DataSci 207 – Applied Machine Learning

Cornelia Paulik, PhD

School of Information

UC Berkeley

Linear Regression – gradient descent

Announcements

- Live session recordings are available in bCourses by end of Thursday
- Live session PP slides are available in my GitHub repo by end of Thursday
- Answer questions in my Slack channel – counts towards participation
- Assignments – start early, don’t wait until the last two days
- Anything else?

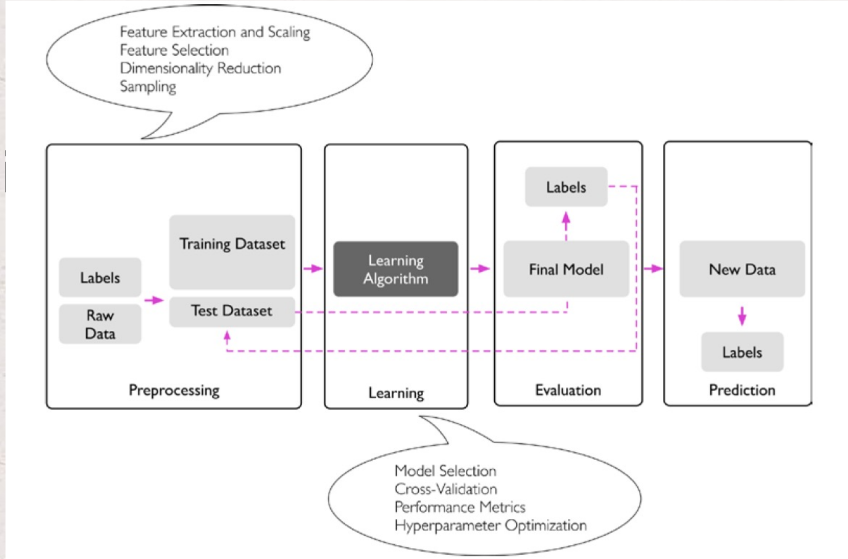
Questions on Final Project

- How to select the data?
- Do we know enough about modeling by the time we give the baseline presentation?
- How to divide tasks within the team?

Questions on Final Project

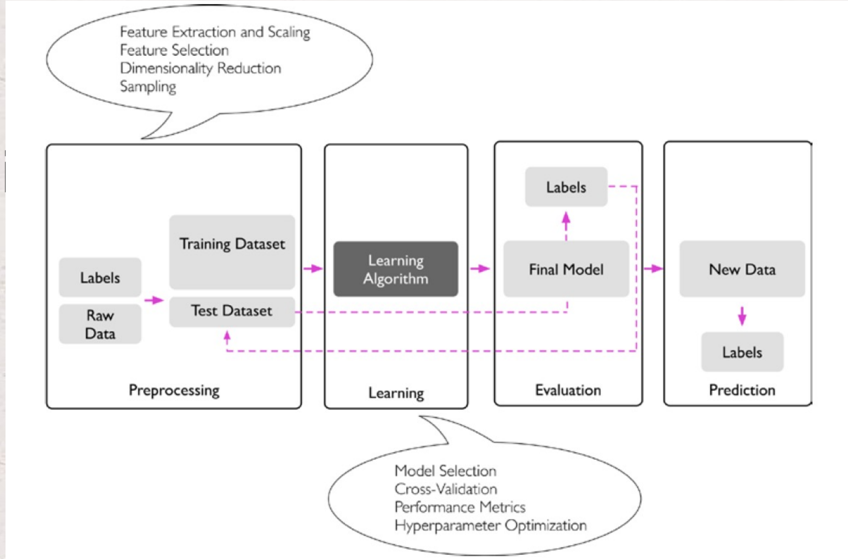
- How to select the data?
- Do we know enough about modeling by the time we g

- How to divide tasks within the team?



Questions on Final Project

- How to select the data?
- Do we know enough about modeling by the time we g
- How to divide tasks within the team?
- What do you mean by “individual grade”?



Last week

- General concepts of Machine Learning (ML)
- Roadmap for building ML systems
- Review of Numpy arrays

Course website:

https://corneliailin.github.io/datasci_w207_summer2024/

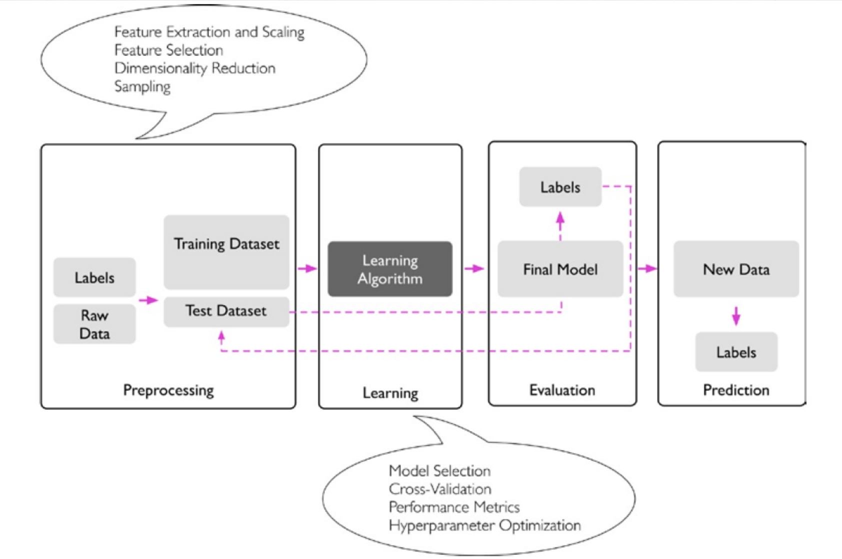


Image source: S. Raschka and V. Mirjalili, Python Machine Learning

Today's learning objectives

- General concepts of Linear regression and Gradient Descent
- Making predictions using the diabetes dataset (1. Linear_regression (gradient descent).ipynb)
- Breakout room exercise
- Introduction to TensorFlow2 (2. Tensorflow_introduction.ipynb)

Linear regression

Q1: What is the **assumed relationship** between outcome (y) and features (X)?

Linear regression

Q1: What is the **assumed relationship** between outcome (y) and features (X)?

$$h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2$$

Linear regression

Q2: What are the **parameters** of the model?

$$h_\theta(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2$$

Linear regression

Q2: What are the **parameters** of the model?

$$h_\theta(x) = \boxed{\theta_0} + \boxed{\theta_1} x_1 + \boxed{\theta_2} x_2$$

Linear regression

Q3: How do we choose the optimal **value** of these **parameters**?

$$h_\theta(x) = \boxed{\theta_0} + \boxed{\theta_1} x_1 + \boxed{\theta_2} x_2$$

Linear regression

Q3: How do we choose the optimal **value** of these **parameters**?

$$h_\theta(x) = \boxed{\theta_0} + \boxed{\theta_1} x_1 + \boxed{\theta_2} x_2$$

Define a cost (loss) function:

$$J(\theta) = \frac{1}{2} \sum_{i=1}^{n} (h_\theta(x^{(i)}) - y^{(i)})^2.$$

and <u>pick parameters to minimize J(θ)</u> so that $h_\theta(x)$ is very close to y (at least in the training data)

- using a search algorithm (e.g., gradient descent), or by
- explicitly taking J(θ) derivatives with respect to the θj's, and setting them to zero.

Gradient descent

Q3: How does gradient descent work?

Gradient descent

Q3: How does gradient descent work?

- Start with some “initial guess” for θ or use transfer learning.
- Continue until hopefully we converge to a value of θ that minimizes J(θ).

Gradient descent

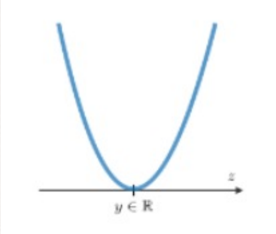
Q4: Can you reach a **local minima** using gradient descent for linear regression?

Gradient descent

Q4: Can you reach a **local minima** using gradient descent for linear regression?

No, gradient descent always converges to the global minima in the linear regression model (assuming the learning rate is not too large).

$$J(\theta) = \frac{1}{2}\sum_{i=1}^{n}(h_\theta(x^{(i)}) - y^{(i)})^2.$$



Linear regression

Gradient descent

Q5: What is the difference between **stochastic** gradient descent (SGD) and **batch** gradient descent (BGD)?

Gradient descent

Q5: What is the difference between **stochastic** gradient descent (SGD) and **batch** gradient descent (BGD)?

- BGD scans through all training examples in a batch before making a single step (costly operation if N is large; choose the batch size wisely)

- SGD starts making progress right away and continues to make progress with each example it looks at. Also:

    - gets θ “close” to the minimum much faster

    - can escape local minima in non-linear models (the gradient on the batch dataset could be 0 at some point, but at that same point, the gradient could be different)