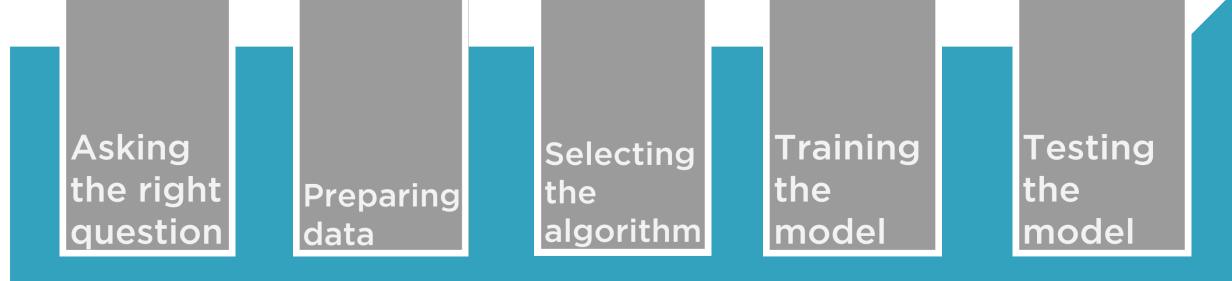
Machine Learning Workflow



Machine Learning Workflow

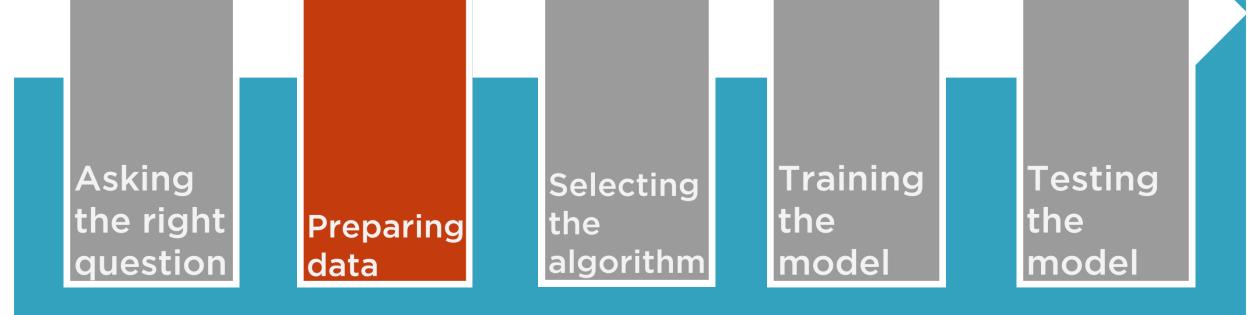
Asking the right question

Preparing data

Selecting the algorithm Training the model

Testing the model

Machine Learning Workflow



Overview



Find the data we need

Inspect and clean the data

Explore the data

Mold the data to Tidy data

Demos in Python in Jupyter Notebook



Tidy Data

Tidy datasets are easy to manipulate, model and visualize, and have a specific structure:

each variable is a column,

each observation is a row,

each type of observational unit is a table.

50-80% of a ML project is spent getting, cleaning, and organizing data



Getting Data

Google

Government databases

Professional or company data sources

Your company

Your department

All of the above

Pima Indian Diabetes Data

Originally from UCI Machine Learning Repository

pima-data.csv - in demo folder, based on UCI data

Female patients at least 21 years old

768 patient observation rows

10 columns

9 feature columns

Number of pregnancies, blood pressure, glucose, insulin level, ...

1 class column

Diabetes - True or False



Closer the data is to what you are predicting, the better



Data will never be in the format you need



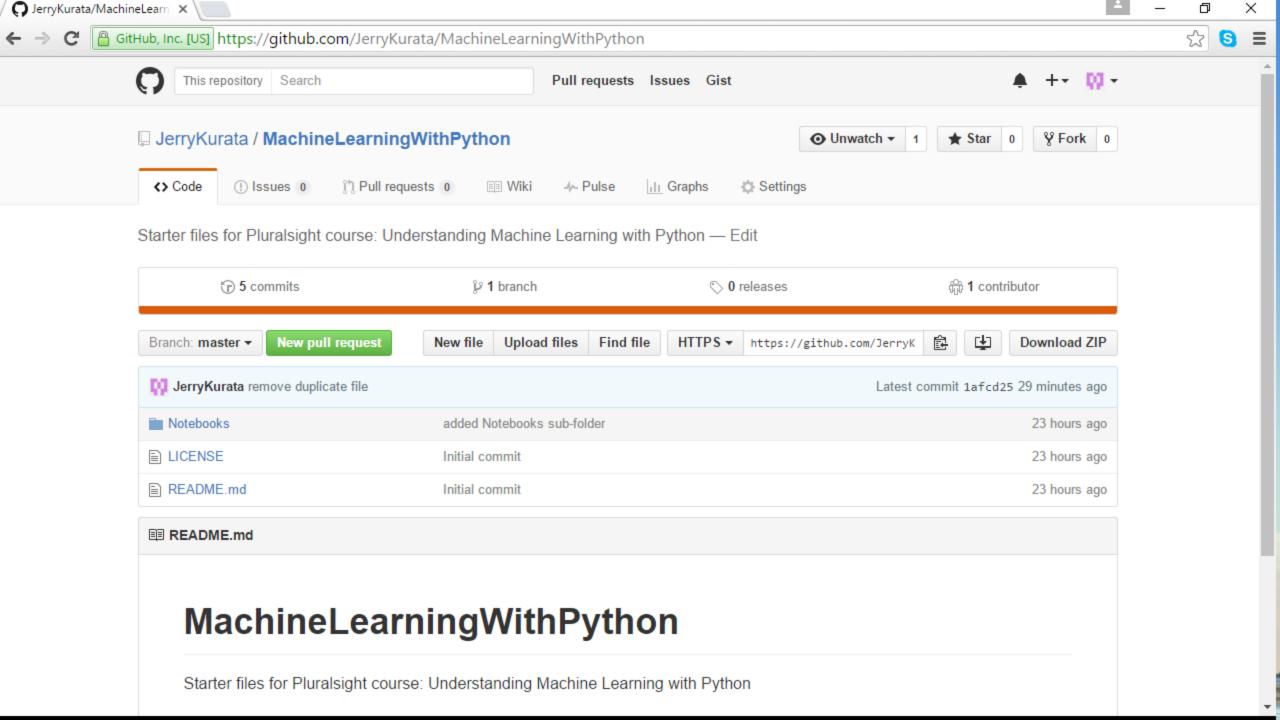
Getting Data and Notebooks

Modified version of Pima Indian Diabetes Data

Notebooks from course

http://bit.ly/ml_python





Demo



Loading Data

Exploring Data

Cleaning Data



Columns to Eliminate

Not used

No values

Duplicates



Correlated Columns

Same information in a different format

- ID and value associated with ID

Add little information

Can cause algorithms to get confused

```
Price = x * Area(sq ft)+ y * Area(sq m)+ z * # of rooms
```



Molding Data

Adjusting data types

Creating new columns, if required



Dealing with missing data

Ignore it

- Algorithms may fail

Impute it - update to "reasonable" values

- Most frequent
- Mean
- Median
- Expert reasonable value



Accurately predicting rare events is difficult



Track how you manipulate data



Change Tracking

Jupyter Notebook

Python Interpreter interaction stored via code cells

Documentation stored via markup cells

Still need source code management (Git, TFS, SVN, etc.)



Summary



Use Pandas to read in demo data

Identified correlated features

Cleaned data

Molded data

Checked True/False ratio

Discussed data rules

