K-Nearest Neighbour

K-NN

Idea Behind K-NN

Birds of the Same Feather Flock Together

Courtesy: www.understandingsociety.ac.uk/2013/07/26/do-birds-of-a-feather-flock-together

Courtesy: http://positivity360.com/post-2/

K - Nearest Neighbors

- This algorithm can be used for classification as well as regression
- The algorithm looks for observations in our training data that are similar or “near” the record to be classified in the predictor space (i.e., records that have values close to X1, X2, . . . , Xp).
- In k-nearest neighbors method, the classifier identifies k observations in the training dataset that are similar to a new record that we wish to classify.
- In k-nearest neighbors method, the regressor identifies k observations in the training dataset that are similar to a new record that we wish to take and gives the average of the response values of those k nearest neighboring observations.

Distance Method

- For record i we have the vector of p measurements (xi1, xi2, . . . , xip), while for record j we have the vector of measurements (xj1, xj2, . . . , xjp).
- The most popular distance measure is the Euclidean distance, dij , which between two cases, i and j, is defined by

$$d_{ij} = \sqrt{(x_{i1} - x_{j1})^2 + (x_{i2} - x_{j2})^2 + \ldots + (x_{ip} - x_{jp})^2}$$

Other Distance Measures

- Numerical Data
  - Correlation-based similarity
  - Statistical distance (also called Mahalanobis distance)
  - Manhattan distance (“city block”)
  - Maximum coordinate distance
- Categorical Data
  - Matching coefficient: $(a + d)/p$
  - Jaquard’s coefficient: $d/(b+c+d)$

K - NN

- The k-nearest neighbors algorithm is a classification method that does not make assumptions about the form of the relationship between the response (Y) and the predictors X1,X2, . . .,Xp.
- This is a nonparametric method because it does not involve estimation of parameters as against the methods like linear regression.

Classification Example: Riding Mowers

- A riding-mower manufacturer **MOW-EASE** took part in a Industrial Exhibition in which it got an opportunity to show a demo of its product to 180 different audience.
- The land owned by each of the audience and their approximate income have been recorded in the file RidingMowers.csv

Glimpse of Data: First 10 Obs.

Index	Income	Lot_Size	Response
0	34	26	Not Bought
1	34	40	Not Bought
2	34	46	Not Bought
3	34	48	Not Bought
4	34	53	Not Bought
5	34	58	Not Bought
6	34	59	Not Bought
7	34	63	Not Bought
8	34	64	Not Bought
9	34	66	Bought

Visualizing the Data

- Here we see that the response has some pattern of farness or nearness

Nearest Observations: K=1

- Consider a person with Income as $ 70,000 and Lot size as 100,000 sq. ft.
- By Euclidean Distance Method, the nearest one observation is the 136th observation.

136	73	102	Not Bought

- As we can see here, that 136th observation person has not bought in spite of showing him the product demo. Hence we can conclude that the person with Income as $ 70,000 and Lot size as 100,000 sq. ft. won’t buy.

Nearest Observations: K=3

- By Euclidean Distance Method, the nearest three observations are 136th, 116th and 141st.

136	73	102	Not Bought
116	67	97	Bought
141	74	98	Not Bought

- As we can see here, that 2 have not bought and 1 has bought in spite of showing him the product demo. Hence we can conclude that the person with Income as $ 70,000 and Lot size as 100,000 sq. ft. won't buy.

Nearest Observations: K=5

- By Euclidean Distance Method, the nearest three observations are 136^{th}, 137^{th}, 116^{th}, 143^{rd} and 141^{st}.

116	67	97	Bought
136	73	102	Not Bought
137	73	105	Bought
141	74	98	Not Bought
143	75	98	Bought

- As we can see here, that 2 have not bought and 3 have bought in spite of showing him the product demo. Hence we can conclude that the person with Income as $ 70,000 and Lot size as 100,000 sq. ft. will buy.

K-NN Classification in Python

- K-NN classifier can be implemented from scikit-learn function KNeighborsClassifier
- KNeighborsClassifier object is instantiated and fit method is called on the object

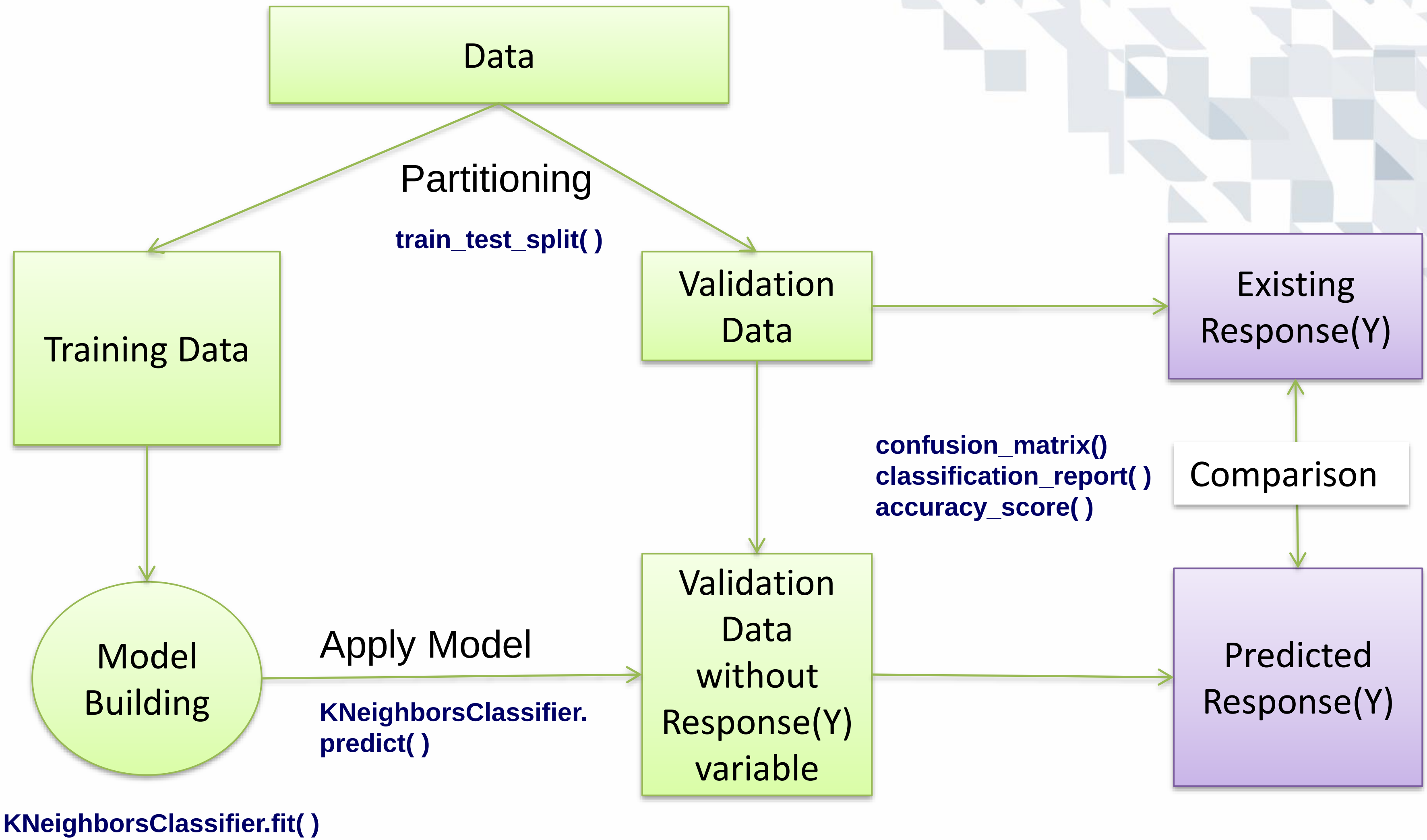
Syntax :

KNeighborsClassifier(n_neighbors, …)

Where

n_neighbors: Number of neighbours (k)

K-NN Classifier



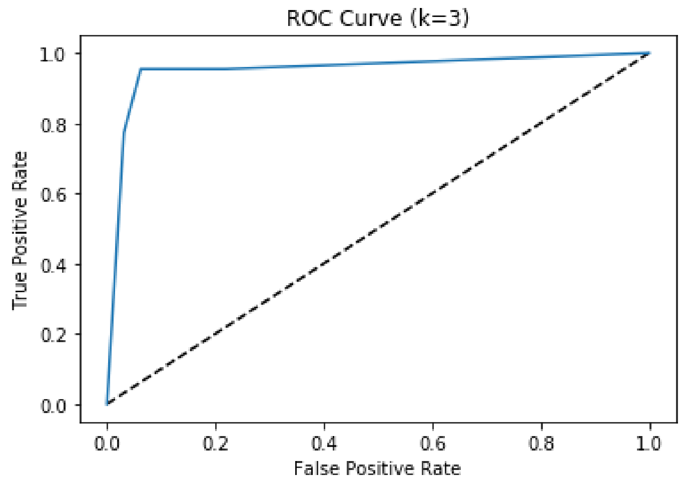
Program and Output

```
In [55]: from sklearn.model_selection import train_test_split
    ...: from sklearn.metrics import confusion_matrix
    ...: from sklearn.metrics import classification_report, accuracy_score
    ...: from sklearn.neighbors import KNeighborsClassifier
    ...:
    ...: X = dum_df.iloc[:,0:2]
    ...: y = dum_df.iloc[:,2]

In [56]: X_train, X_test, y_train, y_test = train_test_split(X, y,test_size = 0.3,
    ...:                                                     random_state=2018,
    ...:                                                     stratify=y)
    ...:
    ...: knn = KNeighborsClassifier(n_neighbors=5)
    ...: knn.fit( X_train , y_train )
    ...: y_pred = knn.predict(X_test)
    ...:
    ...: print(confusion_matrix(y_test, y_pred))
[[30  2]
 [ 1 21]]
```

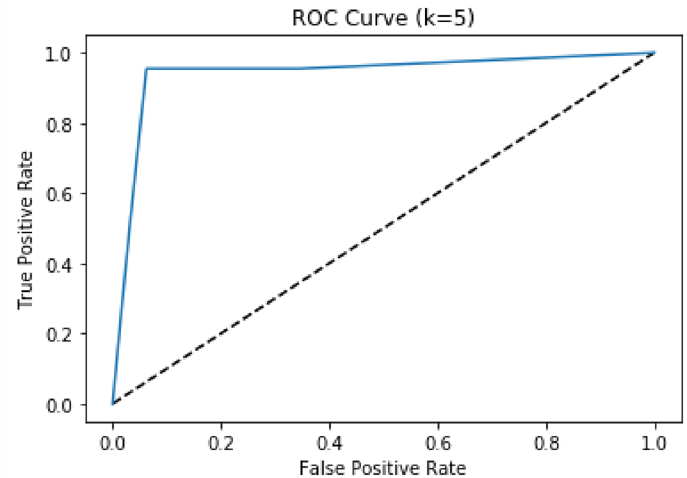
Evaluation

```
In [57]: print(classification_report(y_test, y_pred))
             precision    recall  f1-score   support

          0       0.97      0.94      0.95        32
          1       0.91      0.95      0.93        22

avg / total       0.95      0.94      0.94        54


In [58]: print(accuracy_score(y_test, y_pred))
0.944444444444444
```

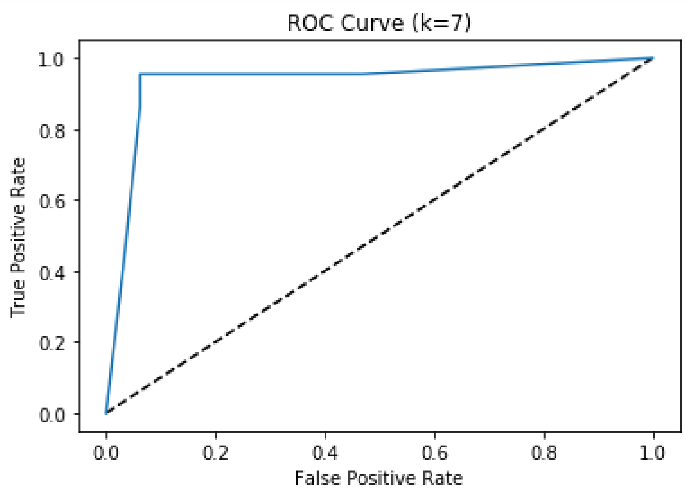
ROC Curves



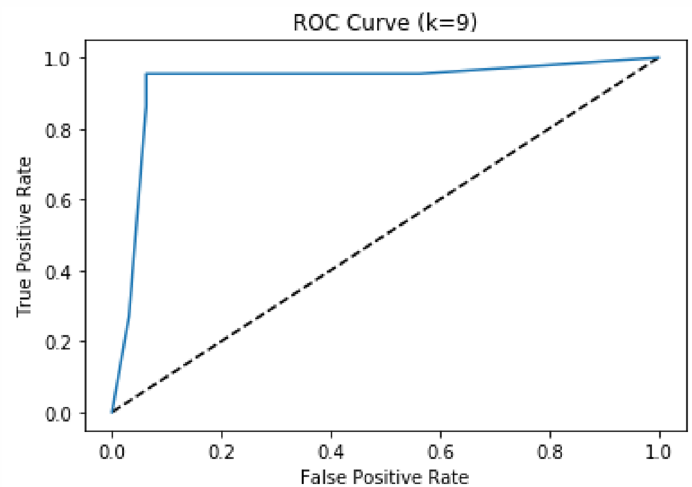
`Out[59]: 0.9517045454545455`



`Out[60]: 0.9403409090909092`



`Out[61]: 0.9332386363636364`



`Out[62]: 0.9268465909090909`

K-NN Regression in Python

- K-NN regressor can be implemented from scikit-learn function KNeighborsRegressor
- KNeighborsRegressor object is instantiated and fit method is called on the object

Syntax :

KNeighborsRegressor(n_neighbors, ...)

Where

n_neighbors: Number of neighbours (k)

Regression Example: Housing Prices

Sales Prices of Houses(Bungalow) in the City of Windsor

- a cross-section from 1987
- number of observations : 546
- Response Variable: Price
- Features: Area(Lot Size), bathrooms, bedrooms, amenities etc.

Program and Output

```
In [1]: import pandas as pd
   ...:
   ...:
   ...: df = pd.read_csv("G:/Statistics (Python)/Cases/Real Estate/Housing.csv")
   ...: dum_df = pd.get_dummies(df.iloc[:,1:11], drop_first=True)
   ...:
   ...: from sklearn.model_selection import train_test_split
   ...: from sklearn.neighbors import KNeighborsRegressor
   ...:
   ...: X = dum_df
   ...: y = df.iloc[:,1]
   ...:
   ...: # Create training and test sets
   ...: X_train, X_test, y_train, y_test = train_test_split(X, y,test_size = 0.3,
   ...:                                                      random_state=2018)
   ...:
   ...: knn = KNeighborsRegressor(n_neighbors=5)
   ...: knn.fit( X_train , y_train )
   ...: y_pred = knn.predict(X_test)
```

Evaluation

```
In [2]: from sklearn.metrics import mean_squared_error,mean_absolute_error,r2_score
   ...: print(mean_squared_error(y_test, y_pred))
937.9804878048791

In [3]: print(mean_absolute_error(y_test, y_pred))
16.536585365853664

In [4]: print(r2_score(y_test, y_pred))
0.9997270312484037
```

Questions?