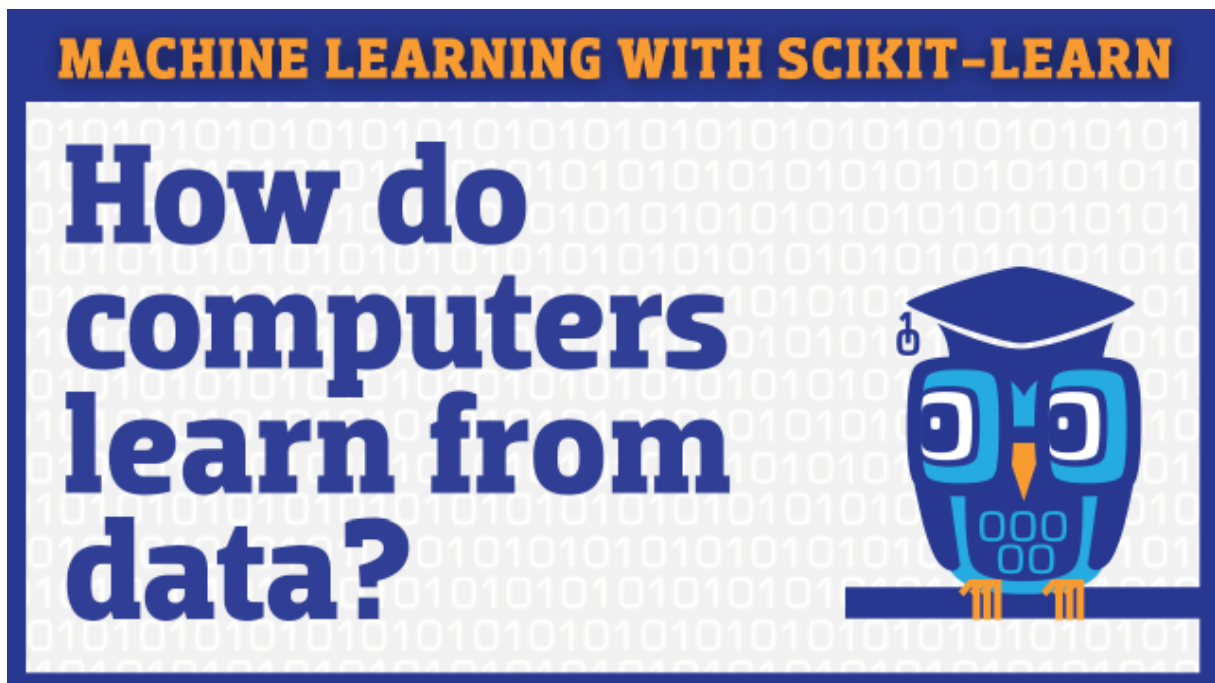

Introduction to Machine Learning with scikit-learn

This video series will teach you how to solve Machine Learning problems using Python's popular scikit-learn library. There are **10 video tutorials** totaling 4.5 hours, each with a corresponding **Jupyter notebook**.

You can watch the entire series on YouTube and view all of the notebooks using nbviewer.

The series is also available as a free online course that includes updated content, quizzes, and a certificate of completion.



Note: The notebooks in this repository have been updated to use Python 3.9.1 and scikit-learn 0.23.2. The original notebooks (shown in the video) used Python 2.7 and scikit-learn 0.16, and can be downloaded from the archive branch. You can read about how I updated the code in this blog post.

Table of Contents

1. What is Machine Learning, and how does it work? (video, notebook)
 - What is Machine Learning?
 - What are the two main categories of Machine Learning?
 - What are some examples of Machine Learning?
 - How does Machine Learning “work”?

-
2. Setting up Python for Machine Learning: scikit-learn and Jupyter Notebook (video, notebook)
 - What are the benefits and drawbacks of scikit-learn?
 - How do I install scikit-learn?
 - How do I use the Jupyter Notebook?
 - What are some good resources for learning Python?
 3. Getting started in scikit-learn with the famous iris dataset (video, notebook)
 - What is the famous iris dataset, and how does it relate to Machine Learning?
 - How do we load the iris dataset into scikit-learn?
 - How do we describe a dataset using Machine Learning terminology?
 - What are scikit-learn's four key requirements for working with data?
 4. Training a Machine Learning model with scikit-learn (video, notebook)
 - What is the K-nearest neighbors classification model?
 - What are the four steps for model training and prediction in scikit-learn?
 - How can I apply this pattern to other Machine Learning models?
 5. Comparing Machine Learning models in scikit-learn (video, notebook)
 - How do I choose which model to use for my supervised learning task?
 - How do I choose the best tuning parameters for that model?
 - How do I estimate the likely performance of my model on out-of-sample data?
 6. Data science pipeline: pandas, seaborn, scikit-learn (video, notebook)
 - How do I use the pandas library to read data into Python?
 - How do I use the seaborn library to visualize data?
 - What is linear regression, and how does it work?
 - How do I train and interpret a linear regression model in scikit-learn?
 - What are some evaluation metrics for regression problems?
 - How do I choose which features to include in my model?
 7. Cross-validation for parameter tuning, model selection, and feature selection (video, notebook)
 - What is the drawback of using the train/test split procedure for model evaluation?
 - How does K-fold cross-validation overcome this limitation?
 - How can cross-validation be used for selecting tuning parameters, choosing between models, and selecting features?
 - What are some possible improvements to cross-validation?
 8. Efficiently searching for optimal tuning parameters (video, notebook)
-

-
- How can K-fold cross-validation be used to search for an optimal tuning parameter?
 - How can this process be made more efficient?
 - How do you search for multiple tuning parameters at once?
 - What do you do with those tuning parameters before making real predictions?
 - How can the computational expense of this process be reduced?

9. Evaluating a classification model (video, notebook)

- What is the purpose of model evaluation, and what are some common evaluation procedures?
- What is the usage of classification accuracy, and what are its limitations?
- How does a confusion matrix describe the performance of a classifier?
- What metrics can be computed from a confusion matrix?
- How can you adjust classifier performance by changing the classification threshold?
- What is the purpose of an ROC curve?
- How does Area Under the Curve (AUC) differ from classification accuracy?

10. Building a Machine Learning workflow (video, notebook)

- Why should you use a Pipeline?
- How do you encode categorical features with OneHotEncoder?
- How do you apply OneHotEncoder to selected columns with ColumnTransformer?
- How do you build and cross-validate a Pipeline?
- How do you make predictions on new data using a Pipeline?
- Why should you use scikit-learn (rather than pandas) for preprocessing?

Bonus Video

At the PyCon 2016 conference, I taught a **3-hour tutorial** that builds upon this video series and focuses on **text-based data**. You can watch the tutorial video on YouTube.

Here are the topics I covered:

1. Model building in scikit-learn (refresher)
2. Representing text as numerical data
3. Reading a text-based dataset into pandas
4. Vectorizing our dataset
5. Building and evaluating a model
6. Comparing models
7. Examining a model for further insight
8. Practicing this workflow on another dataset

9. Tuning the vectorizer (discussion)

Visit this [GitHub repository](#) to access the tutorial notebooks and many other recommended resources.