

CBIO753: Data Mining and Data Warehousing

CBIO 753 - DATA MINING AND DATA WAREHOUSING –LAB

COURSE OBJECTIVES: Implementation of various data mining tools and tests the real data sets using popular data mining tools such as **WEKA**

Total Credits: 1

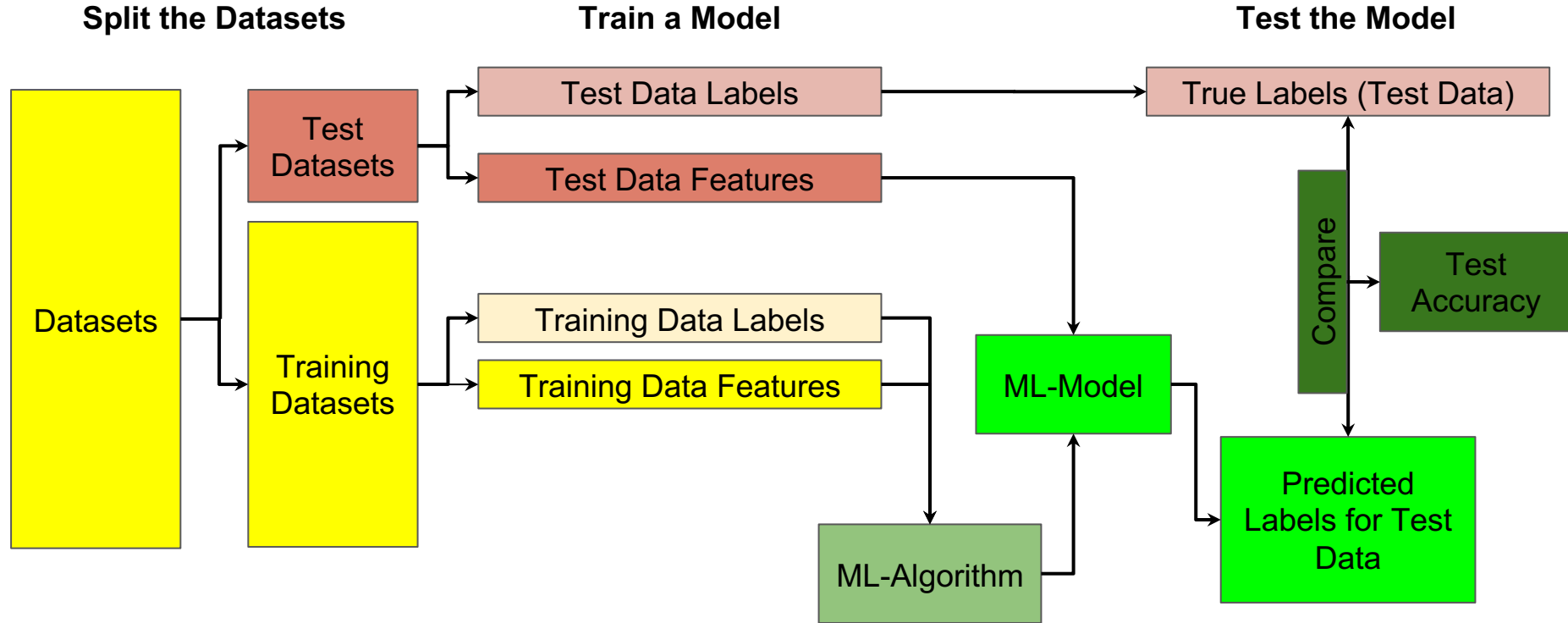
Exercises :

1. Demonstration of Data mining tools: Weka, Tanagra, Rapid miner, Keel, Orange
2. Introduction, Data pre-processing on dataset
3. Association rule process on dataset using apriori algorithm
4. Classification rule process on dataset using j48 algorithm
5. Classification rule process on dataset using id3 algorithm
6. Classification rule process on dataset using naïve bayes algorithm
7. Clustering rule process on dataset using simple k-means

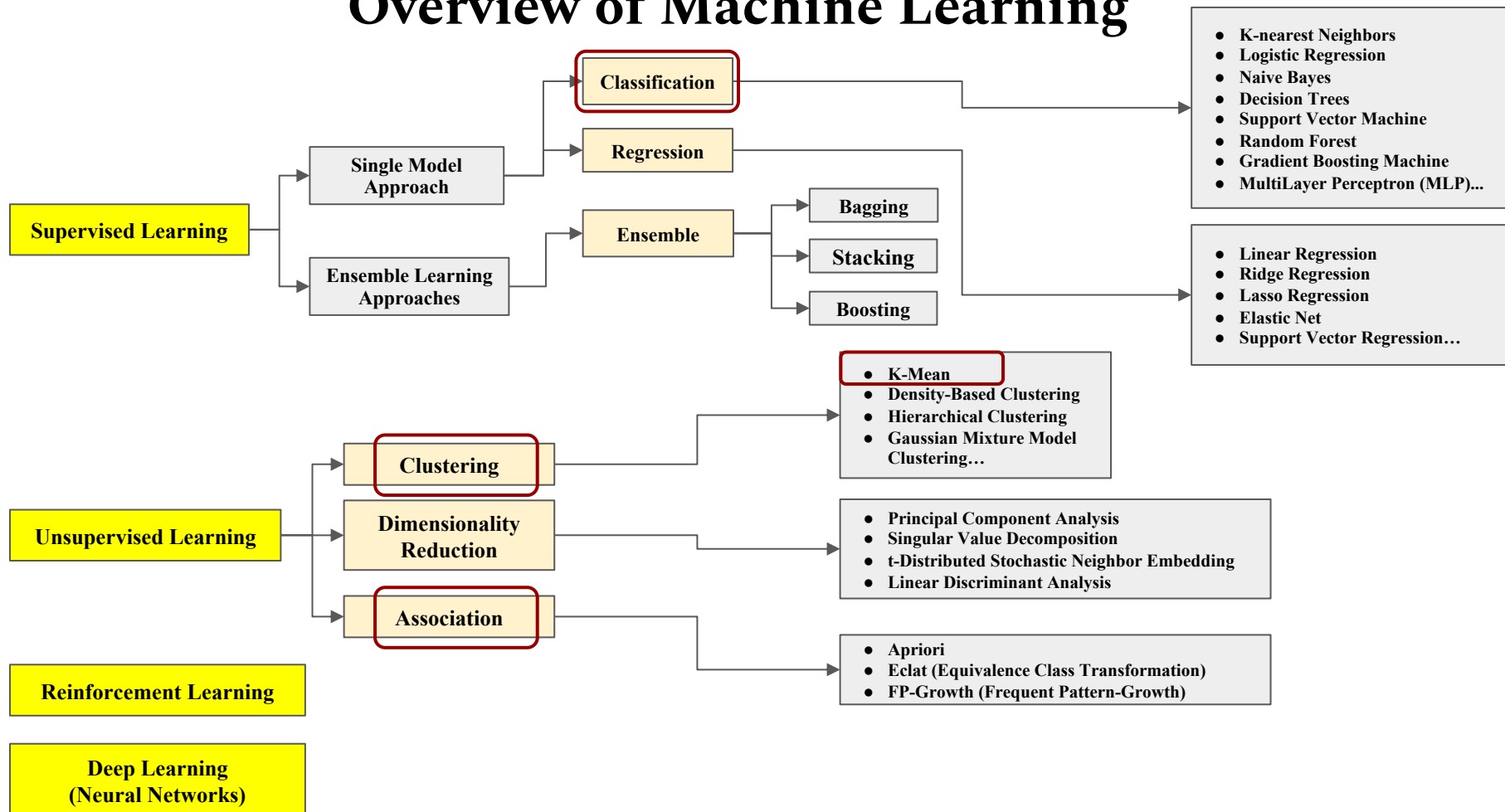
Exercises

- 1. Data Preprocessing**
- 2. Feature selection** (Filter & wrapper methods)
- 3. Association rule** (Apriori Algorithm)
- 4. Classification** (Decision Tree)
- 5. Classification** (Random Forest)
- 6. Classification** (SVM)
- 7. Classification** (Naive Bayes)
- 8. Clustering** (K-Mean)

Machine Learning Workflow



Overview of Machine Learning



Python

```
[1]: print("hello world")
```

hello world

- **Function:** `print()`
- **Argument:** "hello world"
- Input to a function are called **arguments**

function(arguments)

return value -> output

```
[2]: print(print("hello world"))
```

hello world

None

- Innermost is resolve first & then the outside
- Returning '**None**' does not mean that the function does nothing. It simply means that the function does not return any useful data, but it can still perform other actions.

```
[3]: print(type(print))  
  
      <class 'builtin_function_or_method'>
```

- **print** is a function object that is passed as an argument to **type()**
- **print(type(print))** prints the type of the **print** function.
- **print** is not an argument by itself—it's a function object being passed as an argument to the **type()** function

raj.fetch(ball)

`object.methods(argument)`

- **raj** - **object**
 - **fetch** - **method**
 - **ball** - **argument**
-
- Input to a function are called **arguments**
 - class-specific functions present inside the object are called **methods**
 - A specific instance of a class is **object**

```
[4]: x="hello"  
     print(x.upper())
```

HELLO

- **Object:** x (which store the string "hello")
- **Method:** upper()
- **Argument:** none

```
[5]: y=5  
     print(y.upper())
```

```
-----  
AttributeError                                Traceback (most recent call last)  
Cell In[5], line 2  
      1 y=5  
----> 2 print(y.upper())  
  
AttributeError: 'int' object has no attribute 'upper'
```

- **AttributeError:** means that the object doesn't have this **method**

- **dir():** returns all properties and methods of the specified object, without the values
- **Returns details of the object class**

```
[6]: x = 5
      print(dir(x))

['__abs__', '__add__', '__and__', '__bool__', '__divmod__', '__doc__', '__eq__', '__float__']
```

```
[18]: a = x.__add__(5)
      b = x + 5
      print(f'a: {a}')
      print(f'b: {b}')
```

```
a: 10
b: 10
```

Container object

- A container object is an object that contains other objects. Container objects provide a way to store and organize multiple objects, allowing you to access, iterate over, and manipulate them collectively.

```
[19]: train = [1, "string", 5.0, -50, print]  
print(train)
```

```
[1, 'string', 5.0, -50, <built-in function print>]
```

```
[21]: print(train[2])  
print(train.__getitem__(2))
```

```
5.0
```

```
5.0
```

For loop

```
[30]: numbers = [1,2,3,4,5,6]  
      print(numbers)
```

```
      for number in numbers:  
          print(number)
```

```
[1, 2, 3, 4, 5, 6]
```

```
1
```

```
2
```

```
3
```

```
4
```

```
5
```

```
6
```

Key Libraries:

- **NumPy:** For numerical operations.
- **Pandas:** For data manipulation and analysis.
- **Matplotlib/Seaborn:** For data visualization.
- **Scikit-learn:** For machine learning algorithms and tools.
- **TensorFlow/PyTorch:** For deep learning models.

Machine Learning Workflow

- 1. Import the libraries**
- 2. Load Dataset**
- 3. Exploratory Data Analysis**
- 4. Data Pre-processing**
- 5. Building a Machine Learning Model**
- 6. Evaluating the Model**

