Cover sheet for submission of work for assessment

SWINBURNE UNIVERSITY OF TECHNOLOGY

UNIT DETAILS

Unit name	Introduction to Data Science			Class day/time	Friday, 12:30pm
Unit code	COS10022	Assignment no.	2	Due date	8/05/2021
Name of lecturer/teacher	Pei-Wei Tsai				
Tutor/marker's name	Pei-Wei Tsai				

Office use only

Faculty or school date stamp

STUDENT(S)

	Family Name(s)	Given Name(s)	Student ID Number(s)
(1)	Syed	Omair Maqdoom Mohiuddin	102863768
(2)			
(3)			
(4)			
(5)			
(6)			

DECLARATION AND STATEMENT OF AUTHORSHIP

1. I/we have not impersonated, or allowed myself/ourselves to be impersonated by any person for the purposes of this assessment.
2. This assessment is my/our original work and no part of it has been copied from any other source except where due acknowledgement is made.
3. No part of this assessment has been written for me/us by any other person except where such collaboration has been authorised by the lecturer/teacher concerned.
4. I/we have not previously submitted this work for this or any other course/unit.
5. I/we give permission for my/our assessment response to be reproduced, communicated, compared and archived for plagiarism detection, benchmarking or educational purposes.

I/we understand that:

6. Plagiarism is the presentation of the work, idea or creation of another person as though it is your own. It is a form of cheating and is a very serious academic offence that may lead to exclusion from the University. Plagiarised material can be drawn from, and presented in, written, graphic and visual form, including electronic data and oral presentations. Plagiarism occurs when the origin of the material used is not appropriately cited.

Student signature/s

I/we declare that I/we have read and understood the declaration and statement of authorship.

(1)	Syed Omair	(4)	
(2)		(5)	
(3)		(6)	

Further information relating to the penalties for plagiarism, which range from a formal caution to expulsion from the University is contained on the Current Students website at **www.swin.edu.au/student/** Copies of this form can be downloaded from the Student Forms web page at **www.swinburne.edu.au/studentforms/**

BUILDING A PREDICTION MODEL

INTRODUCTION TO DATA SCIENCE (COS10022)

SYED OMAIR MAQDOOM MOHIUDDIN (102863768)

Introduction

Due to lack of knowledge, people accidently get poisoned by eating mushrooms which are not meant for humans (poisonous). The goal of the training dataset that was given is to build a predictive model and help the users prevent accidently be poisoned caused by lack of knowledge in identifying the mushroom type. The dataset contains 8124 data records of different mushrooms with 23 different attributes where the first attribute identifies whether the mushroom is edible and remaining describe the characteristics of a mushroom. The purpose of this report is to develop a predictive model which will help the users to prevent accidently be poisoned caused by lack of knowledge in identifying the mushroom type.

Data Preparation

Data Cleaning

In the dataset, there is an attribute stalk-root which contains some missing values. Data should be clean (no missing data) to achieve the goal which is to build a predictive model which will help the users to prevent accidently be poisoned caused by the lack of knowledge in identifying the mushroom type. Therefore, removing the missing values from stalk-root. Now missing vales are excluded from the dataset. Using Knime Analytics Platform we have cleaned the data using Row Filter node.

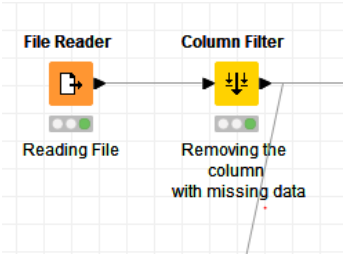


Figure 1: Data Cleaning Using Knime Platform (Column Filter)

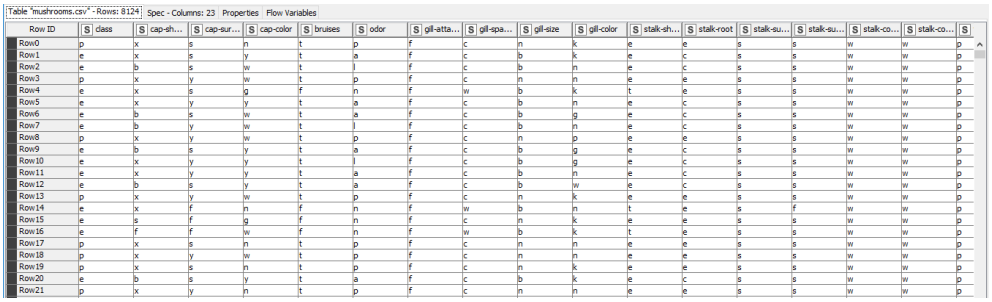


Figure 2: Dataset before cleaning

Before – Rows: 8124/ Columns: 23

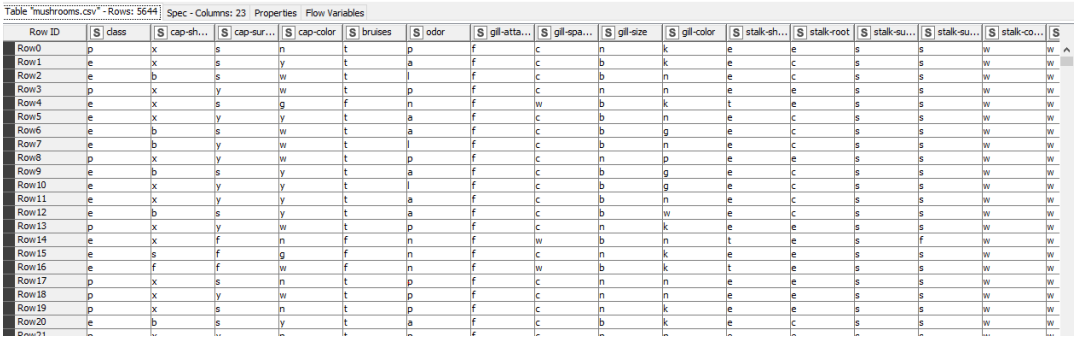


Figure 3: Dataset after cleaning

Removing missing data in stalk-root After- Rows: 5644/ Columns: 23.

To conclude, from the dataset the total number of rows decreased from 8124 to 5644. (2480 tuples were excluded from the dataset)

Data Observation

Attribute	Nominal data	Ordinal data	Continuous data	Discrete data
Class	X			
cap-shape	X			
cap-surface	X			
cap-color	X			
bruises	X			
odor	X			
gill-attachment	X			
gill-spacing		X		
gill-size		X		
gill-color	X			
stalk-shape	X			
stalk-root	X			
stalk-surface-above-ring	X			
stalk-surface-below-ring	X			
stalk-color-above-ring	X			
stalk-color-below-ring	X			
veil-type	X			
veil-color	X			
ring-number		X		
ring-type	X			
spore-print-color	X			
population		X		
habitat	X			

Attribute Selection and Conversion

I selected all the attributes for my model excluding the 'stalk-root' which has some missing data. Selecting all the attributes gives a diverse selection of attributes in the upcoming phases. We have enough attributes to deal with if any error takes place in the prediction process.
There is no requirement of conversion of datatypes in the model which I have chosen. All the attributes are string and using the string type we can build a prediction model. Changing the data type from string to other data type is difficult as not a good practice to be followed.

Data Shuffling

KNIME analytics platforms provides us with shuffling node which helps in easy shuffling of data.

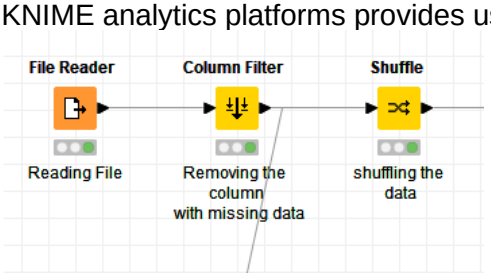


Figure 4: Shuffling the Data.

Data Partition

First partitioning the whole data into 2 sets, one with 1/3 (**Dataset1**) and other with 2/3. Using the 2/3 partitioned data and again partitioning that data into two equal parts (Dataset2 and Dataset3) (50:50).
Now using the first partitioned data (**Dataset1**) and partitioning it into 2 sets. A Training Set (Training 1) and a Test set (Test 1) as 9:1 ratio.
Using the first data from second partition (**Dataset2**) and partitioning it into 2 sets. A Training Set (Training 2) and a Test set (Test 2) as 9:1 ratio.
At last, use the second data set from second partition (**Dataset3**) and partition it into 2 sets. A Training Set (Training 3) and a Test set (Test 3) as 9:1 ratio.

The partition of the whole data is in below form, where training set is 90% and test set is 10%.

The Whole Shuffled Dataset					
Training 1	Test 1	Training 2	Test 2	Training 3	Test 3

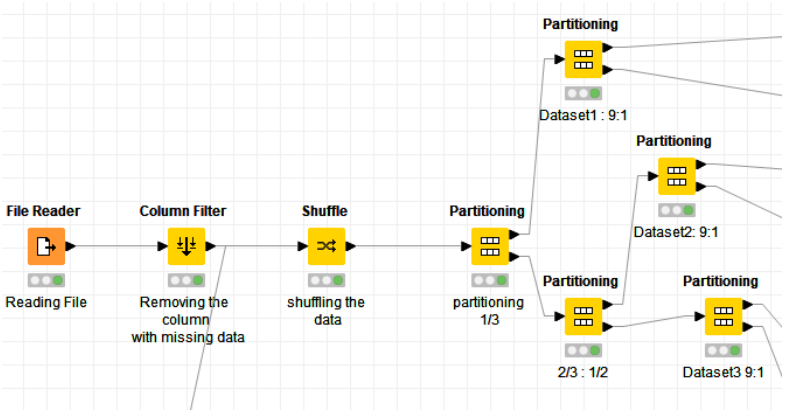


Figure 5: Data Partitioning

Predictive Model

Model Selection

I have selected Random Forest or Random Decision Forest method for my prediction model. It operates by constructing multiple Decision Tress during the training phase. It is an application of stacking.
There are many benefits of using this method for prediction purposes. In this method, it uses multiple tress to reduce the risk of overfitting. It has low training time. It has high accuracy; it runs efficiently on large dataset (bigdata). It produces highly accurate predictions in large dataset (dealing with big data and its analytics). Also, it can maintain accuracy when a large proportion of data is missing. In this dataset we don't have much missing data, but if we want to change the data which has huge amount of data missing, this method will maintain the accuracy in the model.

Model

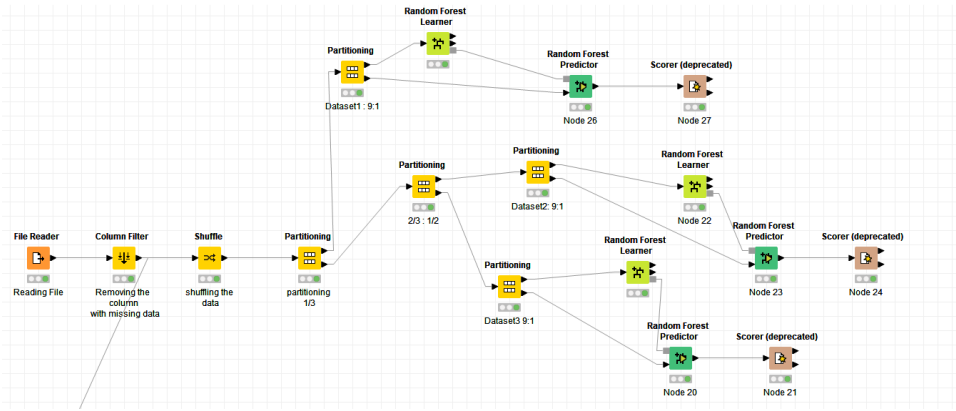


Figure 6: Prediction Model

Using Random Forest learner to train the training set. Then using that data from the learner and taking the data from test set and using Random Forest Predictor to predict the results. It predicts patterns according to an aggregation of the predictions of individual trees in a random forest model. Using the Scorer (deprecated) to compare the two column and generate confusion matrix.

Here we are using class (edible or poisonous) as the target column and other columns for the prediction model (except the stalk-root).

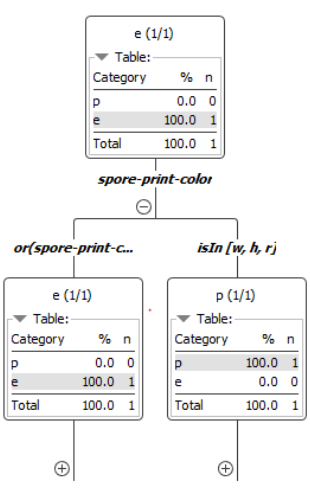


Figure 7: Tree View for Dataset3

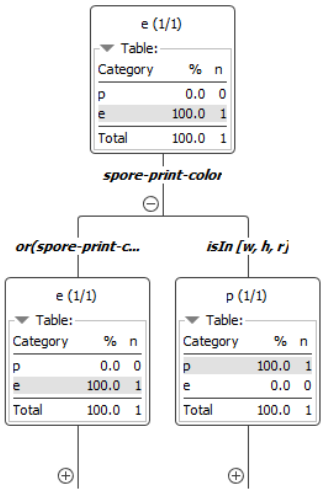


Figure 8: Tree View for Dataset2

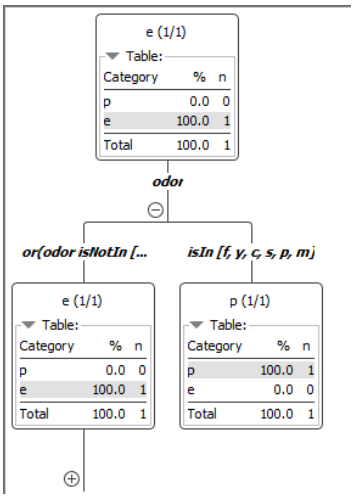


Figure 9: Tree View for Dataset1

When we click the (+) sign we can see the trees with random columns.

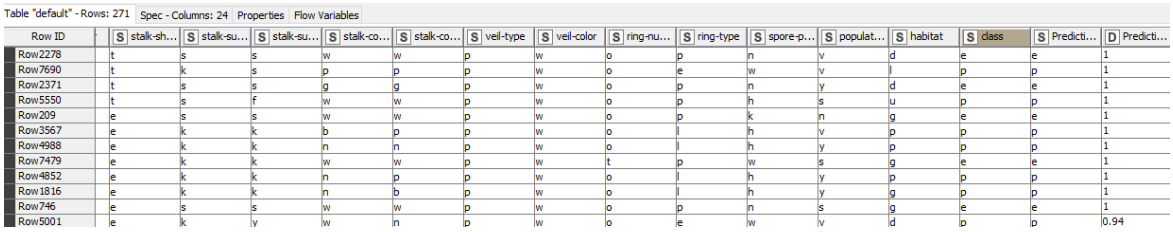


Table "default" - Rows: 271 | Spec - Columns: 24 | Properties | Flow Variables

Row ID	stalk-sh...	stalk-su...	stalk-su...	stalk-co...	stalk-co...	veil-type	veil-color	ring-nu...	ring-type	spore-p...	populat...	habitat	class	Predicti...	Predicti...
Row2278	t	s	s	w	w	p	w	o	p	n	v	d	e	e	1
Row7690	t	k	s	p	p	p	w	o	e	w	v	l	p	p	1
Row2371	t	s	s	g	g	p	w	o	p	n	y	d	e	e	1
Row5550	t	s	f	w	w	p	w	o	p	h	s	u	p	p	1
Row209	e	s	s	w	w	p	w	o	p	k	n	g	e	e	1
Row3567	e	k	k	b	p	p	w	o	l	h	v	p	p	p	1
Row4988	e	k	k	n	n	p	w	o	l	h	y	p	p	p	1
Row7479	e	k	k	w	w	p	w	t	p	w	s	g	e	e	1
Row4852	e	k	k	n	p	p	w	o	l	h	y	p	p	p	1
Row1816	e	k	k	n	b	p	w	o	l	h	y	g	p	p	1
Row746	e	s	s	w	w	p	w	o	p	n	s	g	e	e	1
Row5001	e	k	y	w	n	p	w	o	e	w	v	d	p	p	0.94

Figure 10: Prediction output of Dataset1

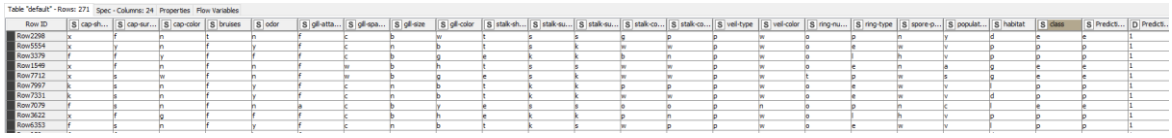


Figure 11: Prediction output of Dataset2

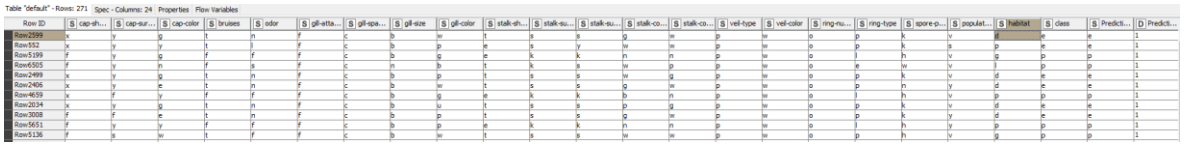
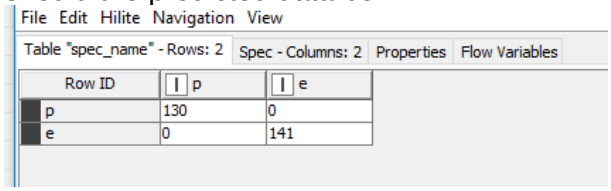


Figure 12: Prediction output of Dataset3

As we can see from the above figures, most of the prediction of data is correct from the prediction model. We can check the prediction using Confidence column, if it is 1 then it is right (absolutely).

Confusion Matrix

Confusion matrix helps us understand how much amount of predicted data is correct, if incorrect then was should the predicted data be.



File Edit Hilite Navigation View

Table "spec_name" - Rows: 2 | Spec - Columns: 2 | Properties | Flow Variables

Row ID	p	e
p	130	0
e	0	141

Figure 13: Confusion Matrix for Dataset1

As it can be seen from the above confusion matrix, the prediction for poisonous is correct. (130) and the prediction for edible is also correct (141). There is no data which is incorrectly predicted.

File Edit Hilite Navigation View

Table "spec_name" - Rows: 2 | Spec - Columns: 2 | Properties | Flow Variables

Row ID	p	e
p	134	0
e	0	137

Figure 14: Confusion Matrix for Dataset2

As it can be seen from the above confusion matrix, the prediction for poisonous is correct. (134) and the prediction for edible is also correct (137). There is no data which is incorrectly predicted.

Confusion matrix - 0:21 - Scorer (deprecated)

File Edit Hilite Navigation View

Table "spec_name" - Rows: 2 | Spec - Columns: 2 | Properties | Flow Variables

Row ID	p	e
p	131	0
e	0	140

Figure 15: Confusion Matrix for Dataset3

As it can be seen from the above confusion matrix, the prediction for poisonous is correct. (131) and the prediction for edible is also correct (140). There is no data which is incorrectly predicted.

These are no errors in the prediction of data, so the chosen method gives us best prediction of the data with accuracy and precision.

Accuracy Statistics

Table "default" - Rows: 3 | Spec - Columns: 11 | Properties | Flow Variables

Row ID	TruePo...	FalsePo...	TrueNe...	FalseN...	Recall	Precision	Sensitivity	Specifity	F-meas...	Accuracy	Cohen'...
p	130	0	141	0	1	1	1	1	1	?	?
e	141	0	130	0	1	1	1	1	1	?	?
Overall	?	?	?	?	?	?	?	?	?	1	1

Figure 16: Accuracy Statistics for Dataset1

Table "default" - Rows: 3 | Spec - Columns: 11 | Properties | Flow Variables

Row ID	TruePo...	FalsePo...	TrueNe...	FalseN...	Recall	Precision	Sensitivity	Specifity	F-meas...	Accuracy	Cohen'...
p	134	0	137	0	1	1	1	1	1	?	?
e	137	0	134	0	1	1	1	1	1	?	?
Overall	?	?	?	?	?	?	?	?	?	1	1

Figure 17: Accuracy Statics for Dataset2

Table "default" - Rows: 3 | Spec - Columns: 11 | Properties | Flow Variables

Row ID	TruePo...	FalsePo...	TrueNe...	FalseN...	Recall	Precision	Sensitivity	Specifity	F-meas...	Accuracy	Cohen'...
p	131	0	140	0	1	1	1	1	1	?	?
e	140	0	131	0	1	1	1	1	1	?	?
Overall	?	?	?	?	?	?	?	?	?	1	1

Figure 18: Accuracy Statistics for Dataset3

Accuracy

We can draw the Accuracy from the Accuracy statistics table provided by Scorer (deprecated). A good model should have a high accuracy % (80% and above).

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \times 100\%$$

Using the formula,
Accuracy for Dataset1 is **100%.**
Accuracy for Dataset2 is **100%.**
Accuracy for Dataset3 is **100%.**

By observing the above accuracies for the datasets, we can conclude that the model has 100% Accuracy (mean).

True Positive Rate

We can draw True Positive Rates (TPR) from the Accuracy statistics table provided by Scorer (deprecated). A well-performed model should have a high TPR (ideally 1). It is also called as “Recall”, it shows what percentage of positive instances a classifier correctly identified.

$$\text{TPR} = \frac{TP}{TP + FN}$$

Using the formula,
TPR for Dataset1 is **1.**
TPR for Dataset2 is **1.**
TPR for Dataset3 is **1.**

By observing the above TPRs for the datasets, we can conclude that the TPR of model is 1(mean).

False Positive Rate

We can draw False Positive Rates (FPR) from the Accuracy statistics table provided by Scorer (deprecated). A well-performed model should have a low FPR (ideally 0). It is also called as “False Alarm Rate”, or “Type I error” it shows what percentage of negatives that a classifier marks as positive.

$$\text{FPR} = \frac{FP}{FP + TN}$$

Using the formula,
FPR for Dataset1 is **0.**
FPR for Dataset2 is **0.**
FPR for Dataset3 is **0.**

By observing the above FPRs for the datasets, we can conclude that the FPR of model is 0(mean).

False Negative Rate

We can draw False Negative Rates (FNR) from the Accuracy statistics table provided by Scorer (deprecated). A well-performed model should have a low FNR (ideally 0). It is also called as “Miss Rate”, or “Type II error” it shows what percentage of positives that a classifier marks as negative.

$$\text{FNR} = \frac{FN}{FN + TP}$$

Using the formula,
FNR for Dataset1 is **0.**
FNR for Dataset2 is **0.**
FNR for Dataset3 is **0.**

By observing the above FNRs for the datasets, we can conclude that the FNR of model is 0(mean).

Precision

We can draw Precision from the Accuracy statistics table provided by Scorer (deprecated). A good model should have a high precision. It is the percentage of instances marked positive that are really positive.

$$\text{Precision} = \frac{TP}{TP + FP}$$

Using the formula,
Precision for Dataset1 is **1.**
Precision for Dataset2 is **1.**
Precision for Dataset3 is **1.**

By observing the above precision for the datasets, we can conclude that the precision of model is 1(mean).

Visualization

Parallel Coordinates Plot was used to visualize the data. Parallel Coordinates plot gives us diverse option to choose the attributes in the plot. We can remove attributes from the plot if required after executing the plot. Using this plot, we can show our findings and support the prediction model. Class, gill-attachment, gill-size, stalk-shape, and ring-number are the attributes used in all the below visualizations.

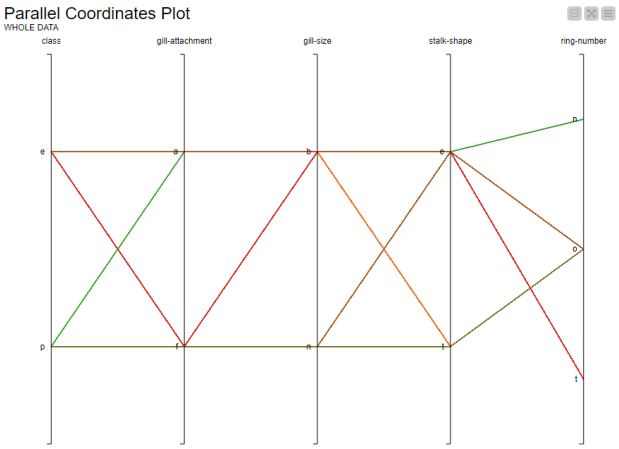


Figure 19: Visualization for whole shuffled data

Above plot shows the visualization for the whole mushroom data. As it can be seen from the graph, each trend was given different colour to different it from other. When we carefully observe the plot, we can find the below trends:

- **Class**: e > **gill-attachment**: a > **gill-size**: b > **stalk-shape**: e > **ring-number**: o.
- **Class**: e > **gill-attachment**: f > **gill-size**: b > **stalk-shape**: e > **ring-number**: t.
- **Class**: p > **gill-attachment**: a > **gill-size**: b > **stalk-shape**: e > **ring-number**: n.
- **Class**: p > **gill-attachment**: f > **gill-size**: n > **stalk-shape**: t > **ring-number**: o.

The above are some identified trends which help in predicting the data.

Using these trends, we can also find relationship between the attributes and nature of the mushroom (edible or poisonous).

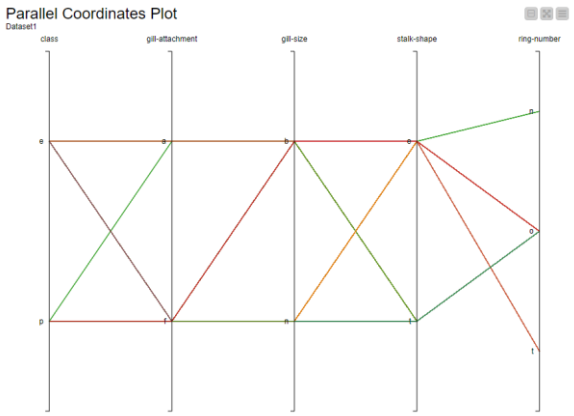


Figure 20: Visualization for Dataset1

Above plot shows the visualization for the first partitioned data (Dataset1). As it can be seen from the graph, each trend was given different colour to different it from other. When we carefully observe the plot, we can find the below trends:

- **Class**: e > **gill-attachment**: a > **gill-size**: b > **stalk-shape**: e > **ring-number**: o.
- **Class**: e > **gill-attachment**: f > **gill-size**: b > **stalk-shape**: e > **ring-number**: o.
- **Class**: p > **gill-attachment**: a > **gill-size**: b > **stalk-shape**: e > **ring-number**: n.
- **Class**: p > **gill-attachment**: f > **gill-size**: b > **stalk-shape**: e > **ring-number**: o.

The above are some identified trends which help in predicting the data.

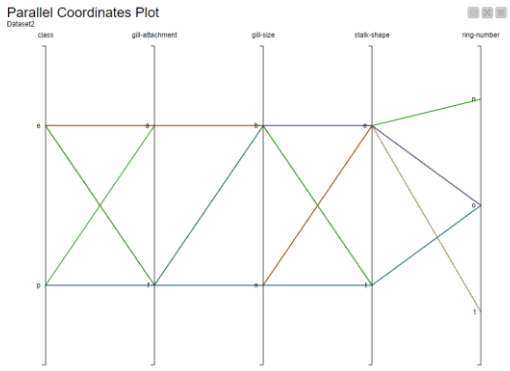


Figure 21: Visualization for Dataset2

Above plot shows the visualization for the second partitioned data (Dataset2). As it can be seen from the graph, each trend was given different colour to different it from other. When we carefully observe the plot, we can find the below trends:

- **Class**: e > **gill-attachment**: a > **gill-size**: b > **stalk-shape**: e > **ring-number**: o.
- **Class**: e > **gill-attachment**: f > **gill-size**: b > **stalk-shape**: t > **ring-number**: o.
- **Class**: p > **gill-attachment**: a > **gill-size**: b > **stalk-shape**: e > **ring-number**: n.
- **Class**: p > **gill-attachment**: f > **gill-size**: n > **stalk-shape**: t > **ring-number**: o.

The above are some identified trends which help in predicting the data.

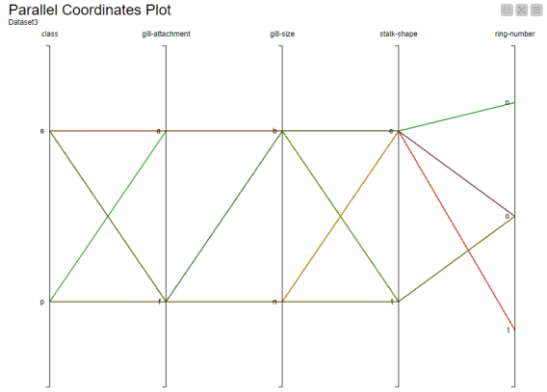


Figure 22: Visualization for Dataset3

Above plot shows the visualization for the third partitioned data (Dataset3). As it can be seen from the graph, each trend was given different colour to different it from other. When we carefully observe the plot, we can find the below trends:

- **Class**: e > **gill-attachment**: a > **gill-size**: b > **stalk-shape**: e > **ring-number**: o.
- **Class**: e > **gill-attachment**: f > **gill-size**: b > **stalk-shape**: t > **ring-number**: o.
- **Class**: p > **gill-attachment**: a > **gill-size**: b > **stalk-shape**: e > **ring-number**: n.
- **Class**: p > **gill-attachment**: f > **gill-size**: n > **stalk-shape**: t > **ring-number**: o.

The above are some identified trends which help in predicting the data. Finding more trends will help in predicting the data easy and accurately.

Conclusion

Model was built by following the best practices. All the phases were implemented without any errors. The proposed model has a high accuracy, high TPR, low FPR and FNR and high precision. This model will help the users understand the mushroom type which will prevent them from getting poisoned. Proposed model deals with huge amount of data, is highly efficient, highly accurate and able to estimate missing data if required (usage of Random Forest). Visualizations can be used to understand the data, its trends, and predict the upcoming data(edible/poisonous).