UNIVERSITI TEKNIKAL MALAYSIA MELAKA

BITI 2513: INTRODUCTION TO DATA SCIENCE

MULTIPLE DISEASE PREDICTION
Task 3

Python Team

NAME:	1) Mohd Hariz Bin Abdul Malek (B031810296) 2) Shivedhassen a/l Balasinggam (B031810360) 3) Muhamad Nur Bin Muhamad Zaidi (B031810314) 4) Agilan a/l Manivasagam (B031810281)
CLASS:	2 BITI S1G1
COURSE:	BITI 2513
LECTURER NAME:	AP. Dr. Sharifah Sakinah Syed Ahmad

Introduction

There are more than tens of thousands of diseases that affect humans that have been discovered in the world. But most of the diseases tend to be cured for two reasons which is because it shows symptoms and second is because there is availability in medicine or treatment. Some diseases show symptoms on the human body physically or mentally.

Early detection of preventive diseases can play a crucial role in timely intervention and management. It also assists in efficient distribution of resources in the healthcare sector. We endeavour to develop a system that facilitates early detection of multiple critical diseases based on their symptoms by using data analysis. The dataset that is being used states many types of disease with their symptoms.

Objective

- To detect any type of disease in an individual based on their symptom.
- To prevent the disease from becoming worse.
- To help the patient with early symptoms detecting it in an early phase.
- To build a model that accurately detects any kind of disease in an individual.

Goal

Our main goal in this project is being able to detect some of the disease in an individual by using our disease detector. This detector helps to find the disease in an individual and alerts them so that they can control or cure their disease from becoming worse. Moreover, we are focused on the younger generation's health. This detector also shows the symptoms of any disease in our body. This detector would help to identify if that person is affected by serious illness, it helps to find it in an early stage and we can start to treat it before it becomes worse.

Besides that , we can also lower the rate of disease in the country as quick actions can be taken in order to eliminate multiple diseases and we can increase the healthy population rate in the country and get the best standing among other countries for the best health quality . Data science and Artificial Intelligence plays an important role in the making of this detector. The technology and knowledge helps to create this useful and technological detector.

Success and Measurement

This project will build a model to analyze the data according to the related attributes or characteristics of a disease's symptoms to diagnose or detect whether the person has a disease or not. The model considered being successful when it is able to diagnose the disease and the success of this project will be measured by the performance of the model which is the accuracy of the model. This project is a success when the model performs good enough and has less errors.

Measurable result

- Model able to diagnose the disease
- Model built able to diagnose with great accuracy

Data Source

The dataset contains 4921 rows including the header which has 4920 instances and 18 attributes or columns which consists of the diseases' symptoms and the diagnosed diseases. The first column in the dataset is labeled 'Disease' while the others labeled 'Symptom_x' in which the x is in increasing order from 1 to 17. Each row is a diagnosed disease based on the symptoms. The purpose of the data is to record the symptoms of each disease diagnosed.

Column entries (attributes):
Disease - The diagnosed disease
Symptom_(1-17) - Diseases' symptom

DATA MANAGEMENT PLAN FOR MULTIPLE DISEASE PREDICTION

Multiple disease prediction project is a project that can predict the disease that has in the human body that gives a sign like a symptom of the disease. This project involves the data analysis on how to predict using data mining techniques. Data mining techniques are used for a variety of applications. In the healthcare industry, data mining plays an important role in predicting diseases. Data mining is the process of selecting, discovering and modeling huge amounts of data. This process has become an increasingly insidious activity in all areas of medical science research.

0. Project Name
Multiple Disease Prediction
1. Description of the data
1.1 Type of study The study of this data is about the multiple type of disease with their symptoms which are collected from the kaggle website dataset **1.2 Type of data** The type of the data that is being recorded is qualitative that deals with characteristics of the symptoms of the disease which only can be observed but not to be measured. The data type for the dataset is object. **1.3 Format and scale of the data** The data format for this dataset is more to string which is the symptom and the type of disease. Most of the data scale is nominal scales or could be simply called as labels for the symptoms for the disease.
2. Data collection/generation
Data collection is one of the important processes of gathering and measuring the information. The method that is being used for the collection of the data should be one of the major concerns because the data that is being collected must be precisely perfect for the future use.

2.1 Methodology for data collection/generation

This is the primary data collection which is qualitative data that is being recorded based on the observation and research.

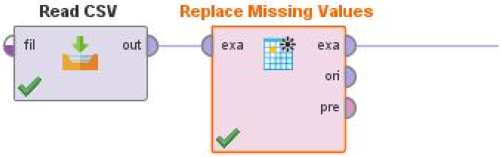
3. Data management, documentation and curation

The dataset provides the column containing symptoms of the disease and the type of the disease. This dataset can be obtained via kaggle website. This website allows you to make any documentation and get the details for the symptom, precaution, disease type and the weights. This data is placed in the kaggle for the long-term storage and preservation with no limit to the retention period.

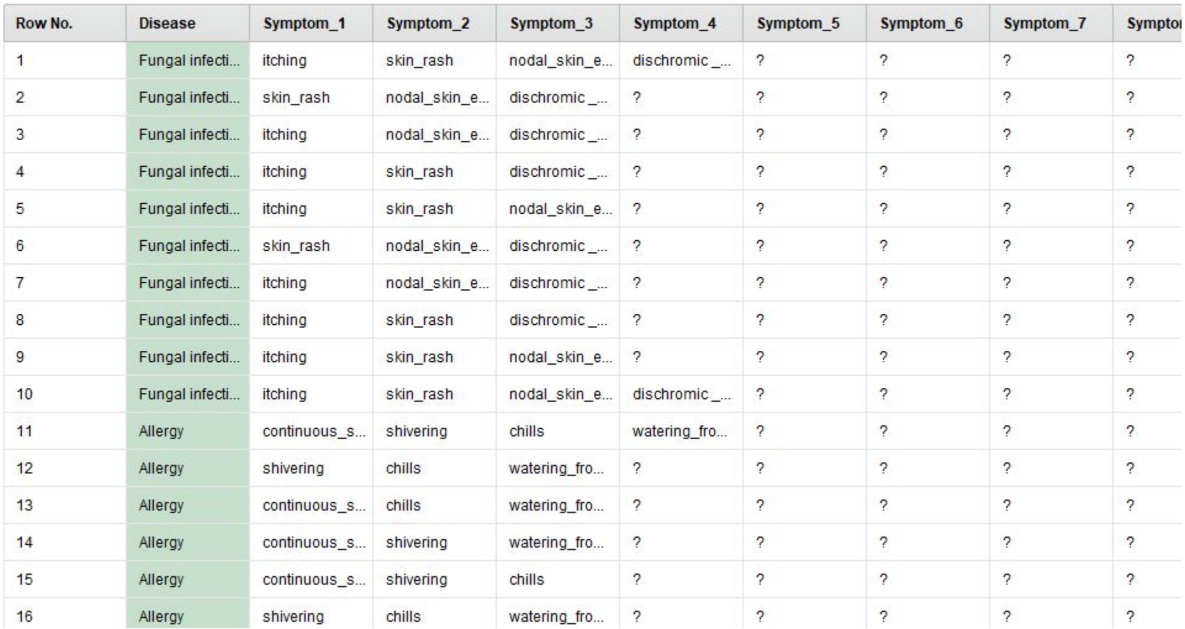
The plan after this process is about getting the best cleaned data by the process of cleaning, restructuring and enriching the raw data available into a more usable format. It is easier to call it a data wrangling process. This will help in the process of decision making, and thus get better insights in less time. Data wrangling, like most data analytics processes, is an iterative one – the practitioner will need to carry out these steps repeatedly in order to produce the results desired.

Data Wrangling

Dataset that we obtained from kaggle is basically already cleaned and there is no need to do data wrangling. The data is already in appropriate and tidy form that is easy to analyze and manipulate. However the dataset contains null values which there might be a little cleaning to do with the data like switching the null values with something that holds value like ‘none’ instead of ‘?’. We are using RapidMiner Studio to do the data wrangling. We use the operator ‘Read CSV’ to read the csv file of the disease dataset. Then, we use the operator ‘Replace Missing Values’ to replace all the null values or ‘?’ in the dataset to ‘none’. Below is the operator used.

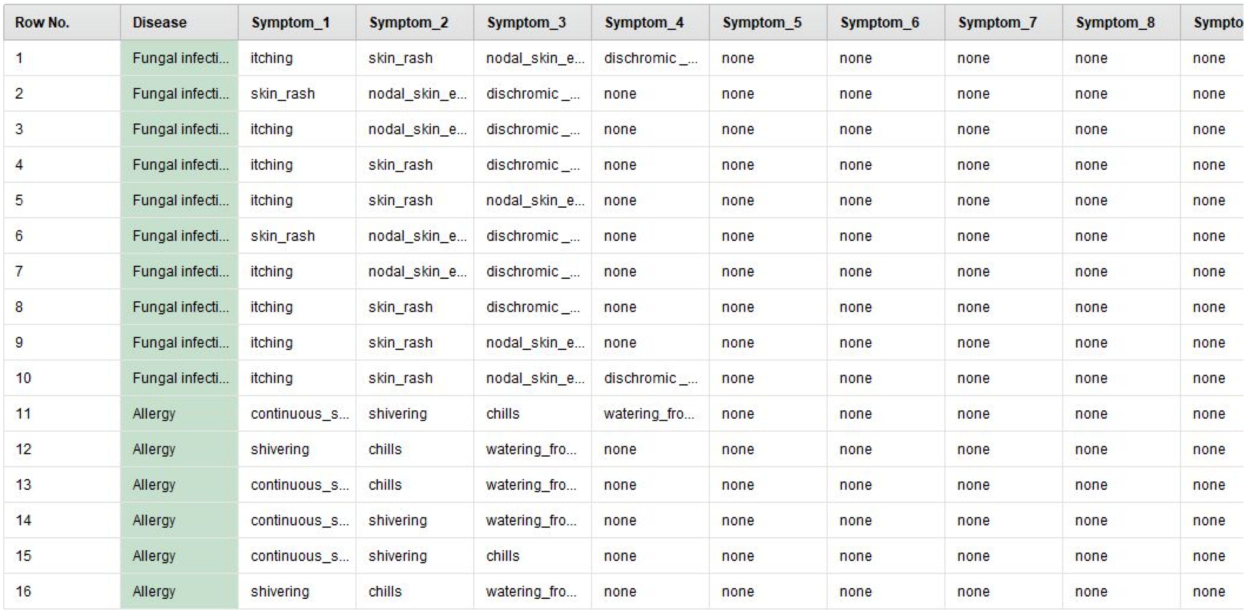


Below is the result of the data wrangling.



Row No.	Disease	Symptom_1	Symptom_2	Symptom_3	Symptom_4	Symptom_5	Symptom_6	Symptom_7	Symptor
1	Fungal infecti...	itching	skin_rash	nodal_skin_e...	dischromic _...	?	?	?	?
2	Fungal infecti...	skin_rash	nodal_skin_e...	dischromic _...	?	?	?	?	?
3	Fungal infecti...	itching	nodal_skin_e...	dischromic _...	?	?	?	?	?
4	Fungal infecti...	itching	skin_rash	dischromic _...	?	?	?	?	?
5	Fungal infecti...	itching	skin_rash	nodal_skin_e...	?	?	?	?	?
6	Fungal infecti...	skin_rash	nodal_skin_e...	dischromic _...	?	?	?	?	?
7	Fungal infecti...	itching	nodal_skin_e...	dischromic _...	?	?	?	?	?
8	Fungal infecti...	itching	skin_rash	dischromic _...	?	?	?	?	?
9	Fungal infecti...	itching	skin_rash	nodal_skin_e...	?	?	?	?	?
10	Fungal infecti...	itching	skin_rash	nodal_skin_e...	dischromic _...	?	?	?	?
11	Allergy	continuous_s...	shivering	chills	watering_fro...	?	?	?	?
12	Allergy	shivering	chills	watering_fro...	?	?	?	?	?
13	Allergy	continuous_s...	chills	watering_fro...	?	?	?	?	?
14	Allergy	continuous_s...	shivering	watering_fro...	?	?	?	?	?
15	Allergy	continuous_s...	shivering	chills	?	?	?	?	?
16	Allergy	shivering	chills	watering_fro...	?	?	?	?	?

Before replacing missing values.



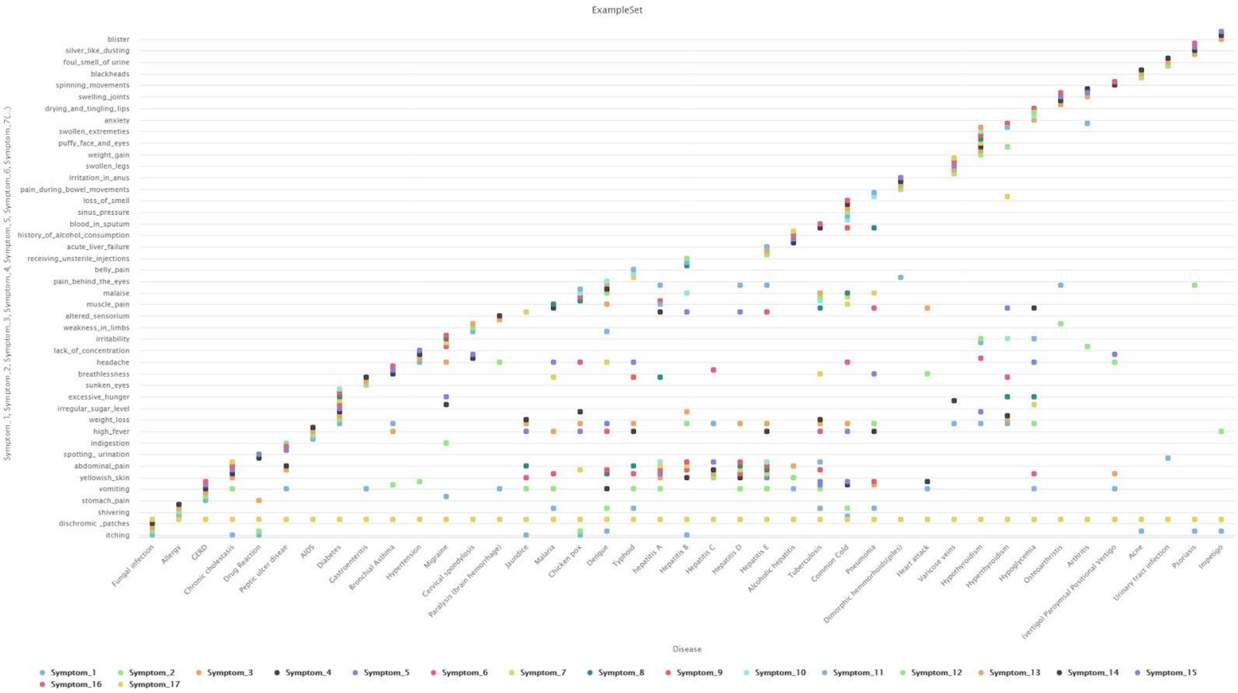
Row No.	Disease	Symptom_1	Symptom_2	Symptom_3	Symptom_4	Symptom_5	Symptom_6	Symptom_7	Symptom_8	Sympto
1	Fungal infecti...	itching	skin_rash	nodal_skin_e...	dischromic _...	none	none	none	none	none
2	Fungal infecti...	skin_rash	nodal_skin_e...	dischromic _...	none	none	none	none	none	none
3	Fungal infecti...	itching	nodal_skin_e...	dischromic _...	none	none	none	none	none	none
4	Fungal infecti...	itching	skin_rash	dischromic _...	none	none	none	none	none	none
5	Fungal infecti...	itching	skin_rash	nodal_skin_e...	none	none	none	none	none	none
6	Fungal infecti...	skin_rash	nodal_skin_e...	dischromic _...	none	none	none	none	none	none
7	Fungal infecti...	itching	nodal_skin_e...	dischromic _...	none	none	none	none	none	none
8	Fungal infecti...	itching	skin_rash	dischromic _...	none	none	none	none	none	none
9	Fungal infecti...	itching	skin_rash	nodal_skin_e...	none	none	none	none	none	none
10	Fungal infecti...	itching	skin_rash	nodal_skin_e...	dischromic _...	none	none	none	none	none
11	Allergy	continuous_s...	shivering	chills	watering_fro...	none	none	none	none	none
12	Allergy	shivering	chills	watering_fro...	none	none	none	none	none	none
13	Allergy	continuous_s...	chills	watering_fro...	none	none	none	none	none	none
14	Allergy	continuous_s...	shivering	watering_fro...	none	none	none	none	none	none
15	Allergy	continuous_s...	shivering	chills	none	none	none	none	none	none
16	Allergy	shivering	chills	watering_fro...	none	none	none	none	none	none

After replacing missing values.

Changing the null values into certain values will help us build a model for the prediction of the data in the future.

Exploratory Data Analysis (EDA)

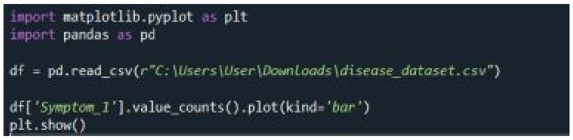
After cleaning and transforming the data into an appropriate form by replacing the missing values, we decided to view and observe the data for what we can or might find in the data if there is a pattern or important information in it. Again we are using RapidMiner Studio to visualize the data that has been cleaned. For the visualization, we are using scatter plots to visualize the data into a graph. Below is the result of the visualization.



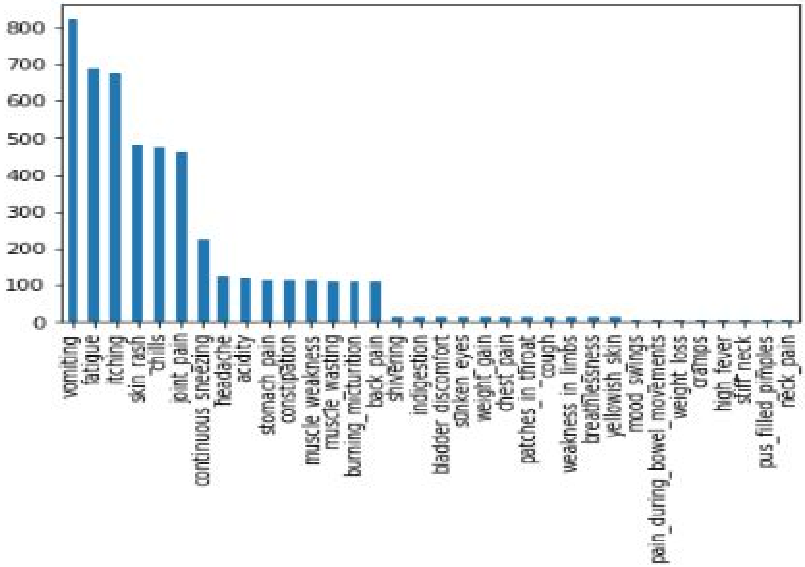
The graph above shows that the x-axis is the disease and the y-axis is the symptoms. From our observation and view, we can see that the symptoms belong to certain diseases. A disease can have a small and large amount of symptoms. From the graph, we can see that there are diseases that have small and large amounts of symptoms. The diseases that have large amounts of symptoms show that the disease is easy to diagnose since the patient or the victim of the disease experience those symptoms. However for the diseases that have less symptoms shown, it indicates that the disease might be difficult to identify or diagnose. From a medical perspective, it is reasonable that most diseases in early stages are difficult to identify and diagnose like cancer for example. Hence the graph helps us identify which disease has less symptoms that make it difficult to diagnose.

Data Visualization And Statistical Summary

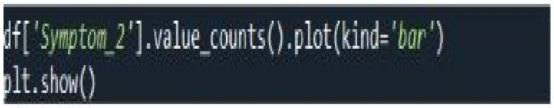
Data visualization is the graphical representation of information and data. By using visual elements like charts, graphs, and maps, data visualization tools provide an accessible way to see and understand trends and patterns in data.



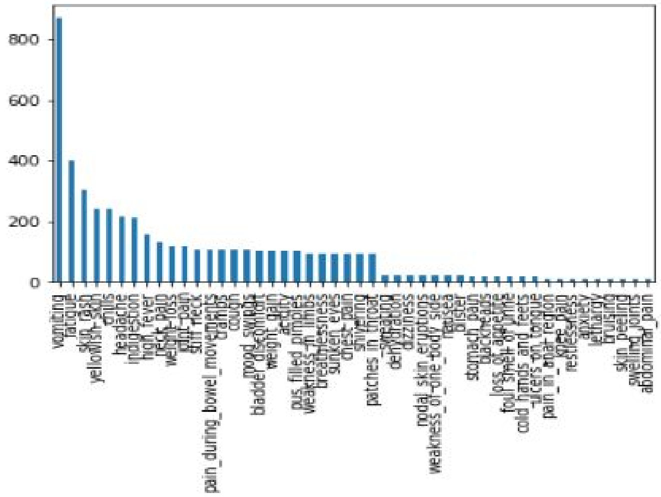
```
import matplotlib.pyplot as plt
import pandas as pd

df = pd.read_csv(r"C:\Users\User\Downloads\disease_dataset.csv")

df['Symptom_1'].value_counts().plot(kind='bar')
plt.show()
```



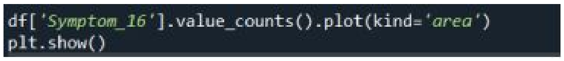
The figure shows that the majority of the people have been vomiting as their symptom for Symptom 1 followed by being fatigue, itching ,skin rash and so on. Symptoms such as shivering, indigestion and chest pain are less experienced by the people.



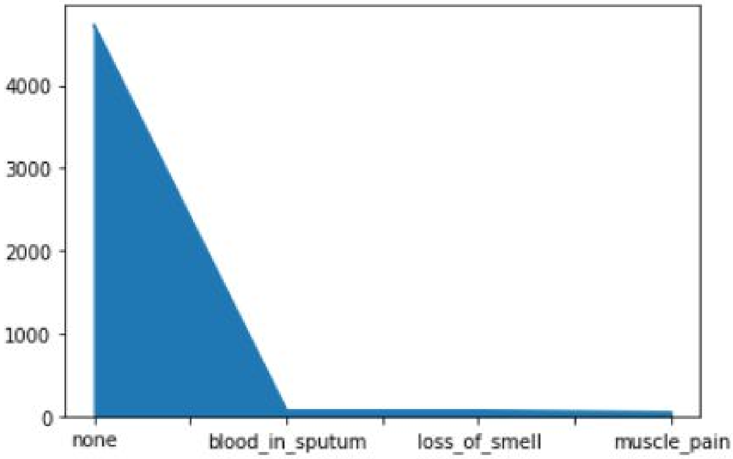
```
df['Symptom_2'].value_counts().plot(kind='bar')
plt.show()
```



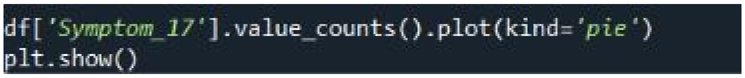
The figure shows that the majority of the people have been vomiting for Symptom 2 followed by fatigue,skin rash,yellowish skin and so on. They also experienced very less symptoms such as swelling joints and abdominal pain.



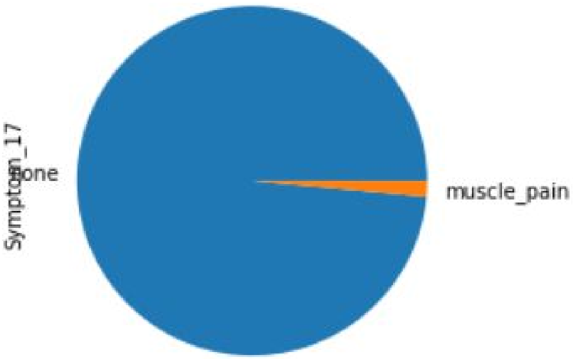
```
df['Symptom_16'].value_counts().plot(kind='area')
plt.show()
```



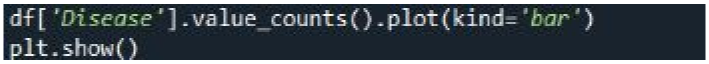
The figure shows that more than 4000 didn't have Symptom 16 and very less people had blood in sputum, loss of smell and muscle pain as the symptoms.



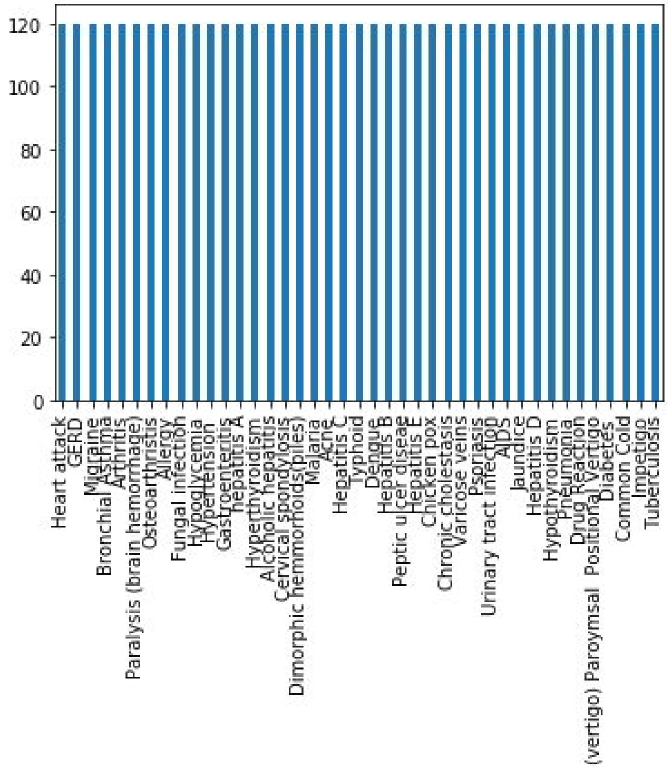
```
df['Symptom_17'].value_counts().plot(kind='pie')
plt.show()
```



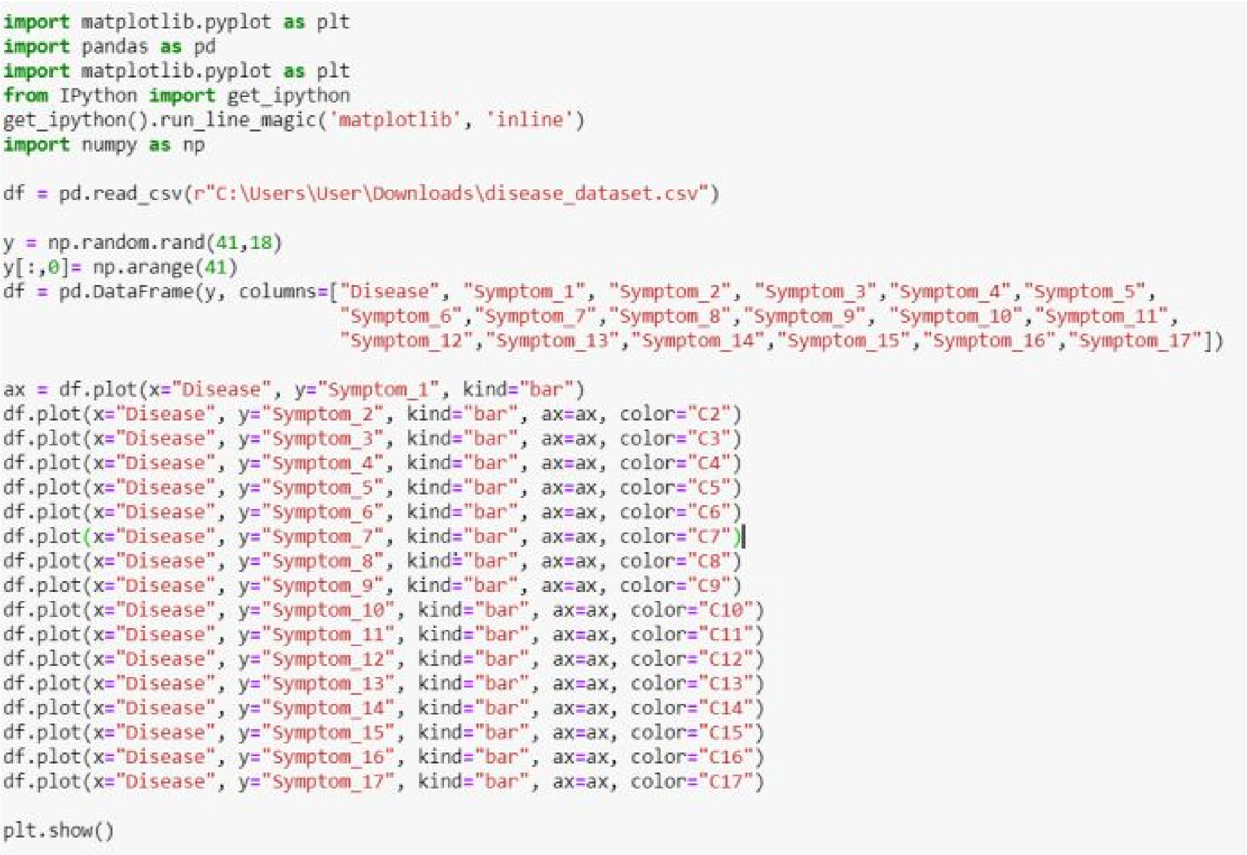
The figure shows that the majority had no symptom 17 and very few people had muscle pain as their 17th symptom.



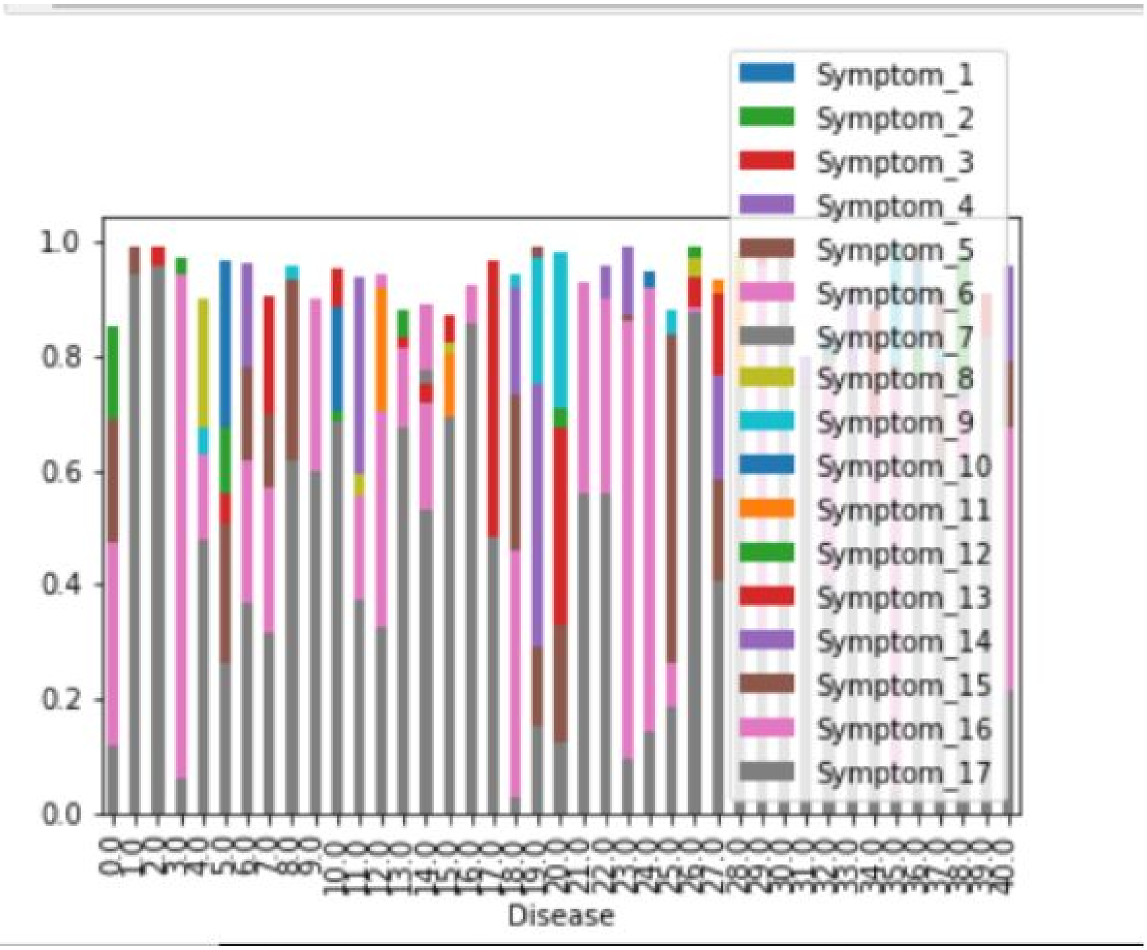
```
df['Disease'].value_counts().plot(kind='bar')
plt.show()
```



The figure shows the list of diseases the people may experience based on the symptoms mentioned previously.



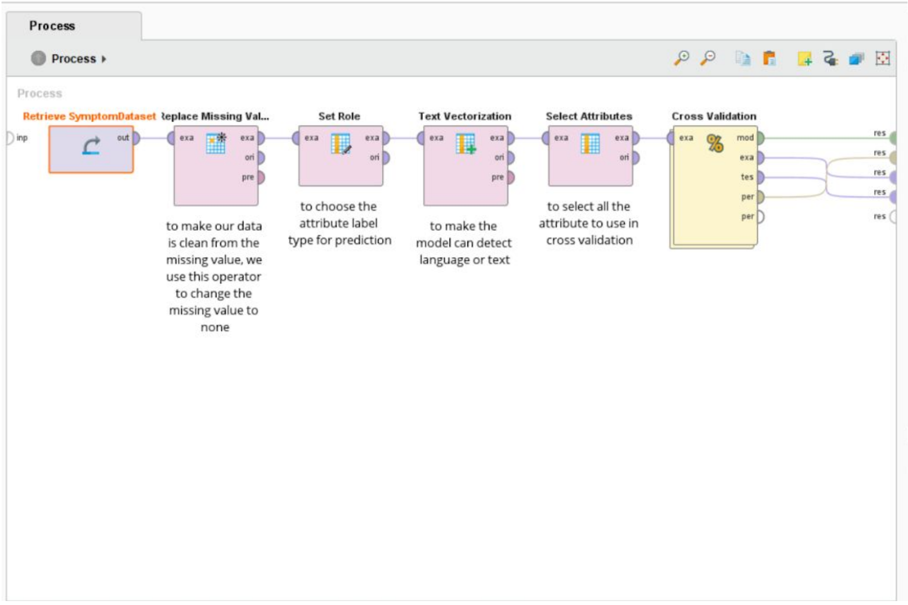
```
import matplotlib.pyplot as plt
import pandas as pd
import matplotlib.pyplot as plt
from IPython import get_ipython
get_ipython().run_line_magic('matplotlib', 'inline')
import numpy as np

df = pd.read_csv(r"C:\Users\User\Downloads\disease_dataset.csv")

y = np.random.rand(41,18)
y[:,0]= np.arange(41)
df = pd.DataFrame(y, columns=["Disease", "Symptom_1", "Symptom_2", "Symptom_3","Symptom_4","Symptom_5",
                              "Symptom_6","Symptom_7","Symptom_8","Symptom_9", "Symptom_10","Symptom_11",
                              "Symptom_12","Symptom_13","Symptom_14","Symptom_15","Symptom_16","Symptom_17"])

ax = df.plot(x="Disease", y="Symptom_1", kind="bar")
df.plot(x="Disease", y="Symptom_2", kind="bar", ax=ax, color="C2")
df.plot(x="Disease", y="Symptom_3", kind="bar", ax=ax, color="C3")
df.plot(x="Disease", y="Symptom_4", kind="bar", ax=ax, color="C4")
df.plot(x="Disease", y="Symptom_5", kind="bar", ax=ax, color="C5")
df.plot(x="Disease", y="Symptom_6", kind="bar", ax=ax, color="C6")
df.plot(x="Disease", y="Symptom_7", kind="bar", ax=ax, color="C7")
df.plot(x="Disease", y="Symptom_8", kind="bar", ax=ax, color="C8")
df.plot(x="Disease", y="Symptom_9", kind="bar", ax=ax, color="C9")
df.plot(x="Disease", y="Symptom_10", kind="bar", ax=ax, color="C10")
df.plot(x="Disease", y="Symptom_11", kind="bar", ax=ax, color="C11")
df.plot(x="Disease", y="Symptom_12", kind="bar", ax=ax, color="C12")
df.plot(x="Disease", y="Symptom_13", kind="bar", ax=ax, color="C13")
df.plot(x="Disease", y="Symptom_14", kind="bar", ax=ax, color="C14")
df.plot(x="Disease", y="Symptom_15", kind="bar", ax=ax, color="C15")
df.plot(x="Disease", y="Symptom_16", kind="bar", ax=ax, color="C16")
df.plot(x="Disease", y="Symptom_17", kind="bar", ax=ax, color="C17")

plt.show()
```



This above figure shows a Stacked Bar Chart. In the stacked bar plot, the bars at each index are literally “stacked” on top of one another. It shows the visual representation of the total symptoms each disease has, and the breakdown of each symptom. Many of the diseases share several symptoms. Based on this stacked bar chart ,majority of the people experience Symptom 16 and Symptom 17 which means they experience no symptoms according to the previously mentioned pie chart.

Data Modelling and Validation

In this project, Rapidminer Studio is used to model the algorithm. The model is shown below:



to make our data is clean from the missing value, we use this operator to change the missing value to none

to choose the attribute label type for prediction

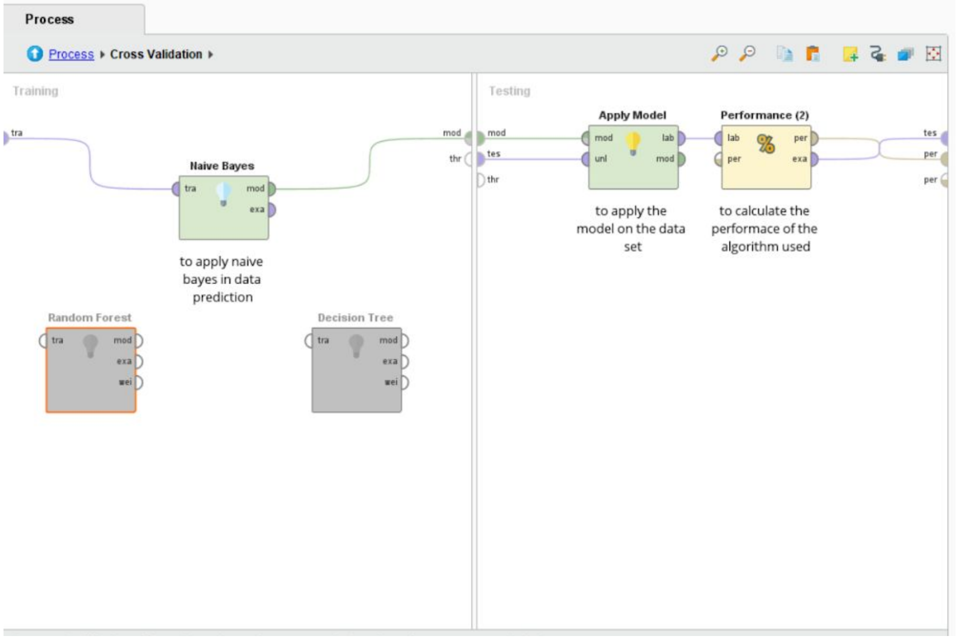
to make the model can detect language or text

to select all the attribute to use in cross validation

Figure 1

Figure 1 is the main model for the three algorithms that we used naive bayes, decision tree and random forest.

Naive Bayes model :

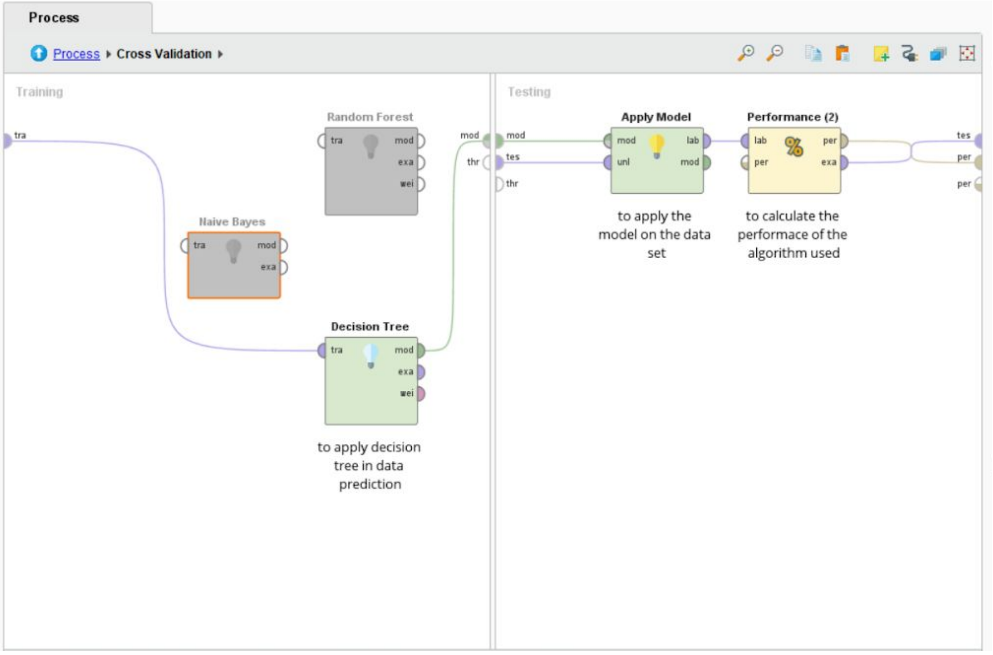


to apply naive bayes in data prediction

to apply the model on the data set

to calculate the performace of the algorithm used

Decision Tree model:

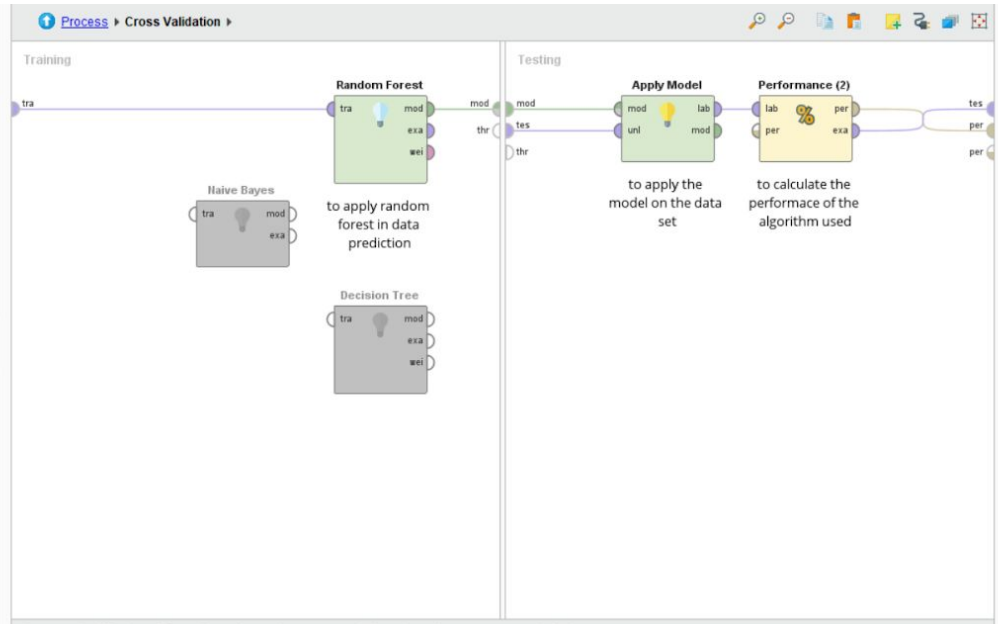


to apply decision tree in data prediction

to apply the model on the data set

to calculate the performace of the algorithm used

Random Forest model:



to apply random forest in data prediction

to apply the model on the data set

to calculate the performace of the algorithm used

Classification Techniques

Naive Bayes

Naive Bayes is a classification algorithm for binary (two-class) and multi-class classification problems. The technique is easiest to understand when described using binary or categorical input values.

Rather than attempting to calculate the values of each attribute value P(d1, d2, d3|h), they are assumed to be conditionally independent given the target value and calculated as P(d1|h) * P(d2|H) and so on.

Decision Tree

A Decision Tree is a simple representation for classifying examples. It is a Supervised Machine Learning where the data is continuously split according to a certain parameter.

Random Forest

Random Forest is an ensemble learning technique which implements a large number of decision trees based on different samples and different feature combinations. These trees are trained on bootstrapped subsets of the Example Set provided at the Input Port. Each node of a tree represents a splitting rule for one specific Attribute. Only a subset of Attributes, specified with the subset ratio criterion, is considered for the splitting rule selection.

Random forests can handle large numbers of variables in a dataset. Also, during the forest building process they generate an internal unbiased estimate of the generalization error. In addition, they can estimate missing data well. A major drawback of random forests is the lack of reproducibility, as the process of building the forest is random. Further, interpreting the final model and subsequent results is difficult since it contains many independent decision trees.

Validation Process

Cross Validation

Cross-validation is a statistical method used to estimate the skill of machine learning models. It is also a resampling procedure used to evaluate machine learning models on a limited data sample. It is commonly used in applied machine learning to compare and select a model for a given predictive modeling problem because it is easy to understand, easy to implement, and results in skill estimates that generally have a lower bias than other methods.

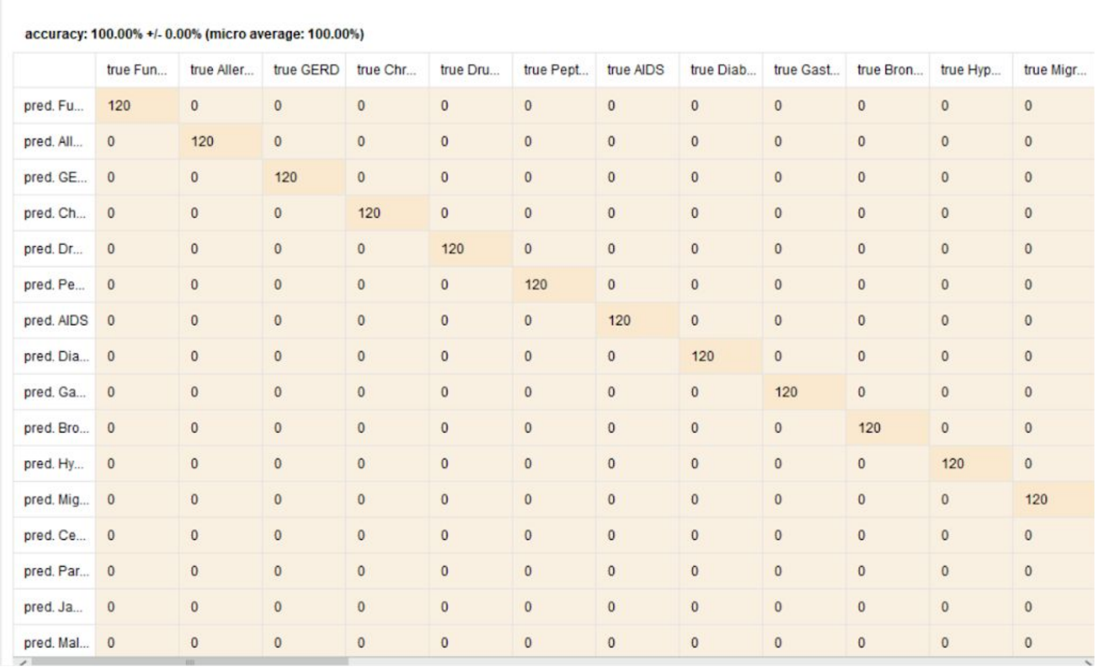
Performance Measure / Results

We used three performance measures to evaluate the three models that we built so that we can compare which one had the best performance in detecting diseases.

Accuracy

Relative number of correctly classified examples or in other words the percentage of correct predictions made. The higher the value the better the accuracy.

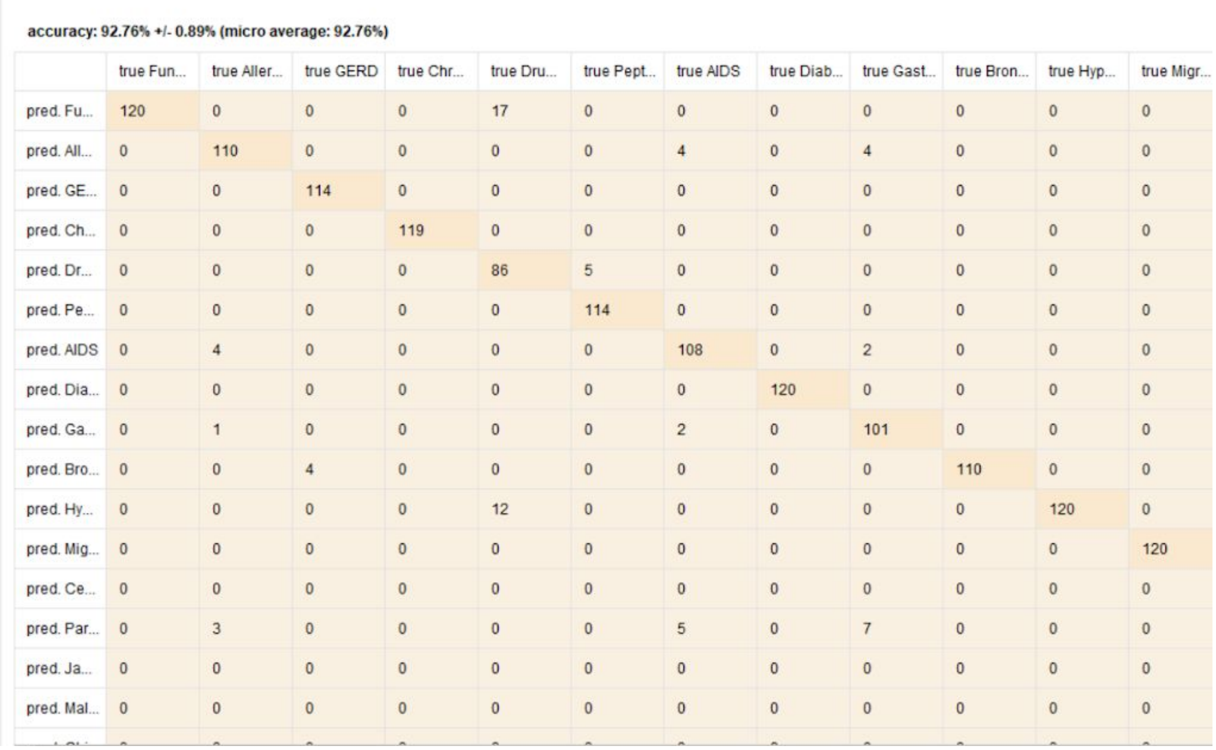
In the model we used the Performance (Classification) operator. Performance (Classification) operator is used with classification tasks only. On the other hand, the Performance operator automatically determines the learning task type and calculates the most common criteria for that type.



accuracy: 100.00% +/- 0.00% (micro average: 100.00%)

	true Fun...	true Aller...	true GERD	true Chr...	true Dru...	true Pept...	true AIDS	true Diab...	true Gast...	true Bron...	true Hyp...	true Migr...
pred. Fu...	120	0	0	0	0	0	0	0	0	0	0	0
pred. All...	0	120	0	0	0	0	0	0	0	0	0	0
pred. GE...	0	0	120	0	0	0	0	0	0	0	0	0
pred. Ch...	0	0	0	120	0	0	0	0	0	0	0	0
pred. Dr...	0	0	0	0	120	0	0	0	0	0	0	0
pred. Pe...	0	0	0	0	0	120	0	0	0	0	0	0
pred. AIDS	0	0	0	0	0	0	120	0	0	0	0	0
pred. Dia...	0	0	0	0	0	0	0	120	0	0	0	0
pred. Ga...	0	0	0	0	0	0	0	0	120	0	0	0
pred. Bro...	0	0	0	0	0	0	0	0	0	120	0	0
pred. Hy...	0	0	0	0	0	0	0	0	0	0	120	0
pred. Mig...	0	0	0	0	0	0	0	0	0	0	0	120
pred. Ce...	0	0	0	0	0	0	0	0	0	0	0	0
pred. Par...	0	0	0	0	0	0	0	0	0	0	0	0
pred. Ja...	0	0	0	0	0	0	0	0	0	0	0	0
pred. Mal...	0	0	0	0	0	0	0	0	0	0	0	0

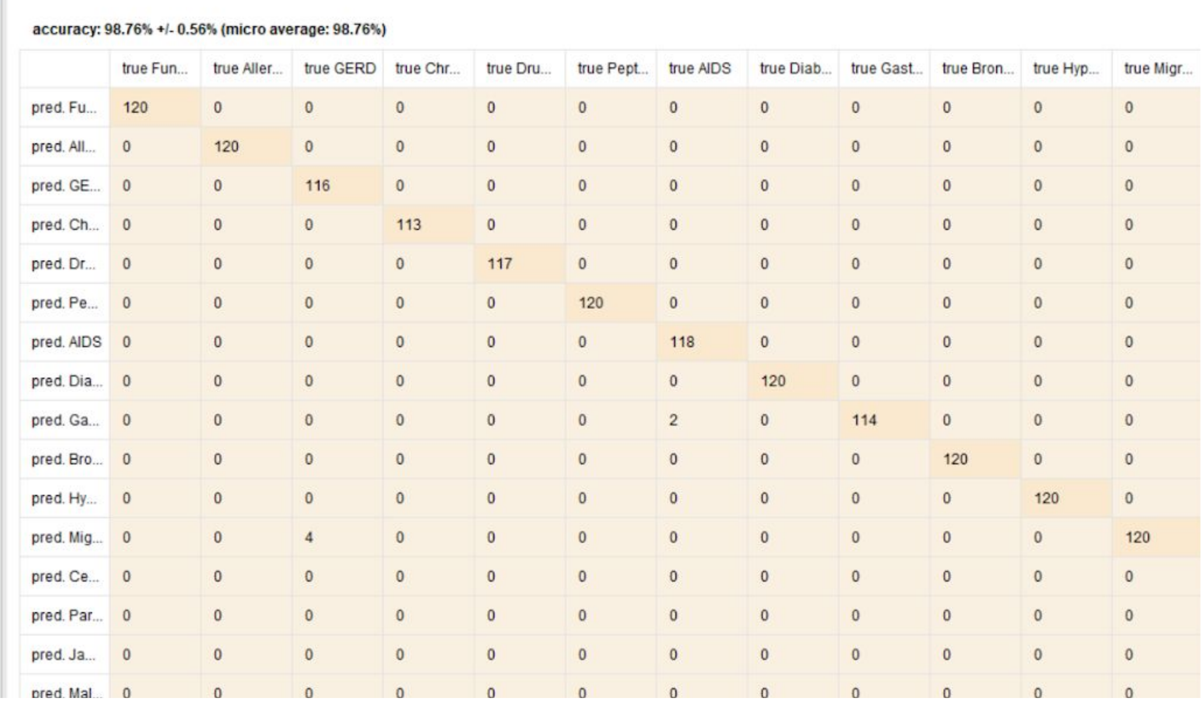
Accuracy for Naive Bayes model



accuracy: 92.76% +/- 0.89% (micro average: 92.76%)

	true Fun...	true Aller...	true GERD	true Chr...	true Dru...	true Pept...	true AIDS	true Diab...	true Gast...	true Bron...	true Hyp...	true Migr...
pred. Fu...	120	0	0	0	17	0	0	0	0	0	0	0
pred. All...	0	110	0	0	0	0	4	0	4	0	0	0
pred. GE...	0	0	114	0	0	0	0	0	0	0	0	0
pred. Ch...	0	0	0	119	0	0	0	0	0	0	0	0
pred. Dr...	0	0	0	0	86	5	0	0	0	0	0	0
pred. Pe...	0	0	0	0	0	114	0	0	0	0	0	0
pred. AIDS	0	4	0	0	0	0	108	0	2	0	0	0
pred. Dia...	0	0	0	0	0	0	0	120	0	0	0	0
pred. Ga...	0	1	0	0	0	0	2	0	101	0	0	0
pred. Bro...	0	0	4	0	0	0	0	0	0	110	0	0
pred. Hy...	0	0	0	0	12	0	0	0	0	0	120	0
pred. Mig...	0	0	0	0	0	0	0	0	0	0	0	120
pred. Ce...	0	0	0	0	0	0	0	0	0	0	0	0
pred. Par...	0	3	0	0	0	0	5	0	7	0	0	0
pred. Ja...	0	0	0	0	0	0	0	0	0	0	0	0
pred. Mal...	0	0	0	0	0	0	0	0	0	0	0	0

Accuracy for Decision Tree model



accuracy: 98.76% +/- 0.56% (micro average: 98.76%)

	true Fun...	true Aller...	true GERD	true Chr...	true Dru...	true Pept...	true AIDS	true Diab...	true Gast...	true Bron...	true Hyp...	true Migr...
pred. Fu...	120	0	0	0	0	0	0	0	0	0	0	0
pred. All...	0	120	0	0	0	0	0	0	0	0	0	0
pred. GE...	0	0	116	0	0	0	0	0	0	0	0	0
pred. Ch...	0	0	0	113	0	0	0	0	0	0	0	0
pred. Dr...	0	0	0	0	117	0	0	0	0	0	0	0
pred. Pe...	0	0	0	0	0	120	0	0	0	0	0	0
pred. AIDS	0	0	0	0	0	0	118	0	0	0	0	0
pred. Dia...	0	0	0	0	0	0	0	120	0	0	0	0
pred. Ga...	0	0	0	0	0	0	2	0	114	0	0	0
pred. Bro...	0	0	0	0	0	0	0	0	0	120	0	0
pred. Hy...	0	0	0	0	0	0	0	0	0	0	120	0
pred. Mig...	0	0	4	0	0	0	0	0	0	0	0	120
pred. Ce...	0	0	0	0	0	0	0	0	0	0	0	0
pred. Par...	0	0	0	0	0	0	0	0	0	0	0	0
pred. Ja...	0	0	0	0	0	0	0	0	0	0	0	0
pred. Mal...	0	0	0	0	0	0	0	0	0	0	0	0

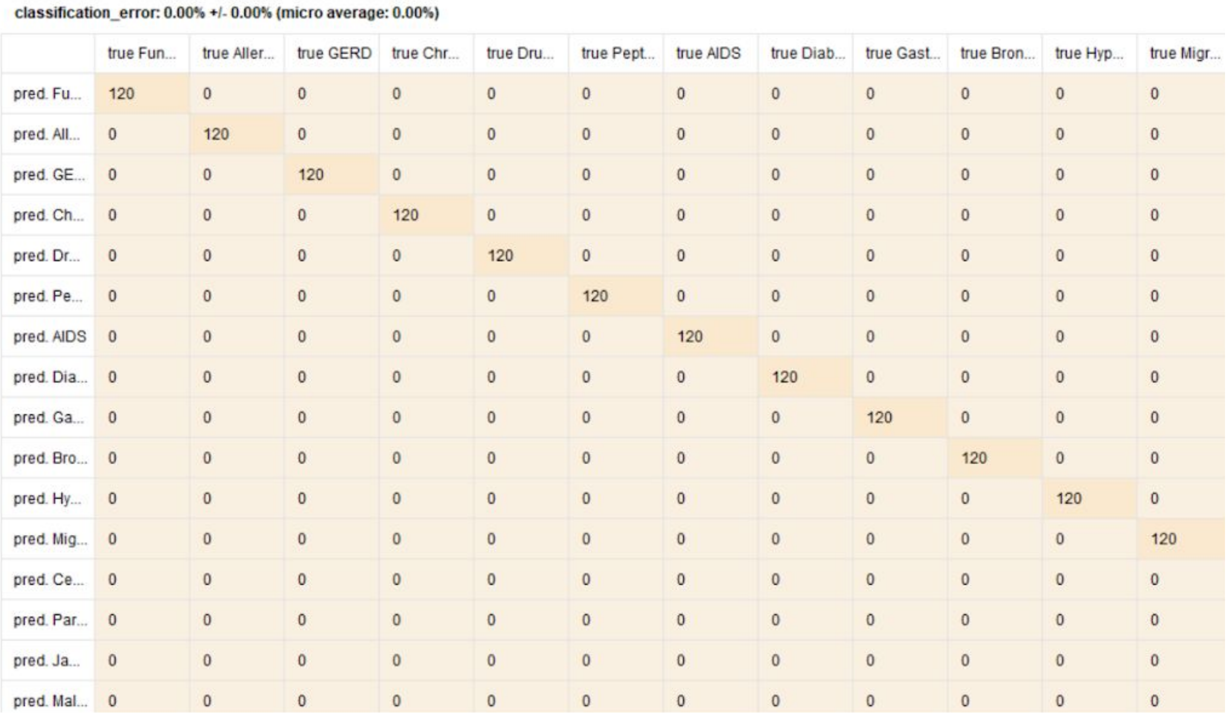
Accuracy for Random Forest model

Classification Error

Relative number of misclassified examples or in other words percentage of incorrect predictions. The below formula is used to calculate classification error rate

Error rate (ERR) is calculated as the number of all incorrect predictions divided by the total number of the dataset. The best error rate is 0.0, whereas the worst is 1.0. The below formula is used to calculate classification error rate:

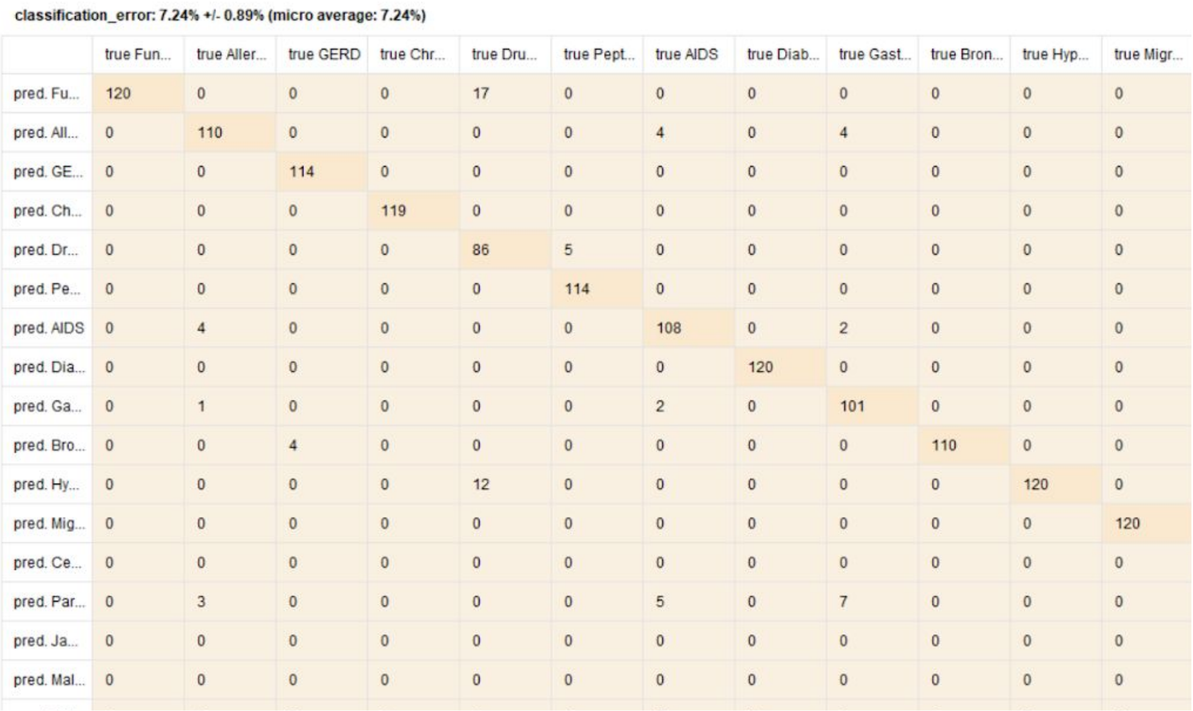
$$\mathrm{ERR} = \frac{\mathrm{FP} + \mathrm{FN}}{\mathrm{TP} + \mathrm{TN} + \mathrm{FN} + \mathrm{FP}} = \frac{\mathrm{FP} + \mathrm{FN}}{\mathrm{P} + \mathrm{N}}$$



classification_error: 0.00% +/- 0.00% (micro average: 0.00%)

	true Fun...	true Aller...	true GERD	true Chr...	true Dru...	true Pept...	true AIDS	true Diab...	true Gast...	true Bron...	true Hyp...	true Migr...
pred. Fu...	120	0	0	0	0	0	0	0	0	0	0	0
pred. All...	0	120	0	0	0	0	0	0	0	0	0	0
pred. GE...	0	0	120	0	0	0	0	0	0	0	0	0
pred. Ch...	0	0	0	120	0	0	0	0	0	0	0	0
pred. Dr...	0	0	0	0	120	0	0	0	0	0	0	0
pred. Pe...	0	0	0	0	0	120	0	0	0	0	0	0
pred. AIDS	0	0	0	0	0	0	120	0	0	0	0	0
pred. Dia...	0	0	0	0	0	0	0	120	0	0	0	0
pred. Ga...	0	0	0	0	0	0	0	0	120	0	0	0
pred. Bro...	0	0	0	0	0	0	0	0	0	120	0	0
pred. Hy...	0	0	0	0	0	0	0	0	0	0	120	0
pred. Mig...	0	0	0	0	0	0	0	0	0	0	0	120
pred. Ce...	0	0	0	0	0	0	0	0	0	0	0	0
pred. Par...	0	0	0	0	0	0	0	0	0	0	0	0
pred. Ja...	0	0	0	0	0	0	0	0	0	0	0	0
pred. Mal...	0	0	0	0	0	0	0	0	0	0	0	0

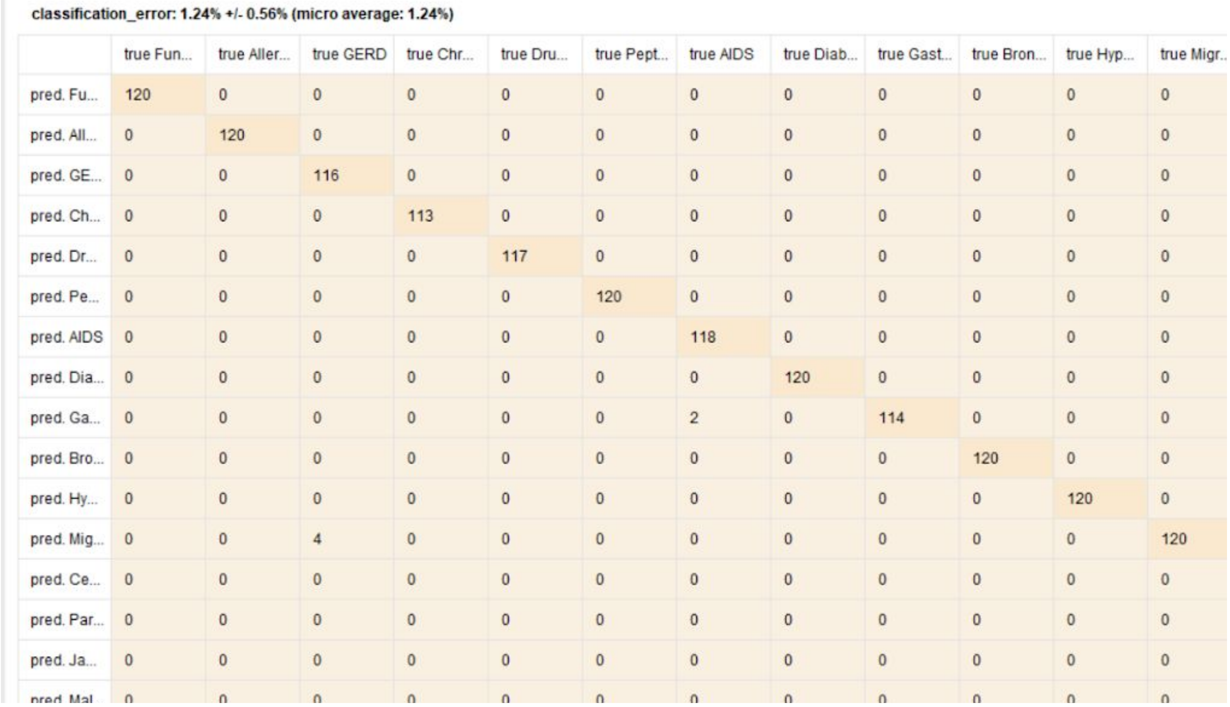
Classification Error for Naive Bayes model



classification_error: 7.24% +/- 0.89% (micro average: 7.24%)

	true Fun...	true Aller...	true GERD	true Chr...	true Dru...	true Pept...	true AIDS	true Diab...	true Gast...	true Bron...	true Hyp...	true Migr...
pred. Fu...	120	0	0	0	17	0	0	0	0	0	0	0
pred. All...	0	110	0	0	0	0	4	0	4	0	0	0
pred. GE...	0	0	114	0	0	0	0	0	0	0	0	0
pred. Ch...	0	0	0	119	0	0	0	0	0	0	0	0
pred. Dr...	0	0	0	0	86	5	0	0	0	0	0	0
pred. Pe...	0	0	0	0	0	114	0	0	0	0	0	0
pred. AIDS	0	4	0	0	0	0	108	0	2	0	0	0
pred. Dia...	0	0	0	0	0	0	0	120	0	0	0	0
pred. Ga...	0	1	0	0	0	0	2	0	101	0	0	0
pred. Bro...	0	0	4	0	0	0	0	0	0	110	0	0
pred. Hy...	0	0	0	0	12	0	0	0	0	0	120	0
pred. Mig...	0	0	0	0	0	0	0	0	0	0	0	120
pred. Ce...	0	0	0	0	0	0	0	0	0	0	0	0
pred. Par...	0	3	0	0	0	0	5	0	7	0	0	0
pred. Ja...	0	0	0	0	0	0	0	0	0	0	0	0
pred. Mal...	0	0	0	0	0	0	0	0	0	0	0	0

Classification Error for Decision Tree model



classification_error: 1.24% +/- 0.56% (micro average: 1.24%)

	true Fun...	true Aller...	true GERD	true Chr...	true Dru...	true Pept...	true AIDS	true Diab...	true Gast...	true Bron...	true Hyp...	true Migr...
pred. Fu...	120	0	0	0	0	0	0	0	0	0	0	0
pred. All...	0	120	0	0	0	0	0	0	0	0	0	0
pred. GE...	0	0	116	0	0	0	0	0	0	0	0	0
pred. Ch...	0	0	0	113	0	0	0	0	0	0	0	0
pred. Dr...	0	0	0	0	117	0	0	0	0	0	0	0
pred. Pe...	0	0	0	0	0	120	0	0	0	0	0	0
pred. AIDS	0	0	0	0	0	0	118	0	0	0	0	0
pred. Dia...	0	0	0	0	0	0	0	120	0	0	0	0
pred. Ga...	0	0	0	0	0	0	2	0	114	0	0	0
pred. Bro...	0	0	0	0	0	0	0	0	0	120	0	0
pred. Hy...	0	0	0	0	0	0	0	0	0	0	120	0
pred. Mig...	0	0	4	0	0	0	0	0	0	0	0	120
pred. Ce...	0	0	0	0	0	0	0	0	0	0	0	0
pred. Par...	0	0	0	0	0	0	0	0	0	0	0	0
pred. Ja...	0	0	0	0	0	0	0	0	0	0	0	0
pred. Mal...	0	0	0	0	0	0	0	0	0	0	0	0

Classification Error for Random Forest model

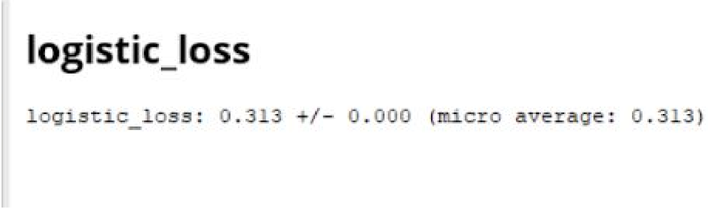
Logistic Loss

The loss function for linear regression is squared loss. The loss function for logistic regression is **Log Loss**, which is defined as follows:

$$\text{Log Loss}=\sum(x,y)\in D-y\log(y')-(1-y)\log(1-y')$$

where:

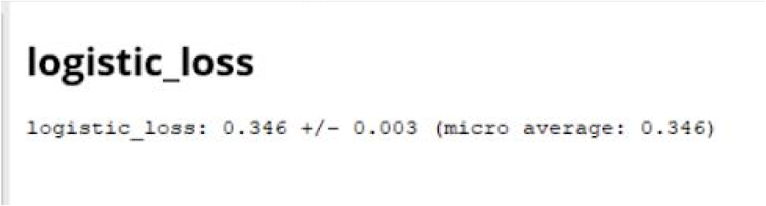
- $(x,y)\in D$ is the data set containing many labeled examples, which are (x,y) pairs.
- y is the label in a labeled example. Since this is logistic regression, every value of y must either be 0 or 1.
- y' is the predicted value (somewhere between 0 and 1), given the set of features in x.



logistic_loss

```
logistic_loss: 0.313 +/- 0.000 (micro average: 0.313)
```

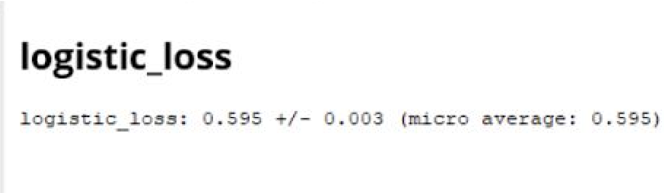
Logistic Loss for Naive Bayes model



logistic_loss

```
logistic_loss: 0.346 +/- 0.003 (micro average: 0.346)
```

Logistic Loss for Decision Tree model



logistic_loss

```
logistic_loss: 0.595 +/- 0.003 (micro average: 0.595)
```

Logistic Loss for Random Forest model

References

1. https://classeval.wordpress.com/introduction/basic-evaluation-measures/#:~:text=Error%20rate%20(ERR)%20is%20calculated,dataset%20(P%20%2B%20N).
2. https://towardsdatascience.com/understanding-random-forest-58381e0602d2
3. https://machinelearningmastery.com/naive-bayes-for-machine-learning/
4. https://towardsdatascience.com/decision-tree-classification-de64fc4d5aac
5. https://docs.rapidminer.com/latest/studio/operators/validation/performance/predictive/performance_classification.html#:~:text=Many%20other%20performance%20evaluation%20operators,Performance%20(Regression)%20operator%20etc.&text=Classification%20is%20a%20technique%20used%20to%20predict%20group%20membership%20for%20data%20instances.
6. https://medium.com/@chiragsehra42/decision-trees-explained-easily-28f23241248
7. https://towardsdatascience.com/decision-trees-explained-3ec41632ceb6
8. https://www.dataquest.io/blog/naive-bayes-tutorial/
9. https://developers.google.com/machine-learning/crash-course/logistic-regression/model-training