ML and AI for Software Developers

4 or 5 Days / Instructor-Led / Format: On-Site or Virtual

Contact Us for Pricing

About this Course: Machine learning (ML) expands the boundaries of what's possible by letting software do things that can't be done algorithmically. From fraud detection and sentiment analysis to spam filtering and facial recognition, it touches lives every day. Deep learning is a subset of machine learning that relies on deep neural networks. It is how computers identify objects in images, translate speech in real time, and perform other tasks that would have been impossible just a few short years ago.

Together, these technologies comprise what is popularly known as Artificial Intelligence, or AI. Learn the basics of machine learning and deep learning and discover how they can be used to solve business problems and write software that is smarter than ever before. And do your learning through a combination of lectures and hands-on exercises designed to impart the knowledge you need to skill up on the hottest technologies in software development.

Key Learning Areas:

- Learn what ML and AI are
- Learn what ML and AI can (and cannot) do
- Learn how to build and consume sophisticated ML and AI models
- Learn how to solve business problems with ML and AI
- Learn how to write intelligent software that incorporates ML and AI
- Learn how to use Intelligence-as-a-Service in the form of Azure AI Services

Who Should Attend:

Software engineers

Prerequisites:

Most coding will be done in Python, so familiarity with Python is helpful but not required.

Course Outline:

Introduction to Machine Learning

Learn what machine learning is, what types of machine-learning models there are, and how to use Scikit-learn to build simple unsupervised- and supervised-learning models using algorithms such as *k*-means clustering and *k*-nearest neighbors.

Regression Models

Learn how to build supervised-learning models that predict numeric values such as how long a flight will be delayed or how much a house might sell for. Also learn how to score regression models for accuracy, how to handle categorical values in datasets, and how to use popular learning algorithms such as linear regression, random forests, and gradient-boosting machines (GBMs) in regression models.

Classification Models

Learn how to build classification models that predict categorical outcomes such as whether a credit-card transaction is fraudulent or what number a hand-written digit represents. Also learn about popular classification learning algorithms such as logistic regression, and get acquainted with precision, recall, sensitivity, specificity, confusion matrices, and other metrics used to score classification models.

Text Classification

Learn how to build machine-learning models that classify textual data – for example, models that predict whether an e-mail is "spam" or "not spam" and models that analyze text for sentiment. Also learn about the Naïve Bayes learning algorithm and how to use textual descriptions of products and services to build intelligent recommender systems.

Support-Vector Machines

Support-vector machines (SVMs) represent the cutting edge of statistical machine learning. They often succeed at finding separation between classes when other do not. Take a deep dive into SVMs, learning what they are and how they work. Also learn about hyperparameter tuning and data normalization, and then put your skills to work building an SVM facial-recognition model.

Principal Component Analysis (PCA)

Principal component analysis, or PCA, is one of the minor miracles of machine learning. It's a dimensionality-reduction technique that reduces the number of dimensions in a dataset without sacrificing a commensurate amount of information. You can use it to visualize high-dimensional data, obfuscate data, remove noise, and much more. Learn what PCA is, how it works, and how to apply it, and then put it to work building an anomaly-detection model that predicts failures in bearings.

Operationalizing Machine-Learning Models

Machine-learning models built in Python are easily consumed in Python apps but consuming them in other languages such as C# and Java is not so straightforward. Learn about ONNX (Open Neural Network Exchange) and other ways to operationalize ML models so they can be consumed by any application, regardless of platform or programming language. Also see an alternative way to build ML models in C# using ML.NET, and novel way to add machine-learning capabilities to Microsoft Excel.

Introduction to Deep Learning

Deep learning is a subset of machine learning that relies primarily on deep neural networks. Learn what neural networks are, how they work, and why they are continually advancing the state of the art in AI.

Neural Networks

Learn how to use Keras and TensorFlow to build and train sophisticated neural networks that perform regression and classification. Also learn how to save and load trained models, how to use dropout layers to combat overfitting, and how to use Keras's callbacks API to customize the training process.

Image Classification with Convolutional Neural Networks (CNNs)

State-of-the-art image classification typically isn't done with traditional neural networks. Rather, it is performed with Convolutional Neural Networks (CNNs), which excel at computer-vision tasks such as identifying objects in images. Learn what CNNs are and how they work and learn how to use transfer learning to build sophisticated CNNs to solve domain-specific problems.

Facial Recognition and Object Detection

Convolutional neural networks can do more than just classify images; they play an important role in modern facial-recognition and object-detection systems, too. Learn how self-driving cars use advanced CNNs to recognize objects around them, and how to build an end-to-end facial recognition system by combining CNNs with popular face-detection algorithms.

Natural Language Processing

Deep neural networks have grown in sophistication to the point that they can process text and speech – sometimes more accurately than humans. Learn how to use Keras and TensorFlow to build and train neural networks that classify text, translate text, and perform other feats of NLP magic.

Azure AI Services

Azure AI Services is a set of cloud-based services and APIs for building intelligent applications in any language and on any platform. Backed by state-of-the-art ML and AI algorithms, they can caption images, analyze text for sentiment, determine whether a photo contains inappropriate content, translate speech in real time, and much more. Learn how to use Azure AI Services to add intelligence to your apps, and then build a Web site that extracts text from uploaded images and translates that text into the language of your choice.

Setup Instructions

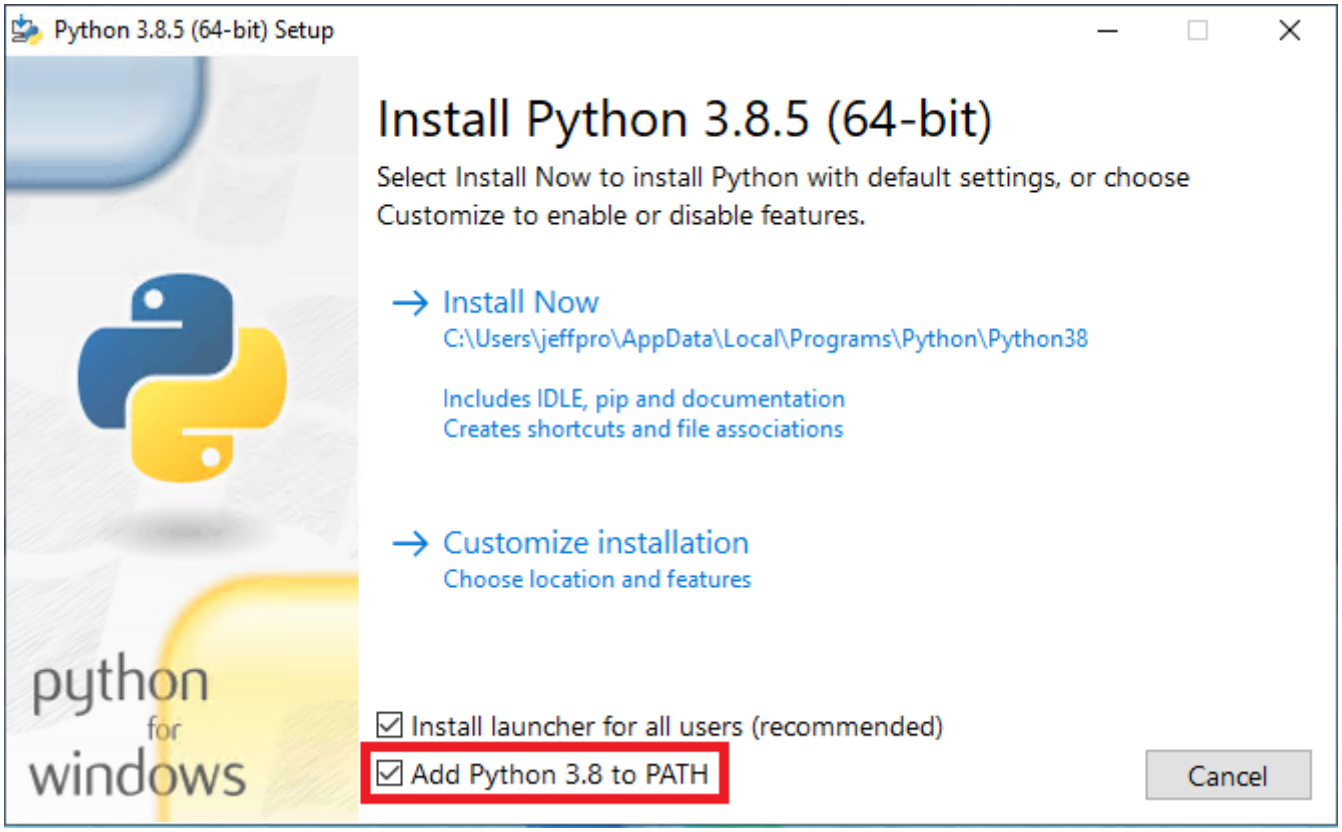
Most of the labs in this course run in Jupyter notebooks, which provide an interactive environment for executing Python code in your browser. There are several ways to get up and running with Jupyter notebooks. Some are cloud-based and require nothing to be installed on your computer. Others require local installs. In order to work the labs, choose one of the following options.

Option #1: Install Python and Jupyter Notebooks on your computer (recommended)

Running Python and Jupyter notebooks locally requires some setup on your computer, but it's the best way to do machine learning. Here are the steps:

1. Install the latest 64-bit version of Python 3 on your computer if it isn't already installed. **Be sure to install the 64-bit version of Python**, because some of the labs use the TensorFlow deep-learning framework, and TensorFlow doesn't run on 32-bit

machines. If the installer offers you the option to "Add Python 3.**x** to PATH" as shown below, check the box to answer yes. The box will probably be unchecked by default.



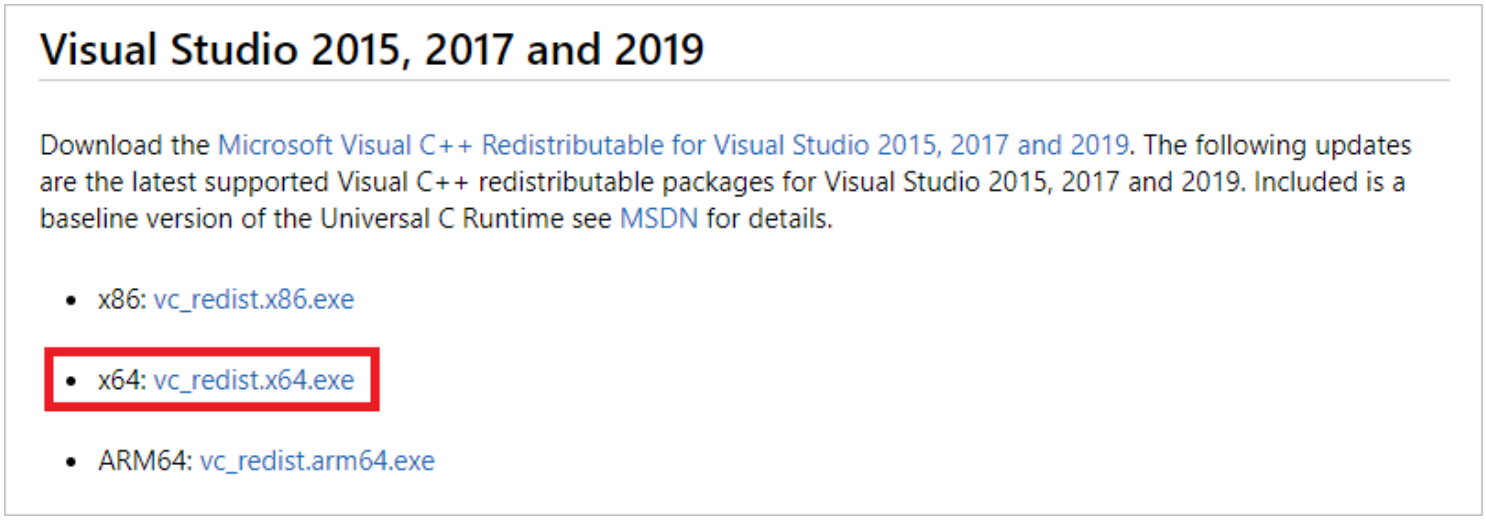
2. Open a Command Prompt or terminal window and execute the following commands in the order shown:

```
pip install notebook
pip install scikit-learn
pip install pandas
pip install matplotlib
pip install seaborn
pip install flask
pip install requests
pip install tensorflow
pip install keras
```

These commands install several Python packages that you'll need to build machine-learning models and neural networks. Some will take a few minutes to complete, even with a fast Internet connection. So be patient!

3. On Windows, TensorFlow requires the 64-bit version of the "Visual Studio 2015, 2017 and 2019 Redistributable" DLL. If your computer runs Windows, the DLL may already be installed. But just in case it's not, go to https://support.microsoft.com/en-

us/help/2977003/the-latest-supported-visual-c-downloads, find the link for the installer for the 64-bit DLL, and click it to install the DLL.



Visual Studio 2015, 2017 and 2019

Download the Microsoft Visual C++ Redistributable for Visual Studio 2015, 2017 and 2019. The following updates are the latest supported Visual C++ redistributable packages for Visual Studio 2015, 2017 and 2019. Included is a baseline version of the Universal C Runtime see MSDN for details.

- x86: vc_redist.x86.exe
- x64: vc_redist.x64.exe
- ARM64: vc_redist.arm64.exe

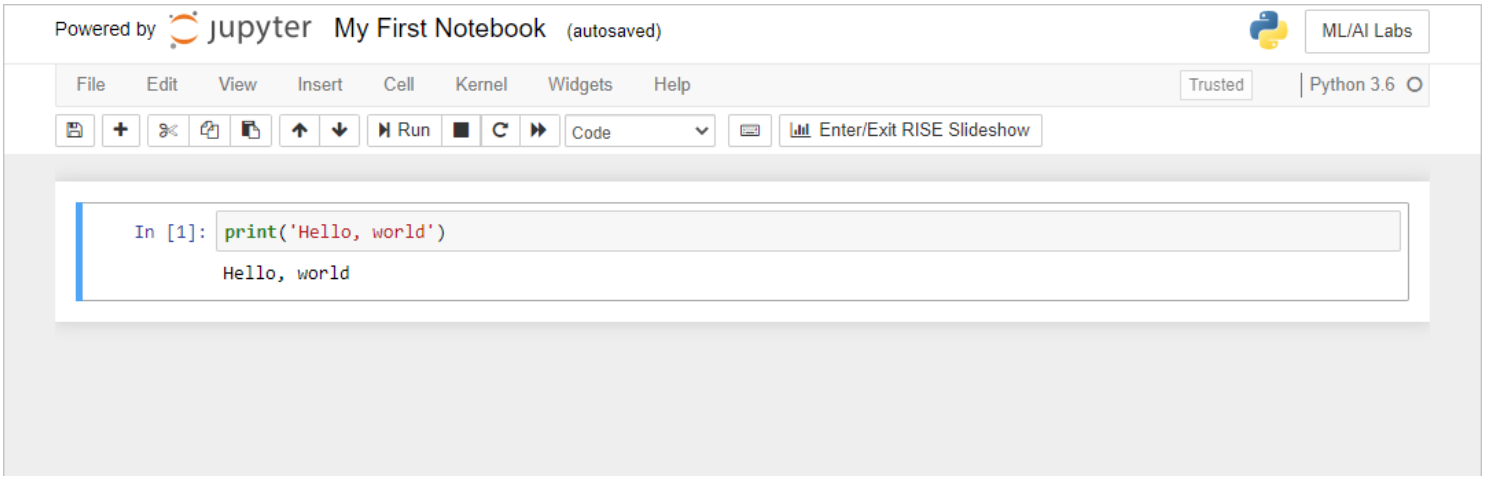
4. Once everything is installed, you can launch Jupyter notebooks by navigating to the directory where your notebooks are stored (or where you want to create new notebooks) and executing the following command:

```
jupyter notebook
```

This launches the Jupyter Notebooks run-time in your browser. You can validate your environment by creating a Jupyter notebook and executing the following statement in a notebook cell:

```
print('Hello, world')
```

If the output is "Hello, world" as pictured below, then Jupyter notebooks are installed and ready to go.



Option #2: Use Visual Studio Code

If you're a fan of Visual Studio Code, you can use it to run Jupyter notebooks rather than use the native Jupyter Notebooks interface. Here's how:

1. Follow the instructions in the previous section to install 64-bit Python, Jupyter Notebooks, and other required packages.

2. Go to https://code.visualstudio.com/ and install Visual Studio Code if it isn't already installed.

3. Go to https://marketplace.visualstudio.com/items?itemName=ms-python.python and install the Python extension for Visual Studio Code.
   Once this is done, you can load, create, edit, and run Jupyter notebooks inside Visual Studio Code and benefit from IntelliSense, Git integration, and more.

Option #3: Use Google Colaboratory ("Colabs")

Colabs are Google's free implementation of Jupyter notebooks in the cloud. Colabs offer a friction-free way to run Jupyter notebooks with common Python packages such as Pandas, Scikit-learn, and TensorFlow already installed. They also include free GPU support and require nothing to be installed on your computer. To get started, go to https://colab.research.google.com/notebooks/intro.ipynb and sign in with your Google account. If you don't have a Google account, you can create one for free.

There are two drawbacks to using Colabs for this course. First, you'll have to upload and download various files throughout the course of the labs rather than simply copy them from one directory to another on your computer. Second, two of the labs in this course require a local Python-based machine-learning environment. If you use Colabs, you'll need to pair up with someone else to work those two labs.