

Team Control Number

23479442

Problem Chosen

E

2023

**IMMC Greater China, Winter
Summary Sheet**

Distinguishing related animal species can be a hard but essential task. Biologists often rely upon sets of characteristics to do classifications. In this problem, we are provided with the data of 564 lizards divided into eight species and all belonged to genus *Darevskia*. The chart provided contains the data of twenty three characteristics, some are pholidosis characteristics, and others are morphometric characteristics. We need to use the data and develop criteria for the biologists to classify these kinds of lizards.

Our major model is based on **logistic regression**. By fitting the data into a sigmoid function, we are able to provide biologists Sigmoid functions to calculate the probability that the lizard belongs to the target species. This method has a huge advantage since it's pretty easy for biologists to calculate on a graphing calculator and classify the lizards.

In problem 1, we are asked to develop a criterion to differentiate the #5 lizard from other species only with FPNr data. This is a binary classification problem and we use training samples to apply our **logistic regression model** and obtain the **one dimensional sigmoid function**. By simply compare the result to 0.5, the biologist can determine the species the lizard belongs to.

In problem 2, we need to differentiate the #5 lizard from other species using two variables out of the measured morphometric and pholidosis characteristics. First, we need to determine the variable pair we are going to use. We do an exhaustive search and examine every possible variable pair. At last, we find out that SCGr and PL together produce the highest mean accuracy. We use these two variables to apply our logistic regression model, which now extends to a **two dimensional sigmoid function**. This time we have a **line decision boundary**.

In problem 3, we need to determine the lizards' sex regardless of their species types and consider their morphometric and pholidosis characteristics. We are implied that these variables might have some linear correlations. Our first method is to do an exhaustive search over all possible ratios of two random variables and use the ratio as the variable of the sigmoid function, and as we did before, we use logistic regression to find the coefficients and the intercept. This method already yields a good accuracy rate. Our second method considers all the twenty-three characteristics to achieve a higher accuracy rate. We use **multi-dimensional logistic regression**.

In problem 4, we use the similar method we used in problem 3. This time we get a new set of training samples and test samples for each subgroup listed in the problem. The set only contains the data of the species in the specific subgroup. Controlling the set of samples can help us have a more accurate result. Then, as before, we apply our multi-dimensional logistic regression model to get the sigmoid functions. We introduce the concept of **multinomial logistic regression** while classifying subgroup c.

In problem 5, we are required to have a criterion that can predict both the lizards' sex and species. Our first method is to develop two **multi-dimensional logistic regression models**,

one for sex, and the other for species. Our second method is to use **decision tree model**. To improve the accuracy of the second method, we use **random forest model**.

Contents

1	Introduction	1
1.1	Background	1
1.2	Problem Restatement	1
1.3	Assumption and Justification	1
2	The Development of Models	2
2.1	Variables	2
2.2	Variable Weight Consideration	2
2.3	Data Visualization	4
2.4	Model Overview	4
3	Solution	6
3.1	Problem 1	6
3.1.1	One Dimensional Logistic Regression	6
3.1.2	Results	6
3.2	Problem 2	7
3.2.1	Two Dimensional Logistic Regression	7
3.2.2	Results	8
3.3	Problem 3	8
3.3.1	Multi-Dimensional Logistic Regression: method 1	8
3.3.2	Results: method 1	9
3.3.3	Results: method 2	10
3.4	Problem 4	10
3.4.1	Multi-dimensional Logistic Regression	10
3.4.2	Multinomial Logistic Regression	11
3.5	Problem 5	13
3.5.1	Logistic Regression	13
3.5.2	Results: method 1	15
3.5.3	Decision Tree	15
3.5.4	Results: method 2	17
3.5.5	Random Forest	17

3.5.6	Results: method 3	18
3.5.7	Discussion of the results	19
4	Strengths and Weaknesses	19
4.1	Strengths	19
4.2	Weaknesses	19
5	Conclusion	19
6	Bibliography	21
7	Appendix	21
7.1	PCA Coefficient	21
7.2	Training Samples and Test Samples for each question	21
7.3	Complete Decision Tree	35
7.4	Complete Random Forest for Species	37
7.5	Complete Random Forest for Gender	43

1 Introduction

1.1 Background

Sorting an animal into different species is crucial to Biology study. However, many closely related animal species are hard to distinguish from each other by their exterior appearance, and it is difficult to find some “key features”. Therefore, we need to develop criteria to sort different species mathematically with only simple measurements. In this report, we take lizards as an example and we developed a series of criteria to sort different species of lizards as accurately as possible with the given data.

1.2 Problem Restatement

We are provided with the measured morphometric and pholidosis characteristics of the lizards, and we need to differentiate the species and sex of the lizards by processing the data.

- **Problem 1:**Build a criterion to differentiate the #5 lizard from other species only with FPNr data.
- **Problem 2:**Build a criterion to differentiate the #5 lizard from all the other species using two variables out of the measured morphometric and pholidosis characteristics.
- **Problem 3:**Build a criterion to predict the lizard’s sex regardless of their biological species based on the morphometric and pholidosis characteristics and their ratios.
- **Problem 4:**Build a set of criteria to sort lizards from subgroups below: #6 and #7; #1 and #2; #3, #4, and #5.
- **Problem 5:**Build a criterion or a set of criteria to predict the lizards’ species and sex in their entire population.

1.3 Assumption and Justification

AssumptionThe data provided in the question is representative enough.

Justification: We need representative data to find a universal criterion. If the data has limitations, the criterion couldn’t be used, leading to a huge bias between the theory and reality.

2 The Development of Models

2.1 Variables

Symbol	Description	Unit
z	Original data	N/A
z'	Processed data	N/A
C	Covariance matrix	N/A
μ	Mean of the training sample	N/A
s	The standard deviation of the training sample	N/A
bl	boundary line	N/A
$P(x)$	The index for whether the lizard belongs to the target or not	N/A
$\sigma(x)$	The sigmoid function, used to obtain the possibility that the lizard belongs to the target from the data	N/A
$f(x)$	The linear equation calculated for the Sigmoid function	N/A
x_i	When $i=1,2,3,4,\dots,23$, in succession, x_i represents the scaled values of a lizard's MBS, VSN, CSN, GSN, FPNr, SDLr, SCSr, SCGr, SMr, MTr, PA, PTMr, aNDsr, SVL, TRL, HL, PL, ESD, HW, HH, MO, FFL, HFL	N/A
w_i	The coefficient of x_i	N/A
G	The Gini Impurity	N/A

2.2 Variable Weight Consideration

To calculate the weight of the 23 variables given, we perform the principal component analysis.

Principal component analysis generates the number of principal components equal as the total number of variables. Each principal component is a linear combination of 23 variables and a weight will be given to each principal component based on its contribution on the variation of data.

To perform PCA, we first need to standardize the data using the following formula:

$$z' = \frac{z - \mu}{s} \quad (2.1)$$

where z is the processed data, x is the training sample, μ is the mean of the training sample and s is the standard deviation of the training sample. We minus the mean of the training sample, and thus center the data in the coordinate system. By dividing the standard deviation of the training sample, we can scale the data into a unified base unit.

After standardizing the data, we need to calculate the covariance matrix of the data. We name this matrix as C . Here, $C_{i,j}$ represents the covariance of variable i and variable j . The covariance matrix of the data is shown in the appendix.

Then, we calculate all the 23 eigenvalues of the covariance matrix, since there are a total of 23 variables in the data to describe a lizard: $\lambda_1 \geq \lambda_2 \geq \lambda_3 \geq \dots \geq \lambda_{23}$, and their corresponding

eigenvectors $u_1, u_2, u_3..u_{23}$. We define $u_{i,j}$ as the j^{th} component of u_i . The 23 principal components are therefore:

$$\begin{cases} PC1 = \sum_{j=1}^{23} u_{1,j}x_j \\ PC2 = \sum_{j=1}^{23} u_{2,j}x_j \\ \dots \\ PC23 = \sum_{j=1}^{23} u_{23,j}x_j \end{cases} \quad (2.2)$$

Here, the eigenvectors $u_1, u_2, u_3..u_{23}$ are shown in the appendix.

The weight of each PC is calculated by the formula:

$$W_{PCi} = \frac{\lambda_i}{\sum_{j=1}^{23} \lambda_j} \quad (2.3)$$

Therefore the score of each variable S_{xi} to the variance of the whole data is calculated by the formula:

$$S_{xi} = \sum_{j=1}^{23} |W_{PCj} \cdot u_{j,i}| \quad (2.4)$$

The calculated weight of each variable W_{xi} is thus:

$$W_{xi} = \frac{S_{xi}}{\sum_{j=1}^{23} S_{xj}} \quad (2.5)$$

The calculated weight of each variable using the given data set is shown below:

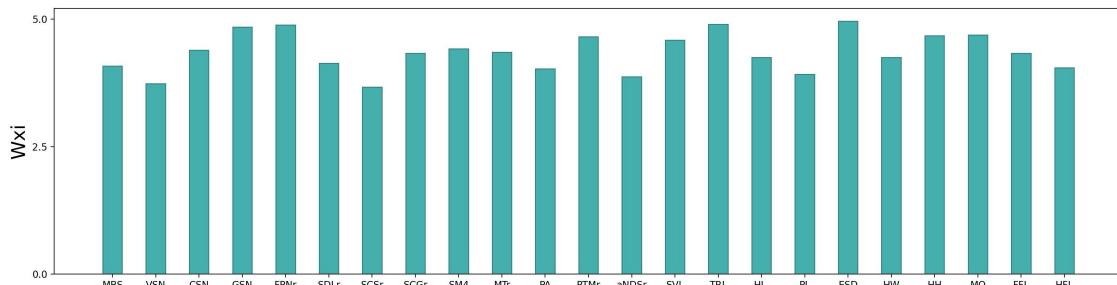


Figure 1: Variable weight

From the graph, it can be seen that the weight of each variable is almost evenly distributed. The variable that contributes the most is ESD, with a weight of 0.0492, and the variable that contributes the least is SCSr, with a weight of 0.0370.

This result sets two guidelines for our following modeling: 1): Because both the range and the variance of the weight of each variable is too small, we decide NOT to consider the weight of the variable in the following modeling. 2): Because each variable contributes almost an equal amount to the final data variance, it is hard to discard any of these variables without lowering the precision. Therefore, we choose NOT to consider removing variables that contribute very little to the result and use the set of 23 variables for modeling.

2.3 Data Visualization

By using the principal component analysis, as discussed in the previous section, we can represent the 23-dimensional data by using the first two principal components, which contribute to over 46 percents of the variation of the data.

Below is the graph of the data set drawn using the first two principal components:

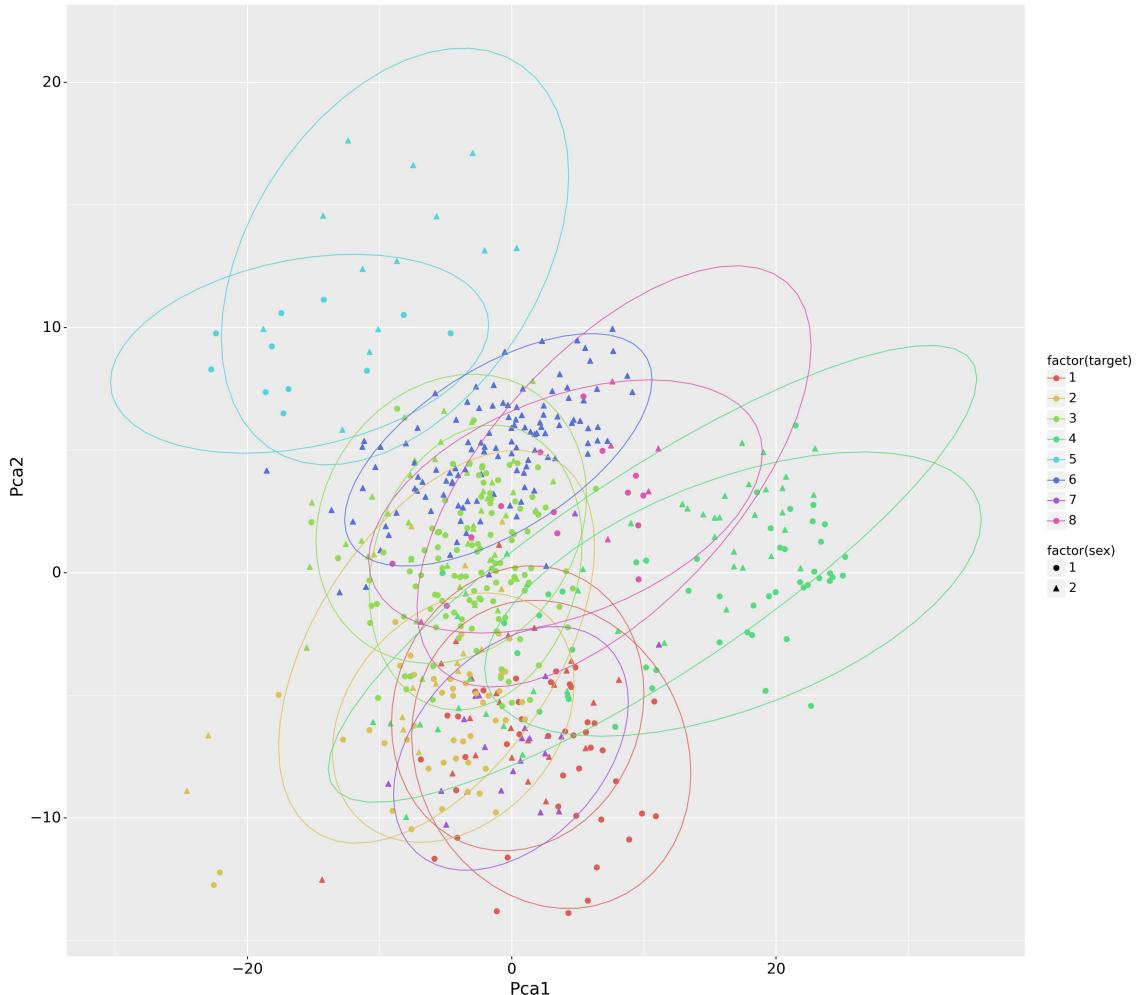


Figure 2: Data set drawn using the first two principal components

2.4 Model Overview

Our model is based on logistic regression. In these problems, logistic regression has a huge advantage since it can sort the data by fitting the model into a Sigmoid Function. By providing the biologists a Sigmoid function, they can calculate the probability and easily categorize the lizards.

We divide the original data into training samples and test samples by a racial of six to four.

In this model, first, as we did in the previous part, we need to fit and scale the data to make the calculation easier.

Then, we have $P(x)$ to show the result of the classification. For data that belongs to the target, $P(x) = 1$ and for data that does not belong to the target, $P(x) = 0$, and this could be

shown as below:

$$P(x) = \begin{cases} 1, & \text{if the lizard belongs to the target} \\ 0, & \text{if the lizard does not belong to the target} \end{cases} \quad (2.6)$$

Thus we can plot the training samples according to their data values and $P(x)$ values in a coordinate system. After that, we use a Sigmoid function $\sigma(x)$ to fit the graph of training samples and obtain $f(x)$. The Sigmoid function describes the possibility that the data belongs to the target, and it could be shown as:

$$\sigma(x) = \frac{1}{1 + e^{-f(x)}}, \text{ where } f(x) = b + \sum_{i=1}^n w_i x_i \quad (2.7)$$

This function has a threshold of $\sigma(x) = 0.5$, which means that if $\sigma(x) \geq 0.5$, then the data belongs to the target; if $\sigma(x) < 0.5$, then the data does not belong to the target.

In order to measure how the model can predict data, we run the model on test samples and we divide the outcome into four categories: True Positive(TP), False Positive(FP), True Negative(TN), and False Negative(FN). We have two methods to compare different models, accuracy, and f1-score.

Accuracy for a model is easier to interpret. It is calculated as:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FN + FP} \quad (2.8)$$

The F1-Score for a model is a bit more complicated. We first need to calculate the Precision and Recall for the model.

$$\text{Precision} = \frac{TP}{TP + FP} = \frac{TP}{\text{Total Predicted Positive}} \quad (2.9)$$

$$\text{Recall} = \frac{TP}{TP + FN} = \frac{TP}{\text{Total Actual Positive}} \quad (2.10)$$

Then we can calculate the F1-Score for the model as:

$$\text{F1 - Score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (2.11)$$

The accuracy is easier to interpret, but it does not take the distribution of the data into account. F1-Score is harder to interpret, but it takes the distribution of the data into account.

Besides logistic regression, we further use other methods, such as decision tree and random forest in problem five. We will discuss them in detail in our next part.

3 Solution

3.1 Problem 1

3.1.1 One Dimensional Logistic Regression

In problem 1, we are asked to differentiate lizards of species #5 from all other species and consider or use only the FPNr data. This is a binary classification problem and we can just simply apply our basic model, logistic regression, to solve this problem.

Since for this single task, we just need to consider the value of x_5 , which is the scaled value of FPNr, we can simplify our model to the following form:

$$f(x) = ax_5 + b$$

$$\sigma(x) = \frac{1}{1 + e^{-ax_5 - b}}$$

By applying our model, we obtain

$$f(x) = -2.752x_5 - 6.715$$

$$\sigma(x) = \frac{1}{1 + e^{-2.752x_5 - 6.715}} \quad (3.1)$$

We can see there is a decision boundary when $\sigma(x) = 0.5$, $x_5 = -2.440$. In this case, the decision boundary is a single point. Therefore, our result can be shown as:

$$\begin{cases} x < -2.199, \sigma(x) \geq 0.5, P(x) = 1 \\ x \geq -2.199, \sigma(x) < 0.5, P(x) = 0 \end{cases} \quad (3.2)$$

We use the test samples to examine the expression and get a precision of 1.00 and an F1-Score of 1.00. We believe that such high accuracy can be attributed to the large difference in values between the FPNr data of #5 species and other species.

3.1.2 Results

The results are shown below.

	True species #5	True species #1-4, 6-8
Classified as species #5	214	0
Classified as species #1-4, 6-8	1	11

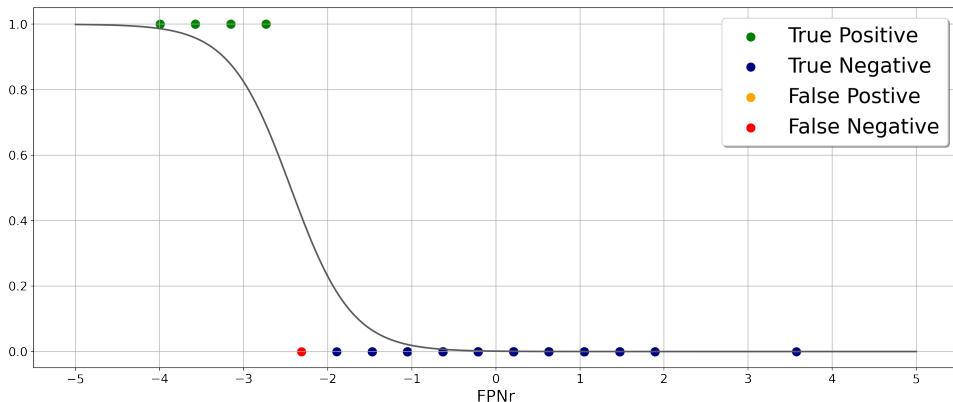


Figure 3: The result of Problem 1

3.2 Problem 2

3.2.1 Two Dimensional Logistic Regression

In problem 2, we are asked to differentiate lizards of species #5 from all other species and consider or use two variables out of the measured morphometric and pholidosis characteristics.

For this problem, we need to first find out the variable pair we are going to use. We do an exhaustive search to perform logistic regression. The morphometric characteristics group and the pholidosis characteristic group, every time we take one variable from each side, and use two dimensional Sigmoid function to apply logistic regression:

$$f(x) = ax_i + bx_j + c$$

$$\sigma(x) = \frac{1}{1 + e^{-ax_i - bx_j - c}}$$

We need to consider two independent values: x_i and x_j . And our decision boundary is no longer a single point, we have a line decision boundary:

$$\begin{aligned} \frac{1}{1 + e^{-ax_i - bx_j - c}} &= 0.5 \\ ax_i + bx_j + c &= 0 \\ \begin{cases} ax_i + bx_j + c \geq 0, \sigma(x) \geq 0.5, P(x) = 1 \\ ax_i + bx_j + c < 0, \sigma(x) < 0.5, P(x) = 0 \end{cases} \end{aligned} \quad (3.3)$$

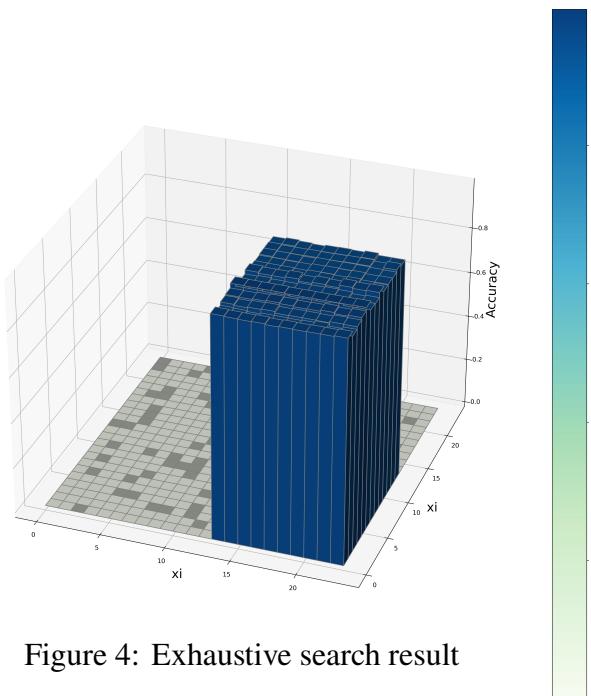


Figure 4: Exhaustive search result

After we try every possible combination of the variables from two sides, as shown in the graph above, we determine that by using the pair of SCGr(x_8) and PL(x_{17}), we will obtain

the largest mean accuracy of 0.99 and F1-score of 0.99 while using the test samples, and the resulting Sigmoid function is:

$$\sigma(x) = \frac{1}{1 + e^{-2.889x_8 - 0.661x_{17} - 6.924}}$$

$$\begin{cases} -2.889x_8 - 0.661x_{17} - 6.924 \geq 0, \sigma(x) \geq 0.5, P(x) = 1 \\ -2.889x_8 - 0.661x_{17} - 6.924 < 0, \sigma(x) < 0.5, P(x) = 0 \end{cases} \quad (3.4)$$

3.2.2 Results

The results are shown below.

	True species #5	True species #1-4, 6-8
Classified as species #5	217	0
Classified as species #1-4, 6-8	0	9

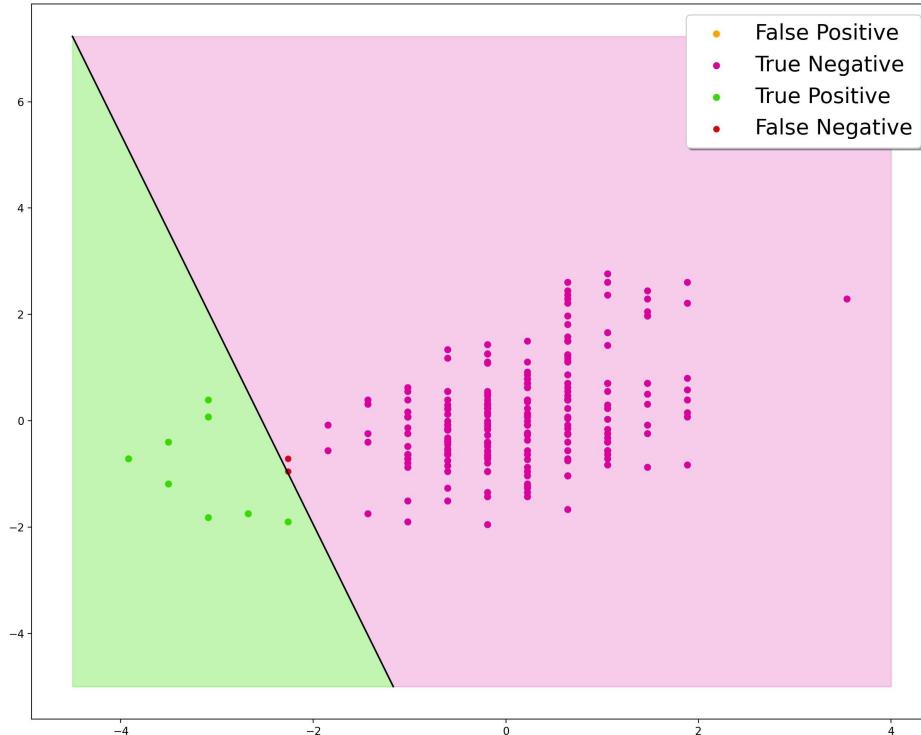


Figure 5: The result of Problem 2

3.3 Problem 3

3.3.1 Multi-Dimensional Logistic Regression: method 1

In problem 3, we are required to separate the two genders, which is a binary classification problem, and we want to determine the criterion with the greatest accuracy while remaining simple.

We do an exhaustive search over all the possible ratios of two random variables in the data list, and apply a logistic regression to find the ratio with the highest accuracy rate. (Some columns are missing here because these columns contain 0 value, which can not be a divisor.)

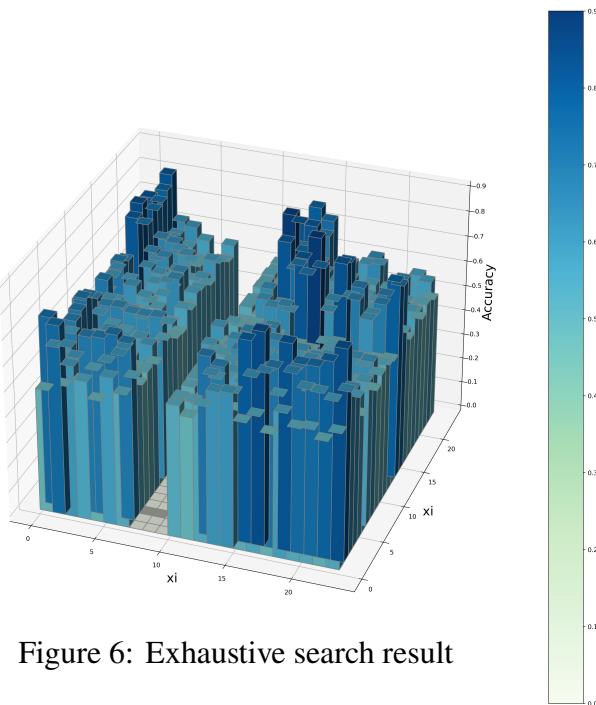


Figure 6: Exhaustive search result

We find that the best data to differentiate the gender is the ratio between SVL and PL and it reach an accuracy rate of 0.907. The $f(x)$ and $\sigma(x)$ obtained from the calculation are:

$$\sigma(x) = \frac{1}{1 + e^{-f(x)}}, \text{ where } f(x) = 2.621 \frac{SVL}{PL} + 0.490 \quad (3.5)$$

3.3.2 Results: method 1

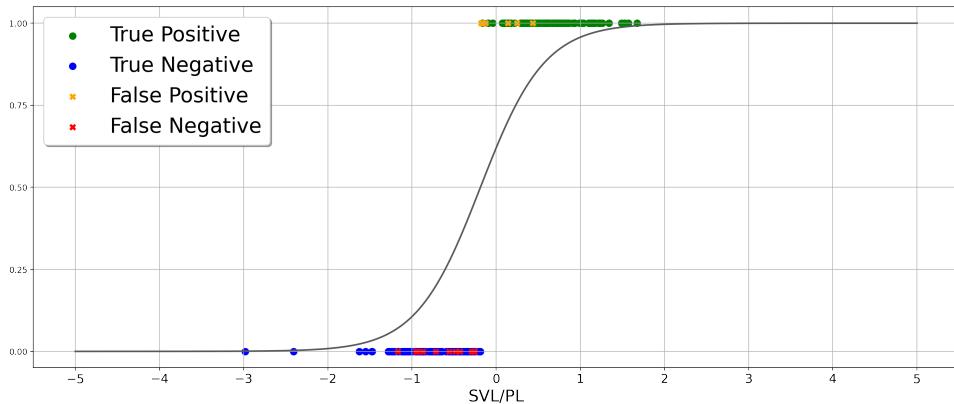


Figure 7: The result of Problem 3 using ratio

	True Male	True Female
Classified as Male	112	9
Classified as Female	12	93

Multi-Dimensional Logistic Regression: method 2 Then we try to achieve a higher accuracy rate, therefore we use all 23 variables and apply a logistic regression. This method has an accuracy of 92.9%. The $f(x)$ and $\sigma(x)$ obtained from the calculation is:

$$\sigma(x) = \frac{1}{1 + e^{-f(x)}}, \text{ where } f(x) = 0.208 + \sum_{i=1}^{23} w_i x_i \quad (3.6)$$

c in the chart below represents the intercept of the function, x_i represents the scaled value of the i th variable given by the question and w_i represents the coefficient of x_i . The value of w_i is shown in the table below.

x_i	MMS	VSN	CSN	GSN	FPNr	SDLr	SCSr	SCGr
w_i	-0.263	1.634	0.287	0.45	-0.072	-0.218	0.086	-0.918
x_i	SMr	MTr	PA	PTMr	aNDSr	SVL	TRL	HL
w_i	-0.036	0.046	0.425	-0.304	-0.396	1.333	1.103	-0.104
x_i	PL	ESD	HW	HH	MO	FFL	HFL	c
w_i	-0.396	0.26	-1.441	-0.967	-0.29	-0.846	-0.867	0.305

3.3.3 Results: method 2

$$\begin{cases} \sigma(x) \geq 0.5, & \text{male} \\ \sigma(x) < 0.5, & \text{female} \end{cases} \quad (3.7)$$

	True Male	True Female
Classified as Male	111	2
Classified as Female	9	104

Hence for this problem, we develop two criteria. Although both of them are quite precise while doing classification, the second one is comparatively more accurate but less convenient, and the first one is more convenient but less accurate. The biologists can choose the criterion depending on the circumstance.

3.4 Problem 4

3.4.1 Multi-dimensional Logistic Regression

In problem 4, we are required to differentiate species in a certain subgroup. Solving this problem requires multi-dimensional logistic regression.

In order to obtain the best possible accuracy rate, we get a new set of training samples and test samples for each subgroup listed in the problem. The set contains only the data of the species in the corresponding subgroup. After that we apply our multi-dimensional logistic regression model described in problem 3.

For subgroup a (species #6 and #7) and subgroup b (species #1 and #2), we use multi-dimensional logistic regression as we did in problem 3. We can treat it as a binary classification problem: for instance, for subgroup a, if the lizard belongs to species #7, $P(x) = 1$; if the lizard belongs to species #6, $P(x) = 0$. We take all the twenty three characteristics into consideration:

For Subgroup A:

$$\sigma(x) = \frac{1}{1 + e^{-f(x)}}, \text{ where } f(x) = -4.465 + \sum_{i=1}^{23} w_i x_i \quad (3.8)$$

x_i	MMS	VSN	CSN	GSN	FPNr	SDLr	SCSr	SCGr
w_i	0.758	0.453	0.462	0.552	0.437	0.08	-0.024	0.72
x_i	SMr	MTr	PA	PTMr	aNDsr	SVL	TRL	HL
w_i	0.527	0.596	0.276	0.401	0.148	0.108	0.037	0.035
x_i	PL	ESD	HW	HH	MO	FFL	HFL	c
w_i	0.05	-0.031	-0.265	-0.333	0.003	0.019	-0.014	-4.465

$$\begin{cases} \sigma(x) \geq 0.5, \text{ species\#7} \\ \sigma(x) < 0.5, \text{ species\#6} \end{cases} \quad (3.9)$$

Results: Subgroup A

We have an accuracy rate of 0.997.

	True species #6	True species #7
Classified as species #6	35	0
Classified as species #7	0	7

For Subgroup B:

$$\sigma(x) = \frac{1}{1 + e^{-f(x)}}, \text{ where } f(x) = -0.797 + \sum_{i=1}^{23} w_i x_i \quad (3.10)$$

x_i	MMS	VSN	CSN	GSN	FPNr	SDLr	SCSr	SCGr
w_i	-0.706	-0.432	0.12	0.048	-0.413	-0.52	-0.627	0.246
x_i	SMr	MTr	PA	PTMr	aNDsr	SVL	TRL	HL
w_i	0.299	-0.835	-1.533	1.226	0.143	-0.604	-0.006	-0.447
x_i	PL	ESD	HW	HH	MO	FFL	HFL	c
w_i	-0.461	0.071	0.133	0.128	0.371	-0.403	-0.691	-0.797

$$\begin{cases} \sigma(x) \geq 0.5, \text{ species\#2} \\ \sigma(x) < 0.5, \text{ species\#1} \end{cases} \quad (3.11)$$

Results: Subgroup B

We have an accuracy rate of 0.929.

	True species #1	True species #2
Classified as species #1	16	0
Classified as species #2	3	9

3.4.2 Multinomial Logistic Regression

For subgroup c: species #3, #4, and #5, we still use logistic regression as the basic algorithm, but it is modified to do more complicated calculations and sorting. As we need to differentiate

every species, the task is no longer a binary classification problem. It turns out to be a multi-classification problem. However, we still use logistic regression in the same way as we did before, but the only difference is that this time we consider every data outside the target as a whole. This means that only for the target $P(x) = 1$ and for the rest, $P(x) = 0$.

Then, we apply logistic regression models in the same way as we did in our second method in problem 3. If we need to classify n species, we have to repeat the process of doing binary regression n times. As we repeat this process three times for the three species in subgroup c, we obtain three functions, each for a species in the subgroup:

$$\begin{cases} \sigma_{species\#3}(x) = \frac{1}{1+e^{-f(x)}}, \text{ where } f(x) = 2.456 + \sum_{i=1}^{23} w_i x_i; \\ \sigma_{species\#4}(x) = \frac{1}{1+e^{-f(x)}}, \text{ where } f(x) = 0.439 + \sum_{i=1}^{23} w_i x_i; \\ \sigma_{species\#5}(x) = \frac{1}{1+e^{-f(x)}}, \text{ where } f(x) = -2.895 + \sum_{i=1}^{23} w_i x_i. \end{cases} \quad (3.12)$$

	Species and Gender Types		
x_i	#3	#4	#5
MMS	0.042	0.487	-0.529
VSN	0.305	0.208	-0.513
CSN	0.06	0.306	-0.366
GSN	-0.481	0.994	-0.513
FPNr	0.008	0.747	-0.755
SDLr	0.435	0.045	-0.48
SCSr	-0.215	0.39	-0.175
SCGr	1.234	-0.655	-0.579
SMr	0.036	0.161	-0.197
MTr	-0.385	0.36	0.025
PA	0.22	-0.15	-0.071
PTMr	0.666	-0.49	-0.175
aNDSr	0.463	-0.249	-0.214
SVL	-0.122	0.198	-0.076
TRL	-0.175	0.231	-0.056
HL	0.195	-0.066	-0.129
PL	-0.067	0.219	-0.152
ESD	0.19	-0.109	-0.081
HW	-0.334	0.361	-0.027
HH	-0.347	0.212	0.135
MO	0.014	-0.002	-0.012
FFL	0.016	0.145	-0.161
HFL	-0.045	0.303	-0.259
c	2.456	0.439	-2.895

After we get the functions, we are able to do multi-classification. By plugging the scaled values of variables into the three functions, we will get three results. By comparing the results and find out the largest one, the species corresponding to the function is the one for the lizard. Remember that our sigmoid function's results are probabilities, hence we can compare those probabilities. A lizard is claimed to belong to that species when it is most likely to belong that species.

For instance, here is a list of the unscaled values of variables for a lizard.

MMS	VSN	CSN	GSN	FPNr	SDLr	SCSr	SCGr
52.0	25.0	11.0	30.0	20.0	26.0	6.0	4.0
SMr	MTr	PA	PTMr	aNDSr	SVL	TRL	HL
2.0	2.0	1.0	1.0	2.5	54.0	28.0	19.8
PL	ESD	HW	HH	MO	FFL	HFL	
3.0	8.2	8.3	5.9	9.7	20.0	32.3	

We scale them and plug the scaled values into the three functions and get the results:

$$\begin{cases} \sigma_{species\#3}(x) = 0.990; \\ \sigma_{species\#4}(x) = 0.999; \\ \sigma_{species\#5}(x) = 6.253 * 10^{-14}. \end{cases} \quad (3.13)$$

The second function yields the greatest result. Hence, we can say that this lizard belongs to species 4.

Results: Subgroup C

We have an accuracy rate of 0.985.

	True species #3	True species #4	True species #5
Classified as species #3	48	1	0
Classified as species #4	0	28	0
Classified as species #5	1	0	4

3.5 Problem 5

3.5.1 Logistic Regression

We split the question into 2 parts. We first develop a multinomial logistic regression model to determine the species of the lizard, then we do the same to determine the gender.

$$\sigma(x) = \frac{1}{1 + e^{-f(x)}}, \text{ where } f(x) = c + \sum_{i=1}^{23} w_i x_i \quad (3.14)$$

And the coefficients and intercepts for all the functions are shown in the tables below. We need to have fourteen functions to be able to classify all the lizards' species and sex. Note that M represents Male and F represents Female.

After we get the functions, we are able to do multi-classification. By plugging the values of variables into fourteen functions, we will get fourteen results. The same as we did for problem 4 subgroup c, by comparing the results and find out the largest one, the type corresponding to the function is the one for the lizard.

x_i	Species and Gender Types						
	#1M	#1F	#2M	#2F	#3M	#3F	#4M
MMS	0.883	0.749	0.649	-0.116	-0.420	-0.910	-0.202
VSN	-0.587	0.269	-0.445	0.115	-0.511	0.114	-1.117
CSN	-0.251	0.203	0.022	0.037	-0.350	-0.578	-0.005
GSN	-0.206	-0.751	0.006	-0.025	-0.422	-0.008	0.992
FPNr	0.143	0.930	0.048	-0.129	-0.386	-0.111	0.390
SDLr	0.564	0.587	0.331	0.241	-0.040	0.872	-0.21
SCSr	0.420	0.455	-0.381	-0.364	-0.103	-0.387	0.298
SCGr	0.585	0.074	0.809	0.711	0.688	0.590	-0.720
SMr	0.002	0.142	-0.053	0.223	0.111	0.081	-0.076
MTr	0.990	0.682	-0.004	-0.246	-0.269	-0.659	0.099
PA	0.193	0.612	-0.957	-0.320	0.469	0.786	-0.547
PTMr	-0.242	-0.434	1.004	0.651	0.377	-0.235	-0.510
aNDSr	-0.371	-0.208	-0.308	0.021	0.757	0.868	0.137
SVL	-0.122	0.440	-0.603	0.165	-0.951	0.384	-0.317
TRL	-0.569	0.400	-0.551	1.111	-0.451	0.764	-0.603
HL	0.840	-0.028	-0.361	-0.787	0.252	-0.273	0.226
PL	0.870	-0.389	0.294	-0.602	0.720	-0.852	0.769
ESD	-0.347	-0.857	-0.379	-0.894	1.201	0.006	1.075
HW	0.805	-0.495	0.616	-0.601	-0.120	-0.811	0.458
HH	-0.362	-0.294	-0.011	-0.721	0.596	-0.067	0.823
MO	0.749	0.132	0.666	-0.089	-0.281	-1.436	-0.242
FFL	-0.329	-0.549	-0.154	-0.780	0.414	0.079	0.817
HFL	1.009	0.146	-0.464	-0.786	0.507	-0.359	1.013
c	-0.262	0.392	0.557	-1.858	3.310	-0.248	-0.457

x_i	Species and Gender Types						
	#4F	#5M	#5F	#6F	#7F	#8M	#8F
MMS	0.202	-0.479	-0.433	-0.610	0.885	-0.396	0.199
VSN	0.128	-0.826	-0.066	0.794	0.979	0.117	1.036
CSN	0.181	-0.702	-0.445	0.381	0.961	0.538	0.009
GSN	1.340	-0.845	-0.408	-0.469	0.483	0.338	-0.024
FPNr	0.529	-0.563	-1.832	0.379	0.390	0.105	0.106
SDLr	0.312	-0.08	-0.422	-0.849	-0.581	-0.961	0.244
SCSr	0.430	0.099	-0.288	-0.055	-0.012	-0.082	-0.029
SCGr	-0.832	-0.657	-0.556	-1.000	0.675	0.218	-0.580
SMr	0.503	-0.365	-0.204	-0.238	0.112	-0.165	-0.074
MTr	0.092	-0.082	-0.040	-0.954	0.213	0.028	0.149
PA	0.128	-0.075	0.399	-0.177	0.485	-0.783	-0.214
PTMr	-0.725	-0.736	-0.061	-0.048	0.514	-0.023	0.468
aNDSr	-0.012	0.044	-0.132	-0.806	-0.042	0.198	-0.146
SVL	0.784	-0.528	0.253	0.414	0.066	-0.037	0.052
TRL	0.688	-0.629	0.461	-0.182	-0.285	-0.282	0.129
HL	-0.106	0.085	-0.270	-0.093	0.009	0.460	0.047
PL	-0.237	0.036	-0.469	-0.091	-0.052	0.221	-0.217
ESD	0.267	0.105	-0.276	0.939	0.418	-0.663	-0.596
HW	0.147	0.211	-0.267	-0.545	-0.422	0.923	0.103
HH	-0.246	0.150	0.137	-0.310	-0.269	0.503	0.069
MO	0.255	-0.103	-0.209	0.562	0.053	0.188	-0.245
FFL	0.000	-0.073	-0.189	-0.106	0.080	0.304	0.487
HFL	-0.127	-0.073	-0.418	-0.503	-0.143	0.034	0.166
c	1.076	-1.346	-1.745	1.261	-0.600	0.381	-0.462

3.5.2 Results: method 1

The accuracy rate is 0.903 and the results in table below:

True Classify \#	#1M	#1F	#2M	#2F	#3M	#3F	#4M	#4F	#5M	#5F	#6F	#7F	#8M	#8F
#1M	18	0	0	0	0	0	0	0	0	0	0	0	0	0
#1F	0	7	0	0	0	0	0	0	0	0	0	0	0	0
#2M	1	0	21	0	0	0	0	0	0	0	0	0	0	0
#2F	0	0	1	6	0	0	0	0	0	0	0	0	0	0
#3M	0	0	0	0	34	1	0	0	0	0	0	0	0	0
#3F	0	0	0	0	4	14	0	0	0	0	0	0	0	0
#4M	0	0	0	0	0	0	15	0	0	0	0	0	0	0
#4F	0	0	0	0	0	0	2	19	0	0	0	0	0	0
#5M	0	0	0	0	0	0	0	0	3	2	0	0	0	0
#5F	0	0	0	0	0	0	0	0	0	2	0	0	0	0
#6F	0	0	0	0	2	0	0	0	0	0	55	0	0	0
#7F	0	0	0	0	0	0	0	0	0	0	0	10	0	0
#8M	0	0	0	0	0	1	0	0	0	0	0	0	5	0
#8F	0	0	0	0	0	0	0	1	0	0	1	0	0	1

3.5.3 Decision Tree

A major issue with logistic regression is that this method is too complicated for biologists as it requires the biologists to get the results of fourteen functions. Therefore, we have another possible method to solve this problem, which is the decision tree. The decision tree is a flowchart-like structure in which each internal node represents a test on an attribute and each branch represents the outcome of the test.

We built this tree with Gini Impurity, and this building method is based on the greedy algorithm. We only consider the best decision in the current step. We need to calculate the Gini Impurity for each leaf under this node first, which is calculated as below:

$$G_{leaf} = 1 - \sum_{i=1}^J p(i) \quad (3.15)$$

where $p(i)$ is the probability of each element sorted after the test and J is the number of classes in this leaf or node. Then, we can calculate the Gini Impurity for this node as the weighted average of all the Gini Impurities of the leaves under this node, which can be written as:

$$G_{node} = \frac{\sum_{i=1}^J p(i) G_{leaf}(J)}{J} \quad (3.16)$$

It is easy to observe that the closer the Gini Impurity is to 0, the better this node can sort elements. When the Gini Impurity equals 0, the element was completely sorted.

Then, we need to find the test criterion with the least Gini Impurity of the node for the greedy algorithm. For a classification feature, we need to sort them from lowest to highest first, then calculate the average of all adjacent data, and calculate the Gini Impurity for each average value, and find the lowest one. We need to repeat this process for every classification feature to find the feature and value with the lowest Gini Impurity as the criterion on this node.

Finally, we need to consider all the leaves as a new node and repeat the previous process until all the leaves have a Gini Impurity of 0.

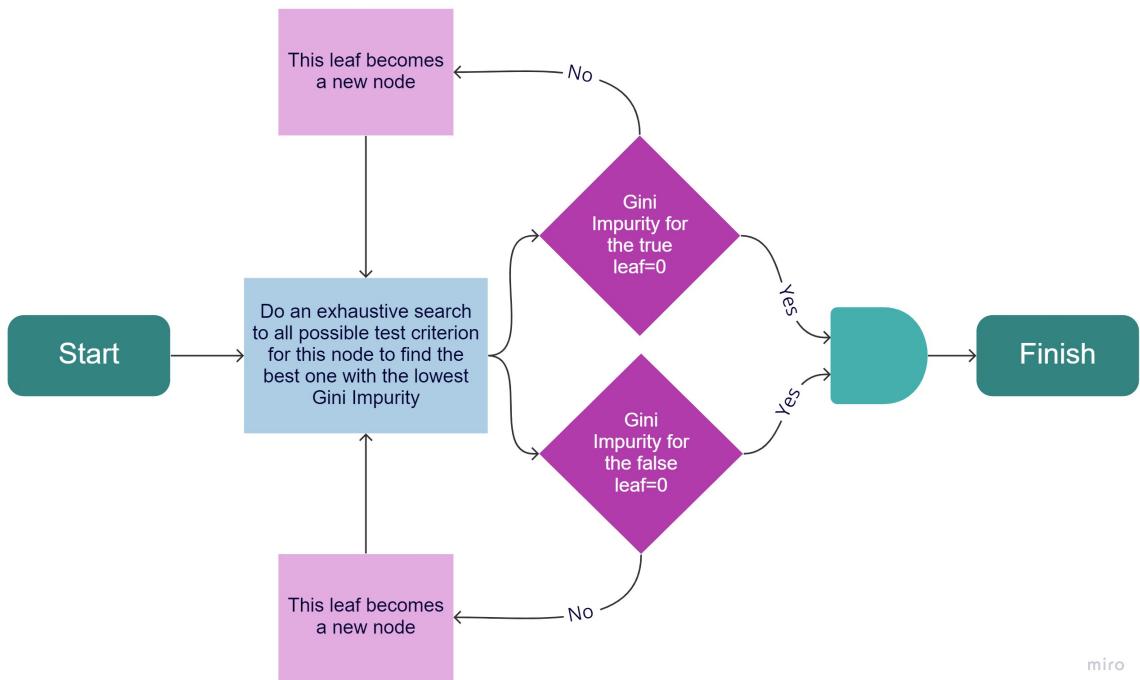


Figure 8: The construction process of a Decision Tree

One of the great advantages of decision trees is that they are easy to use. For biologists, it is so convenient to look at the samples they have obtained and then at the graph to determine the lizard species even without any calculations.

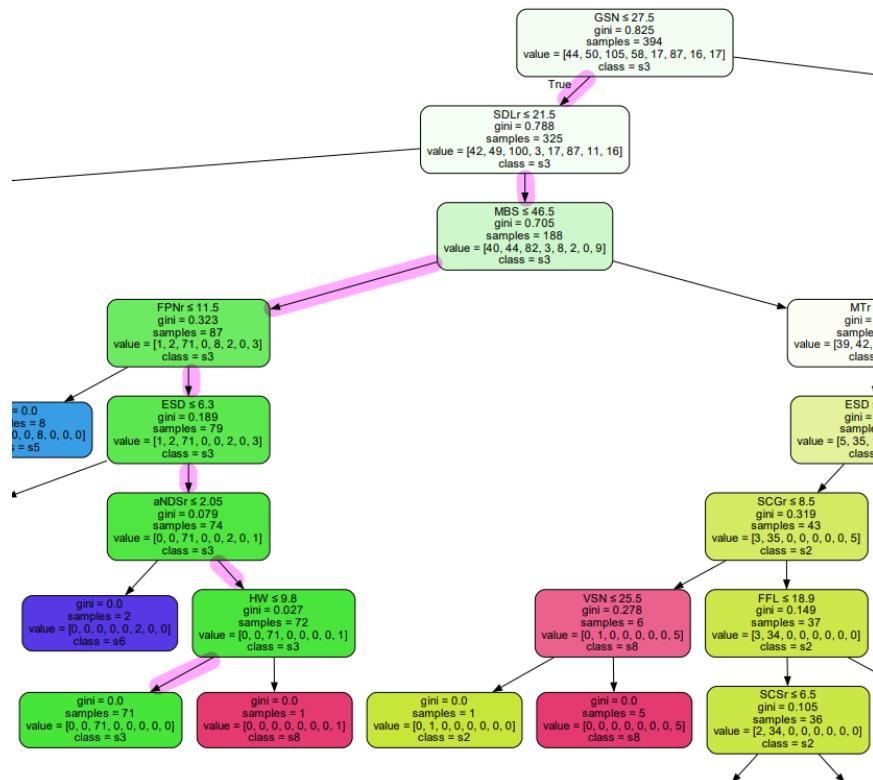


Figure 9: Example of using Decision Tree

#3	MMS	VSN	CSN	GSN	FPNr	SDLr	SCSr	SCGr	SMr	MTr	PA
	45	23	7	21	16	26	6	7	1	2	2
PTMr	aNDSr	SVL	TRL	HL	PL	ESD	HW	HH	MO	FFL	HFL
3	3	57.5	30.6	19.8	13	8.7	8.5	6	9	20.2	31.8

We will take this row of data from the given data set as a sample taken by a biologist. We start from the root node of the decision tree: whether the GSN is less than or equal to 27.5. From the table, we know that this proposition is true, so we move along the left sub-tree of the root node. When this child node is reached, we continue to judge whether $SDLr$ is less than or equal to 21.5. According to the table, we found that the $SDLr$ of this sample is 26 greater than 21.5, so we proceed along the right sub-tree of its points, and so on. Eventually, we stopped at the leaf node in the lower left corner of the picture, where the path is highlighted in pink. The leaf node implies that the species of this sample should be Species #3, which is the same as the given data.

3.5.4 Results: method 2

This method has an accuracy of 0.889 and the results are shown below:

True Classify \	#1	#2	#3	#4	#5	#6	#7	#8
#1	21	1	0	0	0	0	0	0
#2	0	10	0	0	0	0	0	3
#3	0	0	48	0	0	2	0	1
#4	0	1	1	31	0	0	0	2
#5	0	0	0	0	6	1	0	0
#6	0	0	3	0	0	29	0	1
#7	0	0	0	0	1	0	5	0
#8	0	0	0	1	0	1	0	1

3.5.5 Random Forest

Decision Tree Algorithm has a high requirement on the training samples and could easily lead to over-fitting a model. Therefore, we used the Random Forest Algorithm to have higher accuracy and less bias.

The random forest consists of several decision trees. However, there is no direct connection between any decision trees. When we are doing classification works, and the new sample is input, thus every decision tree in the random forest will judge and classify it. Every decision tree will result in its result. If which classification made by all decision trees is the most, then the random forest will take this classification as the final output of the whole random forest.

The number of features is approximately the square root of the original number of features, which in this problem, is 5. Thus in our model, we have five decision trees within a single random forest.

When the training sample with a size of N is input, a random forest will randomly select N samples with replacement, which means the sample repetition may exist. Thus these N samples will be used to train a single decision tree.

When each sample has M attributes, every node of the decision tree needs to be split, and the random forest will randomly select m ($m < M$) details. Then a certain strategy is adopted from the m attributes to choose one attribute as the splitting attribute of the node. During the formation of the decision tree, each node must be split according to the previous stage until it can no longer be split. It constructs the five decision trees by repeating the previous steps, thus forming the random forest.

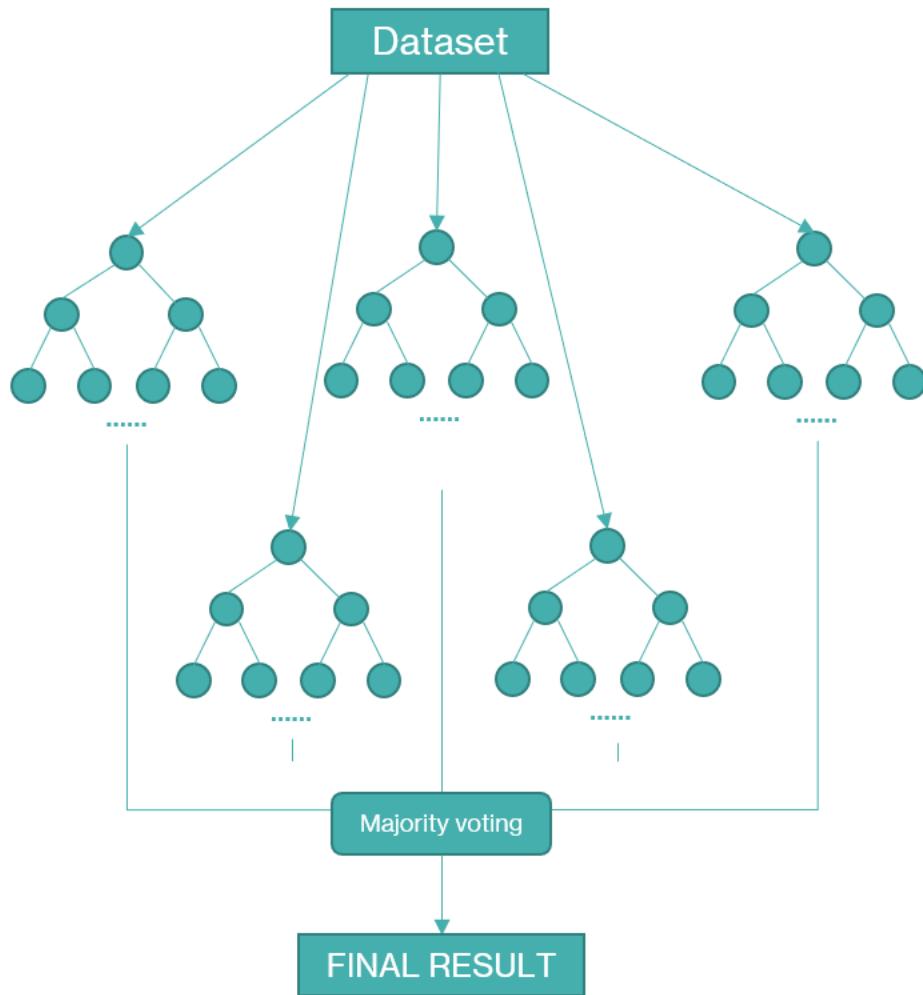


Figure 10: Schematic Diagram of a Random Forest

3.5.6 Results: method 3

This method has an accuracy of 0.918 and the results are shown below:

True Classify \	#1	#2	#3	#4	#5	#6	#7	#8
#1	14	2	0	0	0	0	0	0
#2	2	15	0	0	0	0	0	3
#3	1	0	41	0	0	2	0	1
#4	0	0	0	33	0	0	0	2
#5	0	0	0	0	3	0	0	0
#6	0	0	3	0	0	40	0	0
#7	1	0	0	0	0	0	9	0
#8	0	1	0	2	0	2	0	1

	True Male	True Female
Classified as Male	64	8
Classified as Female	9	89

3.5.7 Discussion of the results

Multinomial logistic regression already helped us develop a criterion composed of several functions to help the biologists divide lizards. However, in this case, it requires the biologists to calculate the results of fourteen functions. This can be inconvenient for the biologists in some circumstances.

Therefore, we introduce another method, decision trees, because it is almost computationally non-existent relative to other algorithms and is one of the most convenient classification methods available to biologists. Of course, using only decision trees sacrifices some accuracy, hence we further develops the random forests model. The accuracy of the results obtained by the five decision trees separately after the main vote will be higher than that of a single decision tree.

4 Strengths and Weaknesses

4.1 Strengths

Our logistic regression model provides the biologists Sigmoid functions so that they can easily calculate the results of the functions on a graphing calculator and compare the values of the results to classify the lizards.

All the results of our model yield high accuracy. Moreover, we introduce several different options for the biologists to let them decide whether to emphasize on convenience or accuracy in different situations.

4.2 Weaknesses

Our logistic regression model is most suitable when the number of variables is relatively small and when it faces a binary classification problem. When the number of variables become larger-for instance, in problem 3,4, and 5, we consider all 23 characteristics-it becomes a little bit more complicated for the biologists to plug in the data. When we face multi-classification problems, in problem 5, we need to get the results of fourteen functions, which is a huge amount of work. (That's why we introduce the decision tree and random forests model.)

Our accuracy while predicting some certain species, especially species#8, is not satisfying enough, not reaching 100%.

5 Conclusion

In conclusion, simplicity and accuracy are two important factors we need to consider while developing the criteria for the biologists to categorize the lizards into different sex and species.

Our model is mainly based on logistic regression. It provides the biologists Sigmoid functions. By calculating the results of the functions on a graphing calculator and compare the values of the results, the biologists can classify the lizards.

In the first question, we use one dimensional logistic regression model to differentiate species #5 from other species. In problem 2, we extend to two dimensional logistic regression model since this time we need to consider a pair of variables. In problem 3, we introduce multi-dimensional logistic regression model and take all the twenty three characteristics into consideration. In problem 4, we use binary classification logistic regression model to classify species in subgroup a and b. And then we introduce our multinomial logistic regression model to divide subgroup c. Finally, in problem 5, we first use our multinomial logistic regression

model to classify all the lizards' species and sex. Realizing that the process turns out to be a little bit more complicated, we introduce other methods to help the biologists. We introduce the decision tree model and further improve it to random forests.

Our models are suitable for biologists to get the results only using a graphing calculator. For multi-dimensional and multinomial problems, we provide the biologists various options to choose depending on the circumstances.

6 Bibliography

- Hosmer, David W. and Stanley Lemeshow. “Applied Logistic Regression.” (1991).[1]
- Hanson, Roger K.. “Logistic regression.” Prediction statistics for psychological assessment. (2022): n. pag.[2]

7 Appendix

7.1 PCA Coefficient

PCA CovARIANCE Matrix: [[1.0000000e+00 -4.85806321e-02 4.45640743e-01 4.96539154e-01 6.38247825e-01 2.99497469e-01 2.36984595e-01 4.68159858e-01 3.78413480e-01 5.15360601e-01 -1.59245841e-01 8.35412416e-02 -1.86899930e-02 1.46340639e-01 6.81210876e-02 1.93601242e-01 2.66228502e-01 -1.17531649e-01 2.40550131e-01 -4.21084529e-02 3.15467567e-01 5.55411265e-02 2.69134957e-01] [-4.85806321e-02 1.00000000e+00 2.26166575e-01 1.03875814e-01 7.65432256e-02 -4.08171881e-01 3.76457674e-03 -2.01916909e-01 -5.28113918e-02 -1.90278237e-01 8.62223349e-02 -2.44982554e-01 1.88989381e-01 2.20140090e-01 -1.56345397e-01 -2.17628065e-01 -7.11574348e-02 -2.63573476e-01 01 -1.3284178e-01 3.61191636e-02 4.72470740e-02 2.07743642e-01] [4.45640743e-01 2.26166575e-01 1.00000000e+00 3.20662453e-01 4.58325701e-01 3.13779674e-02 1.04426763e-01 1.73856098e-01 1.92718938e-01 -2.49031848e-01 9.45017062e-02 -3.08987661e-01 2.89725004e-01 1.35518868e-01 2.02818537e-01 2.68791246e-01 6.69536953e-03 1.94092211e-01 5.90381352e-04 3.89591354e-01 1.50271838e-01 2.82789930e-01] [4.96539154e-01 1.03875814e-01 3.20662453e-01 1.00000000e+00 6.37428183e-01 1.70903590e-01 2.58929840e-01 -5.98439931e-02 3.09524167e-01 2.40170225e-01 -2.57213463e-01 -3.30537832e-01 1.43709142e-01 4.40054024e-01 3.35478369e-01 4.02635967e-01 4.71786829e-01 2.63322172e-01 4.74244434e-01 3.82132801e-01 2.21025698e-01 4.70272058e-01 5.37248053e-01] [6.38247825e-01 7.65432256e-02 4.58325701e-01 6.37428183e-01 1.00000000e+00 2.56916930e-01 2.53443886e-01 2.35975826e-01 2.91655041e-01 3.06758284e-01 -2.80313836e-01 -5.18899322e-02 3.66923585e-02 3.39476246e-01 2.00938169e-01 3.54325020e-01 4.35117915e-01 6.77100467e-02 3.87384990e-01 1.61366639e-01 4.01075976e-01 3.33091225e-01 4.75522747e-01] [2.99497469e-01 -4.08171881e-01 -3.13779674e-02 1.70903590e-01 2.56916930e-01 1.00000000e+00 1.70983127e-01 2.74619052e-01 1.82643315e-01 3.07614153e-01 1.54190230e-01 -1.90864380e-01 2.61566153e-01 -2.11520489e-02 5.21374713e-02 1.47724093e-01 1.67394442e-01 -1.82092370e-03 2.33305205e-01 9.67371356e-02 -7.65886449e-02 4.30683233e-02 2.2378986e-01] [2.36984595e-01 3.76457674e-03 1.04399892e-01 2.58929840e-01 -5.98439931e-02 3.09524167e-01 2.40170225e-01 -2.57213463e-01 -3.30537832e-01 1.43709142e-01 4.40054024e-01 3.35478369e-01 4.02635967e-01 4.71786829e-01 2.63322172e-01 4.74244434e-01 3.82132801e-01 2.21025698e-01 4.70272058e-01 5.37248053e-01] [6.38247825e-01 7.65432256e-02 4.58325701e-01 6.37428183e-01 1.00000000e+00 2.0313836e-01 -5.18899322e-02 3.66923585e-02 3.39476246e-01 2.00938169e-01 3.54325020e-01 4.35117915e-01 -5.98439931e-02 2.35975826e-01 2.74619052e-01 2.12696010e-01 1.00000000e+00 0.232172872e-01 2.84887988e-01 1.94872977e-01 3.58062376e-01 1.34420822e-01 -2.37308909e-01 -1.75890025e-01 -7.80074463e-02 -7.69123418e-02 -2.48453821e-01 -1.3331560e-01 2.31272872e-01 4.95743881e-02 -2.77562320e-01 -9.23692435e-02] [3.78413480e-01 -5.28113918e-02 1.73856098e-01 3.09524167e-01 2.91655041e-01 1.82643315e-01 3.07614153e-01 1.54190230e-01 -1.90864380e-01 2.61566153e-01 -2.11520489e-02 5.21374713e-02 1.47724093e-01 -1.67394442e-01 -1.82092370e-03 2.33305205e-01 9.67371356e-02 -7.65886449e-02 4.30683233e-02 2.2378986e-01] [2.36984595e-01 3.76457674e-03 1.04399892e-01 2.58929840e-01 -5.98439931e-02 3.09524167e-01 2.40170225e-01 -2.57213463e-01 -3.30537832e-01 1.43709142e-01 4.40054024e-01 3.35478369e-01 4.02635967e-01 4.71786829e-01 2.63322172e-01 4.74244434e-01 3.82132801e-01 2.21025698e-01 4.70272058e-01 5.37248053e-01] [6.38247825e-01 7.65432256e-02 4.58325701e-01 6.37428183e-01 1.00000000e+00 2.0313836e-01 -5.18899322e-02 3.66923585e-02 3.39476246e-01 2.00938169e-01 3.54325020e-01 4.35117915e-01 -5.98439931e-02 2.35975826e-01 2.74619052e-01 2.12696010e-01 1.00000000e+00 0.232172872e-01 2.84887988e-01 1.94872977e-01 3.58062376e-01 1.34420822e-01 -2.37308909e-01 -1.75890025e-01 -7.80074463e-02 -7.69123418e-02 -2.48453821e-01 -1.3331560e-01 2.31272872e-01 4.95743881e-02 -2.77562320e-01 -9.23692435e-02] [3.78413480e-01 4.59793702e-02 9.34269946e-02 1.26179827e-01 2.10641416e-01 -1.49414641e-01 -4.29566301e-02 1.62665974e-01 4.00073245e-02 9.29682817e-02 0.96420818e-02 1.62245742e-01] [1.515360601e-01 -1.90278229e-01 1.92718938e-01 2.40170225e-01 3.06758284e-01 3.07614153e-01 2.86410192e-01 2.84887988e-01 4.88720844e-01 1.00000000e+00 6.61739577e-02 -4.13350384e-02 3.08144127e-02 9.86339936e-02 1.04026922e-01 1.48134377e-01 -1.45865886e-01 2.27575342e-01 1.84426704e-02 1.40662265e-01 -2.46652964e-03 1.76740736e-01] [1.59245841e-01 -1.18238797e-01 -2.49031848e-01 -2.74619052e-01 2.12696010e-01 1.00000000e+00 0.232172872e-01 2.84887988e-01 1.94872977e-01 -3.06579502e-02 6.61739577e-02 0.21.00000000e+00 -1.75890025e-01 -7.80074463e-02 -7.69123418e-02 -2.48453821e-01 -1.3331560e-01 2.31272872e-01 4.95743881e-02 -2.77562320e-01 -9.23692435e-02] [3.78413480e-01 -5.28113918e-02 1.73856098e-01 3.09524167e-01 2.91655041e-01 1.82643315e-01 3.07614153e-01 1.54190230e-01 -1.90864380e-01 2.61566153e-01 -2.11520489e-02 5.21374713e-02 1.47724093e-01 -1.67394442e-01 -1.82092370e-03 2.33305205e-01 9.67371356e-02 -7.65886449e-02 4.30683233e-02 2.2378986e-01] [2.36984595e-01 3.76457674e-03 1.04399892e-01 2.58929840e-01 -5.98439931e-02 3.09524167e-01 2.40170225e-01 -2.57213463e-01 -3.30537832e-01 1.43709142e-01 4.40054024e-01 3.35478369e-01 4.02635967e-01 4.71786829e-01 2.63322172e-01 4.74244434e-01 3.82132801e-01 2.21025698e-01 4.70272058e-01 5.37248053e-01] [6.38247825e-01 7.65432256e-02 4.58325701e-01 6.37428183e-01 1.00000000e+00 2.0313836e-01 -5.18899322e-02 3.66923585e-02 3.39476246e-01 2.00938169e-01 3.54325020e-01 4.35117915e-01 -5.98439931e-02 2.35975826e-01 2.74619052e-01 2.12696010e-01 1.00000000e+00 0.232172872e-01 2.84887988e-01 1.94872977e-01 3.58062376e-01 1.34420822e-01 -2.37308909e-01 -1.75890025e-01 -7.80074463e-02 -7.69123418e-02 -2.48453821e-01 -1.3331560e-01 2.31272872e-01 4.95743881e-02 -2.77562320e-01 -9.23692435e-02] [3.78413480e-01 4.59793702e-02 9.34269946e-02 1.26179827e-01 2.10641416e-01 -1.49414641e-01 -4.29566301e-02 1.62665974e-01 4.00073245e-02 9.29682817e-02 0.96420818e-02 1.62245742e-01] [1.515360601e-01 -1.90278229e-01 1.92718938e-01 2.40170225e-01 3.06758284e-01 3.07614153e-01 2.86410192e-01 2.84887988e-01 4.88720844e-01 1.00000000e+00 6.61739577e-02 -4.13350384e-02 3.08144127e-02 9.86339936e-02 1.04026922e-01 1.48134377e-01 -1.45865886e-01 2.27575342e-01 1.84426704e-02 1.40662265e-01 -2.46652964e-03 1.76740736e-01] [1.59245841e-01 -1.18238797e-01 -2.49031848e-01 -2.74619052e-01 2.12696010e-01 1.00000000e+00 0.232172872e-01 2.84887988e-01 1.94872977e-01 -3.06579502e-02 6.61739577e-02 0.21.00000000e+00 -1.75890025e-01 -7.80074463e-02 -7.69123418e-02 -2.48453821e-01 -1.3331560e-01 2.31272872e-01 4.95743881e-02 -2.77562320e-01 -9.23692435e-02] [3.78413480e-01 4.59793702e-02 9.34269946e-02 1.26179827e-01 2.10641416e-01 -1.49414641e-01 -4.29566301e-02 1.62665974e-01 4.00073245e-02 9.29682817e-02 0.96420818e-02 1.62245742e-01] [1.515360601e-01 -1.90278229e-01 1.92718938e-01 2.40170225e-01 3.06758284e-01 3.07614153e-01 2.86410192e-01 2.84887988e-01 4.88720844e-01 1.00000000e+00 6.61739577e-02 -4.13350384e-02 3.08144127e-02 9.86339936e-02 1.04026922e-01 1.48134377e-01 -1.45865886e-01 2.27575342e-01 1.84426704e-02 1.40662265e-01 -2.46652964e-03 1.76740736e-01] [1.59245841e-01 -1.18238797e-01 -2.49031848e-01 -2.74619052e-01 2.12696010e-01 1.00000000e+00 0.232172872e-01 2.84887988e-01 1.94872977e-01 -3.06579502e-02 6.61739577e-02 0.21.00000000e+00 -1.75890025e-01 -7.80074463e-02 -7.69123418e-02 -2.48453821e-01 -1.3331560e-01 2.31272872e-01 4.95743881e-02 -2.77562320e-01 -9.23692435e-02] [3.78413480e-01 4.59793702e-02 9.34269946e-02 1.26179827e-01 2.10641416e-01 -1.49414641e-01 -4.29566301e-02 1.62665974e-01 4.00073245e-02 9.29682817e-02 0.96420818e-02 1.62245742e-01] [1.515360601e-01 -1.90278229e-01 1.92718938e-01 2.40170225e-01 3.06758284e-01 3.07614153e-01 2.86410192e-01 2.84887988e-01 4.88720844e-01 1.00000000e+00 6.61739577e-02 -4.13350384e-02 3.08144127e-02 9.86339936e-02 1.04026922e-01 1.48134377e-01 -1.45865886e-01 2.27575342e-01 1.84426704e-02 1.40662265e-01 -2.46652964e-03 1.76740736e-01] [1.59245841e-01 -1.18238797e-01 -2.49031848e-01 -2.74619052e-01 2.12696010e-01 1.00000000e+00 0.232172872e-01 2.84887988e-01 1.94872977e-01 -3.06579502e-02 6.61739577e-02 0.21.00000000e+00 -1.75890025e-01 -7.80074463e-02 -7.69123418e-02 -2.48453821e-01 -1.3331560e-01 2.31272872e-01 4.95743881e-02 -2.77562320e-01 -9.23692435e-02] [3.78413480e-01 4.59793702e-02 9.34269946e-02 1.26179827e-01 2.10641416e-01 -1.49414641e-01 -4.29566301e-02 1.62665974e-01 4.00073245e-02 9.29682817e-02 0.96420818e-02 1.62245742e-01] [1.515360601e-01 -1.90278229e-01 1.92718938e-01 2.40170225e-01 3.06758284e-01 3.07614153e-01 2.86410192e-01 2.84887988e-01 4.88720844e-01 1.00000000e+00 6.61739577e-02 -4.13350384e-02 3.08144127e-02 9.86339936e-02 1.04026922e-01 1.48134377e-01 -1.45865886e-01 2.27575342e-01 1.84426704e-02 1.40662265e-01 -2.46652964e-03 1.76740736e-01] [1.59245841e-01 -1.18238797e-01 -2.49031848e-01 -2.74619052e-01 2.12696010e-01 1.00000000e+00 0.232172872e-01 2.84887988e-01 1.94872977e-01 -3.06579502e-02 6.61739577e-02 0.21.00000000e+00 -1.75890025e-01 -7.80074463e-02 -7.69123418e-02 -2.48453821e-01 -1.3331560e-01 2.31272872e-01 4.95743881e-02 -2.77562320e-01 -9.23692435e-02] [3.78413480e-01 4.59793702e-02 9.34269946e-02 1.26179827e-01 2.10641416e-01 -1.49414641e-01 -4.29566301e-02 1.62665974e-01 4.00073245e-02 9.29682817e-02 0.96420818e-02 1.62245742e-01] [1.515360601e-01 -1.90278229e-01 1.92718938e-01 2.40170225e-01 3.06758284e-01 3.07614153e-01 2.86410192e-01 2.84887988e-01 4.88720844e-01 1.00000000e+00 6.61739577e-02 -4.13350384e-02 3.08144127e-02 9.86339936e-02 1.04026922e-01 1.48134377e-01 -1.45865886e-01 2.27575342e-01 1.84426704e-02 1.40662265e-01 -2.46652964e-03 1.76740736e-01] [1.59245841e-01 -1.18238797e-01 -2.49031848e-01 -2.74619052e-01 2.12696010e-01 1.00000000e+00 0.232172872e-01 2.84887988e-01 1.94872977e-01 -3.06579502e-02 6.61739577e-02 0.21.00000000e+00 -1.75890025e-01 -7.80074463e-02 -7.69123418e-02 -2.48453821e-01 -1.3331560e-01 2.31272872e-01 4.95743881e-02 -2.77562320e-01 -9.23692435e-02] [3.78413480e-01 4.59793702e-02 9.34269946e-02 1.26179827e-01 2.10641416e-01 -1.49414641e-01 -4.29566301e-02 1.62665974e-01 4.00073245e-02 9.29682817e-02 0.96420818e-02 1.62245742e-01] [1.515360601e-01 -1.90278229e-01 1.92718938e-01 2.40170225e-01 3.06758284e-01 3.07614153e-01 2.86410192e-01 2.84887988e-01 4.88720844e-01 1.00000000e+00 6.61739577e-02 -4.13350384e-02 3.08144127e-02 9.86339936e-02 1.04026922e-01 1.48134377e-01 -1.45865886e-01 2.27575342e-01 1.84426704e-02 1.40662265e-01 -2.46652964e-03 1.76740736e-01] [1.59245841e-01 -1.18238797e-01 -2.49031848e-01 -2.74619052e-01 2.12696010e-01 1.00000000e+00 0.232172872e-01 2.84887988e-01 1.94872977e-01 -3.06579502e-02 6.61739577e-02 0.21.00000000e+00 -1.75890025e-01 -7.80074463e-02 -7.69123418e-02 -2.48453821e-01 -1.3331560e-01 2.31272872e-01 4.95743881e-02 -2.77562320e-01 -9.23692435e-02] [3.78413480e-01 4.59793702e-02 9.3

14.0, 18.0, 16.0, 16.0, 15.0, 15.0, 17.0, 16.0, 16.0, 16.0, 15.0, 20.0, 16.0, 16.0, 17.0, 18.0, 22.0, 16.0, 21.0, 15.0, 14.0, 19.0, 16.0, 19.0, 18.0, 18.0, 21.0, 17.0, 17.0, 16.0, 13.0, 19.0, 17.0, 17.0, 17.0, 14.0, 8.0, 15.0, 17.0, 17.0, 19.0, 15.0, 15.0, 18.0, 18.0, 17.0, 16.0, 16.0, 8.0, 17.0, 18.0, 18.0, 18.0, 18.0, 16.0, 14.0, 19.0, 17.0, 19.0, 15.0, 15.0, 16.0, 18.0, 21.0, 14.0, 15.0, 16.0, 17.0, 20.0, 14.0, 13.0, 16.0, 20.0, 17.0, 18.0, 21.0, 15.0, 14.0, 15.0, 17.0, 18.0, 18.0, 15.0, 13.0, 15.0, 14.0, 17.0, 8.0, 14.0, 16.0, 20.0, 9.0, 14.0, 19.0, 19.0, 17.0, 18.0, 18.0, 14.0, 14.0, 18.0, 18.0, 15.0, 15.0, 16.0, 15.0, 19.0, 18.0, 18.0, 15.0, 16.0, 17.0, 19.0, 17.0, 18.0, 18.0, 15.0, 17.0, 18.0, 15.0, 10.0, 21.0, 15.0, 14.0, 17.0, 16.0, 18.0, 21.0, 18.0, 14.0, 14.0, 15.0, 13.0, 18.0, 16.0, 20.0, 12.0, 16.0, 20.0, 17.0, 16.0, 15.0, 15.0, 19.0, 18.0, 14.0, 19.0, 19.0, 16.0, 20.0, 10.0, 18.0, 8.0, 15.0, 19.0, 21.0, 15.0, 16.0, 18.0, 14.0, 16.0, 15.0, 18.0, 15.0, 11.0, 9.0, 15.0, 20.0, 15.0, 16.0, 17.0, 18.0, 16.0, 19.0, 15.0, 18.0, 21.0, 17.0, 13.0, 20.0, 18.0, 18.0, 16.0, 16.0, 14.0, 18.0, 15.0, 14.0, 19.0, 17.0, 16.0, 16.0, 17.0, 16.0, 14.0, 18.0, 10.0, 18.0, 16.0, 19.0, 18.0, 17.0, 19.0, 16.0, 19.0, 20.0, 14.0, 18.0, 19.0, 17.0, 17.0, 18.0, 18.0, 18.0, 16.0, 17.0, 15.0, 16.0, 21.0, 18.0, 15.0, 16.0, 17.0, 14.0, 11.0, 17.0, 15.0, 18.0, 17.0, 16.0, 20.0, 15.0, 15.0, 18.0, 17.0, 17.0, 17.0, 12.0, 14.0, 16.0, 20.0, 18.0, 16.0, 17.0, 17.0, 15.0, 18.0, 15.0, 18.0, 17.0, 18.0]

TestX: [20.0, 8.0, 19.0, 14.0, 17.0, 14.0, 18.0, 11.0, 20.0, 18.0, 18.0, 16.0, 16.0, 18.0, 14.0, 14.0, 16.0, 16.0, 17.0, 14.0, 18.0, 18.0, 16.0, 16.0, 19.0, 16.0, 18.0, 17.0, 15.0, 20.0, 14.0, 20.0, 19.0, 17.0, 18.0, 16.0, 17.0, 17.0, 15.0, 19.0, 10.0, 17.0, 16.0, 17.0, 15.0, 20.0, 13.0, 18.0, 17.0, 17.0, 15.0, 16.0, 14.0, 17.0, 17.0, 15.0, 18.0, 8.0, 20.0, 16.0, 18.0, 14.0, 18.0, 15.0, 16.0, 12.0, 15.0, 19.0, 18.0, 18.0, 16.0, 17.0, 19.0, 18.0, 15.0, 15.0, 15.0, 17.0, 15.0, 19.0, 15.0, 16.0, 18.0, 25.0, 15.0, 17.0, 20.0, 20.0, 16.0, 18.0, 18.0, 16.0, 16.0, 16.0, 16.0, 19.0, 15.0, 16.0, 19.0, 15.0, 16.0, 17.0, 19.0, 18.0, 16.0, 17.0, 19.0, 18.0, 15.0, 15.0, 17.0, 15.0, 19.0, 15.0, 16.0, 18.0, 25.0, 15.0, 17.0, 20.0, 20.0, 16.0, 18.0, 18.0, 16.0, 16.0, 16.0, 16.0, 19.0, 15.0, 16.0, 17.0, 15.0, 16.0, 13.0, 19.0, 16.0, 16.0, 18.0, 15.0, 15.0, 18.0, 19.0, 16.0, 17.0, 14.0, 16.0, 15.0, 17.0, 16.0, 14.0, 15.0, 20.0, 20.0, 18.0, 16.0, 16.0, 16.0, 20.0, 18.0, 14.0, 19.0, 17.0, 16.0, 17.0, 15.0, 17.0, 18.0, 19.0, 18.0, 19.0, 18.0, 8.0, 17.0, 16.0, 17.0, 19.0, 18.0, 15.0, 15.0, 17.0, 19.0, 17.0, 21.0, 19.0, 16.0, 17.0, 20.0, 21.0, 20.0, 18.0, 20.0, 17.0, 18.0, 16.0, 17.0, 17.0, 16.0, 18.0, 17.0, 18.0, 19.0, 17.0, 15.0, 17.0, 16.0, 15.0, 16.0, 18.0, 10.0, 20.0, 15.0, 15.0, 17.0, 19.0, 15.0, 15.0, 17.0, 19.0, 17.0, 15.0, 15.0, 16.0, 15.0, 15.0, 17.0, 18.0]

Question 2:

TrainX: [[18.0, 57.0], [17.0, 70.0], [20.0, 61.7], [15.0, 59.0], [15.0, 48.0], [15.0, 56.2], [15.0, 52.2], [18.0, 56.6], [15.0, 66.0], [15.0, 53.2], [16.0, 51.5], [19.0, 62.0], [18.0, 52.0], [18.0, 57.8], [20.0, 70.0], [17.0, 56.5], [18.0, 64.2], [21.0, 53.4], [17.0, 61.5], [15.0, 55.0], [16.0, 56.0], [16.0, 54.0], [19.0, 59.0], [17.0, 59.5], [15.0, 55.0], [15.0, 57.8], [18.0, 72.5], [14.0, 53.0], [17.0, 62.5], [10.0, 61.0], [18.0, 48.0], [19.0, 60.0], [17.0, 59.4], [15.0, 51.8], [16.0, 53.0], [15.0, 53.4], [18.0, 68.0], [16.0, 60.5], [18.0, 67.5], [15.0, 58.0], [17.0, 42.0], [18.0, 69.0], [19.0, 57.5], [15.0, 56.9], [16.0, 57.0], [16.0, 54.5], [20.0, 68.5], [17.0, 49.5], [15.0, 54.0], [16.0, 50.0], [15.0, 53.5], [17.0, 50.5], [15.0, 50.0], [17.0, 64.5], [18.0, 60.0], [15.0, 57.0], [16.0, 52.0], [15.0, 62.4], [16.0, 57.2], [16.0, 63.0], [17.0, 58.4], [16.0, 47.5], [17.0, 49.0], [15.0, 56.6], [8.0, 56.5], [10.0, 72.0], [15.0, 48.0], [16.0, 62.3], [16.0, 61.0], [12.0, 57.0], [17.0, 52.0], [15.0, 58.5], [17.0, 59.0], [15.0, 62.4], [15.0, 53.0], [18.0, 47.0], [20.0, 75.5], [12.0, 53.0], [14.0, 61.5], [17.0, 60.8], [15.0, 55.0], [19.0, 72.0], [18.0, 56.2], [19.0, 74.0], [18.0, 58.3], [16.0, 68.0], [17.0, 57.6], [18.0, 71.0], [16.0, 58.4], [18.0, 64.6], [18.0, 55.0], [17.0, 60.5], [19.0, 55.0], [10.0, 46.0], [15.0, 53.0], [16.0, 57.0], [17.0, 50.0], [18.0, 55.0], [18.0, 67.0], [16.0, 45.2], [12.0, 54.0], [16.0, 57.5], [19.0, 52.7], [17.0, 57.3], [10.0, 46.5], [16.0, 60.6], [16.0, 56.5], [18.0, 53.0], [18.0, 55.9], [16.0, 51.8], [17.0, 57.5], [17.0, 58.6], [15.0, 56.5], [14.0, 52.0], [14.0, 56.0], [16.0, 56.5], [14.0, 56.0], [18.0, 58.0], [21.0, 59.6], [15.0, 52.5], [16.0, 55.0], [14.0, 50.0], [16.0, 49.0], [18.0, 71.5], [18.0, 67.0], [17.0, 48.5], [17.0, 55.2], [17.0, 63.0], [14.0, 55.9], [15.0, 60.0], [20.0, 71.0], [16.0, 56.9], [17.0, 52.2], [17.0, 50.5], [8.0, 62.0], [14.0, 58.0], [16.0, 57.0], [17.0, 56.0], [18.0, 63.7], [8.0, 60.0], [14.0, 53.5], [14.0, 58.0], [17.0, 56.0], [17.0, 56.0], [20.0, 70.0], [18.0, 69.0], [14.0, 50.5], [17.0, 62.5], [14.0, 57.0], [11.0, 53.0], [21.0, 59.8], [17.0, 61.5], [16.0, 53.4], [18.0, 65.4], [19.0, 59.0], [15.0, 66.5], [16.0, 56.6], [16.0, 61.6], [16.0, 51.4], [8.0, 53.0], [17.0, 58.0], [20.0, 60.0], [19.0, 60.0], [9.0, 58.0], [17.0, 53.3], [19.0, 60.6], [8.0, 55.0], [19.0, 72.5], [15.0, 53.5], [16.0, 58.0], [18.0, 57.0], [19.0, 54.0], [17.0, 56.0], [15.0, 57.0], [10.0, 50.0], [15.0, 55.0], [16.0, 58.2], [15.0, 53.6], [18.0, 69.0], [14.0, 53.5], [17.0, 53.0], [15.0, 54.7], [17.0, 53.5], [7.0, 53.0], [16.0, 56.5], [16.0, 52.5], [17.0, 53.0], [15.0, 62.0], [16.0, 59.4], [17.0, 56.0], [18.0, 53.0], [18.0, 55.9], [18.0, 56.5], [16.0, 61.0], [17.0, 60.1], [18.0, 73.0], [21.0, 61.2], [16.0, 64.2], [21.0, 59.0], [13.0, 59.5], [17.0, 52.7], [16.0, 53.0], [17.0, 59.7], [18.0, 73.0], [15.0, 59.5], [16.0, 65.5], [16.0, 61.0], [15.0, 55.2], [18.0, 65.0], [16.0, 51.5], [20.0, 61.5], [17.0, 53.6], [16.0, 59.5], [18.0, 55.0], [20.0, 62.0], [19.0, 39.1], [15.0, 62.0], [19.0, 56.5], [16.0, 60.0], [19.0, 48.5], [16.0, 69.0], [16.0, 75.0], [14.0, 55.0], [16.0, 54.8], [15.0, 52.8], [15.0, 57.5], [17.0, 59.8], [15.0, 55.5], [18.0, 55.2], [16.0, 58.6], [17.0, 65.2], [17.0, 58.8], [17.0, 59.0], [17.0, 58.6], [18.0, 51.0], [17.0, 54.0], [20.0, 73.0], [19.0, 55.0], [16.0, 48.5], [17.0, 51.5], [15.0, 48.0], [15.0, 56.5], [14.0, 52.5], [16.0, 57.5], [16.0, 57.2], [16.0, 55.8], [17.0, 57.0], [18.0, 56.8], [15.0, 53.0], [18.0, 69.5], [16.0, 57.0], [18.0, 70.5], [17.0, 58.5], [15.0, 59.5], [15.0, 57.0], [21.0, 64.0], [16.0, 53.9], [17.0, 64.9], [20.0, 70.5], [19.0, 57.5], [15.0, 53.5], [18.0, 59.5], [17.0, 52.0], [18.0, 55.0], [18.0, 51.5], [14.0, 58.0], [15.0, 59.0], [16.0, 55.0], [20.0, 53.5], [14.0, 56.0], [17.0, 67.0], [20.0, 62.0], [16.0, 60.0], [17.0, 57.0], [17.0, 55.0], [20.0, 73.0], [14.0, 54.5], [19.0, 61.5], [18.0, 65.0], [16.0, 59.3], [18.0, 52.1], [18.0, 56.0], [14.0, 57.0], [19.0, 56.0], [17.0, 54.8], [15.0, 56.4], [17.0, 54.0], [15.0, 58.9], [15.0, 47.0], [21.0, 52.3], [15.0, 58.8], [14.0, 59.5], [14.0, 53.0], [18.0, 58.2], [20.0, 52.0], [20.0, 63.0], [18.0, 50.0], [13.0, 66.0], [18.0, 39.3], [19.0, 53.5], [18.0, 70.0], [10.0, 45.5], [18.0, 70.0], [13.0, 46.5], [16.0, 61.0], [19.0, 58.0], [16.0, 58.5], [18.0, 62.5], [15.0, 52.5], [18.0, 57.0], [18.0, 55.0], [16.0, 61.4], [15.0, 57.0], [18.0, 64.5], [15.0, 59.0], [14.0, 56.0], [17.0, 55.0], [16.0, 60.6], [17.0, 55.2], [7.0, 51.0], [15.0, 54.0], [17.0, 56.1], [21.0, 71.5], [17.0,

TestX: [[19.0, 59.4], [15.0, 53.0], [18.0, 60.0], [17.0, 60.0], [9.0, 46.0], [14.0, 48.0], [16.0, 63.5], [16.0, 57.2], [18.0, 61.0], [20.0, 52.0], [14.0, 57.5], [20.0, 58.0], [19.0, 61.5], [14.0, 53.5], [20.0, 57.0], [15.0, 56.5], [20.0, 54.0], [18.0, 74.0], [18.0, 57.0], [17.0, 51.0], [18.0, 51.0], [18.0, 54.0], [16.0, 51.5], [18.0, 74.0], [15.0, 58.3], [15.0, 60.0], [17.0, 51.9], [18.0, 60.5], [18.0, 73.0], [16.0, 59.0], [17.0, 64.5], [20.0, 56.0], [15.0, 54.2], [19.0, 75.0], [17.0, 59.3], [20.0, 71.0], [18.0, 60.0], [15.0, 57.1], [18.0, 58.0], [18.0, 51.0], [18.0, 68.5], [20.0, 59.5], [15.0, 56.7], [16.0, 57.5], [18.0, 50.0], [14.0, 59.2], [16.0, 64.0], [16.0, 59.4], [17.0, 60.0], [16.0, 59.0], [17.0, 55.0], [15.0, 56.0], [17.0, 61.0], [15.0, 61.0], [15.0, 50.0], [18.0, 59.0], [20.0, 55.0], [20.0, 45.8], [19.0, 57.7], [18.0, 67.0], [14.0, 51.0], [15.0, 61.0], [19.0, 68.0], [11.0, 51.5], [17.0, 62.0], [17.0, 49.7], [16.0, 54.0], [15.0, 55.5], [15.0, 59.4], [19.0, 76.0], [13.0, 58.5], [17.0, 57.5], [15.0, 55.5], [16.0, 65.5], [17.0, 58.2], [14.0, 53.5], [17.0, 56.0], [18.0, 71.0], [19.0, 57.0], [18.0, 64.5], [16.0, 54.2], [15.0, 57.0], [18.0, 72.0], [18.0, 61.0], [13.0, 60.0], [22.0, 59.0], [15.0, 55.0], [20.0, 56.0], [18.0, 56.0], [17.0, 58.0], [17.0, 52.9], [18.0, 62.0], [18.0, 72.5], [15.0, 51.5], [17.0, 56.5], [19.0, 62.0], [16.0, 54.0], [17.0, 55.8], [19.0, 52.2], [19.0, 50.0], [16.0, 51.0], [16.0, 51.0], [16.0, 50.0], [16.0, 59.6], [16.0, 56.0], [21.0, 60.0], [18.0, 54.5], [18.0, 60.0], [18.0, 72.0], [15.0, 57.0], [16.0, 56.0], [16.0, 53.5], [16.0, 63.0], [19.0, 52.3], [16.0, 62.7], [13.0, 55.0], [13.0, 56.0], [17.0, 54.7], [20.0, 56.6], [19.0, 57.2], [16.0, 58.5], [14.0, 56.0], [19.0, 54.0], [17.0, 51.5], [14.0, 56.7], [20.0, 72.0], [7.0, 43.0], [19.0, 72.5], [14.0, 58.6], [16.0, 66.6], [16.0, 64.5], [16.0, 56.5], [21.0, 74.0], [19.0, 55.0], [14.0, 45.5], [17.0, 65.5], [16.0, 53.2], [18.0, 54.0], [20.0, 59.0], [14.0, 59.8], [17.0, 62.0], [17.0, 51.0], [17.0, 54.2], [17.0, 62.0], [14.0, 60.0], [17.0, 56.3], [17.0, 49.2], [17.0, 54.2], [18.0, 56.3], [18.0, 49.3], [18.0, 50.0], [16.0, 58.2], [13.0, 58.0], [15.0, 60.0], [15.0, 57.8], [15.0, 65.0], [18.0, 73.0], [16.0, 51.0], [18.0, 53.6], [16.0, 51.7], [17.0, 56.7], [18.0, 52.8], [17.0, 60.0], [16.0, 52.0], [14.0, 53.0], [17.0, 57.0], [16.0, 53.3], [20.0, 57.0], [17.0, 55.3], [16.0, 54.9], [21.0, 62.6], [15.0, 57.0], [20.0, 72.0], [19.0, 53.0], [21.0, 56.5], [11.0, 45.5], [18.0, 53.0], [17.0, 62.8], [15.0, 55.0], [21.0, 58.5], [15.0, 55.1], [14.0, 58.0], [15.0, 49.5], [19.0, 66.5], [15.0, 65.0], [14.0, 61.0], [17.0, 58.0], [19.0, 38.0], [16.0, 49.0], [20.0, 55.0], [15.0, 55.2], [13.0, 56.0], [18.0, 65.3], [14.0, 48.0], [19.0, 61.0], [17.0, 59.0], [16.0, 58.5], [18.0, 61.5], [20.0, 60.7], [19.0, 54.8], [17.0, 57.0], [16.0, 59.0], [17.0, 63.3], [18.0, 55.0], [16.0, 64.4], [18.0, 63.0], [18.0, 61.2], [19.0, 64.5], [16.0, 70.0], [18.0, 57.0], [18.0, 70.0], [18.0, 72.0], [16.0, 64.5], [18.0, 56.5], [19.0, 55.5], [19.0, 62.3], [17.0, 54.0], [9.0, 51.5], [21.0, 58.0], [16.0, 55.0], [14.0, 56.0], [16.0, 62.0], [17.0, 53.5], [16.0, 59.0], [20.0, 70.5]]

Question 3 ratio:

TrainX =

[5.036231884057971, 4.6415094339622645, 4.351851851852, 4.776785714285714, 4.825174825174825, 8.043478260869566, 4.435483870967742, 4.285714285714286, 4.042553191489362, 4.265734265734266, 5.105691056910569, 4.448818897637795, 4.365079365079366, 4.3181818181818, 4.32, 4.966101694915254, 4.790322580645161, 4.453781512605042, 4.521739130434782, 4.584070796460177, 4.852173913043478, 4.696969696969697, 4.768518518518, 4.62809917355372, 4.745454545454546, 5.0, 4.2734375, 4.769230769230769, 4.84375, 4.2708333333333, 5.07923076923077, 4.64, 4.80314962099213, 4.314516129032258, 4.444444444444444, 4.66666666666667, 4.908256880739345, 3.5, 4.467213114754099, 4.47945205479452, 4.483606557377049, 4.4071428571428575, 4.576991150424477, 4.517543859649122, 4.690375, 4.11578947368421, 4.741379310344827, 4.9180276885246, 4.8111965811965812, 4.4296875, 4.417475728155339, 4.201388888888888, 4.27417943548387097, 4.683760683760684, 4.026845637583892, 4.869565217391305, 4.946902654867256, 4.645669291338583, 4.2481203007518795, 4.6, 4.214876303507518795, 4.67287919621682, 4.46564585491632, 4.20353982300885, 4.3958333333333, 5.070325, 4.96478732394366, 4.21383477978471, 5.043103448275862, 5.56359077443606, 4.375, 4.34108527131789, 7.12641509433962, 4.214876303507518795, 4.63278673478633, 4.104938271604934, 4.573643410852713]

4.126506024096385, 4.583333333333333, 4.638655462184874, 4.5, 5.321100917431193, 5.041322314049587, 4.878048780487805, 4.793388429752066, 4.827586206896552, 4.9298245614035086, 4.23728813559322, 4.402985074626866, 5.126050420168067, 4.236641221374046, 4.846153846153847, 4.793388429752066, 5.148760330578512, 4.078947368421053, 4.743119266055046, 4.082191780821918, 4.663934426229508, 5.140845070422536, 4.436619718309859, 4.263565891472868, 4.456692913385827, 4.5528455284552845, 4.322033898305085, 4.28448275862069, 4.912, 4.275362318840579, 4.09420289850725, 4.273504273504273, 4.9692307692307685, 4.278688524590164, 4.351145038167939, 5.084745762711864, 4.1735537190082646, 4.867256637168142, 4.186046511627906, 4.471544715447155, 4.9576271186440675, 4.4639999999999995, 4.547826086956522, 4.301470588235294, 4.892086330935252, 4.790322580645161, 4.793893129770992, 4.444444444444444, 4.608695652173913, 4.264705882352941, 4.62184739495798, 4.1640624999999999, 5.193798449612403, 4.45125, 4.918032788685246, 4.960629921259843, 4.33333333333333, 5.140186915887851, 5.0, 4.172661870503597, 4.360465116279065, 4.2063492063492065, 4.261744966429525, 4.507936507936508, 4.3359375, 4.354838709677419, 4.655172413793103, 4.351145038167939, 4.6727272727273, 4.758620689655173, 4.32, 5.034965034965035, 4.36936936936937, 4.403846153846153, 5.125, 4.5, 4.827586206896552, 4.338235294117647, 4.966666666666667, 4.424778761061947, 4.901785714285714, 4.351851851851851, 4.646017699115044, 4.227642276422764, 4.695652173913044, 4.956774719354839, 4.168, 4.819672131174541, 4.253731343283582, 4.4166666666666667, 4.924242424242425, 4.235294117647059, 4.2024539877300615, 4.50348275862069, 4.298780487804878, 5.040322580645161, 4.423076923070755, 4.4921875, 4.724409448818898, 3.2544378698224845, 4.583333333333333, 4.4444444444444445, 4.953703703703703, 5.3090909090909, 4.7069230769230775, 4.4921875, 4.724409448818898, 3.2544378698224845, 4.583333333333333, 4.4444444444444445, 4.953703703703703, 5.3090909090909, 4.7069230769230775, 4.4921875, 4.724409448818898, 3.2544378698224845, 4.583333333333333, 4.4444444444444445, 4.715447154471544, 5.136752136752137, 4.33333333333333, 4.30656943065694, 4.378698224852071, 4.390243902439024, 4.5528455284552845, 4.830508474576271, 4.951923076923077, 4.386792452830189, 4.285714285714286, 4.705882352941177, 4.855072463768115, 4.516129032258064, 5.391304347826087, 4.8062015503875966, 4.47457627118644, 4.688524590163935, 5.1171875, 4.503816793893125, 5.140186915887851, 5.0, 4.172661870503597, 4.360465116279065, 5.30, 5.3063063063063074, 4.210526162162162, 4.852459016393444, 4.423076923076923, 4.627897196216162, 5.188034188034188, 4.292035598230088, 4.67878908256887, 4.071942446043166, 4.246153846153846, 5.126126126126126, 4.852459016393444, 4.226190476190476, 4.982758620689655, 4.958333333333333, 4.4444444444444444, 4.322033898305085, 5.152, 5.073529411764706, 4.6302321008040336, 4.504273504273505, 4.363636363636363, 4.296296296296297, 4.478527607361963, 4.155405405405405, 4.24, 4.152857142857143, 4.256198347104738, 4.764227642276422, 4.34959345934593459, 4.900763358778626, 4.442622950819673, 4.9557521238983, 4.377049180327869, 4.206895521724138, 5.43809528095238, 4.401785714285714, 4.681818181818182, 5.083333333333333, 4.8999999999999995, 4.34959345934593459, 4.95, 4.962686567164179, 4.967213114754099, 4.8584905660337736, 4.716981132075472, 4.75177304964539, 4.566923076923077, 4.566923133858268, 4.700854700854701, 4.9523809523809526, 5.1171875, 4.503816793893125, 5.140186915887851, 4.35145038167939, 4.67580620689655173, 4.3165672589982, 4.9557521238983, 4.423076923076923, 4.627897196216162, 5.188034188034188, 4.292035598230088, 4.67878908256887, 4.071942446043166, 4.246153846153846, 5.126126126126126, 4.852459016393444, 4.226190476190476, 4.982758620689655, 4.958333333333333, 4.4444444444444444, 4.322033898305085, 5.152, 5.073529411764706, 4.6302321008040336, 4.504273504273505, 4.3636363636363, 4.296296296296297, 4.478527607361963, 4.155405405405405, 4.24, 4.152857142857143, 4.256198347104738, 4.764227642276422, 4.34959345934593459, 4.900763358778626, 4.442622950819673, 4.9557521238983, 4.377049180327869, 4.206895521724138, 5.43809528095238, 4.401785714285714, 4.681818181818182, 5.083333333333333, 4.8999999999999995, 4.34959345934593459, 4.95, 4.962686567164179, 4.967213114754099, 4.8584905660337736, 4.716981132075472, 4.75177304964539, 4.566923076923077, 4.566923133858268, 4.700854700854701, 4.9523809523809526, 5.1171875, 4.503816793893125, 5.140186915887851, 4.35145038167939, 4.67580620689655173, 4.3165672589982, 4.9557521238983, 4.423076923076923, 4.627897196216162, 5.188034188034188, 4.292035598230088, 4.67878908256887, 4.071942446043166, 4.246153846153846, 5.126126126126126, 4.852459016393444, 4.226190476190476, 4.982758620689655, 4.958333333333333, 4.4444444444444444, 4.322033898305085, 5.152, 5.073529411764706, 4.6302321008040336, 4.504273504273505, 4.3636363636363, 4.296296296296297, 4.478527607361963, 4.155405405405405, 4.24, 4.152857142857143, 4.256198347104738, 4.764227642276422, 4.34959345934593459, 4.900763358778626, 4.442622950819673, 4.9557521238983, 4.377049180327869, 4.206895521724138, 5.43809528095238, 4.401785714285714, 4.681818181818182, 5.083333333333333, 4.8999999999999995, 4.34959345934593459, 4.95, 4.962686567164179, 4.967213114754099, 4.8584905660337736, 4.716981132075472, 4.75177304964539, 4.566923076923077, 4.566923133858268, 4.700854700854701, 4.9523809523809526, 5.1171875, 4.503816793893125, 5.140186915887851, 4.35145038167939, 4.67580620689655173, 4.3165672589982, 4.9557521238983, 4.423076923076923, 4.627897196216162, 5.188034188034188, 4.292035598230088, 4.67878908256887, 4.071942446043166, 4.246153846153846, 5.126126126126126, 4.852459016393444, 4.226190476190476, 4.982758620689655, 4.958333333333333, 4.4444444444444444, 4.322033898305085, 5.152, 5.073529411764706, 4.6302321008040336, 4.504273504273505, 4.3636363636363, 4.296296296296297, 4.478527607361963, 4.155405405405405, 4.24, 4.152857142857143, 4.256198347104738, 4.764227642276422, 4.34959345934593459, 4.900763358778626, 4.442622950819673, 4.9557521238983, 4.377049180327869, 4.206895521724138, 5.43809528095238, 4.401785714285714, 4.681818181818182, 5.083333333333333, 4.8999999999999995, 4.34959345934593459, 4.95, 4.962686567164179, 4.967213114754099, 4.8584905660337736, 4.716981132075472, 4.75177304964539, 4.566923076923077, 4.566923133858268, 4.700854700854701, 4.9523809523809526, 5.1171875, 4.503816793893125, 5.140186915887851, 4.35145038167939, 4.67580620689655173, 4.3165672589982, 4.9557521238983, 4.423076923076923, 4.627897196216162, 5.188034188034188, 4.292035598230088, 4.67878908256887, 4.071942446043166, 4.246153846153846, 5.126126126126126, 4.852459016393444, 4.226190476190476, 4.982758620689655, 4.958333333333333, 4.4444444444444444, 4.322033898305085, 5.152, 5.073529411764706, 4.6302321008040336, 4.504273504273505, 4.3636363636363, 4.296296296296297, 4.478527607361963, 4.155405405405405, 4.24, 4.152857142857143, 4.256198347104738, 4.764227642276422, 4.34959345934593459, 4.900763358778626, 4.442622950819673, 4.9557521238983, 4.377049180327869, 4.206895521724138, 5.43809528095238, 4.401785714285714, 4.681818181818182, 5.083333333333333, 4.8999999999999995, 4.34959345934593459, 4.95, 4.962686567164179, 4.967213114754099, 4.8584905660337736, 4.716981132075472, 4.75177304964539, 4.566923076923077, 4.566923133858268, 4.700854700854701, 4.9523809523809526, 5.1171875, 4.503816793893125, 5.140186915887851, 4.35145038167939, 4.67580620689655173, 4.3165672589982, 4.9557521238983, 4.423076923076923, 4.627897196216162, 5.188034188034188, 4.292035598230088, 4.67878908256887, 4.071942446043166, 4.246153846153846, 5.126126126126126, 4.852459016393444, 4.226190476190476, 4.982758620689655, 4.958333333333333, 4.4444444444444444, 4.322033898305085, 5.152, 5.073529411764706, 4.6302321008040336, 4.504273504273505, 4.3636363636363, 4.296296296296297, 4.478527607361963, 4.155405405405405, 4.24, 4.152857142857143, 4.256198347104738, 4.764227642276422, 4.34959345934593459, 4.900763358778626, 4.442622950819673, 4.9557521238983, 4.377049180327869, 4.206895521724138, 5.43809528095238, 4.401785714285714, 4.681818181818182, 5.083333333333333, 4.8999999999999995, 4.34959345934593459, 4.95, 4.962686567164179, 4.967213114754099, 4.8584905660337736, 4.716981132075472, 4.75177304964539, 4.566923076923077, 4.566923133858268, 4.700854700854701, 4.9523809523809526, 5.1171875, 4.503816793893125, 5.140186915887851, 4.35145038167939, 4.67580620689655173, 4.3165672589982, 4.9557521238983, 4.423076923076923, 4.627897196216162, 5.188034188034188, 4.292035598230088, 4.67878908256887, 4.071942446043166, 4.246153846153846, 5.126126126126126, 4.852459016393444, 4.226190476190476, 4.982758620689655, 4.958333333333333, 4.4444444444444444, 4.322033898305085, 5.152, 5.073529411764706, 4.6302321008040336, 4.504273504273505, 4.3636363636363, 4.296296296296297, 4.478527607361963, 4.155405405405405, 4.24, 4.152857142857143, 4.256198347104738, 4.764227642276422, 4.34959345934593459, 4.900763358778626, 4.442622950819673, 4.9557521238983, 4.377049180327869, 4.206895521724138, 5.43809528095238, 4.401785714285714, 4.681818181818182, 5.083333333333333, 4.8999999999999995, 4.34959345934593459, 4.95, 4.962686567164179, 4.967213114754099, 4.8584905660337736, 4.716981132075472, 4.75177304964539, 4.566923076923077, 4.566923133858268, 4.700854700854701, 4.9523809523809526, 5.1171875, 4.503816793893125, 5.140186915887851, 4.35145038167939, 4.67580620689655173, 4.3165672589982, 4.9557521238983, 4.423076923076923, 4.627897196216162, 5.188034188034188, 4.292035598230088, 4.67878908256887, 4.071942446043166, 4.246153846153846, 5.126126126126126, 4.852459016393444, 4.226190476190476, 4.982758620689655, 4.958333333333333, 4.4444444444444444, 4.322033898305085, 5.152, 5.073529411764706, 4.6302321008040336, 4.504273504273505, 4.3636363636363, 4.296296296296297, 4.478527607361963, 4.155405405405405, 4.24, 4.152857142857143, 4.256198347104738, 4.764227642276422, 4.34959345934593459, 4.900763358778626, 4.442622950819673, 4.9557521238983, 4.377049180327869, 4.206895521724138, 5.43809528095238, 4.401785714285714, 4.681818181818182, 5.083333333333333, 4.8999999999999995, 4.34959345934593459, 4.95, 4.962686567164179, 4.967213114754099, 4.8584905660337736, 4.716981132075472, 4.75177304964539, 4.566923076923077, 4.566923133858268, 4.700854700854701, 4.9523809523809526, 5.1171875, 4.503816793893125, 5.140186915887851, 4.35145038167939, 4.67580620689655173, 4.3165672589982, 4.9557521238983, 4.423076923076923, 4.627897196216162, 5.188034188034188, 4.292035598230088, 4.67878908256887, 4.071942446043166, 4.246153846153846, 5.126126126126126, 4.852459016393444, 4.226190476190476, 4.982758620689655, 4.958333333333333, 4.4444444444444444, 4.322033898305085, 5.152, 5.073529411764706, 4.6302321008040336, 4.504273504273505, 4.3636363636363, 4.296296296296297, 4.478527607361963, 4.155405405405405, 4.24, 4.152857142857143, 4.256198347104738, 4.764227642276422, 4.34959345934

19.4, 29.0], [43.0, 27.0, 8.0, 22.0, 17.0, 18.0, 6.0, 5.0, 0.0, 1.0, 2.0, 4.0, 2.1, 56.1, 28.5, 18.8, 12.6, 8.3, 7.4, 5.5, 10.8, 20.3, 29.0], [43.0, 26.0, 10.0, 22.0, 15.0, 20.0, 5.0, 7.0, 1.0, 0.0, 2.0, 3.0, 2.0, 54.2, 27.3, 18.5, 12.0, 17.5, 6.6, 4.2, 8.3, 19.2, 30.0]]

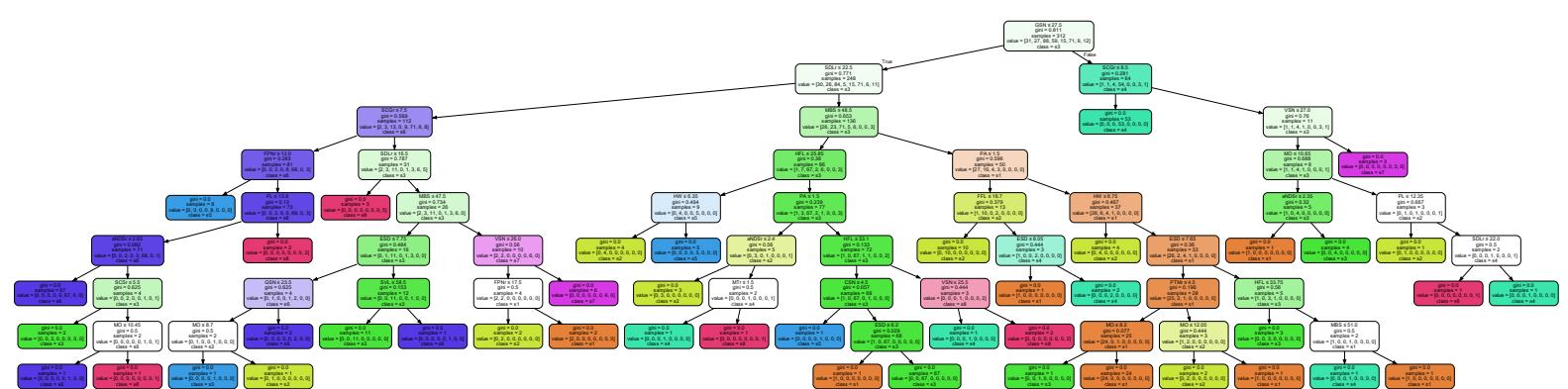
TestX: [44.0, 27.0, 8.0, 24.0, 16.0, 19.0, 5.0, 6.0, 1.0, 1.0, 2.0, 3.0, 2.1, 51.0, 25.5, 17.0, 10.9, 7.1, 6.2, 4.1, 9.4, 17.5, 27.4], [45.0, 25.0, 9.0, 25.0, 17.0, 19.0, 6.0, 6.0, 0.0, 1.0, 1.0, 2.9, 49.2, 22.2, 16.2, 11.6, 7.2, 7.0, 4.7, 9.9, 18.8, 27.8], [43.0, 27.0, 8.0, 23.0, 16.0, 21.0, 5.0, 4.0, 1.0, 1.0, 1.0, 2.0, 2.2, 6.1, 30.4, 18.8, 12.8, 7.9, 7.8, 4.5, 11.9, 20.3, 29.9], [43.0, 26.0, 8.0, 22.0, 18.0, 19.0, 5.0, 6.0, 1.0, 1.0, 1.0, 4.0, 2.1, 63.7, 33.5, 20.2, 12.8, 8.3, 8.0, 5.3, 10.9, 20.0, 30.8], [54.0, 28.0, 11.0, 20.0, 25.0, 19.0, 20.0, 6.0, 6.0, 1.0, 2.0, 2.0, 4.0, 2.6, 57.0, 28.0, 18.6, 11.8, 7.7, 6.6, 4.1, 9.5, 18.2, 26.8], [46.0, 28.0, 11.0, 22.0, 17.0, 20.0, 5.0, 7.0, 1.0, 1.0, 1.0, 4.0, 1.9, 55.0, 28.7, 18.0, 11.9, 7.6, 6.5, 4.7, 10.3, 19.6, 28.2], [41.0, 27.0, 9.0, 23.0, 17.0, 19.0, 6.0, 7.0, 1.0, 1.0, 4.0, 24.4, 50.0, 25.2, 17.8, 11.6, 7.3, 6.8, 4.6, 10.7, 18.5, 28.3], [43.0, 28.0, 9.0, 24.0, 16.0, 20.0, 6.0, 5.0, 1.0, 2.0, 2.0, 4.0, 2.0, 55.8, 27.0, 16.9, 11.5, 7.6, 7.1, 5.3, 8.9, 19.2, 25.3], [46.0, 26.0, 9.0, 21.0, 16.0, 22.0, 5.0, 7.0, 1.0, 1.0, 2.0, 3.0, 1.7, 48.5, 23.5, 15.8, 10.4, 6.7, 6.4, 4.2, 7.7, 17.8, 26.7], [56.0, 29.0, 12.0, 28.0, 17.0, 19.0, 6.0, 12.0, 2.0, 2.0, 2.0, 5.0, 2.1, 60.1, 33.4, 16.7, 11.7, 7.7, 6.4, 4.6, 9.6, 17.8, 27.2], [43.0, 28.0, 8.0, 23.0, 16.0, 18.0, 5.0, 5.0, 1.0, 1.0, 1.0, 2.0, 2.3, 60.8, 28.5, 18.6, 12.7, 8.4, 7.6, 5.6, 10.6, 19.8, 31.0], [48.0, 26.0, 9.0, 24.0, 17.0, 19.0, 6.0, 7.0, 0.0, 1.0, 2.0, 4.0, 2.2, 57.3, 28.8, 18.6, 12.2, 7.6, 6.8, 4.6, 9.0, 18.6, 29.4], [41.0, 27.0, 9.0, 23.0, 16.0, 20.0, 5.0, 6.0, 0.0, 1.0, 2.0, 4.0, 2.6, 65.5, 35.2, 19.0, 12.8, 8.4, 7.8, 6.0, 10.8, 20.2, 28.7], [42.0, 27.0, 9.0, 22.0, 15.0, 20.0, 5.0, 5.0, 1.0, 1.0, 2.0, 3.0, 2.3, 66.5, 36.2, 19.0, 13.4, 8.7, 7.4, 5.6, 12.0, 20.4, 30.3], [54.0, 27.0, 11.0, 25.0, 19.0, 20.0, 6.0, 14.0, 1.0, 2.0, 2.0, 5.0, 2.2, 59.0, 30.5, 18.8, 12.0, 7.5, 6.6, 4.5, 10.7, 18.6, 27.7], [43.0, 27.0, 8.0, 24.0, 16.0, 20.0, 5.0, 6.0, 1.0, 1.0, 1.0, 3.0, 2.0, 63.0, 33.8, 19.2, 13.0, 8.4, 7.7, 5.5, 11.4, 21.1, 32.4], [43.0, 27.0, 9.0, 22.0, 15.0, 19.0, 6.0, 1.0, 1.0, 2.0, 4.0, 2.7, 57.8, 31.0, 16.6, 11.5, 7.7, 6.9, 4.4, 10.2, 18.6, 27.0], [41.0, 28.0, 9.0, 23.0, 16.0, 19.0, 6.0, 5.0, 1.0, 1.0, 2.0, 4.0, 2.3, 60.0, 31.8, 19.3, 12.5, 7.7, 7.0, 4.9, 8.8, 20.2, 30.3], [41.0, 26.0, 10.0, 52.5, 27.3, 17.0, 11.2, 7.0, 6.4, 4.4, 9.6, 18.5, 27.4], [43.0, 26.0, 12.0, 20.0, 16.0, 19.0, 5.0, 5.0, 1.0, 1.0, 2.0, 4.0, 2.3, 60.0, 31.8, 19.3, 12.5, 7.7, 7.0, 4.9, 8.8, 20.2, 30.3], [41.0, 26.0, 10.0, 24.0, 16.0, 20.0, 6.0, 9.0, 1.0, 1.0, 2.0, 4.0, 2.5, 54.8, 26.8, 18.0, 11.7, 7.4, 7.0, 5.3, 9.7, 19.7, 29.5], [44.0, 27.0, 8.0, 25.0, 16.0, 20.0, 6.0, 6.0, 1.0, 1.0, 1.0, 2.0, 2.1, 62.4, 32.3, 19.7, 12.6, 8.3, 7.8, 5.1, 10.6, 19.7, 30.0], [45.0, 27.0, 10.0, 23.0, 16.0, 20.0, 6.0, 5.0, 1.0, 1.0, 2.0, 4.0, 2.5, 51.8, 24.7, 12.7, 11.3, 7.4, 7.1, 4.7, 9.3, 17.6, 7.8, 7.5, 4.7, 10.3, 19.5, 28.3], [42.0, 26.0, 9.0, 23.0, 15.0, 19.0, 5.0, 5.0, 1.0, 1.0, 1.0, 2.0, 2.1, 60.4, 31.2, 19.0, 12.9, 7.7, 7.0, 4.9, 8.8, 20.2, 30.3], [44.0, 29.0, 9.0, 25.0, 16.0, 21.0, 6.0, 5.0, 1.0, 1.0, 2.0, 4.0, 2.5, 51.8, 24.7, 12.7, 11.3, 7.4, 7.1, 4.7, 9.3, 17.6, 8.0], [42.0, 27.0, 10.0, 21.0, 15.0, 19.0, 6.0, 3.0, 0.0, 1.0, 1.0, 3.0, 2.1, 57.8, 28.1, 18.7, 11.6, 7.5, 7.3, 5.0, 10.8, 16.8, 29.2], [44.0, 26.0, 9.0, 22.0, 16.0, 18.0, 6.0, 4.0, 0.0, 1.0, 2.0, 4.0, 2.0, 50.5, 25.2, 18.2, 18.0, 12.0, 7.4, 7.2, 4.0, 8.5, 18.0, 28.6], [44.0, 27.0, 10.0, 26.0, 15.0, 19.0, 5.0, 4.0, 0.0, 1.0, 1.0, 4.0, 2.1, 65.0, 31.4, 19.9, 13.2, 8.4, 8.6, 5.6, 11.3, 19.0, 27.0], [42.0, 28.0, 8.0, 2.0, 7.5, 25.0, 17.0, 20.0, 6.0, 4.0, 1.0, 1.0, 2.0, 3.0, 2.4, 57.6, 28.4, 18.1, 11.8, 7.7, 7.2, 5.0, 10.4, 19.4, 27.9], [46.0, 26.0, 8.0, 23.0, 15.0, 19.0, 5.0, 6.0, 1.0, 1.0, 2.0, 4.0, 2.2, 51.5, 24.3, 16.7, 11.0, 7.3, 6.7, 4.8, 10.0, 17.8, 26.5], [43.0, 27.0, 9.0, 23.0, 17.0, 20.0, 5.0, 5.0, 1.0, 1.0, 2.0, 4.0, 2.5, 62.0, 34.2, 20.7, 12.9, 8.2, 8.2, 5.3, 10.9, 20.2], [30.5], [43.0, 28.0, 9.0, 25.0, 16.0, 20.0, 6.0, 6.0, 1.0, 1.0, 2.0, 3.0, 2.1, 64.5, 33.4, 19.6, 12.7, 8.2, 7.8, 5.4, 10.6, 20.9, 31.4], [55.0, 28.0, 10.0, 29.0, 19.0, 20.0, 5.0, 5.0, 1.0, 1.0, 2.0, 4.0, 2.5, 60.2, 30.5, 18.5, 13.2, 8.7, 7.4, 5.5, 11.0, 20.8, 31.5], [54.0, 28.0, 10.0, 28.5, 17.9, 11.8, 7.2, 6.5, 4.2, 10.8, 17.7, 26.0], [45.0, 27.0, 9.0, 20.0, 15.0, 19.0, 6.0, 5.0, 0.0, 1.0, 1.0, 4.0, 2.1, 62.0, 30.5, 18.5, 13.2, 8.7, 7.4, 5.5, 11.0, 20.8, 31.5], [54.0, 28.0, 10.0, 25.0, 17.0, 20.0, 5.0, 12.0, 1.0, 2.0, 2.0, 5.0, 2.5, 50.5, 25.8, 15.8, 10.8, 6.8, 6.0, 3.8, 9.2, 16.3, 26.3], [44.0, 27.0, 9.0, 23.0, 17.0, 20.0, 5.0, 6.0, 1.0, 2.0, 3.0, 2.1, 59.7, 31.0, 18.4, 12.2, 7.8, 7.6, 5.2, 10.0, 19.8, 28.6], [57.0, 29.0, 11.0, 30.0, 17.0, 20.0, 6.0, 11.0, 1.0, 3.0, 2.0, 4.0, 2.5, 58.6, 29.0, 18.4, 11.8, 7.4, 6.3, 4.3, 9.6, 19.3, 30.2], [43.0, 26.0, 9.0, 25.0, 16.0, 17.0, 6.0, 4.0, 1.0, 2.0, 3.0, 2.3, 58.5, 31.4, 18.1, 11.8, 7.3, 6.6, 4.8, 10.0, 19.7, 29.7], [44.0, 27.0, 10.0, 24.0, 16.0, 20.0, 6.0, 6.0, 1.0, 1.0, 3.0, 3.0, 2.1, 64.2, 31.3, 20.4, 13.1, 8.1, 7.3, 5.3, 10.7, 21.1, 31.8], [43.0, 27.0, 9.0, 24.0, 15.0, 19.0, 7.0, 6.0, 1.0, 1.0, 2.0, 3.0, 2.0, 55.2, 28.3, 17.7, 11.9, 7.7, 7.1, 4.8, 11.5, 19.6, 28.9], [43.0, 26.0, 9.0, 24.0, 17.0, 20.0, 6.0, 6.0, 1.0, 1.0, 2.0, 3.0, 2.1, 64.2, 31.3, 20.4, 13.1, 8.1, 7.3, 5.3, 10.7, 21.1, 31.8], [43.0, 27.0, 9.0, 26.6, 16.6, 11.0, 7.2, 6.8, 4.4, 8.6, 18.0, 26.0]]

Question 4 Subset b:

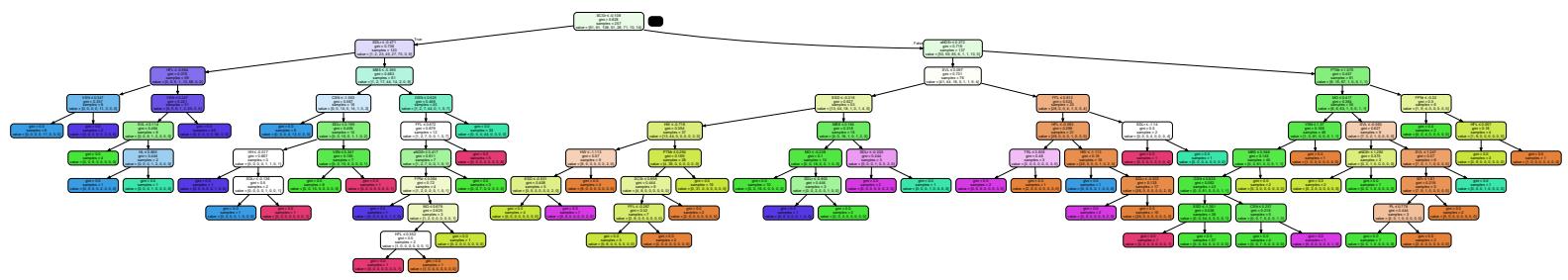
4.0, 1.0, 2.0, 2.0, 1.0, 2.0, 72.5, 40.0, 23.3, 14.3, 8.9, 9.0, 6.4, 9.9, 22.4, 33.0], [48.0, 25.0, 8.0, 28.0, 14.0, 25.0, 5.0, 9.0, 2.0, 2.0, 4.0, 2.6, 56.0, 32.2, 17.5, 12.3, 7.9, 8.5, 6.0, 7.8, 17.5, 29.0], [48.0, 25.0, 9.0, 31.0, 20.0, 24.0, 6.0, 2.0, 1.0, 2.0, 1.0, 1.0, 2.8, 72.0, 40.6, 21.6, 14.3, 9.0, 9.2, 6.5, 10.0, 23.4, 35.5], [50.0, 24.0, 8.0, 25.0, 14.0, 19.0, 6.0, 10.0, 1.0, 2.0, 2.0, 3.0, 2.7, 61.0, 30.0, 21.2, 14.0, 9.4, 8.6, 5.8, 11.2, 20.5, 31.0], [44.0, 24.0, 5.0, 24.0, 17.0, 19.0, 6.0, 9.0, 0.0, 0.0, 3.0, 4.0, 30.0, 52.0, 25.4, 17.3, 11.7, 7.8, 6.8, 5.5, 10.0, 19.0, 29.8], [43.0, 23.0, 7.0, 20.0, 16.0, 24.0, 5.0, 9.0, 0.0, 0.0, 2.0, 4.0, 27.5, 56.5, 27.5, 19.1, 12.6, 8.4, 7.5, 5.7, 9.7, 20.0, 29.5], [46.0, 27.0, 7.0, 24.0, 15.0, 26.0, 5.0, 7.0, 2.0, 2.0, 2.0, 4.0, 35.5, 30.0, 35.5, 16.8, 10.7, 7.3, 7.1, 4.8, 7.6, 17.5, 26.7], [38.0, 21.0, 5.0, 19.0, 9.0, 20.0, 6.0, 1.0, 0.0, 2.0, 2.0, 1.0, 2.0, 38.0, 24.1, 16.2, 11.1, 7.4, 7.1, 5.5, 7.4, 16.8, 25.3], [51.0, 23.0, 8.0, 32.0, 19.0, 24.0, 6.0, 6.0, 2.0, 2.0, 1.0, 1.0, 2.9, 62.0, 30.0, 24.0, 14.2, 9.1, 9.7, 7.2, 10.1, 23.2, 37.4]]

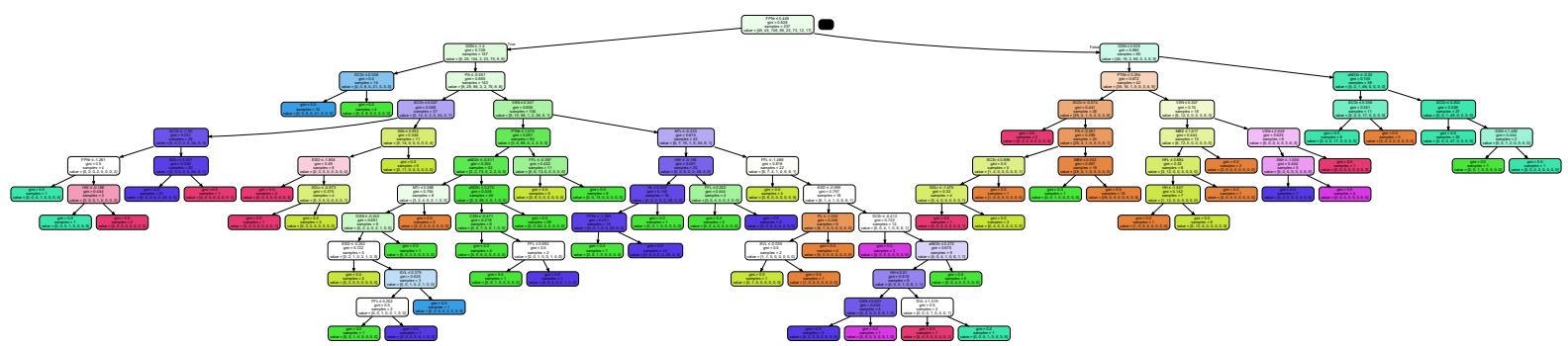
Question 5:

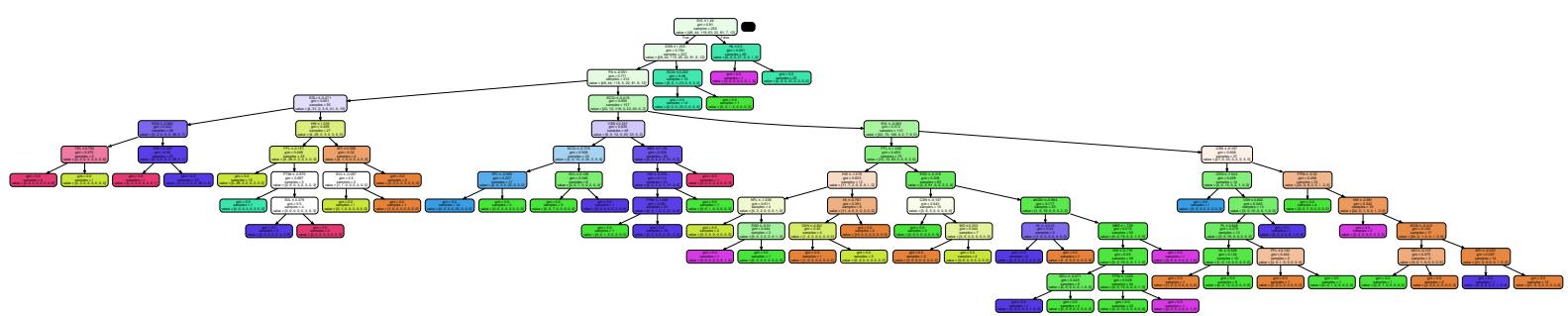
2.2, 58.0, 31.4, 18.3, 12.2, 7.2, 8.1, 5.6, 8.6, 19.5, 28.7], [45.0, 26.0, 8.0, 31.0, 18.0, 24.0, 6.0, 4.0, 2.0, 2.0, 1.0, 2.6, 63.0, 35.9, 17.8, 12.8, 7.9, 8.9, 6.0, 9.3, 21.5, 32.9], [43.0, 24.0, 7.0, 21.0, 14.0, 24.0, 7.0, 7.0, 1.0, 1.0, 2.0, 2.8, 58.0, 27.9, 18.8, 12.5, 8.3, 8.7, 5.8, 8.5, 19.2, 31.0], [55.0, 28.0, 11.0, 25.0, 17.0, 20.0, 7.0, 12.0, 1.0, 3.0, 2.0, 4.0, 2.1, 59.5, 30.6, 18.8, 12.0, 7.8, 6.5, 4.9, 10.3, 18.6, 26.6], [48.0, 24.0, 9.0, 23.0, 17.0, 24.0, 7.0, 12.0, 2.0, 2.0, 2.0, 2.8, 48.5, 27.8, 16.5, 11.1, 7.7, 6.8, 4.8, 8.8, 17.8, 26.8], [45.0, 25.0, 7.0, 23.0, 13.0, 6.0, 10.0, 0.0, 0.0, 3.0, 4.0, 2.7, 58.5, 32.7, 17.9, 11.6, 7.3, 7.0, 5.0, 7.9, 16.9, 27.0], [42.0, 26.0, 8.0, 22.0, 16.0, 24.0, 6.5, 7.2, 4.9, 10.1, 7.7, 12.2, 24.0, 33.3], [50.0, 24.0, 8.0, 24.0, 15.0, 24.0, 4.0, 12.0, 1.0, 1.0, 4.0, 2.1, 54.7, 28.6, 18.4, 12.8, 6.8, 7.3, 5.2, 11.3, 18.7, 26.8], [46.0, 23.0, 6.0, 23.0, 17.0, 18.0, 5.0, 8.0, 3.0, 2.0, 3.0, 2.4, 56.5, 30.5, 19.3, 13.0, 8.9, 7.6, 5.2, 9.5, 19.3, 29.5], [39.0, 24.0, 8.0, 17.0, 8.0, 16.0, 6.0, 9.0, 1.0, 2.0, 3.0, 3.0, 2.0, 6.2, 38.0, 18.3, 11.5, 7.4, 6.4, 5.4, 10.6, 18.5, 28.3], [43.0, 25.0, 6.0, 22.0, 18.0, 25.0, 5.0, 10.0, 1.0, 1.0, 3.0, 3.0, 2.8, 61.0, 38.3, 18.4, 12.1, 7.9, 7.0, 5.0, 8.0, 18.7, 28.5], [38.0, 21.0, 6.0, 16.0, 10.0, 25.0, 6.0, 0.0, 1.0, 2.0, 3.0, 1.0, 2.7, 46.5, 22.7, 15.7, 10.6, 7.3, 7.7, 5.1, 8.7, 16.5, 24.3], [45.0, 24.0, 9.0, 27.0, 14.0, 19.0, 5.0, 7.0, 2.0, 3.0, 2.0, 3.0, 3.0, 5.0, 28.6, 19.5, 11.7, 8.4, 7.2, 5.5, 9.5, 20.8, 31.0], [47.0, 22.0, 9.0, 24.0, 19.0, 26.0, 6.0, 12.0, 1.0, 2.0, 1.0, 6.0, 2.3, 55.0, 27.9, 18.6, 12.3, 6.2, 8.4, 4.3, 9.1, 17.3, 28.2], [44.0, 26.0, 9.0, 22.0, 16.0, 18.0, 6.0, 4.0, 0.0, 1.0, 1.0, 4.0, 2.3, 63.0, 33.9, 19.0, 12.7, 8.3, 7.7, 5.9, 10.9, 20.3, 29.7], [45.0, 24.0, 8.0, 28.0, 18.0, 23.0, 6.0, 8.0, 1.0, 2.0, 2.0, 1.0, 2.7, 70.5, 41.8, 21.7, 14.2, 9.3, 8.5, 6.2, 11.4, 22.0, 36.0], [41.0, 23.0, 7.0, 21.0, 13.0, 26.0, 5.0, 8.0, 1.0, 2.0, 5.0, 2.6, 55.0, 31.1, 18.3, 12.0, 8.0, 7.3, 5.3, 9.7, 19.2, 29.2], [43.0, 27.0, 8.0, 22.0, 17.0, 18.0, 6.0, 5.0, 0.0, 1.0, 2.0, 4.0, 2.1, 56.1, 28.5, 18.8, 12.6, 8.3, 7.4, 5.5, 10.8, 23.0, 29.1], [55.0, 28.0, 10.0, 29.0, 19.0, 20.0, 5.0, 10.0, 1.0, 2.0, 2.0, 5.0, 2.5, 54.0, 28.5, 17.9, 11.8, 7.2, 6.5, 4.2, 10.8, 17.7, 26.0], [42.0, 26.0, 7.0, 21.0, 17.0, 21.0, 5.0, 4.0, 1.0, 2.0, 3.0, 2.7, 62.8, 33.1, 19.0, 13.1, 8.4, 7.7, 4.2, 11.7, 21.2, 31.0], [51.0, 26.0, 9.0, 27.0, 19.0, 26.0, 8.0, 11.0, 2.0, 5.0, 2.0, 2.5, 55.5, 29.9, 19.5, 12.2, 8.4, 7.3, 5.5, 9.5, 20.2, 30.7], [50.0, 22.0, 11.0, 18.0, 26.0, 6.0, 4.0, 2.0, 2.0, 1.0, 2.0, 3.0, 3.0, 5.0, 28.6, 19.5, 11.7, 8.4, 7.2, 5.5, 9.5, 20.8, 32.0], [38.8, 23.0, 17.2, 10.7, 11.2, 7.9, 10.8, 26.5, 42.0], [48.0, 23.0, 12.0, 28.0, 18.0, 25.0, 6.0, 7.0, 1.0, 3.0, 1.0, 1.0, 2.6, 67.0, 37.8, 22.3, 15.9, 10.1, 11.5, 8.0, 10.9, 19.3, 38.3], [49.0, 25.0, 12.0, 24.0, 17.0, 25.0, 9.0, 10.0, 2.0, 5.0, 2.0, 2.3, 62.0, 29.0, 21.1, 14.0, 7.2, 9.1, 5.1, 10.2, 20.0, 33.0], [53.0, 25.0, 9.0, 22.0, 17.0, 25.0, 4.0, 10.0, 2.0, 4.0, 1.0, 3.0, 2.5, 56.7, 30.5, 19.4, 12.8, 6.7, 8.2, 5.1, 10.3, 18.6, 27.8], [47.0, 25.0, 10.0, 29.0, 21.0, 25.0, 7.0, 6.0, 2.0, 2.0, 1.0, 2.4, 60.0, 32.6, 18.9, 12.3, 7.6, 8.4, 5.5, 20.6, 31.5, 39.3], [57.0, 28.0, 11.0, 27.0, 17.0, 20.0, 5.0, 11.0, 2.0, 2.0, 5.0, 2.2, 57.0, 30.0, 18.0, 11.8, 7.4, 6.1, 4.7, 19.8, 27.0], [49.0, 23.0, 8.0, 26.0, 14.0, 24.0, 6.0, 11.0, 2.0, 2.0, 3.0, 5.0, 2.8, 53.0, 25.2, 18.5, 11.7, 7.7, 7.3, 4.6, 8.5, 19.4, 29.5], [42.0, 24.0, 10.0, 18.0, 12.0, 17.0, 5.0, 10.0, 1.0, 2.0, 2.0, 4.0, 2.4, 57.0, 29.8, 20.0, 12.6, 8.4, 8.0, 5.8, 9.4, 20.3, 31.5], [44.0, 23.0, 8.0, 23.0, 18.0, 25.0, 6.0, 10.0, 1.0, 2.0, 3.0, 2.4, 57.6, 28.4, 18.1, 11.8, 7.7, 7.2, 5.0, 10.4, 19.4, 27.9], [50.0, 24.0, 8.0, 25.0, 14.0, 19.0, 6.0, 10.0, 1.0, 2.0, 3.0, 2.7, 61.0, 30.0, 21.2, 14.0, 9.4, 8.6, 5.8, 11.2, 20.5, 31.0], [36.0, 25.0, 7.0, 17.0, 9.0, 16.0, 5.0, 3.0, 1.0, 2.0, 3.0, 3.0, 2.2, 60.0, 36.0, 16.9, 11.4, 7.3, 6.5, 5.2, 10.1, 18.2, 26.0], [44.0, 27.0, 8.0, 23.0, 16.0, 19.0, 5.0, 5.0, 1.0, 0.0, 3.0, 3.0, 2.6, 59.4, 30.8, 18.1, 12.4, 7.8, 7.2, 5.4, 9.9, 20.5, 28.6], [45.0, 22.0, 9.0, 28.0, 18.0, 23.0, 6.0, 9.0, 3.0, 1.0, 2.0, 3.0, 2.0, 53.4, 29.8, 16.7, 11.0, 5.7, 7.0, 4.8, 9.2, 16.4, 28.0], [49.0, 24.0, 9.0, 31.0, 19.0, 25.0, 6.0, 2.0, 1.0, 2.0, 1.0, 2.5, 72.0, 39.5, 24.4, 17.3, 10.7, 10.6, 7.9, 12.5, 26.3, 38.5], [50.0, 28.0, 10.0, 33.0, 20.0, 24.0, 7.0, 4.0, 1.0, 3.0, 3.0, 2.8, 53.0, 25.2, 18.5, 11.7, 7.7, 7.3, 4.6, 8.5, 19.4, 29.5], [42.0, 24.0, 10.0, 18.0, 12.0, 17.0, 5.0, 10.0, 1.0, 2.0, 2.0, 4.0, 2.4, 57.0, 29.8, 20.0, 12.6, 8.4, 8.0, 5.8, 9.4, 20.3, 31.5], [44.0, 23.0, 8.0, 23.0, 18.0, 25.0, 5.0, 10.0, 1.0, 2.0, 3.0, 2.4, 57.6, 28.4, 18.1, 11.8, 7.7, 7.2, 5.0, 10.4, 19.4, 27.9], [50.0, 24.0, 8.0, 25.0, 14.0, 19.0, 6.0, 10.0, 1.0, 2.0, 3.0, 2.7, 61.0, 30.0, 21.2, 14.0, 9.4, 8.6, 5.8, 11.2, 20.5, 31.0], [36.0, 25.0, 7.0, 17.0, 9.0, 16.0, 5.0, 3.0, 1.0, 2.0, 3.0, 3.0, 2.2, 60.0, 36.0, 16.9, 11.4, 7.3, 6.5, 5.2, 10.1, 18.2, 26.0], [44.0, 27.0, 8.0, 23.0, 16.0, 19.0, 5.0, 5.0, 1.0, 0.0, 3.0, 3.0, 2.6, 59.4, 30.8, 18.1, 12.4, 7.8, 7.2, 5.4, 9.9, 20.5, 28.6], [45.0, 22.0, 9.0, 28.0, 18.0, 23.0, 6.0, 9.0, 3.0, 1.0, 2.0, 3.0, 2.0, 53.4, 29.8, 16.7, 11.0, 5.7, 7.0, 4.8, 9.2, 16.4, 28.0], [49.0, 24.0, 9.0, 31.0, 19.0, 25.0, 6.0, 2.0, 1.0, 2.0, 1.0, 2.5, 72.0, 39.5, 24.4, 17.3, 10.7, 10.6, 7.9, 12.5, 26.3, 38.5], [50.0, 28.0, 10.0, 33.0, 20.0, 24.0, 7.0, 4.0, 1.0, 3.0, 3.0, 2.8, 53.0, 25.2, 18.5, 11.7, 7.7, 7.3, 4.6, 8.5, 19.4, 29.5], [42.0, 24.0, 10.0, 18.0, 12.0, 17.0, 5.0, 10.0, 1.0, 2.0, 2.0, 4.0, 2.4, 57.0, 29.8, 20.0, 12.6, 8.4, 8.0, 5.8, 9.4, 20.3, 31.5], [44.0, 23.0, 8.0, 23.0, 18.0, 25.0, 5.0, 10.0, 1.0, 2.0, 3.0, 2.4, 57.6, 28.4, 18.1, 11.8, 7.7, 7.2, 5.0, 10.4, 19.4, 27.9], [50.0, 24.0, 8.0, 25.0, 14.0, 19.0, 6.0, 10.0, 1.0, 2.0, 3.0, 2.7, 61.0, 30.0, 21.2, 14.0, 9.4, 8.6, 5.8, 11.2, 20.5, 31.0], [36.0, 25.0, 7.0, 17.0, 9.0, 16.0, 5.0, 3.0, 1.0, 2.0, 3.0, 3.0, 2.2, 60.0, 36.0, 16.9, 11.4, 7.3, 6.5, 5.2, 10.1, 18.2, 26.0], [44.0, 27.0, 8.0, 23.0, 16.0, 19.0, 5.0, 5.0, 1.0, 0.0, 3.0, 3.0, 2.6, 59.4, 30.8, 18.1, 12.4, 7.8, 7.2, 5.4, 9.9, 20.5, 28.6], [45.0, 22.0, 9.0, 28.0, 18.0, 23.0, 6.0, 9.0, 3.0, 1.0, 2.0, 3.0, 2.0, 53.4, 29.8, 16.7, 11.0, 5.7, 7.0, 4.8, 9.2, 16.4, 28.0], [49.0, 24.0, 9.0, 31.0, 19.0, 25.0, 6.0, 2.0, 1.0, 2.0, 1.0, 2.5, 72.0, 39.5, 24.4, 17.3, 10.7, 10.6, 7.9, 12.5, 26.3, 38.5], [50.0, 28.0, 10.0, 33.0, 20.0, 24.0, 7.0, 4.0, 1.0, 3.0, 3.0, 2.8, 53.0, 25.2, 18.5, 11.7, 7.7, 7.3, 4.6, 8.5, 19.4, 29.5], [42.0, 24.0, 10.0, 18.0, 12.0, 17.0, 5.0, 10.0, 1.0, 2.0, 2.0, 4.0, 2.4, 57.0, 29.8, 20.0, 12.6, 8.4, 8.0, 5.8, 9.4, 20.3, 31.5], [44.0, 23.0, 8.0, 23.0, 18.0, 25.0, 5.0, 10.0, 1.0, 2.0, 3.0, 2.4, 57.6, 28.4, 18.1, 11.8, 7.7, 7.2, 5.0, 10.4, 19.4, 27.9], [50.0, 24.0, 8.0, 25.0, 14.0, 19.0, 6.0, 10.0, 1.0, 2.0, 3.0, 2.7, 61.0, 30.0, 21.2, 14.0, 9.4, 8.6, 5.8, 11.2, 20.5, 31.0], [36.0, 25.0, 7.0, 17.0, 9.0, 16.0, 5.0, 3.0, 1.0, 2.0, 3.0, 3.0, 2.2, 60.0, 36.0, 16.9, 11.4, 7.3, 6.5, 5.2, 10.1, 18.2, 26.0], [44.0, 27.0, 8.0, 23.0, 16.0, 19.0, 5.0, 5.0, 1.0, 0.0, 3.0, 3.0, 2.6, 59.4, 30.8, 18.1, 12.4, 7.8, 7.2, 5.4, 9.9, 20.5, 28.6], [45.0, 22.0, 9.0, 28.0, 18.0, 23.0, 6.0, 9.0, 3.0, 1.0, 2.0, 3.0, 2.0, 53.4, 29.8, 16.7, 11.0, 5.7, 7.0, 4.8, 9.2, 16.4, 28.0], [49.0, 24.0, 9.0, 31.0, 19.0, 25.0, 6.0, 2.0, 1.0, 2.0, 1.0, 2.5, 72.0, 39.5, 24.4, 17.3, 10.7, 10.6, 7.9, 12.5, 26.3, 38.5], [50.0, 28.0, 10.0, 33.0, 20.0, 24.0, 7.0, 4.0, 1.0, 3.0, 3.0, 2.8, 53.0, 25.2, 18.5, 11.7, 7.7, 7.3, 4.6, 8.5, 19.4, 29.5], [42.0, 24.0, 10.0, 18.0, 12.0, 17.0, 5.0, 10.0, 1.0, 2.0, 2.0, 4.0, 2.4, 57.0, 29.8, 20.0, 12.6, 8.4, 8.0, 5.8, 9.4, 20.3, 31.5], [44.0, 23.0, 8.0, 23.0, 18.0, 25.0, 5.0, 10.0, 1.0, 2.0, 3.0, 2.4, 57.6, 28.4, 18.1, 11.8, 7.7, 7.2, 5.0, 10.4, 19.4, 27.9], [50.0, 24.0, 8.0, 25.0, 14.0, 19.0, 6.0, 10.0, 1.0, 2.0, 3.0, 2.7, 61.0, 30.0, 21.2, 14.0, 9.4, 8.6, 5.8, 11.2, 20.5, 31.0], [36.0, 25.0, 7.0, 17.0, 9.0, 16.0, 5.0, 3.0, 1.0, 2.0, 3.0, 3.0, 2.2, 60.0, 36.0, 16.9, 11.4, 7.3, 6.5, 5.2, 10.1, 18.2, 26.0], [44.0, 27.0, 8.0, 23.0, 16.0, 19.0, 5.0, 5.0, 1.0, 0.0, 3.0, 3.0, 2.6, 59.4, 30.8, 18.1, 12.4, 7.8, 7.2, 5.4, 9.9, 20.5, 28.6], [45.0, 22.0, 9.0, 28.0, 18.0, 23.0, 6.0, 9.0, 3.0, 1.0, 2.0, 3.0, 2.0, 53.4, 29.8, 16.7, 11.0, 5.7, 7.0, 4.8, 9.2, 16.4, 28.0], [49.0, 24.0, 9.0, 31.0, 19.0, 25.0, 6.0, 2.0, 1.0, 2.0, 1.0, 2.5, 72.0, 39.5, 24.4, 17.3, 10.7, 10.6, 7.9, 12.5, 26.3, 38.5], [50.0, 28.0, 10.0, 33.0, 20.0, 24.0, 7.0, 4.0, 1.0, 3.0, 3.0, 2.8, 53.0, 25.2, 18.5, 11.7, 7.7, 7.3, 4.6, 8.5, 19.4, 29.5], [42.0, 24.0, 10.0, 18.0, 12.0, 17.0, 5.0, 10.0, 1.0, 2.0, 2.0, 4.0, 2.4, 57.0, 29.8, 20.0, 12.6, 8.4, 8.0, 5.8, 9.4, 20.3, 31.5], [44.0, 23.0, 8.0, 23.0, 18.0, 25.0, 5.0, 10.0, 1.0, 2.0, 3.0, 2.4, 57.6, 28.4, 18.1, 11.8, 7.7, 7.2, 5.0, 10.4, 19.4, 27.9], [50.0, 24.0, 8.0, 25.0, 14.0, 19.0, 6.0, 10.0, 1.0, 2.0, 3.0, 2.7, 61.0, 30.0, 21.2, 14.0, 9.4, 8.6, 5.8, 11.2, 20.5, 31.0], [36.0, 25.0, 7.0, 17.0, 9.0, 16.0, 5.0, 3.0, 1.0, 2.0, 3.0, 3.0, 2.2, 60.0, 36.0, 16.9, 11.4, 7.3, 6.5, 5.2, 10.1, 18.2, 26.0], [44.0, 27.0, 8.0, 23.0, 16.0, 19.0, 5.0, 5.0, 1.0, 0.0, 3.0, 3.0, 2.6, 59.4, 30.8, 18.1, 12.4, 7.8, 7.2, 5.4, 9.9, 20.5, 28.6], [45.0, 22.0, 9.0, 28.0, 18.0, 23.0, 6.0, 9.0, 3.0, 1.0, 2.0, 3.0, 2.0, 53.4, 29.8, 16.7, 11.0, 5.7, 7.0, 4.8, 9.2, 16.4, 28.0], [49.0, 24.0, 9.0, 31.0, 19.0, 25.0, 6.0, 2.0, 1.0, 2.0, 1.0, 2.5, 72.0, 39.5, 24.4, 17.3, 10.7, 10.6, 7.9, 12.5, 26.3, 38.5], [50.0, 28.0, 10.0, 33.0, 20.0, 24.0, 7.0, 4.0, 1.0, 3.0, 3.0, 2.8, 53.0, 25.2, 18.5, 11.7, 7.7, 7.3, 4.6, 8.5, 19.4, 29.5], [42.0, 24.0, 10.0, 18.0, 12.0, 17.0, 5.0, 10.0, 1.0, 2.0, 2.0, 4.0, 2.4, 57.0, 29.8, 20.0, 12.6, 8.4, 8.0, 5.8, 9.4, 20.3, 31.5], [44.0, 23.0, 8.0, 23.0, 18.0, 25.0, 5.0, 10.0, 1.0, 2.0, 3.0, 2.4, 57.6, 28.4, 18.1, 11.8, 7.7, 7.2, 5.0, 10.4, 19.4, 27.9], [50.0, 24.0, 8.0, 25.0, 14.0, 19.0, 6.0, 10.0, 1.0, 2.0, 3.0, 2.7, 61.0, 30.0, 21.2, 14.0, 9.4, 8.6, 5.8, 11.2, 20.5, 31.0], [36.0, 25.0, 7.0, 17.0, 9.0, 16.0, 5.0, 3.0, 1.0, 2.0, 3.0, 3.0, 2.2, 60.0, 36.0, 16.9, 11.4, 7.3, 6.5, 5.2, 10.1, 18.2, 26.0], [44.0, 27.0, 8.0, 23.0, 16.0, 19.0, 5.0, 5.0, 1.0, 0.0, 3.0, 3.0, 2.6, 59.4, 30.8, 18.1, 12.4, 7.8, 7.2, 5.4, 9.9, 20.5, 28.6], [45.0, 22.0, 9.0, 28.0, 18.0, 23.0, 6.0, 9.0, 3.0, 1.0, 2.0, 3.0, 2.0, 53.4, 29.8, 16.7, 11.0, 5.7, 7.0, 4.8, 9.2, 16.4, 28.0], [49.0, 24.0, 9.0, 31.0, 19.0, 25.0, 6.0, 2.0, 1.0, 2.0, 1.0, 2.5, 72.0, 39.5, 24.4, 17.3, 10.7, 10.6, 7.9, 12.5, 26.3, 38.5], [50.0, 28.0, 10.0, 33.0, 20.0, 24.0, 7.0, 4.0, 1.0, 3.0, 3.0, 2.8, 53.0, 25.2, 18.5, 11.7, 7.7, 7.3, 4.6, 8.5, 19.4, 29.5], [42.0, 24.0, 10.0, 18.0, 12.0, 17.0, 5.0, 10.0, 1.0, 2.0, 2.0, 4.0, 2.4, 57.0, 29.8, 20.0, 12.6, 8.4, 8.0, 5.8, 9.4, 20.3, 31.5], [44.0, 23.0, 8.0, 23.0, 18.0, 25.0, 5.0, 10.0, 1.0, 2.0, 3.0, 2.4, 57.6, 28.4, 18.1, 11.8, 7.7, 7.2, 5.0, 10.4, 19.4, 27.9], [50.0, 24.0, 8.0, 25.0, 14.0, 19.0, 6.0, 10.0, 1.0, 2.0, 3.0, 2.7, 61.0, 30.0, 21.2, 14.0, 9.4, 8.6, 5.8, 11.2, 20.5, 31.0], [36.0, 25.0, 7.0, 17.0, 9.0, 16.0, 5.0, 3.0, 1.0, 2.0, 3.0, 3.0, 2.2, 60.0, 36.0, 16.9, 11.4, 7.3, 6.5, 5.2, 10.1, 18.2, 26.0], [44.0, 27.0, 8.0, 23.0, 16.0, 19.0, 5.0, 5.0, 1.0, 0.0, 3.0, 3.0, 2.6, 59.4, 30.8, 18.1, 12.4, 7.8, 7.2, 5.4, 9.9, 20.5, 28.6], [45.0, 22.0, 9.0, 28.0, 18.0, 23.0,

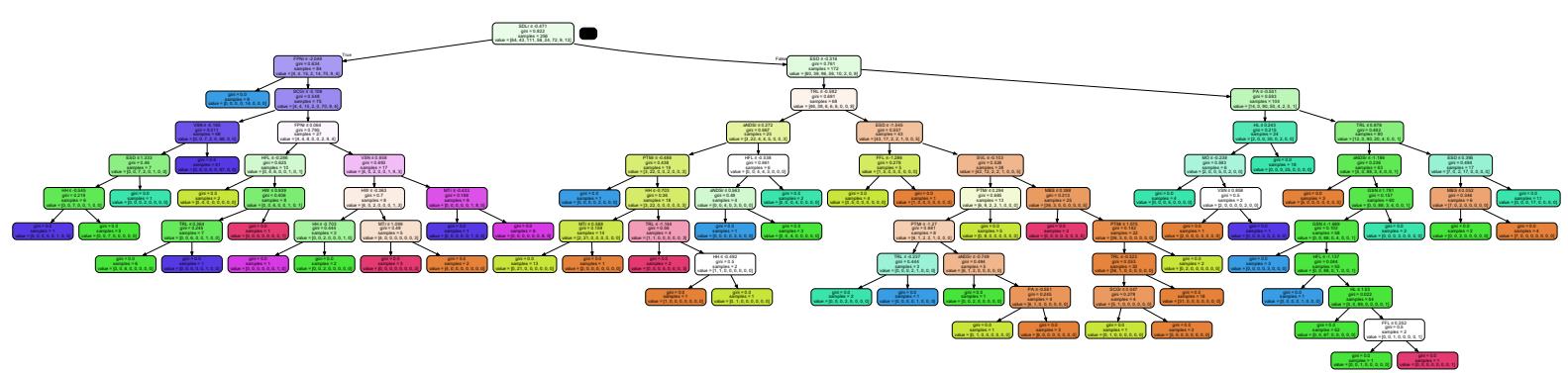


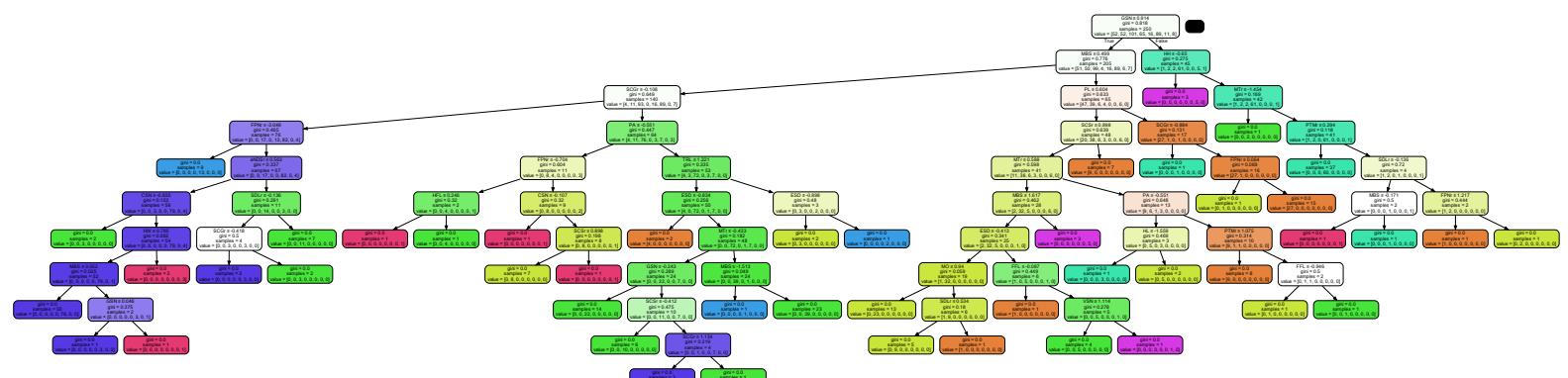
7.4 Complete Random Forest for Species











7.5 Complete Random Forest for Gender

