Metadata for Section 3

Section Description: In this section we see a zoo of classifiers, how to train them, when to use them, and how they perform on the Titanic and iris datasets.

|  |
| --- |
| **Metadata**: Spot the problem, highlight it, and design the solution in 3 core steps |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Video Number** | **Video Title** | **Problem / Solution (Not more than 50 words)** | **Step 1 (Not more than 10 words)** | **Step 2(Not more than 10 words)** | **Step 3(Not more than 10 words)** |
| 3.1 | Classifying Data in Python Using the k-Nearest Neighbors (KNN) | How do we train KNN classifiers? We do so using the KNN object from scikit-learn | Import KNN from scikit-learn | Choose k | Train the classifier |
| 3.2 | Working With Decision Trees | How do we train decision trees? We use the DecisionTreeClassifier object from scikit-learn | Import DecisionTreeClassifier from sklearn | Choose the maximum depth of the tree | Train the classifier |
| 3.3 | Machine Learning Using Random Forests | How do we train random forests? We use the RandomForestClassifier object from scikit-learn | Import RandomForestClassifier from sklearn | Choose the maximum tree depth and number of trees | Train the classifier |
| 3.4 | Making Predictions Using the Naïve Bayes Algorithm | How do we train naïve Bayes classifiers? We can use objects from the naïve\_bayes module of scikit-learn, such as BernoulliNB | Decide which flavor of naïve Bayes classification to use | Import the appropriate object from naïve\_bayes | Train the classifier |
| 3.5 | Working with Support Vector Machines (SVM) for Classification and Detection | How do we train SVMs? We use the SVC object from scikit-learn. | Import SVC from sklearn | Choose C and the kernel function to use | Train the classifier |
| 3.6 | Logistic Regression with Machine Learning | How do we perform logistic regression? We use the LogisticRegression object from scikit-learn. | Import LogisticRegression from sklearn | Choose C | Train the classifier |
| 3.7 | Going Beyond Binary | What if our data belongs to multiple classes? We can adapt our classifiers to identifying one of many classes. | Choose a classifier | If necessary, decide between one-vs-one or one-vs-all multiclass schemes | Train the classifier |