# Machine Learning For Absolute Beginners

Third Edition

## Chapter 1: Preface

## Chapter 2: What Is Machine Learning?

## Chapter 3: Machine Learning Categories

## Chapter 4: Machine Learning Toolbox

## Chapter 5: Data Scrubbing

* Data scrubbing = refining your dataset to make it more workable, the ‘clean-up’ process
* Non-numeric or categorical row values can be problematic to merge while preserving the true value of the original data
  + i.e. merging countries into regions can easily become skewed with 10 countries in one area and 1 country in the other
* One-hot encoding = convert text based values into numeric/binary values
  + i.e True/False is 0/1
  + con: can be more features (columns) in a table
  + pro: broader spectrum of machine learning algorithms available
* Binning
  + typically continuous variables are best kept numerical, but sometimes if the algorithm requires it the data will be binned
* Normalization
  + helps improve accuracy of algorithms even though it’s not required
  + rescaling can help normalize variance, less exaggeration
    - define feature as between [-1,1], controlled for min and max
  + normalization isn’t recommended for data with extreme ranges, a normalized range may be too narrow
    - instead use standardization
* Standardization
  + Converts unit variance to a standard normal distribution with a mean of zero and a standard deviation (σ) of one
    - An extremely high or low value would be expressed as three or more standard deviations from the mean
  + Standardization is usually recommended before support vector machines (SVM), principal component analysis (PCA), and k-nearest neighbors (k-NN)
* Missing data
  + Use the mode value (most common vaue) in place of missing data for binary and categorical variables
  + Use the median value in place of missing data for continuous variable
  + Last resort, use rows missing values at the cost of less data

## Chapter 6: Setting Up Your Data

* Will need to split your data into training and test data
  + The training/test ratio is usually 70/30 or 80/20
* Before you split data it’s important to randomize the order of rows
  + Scikit-learn provides a built-in command that randomizes/shuffles
* In supervised learning…
  + A model is developed by feeding the machine the training data and analyzing the relationships between the features (X) of the input data and the final output (y)
* After randomization you must measure how the model performed
  + AUC = Area Under the Curve
  + ROC = Receiver Operating Characteristic
  + Confusion matrix
  + Recall
  + Accuracy
  + RMSE = mean absolute error and root mean error
    - Uses to assess models with a numerical output
  + MAE = Mean Absolute Error
    - Average of the errors in a set of predictions on a numeric/continuous scale
* Cross validation
  + How will you model handle new data when it’s not split?
  + Cross validation maximizes the availability of training data and splits data into various combinations and tests each specific combination
    - Two different methods: exhaustive cross validation and k-fold validation
  + Exhaustive cross validation
    - Test all possible combinations to divide the original sample into a training set and test set
  + *k*-fold validation (pg. 50 helpful diagram)
    - More common
    - Splitting data into *k* assigned buckets and reserving one of those buckets for testing the training model at each round
    - Steps
      * Data is assigned randomly to *k* buckets of equal size
      * One bucket is reserved as the test bucket and used to measure and evaluate the performance of the remaining buckets (*k*-1)
      * The cross validation technique is repeated *k* times (“folds”)
      * Repeated until all buckets have been used as a test set
    - By maximizing the training/test sets and averaging models otuputs you minimize the prediction error incurred on a fixed split (80/20)
* How much data do you need?
  + For machine learning a full range of feature combinations is ideal
  + At a minimum a dataset should contain 10x as many data points as the total number of features
    - i.e. small dataset with 5 features should have 50 rows
  + Datasets < 10,000 samples
    - Clustering and dimensionality algorithms are effective
  + Datasets < 100,000 samples
    - Regression analysis and classification algorithms
  + Neural networks > 100k
  + Scikit-learn has a great cheat sheet for matching algorithms to different datasets
    - <https://scikit-learn.org/1.3/tutorial/machine_learning_map/index.html>
  + You can also find tutorial of python implemented algorithms mentioned in the book here
    - <https://scatterplotpress.com/p/ml-code-exercises>

## Chapter 7: Linear Regression

* Supervised learning algorithm
* Predict an unknown variable using the results you do know
* Simple linear regression
  + Straight line to describe linear relationships
  + The technical term for a linear regression line is hyperplane
    - In a 2D space a hyperplane serves as a (flat) trendline
  + Goal: minimize the distance between data points and hyperplane
  + Residual/error = distance from best fit line to observed values
  + y = bx + a
    - y is the dependent variable
    - x is the independent variable
    - b is the slope and explains the relationship between x and y
    - a is the point where the hyperplane crosses the y-axis, the y-intercept
  + manual calculations
    - a =
    - b =
* Multiple linear regression
  + More applicable to machine learning
  + y = a + b1x1 + b2x2 + b3x3 + …
  + multiple independent variables each with their own coefficient
  + X and y values used from the training data to create a model, then multiple linear regression formula can be used to make a prediction (y) using the X values from the test data
* Output must be a continuous variable but input can be continuous or cateforical
  + Using one-hot coding (0/1 for female/male)
* If a strong relationship exists between two independent variables the issue of multi-collinearity arises
  + Tend to cancel each other out and provide the model with no information
  + Analyze relationship/correlation between independent variable beforehand

## Chapter 8: Logistic Regression

* Linear regression is good for predicting a continuous outcome, but logistic regression is good for predicting a categorical outcome
  + Will a customer return?
* Logistic regression is a supervised learning technique but predicts a qualitative outcome rather than quantitative
  + Fraud detection, disease diagnosis
* Sigmoid function
  + Produces S-shaped curve that can convert any number of map into a numerical value between 0 and 1 without ever reaching those exact limits
  + Independent variables are converted into an expression of probability in relation to the dependent variable
  + y =
    - e = Euler’s constant, 2.718
* logistic regression assigns each data point to a discrete class
  + when binary a cutoff can be at 0.5
* If the outcome is not binary you can do multinomial logistic regression
  + However, logistic regression’s core strength is with binary prediction. Otherwise, you may want to look at decision trees or SVMs
* Tips for logistic regression
  + Data should contain no missing values
  + Independent variables are not strongly correlated with each other
  + 30-50 data points as a starting point, doesn’t work well with large data sets

## Chapter 9: *k*-Nearest Neighbors (*k*-NN)

* Supervised learning algorithm, classification technique
* Classifies new data points based on their position to nearby data points
* Similar to a ‘popularity contest’
  + New kid at school
  + Looks around to see what everyone is wearing
  + Copy whatever the trend was nearest to you
* *k* determines how may data points you use to classify the new data point
  + the chosen number *k* is crucial to determine the results
  + it’s useful to test numerous *k* combinations to find the best fit
  + too low will increase bias and lead to misclassification
  + too high will make it computationally expensive
  + uneven number will help eliminate the possibility of a statistical stalemate
  + Scitkit-learn *k*-NN algorithm has *k* set to 5 by default
* The scale of individual variables has a large impact on k-NN
  + Data needs to be scaled to standardize variance
* Works best with continuous variables
* Not recommended for large datasets
* Challenging to apply to high dimensional data with a large number of features (columns)

## Chapter 10: *k*-Means Clustering

* Unsupervised learning
* Clustering data points that share similar attributes
* Attempts to divide data into *k* number of groups, effective at identifying new patterns
* How does it work?
  + Select a centroid for each cluster at random
  + Data points are assigned to the nearest centroid by measuring the Euclidean distance
    - The Euclidean distance between two points in Euclidean space is the length of the line segment between them
    - Euclidean distance can by calculated using Pythagorean theorem
  + After all data points have been allocated to a centroid, calculate the mean value of data points in the cluster
    - Uses the average to update centroid coordinates
  + Repeat until centroids don’t move
    - Multiple iterations
* Setting k
  + A scree plot can be used for guidance when determing what to set k equal to
    - A scree plot compares the sum of squared error (SSE) for each variation of total clusters
    - SSE is measured as the sum of the squared distance between the centroid and other neighbors in the cluster
    - SSE drops as more clusters are produced
    - Look for the elbow
    - Another useful way to determine k

## Chapter 11: Bias and Variance

* Hyperparameters = lines of code that act as the algorithm’s settings
* Bias = gap between the value predicted by your model and the actual patterns of the data
  + High bias – values skewed in a direction away from true values
* Variance = how scattered your predictions are in relation to each other
* Ideally, you want low variance and low bias
* Bias and variance both contribute to error but it’s the prediction error you want to minimize, not the bias or variance specifically
* Under and overfitting
  + Underfitting = low variance, high bias
    - Has not scratched the surface of underlying data
  + Overfitting = high variance, low bias
    - Natural temptation is adding more complexity to the model
* Linear regression rarely encounters overfitting but more susceptible to underfitting
* Switching from linear to non-linear regression reduces bias and increases variance
* Increasing k in k-NN minimizes variances (by averaging more neighbors)
* Reducing variance by switching from single decision tree to random forests with many decision trees
* regularization
  + advanced strategy to combat overfitting
  + makes the model simpler
  + this add-on hyperparameter amplifies bias error
    - must keep variance high while other hyperparameters are being tested
    - setting the regularization hyperparameter to high value avoids overfitting

## Chapter 12: Support Vector Machines (SVM)

* SVM is mostly a classification technique to predict categorical outcomes
* SVM is similar to linear regression in that its used to filter data into a binary or multiclass target variable, but there’s a different emphasis on the location of the classification line
  + A logistic regression boundary minimizes the distance between all data points and the decision boundary
  + A SVM boundary separates both classes and maximizes the distances between the boundary and the two classes of data points. SVM also has a margin to help support new data points.
  + Logistic regression goes out of its way to fit outliers, SVM is more sensitive to outliers and minimizes their impact
  + SVM can boundary can be modified to ignore misclassified data using a hyperparameter called C
* The tradeoff in SVM is between a wide margin/more mistakes and a narrow margin/less mistakes
  + Ideally, find a perfect balance between not too strict/loose
* modifying the C hyperparameter
  + ‘soft margin’
  + Low C value, closer to 0 = wider margin
  + High C value = cost of misclassification is high
* Kernel trick – used to classify high-dimensional data
  + Used when a linear line cannot be used for a decision boundary
* SVM is sensitive to feature scales and you made need to rescale the data prior to training
  + Using standardization, convert data to a normal distribution with a mean of 0
  + StandardScaler is used with Scitkit-learn
* SVM is not recommended for data with a low feature:row ratio
  + Remember features are columns

## Chapter 13: Artificial Neural Networks (ANN)

* Artificial neural networks, aka neural networks, is a popular ML technique for analyzing data through a network of decision layers
  + Named because if its resemblance to brain
  + Brain has interconnected neurons with dendrites that receive input. From the inputs, the neuron produces an electric signal output from the axon and emits signals through axon terminals to other neurons.
  + ANN consists of interconnected decision functions, known as *nodes*, which interact with each other through axon-like *edges*
* The nodes of a neural network are separated into layers and generally start with a wide base
  + First layer = consists of raw input data (numeric values, text, image pixels, sound) divided into nodes
  + Each input node sends info to the next layer of nodes via network edges
* Each edge in the network has an numeric weight that can be altered based on experience
  + If the sum of the connected edges satisfies a set threshold, known as the activation function, this activates a neuron at the next layer
  + If the sum of the activation function is nothe met, nothing happens
  + All or none
* Using supervised learning the models predicted output is compared to the actual output (that’s known to be correct) and the difference between the two is the cost value
* Back-propagation – process of training the neural network
  + Start from the output layer and work your way backwards
* The black-box dilemma
  + Can be accurate at predicting outputs but not tell you the relationship between variables
* When should you use a neural network then? Large number of features and complex patterns. Problems complex for a computer and easy for a human (i.e. CAPTCHA)
  + Getting a fast accurate answer is more important than the relationship between variables
* Building a ANN
  + Typical network divided into input, hidden, and output layers
  + Data is received by the input layer
  + The hidden layer analyzes and processes input features
  + The output layer shows the final result
  + As more hidden layers are added, model accuracy improves
  + Models with a deep number of layers is often referred to as deep learning
  + There are many ways to assemble a ANN
    - Simplest method is the feed-forward network
      * Signals flow in one direction, no loop
      * The most basic form is perceptron
        + Binary output
        + Sensitive to small changes
      * Sigmoid neuron
        + Values between 0-1
        + Flexibility to form small changes in edge weights
        + Con: unable to generate negative values
      * Hyperbolic tangent function
* Multilayer perceptrons (MLP)
  + Predict categorical or continuous target variable
  + Aggregate multiple models into a unified prediction model
  + Each node in the hidden layer is a sigmoid function
  + Output nodes are still binary
  + Ideal for large and complex data sets no computational or time restraints
* Deep learning (DL)
  + A ANN with a deep number of layers, can be used to interpret a high number of input features and breakdown complex patterns into simpler patterns
  + Stacking of at least 5-10 node layers
  + Object recognition with self-driving cars uses upwards of 150 layers
  + Often used to analyze data trends over a period of time
* Quiz
  + Q: what is a more transparent classification algorithm that we could use in place of multilayer perceptron (MLP)
  + A: Logistic regression
    - NOT linear regression because it used to predict continuous variables
    - NOT multiple linear regression because still continuous variables
    - NOT k-means clustering because it’s for identifying patterns
    - Logistic regression is used for classification

## Chapter 14: Decision Trees

* Although ANN is very powerful it’s not always the ‘cure all’. Sometimes decision trees trump ANN.
  + Cons of ANN: computational resources, can require tons of examples, black-box dilemma (model structure is concealed)
* Decision trees
  + Less data, less computationally demanding
  + Mainly used for classification problems but can also be used as a regression model to predict numeric outcomes
  + Examples: picking a scholarship recipient, predicting e-commerce sales, select the right job applicant
  + Can easily display decision trees graphically
* Building a decision tree
  + Root node, edges, leaves, branches
  + Goal is to keep tree as small as possible
    - Do this by splitting data into homogenous groups and minimize the level of data entropy on the next branch
    - Entropy is the mathematical concept that explains the measure of variance in the data among different classes
    - Want each layer to be more homogenous than the previous partition
      * Do so with a greedy algorithm
* Greedy algorithm
  + Recursive partitioning – splitting data into sub-partitions is repeated until a stop criterion is met
    - All leaves contain less than 3-5 items
    - Branch produces a result that places all items in one binary leaf
* Calculating entropy
  + Information value = entropy calculation number, measured in bits (using base 2 log)
  + Calculated using the data points at each node
  + (-p1logp1 – p2logp2)/log2
* Susceptible to overfitting the training data
  + The variable used to split the first round of data does not guarantee a more accurate model
  + Bigger consequences for splits higher in the tree
* Bagging
  + Growing multiple decision trees using a randomized selection of input data and averaging the output (for regression) or voting (classification)
  + Bootstrap sampling – needs to be an element of randomness
  + Helps handle outliers and reduces variance
* Random forests
  + Similar to bagging, but the number of variables in each split is capped
  + With bagging trees look similar because the use the same variable early in their decision structure
  + Random forests force a subset of variables to be used which gives other variables a greater chance of selection
  + Average unique and uncorrelated tress to give a less variable and more reliable decision
  + Since the model is trained using a subset of variables it’s considered a semi-supervised learning technique
  + Favors a high number of tress, 100+
  + Diminishing rate of effective when adding more trees, at some point focus on hyperparameters
* Boosting
  + Aggregates a large pool of decision trees
  + Combine weak models into one strong model
  + Add weights to a tree based on misclassified cases in the previous tree
    - Like teachers offer tutoring after students performed bad on a test
  + Gradient boosting is a popular boosting algorithm
    - Instead of choosing random variables, gradient boosting choose variables that increase accuracy in the next decision tree
    - Trees are grown sequentially
    - Mistakes recorded in the training are accounted for in the next iteration
    - Iterate until low error
  + Mitigates overfitting and uses fewer trees than random forests
    - With random forests you can add more decision trees to mitigate overfitting
    - This is not the case with boosting. This is because boosting learns from error. So, it can try to closely fit to outliers.
  + Slower than random forests
    - Random forests can be processed in parallel
    - Boosting is sequential
* Boosting, random forests, and bagging all lose the simplicity of visualization and interpretation

## Chapter15: Ensemble Modeling

* Ensemble modeling – combining algorithms and models to increase prediction
  + Sequential
  + Parallel and homogenous
  + heterogenous
* Sequential model – prediction errors are reduced by adding weights to classifiers that previously misclassified data
  + Gradient boosting, Adaboosting
* Parallel models – work concurrently and reduce error by averaging
  + Random forests
* Homogenous ensemble – same technique, numerous variations
  + Multiple decision trees working together for a single prediction
* Heterogenous ensemble – multiple techniques
  + k-means clustering + decision tree, NN + decision tree
  + it’s important for techniques to complement each other
  + NN require complete data whereas decision trees can handle missing values
    - Increases accuracy
    - Tradeoff is not easily interpretable
* 4 main methods of ensemble modeling
  + Bagging
    - Parallel model, averaging using a homogenous ensemble
    - Draws upon randomly drawn data and combines predictions to design a unified model
  + Boosting
    - Homogenous ensemble, sequential
    - Addresses error and data misclassified by the previous iteration
    - Gradient boosting, AdaBoost
  + A bucket of models
    - Heterogenous
    - Trains multiple different algorithms models using the same training data and picks the one that performed most accurately on the test data
  + Stacking
    - Runs multiple models simultaneously on the data and combines those results to produce a final model
    - Heterogenous
    - Does not assign equal trust to each model, does not average
    - Attempts to identify and add emphasis to well-performing mdoels
    - Smooth out error rate of models at the base level (level-0) before pushing those outputs to the level-1 model where they are combinaed and consolidated into a final prediction
    - Netflix held a competition for the best recommender system and this model one when combining 100s of prediction algorithms

## Chapter 16: Development Environment

* Jupyter Notebook – open-source web app that allows for editing and sharing of code notebooks
  + Can use the desktop version or command line version
  + I used it with a conda environment from the CL

## Chapter 17: Building a Model in Python

* Gradient boosting example, a supervised learning model, in Jupyter Notebook

## Additional Exercises/Notes from Author on Web

* Evaluation of classification
  + Accuracy
    - Accuracy = # correctly predicted / # predicted
  + Confusion matrix
    - Aka error matrix
    - Table that summaries the performance of the model including the # of false-positives and false-negatives
    - A red and blue squares

      Description automatically generated
    - Accuracy can still be calculated by (false-positive + false-negative) / total data points
    - (29 + 12) / (134 + 12 + 29 + 125) = 0.1366
  + Classification report
    - Report with three evaluation metrics: precision, recall, F-1 score
    - **Precision**
      * ratio of correctly predicted true-positives to the total of predicted positive cases.
      * A high precision score translates to a low number of false-positives.
      * This metric addresses the model’s accuracy at predicting a positive outcome and avoiding a false positive prediction.
    - **Recall**
      * ratio of correctly predicted true-positives to the total of actual positive cases.
      * This metric addresses the question of how many positive outcomes were rightly classified as positive?
      * numerator is the same for both precision and recall, while the denominator is different.
    - **F1-score**
      * weighted average of precision and recall
      * It’s typically used as a metric for model-to-model comparison rather than internal model accuracy.
      * generally lower than the accuracy score due to the way recall and precision are calculated.
    - **Support** is not an evaluation metric per se but rather a tally of the number of positive and negative cases respectively.
* Evaluation for Regression
  + Two most common are mean absolute error and
  + **MAE**
    - measures the average of the errors in a set of predictions
    - i.e. how far the regression line is to the actual data points
  + **RMS**
    - measures the standard deviation of the prediction errors, which informs how concentrated or spread-out prediction errors are in relation to an optimal fit
  + Given errors are squared before they are averaged, RMSE is far more sensitive to large errors than MAE
  + On the other hand, RMSE is not as easy to interpret as MAE as it doesn’t describe the average error of the model’s predictions.
  + Subsequently, RMSE is used more as a feedback mechanism to penalize poor predictions rather than to investigate the average error for each prediction.