Python4ML

An open-source course for everyone

Team: James Hopkins | Brendan Sherman | Zachery Smith | Eric Wynn
Client: Amirsina Torfi
Instructor: Dr. Edward Fox
CS 4624: Multimedia, Hypertext, and Information Access
Virginia Tech, Blacksburg VA 24061

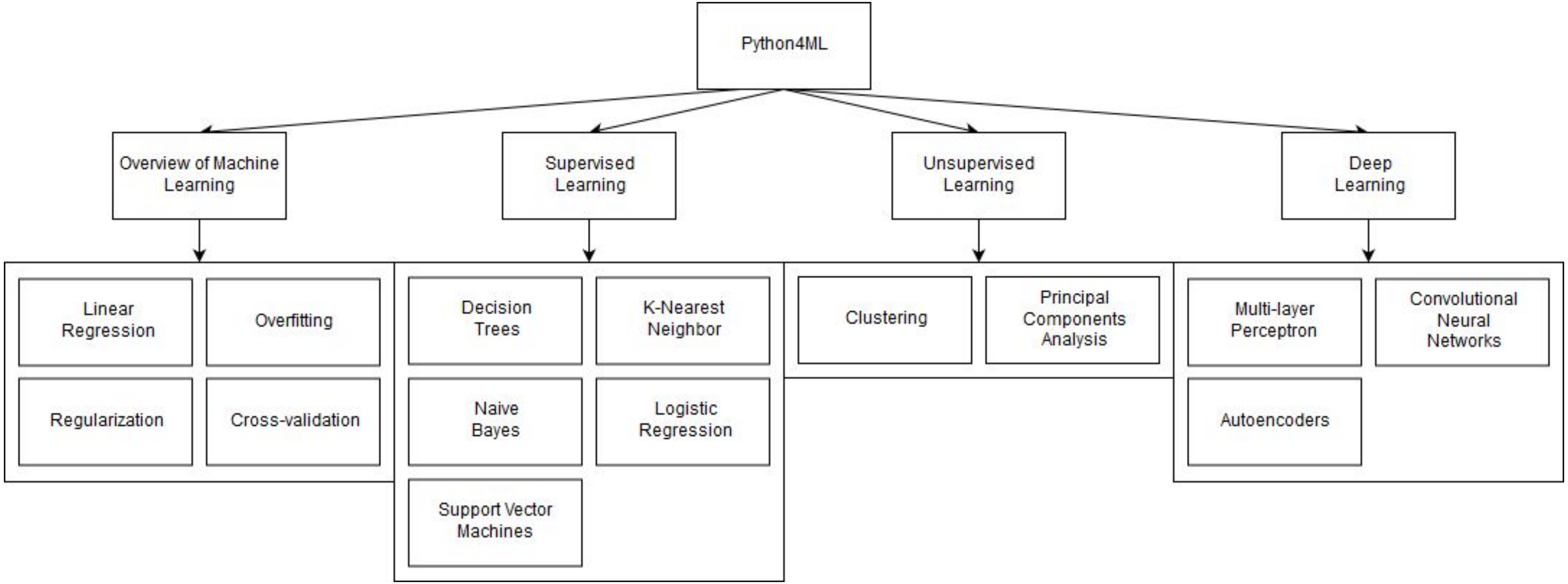
5/12/2019

Outline

- Summary
- Deliverables
- Documentation
- Testing Plans
- Post-Semester Work
- Lessons Learned

Summary

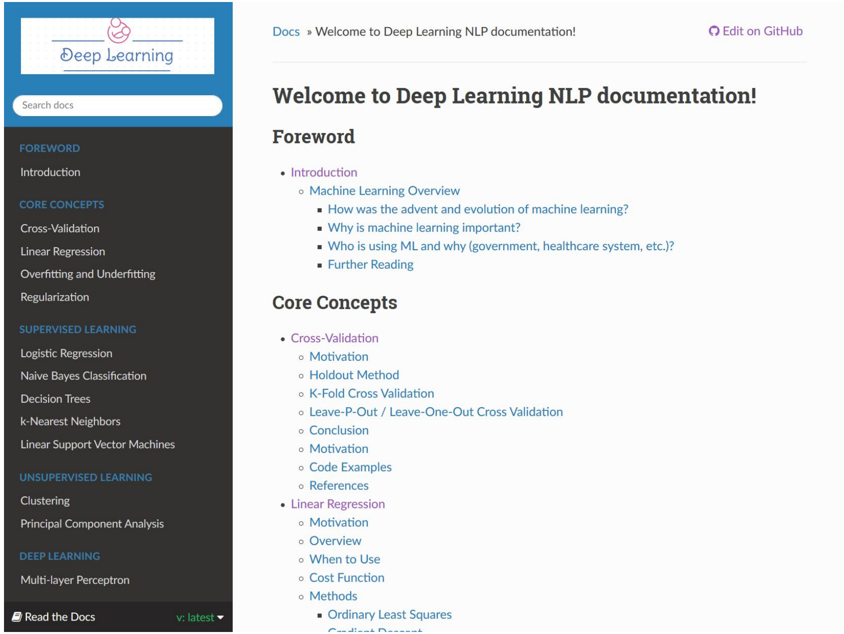
Creating an open-source guide to Machine Learning using Python



Deliverables

Course website

- Contains all module write-ups
- Entire site can be downloaded as PDF or other formats
- Links to code examples on GitHub



FOREWORD

Introduction

CORE CONCEPTS

Cross-Validation

Linear Regression

Overfitting and Underfitting

Regularization

SUPERVISED LEARNING

Logistic Regression

Naive Bayes Classification

Decision Trees

k-Nearest Neighbors

Linear Support Vector Machines

UNSUPERVISED LEARNING

Clustering

Principal Component Analysis

DEEP LEARNING

Multi-layer Perceptron

Docs » Welcome to Deep Learning NLP documentation!

Welcome to Deep Learning NLP documentation!

Foreword

- Introduction
  - Machine Learning Overview
    - How was the advent and evolution of machine learning?
    - Why is machine learning important?
    - Who is using ML and why (government, healthcare system, etc.)?
    - Further Reading

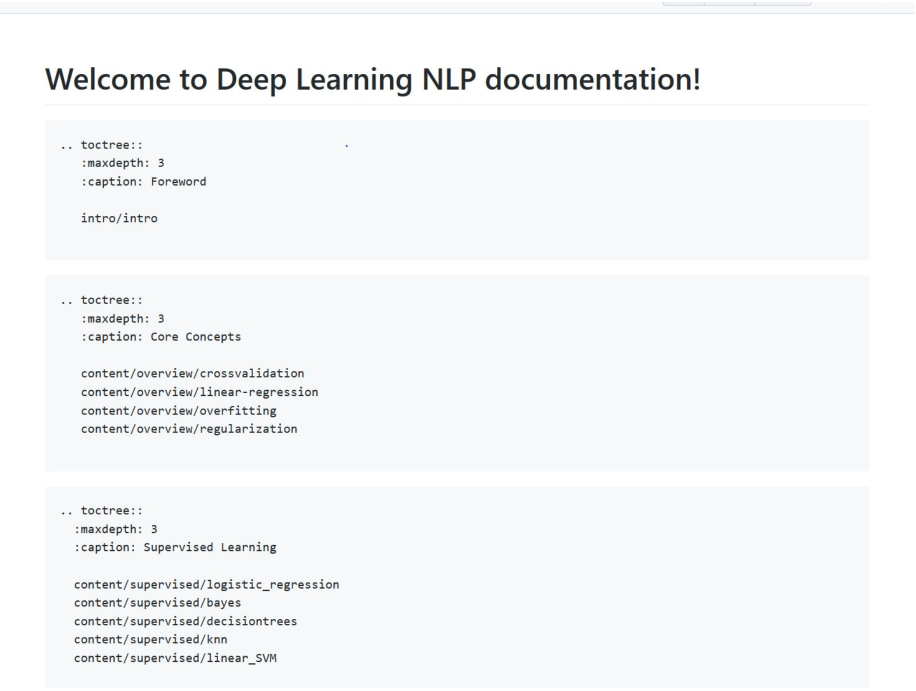
Core Concepts

- Cross-Validation
  - Motivation
  - Holdout Method
  - K-Fold Cross Validation
  - Leave-P-Out / Leave-One-Out Cross Validation
  - Conclusion
  - Motivation
  - Code Examples
  - References
- Linear Regression
  - Motivation
  - Overview
  - When to Use
  - Cost Function
  - Methods
    - Ordinary Least Squares
    - Gradient Descent

Deliverables

GitHub repository

- All course content is open-source
- Anyone can contribute and suggest changes
- Highly structured system for files



Welcome to Deep Learning NLP documentation!

```
.. toctree::
   :maxdepth: 3
   :caption: Foreword

   intro/intro
```

```
.. toctree::
   :maxdepth: 3
   :caption: Core Concepts

   content/overview/crossvalidation
   content/overview/linear-regression
   content/overview/overfitting
   content/overview/regularization
```

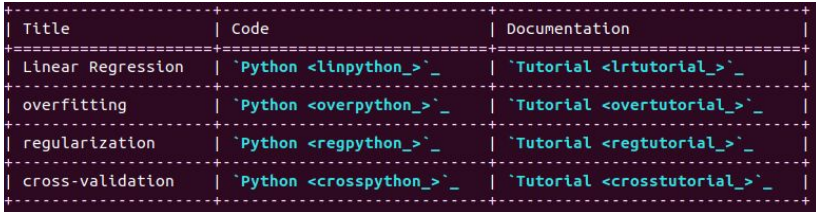
```
.. toctree::
  :maxdepth: 3
  :caption: Supervised Learning

  content/supervised/logistic_regression
  content/supervised/bayes
  content/supervised/decisiontrees
  content/supervised/knn
  content/supervised/linear_SVM
```

Documentation

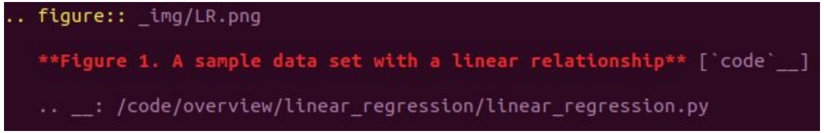
Documentation written in reStructuredText (rST), a form of markup

Seamless integration with Sphinx



```
+----------------------+-----------------------------+--------------------------------+
| Title                | Code                        | Documentation                  |
+======================+=============================+================================+
| Linear Regression    | `Python <linpython_>`_      | `Tutorial <lrtutorial_>`_      |
+----------------------+-----------------------------+--------------------------------+
| overfitting          | `Python <overpython_>`_     | `Tutorial <overtutorial_>`_    |
+----------------------+-----------------------------+--------------------------------+
| regularization       | `Python <regpython_>`_      | `Tutorial <regtutorial_>`_     |
+----------------------+-----------------------------+--------------------------------+
| cross-validation     | `Python <crosspython_>`_    | `Tutorial <crosstutorial_>`_   |
+----------------------+-----------------------------+--------------------------------+
```

Clean tables

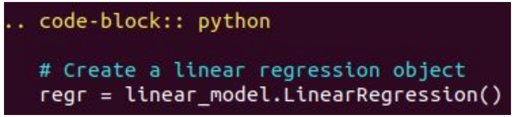


```
.. figure:: _img/LR.png

   **Figure 1. A sample data set with a linear relationship** [`code`__]

   .. __: /code/overview/linear_regression/linear_regression.py
```

Annotated, linked figures



```
.. code-block:: python

   # Create a linear regression object
   regr = linear_model.LinearRegression()
```

Embeddable code

Demo

https://machine-learning-course.readthedocs.io/en/latest/

Project Stats

- 6500+ lines of content
- 100+ pages of course documentation (another 60+ in the final report)
- 70+ sources
- 25+ unique python examples

Testing Plans

We have been testing scripts as they are added, but the whole team will come together to re-test each one for the final deliverable.

Every navigation link in the site needs to be tested.

We had sample users go through the modules and provide us with feedback.

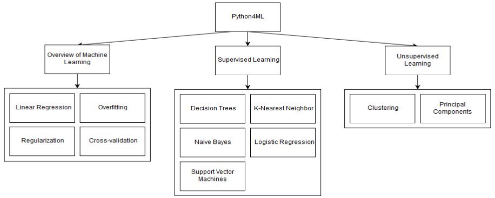
End of Semester Plans

- Complete the last two topics on Neural Networks
- Build final site, test navigation, and test every script
- Get feedback from user testing

Python4ML

A Machine Learning Course for Everyone

Topics


Python4ML

Overview of Machine Learning

Supervised Learning

Unsupervised Learning

Linear Regression

Overfitting

Regularization

Cross-validation

Decision Trees

K-Nearest Neighbor

Naive Bayes

Logistic Regression

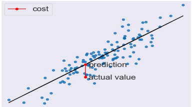
Support Vector Machines

Clustering

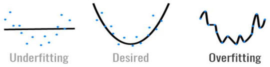
Principal Components

Overview

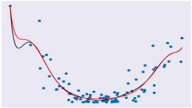
Linear Regression



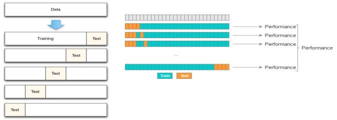
Overfitting / Underfitting



Regularization

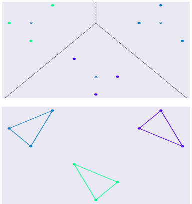


Cross Validation

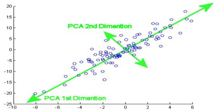


Unsupervised Learning

Clustering



Principal Component Analysis



WHAT?

Python4ML is an open-source course for machine learning using the Python programming language.

HOW?

The course is made up of reStructuredText documents and example programs written in Python, using libraries such as scikit learn.

WHO?

This course is being developed with Virginia Tech's Open Source for Science organization, led by our client Amirsina Torfi.

WHERE?

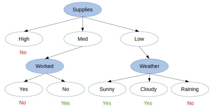
The course is available on GitHub. The code and live course site can be found by scanning the QR code on this poster.

WHY?

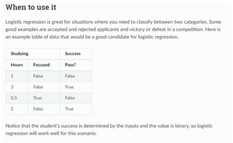
The course is aimed at those with little knowledge of machine learning. We want to facilitate education in an open-source context, bringing important topics together in a high-level overview of ML.

Supervised Learning

Decision Trees



Logistic Regression



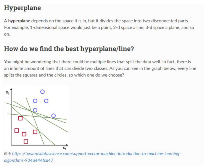
When to use it

Logistic regression is great for situations where you need to classify between two categories. Some good examples are accepted and rejected applicants and victory or defeat in a competition. Here is an example table of data that would be a good candidate for logistic regression.

Studying		Success
Hours	Focused	Pass?
1	False	False
3	False	True
0.5	True	False
2	False	True

Notice that the student's success is determined by the inputs and the value is binary, so logistic regression will work well for this scenario.

Support Vector Machines



Hyperplane

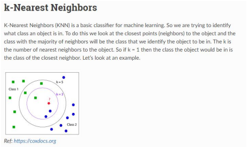
A **hyperplane** depends on the space it is in, but it divides the space into two disconnected parts. For example, 1-dimensional space would just be a point, 2-d space a line, 3-d space a plane, and so on.

How do we find the best hyperplane/line?

You might be wondering that there could be multiple lines that split the data well. In fact, there is an infinite amount of lines that can divide two classes. As you can see in the graph below, every line splits the squares and the circles, so which one do we choose?

Ref: https://towardsdatascience.com/support-vector-machine-introduction-to-machine-learning-algorithms-934a444fca47

K-Nearest Neighbors

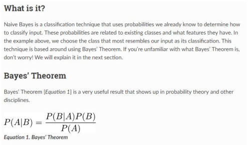


k-Nearest Neighbors

K-Nearest Neighbors (KNN) is a basic classifier for machine learning. So we are trying to identify what class an object is in. To do this we look at the closest points (neighbors) to the object and the class with the majority of neighbors will be the class that we identify the object to be in. The k is the number of nearest neighbors to the object. So if k = 1 then the class the object would be in is the class of the closest neighbor. Let's look at an example.

Ref: https://coxdocs.org

Naive Bayes



What is it?

Naive Bayes is a classification technique that uses probabilities we already know to determine how to classify input. These probabilities are related to existing classes and what features they have. In the example above, we choose the class that most resembles our input as its classification. This technique is based around using Bayes' Theorem. If you're unfamiliar with what Bayes' Theorem is, don't worry! We will explain it in the next section.

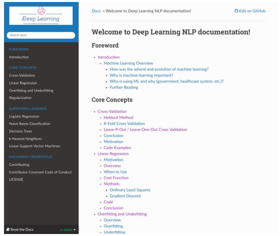
Bayes' Theorem

Bayes' Theorem [*Equation 1*] is a very useful result that shows up in probability theory and other disciplines.

$$P(A|B) = \frac{P(B|A)P(B)}{P(A)}$$

Equation 1. Bayes' Theorem

Live Course Site



Deep Learning

Search docs

FOREWORD

Introduction

CORE CONCEPTS

Cross-Validation

Linear Regression

Overfitting and Underfitting

Regularization

SUPERVISED LEARNING

Logistic Regression

Naive Bayes Classification

Decision Trees

k-Nearest Neighbors

Linear Support Vector Machines

DOCUMENT CREDENTIALS

Contributing

Contributor Covenant Code of Conduct

LICENSE

Read the Docs v: latest

Docs » Welcome to Deep Learning NLP documentation! Edit on GitHub

Welcome to Deep Learning NLP documentation!

Foreword

- Introduction
  - Machine Learning Overview
    - How was the advent and evolution of machine learning?
    - Why is machine learning important?
    - Who is using ML and why (government, healthcare system, etc.)?
    - Further Reading

Core Concepts

- Cross-Validation
  - Holdout Method
  - K-Fold Cross Validation
  - Leave-P-Out / Leave-One-Out Cross Validation
  - Conclusion
  - Motivation
  - Code Examples
- Linear Regression
  - Motivation
  - Overview
  - When to Use
  - Cost Function
  - Methods
    - Ordinary Least Squares
    - Gradient Descent
  - Code
  - Conclusion
- Overfitting and Underfitting
  - Overview
  - Overfitting
  - Underfitting

Technologies

Learn More

Team: James Hopkins, Brendan Sherman, Zachery Smith, and Eric Wynn

Client: Amirsina Torfi,
Head of Open Source for Science @ VT

4/30/2019
Instructor: Edward A. Fox

Scan me

Post-Semester Work

This course can be expanded outside this capstone or in future semesters by:

- Writing more example scripts and content
- Integrating the project with Docker:
  - A containerized system, similar to virtual machines
  - Can pre-package dependencies together for each section and provide users with a single command to run scripts
  - Users won’t need python **OR** dependencies installed

Lessons Learned / Takeaways

- Importance of high quality documentation
    - As important as or more important than actual code
    - Makes reviews faster and documents easier to understand
- Teaching about a subject requires deep understanding
    - Had to know why specific decisions were made at every step
    - Required lots of research on the topics

Lessons Learned / Takeaways

- Importance of finishing personal assignments in a timely manner
    - Several reviewing stages that required individual approval
    - Delayed when someone doesn't respond or approve changes

Acknowledgements

Client: Amirsina Torfi

- PhD student in Computer Science
- Focused on deep learning, neural networks

Scikit-Learn

- Open-source Python Machine Learning library