

INTERNSHIP REPORT

|  |  |
| --- | --- |
| INTERNSHIP DETAILS | |
| NAME (‘s): | MUZAMMIL AHMED V T  SYED HANNAN |
| College Name: | HKBK COLLEGE OF ENGINEERING |
| Project Title: | DISEASE PREDICTION USING SYMPTOMS |
| Program: | 16/08/2023 to 16/09/2023 (1 Month Internship Program)  ON  DATA SCIENCE AND MACHINE LEARNING |
| Project Handled and Trained By: | This project was supervised by our experienced trainer Hemesh Muniraji sir providing us with Knowledge, Guidance, Insights and directions throughout our work experience. The mentorship ensured the development and implementation of the disease prediction project. |

TEAM OF 2

**ABSTRACT**

Disease prediction is a critical area in healthcare that has witnessed significant advancements with the integration of machine learning (ML) techniques. This abstract provides a comprehensive overview of the state-of-the-art in disease prediction using ML, emphasizing the pivotal role it plays in early diagnosis and proactive healthcare management. Machine learning models have revolutionized disease prediction by leveraging vast amounts of patient data, including electronic health records, genomic data, and lifestyle information. This approach enables healthcare professionals to identify risk factors and predict diseases with unprecedented accuracy.

The first section of this abstract discusses the data sources and preprocessing steps involved in disease prediction. We delve into the challenges related to data quality, privacy, and ethical concerns, highlighting the need for robust data handling techniques and ensuring patient privacy and consent. The second section highlights the various ML algorithms employed for disease prediction. Supervised learning techniques such as logistic regression, decision trees, and support vector machines are discussed, along with the growing popularity of deep learning methods, including convolutional neural networks and recurrent neural networks. The third section explores real-world applications of disease prediction using ML across various medical domains, such as cardiovascular diseases, cancer, diabetes, and infectious diseases. These applications demonstrate the tangible benefits of ML in enhancing diagnostic accuracy, optimizing treatment plans, and ultimately improving patient outcomes.

In conclusion, disease prediction using machine learning has emerged as a powerful tool in modern healthcare. By effectively leveraging diverse data sources and advanced algorithms, ML has the potential to transform healthcare delivery, making it more proactive, efficient, and patient-centred. However, ongoing research and collaboration between healthcare professionals and data scientists are essential to overcome challenges and unlock the full potential of ML in disease prediction.

**INTRODUCTION**

Disease prediction has long been a cornerstone of modern healthcare, with early detection and intervention playing a pivotal role in improving patient outcomes and reducing the burden on healthcare systems. In recent years, the integration of machine learning (ML) techniques into healthcare has ushered in a new era of disease prediction, characterized by unparalleled accuracy, scalability, and adaptability. This introduction provides a glimpse into the transformative potential of ML in the field of disease prediction.

Machine learning, a subset of artificial intelligence, equips healthcare professionals with powerful tools to sift through vast troves of patient data, uncover hidden patterns, and make informed predictions about the likelihood of diseases. This paradigm shift is fueled by the availability of electronic health records, genomic data, and wearable devices, all of which provide an unprecedented wealth of information for training predictive models.

As we embark on this exploration of disease prediction using machine learning, we will delve into the various data sources, algorithms, real-world applications, challenges, and future prospects that define this exciting frontier in healthcare. From predicting cardiovascular diseases to identifying early-stage cancers and monitoring the progression of chronic conditions, ML has the potential to revolutionize healthcare by offering timely, data-driven insights to clinicians and researchers alike.

Numpy : is used for deal with the array.

Matplotlib: is used to create map.

Tensorflow: is a deep learning model in python used form training and predicting the data by

using deep neural network.

Sklearn : is used for machine learning to build model

We applied Long Short-Term Memory (LSTM). Recurrent neural networks (RNN) are

expanded upon by LSTM networks. where the LSTM model is used to predict the data. After processing, we take the predicted data from the LSTM model. The projected data is then used with Matplotlib to create a graph.

**CODE**

# Import Dependencies

import csv

import pandas as pd

import numpy as np

from collections import defaultdict

import seaborn as sns

import matplotlib.pyplot as plt

%matplotlib inline

# Read Raw Dataset

df = pd.read\_excel('raw\_data.xlsx')

df.head()

# Fill all NaN with the values above

data = df.fillna(method='ffill')

data.head()

# Process Disease and Symptom Names

def process\_data(data):

data\_list = []

data\_name = data.replace('^','\_').split('\_')

n = 1

for names in data\_name:

if (n % 2 == 0):

data\_list.append(names)

n += 1

return data\_list

# Data Cleanup

disease\_list = []

disease\_symptom\_dict = defaultdict(list)

disease\_symptom\_count = {}

count = 0

for idx, row in data.iterrows():

# Get the Disease Names

if (row['Disease'] !="\xc2\xa0") and (row['Disease'] != ""):

disease = row['Disease']

disease\_list = process\_data(data=disease)

count = row['Count of Disease Occurrence']

# Get the Symptoms Corresponding to Diseases

if (row['Symptom'] !="\xc2\xa0") and (row['Symptom'] != ""):

symptom = row['Symptom']

symptom\_list = process\_data(data=symptom)

for d in disease\_list:

for s in symptom\_list:

disease\_symptom\_dict[d].append(s)

disease\_symptom\_count[d] = count

# See that the data is Processed Correctly

disease\_symptom\_dict

# Count of Disease Occurence w.r.t each Disease

disease\_symptom\_count

# Save cleaned data as CSV

f = open('cleaned\_data.xlsx', 'w')

with f:

writer = csv.writer(f)

for key, val in disease\_symptom\_dict.items():

for i in range(len(val)):

writer.writerow([key, val[i], disease\_symptom\_count[key]])

# Remove any rows with empty values

df.replace(float('nan'), np.nan, inplace=True)

df.dropna(inplace=True)

from sklearn import preprocessing

n\_unique = len(df['Symptom'].unique())

n\_unique

df.dtypes

# Encode the Labels

from sklearn.preprocessing import LabelEncoder

from sklearn.preprocessing import OneHotEncoder

label\_encoder = LabelEncoder()

integer\_encoded = label\_encoder.fit\_transform(df['Symptom'])

print(integer\_encoded)

# One Hot Encode the Labels

onehot\_encoder = OneHotEncoder(sparse=False)

integer\_encoded = integer\_encoded.reshape(len(integer\_encoded), 1)

onehot\_encoded = onehot\_encoder.fit\_transform(integer\_encoded)

print(onehot\_encoded)

onehot\_encoded[0]

len(onehot\_encoded[0])

cols = np.asarray(df['Symptom'].unique())

cols

# Create a new dataframe to save OHE labels

df\_ohe = pd.DataFrame(columns = cols)

df\_ohe.head()

for i in range(len(onehot\_encoded)):

df\_ohe.loc[i] = onehot\_encoded[i]

df\_ohe.head()

len(df\_ohe)

# Disease Dataframe

df\_disease = df['Disease']

df\_disease.head()

# Concatenate OHE Labels with the Disease Column

df\_concat = pd.concat([df\_disease,df\_ohe], axis=1)

df\_concat.head()

df\_concat.drop\_duplicates(keep='first',inplace=True)

df\_concat.head()

len(df\_concat)

cols = cols[1:]

# Since, every disease has multiple symptoms, combine all symptoms per disease per row

df\_concat = df\_concat.groupby('Disease').sum()

df\_concat = df\_concat.reset\_index()

df\_concat[:5]

len(df\_concat)

df\_concat.to\_csv("training\_dataset.csv", index=False)

# One Hot Encoded Features

X = df\_concat[cols]

# Labels

y = df\_concat['Disease']

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

from sklearn.naive\_bayes import MultinomialNB

from sklearn import tree

from sklearn.tree import DecisionTreeClassifier, export\_graphviz

# Train Test Split

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=101)

len(X\_train), len(y\_train)

len(X\_test), len(y\_test)

dt = DecisionTreeClassifier()

clf\_dt=dt.fit(X, y)

clf\_dt.score(X, y)

export\_graphviz(dt,

out\_file='./tree.dot',

feature\_names=cols)

!pip install graphviz

from graphviz import Source

from sklearn import tree

graph = Source(export\_graphviz(dt,

out\_file=None,

feature\_names=cols))

png\_bytes = graph.pipe(format='png')

with open('tree.png','wb') as f:

f.write(png\_bytes)

from IPython.display import Image

Image(png\_bytes)

disease\_pred = clf\_dt.predict(X)

disease\_real = y.values

for i in range(0, len(disease\_real)):

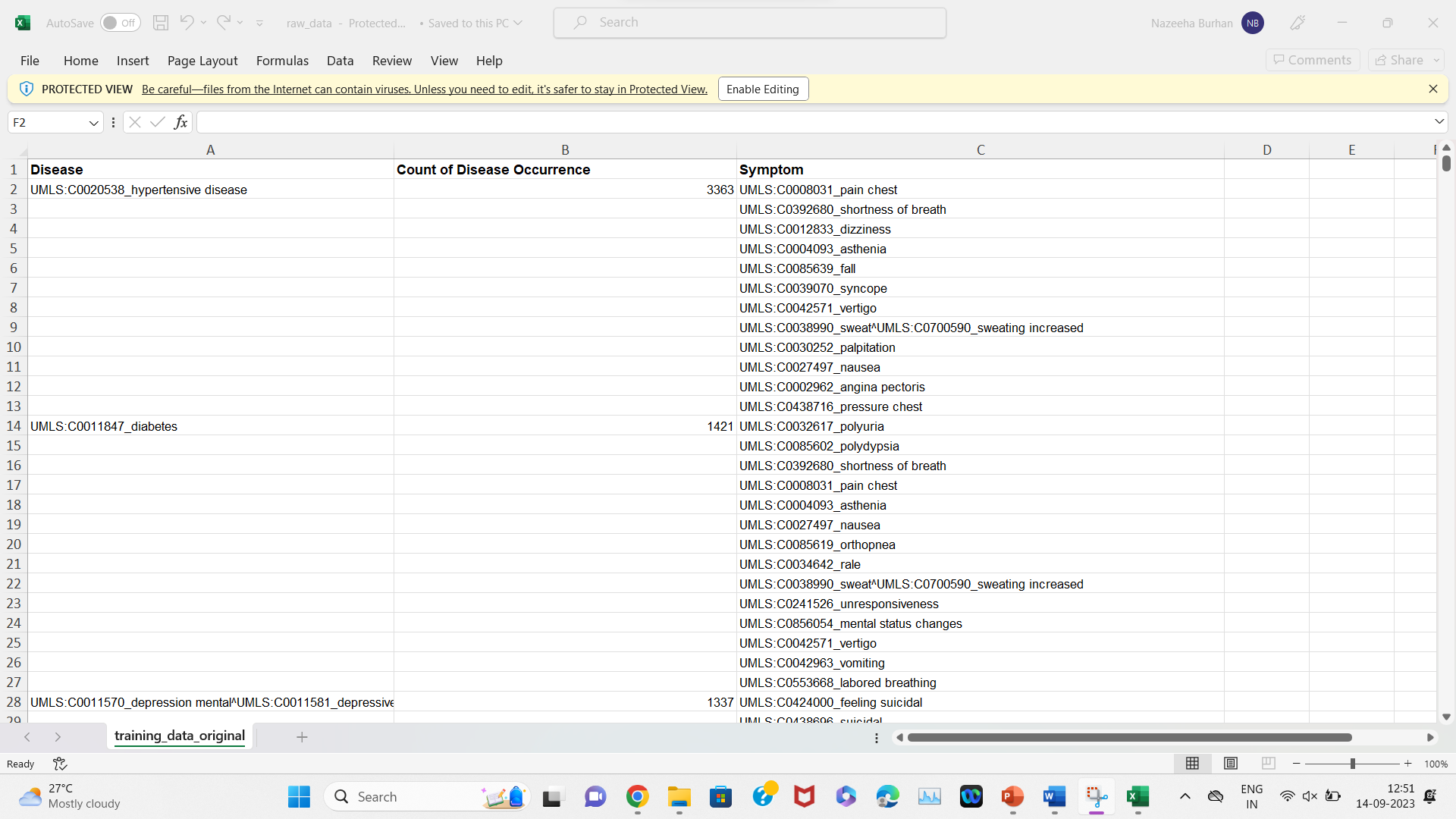
if disease\_pred[i]!=disease\_real[i]:

print ('Pred: {0}\nActual: {1}\n'.format(disease\_pred[i], disease\_real[i]))

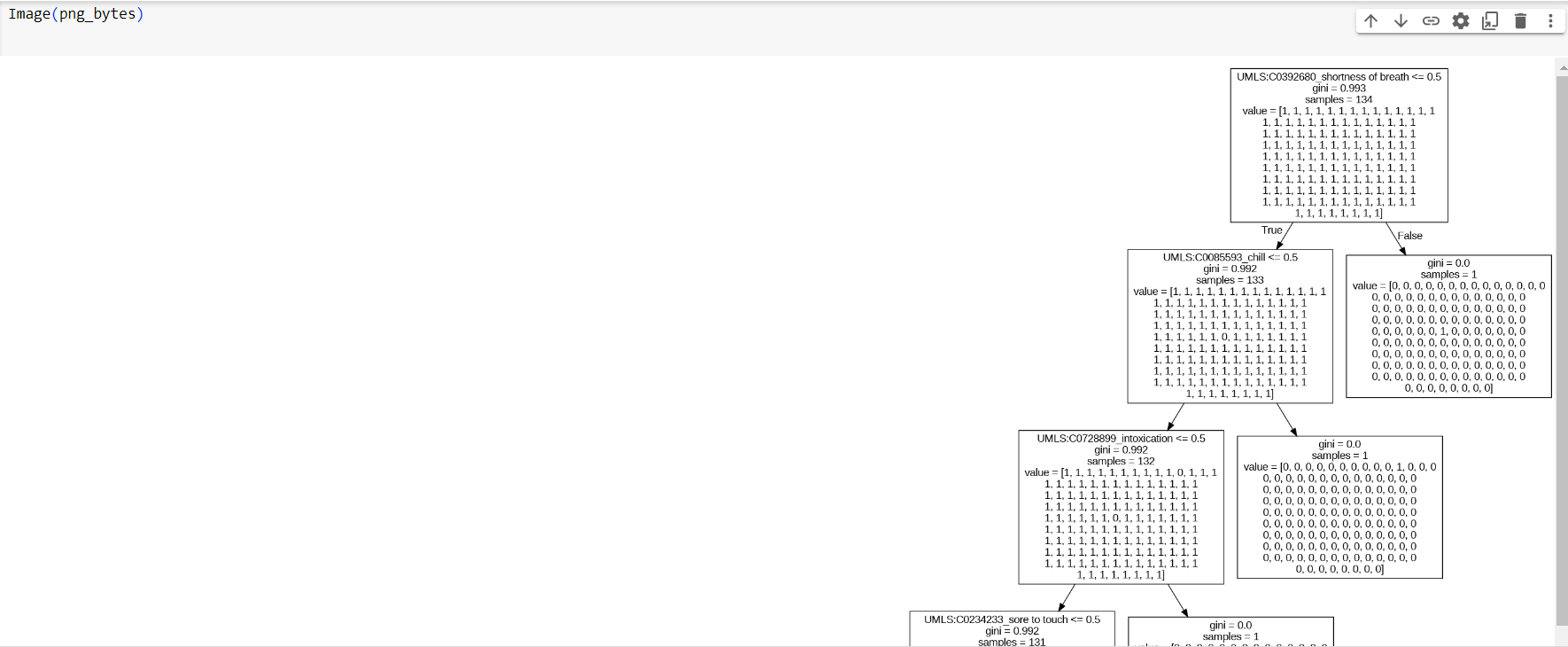
**SCREENSHOTS**

RAW\_DATA.XLSV FILE

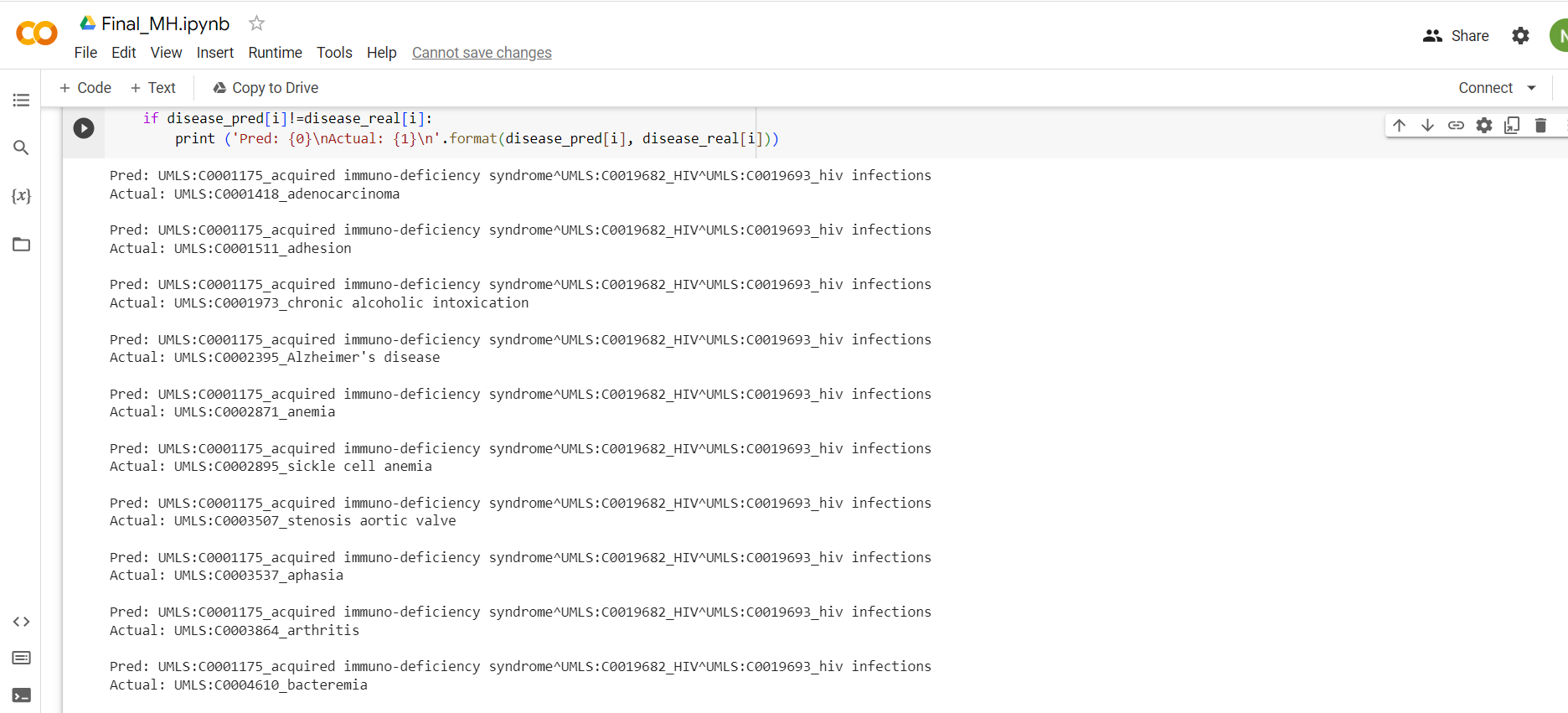




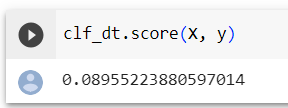
TREE IMAGES



PREDICTION



TIME TAKEN



THE END