

Machine Learning

Types and Methods of ML

Computer & Health Informatics
Department

Introduction

- There are various ML algorithms, techniques and methods that can be used to build models for solving real-life problems by using data.
- In this chapter, we are going to discuss such different kinds of methods.

Different Types of Methods

- The following are various ML methods based on some broad categories –
 - Based on human supervision
 - Unsupervised Learning
 - Semi-supervised Learning
 - Reinforcement Learning

Supervised Learning

- **How it works:** This algorithm consist of a target / outcome variable (or dependent variable) which is to be predicted from a given set of predictors (independent variables).
- Using these set of variables, we generate a function that map inputs to desired outputs.
- The training process continues until the model achieves a desired level of accuracy on the training data.
- Examples of Supervised Learning: Decision Tree

Unsupervised Learning

- **How it works:** In this algorithm, we do not have any target or outcome variable to predict / estimate.
- It is used for clustering population in different groups, which is widely used for segmenting customers in different groups for specific intervention.
- Examples of Unsupervised Learning: K-means.

Reinforcement Learning

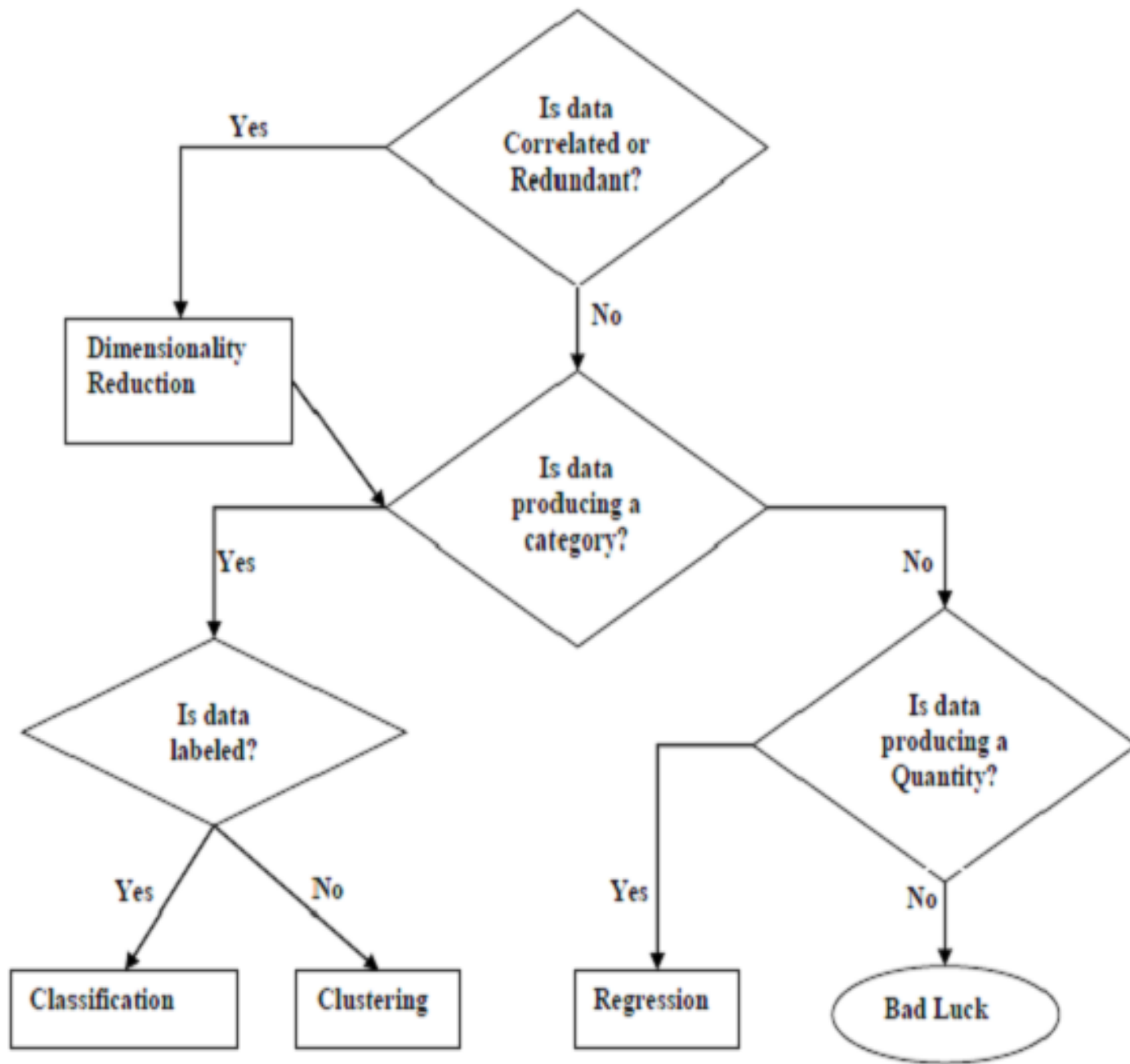
- How it works: Using this algorithm, the machine is trained to make specific decisions.
- It works this way: the machine is exposed to an environment where it trains itself continually using trial and error.

Reinforcement Learning

- This machine learns from past experience and tries to capture the best possible knowledge to make accurate business decisions.
- Example of Reinforcement Learning: Markov Decision Process

Tasks Suited for Machine Learning

- The following diagram shows what type of task is appropriate for various ML problems :



Based on learning ability

- In the learning process, the following are some methods that are based on learning ability :
 - Batch Learning
 - Online Learning

Batch Learning

- In many cases, we have end-to-end Machine Learning systems in which we need to train the model in one go by using whole available training data.
- Such kind of learning method or algorithm is called **Batch or Offline learning**.
- It is called Batch or Offline learning because it is a one-time procedure and the model will be trained with data in one single batch.

Batch Learning

- The following are the main steps of Batch learning methods

–

Step 1 – First, we need to collect all the training data for start training the model.

Step 2 – Now, start the training of model by providing whole training data in one go.

Step 3 – Next, stop learning/training process once you got satisfactory results/performance.

Step 4 – Finally, deploy this trained model into production. Here, it will predict the output for new data sample.

Online Learning

- It is completely opposite to the batch or offline learning methods.
- In these learning methods, the training data is supplied in multiple incremental batches, called mini-batches, to the algorithm.

Online Learning

- Followings are the main steps of Online learning methods –

Step 1 – First, we need to collect all the training data for starting training of the model.

Step 2 – Now, start the training of model by providing a mini-batch of training data to the algorithm.

Step 3 – Next, we need to provide the mini-batches of training data in multiple increments to the algorithm.

Step 4 – As it will not stop like batch learning hence after providing whole training data in mini-batches, provide new data samples also to it.

Step 5 – Finally, it will keep learning over a period of time based on the new data samples.

Based on Generalization Approach

- In the learning process, followings are some methods that are based on generalization approaches
 - Instance based Learning
 - Model based Learning

Instance based Learning

- Instance based learning method is one of the useful methods that build the ML models by doing generalization based on the input data.
- It is opposite to the previously studied learning methods in the way that this kind of learning involves ML systems as well as methods that uses the raw data points themselves to draw the outcomes for newer data samples without building an explicit model on training data.

Instance based Learning

- In simple words, instance-based learning basically starts working by looking at the input data points and then using a similarity metric, it will generalize and predict the new data points.

Model based Learning

- In Model based learning methods, an iterative process takes place on the ML models that are built based on various model parameters, called hyperparameters and in which input data is used to extract the features.

Model based Learning

- In this learning, hyperparameters are optimized based on various model validation techniques.
- That is why we can say that Model based learning methods uses more traditional ML approach towards generalization.

Data Loading for ML Projects

- Suppose if you want to start a ML project then what is the first and most important thing you would require?
- It is the data that we need to load for starting any of the ML project.
- With respect to data, the most common format of data for ML projects is CSV (comma-separated values).

Data Loading for ML Projects

- Basically, CSV is a simple file format which is used to store tabular data (number and text) such as a spreadsheet in plain text.
- In Python, we can load CSV data into with different ways but before loading CSV data we must have to take care about some considerations.

Consideration While Loading CSV data

- CSV data format is the most common format for ML data, but we need to take care about following major considerations while loading the same into our ML projects.

Consideration While Loading CSV data

- File Header
- In CSV data files, the header contains the information for each field.
- We must use the same delimiter for the header file and for data file because it is the header file that specifies how should data fields be interpreted.
- The following are the two cases related to CSV file header which must be considered –

Consideration While Loading CSV data

- **Case-I: When Data file is having a file header** – It will automatically assign the names to each column of data if data file is having a file header.
- **Case-II: When Data file is not having a file header** – We need to assign the names to each column of data manually if data file is not having a file header.
- In both the cases, we must need to specify explicitly whether our CSV file contains header or not.

Consideration While Loading CSV data

- **Comments**
- Comments in any data file are having their significance.
- In CSV data file, comments are indicated by a hash (#) at the start of the line.
- We need to consider comments while loading CSV data into ML projects because if we are having comments in the file then we may need to indicate, depends upon the method we choose for loading, whether to expect those comments or not.

Consideration While Loading CSV data

- **Delimiter**
- In CSV data files, comma (,) character is the standard delimiter.
- The role of delimiter is to separate the values in the fields.
- It is important to consider the role of delimiter while uploading the CSV file into ML projects because we can also use a different delimiter such as a tab or white space.
- But in the case of using a different delimiter than standard one, we must have to specify it explicitly.

Consideration While Loading CSV data

- **Quotes**
- In CSV data files, double quotation (“ ”) mark is the default quote character.
- It is important to consider the role of quotes while uploading the CSV file into ML projects because we can also use other quote character than double quotation mark.
- But in case of using a different quote character than standard one, we must have to specify it explicitly.

Methods to Load CSV Data File

- While working with ML projects, the most crucial task is to load the data properly into it.
- The most common data format for ML projects is CSV and it comes in various flavors and varying difficulties to parse.
- In this section, we are going to discuss about three common approaches in Python to load CSV data file –

Methods to Load CSV Data File

- **Load CSV with Python Standard Library**
- The first and most used approach to load CSV data file is the use of Python standard library which provides us a variety of built-in modules namely *csv module* and the *reader()*function.
- The following is an example of loading CSV data file with the help of it –

Methods to Load CSV Data File

- **Example**
- In this example, we are using the *iris flower data set* which can be downloaded into our local directory. After loading the data file, we can convert it into
- NumPy :array and use it for ML projects.
- Following is the Python script for loading CSV data file
- First, we need to import the csv module provided by Python standard library as follows –

Methods to Load CSV Data File

```
import csv
```

- Next, we need to import Numpy module for converting the loaded data into NumPy array.

```
import numpy as np
```

- Now, provide the full path of the file, stored on our local directory, having the CSV data file –

```
path = r"c:\iris.csv"
```

Methods to Load CSV Data File

- Next, use the `csv.reader()` function to read data from CSV file :

```
with open(path, 'r') as f:  
    reader = csv.reader(f, delimiter = ',')  
    headers = next(reader) data = list(reader)  
    data = np.array(data).astype(float)
```


Methods to Load CSV Data File

- We can print the names of the headers with the following line of script –

```
print(headers)
```

- The following line of script will print the shape of the data i.e. number of rows & columns in the file

```
print(data.shape)
```

Next script line will give the first three line of data file

```
print(data[:3])
```

Methods to Load CSV Data File

- Output

```
['sepal_length', 'sepal_width', 'petal_length',  
'petal_width']
```

```
(150, 4)
```

```
[[5.1 3.5 1.4 0.2]
```

```
[4.9 3. 1.4 0.2]
```

```
[4.7 3.2 1.3 0.2]]
```

Load CSV with NumPy

- Another approach to load CSV data file is *NumPy* and *numpy.loadtxt()* function.
- The following is an example of loading CSV data file with the help of it –

Load CSV with NumPy

- **Example**
- In this example, we are using the Pima Indians Dataset having the data of diabetic patients.
- This dataset is a numeric dataset with no header.
- It can also be downloaded into our local directory.
- After loading the data file, we can convert it into *NumPy* array and use it for ML projects.
- The following is the Python script for loading CSV data file —

```
from numpy import loadtxt
path = r"C:\pima-indians-diabetes.csv"
datapath= open(path, 'r')
data = loadtxt(datapath, delimiter=",")
print(data.shape)
print(data[:3])
```

Output

```
(768, 9)
[[ 6. 148. 72. 35. 0. 33.6 0.627 50. 1.]
 [ 1. 85. 66. 29. 0. 26.6 0.351 31. 0.]
 [ 8. 183. 64. 0. 0. 23.3 0.672 32. 1.]]
```

Thank you