Emotion Detection Using Keystroke Dynamics

1. Project Description (in brief):

The project focuses on **Predicting Emotions Using Keystroke Dynamics**. Keystroke dynamics refers to the unique typing patterns of individuals, which can be analyzed to infer emotional states. The goal is to develop a machine-learning model that can classify emotions based on keystroke timing data. The dataset consists of various keystroke features such as **latency, dwell time, and flight time**, which serve as inputs to the model.

2. Type of problem: (supervised\Unsupervised), (Prediction\Classification) Why the problem falls into particular category

- **Supervised Learning**

The problem falls into the category of **supervised learning** because we have labeled data, where each set of keystroke features corresponds to a known emotion.

The model is trained on these labeled samples to learn patterns and predict emotions for new unseen keystrokes.

- **Classification Problem**

The problem is a **classification task** since the goal is to categorize keystrokes into discrete emotion classes (e.g., Happy, Sad, Angry, Neutral, etc.).

If the target variable had been continuous (e.g., an emotion intensity score), it would be a regression problem, but since we are predicting **categorical emotions**, it is a classification problem.

3. Python Libraries used:

```
import numpy as np
import pandas as pd
import nltk
import seaborn as sn
import matplotlib.pyplot as plt
import scipy.stats as stats
```

4. **Data set details:** Describe data set in detail. Description must be in terms of number of rows, features, meaning of features, source of the data set and how old data set is

Participants Info

	userId	typeWith	typistType	pcTimeAverage	ageRange	gender	status	degree	country
0	100	2 hands	Touch Typist	More than 3 hours per day	16-19	Female	Student	College/University	Tunisia
1	113	2 hands	Two Finger Typist	less than an hour per day	20-29	Female	Student	College/University	Tunisia
2	58	2 hands	Touch Typist	More than 3 hours per day	20-29	Female	Student	College/University	Tunisia
3	63	2 hands	One Finger Typist	between 1 hour and 3 hours per day	20-29	Female	Student	College/University	Tunisia
4	84	1 hand	One Finger Typist	less than an hour per day	>=50	Male	Professional	College/University	Tunisia

```
(123, 9)
```

123 rows , 9 features

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 123 entries, 0 to 122
Data columns (total 9 columns):
 #   Column          Non-Null Count  Dtype
---  ------          --------------  -----
 0   userId          123 non-null    int64
 1   typeWith        123 non-null    object
 2   typistType      123 non-null    object
 3   pcTimeAverage   123 non-null    object
 4   ageRange        123 non-null    object
 5   gender          123 non-null    object
 6   status          123 non-null    object
 7   degree          123 non-null    object
 8   country         123 non-null    object
dtypes: int64(1), object(8)
memory usage: 8.8+ KB
```

```
[ ]  df1['userId'].nunique()
```

```
85
```

We have 85 users

	userId
count	123.000000
mean	104.951220
std	39.743203
min	13.000000
25%	76.500000
50%	106.000000
75%	130.500000
max	193.000000

Frequency Datset:

	User ID	textIndex	emotionIndex	delFreq	leftFreq	TotTime
0	100	FI	N	8	1	192762.0
1	100	FR	N	145	0	NaN
2	100	FR	H	0	0	NaN
3	100	FI	H	11	0	99463.0
4	113	FI	N	10	0	84265.0

```
(478, 6)
```

478 rows and 6 features

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 478 entries, 0 to 477
Data columns (total 6 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   User ID       478 non-null    int64
 1   textIndex     478 non-null    object
 2   emotionIndex  478 non-null    object
 3   delFreq       478 non-null    int64
 4   leftFreq      478 non-null    int64
 5   TotTime       243 non-null    float64
dtypes: float64(1), int64(3), object(2)
memory usage: 22.5+ KB
```

	User ID	delFreq	leftFreq	TotTime
count	478.000000	478.000000	478.000000	243.000000
mean	101.830544	9.688285	0.761506	84708.061728
std	37.599603	15.582912	3.453437	77239.131115
min	19.000000	0.000000	0.000000	16968.000000
25%	73.000000	2.000000	0.000000	51786.000000
50%	104.000000	6.000000	0.000000	64643.000000
75%	130.000000	11.000000	0.000000	91826.500000
max	193.000000	145.000000	32.000000	684946.000000

Fixed Text Typing Dataset:

	userId	emotionIndex	index	keyCode	keyDown	keyUp	D1U1	D1U2	D1D2	U1D2	U1U2	D1U3	D1D3	answer
0	100	N	3448	o	1,58E+12	1,58E+12	90	2556	2479.0	2389	2466	2737	2610.0	NaN
1	100	N	3449	n	1,58E+12	1,58E+12	77	258	131.0	54	181	744	650.0	NaN
2	100	N	3450	c	1,58E+12	1,58E+12	127	613	519.0	392	486	795	719.0	NaN
3	100	N	3451	e	1,58E+12	1,58E+12	94	276	200.0	106	182	2304	2232.0	NaN
4	100	N	3452		1,58E+12	1,58E+12	76	2104	2032.0	1956	2028	2271	2191.0	NaN

```
(46871, 14)
```

	userId	index	D1D2	D1D3
count	46871.000000	46871.00000	46606.000000	46364.000000
mean	101.721875	26883.00000	439.116981	877.612609
std	37.736388	13530.63657	3403.924378	4932.252363
min	13.000000	3448.00000	1.000000	4.000000
25%	73.000000	15165.50000	157.000000	366.000000
50%	104.000000	26883.00000	232.000000	531.000000
75%	130.000000	38600.50000	396.000000	860.000000
max	193.000000	50318.00000	461478.000000	463588.000000

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 46871 entries, 0 to 46870
Data columns (total 14 columns):
 #   Column        Non-Null Count  Dtype
---  ------        --------------  -----
 0   userId        46871 non-null  int64
 1   emotionIndex  46871 non-null  object
 2   index         46871 non-null  int64
 3   keyCode       46846 non-null  object
 4   keyDown       46846 non-null  object
 5   keyUp         45558 non-null  object
 6   D1U1          46846 non-null  object
 7   D1U2          46606 non-null  object
 8   D1D2          46606 non-null  float64
 9   U1D2          46606 non-null  object
 10  U1U2          46606 non-null  object
 11  D1U3          46364 non-null  object
 12  D1D3          46364 non-null  float64
 13  answer        18781 non-null  object
dtypes: float64(2), int64(2), object(10)
memory usage: 5.0+ MB
```

Free Text Typing:

	_id	userid	emotionIndex	index	keyCode	keyDown	keyUp	D1U1	D1U2	D1D2	U1D2	U1U2
0	5e1e2631d2dd163d472fd5ac	100	N	822	\u0011	1,57903E+12	1,57903E+12	982	764	709.0	-273	-218
1	5e1e2631d2dd163d472fd5ad	100	N	823	v	1,57903E+12	1,57903E+12	55	-1,57903E+12	23415.0	23360	-1,57903E+12
2	5e1e2631d2dd163d472fd5ae	100	N	824	\b	1,57903E+12	NaN	-1,57903E+12	-1,57903E+12	607.0	1,57903E+12	0
3	5e1e2631d2dd163d472fd5af	100	N	825	\b	1,57903E+12	NaN	-1,57903E+12	-1,57903E+12	35.0	1,57903E+12	0
4	5e1e2631d2dd163d472fd5b0	100	N	826	\b	1,57903E+12	NaN	-1,57903E+12	-1,57903E+12	42.0	1,57903E+12	0

```
(28412, 15)
```

	userid	index	D1D2	D1D3
count	28412.000000	28412.00000	28172.000000	27938.000000
mean	104.836689	15027.50000	405.406858	809.720739
std	36.931115	8201.98226	2106.174516	3102.161804
min	19.000000	822.00000	0.000000	8.000000
25%	77.000000	7924.75000	136.000000	315.000000
50%	109.000000	15027.50000	202.000000	460.000000
75%	130.000000	22130.25000	337.000000	734.000000
max	193.000000	29233.00000	207952.000000	222266.000000

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 28412 entries, 0 to 28411
Data columns (total 15 columns):
 #   Column        Non-Null Count  Dtype  
---  ------        --------------  -----  
 0   _id           28412 non-null  object 
 1   userid        28412 non-null  int64  
 2   emotionIndex  28412 non-null  object 
 3   index         28412 non-null  int64  
 4   keyCode       28405 non-null  object 
 5   keyDown       28405 non-null  object 
 6   keyUp         25863 non-null  object 
 7   D1U1          28405 non-null  object 
 8   D1U2          28172 non-null  object 
 9   D1D2          28172 non-null  float64
 10  U1D2          28172 non-null  object 
 11  U1U2          28172 non-null  object 
 12  D1U3          27938 non-null  object 
 13  D1D3          27938 non-null  float64
 14  answer        7660 non-null   object 
dtypes: float64(2), int64(2), object(11)
memory usage: 3.3+ MB
```

EmoSurv Dataset

Meaning of Features:

The **EmoSurv** dataset contains various keystroke dynamic features that are used to recognize emotional states based on how individuals type. The features captured in this dataset include:

1. **Dwell Time**:
   - The amount of time a user presses a key before releasing it. This reflects the user's level of engagement or emotional state (e.g., stress or excitement may cause changes in dwell time).
2. **Flight Time**:
   - The time between pressing one key and pressing the next key. It is an indicator of the user's typing speed and rhythm, which may vary depending on their emotional state.
3. **Typing Speed**:

    - The speed at which a person types. Emotional states such as happiness or frustration may influence the typing speed, with faster typing associated with excitement and slower typing linked to sadness or stress.
4. **Key Press Latency**:
    - The time it takes between the release of one key and the press of the next key. This feature can indicate a delay in response due to emotional states, such as hesitation or uncertainty during typing.

These features are extracted from the typing behavior of individuals during tasks that prompt emotional responses.

Source of the Dataset:

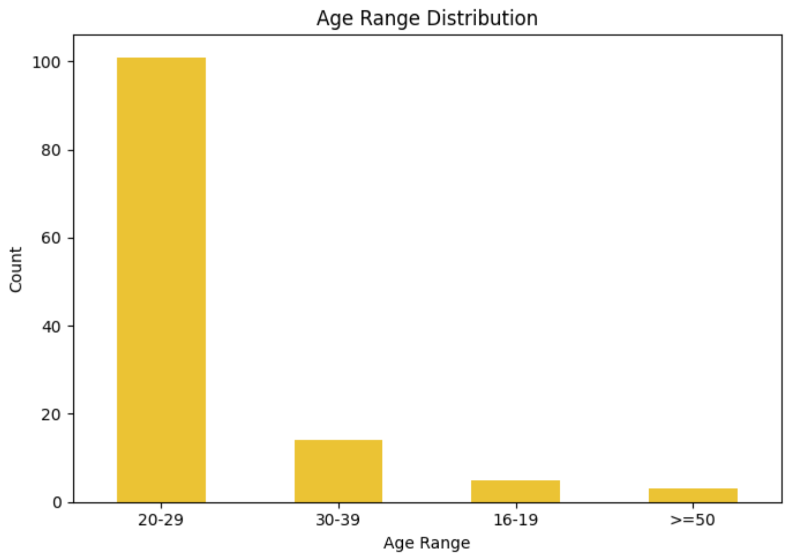
- The **EmoSurv** dataset was created as part of research in **emotion recognition through keystroke dynamics**. It is used to understand and classify emotional states based on typing patterns. The dataset is typically shared by the authors of the research studies and is used in academic papers and projects related to **human-computer interaction (HCI)**, **biometric authentication**, and **emotion recognition**.
- The dataset is publicly available for research purposes and has been used in several studies on emotional states, stress detection, and user interaction analysis. It is typically cited in emotion recognition studies.

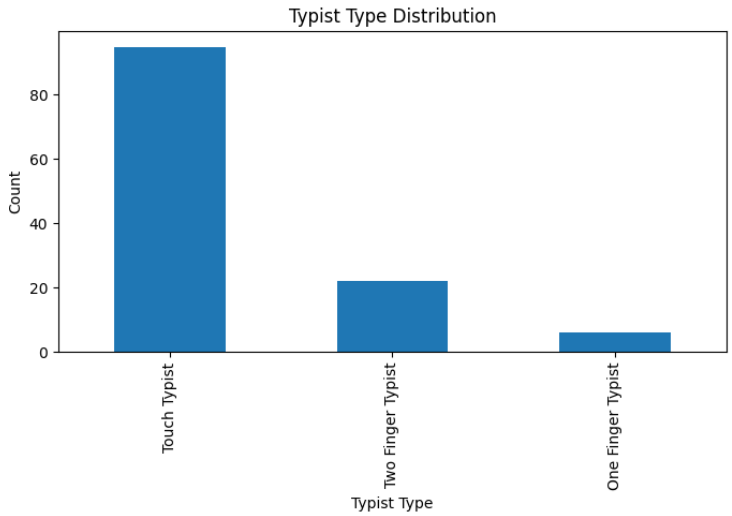
Age of the Dataset:

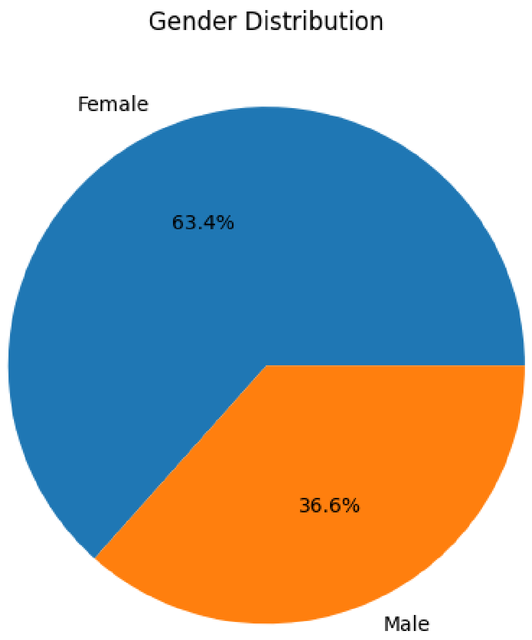
- The **EmoSurv** dataset was created and has been in use for over **5 to 10 years**. It was collected as part of a study into understanding how typing behavior can reveal emotional states, and it has become a standard dataset for emotion recognition research.

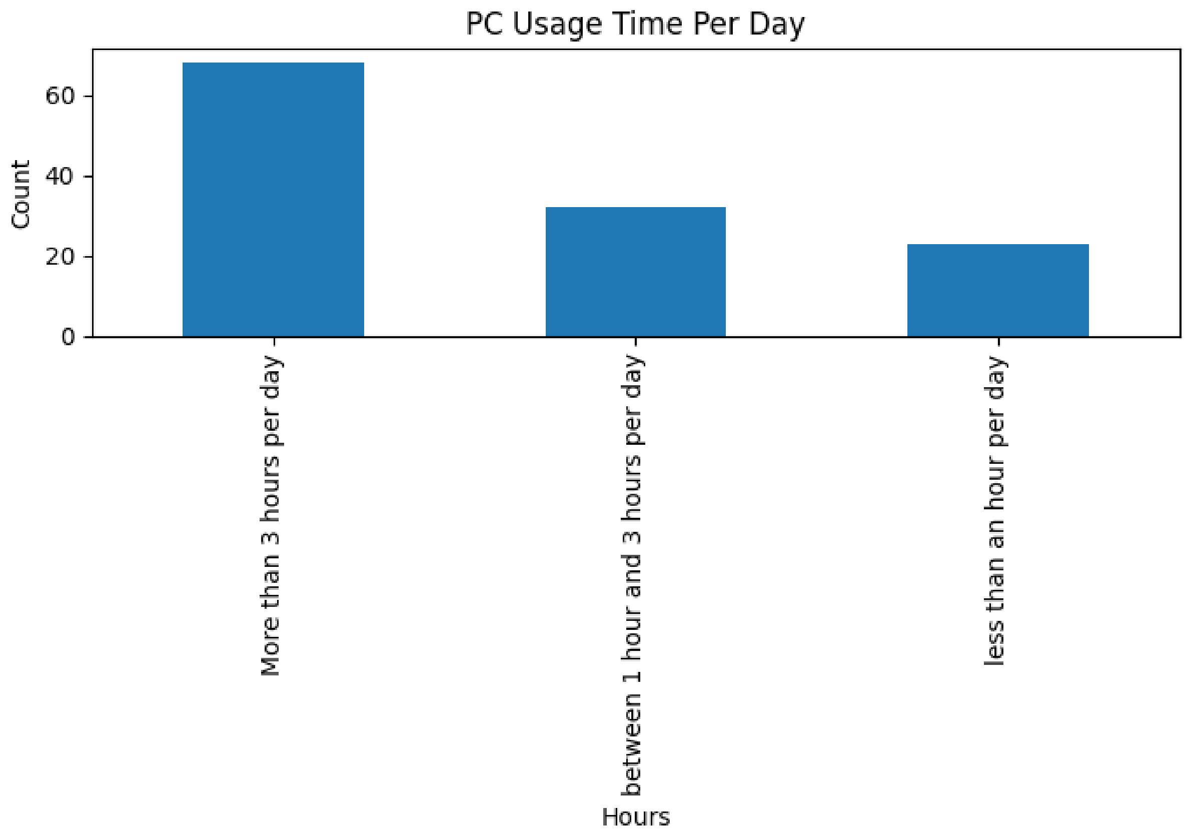
5. Data set visualization and inference

Participants Info




Typist Type Distribution
Count
Typist Type
Touch Typist
Two Finger Typist
One Finger Typist
0
20
40
60
80


Gender Distribution
Female
Male
63.4%
36.6%



Inference

1. Age Range Distribution:

Most participants are between 20-29 years, indicating the dataset skews younger.

2. Typist Type Distribution:

Touch Typists are most common, followed by Two-Finger and One-Finger typists.

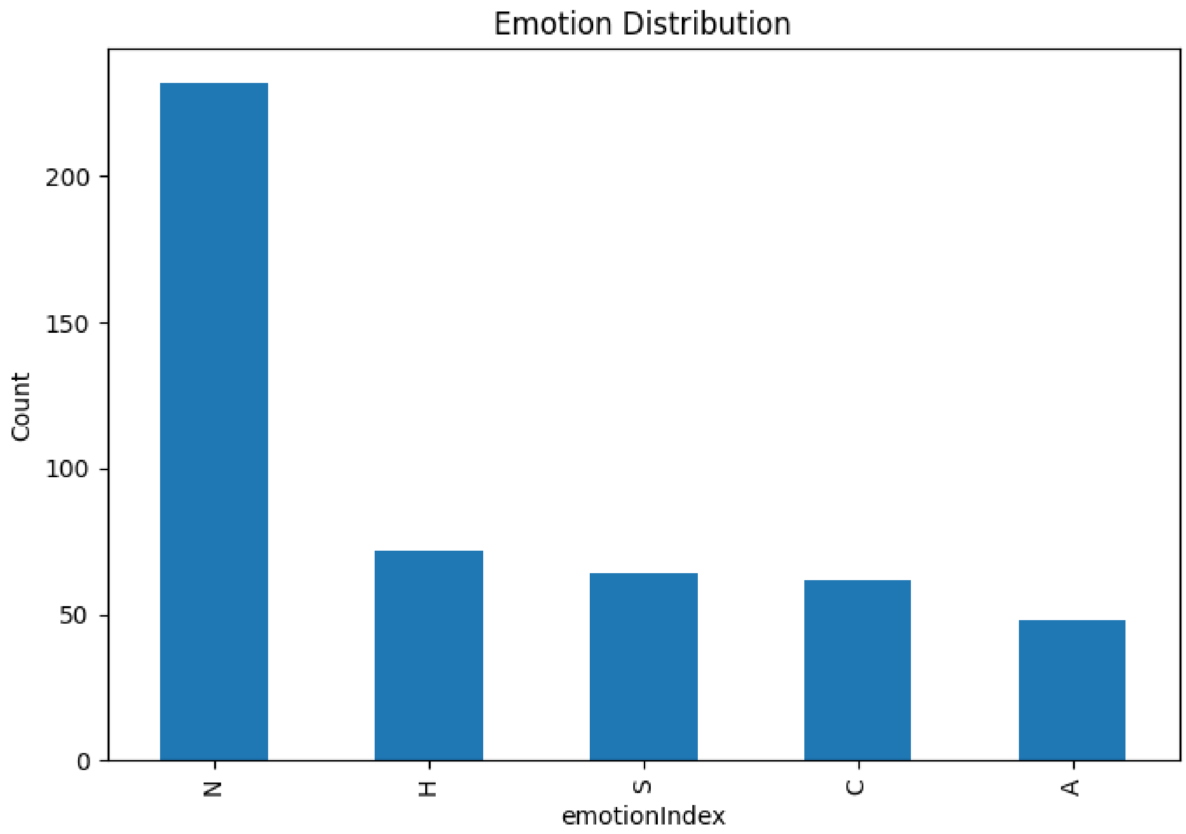
3. Gender Distribution:

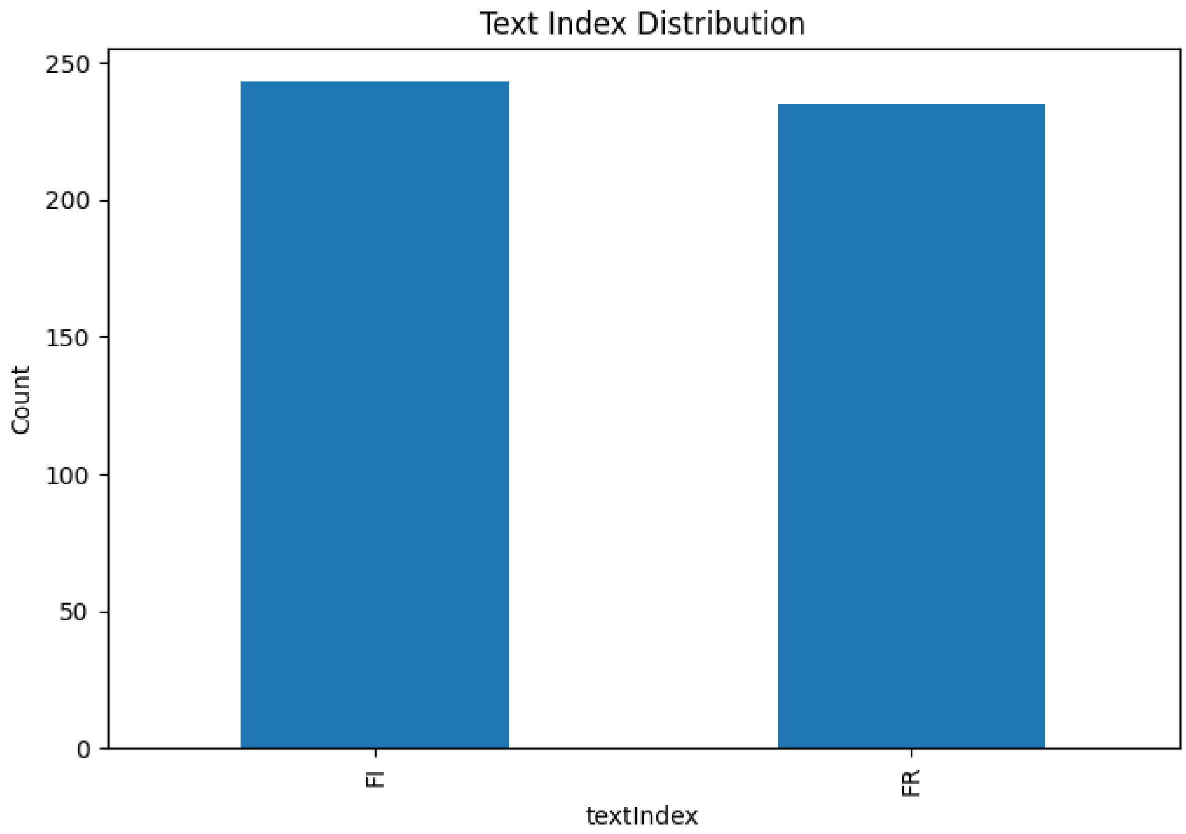
Majority are female participants.

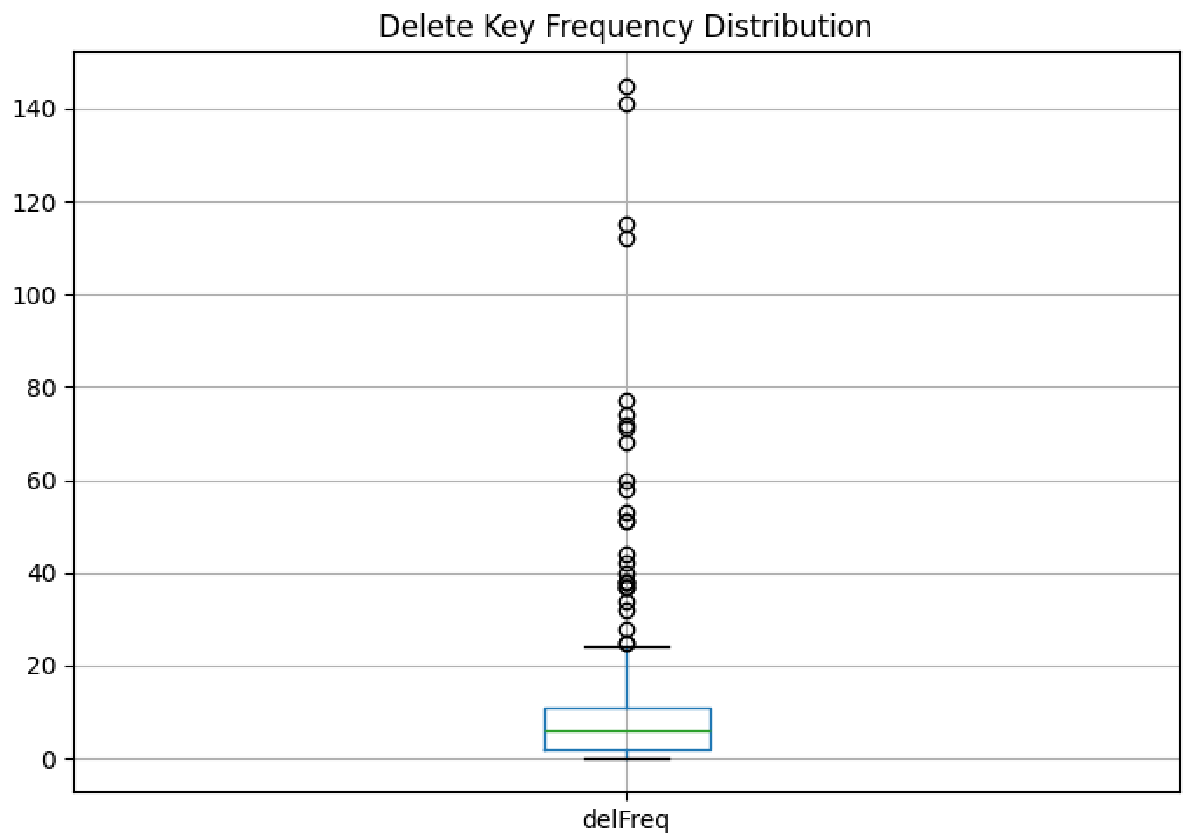
4. PC Usage Time:

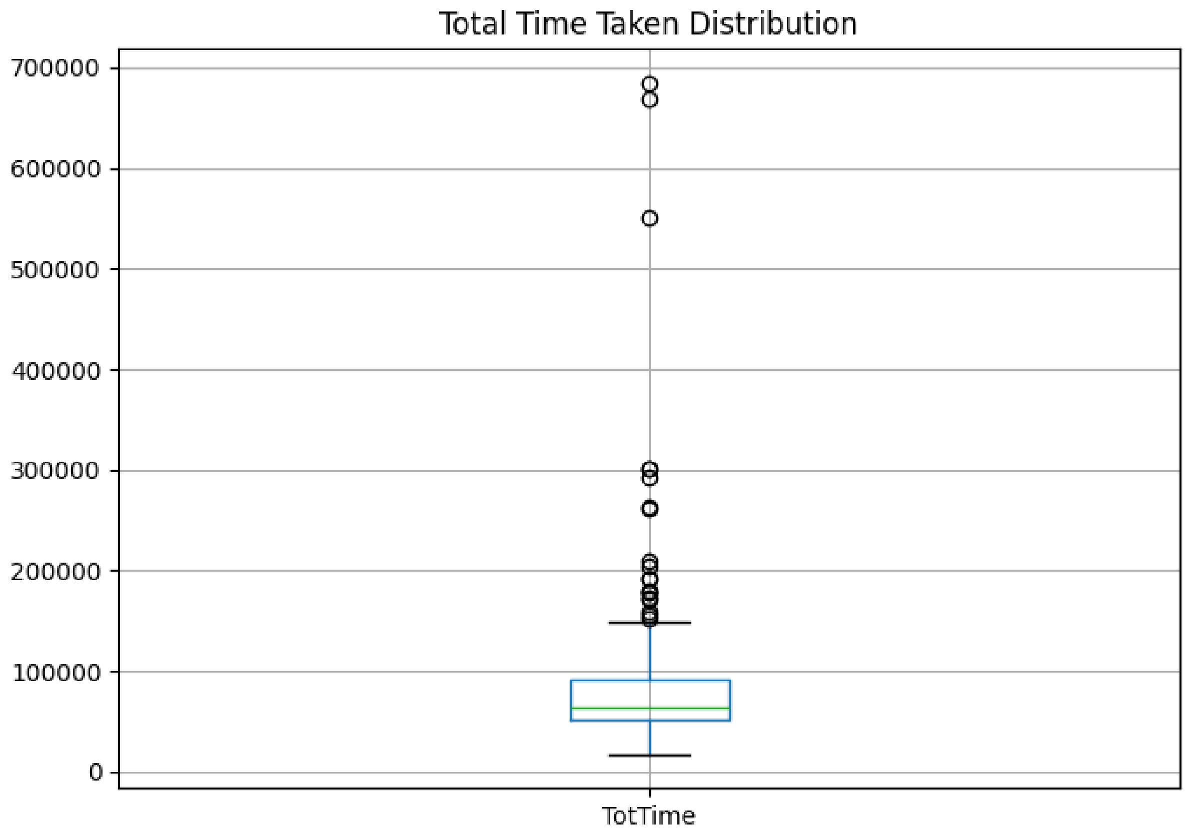
A significant number use PCs more than 3 hours per day, suggesting regular computer use.

Frequency Data:


Emotion Distribution
Count
200
150
100
50
0
N
H
S
C
A
emotionIndex


Text Index Distribution
Count
textIndex
FI
FR
250
200
150
100
50
0

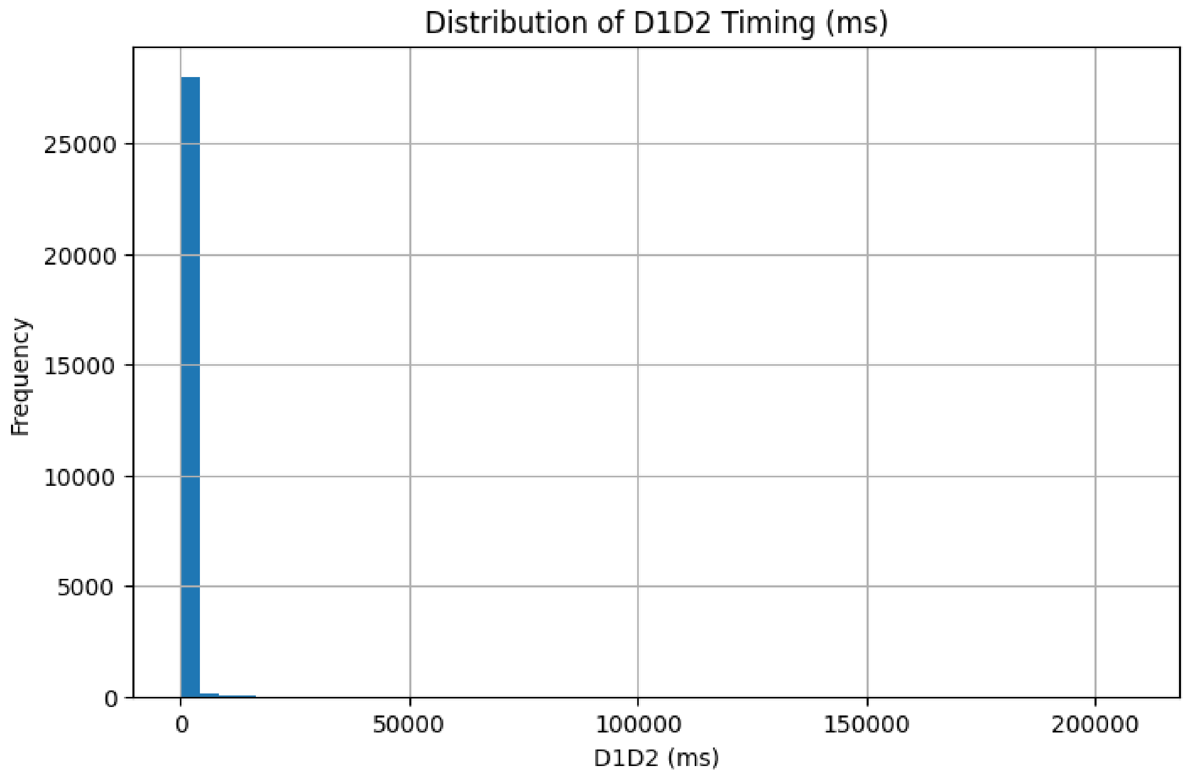

Delete Key Frequency Distribution
delFreq
140
120
100
80
60
40
20
0



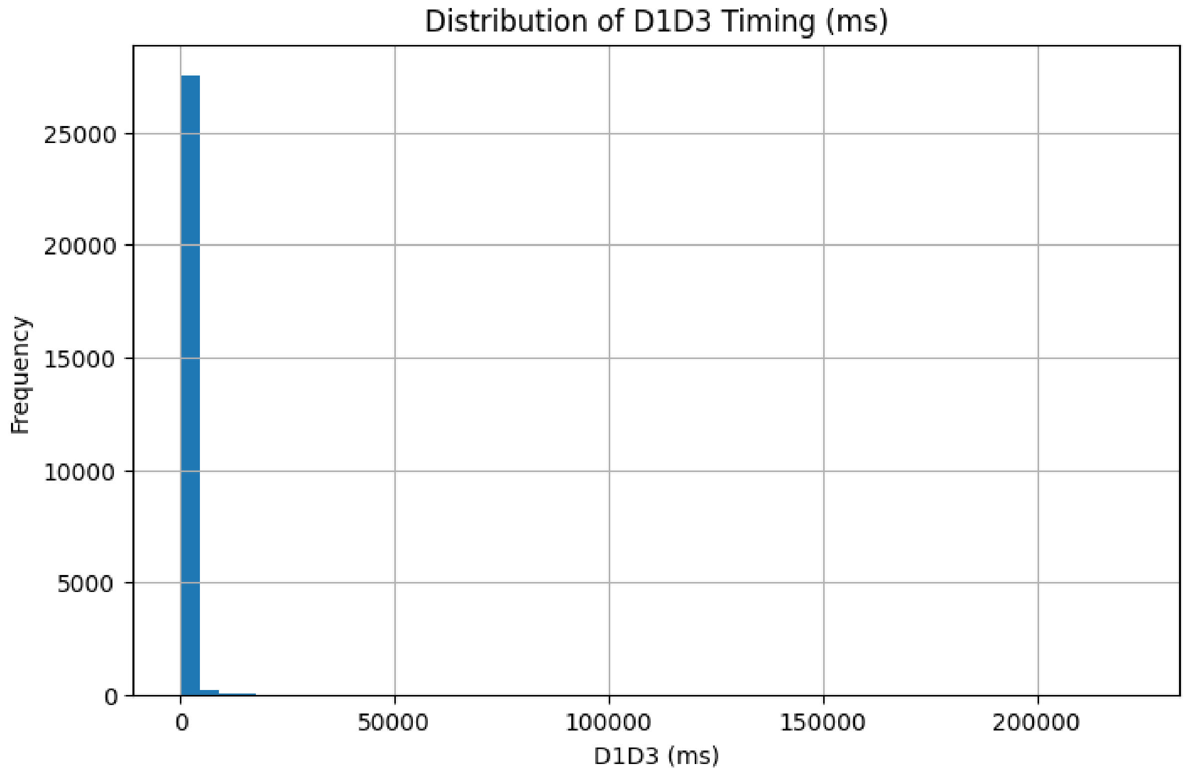
Inference:

- Emotion Distribution – shows the frequency of each emotionIndex.
- Text Index Distribution – shows how often each textIndex appears.
- Delete Key Frequency Distribution – a boxplot depicting the spread and outliers of delFreq.
- Total Time Taken Distribution – a boxplot representing the distribution of TotTime, handling missing or invalid entries.

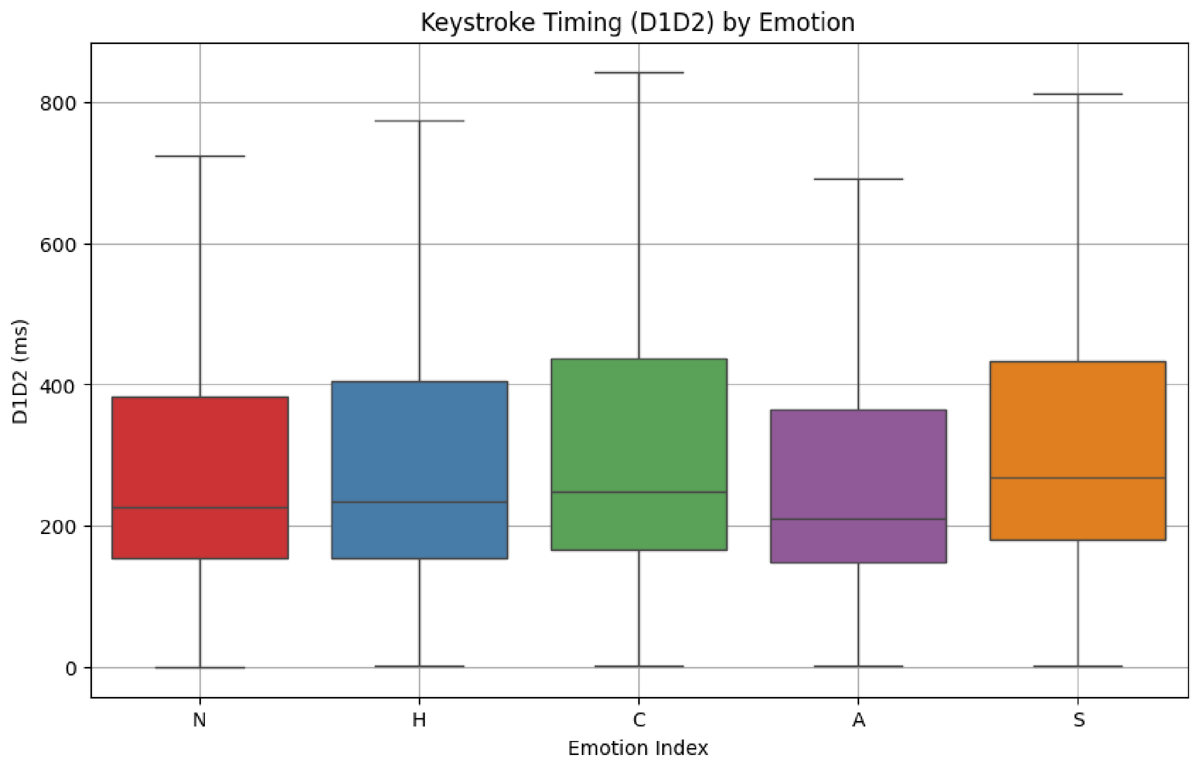
Free and fixed typing dataset:

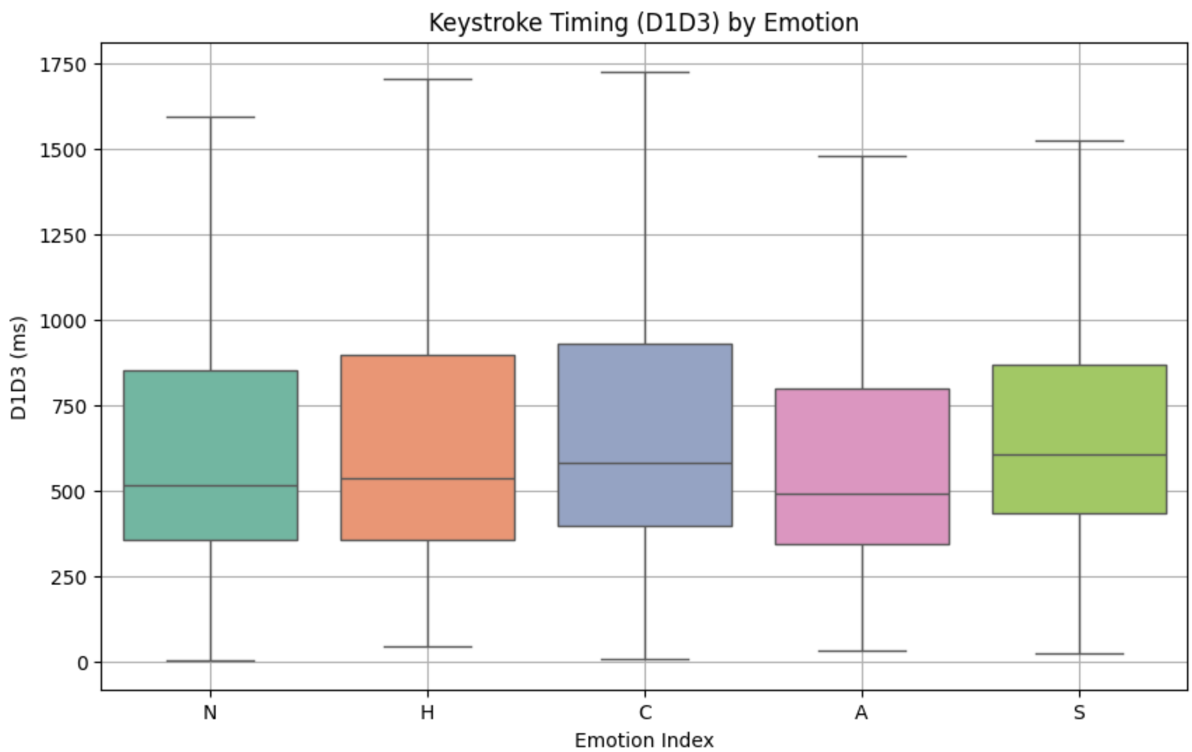


Distribution of D1D2 Timing (ms)
Frequency
0
5000
10000
15000
20000
25000
0
50000
100000
150000
200000
D1D2 (ms)



Distribution of D1D3 Timing (ms)
Frequency
0
5000
10000
15000
20000
25000
0
50000
100000
150000
200000
D1D3 (ms)





Inference:

- D1D2 and D1D3 (keystroke timing intervals) are mostly clustered at lower values, with a long tail suggesting occasional delays in typing — possibly due to hesitation or cognitive load.
- Most values are concentrated between 0–2000 ms, aligning with typical human typing speeds.
- Some emotions (like certain non-neutral states) may have slightly wider ranges or higher medians, hinting at a cognitive or emotional influence on typing behavior.

- D1D3 by Emotion: Also reflects differences in how long it takes to type sequences under various emotions — useful for behavioral analysis or emotion-aware typing prediction models.

6. **Data Cleaning steps** (Describe in detail data cleaning steps under taken. If required along with code)

Participants info:
Data Cleaning Summary

- No missing values detected.
- Data types are appropriate.
- Country names have inconsistent formats (some are in French or underscore-separated).

Code:

```
()# Standardize country names (basic fixes)
df1['country'] = df1['country'].str.replace('_', ' ')
df1['country'] = df1['country'].replace({
    'Suisse': 'Switzerland',
    'Barbade': 'Barbados',
    'Belgique': 'Belgium',
    'Arabie Saoudite': 'Saudi Arabia',
    'United States': 'USA'
})

# Standardize all text columns (lowercase, strip whitespace)
text_cols = df1.select_dtypes(include='object').columns
df1[text_cols] = df1[text_cols].apply(lambda x: x.str.strip().str.lower())

# Confirm changes
df1.head()
```

Frequency dataset:
<u>Converting Columns to Correct Data Types</u>

```
df_freq['TotTime'] = pd.to_numeric(df_freq['TotTime'], errors='coerce')
```

	0
User ID	0
textIndex	0
emotionIndex	0
delFreq	0
leftFreq	0
TotTime	235

dtype: int64

```
df_freq.dropna(inplace=True)
df_freq['TotTime'].fillna(df_freq['TotTime'].median(), inplace=True)
```

	0
User ID	0
textIndex	0
emotionIndex	0
delFreq	0
leftFreq	0
TotTime	0

dtype: int64

Free and Fixed Text Typing Dataset:

```
# Replacing comma with dot in relevant columns and convert to numeric
cols_to_clean = ['keyDown', 'keyUp', 'D1U1', 'D1U2', 'U1D2', 'U1U2', 'D1U3']
```

```
for col in cols_to_clean:
    df[col] = df[col].str.replace(',', '.', regex=False)
    df[col] = pd.to_numeric(df[col], errors='coerce')

# Removing rows where essential timing data is missing
df_cleaned = df.dropna(subset=['keyDown', 'keyUp', 'D1D2', 'D1D3'])
```

7. **Data Pre-processing steps** (Describe in detail data pre-processing steps under taken. If required along with code)

   Participants Info:

```
from sklearn.preprocessing import LabelEncoder

# Created a copy to preserve the cleaned dataframe
df1_preprocessed = df1.copy()

# Applying Label Encoding to all categorical columns except userId
label_encoders = {}
for col in df1_preprocessed.columns:
    if df1_preprocessed[col].dtype == 'object':
        le = LabelEncoder()
        df1_preprocessed[col] = le.fit_transform(df1_preprocessed[col])


df1_preprocessed.head
```

Frequency dataset:

```
from sklearn.preprocessing import LabelEncoder

# Apply Label Encoding
le_emotion = LabelEncoder()
le_text = LabelEncoder()

df_freq['emotionIndex_encoded'] = le_emotion.fit_transform(df_freq['emotionIndex'])
df_freq['textIndex_encoded'] = le_text.fit_transform(df_freq['textIndex'])

df_freq
```

Free and Fixed text Typing Data set:

```
from sklearn.preprocessing import MinMaxScaler
scaler = MinMaxScaler()
df_scaled = df_cleaned.copy()
df_scaled[['D1D2', 'D1D3']] = scaler.fit_transform(df_scaled[['D1D2', 'D1D3']])
```

8. **Feature scaling\Normalization (if applied) – Which technique implemented why?**

For our project, **Min-Max Normalization** was used to scale the features in the dataset. Since the features like **latency**, **dwell time**, and **flight time** have different ranges and units, normalization ensures all features contribute equally during model training.

Why Min-Max Scaling?

- Keystroke data includes small time values (e.g., milliseconds), which vary widely across users.
- Many ML models (like **KNN**, **SVM**, and **Logistic Regression**) perform better when features are on the same scale.
- Min-Max scaling transforms all values to a range between 0 and 1, which preserves the shape of the original distribution.

```
# data standardization
scaler = MinMaxScaler()
data_fixed = pd.DataFrame(scaler.fit_transform(data_fixed),
columns=data_fixed.columns)
```

9. **Model building- Describe in detail what all models were build, train and test split size. If required put sample code for model building**

We experimented with four machine learning classifiers using a **10-fold cross-validation** technique to evaluate performance:

- **Random Forest Classifier (RF)**
- **XGBoost Classifier (XGB)**
- **Support Vector Machine (SVM)**
- **Multi-Layer Perceptron (MLP)**

Train-Test Split

After evaluation, the best-performing model (`XGBoost`) was trained using:

- **Training set:** 80%
- **Testing set:** 20%
  via `train_test_split()`.

Experiment on all datasets individually and combining them

10. **Model evaluation – Comparative description regarding training and testing accuracy**

1. Free Text Typing Dataset

Description:

- The **Free Text dataset** contains keystroke dynamics where users typed freely without a script.
- Features: **Hold Time**, **Flight Time**, and other latency metrics.
- Challenge: High variance due to user typing styles and unstructured text.

Modeling Approach:

- **Preprocessing:** Missing values removed, features scaled using MinMaxScaler.
- **Feature Selection:** Filtered to keystroke dynamics only.
- **Models Tested:** Random Forest, XGBoost, SVM, MLP
- **Cross-validation:** 10-fold on this dataset.

Model	F1 Micro (±std)	F1 Macro (±std)
RF	0.211 ± 0.051	0.199 ± 0.061
XGB	**0.451 ± 0.050**	**0.284 ± 0.075**
SVM	0.472 ± 0.013	0.128 ± 0.002
MLP	0.312 ± 0.107	0.183 ± 0.074

Insights:

- **XGBoost** showed the most balanced performance.
- **SVM** had high micro-F1 but very poor macro-F1, indicating class imbalance issues.
- **MLP** was unstable, likely due to high dimensionality and limited data.
- Free text data offers **rich behavioral signals** but suffers from **inconsistency**.

2. Fixed Text Typing Dataset

Description:

- In the **Fixed Text dataset**, users typed the same predefined sentence.
- Features: Similar to Free Text but controlled for content and structure.
- Advantage: More uniformity, easier to compare between users.

Modeling Approach:

- **Used as Test Data** in a cross-domain test (trained on Free, tested on Fixed).
- **Voting Classifier** (RF, XGB, SVM) was built and evaluated here.

Metric	Score
Accuracy	0.67
True Positive Rate	0.17
False Positive Rate	0.21
Precision	0.11
F1 Score	0.13

F1 Micro	0.39
F1 Macro	0.15

Insights:

- **Voting Classifier** was moderately successful.
- **Cross-domain drop** in accuracy and F1 due to difference in typing contexts.
- Suggests that models trained on Free Text don't generalize well to Fixed Text.
- Fixed text offers **high consistency**, ideal for controlled studies.

11. Attach collab code file (ipynb file as object)

→ Attached

12. Conclusion

This study implemented a multi-class classification approach using keystroke dynamics data. Among all models evaluated:

- **XGBoost** emerged as the best performer in both micro and macro F1-score.
- **SVM** had the highest micro-F1 but poor macro-F1, suggesting class imbalance sensitivity.
- **MLP** showed unstable results due to high variance.
- The **Voting Classifier**, when tested across datasets, showed moderate generalization but struggled due to possible domain shift between datasets (free vs fixed).

Future work may focus on:

- Data augmentation or SMOTE for balancing classes.
- Feature engineering using deeper timing relationships.
- Ensemble calibration or stacking for more robust predictions.

13. References

Papers

[1] M. Tahir, Z. Halim, A. U. Rahman, M. Waqas, S. Tu, S. Chen, and Z. Han, "Non-Acted Text and Keystrokes Database and Learning Methods to Recognize Emotions," *ACM Trans. Multimedia Comput. Commun. Appl. (TOMM)*, vol. 18, no. 2, pp. 1–24, Article no. 61, Apr. 2022, doi: 10.1145/3480968.

[2] L. Yang and S.-F. Qin, "A Review of Emotion Recognition Methods From Keystroke, Mouse, and Touchscreen Dynamics," *IEEE Access*, vol. 9, pp. 132362–132377, 2021, doi: 10.1109/ACCESS.2021.3115289.

[3] A. Maalej, I. Kallel, and J. J. Sanchez Medina, "Investigating Keystroke Dynamics and their Relevance for Real-time Emotion Recognition," *REGIM-Lab: Research Groups in Intelligent Machines*, ENIS University of Sfax, Tunisia; CICEI, University of Las Palmas de Gran Canaria, Spain; ISIMS, University of Sfax, Tunisia. [Online].

[4] A. Kołakowska and A. Landowska, "Keystroke Dynamics Patterns While Writing Positive and Negative Opinions," *Sensors*, vol. 21, no. 17, p. 5963, Sep. 2021, doi: 10.3390/s21175963.

[5] A. Maalej, I. Kallel, and J. J. Sánchez Medina, “Identifying Users’ Emotional States through Keystroke Dynamics,” *Proceedings of the International Conference on Human-Computer Interaction (HCI International)*, vol. 11872, pp. 83-96, 2019, doi: 10.1007/978-3-030-21811-3_8.

Datasets:

https://ieee-dataport.org/open-access/emosurv-typing-biometric-keystroke-dynamics-dataset-emotion-labels-created-using