1) If you have a training set with millions of features, use the Lasso() Linear Regression algorithm

2) The SGD algorithm might suffer if the features in the trainings set have very different scales

3) To remedy the above, scale all the features

4) It is not a good idea to stop a mini-batch gradient descent immediately when the validation error goes up

5) Decision trees have linear runtime

6) If a decision tree is underfitting a training set then scaling the input features wont make a difference

7) If a decision tree is overfitting the training set it's a good idea to try decreasing max\_depth

8) The Gini impurity of a node is generally lower than it's parent, but not always

9) The depth of a full decision tree trained without restrictions is log\_2(number of examples)

10) If an SVM classifier with an RBF kernel is underfitting increase gamma

11) Use the primal form or the SVM problem to train a model on a training set with millions of instances and hundreds of features

12) An SVM classifier can output a confidence score when it classifies an instance, but not a probability

13) It is important to scale the inputs when using SVM's as it helps finding a larger margin/wider street

14) A support vector is every example that sits on the decision boundary

15) The fundamental idea behind SVM's is putting a wide street or large margin between the two classes

16) For an item that needs to be categorised into 2 sets of categories implement 2 logistic regression classifiers, one or each set

17) Use Elastic Net instead os Lasso when we suspect that many features are highly correlated with some other features

18) Use Lasso instead of Ridge Regression when we suspect many features are irrelevant

19) Use Ridge Regression instead of plain Linear Regression (without regularisation) when LR was shown to overfit on the problem in question

20) Using Ridge Regression, if the training error and validation error are almost equal and fairly high, this is underfitting caused by a high bias, therefore you should reduce alpha

21) Using a constant learning rate, batch gradient descent will converge to an exact optimum

22) ^ Stochastic gradient descent will get close to an exact optimum the fastest

23) Use Clustering to segment customers into multiple groups

24) spam detection is a supervised problem

25) Memory based algorithms like nearest neighbor rely on a similarity measure to make predictions

26) bad karma if you tune hyperparams in a test set + might fool oneself into believing a higher accuracy

1) Regression algorithm, training and validation error are almost equal, but also rather high: this means the model is underfitted and the alpha weight penalty should be lowered

2) To make SVM classifiers less sensitive to the outliers you should decrease the c/cost of misclassification parameter

3/4) Recall is the number of true positives divided by the number of true positives AND false negatives (aka total number of positive classes in the example)

5) larger cost = larger epsilon, decrease epsilon decreases the epsilon boundaries

6) Decision tress can be used for regression, are sometimes used in ensembles, and are building blocks for Random Forrests

7) The mean square error can sometimes overemphasise outliers

8) Training an SGDclassifier with too large a learning rate can cause the solution to not converge

9) If an AdaBoost is underfitting then increase the learning rate of the boost algorithm to increase/decrease the weights to a small degree

10) Using X fold cross validation, models are trained with X-1/X amount of the training data

11) To simplify a model \*DONT\* provide more attributes, more training examples or use one hot encoding

12) In bagging, out-of-bag-evaluation is almost like having a validation set for free

13) When training an SVM classifier with RBF kernel, if all the decision boundaries are squiggly then decrease gamma and/or C

14) Training decision trees is a linear process, for each minute it trains X many examples

15) The number of support vectors decrease when we increase the cost parameter C in an SVC classifier with RBF kernel

16) Decision trees kind of work like binary search trees going down a true/false path

17) A validation set is used to tune the hyperparameters

18) If a linear kernel SVM classifier is failing to give an accurate classification of a dataset then try changing the kernel to RBF

19) Mean Squared Error is a commonly used cost function for linear regressions (not hinge loss or cross entropy loss)

20) If an item is to be categorised in 2 separate categories then use two separate classifiers

21) With PolynomialFeatures class for preprocessing, if the training error is smaller than the validation error then the model is overfitted and the degree of the polynomial kernel should be reduced

22) Multi-label learning predicts more than one binary target

23) The RandomForrest classifier can output probabilities for all class labels when it classifies an instance

24) K-nearest neighbor classifier is an example of instance based learning (NOT elastic net, SGD classifier)

25) Small changes in the values of the data can cause large changes to the structure of the decision tree means that training a decision tree is instable

26) Use the lowest validation loss in AdaBoost ensemble

27) The Hinge Loss is the function max(0, 1-t) where it is equal to 0 when t >= 1

28) Using a trained scikit-learn pipeline ensures that the test data will be preprocessed the same way as the training data - this is always true

29) Class probability = 1-((value/samples)^2 + (value/samples)^2 + ...)

30) If you have missing values in your data, you can: remove data with missing values, replace the missing value with the most common value for this feature, or replace it with a median value

31) Logistic regression is an example of instance based learning

32) Stochastic Gradient Descent will get to an exact optimum faster than batch gradient descent or mini batch gradient descent

33) A random patch is a random sample of samples and features

34) if a decision tree is underfitting the training set, it is a good idea to increase the max depth

35) When training a SVM classifier with 400,000 examples and 50 features, set "dual" to be "false"

36) A hard vote selects the option that the majority have selected

37) Dont use the preprocessing methods (standardscaler, ordinalencoder and onehotencoding) if there is the occasional missing value

38) If you tune hyperparameters during the test set you might fool yourself to believe a higher than acutual accuracy, and all the bad karma will follow you

39) If your training data has more features than examples and the features are correlated with each other, use ElasticNet()

40) If your training data has more features than examples and the features are uncorrelated, use Lasso()

41) Clustering is a type os algorithm that segments your customers into multiple groups

42) Predicting the future price of an item - use supervised regression

43) The ROC curve plots the true positive rate over the false positive rate for all possible thresholds

44) If the ROC curve of a classifier A is always higher than that of classifier B, then A is always better than B

45) If a model underfits use more features

46) Predicting the species of an animal in a image is a classification problem

47) Ensembles combine two or more classifiers for more accurate results

48) If scikit-learn pipeline ends with a classifier it can be used like a classifier - this is always true

49) K-old cross-validation splits the training data into k equal sized folds

50) The CART decision algorithm greedily chooses a split point that minimises the overall Gini impurity

Ensembles

- A Decision Tree Regressor with n-leaf nodes creates a set of #n values that gives the smalled mean squared error

- Decision trees can be unstable, and small changes in the data can cause the structure of the tree to change signifigantly

- To simplify a deep decision tree without losing classification accuracy we can rotate the data

- Bagging ensembles: the algorithm can be parallelized and run on multiple machines, each bag is very likely o be unique and each classifier is trained on a unique training set, some examples may be sampled more than once

- The dataset in each decision tree in a random forrest classifier is built by sampling both random features and random examples from the dataset

- Boosted ensembles cannot be parallelized as they typically require the result rom the previous stage to build the next stage

- Stacked ensembles can be partially parallelized as some parts can be but the meta-learner requires the output of the individual classifier

- If a Boosted ensemble is overfitting, try lower the number of ensemble members, complexity of the individual classifier, or the learning rate of the boosting algorithm

- Ensembles generally perform better than individual classifiers when the classifiers are diverse. If the classifiers are not diverse the performance wont be any better

Misc Notes

Machine learning is the science of programming computers so that they can learn from data

ML automatically adapts to change

Great tool to use instead of a complicated approach or no known algorithm

ML can dig into large amounts of data to extract any patterns that might not have been apparent - data mining

```To summarize, Machine Learning is great for:

• Problems for which existing solutions require a lot of fine-tuning or long lists of rules: one Machine Learning algorithm can often simplify code and perform better than the traditional approach.

• Complex problems for which using a traditional approach yields no good solution: the best Machine Learning techniques can perhaps find a solution.

• Fluctuating environments: a Machine Learning system can adapt to new data.

• Getting insights about complex problems and large amounts of data.```

Regression = predicting values

Supervised Learning - the training set includes the desired solutions: called labels

- Tasks: Classification (separating data into categories ie spam), Regression (predict a target numeric value based on a set of predictors/features)

```

Here are some of the most important supervised learning algorithms (covered in this book):

• k-Nearest Neighbors

• Linear Regression

• Logistic Regression

• Support Vector Machines (SVMs)

• Decision Trees and Random Forests

• Neural networks

```

Unsupervised Learning - the training data is unlabelled and the system tried to learn without a teacher

- Creates groups/clusters

Dimensionality reduction - simplify data without losing much info, merging features: feature extraction

Anomaly Detection - odd behaviour

Semisupervised - some data labelled as labelling all of it can be time consuming

Reinforcement - observes an environment and receives rewards/punishments based on its actions, this allows it to find an optimal path ie alpha go

Batch Learning - MLA must be trained with all the available data, this takes time so usually done offline

Online Learning - MLA learns in small batches or incrementally, great for on the fly or continuous flow, great for limited resources

- Important Parameter: learning rate is how fast the algorithm can adapt to changing data

- Big Challenge: Bad data will result in the MLA being gradually less effective

Instance Based - learns examples by heart, then examines new cases by measuring similarity

Model Based - builds a model from a set of examples and uses this to predict future samples

- Fitness function: used to evaluate how good a model is

- Cost function: used to evaluate how bad a model is

Typical ML Project - study data, select model, train on training data, make predictions for new data

Main Challenges - Insufficient quantity of training data, nonrepresentitive (doesnt represent new cases that you want to generalise), correctly sampled data (otherwise sampling bias and/or noise), poor quality data (filled with noise/errors/outliers), Irrelevant features (we want the most useful features), Overfitting (over generalising - gather more data, reduce noise, simplify model/regularisation), Under-fitting (select more powerful model, feed better features, reduce constraints)

Testing/Validating - train and test set, test set evaluates how well the data did. If training error is low and generalisation error is high, data is overfitted.

Comparing Models - compare test set results, choose hyperparameter: have validation set and choose the best model based on this validation set result

Performance Measures - Regression: Mean Root Square Error, Mean Absolute Error

- List and verify assumptions early on to catch any errors

Preparing Data - fillna(): remove missing values, feature scaling (minmax: everything is between 0 and 1, standardization: uses mean and std deviation)

Classifiers

- Binary: either yes or no answers, use cross validation to evaluate a model, accuracy isn't the ideal performance measurement when you have skewed data, use confusion matrix instead (TP, FP, FN, TN)

precision = TP/(TP+FP)

recall = TP/(TP+FN)

F1 = 2 x (precision x recall)/(precision + recall) is a simple way to compare classifiers

- Precision/Recall Tradeoff: raising threshold decreases recall

- ROC curve: plots true positive rate (recall) against false positive rate

Linear Regression - makes a prediction by computing a weighted sum of the input features plus a constant called the bias term

Gradient Descent - generic optimization algorithm, tweak params iteratively to minimize cost function: steps, Set a large number of iterations for the learning rate but interrupt when the gradient vector becomes tiny

SGD - picks random instance, much less regular than batch instance

Mini Batch - combines the previous two, does SGD but on mini batches instead of whole dataset

Polynomial Regression - for nonlinear data, add powers of each feature as new features then train linear model on this extended set of features

Ridge Regression - regularized version of Linear Regression, regularization term is added to the cost function, forces data to be fitted and keeps woights as small as possible

Lasso Regression - like Ridge Regression but adds regularization term to the cost function, tends to eliminate the weights of the least important factors

Elastic Net - Middle ground between Ridge and Lasso, preferred over Lasso because L can behave erratically when num of features>num of training instances or when several features are strongly correlated

Early Stopping - when validation error reaches minimum stop training

Logistic Regression - Estimate the probability of something belonging to a particular class, outputs logistic of weighted sum of input features

Linear SVM - adding boundaries to a linear line, fitting the widest possible street

Nonlinear - Like linear but uses polynomial features to separate areas, change polynomial degree to cope with over/underfitting

Gaussian - hyperparameters gamma and C

SVM Regression - tries to fit as many instances on the street while limiting margin violations, width of street controlled by epsilon

Decision Trees - a big long if statement that creates boundaries

CART - used to train trees, splits training into 2 sets using a single feature and a threshold, then does same for subsets etc until max depth reached

- Instability: sensitive to training set rotation

Voting Classifiers - train many classifiers and get the majority vote

Bagging and Pasting - use the same training algorithm for each predictor and train them on different subsets of the training set. bagging: done with replacement (allows training instances to be sampled several times for the same predictor), pasting: done without relpacement

Random Forrest - uses decision trees with bagging, extra randomness by searching for the best feature among a random set of features, extra trees introduce more randomness by using random thresholds for each feature rather than searching for the best possible thresholds, feature importance

Boosting - combines several weak learners into a strong learner

AdaBoost - trains, calculates weight of misclassified, rinse and repeat with a different classifier etc

Gradient Boosting - Like Ada but tries to fit the new predictor to the residual errors made by the previous predictor instead of tweaking weights

Stacking - instead of using a trivial function to aggregate a bunch of predictors, just use another predictor to do it (called a blender), blender is trained on split data