Project Guidelines

CS 438/638

1. Choose a classification problem

- Find a classification problem that is interesting to you.
- Either find a sufficient number of training examples or you generate them.
- Do not choose a problem with a single training feature.
- Discuss what is the output feature that you will learn, what are the input features that you will use, and why you think they are related.
- Maybe you don’t have obvious training features, but you need to generate them by processing some data. Discuss it.

2. Analyze how the training data is distributed

- Look at how the values in your training features are distributed. Do you need feature scaling?
- Look at how correlated are your features. If some of them are highly correlated, you may consider filtering out redundant features.
- Use visualization tools to document your work (box plots, histograms, etc).
- If you have way too many features, you may benefit from some feature selection. Discuss how you reduced the number of features.

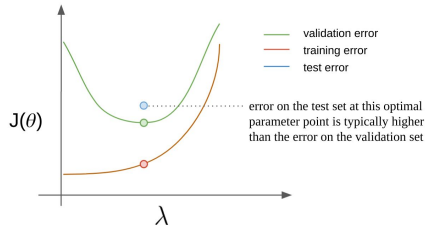
3. Apply machine learning algorithms and tune them

Work on applying at least one of the following algorithms on your dataset.

- Logistic regression
- Support vector machine
- Neural networks

Tune your parameters:

- Randomly separate the data into training, validation and test sets.
- Tune the parameters in these algorithms using the validation set. Show us how you decided.

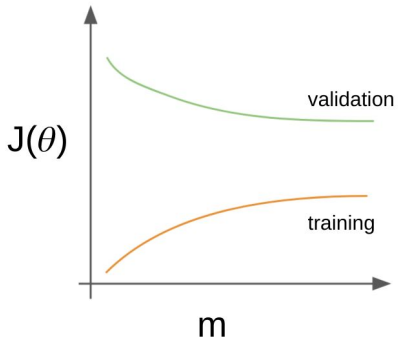


4. Produce alternative models

- To improve your model, consider using more data features or less data features.
- You can increase the features by generating derivative features (polynomial or multiplicative). You can decrease them by eliminating features based on some rationale.
- Or instead of changing the features used, you can consider applying an alternative learning method to your problem.
- Tune the hyperparameters of alternative settings just like you tuned the original one.

5. Learning curve

- For each learning setting, generate a learning curve and decide if you can benefit from collecting more training examples.
- Comment on the quality of the fit for each model (underfit / overfit) and demonstrate your reasons.



6. Analyze your success

- Execute each method on the same test data, and compare their ROC curves and their AUC.
- Calculate your precision and recall.
- Compare the errors that these methods make. Put those errors in a Venn diagram so that we can see if the methods make similar errors.
- Select some sample errors and comment on why they are misclassified.

Maybe it is not very hard to understand why some of the examples are classified incorrectly.

7. Communicate your results

As a result of your project, you need to prepare the following.

- Your code along with your inputs and outputs, and instructions to reproduce your results
- A 10 minutes presentation for the class
- A report as a PDF file. Make sure it has the following sections:
  - Description of the problem and the data
  - Parameter tuning with charts
  - Generation and tuning of alternative models
  - Learning curve analysis
  - Performance and error analysis