Application of Decision Tree in Predicting Final Stages of Stellar Evolution

Najwa Kahani Fatima - 13523043[1]
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
[1]najwanewa1010@gmail.com, 13523043@std.stei.itb.ac.id

***Abstract*—Stellar evolution is an important field to be studied since it provides information on how the star behave and evolved throughout the time. The main source of energy of the Earth, the sun, is also a star that is in its main sequence stage. The center of the Milky Way galaxy, is a black hole which is a result of a massive star evolution. The scientist studied their evolution and create a theory on how the star evolved based on its physical properties, such as mass. From the information of their mass, final stages of stellar evolution can be predicted. This paper will examine this topic and experiment on using decision tree to predict the final stages of stellar evolution.**

***Keywords*—Decision Tree, Stellar Evolution**

I. INTRODUCTION

The growing knowledge of mathematics and computer science has given more opportunities for more advanced discoveries in the world of physics and astronomy. One of the most extraordinary milestones was the first direct image of the supermassive black hole at the center of the Messier 87 Galaxy using the Event Horizon Telescope, which used Continuous High-resolution Image Reconstruction using Patch priors (CHIRP) to successfully acquire the image [1].

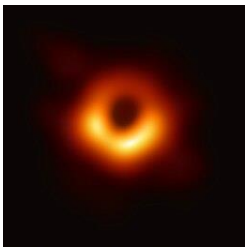

Figure 1. The first direct image of supermassive blackhole at the core of Messier 87 (retrieved from www.eso.org).

A black hole originates from a common star, just like the sun, yet has different properties. It is a result of a star evolution. Understanding stellar evolution becomes very important in order to know the history and the origin of the universe.

The fact that every star has their own properties, classes, and their own journey – from birth to die – hints scientist about the life cycle of stars. Determined by the mass, luminosity, and other physical properties, scientist could discover the common pattern and map them into a theory of stellar evolution. The theory could predict the journey of a star, its final stage, even its history before becoming the star people know today.

The process of predicting the final stage of stellar evolution can be implemented using the decision tree, a common model used in data mining, a branch of computer science. Decision tree is one of the applications of the tree theory, one of the materials in the curriculum of IF1220 Discrete Mathematics.

II. TREE THEORY

Reference [2] shows that a tree is a part of graph theory since it is made and originated from graph. The topics that will be discussed in this section are tree, rooted tree, and its properties.

A. Trees

A tree is a connected undirected graph that contains no simple circuits. A graph $G = (V, E)$ is a discrete structure consisting of vertices (V) or nodes and edges (E) that connect those vertices. Since a tree cannot contain multiple edges or loops, it is must be a simple graph. A simple graph is a graph that each edge connects two different vertices and no two edges connect the same pair of vertices.

A tree must also satisfy the theorem which there is only one unique simple path between any two of its vertices. A graph that contains no simple circuits but not necessarily connected is called forests.

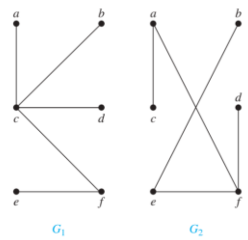

Figure 2. G_1 and G_2 are trees. A graph that contains both G_1 and G_2 is called forest (retrieved from [2])

B. Rooted Trees

In tree applications, a specific vertex of a tree is declared as the root. Directions can be assigned on each edge originated from the root because there is a unique path from the root to each vertex of the graph. A tree with a specified root produces a directed graph which called a rooted tree.

A tree might become several types of rooted tree based on the vertex that is chosen as the root. In the figure below, a tree T can be converted to a rooted tree with vertex a as the root and vertex

c as the root.

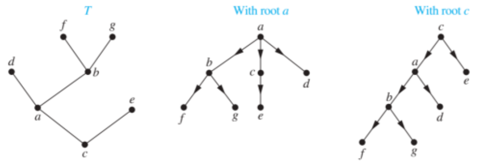

Figure 3. A tree T and rooted three with designated root a (middle) and root c (right) (retrieved from [2]).

A rooted tree T has terminologies that is similar to genealogical origins. Let v is a vertex in T except the root, then the parent of v is the unique vertex u so that there is a directed edge from u to v. If u is the parent of v, then v is the child of u. If vertices v and x have the same parent, then v and x are siblings. The ancestors of a vertex are the vertices in the path from the root to its vertex, other than the vertex itself and including the root. The vertices that have v as the ancestor is called descendants. A leaf is a vertex that has no children. Vertices that have children are internal vertices. A subtree is a subgraph that consist of the specified root and all of its descendants.

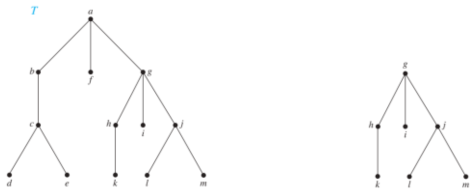

Figure 4. A rooted tree T with root a (left) and a subtree of T rooted at g (right) (retrieved from [2]).

An *m-ary* tree is a rooted tree in which every internal vertex has no more than m children. A *full m-ary* tree is a tree that its every internal vertex has exactly m children. If an *m-ary* tree has $m = 2$, then it is called binary tree.

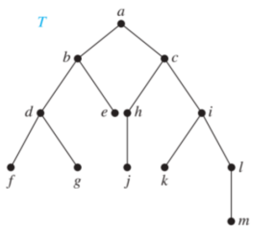

Figure 5. Binary tree (retrieved from [2]).

A rooted tree where the children of every internal vertex are ordered is called an ordered rooted tree. In a binary tree, the first child is called the left child and the second one is called the right child. If the child is not a leaf, then it is called a left subtree or a right subtree based on the order.

C. *Properties of Trees*

A tree that has n vertices have $n - 1$ edges. This theorem is proved by mathematical induction. For instance, in Figure 5, the tree T has 13 vertices and 12 edges. In figure 4, the subtree with root g has 7 vertices and 6 edges.

In another hand, a *full m-ary* tree with i internal vertices contains $n = mi + 1$ vertices. This can be proofed by the fact that each internal vertex i has m children which makes $n = mi$. The plus one is completing the number of n vertices including the root, results in $n = mi + 1$.

Another properties in a rooted tree is called level and height (h). The level of a vertex v in a rooted tree is the length of the unique path from the root to its vertex. The root is at level zero. The height or depth of a rooted tree is the maximum levels of vertices. In Figure 3, the height of the rooted tree with c as its root is 3 and the vertex d is at level 2. A rooted tree is classified as balanced tree if all leaves are at levels h or $h - 1$.

III. Decision Tree

A rooted tree can be utilized as a model to solve a problem in which a series of decisions or comparisons leads to a solution. A decision tree is one of the applications in using tree theory to solve a problem that will leads to an answer that exist in the leaves of the tree. Its internal vertex corresponds to a decision. The subtree at these vertices will become each possible outcome of the decision [2].

In data mining, a decision tree is a predictive model that can be used to represent classifier and regression models which refer to a hierarchical model of decisions and their consequences [3]. A decision tree that is used for classification task is referred as classification tree.

A classification tree is used to classify an object based into a predefined set of classes based on their attribute values or properties. This is mostly used as an exploratory technique. Below is an example of a classification tree.

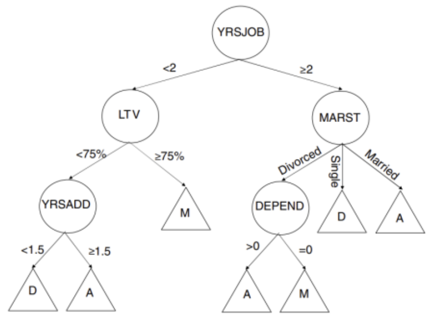

Figure 6. Example of a decision tree (classification tree) that is used to classify underwriting process of mortgage application (retrieved from [3]).

In a decision tree, every internal node splits the instance space to two or more sub-spaces according to a certain discrete function from the input properties. Meanwhile, each leaf is representing the class with the most appropriate target value.

IV. Stellar Evolution

This section explains about the theoretical evolutionary paths of stellar system with very basic stellar properties. In real condition, the evolution of each star remains mystery because when the chemical composition of a star changes with time, a new model is always computed [4].

A. The Properties

Star properties are obtained from observation. These data provide abundance information that can be derived. In the context of stellar evolution, apparent magnitude (m), color index (CI), and parallax (p) could provide important parameter to determine stellar evolution stage with the help of the Hertzsprung-Russel (HR) Diagram.

The absolute magnitude (M) of star can be calculated from the equation bellow (neglecting absorption factor), with d is distance in parsec (pc).

$$M = m + 5 - 5\log d$$

$$d\ (pc) = \frac{1}{p('')}$$

In another side, the effective temperature (T_{eff}) is formulized as follows.

$$\log T_{eff} = 3.988 - 0.881 \times CI + 0.111 \times CI^2$$

The luminosity (L) in the sun luminosity unit can be derived from the following equation, with $M_{sun} = {\sim}4.83$.

$$\frac{L}{L_{sun}} = 10^{(M - M_{sun})/2.5}$$

Lastly, one of the most important features in stellar evolution, star mass, can be calculated as below, with L is luminosity and M is mass.

$$L \propto M^{\alpha}$$

$$T_{eff} > 30{,}000\ K\ ;\ \alpha = 3.5$$
$$T_{eff} > 10{,}000\ K\ ;\ \alpha = 4$$
$$T_{eff} > 6{,}000\ K\ ;\ \alpha = 4.5$$
$$T_{eff} < 6{,}000\ K\ ;\ \alpha = 5$$

In order to explore the properties of star that is related to its evolution, the evolutionary time scale is discussed. However, in every stage and phase of the evolution, different properties are calculated. The three important basic evolutionary time scales are the nuclear timescale t_n, the thermal time scale t_t and the dynamical (freefall) time scale t_d [4].

The Nuclear Time Scale

It is the time in which a star radiates all the available energy by nuclear reactions. In another words, it is the time in which all available hydrogen is turned into helium. Based on the theoretical considerations and computations, only 10% of the total mass (M) of hydrogen of a star can be turned before more rapid evolutionary mechanism occurs. In addition, only 0.7% of the rest mass is turned into energy by hydrogen burning. Therefore, the nuclear time scale is formulized as below, where c is the speed of light and L is the luminosity of the star.

$$t_n \approx \frac{0.007 \times 0.1\ Mc^2}{L}\ s$$

The Thermal Time Scale

This time scale is calculated when the nuclear energy production were suddenly turned off. It is the time in which a star would radiate all its thermal energy and is derived from the virial theorem. The thermal timescale is roughly estimated as below, where G is the constant of gravity and R is the stellar radius.

$$t_t = \frac{0.5\ GM^2/R}{L}\ s$$

The Dynamical Time Scale

This is the time scale the star would take to collapse if the pressure that support it against gravity were suddenly removed. It is calculated from the half period in Kepler's third law with the semimajor axis of the particle orbit corresponds to half of the stellar radius.

$$t_d = \frac{2\pi}{2}\sqrt{\frac{\left(\frac{R}{2}\right)^3}{Gm}} \approx \sqrt{\frac{R^3}{GM}}\ s$$

B. The Birth of Stars

The evolution of stars starts from the formation of a protostar, which made of the process of a gravitational collapse of condensations in the interstellar medium which occurs on the dynamical time scale. After the collapse, it will release gravitational potential energy and it is transformed into thermal energy of the gas and into radiation. It results in the increasing density and pressure near the center of the cloud. This is called a protostar which consist mainly of hydrogen in molecular form. In HR Diagram, the protostar will come up and settles at a point based on their initial mass on the Hayashi Track.

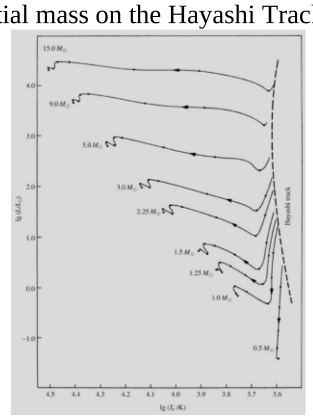

Figure 7. The Star Early Stages on Hertzsprung-Russel Diagram (retrieved from [4])

At temperature $1.8\ K$, the molecules start to dissociated into atoms. Then at $10^4\ K$ the hydrogen is ionized, then the helium, and lastly at $10^5\ K$, the gas becomes completely ionized. The contraction of the protostar stops when the gas is turned into plasma. After reaching the hydrostatic equilibrium, the star evolution will take place in thermal time scale.

As the star passing the Hayashi track, the center temperature is increasing, resulting in the energy transfer through radiation. When the central temperature is high enough, the star would start the nuclear reaction from hydrogen burning. It marks the start of the main sequence phase.

C. The Main Sequence Phase

The main sequence phase is the longest part of the life of stars since it takes place on nuclear time scale. It acts on its time scale because the only source of stellar energy in this phase is released by the burning of hydrogen in the core.

For instance, a solar mass star, the main sequence will last for about 10,000 million year, meanwhile more massive star would evolve more rapidly. This happens because more massive star radiates much more power. However, when the star is too massive, the force of gravity cannot resist the radiation pressure and they cannot be formed during contraction phase. In the other hand, the star that pass below $0.08\,M_{sun}$ will never be hot enough to begin hydrogen burning, which called the brown dwarf.

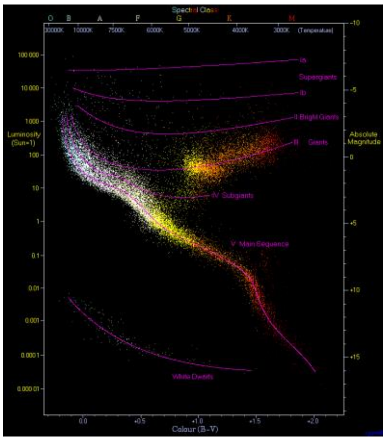

Figure 8. The Hertzsprung-Russel Diagram (retrieved from atlasoftheuniverse.com)

The star which located in the upper main sequence in the HR Diagram are very massive, resulting in the very high central temperature. Therefore, the CNO cycle can operate. Since the outward energy flux becomes very large, the radiative transport cannot be maintained, thus they have a convective core, i.e. the energy is transported by material motions. However, the energy outside of the core is still carried by radiation.

On the lower main sequence, the central temperature is lower. Thus, the energy is generated by pp chain and the core remains radiative. Stars with masses between $0.08 - 0.26\,M_{sun}$ have very simple evolution and finally evolved by contracting and become white dwarfs.

D. The Giant Phase

When the hydrogen is exhausted at the center, the main sequence phase ends and the giant phase begins. In the giant phase, the star stars to burn the helium core. The evolution in this phase depends strongly on the stellar mass.

For the low-mass stars ($\leq 2.3\,M_{sun}$), the density become so high eventually which results in the increasing core temperature. If the mass is higher than $0.26\,M_{sun}$, the helium will burn to carbon in triple alpha process. Only a few seconds after the ignition of helium, the helium flash occurs, an explosion that indicate that the star settles into a new state in which the star is steadily burning to carbon. In HR Diagram, this is located in the horizontal giant branch.

In intermediate-mass star ($2.3\,M_{sun} \leq M \leq 8\,M_{sun}$, the helium burning can set in non-catastrophically as the central regions contract. The star would move from red giant branch towards blue giant area, then back towards the Hayashi track again. Low and intermediate-mass giants ($M \leq 8 M_{sun}$) are never hot enough to ignite carbon burning, thus remains as carbon-oxygen white dwarfs.

For stars with mass around ($10\,M_{sun}$), the carbon or oxygen may ignite explosively through the carbon and oxygen flash. This phenomenon is very powerful that may make the stars explode as supernova. For higher-mass star ($\geq 15\,M_{sun}$), the burning will continue to silicon burning and other burning process that leads iron which results in a sequence of layers with differing composition.

E. Final Stages

The final endpoint of stellar evolution relies on the mass and central density in final equilibrium when the body has cooled. There are two maxima mass in this stage. First the Chandrasekar mass, $M_{Ch} \approx 1.2 - 1.4\,M_{sun}$ and the Oppenheimer-Volkoff mass, $M_{OV} \approx 1.5 - 2\,M_{sun}$. For star with mass less than M_{Ch}, the star will become a white dwarf, then black dwarf when it reaches final equilibrium. If the mass is larger than M_{Ch} but smaller than M_{OV}, it will become a completely degenerate neutron star. A massive star larger than M_{OV} will go on contracting past the density of a neutron star and forming a black hole. The only endpoint of stellar evolution based on theory are two stable states: a black hole or explosive disruption.

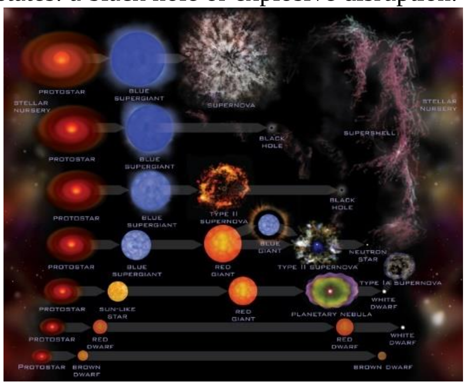

Figure 9. Stellar Evolution Diagram (retrieved from chandra.harvard.edu)

V. Proposed Method

Predicting the final stages of stellar evolution can be carried out using the concept of Decision Tree. In this experiment, the properties that are needed for the input are the observational properties such apparent magnitude, parallax, and color index. The program will classify the star class and predict the final stage of the stellar evolution. The program will derive the essential element to predict the final stages of stellar evolution, which is the star mass. The constraint of this experiment is at the parameter to predict the final stage, in which only uses stellar mass because it is the fundamental properties that effect the other properties such seen in part IV.A [5]. This experiment also designed for non-binary stars. Below is the decision path of the program.

```
Current Stage Classification:
|-- mass < 0.08
|   |-- "Brown Dwarf"
|-- mass >= 0.08
    |-- temp > 30000
    |   |-- luminosity > 1000
    |       |-- "O Star"
    |-- temp > 10000
    |   |-- luminosity > 100
    |       |-- "B Star"
    |-- temp > 7500
    |   |-- "A Star"
    |-- temp > 6000
    |   |-- "F Star"
    |-- temp > 5000
    |   |-- "G Star"
```

```
        |-- temp > 3500
        |    |-- "K Star"
        |-- temp <= 3500
            |-- luminosity > 1000
            |    |-- "Red Supergiant"
            |-- luminosity > 100
            |    |-- "Red Giant"
            |-- luminosity < 0.01 and temp > 4000
            |    |-- "White Dwarf"
            |-- else
                |-- "M Star"

Final Stage Classification:
|-- mass < 0.08
|    |-- "Brown Dwarf"
|-- mass >= 0.08
    |-- mass < 8
    |    |-- "White Dwarf"
    |-- mass < 25
    |    |-- "Neutron Star"
    |-- mass >= 25
        |-- "Black Hole"
```

VI. PROGRAM EXPERIMENT

A. Program

The program uses Python with several libraries such as NumPy, pandas, scikit-learn, and data classes. The sun properties will be the fundamental reference for the calculation. The experiment is carried out as follows.

1. Importing fundamental libraries

```python
import numpy as np
import pandas as pd
from sklearn.tree import DecisionTreeClassifier, export_text
from dataclasses import dataclass
```

2. Creating class for star parameters

```python
@dataclass
class StarParameters:
    apparent_magnitude: float
    distance_parsecs: float
    b_v_color: float
```

3. Creating decision tree classifier

a. Initialization

```python
class StellarDecisionTreeClassifier:
    def __init__(self):
        # Solar reference values
        self.solar_luminosity = 1.0
        self.solar_temperature = 5778
        self.solar_absolute_magnitude = 4.75

        # Train the decision trees
        self.current_stage_classifier = self._train_current_stage_tree()
        self.final_stage_classifier = self._train_final_stage_tree()
```

```python
    def _generate_training_data(self, n_samples=1000):
        # Random values
        temperatures = np.random.uniform(2000, 50000, n_samples)
        luminosities = np.exp(np.random.uniform(-5, 15, n_samples))
        masses = np.random.uniform(0.08, 150, n_samples)

        X = np.column_stack([temperatures, luminosities, masses])

        current_stages = []
        final_stages = []

        for temp, lum, mass in zip(temperatures, luminosities, masses):
            # Current stage classification
            if mass < 0.08:
                current_stages.append("Brown Dwarf")
            elif temp > 30000 and lum > 1000:
                current_stages.append("O Star")
            elif temp > 10000 and lum > 100:
                current_stages.append("B Star")
            elif temp > 7500:
                current_stages.append("A Star")
            elif temp > 6000:
                current_stages.append("F Star")
            elif temp > 5000:
                current_stages.append("G Star")
            elif temp > 3500:
                current_stages.append("K Star")
            elif lum > 1000 and temp < 5000:
                current_stages.append("Red Supergiant")
            elif lum > 100 and temp < 5000:
                current_stages.append("Red Giant")
            elif lum < 0.01 and temp > 4000:
                current_stages.append("White Dwarf")
            else:
                current_stages.append("M Star")

            # Final stage classification
            if mass < 0.08:
                final_stages.append("Brown Dwarf")
            elif mass < 8:
                final_stages.append("White Dwarf")
            elif mass < 25:
                final_stages.append("Neutron Star")
            else:
                final_stages.append("Black Hole")

        return X, current_stages, final_stages
```

```python
    def _train_current_stage_tree(self):
        X, current_stages, _ = self._generate_training_data()
        clf = DecisionTreeClassifier(max_depth=6, random_state=42)
        clf.fit(X, current_stages)
        return clf

    def _train_final_stage_tree(self):
        X, _, final_stages = self._generate_training_data()
        clf = DecisionTreeClassifier(max_depth=4, random_state=42)
        clf.fit(X, final_stages)
        return clf
```

b. Calculate stellar parameters

```python
def calculate_stellar_parameters(self, star: StarParameters):
        # Absolute magnitude
        distance_modulus = 5 * np.log10(star.distance_in_pc) - 5
        absolute_magnitude = star.apparent_magnitude - distance_modulus

        # Temperature from color index
        log_temp = 3.988 - 0.881 * (star.color_index) + 0.111 * (star.color_index)**2
        temperature = 10**log_temp

        # Luminosity
        luminosity = 10**((self.solar_absolute_magnitude - absolute_magnitude)/2.5)

        # Mass
        if temperature > 30000:
            mass = luminosity**(1/3.5)
        elif temperature > 10000:
            mass = luminosity**(1/4)
        elif temperature > 6000:
            mass = luminosity**(1/4.5)
        else:
            mass = luminosity**(1/5)

        return temperature, luminosity, mass
```

c. Predict final stage

```python
def predict_stages(self, star: StarParameters):
        # Stellar parameters
        temperature, luminosity, mass = self.calculate_stellar_parameters(star)

        # Feature array
        X_pred = np.array([[temperature, luminosity, mass]])

        # Make predictions
        current_stage = self.current_stage_classifier.predict(X_pred)[0]
        final_stage = self.final_stage_classifier.predict(X_pred)[0]

        # Decision paths
        current_stage_path = export_text(self.current_stage_classifier,
                                         feature_names=['Temperature', 'Luminosity', 'Mass'])
        final_stage_path = export_text(self.final_stage_classifier,
                                       feature_names=['Temperature', 'Luminosity', 'Mass'])

        # Feature importances
        feature_importance = pd.DataFrame({
            'Feature': ['Temperature', 'Luminosity', 'Mass'],
            'Current Stage Importance': self.current_stage_classifier.feature_importances_,
            'Final Stage Importance': self.final_stage_classifier.feature_importances_
        })

        return {
            'parameters': {
                'temperature': temperature,
                'luminosity': luminosity,
                'mass': mass
            },
            'predictions': {
                'current_stage': current_stage,
                'final_stage': final_stage
            },
            'decision_paths': {
                'current_stage_path': current_stage_path,
                'final_stage_path': final_stage_path
            },
            'feature_importance': feature_importance
        }
```

4. Main program

```python
def main():
    # Receive input
    m_input =  float(input("Input star apparent magnitude: "))
    d_input = float(input("Input star distance (parsec): "))
    ci_input = float(input("Input star color index: "))
    star = StarParameters(
        apparent_magnitude = m_input,
        distance_in_pc = d_input,
        color_index = ci_input
    )

    classifier = StellarDecisionTreeClassifier()

    print(f"\nPredicting Stellar Final Stage Evolution Program")
    print("=" * 48)

    result = classifier.predict_stages(star)

    print("\nCalculated Parameters:")
    for param, value in result['parameters'].items():
        print(f"  {param}: {value:.2f}")

    print("\nPredicted Stages:")
    print(f"  Current Stage: {result['predictions']['current_stage']}")
    print(f"  Final Stage: {result['predictions']['final_stage']}")

    print("\nFeature Importance:")
    print(result['feature_importance'])
```

B. Experiment

The program will be experimented for the stars below.

Star	Apparent Magnitude	Color Index	Distance (parsec)
Star 1	9.54	1.57	1.83
Star 2	11.05	1.82	1.30
Star 3	6.5	2.00	1200
Star 4	4.55	−0.10	2300
Star 5	0.50	1.85	152

Below is the program result of the experiment for Star 1 and the table for the remaining inputs.

```
Star #1
--------------------

Calculated Parameters:
  temperature: 755.85
  luminosity: 0.00
  mass: 0.21

Predicted Stages:
  Current Stage: M Star
  Final Stage: White Dwarf

Feature Importance:
       Feature  Current Stage Importance  Final Stage Importance
0  Temperature                  0.514138                     0.0
1   Luminosity                  0.485862                     0.0
2         Mass                  0.000000                     1.0
```

Star	Current Stage	Final Stage
Star 1	M Star	White Dwarf
Star 2	M Star	White Dwarf
Star 3	Red Supergiant	White Dwarf
Star 4	B Star	Neutron Star
Star 5	Red Supergiant	White Dward

VII. CONCLUSION

Decision tree can be applied in the field of astronomy, particularly in stellar evolution since the prediction process is applicable and similar to tree. However, there are still many further improvements in order to build a more accurate prediction by considering other physical properties. Binary stars can also be considered in further experiment.

VIII. ACKNOWLEDGMENT

The author would like to thank God for the guidance throughout the process of learning and writing this paper. The author would also like to deliver biggest gratitude to IF1220 Discrete Mathematics lecturers for sharing and guiding the student in learning the materials throughout the semester. The author would also like to thank to family and friends who have accompanied the journey of joy and sorrow since the start of the author's university journey.

REFERENCES

[1] Bouman, Katherine L.; Johnson, Michael D.; Zoran, Daniel; Fish, Vincent L.; Doeleman, Sheperd S.; Freeman, William T. (2016). "Computational Imaging for VLBI Image Reconstruction". 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). pp. 913–922. arXiv:1512.01413. doi:10.1109/CVPR.2016.105. hdl:1721.1/103077. ISBN 978-1-4673-8851-1. S2CID 9085016.Xxx

[2] Rosen, Kenneth. (2007). Discrete Mathematics and Its Applications. 674-819.

[3] Rokach, Lior.; Maimon, Oded. (2014). Data Mining with Decision Trees: Theory and Applications. 1-16.

[4] Karttunen, H., Kröger, P., Oja, H., Poutanen, M., & Donner, KJ (2007). Fundamental astronomy. (5th ed ed.) Springer.

[5] Stellar Evolution. Accessed on January, 6th, 2025 from www.aavso.org/stellar-evolution

[6] Sekiguchi, M., Fukugita, M. (2000). A Study of the B−V Color-Temperature Relation. DOI 10.1086/301490

[7] Degl'Innocenti, S. (2016). Intoduction to Stellar Evolution. J. Phys.: Conf. Ser. 703 012002

STATEMENT

I hereby declare that the paper I wrote is my own writing, not an adaptation or translation of someone else's paper, and not plagiarized.

Bandung, 5 January 2025

Najwa Kahani Fatima - 13523043