Lesson 18 – Machine Learning Applied:

**Questions for Mentor:**

**Model Evaluation:**

* Evaluation metric
  + A way to quantify performance of ML model
  + Not the same as a loss function
* Supervised Learning Metrics
* Classification metrics
  + Accuracy
    - # of correct predictions / total # of predictions
    - .score() method
    - Many fallacies – if 95% are positive, and our model just says everything is positive, we get 95% accuracy
  + Confusion matrix
    - Not a metric
    - Helps gain insight into the type of errors a model is making
    - Helps to understand some other metrics
    - Table of true negatives (TN), false negatives (FN), true positives (TP), false positives (FP)
  + Precision
    - TP / (TP + FP)
    - Used when want to minimize false positives
  + Recall
    - TP / (TP + FN)
    - Used when want to minimize false negatives
  + F1 score
    - Summarizes confusion matrix in one number
  + Matthews correlation coefficient
    - Takes into account all fields of confusion matrix
    - Doesn’t extend well into multi-class problem
  + ROC Curve
    - Plot of true positive rate on Y, false positive rate on X
  + AUC
    - Area under ROC curve
  + Precision/Recall curve
    - Plot recall on X, precision on Y
  + Log loss
  + Takes into account uncertainty of model predictions
  + Larger penalty for confident false predictions
* Regression metrics
  + Default .score() method is the R^2 score
    - Has intuitive scale
    - Doesn’t depend on target units
  + Mean absolute error
    - Avg of absolute value of residuals
    - More robust to outliers – not squared
  + Mean squared error
    - Avg of squared value of residuals
    - More sensitive to outliers because residuals are squared
  + Root Mean Squared error
    - Root of MSE
    - Gives high weight to large errors
  + Root mean squared logarithmic error
    - Like RMSE but you get log of y+1 instead of y
* Takeaways
  + There’s a reason for this variety
  + No ‘one size fits all’ metric
  + Get to know data
    - Outliers
    - Class imbalances
    - Etc
  + Keep in mind business objective of ML problem

**Regression Metrics:**

* Adjusted R^2 considers marginal improvement added by an additional term in model
* RMSE is the best standard but every ML problem is different

**Hyberparameters:**

* Applied deep learning
  + Idea -> code -> experiment
  + Empirical process – try a lot of things and see what works
* Model parameter = w^tx = y
  + X = vector of features of data
  + Y = target value
  + W = weight vector
* Hyperparameters
  + Values that must be specified outside of training procedure
  + i.e. lasso and ridge regression that add regularization term to linear regression
  + control capacity of model
    - prevents overfitting
  + Tuning
    - Finding the best set of hyperparameters to use as model parameters

**Grid Search & Randomized Search in Python:**

* Grid search great for small data sets – will get slowed down with larger ones
* Random search saves time with large datasets and doesn’t sacrifice much in accuracy

**Bayesian parameter optimization:**

* Tuning tips
  + Keep open mind
  + Don’t do grid search
    - Some hyperparameters don’t matter
    - Gridsearch creates duplicate points, only differ with irrelevant HPs
    - Pick HPs randomly
  + Try to eliminate hyperparameters
  + To see a clear pattern it can take way longer than you expect
* Bayesian linear regression
  + Considers various plausible for how the data were generated
  + Makes predictions using all possible regression weights, weighted by their posterior probability
  + We can turn them non-linear as well
  + Surrogate function
    - Model for how well HP will do on our data
  + Acquisition function
    - Tells us how promising a candidate is
  + Don’t care as much about points we’re confident in. care more about uncertain
* Bayesian optimizer
  + Takes a function and hyperparameters and tests n iterations
* L1 regularization is lasso regression
* L2 regularization is Ridge regression