Introduction to Machine Learning

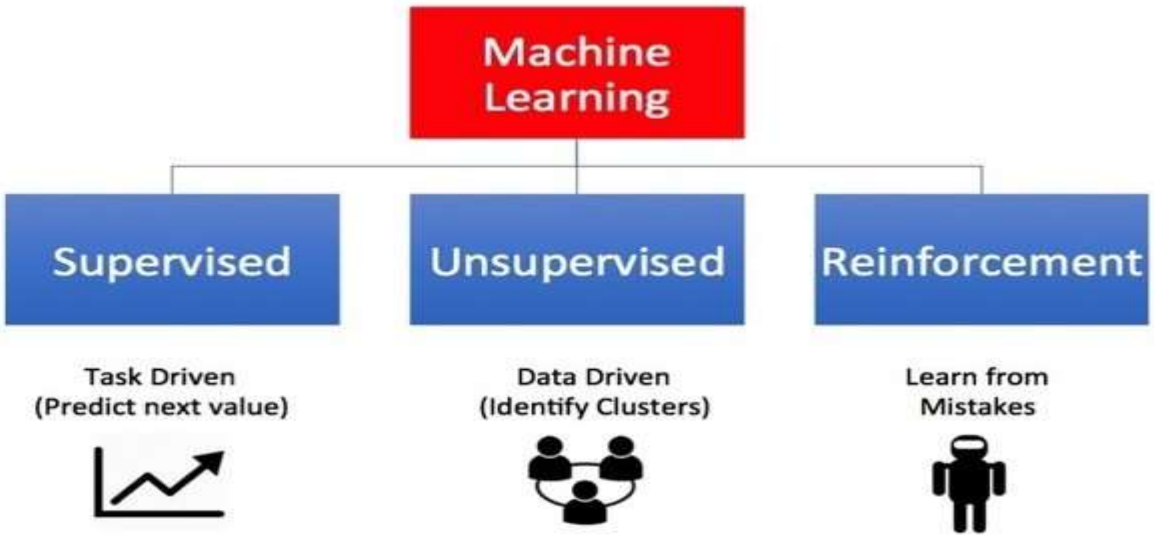
Artificial Intelligence Technique that enables machines to think and learn, making them smart to mimic human behavior

Machine Learning Subset of AI technique that uses statistical methods to enable machines to improve with experience.

Deep Learning Subset of Machine Learning that makes computation of multi-layer neural network feasible. It uses Neural Network to simulate human-like decision making.

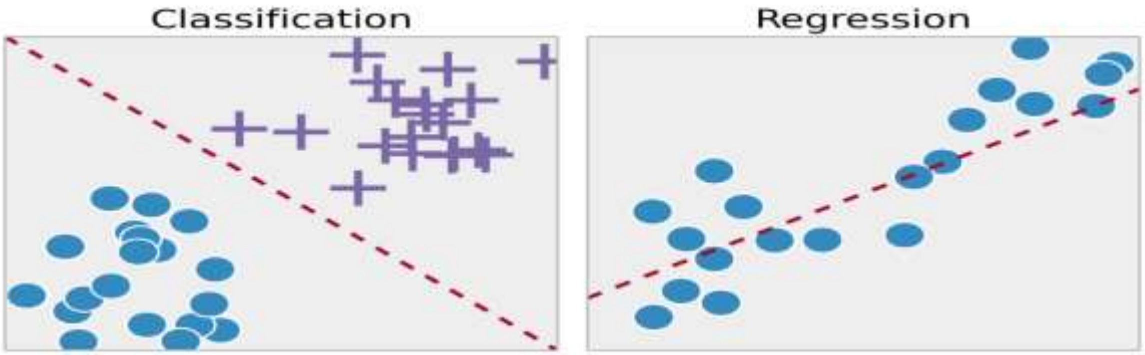
Deals with building and teaching a computer program or algorithm how to progressively/ iteratively improve upon a set task that it is given. On the research-side of things, machine learning is viewed through the lens of theoretical and mathematical modeling of how this process works.

Types of Machine Learning



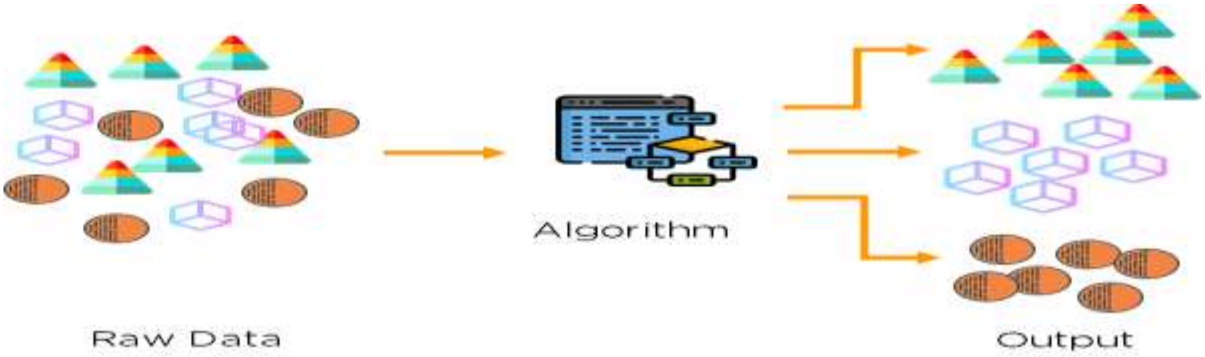
Supervised Learning

Given labelled data, we feed a learning algorithm these example-label pairs one by one, allowing the algorithm to predict the label for each example, and giving it feedback as to whether it predicted the right answer or not. ***Predicting Modelling, Spam Classification, Face Recognition***



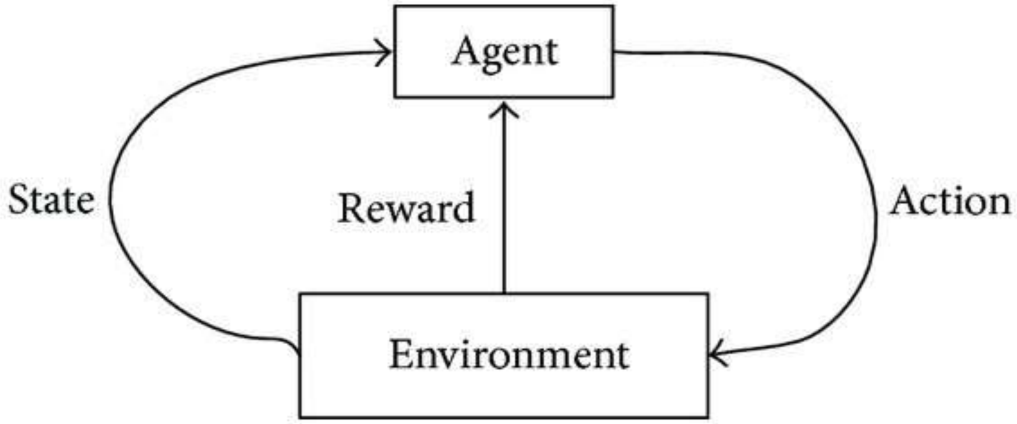
Unsupervised Learning: ***"The outcome or output for the given inputs is unknown",***

Data set features no labels. Instead, the algorithm is fed a lot of data and given the tools to understand the properties, associations and patterns of the data. From there, it can learn to group, cluster, and/or organize the data in a way such that a human (or other intelligent algorithm) can come in and make sense of the newly organized data. Clustering and Association Rules Problems: Anomaly Detection and used in Descriptive Modelling. Recommender Systems,



Semi-supervised Learning: It is in-between that of ***Supervised and Unsupervised Learning***. The combination is used to produce the desired results and it is the most important in real-world scenarios where all the data available are a combination of ***labelled and unlabeled data***.

Reinforced Learning: The model is exposed to an ***environment where it gets trained by trial and error method***, learning from mistakes, where it is trained to make a much specific decision. The machine learns from past experience and tries to capture the best possible knowledge to make ***accurate decisions*** based on the feedback received. Basic Reinforcement ***Markov Decision Process,*** most popular algorithms used: ***Q-Learning***, ***Deep Adversarial Networks.*** Its practical applications include computer playing board games such as ***chess*** and ***GO***, **Self-driving cars**, video Games, Industrial Simulation, Resource Management



Reinforced Learning workflow

We shall cover the core concepts that are at the heart of the Applied Machine Learning Field. Comprehensive understanding of these building blocks will help you build state of the art ML systems.

Suppose that we are designing a machine learning model. How do we arrive at the conclusion that a model is a good machine learning model?

By checking if it generalizes any new input data from the problem domain in a proper way.

Generalization in Machine Learning

This refers to the model's ability to adapt properly and give sensible outputs to set of new, previously unseen data, drawn from the same distribution as the one used to create the model. It checks how well the concepts learned by the model apply to specific examples not seen by the model when it was learning.

Generalization is bound by the **two undesirable outcomes —** high bias and high variance (To ***be Discussed Later***). Detecting whether the model suffers from either one ***is the sole responsibility of the model developer***.

The performance and application of the model relies heavily on the generalization of the model.

A model that generalizes well is a model that is neither underfit nor overfit.

Inductive Learning:

This is the learning of the target function (dependent variable/ target variable) from training data. Target Function/ variable is the variable whose values are modelled and predicted by the features.
Deduction: It is the other way round and seeks to learn specific concepts from general rules.

Purpose of Machine Learning

The goal of a good ML Model is to generalize well from the training data to any data from the problem domain. This allows us to make instance predictions in the future on data the model has never seen. We aim to achieve ***low bias and low variance*** and have the algorithm achieve good prediction performance

Building on this idea of how well an ML Model learns and generalizes to new data, we get introduced to two terminologies: Overfitting and Underfitting. They refer to deficiencies that the model's performance might suffer from. Ie" knowing "how off" the model's predictions are: knowing how close it is to overfitting or underfitting.

Statistical Fit: How well the approximation function matches the target function

Statistics often describe the goodness of fit which refers to measures used to estimate how well the approximation of the function matches the target function.

Overfitting

The model, models the training data so well and learns the detail and ***noise*** in the training data to the extent that it negatively impacts the performance of the model on new data. The irrelevant information or random fluctuations in the training data is picked up and learned as concepts by the model. The problem is that these concepts do not apply to new data and negatively impact the models ability to generalize. The overall cost is really small and the model generalization becomes unreliable. It happens when we train our model a lot over noisy dataset

It is more likely with nonparametric and nonlinear models that have more flexibility when learning a target function. As such, many nonparametric machine learning algorithms also include parameters or techniques to limit and constrain how much detail the model learns. ***(Check out the parametric and non-parametric machine learning models)***

For example, decision trees are a nonparametric machine learning algorithm that is very flexible and is subject to over fitting training data. This problem can be addressed by pruning a tree after it has learned in order to remove some of the detail it has picked up.

Specifically, overfitting occurs if the model or algorithm shows **low bias** but **high variance**.

Underfitting

This is the case where the model has " not learned enough" from the training data, and possibly not captured the dominant underlying trend of the many patterns thus resulting in low generalization and unreliable predictions. The model can neither model the training data nor generalize the new data

Specifically, underfitting occurs if the model or algorithm shows **low variance** but **high bias**

Underfitting is quite simple to overcome and can be avoided by using more data and reducing features by feature selection

It’s like, what if I send a 3rd grade kid to a Differential Calculus Class, the kid is only familiar with the basic arithmetic operations. That is what it is! If the data contains too much information that the model cannot take, the model is going to underfit for sure.

Happens when we have less training data but quite high amount of features, or when trying to build a linear model with a non-linear data. In such cases the rules of the machine learning model are too easy and flexible to be applied on such a minimal data and therefore the model will probably make a lot of wrong predictions.

A Good (Just) Fit in Machine Learning

Goal: Select a good model that’s at the sweet spot between underfitting and overfitting.

To understand this goal, we look at the performance of a ML algorithm over time during the model training process. Here we plot both the skill on the training data and the skill on a test dataset (held back during the training process).

Over time, as the algorithm learns, the error for the model on the training data goes down and so does the error on the test dataset. If we train for too long, the performance on the training dataset continues to decrease because the model is overfitting and learning the irrelevant detail and noise in the training dataset. At the same time the error for the test set starts to rise again as the model’s ability to generalize decreases.

The **sweet spot** is the point just before the error on the test dataset starts to increase where the model has good skill on both the training dataset and the unseen test dataset.

You can perform this experiment with your favorite machine learning algorithms. This is often not useful technique in practice, because by choosing the stopping point for training using the skill on the test dataset it means that the test set is no longer "unseen" or a standalone objective measure. Some knowledge (a lot of useful knowledge) about that data has leaked into the training procedure. (*We will cover **Data Leakage** in later topics*)

There are two additional techniques you can use to help find the sweet spot in practice: **resampling methods and a validation dataset.**

How to Limit Overfitting

Both overfitting and underfitting leads to poor model performance.

The **most** common problem in applied machine learning is over fitting: with a good performance metric it is easy to detect underfitting.

At times: overfitting is such a big problem because the evaluation of ML algorithms on the training data is different from the evaluation we actually care the most about, namely how well the algorithm performs on unseen data.

There are **two important** techniques that you can use when evaluating machine learning algorithms to limit overfitting:

1. *Use a resampling technique to estimate model accuracy.*
2. *Hold back a validation dataset.*

The most popular resampling technique is k-fold cross validation. It allows you to train and test your model iteratively k-times on different subsets of training data called folds, while using the remaining fold as the test set (called the "holdout fold") and building up an estimate of the performance of a machine learning model on unseen data.

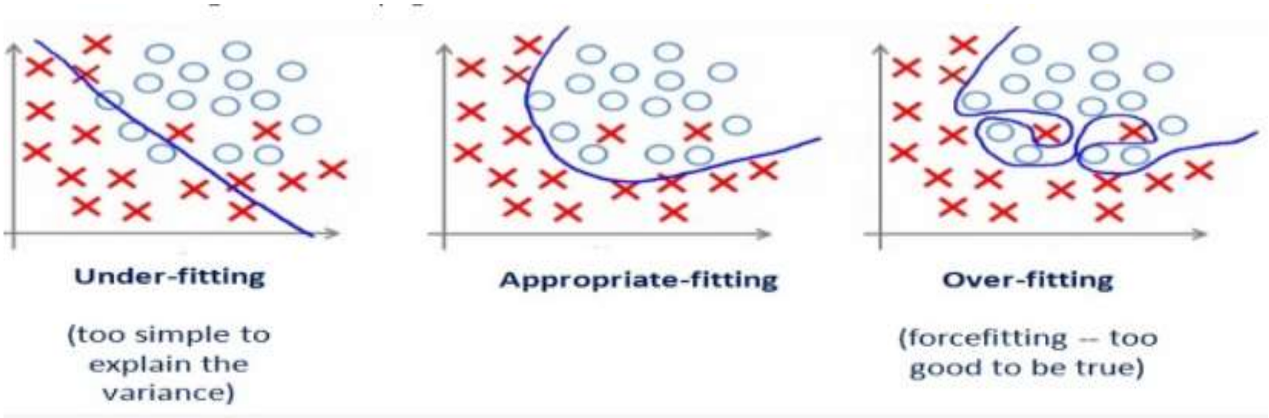
Cross-validation allows you to tune hyperparameters with only your original training set. This allows you to keep your test set as a truly unseen dataset for selecting your final model.

A validation dataset: this is a subset/ sample of your training data that you hold back from training your ml model that is used to give an estimate of model skill while tuning the model's hyper parameters

After you have selected and tuned your machine learning algorithms on your training dataset you can evaluate the learned models on the validation dataset to get a final objective idea of how the models might perform on unseen data.

Using cross validation is a **Gold Standard** in applied machine learning for estimating model accuracy on unseen data. If you have the data, always strive to subset a sample as a validation dataset: Excellent practice.

Check out other techniques used to limit overfitting in ML models: (Early Stopping, Regularization, Training with More Data, removing features using Feature Selection Methods)


Under-fitting
(too simple to explain the variance)
Appropriate-fitting
Over-fitting
(forcefitting -- too good to be true)

By looking at the graph on the left side we can predict that the line does not cover all the points shown in the graph. Such model tend to cause underfitting of data .It also called High Bias.

Where as the graph on right side, shows the predicted line covers all the points in graph. In such condition you can also think that it’s a good graph which cover all the points. But that’s not actually true, the predicted line into the graph covers all points which are noise and outlier. Such model are also responsible to predict poor result due to its complexity.It is also called High Variance.

Now, Looking at the middle graph it shows a pretty good predicted line. It covers majority of the point in graph and also maintains the balance between bias and variance.

The Bias-Variance Trade-Off in Machine Learning

Discover the Bias-Variance Trade-Off and how to use it to better understand supervised machine learning algorithms, minimize bias and variance to build accurate models and get better performance on your data.

Learn how to avoid the mistakes of overfitting and underfitting

These are the core parameters to tune while training a ML/ DL model.

Examples of Biases

The initial roll out of the Google’s Facial Recognition Feature: the users of varying faces were often incorrectly tagged as inhuman or ignored correctly.

Last year Amazon scrapped off their AI Hiring and Recruiting engine that showed gender and racial bias against women. They built the tool with the aims of reviewing job applicants’ resumes to mechanize the search for the top talent. But the system was not rating the candidates in a gender-neutral way.

This is a phenomenon when a model/ algorithm produces results that are systematically prejudiced due to erroneous assumptions in the **machine learning** process. Tendency of an ML model to **consistently learn the wrong relations** by **not taking in account all the features** given for the training.

Bias Error

- Bias is the simplifying assumptions made by the model to make the target function easier to learn and approximate.
  - Examples of low-bias machine learning algorithms include: Decision Trees, k-Nearest Neighbors and Support Vector Machines.
  - Examples of high-bias machine learning algorithms include: Linear Regression, Linear Discriminant Analysis and Logistic Regression.

A **ML model with high bias won't be able to learn relations between features effectively** and hence would **Underfit** on the dataset leading to **low accuracy while predicting**. The model misses relevant relations between the input features and the target outputs.

Variance: (variance in statistics is the same as variance in ML)

In the context of ML this is the **amount by which the target function changes** while it's being trained on data. Alternatively, it's the **flexibility of the Model to tune itself with the data points** in the given training dataset.

Variance Error

- Variance is the amount that the estimate of the target function will change given different training data was used. *Occurs due to a model's sensitivity to small fluctuations in the training set*.
  - Examples of low-variance machine learning algorithms include: Linear Regression, Linear Discriminant Analysis and Logistic Regression.

  - Examples of high-variance machine learning algorithms include: Decision Trees, k-Nearest Neighbors and Support Vector Machines.

An ML model with **High variance** causes it to **become highly flexible** with respect to the data points of the dataset. Such a condition causes a model to **Overfit** on the training data leading to low accuracy while predicting. Models with randomness and noise in detail

Bias-Variance Trade-Off

Trade-off is the tension between the error introduced by the bias and the variance and proves out to be the best way to ensure that model is **sufficiently fit** on the data and performs well on new data.

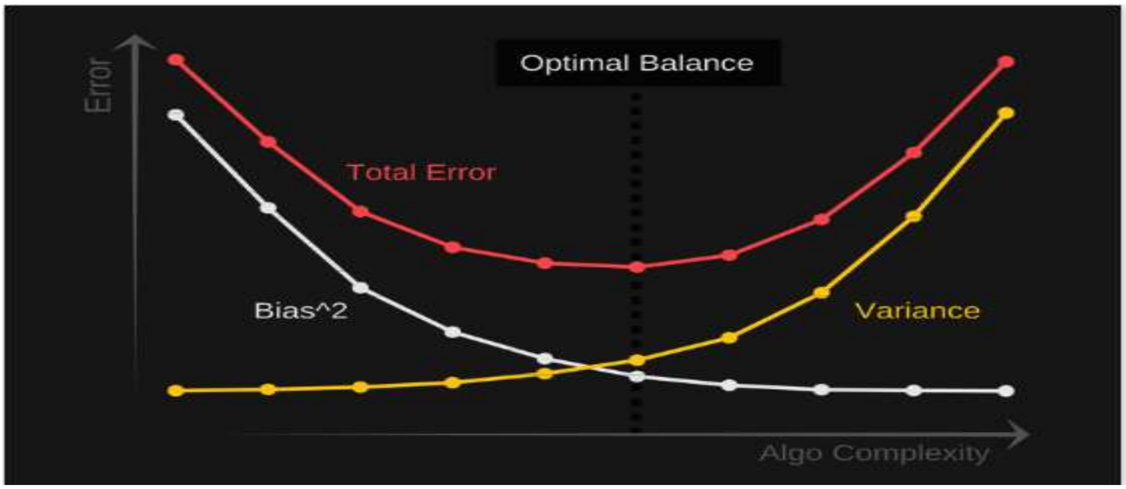
Parametric or linear machine learning algorithms often have a high bias but a low variance. Non-parametric or non-linear machine learning algorithms often have a low bias but a high variance.
Parameterization of ML algorithms is often a battle to balance out bias and variance.

If our model is too simple and has very few parameters then it may have high bias and low variance. On the other hand if our model has large number of parameters then it's going to have high variance and low bias. So we need to find the right/good balance without overfitting and underfitting the data.

This tradeoff in complexity is why there is a tradeoff between bias and variance. An algorithm can't be more complex and less complex at the same time.

Total Error

To build a good model, we need to find a good balance between bias and variance such that it minimizes the total error.



An optimal balance of bias and variance would never overfit or underfit the model. Therefore understanding bias and variance is critical for understanding the behavior of prediction models. Machine learning processes find that optimal balance:

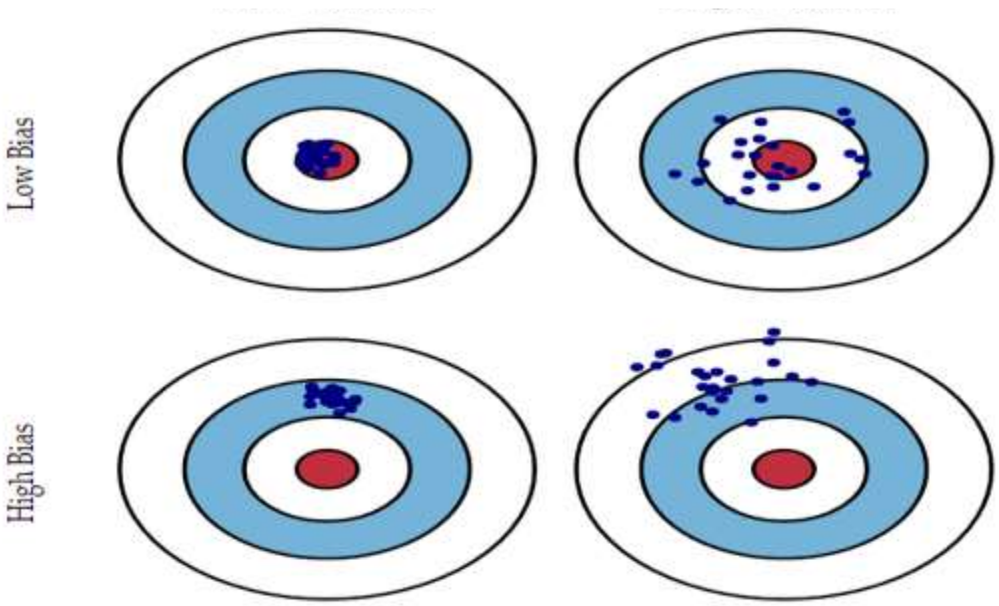
A model is good if it neither Underfits or Overfits.

A proper machine learning workflow includes:

- Separate training and test sets
- Trying appropriate algorithms (No Free Lunch)
- Fitting model parameters
- Tuning impactful hyperparameters
- Proper performance metrics
- Systematic cross-validation

Finally, as you might have already concluded, an optimal balance of bias and variance leads to a model that is neither overfit nor underfit:

Here's an image to help visualize the trade-off.



As you can see a **good model has low bias and relatively low variance**.

How to reduce variance and bias in machine learning models

https://machinelearningmastery.com/how-to-reduce-model-variance/

Tips, Tricks and Advice to starting out on Machine Learning

Career Paths to follow in Machine Learning

· **Research and Development -** learn the math first. In research the job is to come up with new, or substantially improve existing algorithms. Can’t do that without math.

Industry, applied - learn to program and handle data first because 90% of applied machine learning is data wrangling and programming. You should not stop there though. Data wrangling and machine learning without a good grasp on statistics is not a good combination.

- The foremost step to begin with ML is to get comfortable with a programming language, most preferably Python. Python has an easy learning curve and a host of useful libraries well-suited for different ML tasks.

- After you’ve become well-versed with programming, break the ice with the basic ML concepts including supervised learning, unsupervised learning, reinforcement learning and then move on to more complex ones like Deep Learning.
- Brush up on your Mathematics and Statistical skills. Revisit Linear Algebra, Multivariable Calculus, along with the fundamental Statistical concepts like probability distributions, statistical significance, hypothesis testing, variance and standard deviation, Bayesian theory etc.
- Once you feel like you’ve become comfortable with all the things mentioned above, try your hand at practical applications of ML.
- Experiment with ML algorithms and build simple applications/projects.
- Participate in Kaggle/ Zindi and other online competitions. These competitions help you assess your strengths and weaknesses compared to your rivals and also are a great opportunity for landing jobs - top recruiters regularly monitor online competitions in the field.

Golden Advice

You must always harbor the will to up skill - read Data Science articles, subscribe to newsletters, attend conferences/seminars, etc.

Newsletters to subscribe to: Towards Data Science, Analytics Vidhya, Data Science Central, Reddit. They'll keep you updated with latest improvements and algorithms taking place currently, and in the field of machine learning, the person most updated with these news is the winner. So keep reading, and keep on learning and enhancing your skills.

Recommendations

I would highly recommend going through the following amazing books

- Master Machine Learning Algorithms by Jason Brownlee.
- Hands-On Machine Learning with Scikit Learn and Tensorflow: Concepts, Tools and Techniques to Build Intelligent Systems by O’ Reilly.