LONDON METROPOLITAN UNIVERSITY

islington college
(इस्लिङ्टन कलेज)

Module Code & Module Title

CU6051NI - Artificial Intelligence

Assessment Weightage & Type

75% Individual Coursework

Year and Semester

2020-21 Autumn

Student Name: Prashanna GC

London Met ID: 19031368

College ID: NP01CP4A190249

Assignment Due Date: Jan 26, 2022

Assignment Submission Date: Jan 26, 2022

I confirm that I understand my coursework needs to be submitted online via Google Classroom under the relevant module page before the deadline in order for my assignment to be accepted and marked. I am fully aware that late submissions will be treated as non-submission and a marks of zero will be awarded.

Table of Contents

LIST OF FIGURES:

1. Introduction

1.1. Introduction:

Machine Learning is a branch of computer science and artificial intelligence where the study of algorithms and data trains a computer without being explicitly programmed. It provides the machine with an ability to learn from the experience of its own program which improves itself without doing any further coding, so this process is also referred as making computer machines more human-like behavior in decision makings. Different algorithms are required to train the computer machines for reducing human involvement by using the subset of Artificial Intelligence i.e., Machine Learning.

The process involves various quality datasets and correct choice of algorithm during the training phase of computer machine but later the learning phase of the machine is automated depending on the system. The input and output both are provided to the algorithm used by the ML system in the form of training datasets. And the system works out a program for itself. Along with the training datasets, the system also requires validation sets to acquire progress in the machine learning model.

(Great Learning Team, 2021)

1.2. Introduction of the topic used:

In a machine learning model, the model only deals with either structured or semi structured data because ML doesn't need to directly stimulate human behaviour in solving complexity. ML deals with training, validation and testing during its learning phase. From most types of machines learning, Supervised Learning is selected as an AI topic in this given task.

Since with supervised learning, the machine learning model is trained by feeding the machine with large number of training sets. These training sets has multiple collection of labelled data points which trains the system to gather data as input itself from its previous ML deployment and generate an output according to the gathered

input (Algorithmia, 2020). This working process of supervised learning is like the process of how humans learn because like in humans, the machine also collects data from its past project experience and the newer output is predicted more accurately. So, the major role of supervised learning is recognized as providing both input and output data to the machine learning model which are later identified as mapping function by the supervised learning algorithm. The input takes as variable(x) and the output as variable(y) is mapped with a mapping function using supervised learning algorithm. After the completion of training process, furthermore the model is tested according to the test data then the output is predicted.

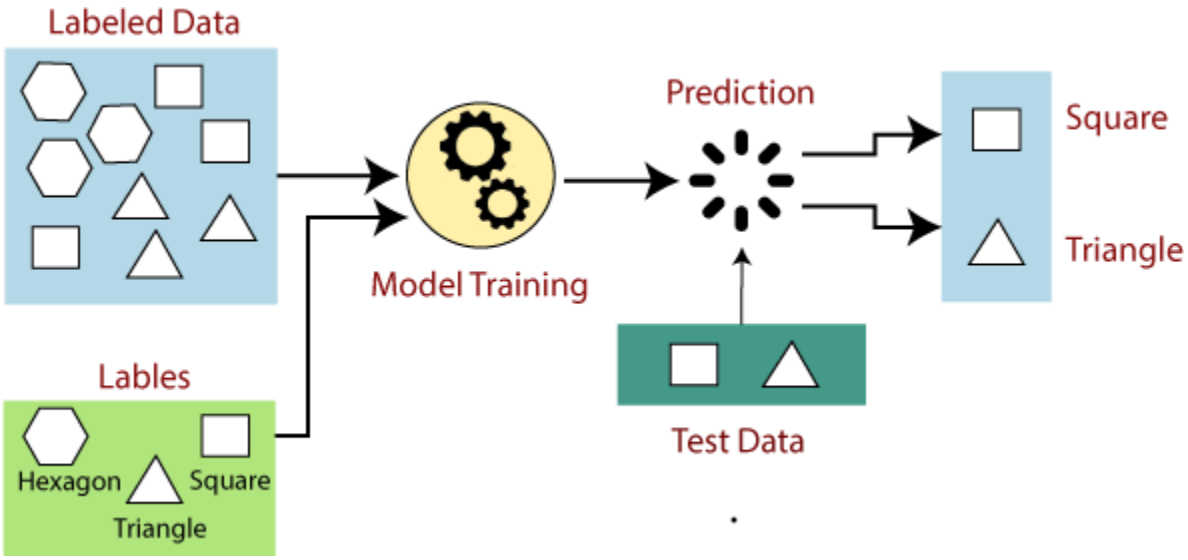

Figure 1: The working process of Supervised Learning

(JavaTpoint, 2011-2021)

1.3. Introduction of the chosen problem topic:

Supervised learning acts as if the output is already known and the algorithms are corrected each time to generate the best effective result. The lifecycle of a ML model depends on the algorithm used in the model. It manages all the sections of the model when it is in production phase. The coursework includes training a machine learning model to predict the disease when the user provides his/her symptoms as the input. This project leads to recognize the illness to avoid any risk factors before the professional medical procedures. The characterization of the illness can be done with various calculations like Random Forest, LightGBM, Decision Tree etc. This framework of ML model uses the algorithms to provide a solution for the user or patient to research about the diseases and its medical procedures to process. The use of machine learning model has been successful for the analysis of medical data including image analysis. It has played an integral part in decision makings for different kind of treatments and supporting professional medical procedures in multiple health sectors. Moreover, not only large medical organizations but also an individual can use the AI-powered tools to research and get to know about the disease based on symptoms.

(Kolli, Aug 2021)

2. Background

2.1. Research work done on the chosen topic:

People can face medical emergencies in any specific period. Some people neglect small health issues which turns out causing a serious health problem. Anyone who should be evaluated with an urgent health care depending on the severity if the symptoms should not be ignoring the medical procedures. Realizing simple symptoms and taking preventions or consulting a doctor in a hospital is very compulsory as well as beneficial. Health problems which are delay cared increases the chances of serious and irresponsible damage to the patient. Any slight symptoms of either coughing, short

breathing, headache, stomachache etc. is ignored especially by people of Nepal which

is not a good way of maintaining good health. Scheduled doctor appointments are also skipped by the patient when they start getting well.

Nowadays, hospitals in different sectors have been using database system to record the data including all the medical reports of each patient, doctor’s appointment schedule, provided medical precautions etc. With the use of information system, hospital management system has become responsible and dependable with the improvements in medical technologies. But this enrolls only the patients who visits the hospital and uses the hospital medical facilities. Implementation of AI technology in medical field has increased with the development of computer science. Furthermore, AI tools are being used to research and analyze CT scans, X-Rays, MRI'S and other image analysis. In the process of image reading, AI tools can even capture and record the information which can be missed even in the expertise medical supervision.

The use of AI in medical sectors is defined more with research and new designed algorithms to help monitor patients with AI-powered tools. Disease detection is to be done in this project and the diagnosis of the disease is observed with the ML model which should be resulting positive reports by tracking vital signs of the patient symptoms. This will also help to recognize complex condition and helps to avoid the risk factors. Algorithms are used to analyze and separate the context of different types of medical information. (For example: if a patient has multiple medical reports which are pervious and latest reports then the trained AI algorithm uses the natural language processing to identify the correct medication researching the patient's medical history.

(IBM, 2021)

As a means of developing a medical diagnosis system based on machine learning algorithms for disease prediction numerous machine learning techniques can be used. But this project has discussion on creating a disease prediction system using KNN algorithm. There should be multiple data regarding the diseases to acquire the most predictable value. The data collection that needs to be processed should be accurate and include all the information like age, gender of an individual, and others to generate the most predictable output that the individual might be suffering from

(Khakharia, 2020). According to my research, KNN algorithm is the most used algorithm

since it provides the most accurate prediction in comparison to the other algorithms. This machine learning model can be a help for the doctors of different medical fields to find details before the early diagnosis of a patient and can also ensure about the diseases.

2.2. Review and Analysis:

Throughout the internet, I have found many research papers relating machine learning and its model which is like that of my project. To learn and understand about the algorithms and how does different algorithm work respectively to the model I have viewed multiple research which includes articles on disease predictions based on symptoms and as well as predicting best nearest hospitals etc. The ML model initiative is found with multiple variety and so high complexity, but its framework remains same. With the use of machine learning techniques, the ML model was developed in the researched project. But, to do so the major steps like gathering data, cleaning the data, building the model etc. was done and later it also included the testing of the model. The raw data collected in the initial phase of building the model consists of different required datasets, which was divided into gender, age group, with the symptoms category to detect the disease.

2.2.1. Research 1: Disease prediction with Decision Tree algorithm

Decision Tree Algorithm belongs to the supervise machine learning category which can fall under both classification and regression analysis. It basically is a tree model with many branches. These branches are separated as decision nodes and leaf nodes which helps to make decision in a standardized manner. The tree diagram has a root node which is extended according to the decision operation conducted and splits. The input is placed at the node and classifies the input. All the required entropy and

proportional are calculated which helps to calculate the maximum GAIN for the datasets.

The entropy is a metric which is used to calculate the uncertainty. With a given number of variables, information gain can be collected which measures how much uncertainty in the target is decreased.

Entropy (E) = $-p_+ log2(p+) – p_- log2(p-)$

Where,

p is proportional

p_- = negative(-ve) proportional

i.e., p_- = (total number of NO) / (total number)

p_+ = positive(+ve) proportional

i.e., p_+ = (total number of YES) / (total number)

And,

Gain (S,A) = Entropy(E) – E_v Values(S_v/S) * Entropy

This algorithm consists of various conditions in the sets of trees to generate the model of data at every node of the decision tree. The working process of this algorithm is as follows:

- The initial phase of implementing this algorithm is by taking the datasets with required feature and targeted attribute.
- The algorithm is provided with the training datasets.
- The ML model will accept the training sets as input, to process among itself to generate the output.
- Decision tree includes an if-else condition in which many decision tree operations

are conducted to obtain the best prediction.

(K. Venkatesh, 2021)

2.2.2. Research 2: AI-Based Smart Clinical Disease prediction with Random Forest Classifier

The Random Forest Classifier belongs to supervised machine learning category, but this is also recognized as the derived form of decision tree algorithm. Mostly random forest algorithm only supports ecommerce and banking industries in predicting the profit rate, loss rate and the financial outcomes. It also consists of decision tress which holds various branches of data and the forest generated by this algorithm is trained with the help of bagging or bootstrap aggregating. Since it consists of multiple of decision trees, so it operates as a group by each individual trees generating a prediction. The most accurate prediction among the class is determine as the output. This is the reason why random tree classifier is also chosen highly in ML model since operating models as a committee is intended to surpass the individual constituent models. The prediction made with this algorithm is not much of higher accuracy in comparison to other algorithms. Although it predicts the output by calculating the average or mean of the output from different trees. The datasets with large number of input variables can also be used with this algorithm.

Suppose there are “N” number of data points and “M” number of input variables in datasets. And if “K” is the number of sampling group. Then the following is formed after the decision tree for each class is formed:

Ni = total number of data

Mi = input variables in group

Where “i” is the grouping value i.e., 1, 2, 3,…..k. (Nishant Yede, 2021)

This Ni and Mi creates a decision tree model for each class. And the tree model is based on m-try features which helps to calculate all the possible probabilities. Then the data split produced by the lead node is read while random forest ensemble uses the high probability value to classify the actual prediction (V. Jackins, 04 November 2020).

3. Solution

3.1. Explanation of the proposed solution:

In this coursework, I have decided to create a machine learning model which should be able to identify the disease of the patient based on the symptoms provided as the input by the user. KNN algorithm is used for this ML model. The training sets and test sets are provided in the KNN algorithm to test and train the model. After the training process is completed the performance output should be the prediction of the disease.

For this machine learning model, classification and regression algorithms are the most essential components of supervised learning. According to my research, the ML model of this coursework will have to deal with multinomial classification which is a type of classification that deals with the situation when the model classes have several target variables. Also, logistic regression can be integral part for the effective regression approach which will eventually solve addressing of binary classification problems if in case any occurs. For this coursework, I have categorized and research datasets from the website Kaggle for the disease prediction system based on symptoms. Also, another site i.e., data.world has some important required datasets which can be used to select the datasets. The records found here are qualitative records which has also been used and approved by expertise users. The datasets include multiple rows and columns of data that has the required information on symptoms for various diseases.

3.2. Explanation of the AI algorithm used:

The algorithm selected for this machine learning model is K Nearest Neighbors. This is the most used algorithm with the highest accuracy based on the research papers of multiple expertise users who has already worked with multiple algorithms before. It is also one of the basis supervised machine learning algorithm methods for classification problems. Each unlabelled data is set according to the majority label among its K-nearest neighbors. The working process includes prediction based on the K values which is like the training the model. All the available classes are stored and are the new data are classified on similarity measures. The algorithm implementation finds that if a new point of the class is comparable to the neighboring points. The other reason why KNN algorithms is used for this coursework is because this is the most applicable algorithm for search applications. The number of nearest neighbors of that new point that requited to be predicted is denoted by K in KNN method. KNN has high learning ability which will memorizes the training datasets and does all the calculations when the prediction is needed. This algorithm works by adding a new point to a dataset. The class in which the point belongs is determined and to predict the output, it selects the value of K.

> For example: If the K value is 8, the algorithm will select 8 neighbor points among those which has the shortest distance between the previous and new points. This helps to find, in which the class the new the point belongs.

But in order to find the best K value, several tests are done which includes the cross-validation techniques. And similarly, to find the best shortest distance between the points we need to use either Manhattan distance or Euclidean distance. The Manhattan distance helps to find the distance by calculated by adding the absolute difference of the two real vectors. Whereas the Euclidean distance is calculated by adding the absolute difference of two points in a Euclidean space.

To predict the diseases based on the symptoms, the KNN algorithm is used but to implement this algorithm in the more various task needs to be performed. Some of the important steps are given as follows:

i. Data Collection:

This is the I nitial but mostly important task which holds the factual medical information, results of the medication, information on symptoms and diseases etc. and others. The datasets are categorized based on these data. The data should also be split to further train and test the model. The test data is spitted in the class, we can use sklearn packages for testing and training.

ii. Model Implementation:

This includes importing libraries and reading the datasets for high accuracy calculations, data visualization and analyzation etc.

```
1  import pandas as pd
2  pd.options.mode.chained_assignment = None
3  import matplotlib.pyplot as pt
4  from matplotlib import rcParams
5  import seaborn as sns
6
7  from sklearn.neighbors import KNeighborsClassifier
8  from sklearn.metrics import classification_report,confusion_matrix
9  from sklearn.model_selection import cross_val_score
10 import sklearn.metrics as met
11 from sklearn.model_selection import train_test_split
12 from sklearn.preprocessing import StandardScaler
```

Figure 2: Importing libraries for Model Implementation

The datasets need to be imported, If the datasets is in CSV format it can

be imported like lines of code in Figure 3:

```
1   data=pd.read_csv('data.csv').drop(["id"],axis=1)
```

Figure 3: Importing the dataset

iii. Splitting the dataset:

The splitting datasets is important for testing and training the ML model. This is important because the separate independent and dependent variables need to be recognized. The data should be converted to a numpy array and then categorical variable encoding should be done. All the sample data which fits the model requirement should be tested with the model which helps to find the maximum and average accuracy. Also, the sample datasets should also be evaluated on the final ML model.

(Sewwandi, 2020)

Since, at the beginning we do not know the exact value of K, so in the model K value is kept as ‘n_neighbors’ value as 1 before fitting the model with the training data.

3.3. Pseudocode of the solution:

IMPORT libraries

IMPORT dataset

LOAD dataset

INITIALIZE value of K to chosen number of neighbors (n=1)

CALCULATE distance between query and current data

ADD distance and the index of datasets

SORT distance in increasing order

SELECT top K rows

PREDICT class for new data

GET labels from selected K rows

IF Regression

RETURN mean of K labels

ELSE IF Classification

RETURN mode of K labels

END IF

3.4. Diagrammatic representations of the solution (Flow Chart)

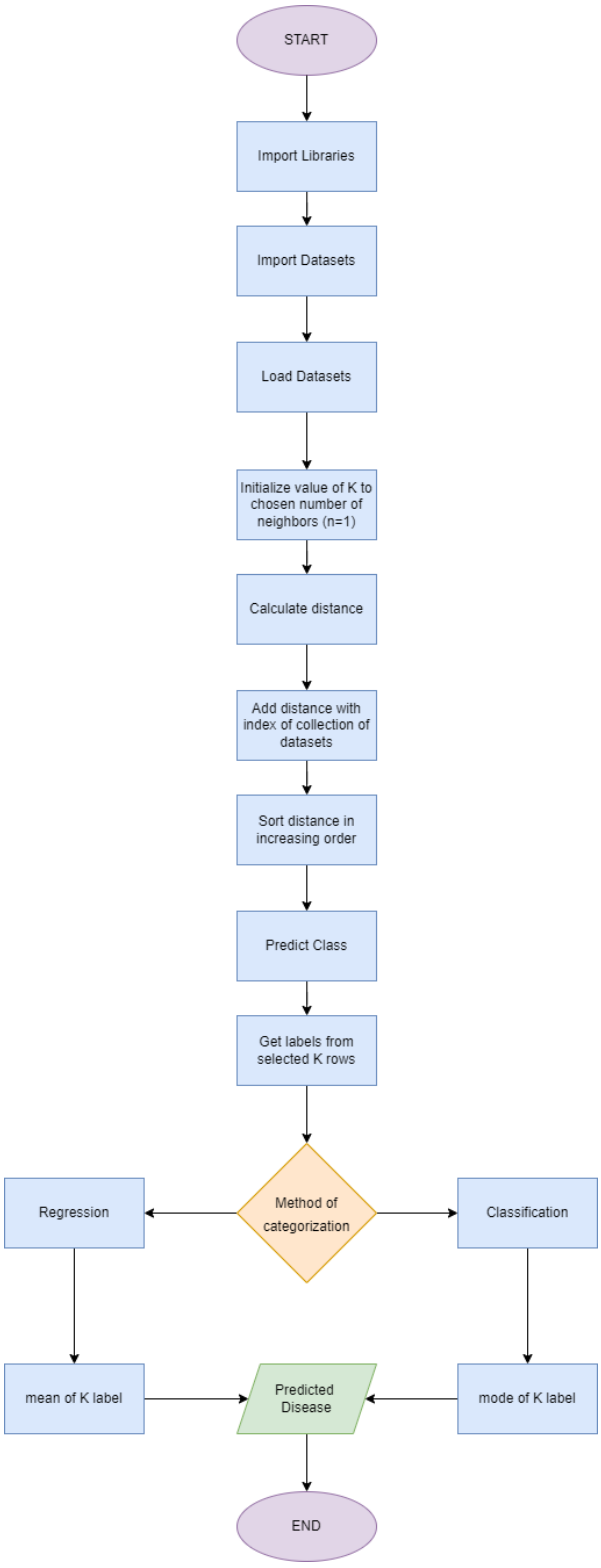

Figure 4: Flow Chart

3.5. Development Process

For the development of the system, Jupyter Notebook an open source web application was used. The development process for this system includes multiple steps, some of the major steps are

- Importing and Loading Datasets
- Training datasets
- Testing datasets
- Initialize K Nearest Neighbor
- Initialize five symptoms as input
- Print closest disease

This system is developed to predict the disease based on five symptoms provided by the user as input. With the help of Jupyterlab from Anaconda platform the interface prototype is constructed. For the construction libraries Numpy, Pandas and Matplotlib.pyplot were used. At the very beginning after importing the required libraries, the dataset loaded and analyzed for developing the model. After much research on datasets, I was able to find this appropriate datasets for my model and the datasets was which splitted and separated testing and training my ML model. The separate independent and dependent variables was recognized, and the remaining portion like converting the variables in a numpy array and encoding the array was done in my ML model.

- Importing the Libraries:

  i. The Matplotlib.pyplot library was used for the visualization of histogram/bar graphs of column data and for the scattering of matrix inversion of kernel density plots.
  ii. The Numpy library was used for the mathematical calculation in the ML model.
  iii. The Pandas library was used for the analysis of datsets.
  iv. The sklearn library was used importing necessary elements like

KNeighborsClassifiers, classification_report,confusion_matrix, and accuracy_score.

```
import matplotlib.pyplot as plt
import numpy as np
import pandas as pd
```

```
from sklearn.neighbors import KNeighborsClassifier
```

```
from sklearn.metrics import classification_report,confusion_matrix,accuracy_score
```

Figure 5: Importing Libraries for development of application

- Reading and Diplaying Datasets:

By the help of pandas library and its inbuilt read_csv() funtion, the datasets was imported and loaded. The main dataset used in this model which is training.csv dataset includes two columns “Disease” and “Symptoms”. The dataset was processed to help classify the data and the model was trained with this data.

	itching	skin_rash	nodal_skin_eruptions	continuous_sneezing	shivering	chills	joint_pain	stomach_pain	acidity	ulcers_on_tongue	...	blackheads	scu
0	1	1	1	0	0	0	0	0	0	0	...	0	
1	0	1	1	0	0	0	0	0	0	0	...	0	
2	1	0	1	0	0	0	0	0	0	0	...	0	
3	1	1	0	0	0	0	0	0	0	0	...	0	
4	1	1	1	0	0	0	0	0	0	0	...	0	
...	...	...	...	...	...	...	...	...	...	...	...	...	
4915	0	0	0	0	0	0	0	0	0	0	...	0	
4916	0	1	0	0	0	0	0	0	0	0	...	1	
4917	0	0	0	0	0	0	0	0	0	0	...	0	
4918	0	1	0	0	0	0	1	0	0	0	...	0	
4919	0	1	0	0	0	0	0	0	0	0	...	0	

4920 rows × 133 columns

Figure 6: Datasets

- Storing data in List:

A list named l1 is created to store the various symptoms according to various diseases.

```
#List of the symptoms is listed here in list l1.

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_s
    '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']
```

Figure 7: Storing values in list

And, a list named diseases is created to store the various diseases for the prediction according to the symptoms.

- Replacing the training data:

The training datasets which contains the diseases and symptoms is utilized for the ML model. The read_csv() function is used to analysis the datasets and initialized to store the values in a dataframe df. And, after the training datasets was read and loaded. The values obtained from importing the training.csv datasets was replaced in pandas by using an inbuilt function called replace(). So,

the column with number of names of diseases is replaced by the numbers from 0 to n-1 where n is the number of diseases present in the datasets. Then, the head() function is used to display the five top rows of the dataframe df.

```
#Reading the training .csv file
df=pd.read_csv("training.csv")
DF= pd.read_csv('training.csv', index_col='prognosis')
#Replacing the values in the imported file by pandas by the inbuilt function replace in pandas.

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)
#df.head()
DF.head()
```

prognosis	itching	skin_rash	nodal_skin_eruptions	continuous_sneezing	shivering	chills	joint_pain	stomach_pain	acidity	ulcers_on_tongue	...	pus_filled_p
Fungal infection	1	1	1	0	0	0	0	0	0	0	...	
Fungal infection	0	1	1	0	0	0	0	0	0	0	...	
Fungal infection	1	0	1	0	0	0	0	0	0	0	...	
Fungal infection	1	1	0	0	0	0	0	0	0	0	...	
Fungal infection	1	1	1	0	0	0	0	0	0	0	...	

5 rows × 132 columns

Figure 8: Replacing and Displaying Training Datasets

- Distribution graph

In this plotPerColumnDistribution() function, the codes to create and display the histogram/bar graph of column data is done. The Matplotlib.pyplot library is imported to use this function for the visualization of distribution graph of the columns of the training.csv file.

```
# Distribution graphs (histogram/bar graph) of column data
def plotPerColumnDistribution(df1, nGraphShown, nGraphPerRow):
    nunique = df1.nunique()
    df1 = df1[[col for col in df if nunique[col] > 1 and nunique[col] < 50]] # For displaying purposes, pick columns that have be
    nRow, nCol = df1.shape
    columnNames = list(df1)
    nGraphRow = (nCol + nGraphPerRow - 1) / nGraphPerRow
    plt.figure(num = None, figsize = (6 * nGraphPerRow, 8 * nGraphRow), dpi = 80, facecolor = 'w', edgecolor = 'k')
    for i in range(min(nCol, nGraphShown)):
        plt.subplot(nGraphRow, nGraphPerRow, i + 1)
        columnDf = df.iloc[:, i]
        if (not np.issubdtype(type(columnDf.iloc[0]), np.number)):
            valueCounts = columnDf.value_counts()
            valueCounts.plot.bar()
        else:
            columnDf.hist()
        plt.ylabel('counts')
        plt.xticks(rotation = 90)
        plt.title(f'{columnNames[i]} (column {i})')
    plt.tight_layout(pad = 1.0, w_pad = 1.0, h_pad = 1.0)
    plt.show()
```

The output of the histogram/bar graph is given below:

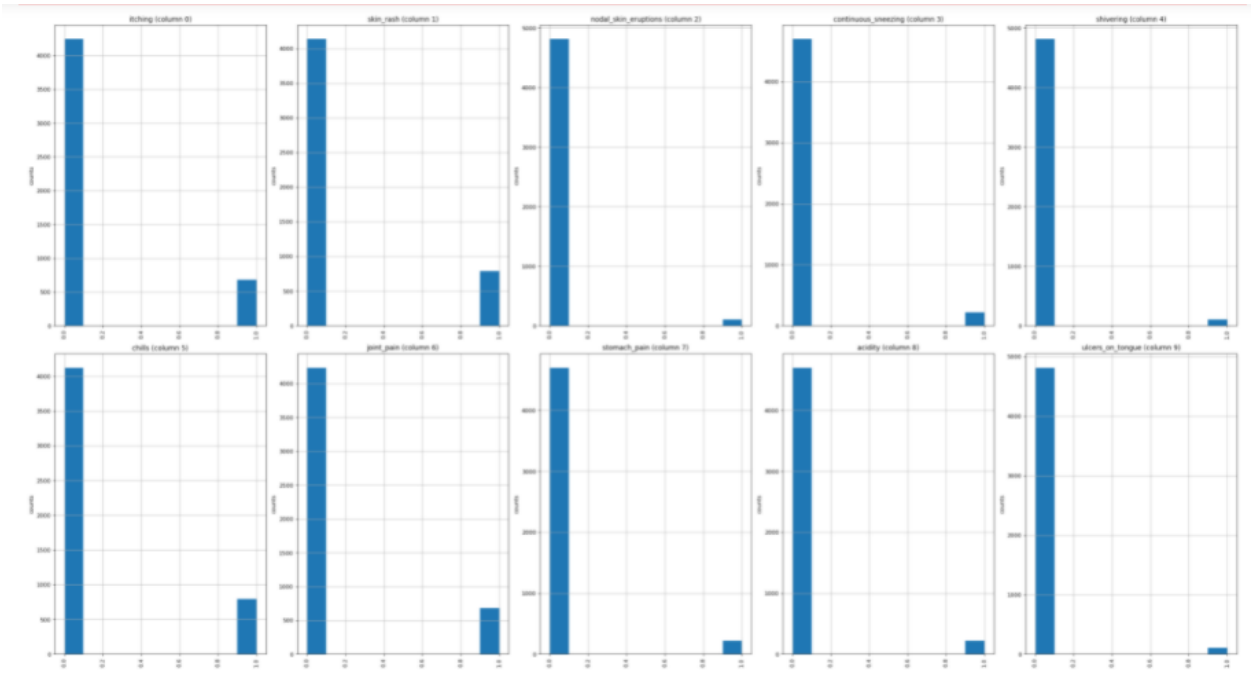

Figure 9: Implementaion and Displaying Distribution graph

- Scattering the Matrix:

  Using the plotScatterMatrix, the scattering of the matrix and density plots of the column of the training.csv file is done.

```
# Scatter and density plots
def plotScatterMatrix(df1, plotSize, textSize):
    df1 = df1.select_dtypes(include =[np.number]) # keep only numerical columns
    # Remove rows and columns that would lead to df being singular
    df1 = df1.dropna('columns')
    df1 = df1[[col for col in df if df[col].nunique() > 1]] # keep columns where there are more than 1 unique values
    columnNames = list(df)
    if len(columnNames) > 10: # reduce the number of columns for matrix inversion of kernel density plots
        columnNames = columnNames[:10]
    df1 = df1[columnNames]
    ax = pd.plotting.scatter_matrix(df1, alpha=0.75, figsize=[plotSize, plotSize], diagonal='kde')
    corrs = df1.corr().values
    for i, j in zip(*plt.np.triu_indices_from(ax, k = 1)):
        ax[i, j].annotate('Corr. coef = %.3f' % corrs[i, j], (0.8, 0.2), xycoords='axes fraction', ha='center', va='center', size
    plt.suptitle('Scatter and Density Plot')
    plt.show()
```

The output after the scattering is given below:

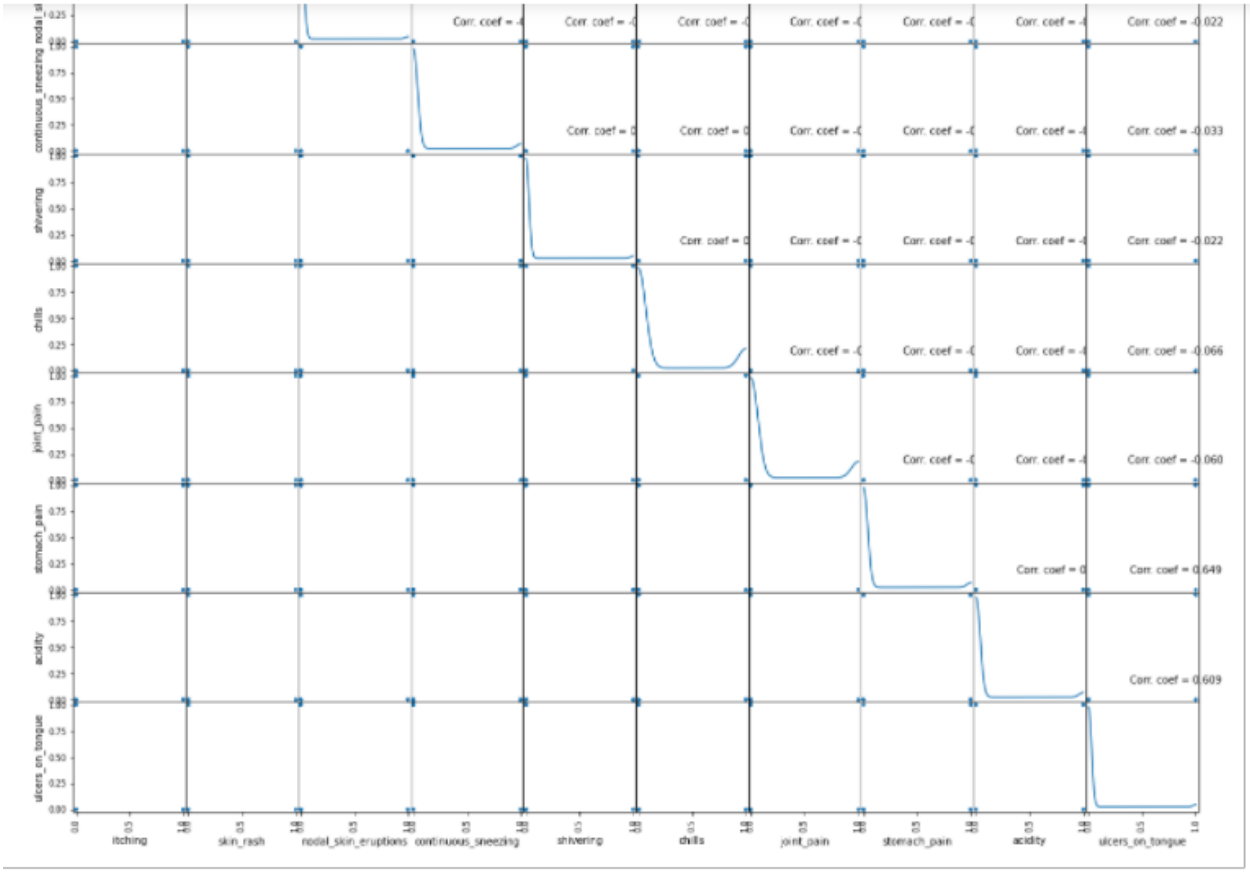

Figure 10: Implementation and Displaying Scattering of Matrix

- Replacing the testing data

The testing datasets which contains the diseases and symptoms is utilized for the ML model. The read_csv() function is to analysis the testing datasets and initialized to store the values in a dataframe tr. And, after the testing datasets was read and loaded. The values obtained from importing the testing.csv datasets was replaced in pandas by using an inbuilt function called replace().

```
#Reading the  testing.csv file
tr=pd.read_csv("testing.csv")

#Using inbuilt function replace in pandas for replacing the values

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)
tr.head()
```

	itching	skin_rash	nodal_skin_eruptions	continuous_sneezing	shivering	chills	joint_pain	stomach_pain	acidity	ulcers_on_tongue	...	blackheads	scurrin
0	1	1	1	0	0	0	0	0	0	0	...	0	
1	0	0	0	1	1	1	0	0	0	0	...	0	
2	0	0	0	0	0	0	0	1	1	1	...	0	
3	1	0	0	0	0	0	0	0	0	0	...	0	
4	1	1	0	0	0	0	0	1	0	0	...	0	

5 rows × 133 columns

Figure 11: Replacing and Displaying Testing Datasets

- Implementation of KNN algorithm

```
from sklearn.neighbors import KNeighborsClassifier

knn=KNeighborsClassifier(n_neighbors=5,metric='minkowski',p=2)
knn=knn.fit(X,np.ravel(y))
```

```
from sklearn.metrics import classification_report,confusion_matrix,accuracy_score

y_pred=knn.predict(X_test)
print("kNearest Neighbour")
print("Accuracy")
print(accuracy_score(y_test, y_pred))
print(accuracy_score(y_test, y_pred,normalize=False))
print("Confusion matrix")
conf_matrix=confusion_matrix(y_test,y_pred)
print(conf_matrix)
```

```
kNearest Neighbour
Accuracy
0.9512195121951219
39
Confusion matrix
[[1 0 0 ... 0 0 0]
 [0 1 0 ... 0 0 0]
 [0 0 0 ... 0 0 0]
 ...
 [0 0 0 ... 1 0 0]
 [0 0 0 ... 0 1 0]
 [0 0 0 ... 0 0 1]]
```

Figure 12: Implementation of KNN algorithm

For training the model KNN algorithm is used and the prediction of the nearest disease based on the symptoms is done. The accuracy as per the KNN algorithm is gained 96% accuracy. Along with accuracy the total number of correctly classified accuracy_score and the confusion matrix is displayed as output.

- User Input Field

```
global symptoms1
global symptoms2
global symptoms3
global symptoms4
global symptoms5

def inputFuntion():
    global symptoms1
    global symptoms2
    global symptoms3
    global symptoms4
    global symptoms5

    symptoms1=input("Enter first symptom: ")
    symptoms2=input("Enter second symptom: ")
    symptoms3=input("Enter third symptom: ")
    symptoms4=input("Enter fourth symptom: ")
    symptoms5=input("Enter fifth symptom: ")
    print(len(symptoms1))

inputFuntion()
```

```
Enter first symptom:  redness_of_eyes
Enter second symptom: yellowing_of_eyes
Enter third symptom: blurred_and_distorted_vision
Enter fourth symptom: puffy_face_and_eyes
Enter fifth symptom: watering_from_eyes
16
```

Figure 13: User Input Field

Here in this figure, five global variable is given in which each of the input provided by the user as symptoms is stored. And a inputFunction() created and called which provides the input field for the user to provide the symptoms.

- Prediction

```
print(symptoms1)

if (symptoms1 == "" or symptoms2 == "" or symptoms3 == "" or symptoms4 == "" or symptoms5 == ""):
    print("Please fill all the symptoms fields.")
    inputFuntion()
else:
    all_symptoms = [symptoms1,symptoms2,symptoms3,symptoms4,symptoms5]

    for k in range(0,len(l1)):
        for z in all_symptoms:
            if(z==l1[k]):
                l2[k]=1

    inputtest = [l2]
    predict = knn.predict(inputtest)
    predicted=predict[0]
    print(predicted)
```

```
h='no'
for a in range(0,len(disease)):
    if(predicted == a):
        h='yes'
#         print(a)
        break
if (h=='yes'):
    print("Disease: " + disease[a])
else:
    print("Not Found")
```

Figure 14: Prediction Step

In this above step, the validation, prediction and displaying the predicted disease is shown. All the five symptoms provided by the user is stored in a list named all_symptoms, and with the use of KNN algorithm provided above the prediction is done for the disease. The user must enter five symptoms for the prediction of the disease. So, if any of the input field is left empty then an error message instructing the user to fill all the symptoms field is displayed.

- Output

```
h='no'
for a in range(0,len(disease)):
    if(predicted == a):
        h='yes'
#         print(a)
        break
if (h=='yes'):
    print("Disease: " + disease[a])
else:
    print("Not Found")
```

```
Disease: Allergy
```

Figure 15: Output of the program

Based on the five symptoms provided the closest disease predicted is given as output.

3.5.1. Tools Used

a. Anaconda Navigator:

Anaconda Navigator is a graphical user interface (GUI) which is mainly used for running python programs. The python programs can be run easily with anaconda navigator without having the user to use system terminal to install packages, manage environment etc. Every necessary packages and tools are provided inside navigator by default which helps the user to run programs without much hard work in typing commands in windows terminal.

b. Jupyter Notebook

Jupyter Notebook is an open-source web application which is used in multiple platform like for data visualization, code sharing, graphics etc. It is also know as multi-language computing environment because it is able to support more than 40 programming language to its users. From raw samples of data or text to its graphics, visualization etc. Jupyter Notebook can be used to convert these data into interactive visible content. Each of the files used in a notebook has a kernel to control those documents.

(Sejuti, 2022)

3.5.2. Libraries Used

a. Pandas

Pandas is a python library which is mainly used for manipulating data. It has two data structures which are Series and Dataframe. This data structure has its fata manipulated by pandas with the conjunction of pandas with other libraries. This library takes Numpy libraries as replication by building Pandas on top of Numpy library. With the help of Pandas libraries, the data can be used as input which is later used for plotting functions of Matplotlib, as well as used for statistical analysis in SciPy.

(Aggarwal, 2021)

b. Numpy

Numerical Python which is shortly known as Numpy, is a python library which includes multidimensional array objects. On these array objects mathematical and logical operations are performed using Numpy python library. These operations includes shape manipulation, algebra and random number generation. This package mainly is used along with Scientific Python package and Matpot library. So, this combination is also known to use for the replacement of Matlab.

(mygreatlearning, 2022)

c. Matplotlib.pyplot

Pyplot is a python library which is used for 2D graphics in web application server, GUI, python shell etc. And this library has set of functions which is included in the Matplotlib visualization package. With the help of this library 2D graphics is constructed, and the 2D graphics includes plot functions for manipulation of figures, defining a plotting area, plotting lines, and other graphs.

3.6. Achieved Results

The screenshots provided here are the achieved results as output of each steps while building the model.

- Importing and loading Datasets

```
datasets=pd.read_csv("training.csv")
```

```
datasets.head
```

```
<bound method NDFrame.head of       itching  skin_rash  nodal_skin_eruptions  continuous_sneezing  \
0           1          1                     1                    0
1           0          1                     1                    0
2           1          0                     1                    0
3           1          1                     0                    0
4           1          1                     1                    0
...       ...        ...                   ...                  ...
4915        0          0                     0                    0
4916        0          1                     0                    0
4917        0          0                     0                    0
4918        0          1                     0                    0
4919        0          1                     0                    0

      shivering  chills  joint_pain  stomach_pain  acidity  ulcers_on_tongue  \
0             0       0           0             0        0                 0
1             0       0           0             0        0                 0
2             0       0           0             0        0                 0
3             0       0           0             0        0                 0
4             0       0           0             0        0                 0
...         ...     ...         ...           ...      ...               ...
4915          0       0           0             0        0                 0
4916          0       0           0             0        0                 0
4917          0       0           0             0        0                 0
4918          0       0           1             0        0                 0
4919          0       0           0             0        0                 0
```

Figure 16: Screenshots of Importing and loading Datasets

- Replacing the imported file in pandas

```
#Reading the training .csv file
df=pd.read_csv("training.csv")
DF= pd.read_csv('training.csv', index_col='prognosis')
#Replacing the values in the imported file by pandas by the inbuilt function replace in pandas.

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)
#df.head()
DF.head()
```

prognosis	itching	skin_rash	nodal_skin_eruptions	continuous_sneezing	shivering	chills	joint_pain	stomach_pain	acidity	ulcers_on_tongue	...	pus_filled_p
Fungal infection	1	1	1	0	0	0	0	0	0	0	...	
Fungal infection	0	1	1	0	0	0	0	0	0	0	...	
Fungal infection	1	0	1	0	0	0	0	0	0	0	...	
Fungal infection	1	1	0	0	0	0	0	0	0	0	...	
Fungal infection	1	1	1	0	0	0	0	0	0	0	...	

5 rows × 132 columns

Figure 17: Screenshots of Replacing and Displaying Testing Datasets values

- Histogram/Bar graph of training datasets

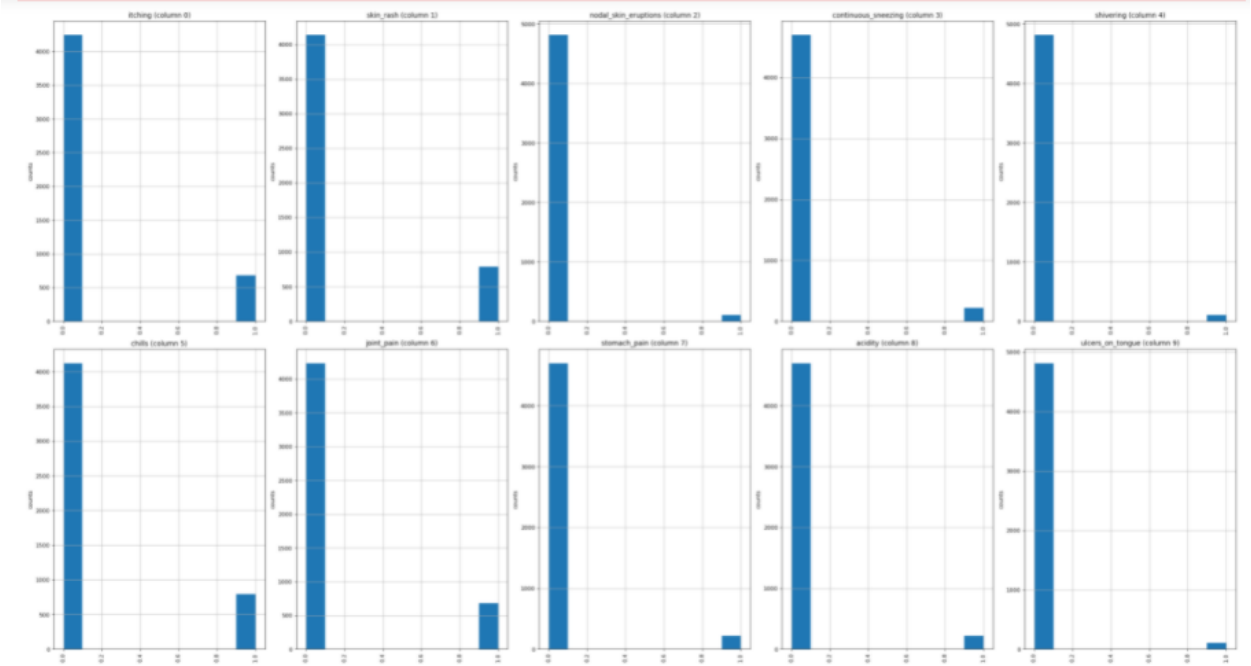

Figure 18: Screenshots of Distribution Graph

- Scattering the datasets matrix

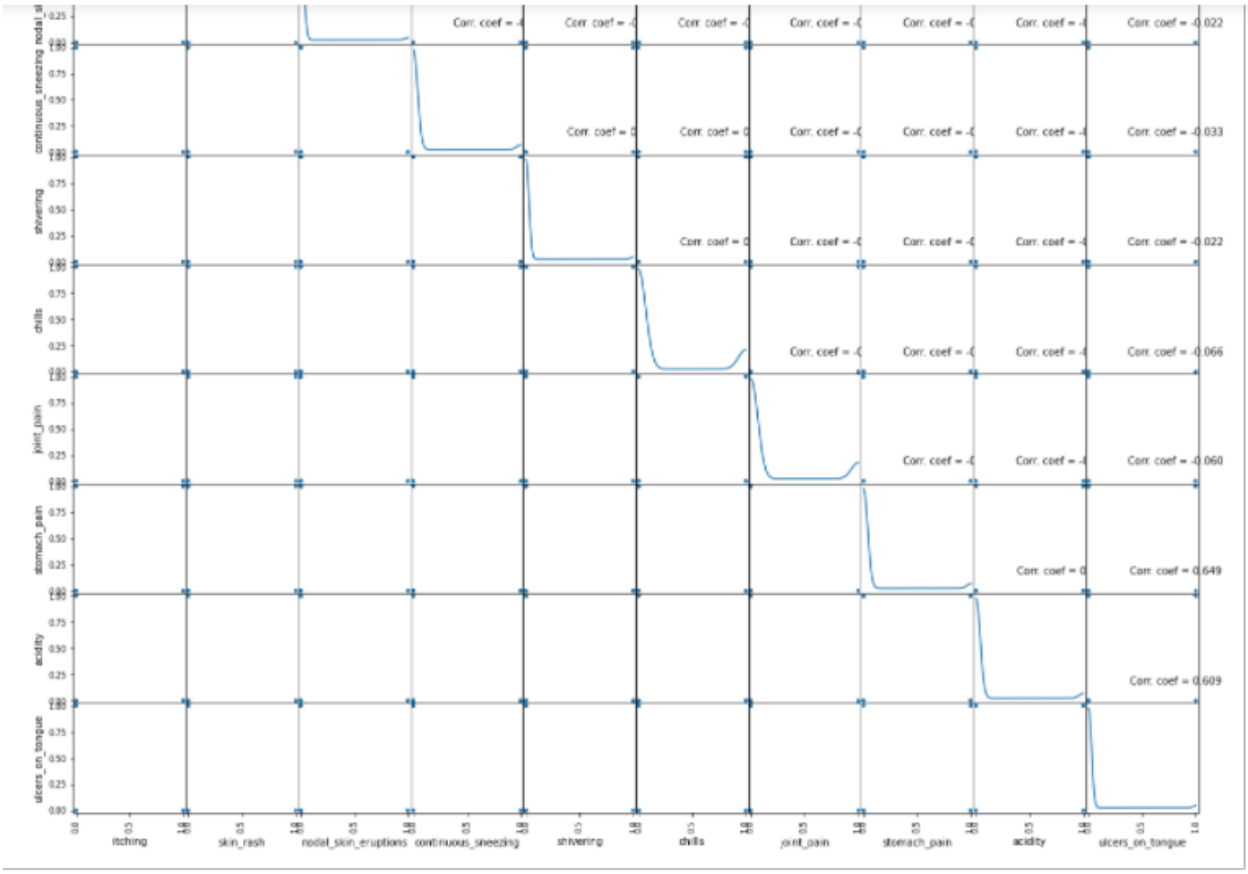

Figure 19: Screenshots of Scattering of Matrix

- Accuracy Output and Confusion matrix

```
kNearest Neighbour
Accuracy
0.9512195121951219
39
Confusion matrix
[[1 0 0 ... 0 0 0]
 [0 1 0 ... 0 0 0]
 [0 0 0 ... 0 0 0]
 ...
 [0 0 0 ... 1 0 0]
 [0 0 0 ... 0 1 0]
 [0 0 0 ... 0 0 1]]
```

Figure 20: Screenshots of Accuracy score

- User Input Fields

```
global symptoms1
global symptoms2
global symptoms3
global symptoms4
global symptoms5

def inputFuntion():
    global symptoms1
    global symptoms2
    global symptoms3
    global symptoms4
    global symptoms5

    symptoms1=input("Enter first symptom: ")
    symptoms2=input("Enter second symptom: ")
    symptoms3=input("Enter third symptom: ")
    symptoms4=input("Enter fourth symptom: ")
    symptoms5=input("Enter fifth symptom: ")
    print(len(symptoms1))

inputFuntion()
```

```
Enter first symptom:  redness_of_eyes
Enter second symptom: yellowing_of_eyes
Enter third symptom: blurred_and_distorted_vision
Enter fourth symptom: puffy_face_and_eyes
Enter fifth symptom: watering_from_eyes
16
```

Figure 21: Screenshots of input fields for user

- Prediction as output of the system

```
print(symptoms1)

if (symptoms1 == "" or symptoms2 == "" or symptoms3 == "" or symptoms4 == "" or symptoms5 == ""):
    print("Please fill all the symptoms fields.")
    inputFuntion()
else:
    all_symptoms = [symptoms1,symptoms2,symptoms3,symptoms4,symptoms5]

    for k in range(0,len(l1)):
        for z in all_symptoms:
            if(z==l1[k]):
                l2[k]=1

    inputtest = [l2]
    predict = knn.predict(inputtest)
    predicted=predict[0]
    print(predicted)
```

Figure 22: Screenshots of prediction steps

```
h='no'
for a in range(0,len(disease)):
    if(predicted == a):
        h='yes'
#         print(a)
        break
if (h=='yes'):
    print("Disease: " + disease[a])
else:
    print("Not Found")
```

```
Disease: Allergy
```

Figure 23: Screenshots of output of program

4. Conclusion

4.1. Analysis of Work:

This coursework was all about preparing a ML model based on the proposal we submitted as first draft of coursework by researching about AI topics. This proposal coursework included understanding of ML topic and how a problem domain should be researched. Along with the problem domain also the solution for that problem was researched and explained. According to the proposed solution and problem domain, a ML model was to be developed. We were provided with list of material via Google Classroom app to learn and understand about Artificial Intelligence and the algorithms to use in the model. So, based on the knowledge of algorithms from lecture and

workshop sessions, this AI application was developed.

My chosen topic was about an ML model which should be able to detect diseases based on the symptoms provided by the users as input. This coursework includes the prototype of the machine leaning model application. This topic was chose by me to extend my interest and knowledge in the field of AI. An AI application to predict the disease of the user based on the symptoms has many advantages in the field of medical sections. Also, this report includes the details of the algorithms and how these algorithms are implemented in a machine learning model. With this, the development process and screenshots of the system is also included. The algorithm used for this model was done according to the problem domain as well as for solution for this problem.

4.2. How solution addresses real world problem

Normally, people don’t pay much attention on simple symptoms and choses to ignore the problem. Any kind of simple symptoms like headache, coughing etc. may lead to major health problem later. People are not even aware about what diseases they can have. Any people who neglect the simple symptoms might be needing an urgent health care under the expertise supervision. Depending on the severity of the symptoms, the medical procedures should not be skipped.

To help detect the disease without consulting a doctor directly, this ML model developed can be used efficiently. Determining simple symptoms to figure out the disease urgently, whether the disease is critically or not this model can be handy. Such system can decrease the rush at OPDs of hospitals and also decrease the workload on medical staffs. People don’t need to rush in busy hours to consult the medical expertise. By tracking vital signs of the patient symptoms, the medical procedures can be continued. Even before consulting doctors and visiting hospitals directly, the details of the disease can be known which will help to take preventions and ease the patient

discomfort. Lack of medical care or even delayed medical care can lead a human body in serious disadvantage so this machine learning model developed with the use of KNN algorithm can be used in real world scenario to decrease risk factors of patients.

5. References

Aggarwal, N. (2021, Dec 09). *introduction-to-pandas-in-python*. Retrieved 2021, from geeksforgeeks: https://www.geeksforgeeks.org/introduction-to-pandas-in-python/

Algorithmia. (2020, Jan 07). *how-machine-learning-works*. Retrieved from algorithmia: https://algorithmia.com/blog/how-machine-learning-works

Great Learning Team. (2021, Mar 24). *what-is-machine-learning*. Retrieved from mygreatlearning: https://www.mygreatlearning.com/blog/what-is-machine-learning

IBM. (2021). *artificial-intelligence-medicine*. Retrieved from ibm: https://www.ibm.com/topics/artificial-intelligence-medicine

JavaTpoint. (2011-2021). *supervised-machine-learning*. Retrieved from JavaTpoint: https://www.javatpoint.com/supervised-machine-learning

K. Venkatesh, K. D. (2021). Identification of Disease Prediction Based on Symptoms Using Machine Learning. *A Journal Of Composition Theory*, 8.

Khakharia, R. K. (2020, Oct 08). *papers.cfm*. Retrieved from SSRN: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3661426

Kolli, B. T. (Aug 2021). Symptoms_Based_Multiple_Disease_Prediction_Model_using_Machine_Learning_Approach. *Internation Journal of Innovative Technology and Exploring Engineering*, 7.

mygreatlearning. (2022, Jan 11). *python-numpy-tutorial*. Retrieved 2022, from mygreatlearning: https://www.mygreatlearning.com/blog/python-numpy-tutorial/

Nishant Yede, R. K. (2021). General Disease Prediction Based On Symptoms Provided By Patient. *Open Access International Journal Of Science And Engineering*, 6.

Sejuti. (2022). *why-jupyter-notebooks-are-so-popular-among-data-scientists*. Retrieved from analyticsindiamag: https://analyticsindiamag.com/why-jupyter-notebooks-are-so-popular-among-data-scientists/

Sewwandi, T. (2020, Sep 13). *predicting-cardiovascular-disease-using-k-nearest-neighbors-algorithm*. Retrieved from towardsdatascience: https://towardsdatascience.com/predicting-cardiovascular-disease-using-k-nearest-neighbors-algorithm-614b0ecbf122

V. Jackins, S. V. (04 November 2020). AI-based smart prediction of clinical disease using random forest classifier and Naive Bayes. *The Journal of Supercomputing*, 77.