Martin Jimenez
ML
HW2

Code GIT: https://github.com/jimenezm3atwit/ML.git

1A. **In two-to-three sentences, how can abstract superclasses in object-oriented Programming facilitates modularity?**

Abstract superclasses in object-oriented programming implement common behaviors and interfaces that can be shared by subclasses. while subclasses implement their unique functionality with consistency in the interaction with the rest of the system. This allows for modularity, since code can be extended on or replaced without needing much significant changes to the larger application.

The modular elements in non-object-oriented languages should be documented which means describing the purpose of each function, the inputs and outputs, and then finally the interaction between the various functions, so that the next user or your peers can know how to link and work with the modules.

1B. **If we do not provide functions that manipulate each of the variables (set methods), then how do you think these variables receive their assignments?**

Variables should not change except in controlled ways, for example during training or initialization Variables are often initialized or specialized through methods that place them in a context relevant to the model with no direct way of manipulation. This then helps with encapsulation: where there is minimal direct access, thus reducing the risk of unintentional modifications. This is a preferred template for a machine learning algorithm, as it helps maintain the internal state of a model.so it is easier to be consistent and to avoid bugs.
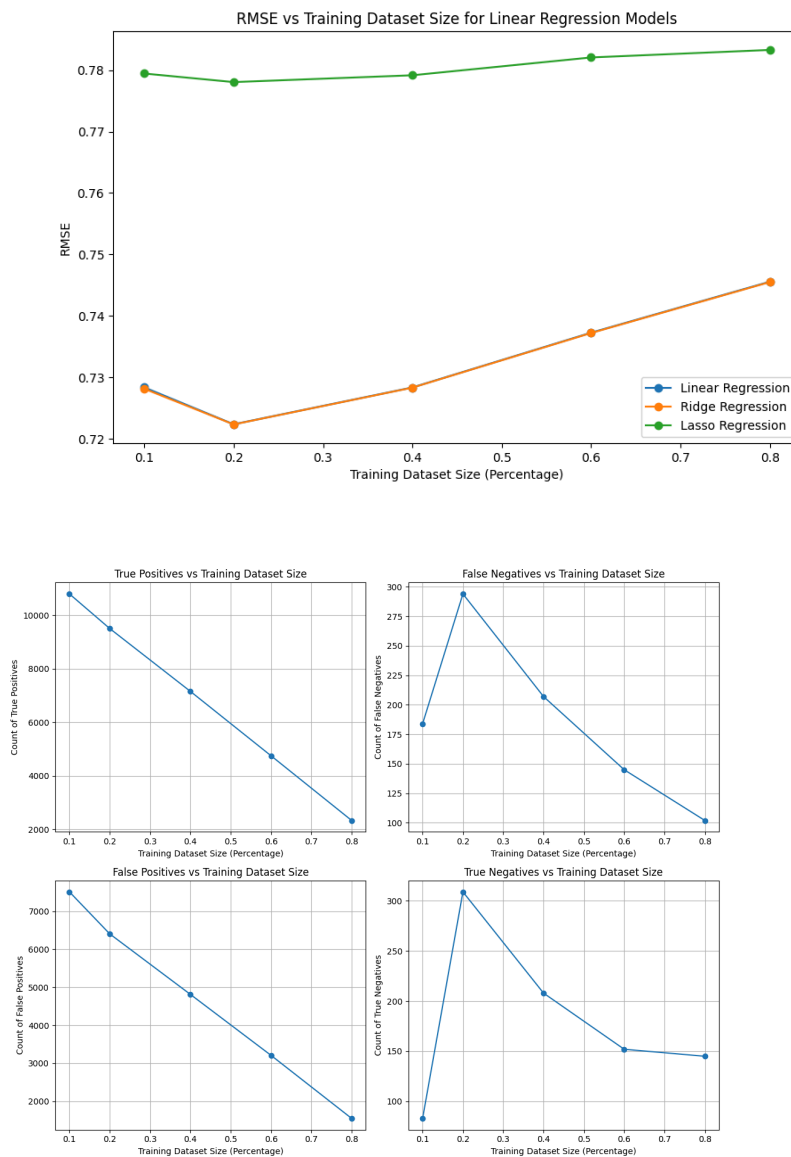
1C. which is more appropriate for hyperparameters, which is more appropriate for parameters, and why?
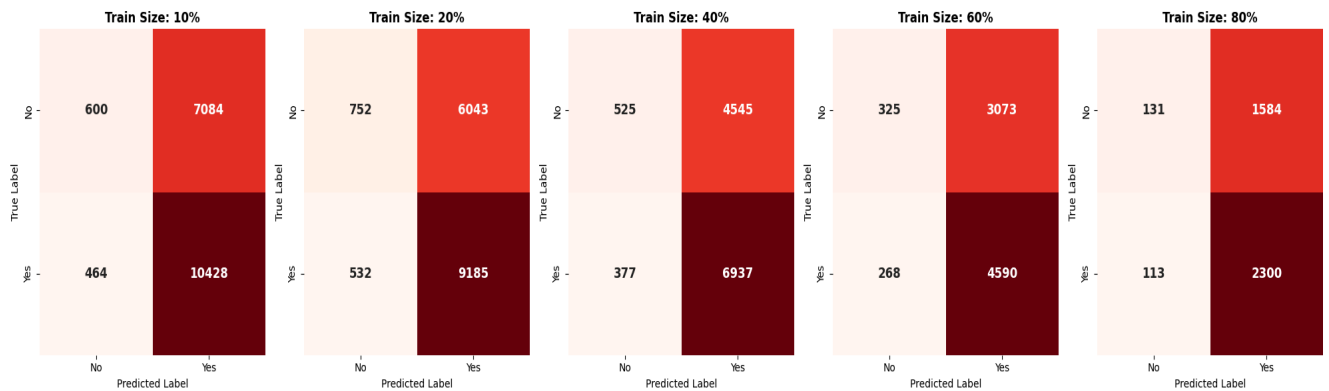
In practice, a hash map/dictionary is more suitable for hyperparameters, since they usually have distinctive names and their types may be different; learning rate, regularization strength, or the number of iterations are examples

Learned parameters will be better with an array or matrix since they normally are numbers in some sort of set arrangement, such as weight vectors or coefficient matrices

3:

Most homes are marked as "willing to purchase," with 12,126 of 20,640 getting a 1, meaning buyers are generally interested at the predicted price. From the confusion matrices, True Positive and True Negative counts drop as we use more training data, but False Positives and False Negatives stay pretty high, which isn't great.

Train Size: 10%   **Train Size: 20%**   **Train Size: 40%**   **Train Size: 60%**   **Train Size: 80%**

	Predicted: No	Predicted: Yes
True No	600	7084
True Yes	464	10428

	Predicted: No	Predicted: Yes
True No	752	6043
True Yes	532	9185

	Predicted: No	Predicted: Yes
True No	525	4545
True Yes	377	6937

	Predicted: No	Predicted: Yes
True No	325	3073
True Yes	268	4590

	Predicted: No	Predicted: Yes
True No	131	1584
True Yes	113	2300

Q4A:

Flubsy's idea of calculating gradients by using tiny offsets makes sense at first, but it runs into big problems when the function isn't smooth or continuous. Imagine trying to find the gradient at a sharp point, like on an absolute value function ... .there's no single direction for the slope, since it changes suddenly!The same happens with piecewise/sub functions that have jumps, where the value changes suddenly, making it impossible to calculate a smooth gradient. In those cases, Flubsy's approach just doesn't give meaningful results, and you'd end up with errors or incorrect gradients that mess up the whole optimization.

4B:

So again In theory, Flubsy's idea sounds reasonably good because he uses very small changes, in order to find the gradient. This can lead to issues because of how computers deal with super small numbers. Since epsilon is so small, it can be subject to underflow, where the difference between two numbers is so small that the computer rounds it down to zero. That means, other than the right rate of change, you get something that doesn't make sense for the application or even just zero which in turn messes up a lot of the gradient calculations.

External Reasources:

How to use Skylearn:
1. https://youtube.com/playlist?list=PLcQVY5V2UY4LNmObS0gqNVyNdVfXnHwu8&si=tITAnyw5P7bmjSlE

2. https://www.digitalocean.com/community/tutorials/python-scikit-learn-tutorial

How to do Data Analysis:
1. https://www.youtube.com/watch?v=a9UrKTVEeZA
2. https://www.w3schools.com/python/matplotlib_intro.asp

Linear Reg:
1. https://www.w3schools.com/python/python_ml_linear_regression.asp

2. https://www.geeksforgeeks.org/ml-understanding-data-processing/