RAJKIYA ENGINEERING COLLEGE BIJNOR

(Affiliated to Dr. A.P.J Abdul Kalam Technical University)

SESSION 2020-21

INDUSTRIAL TRAINING REPORT

ON

MACHINE LEARNING

Submitted in the partial fulfillment of the requirement for

Bachelor of Technology

In

Information Technology

Submitted By-
RAHUL
1773513037

Submitted To-
Mr. Santosh Kumar
(Assistant Professor)
(Information Technology)

ACKNOWLEDGEMENT

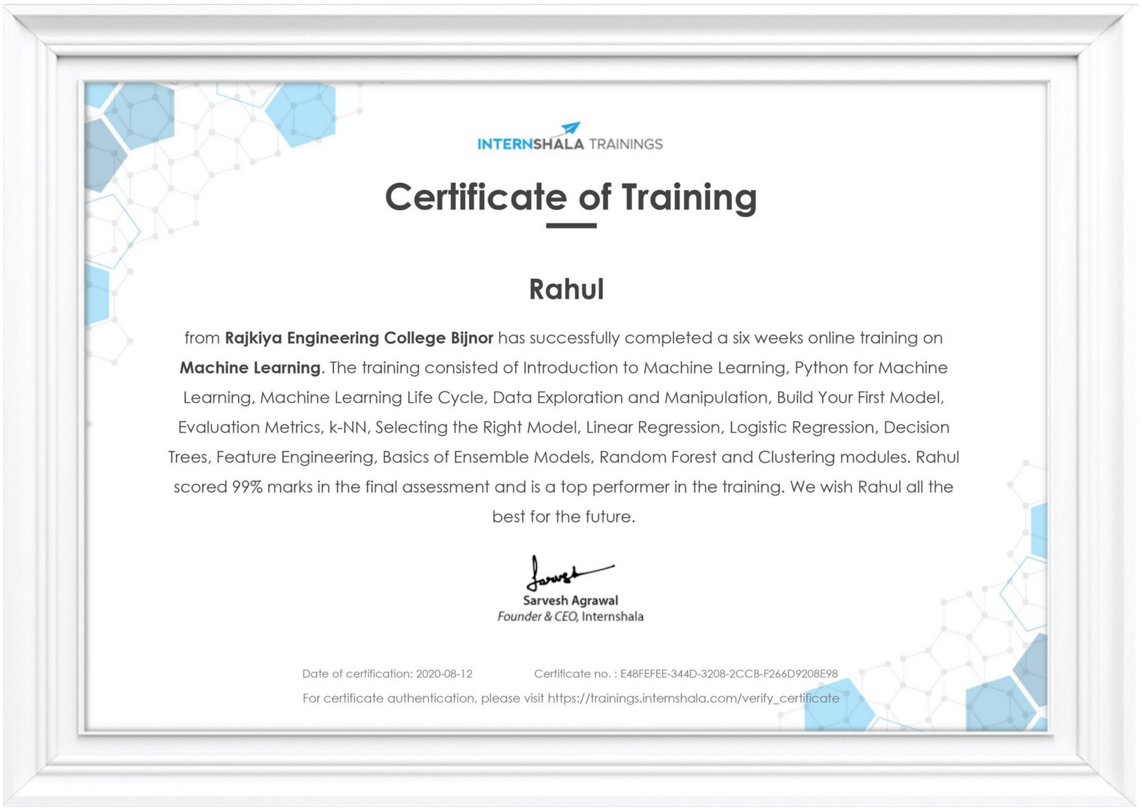
It is always a pleasure to remind the fine people in the Engineering Workshops for their sincere guidance I received to uphold my practical as well as theoretical skills in engineering.

I am very pleased to express our gratitude to **MR. Santosh Kumar** (Asst. Prof.) for his valuable guidance, encouragement and facilities provided during period of our Industrial training.

I would also like to acknowledge with much appreciation the crucial role of the faculty members, Technical staff who gave the permission to use all required resources.

Last but not least, many thanks go to the Family members & friends who have given his full effort in guiding the team in achieving the goal as well as his encouragement to maintain our progress on track. I would to appreciate the guidance given by other supervisor as well as the panels especially in our Industrial training.

INDUSTRIAL TRAINING CERTIFICATE



INTERNSHALA TRAININGS

Certificate of Training

Rahul

from **Rajkiya Engineering College Bijnor** has successfully completed a six weeks online training on **Machine Learning**. The training consisted of Introduction to Machine Learning, Python for Machine Learning, Machine Learning Life Cycle, Data Exploration and Manipulation, Build Your First Model, Evaluation Metrics, k-NN, Selecting the Right Model, Linear Regression, Logistic Regression, Decision Trees, Feature Engineering, Basics of Ensemble Models, Random Forest and Clustering modules. Rahul scored 99% marks in the final assessment and is a top performer in the training. We wish Rahul all the best for the future.

[illegible]

Sarvesh Agrawal
Founder & CEO, Internshala

Date of certification: 2020-08-12 Certificate no. : E48FEFEE-344D-3208-2CCB-F266D9208E98

For certificate authentication, please visit https://trainings.internshala.com/verify_certificate

ABSTRACT

This Report presents the experience and skills gained during my 6 week of industrial training undertaken at **INTERNSHALA** (Work from home).

My training was on Machine Learning with Python, including Machine learning algorithms and mathematics behind algorithms (Statistical, Probability etc).

During the period, I acquired practical knowledge and skills in using Machine Learning.

Machine learning fields, which can be briefly defined as enabling computers make successful prediction using past experience, has exhibited an impressive development.

TABLE OF CONTENT

1. INTRODUCTION TO MACHINE LEARNING

1.1 overview of machine learning

Machine learning is a very hot topic for many key reasons, and because it provides the ability to automatically obtain deep insights, recognize unknown patterns, and create high performing predictive models from data, all without requiring explicit programming instructions.

“A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P if its performance at tasks in T, as measured by P, improves with experience E”. -Tom Mitchell, Carnegie Mellon University

Machine learning involves a computer to be trained using a given data set, and use this training to predict the properties of a given new data.The process of training and prediction involves the use of specialized algorithms. We feed the training data to an algorithm, and the algorithm uses this training data to give predictions on a new test data.

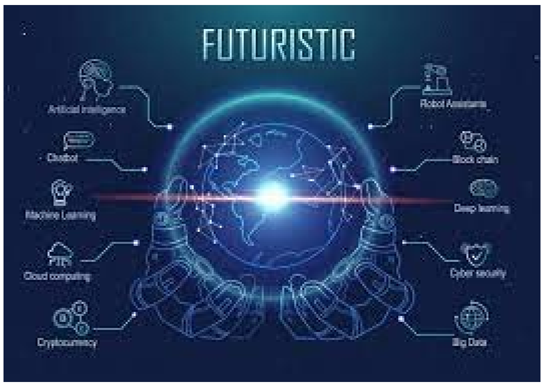
1.1 FUTURE OF MACHINE LEARNING

Along with this, we will also study real-life Machine Learning Future applications to understand companies using machine learning.

Basically, it’s an application of artificial intelligence. Also, it allows software applications to become accurate in predicting outcomes.

Google says**” Machine Learning is the future”**, so the future of machine learning is going to be very bright.

As humans become more addicted to machines, we’re witness to a new revolution that’s taking over the world and that is going to be the future of Machine Learning.



2. PYTHON

2.1 Brief introduction

Python is a widely used general-purpose, high level programming language. It was created by Guido van Rossum in 1991 and further developed by the Python Software Foundation. It was designed with an emphasis on code readability, and its syntax allows programmers to express their concepts in fewer lines of code.

Python is a programming language that lets you work quickly and integrate systems more efficiently.

Python is used in Data science and machine learning because of a lot of libraries available in python.

2.2 Python for ML

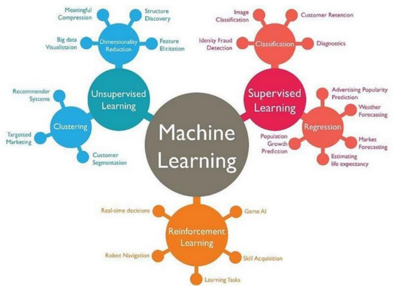
Machine learning is a type of artificial intelligence (AI) that provides computers with the ability to learn without being explicitly programmed. Machine learning focuses on the development of Computer Programs that can change when exposed to new data. In this article, we'll see the basics of Machine Learning, and implementation of a simple machine learning algorithm using python.

3. TECHNOLOGY LEARNT

3.1 TECHNIQUE OF ML

3.1.1 SUPERVISED MACHINE LEARNING

Supervised machine learning is a subcategory of machine learning and intelligence. It is defined by its use of labeled datasets to train algorithms to classify data or predict outcomes accurately. As input data is fed into the model, it adjusts its weights through a reinforcement learning process, which ensures that the model has been fitted appropriately. Supervised learning helps organizations solve a variety of real-world problems at scale, such as classifying spam in a separate folder from your inbox.



Unsupervised machine learning: Unsupervised learning is the training of

machines using information that is neither classified nor labeled and allowing the algorithm to act on that information without guidance. Here the task of the machine is to group unsorted information according to similarities, patterns and differences without any prior training of data.

3.1.2 REINFORCEMENT ML:

Reinforcement learning is the training of machine learning models to make a sequence of decisions. The agent learns to achieve a goal in an uncertain, potentially complex environment. In reinforcement learning, artificial intelligence faces a game-like situation. The computer employs trial and error to come up with a solution to the problem. To get the machine to do what the programmer wants, the artificial intelligence gets either rewards or penalties for the actions it performs. Its goal is to maximize the total reward

3.2 DATA PROCESSING

Data Processing is a task of converting data from a given form to a much more usable and desired form i.e. making it more meaningful and informative. Using Machine Learning algorithms, mathematical modelling and statistical knowledge, this entire process can be automated.

3.2.1 DATA PREPARATION

Machine learning helps us find patterns in data—patterns we then use to make predictions about new data points. To get those predictions right, we must *construct* the data set and *transform* the data correctly.

There are some steps of data processing are:

1. Data collection
2. Data exploration and profiling
3. Formatting data and make it consistent
4. Improving data quality
5. Feature engineering and feature scaling
6. Splitting data in to training and evaluation sets

3.2.2 DATA EXPLORATION

Data exploration is an approach similar to initial data analysis, whereby a data analyst uses visual exploration to understand what is in a dataset and the characteristics of the data, rather than through traditional data management systems.These characteristics can include size or amount of data, completeness of the data, correctness of the data, possible relationships amongst data elements or files/tables in the data.

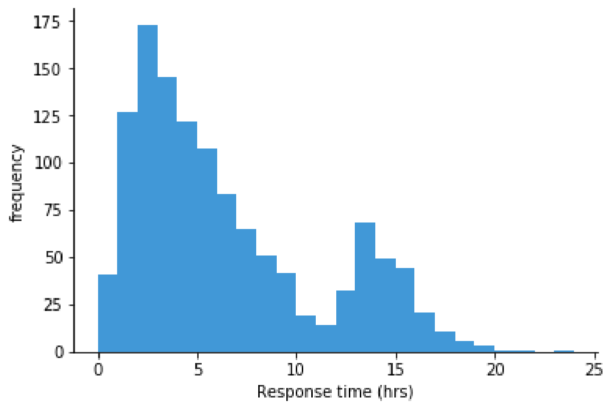
3.2.3 DATA VISUALISATION

With the help of data visualization, we can see how the data looks like and what kind of correlation is held by the attributes of data. It is the fastest way to see if the features correspond to the output. With the help of following Python recipes, we can understand ML data with statistics.

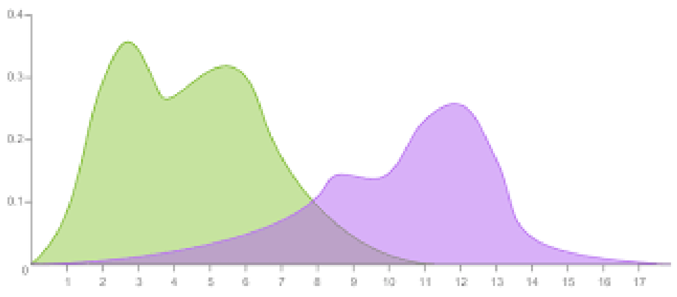
The simplest type of visualization is single-variable or “univariate” visualization. With the help of univariate visualization, we can understand each attribute of our dataset independently.

The following are some techniques in Python to implement univariate visualization −

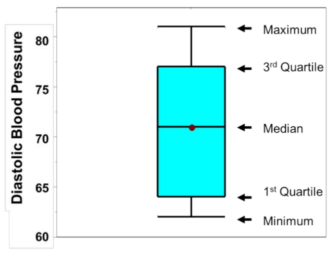
- HISTOGRAM



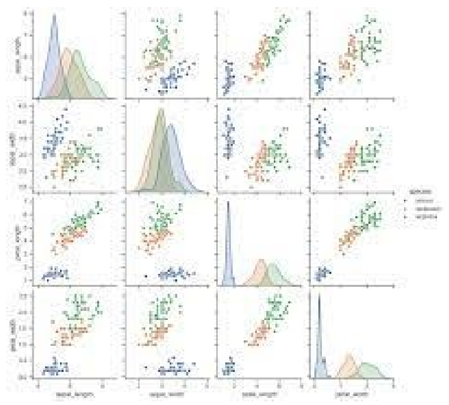
- DENSITY PLOTS



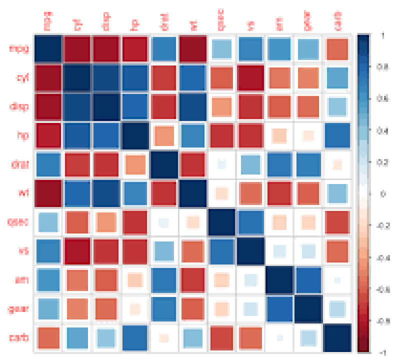
- BOX AND WHISKER PLOT



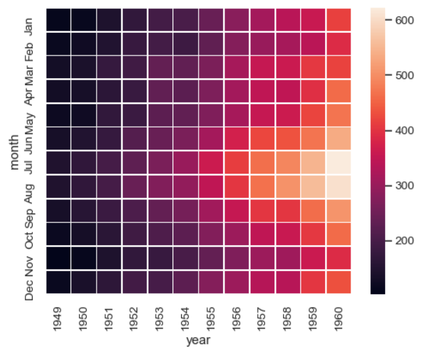
- SCATTER MATRIX PLOT



- CORRELATION MATRIX PLOT



- HEAT MAP PLOT



3.2.4 FEATURE ENGINEERING

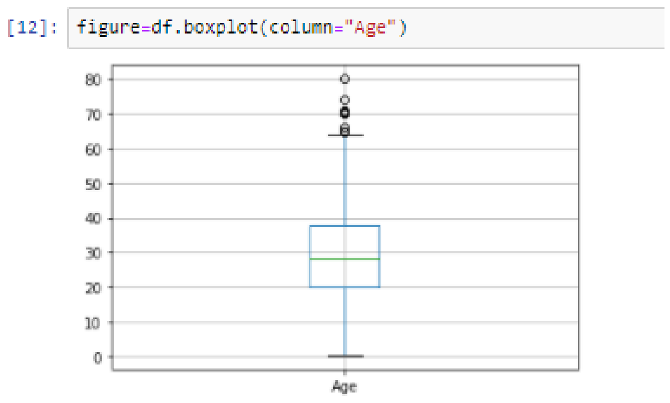
We know that machine learning algorithms use some input data to produce results. But quite often, the data you've been given might not be enough for designing a good machine learning model. That's where the power of feature engineering comes into play.

LIst of feature engineering techniques :

1. **Imputation:** Missing values are one of the most common problems you can encounter when you prepare your data for machine learning. The reason for the missing values might be human errors, interruptions in the data

flow, privacy concerns, etc.

2. Handling Outliers: Before mentioning how outliers can be handled, I want to state that the best way to detect outliers is to demonstrate the data visually. All other statistical methodologies are open to making mistakes, whereas visualizing the outliers gives a chance to take a decision with high precision.



```
[12]: figure=df.boxplot(column="Age")
```

3. Binning: The main motivation of binning is to make the model more robust and prevent overfitting. However, it has a cost on the performance. Every time you bin something, you sacrifice information and make your data more regularized.

4. Log Transform: Logarithm transformation (or log transform) is one of the most commonly used mathematical transformations in feature engineering. Here are the benefits of using log transform:

5. One-Hot Encoding: This method changes your categorical data, which is challenging to understand for algorithms, to a numerical format and enables you to group your categorical data without losing any information.

6. Grouping Operations: Using a pivot table or grouping based on aggregate functions using lambda.

7. Scaling: In most cases, the numerical features of the dataset do not have a certain range and they differ from each other. In order for a symmetric dataset, scaling is required.

$$x_{i,j}^{*} = \frac{x_{i,j} - x_j^{min}}{x_j^{max} - x_j^{min}}$$

3.3 PYTHON LIBRARIES USED IN ML

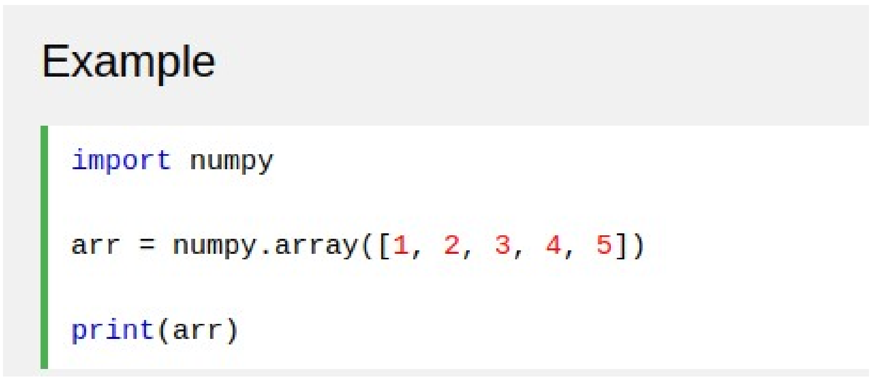
THERE ARE MANY LIBRARY USED IN MACHINE LEARNING :

NUMPY: NumPy is a Python library used for working with arrays.

It also has functions for working in the domain of linear algebra, fourier transform, and matrices. In Python we have lists that serve the purpose of arrays, but they are slow to process.

NumPy aims to provide an array object that is up to 50x faster than traditional Python lists.

The array object in NumPy is called ndarray, it provides a lot of supporting functions that make working with ndarray very easy.


Example

```
import numpy

arr = numpy.array([1, 2, 3, 4, 5])

print(arr)
```

PANDAS : Pandas is a Python library used for working with data sets.

It has functions for analyzing, cleaning, exploring, and manipulating data.

Pandas is a Python library used for working with data sets.It has functions for analyzing, cleaning, exploring, and manipulating data.

Pandas gives you answers about the data. Like:

- Is there a correlation between two or more columns?
- What is average value?
- Max value?
- Min value?

Example

```python
import pandas

mydataset = {
  'cars': ["BMW", "Volvo", "Ford"],
  'passings': [3, 7, 2]
}

myvar = pandas.DataFrame(mydataset)

print(myvar)
```

MATPLOTLIB: Matplotlib is a low level graph plotting library in python that serves as a visualization utility.

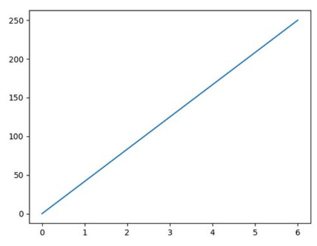
Matplotlib was created by John D. Hunter. Matplotlib is open source and we can use it freely.

Matplotlib is mostly written in python, a few segments are written in C, Objective-C and Javascript for Platform compatibility.

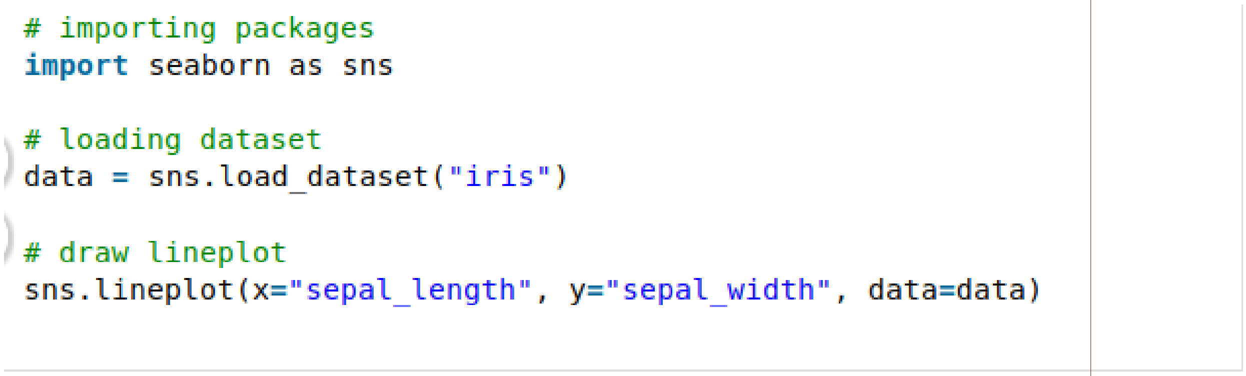
Most of the Matplotlib utilities lies under the pyplot submodule, and are usually imported under the plt alias:

```python
import matplotlib.pyplot as plt
```

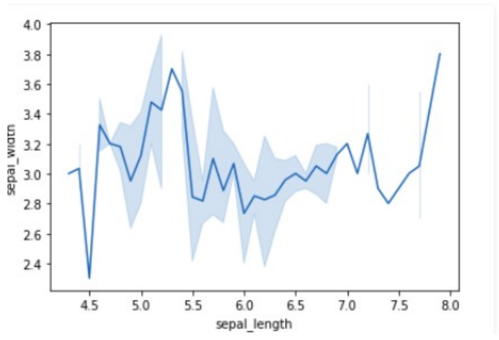
```python
import matplotlib.pyplot as plt
import numpy as np

xpoints = np.array([0, 6])
ypoints = np.array([0, 250])

plt.plot(xpoints, ypoints)
plt.show()
```



SEABORN : One of the best but also more challenging ways to get your insights across is to visualize them: that way, you can more easily identify patterns, grasp difficult concepts or draw the attention to key elements. When you’re using Python for data science, you’ll most probably have already used Matplotlib, a 2D plotting library that allows you to create publication-quality figures. Another complimentary package that is based on this data visualization library is Seaborn, which provides a high-level interface to draw statistical graphics.



```
# importing packages
import seaborn as sns

# loading dataset
data = sns.load_dataset("iris")

# draw lineplot
sns.lineplot(x="sepal_length", y="sepal_width", data=data)
```



4. SUPERVISED MACHINE LEARNING

Supervised learning as the name indicates the presence of a supervisor as a teacher. Basically supervised learning is a learning in which we teach or train the machine using data which is well labeled that means some data is already tagged with the correct answer. After that, the machine is provided with a new set of examples(data) so that supervised learning algorithm analyses the training data(set of training examples) and produces a correct outcome from labeled data.

4.1 REGRESSION

Regression analysis consists of a set of machine learning methods that allow us to predict a continuous outcome variable (y) based on the value of one or multiple predictor variables (x). Briefly, the goal of a regression model is to build a mathematical equation that defines y as a function of the x variables.

4.1.1 LINEAR REGRESSION

Simple linear regression is a type of regression analysis where the number of independent variables is one and there is a linear relationship between the independent(x) and dependent(y) variable. The red line in the above graph is referred to as the best fit straight line. Based on the given data points, we try to plot a line that models the points the best.

The line can be modelled based on the linear equation shown below.

```
y = a_0 + a_1 * x      ## Linear Equation
```

Cost Function

The cost function helps us to figure out the best possible values for a_0 and a_1 which would provide the best fit line for the data points. Since we want the best values for a_0 and a_1, we convert this search problem into a minimization problem where we would like to minimize the error between the predicted value and the actual value.

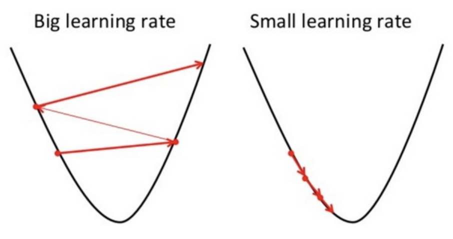
$$minimize \frac{1}{n}\sum_{i=1}^{n}(pred_i - y_i)^2$$

$$J = \frac{1}{n}\sum_{i=1}^{n}(pred_i - y_i)^2$$

Minimization and Cost Function

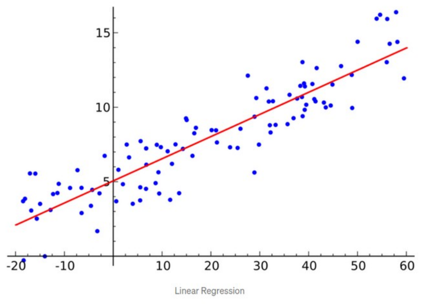
GRADIENT DESCENT:

The next important concept needed to understand linear regression is gradient descent. Gradient descent is a method of updating a_0 and a_1 to reduce the cost function(MSE). The idea is that we start with some values for a_0 and a_1 and then we change these values iteratively to reduce the cost. Gradient descent helps us on how to change the values.



Gradient Descent

Regression is a method of modelling a target value based on independent predictors. This method is mostly used for forecasting and finding out cause and effect relationship between variables. Regression techniques mostly differ based on the number of independent variables and the type of relationship between the independent and dependent variables.



The motive of the linear regression algorithm is to find the best values for a_0 and a_1. Before moving on to the algorithm, let’s have a look at two important concepts you must know to better understand linear regression.

4.1.2 POLYNOMIAL REGRESSION

Polynomial Regression is a regression algorithm that models the relationship between a dependent(y) and independent variable(x) as nth degree polynomial. The Polynomial Regression equation is given below:

$$y= b_0+b_1x_1+ b_2x_1^2+ b_2x_1^3+...... b_nx_1^n$$

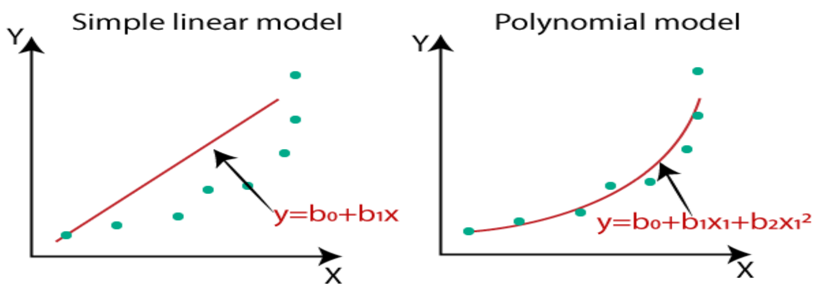
- It is also called the special case of Multiple Linear Regression in ML. Because we add some polynomial terms to the Multiple Linear regression equation to convert it into Polynomial Regression.
- It is a linear model with some modification in order to increase the accuracy.
- The dataset used in Polynomial regression for training is of non-linear nature.
- It makes use of a linear regression model to fit the complicated and non-linear functions and datasets.
- Hence, *"In Polynomial regression, the original features are converted into Polynomial features of required degree (2,3,..,n) and then modeled using a linear model."*

Need for Polynomial Regression:

The need of Polynomial Regression in ML can be understood in the below points:

- If we apply a linear model on a **linear dataset**, then it provides us a good result as we have seen in Simple Linear Regression, but if we apply the same model without any modification on a **non-linear dataset**, then it will produce a drastic output. Due to which loss function will increase, the error rate will be high, and accuracy will be decreased.

- So for such cases, **where data points are arranged in a non-linear fashion, we need the Polynomial Regression model**. We can understand it in a better way using the below comparison diagram of the linear dataset and non-linear dataset.



4.2 CLASSIFICATION:

Classification is the process of predicting the class of given data points. Classes are sometimes called as targets/ labels or categories. Classification predictive modeling is the task of approximating a mapping function (f) from input variables (X) to discrete output variables (y).

1. Lazy learners

Lazy learners simply store the training data and wait until a testing data appear. When it does, classification is conducted based on the most related data in the stored training data. Compared to eager learners, lazy learners have less training time but more time in predicting.

Ex. k-nearest neighbor, Case-based reasoning

2. Eager learners

Eager learners construct a classification model based on the given training data before receiving data for classification. It must be able to commit to a single hypothesis that covers the entire instance space. Due to the model construction, eager learners take a long time for train and less time to predict.

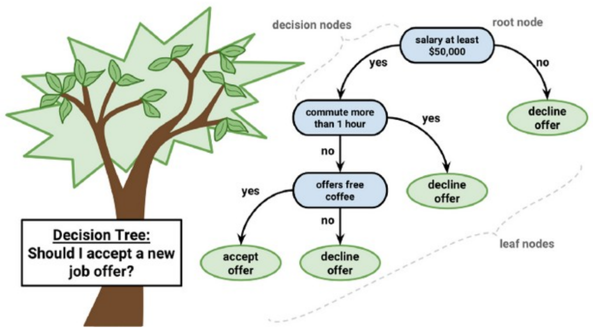
Ex. Decision Tree, Naive Bayes, Artificial Neural Networks

Classification algorithms

There is a lot of classification algorithms available now but it is not possible to conclude which one is superior to another. It depends on the application and nature of available data set. For example, if the classes are linearly separable, the linear classifiers like Logistic regression, Fisher's linear discriminant can outperform sophisticated models and vice versa.

Decision Tree:

Decision tree builds classification or regression models in the form of a tree structure. It utilizes an if-then rule set which is mutually exclusive and exhaustive for classification. The rules are learned sequentially using the training data one at a time. Each time a rule is learned, the tuples covered by the rules are removed. This process is continued on the training set until meeting a termination condition.



Naive Bayes

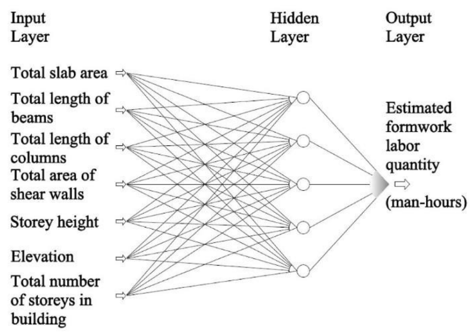
Naive Bayes is a probabilistic classifier inspired by the Bayes theorem under a simple assumption which is the attributes are conditionally independent.

$$P(\mathbf{X}|C_i) = \prod_{k=1}^{n} P(x_k|C_i) = P(x_1|C_i) \times P(x_2|C_i) \times \ldots \times P(x_n|C_i)$$

The classification is conducted by deriving the maximum posterior which is the maximal P(Ci|X) with the above assumption applying to Bayes theorem. This assumption greatly reduces the computational cost by only counting the class distribution. Even though the assumption is not valid in most cases since the attributes are dependent, surprisingly Naive Bayes has able to perform impressively.

Naive Bayes is a very simple algorithm to implement and good results have obtained in most cases. It can be easily scalable to larger datasets since it takes linear time, rather than by expensive iterative approximation as used for many other types of classifiers.

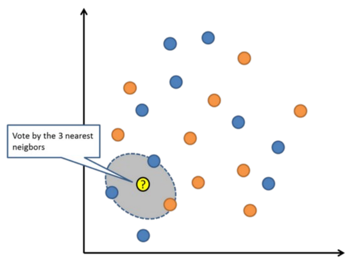
Artificial Neural Networks



Artificial Neural Network is a set of connected input/output units where each connection has a weight associated with it started by psychologists and neurobiologists to develop and test computational analogs of neurons. During the learning phase, the network learns by adjusting the weights so as to be able to predict the correct class label of the input tuples.

There are many network architectures available now like Feed-forward, Convolutional, Recurrent etc. The appropriate architecture depends on the application of the model. For most cases feed-forward models give reasonably accurate results and especially for image processing applications, convolutional networks perform better.

k-Nearest Neighbor (KNN)



k-Nearest Neighbor is a lazy learning algorithm which stores all instances correspond to training data points in n-dimensional space. When an unknown discrete data is received, it analyzes the closest k number of instances saved (nearest neighbors)and returns the most common class as the prediction and for real-valued data it returns the mean of k nearest neighbors.

In the distance-weighted nearest neighbor algorithm, it weights the contribution of each of the k neighbors according to their distance using the following query giving greater weight to the closest neighbors.

$$w \equiv \frac{1}{d(x_q, x_i)^2}$$

Distance calculating query

Evaluating a classifier

After training the model the most important part is to evaluate the classifier to verify its applicability.

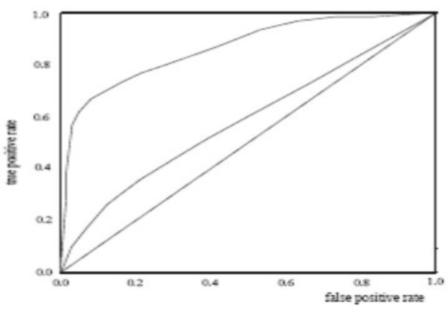
Holdout method

Cross-validation

Over-fitting is a common problem in machine learning which can occur in most models. k-fold cross-validation can be conducted to verify that the model is not over-fitted. In this method, the data-set is randomly partitioned into *k mutually exclusive* subsets, each approximately equal size and one is kept for testing while others are used for training. This process is iterated throughout the whole k folds.

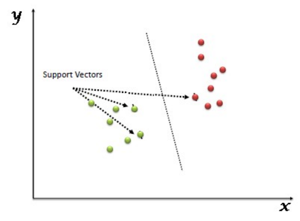
ROC curve (Receiver Operating Characteristics)

ROC curve is used for visual comparison of classification models which shows the trade-off between the true positive rate and the false positive rate. The area under the ROC curve is a measure of the accuracy of the model. When a model is closer to the diagonal, it is less accurate and the model with perfect accuracy will have an area of 1.0

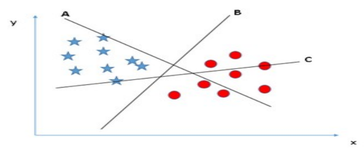


4.2.2 What is Support Vector Machine?

“Support Vector Machine” (SVM) is a supervised Machine learning algo which can be used for both classification or regression challenges. However, it is mostly used in classification problems. In the SVM algorithm, we plot each data item as a point in n-dimensional space (where n is number of features you have) with the value of each feature being the value of a particular coordinate. Then, we perform classification by finding the hyper-plane that differentiates the two classes very well (look at the below snapshot).



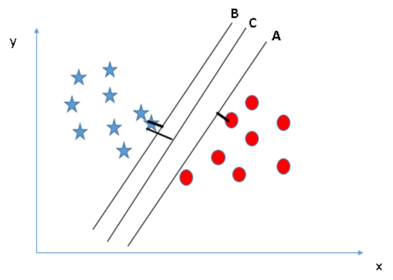
Identify the right hyper-plane (Scenario-1): Here, we have three hyper-planes (A, B and C). Now, identify the right hyper-plane to classify star and circle.



you need to remember a thumb rule to identify the right hyper-plane: “Select the hyper-plane which segregates the two classes better”. In this scenario, hyper-plane “B” has excellently performed this job.

Identify the right hyper-plane (Scenario-2): Here, we have three hyper-planes (A, B and C) and all are segregating the classes well. Now, How can we identify the right hyper-plane?

Here, maximizing the distances between the nearest data point (either class) and hyper-plane will help us to decide the right hyper-plane. This distance is called Margin.

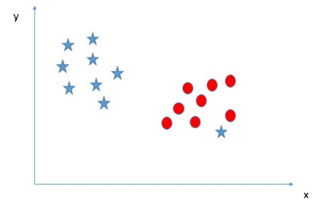


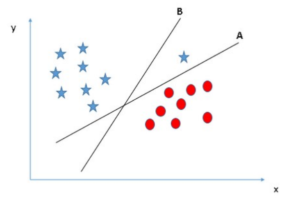
- Above, you can see that the margin for hyper-plane C is high as compared to both A and B. Hence, we name the right hyper-plane as C. Another lightning reason for selecting the hyper-plane with higher margin is robustness. If we select a hyper-plane having low margin then there is a high chance of miss-classification.

● **Identify the right hyper-plane (Scenario-3):**Hint: Use the rules as discussed in previous section to identify the right hyper-plane

Some of you may have selected the hyper-plane **B** as it has higher margin compared to **A.** But, here is the catch, SVM selects the hyper-plane which classifies the classes accurately prior to maximizing margin. Here, hyper-plane B has a classification error and A has classified all correctly. Therefore, the right hyper-plane is **A.**

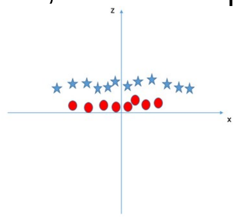
● **Can we classify two classes (Scenario-4)?:** Below, I am unable to segregate the two classes using a straight line, as one of the stars lies in the territory of other(circle) class as an outlier.





● As I have already mentioned, one star at other end is like an outlier for star class. The SVM algorithm has a feature to ignore outliers and find the hyper-plane that has the maximum margin. Hence, we can say, SVM classification is robust to outliers.

● SVM can solve this problem. Easily! It solves this problem by introducing additional features. Here, we will add a new feature z=x^2+y^2. Now, let's plot the data points on axis x and z:



5. UNSUPERVISED MACHINE LEARNING

Unsupervised learning is the training of machines using information that is neither classified nor labeled and allowing the algorithm to act on that information without guidance. Here the task of the machine is to group unsorted information according to similarities, patterns and differences without any prior training of data.

5.1 CLUSTERING

Clustering or cluster analysis is a machine learning technique, which groups the unlabelled dataset. It can be defined as *"A way of grouping the data points into different clusters, consisting of similar data points. The objects with the possible similarities remain in a group that has less or no similarities with another group."*

The clustering technique can be widely used in various tasks. Some most common uses of this technique are:

- Market Segmentation
- Statistical data analysis
- Social network analysis
- Image segmentation
- Anomaly detection, etc.

5.1.1 CLUSTERING ALGORITHMS

The Clustering algorithms can be divided based on their models that are explained above. There are different types of clustering algorithms published, but only a few are commonly used. The clustering algorithm is based on the kind of data that we are using. Such as, some algorithms need to guess the number of clusters in the given dataset, whereas some are required to find the minimum distance between the observation of the dataset.

Here we are discussing mainly popular Clustering algorithms that are widely used in machine learning:

- **K-Means algorithm**
- **Mean-shift algorithm**
- **DBSCAN Algorithm**
- **Expectation-Maximization Clustering using GMM**

Applications of Clustering:

1. In identification of cancer cell
2. In search engine
3. Customer segmentation
4. In Biology
5. In land use

5.1.2 K MEANS CLUSTERING

K-means clustering is one of the simplest and popular unsupervised machine learning algorithms.

You'll define a target number *k*, which refers to the number of centroids you need in the dataset. A centroid is the imaginary or real location representing the center of the cluster. Every data point is allocated to each of the clusters through reducing the in-cluster sum of squares.

In other words, the K-means algorithm identifies the k number of centroids, and then allocates every data point to the nearest cluster, while keeping the centroids as small as possible.

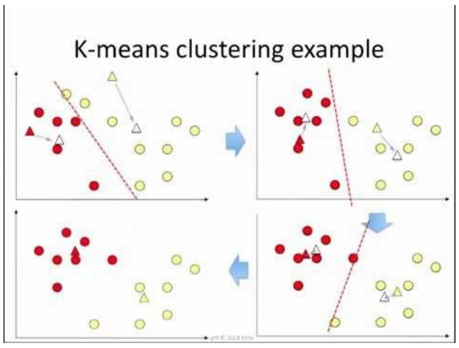
The *'means'* in the K-means refers to averaging of the data; that is, finding the centroid.

HOW K MEANS WORKS:

To process the learning data, the K-means algorithm in data mining starts with a first group of randomly selected centroids, which are used as the beginning points for every cluster, and then performs iterative (repetitive) calculations to optimize the positions of the centroids

It halts creating and optimizing clusters when either:

- The centroids have stabilized — there is no change in their values because the clustering has been successful.
- The defined number of iterations has been achieved.



6. MODEL SELECTION AND EVALUATION

It is a topic central to the process of building good (supervised) machine learning models: model selection. This is not to say that model selection is the centerpiece of the data science workflow — without high-quality data, model building is vanity. Nevertheless, model selection plays a crucial role in building good machine learning models.

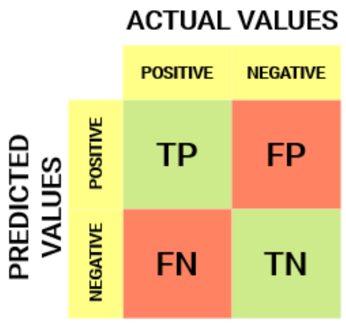
6.1 OVERVIEW

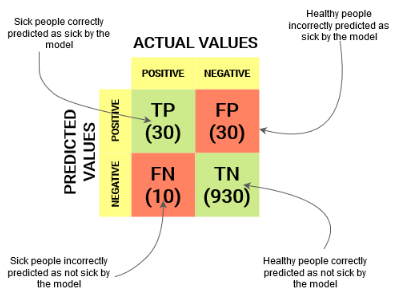
Model evaluation aims at estimating the generalization error of the selected model, i.e., how well the selected model performs on unseen data. Obviously, a good machine learning model is a model that not only performs well on data seen during training (else a machine learning model could simply memorize the training data), but also on unseen data.

6.2 CONFUSION MATRIX

A Confusion matrix is an N x N matrix used for evaluating the performance of a classification model, where N is the number of target classes. The matrix compares the actual target values with those predicted by the machine learning model. This gives us a holistic view of how well our classification model is performing and what kinds of errors it is making.

For a binary classification problem, we would have a 2 x 2 matrix as shown below with 4 values:




Sick people correctly predicted as sick by the model
Healthy people incorrectly predicted as sick by the model
ACTUAL VALUES
POSITIVE
NEGATIVE
PREDICTED VALUES
POSITIVE
NEGATIVE
TP (30)
FP (30)
FN (10)
TN (930)
Sick people incorrectly predicted as not sick by the model
Healthy people correctly predicted as not sick by the model

True Positive (TP)

- The predicted value matches the actual value
- The actual value was positive and the model predicted a positive value

True Negative (TN)

- The predicted value matches the actual value
- The actual value was negative and the model predicted a negative value

False Positive (FP) – Type 1 error

- The predicted value was falsely predicted
- The actual value was negative but the model predicted a positive value
- Also known as the Type 1 error

False Negative (FN) – Type 2 error

- The predicted value was falsely predicted
- The actual value was positive but the model predicted a negative value
- Also known as the Type 2 error

6.3 RMSE VALUE

Root Mean Square Error is the measure of how well a regression line fits the data points. RMSE can also be construed as Standard Deviation in the residuals.

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N} (Predicted_i - Actual_i)^2}{N}}$$

7. PROJECT

WINE QUALITY TEST PROJECT

First I have done some exploration on the data using matplotlib and seaborn. Then, I use different classifier models to predict the quality of the wine.

1. Random Forest Classifier
2. Stochastic Gradient Descent Classifier
3. Support Vector Classifier(SVC)

Then I use cross validation evaluation technique to optimize the model performance.

1. Grid Search CV
2. Cross Validation Score

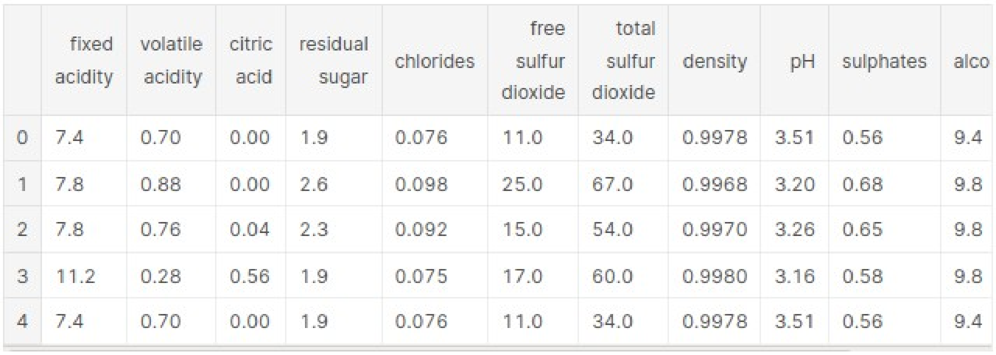
8. IMPLEMENTATION

Importing required packages:

```
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.ensemble import RandomForestClassifier
from sklearn.svm import SVC
from sklearn.linear_model import SGDClassifier
from sklearn.metrics import confusion_matrix, classification_report
from sklearn.preprocessing import StandardScaler, LabelEncoder
from sklearn.model_selection import train_test_split, GridSearchCV,
cross_val_score
%matplotlib inline
```

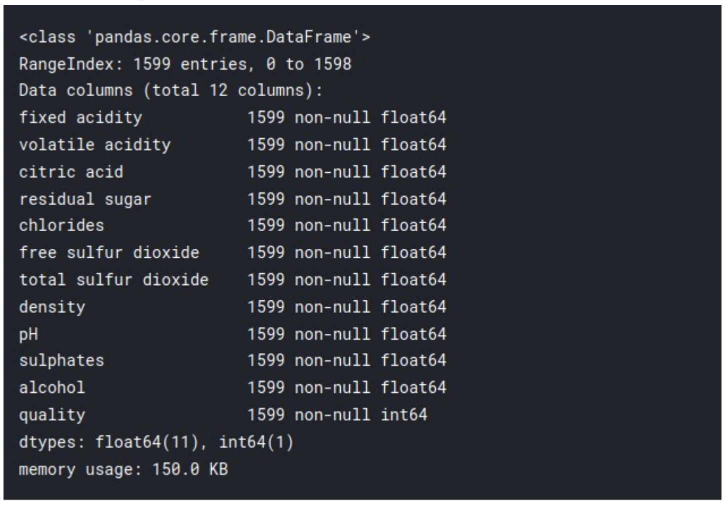
Loading dataset:

```
wine = pd.read_csv('../input/winequality-red.csv')
#How data is distributed
wine.head()
```



	fixed acidity	volatile acidity	citric acid	residual sugar	chlorides	free sulfur dioxide	total sulfur dioxide	density	pH	sulphates	alco
0	7.4	0.70	0.00	1.9	0.076	11.0	34.0	0.9978	3.51	0.56	9.4
1	7.8	0.88	0.00	2.6	0.098	25.0	67.0	0.9968	3.20	0.68	9.8
2	7.8	0.76	0.04	2.3	0.092	15.0	54.0	0.9970	3.26	0.65	9.8
3	11.2	0.28	0.56	1.9	0.075	17.0	60.0	0.9980	3.16	0.58	9.8
4	7.4	0.70	0.00	1.9	0.076	11.0	34.0	0.9978	3.51	0.56	9.4

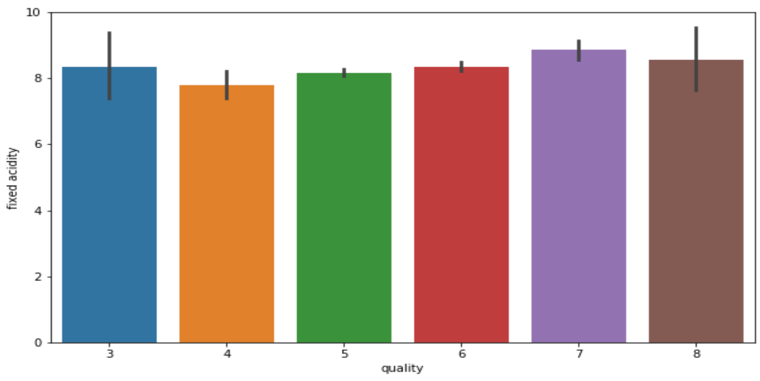
#Information about the data columns
wine.info()



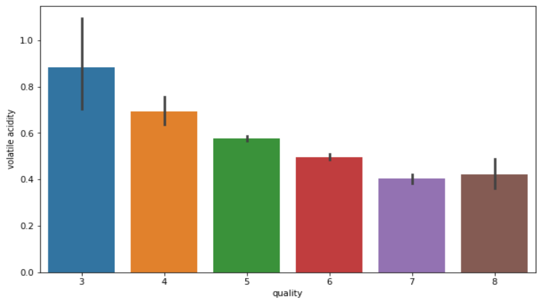
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1599 entries, 0 to 1598
Data columns (total 12 columns):
fixed acidity           1599 non-null float64
volatile acidity        1599 non-null float64
citric acid             1599 non-null float64
residual sugar          1599 non-null float64
chlorides               1599 non-null float64
free sulfur dioxide     1599 non-null float64
total sulfur dioxide    1599 non-null float64
density                 1599 non-null float64
pH                      1599 non-null float64
sulphates               1599 non-null float64
alcohol                 1599 non-null float64
quality                 1599 non-null int64
dtypes: float64(11), int64(1)
memory usage: 150.0 KB
```

#Plotting to know how data column are distributed

```
        fig = plt.figure(figsize = (10,6))
sns.barplot(x = 'quality', y = 'fixed acidity', data = wine)
```

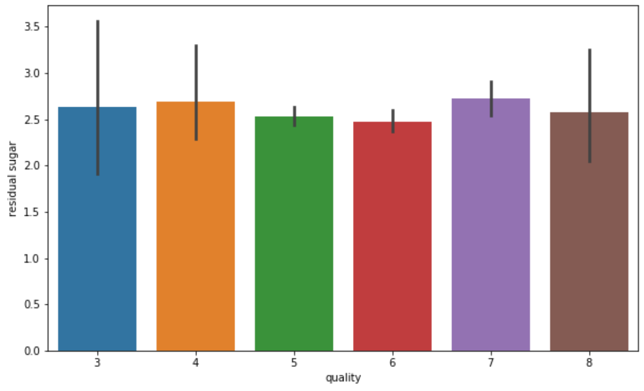


```
fig = plt.figure(figsize = (10,6))
sns.barplot(x = 'quality', y = 'volatile acidity', data = wine)
```

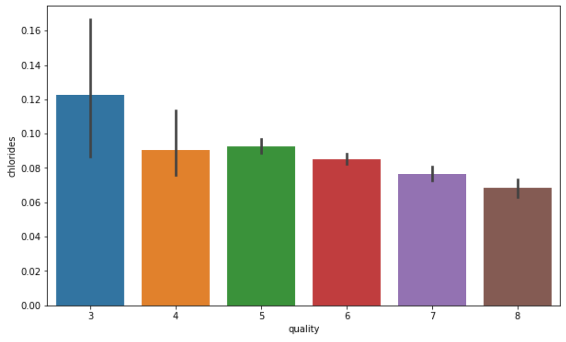


#Composition of citric acid go higher as we go higher in the quality of the wine

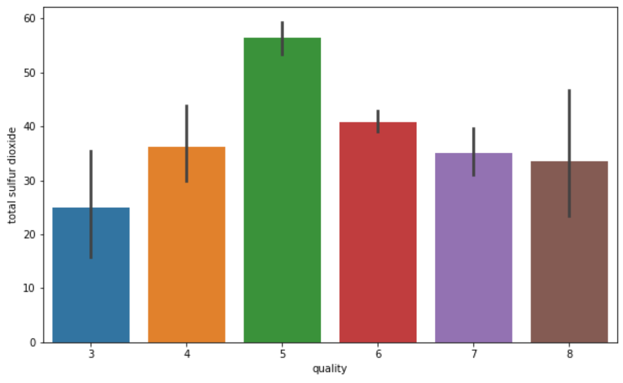
```
fig = plt.figure(figsize = (10,6))
sns.barplot(x = 'quality', y = 'residual sugar', data = wine)
```



```
fig = plt.figure(figsize = (10,6))
sns.barplot(x = 'quality', y = 'chlorides', data = wine)
```

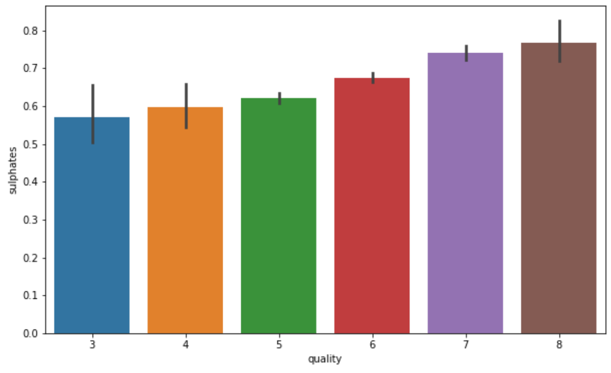


```
fig = plt.figure(figsize = (10,6))
sns.barplot(x = 'quality', y = 'total sulfur dioxide', data = wine)
```



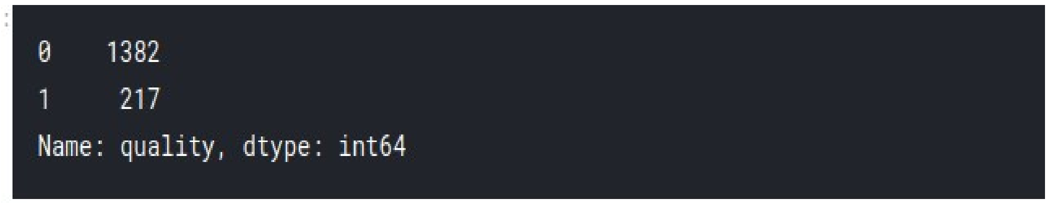
```
fig = plt.figure(figsize = (10,6))
```

sns.barplot(x = 'quality', y = 'sulphates', data = wine)



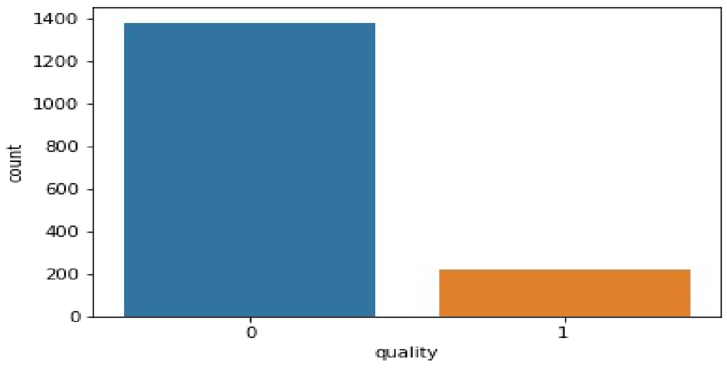
Preprocessing Data for performing Machine learning algorithms

```
#Making binary classificaion for the response variable.
#Dividing wine as good and bad by giving the limit for the quality

bins = (2, 6.5, 8)
group_names = ['bad', 'good']
wine['quality'] = pd.cut(wine['quality'], bins = bins, labels = group_names)

label_quality = LabelEncoder()

wine['quality'] = label_quality.fit_transform(wine['quality'])

wine['quality'].value_counts()
```



```
0    1382
1     217
Name: quality, dtype: int64
```

sns.countplot(wine['quality'])



#Now seperate the dataset as response variable and feature variabes
X = wine.drop('quality', axis = 1)
y = wine['quality']

#Train and Test splitting of data

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2, random_state = 42)

#Applying Standard scaling to get optimized result
sc = StandardScaler()

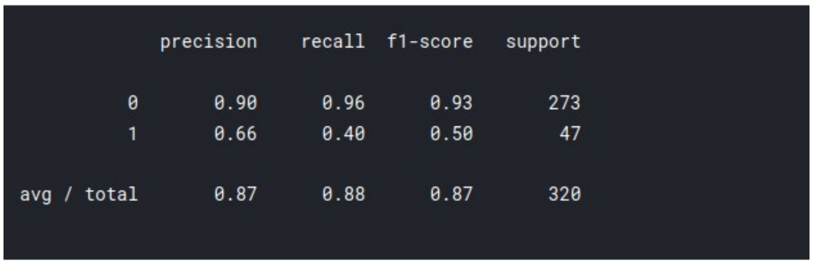
X_train = sc.fit_transform(X_train)
X_test = sc.fit_transform(X_test)

Our training and testing data is ready now to perform machine learning algorithm

Random Forest Classifier

rfc = RandomForestClassifier(n_estimators=200)
rfc.fit(X_train, y_train)
pred_rfc = rfc.predict(X_test)

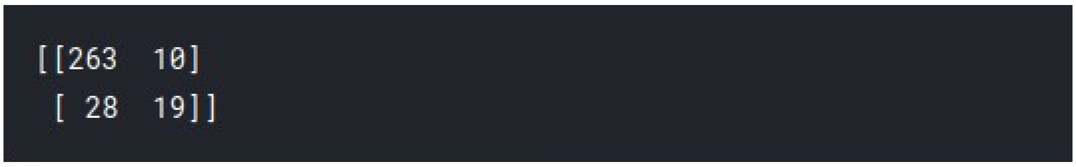
print(classification_report(y_test, pred_rfc))



```
             precision    recall  f1-score   support

          0       0.90      0.96      0.93       273
          1       0.66      0.40      0.50        47

avg / total       0.87      0.88      0.87       320
```

Random forest gives the accuracy of 87%

#Confusion matrix for the random forest classification
print(confusion_matrix(y_test, pred_rfc))

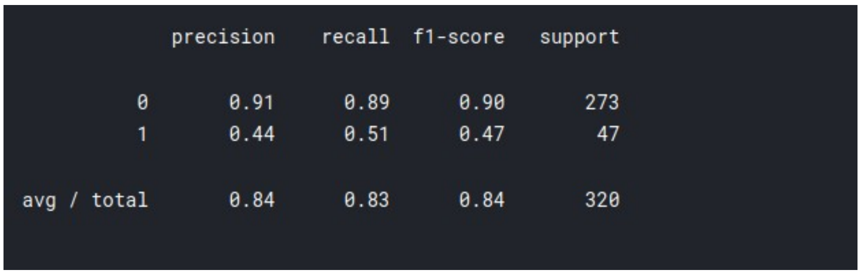


```
[[263  10]
 [ 28  19]]
```

Stochastic Gradient Descent Classifier

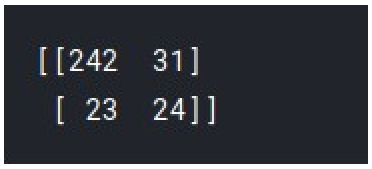
sgd = SGDClassifier(penalty=None)
sgd.fit(X_train, y_train)
pred_sgd = sgd.predict(X_test)

print(classification_report(y_test, pred_sgd))


```
             precision    recall  f1-score   support

          0       0.91      0.89      0.90       273
          1       0.44      0.51      0.47        47

avg / total       0.84      0.83      0.84       320
```

84% accuracy using stochastic gradient descent classifier

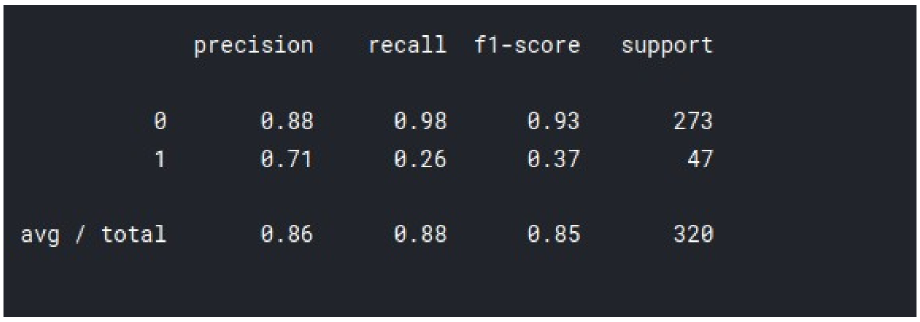
print(confusion_matrix(y_test, pred_sgd))


```
[[242  31]
 [ 23  24]]
```

Support Vector Classifier

svc = SVC()
svc.fit(X_train, y_train)
pred_svc = svc.predict(X_test)

print(classification_report(y_test, pred_svc))

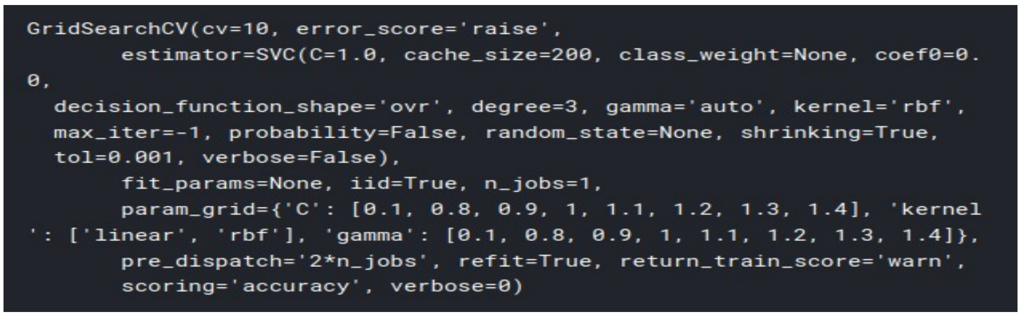

```
             precision    recall  f1-score   support

          0       0.88      0.98      0.93       273
          1       0.71      0.26      0.37        47

avg / total       0.86      0.88      0.85       320
```

Support vector classifier gets 86%

Let's try to increase our accuracy of models

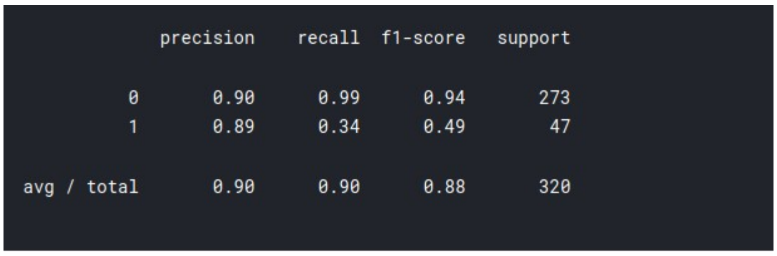
Grid Search CV

```
#Finding best parameters for our SVC model
param = {
    'C': [0.1,0.8,0.9,1,1.1,1.2,1.3,1.4],
    'kernel':['linear', 'rbf'],
    'gamma' :[0.1,0.8,0.9,1,1.1,1.2,1.3,1.4]
}
grid_svc = GridSearchCV(svc, param_grid=param, scoring='accuracy', cv=10)

grid_svc.fit(X_train, y_train)
```



```
GridSearchCV(cv=10, error_score='raise',
       estimator=SVC(C=1.0, cache_size=200, class_weight=None, coef0=0.
0,
  decision_function_shape='ovr', degree=3, gamma='auto', kernel='rbf',
  max_iter=-1, probability=False, random_state=None, shrinking=True,
  tol=0.001, verbose=False),
       fit_params=None, iid=True, n_jobs=1,
       param_grid={'C': [0.1, 0.8, 0.9, 1, 1.1, 1.2, 1.3, 1.4], 'kernel
': ['linear', 'rbf'], 'gamma': [0.1, 0.8, 0.9, 1, 1.1, 1.2, 1.3, 1.4]},
       pre_dispatch='2*n_jobs', refit=True, return_train_score='warn',
       scoring='accuracy', verbose=0)
```

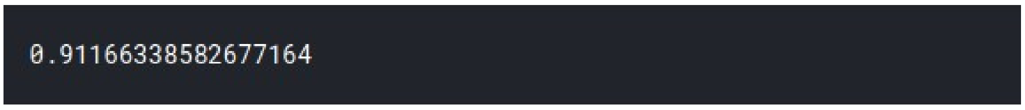
```
#Best parameters for our svc model
grid_svc.best_params_


#Let's run our SVC again with the best parameters.
svc2 = SVC(C = 1.2, gamma =  0.9, kernel= 'rbf')
svc2.fit(X_train, y_train)
pred_svc2 = svc2.predict(X_test)
print(classification_report(y_test, pred_svc2))
```



```
             precision    recall  f1-score   support

          0       0.90      0.99      0.94       273
          1       0.89      0.34      0.49        47

avg / total       0.90      0.90      0.88       320
```

SVC improves from 86% to 90% using Grid Search CV

```
#Now lets try to do some evaluation for random forest model using cross validation.
rfc_eval = cross_val_score(estimator = rfc, X = X_train, y = y_train, cv = 10)
rfc_eval.mean()
```



```
0.91166338582677164
```

Random forest accuracy increases from 87% to 91 % using cross validation score

CONCLUSION

During my industrial training, there are many changes from the point of learning environments and discussion among colleagues. It can directly increase the dedication and rational attitude toward myself.

However, there are still some weaknesses that can be improved in the future. Therefore I conclude that the industrial training program has provided many benefits to students even if there are minor flaws that are somewhat disfiguring , so that this weakness can be rectified in the future.

I can conclude that this industry is through training I received a lot of exposure in the computing world. I would also like to thank the INTERNSHALA for their own experience with having Industrial Training like this.

Signature:

Name: **RAHUL**

Roll: **1773513037**