Machine Learning Mini Project Report:

Submitted in partial fulfillment of the requirements

of the degree

BACHELOR OF ENGINEERING IN **COMPUTER ENGINEERING**

By

Student Name

1.Aaryan Manawat - 37

2.Abhiman Rajput - 51

3.Hitesh Ramrakhayni - 52

4.Nikhil Haswani - 21

Supervisor

Prof. GEOCEY SHEJY

Department of Computer Engineering

Vivekanand Education Society's Institute of Technology

HAMC, Collector's Colony, Chembur,

Mumbai-400074

University of Mumbai

(AY 2021-22)

CERTIFICATE

This is to certify that the Mini Project entitled **" CVS ANALYZER "** is a bonafide work of **Aaryan Manawat(37), Abhiman Singh(51), Hitesh Ramrakhyani(52) and Nikhil Haswani(21)** submitted to the University of Mumbai in partial fulfillment of the requirement for the award of the degree of **"Bachelor of Engineering"** in **"Computer Engineering" .**

(Prof. Geocey Shejy**)**
Supervisor

(Prof._____)
Head of Department

(Prof._____)
Principa

Mini Project Approval

This Mini Project entitled **"CVS ANALYZER "** by **Aaryan Manawat(37), Abhiman Singh(51), Hitesh Ramrakhyani(52) and Nikhil Haswani(21)** is approved for the degree of **Bachelor of Engineering** in **Computer Engineering.**

Examiners

1……
(Internal Examiner Name & Sign)

2……
(External Examiner name & Sign)

Date:

Place

Contents

Abstract:

Our project identifies CVS (Computer Vision Syndrome) amongst people with high amounts of screen time on a day to day basis like software engineers and recommending treatments and cures to prevent CVS.

From our research, we have found that there are very few software or applications or even data and papers based on CVS analysis amongst software engineers, who spend hours in front of the screen on a day to day basis and hence, need a solution so that they can take better care of their eyes

Acknowledgments:

We would like to express our gratitude towards the head of the Computer Department Dr (Mrs) Nupur Giri and our mentor Prof Geocey Shejy of the Computer Department for providing support and guidance. We got to learn a lot more about this project which will be very helpful for us .

List of figures:

List of tables:

Introduction:

“CVS Analyzer” identifies CVS(Computer Vision Syndrome) amongst people who face high amounts of screen time on a day to day basis like software engineers and recommends treatments and cures to prevent CVS and take better care of their eyes.

Motivation:

As computers and mobile phones become part of our everyday lives, more and more people are experiencing a variety of ocular symptoms related to computer use.

These include eyestrain, tired eyes, irritation, redness, blurred vision and double vision collectively referred to as Computer Vision Syndrome (CVS).

Therefore, there is a need for a "**CVS Analyzer**" especially for individuals like software engineers who spend hours in front of a screen daily, to maintain the good health and well-being of the individuals in the society.

Problem statement and Objective:

With everything transitioning online especially due to the pandemic and rise of metaverse,cryptocurrency,nfts,etc. which only suggests that the future is heading towards the virtual space and hence, computers and mobile phones have become part of our everyday lives therefore,more and more people are experiencing a variety of ocular symptoms related to computer use.,which include eyestrain, tired eyes, irritation, redness, blurred vision and double vision collectively referred to as Computer Vision Syndrome (CVS).

The objective of our project is to identify CVS (Computer Vision Syndrome) amongst people with high amounts of screen time on a day to day basis like software engineers and recommending treatments and cures to prevent CVS and take better care of their eyes.

Organization of report:

This report consists of a literature survey and proposed system.

The literature survey includes all the research papers that we have gone through to learn and understand more about the situation and our topic and also to extract data to work on to develop our software to predict CVS.

The proposed system consists of the algorithms,design,data preparation,framework, experiment and its conclusion and next work plan.

Literature Survey:

1.Survey of Existing System:

We have studied about CVS and extracted data from the following papers mentioned in the below:

Sr.no.	Title of the Paper	Author Name	Year of Publication
1	CVS in Health Personnel for Software Engineers	Eva Maria Artime Rios,Fernando Sanchez Lasheras	2019
2	CVS among Computer Office Workers	Y.S.Perera,S.Kulatunga	2016
3	CVS among Computer Based Employees	Sultan H.Al. Rashidi,H.Alhumaidan	2017
4	A Study of Computer Vision Syndrome at the Workplace - Prevalence and Causative Factors	Soumya Harapanahalli Venkatesh,Anita T Girish, Shashikala,Praveen Kulkarni,Snigdha Mannava, Rajendra Rajarathnam	2016

2.Limitation of existing system or research gap:

➔ Not much survey has been done on Computer Vision Syndrome as of now.

➔ The data collected is not much to detect and completely cure the occupational hazards related to Software Engineers.

➔ The research done related to this syndrome is much lesser as compared to others.

3.Mini project contribution:

With everything moving towards the virtual space as seen in the past couple of pandemic hit years, a software like "**CVS Analyzer**" especially for individuals like software engineers who spend hours in front of a screen daily, will help maintain the good eye health and well-being of the individuals in the society.

Proposed System:

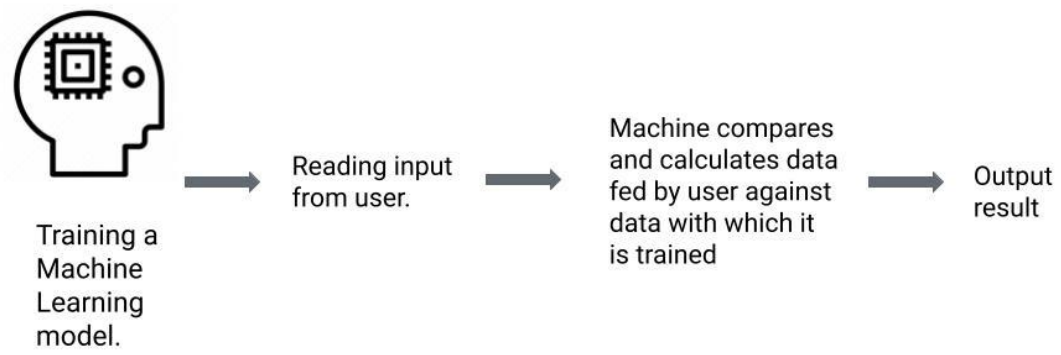
1.Introduction:

➔ The system is going to analyse Computer Vision Syndrome on people working continuously in front of Computer screen like Software Engineers.

➔ The system is going to collect the data which we can call symptoms during or immediately following Computer work.

➔ The proposed system will detect the Occupational Hazards during the process

2.Architecture/Framework:



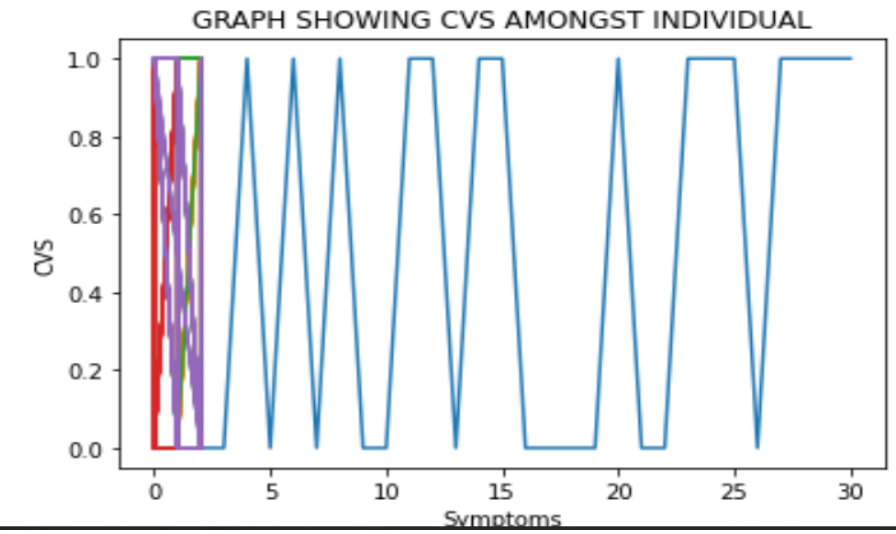
3.Algorithm and Process design:

First we will be constructing the data and then transforming it to train our Machine Learning model with it,for data preprocessing and filtering we have used two algorithms- Logistic Regression and Naive Bayes.

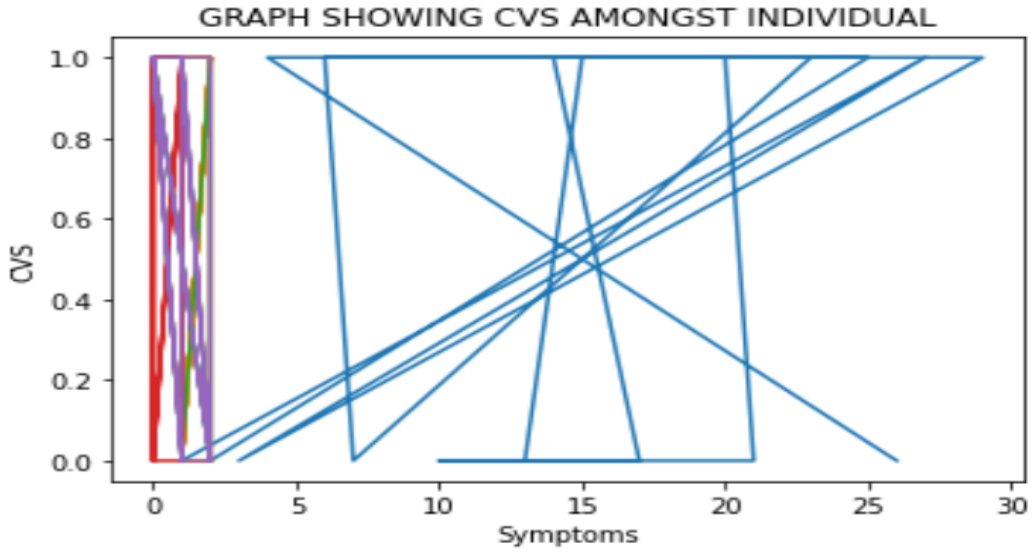
We have build the following classifier to preprocess and filter our data:

```
#Importing Libraries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
#Importing Dataset
training_data =
pd.read_csv(r"C:\Users\dhruv\AppData\Roaming\Microsoft\Win
dows\Start Menu\Programs\Anaconda3 (64-bit)\Dataset2.csv")

training_data

#Assigning numbers to word input in Gender column for
betterment analysis.
```

```
training_data['Gender'].unique()
training_data = training_data.replace(to_replace='m',
value=1)
training_data = training_data.replace(to_replace='f',
value=2)
training_data

#Assigning numbers to word input in Eye Strain column for
betterment analysis.
training_data['Eye Strain'].unique()
training_data = training_data.replace(to_replace='high',
value=2)
training_data = training_data.replace(to_replace='medium',
value=1)
training_data = training_data.replace(to_replace='low',
value=0)
training_data

#Assigning numbers to word input in Anti-Glare Filter
column for betterment analysis.
training_data['Anti-Glare Filter'].unique()
training_data = training_data.replace(to_replace='yes',
value=0)
training_data = training_data.replace(to_replace='no',
value=1)
training_data

#training_data = training_data.drop(["Sr. No."], axis=1)

 # plotting the points
plt.plot(X, y)

# naming the x axis
plt.xlabel('Symptoms')
# naming the y axis
plt.ylabel('CVS')

# giving a title to my graph
plt.title('GRAPH SHOWING CVS AMONGST INDIVIDUAL')

# function to show the plot
plt.show()
#overall data graph
```




```
 # plotting the points
plt.plot(X_train, y_train)

# naming the x axis
plt.xlabel('Symptoms')
# naming the y axis
plt.ylabel('CVS')

# giving a title to my graph
plt.title('GRAPH SHOWING CVS AMONGST INDIVIDUAL')

# function to show the plot
plt.show()
#training data graph
```



```
  #splitting training and test data
from sklearn.model_selection import train_test_split
X = training_data.drop(["cvs"], axis=1)
y = training_data["cvs"]
X_train, X_test, y_train, y_test = train_test_split(X, y,
test_size=0.33)

#X = training_data.copy()
#X = X.drop(["cvs"], axis=1)
#Y = training_data["cvs"]

#Applying Logistic Regression to the dataset
from sklearn.linear_model import LogisticRegression
clf = LogisticRegression()
clf.fit(X_train,y_train)

#test_X = test_data.copy()
#test_X = test_X.drop(["cvs"], axis=1)
#test_Y = test_data["cvs"]

clf.predict(X_test)

OUTPUT:   array([0, 0, 0, 0, 0, 1, 1, 1, 1, 1], dtype=int64)

y_test

#Finding the accuracy of the model
print("Accuracy : {}".format(clf.score(X_test,y_test)))

OUTPUT: Accuracy : 0.5

#Applying Gaussian Naive Bayes to our dataset
from sklearn.naive_bayes import GaussianNB
nb = GaussianNB()
nb.fit(X_train,y_train)
#Predicting the output
y_pred=nb.predict(X_test)

y_pred

OUTPUT: array([0, 0, 0, 0, 0, 1, 1, 1, 1, 1], dtype=int64)

y_test
```

```
#Finding the accuracy of the model
print("Accuracy : {}".format(nb.score(X_test,y_test)))
```

```
OUTPUT: Accuracy : 0.5
```

3.Details of hardware and software:

Technologies to be used-

- Python
- R
- Android
- Tensorflow
- Matplotlib
- Numpy libraries
- Pandas

Tools to be used-

- Nvidia GPU
- Google Colaboratory
- MySQL
- Firebase
- Jupyter Notebooks

4.Experiment and Results:

We first built our dataset from the research paper, then using jupyter notebooks, we imported the dataset into it and then split the dataset into training dataset and test data set.

We then ran our data through two algorithms, Logistic Regression and Naive Bayes, which helped us compare the algorithms and also whether our data is skewed, thus helping us to provide better accuracy.

Since both algorithms cannot evaluate string based data we had to first replace the string values to numeric values

We have also plotted our overall data and training data, which tells us that our data is relatively small and skewed, and therefore we need to improve our data's quantity and also include more number of symptoms to obtain better accuracy and results.

5.Conclusion and Future Work:

Software engineers spend hours in front of a screen on a daily basis, hence to maintain good eye health, diagnosis of CVS are required. Our project solves this problem by analyzing CVS amongst software engineers.

By using tools and technology like Machine Learning we will build an app that diagnoses CVS in software engineers and recommends cures and treatments and also recommends tips from time to time to maintain good eye health.

- ➔ Implementing image processing to diagnose CVS by scanning the user's eyes in real time.
- ➔ Implementing other health hazards like body posture for software engineers.
- ➔ Adding health hazards for other professions in the system, which may also be very useful for insurance companies to design packages specifically for a profession, so that more allowance can be given towards health hazards which they may face at their job.

References:

- ➔ https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5669505/
- ➔ https://www.researchgate.net/publication/333978330_Prediction_of_Computer_Vision_Syndrome_in_Health_Personnel_by_Means_of_Genetic_Algorithms_and_Binary_Regression_Trees
- ➔ https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4784392/
- ➔ https://www.researchgate.net/publication/337621491_Prevalence_of_Computer_Vision_Syndrome_CVS_Symptoms_and_Its_Awareness_Among_Software_Engineering_Students_of_Twin_Cities
- ➔ https://www.analyticsvidhya.com/blog/2021/01/a-guide-to-the-naive-bayes-algorithm/