01Data_Import_v1

September 21, 2022

505-01-FA22

Team 1

Final Project

Import libraries

[1]:
```
import pandas as pd
import numpy as np
import matplotlib.pylab as plt
from sklearn.preprocessing import StandardScaler, OneHotEncoder
from sklearn.model_selection import train_test_split, GridSearchCV
from sklearn.metrics import accuracy_score, r2_score
from sklearn.neighbors import NearestNeighbors, KNeighborsClassifier,␣
 ↪KNeighborsRegressor
from sklearn.linear_model import LogisticRegression, LogisticRegressionCV,␣
 ↪LinearRegression
from sklearn.tree import DecisionTreeRegressor
import dmba
from dmba import classificationSummary, regressionSummary, gainsChart,␣
 ↪liftChart, backward_elimination, stepwise_selection
from dmba.metric import AIC_score, adjusted_r2_score

%matplotlib inline
```

Load data and display characteristics

Air Quality data set

https://archive.ics.uci.edu/ml/datasets/Air+Quality

[2]:
```
air_df01 = pd.read_csv(r'C:
 ↪\Users\acarr\Documents\GitHub\ads505_business_proj\data\Potential Data␣
 ↪Sets\Air Quality\AirQualityUCI.csv', sep=';')
air_df01 = air_df01.drop(['Unnamed: 15', 'Unnamed: 16'], axis=1)
display(air_df01.shape) # Display dimensions
display(air_df01.describe()) # Display simple descriptive stats
display(air_df01.head()) # Display first 5 rows
```

```
display(air_df01.info())
```

```
(9471, 15)
```

```
       PT08.S1(CO)     NMHC(GT)  PT08.S2(NMHC)      NOx(GT)  PT08.S3(NOx)  \
count  9357.000000  9357.000000    9357.000000  9357.000000   9357.000000
mean   1048.990061  -159.090093     894.595276   168.616971    794.990168
std     329.832710   139.789093     342.333252   257.433866    321.993552
min    -200.000000  -200.000000    -200.000000  -200.000000   -200.000000
25%     921.000000  -200.000000     711.000000    50.000000    637.000000
50%    1053.000000  -200.000000     895.000000   141.000000    794.000000
75%    1221.000000  -200.000000    1105.000000   284.000000    960.000000
max    2040.000000  1189.000000    2214.000000  1479.000000   2683.000000

           NO2(GT)  PT08.S4(NO2)  PT08.S5(O3)
count  9357.000000   9357.000000  9357.000000
mean     58.148873   1391.479641   975.072032
std     126.940455    467.210125   456.938184
min    -200.000000   -200.000000  -200.000000
25%      53.000000   1185.000000   700.000000
50%      96.000000   1446.000000   942.000000
75%     133.000000   1662.000000  1255.000000
max     340.000000   2775.000000  2523.000000
```

	Date	Time	CO(GT)	PT08.S1(CO)	NMHC(GT)	C6H6(GT)	PT08.S2(NMHC)	\
0	10/03/2004	18.00.00	2,6	1360.0	150.0	11,9	1046.0	
1	10/03/2004	19.00.00	2	1292.0	112.0	9,4	955.0	
2	10/03/2004	20.00.00	2,2	1402.0	88.0	9,0	939.0	
3	10/03/2004	21.00.00	2,2	1376.0	80.0	9,2	948.0	
4	10/03/2004	22.00.00	1,6	1272.0	51.0	6,5	836.0	

	NOx(GT)	PT08.S3(NOx)	NO2(GT)	PT08.S4(NO2)	PT08.S5(O3)	T	RH	\
0	166.0	1056.0	113.0	1692.0	1268.0	13,6	48,9	
1	103.0	1174.0	92.0	1559.0	972.0	13,3	47,7	
2	131.0	1140.0	114.0	1555.0	1074.0	11,9	54,0	
3	172.0	1092.0	122.0	1584.0	1203.0	11,0	60,0	
4	131.0	1205.0	116.0	1490.0	1110.0	11,2	59,6	

	AH
0	0,7578
1	0,7255
2	0,7502
3	0,7867
4	0,7888

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 9471 entries, 0 to 9470
Data columns (total 15 columns):
 #   Column         Non-Null Count  Dtype
```

```
---  ------          --------------  -----
 0   Date            9357 non-null   object
 1   Time            9357 non-null   object
 2   CO(GT)          9357 non-null   object
 3   PT08.S1(CO)     9357 non-null   float64
 4   NMHC(GT)        9357 non-null   float64
 5   C6H6(GT)        9357 non-null   object
 6   PT08.S2(NMHC)   9357 non-null   float64
 7   NOx(GT)         9357 non-null   float64
 8   PT08.S3(NOx)    9357 non-null   float64
 9   NO2(GT)         9357 non-null   float64
 10  PT08.S4(NO2)    9357 non-null   float64
 11  PT08.S5(O3)     9357 non-null   float64
 12  T               9357 non-null   object
 13  RH              9357 non-null   object
 14  AH              9357 non-null   object
dtypes: float64(8), object(7)
memory usage: 1.1+ MB

None
```

Add color to Jupyter NB Markdown (Santopay, 2019)

Discussion:
* There are no null values. Some of the columns use the European convention of "," to indicate decimal place, so those would need to be converted.
* This isn't strictly a "business" data set, but we could prob craft a convincing story.
* We would also need to determine which feature is our target. It is not clear which is the intended dependent variable.

E-shop clothing data set

https://archive.ics.uci.edu/ml/datasets/clickstream+data+for+online+shopping

[3]:
```
esh_df01 = pd.read_csv(r'C:
  ↪\Users\acarr\Documents\GitHub\ads505_business_proj\data\Potential Data␣
  ↪Sets\E-shop\e-shop clothing 2008.csv', sep=';')
display(esh_df01.shape) # Display dimensions
display(esh_df01.describe()) # Display simple descriptive stats
display(esh_df01.head()) # Display first 5 rows
display(esh_df01.info())
```

```
(165474, 14)

           year          month            day          order        country  \
count  165474.0  165474.000000  165474.000000  165474.000000  165474.000000
mean     2008.0       5.585887      14.524554       9.817476      26.952621
std         0.0       1.328160       8.830374      13.478411       7.150691
min      2008.0       4.000000       1.000000       1.000000       1.000000
25%      2008.0       4.000000       7.000000       2.000000      29.000000
```

```
50%      2008.0       5.000000      14.000000       6.000000      29.000000
75%      2008.0       7.000000      22.000000      12.000000      29.000000
max      2008.0       8.000000      31.000000     195.000000      47.000000

          session ID  page 1 (main category)         colour       location  \
count  165474.000000           165474.000000  165474.000000  165474.000000
mean    12058.417056                2.400842       6.227655       3.258198
std      7008.418903                1.144420       4.235606       1.713206
min         1.000000                1.000000       1.000000       1.000000
25%      5931.000000                1.000000       3.000000       2.000000
50%     11967.500000                2.000000       4.000000       3.000000
75%     18219.000000                3.000000       9.000000       5.000000
max     24026.000000                4.000000      14.000000       6.000000

       model photography          price        price 2           page
count      165474.000000  165474.000000  165474.000000  165474.000000
mean            1.260071      43.802507       1.488167       1.710166
std             0.438674      12.548131       0.499861       0.982412
min             1.000000      18.000000       1.000000       1.000000
25%             1.000000      33.000000       1.000000       1.000000
50%             1.000000      43.000000       1.000000       1.000000
75%             2.000000      52.000000       2.000000       2.000000
max             2.000000      82.000000       2.000000       5.000000
   year  month  day  order  country  session ID  page 1 (main category)  \
0  2008      4    1      1       29           1                       1
1  2008      4    1      2       29           1                       1
2  2008      4    1      3       29           1                       2
3  2008      4    1      4       29           1                       2
4  2008      4    1      5       29           1                       2

  page 2 (clothing model)  colour  location  model photography  price  \
0                     A13       1         5                  1     28
1                     A16       1         6                  1     33
2                      B4      10         2                  1     52
3                     B17       6         6                  2     38
4                      B8       4         3                  2     52

   price 2  page
0        2     1
1        2     1
2        1     1
3        2     1
4        1     1

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 165474 entries, 0 to 165473
Data columns (total 14 columns):
 #   Column                   Non-Null Count   Dtype
```

```
---   ------                    --------------   -----
 0    year                      165474 non-null  int64
 1    month                     165474 non-null  int64
 2    day                       165474 non-null  int64
 3    order                     165474 non-null  int64
 4    country                   165474 non-null  int64
 5    session ID                165474 non-null  int64
 6    page 1 (main category)    165474 non-null  int64
 7    page 2 (clothing model)   165474 non-null  object
 8    colour                    165474 non-null  int64
 9    location                  165474 non-null  int64
 10   model photography         165474 non-null  int64
 11   price                     165474 non-null  int64
 12   price 2                   165474 non-null  int64
 13   page                      165474 non-null  int64
dtypes: int64(13), object(1)
memory usage: 17.7+ MB
```

```
None
```

Discussion:
* Like the first one, this dataset is pretty clean.
* We would have to figure out what we are looking to predict, since this is a page-click dataset. We could bin the number of clicks and enginneer a response feature with a class for each bin.
* N is very large.

Seoul Bikes data set

https://archive.ics.uci.edu/ml/datasets/Seoul+Bike+Sharing+Demand

[4]:
```
bke_df01 = pd.read_csv(r'C:
  ↪\Users\acarr\Documents\GitHub\ads505_business_proj\data\Potential Data␣
  ↪Sets\Seoul Bikes\SeoulBikeData.csv', sep=',')
display(bke_df01.shape) # Display dimensions
display(bke_df01.describe()) # Display simple descriptive stats
display(bke_df01.head()) # Display first 5 rows
display(bke_df01.info())
```

```
(8760, 14)
```

```
       Rented Bike Count         Hour  Temperature(C)  Humidity(%)  \
count        8760.000000  8760.000000     8760.000000  8760.000000
mean          704.602055    11.500000       12.882922    58.226256
std           644.997468     6.922582       11.944825    20.362413
min             0.000000     0.000000      -17.800000     0.000000
25%           191.000000     5.750000        3.500000    42.000000
50%           504.500000    11.500000       13.700000    57.000000
75%          1065.250000    17.250000       22.500000    74.000000
max          3556.000000    23.000000       39.400000    98.000000
```

```
       Wind speed (m/s)  Visibility (10m)  Dew point temperature(C)  \
count       8760.000000       8760.000000               8760.000000
mean           1.724909       1436.825799                  4.073813
std            1.036300        608.298712                 13.060369
min            0.000000         27.000000                -30.600000
25%            0.900000        940.000000                 -4.700000
50%            1.500000       1698.000000                  5.100000
75%            2.300000       2000.000000                 14.800000
max            7.400000       2000.000000                 27.200000

       Solar Radiation (MJ/m2)  Rainfall(mm)  Snowfall (cm)
count              8760.000000   8760.000000    8760.000000
mean                  0.569111      0.148687       0.075068
std                   0.868746      1.128193       0.436746
min                   0.000000      0.000000       0.000000
25%                   0.000000      0.000000       0.000000
50%                   0.010000      0.000000       0.000000
75%                   0.930000      0.000000       0.000000
max                   3.520000     35.000000       8.800000
         Date  Rented Bike Count  Hour  Temperature(C)  Humidity(%)  \
0  01/12/2017                254     0            -5.2           37
1  01/12/2017                204     1            -5.5           38
2  01/12/2017                173     2            -6.0           39
3  01/12/2017                107     3            -6.2           40
4  01/12/2017                 78     4            -6.0           36

   Wind speed (m/s)  Visibility (10m)  Dew point temperature(C)  \
0               2.2              2000                     -17.6
1               0.8              2000                     -17.6
2               1.0              2000                     -17.7
3               0.9              2000                     -17.6
4               2.3              2000                     -18.6

   Solar Radiation (MJ/m2)  Rainfall(mm)  Snowfall (cm) Seasons     Holiday  \
0                      0.0           0.0            0.0  Winter  No Holiday
1                      0.0           0.0            0.0  Winter  No Holiday
2                      0.0           0.0            0.0  Winter  No Holiday
3                      0.0           0.0            0.0  Winter  No Holiday
4                      0.0           0.0            0.0  Winter  No Holiday

  Functioning Day
0             Yes
1             Yes
2             Yes
3             Yes
4             Yes

<class 'pandas.core.frame.DataFrame'>
```

```
RangeIndex: 8760 entries, 0 to 8759
Data columns (total 14 columns):
 #   Column                     Non-Null Count  Dtype
---  ------                     --------------  -----
 0   Date                       8760 non-null   object
 1   Rented Bike Count          8760 non-null   int64
 2   Hour                       8760 non-null   int64
 3   Temperature(C)             8760 non-null   float64
 4   Humidity(%)                8760 non-null   int64
 5   Wind speed (m/s)           8760 non-null   float64
 6   Visibility (10m)           8760 non-null   int64
 7   Dew point temperature(C)   8760 non-null   float64
 8   Solar Radiation (MJ/m2)    8760 non-null   float64
 9   Rainfall(mm)               8760 non-null   float64
 10  Snowfall (cm)              8760 non-null   float64
 11  Seasons                    8760 non-null   object
 12  Holiday                    8760 non-null   object
 13  Functioning Day            8760 non-null   object
dtypes: float64(6), int64(4), object(4)
memory usage: 958.2+ KB
```

```
None
```

Discussion:
* Clear response feature for regression
* No null values
* Sufficient n, but low feature count (p).
* Mix of categorical and numerical features.

References

Satopay, H. (2018, November 18). The ultimate markdown guide (for Jupyter Notebook). *Medium* https://medium.com/analytics-vidhya/the-ultimate-markdown-guide-for-jupyter-notebook-d5e5abf728fd