948
- Cross Validation, 20 folds: 0.949
- Cross Validation, 20 folds, stratified: 0.949
- Random Sampling, 2 repeat, 66% sample size, stratified: 0.939
- Random Sampling, 2 repeat, 66% sample size: 0.94
- Random Sampling, 10 repeat, 5% sample size, stratified: 0.805
- Random Sampling, 10 repeat, 5% sample size: 0.792
- Random Sampling, 10 repeat, 66% sample size, stratified: 0.938
- Random Sampling, 10 repeat, 66% sample size: 0.937

Overall, the random forest classifier did a fairly good job at performing accurate classifications. Similarly to other classifiers, there was no notable change in classification accuracy when classifying with only Tc and without Tc. There was also no notable change when outliers were removed, or when missing values were imputed.

6 Results

Below is a table that condenses the evaluation scores of each classifier, for both the training and testing sets. (Note that the rule-based section uses the entropy rule set).

	Decision Tree		Rule - Based		Nearest - Neighbor		Naïve Bayes		Neural Network		Random Forest	
	Training	Testing	Training	Testing	Training	Testing	Training	Testing	Training	Testing	Training	Testing
Cross Validation, 2 folds, stratified	0.988	0.994	0.999	0.997	0.929	0.893	0.000	0.000	0.980	0.936	0.952	0.925
Cross Validation, 2 folds, not stratified	0.988	0.992	0.999	0.999	0.929	0.893	0.000	0.000	0.980	0.929	0.954	0.925
Cross Validation, 10 folds	0.988	0.994	0.999	0.999	0.945	0.919	0.001	0.001	0.989	0.972	0.968	0.946
Cross Validation, 10 folds, stratified	0.988	0.994	0.999	0.999	0.945	0.919	0.001	0.001	0.989	0.972	0.969	0.948
Cross Validation, 20 folds	0.988	0.995	0.999	0.999	.946	0.920	0.001	0.003	0.990	0.974	0.969	0.949
Cross Validation, 20 folds, stratified	0.988	0.995	0.999	0.999	0.946	0.920	0.001	0.003	0.990	0.974	0.969	0.949
Random Sampling, 2 repeat, 66% sample size, stratified		0.995		0.998		0.908		0.000		0.957		0.939
Random Sampling, 2 repeat, 66% sample size	0.987	0.992	0.999	0.998	0.938	0.910	0.001	0.001	0.985	0.952	0.966	0.94
Random Sampling, 10 repeat, 5% sample size, stratified		0.986		0.986		0.745		0.005		0.460		0.805
Random Sampling, 10 repeat, 5% sample size	0.989	0.983	0.990	0.979	0.812	0.74	0.003	0.006	0.692	0.463	0.857	0.792
Random Sampling, 10 repeat, 66% sample size, stratified		0.995		0.998		0.908		0.000		0.956		0.938
Random Sampling, 10 repeat, 66% sample size	0.989	0.993	0.999	0.998	0.938	0.908	0.001	0.001	0.985	0.952	0.963	0.937
Average Cross Validation Score	0.988	.994	0.999	.999	0.940	.911	6.6E-4	.001	0.986	.960	0.964	.940
Average Random Sampling Score	0.988	.991	0.996	.993	0.896	.853	0.002	.002	0.887	.790	0.929	.892
Combined Average Score	0.988	.993	0.998	.996	0.918	.882	0.001	.002	0.937	.875	0.946	.916

7 Conclusion

We predicted that Decision Tree would be the best classifier. However from our analysis, we see that the Rule-Based classifier performs slightly better. We anticipated that Decision Tree would perform better because tree classifiers are known to handle missing values really well. However, both Decision Tree and Rule-Base performed tremendously, with the Rule-Based classifier doing slightly better.

An explanation of each of the classifiers we modeled are provided below:

1. Decision Tree

   There are many positive characteristics of decision tree induction. Some of them are applicability since it does not need prior assumptions, expressiveness, etc. The decision tree classifier has an overall score of 0.988 for training set and a score of 0.993 for test set. Some of the reason why this classifier performed well could be because it can handle missing values (there were several missing values in the dataset), handle interactions among attributes (many attributes depend on each other), and handle redundant attributes.

2. Rule-Based

   Rule-Based classifier performed slightly better than the decision tree. Some of the positive characteristics are that the expressiveness of a rule set is almost equivalent to that of a decision tree, used to produce descriptive models that are easier to interpret. The rule-based classifier has an overall score of 0.998 for training set and a score of 0.996 for test set. Rule-based classifier does not perform well with missing data however, in our dataset, the missing values mostly appear in irrelevant attributes.

3. Nearest-Neighbor Classifier

   Nearest-neighbor classifier performed substantially well. Some of the positive characteristics are that since nearest-neighbor is an unsupervised classifier, it does not require model building. It is also great with interactive attributes. The nearest-neighbor classifier has an overall score of 0.918 for training set and a score of 0.882 for test set. This classifier is susceptible to noise and bad at handling missing values. Hence, the nearest-neighbor classifier performed a bit worse than the above two classifiers, but overall it performed well.

4. Naïve Bayes Classifier

   Some of the advantages of using Naïve Bayes Classifier is that it isolates noise, and can handle missing values well. The Naïve Bayes classifier has an overall score of 0.001 for training set and a score of 0.002 for test set. The reason for such poor performance is because Naïve Bayes cannot handle correlated attributes. There are many attributes in our dataset that relates back to internal temperature of a star. Given the nature of our dataset, it is hard to isolate pure attributes because a lot of physical notions are dependent on each other. Hence, Naïve Bayes's performance was extremely weak.

5. Neural Network

   Neural networks are powerful classification models that are able to learn highly complex and nonlinear decision boundaries purely from the data. The neural network classifier has an overall score of 0.937 for training set and a score of 0.875 for test set. Neural networks cannot handle instances with missing values well. This model is also easily susceptible to overfitting that leads to poor generalization performance when there are large number of redundant attributes. These seem to be the case for a lower performance with our dataset compared with other classifiers.

6. Random Forest

   Random forest is a class of ensemble methods specifically designed for decision tree classifiers. It is designed not to overfit. If the data contain groups of correlated features of similar relevance for the output, then smaller groups are favored over larger groups. The random forest classifier has an overall score of 0.946 for training set and a score of 0.916 for test set. This reduction in accuracy from the standard decision tree may be due to the nature of how the ensemble of decision trees is created with the random forest model. This model attempts to create numerous decision trees with minimal correlation. It may be the case that in the attempt to create these uncorrelated trees, some of the trees in the ensemble result to using attributes that are not strong predictors of the final spectral class. This may occasionally cause the ensemble to come to an incorrect classification, and may be responsible for the observed decrease in classification accuracy.

We also implemented two different validation methods for each classifier.

1. Cross Validation

   We noticed that when the number of folds is increased, the accuracy level increases slightly, and there is not a significant difference between stratified and non-stratified classification accuracies. This may be due to the fact that cross validation uses the entire dataset to test; thus, there are not certain data points that are ever excluded in the performance evaluation, unlike with random sampling. Overall, the number of folds in the cross validation and whether or not the folds are stratified seems to have little influence on the reported classification accuracy. The cross validation results have little variance.

2. Random Sampling

   Random sampling seems to output slightly lower classification accuracy than the cross validation. The reason for this could be that random sampling only uses a subset of the entire testing data to compute its classification accuracy score. This means that the entire data set is not being used with random sampling to compute the classification accuracies. This means that some data is left out, which may lead to the smaller classification accuracy scores. This is particularly visible with the 5% sample size, as only a very small selection of the data is used at a time to compute the classification accuracies. Additionally, the difference between stratified and non-stratified is more pronounced with random sampling than with cross-validation. This may be due to the fact that stratified random sampling ensures that each class is represented and not skipped over as they could be with a standard random sampling. It is logical that the difference between stratified and non stratified is greater with random sampling than cross-validation, as cross validation ultimately ensures that the entire data set is used for computing the classification accuracies; thus, no class is entirely ignored with cross-validation, even if it is not stratified cross validation.

Overall, the decision tree and rule based classifiers performed very well with this dataset, with over a 98% classification accuracy. Nearest neighbor, neural network, and random forest also performed quite well, with over 85% classification accuracy. Finally, the Naive Bayes classifier performed extremely poorly, with less than 1% classification accuracy. The difference in measured performance between these different classifiers help to demonstrate how well these classifiers can adapt to a dataset with many missing values, and heavily correlated attributes.

It is also noteworthy that none of the classifiers had an appreciable difference in evaluated classification accuracy when classifying solely based on temperate, or when classifying without the temperature class. This suggests that, while temperature is a very strong indicator of the spectral type, the other attributes can be utilized in conjunction with the temperature class to build slightly more versatile classification models.

Overall, cross validation appears to be a more consistent and reliable method of computing the classification accuracy. There is more consistency in the cross validation classification accuracies when compared with the random sampling classification accuracies.

In closing, with this dataset, the rule-based and decision tree classifiers were able to create the most accurate classification models, no matter the method of evaluation.

8 References

https://en.wikipedia.org/wiki/Stellar_classification
https://en.wikipedia.org/wiki/Random_forest#Properties
https://towardsdatascience.com/understanding-random-forest-58381e0602d2
https://en.wikipedia.org/wiki/Neural_network
Pang-Ning Tan, Michael Steinbach, Anuj Karpatne, and Vipin Kumar. 2018. Introduction to Data Mining (2nd Edition)(2nd ed.). Pearson.

A Additional Data Set Information

Additional information about this dataset that was obtained from the University of Strasbourg site is provided on the following pages. Furthermore, if more detailed summary statistics are desired, please refer to the initial report submitted.

```
#

#   VizieR Astronomical Server vizier.u-strasbg.fr

#    Date: 2022-02-21T02:48:10 [V1.99+ (14-Oct-2013)]

#   In case of problem  " please report to:     cds-question@unistra.fr"

#

#

"#Coosys        J2000:  eq_FK5 J2000"

"#INFO  votable-version=1.99+ (14-Oct-2013)"

"#INFO  -ref=VIZ6212fc15162bfe"

"#INFO  -out.max=999999"

"#INFO  queryParameters=23"

#-oc.form=dec

#-out.max=999999

#-out.add=_r

#-out.add=_RAJ  _DEJ

#-sort=_r

#-order=I

#-out.src=V/137D/XHIP

#-nav=cat:V/137D&tab:{V/137D/XHIP}&key:source=V/137D/XHIP&HTTPPRM:&&-ref=VIZ6212fc15
162bfe&-out.max=50&-out.form=HTML Table&-out.add=_r&-out.add=_RAJ
_DEJ&-sort=_r&-oc.form=sexa&-c.eq=J2000&-c.r=
2&-c.u=arcmin&-c.geom=r&-order=I&-out=SpType&-out=Tc&-out=Lc&-out=[Fe/H]&-out=age&-o
ut=Npl&-ignore=Simbad=*&Simbad=Simbad&-out=B-V&-out=KMag&-out=Lum&-out=Name&-file=.&
-meta.ucd=2&-meta=1&-meta.foot=1&-usenav=1&-bmark=POST&-out.src=V/137D/XHIP

#-c.eq=J2000

#-c.r=  2

#-c.u=arcmin

#-c.geom=r
```

```
#-source=V/137D/XHIP

#-out=SpType

#-out=Tc

#-out=Lc

#-out=[Fe/H]

#-out=age

#-out=Npl

#-out=B-V

#-out=KMag

#-out=Lum

#-out=Name

#



#RESOURCE=yCat_5137

#Name: V/137D

#Title: Extended Hipparcos Compilation (XHIP) (Anderson+         2012)

"#Coosys        J2000_1991.250: eq_FK5 J2000"

"#Table V_137D_XHIP:"

#Name: V/137D/XHIP

#Title: XHIP catalog: astrometry         spectrography   space motions   exoplanets
 photometry      and references
"#Column        _RAJ2000        (F14.10)        Right ascension (FK5"
Equinox=J2000.0) at Epoch=J2000 " proper motions taken into account
[ucd=pos.eq.ra]"

"#Column        _DEJ2000        (F14.10)        Declination (FK5"
Equinox=J2000.0) at Epoch=J2000 " proper motions taken into account
[ucd=pos.eq.dec]"

"#Column        SpType  (a26)   Spectral type (MK"       HD     " or other)
```

```
[ucd=src.spType]"

"#Column        Tc      (I3)    ]0/140[? Temperature class codified (10) [NULL
integer written as an empty string]     [ucd=src.spType]"

"#Column        Lc      (I1)    [1/6]? Luminosity class codified (11) [NULL integer
written as an empty string]     [ucd=src.class.luminosity]"

"#Column        [Fe/H]  (F5.2)  ? Iron abundance        [ucd=phys.abund.Fe]"

"#Column        age     (F4.1)  ? Age"  " in billions of years  [ucd=time.age]"

"#Column        Npl     (I1)    ? Number of exoplanets (known in April 2012) [NULL
integer written as an empty string]     [ucd=meta.number]"

"#Column        B-V     (F6.3)  ? Johnson B-V color index       [ucd=phot.color"
em.opt.B        em.opt.V]

"#Column        KMag    (F6.2)  ? Absolute Magnitude K (23)     [ucd=phys.magAbs"
em.IR.K]

"#Column        Lum     (F9.2)  ? Stellar luminosity (23)
[ucd=phys.luminosity]"

"#Column        Name    (a48)   Star name(s)    [ucd=meta.id]"
```

B Additional Decision Trees

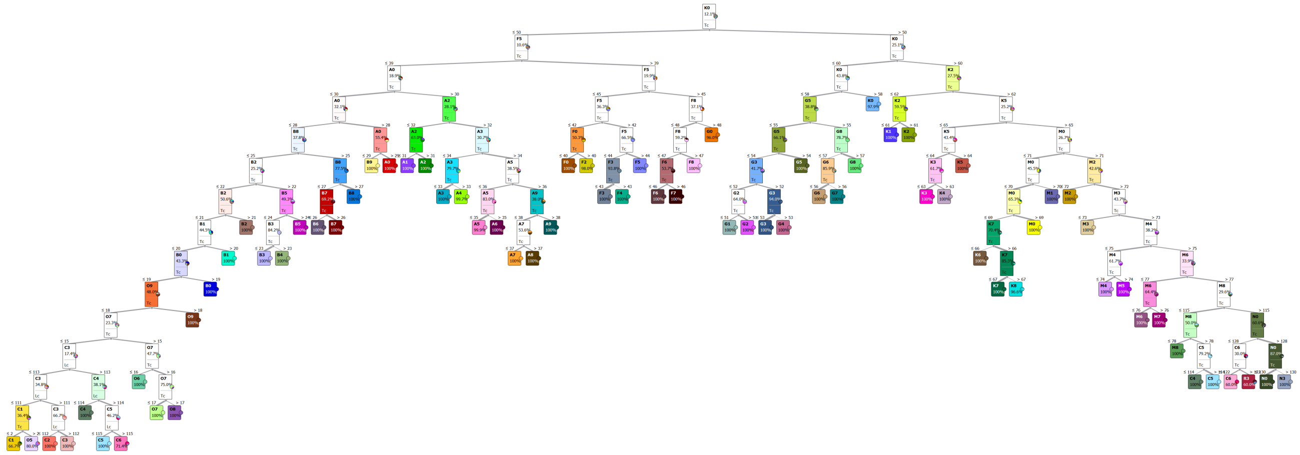


Figure 11: Decision tree made with outliers removed (143 nodes and 72 leaves)

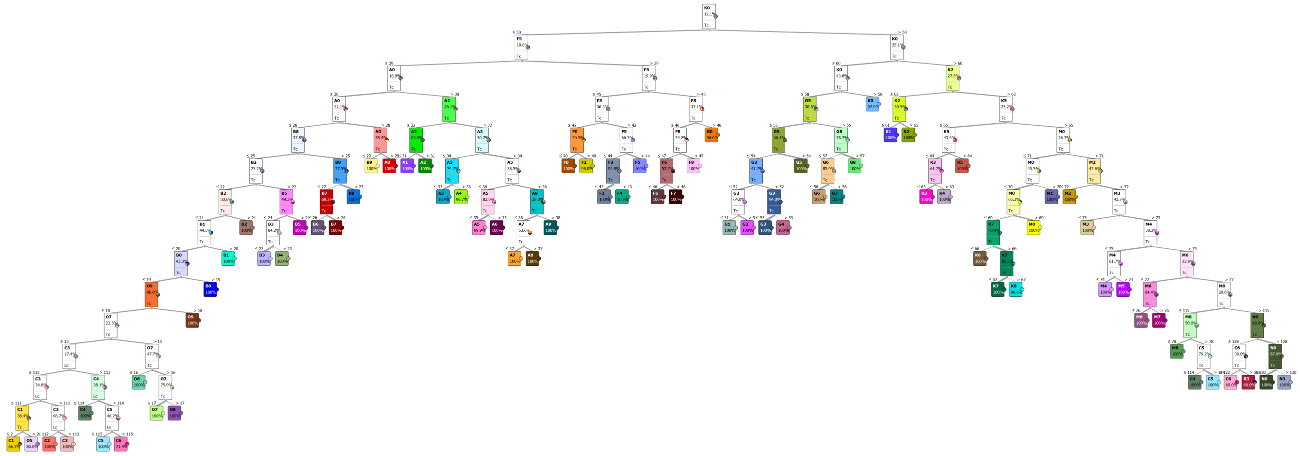


Figure 12: Decision tree made with imputed missing values (179 nodes, 90 leaves)

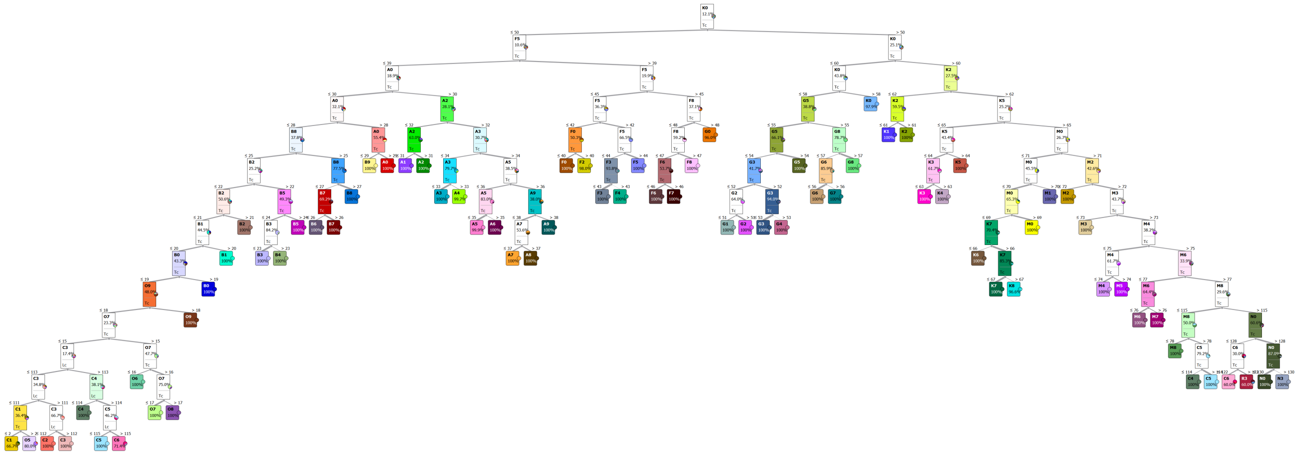
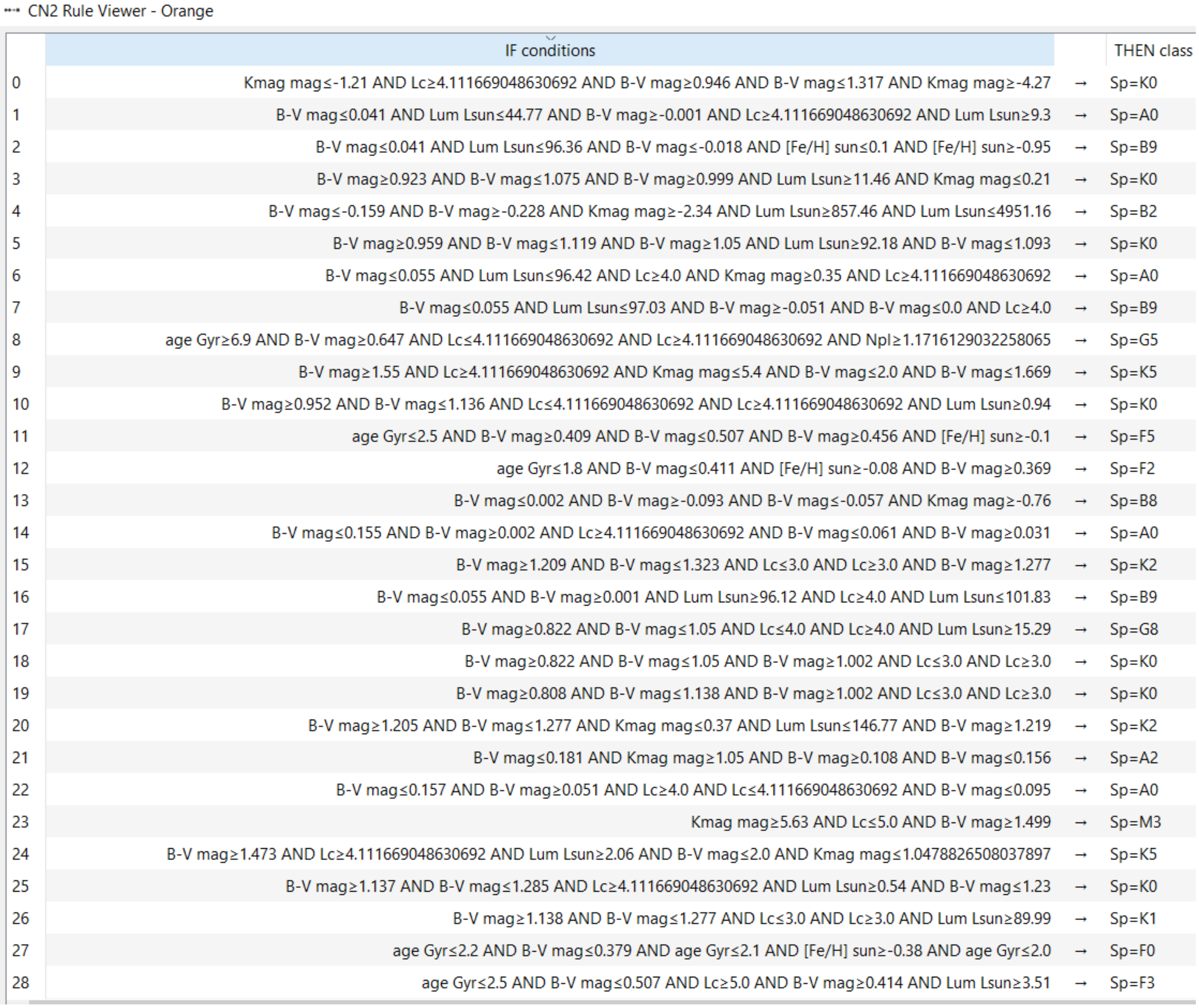


Figure 13: Decision tree made with only Tc (163 nodes, 82 leaves)

The decision tree made without Tc performed very comparably to the decision trees with all attributes, as well as the tree only using Tc. However, it is notably more complicated, with 14751 nodes and 7378 leaves. Due to this tree's complexity, it is not included in this appendix.

C Additional Rule Sets



CN2 Rule Viewer - Orange

	IF conditions		THEN class
0	Kmag mag≤-1.21 AND Lc≥4.111669048630692 AND B-V mag≥0.946 AND B-V mag≤1.317 AND Kmag mag≥-4.27	→	Sp=K0
1	B-V mag≤0.041 AND Lum Lsun≤44.77 AND B-V mag≥-0.001 AND Lc≥4.111669048630692 AND Lum Lsun≥9.3	→	Sp=A0
2	B-V mag≤0.041 AND Lum Lsun≤96.36 AND B-V mag≤-0.018 AND [Fe/H] sun≤0.1 AND [Fe/H] sun≥-0.95	→	Sp=B9
3	B-V mag≥0.923 AND B-V mag≤1.075 AND B-V mag≥0.999 AND Lum Lsun≥11.46 AND Kmag mag≤0.21	→	Sp=K0
4	B-V mag≤-0.159 AND B-V mag≥-0.228 AND Kmag mag≥-2.34 AND Lum Lsun≥857.46 AND Lum Lsun≤4951.16	→	Sp=B2
5	B-V mag≥0.959 AND B-V mag≤1.119 AND B-V mag≥1.05 AND Lum Lsun≥92.18 AND B-V mag≤1.093	→	Sp=K0
6	B-V mag≤0.055 AND Lum Lsun≤96.42 AND Lc≥4.0 AND Kmag mag≥0.35 AND Lc≥4.111669048630692	→	Sp=A0
7	B-V mag≤0.055 AND Lum Lsun≤97.03 AND B-V mag≥-0.051 AND B-V mag≤0.0 AND Lc≥4.0	→	Sp=B9
8	age Gyr≥6.9 AND B-V mag≥0.647 AND Lc≤4.111669048630692 AND Lc≥4.111669048630692 AND Npl≥1.1716129032258065	→	Sp=G5
9	B-V mag≥1.55 AND Lc≥4.111669048630692 AND Kmag mag≤5.4 AND B-V mag≤2.0 AND B-V mag≤1.669	→	Sp=K5
10	B-V mag≥0.952 AND B-V mag≤1.136 AND Lc≤4.111669048630692 AND Lc≥4.111669048630692 AND Lum Lsun≥0.94	→	Sp=K0
11	age Gyr≤2.5 AND B-V mag≥0.409 AND B-V mag≤0.507 AND B-V mag≥0.456 AND [Fe/H] sun≥-0.1	→	Sp=F5
12	age Gyr≤1.8 AND B-V mag≤0.411 AND [Fe/H] sun≥-0.08 AND B-V mag≥0.369	→	Sp=F2
13	B-V mag≤0.002 AND B-V mag≥-0.093 AND B-V mag≤-0.057 AND Kmag mag≥-0.76	→	Sp=B8
14	B-V mag≤0.155 AND B-V mag≥0.002 AND Lc≥4.111669048630692 AND B-V mag≤0.061 AND B-V mag≥0.031	→	Sp=A0
15	B-V mag≥1.209 AND B-V mag≤1.323 AND Lc≤3.0 AND Lc≥3.0 AND B-V mag≥1.277	→	Sp=K2
16	B-V mag≤0.055 AND B-V mag≥0.001 AND Lum Lsun≥96.12 AND Lc≥4.0 AND Lum Lsun≤101.83	→	Sp=B9
17	B-V mag≥0.822 AND B-V mag≤1.05 AND Lc≤4.0 AND Lc≥4.0 AND Lum Lsun≥15.29	→	Sp=G8
18	B-V mag≥0.822 AND B-V mag≤1.05 AND B-V mag≥1.002 AND Lc≤3.0 AND Lc≥3.0	→	Sp=K0
19	B-V mag≥0.808 AND B-V mag≤1.138 AND B-V mag≥1.002 AND Lc≤3.0 AND Lc≥3.0	→	Sp=K0
20	B-V mag≥1.205 AND B-V mag≤1.277 AND Kmag mag≤0.37 AND Lum Lsun≤146.77 AND B-V mag≥1.219	→	Sp=K2
21	B-V mag≤0.181 AND Kmag mag≥1.05 AND B-V mag≥0.108 AND B-V mag≤0.156	→	Sp=A2
22	B-V mag≤0.157 AND B-V mag≥0.051 AND Lc≥4.0 AND Lc≤4.111669048630692 AND B-V mag≤0.095	→	Sp=A0
23	Kmag mag≥5.63 AND Lc≤5.0 AND B-V mag≥1.499	→	Sp=M3
24	B-V mag≥1.473 AND Lc≥4.111669048630692 AND Lum Lsun≥2.06 AND B-V mag≤2.0 AND Kmag mag≤1.0478826508037897	→	Sp=K5
25	B-V mag≥1.137 AND B-V mag≤1.285 AND Lc≥4.111669048630692 AND Lum Lsun≥0.54 AND B-V mag≤1.23	→	Sp=K0
26	B-V mag≥1.138 AND B-V mag≤1.277 AND Lc≤3.0 AND Lc≥3.0 AND Lum Lsun≥89.99	→	Sp=K1
27	age Gyr≤2.2 AND B-V mag≤0.379 AND age Gyr≤2.1 AND [Fe/H] sun≥-0.38 AND age Gyr≤2.0	→	Sp=F0
28	age Gyr≤2.5 AND B-V mag≤0.507 AND Lc≥5.0 AND B-V mag≥0.414 AND Lum Lsun≥3.51	→	Sp=F3

Figure 14: Laplace Rules without Tc