**Exercise #1: Using Machine Learning Algorithm to predict Heart Disease**

**Group:**

|  |  |  |  |
| --- | --- | --- | --- |
| **#** | **Name** | **Student ID** | **Responsibility** |
| **1** | **The Cuong, Bui** | **C0770339** |  |
| **2** | **Chiranjeev Singh** | **C0760944** |  |
| **3** | **Gurpreet Singh Virdi** | **C0762178** |  |
| **4** | **Roshan** |  |  |
| **5** | **Manoj Moond** |  |  |

Abstract— Machine learning is playing an important roles in different edges of life and business. The theory of statistics and mathematical can be applied to predict the results in real life. Big Data and Data Science is developing bring huge demands for machine is growing tremendously to the explosion in data volume. Information Technology industry is continuously growing forward to fulfil the demand of Big Data tools and techniques as well as platforms to support the majurity of Machine Learning in applications. This exercise is dealing with Heart Disease prediction using Linear Regression learning and Python library for Data Science in Health Care fields. At the result, we will come to the performance analysis of the machine Learning model. Keywords: Machine Learning, Supervised Learning, Logistic Regression Algorithm

1. **Introduction**
2. **Machine Learning Introduction**
3. **Programming Language and tools used for exercise.**
4. **Understanding Features and data sets of heart diseases.**
5. **Process to build Machine Learning Model to predict Heart Diseases**
6. **Introduction:**

Heart diseases reflects a various conditions in which can affect human’s heart.

Diseases relates to the heart disease are various include coronary artery disease (blood vessel diseases), arrhythmias (heart rhythm problems) and congenital heart defects (heart defects you're born with) and so on.

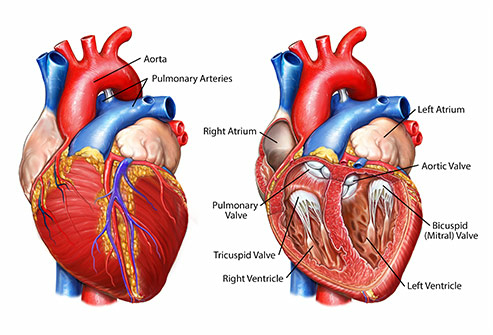


Figure 1

Figure 1: Heart parts

Heart diseases have been considered as the serious causes of high death rate of human life. Hence scientist has been spent a lots of effort in researching the effective methods to quickly diagnostic and prevent heart disease and it has become more than necessary. With a huge data from large number of hospitals and clinics in all over the world, it can help to build a good data-driven systems to analyze and predict heart diseases and then can make the research and prevention process more effective. From that it can ensure that more people can live a healthy life.

With the development and maturity of Big Data along with new technologies such as Cloud Computing, nowadays, Machine Learning is attracting data scientists apply across many spheres around the world. The healthcare industry is one of the realistic cases in which Machine Learning can play an essential role in predicting presence of Locomotor disorders, Heart diseases and so on. And machine learning the predictions made by Machine Learning are quite accurate and promising coming more accuracy. With the maturity and increasing experience of Data Scientist in applying high technology in researching heart diseases, it is promising that Machine Learning can provide a pivotal insights to hospital or doctors to adapt their diagnosis and treatment method for each individual patient.

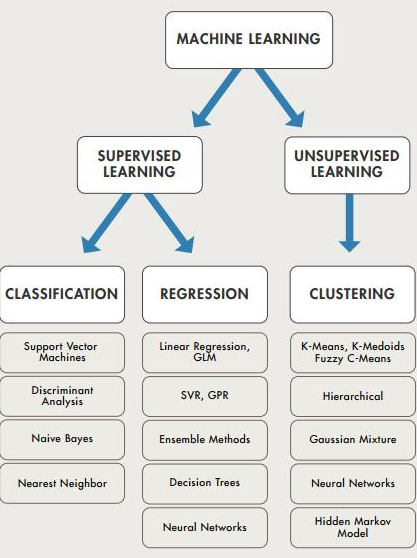
Regarding to above reasons, our group are pleased to apply Machine Learning to predict heart disease as an exercise of Big Data Fundamental course of fall 2019 at Lambton College. This is not only to help us enrich knowledge, experience in Machine Learning as the objective of the course but also help us to have a right feeling that we are going to the right course in which we can participate in the most top health problem of the world in which we can contribute our knowledge to insights for society.

1. **Machine Learning Introduction:**

**MACHINE LEARNING**

Machine Learning can be characterized as a field of concentrate that enables to the system for self-learn without being expressly modified. It tends to be considered as a part of computerized reasoning dependent on the possibility that frameworks can gain from information, recognize patterns and settle on choices with negligible human mediation.

Two of the most widely adopted machine learning methods are supervised learning and unsupervised learning which are explained further below.



**Supervised Learning**

Supervised Learning is a sort of framework wherein both input data and desired yield information are provided. A supervised learning calculation takes a known dataset of informational collection and its known reactions to the information (yield), accordingly it is regularly utilized in applications where recorded information predicts likely future events. For instance, it can foresee when credit card exchanges are probably going to be fraudulent or which insurance client is probably going to record a claim.

Supervised learning has examples of input out pairs and with the help of these historical data predictions are made.

Supervised learning can be mainly grouped as follows.

**Classification**

Classification can be described as a predictive modelling wherein categories or labels are predicted.

It includes the process of predicting an output (Z) by approximating a mapping function ( lets say (F)) from the given input data and there by giving a discrete value as output

**Regression**

Regression is yet another method in supervised learning which has the same characteristic as classification however when an output (Z) is predicted a continuous value is obtained as output which is mostly numeric.

**Un**s**upervised Learning**

Unsupervised learning is used against data that has no historical labels. This type of learning analyses the data in the first instance which helps to identify certain patterns and similarities which in turn will lead to successful predictions. Unsupervised learning algorithm has a tendency to perform well while using with transactional data.

For instance similar customers with identical traits are easily identified. As a result a common methodology can be used in order to work with them during marketing campaigns. As a result this will give maximum benefits and in turn will boost the business.It will also help us to understand the differences in traits that separates individual customers or consumer segments from each other**.**

The Algorithm of unsupervised learning can be further divided as follows:

* Clustering
* Association

**Clustering**

By definition Clustering is nothing but the process by which a datasets are classified in such a way that objects in a particular group depict more resemblance with each other while compared to objects in a different group.

Here the term group can be referred as a cluster and hence the terminology clustering was coined as a standard.

**Association**

Association is yet another part of unsupervised machine learning which is based on finding identical traits and similar relations or characteristics among different variables in database.

Association basically is utilized to obtain common dimensions in any given database.

**Tools to be used:**

**Programming Language:**

Among Python, R and other programming languages, group decided to use Python for our exercise as it can help us learn practical language for our future work.

**Jupiter:**

A popular interface for Machine Learning. Using notebook format with input cells contains code and output cells contains the results of code.

**Advantage:**

It can iterative code quickly, instantly visualizing the result of modification.

**Pandas:**



*Figure 6*

Is the reference module to efficiently manipulate millions rows of data in Python.

**Scikit learn:**

****

Figure 7

Is one of the reference modules for Machine Learning in Python.

**Numpy and SciPy and Matplotlib:**



Figure 8

These are more convenient modules for data computerization and data visualization.

**Understanding Data sets of Heart Diseases:**

For this exercise, we used the most recommended data set called Cleveland Heart Disease from a study of heart disease from UCI Machine Learning.

Repository which is maintained by the Center for Machine Learning and Intelligent Systems of the University of California, Irvine.

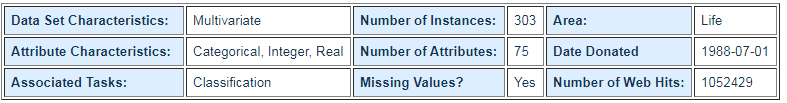


Figure 9

In particular, the Cleveland data set is the only one that has been used by Machine Learning researchers to this date. The main target of most of the research is to detect the presence/absence of heart disease from patient’s data. Experiments with the Cleveland data sets have been concentrated on simply attempting to distinguish presence (values 1,2,3,4) from absence (value 0).

For exercise purpose and initial start, our group select the “processed” of Cleveland data to ensure higher accuracy of prediction.

The dataset consists of 76 attributes and 303 individual data. However, all published experiments refer to using a subset of only 14 of them, which are described below.

1. *Age* : the most important risk factor in developing cardiovascular or heart diseases. It is estimated that the high risk of coronary heart disease and stroke for people over 55 or 65 and older.
2. 2. Men is considered greater risk of heart disease than pre-menopausal women. However, If a female has diabetes, the higher possibility to develop heart disease than a male. *Sex* : 1 = male, 0 = female
3. Angina *(Chest-pain type)* : when heart muscle doesn’t get enough oxygen-rich blood, it causes Chest pain or discomfort. 1 = typical angina, 2 = atypical angina, 3 = non — anginal pain, 4 = asymptotic
4. *Resting Blood Pressure* (in mmHg (unit)): high blood pressure can damage arteries that feed heart over time.
5. *Serum Cholestrol (* mg/dl (unit)): A high level of low-density lipoprotein (LDL) (“good”) cholesterol is most likely to narrow arteries, a high level of high-density lipoprotein (HDL) cholesterol (“good”) lowers your risk of heart attack.
6. *Fasting Blood Sugar* : blood sugar levels in human’s body can rise up if it is not produced enough of a hormone secreted by body’s pancreas (insulin). If fasting blood sugar > 120mg/dl then : 1 (true), else : 0 (false)
7. *Resting ECG(*electrocardiographic): 0 = normal, 1 = having ST-T wave abnormality, 2 = left ventricular hyperthrophy
8. *Max heart rate achieved*: The increase in the cardiovascular risk, associated with the acceleration of heart rate.
9. *Exercise induced angina* : The pain or discomfort associated with angina usually feels tight, gripping or squeezing, and can vary from mild to severe. 1 = yes, 0 = no
10. *ST depression induced by exercise relative to rest* : displays the value which is integer or float.
11. *Peak exercise ST segment* : A treadmill ECG stress test is considered abnormal when there is a horizontal or down-sloping ST-segment depression ≥ 1 mm at 60–80 ms after the J point. 1 = upsloping, 2 = flat, 3 = downsloping
12. *Number of major vessels (0–3) colored by flourosopy*.
13. *Thal* (thalassemia): 3 = normal,6 = fixed defect, 7 = reversible defect
14. *Diagnosis of heart disease*: 0 = absence, 1, 2, 3, 4 = present.

**Process to build Machine Learning Model to predict Heart Diseases**

Building Machine Learning process

To build Machine learning, it should consider the following procedures

* Data Preparation
* Feature Engineering
* Data Modeling
* Performance Measure

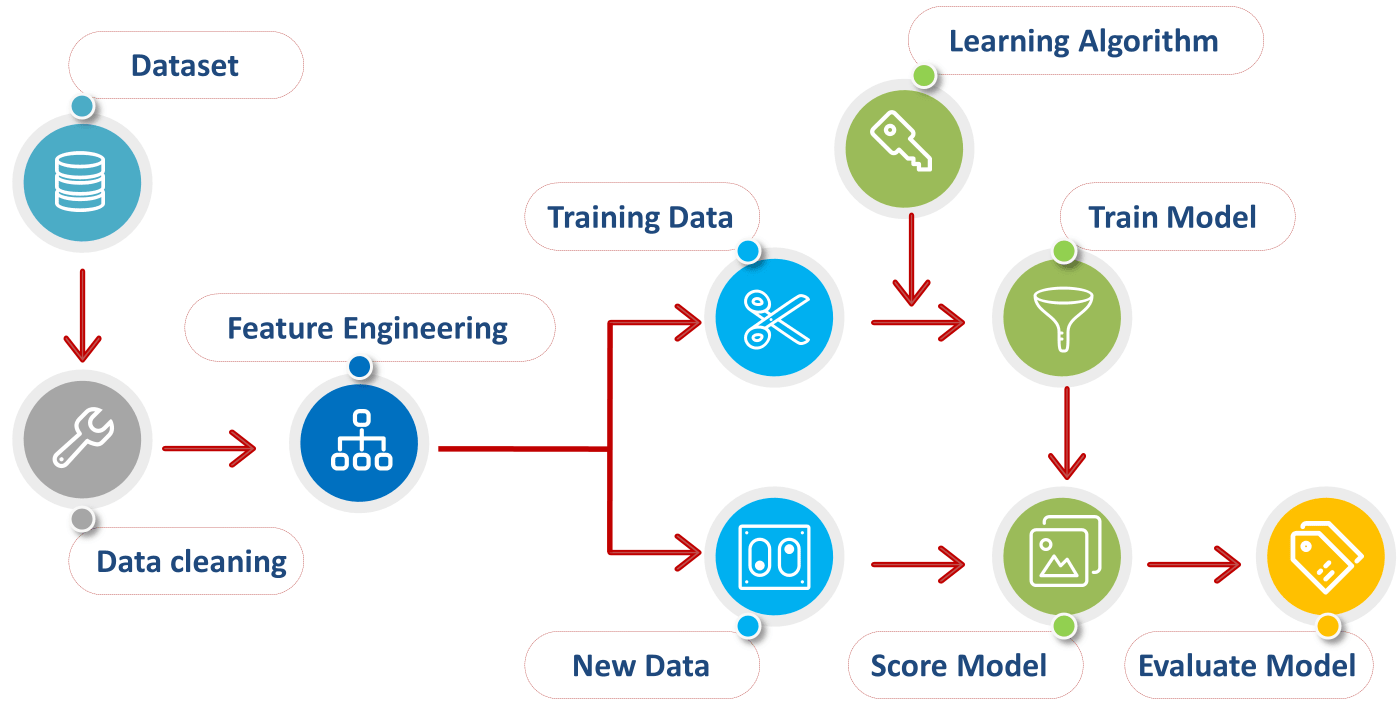


Figure 2

**However, not the model performance does not always achieve the best performance at first. Sometimes, it retains the low performance and need to be iteratively rebuild the model. It comes to a new process called Iterative Machine Learning Process. In that, it improves the performance through the Process of building Machine algorithm over and over until it achieves the expected performance.**

Figure 3

**Data Preparation:**

Data comes to the use of Machine Learning is not always meaningful, it requires several preprocessing to be using for Data modeling. In this stage, it includes import raw data, clean data with missing values, un format fata and outliers.

**Query data**: using pandas library to read csv

*url = "https://archive.ics.uci.edu/ml/machine-learning-databases/heart-disease/"*

*data\_file = "processed.cleveland.data"*

*missing\_values = ["n/a", "na", "--", "?"]*

*features = [ "age", "sex", "cp", "trestbps", "chol","fbs", "restecg",*

*"thalach","exang", "oldpeak","slope", "ca", "thal"]*

*target = "num"*

*method = 'median'*

*# Importing the datasets*

*Dataframe = pandas.read\_csv(url + data\_file, names=features + [target], na\_values = missing\_values)*

**Clean data**: using sklearn library to preprocess the data

from sklearn.preprocessing import Imputer

imputer = Imputer(missing\_values = 'NaN', strategy = 'mean', axis = 0)

imputer = imputer.fit(X[:, 0:13])

X[:, 0:13] = imputer.transform(X[:, 0:13])

In this exercise, the imputation strategy is mean. Other strategies include median, most frequent

**Remove outliers:**

**Format data: to modify data to fit constraints of algorithms.**

**The most common transformation is the encoding the categorical variables. Besides, it also has Replacing values, One-Hot encoding, Binary encoding, Backward difference encoding, Miscellaneous features**

**Label Encoding is to encode categorical values with a technique called “Label encoding in which it convert every value in a feature to a number. Number labels are always between 0 and n\_categories-1. Using method “.cat.codes” of Pandas Data frame.**

**# Label Encoder**

**number = LabelEncoder()**

**for feature in features:**

**X[feature] = number.fit\_transform(X[feature].astype('str'))**

## ****Step 5: Splitting the Data set into Training set and Test Set****

Now we divide our data into two sets, one for training our model called the**training set** and the other for testing the performance of our model called the **test set**. The split is generally 80/20. To do this we import the “train\_test\_split” method of “sklearn.model\_selection” library.

1. **X\_train** (training part of the matrix of features),
2. **X\_test** (test part of the matrix of features),
3. **Y\_train** (training part of the dependent variables associated with the X train sets, and therefore also the same indices) ,
4. **Y\_test** (test part of the dependent variables associated with the X test sets, and therefore also the same indices).

**from sklearn.model\_selection import train\_test\_split**

**# #Splitting the X into X\_train and X\_test**

**print("\*"\*120)**

**print(color.BOLD + "Splitting the X into X\_train and X\_test" + color.END)**

**print()**

**y = df[target]**

**X = df[features]**

**print("\*"\*120)**

**print(color.UNDERLINE + "The Training data set is split into train set and test set (%d:%d)" %(100-test\_size, test\_size) + color.END)**

**X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size = test\_size, random\_state = random\_state)**

**Feature Engineering:**

**After Data preparation, sometimes data still contains irrelevant data, unmeaningful data for machine Learning algorithm to ensure data becomes informative, discriminative, and non-redundant. The results from this is that**  It can improve model performance on hidden data.

**Feature engineering includes below common parts:**

**Feature construction is to construct raw data to informative features in which best represent for the problem and the algorithm can be understood.**

**Feature Transformation:**

It is the process of transforming a feature into new one with a specific function.

**Example:**

Scaling (the most important), log (reduce heteroscedasticity)

from sklearn.preprocessing import MinMaxScaler

**scaler = MinMaxScaler(feature\_range=(0, 1))**

**X = scaler.fit\_transform(X)**

**Dimension reduction (selection and extraction): is the process to reduce**

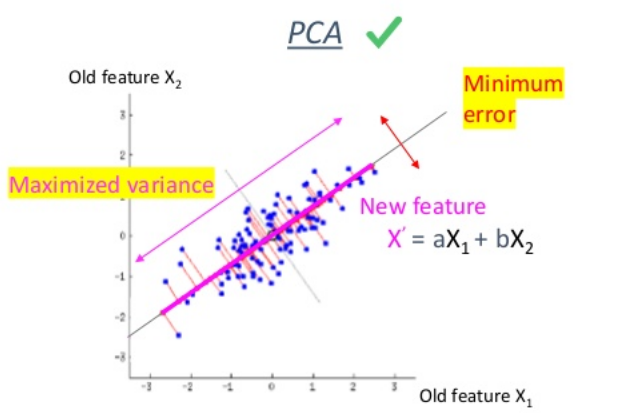
the number of irrelevant features used to build the models with the goal of keeping only informative, discriminative non-redundant features. It results to the advantage of faster computation, less storage space, improve model performance, data visualization.

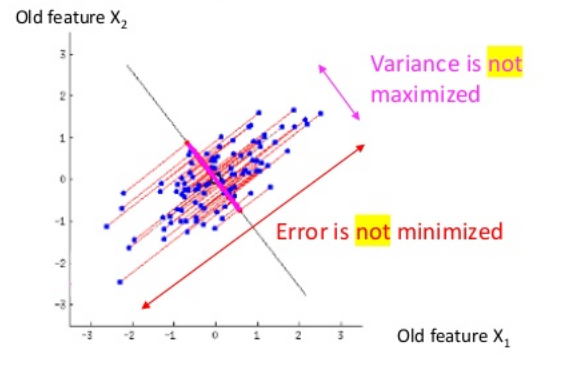
**It includes feature**

* **Feature selection: is the process to select** the most "importance"/influence on the target variable list of existing features. There are several techniques can be used to solve this problem such as Linear Regression, Decision Trees, calculation of "importance" weights (ReliefF, Fisher score), Recursive Feature Elimination(RFE), Variance threshold filter filter, High Correlation Filter.
* **Feature extraction:** generate features from data in a format that is difficult to analyse and build derived features in which are more relevant. The most common feature extraction is Principle Component Analysis (PCA).

**PCA makes orthogonal projection on a linear space to determine new features named as Principal components/features.**

Principlal component is built along an axis so that it is, as much as possible, discriminative (its variance is maximized), informative( the error to the original values is minimized), independence (non-redundant) from other features.





from sklearn.decomposition import PCA

pca = PCA(n\_components=no\_selected\_features)

X = pca.fit\_transform(X)

* **Feature transformation:** Transformation of existing features in order to create new ones based on the old ones. A very popularly used technique for dimensionality reduction is Principal Component Analysis (pca) that uses some orthogonal transformation in order to produce a set of linearly non-correlated variables based on the initial set of variables.

**Data Modeling:**

In this stage, it needs to find the most suitable learning algorithm to train model using dataset.

After understanding the dataset in processed Cleveland Heart Disease from a study of heart disease from UCI Machine Learning, the dataset has both independent (features) and dependent data (label), the Machine Learning Algorithm should be Supervised learning. Even the most problem of machine learning is to predict the presence or absence of heart disease in patient, there are many model using classification algorithm to solve the problem. Moreover, the independent features are multivariable and dependent data (label) is continuous, it may be unbalanced to use classification algorithm to build model. Therefore, based on the characteristic of dataset and problem, the selected algorithm is considered as Linear Regression.



from sklearn.linear\_model import LinearRegression

mln\_model=linear\_model.LinearRegression()

mln\_model.fit(X\_train,y\_train)

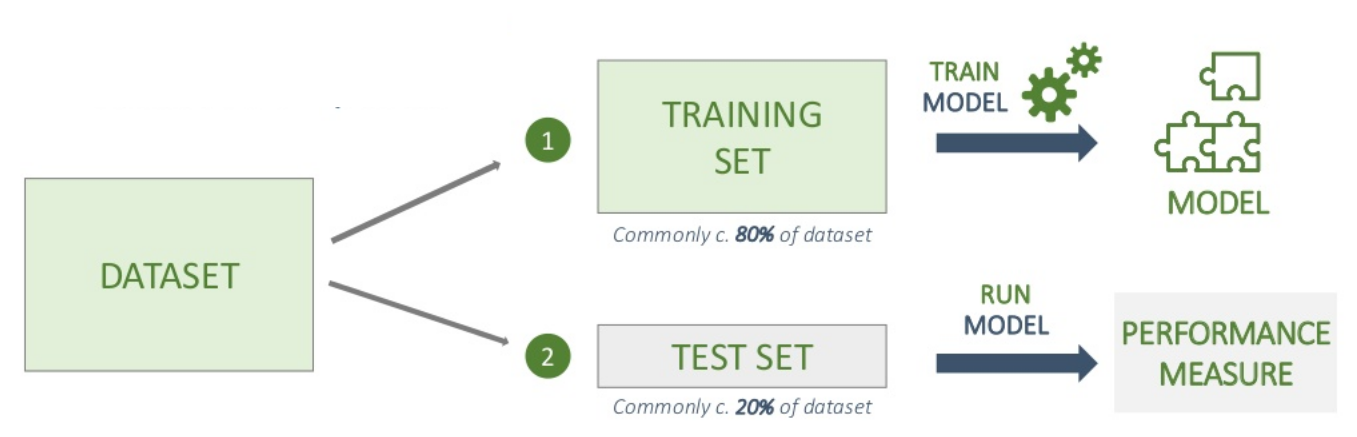
**Model Performance:**

After developing Machine Learning Model to predict heart disease problem. It is now time to validate how model performance works.

There are 2 steps of Model performance process:

Use the training model to predict the label (heart disease) from test and sometimes training input include methods such as:

**Training set and test set**: This method is simple by split dataset into training and test set with ratio 80:20. It can be simple and quick to test different algorithms at the beginning. However, the performance may be biased due to it highly depends on data from test set.



#Predicting the values of traing set

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size = test\_size, random\_state = random\_state)

y\_pred\_train = mln\_model.predict(X\_train)

y\_pred\_train = np.round(y\_pred\_train)

#Predicting the values of test set

y\_pred\_test = mln\_model.predict(X\_test)

y\_pred\_test = np.round(y\_pred\_test)

Use some indicator or performance metric to evaluate and compare the variance between the predicted values and the real label values.

plt.scatter(y\_test,y\_pred\_test)

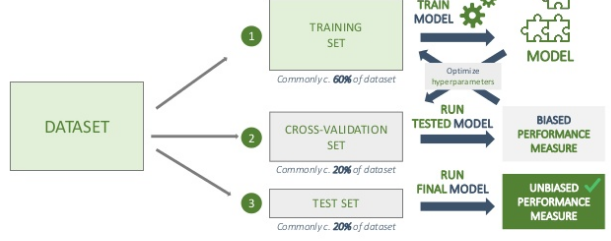
#Check the difference between the actual value and predicted value.

df1 = pd.DataFrame({'Actual': y\_test, 'Predicted': y\_pred\_test})

**To improve performance, cross-validation is an alternative option. There is extensive cross-validation functions in scikit-learn including methods for searching various sets of factors and different models.**

**Moreover,** cross-validation can be applicable to any model.

Cross-validation is a resampling procedure used to evaluate machine learning models on a limited data sample.



## k-Fold Cross-Validation procedures.

1. Randomly Shuffle the data set.
2. Split the data set into k groups (commonly k=10)
3. For each group:

* Select one group as a test data set
* Select the remaining groups as a training data set
* Fit a model on the training set and evaluate it on the test set
* Retain the evaluation score and discard the model
* Summarize the skill of the model using the sample of model evaluation scores

kf = KFold(n\_splits=k\_fold, random\_state=101, shuffle=False)

rmse\_values = []

scores = []

test\_group = []

for train\_index, test\_index, in kf.split(df):

train = df.iloc[train\_index]

test = df.iloc[test\_index]

X\_train = train[features]

y\_train = train[target]

X\_test = test[features]

y\_test = test[target]

mln\_model.fit(X\_train, y\_train)

y\_test\_pred = mln\_model.predict(X\_test)

scores.append(mln\_model.score(X\_test, y\_test))

mse = mean\_squared\_error(y\_test, y\_test\_pred)

rmse = np.sqrt(mse)

rmse\_values.append(rmse)

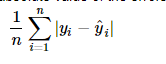
avg\_rmse = np.mean(rmse\_values)

## Model Evaluation Metrics for Regression

Model Evaluation can be a tough task in which it decides if the training model is efficient or not. Select correct metrics to evaluate model is the most important part. To evalulate model for classification problems, it can use classification accuracy as performance evaluation metric. However, for regression problem. There are other metrics to evaluate performances as below.

Normally, dataset is split into training and testing sets and use training set to build the model and testing set to test the training model. After that, evaluating the model performance based on an error metric to determine the accuracy of the model.

Mean Absolute Error: is the mean of the absolute value of the errors



metrics.mean\_absolute\_error(y\_test, y\_pred\_test)

Mean Squared Error (MSE): is the mean of squared errors



mse = mean\_squared\_error(y\_test, y\_test\_pred)

Root Mean squared Error (RMSE) is the squared root of the mean of the squared errors.

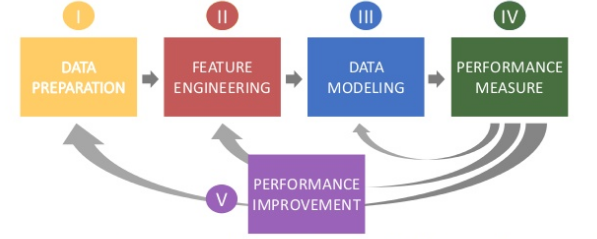


rmse = np.sqrt(mse)

However, it may be not very reliable as the splitting rate of training and test may be unbalance. K-fold Cross Validation(CV) described above is giving a better solutions by dividing the data into k folds and ensuring that each fold is used as a testing set at some point.

**Conclusion:**

**Sometimes the performance metrics show that the performance is not well as expected. There can be many reasons such as data set is underfitting (high bias) or overfitting (high variance) through building Machine Learning model process. It is good to apply iterative Machine Learning modeling to review backward each previous steps to make sure the dataset best fit the best training model to result the best performance.**



References:

<https://www.slideshare.net/CharlesVestur/building-a-performing-machine-learning-model-from-a-to-z>

<https://www.ritchieng.com/machine-learning-evaluate-linear-regression-model/>

<https://towardsdatascience.com/>

<http://machinelearningmastery.com>