DataSci 420

Lesson 1

Seth Mottaghinejad

today's agenda

- about me and my teaching style
- assignments, quizzes and milestones
- participation and grading
- grading expectations and supplimentary material
- coding environment
- overview of machine learning

about me

- I have been working in data science / analytics for over 10 years
- background in statistics, self-taught programmer
- worked across many industries
- I love teaching and include a lot of **hands-on** work
- on average, we spend about 40% on lecture and 60% on hands-on
- let's take frequent short breaks to fit online format

assignments, quizzes and milestones

- assignments are due by **11:59 PM each Sunday following lecture**
- I will make **no exceptions** about assignment due dates
- quizzes are taken during the lecture and **answered in chat window**
- assignments and milestones are graded by our grader
- milestones are similar to assignment but more project-oriented
- questions about the assignments should be posted on the **discussion board** on Canvas, where I or grader will answer them (or students if they wish)

grading expectations

- please use the discussion boards for questions on assignment & milestones
- when necessary, it's generally OK to make an assumption about the assignment reqs - **as long as you state your assumption**
- more important to be on time than perfect:
  - meet the reqs with working code
  - write good comments for your code
  - write good explanations showing your line of thinking

participation and grading

activity	what you need to do	grade
participation	be active in **discussion boards**	16%
quizzes	answer **in chat window** during lecture	22%
assignments	submit by **11:59 PM every Sunday**	22%
milestone 1	due same time as assignment 4	10%
milestone 2	due same time as assignment 7	10%
milestone 3	due same time as assignment 10	20%

break time

coding environment

- install Anaconda (use **Python 3.7**), which installs everything we need
- we will be using browser-based Jupyter notebooks as our Python environment
- basics of Jupyter notebooks:
  - code cells and Markdown cells
  - running and re-running code cells - **order matters!**
  - magics are very useful shortcuts

break time

overview of data formats

- different sources of data?
  - **tabular (structured)**: relational tables (`SQL`), matrices, `DataFrame` , ...
  - **semi-structured**: `JSON` , `XML` , `Mongo DB` , graph datatabases, ...
  - **unstructured**: raw text, images, sound, video, ...
- most ML algorithms only work with tabular data
- once data is made tabular, we still need to do a lot of pre-processing prior to ML

lab time

- here's an example of a single record in semi-structured data:

```
{
  author : [Walter, Habib],
  title : "How I learned to drink water",
  language : "Eng",
  year_pulished : [2008, 2012]
}
```

- represent it using tabular format (there is more than one way)
- propose extra columns that can be *extracted* from the above data

what is machine learning?

- an algorithm is a self-contained set of **rules or instructions** used to *solve problems*
- **machine learning** is the field of study that gives computers the ability to learn *without being explicitly programmed*, by using data to learn
- the *problems* ML algorithms try to solve are usually
  - prediction: **supervised learning**
  - finding structure in data: **unsupervised learning**
  - ruling over humans: **reinforcement learning** (not covered here)

supervised learning

- also called **predictive modeling** or **inference**
- look at some current examples (**labeled data**) and find a model that can predict future examples (**unlabeled data**)
- the **target variable** or **label** is we want to predict
  - **regression** algorithms are used with a **numeric target**
  - **classification** algorithms are used with a **categorical target**
- by comparing predictions with the actual labels, we can evaluate our model's accuracy (hence the term *supervised*)

simplified example: credit rating

- **data**: age, income, credit score (300-800)
- **algorithm**: linear regression

  ```
  a + b * age + c * income + some error = credit score
  ```

- after feeding it data (called **training**)

  ```
  5.8 + 3.9 * age + 1.98 * income + error = credit score
  ```

- prediction equation (used for **scoring**)

  ```
  5.8 + 3.9 * age + 1.98 * income = predicted credit score
  ```

in pseudo-code

- in Python we call `.fit()` to train and `.predict()` to score
- choose the algorithm:

```
lin_reg = LinearRegression(alpha = .005)
```

- train the algorithm on data

```
lin_reg.fit(X_train, y_train)
```

- predict on any new data

```
lin_reg.predict(X_test)
```

break time

everyone has their jargon

- **rows** - observation, example, sample, record, data points, item, instance
- **columns** - variables, attributes, properties, features, fields, dimensions
  - **target** - label, response variables, dependent variable, outcome
  - **features** - explanatory variable, independent variable, predictors, covariates
    - **numeric features**: dates, counts, amounts, etc.
    - **categorical features**: grouping variables, identifiers

lab time

- a basic example of an algorithm is a recipe
- however, a recipe is not a machine learning algorithm because the instructions are explicit and rely on expert knowledge
- what would it look like if we used ML to **learn** the recipe for something simple, like **guacamole**
  - what would the data look like? think of some features?
  - what would be the target?
- to simplicity things, assume that all ingredients are known (or limited), but their amount isn't

supervised learning algorithms

- **linear regression**: regression
- **logistic regression**: classification (even though it's called logistic *regression*)
- **tree-based algorithms**: more commonly for classification
- **support vector machines** (SVMs): binary classification
- **neural networks**: classification and regression
  - **deep neural networks** (deep learning)
    - image recognition and natural language processing (NLP)

break time

unsupervised learning

- also called data-mining / pattern recognition / structure discovery
- look at **unlabeled data** and find general patterns
- more subjective and difficult to evaluate and interpret, and hence it is far **less common** than supervised learning
- **clustering** is the most common example
  - k-means clustering
  - variable clustering / dimensionality reduction
  - word clouds

lab time

- let's say we have an guacamole dataset like the one we proposed in the prior lab, with or without the target
- what would it look like if we used *unsupervised* ML to discover some patterns in the data
  - what would an example of a pattern look like?
  - should we include the target or not?
- to simplicity things, assume that all ingredients are known (or limited), but their amount isn't

mixing it up

- in practice, the line between supervised and unsupervised ML can get blurry, for example
  - you can use k-means clustering to cluster the features in your data, then use the cluster itself as one of the feature to predict the target
  - you can run a clustering algorithm on labeled data and include the target itself in order to understand how it relates to the other features in the data

the end