AN OVERVIEW OF PREDICTIVE MODELLING USING R AND PYTHON

D V JEYANTHI,
Assistant Professor, Department of Computer Science, Sourashtra College, Madurai -625004.
Email : dvjeyanthi@gmail.com

ABSTRACT

Big data analytics is the process of collecting, organizing and analyzing large sets of data to discover patterns and other useful information. To analyze such a large volume of data, big data analytics is typically performed using specialized analytics software tools and applications. Data Scientists working in Data Science have big confusion on selecting the language for Predictive Modelling. Predictive Modelling is performed on various areas like actuarial science, health care, financial services, insurance, stock market, satellite image predictions, natural language processing, business, social media, location services, sensors and much more. R is an open source programming language and software environment for statistical computing and graphics. Python is a general purpose programming language used by data scientists, easy to learn, integrates well with other databases and tools like Spark and Hadoop. It has great computational intensity and has powerful data analytics libraries. This paper provides an overview of packages and libraries available in R and Python. It also compares various facts in R and Python. This helps the predictive modellers for selecting Predictive Modelling Language to meet their objective.
1. INTRODUCTION

Predictive modelling is a process that uses data mining and probability to forecast outcomes. Each model is made up of a number of predictors, which are variables that are likely to influence future results. Once data has been collected for relevant predictors, a statistical model is formulated. Predictive analytics is the practice of extracting information from existing data sets in order to determine patterns and predict future outcomes and trends. Predictive models and analysis are typically used to forecast future probabilities. Applied to business, predictive models are used to analyze current data and historical facts in order to better understand customers, products and partners and to identify potential risks and opportunities for a company. It uses a number of techniques, including data mining, statistical modeling and machine learning to help analysts make future business forecasts.

2. R

2.1 Introduction

R is an open source programming language and software environment for statistical computing and graphics that is supported by the R Foundation for Statistical Computing. The R language is widely used among statisticians and data miners for developing statistical software and data analysis.

R is an integrated suite of software facilities for data manipulation, calculation and graphical display. It includes an effective data handling and storage facility. It provides a suite of operators for calculations on arrays, matrices. It provides a large, coherent, integrated collection of intermediate tools for data analysis. Provides graphical facilities for data analysis and display either on-screen or on hardcopy, and a well-developed, simple and effective programming language which includes conditionals, loops, user-defined recursive functions and input and output facilities.

Many users think of R as a statistics system. It of an environment within which statistical techniques are implemented. R can be extended easily via packages.

2.2 Packages in R

R provides packages like Ggplot2, googleVis for Data Visualization, plyr, data.table for Data Transformation, DissForest, MissMDA for Missing Value Imputations, Outliers, EVIr for Outlier Detection, Features, RRF for Feature Selection, FactoMineR, CCP for Dimension Reduction, Car, randomforest for Continuous Regression, Rminer, CoreLearn for Ordinal Regression, Care, BifRF for Classification, CBA, RankCluster for Clustering, Forecast, LTSA for TimeSeries, Survival, Basta for Survival, LSMMeans, Comparison for General Model Validation, RegTest, ACD for Regression Validation, BinomTools, BAIM for Classification Validation, ClustEval, SigClust for Clustering Validation, PROC, TimeRoc for ROC Analysis, Other libraries like RCPP, parallel, tm, twitteR, swirl, reshape2, Rmomygo, Rmarkdown and much more.

Importing Data: R offers wide range of packages for importing data available in any format such as .txt, .csv, .json, .sql etc. To import large files of data quickly, it is advisable to install and use data.table, readr, RMySQL, sqldf, jsonlite.

Data Visualization: R has in-built plotting commands to create simple graphs. For advanced graphics. Hence install ggplot2.

Data Manipulation: R has packages for data manipulation, dplyr, plyr, tidyr, lubridate, stringr packages allows to do basic and advanced computations quickly..

Modeling / Machine Learning: For modeling, caret package in R is powerful enough to cater to every need for creating machine learning model. Install packages algorithms wise such as randomForest, rpart, gbm etc
Packages such as dplyr, tidyr, readr, data.table, SparkR, ggplot2 have made data manipulation, visualization and computation much faster.

3.1 PYTHON

Python is a functional and flexible programming language that is powerful enough for experienced programmers to use. Python is a well-developed programming language that is adaptable for both small and large development projects. Companies of all sizes and in all areas from the biggest investment banks to the smallest social/mobile web app startups are using Python to run their business and manage their data. Python for Big Data analysis is all about manipulating, processing, cleaning, and crunching Big Data in Python.

It is an interpreted language rather than compiled language and hence might take up more CPU time. It is for data science and production of web based analytics products.

3.2 Python Libraries

Following are the list of libraries available for scientific computations and data analysis:

**NumPy** stands for Numerical Python. The most powerful feature of NumPy is n-dimensional array. This library also contains basic linear algebra functions, Fourier transforms, advanced random number capabilities and tools for integration with other low level languages like Fortran, C and C++

**SciPy** stands for Scientific Python. SciPy is built on NumPy. It is one of the most useful library for variety of high level science and engineering modules like discrete Fourier transform, Linear Algebra, Optimization and Sparse matrices.

**Matplotlib** for plotting vast variety of graphs, starting from histograms to line plots to heat plots. Pylab feature can be used in ipython notebook (ipython notebook –pylab = inline) to use these plotting features inline. If inline option is ignored, then pylab converts ipython environment to an environment, very similar to Matlab. Latex commands can be used to add math to the plot.

**Pandas** for structured data operations and manipulations. It is extensively used for data munging and preparation. Pandas were added relatively recently to Python and have been instrumental in boosting Python’s usage in data scientist community.

**Scikit Learn** for machine learning. Built on NumPy, SciPy and matplotlib, this library contains a lot of efficient tools for machine learning and statistical modeling including classification, regression, clustering and dimensionality reduction.

**Statsmodels** for statistical modeling. Statsmodels is a Python module that allows users to explore data, estimate statistical models, and perform statistical tests. An extensive list of descriptive statistics, statistical tests, plotting functions, and result statistics are available for different types of data and each estimator.

**Seaborn** for statistical data visualization. Seaborn is a library for making attractive and informative statistical graphics in Python. It is based on matplotlib. Seaborn aims to make visualization a central part of exploring and understanding data.

**Bokeh** for creating interactive plots, dashboards and data applications on modern web browsers. It empowers the user to generate elegant and concise graphics in the style of D3.js. Moreover, it has the capability of high performance interactivity over very large or streaming datasets.

**Blaze** for extending the capability of Numpy and Pandas to distributed and streaming datasets. It can be used to access data from a multitude of sources including Bcolz, MongoDB, SQLAlchemy, Apache Spark, PyTables, etc. Together with Bokeh, Blaze can act as a very powerful tool for creating effective visualizations and dashboards on huge chunks of data.

**Scrappy** for web crawling. It is a very useful framework for getting specific patterns of data. It has the capability to start at a website home url
and then dig through web-pages within the website to gather information.

SymPy for symbolic computation. It has wideranging capabilities from basic symbolic arithmetic to calculus, algebra, discrete mathematics and quantum physics. Another useful feature is the capability of formatting the result of the computations as LaTeX code.

Requests for accessing the web. It works similar to the the standard python library urllib2 but is much easier to code.

4. R Vs. Python for Predictive Modelling

Predictive Modelling is the process of using known results to create, process, and validate a model that can be used to forecast future outcomes. Predictive Modeling is a tool used in Predictive Analytics, a data mining technique which attempts to answer the question “what might possibly happen in the future?”. Most widely used predictive modeling techniques are regression and neural networks. In the field of statistics, regression refers to a linear relationship between the input and output variables. A predictive model with a linear function requires one predictor or feature in order to predict the output/outcome.

In the field of statistics, regression refers to a linear relationship between the input and output variables. A predictive model with a linear function requires one predictor or feature in order to predict the output/outcome.

4.1 Statistics Language

R was primarily built to help data scientists to run complex data science algorithms while Python evolved as a general purpose programming language.

Any analytics project related to Predictive Analytics is done in two phases:1. Model Building 2. Real time prediction

As R was built only for data scientists and statisticians, it beats Python in first phase but the revolution of production system was concurrent to the evolution of Python, hence Python easily integrates with your production code written in other languages like Java or C++ etc.

Practically, when it comes to Predictive Analytics or Machine Learning both languages have pretty good packages written.

4.2 Handling Big Data Sets

Data scientists or statisticians were able to handle the data and running Predictive Analytics using R which stores data in computers’ RAM.

This is one of the major drawbacks of R that it does just in-memory computations.

R can be installed on a server and run the machine algorithms or any other statistical analysis.

Apart from the option of server installation, R and Python - both have capability to connect to Hadoop HDFS and do parallel computing.

4.3 Visualization

Data Visualization is indeed the first part which is needed even before running the first iteration of the model. There are various examples where graphs can tell a story better than a machine learning algorithm. Python couldn’t compete with R when it comes to data visualization. ggplot is the best tool for statistical data visualizations.

4.4 Learning

R assumes that the objective is “Statistical Learning” and tries to make it cooler for you to understand and diagnose the predictive model built.

Scikit-learn is the mostly used Python package for machine learning which helps to tune the model or switch between different models but it’s hard to diagnose the model with Scikit-learn in Python.

It difficult to code in R. Python codes are easy. Hence, learning curve of R is proven to be steeper than Python.
4.5 Code Repositories

CRAN is a huge repository for R packages. Packages are collection of R functions, data and compiled code. Installed in single line code.

PyPi Python package is little bit complicated.

Python is easier to adapt for people with programming background using other languages like JAVA, FORTRAN, C++ etc.

In general when we compare the feature of R and Python:

<table>
<thead>
<tr>
<th>FEATURE</th>
<th>WHICH LANGUAGE IS BETTER</th>
</tr>
</thead>
<tbody>
<tr>
<td>MODEL BUILDING</td>
<td>BOTH SIMILAR</td>
</tr>
<tr>
<td>MODEL INEPRETABILITY</td>
<td>R</td>
</tr>
<tr>
<td>PRODUCTION</td>
<td>PYTHON</td>
</tr>
<tr>
<td>COMMUNITY SUPPORT</td>
<td>R</td>
</tr>
<tr>
<td>LIBRARIES</td>
<td>BOTH SIMILAR</td>
</tr>
<tr>
<td>VISUALISATION</td>
<td>R</td>
</tr>
<tr>
<td>LEARNING CURVE</td>
<td>PYTHON</td>
</tr>
</tbody>
</table>

5. Conclusion

R is for complex statistical model. R community is much stronger tha Python community. R was built specifically to help Data Science. Statistics/Analytics peoples feels more comfortable with R. People good in mathematics prefers R. For Business models, HealthCare, Insurance, Social Meadia R is helpful. ComputerScience Developers feels more comfortable with python since it is a general purpose programming language. Python provides wonderful keras package. For Image Recognition, Video Analysis, Natural Language Processing Python works better. Selection of R or Python depends on the objective on which predictive model is going to be built.

6. References


2. Python for Data Analysis - Data Wrangling with Pandas, NumPy, and Ipython from Wes McKinney published by O'Reilly Media