EMOTION STATE ANALYSIS OF AN INDIVIDUAL BASED ON HIS HANDWRITING AND DRAWING

A PROJECT REPORT FOR J COMPONENT

MACHINE LEARNING

(CSE4020)

Submitted to

PROF. SHARMILA BANU K

Submitted by

Swattik Maiti: 18BCE0995

Palak Kishore: 18BCE2312

School of Computer Science and Engineering

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

May, 2021

INTRODUCTION

Health care mainly relies on the early detection of illnesses, and clinical tests have been developed to diagnose diseases and follow their evolution. Among tests, those based on human activities such as speech, handwriting and body movements have the advantage of being non-invasive and are valuable tools for complementing clinical examination and laboratory analyses. In particular, simple pen and paper tests can detect cognitive impairments through handwriting: lack of legibility, jagging and perseveration of letters are well-known effects of Alzheimer (AD) and Parkinson (PD) diseases.

The importance of detecting early signs of illnesses can be extended to the detection of negative emotions since emotions such as depression, anxiety and stress influence health. In the present study we focus on stress. The most comprehensive definition of stress is “a negative emotional experience accompanied by predictable biochemical, physiological and behavioural changes”. The causes of stress are extremely diverse, ranging from difficulties to handle everyday experiences and changes to traumatic events such as surviving to a natural disaster. Stress is one of the natural responses to changes and challenges of everyday life, however, when persisting over a long time period, they may contribute to serious health problems, such as heart disease, high blood pressure, diabetes, etc.

There exists multiple negative emotions such as depression, anxiety and stress, which are seen as distinct states but have largely overlapping clinical symptoms. In the present study we propose to detect stress through handwriting as it is a daily human activity. Our approach consists of using a dataset which was created by collecting individual’s handwriting through a computerized platform and we will be predicting their emotional state through a machine-learning approach. The database used in this paper is called ”EMOTHAW” (Emotion recognition from handwriting and drawing) which includes samples of participants whose emotional states are assessed by the Depression Anxiety Stress Scales (DASS) questionnaire.

Random forests or random decision forests are an ensemble learning method for classification, regression and other tasks that operates by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean/average prediction (regression) of the individual trees. Random decision forests correct for decision trees' habit of overfitting to their training set.

A multilayer perceptron (MLP) is a class of feedforward artificial neural network (ANN). An MLP consists of at least three layers of nodes: an input layer, a hidden layer and an output layer. Except for the input nodes, each node is a neuron that uses a nonlinear activation function. MLP utilizes a supervised learning technique called backpropagation for training. Its multiple layers and non-linear activation distinguish MLP from a linear perceptron. It can distinguish data that is not linearly separable.

LITERATURE SURVEY

S.NO.	Author	Title	Abstract	Methodology
1.	Leimapokpam Dorendro Singh, Mutum Malemnganba, Md Ameer Humjah	Psychological analysis based on handwriting pattern with machine learning (SVM)	The objective of this project is to develop a system that takes an image document containing the handwriting of a person and output a few of his/her personality traits based on some selected handwriting features. Carefully analyzing all the significant characteristics of a handwriting manually is not only time consuming but prone to errors as well. Automating the analysis on a few selected characteristics of handwriting will speed up the process and reduce the errors.	The proposed methodology focuses on developing a system that can predict the personality traits with the aid of machine learning without human intervention. To make this happen, they consider seven handwriting features: (i) size of letters, (ii) slant of the writing, (iii) baseline, (iv) pen pressure, (v) spacing between letters, (vi) spacing between words and (vii) top margin in a document to predict eight personality traits of a writer. After extracting all these features from the image containing the handwriting, eight support vector machines are trained which output each personality trait of the writer.
2.	Jesus Alberto Martinez Mendoza	Handwriting Analysis using Graphology & Machine Learning	A person's handwriting is as unique as their personality, which makes it tempting to connect the two. Graphology is the analysis of the physical characteristics and patterns of handwriting claiming to be able to identify the writer, indicating the psychological state at the time of writing, or evaluating personality characteristics.	The algorithm applied is Convolutional Neural Network (ConvNet/CNN). In this algorithm, features are extracted in the convolutional layers, where a kernel is passed over the image to extract a certain feature. In the end result, multiple kernels learn all the features within a dataset, in order to make classifications. This solves the issue of feature extraction in OCR methods.
3.	Sol Simpson	OpenHandWrite-Python tools for the recording and analysis of handwriting captured via pen tablet type devices.	OpenHandWrite is a suite of programs designed to provide behavioural scientists with tools for capturing and analysing pen movement. These programs provide accurately timed capture of pen movement data from digitising tablets and tabletPCs, and a markup and analysis tool that allows users to manually segment the pen trace into meaningful units (sentences, words, syllables, letters, lines, strokes) and then computes by-segment summary statistics.	Pen sample timing uses a parallel event-handeling technology (ioHub) that avoids quantising issues associated with windows swap and USB polling, therefore giving very accurate sample timing and more or less no sample skipping. Integration into an existing flexible and fully-featured experiment development environment ([PsychoPy])
4.	Sagarika Ganguly	Handwriting Analysis for	The project has been created with a GUI handle ,	Used SVM classifier to train a supervised learning model.

		Personality Trait Detection	where you can do the following : Browse the image of the handwriting you would like to detect the personality trait of. Perform Thresholding upon the browsed image (so that the image gets converted into a binary matrix where 1 is white and 0 is black) Feature Extraction button to view the features extracted , in the command window And finally , in the Classify button , you can view the results.	Analysis done with the help of HVM graphs built on Matlab.
5.	Deepak Kumar Jha, Sankalp Verma, Swattik Maiti, Shubhankar Shankar	Happy Hours - Employee Burn Rate Predictor	Happy Hours is an online web application that can manage the well-being status of the employees in a company through an interactive dashboard provided to the Human Resources department and also the Upper Management of the company. The web app can clearly demonstrate the status of the employees working there and how are they feeling with the amount of work put on them. In addition to this, the web app can identify a person's personality over 4 different axes by analyzing their publicly available social media posts. This can provide a deeper insight as to how well will they be able to cope with pressure.	This project implements 2 main models. 1) One Machine Learing model to predict the burnout rate of an employee given the basic information of the employee from a HR database. The Model implemented is a Linear Regression Model. 2) Our application also uses a Deep Learning model to perdict a person's personality. Personality is divided into 4 different axes of perosanilities. We have made 4 models with 4 different weights and biases for each of the personality traits using posts made by the user.

GAP IDENTIFIED

The topic of emotion analysis based on handwriting and drawing patterns is a fairly less explored topic. Hence any work towards this topic is unique. Previous papers have tried different datasets to create models to analyse this topic. Here we use EMOTHAW which is arguably the latest and best database drawing a relation between handwriting patterns and emotional state of individuals. In our project we try to draw out features that contribute to the mental stress of a person by utilizing EMOTHAW and come up with two different models to analyse the same. One of these is a neural network based model which shows promising results.

DATA COLLECTION

In this project, we have used EMOTHAW, the first publicly available database which relates emotional states to handwriting and drawing. We emailed the author of the paper "EMOTHAW: A Novel Database for Emotional State Recognition from Handwriting and Drawing" who granted us the permission to use the said dataset for non-commercial purposes only. The database we received contained samples of 129 participants whose emotional states, namely anxiety, depression, and stress, are assessed by the DASS questionnaire.

Seven tasks were recorded using INTUOS WACOM series 4 digitizing tablets and a special writing device named Intuos Inkpen, the tasks included drawing two pentagons and a house, words copied in handprint, drawing a clock and circles with right and left hand, and lastly, one sentence copied in cursive writing. Records consist in pen positions i.e. x-position and y-position, pen status i.e. on-paper and in-air, time stamp, pressure, pen azimuth, and altitude.

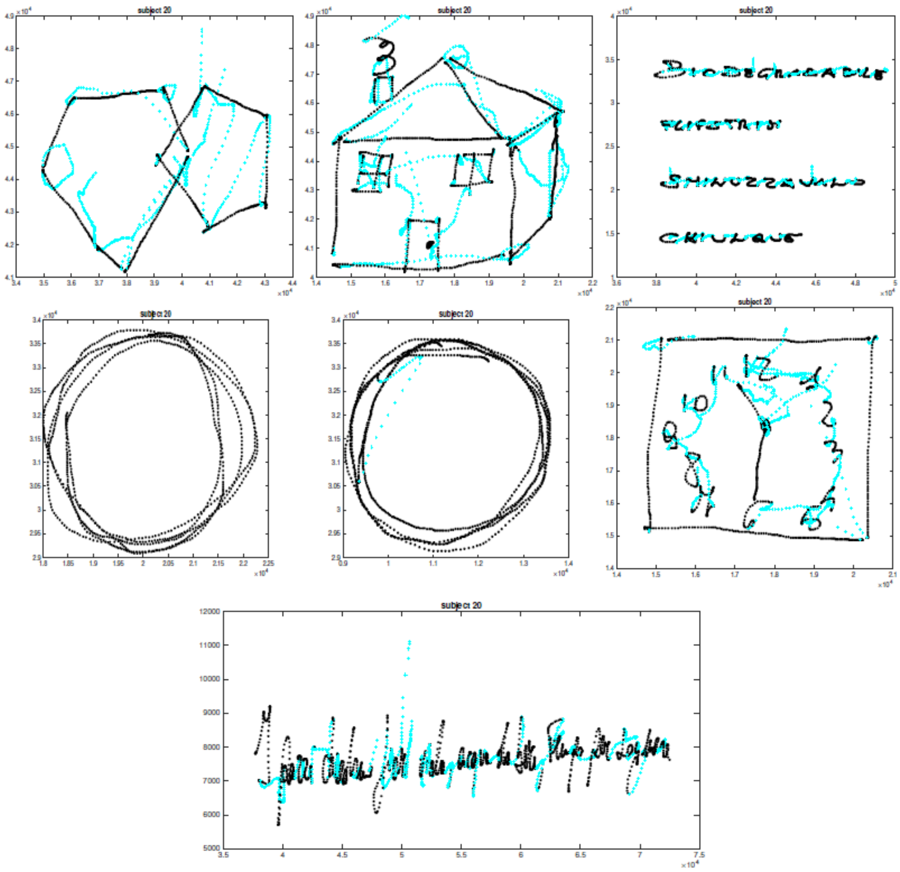


Fig: Writing and drawing samples collected from all tasks. Pen-down and pen-up data points are in black and blue, respectively.

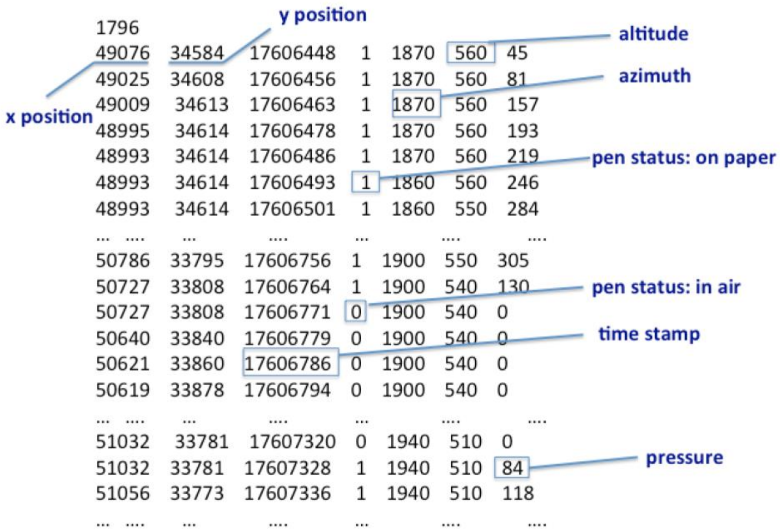


Fig: Extract of an svc file corresponding to the pentagon drawing task.

Using EMOTHAW database, we created a new dataset which contained the following features

1. on-paper duration (pentagon)
2. on-paper duration (house)
3. total duration (clock)
4. total duration (cursive)
5. number of pen down strokes (cursive)
6. average pressure (cursive)

A drawing or writing task consists of a sequence of strokes which can be denoted by $\{s^{(1)}, s^{(2)}, \ldots s^{(2K+1)}\}$. In this context, an on-paper stroke corresponds to consecutive drawing points achieved without lifting the pen i.e. $\{s^{(2k+1)}\}_{k=0,\ldots,K}$ is the set of on-paper strokes and K+1 is the number of on-paper strokes, an on-paper duration corresponds to the total time spent on paper while completing a task i.e. $\sum_{k=0}^{K} d(2k+1)$ and total duration corresponds to total time spent while completing a particular task i.e. $t^{(2K+1)}_{|s(2K+1)|} - t^{(1)}_1$ where t=time stamps and $d_i = t^{(i)}_{|s(i)|} - t^{(i)}_1$.

	A	B	C	D	E	F	G	H
1	On Paper Duration (pentagons)	On paper Duration (house)	total duration (clock)	total duration (cursive)	number of pen-down strokes (cursive)	Pressure average (cursive)	DASS Scores	Stress
2	7107	11023	21398	23531	33	297.58727	13	Normal
3	10152	14305	39205	26279	42	348.658005	21	Moderate
4	7350	10920	22215	25618	39	97.101751	17	Moderate
5	11236	9118	44864	23274	34	431.931921	17	Moderate
6	13971	19535	69047	30511	45	415.328375	27	Severe
7	9787	21296	36034	31219	41	229.998456	25	Moderate
8	15017	16775	34059	27007	44	299.951981	32	Severe
9	8485	21084	31387	27031	38	146.31625	35	Severe
10	9240	12393	24701	24720	42	283.69945	25	Moderate
11	3872	7800	58595	30735	52	230.202721	15	Normal
12								

Fig: Extracted Dataset

Our target variable is Stress which can be normal, moderate or severe depending on the DASS scores.

STRESS	DASS SCORE
Normal	0 - 16
Moderate	17 - 25
Severe	25+

TABLE: DASS score range according to emotional state level.

IMPLEMENTATION PROCESS

1. Importing the libraries

jupyter Emothaw Last Checkpoint: Last Friday at 00:22 (autosaved) Logout

File Edit View Insert Cell Kernel Widgets Help Trusted Python 3

Run Code

In [1]:
```
#import libraries
import pandas as pd
import numpy as np
from sklearn import preprocessing
import seaborn as sb
import matplotlib.pyplot as plt
```

```
C:\Users\Swattik\AppData\Local\Continuum\anaconda3\lib\site-packages\statsmodels\tools\_testing.py:19: FutureWarning: pandas.ut
il.testing is deprecated. Use the functions in the public API at pandas.testing instead.
  import pandas.util.testing as tm
```

2. Loading the dataset

In [97]:
```
data = pd.read_csv("final_ds2.csv")
data
```

Out[97]:

	Subject	On Paper Duration (pentagons)	On paper Duration (house)	total duration (clock)	total duration (cursive)	number of pen-down strokes (cursive)	Pressure average (cursive)	DASS Scores	Stress
0	1	7107	11023	21398	23531	33	297.587270	13	Normal
1	3	10152	14305	39205	26279	42	348.658005	21	Moderate
2	4	7350	10920	22215	25618	39	97.101751	17	Moderate
3	6	11236	9118	44864	23274	34	431.931921	17	Moderate
4	33	13971	19535	69047	30511	45	415.328375	27	Severe
5	34	9787	21296	36034	31219	41	229.998456	25	Moderate
6	47	15017	16775	34059	27007	44	299.951981	32	Severe
7	81	8485	21084	31387	27031	38	146.316250	35	Severe
8	93	9240	12393	24701	24720	42	283.699450	25	Moderate
9	125	3872	7800	58595	30735	52	230.202721	15	Normal

In [98]:
```
data = data.drop('Subject',axis = 1)
data = data.drop('Pressure average (cursive)',axis = 1)
```

3. Feature Engineering

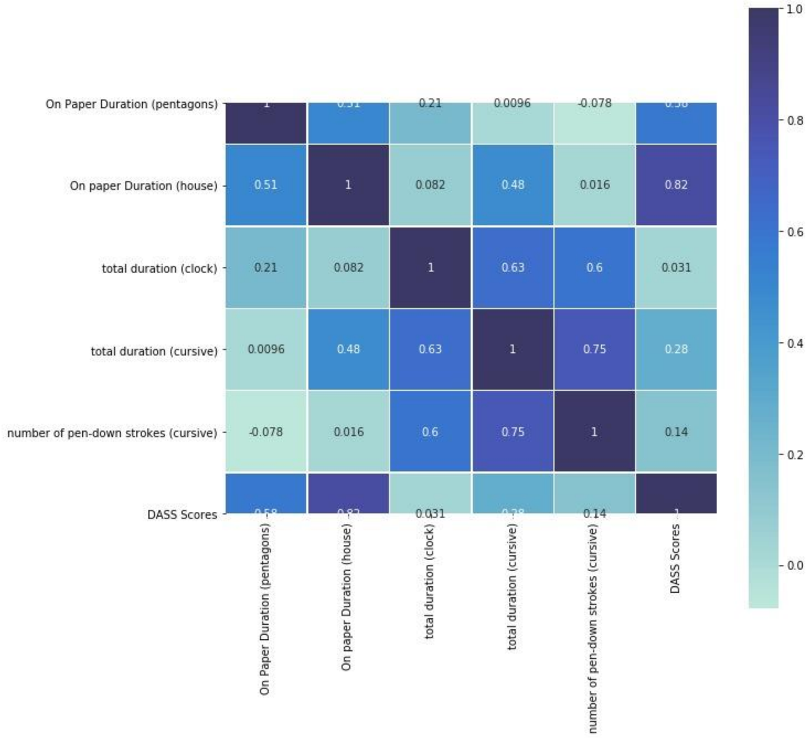
In [99]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10 entries, 0 to 9
Data columns (total 7 columns):
 #   Column                                Non-Null Count  Dtype 
---  ------                                --------------  ----- 
 0   On Paper Duration (pentagons)         10 non-null     int64 
 1   On paper Duration (house)             10 non-null     int64 
 2   total duration (clock)                10 non-null     int64 
 3   total duration (cursive)              10 non-null     int64 
 4   number of pen-down strokes (cursive)  10 non-null     int64 
 5   DASS Scores                           10 non-null     int64 
 6   Stress                                10 non-null     object
dtypes: int64(6), object(1)
memory usage: 688.0+ bytes
```

4. Plotting the Heatmap

```
In [100]: corr = data.corr()
          f,axes = plt.subplots(1,1,figsize = (10,10))
          sb.heatmap(corr,square=True,annot = True,linewidth = .5,center = 1.4,ax = axes)

Out[100]: <matplotlib.axes._subplots.AxesSubplot at 0x1d26cce64c8>
```



5. Label Encoding the target variable

```
In [102]: #Label Encoding
          data['Stress'] = data['Stress'].replace({'Normal':'1','Moderate':'2', 'Severe':'3'})
          data

Out[102]:
```

	On Paper Duration (pentagons)	On paper Duration (house)	total duration (clock)	total duration (cursive)	number of pen-down strokes (cursive)	Stress
0	7107	11023	21398	23531	33	1
1	10152	14305	39205	26279	42	2
2	7350	10920	22215	25618	39	2
3	11236	9118	44864	23274	34	2
4	13971	19535	69047	30511	45	3
5	9787	21296	36034	31219	41	2
6	15017	16775	34059	27007	44	3
7	8485	21084	31387	27031	38	3
8	9240	12393	24701	24720	42	2
9	3872	7800	58595	30735	52	1

6. Separating the target variable and feature matrix

In [8]:
```
y = data["Stress"]
y
```

Out[8]:
```
0    1
1    2
2    2
3    2
4    3
5    2
6    3
7    3
8    2
9    1
Name: Stress, dtype: object
```

In [9]:
```
X = data.drop('Stress',axis = 1)
X
```

Out[9]:

	On Paper Duration (pentagons)	On paper Duration (house)	total duration (clock)	total duration (cursive)	number of pen-down strokes (cursive)
0	7107	11023	21398	23531	33
1	10152	14305	39205	26279	42
2	7350	10920	22215	25618	39
3	11236	9118	44864	23274	34
4	13971	19535	69047	30511	45
5	9787	21296	36034	31219	41
6	15017	16775	34059	27007	44
7	8485	21084	31387	27031	38
8	9240	12393	24701	24720	42
9	3872	7800	58595	30735	52

7. Scaling the features

In [10]:
```
# Scaling
scaler = preprocessing.StandardScaler().fit(X)
X = scaler.transform(X)
```

In [11]:
```
X=pd.DataFrame(X)
X
```

Out[11]:

	0	1	2	3	4
0	-0.807749	-0.719998	-1.129147	-1.242422	-1.528321
1	0.170338	-0.025376	0.071075	-0.256094	0.191040
2	-0.729695	-0.741797	-1.074080	-0.493344	-0.382080
3	0.518531	-1.123183	0.452501	-1.334666	-1.337281
4	1.397043	1.081531	2.082476	1.262881	0.764161
5	0.053096	1.454239	-0.142656	1.517001	0.000000
6	1.733029	0.497389	-0.275774	0.005204	0.573121
7	-0.365120	1.409371	-0.455871	0.013819	-0.573121
8	-0.122606	-0.430043	-0.906519	-0.815659	0.191040
9	-1.846866	-1.402132	1.377994	1.343281	2.101442

8. Splitting into training and testing data

Model Training

```
In [87]: from sklearn.model_selection import train_test_split
         x_train,x_test,y_train,y_test=train_test_split(X,y,test_size=0.3)
```

9. Training the Random Forest Classifier

```
In [91]: #Import Random Forest Model
         from sklearn.ensemble import RandomForestClassifier

         #Create a Gaussian Classifier
         clf=RandomForestClassifier(n_estimators=100)


         clf.fit(x_train,y_train)

         y_pred_rfc=clf.predict(x_test)
```

10. Actual y-test values

```
In [88]: y_test

Out[88]: 4    3
         2    2
         3    2
         Name: Stress, dtype: object
```

11. Values predicted by RFC

```
In [92]: y_pred_rfc

Out[92]: array(['3', '1', '2'], dtype=object)

In [93]: from sklearn import metrics
         print("Accuracy using RFC:",metrics.accuracy_score(y_test, y_pred_rfc))

         Accuracy using RFC: 0.6666666666666666
```

12. Training the Neural Network

```
In [94]: from sklearn.neural_network import MLPClassifier

         mlp = MLPClassifier(hidden_layer_sizes=(10,10,10), activation='relu', solver='adam', max_iter=500)
         mlp.fit(x_train,y_train)

         y_pred_mlp=mlp.predict(x_test)
```

13. Values predicted by MLPC

```
In [95]: y_pred_mlp

Out[95]: array(['3', '2', '1'], dtype='<U1')

In [96]:
         print("Accuracy using MLP:",metrics.accuracy_score(y_test, y_pred_mlp))

         Accuracy using MLP: 0.6666666666666666
```

ERROR HANDLING

1. Our initial dataset included total duration (pentagons), total duration (house), total duration (clock), total duration (cursive), number of pen-down strokes (cursive) and pressure average (cursive). The initial target variable was stress i.e. the DASS score.

	Subject	total duration (pentagons)	total duration (house)	total duration (clock)	total duration (cursive)	number of pen-down strokes (cursive)	Pressure average (cursive)	Stress
0	1	11986	22182	21398	23531	33	297.587270	13
1	3	22915	48424	39205	26279	42	348.658005	3
2	4	12185	19275	22215	25618	39	97.101751	19
3	6	13273	23193	44864	23274	34	431.931921	17
4	33	20490	39464	69047	30511	45	415.328375	27
5	34	22589	31152	36034	31219	41	229.998456	9
6	47	15920	25795	34059	27007	44	299.951981	32
7	81	11088	39395	31387	27031	38	146.316250	35
8	93	11140	21315	24701	24720	42	283.699450	25
9	125	22547	29631	58595	30735	52	230.202721	15

Fig: Initial Dataset

Using the initial dataset, our initial accuracy turned out to be 0.

Model Training

```
In [12]: #Import Random Forest Model
         from sklearn.ensemble import RandomForestClassifier

         #Create a Gaussian Classifier
         clf=RandomForestClassifier(n_estimators=100)

         #Train the model using the training sets y_pred=clf.predict(X_test)
         clf.fit(x_train,y_train)

         y_pred=clf.predict(x_test)

In [13]: y_pred

Out[13]: array([ 9, 19, 25], dtype=int64)

In [14]: from sklearn import metrics
         print("Accuracy:",metrics.accuracy_score(y_test, y_pred))

         Accuracy: 0.0
```

Fig: Initial Accuracy

We identified the correlation between features using seaborn heatmap. We found that total duration (pentagon), total duration (house) and pressure (cursive) were having a negative correlation towards our target variable DASS score.

To solve this, we replaced total duration (pentagon) with on-paper duration (pentagon), total duration (house) with on-paper duration (house) and dropped pressure (cursive).

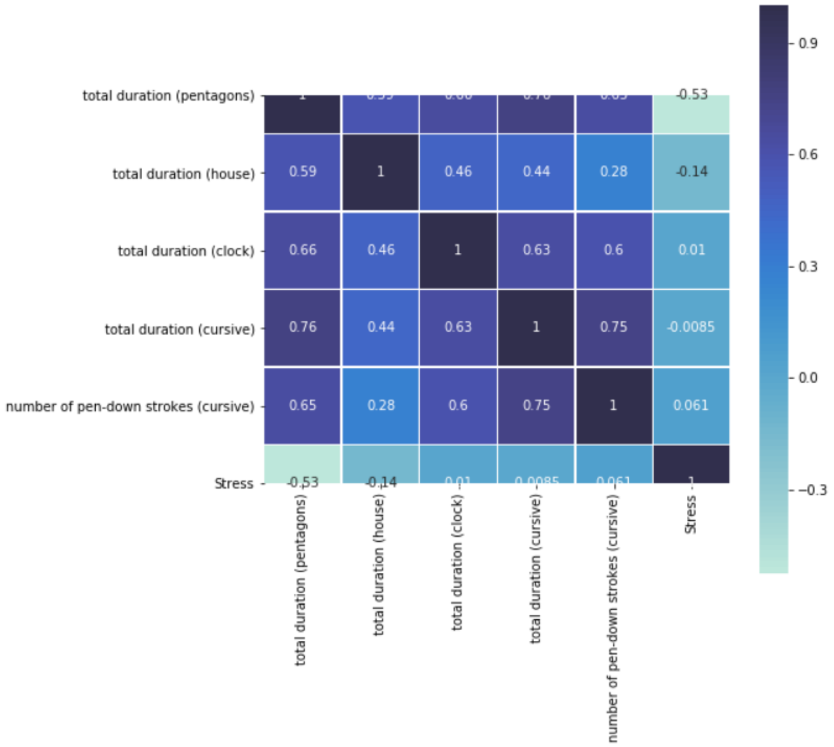


Fig: Initial Heatmap

2. Next, we identified that our accuracy was constantly low because our initial target variable was DASS score which is a numerical value. Since this is a classification problem, it would be very difficult for the model to predict for a range between 0 to 25+. In multiple cases our predicted value and actual value were very close but not equal, which is expected considering the size of our dataset.

   To solve this problem, we reduced the range of our target variable, Stress, to three categories, namely normal, moderate or severe based on the DASS score.

INDIVIDUAL CONTRIBUTION

1. Swattik Maiti
   Worked on calculating the values of individual features from the dataset and identifying relevant features. Also contributed in the feature engineering process and designing the neural network classification model. Helping out in the documentation process.

2. Palak Kishore
   Worked on calculating the values of individual features and compiling the final dataset to be used for the project. Major contributor in the error handling process. Training the random forest classifier model. Also worked on the Documentation.

CONCLUSION

The topic of emotional state analysis based on an individual's handwriting and drawing is a topic which is unique and fairly less researched compared to other, more mainstream subjects. In this project, we utilised the EMOTHAW database to a great extent, producing relevant features which, as proven, directly impacts the emotional stress experienced by a person. The two machine learning models that we created gave results which exceeded our expectations considering the severe lack of data.

The major constraint that we faced throughout the project was the cost of the process by which the features are calculated from the raw database. As we have done the majority of the process manually, we could only feasibly come up with a dataset with 10 samples, thus greatly impacting our scope to make a more robust machine learning model.

However, this research could be used as a stepping stone for further research on the topic. Automating the process of calculating each feature from the database is of primary priority for future research. With more data and more relevant features, a better neural network can be trained which will in turn create a more accurate model.

REFERENCE

[1] https://github.com/Malemm/ml-graphology
[2] https://github.com/jesusmartinoza/Graphology-Machine-Learning
[3] https://github.com/isolver/OpenHandWrite
[4] https://github.com/sagarikaganguly719/handwritingAnalysis
[5] https://github.com/PerciValXIII/SAP_Hack-Happy-Hours-Burn_Rate_Predictor

TEAM MEMBERS

Swattik Maiti
18BCE0995

Palak Kishore
18BCE2312

LINK TO GITHUB PROJECT: https://github.com/PerciValXIII/Emotion-state-analysis-of-an-individual-based-on-handwriting-and-drawing