**AI Project Report**



**Disease Prediction System using**

**Machine Learning with Python**

By : Neha Kalra(17csu122) and Prerna Pathak(17csu143)

# Table of Contents:

1. ABSTRACT
2. INTRODUCTION
3. DESCRIPTION
4. ANALYSIS
5. CONCLUSION
6. BIBLIOGRAPHY

**ABSTRACT:**

Machine learning is an application of artificial intelligence (AI) that provides systems the ability to automatically learn and improve from experience without being explicitly programmed. **Machine learning focuses on the development of computer programs**that can access data and use it learn for themselves.

**INTRODUCTION:**

At present, when one suffers from particular disease, then the person has to visit to doctor which is time consuming and costly too. Also if the user is out of reach of doctor and hospitals it may be difficult for the user as the disease can not be identified. So, if the above process can be completed using a automated program which can save time as well as money, it could be easier to the patient which can make the process easier.

Healthcare industry has become big business. The healthcare industry produces large amounts of health-care data daily that can be used to extract information for predicting disease that can happen to a patient in future while using the treatment history and health data. This hidden information in the healthcare data will be later used for affective decision making for patient’s health. Also, this area need improvement by using the informative data in healthcare. Major challenge is how to extract the information from these data because the amount is very large so some data mining and machine learning techniques can be used. Also, the expected outcome and scope of this project is that if disease can be predicted than early treatment can be given to the patients which can reduce the risk of life and save life of patients and cost to get treatment of diseases can be reduced up to some extent by early recognition.

**SCOPE**

Here the scope of the project is that integration of clinical decision support with computerbased patient records could reduce medical errors, enhance patient safety, decrease unwanted practice variation, and improve patient outcome. The user can select various symptoms and can find the diseases with their probabilistic figures.

**PURPOSE**

The purpose of this system is to provide prediction for the general and more commonly occurring disease that when unchecked can turn into fatal disease. The system applies data mining techniques and decision tree algorithms. This system will predict the most possible disease based on the given symptoms and measures required to avoid the aggression of disease, it will also help the doctors analyse the pattern of presence of diseases in the society. In this project, the disease prediction system will carry out data mining in its preliminary stages, the system will be trained using machine learning and data mining.

**OBJECTIVE**

 The main objective of this research is to develop a disease prediction system. The system can discover and extract hidden knowledge associated with diseases from a historical data set.

 Provides new approach to concealed patterns in the data.

 Helps avoid human biasness.

 Reduce the cost of medical tests.

**Description:**

We did programming in Python on **IDLE** (short for **integrated development environment** or **integrated development and learning environment**[]](https://en.wikipedia.org/wiki/IDLE#cite_note-3)) which is an [integrated development environment](https://en.wikipedia.org/wiki/Integrated_development_environment) for [Python](https://en.wikipedia.org/wiki/Python_(programming_language)), which has been bundled with the default implementation of the language. It is packaged as an optional part of the Python packaging with many [Linux distributions](https://en.wikipedia.org/wiki/Linux_distributions). It is completely written in Python and the [Tkinter](https://en.wikipedia.org/wiki/Tkinter) GUI toolkit ([wrapper](https://en.wikipedia.org/wiki/Wrapper_function) functions for [Tcl](https://en.wikipedia.org/wiki/Tcl)/[Tk](https://en.wikipedia.org/wiki/Tk_(framework))).

IDLE is intended to be a simple and suitable for beginners, especially in an educational environment. To that end, it is cross-platform, and avoids feature clutter.

**Why Python ?**

The language's core philosophy is summarized in the document The Zen of Python (PEP 20), which includes aphorisms such as…

Beautiful is better than ugly

Simple is better than complex

Complex is better than complicated

Readability counts

Explicit is better than implicit

**Python Libraries that would be need to achieve the task:  
1. Numpy  
2. Pandas  
3. Sci-kit Learn**

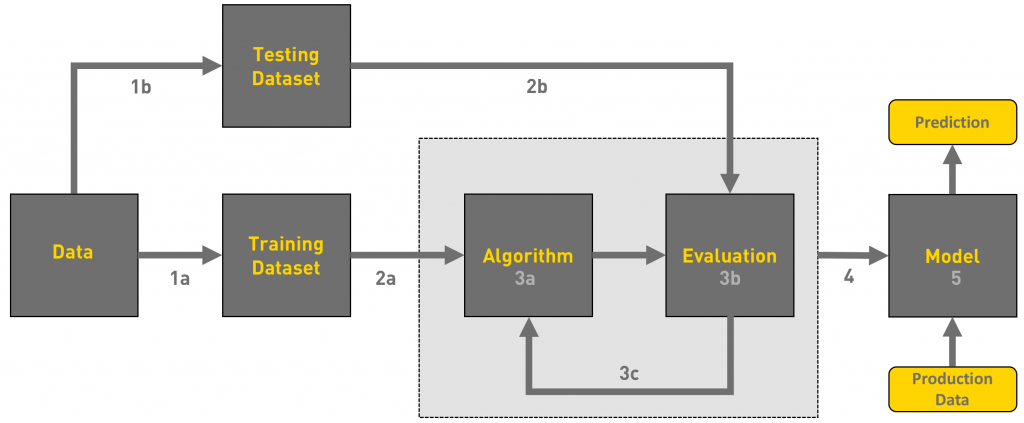
**4. Tkinter**

**First we have to check which version of pip is installed in our system**

1. $ pip --version
2. pip 18.1 from /home/zhaosong/anaconda3/lib/python3.7/site-packages/**pip** (python 3.7)

**Following commands are used to install the above libraries**

1. pip install numpy
2. pip install pandas
3. pip install scipy

****

**We can define the machine learning workflow stages as:**

**1.** Gathering data

**2.** Data pre-processing

**3.** Researching the model that will be best for the type of data

**4.** Training and testing the model

**5.** Evaluation

**1. Gathering Data**

The process of gathering data depends on the type of project we desire to make, if we want to make an ML project that uses real-time data, then we can build an IoT system that using different sensors data. The data set can be collected from various sources such as a file, database, sensor and many other such sources but the collected data cannot be used directly for performing the analysis process as there might be a lot of missing data, extremely large values, unorganized text data or noisy data. Therefore, to solve this problem Data Preparation is done.

We can also use some free data sets which are present on the

internet. [**Kaggle**](http://www.kaggle.com/)and [**UCI Machine learning Repository**](https://archive.ics.uci.edu/ml/datasets.html) are the repositories that are used the most for making Machine learning models. Kaggle is one of the most visited websites that is used for practicing machine learning algorithms, they also host competitions in which people can participate and get to test their knowledge of machine learning.

**2. Data pre-processing**

Data pre-processing is one of the most important steps in machine learning. It is the most important step that helps in building machine learning models more accurately. In machine learning, there is an 80/20 rule. Every data scientist should spend 80% time for data pre-processing and 20% time to actually perform the analysis.

Data pre-processing is a process of cleaning the raw data i.e. the data is collected in the real world and is converted to a clean data set. In other words, whenever the data is gathered from different sources it is collected in a raw format and this data isn’t feasible fortheanalysis.  
Therefore, certain steps are executed to convert the data into a small clean data set, this part of the process is called as data pre-processing.

As we know that data pre-processing is a process of cleaning the raw data into clean data, so that can be used to train the model. So, we definitely need data pre-processing to achieve good results from the applied model in machine learning and deep learning projects.

Most of the real-world data is messy, some of these types of data are:

1. **Missing data:** Missing data can be found when it is not continuously created or due to technical issues in the application (IOT system).

2. **Noisy data:** This type of data is also called outliners, this can occur due to human errors (human manually gathering the data) or some technical problem of the device at the time of collection of data.

3. **Inconsistent data:** This type of data might be collected due to human errors (mistakes with the name or values) or duplication of data.

Three Types of Data

1. Numeric e.g. income, age

2. Categorical e.g. gender, nationality

3. Ordinal e.g. low/medium/high

## How can data pre-processing be performed?

These are some of the basic pre — processing techniques that can be used to convert raw data.

1. **Conversion of data:** As we know that Machine Learning models can only handle numeric features, hence categorical and ordinal data must be somehow converted into numeric features.

2. **Ignoring the missing values:** Whenever we encounter missing data in the data set then we can remove the row or column of data depending on our need. This method is known to be efficient but it shouldn’t be performed if there are a lot of missing values in the dataset.

3. **Filling the missing values:** Whenever we encounter missing data in the data set then we can fill the missing data manually, most commonly the mean, median or highest frequency value is used.

4.**Machine learning:** If we have some missing data then we can predict what data shall be present at the empty position by using the existing data

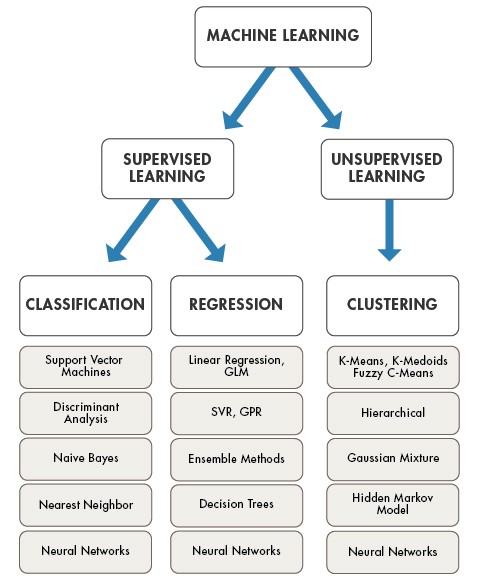
5. **Outliers detection:** There are some error data that might be present in our data set that deviates drastically from other observations in a data set. [Example: human weight = 800 Kg; due to mistyping of extra 0]

**3. Researching the model that will be best for the type of data**

Our main goal is to train the best performing model possible, using the pre-processed data.

We used classification algorithms.

* **Decision Trees**
* **Random Forest**



# 4. Training and testing the model on data

For training a model we initially split the model into 2 sections which are ‘**Training data**’ and ‘**Testing data**’.

You train the classifier using ‘**training data set**’, and then test the performance of your classifier on unseen ‘**test data set**’. An important point to note is that during training the classifier only the training set is available. The test data set must not be used during training the classifier. The test set will only be available during testing the classifier.



**Training set:** The training set is the material through which the computer learns how to process information. Machine learning uses algorithms to perform the training part. A set of data used for learning , that is to fit the parameters of the classifier.

**Test set:** A set of unseen data used only to assess the performance of a fully-specified classifier.

Once the model is trained we can use the same trained model to predict using the testing data i.e. the unseen data. Once this is done we can develop a confusion matrix, this tells us how well our model is trained. A confusion matrix has 4 parameters, which are ‘**True positives’**,**‘True Negatives’**, **‘False Positives’**and ‘**False Negative’**. We prefer that we get more values in the True negatives and true positives to get a more accurate model. The size of the Confusion matrix completely depends upon the number of classes.

* **True positives:** These are cases in which we predicted TRUE and our predicted output is correct.
* **True negatives:** We predicted FALSE and our predicted output is correct.
* **False positives:** We predicted TRUE, but the actual predicted output is FALSE.
* **False negatives:** We predicted FALSE, but the actual predicted output is TRUE.

We can also find out the accuracy of the model using the confusion matrix.

1. *Accuracy (all****correct****/ all) = TP + TN / TP + TN + FP + FN*
2. *Misclassification (all****incorrect****/ all) = FP + FN / TP + TN + FP + FN*
3. *Precision (****true****positives /****predicted****positives) = TP / TP + FP*

I .e. for example:

Accuracy = (100 + 50) / 165 = 0.9090 (90.9% accuracy)

*Confusion matrices have two types of errors:*

*Type I and Type II. False Positive contains one negative word (False) so it’s a Type I error. False Negative has two negative words (False + Negative) so it’s a Type II error.*

**5. Evaluation**

Model Evaluation is an integral part of the model development process. It helps to find the best model that represents our data and how well the chosen model will work in the future.

To improve the model we might tune the hyper-parameters of the model and try to improve the accuracy and also looking at the confusion matrix to try to increase the number of true positives and true negatives.

**Project Procedure**

**Gathering Data:**

Training Data – Training.csv

Testing Data – Testing.csv

**Data Pre Processing:**

As the data present in testing.csv and training.csv have values in the form of 0’s and 1’s

And also no missing value is formed so data pre processing is done

**Choosing the Correct Model for Analysis:**

* 1. As we want to predict the disease of the patient so classification algorithm is the best suited for this type
  2. Classification Algorithm has many algorithms providing various levels of accuracy
  3. The two best models namely Random forest and Decision tree will be used here

**Project Code:**

from tkinter import \*

import numpy as np

import pandas as pd

l1=['back\_pain','constipation','abdominal\_pain','diarrhoea','mild\_fever','yellow\_urine',

'yellowing\_of\_eyes','acute\_liver\_failure','fluid\_overload','swelling\_of\_stomach',

'swelled\_lymph\_nodes','malaise','blurred\_and\_distorted\_vision','phlegm','throat\_irritation',

'redness\_of\_eyes','sinus\_pressure','runny\_nose','congestion','chest\_pain','weakness\_in\_limbs',

'fast\_heart\_rate','pain\_during\_bowel\_movements','pain\_in\_anal\_region','bloody\_stool',

'irritation\_in\_anus','neck\_pain','dizziness','cramps','bruising','obesity','swollen\_legs',

'swollen\_blood\_vessels','puffy\_face\_and\_eyes','enlarged\_thyroid','brittle\_nails',

'swollen\_extremeties','excessive\_hunger','extra\_marital\_contacts','drying\_and\_tingling\_lips',

'slurred\_speech','knee\_pain','hip\_joint\_pain','muscle\_weakness','stiff\_neck','swelling\_joints',

'movement\_stiffness','spinning\_movements','loss\_of\_balance','unsteadiness',

'weakness\_of\_one\_body\_side','loss\_of\_smell','bladder\_discomfort','foul\_smell\_of urine',

'continuous\_feel\_of\_urine','passage\_of\_gases','internal\_itching','toxic\_look\_(typhos)',

'depression','irritability','muscle\_pain','altered\_sensorium','red\_spots\_over\_body','belly\_pain',

'abnormal\_menstruation','dischromic \_patches','watering\_from\_eyes','increased\_appetite','polyuria','family\_history','mucoid\_sputum',

'rusty\_sputum','lack\_of\_concentration','visual\_disturbances','receiving\_blood\_transfusion',

'receiving\_unsterile\_injections','coma','stomach\_bleeding','distention\_of\_abdomen',

'history\_of\_alcohol\_consumption','fluid\_overload','blood\_in\_sputum','prominent\_veins\_on\_calf',

'palpitations','painful\_walking','pus\_filled\_pimples','blackheads','scurring','skin\_peeling',

'silver\_like\_dusting','small\_dents\_in\_nails','inflammatory\_nails','blister','red\_sore\_around\_nose',

'yellow\_crust\_ooze']

disease=['Fungal infection','Allergy','GERD','Chronic cholestasis','Drug Reaction',

'Peptic ulcer diseae','AIDS','Diabetes','Gastroenteritis','Bronchial Asthma','Hypertension',

' Migraine','Cervical spondylosis',

'Paralysis (brain hemorrhage)','Jaundice','Malaria','Chicken pox','Dengue','Typhoid','hepatitis A',

'Hepatitis B','Hepatitis C','Hepatitis D','Hepatitis E','Alcoholic hepatitis','Tuberculosis',

'Common Cold','Pneumonia','Dimorphic hemmorhoids(piles)',

'Heartattack','Varicoseveins','Hypothyroidism','Hyperthyroidism','Hypoglycemia','Osteoarthristis',

'Arthritis','(vertigo) Paroymsal Positional Vertigo','Acne','Urinary tract infection','Psoriasis',

'Impetigo']

l2=[]

for x in range(0,len(l1)):

l2.append(0)

# TRAINING DATA df -------------------------------------------------------------------------------------

df=pd.read\_csv("Training.csv")

df.replace({'prognosis':{'Fungal infection':0,'Allergy':1,'GERD':2,'Chronic cholestasis':3,'Drug Reaction':4,

'Peptic ulcer diseae':5,'AIDS':6,'Diabetes ':7,'Gastroenteritis':8,'Bronchial Asthma':9,'Hypertension ':10,

'Migraine':11,'Cervical spondylosis':12,

'Paralysis (brain hemorrhage)':13,'Jaundice':14,'Malaria':15,'Chicken pox':16,'Dengue':17,'Typhoid':18,'hepatitis A':19,

'Hepatitis B':20,'Hepatitis C':21,'Hepatitis D':22,'Hepatitis E':23,'Alcoholic hepatitis':24,'Tuberculosis':25,

'Common Cold':26,'Pneumonia':27,'Dimorphic hemmorhoids(piles)':28,'Heart attack':29,'Varicose veins':30,'Hypothyroidism':31,

'Hyperthyroidism':32,'Hypoglycemia':33,'Osteoarthristis':34,'Arthritis':35,

'(vertigo) Paroymsal Positional Vertigo':36,'Acne':37,'Urinary tract infection':38,'Psoriasis':39,

'Impetigo':40}},inplace=True)

X= df[l1]

y = df[["prognosis"]]

np.ravel(y)

# TESTING DATA tr --------------------------------------------------------------------------------

tr=pd.read\_csv("Testing.csv")

tr.replace({'prognosis':{'Fungal infection':0,'Allergy':1,'GERD':2,'Chronic cholestasis':3,'Drug Reaction':4,

'Peptic ulcer diseae':5,'AIDS':6,'Diabetes ':7,'Gastroenteritis':8,'Bronchial Asthma':9,'Hypertension ':10,

'Migraine':11,'Cervical spondylosis':12,

'Paralysis (brain hemorrhage)':13,'Jaundice':14,'Malaria':15,'Chicken pox':16,'Dengue':17,'Typhoid':18,'hepatitis A':19,

'Hepatitis B':20,'Hepatitis C':21,'Hepatitis D':22,'Hepatitis E':23,'Alcoholic hepatitis':24,'Tuberculosis':25,

'Common Cold':26,'Pneumonia':27,'Dimorphic hemmorhoids(piles)':28,'Heart attack':29,'Varicose veins':30,'Hypothyroidism':31,

'Hyperthyroidism':32,'Hypoglycemia':33,'Osteoarthristis':34,'Arthritis':35,

'(vertigo) Paroymsal Positional Vertigo':36,'Acne':37,'Urinary tract infection':38,'Psoriasis':39,

'Impetigo':40}},inplace=True)

X\_test= tr[l1]

y\_test = tr[["prognosis"]]

np.ravel(y\_test)

# ------------------------------------------------------------------------------------------------------

def DecisionTree():

from sklearn import tree

clf3 = tree.DecisionTreeClassifier() # empty model of the decision tree

clf3 = clf3.fit(X,y)

# calculating accuracy-------------------------------------------------------------------

from sklearn.metrics import accuracy\_score

y\_pred=clf3.predict(X\_test)

print(accuracy\_score(y\_test, y\_pred))

psymptoms = [Symptom1.get(),Symptom2.get(),Symptom3.get(),Symptom4.get(),Symptom5.get()]

for k in range(0,len(l1)):

# print (k,)

for z in psymptoms:

if(z==l1[k]):

l2[k]=1

inputtest = [l2]

predict = clf3.predict(inputtest)

predicted=predict[0]

h='no'

for a in range(0,len(disease)):

if(predicted == a):

h='yes'

break

if (h=='yes'):

t1.delete("1.0", END)

t1.insert(END, disease[a])

else:

t1.delete("1.0", END)

t1.insert(END, "Not Found")

def randomforest():

from sklearn.ensemble import RandomForestClassifier

clf4 = RandomForestClassifier()

clf4 = clf4.fit(X,np.ravel(y))

# calculating accuracy-------------------------------------------------------------------

from sklearn.metrics import accuracy\_score

y\_pred=clf4.predict(X\_test)

print(accuracy\_score(y\_test, y\_pred))

psymptoms = [Symptom1.get(),Symptom2.get(),Symptom3.get(),Symptom4.get(),Symptom5.get()]

for k in range(0,len(l1)):

for z in psymptoms:

if(z==l1[k]):

l2[k]=1

inputtest = [l2]

predict = clf4.predict(inputtest)

predicted=predict[0]

h='no'

for a in range(0,len(disease)):

if(predicted == a):

h='yes'

break

if (h=='yes'):

t2.delete("1.0", END)

t2.insert(END, disease[a])

else:

t2.delete("1.0", END)

t2.insert(END, "Not Found")

# gui\_stuff------------------------------------------------------------------------------------

root = Tk()

root.title("DISEASE PREDICTOR SYSTEM")

root.geometry("643x475")

root.resizable(False,False)

# entry variables

Symptom1 = StringVar()

Symptom1.set(None)

Symptom2 = StringVar()

Symptom2.set(None)

Symptom3 = StringVar()

Symptom3.set(None)

Symptom4 = StringVar()

Symptom4.set(None)

Symptom5 = StringVar()

Symptom5.set(None)

Name = StringVar()

# Heading

w2 = Label(root, font=("arial",13,"bold") , text="Disease Prediction System",width = 50 , height = 3 , fg="yellow", bg="black")

w2.grid(row=1)

f1=Frame(root,width=1363,height=400,bg="blue",highlightbackground="yellow",highlightthickness=20,bd=20,relief="ridge")

f1.grid(sticky=W)

# labels

NameLb = Label(f1, text="Name of the Patient", fg="yellow", bg="black")

NameLb.grid(row=6, column=0, pady=15, sticky=W)

S1Lb = Label(f1, text="Symptom 1", fg="yellow", bg="black")

S1Lb.grid(row=7, column=0, pady=10, sticky=W)

S2Lb = Label(f1, text="Symptom 2", fg="yellow", bg="black")

S2Lb.grid(row=8, column=0, pady=10, sticky=W)

S3Lb = Label(f1, text="Symptom 3", fg="yellow", bg="black")

S3Lb.grid(row=9, column=0, pady=10, sticky=W)

S4Lb = Label(f1, text="Symptom 4", fg="yellow", bg="black")

S4Lb.grid(row=10, column=0, pady=10, sticky=W)

S5Lb = Label(f1, text="Symptom 5", fg="yellow", bg="black")

S5Lb.grid(row=11, column=0, pady=10, sticky=W)

lrLb = Label(f1, text="DecisionTree", fg="white", bg="red")

lrLb.grid(row=15, column=0, pady=10,sticky=W)

destreeLb = Label(f1, text="RandomForest", fg="white", bg="red")

destreeLb.grid(row=17, column=0, pady=10, sticky=W)

# entries

OPTIONS = sorted(l1)

NameEn = Entry(f1, textvariable=Name)

NameEn.grid(row=6, column=1)

S1En = OptionMenu(f1, Symptom1,\*OPTIONS)

S1En.grid(row=7, column=1)

S2En = OptionMenu(f1, Symptom2,\*OPTIONS)

S2En.grid(row=8, column=1)

S3En = OptionMenu(f1, Symptom3,\*OPTIONS)

S3En.grid(row=9, column=1)

S4En = OptionMenu(f1, Symptom4,\*OPTIONS)

S4En.grid(row=10, column=1)

S5En = OptionMenu(f1, Symptom5,\*OPTIONS)

S5En.grid(row=11, column=1)

dst = Button(f1, text="DecisionTree", command=DecisionTree,bg="green",fg="yellow")

dst.grid(row=8, column=3,padx=10)

rnf = Button(f1, text="Randomforest", command=randomforest,bg="green",fg="yellow")

rnf.grid(row=9, column=3,padx=10)

#textfileds

t1 = Text(f1, height=1, width=40,bg="orange",fg="black")

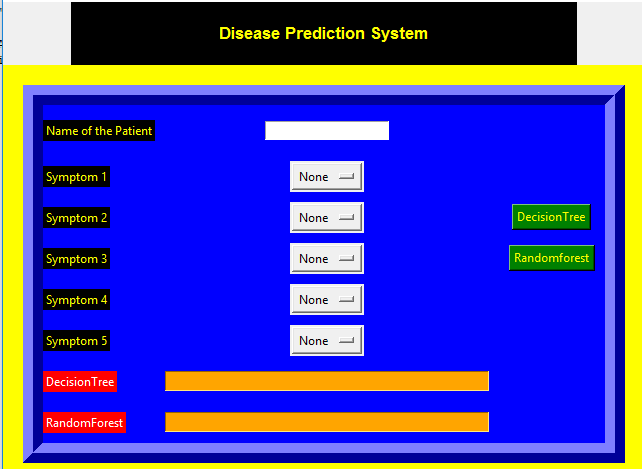
t1.grid(row=15, column=1, padx=10)

t2 = Text(f1, height=1, width=40,bg="orange",fg="black")

t2.grid(row=17, column=1 , padx=10)

root.mainloop()

**GUI:**

****

**ANALYSIS:**

Machine learning uses programmed algorithms that receive and analyse input data to spredict output values within an acceptable range. As new data is fed to these algorithms, they learn and optimise their operations to improve performance, developing ‘intelligence’ over time.

Main goal of this was to provide development opportunities that enhance knowledge, develop skills. Machine learning, one of the top emerging sciences, has an extremely broad range of applications in fields such as Business Analytics where big companies use their extreme amount of raw data to figure out it into a meaningful data using Graphs and plots to make decision about prices and future investments

This was very important so that I can know more about our emerging fields. Any firm needs to work on common raw data so that it is understandable and meaningful to the people who wants to make decision using graphs and predictable methods.

During this period I found myself to be efficient at work. My major strengths were that I was punctual, flexible, knowledgeable and compassionate.

**CONCLUSION:**

This gave as an exposure and I learnt a lot of things regarding our as well as others fields. This was really a very knowledgeable and efficient.I got to know more about Data Analysis and learn something new.

**BIBLIOGRAPHY:**

[https://towardsdatascience.com](https://towardsdatascience.com/)

<https://www.python.org/downloads/>s

### [Disease prediction by machine learning over big data from healthcare communities](https://ieeexplore.ieee.org/abstract/document/7912315/)

[M Chen](https://scholar.google.co.in/citations?user=_HyHTrsAAAAJ&hl=en&oi=sra), Y Hao, [K Hwang](https://scholar.google.co.in/citations?user=R7eCjGYAAAAJ&hl=en&oi=sra), L Wang, L Wang - Ieee Access, 2017 - ieeexplore.ieee.org

With big data growth in biomedical and healthcare communities, accurate analysis of  
medical data benefits early **disease**detection, patient care, and community services.  
…