Machine Learning Documentation

Sunpreet Singh Chahal

Table of Contents:

1. [Foreword](#Foreword) (Pg. 3)
2. [Introduction](#Introduction) (Pg. 4 – Pg. 6)
   1. [Model Training](#ModelTraining)
   2. [Machine Learning vs. Deep Leaning vs. Neural Networks](#MLvsDL)
   3. [Challenges of Machine Learning](#Challenges_of_ML)
3. Machine Learning Dictionary (Pg. X – Pg. X)
   1. [General Fundamentals](#GeneralTerms)
      1. [Model Parameters](#Model_Parameters)
      2. [Model Hyperparameters](#Model_Hyperarameters)
4. [Machine Learning Models](#ML_Models) (Pg. X – Pg. X)
   1. Supervised
      1. Regression
         1. [OLS](#Regression_OLS)
         2. [Ridge](#Regression_Ridge)
         3. [Lasso](#Regression_Lasso)
         4. [Decision Trees](#Regression_DecisionTrees)
         5. [Random Forest](#Regression_RandomForest)

Foreword:

After spending some time on learning a handful of machine learning models, I realised that I had made progress on learning the code for some of the models, however, I made minimal progress on understanding the models well enough to where I could converse the results that the model produces.

I will still continue to play around with various models to develop my practical knowledge within this domain.

The end-goal of the document is to gain a deeper and more well-rounded understanding of machine learning.

*This document’s purpose is to serve as personal journaling of what I’ve learned so-far and for personal reference.*

*Some references include:*

* *IBM*
* *StatQuest*

***GitHub Repository - ML Model Overview & Documentation:***

[*SunpreetSChahal/MachineLearning-Learning: Indapendant ML Learning - Reference (github.com)*](https://github.com/SunpreetSChahal/MachineLearning-Learning)

***GitHub Repository - Separate Exemplar ML Projects:***

[*SunpreetSChahal/ML\_Practice\_Projs (github.com)*](https://github.com/SunpreetSChahal/ML_Practice_Projs)

Introduction:

A branch of artificial intelligence; machine learning uses data and algorithms to train computers to imitate how humans think. Using patterns within the data to produce hidden insights, gradually improving the accuracy of the models as new data comes in.

Here I’ll cover some general topics regarding machine learning, such as model training, differentiating ML vs. Deep Learning and the challenges of ML.

**Model Training:**

As data gets fed into the model, its accuracy improves – this is called training a model. Training your model is a key step in the process, this step allows you to, 1) see if your model is providing you your desired result and 2) if the model is getting increasingly accurate.

For supervised learning, your training set must have a desired value or *target value*, which the model will aim to produce by finding patterns within the provided dataset. Your training set is your “gold standard” – you fit your model using this set.

Once fitted you feed your model a validation/test set. This is where you test how well you model has been fitted. This could be through splitting your entire data set into training and test sets or by having dedicated sets for each function. If splitting the data; do a split of 70:30 (70% of the data is for testing and 30% of the data is for training.)

For example, you have a model which will predict the species of various animals. You train your model to take in various inputs of the animals’ features and get a desired result of what species each animal is. Once fitted, you then provide the model with random set of features and see whether the model produces an accurate result.

As big data grows year over year, it is no surprise that data scientists are in high demand to be able to gain business insight with the use of machine learning. The amount of data isn’t project to slow down anytime soon, so, this demand is set to climb at a consistent pace.

**Machine Learning vs. Deep Learning:**

Often used interchangeably, though are quite different. Machine learning is a sub-field of AI, Deep Learning is a sub-field of ML and Neural Networks are a sub-field of DL.

Machine learning has significant human intervention. We set the features of the models. With the case usually being that structured data is required.

*Deep learning* has the ability to eliminate most of the human interaction that traditional ML requires. Though deep learning models can benefit from using labelled data, unstructured data can still be used effectively. DL models can automatically find patterns within the input data, highlighting differentiations without human interference. In areas such as NLP, facial recognition and speech recognition, deep learning excels.

\*\*Neural Networks\*\*

**Real-World Machine Learning Application:**

\*\* Write up some examples – IBM \*\*

**Challenges of Machine Learning:**

Other than technical challenges of machine learning, such as underfitting models, overfitting models or a lack of good quality data – there are grander, ethical issue, which we’ll focus on in this section.

*How far should we let ML go?*

Issues around letting ML run wild is a genuine concern. For example, with fully autonomous cars, what’ll happen when the cars inevitably get into an accident? Currently, whoever’s behind the wheel is liable, this can’t be the case with autonomous cars. How much we should *limit* ML is a real concern, making sure we are still in control.

*How will ML effect the job market?*

A concern regarding ML taking over a wide array of job has been in the minds of everyone. Will ML take over my role? Will there be a point where there are no jobs to do? No, at least, as far as we know now. There will always be talent required for physical work, as well as, talent to manage systems across businesses. Machines are efficient as producing an output based on parameters and the input, however, only humans are capable of extracting *deep* value from the output – *What is going on? Why is this output significant? How could we aim for a better output?* Etc.

*How do we deal with model bias?*

Use of ML for mass surveillance, monitoring, profiling and violations of human rights is strongly disagreed with. Though, we can’t say there *won’t* be wrongful use of ML, we can remain confident that reputable businesses are using ML for the benefit of the customer and to keep healthy market competition. If there are any hints of discrimination or bias the projects are to be scrapped immediately, e.g. Amazon hiring candidates for technical roles, however, unintentionally had gender bias within their selection pool – this process was scrapped shortly.

ML Model Fundamentals:

In this section I’ll cover various important parameters and keywords relevant to machine leaning.

**General Terms:**

* **Underfitting Model**: A model that is too simple for a dataset that is too complex. [SSE](#SumOfSquaredErrors) is very high, doesn’t work well with the training set, and by proxy, the testing set.
* **Overfitting Model**: A model that a very small amount [SSE](#SumOfSquaredErrors) with the training set. The model learns the training data very well, understanding its patterns completely, but fails to be able to generalize; won’t perform well against testing data. Doesn’t give you an idea, which ML is all about. The goal of our model should be to generalize.
* **Variance**: Difference in how well the model fits between different datasets (training & testing sets).

A good model will have moderate **SSE** with **low variance**. A model that is blended between complex and simple.

**Model Parameters:**

Model parameters are variables that are used to fit the model. They are there to make sure your model is w*orking* – though not yet optimised. That is where *hyperparameters* come in.

**Model Hyperparameters:**

Model hyperparameters are variables that are used to fine-tune and optimise your model for the best model performance. Altering the learning process of the model.

It is important to consider hyperparameters to make sure you’re getting optimal model performance.

Examples of Hyperparameters:

* Train/Test splits
* Learning rate (Gradient Descent)
* n\_estimators (Random Forest)

\*\*For each hyperparameter that is specific for a specific model or function; hyper link it to that function. E.g. n\_estimators for Random Forest; hyperlink *Random Forest* to its specific model description. \*\*

ML Models:

In this section I will cover various machine learning models. For each model I will cover:

* What the model is.
* The use case for the model.
* Key hyperparameters for the model – if applicable.
* Key documentation for the model.

*To find reference examples of each model, refer to the* [*foreword page*](#Foreword)*, where my relevant GitHub links can be found.*

There are three types of machine learning models: Supervised, Unsupervised & Reinforcement Learning. Detailed notes on each class of model can be found in their respective sections.

Diagram

Description automatically generated

**Supervised Learning Models:**

A model that uses labelled data to train the machine. The output(s)/target(s) are known, the machine just needs to class the input accordingly.

For example, our data has two columns: experience (explanatory) and salary (response). We can use a supervised model to predict; what our salary would be given *x* years of experience.

There are two sub-categories within supervised learning – Regression & Classification models.

**Regression Models:**

Regression models are best utilised where the response variable is continuous.

* Time
* Money

**Ordinary Least Square Linear Regression:**

Formula; y = mx + b

**What is OLS Linear Regression?**

OLS finds a line that results in the minimum [SSE](#SumOfSquaredErrors).

Chart, scatter chart

Description automatically generated

In the example above, our linear regression model takes the explanatory value of “Area (Square meters)” against the response value of “Price ($)”. The model fits a line that encapsulates as many data points as possible and allows us to accurately predict what the price of a property would be relative to a given area size.

**What is the use case for OLS Linear Regression?**

Linear regressions are very popular due to their relative simplicity and ease of training. The use cases are very broad; if you have linear (sequential/homoscedastic) data, you can utilise linear regression – a scatter plot can help you find out if your data is linear or not.

The larger the dataset, the more squared residuals you can measure and in-turn a better fitted line, if you have less data…\*\*\*

**What are the key hyperparameters for this model?**

N/A

**Key documentation:**

[SkLearn – OLS Linear Regression Documentation](https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.LinearRegression.html?highlight=linear%20regression#sklearn.linear_model.LinearRegression)

**Ridge Regression:**

Formula; [SSE](#SumOfSquaredErrors) + ([λ](#Lambda) x *Slope2*)

* *Slope*: y = **m**x + b

Essentially, OLS + Ridge penalty ([λ](#Lambda) x *Slope2*)

**What is Ridge Regression?**

When you suspect that your [OLS regression](#Regression_OLS) line is overfitting the training data, ridge regression may come in handy. By introducing a s*mall* amount of bias, we can fit a better, more generalized line for the training set.

The small amount of bias, though dropping the accuracy with the training set, improves long-term performance – by dropping variance.

**What is the use case for Ridge Regression?**

Ridge regression may be used when you suspect overfitting with OLS. Using this method, we can introduce some bias and lower the variance of the fitted line.

**What are the key parameters for this model?**

* [λ](#Lambda): fine tune this value to get the best RR line.
  + To get the best value, cross validation can be of help.
    - *New section for Cross Validation not yet complete.*
* \*\*\*Refer to documentation

**Key Documentation:**

[SkLearn – Ridge Regression Documentation](https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.Ridge.html)

**Lasso Regression:**

Formula; [SSE](#SumOfSquaredErrors) + ([λ](#Lambda) x |*Slope*|)

* *Slope*: y = **m**x + b
* |*Slope*|: The absolute value of the *slope*

Essentially, OLS + Lasso penalty ([λ](#Lambda) x |*Slope*|)

**What is Lasso Regression?**

Though quite similar to [Ridge Regression](#Regression_Ridge), Lasso Regression can shrink it’s slope to 0, where as Ridge Regression can only shrink close to 0.

This due to Lasso Regression using the absolute value of the *slope*.

**What is the use case for Lasso Regression?**

Where we may have useless features for our model that don’t help us with our problem, Lasso Regression makes it so we can remove these useless features. By eventually withering them away; as [λ](#Lambda) increases, the value of the features nears 0.

**What are the key parameters for this model?**

* \*\*\*Refer to documentation

**Key Documentation:**

[SkLearn – Lasso Regression Documentation](https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.Lasso.html)

**Decision Trees - Regression:**

**What are decision trees?**

As its name suggest, this model produces a result or *decision* using a tree. A tree in the sense of starting with the input data and branching out into the different features within the data.

**What is the use case for decision trees?**

* **\*\*\***Refer to source on PC

**What are the key parameters for this model?**

* **\*\*\***Refer to documentation

**Key Documentation:**

[SkLearn – Decision Tree (Regression) Documentation](https://scikit-learn.org/stable/modules/generated/sklearn.tree.DecisionTreeRegressor.html)

**Random Forest - Regression:**

\*\*Bootstrapped datasets\*\* / \*\*Bagging\*\*

**What is Random Forest?**

Using ensemble learning (multiple learning algorithms), Random forest models build upon [decision trees](#Regression_DecisionTrees). Instead of a single tree, this model uses multiple – hence the name, Random *Forest*.

This model uses various, smaller samples of the same dataset and finds the mode of each result. For example, there are 4 trees, 3 produce the result “*2*” and 1 produces the result “*15*”. The final output would be *2*. The usage of a *popularity* contest between each tree results in improved predictive accuracy and minimizing over-fitting your model

**What is the use case for Random Forest?**

Whenever you’d use a decision tree, Random Forest is, more often than not, the better choice.

If you have time and a larger dataset, Random Forest is guaranteed to produce a better result, compared to if you were to use a decision tree.

**What are the key parameters for this model?**

* \*\*\*Refer to documentation

**Key Documentation:**

[SkLearn – Random Forest (Regression) Documentation](https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestRegressor.html)

**Classification Models:**

Classifications models are models that put the response variable into a category/class.

* Vehicle type
* Email spam or not

Mathematical Theory Behind ML

In this section I will be looking at the mathematical theory behind machine learning. Having a good understanding of the mathematical theory allows you to; know what model to use in any given problem/situation. Understand how the model works – not just the code, but why the model does what it does. Understanding what the parameters are, that allow you to fit the most optimised model.

**Terms:**

* **λ/Lambda**: Any value from 0 to infinity
* **Sum of Squared Errors**: (SSE) Measure the distances from the fitted line to the data point, square them, then sum them. *Errors* refers to the data points the fitted line missed, this gives us an idea of how well or badly our model is fitted – refer to over/underfitting. *Squaring* making sure the negative distances don’t overwrite the positive distances.

**Linear Algebra:**

xxx