Diabetes Classification using NB, KNN, SGD and DT classifiers: An approach to Grid Search and Random Search parameter tuning in Python

Applied Machine Learning and Data Science Recipe - 034

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

a) How to organise a Predictive Modelling Machine Learning project step by step.
b) What are the different steps in Predictive Modelling and Applied Machine Learning.
c) How to summarise and present feature variables in Predictive Modelling (Descriptive statistics).
d) How to visualise features through histogram, density plot, box plot and scatter matrix.
e) How to find correlations among features variables.
f) How to visualise target variables.
g) How to do data analysis for feature and target variables.
h) How to utilise **sklearn** and **pandas** packages in Python.
i) How to implement **NB, KNN, SGD and DT classifiers** for Binary Classification in Python.
j) How to setup **NB, KNN, SGD and DT** hyper-parameters: **manual** and **automatic** tuning in Python.
k) How to setup **RandomSearchCV** and **GridSearchCV** for parameter tuning in Python.
l) How to perform **K-fold Cross Validation** in Python.
m) How to compare **classifiers** with Accuracy and Kappa in Python.
n) How to implement an end-to-end Data Science Project using MySQL, scikit-learn 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

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

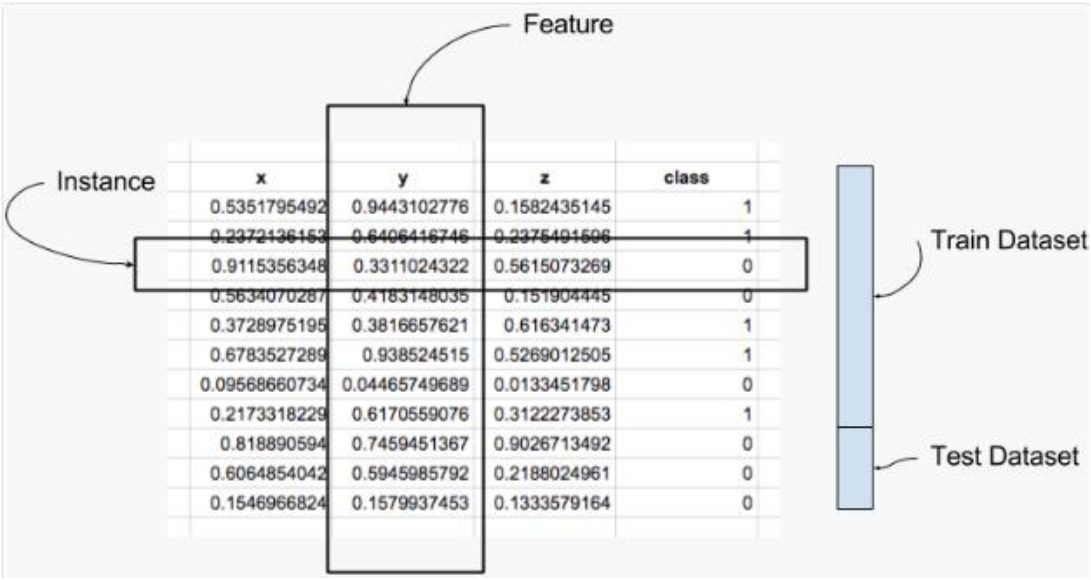
A data scientist should use algorithm tuning to further achieve the most out of the better performing algorithm on the data available.

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 OUTOUT 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.

Result from this Data Science Recipe

Disclaimer

The information and codes presented within this recipe 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 recipe 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)