ADS505_Final1

October 13, 2022

1 Predicting Bike Rental Counts in Seoul Based on the Weather and Holiday Information for a Stable Supply

2 Team 5

2.1 Kyle Dalope

3 ADS505-01-FA22

[51]:

```
! pip install dmba
```

```
Looking in indexes: https://pypi.org/simple, https://us-python.pkg.dev/colab-
wheels/public/simple/
Requirement already satisfied: dmba in /usr/local/lib/python3.7/dist-packages
(0.1.0)
```

[84]:

```
#Imports Required

import pandas as pd
import matplotlib.pylab as plt
import numpy as np

from sklearn import preprocessing
from sklearn.ensemble import RandomForestClassifier
from sklearn.neighbors import KNeighborsClassifier
from sklearn.feature_selection import RFE
from sklearn.linear_model import LinearRegression
import statsmodels.api as sm
from sklearn.model_selection import train_test_split
from sklearn.metrics import confusion_matrix
from dmba import gainsChart, liftChart
from dmba import regressionSummary
from dmba import adjusted_r2_score, AIC_score, BIC_score
#from dmba import classificationSummary
```

```
from sklearn.metrics import f1_score, precision_score, recall_score,␣
 ↪accuracy_score

%matplotlib inline
```

4 EDA

[53]:
```
#Data import

SeoulBike_df = pd.read_csv("SeoulBikeData.csv", encoding = 'unicode_escape',
                           parse_dates=[0])
```

[54]:
```
SeoulBike_df.head()
```

[54]:
```
        Date  Rented Bike Count  Hour  Temperature(°C)  Humidity(%)  \
0 2017-01-12                254     0             -5.2           37
1 2017-01-12                204     1             -5.5           38
2 2017-01-12                173     2             -6.0           39
3 2017-01-12                107     3             -6.2           40
4 2017-01-12                 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
```

[55]:
```
SeoulBike_df.describe() #Statistical summary
```

[55]:
```
       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  
```

[56]:
```python
SeoulBike_df.info() #Observe data types
```

```
<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   datetime64[ns]
 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: datetime64[ns](1), float64(6), int64(4), object(3)
memory usage: 958.2+ KB
```

[57]:
```python
SeoulBike_df.isnull().sum() #Observe if any missing data exists
```

[57]:
```
Date                         0
Rented Bike Count            0
Hour                         0
Temperature(°C)              0
Humidity(%)                  0
Wind speed (m/s)             0
Visibility (10m)             0
Dew point temperature(°C)    0
Solar Radiation (MJ/m2)      0
Rainfall(mm)                 0
Snowfall (cm)                0
Seasons                      0
Holiday                      0
Functioning Day              0
dtype: int64
```

5 Data Pre-processing

[58]:
```python
#Reformat Column Names

SeoulBike_df = SeoulBike_df.copy()

SeoulBike_df.columns = [d.replace(' ', '_').replace('.', '') for d in␣
 ↪SeoulBike_df.columns]
SeoulBike_df.head()
```

[58]:
```
        Date  Rented_Bike_Count  Hour  Temperature(°C)  Humidity(%)  \
0 2017-01-12                254     0             -5.2           37
1 2017-01-12                204     1             -5.5           38
2 2017-01-12                173     2             -6.0           39
3 2017-01-12                107     3             -6.2           40
4 2017-01-12                 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
```

[59]:

```
SeoulBike_df1 = SeoulBike_df.copy()

SeoulBike_df1 = pd.get_dummies(SeoulBike_df1, columns= ['Seasons', 'Holiday',␣
 ↪'Functioning_Day'],
                               prefix_sep='_')
SeoulBike_df1
```

[59]:

```
           Date  Rented_Bike_Count  Hour  Temperature(°C)  Humidity(%)  \
0    2017-01-12                254     0             -5.2           37
1    2017-01-12                204     1             -5.5           38
2    2017-01-12                173     2             -6.0           39
3    2017-01-12                107     3             -6.2           40
4    2017-01-12                 78     4             -6.0           36
…         …                …   …         …          …
8755 2018-11-30               1003    19              4.2           34
8756 2018-11-30                764    20              3.4           37
8757 2018-11-30                694    21              2.6           39
8758 2018-11-30                712    22              2.1           41
8759 2018-11-30                584    23              1.9           43

      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
…                …               …                        …
8755               2.6              1894                      -10.3
8756               2.3              2000                       -9.9
```

```
8757               0.3               1968                        -9.9
8758               1.0               1859                        -9.8
8759               1.3               1909                        -9.3

      Solar_Radiation_(MJ/m2)  Rainfall(mm)  Snowfall_(cm)  Seasons_Autumn  \
0                         0.0           0.0            0.0               0
1                         0.0           0.0            0.0               0
2                         0.0           0.0            0.0               0
3                         0.0           0.0            0.0               0
4                         0.0           0.0            0.0               0
…                       …           …            …             …
8755                      0.0           0.0            0.0               1
8756                      0.0           0.0            0.0               1
8757                      0.0           0.0            0.0               1
8758                      0.0           0.0            0.0               1
8759                      0.0           0.0            0.0               1

      Seasons_Spring  Seasons_Summer  Seasons_Winter  Holiday_Holiday  \
0                  0               0               1                0
1                  0               0               1                0
2                  0               0               1                0
3                  0               0               1                0
4                  0               0               1                0
…              …             …             …              …
8755               0               0               0                0
8756               0               0               0                0
8757               0               0               0                0
8758               0               0               0                0
8759               0               0               0                0

      Holiday_No Holiday  Functioning_Day_No  Functioning_Day_Yes
0                      1                   0                    1
1                      1                   0                    1
2                      1                   0                    1
3                      1                   0                    1
4                      1                   0                    1
…                  …                 …                  …
8755                   1                   0                    1
8756                   1                   0                    1
8757                   1                   0                    1
8758                   1                   0                    1
8759                   1                   0                    1

[8760 rows x 19 columns]
```

[80]:
```
#Preprocess the data column
SeoulBike_df2 = SeoulBike_df1.copy()
```

```
SeoulBike_df2['Date_year'] = SeoulBike_df2['Date'].dt.year
SeoulBike_df2['Date_month'] = SeoulBike_df2['Date'].dt.month
SeoulBike_df2['Date_day'] = SeoulBike_df2['Date'].dt.day
```

[69]:
```
#Check dataframe
SeoulBike_df2
```

[69]:
```
           Date  Rented_Bike_Count  Hour  Temperature(°C)  Humidity(%)  \
0    2017-01-12                254     0             -5.2           37
1    2017-01-12                204     1             -5.5           38
2    2017-01-12                173     2             -6.0           39
3    2017-01-12                107     3             -6.2           40
4    2017-01-12                 78     4             -6.0           36
…         …                …   …              …          …
8755 2018-11-30               1003    19              4.2           34
8756 2018-11-30                764    20              3.4           37
8757 2018-11-30                694    21              2.6           39
8758 2018-11-30                712    22              2.1           41
8759 2018-11-30                584    23              1.9           43

      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
…                …                …                         …
8755               2.6              1894                      -10.3
8756               2.3              2000                       -9.9
8757               0.3              1968                       -9.9
8758               1.0              1859                       -9.8
8759               1.3              1909                       -9.3

      Solar_Radiation_(MJ/m2)  Rainfall(mm)  …  Seasons_Summer  \
0                         0.0           0.0  …               0
1                         0.0           0.0  …               0
2                         0.0           0.0  …               0
3                         0.0           0.0  …               0
4                         0.0           0.0  …               0
…                       …           …  …             …
8755                      0.0           0.0  …               0
8756                      0.0           0.0  …               0
8757                      0.0           0.0  …               0
8758                      0.0           0.0  …               0
8759                      0.0           0.0  …               0
```

```
      Seasons_Winter  Holiday_Holiday  Holiday_No Holiday  Functioning_Day_No  \
0                  1                0                   1                   0
1                  1                0                   1                   0
2                  1                0                   1                   0
3                  1                0                   1                   0
4                  1                0                   1                   0
…                …                …                   …                   …
8755               0                0                   1                   0
8756               0                0                   1                   0
8757               0                0                   1                   0
8758               0                0                   1                   0
8759               0                0                   1                   0

      Functioning_Day_Yes  Date_year  Date_month  Date_week  Date_day
0                       1       2017           1          2        12
1                       1       2017           1          2        12
2                       1       2017           1          2        12
3                       1       2017           1          2        12
4                       1       2017           1          2        12
…                     …        …          …         …       …
8755                    1       2018          11         48        30
8756                    1       2018          11         48        30
8757                    1       2018          11         48        30
8758                    1       2018          11         48        30
8759                    1       2018          11         48        30

[8760 rows x 23 columns]
```

[76]:

```
# Separate out predictors and outcome variable
X = SeoulBike_df2.drop(columns=['Rented_Bike_Count', 'Date', 'Date_week'],␣
 ↪axis=0)
y = SeoulBike_df2['Rented_Bike_Count']

# partition the data into training (60%) and validation (40%) sets. use␣
 ↪random_state=1 for reproducibility of results
train_X, valid_X, train_y, valid_y = train_test_split(X, y, test_size=0.4,␣
 ↪random_state=1)
```

[77]:

```
print(train_X.shape, train_y.shape)
print(valid_X.shape, valid_y.shape)
```

```
(5256, 20) (5256,)
(3504, 20) (3504,)
```

[81]:

```
train_X
```

```
[81]:       Hour  Temperature(°C)  Humidity(%)  Wind_speed_(m/s)  Visibility_(10m)  \
      3631     7             18.0           82               0.7               264
      790     22             -3.4           33               2.4              2000
      7841    17             15.7           63               3.2              1118
      934     22              0.1           68               4.6               740
      6620    20             26.9           56               1.6              2000
      …    …              …          …               …               …
      2895    15             18.9           28               3.7              1769
      7813    13             19.4           35               1.2               678
      905     17              2.3           33               0.2              1515
      5192     8             23.6           93               0.9               308
      235     19              1.6           53               4.0              2000

            Dew_point_temperature(°C)  Solar_Radiation_(MJ/m2)  Rainfall(mm)  \
      3631                       14.8                     0.11           0.0
      790                       -17.4                     0.00           0.0
      7841                        8.6                     0.39           0.0
      934                        -5.1                     0.00           0.0
      6620                       17.3                     0.00           0.0
      …                         …                      …           …
      2895                        0.0                     2.09           0.0
      7813                        3.5                     1.73           0.0
      905                       -12.3                     0.13           0.0
      5192                       22.3                     0.21           0.0
      235                        -6.9                     0.00           0.0

            Snowfall_(cm)  Seasons_Autumn  Seasons_Spring  Seasons_Summer  \
      3631            0.0               0               1               0
      790             0.0               0               0               0
      7841            0.0               1               0               0
      934             1.0               0               0               0
      6620            0.0               1               0               0
      …              …             …             …             …
      2895            0.0               0               1               0
      7813            0.0               1               0               0
      905             0.0               0               0               0
      5192            0.0               0               0               1
      235             0.0               0               0               0

            Seasons_Winter  Holiday_Holiday  Holiday_No Holiday  Functioning_Day_No  \
      3631               0                1                   0                   0
      790                1                0                   1                   0
      7841               0                0                   1                   0
      934                1                0                   1                   0
      6620               0                0                   1                   0
      …              …              …                 …                 …
      2895               0                0                   1                   0
```

```
7813               0                0                   1                   0
905                1                0                   1                   0
5192               0                0                   1                   0
235                1                0                   1                   0

      Functioning_Day_Yes  Date_year  Date_month  Date_day
3631                    1       2018           1         5
790                     1       2018           2         1
7841                    1       2018          10        23
934                     1       2018           8         1
6620                    1       2018           2         9
…                   …        …         …       …
2895                    1       2018           3        31
7813                    1       2018          10        22
905                     1       2018           7         1
5192                    1       2018           5         7
235                     1       2017          10        12

[5256 rows x 20 columns]
```

[82]:

```python
#Preliminary Model to determine if all variables are to be included
sb_lm = LinearRegression()
sb_lm.fit(train_X, train_y)

# print coefficients
print('intercept ', sb_lm.intercept_)
display(pd.DataFrame({'Predictor': X.columns, 'coefficient': sb_lm.coef_}))

# print performance measures
regressionSummary(train_y, sb_lm.predict(train_X))
```

```
intercept  222171.79999533866

                    Predictor  coefficient
0                        Hour    27.575007
1             Temperature(°C)     9.439506
2                 Humidity(%)   -13.157778
3            Wind_speed_(m/s)    11.186503
4            Visibility_(10m)     0.003515
5   Dew_point_temperature(°C)    17.756664
6     Solar_Radiation_(MJ/m2)   -72.818209
7                Rainfall(mm)   -49.360981
8               Snowfall_(cm)    38.756531
9              Seasons_Autumn   184.254745
10             Seasons_Spring    31.885788
11             Seasons_Summer    25.067836
12             Seasons_Winter  -241.208369
13            Holiday_Holiday   -67.900269
```

```
14         Holiday_No Holiday    67.900269
15         Functioning_Day_No  -466.229531
16        Functioning_Day_Yes   466.229531
17                  Date_year  -109.847549
18                 Date_month    -0.336785
19                   Date_day    -1.285028


Regression statistics

               Mean Error (ME) : 0.0000
Root Mean Squared Error (RMSE) : 432.1178
     Mean Absolute Error (MAE) : 321.9594
```

[85]:

```python
# get predictions based on train_X
pred_y = sb_lm.predict(train_X)

#calculate adjusted r2 and information criteria measures
print('adjusted r2 : ', adjusted_r2_score(train_y, pred_y, sb_lm))
print('AIC : ', AIC_score(train_y, pred_y, sb_lm))
print('BIC : ', BIC_score(train_y, pred_y, sb_lm))
```

```
adjusted r2 :  0.5533662344890359
AIC :  78754.03677098356
BIC :  78898.51353330717
```

6 Model Selections

[ ]:

[ ]:

[ ]:

7 Model Evaluation

[ ]:

[ ]:

[ ]:

8 Final Model Selection and Conclusion

```
[ ]:
```

```
[89]: !sudo apt-get install texlive-xetex texlive-fonts-recommended␣
       ↪texlive-plain-generic
      !jupyter nbconvert --to pdf /content/ADS505_Final.ipynb
```

```
Reading package lists… Done
Building dependency tree
Reading state information… Done
texlive-fonts-recommended is already the newest version (2017.20180305-1).
texlive-plain-generic is already the newest version (2017.20180305-2).
texlive-xetex is already the newest version (2017.20180305-1).
The following package was automatically installed and is no longer required:
  libnvidia-common-460
Use 'sudo apt autoremove' to remove it.
0 upgraded, 0 newly installed, 0 to remove and 12 not upgraded.
[NbConvertApp] WARNING | pattern '/content/ADS505_Final.ipynb' matched no files
This application is used to convert notebook files (*.ipynb)
        to various other formats.

        WARNING: THE COMMANDLINE INTERFACE MAY CHANGE IN FUTURE RELEASES.

Options
=======
The options below are convenience aliases to configurable class-options,
as listed in the "Equivalent to" description-line of the aliases.
To see all configurable class-options for some <cmd>, use:
    <cmd> --help-all

--debug
    set log level to logging.DEBUG (maximize logging output)
    Equivalent to: [--Application.log_level=10]
--show-config
    Show the application's configuration (human-readable format)
    Equivalent to: [--Application.show_config=True]
--show-config-json
    Show the application's configuration (json format)
    Equivalent to: [--Application.show_config_json=True]
--generate-config
    generate default config file
    Equivalent to: [--JupyterApp.generate_config=True]
-y
    Answer yes to any questions instead of prompting.
    Equivalent to: [--JupyterApp.answer_yes=True]
--execute
```

```
    Execute the notebook prior to export.
    Equivalent to: [--ExecutePreprocessor.enabled=True]
--allow-errors
    Continue notebook execution even if one of the cells throws an error and
include the error message in the cell output (the default behaviour is to abort
conversion). This flag is only relevant if '--execute' was specified, too.
    Equivalent to: [--ExecutePreprocessor.allow_errors=True]
--stdin
    read a single notebook file from stdin. Write the resulting notebook with
default basename 'notebook.*'
    Equivalent to: [--NbConvertApp.from_stdin=True]
--stdout
    Write notebook output to stdout instead of files.
    Equivalent to: [--NbConvertApp.writer_class=StdoutWriter]
--inplace
    Run nbconvert in place, overwriting the existing notebook (only
            relevant when converting to notebook format)
    Equivalent to: [--NbConvertApp.use_output_suffix=False
--NbConvertApp.export_format=notebook --FilesWriter.build_directory=]
--clear-output
    Clear output of current file and save in place,
            overwriting the existing notebook.
    Equivalent to: [--NbConvertApp.use_output_suffix=False
--NbConvertApp.export_format=notebook --FilesWriter.build_directory=
--ClearOutputPreprocessor.enabled=True]
--no-prompt
    Exclude input and output prompts from converted document.
    Equivalent to: [--TemplateExporter.exclude_input_prompt=True
--TemplateExporter.exclude_output_prompt=True]
--no-input
    Exclude input cells and output prompts from converted document.
            This mode is ideal for generating code-free reports.
    Equivalent to: [--TemplateExporter.exclude_output_prompt=True
--TemplateExporter.exclude_input=True]
--log-level=<Enum>
    Set the log level by value or name.
    Choices: any of [0, 10, 20, 30, 40, 50, 'DEBUG', 'INFO', 'WARN', 'ERROR',
'CRITICAL']
    Default: 30
    Equivalent to: [--Application.log_level]
--config=<Unicode>
    Full path of a config file.
    Default: ''
    Equivalent to: [--JupyterApp.config_file]
--to=<Unicode>
    The export format to be used, either one of the built-in formats
            ['asciidoc', 'custom', 'html', 'latex', 'markdown', 'notebook',
'pdf', 'python', 'rst', 'script', 'slides']
```

```
            or a dotted object name that represents the import path for an
            `Exporter` class
    Default: 'html'
    Equivalent to: [--NbConvertApp.export_format]
--template=<Unicode>
    Name of the template file to use
    Default: ''
    Equivalent to: [--TemplateExporter.template_file]
--writer=<DottedObjectName>
    Writer class used to write the
                                        results of the conversion
    Default: 'FilesWriter'
    Equivalent to: [--NbConvertApp.writer_class]
--post=<DottedOrNone>
    PostProcessor class used to write the
                                        results of the conversion
    Default: ''
    Equivalent to: [--NbConvertApp.postprocessor_class]
--output=<Unicode>
    overwrite base name use for output files.
                can only be used when converting one notebook at a time.
    Default: ''
    Equivalent to: [--NbConvertApp.output_base]
--output-dir=<Unicode>
    Directory to write output(s) to. Defaults
                                  to output to the directory of each notebook.
To recover
                                  previous default behaviour (outputting to the
current
                                  working directory) use . as the flag value.
    Default: ''
    Equivalent to: [--FilesWriter.build_directory]
--reveal-prefix=<Unicode>
    The URL prefix for reveal.js (version 3.x).
            This defaults to the reveal CDN, but can be any url pointing to a
copy
            of reveal.js.
            For speaker notes to work, this must be a relative path to a local
            copy of reveal.js: e.g., "reveal.js".
            If a relative path is given, it must be a subdirectory of the
            current directory (from which the server is run).
            See the usage documentation
            (https://nbconvert.readthedocs.io/en/latest/usage.html#reveal-js-
html-slideshow)
            for more details.
    Default: ''
    Equivalent to: [--SlidesExporter.reveal_url_prefix]
--nbformat=<Enum>
```

```
    The nbformat version to write.
            Use this to downgrade notebooks.
    Choices: any of [1, 2, 3, 4]
    Default: 4
    Equivalent to: [--NotebookExporter.nbformat_version]

Examples
--------

    The simplest way to use nbconvert is

            > jupyter nbconvert mynotebook.ipynb

            which will convert mynotebook.ipynb to the default format (probably
HTML).

            You can specify the export format with `--to`.
            Options include ['asciidoc', 'custom', 'html', 'latex', 'markdown',
'notebook', 'pdf', 'python', 'rst', 'script', 'slides'].

            > jupyter nbconvert --to latex mynotebook.ipynb

            Both HTML and LaTeX support multiple output templates. LaTeX
includes
            'base', 'article' and 'report'.  HTML includes 'basic' and 'full'.
You
            can specify the flavor of the format used.

            > jupyter nbconvert --to html --template basic mynotebook.ipynb

            You can also pipe the output to stdout, rather than a file

            > jupyter nbconvert mynotebook.ipynb --stdout

            PDF is generated via latex

            > jupyter nbconvert mynotebook.ipynb --to pdf

            You can get (and serve) a Reveal.js-powered slideshow

            > jupyter nbconvert myslides.ipynb --to slides --post serve

            Multiple notebooks can be given at the command line in a couple of
            different ways:

            > jupyter nbconvert notebook*.ipynb
            > jupyter nbconvert notebook1.ipynb notebook2.ipynb
            or you can specify the notebooks list in a config file, containing::

                c.NbConvertApp.notebooks = ["my_notebook.ipynb"]

            > jupyter nbconvert --config mycfg.py

To see all available configurables, use `--help-all`.
```