**Section 6.8: Support Vector Machines in Python**

**Duration:** 2.5 hours

**Concepts:**

* Maximal margin classifier
* Support vector classifier
* Support vector machine

**Textbook section:** An Introduction to Statistical Learning in Python, Chapter 9

|  |  |
| --- | --- |
| **Materials and Resources** | **Learning Goals** |
| * Computers for students with Jupyter Notebook * Support Vector Machines Slides * Support Vector Machines Exercises Jupyter Notebook file | * Using hyperplanes for binary classification * The concepts for each classifier listed above * How to implement SVMs in Python |

|  |  |  |
| --- | --- | --- |
| **Duration** | **Lesson Section** | **Learning Objectives** |
| 35 mins | Go through the first part of the slide deck until an exercise slide is reached. | * What is a hyperplane? * Binary classification using a separating hyperplane * The maximal margin classifier |
| 20 mins | Go through the “Maximal Margin Classifier” sections in the Jupyter Notebook file as a class. | * Generating data * Fitting a maximal margin classifier with `SVC()` * Plotting the classification * Identifying support vector * Making predictions with the classifier on a test set |
| 15 mins | Go through the support vector classifier section. | * Soft margin * Comparing the maximal margin classifier and the support vector classifier * The tuning parameter C |
| 25 mins | Go through the “Support Vector Classifier” section in the Jupyter Notebook file as a class. | * Fitting a support vector classifier with `SVC()` * Identifying support vectors * What is the `cost` argument * Use the `tune()` function to pick the best `cost` value. |
| 15 mins | Go through the Support Vector Machines section. | * Support vector machine * Kernel * Comparing SVM with support vector classifier * SVM with more than two classes   + One-versus-one   + One-versus-all |
| 20 mins | Go through the “Support Vector Machine” section in the Jupyter Notebook file as a class. | * Fitting an SVM with a radial kernel using `SVC(kernel="rbf")` * Use the `tune()` function to pick the best `cost` and `gamma` values. |