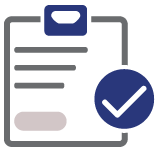
Terminology

The table below outlines the different course activities:

|  |  |
| --- | --- |
| ACTIVITY | DESCRIPTION |
| Demonstration | The Instructor will demonstrate the workflow, outlining the key concept(s). The student is not expected to replicate the Instructors demonstration; but understand the key concept(s) and workflow. |
| Lab | The Instructor will outline the key concepts, features and options. The student is expected to follow along with the instructor so that they understand the key concept(s), features and options for the Exercise |

The icon indicates an Info Tip. Info Tips help users understand unfamiliar workflows or actions.

The icon indicates that you need to be careful when implementing or configuring the step / option(s).

*The icon indicates Best Practice. A Best Practice is a method or technique that has been generally accepted as a standard way of doing things.*

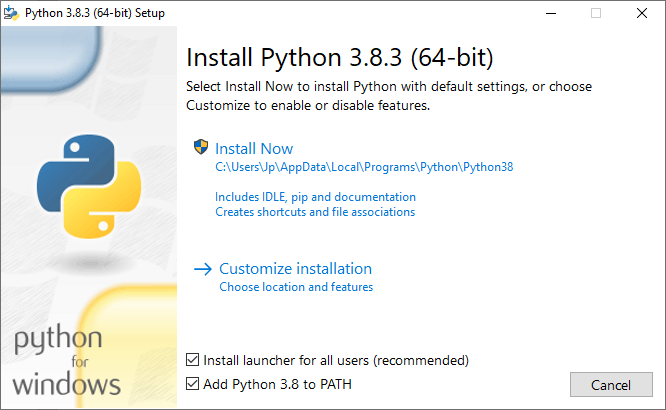
Pre-requisites

The following pre-requisites need to be completed:

* Install Python for Windows
* Install R for Windows
* Set R Environmental Variables
* Install R Studio for Windows
* Configure Pentaho Data Integration with R

Installation of Python

1. To install: download Python from [Python for Windows](https://www.python.org/downloads/windows/)
2. Click on: the option " Download Windows x86-64 executable installer" on the page.
3. Run: python-3.8.3-amd64.exe file and follow the installation instructions.

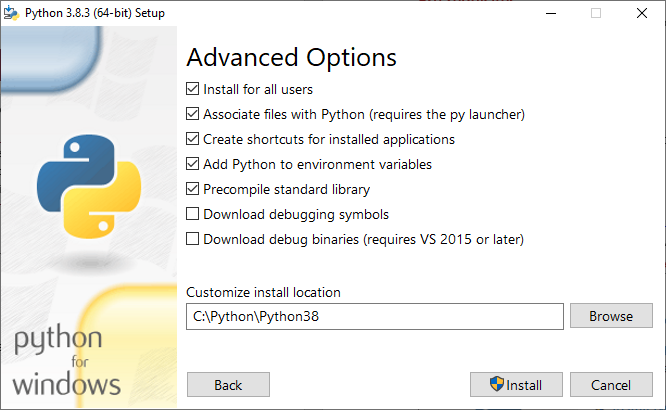


1. Ensure you select:

* Customize Installation
* Install launcher
* Add Python 3.8 to PATH

1. Keep default options and click: Next
2. Ensure you select:

* Install for all users
* Precompile standard library
* Change the Path to: C:\Python\Python38

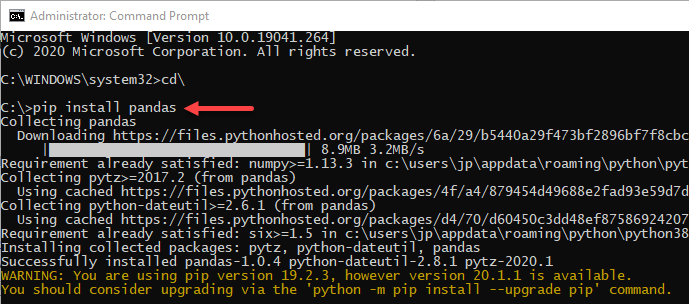


1. Click: Install

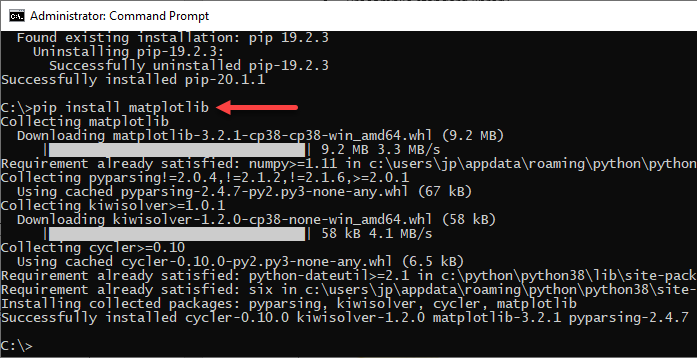
The following libraries need to be installed:

* pandas
* matplotlib
* py4j
* numpy
* TPOT

1. Open a Command Prompt (Admin) window
2. Enter the following command to install pandas: pip install pandas



1. Repeat to download and install the other required libraries



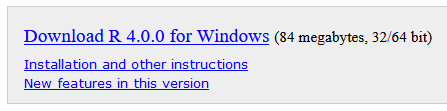
You may need to restart your system.

Installation of R

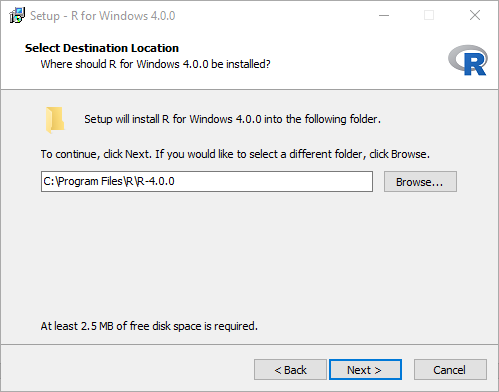
1. To install: download R from [r-project](https://cran.r-project.org/mirrors.html)

* Select a CRAN location (a mirror site) and click the corresponding link

1. Click on: the "Download R for Windows" link at the top of the page.
2. Click on: “install R for the first time" link at the top of the page.

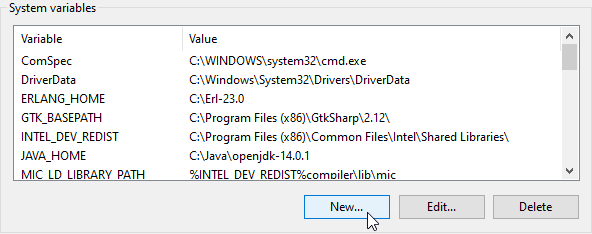


1. Click: Download R <version> for Windows
2. Run: R-4.0.0-win.exe file and follow the installation instructions.

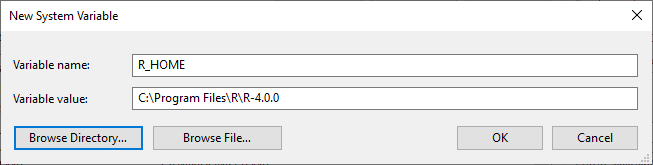


Set R Environmental Variables

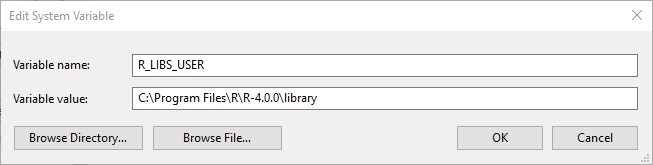
1. Go to: Control Panel > System > Advanced System Settings.
2. Click: Environment Variables button.
3. Under System Variables, click: New.



1. In the Variable Name field, enter: R\_HOME
2. Browse for the directory: C:\Program Files\R\R-4.0.0

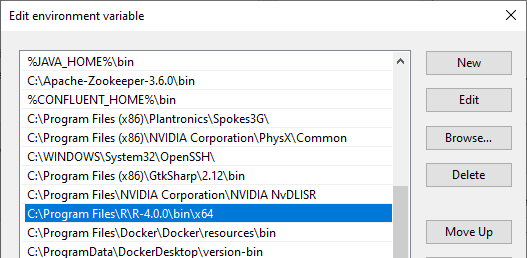


1. Repeat to add the variable: **R\_LIBS\_USER**
2. Browse for the directory: C:\Program Files\R\R-3.6.3\library



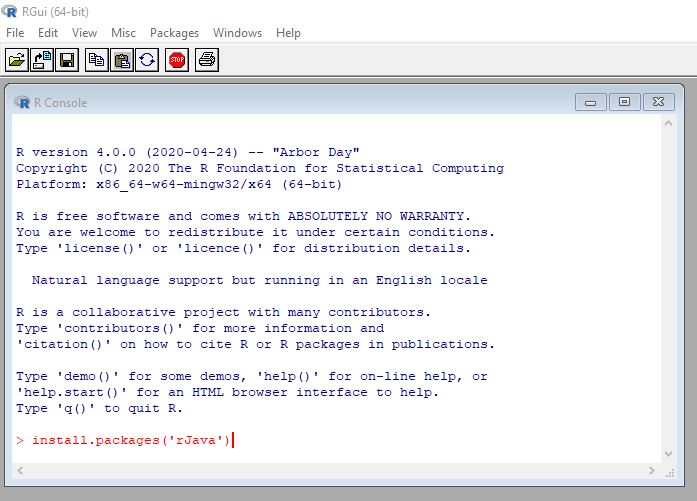
1. Add to the Path the location of the R executable: %R\_HOME%\bin\x64

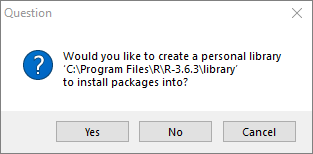
* Ensure the path references rcmd.exe and r.dll



1. Start R. In the R Console: Run the following command:

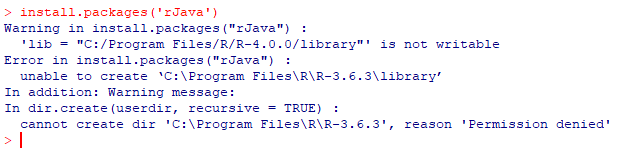
install.packages('rJava')

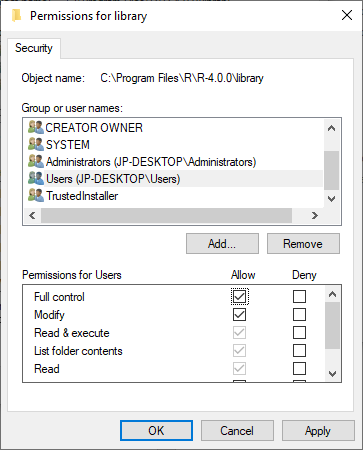


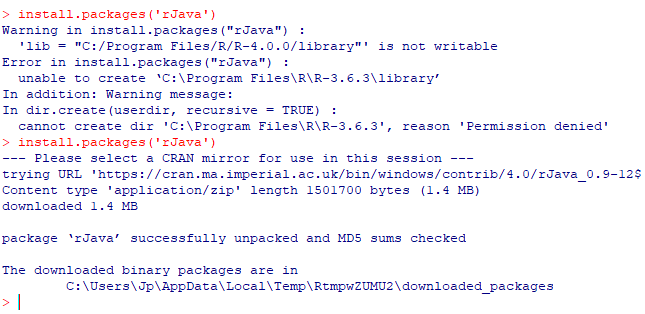
* If prompted with "Would you like to use a personal library instead?" click Yes.
* 
* If prompted with the path of the library, click Yes.
* When prompted for the CRAN mirror, choose a country then click OK.



*You may be denied permission writing to the library folder. You will need to change the permission for the folder.*



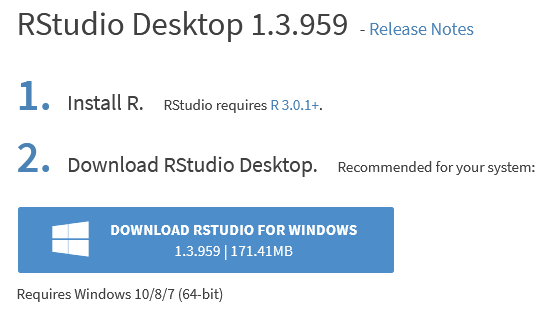




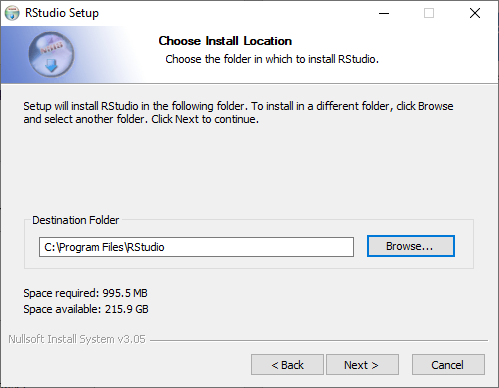
1. After rJava has successfully been installed, type q() to quit the R console.
2. Click Yes to close the workplace image.
3. Close R.

To Install RStudio

1. To install: download R Studio from [R Studio IDE](https://rstudio.com/products/rstudio/download/#download)
2. Click on the "Download RStudio for Windows" button.



1. Run: RStudio-1.3.959.exe file and follow the installation instructions.



Configure Pentaho Data Integration with R

In the rJava directory there is a jri.dll file that needs to be copied into the libswt directory of Spoon.

1. Stop: Spoon, if it's running
2. Find: %R\_LIBS\_USER%/rJava/jri/x64/jri.dll
3. Copy: jri.dll to the following directory

Windows: ﻿[Pentaho directory]/client-tools/data-integration/libswt/win64

Linux: [Pentaho directory]/client-tools/data-integration/libswt/linux



*Further details can be found at:* [R on PDI](https://support.pentaho.com/hc/en-us/articles/360000307943-Pentaho-Data-Integration#WPR)

Verifying Your Installation

1. Open a new transformation in PDI.
2. Drag an R Script Executor step onto the canvas.
3. Double-click the step and select the middle tab, R Script. You will see some comments at the top of the window:

# The main output is expected to be a data frame, unless "Output

# from script is text" is checked. So, to output a data frame the

# last statement in the script should be the name of the frame.

# In the case that the output is text (as would be seen on the

# R console), the last statement should be a "print" statement in

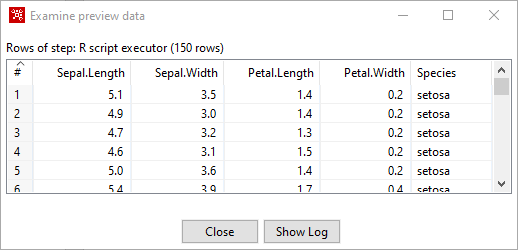
# order to print the object required.

1. Beneath the comment above, enter this code:

library(datasets)

iris

1. Once you have entered this code in the R Script tab, click the **Test Script** button on this tab.



# Lab 1: Credit Card Fraud - AutoML

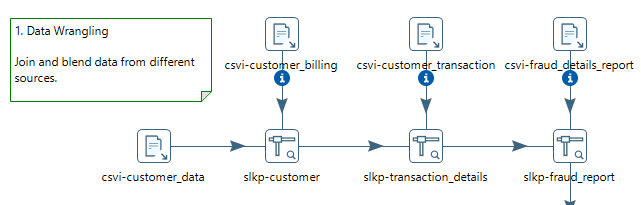
|  |  |  |
| --- | --- | --- |
| Introduction | Imagine that a direct retailer wants to reduce losses due to orders involving fraudulent use of credit cards. They accept orders via phone and their web site, and ship goods directly to the customer. Basic customer details, such as customer name, date of birth, billing address and preferred shipping address, are stored in a relational database.  Orders, as they come in, are stored in a database. There is also a report of historical instances of fraud contained in a CSV spreadsheet. | |
| Objectives | In this guided demonstration, you will:   * Data Prep * Configure Python Executor step. * Build and Train a Forest Tree Model. * Deploy and Test the model. |

Step 1- Data Preparation

With the goal of preparing a dataset for ML, we can use PDI to combine these disparate data sources and engineer some features for learning from it. The following figure shows a transformation demonstrating an example of just that and includes some steps for deriving new fields. To begin with customer data is joined from several data sources, and then blended with transactional data and historical fraud occurrences contained in a CSV file.

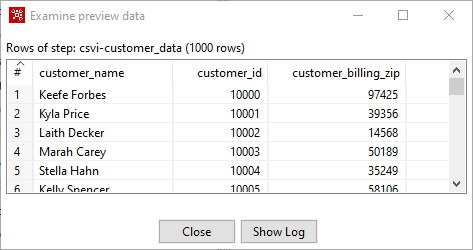
1. In Spoon open the following main Job:

C:\Machine--Learning\Lab 01 AutoML\tr\_autoML.ktr



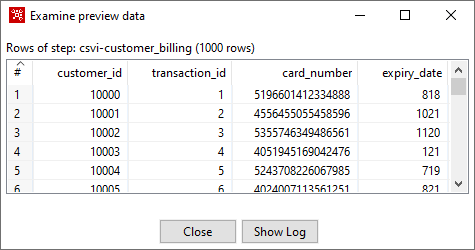
1. Browse the various customer data sources:

csvi-customer\_data



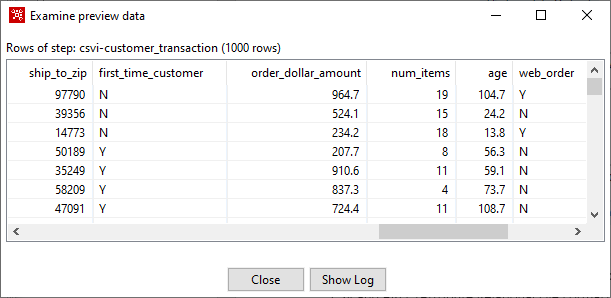
* Here you will find the customer\_billing\_zip codes, which will be used in feature engineering.

csvi-customer\_billing



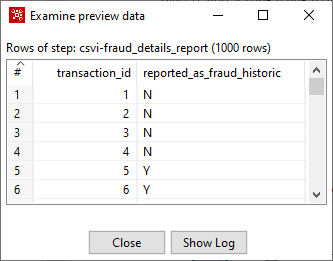
* References the customer transaction

csvi-customer\_transaction



* Customer transaction details
* Feature engineering for ship\_to\_zip
* The transaction details (x variables) are used by the decision trees to determine whether the transaction is fraudulent (y variable). The Boolean values will need to be changed into numbers for the randomForest algorithm.

csvi-fraud\_details\_report

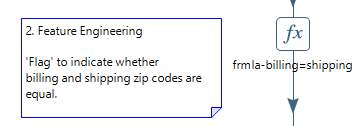


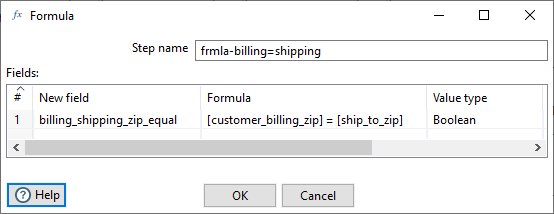
* Indicates whether historically the transaction was fraudulent.

Step 2 - Feature Engineering

The Feature Engineering is set to: billing zip code = shipping zip code

1. Open the step: frmla-billing=shipping





There are steps for deriving additional fields that might be useful for predictive modeling. These include computing the customer's age, extracting the hour of the day the order was placed, and setting a flag to indicate whether the shipping and billing addresses have the same zip code.

Step 3 - Test Machine Learning Models to Identify the Most Accurate Model

So, what does the data scientist do at this point?

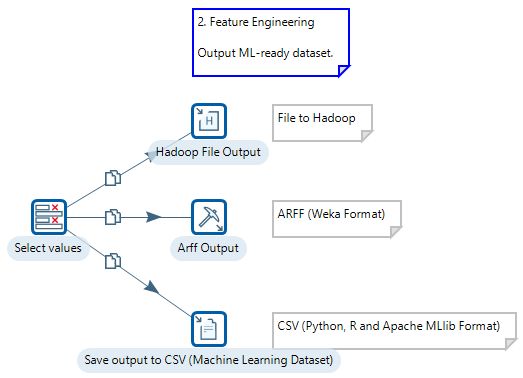
Typically, they will want to get a feel for the data by examining simple summary statistics and visualizations, followed by applying quick techniques for assessing the relationship between individual attributes (fields) and the target of interest which, in this example, is the reported\_as\_fraud\_historic" field.

Following that, if there are attributes that look promising, quick tests with common supervised classification algorithms will be next on the list. This comprises the initial stages of experimental data mining - i.e. the process of determining which predictive techniques are going to give the best result for a given problem.

TPOT: py-auto\_ml

A Tree-based Pipeline Optimization Tool for Automating Machine Learning (TPOT) is a Python Automated Machine Learning tool that optimizes machine learning pipelines using genetic programming. TPOT will automate the most tedious part of machine learning by intelligently exploring thousands of possible pipelines to find the best one for your data.

1. Open the step: py-auto\_ml



This process culminates with output of flattened (a Data Scientist’s preferred data shape) data in both CSV and ARFF (Attribute Relational File Format) data, the latter being the native file format used by PDM (Pentaho Data Mining, AKA WEKA). We end up with 100,000 examples (rows).

From this list, for the purposes of predictive modelling, we can drop the customer name, ID fields, email addresses, phone numbers and physical addresses. These fields are unlikely to be useful for learning purposes and, in fact, can be detrimental due to the large number of distinct values they contain.

# Lab 2: Credit Card Fraud - randomForest

|  |  |  |
| --- | --- | --- |
| Introduction | A Logistics Company wish to track telemetrics from their fleet of delivery trucks. This will help with predictive maintenance and optimize delivery routes. | |
| Objectives | In this guided demonstration, the instructor will guide you through:   * Create a Topic: SensorData * Publish / Subscribe messages * Configure AMQP Producer step * Configure AMQP Consumer step * Configure Get records from stream step |

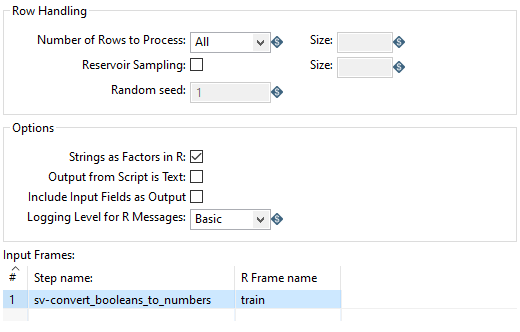
The model that will be used: **Random Forest**

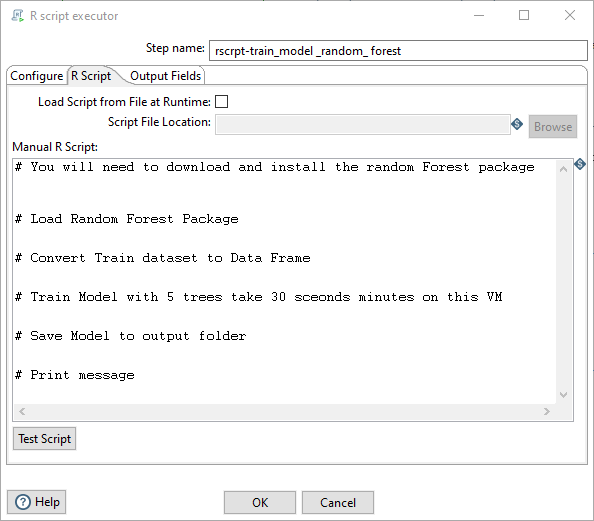
To configure the model for ‘training’:

1. Double-click on the ‘Train Model using Random Forest’ step to bring up the configuration settings.
2. Under the Configure tab, ensure the Input Frames points to the Step name:

sv-convert\_booleans\_to\_numbers

And that the R Frame name is: train

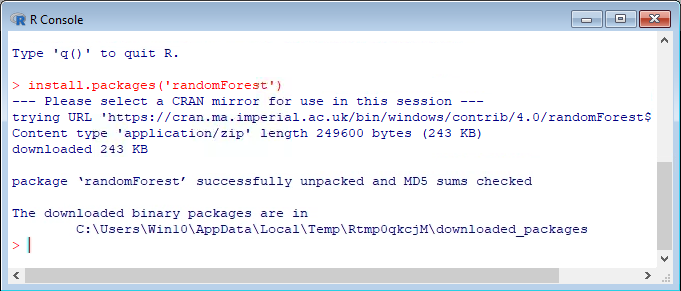


1. Select the R script tab. Copy and Paste the code snippets based on the Comments. 
2. The required script is located:

C:\Machine--Learning\Lab Credit Card Fraud\scripts\train\_model.txt

Install randomForest library

1. Click on: R Console icon
2. Enter the following command: install.packages(‘randomForest’)



Load Random

*The complete solution can be found at:*

C:\Streaming--Data\03\_JMS\Lab Apache Artemis\solution