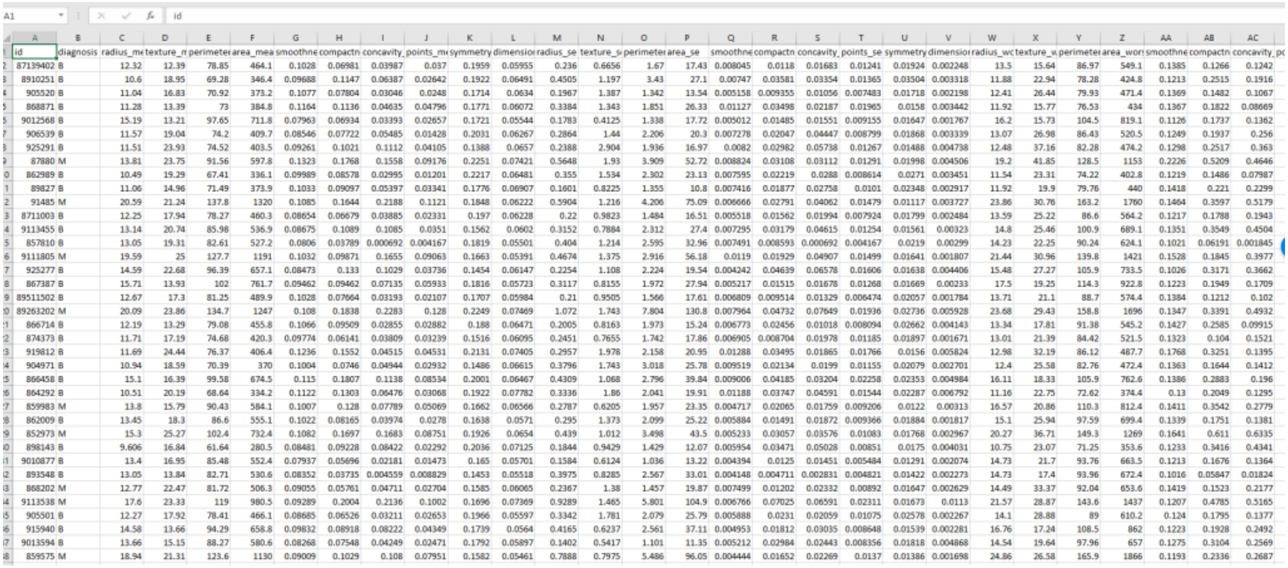
Prediction via KNN (K Nearest Neighbours) R codes: Part 2

Posted on *March 23, 2017*

In the previous post (Part 1), I have explained the concepts of KNN and how it works. In this post, I will explain how to use KNN for predict whether a patient with Cancer will be Benign or Malignant. This example is get from Brett book[1]. Imagine that we have a dataset on laboratory results of some patients that some of them already Benign or Malignant. See below picture.

the first column is patient ID, the second one is the diagnosis for each patient: B stand for Benign and M stand for Malignant. the other columns are the laboratory results (I am not good on understanding them!)

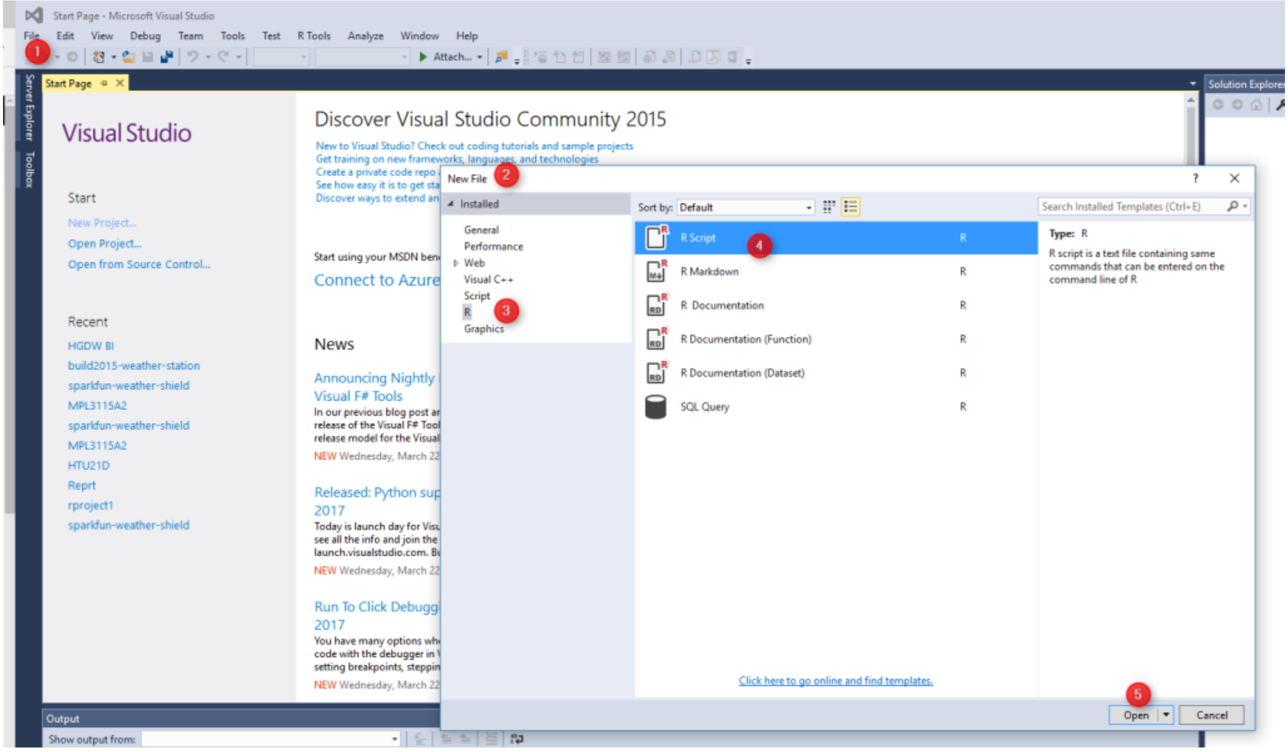


A1 id

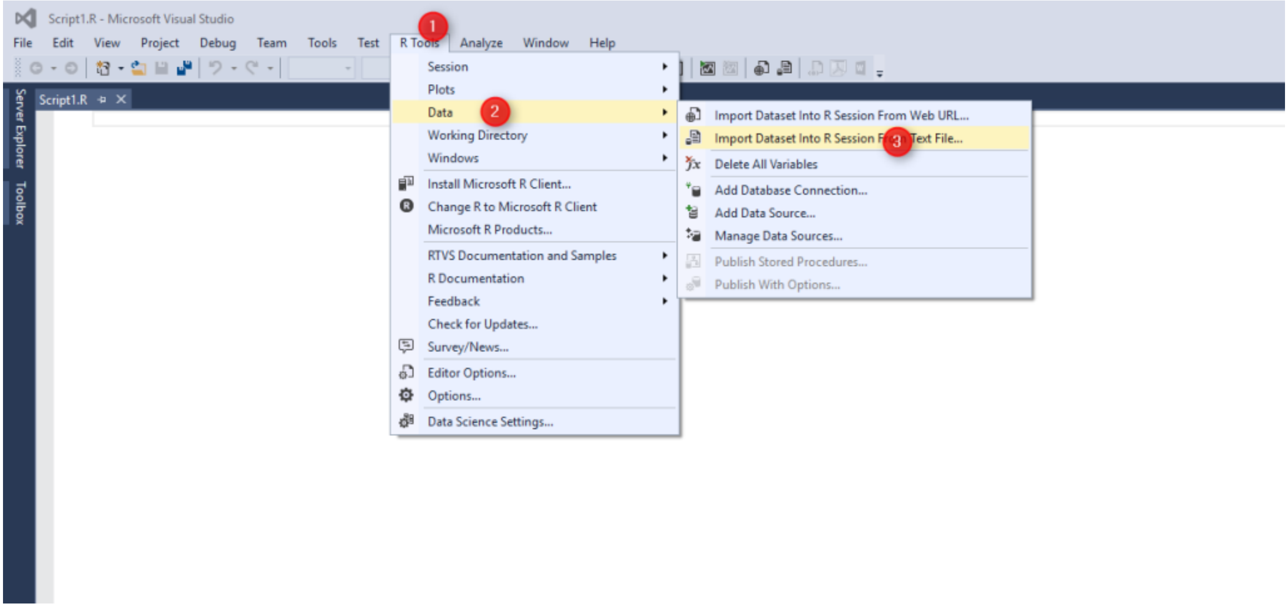
	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	AA	AB	AC
1	id	diagnosis	radius_me	texture_m	perimeter	area_mea	smoothne	compactn	concavity_	points_me	symmetry	dimensior	radius_se	texture_s	perimeter	area_se	smoothne	compactn	concavity_	points_se	symmetry	dimensior	radius_wc	texture_w	perimeter	area_wor	smoothne	compactn	concavity_
2	87139402	B	12.32	12.39	78.85	464.1	0.1028	0.06981	0.03987	0.037	0.1959	0.05955	0.236	0.6656	1.67	17.43	0.008045	0.0118	0.01683	0.01241	0.01924	0.002248	13.5	15.64	86.97	549.1	0.1385	0.1266	0.1242
3	8910251	B	10.6	18.95	69.28	346.4	0.09688	0.1147	0.06387	0.02642	0.1922	0.06491	0.4505	1.197	3.43	27.1	0.00747	0.03581	0.03354	0.01365	0.03504	0.003318	11.88	22.94	78.28	424.8	0.1213	0.2515	0.1916
4	905520	B	11.04	16.83	70.92	373.2	0.1077	0.07804	0.03046	0.0248	0.1714	0.0634	0.1967	1.387	1.342	13.54	0.005158	0.009355	0.01056	0.007483	0.01718	0.002198	12.41	26.44	79.93	471.4	0.1369	0.1482	0.1067
5	868871	B	11.28	13.39	73	384.8	0.1164	0.1136	0.04635	0.04796	0.1771	0.06072	0.3384	1.343	1.851	26.33	0.01127	0.03498	0.02187	0.01965	0.0158	0.003442	11.92	15.77	76.53	434	0.1367	0.1822	0.08669
6	9012568	B	15.19	13.21	97.65	711.8	0.07963	0.06934	0.03393	0.02657	0.1721	0.05544	0.1783	0.4125	1.338	17.72	0.005012	0.01485	0.01551	0.009155	0.01647	0.001767	16.2	15.73	104.5	819.1	0.1126	0.1737	0.1362
7	906539	B	11.57	19.04	74.2	409.7	0.08546	0.07722	0.05485	0.01428	0.2031	0.06267	0.2864	1.44	2.206	20.3	0.007278	0.02047	0.04447	0.008799	0.01868	0.003339	13.07	26.98	86.43	520.5	0.1249	0.1937	0.256
8	925291	B	11.51	23.93	74.52	403.5	0.09261	0.1021	0.1112	0.04105	0.1388	0.0657	0.2388	2.904	1.936	16.97	0.0082	0.02982	0.05738	0.01267	0.01488	0.004738	12.48	37.16	82.28	474.2	0.1298	0.2517	0.363
9	87880	M	13.81	23.75	91.56	597.8	0.1323	0.1768	0.1558	0.09176	0.2251	0.07421	0.5648	1.93	3.909	52.72	0.008824	0.03108	0.03112	0.01291	0.01998	0.004506	19.2	41.85	128.5	1153	0.2226	0.5209	0.4646
10	862989	B	10.49	19.29	67.41	336.1	0.09989	0.08578	0.02995	0.01201	0.2217	0.06481	0.355	1.534	2.302	23.13	0.007595	0.02219	0.0288	0.008614	0.0271	0.003451	11.54	23.31	74.22	402.8	0.1219	0.1486	0.07987
11	89827	B	11.06	14.96	71.49	373.9	0.1033	0.09097	0.05397	0.03341	0.1776	0.06907	0.1601	0.8225	1.355	10.8	0.007416	0.01877	0.02758	0.0101	0.02348	0.002917	11.92	19.9	79.76	440	0.1418	0.221	0.2299
12	91485	M	20.59	21.24	137.8	1320	0.1085	0.1644	0.2188	0.1121	0.1848	0.06222	0.5904	1.216	4.206	75.09	0.006666	0.02791	0.04062	0.01479	0.01117	0.003727	23.86	30.76	163.2	1760	0.1464	0.3597	0.5179
13	8711003	B	12.25	17.94	78.27	460.3	0.08654	0.06679	0.03885	0.02331	0.197	0.06228	0.22	0.9823	1.484	16.51	0.005518	0.01562	0.01994	0.007924	0.01799	0.002484	13.59	25.22	86.6	564.2	0.1217	0.1788	0.1943
14	9113455	B	13.14	20.74	85.98	536.9	0.08675	0.1089	0.1085	0.0351	0.1562	0.0602	0.3152	0.7884	2.312	27.4	0.007295	0.03179	0.04615	0.01254	0.01561	0.00323	14.8	25.46	100.9	689.1	0.1351	0.3549	0.4504
15	857810	B	13.05	19.31	82.61	527.2	0.0806	0.03789	0.000692	0.004167	0.1819	0.05501	0.404	1.214	2.595	32.96	0.007491	0.008593	0.000692	0.004167	0.0219	0.00299	14.23	22.25	90.24	624.1	0.1021	0.06191	0.001845
16	9111805	M	19.59	25	127.7	1191	0.1032	0.09871	0.1655	0.09063	0.1663	0.05391	0.4674	1.375	2.916	56.18	0.0119	0.01929	0.04907	0.01499	0.01641	0.001807	21.44	30.96	139.8	1421	0.1528	0.1845	0.3977
17	925277	B	14.59	22.68	96.39	657.1	0.08473	0.133	0.1029	0.03736	0.1454	0.06147	0.2254	1.108	2.224	19.54	0.004242	0.04639	0.06578	0.01606	0.01638	0.004406	15.48	27.27	105.9	733.5	0.1026	0.3171	0.3662
18	867387	B	15.71	13.93	102	761.7	0.09462	0.09462	0.07135	0.05933	0.1816	0.05723	0.3117	0.8155	1.972	27.94	0.005217	0.01515	0.01678	0.01268	0.01669	0.00233	17.5	19.25	114.3	922.8	0.1223	0.1949	0.1709
19	89511502	B	12.67	17.3	81.25	489.9	0.1028	0.07664	0.03193	0.02107	0.1707	0.05984	0.21	0.9505	1.566	17.61	0.006809	0.009514	0.01329	0.006474	0.02057	0.001784	13.71	21.1	88.7	574.4	0.1384	0.1212	0.102
20	89263202	M	20.09	23.86	134.7	1247	0.108	0.1838	0.2283	0.128	0.2249	0.07469	1.072	1.743	7.804	130.8	0.007964	0.04732	0.07649	0.01936	0.02736	0.005928	23.68	29.43	158.8	1696	0.1347	0.3391	0.4932
21	866714	B	12.19	13.29	79.08	455.8	0.1066	0.09509	0.02855	0.02882	0.188	0.06471	0.2005	0.8163	1.973	15.24	0.006773	0.02456	0.01018	0.008094	0.02662	0.004143	13.34	17.81	91.38	545.2	0.1427	0.2585	0.09915
22	874373	B	11.71	17.19	74.68	420.3	0.09774	0.06141	0.03809	0.03239	0.1516	0.06095	0.2451	0.7655	1.742	17.86	0.006905	0.008704	0.01978	0.01185	0.01897	0.001671	13.01	21.39	84.42	521.5	0.1323	0.104	0.1521
23	919812	B	11.69	24.44	76.37	406.4	0.1236	0.1552	0.04515	0.04531	0.2131	0.07405	0.2957	1.978	2.158	20.95	0.01288	0.03495	0.01865	0.01766	0.0156	0.005824	12.98	32.19	86.12	487.7	0.1768	0.3251	0.1395
24	904971	B	10.94	18.59	70.39	370	0.1004	0.0746	0.04944	0.02932	0.1486	0.06615	0.3796	1.743	3.018	25.78	0.009519	0.02134	0.0199	0.01155	0.02079	0.002701	12.4	25.58	82.76	472.4	0.1363	0.1644	0.1412
25	866458	B	15.1	16.39	99.58	674.5	0.115	0.1807	0.1138	0.08534	0.2001	0.06467	0.4309	1.068	2.796	39.84	0.009006	0.04185	0.03204	0.02258	0.02353	0.004984	16.11	18.33	105.9	762.6	0.1386	0.2883	0.196
26	864292	B	10.51	20.19	68.64	334.2	0.1122	0.1303	0.06476	0.03068	0.1922	0.07782	0.3336	1.86	2.041	19.91	0.01188	0.03747	0.04591	0.01544	0.02287	0.006792	11.16	22.75	72.62	374.4	0.13	0.2049	0.1295
27	859983	M	13.8	15.79	90.43	584.1	0.1007	0.128	0.07789	0.05069	0.1662	0.06566	0.2787	0.6205	1.957	23.35	0.004717	0.02065	0.01759	0.009206	0.0122	0.00313	16.57	20.86	110.3	812.4	0.1411	0.3542	0.2779
28	862009	B	13.45	18.3	86.6	555.1	0.1022	0.08165	0.03974	0.0278	0.1638	0.0571	0.295	1.373	2.099	25.22	0.005884	0.01491	0.01872	0.009366	0.01884	0.001817	15.1	25.94	97.59	699.4	0.1339	0.1751	0.1381
29	852973	M	15.3	25.27	102.4	732.4	0.1082	0.1697	0.1683	0.08751	0.1926	0.0654	0.439	1.012	3.498	43.5	0.005233	0.03057	0.03576	0.01083	0.01768	0.002967	20.27	36.71	149.3	1269	0.1641	0.611	0.6335
30	898143	B	9.606	16.84	61.64	280.5	0.08481	0.09228	0.08422	0.02292	0.2036	0.07125	0.1844	0.9429	1.429	12.07	0.005954	0.03471	0.05028	0.00851	0.0175	0.004031	10.75	23.07	71.25	353.6	0.1233	0.3416	0.4341
31	9010877	B	13.4	16.95	85.48	552.4	0.07937	0.05696	0.02181	0.01473	0.165	0.05701	0.1584	0.6124	1.036	13.22	0.004394	0.0125	0.01451	0.005484	0.01291	0.002074	14.73	21.7	93.76	663.5	0.1213	0.1676	0.1364
32	893548	B	13.05	13.84	82.71	530.6	0.08352	0.03735	0.004559	0.008829	0.1453	0.05518	0.3975	0.8285	2.567	33.01	0.004148	0.004711	0.002831	0.004821	0.01422	0.002273	14.73	17.4	93.96	672.4	0.1016	0.05847	0.01824
33	868202	M	12.77	22.47	81.72	506.3	0.09055	0.05761	0.04711	0.02704	0.1585	0.06065	0.2367	1.38	1.457	19.87	0.007499	0.01202	0.02332	0.00892	0.01647	0.002629	14.49	33.37	92.04	653.6	0.1419	0.1523	0.2177
34	9113538	M	17.6	23.33	119	980.5	0.09289	0.2004	0.2136	0.1002	0.1696	0.07369	0.9289	1.465	5.801	104.9	0.006766	0.07025	0.06591	0.02311	0.01673	0.0113	21.57	28.87	143.6	1437	0.1207	0.4785	0.5165
35	905501	B	12.27	17.92	78.41	466.1	0.08685	0.06526	0.03211	0.02653	0.1966	0.05597	0.3342	1.781	2.079	25.79	0.005888	0.0231	0.02059	0.01075	0.02578	0.002267	14.1	28.88	89	610.2	0.124	0.1795	0.1377
36	915940	B	14.58	13.66	94.29	658.8	0.09832	0.08918	0.08222	0.04349	0.1739	0.0564	0.4165	0.6237	2.561	37.11	0.004953	0.01812	0.03035	0.008648	0.01539	0.002281	16.76	17.24	108.5	862	0.1223	0.1928	0.2492
37	9013594	B	13.66	15.15	88.27	580.6	0.08268	0.07548	0.04249	0.02471	0.1792	0.05897	0.1402	0.5417	1.101	11.35	0.005212	0.02984	0.02443	0.008356	0.01818	0.004868	14.54	19.64	97.96	657	0.1275	0.3104	0.2569
38	859575	M	18.94	21.31	123.6	1130	0.09009	0.1029	0.108	0.07951	0.1582	0.05461	0.7888	0.7975	5.486	96.05	0.004444	0.01652	0.02269	0.0137	0.01386	0.001698	24.86	26.58	165.9	1866	0.1193	0.2336	0.2687

We want to create a prediction models for a new patient with sp ory resu predict whether this patient will be Benign or Malignant.

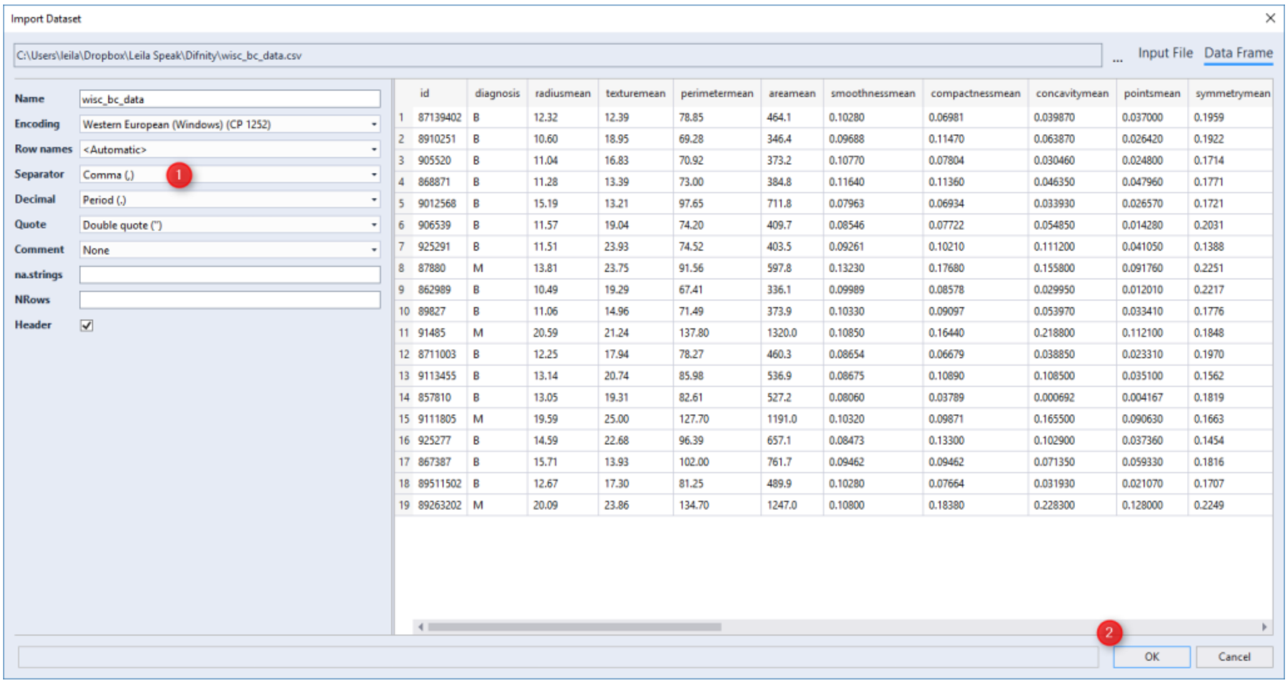
For this demo, I will use R environment in Visual Studio. Hence, after opening Visual Studio 2015, select File, New file and then under the General tab find “R”. I am going to write R codes in R scripts (Number 4) and then create a R scripts there.


Start Page - Microsoft Visual Studio
File Edit View Debug Team Tools Test R Tools Analyze Window Help
1
Visual Studio
Discover Visual Studio Community 2015
Start
New Project...
Open Project...
Open from Source Control...
Recent
News
New File
2
Installed
General
Performance
Web
Visual C++
Script
R
3
Graphics
Sort by: Default
Search Installed Templates (Ctrl+E)
R Script
4
R Markdown
R Documentation
R Documentation (Function)
R Documentation (Dataset)
SQL Query
Type: R
R script is a text file containing same commands that can be entered on the command line of R
Click here to go online and find templates.
5
Open
Cancel
Output
Show output from:

After creating an empty R scripts. Now I am going to import data. choose “R Tools”, then in Data menu, then click on the “Import Dataset into R session”.


Script1.R - Microsoft Visual Studio
File Edit View Project Debug Team Tools Test R Tools Analyze Window Help
1
Script1.R
Session
Plots
Data
2
Working Directory
Windows
Install Microsoft R Client...
Change R to Microsoft R Client
Microsoft R Products...
RTVS Documentation and Samples
R Documentation
Feedback
Check for Updates...
Survey/News...
Editor Options...
Options...
Data Science Settings...
Import Dataset Into R Session From Web URL...
Import Dataset Into R Session From Text File...
3
Delete All Variables
Add Database Connection...
Add Data Source...
Manage Data Sources...
Publish Stored Procedures...
Publish With Options...

You will see below window. It shows all the columns and the sample of data. The SCV file that I am used for this post has been produced by [1]. It is a CSV file with delimiter (number 1) by Comma.



C:\Users\leila\Dropbox\Leila Speak\Difnity\wisc_bc_data.csv

Name: wisc_bc_data
Encoding: Western European (Windows) (CP 1252)
Row names: <Automatic>
Separator: Comma (,)
Decimal: Period (.)
Quote: Double quote (")
Comment: None
na.strings:
NRows:
Header: ☑

	id	diagnosis	radiusmean	texturemean	perimetermean	areamean	smoothnessmean	compactnessmean	concavitymean	pointsmean	symmetrymean
1	87139402	B	12.32	12.39	78.85	464.1	0.10280	0.06981	0.039870	0.037000	0.1959
2	8910251	B	10.60	18.95	69.28	346.4	0.09688	0.11470	0.063870	0.026420	0.1922
3	905520	B	11.04	16.83	70.92	373.2	0.10770	0.07804	0.030460	0.024800	0.1714
4	868871	B	11.28	13.39	73.00	384.8	0.11640	0.11360	0.046350	0.047960	0.1771
5	9012568	B	15.19	13.21	97.65	711.8	0.07963	0.06934	0.033930	0.026570	0.1721
6	906539	B	11.57	19.04	74.20	409.7	0.08546	0.07722	0.054850	0.014280	0.2031
7	925291	B	11.51	23.93	74.52	403.5	0.09261	0.10210	0.111200	0.041050	0.1388
8	87880	M	13.81	23.75	91.56	597.8	0.13230	0.17680	0.155800	0.091760	0.2251
9	862989	B	10.49	19.29	67.41	336.1	0.09989	0.08578	0.029950	0.012010	0.2217
10	89827	B	11.06	14.96	71.49	373.9	0.10330	0.09097	0.053970	0.033410	0.1776
11	91485	M	20.59	21.24	137.80	1320.0	0.10850	0.16440	0.218800	0.112100	0.1848
12	8711003	B	12.25	17.94	78.27	460.3	0.08654	0.06679	0.038850	0.023310	0.1970
13	9113455	B	13.14	20.74	85.98	536.9	0.08675	0.10890	0.108500	0.035100	0.1562
14	857810	B	13.05	19.31	82.61	527.2	0.08060	0.03789	0.000692	0.004167	0.1819
15	9111805	M	19.59	25.00	127.70	1191.0	0.10320	0.09871	0.165500	0.090630	0.1663
16	925277	B	14.59	22.68	96.39	657.1	0.08473	0.13300	0.102900	0.037360	0.1454
17	867387	B	15.71	13.93	102.00	761.7	0.09462	0.09462	0.071350	0.059330	0.1816
18	89511502	B	12.67	17.30	81.25	489.9	0.10280	0.07664	0.031930	0.021070	0.1707
19	89263202	M	20.09	23.86	134.70	1247.0	0.10800	0.18380	0.228300	0.128000	0.2249

After importing the dataset, now we are going to see the summary of data by Function “STR”. this function shows the summary of column’s data and the data type of each column.

```
str(wisc_bc_data)
```

the result will be:

```
data.frame':	569 obs. of  32 variables:
 $ id               : int  87139402 8910251 905520 868871 9012568 9065
 $ diagnosis        : Factor w/ 2 levels "B","M": 1 1 1 1 1 1 1 2 1 1
 $ radius_mean      : num  12.3 10.6 11 11.3 15.2 ...
 $ texture_mean     : num  12.4 18.9 16.8 13.4 13.2 ...
 $ perimeter_mean   : num  78.8 69.3 70.9 73 97.7 ...
 $ area_mean        : num  464 346 373 385 712 ...
 $ smoothness_mean  : num  0.1028 0.0969 0.1077 0.1164 0.0796 ...
 $ compactness_mean : num  0.0698 0.1147 0.078 0.1136 0.0693 ...
 $ concavity_mean   : num  0.0399 0.0639 0.0305 0.0464 0.0339 ...
 $ points_mean      : num  0.037 0.0264 0.0248 0.048 0.0266 ...
 $ symmetry_mean    : num  0.196 0.192 0.171 0.177 0.172 ...
 $ dimension_mean   : num  0.0595 0.0649 0.0634 0.0607 0.0554 ...
 $ radius_se        : num  0.236 0.451 0.197 0.338 0.178 ...
 $ texture_se       : num  0.666 1.197 1.387 1.343 0.412 ...
 $ perimeter_se     : num  1.67 3.43 1.34 1.85 1.34 ...
 $ area_se          : num  17.4 27.1 13.5 26.3 17.7 ...
 $ smoothness_se    : num  0.00805 0.00747 0.00516 0.01127 0.00501 ...
 $ compactness_se   : num  0.0118 0.03581 0.00936 0.03498 0.01485 ...
 $ concavity_se     : num  0.0168 0.0335 0.0106 0.0219 0.0155 ...
 $ points_se        : num  0.01241 0.01365 0.00748 0.01965 0.00915 ...
 $ symmetry_se      : num  0.0192 0.035 0.0172 0.0158 0.0165 ...
 $ dimension_se     : num  0.00225 0.00332 0.0022 0.00344 0.00177 ...
 $ radius_worst     : num  13.5 11.9 12.4 11.9 16.2 ...
 $ texture_worst    : num  15.6 22.9 26.4 15.8 15.7 ...
 $ perimeter_worst  : num  87 78.3 79.9 76.5 104.5 ...
 $ area_worst       : num  549 425 471 434 819 ...
 $ smoothness_worst : num  0.139 0.121 0.137 0.137 0.113 ...
 $ compactness_worst: num  0.127 0.252 0.148 0.182 0.174 ...
 $ concavity_worst  : num  0.1242 0.1916 0.1067 0.0867 0.1362 ...
 $ points_worst     : num  0.0939 0.0793 0.0743 0.0861 0.0818 ...
 $ symmetry_worst   : num  0.283 0.294 0.3 0.21 0.249 ...
 $ dimension_worst  : num  0.0677 0.0759 0.0788 0.0678 0.0677 ...
>
```

Now we want to keep the original dataset, so we put data in a temp variable “wbcd”

```
wbcd <- wisc_bc_data
```

The first column of data "id" could not be that much important in prediction, so we eliminate the first column from dataset.

```
wbcd<-wbcd[-1]
```

We want to look at the statistical summary of each column: such as min, max, mid, mean value of each columns.

```
summary(wbcd)
```

The result of running the code will be as below, as you can see for first column (we already delete the id column), we have 357 cases that are Benign and 212 Malignant cases. also for all other laboratory measurement we can see the min, max, median, mean. 1st Qu, and 3rd Qu.

```
diagnosis  radius_mean      texture_mean   perimeter_mean     area_mean
 B:357     Min.   : 6.981   Min.   : 9.71   Min.   : 43.79   Min.   : 1
 M:212     1st Qu.:11.700   1st Qu.:16.17   1st Qu.: 75.17   1st Qu.: 4
           Median :13.370   Median :18.84   Median : 86.24   Median : 5
           Mean   :14.127   Mean   :19.29   Mean   : 91.97   Mean   : 6
           3rd Qu.:15.780   3rd Qu.:21.80   3rd Qu.:104.10   3rd Qu.: 7
           Max.   :28.110   Max.   :39.28   Max.   :188.50   Max.   :25
   radius_se        texture_se      perimeter_se       area_se        s
 Min.   :0.1115   Min.   :0.3602   Min.   : 0.757   Min.   :  6.802   M
 1st Qu.:0.2324   1st Qu.:0.8339   1st Qu.: 1.606   1st Qu.: 17.850   1
 Median :0.3242   Median :1.1080   Median : 2.287   Median : 24.530   M
 Mean   :0.4052   Mean   :1.2169   Mean   : 2.866   Mean   : 40.337   M
 3rd Qu.:0.4789   3rd Qu.:1.4740   3rd Qu.: 3.357   3rd Qu.: 45.190   3
 Max.   :2.8730   Max.   :4.8850   Max.   :21.980   Max.   :542.200   M
  radius_worst   texture_worst   perimeter_worst    area_worst     smoo
 Min.   : 7.93   Min.   :12.02   Min.   : 50.41   Min.   : 185.2   Min
 1st Qu.:13.01   1st Qu.:21.08   1st Qu.: 84.11   1st Qu.: 515.3   1st
 Median :14.97   Median :25.41   Median : 97.66   Median : 686.5   Medi
 Mean   :16.27   Mean   :25.68   Mean   :107.26   Mean   : 880.6   Mean
 3rd Qu.:18.79   3rd Qu.:29.72   3rd Qu.:125.40   3rd Qu.:1084.0   3rd
 Max.   :36.04   Max.   :49.54   Max.   :251.20   Max.   :4254.0   Max
>
```

Data Wrangling

first of all, we want to have a dataset that is easy to read. the first data cleaning is about replacing the "B" value with Benign and "M" value with Malignant in diagnosis column. this replacement makes the data to be more informative. Hence we employ below code:

```
wbcd$diagnosis<- factor(wbcd$diagnosis, levels = c("B", "M"), labels =
```

Factor is a function that gets the column name in a dataset, and we can identify the labels with out consuming memories)

there is another issue in data. the numbers are not normalized!

what is data normalization : that mean they are not in a same scale. for instance for radius mean all numbers between 6 to 29 while for column smoothness_mean is between 0.05 to 0.17. for performing the predict analysis using KNN, as we use distance calculation (Part 1), it is important all numbers should be in same range[1].

normalization can be done by below formula

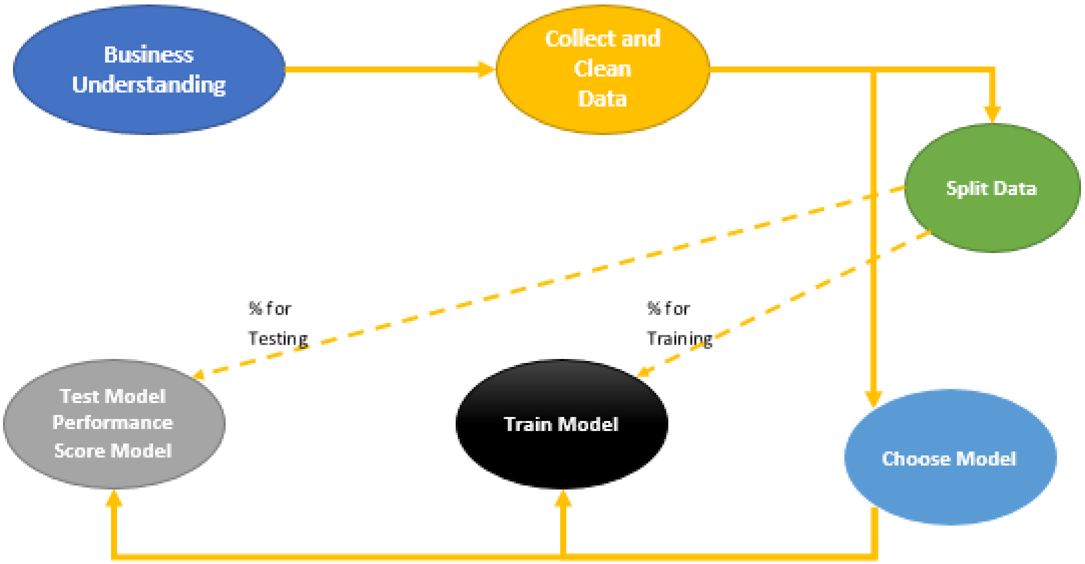
```
normalize <- function(x) {
  return ((x - min(x)) / (max(x) - min(x))) }
```

now we are going to apply this function in all numeric columns in wbcd dataset. There is a function in R that apply a function over a dataset:

```
wbcd_n <- as.data.frame(lapply(wbcd[2:31], normalize))
```

"lapply" gets the dataset and function name, then apply the function on all dataset. in this example because the first column is text (diagnosis), we apply "normalize" function on columns 2 to 31.Now our data is ready for creating a KNN model.

from machine learning process we need a dataset for training model and another for testing model (from Market basket analysis post)



Hence, we should have two different dataset for train and test. in this example, we going to have row number 1 to 469 for training and creating model and from row number 470 to 569 for testing the model.

```
wbcd_train <- wbcd_n[1:469, ]
 wbcd_test <- wbcd_n[470:569, ]
```

so wbcd_train we have 469 rows of data and the rest in wbcd_test. also we need the prediction label for result

```
wbcd_train_labels <- wbcd[1:469, 1]
wbcd_test_labels <- wbcd[470:569, 1]
```

So data is ready, now we are going to train model and create KNN algorithm.

For using KNN there is a need to install package “Class”

```
install.packages("class")
```

Now we able to call function KNN to predict the patient diagnosis. KNN function accept the training dataset and test dataset as second arguments. moreover the prediction label also need for result. we want to use KNN based on the discussion on Part 1, to identify the number K (K nearest Neighbour), we should calculate the square root of observation. here for 469 observation the K is 21.

```
wbcd_test_pred <- knn(train = wbcd_train, test = wbcd_test,cl= wbcd_tra
```

the result is “wbcd_test_pred” holds the result of the KNN prediction.

```
[1] Benign    Benign    Benign    Benign    Malignant Benign    Maligna
 [19] Malignant Benign    Malignant Malignant Malignant Malignant Beni
 [37] Benign    Benign    Benign    Malignant Malignant Benign    Mali
 [55] Benign    Benign    Benign    Benign    Malignant Benign    Beni
 [73] Benign    Benign    Malignant Malignant Benign    Benign    Beni
 [91] Benign    Malignant Benign    Benign    Benign    Benign    Beni
Levels: Benign Malignant
```

we want to evaluate the result of the model by installing “gmodels” a packages that shows the evaluation performance.

```
install.packages("gmodels")
require("gmodels")
library("gmodels")
```

we employ a function name “CrossTable”. it gets label as first input, the prediction result as second argument.

```
CrossTable(x = wbcd_test_labels, y = wbcd_test_pred,
prop.chisq = FALSE)
```

The result of “Cross table” will be as below. we have 100 observation. the tables show the result of evaluation and see how much the KNN prediction is accurate. the first row and first column shows the true positive (**TP**) cases, means the cases that already Benign and KNN predicts Benign. The first row and second column shows number of cases that already Benign and KNN predict they are Malignant (**TN**). The second row and first column is Malignant in real world but KNN predict they are Benign (**FP**). finally the last column and last row is False Negative (**FN**) that means cases that they Malignant and KNN predict as Malignant.

```
Total Observations in Table:  100


                 | wbcd_test_pred 
wbcd_test_labels |    Benign | Malignant | Row Total | 
-----------------|-----------|-----------|-----------|
          Benign |        61 |         0 |        61 | 
                 |     1.000 |     0.000 |     0.610 | 
                 |     0.968 |     0.000 |           | 
                 |     0.610 |     0.000 |           | 
-----------------|-----------|-----------|-----------|
       Malignant |         2 |        37 |        39 | 
                 |     0.051 |     0.949 |     0.390 | 
                 |     0.032 |     1.000 |           | 
                 |     0.020 |     0.370 |           | 
-----------------|-----------|-----------|-----------|
    Column Total |        63 |        37 |       100 | 
                 |     0.630 |     0.370 |           | 
-----------------|-----------|-----------|-----------|
```

so as much as TP and FN is higher the prediction is better. in our example TP is 61 and FN is 37, moreover the TN and TP is just 0 and 2 which is good.

to calculate the accuracy we should follow the below formula:

accuracy <- (tp + tn) / (tp + fn + fp + tn)

Accuracy will be (61+37)/(61+37+2+0)=98%

In the next post I will explained how to perform KNN in Power BI (data wrangling, modelling and visualization).

[1].Machine Learning with R,Brett Lantz, Packt Publishing,2015.

Leila Etaati

Trainer, Consultant, Mentor
Leila is the first Microsoft AI MVP in New Zealand and Australia, She has Ph.D. in Information System from the University Of Auckland. She is the Co-director and data scientist in RADACAD Company with more than 100 clients in around the world. She is the co-organizer of Microsoft Business Intelligence and Power BI Use group (meetup) in Auckland with more than 1200 members, She is the co-organizer of three main conferences in Auckland: SQL Saturday Auckland (2015 till now) with more than 400 registrations, Difinity (2017 till now) with more than 200 registrations and Global AI Bootcamp 2018. She is a Data Scientist, BI Consultant, Trainer, and Speaker. She is a well-known International Speakers to many conferences such as Microsoft ignite, SQL pass, Data Platform Summit, SQL Saturday, Power BI world Tour and so forth in Europe, USA, Asia, Australia, and New Zealand. She has over ten years' experience working with databases and software systems. She was involved in many large-scale projects for big-sized companies. She also AI and Data Platform Microsoft MVP. Leila is an active Technical Microsoft AI blogger for RADACAD.

This entry was posted in *Analytics*, *Azure Machine Learning*, *Business Intelligence*, *Data Mining*, *Power BI*, *R* and tagged *Advance Analytics*, *AzureML*, *KNN*, *Machine Learning*, *Predictive Analysis*, *R*. Bookmark the *permalink*.

← Prediction via KNN (K Nearest Neighbours) Concepts: Part 1

Prediction via KNN (K Nearest Neighbours) KNN Power BI: Part 3 →

Leave a Reply

Enter your comment here...

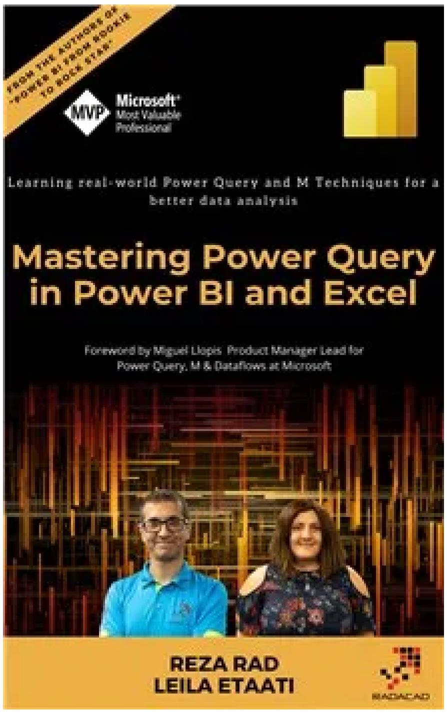
POWER BI SUMMIT

Power BI Summit
Virtual Conference 7th-11 March 2022
More than 100 Sessions!
Super Early Bird Special $79.99
Speakers:
Power BI Product Managers
Regional Directors
MVPs and Experts

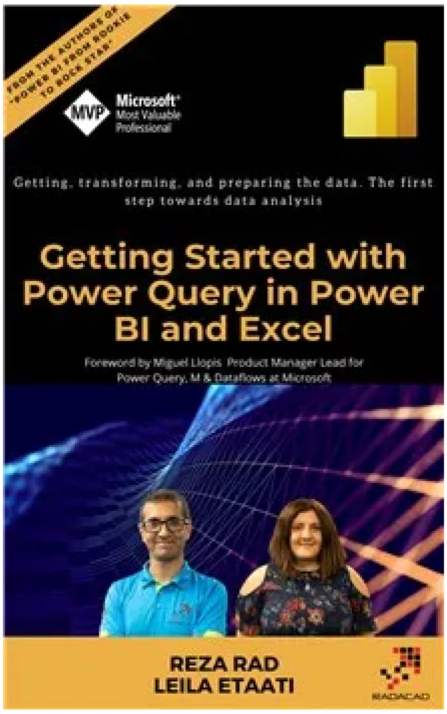
BOOK: POWER BI DAX SIMPLIFIED

FROM THE AUTHOR OF "POWER BI FROM ROOKIE TO ROCK STAR"
DAX and calculation language of Power BI demystified by practical examples
POWER BI DAX SIMPLIFIED
Measure = SUMX
(
Table,
Expression
)
ALL
(
Table,
Or column(s)
)
Microsoft
MVP
Microsoft Most Valuable Professional
2021
REZA RAD
RADACAD

BOOK: MASTERING POWER QUERY IN POWER BI AND EXCEL


FROM THE AUTHORS OF "POWER BI FROM ROOKIE TO ROCK STAR"
MVP Microsoft Most Valuable Professional
Learning real-world Power Query and M Techniques for a better data analysis
Mastering Power Query in Power BI and Excel
Foreword by Miguel Llopis Product Manager Lead for Power Query, M & Dataflows at Microsoft
REZA RAD
LEILA ETAATI
RADACAD

BOOK: GETTING STARTED WITH POWER QUERY IN POWER BI AND EXCEL


FROM THE AUTHORS OF "POWER BI FROM ROOKIE TO ROCK STAR"
MVP Microsoft Most Valuable Professional
Getting, transforming, and preparing the data. The first step towards data analysis
Getting Started with Power Query in Power BI and Excel
Foreword by Miguel Llopis Product Manager Lead for Power Query, M & Dataflows at Microsoft
REZA RAD
LEILA ETAATI
RADACAD

OUR UPCOMING LIVE COURSES

RADACAD
Power BI from Rookie to Rock Star
MVP Microsoft Most Valuable Professional

SEARCH IN ARTICLES

Search …

TRAINING ACADEMY

COACHING & CONSULTING

Free Books, And Article Updates

Enter your email below for our free books on Power BI and AI, the Power BI Helper tool, and getting updates about our latest articles.

Enter your email here...

Give me free stuff

FOLLOW RADACAD ON TWITTER

Follow @Radacad_com 1,410 followers

FOLLOW REZA ON TWITTER

Follow @Rad_Reza 7,883 followers

FOLLOW LEILA ON TWITTER

Follow @leila_etaati 5,115 followers

RECENT POSTS

- ConcatenateX in Power BI and DAX: Concatenate Values of a Column
- KPI Visual in Power BI Explained
- Power BI Pattern: Find Start and End Date of Task from Children
- Power BI Calendar Visual
- Search Bar in Power BI Slicer
- Power BI Date or Calendar Table Best Method: DAX or Power Query?
- Power Query Transformation Pattern: Convert Status Log Dates Table to Start and End Date Table in Power BI
- Power BI Modeling: Change the Status Log Table to a From-To Table
- Generate Year Month Day Combinations in Power BI Report using Power Query Date.ToText Function
- Add Leading Zeros to a Number in Power BI Using Power Query

CATEGORIES

- AI (57)
- AI Builder (14)
- Analytics (103)
- Analytics with Power BI and R (44)
- Anomaly Detection (2)
- Architecture (3)
- Automated Machine Learning (8)
- AutoML (11)
- Awards (6)

Azure (48)

Azure Data Analysis (14)

Azure Data Factory (14)

Azure Data Lake (9)

Azure Event Hub (2)

Azure Machine Learning (67)

Azure ML (34)

Azure ML workbench (17)

Azure SQL Database (3)

Azure Stream Analytics (2)

Barcode Scanner (1)

Big Data (6)

Business Intelligence (48)

Career (1)

Charticulator (3)

Chatbot (1)

Classification (2)

Cognitive Service (14)

Cognitive Services (16)

Cortana Analytics (24)

Data Mining (63)

Data Science (47)

Data Science Virtual Machine (6)

Databricks (3)

Dataflow (20)

DataScience (36)

DAX (101)

Decision Tree (8)

Decomposition Tree (2)

DecompositionTree (1)

Design (5)

Design Patterns (10)

DQS (2)

ETL (32)

Events (3)

Excel (7)

face recognition (3)

Form Processing (3)

HDInsight (3)

Image Processing (5)

IOT (9)

Key Influencers (3)

M (17)

Machine Learning Services (23)

MDS (12)

Microsoft Flow (10)

ML Services (12)

Modelling (34)

News (4)

Object Detection (1)

OCR (6)

pbiviz (2)

Performance Tuning (1)

Power App (12)

Power Automate (6)

Power BI (523)

Power BI from Rookie to Rockstar (373)

Power BI Helper (21)

Power BI Helper Guides (6)

Power BI REST API (7)

Power BI Showcase (1)

Power Query (157)

PowerApps (16)

Presentations (82)

Publications (11)

Python (13)

Q&A (3)

Q&A Visual (2)

R (60)

R Packages (20)

R scripts (23)

R services (15)

Ranking Chart (1)

Regression (1)

RVIZ (10)

Salary Survey (4)

Scripting (34)

Sharing (4)

Slope Chart (1)

SQL Server (50)

SQL Server 2016 (10)

SQL Server 2017 (2)

SQL Server 2019 (1)

SSAS (9)

SSDT (2)

SSIS (104)

SSIS Catalog (8)

SSIS Tutorial Videos (13)

SSRS (1)

T-SQL (16)

Text Analytics (1)

Text Classification (1)

Text Recognizer (1)

Time Series (5)

Training (2)

Uncategorized (67)

Visualization (88)

XML (2)

TAGS

Advanced Analytics (15) Advanced Power BI (49) AI (10) AI Builder (13) Azure (22) Azure Data Factory (16) Azure Machine Learning (42) Azure ML (12) Business Intelligence (18) Cognitive services (11) Conference (53) Dataflow (16) Data Mining (47) Data science (18) Data Visualization (29) Data Warehouse (15) Date Dimension (10) DAX (120) Design Patterns (11) ETL (26) Excel (10) Foreach Loop (11) M (13) Machine Learning (81) MDS (15) Microsoft Flow (12) Modeling (32) Power Apps (21) Power BI (510) Power BI Desktop (42) Power BI from Rookie to Rock Star (341) Power BI from Rookie to Rockstar (27) Power BI Helper (24) Power Query (155) R (30) Row Level Security (18) R scripts (11) Sharing (11) SQL Server (48) SSAS (14) SSIS (107) T-SQL (14) TechEd (11) Tutorial Videos (12) Visualization (71)

training@radacad.com

USA: +1 206 905 9306
New Zealand: +64 9 889 1767
a: +61 2

USA | Australia | New Zealand