Mini Course 2: Machine Learning I

Fundamental ML Concepts, ML Pipelines, Regression

Note on Resources

- This is meant to be a light introduction, so we won't go over the gritty details
- StatQuest is a great resource for learning linear algebra, and statistics, and machine learning!
    - Josh Starmer is very funny, and has great visuals ☺

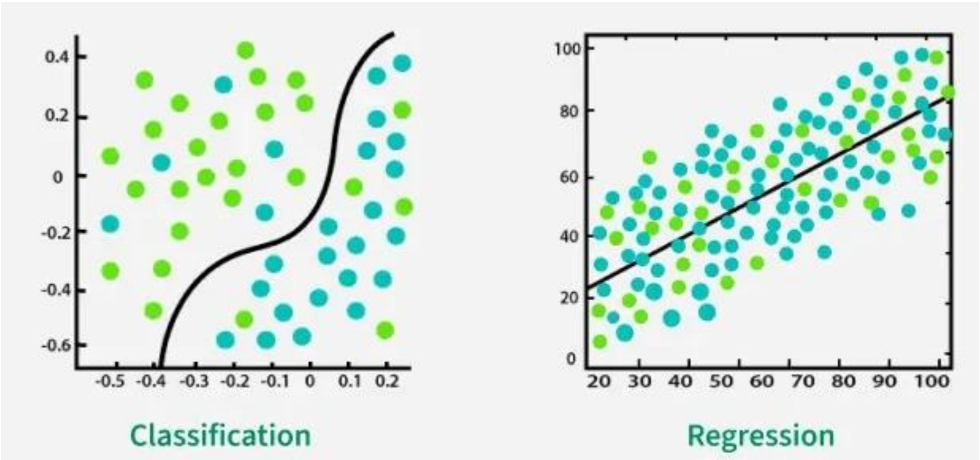
What we’ll go over today

- Fundamental concepts in ML
- Essential ML Model Training Pipeline
- Linear Regression
- Multivariate Linear Regression
- Transforming Nonlinear Distributions
- Overfitting and Underfitting
- Exercise in Jupyter Notebook!

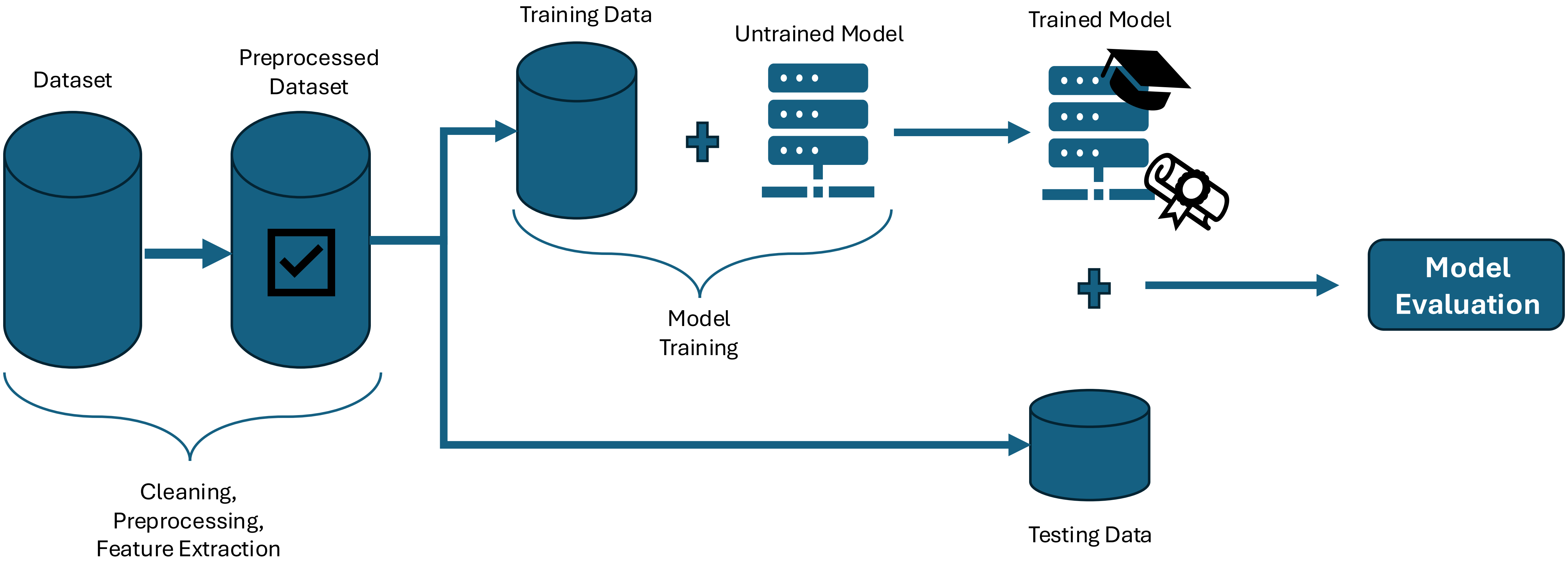
Fundamental Concepts in ML

- Each individual row in a dataset is an **observation** or **sample**
- **Features** are the parts of the data used for learning
  - An **n-dimensional** feature space uses n features
- **Predictors (or labels)** are what is being learned
- **Supervised machine learning** uses features to estimate predictors
- **Unsupervised machine learning** learns the underlying distribution without labels
- **Regression** learns continuous predictors
- **Classification** learns discrete predictors (e.g., labels like 'animal')

Make	Engine Size	Price
Toyota	130	13495
Toyota	152	17540
Subaru	122	11204
Honda	144	14582



Essential ML Model Training Pipeline



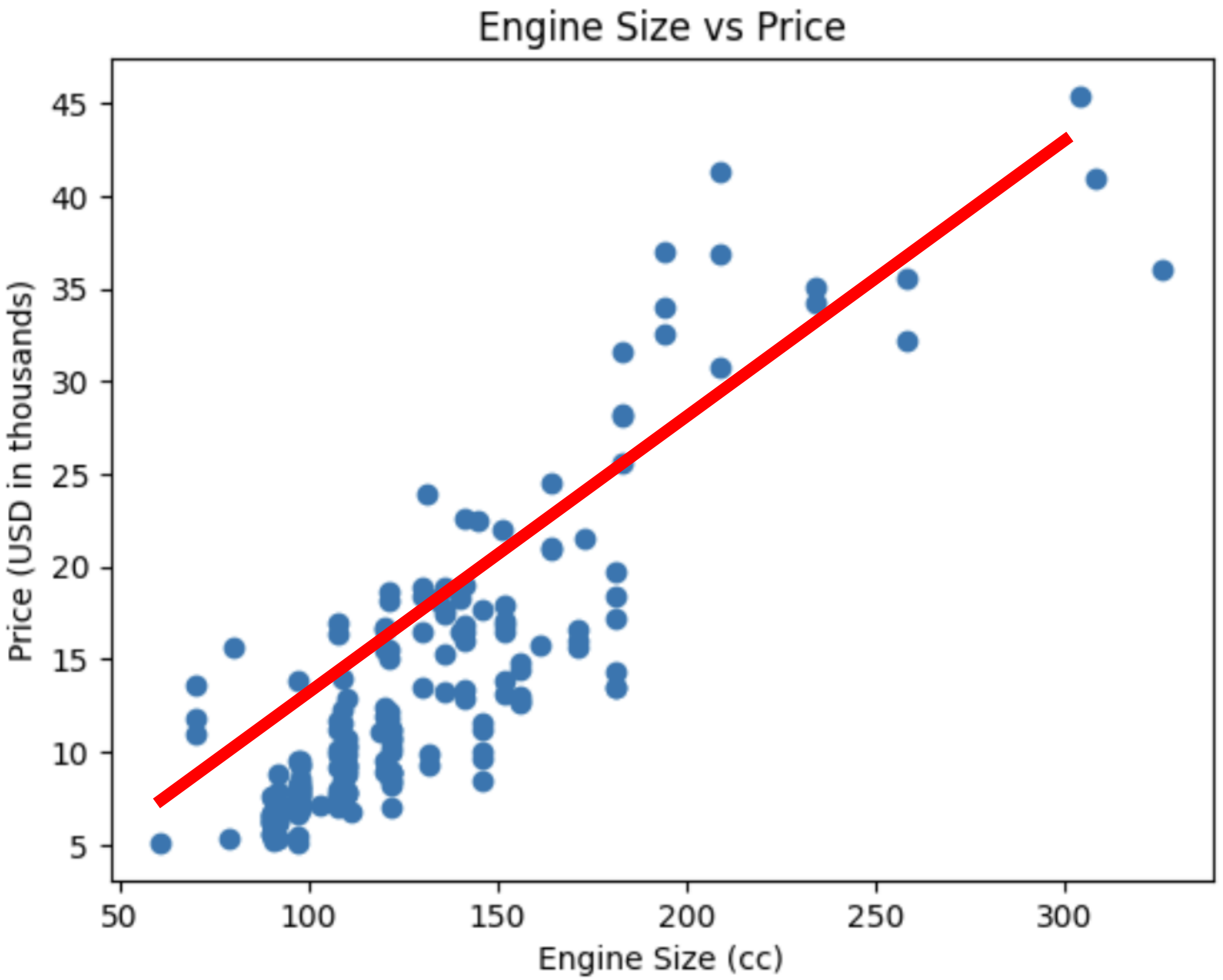
Linear Regression

- Goal: learn a weight matrix (w) that best transforms features into the predictor
- This means we want to minimize the difference between y and w*x (+b)
- For a line, each x and y are single observations, w is a single weight (slope) value
- Let’s take a look at an example

$$\underset{\text{predictor}}{\underline{y}} = \underset{\text{weights}}{\underline{w}} * \underset{\text{feature}}{\underline{x}} + \underset{\text{intercept}}{\underline{b}}$$

Example: Engine Size vs. Price

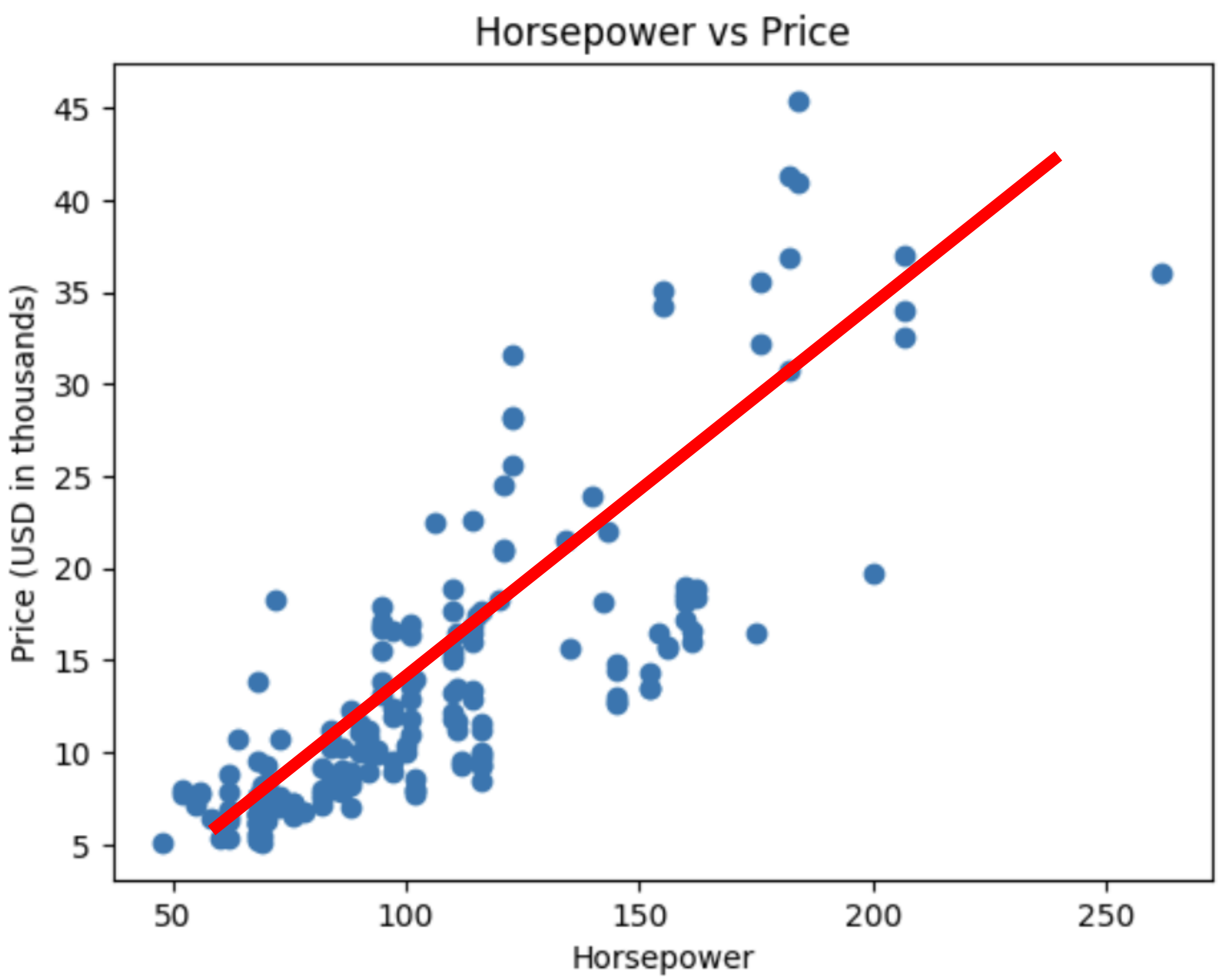
- Take the automobile dataset from the previous week, for example
- We plotted engine size vs. price and found a linear relationship
- We can try to train a linear regression model, which will learn a **line of best fit**
  - This line can now be used (and updated) on new data!



Multivariate Linear Regression

$$y = w * x + b$$

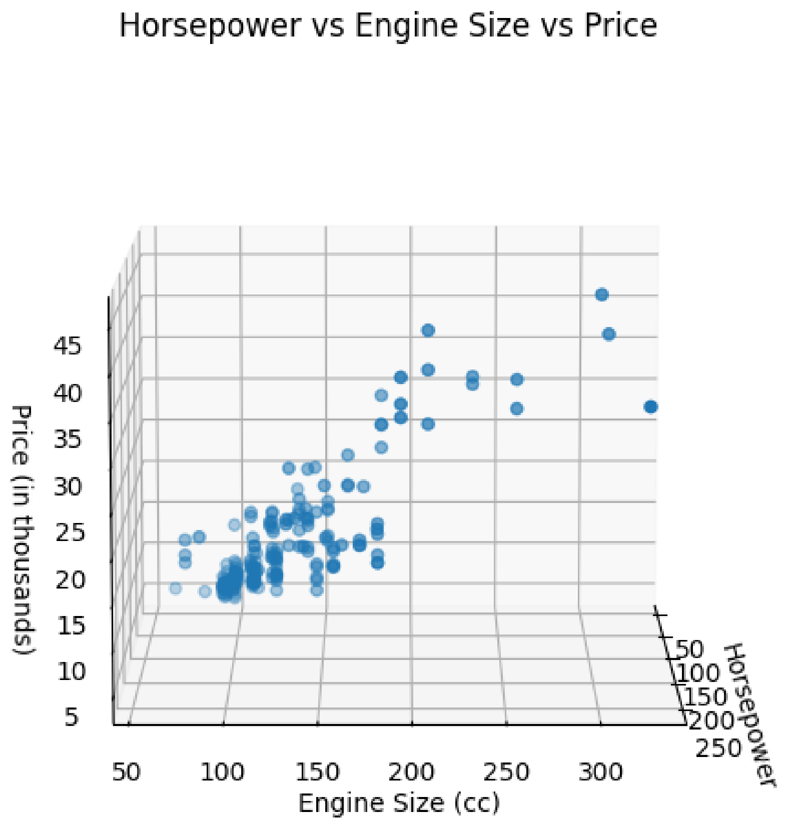
- Horsepower also predicts price quite well. Can we combine it with engine size to improve prediction?
  - Yes! This would mean that each observation in x has *two* elements (a 2-dimensional vector)
- Linear regression generalized: fit a (d-1)-dimensional hyperplane on a d-dimensional feature space
- Increasing the number of features *can* help prediction, but can also lead to **overfitting**



Multivariate Linear Regression

$$y = w * x + b$$

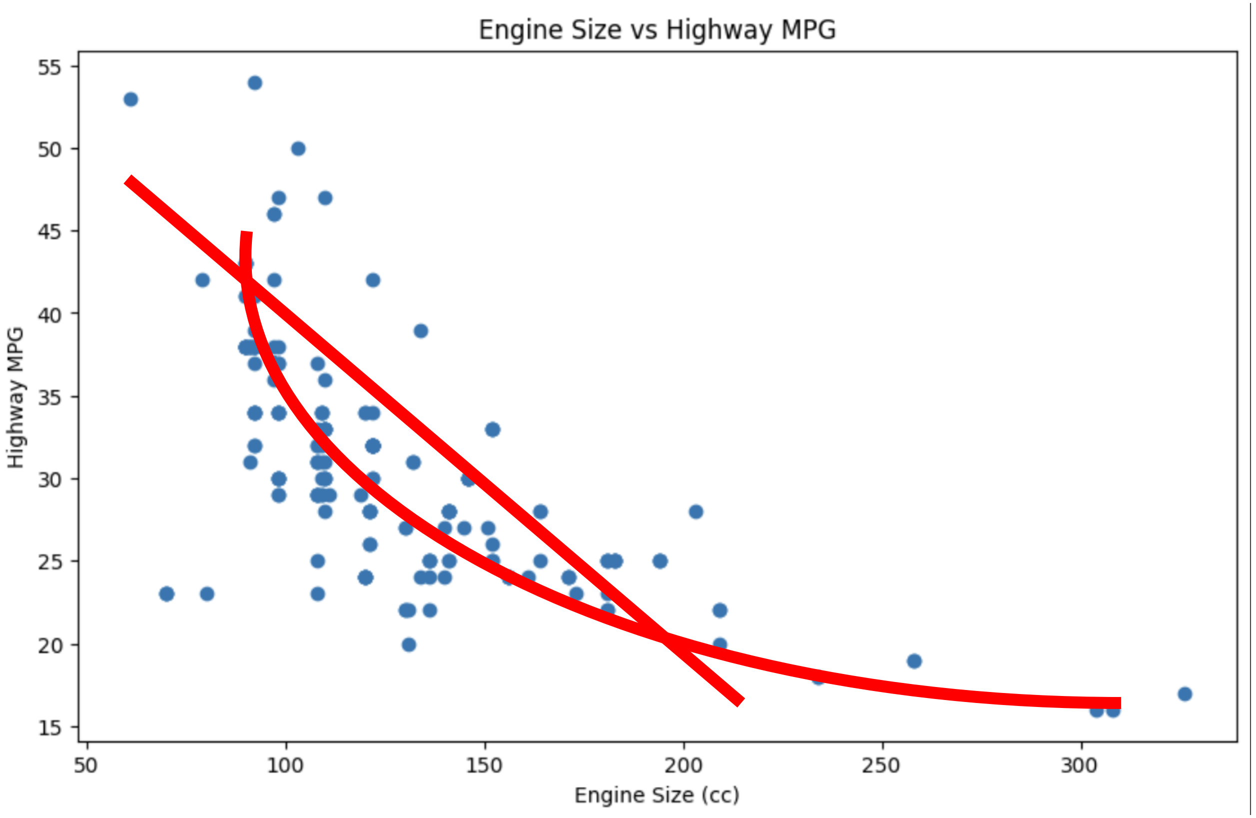
- Horsepower also predicts price quite well. Can we combine it with engine size to improve prediction?
  - Yes! This would mean that each observation in x has *two* elements (a 2-dimensional vector)
- Linear regression generalized: fit a (d-1)-dimensional hyperplane on a d-dimensional feature space
- Increasing the number of features *can* help prediction, but can also lead to **overfitting**



Nonlinear Features

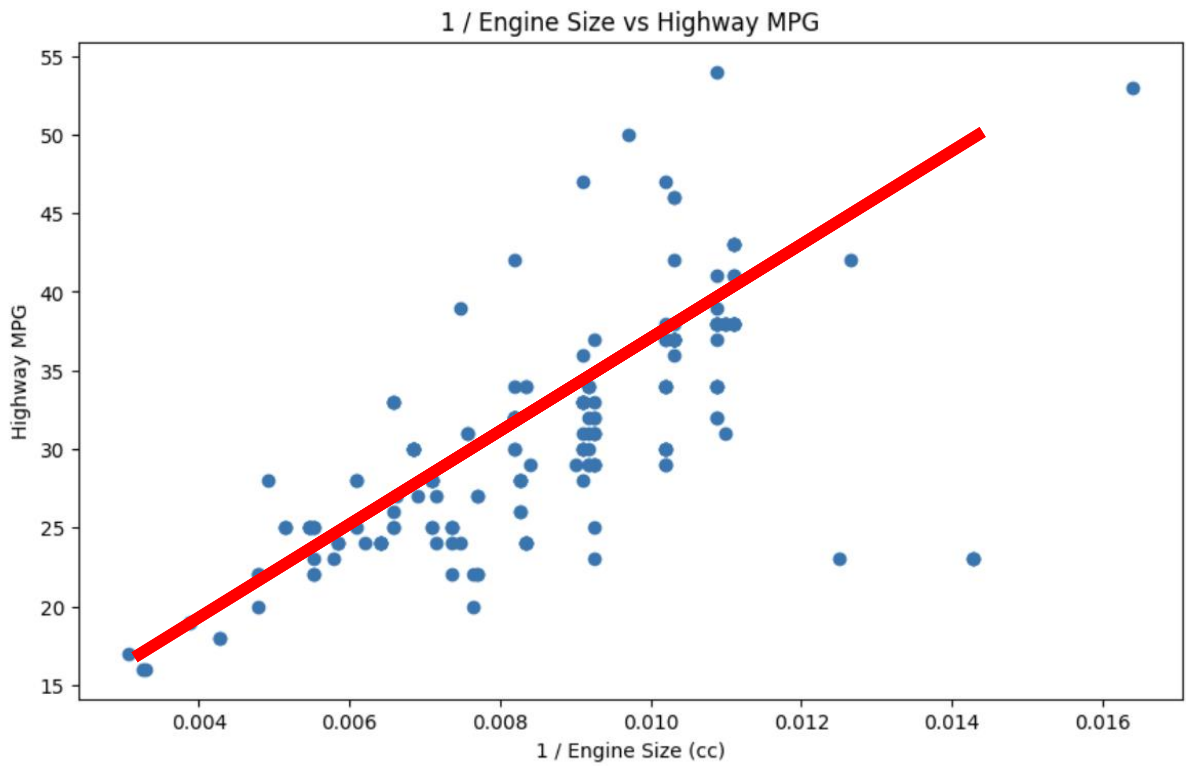
$$y = w * x^{-1} + b$$

- What if our data is not exactly linear?
  - The relationship between engine size and highway MPG is not exactly linear
- There are transformations that we can try to <u>linearize</u> the data
  - **Note:** This is why it's important to build intuitions about our data via exploration!
- We can *transform* feature **x** (engine size) into **1/x** (MPG) before training in order to linearize it
  - This transformation is part of <u>preprocessing</u>



Nonlinear Features

- What if our data is not exactly linear?
    - The relationship between engine size and highway MPG is not exactly linear
- There are transformations that we can try to <u>linearize</u> the data
    - **Note:** This is why it's important to build intuitions about our data via exploration!
- We can *transform* feature **x** (engine size) into **1/x** (MPG) before training in order to linearize it
    - This transformation is part of <u>preprocessing</u>

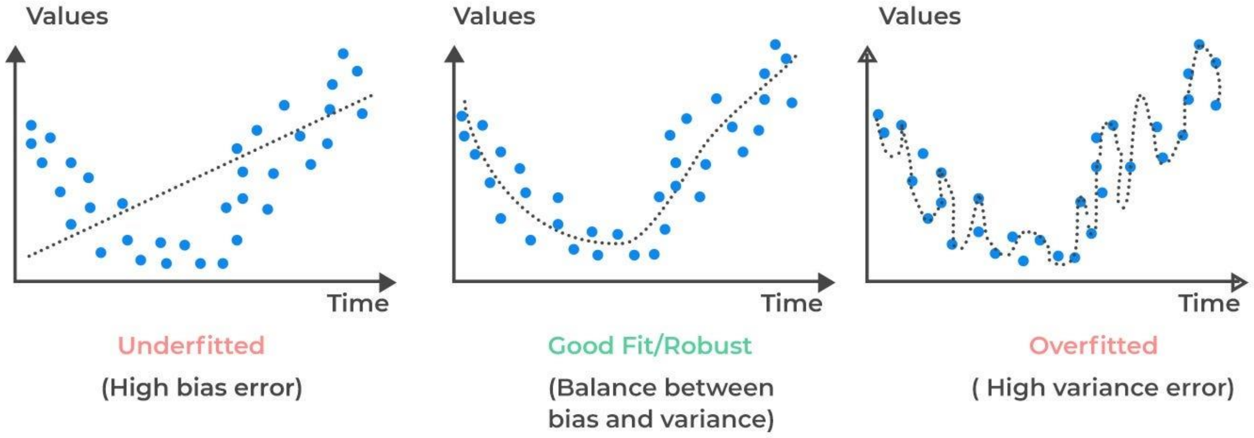


Quick Check-In!

- What are some potential features in the data you're working with? What are potential predictors/labels?
  - Are they continuous or discrete?
- Can you apply regression to the data you're working with this summer?
  - If so, how?
  - If not, why not?

Note on Overfitting and Underfitting

- Theoretically, we could define a polynomial that would fit our data perfectly
- But that would lead to a model that doesn't generalize to new data
- We must consider the tradeoff of **bias** (overfitting) and **variance** (underfitting) while creating machine learning models



Jupyter Notebook Time!

- Navigate to your local copy of the workshops repository
  - Return to previous instructions if you don't have this yet
- **Note:** You need to update your filesystem, as the structure of the repo has changed. Make sure you have folders called mc1/ and mc2/
- Run: **`git pull origin main`**
- Enter the mc2/ folder and open the file mc2.ipynb
- Follow the instructions in the document