Python for scientific research

Advanced topics

John Joseph Valletta

University of Exeter, Penryn Campus, UK

June 2017

UNIVERSITY OF EXETER | DOCTORAL COLLEGE

Researcher Development

Statsmodels

statsmodels is a package for the estimation of different statistical models

```python
import statsmodels.api as sm
import statsmodels.formula.api as smf

# Read Boston housing data set (a Pandas data frame)
df = sm.datasets.get_rdataset("Boston", "MASS").data
df.head()
```

```
      crim    zn  indus  chas    nox     rm   age     dis  rad  tax  ptratio  \
0  0.00632  18.0   2.31     0  0.538  6.575  65.2  4.0900    1  296     15.3
1  0.02731   0.0   7.07     0  0.469  6.421  78.9  4.9671    2  242     17.8
2  0.02729   0.0   7.07     0  0.469  7.185  61.1  4.9671    2  242     17.8
3  0.03237   0.0   2.18     0  0.458  6.998  45.8  6.0622    3  222     18.7
4  0.06905   0.0   2.18     0  0.458  7.147  54.2  6.0622    3  222     18.7

    black  lstat  medv
0  396.90   4.98  24.0
1  396.90   9.14  21.6
2  392.83   4.03  34.7
3  394.63   2.94  33.4
4  396.90   5.33  36.2
```

Statsmodels: Generalised linear model

```
# Dependent variable
# medv - median value of owner-occupied homes in $1000's

# Covariates
# crim - per capita crime rate by town
# nox - nitric oxides concentration (parts per 10 million)
# rm - average number of rooms per dwelling
# indus - proportion of non-retail business acres per town.
# rad - index of accessibility to radial highways

# Set up model
model = smf.glm(formula="medv ~ crim + nox + rm + indus + rad",
                family=sm.families.Gaussian(),
                data=df)

# Fit model
model = model.fit()
```

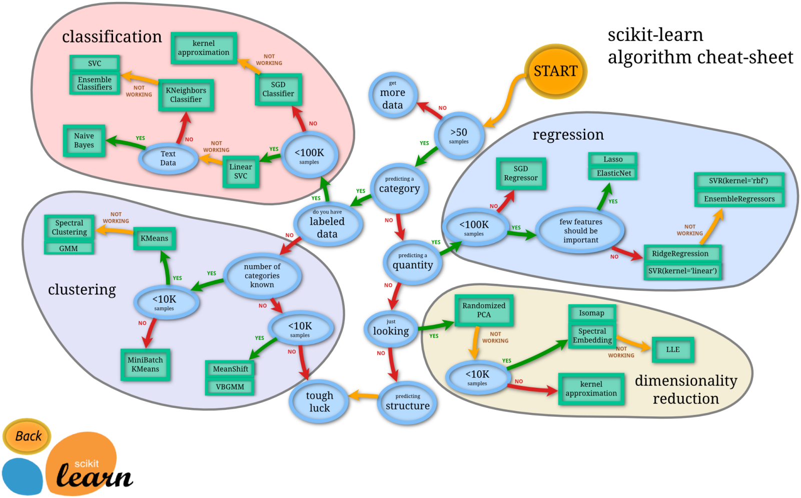
Statsmodels: Generalised linear model

```python
# Show summary
model.summary()
```

```
                 Generalized Linear Model Regression Results
==============================================================================
Dep. Variable:                   medv   No. Observations:                  506
Model:                            GLM   Df Residuals:                      500
Model Family:                Gaussian   Df Model:                            5
Link Function:               identity   Scale:                   36.7202989747
Method:                          IRLS   Log-Likelihood:                -1626.6
Date:                Tue, 30 May 2017   Deviance:                       18360.
Time:                        15:00:23   Pearson chi2:                 1.84e+04
No. Iterations:                     2
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
Intercept    -19.9794      3.267     -6.115      0.000     -26.383     -13.576
crim          -0.1608      0.040     -3.973      0.000      -0.240      -0.081
nox           -5.7432      3.784     -1.518      0.129     -13.160       1.674
rm             7.7004      0.420     18.353      0.000       6.878       8.523
indus         -0.1361      0.065     -2.089      0.037      -0.264      -0.008
rad           -0.0628      0.047     -1.342      0.179      -0.154       0.029
==============================================================================
```

Scikit-learn

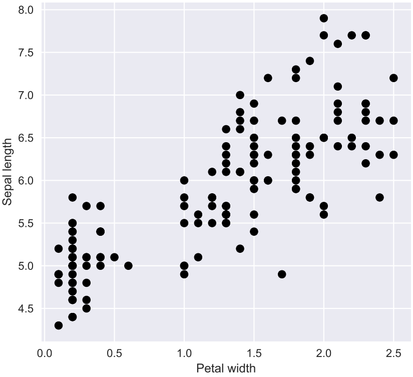
`sklearn` is a package for machine learning algorithms



Scikit-learn

```
from sklearn import datasets

# Load popular iris data set
iris = datasets.load_iris()
xTrain = iris.data # petal/sepal width/length
yTrain = iris.target
```

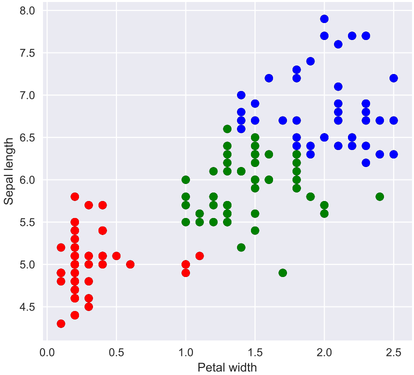


Scikit-learn: k-means clustering

```
from sklearn.cluster import KMeans

# Set up k-means clustering model
model = KMeans(n_clusters=3)

# Fit model with supplied data (unsupervised)
model.fit(xTrain)
```

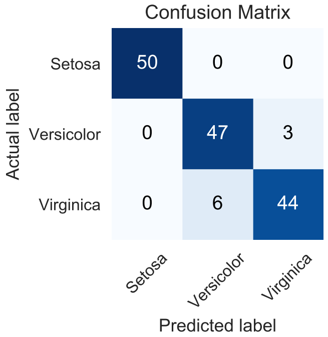


Scikit-learn: Random forests

```
from sklearn.ensemble import RandomForestClassifier

# Set up Random Forest model
model = RandomForestClassifier(n_estimators=500, oob_score=True)

# Fit model with supplied data (supervised)
model.fit(xTrain, yTrain)
```



Networkx

networkx is a package for the creation, manipulation, and study of the structure, dynamics, and functions of complex networks

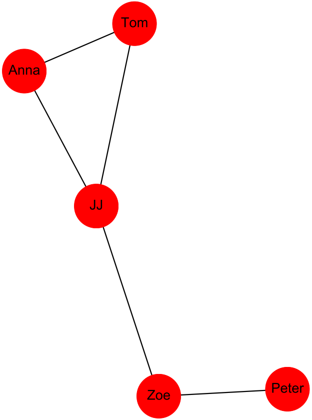
```
import networkx as nx

# Creating a new empty Graph object
g = nx.Graph()

# Adding nodes
g.add_nodes_from(["Tom", "Zoe",
                  "JJ", "Anna", "Peter"])

# Adding an edge
g.add_edge("JJ", "Zoe")
g.add_edge("JJ", "Anna")
g.add_edge("JJ", "Tom")
g.add_edge("Peter", "Zoe")
g.add_edge("Tom", "Anna")

# Draw network
nx.draw(g, node_size=1500,
        with_labels=True)
```



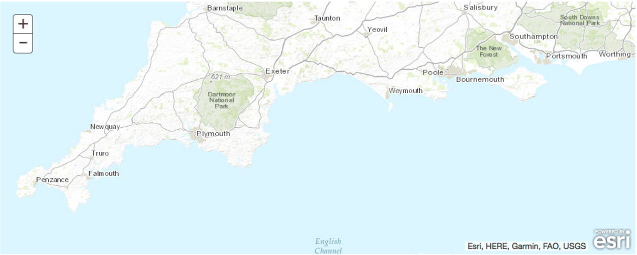
ArcGIS

arcgis is a package for GIS visualisation and analysis, spatial data management and GIS system administration tasks

```
from arcgis.gis import GIS

# Create a GIS object (anonymous user)
gis = GIS()

# Get map of Cornwall
myMap = gis.map("Cornwall, UK")
myMap
```



MyGene

mygene provides web services to query/retrieve gene annotation data from MyGene.info

```python
import mygene

# Instantiate MyGeneInfo class
mg = mygene.MyGeneInfo()

# Query for the name and EnsemblID of genes Ifng and Ccl3
mg.querymany(qterms=["Ifng", "Ccl3"], scopes="symbol",
             fields=["name", "ensembl"], species="human")
```

```
[{’_id’: ’3458’,
  ’_score’: 98.66912,
  ’ensembl’: {’gene’: ’ENSG00000111537’,
   ’protein’: ’ENSP00000229135’,
   ’transcript’: ’ENST00000229135’,
   ’translation’: [{’protein’: ’ENSP00000229135’, ’rna’: ’ENST00000229135’}]},
  ’name’: ’interferon gamma’,
  ’query’: ’Ifng’},
....
....
....
```

Tweepy

tweepy provides access to Twitter for posting/reading tweets

How about analysing tweets from Mr President?

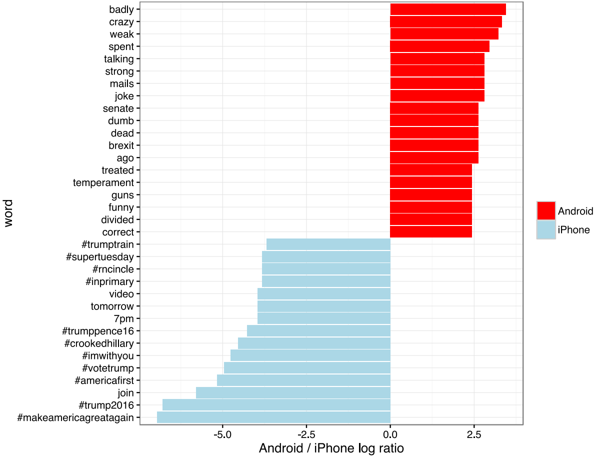


Image taken from here (analysis done in R)