**Section 6.3: Classification in Python**

**Duration:** 2.5 hours

**Concepts:**

* Logistic regression
* Bayes classifier
* Linear discriminant analysis
* Quadratic discriminant analysis
* Naive Bayes
* K-nearest neighbours

**Textbook section:** An Introduction to Statistical Learning in Python, Chapter 4 & Chapter 2, section 2.2.3

|  |  |
| --- | --- |
| **Materials and Resources** | **Learning Goals** |
| * Computers for students with Jupyter Notebook * Classification Slides * Classification Exercises Jupyter Notebook file | * Classification methods theory * Bias-variance trade off * Implementation of classification methods in Python |

|  |  |  |
| --- | --- | --- |
| **Duration** | **Lesson Section** | **Learning Objectives** |
| 20 mins | Go through the introduction and logistic regression section of the slide deck. | * Logistic regression model * Odds and log odds * Making predictions * Multiple predictors |
| 20 mins | Go through the Jupyter Notebook Getting Started and Logistic Regression Sections as a class. | * Use `sm.GLM()` to fit a logistic regression * Interpret summary * Use `predict()` to find the training and test error rate |
| 30 mins | Go through the linear discriminant analysis section of the slide deck. | * Bayes classifier * LDA * Confusion matrix * Threshold * ROC |
| 15 mins | Go through the Jupyter Notebook “Linear Discriminant Analysis” section as a class. | * Use `lda.fit()` to fit an LDA model * Interpret the results * `predict()` on test set * Calculate test error rate |
| 8 mins | Go through the Quadratic Discriminant Analysis section of the slide deck. | * LDA vs QDA |
| 15 mins | Go through the Jupyter Notebook Quadratic Discriminant Analysis section as a class. | * Use `qda.fit()` to fit a QDA |
| 5 mins | Go through the Naive Bayes section of the slide deck. | * Naive Bayes |
| 15 mins | Go through the Jupyter Notebook Naive Bayes section as a class. | * Use `NB.fit()` to fit a naive Bayes model |
| 8 mins | Go through the K-Nearest Neighbour section of the slide deck. | * KNN * Indicator variable |
| 15 mins | Go through the Jupyter Notebook K-Nearest Neighbours section as a class. | * Use `fit()` to fit a KNN model * Compare results for different values of K |