

Multiclass Classification using Support Vector Machine (SVM) in Python

*Applied Machine Learning and Data Science
Recipe - 009*

In this ***Data Science Recipe***, the reader will learn

- a) Different types of Machine Learning problems.
- b) How to organise a Predictive Modelling Machine Learning project.
- c) Implementation steps in Applied Machine Learning project.
- d) Different elements of data used for predictive modelling.
- e) How to install Python and MySQL.
- f) How to load Dataset from RDBMS.
- g) How to summarize and visualize Dataset.
- h) How to implement **SVM Algorithms** for **Multiclass Classification** in Python.
- i) How to manually tune parameters of **SVM Models in scikit-learn**.
- j) How to train a model and perform Cross Validation (CV).
- k) How to compare CV results of different models/algorithms.
- l) How to report results for trained models and compare.
- m) How to implement an end-to-end Data Science Project using MySQL and Python.

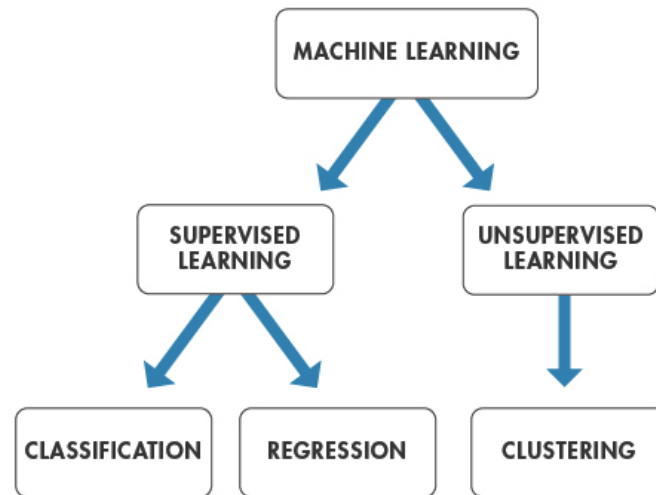
What is Machine Learning?

Machine learning is the science of getting computers to act without being explicitly program. It is a subset of AI: Artificial Intelligence. Predictive modelling is a branch of Machine Learning that particularly deals with tabular data to explicitly find patterns and/or insights from the data available.

Types of Machine Learning Problems

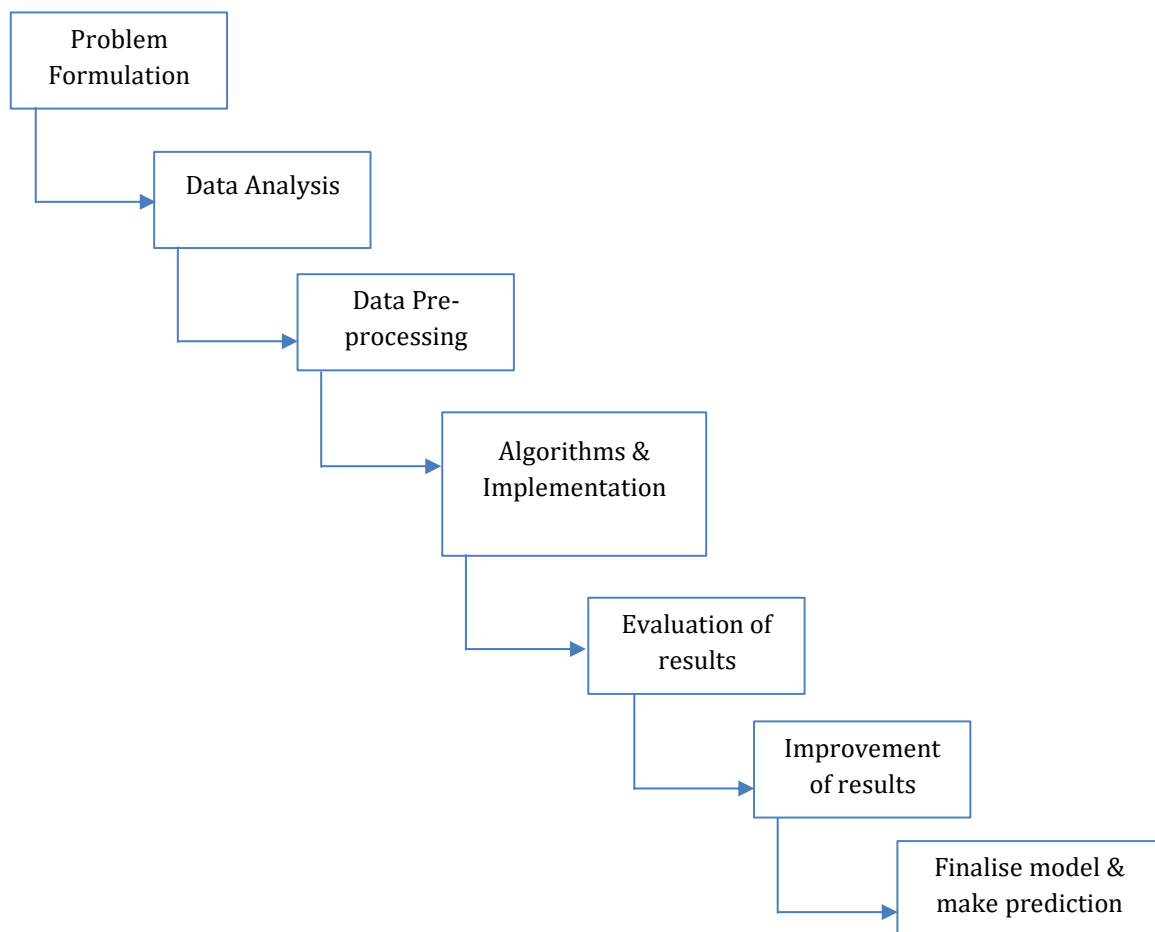
There are common classes of problems in Machine Learning. The problems discussed below are standards for most of the ML based predictive modelling problems.

- **Classification (or Supervised Learning):** Data are labelled meaning that they are assigned to classes, for example spam/non-spam or fraud/non-fraud. The decision being modelled is to assign labels to new unlabelled pieces of data. Classification should be Binary classification and Multi-class classification.
- **Regression (or Supervised Learning):** Data are labelled with a real value (think of a real number) rather than a label/class. Examples that are easy to understand are time series data like the price of a stock over time, monthly sales volume of a store etc. The decision being modelled is what value to predict for new unpredicted data.
- **Clustering (or Unsupervised Learning):** Data are not labelled, but can be divided into groups based on similarity and other measures of natural structure in the data.



Steps to setup a Predictive Modelling project

There are several distinctive steps involved in a predictive modelling machine learning project. These steps are very similar to the Water-Fall method in Rapid Application Development (RAD) in software engineering. An overview of these steps is given below.



Problem formulation

The first and initial step in predictive modelling machine learning is to define and formulise a problem. A data scientist (or machine learning engineer or developer) should investigate and characterise the problem to better understand the objectives and goals of the project i.e. whether it is a 'classification' or 'regression' or 'clustering' problem.

Data Analysis

A data scientist should utilise some well-understood descriptive statistics and visualisation techniques to the data available. This descriptive exploratory data analysis would help to better understand the structure of data.

Data Pre-processing

A data scientist should utilise data transformations, missing value treatment etc. in order to better expose the structure of the prediction problem to modelling algorithms.

Algorithms

A data scientist should choose out of bag predictive modelling machine learning algorithms to fit the data available. Data must be split into train and test data to report performance of each algorithm tested.

Evaluation

A data scientist should evaluate the model to report the performance using some well understood evaluation techniques such as confusion matrix for classification, RMSE estimation for regression etc.

Improvement

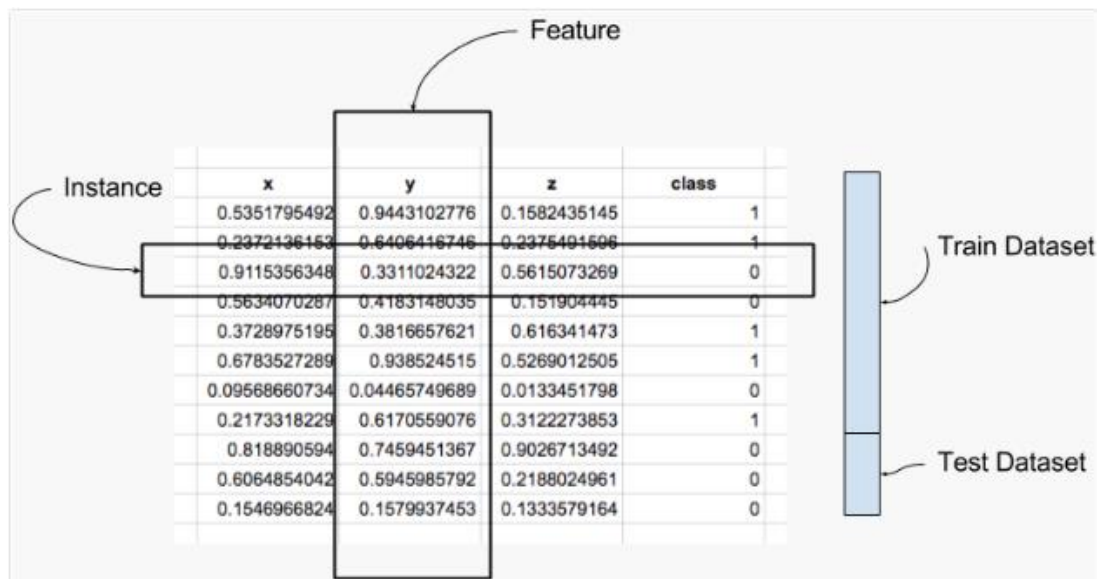
A data scientist should use algorithm tuning to further achieve the most out of the better performing algorithm on the data available.

Finalisation and prediction

Finally, the tuned model needs to finalise for making predictions on unseen data and the outcomes of the model need to be presented.

Different elements of data in predictive modelling

A predictive modelling machine learning project is primarily focused on 2D tabular data i.e. data are stored in spreadsheet and/or in database. Here a spreadsheet is shown below to describe different elements of data available.



Instance: An individual row of data in a tabular dataset is called an instance.

Feature: A single column of data in a tabular dataset is called a feature. It is also known as attribute of a data instance. There are INPUT features and OUTPUT features in a typical dataset. Sometimes OUTPUT feature(s) needs to drive from the INPUT features.

Datasets: A collection of instances and features used in predictive modelling machine learning projects is known as datasets. A dataset is usually divided into three independent datasets: a) Training dataset, b) Testing dataset and c) Validation dataset.

Training dataset: A collection of instances and features used to fit an algorithm.

Testing dataset: A collection of instances and features used to test the fitted algorithm.

Validation dataset: A collection of instances and features used to evaluate the performance of the model or fitted algorithm.

Installing Python (Anaconda 3) and MySQL

Python can be installed by using open source data science eco-systems “Anaconda 3”. The Anaconda distribution includes all necessary Python libraries for Applied Machine Learning and Data Science. Anaconda-python can be downloaded from <https://www.anaconda.com/download/#windows>

MySQL 5.7 (community version) can be downloaded from <https://dev.mysql.com/downloads/>

The Python-MySQL connector (pymysql) can be install by using conda through command prompt. The command should be: conda install pymysql

Once these software(s) are installed, the system is ready to explore data science recipes.

A gentle introduction to IRIS Flower Classification using SCIKIT-LEARN SVM Models with **Parameter Tuning**

In this data science recipe, IRIS Flower data is used to check different machine learning algorithms that are available in scikit-learn package. IRIS data is freely downloaded from UCI machine learning repository [1]. The goal of this exercise is to correctly classify each flower given its attributes. IRIS is a small and well understood dataset for classification problem. Here the author is going to present an end to end predictive modelling machine learning recipe for this classification project using **different SVM Models** available in scikit-learn as well as their tuning the hyper-parameters of these models. Interested reader could follow (www.scikit-learn.org) for mathematical formulation of those models/algorithms.

The summary of this recipe is as follows. Detailed self-instructed Python code is given in **DSR-009.py & DSR-009-codes.PDF**

```
"""
@author:
    Dr. Nilimesh Halder (Principle Data Scientist @ Crown)
    BSc in Computer Science and Engineering,
    @ Khulna University, Bangladesh.
    PhD in Artificial Intelligence and Applied Machine Learning,
    @ The University of Western Australia, Australia.
"""

# -----
# Steps in Applied Machine Learning:
#
# 1. Load Library
# 2. Load Dataset to which Machine Learning Algorithm to be applied
#    Either a) load from a CSV file or b) load from a Database
# 3. Summarisation of Data to understand dataset (Descriptive Statistics)
# 4. Visualisation of Data to understand dataset (Plots, Graphs etc.)
# 5. Data pre-processing & Data transformation
#    (split into train-test datasets)
# 6. Application of a Machine Learning Algorithm to training dataset
#    a) setup a ML algorithm and parameter settings
#    b) cross validation setup with training dataset
#    c) training & fitting Algorithm with training Dataset
#    d) evaluation of trained Algorithm (or Model) and result
#    e) saving the trained model for future prediction
# 7. Finalise the trained model and make prediction
..
```

The project is divided into several small sections. These are:

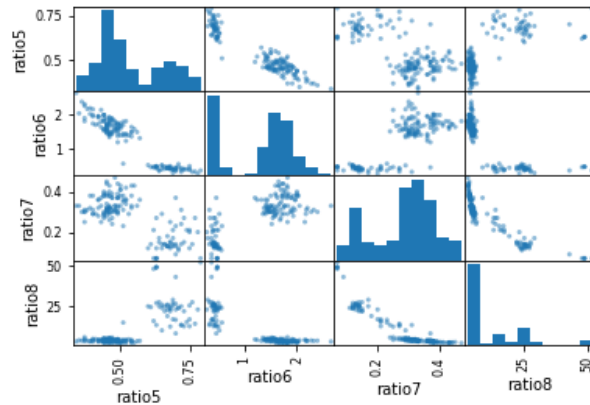
1. **Loading necessary libraries**
2. **Load Dataset Either from a CSV file or from a MySQL Table**
3. **Summarisation of Data to understand Dataset (Descriptive Statistics)**
4. **Visualisation of Data to understand Dataset (Plots, Graphs etc.)**
5. **Data pre-processing & Data transformation (split into train-test datasets)**
6. **Application of a Machine Learning Algorithm to training dataset**
 - a) **setup a ML algorithm and parameter settings**
 - b) **cross validation setup with training dataset**
 - c) **training & fitting Algorithm with training Dataset**

d) evaluation of trained Algorithm (or Model) and result

e) saving the trained model for future prediction

7. Load the saved model and apply it to new dataset for prediction

Outcomes



Cross Validation Results

Support Vector Machine: SVC – PS-1:
Mean Accuracy: 0.939373 (Std: 0.060205)

Support Vector Machine: SVC – PS-2:
Mean Accuracy: 0.929373 (Std: 0.061742)

Support Vector Machine: SVC – PS-3:
Mean Accuracy: 0.350351 (Std: 0.011104)

Support Vector Machine: NuSVC – PS-1:
Mean Accuracy: 0.939373 (Std: 0.060205)

Support Vector Machine: NuSVC – PS-2:
Mean Accuracy: 0.859323 (Std: 0.060465)

Support Vector Machine: NuSVC – PS-3:
Mean Accuracy: 0.310877 (Std: 0.077683)

Support Vector Machine: LinearSVC – PS-1:
Mean Accuracy: 0.919373 (Std: 0.069279)

Support Vector Machine: LinearSVC – PS-2:
Mean Accuracy: 0.919373 (Std: 0.069279)

Disclaimer

The information and recipe presented within this eArticle is only for educational and coaching purposes for beginners and app-developers. Anyone can practice and apply the recipe presented here, but the reader is taking full responsibility for his/her actions. The author of this recipe (code / program) has made every effort to ensure the accuracy of the information was correct at time of publication. The author does not assume and hereby disclaims any liability to any party for any loss, damage, or disruption caused by errors or omissions, whether such errors or omissions result from accident, negligence, or any other cause. Some of the information presented here could be also found in public knowledge domains.

Acknowledgement

The author of this eArticle thanks to all readers of **SETScholars: Applied Machine Learning and Data Science Recipes**. The author also thanks to the publishers and contributors of open source datasets at **UCI Machine Learning Repository** [1] and **Kaggle** [2] as well as acknowledges their kind efforts.

References

- [1] UCI Machine Learning Repository (<http://archive.ics.uci.edu/ml/index.php>).
- [2] Kaggle Data Science (<https://kaggle.com>)
- [3] SCIKIT-LEARN (<http://scikit-learn.org>)