Building the Models

St. Joseph's University, Bengaluru

Basics of Data Science

Data Modeling

Data Modeling involves the following steps:

1. Model and variable selection
2. Model execution
3. Model diagnostic and model comparision

Model and Variable selection

When doing model and variable selection one must think about:

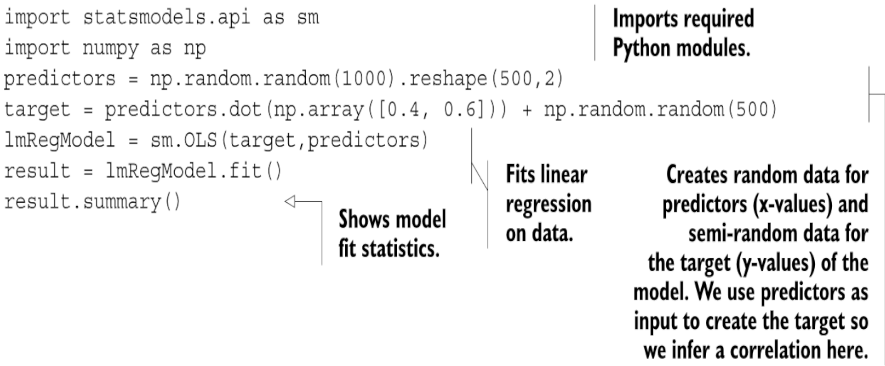
1. Must the model be moved to a production environment and, if so, would it be easy to implement?
2. How difficult is the maintenance on the model: how long will it remain relevant if left untouched?
3. Does the model need to be easy to explain?

Model Execution

After choosing model and required variables, we must implement the model in code.

Model Execution

Example 1



```
import statsmodels.api as sm
import numpy as np
predictors = np.random.random(1000).reshape(500,2)
target = predictors.dot(np.array([0.4, 0.6])) + np.random.random(500)
lmRegModel = sm.OLS(target,predictors)
result = lmRegModel.fit()
result.summary()
```

Imports required Python modules.

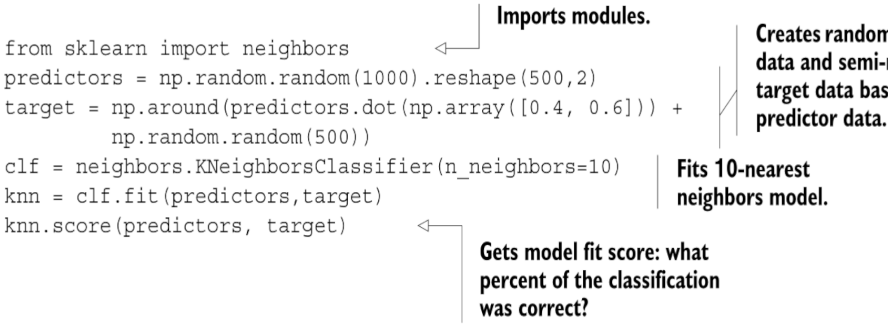
Creates random data for predictors (x-values) and semi-random data for the target (y-values) of the model. We use predictors as input to create the target so we infer a correlation here.

Fits linear regression on data.

Shows model fit statistics.

Model Execution

Example 1



Imports modules.

```
from sklearn import neighbors
predictors = np.random.random(1000).reshape(500,2)
target = np.around(predictors.dot(np.array([0.4, 0.6])) +
         np.random.random(500))
clf = neighbors.KNeighborsClassifier(n_neighbors=10)
knn = clf.fit(predictors,target)
knn.score(predictors, target)
```

Creates random
data and semi-
target data bas
predictor data.

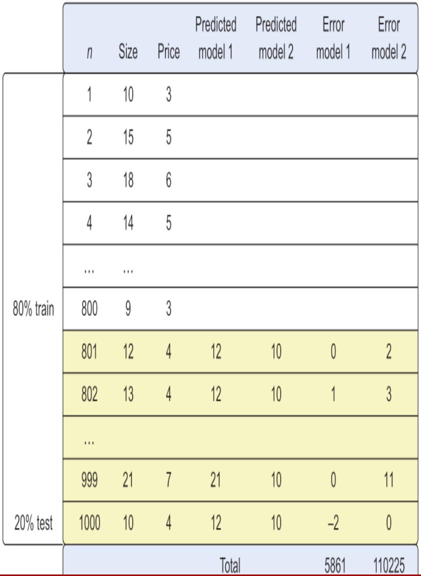
Fits 10-nearest neighbors model.

Gets model fit score: what percent of the classification was correct?

Model Diagnostics

How do you check model performance?
Example:

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(\hat{Y}_i - Y_i)^2$$



	n	Size	Price	Predicted model 1	Predicted model 2	Error model 1	Error model 2
	1	10	3				
	2	15	5				
	3	18	6				
	4	14	5				
	...	...					
80% train	800	9	3				
	801	12	4	12	10	0	2
	802	13	4	12	10	1	3
	...						
	999	21	7	21	10	0	11
20% test	1000	10	4	12	10	–2	0
				Total		5861	110225