FREEZING OF GAIT PREDICTION IN PARKINSON'S DISEASE

BY: AGAM GUPTA
PRACHI GOEL
DEEPAK KUMAR
HARSH YADAV

CONTENT

1 PROBLEM INTRODUCTION

2 PREVIOUS KNOWLEDGE

3 MODEL SELECTION

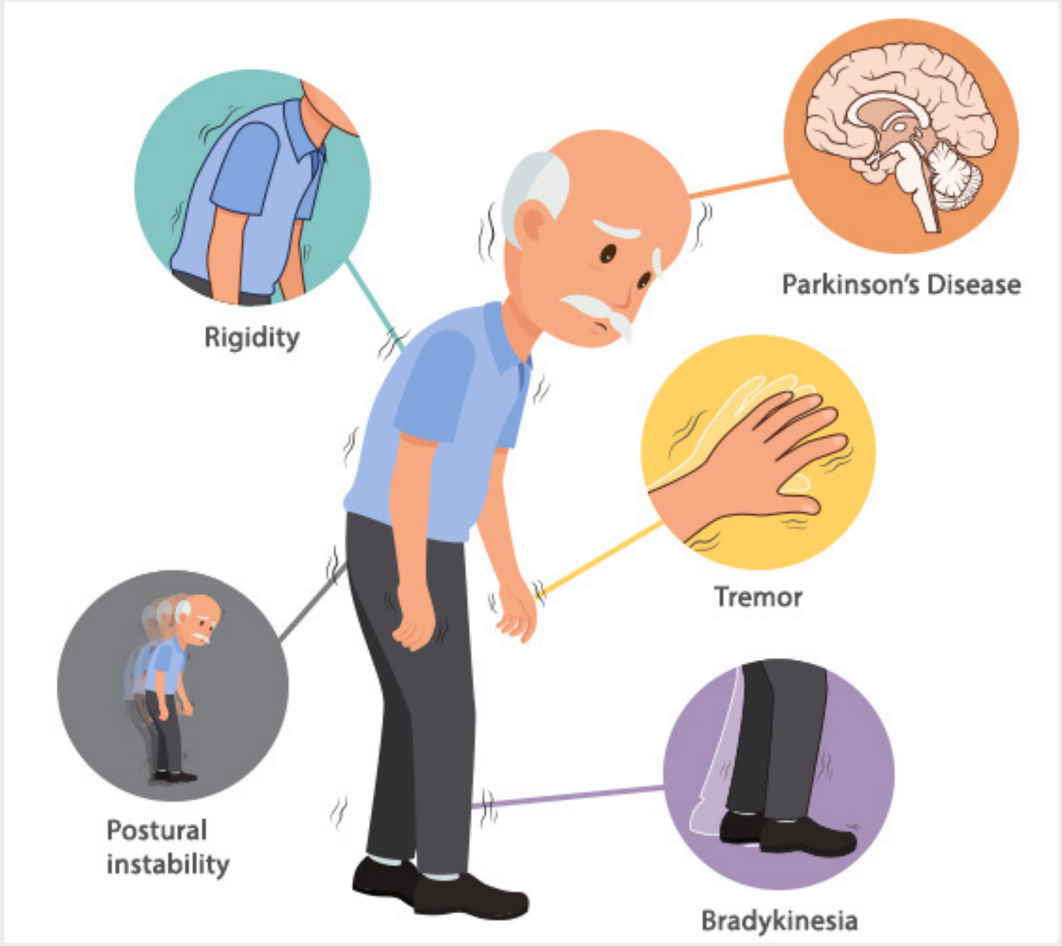
4 MODEL EVALUATION

Parkinson's Disease

- Parkinson's Disease is a chronic and progressive neurological disorder that affects movement and coordination
- Affects the nervous system and causes movement problems, tremors, and stiffness
- It is diagnosed based on the patient's medical history, physical examination, and neurological tests, but that data mining techniques can help improve the accuracy and speed of diagnosis

Symptoms

1. Tremor in hands, arms, legs, jaw, or head.
2. Muscle stiffness.
3. Slowness of movement
4. Impaired balance and coordination.
5. Depression and other emotional changes.
6. Difficulty swallowing, chewing, and speaking.
7. Urinary problems.
8. Skin problems.



SOURCE:https://neurologysleepcentre.com/blog/what-is-parkinsons-disease/

OVERVIEW OF THE PROBLEM

PROBLEM

Detect the start and stop of freezing episode and occurence in series of 3 types of fog event:

1. Start hesitation
2. Turn
3. Walking

GAIT FEATURES

Human gait is a sequence of involuntary movements, cyclic repeated and triggered by voluntary movement.

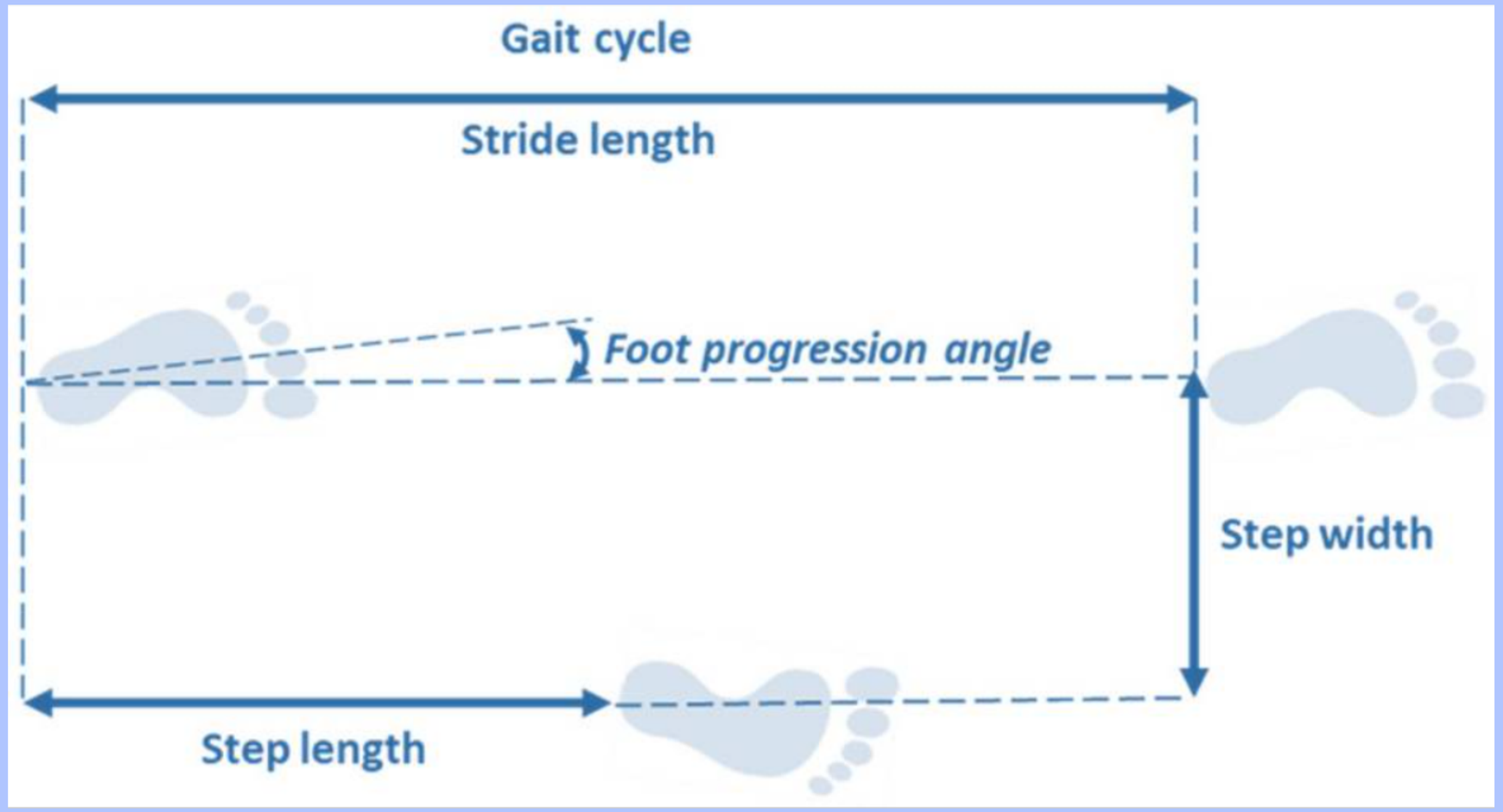
Components used to objectively measureand analyze gait cycle are:-

- Spatiotemporal features
- Kinematics features
- Kinetics features

1 SPATIOTEMPORAL FEATURES

- **Gait Cycle**:-The time from initial contact to initial contact on the same foot including both the stance phase and swing phase.
- **Stance Phase**:-The period during which the foot is in contact with the support surface during one gait cycle.
- **Swing Phase**:- The period during which the foot is airborne during one gait cycle.
- **Double Limb Support**:-The period during which both feet are in contact with the support surface during one gait cycle.

- **Single Limb Support:-**The period during which only one foot is in contact with the support surface during one gait cycle.
- **Step Duration :-** The period between 2 successive events of the same type on opposite limbs.
- **Stride Length :-** The linear distance between 2 successive events (initial contact) on the same limb.
- **Step Length:-** The linear distance between 2 successive events of same type on opposite limbs.
- **Step Width :-** The horizontal distance between 2 points on opposite limbs.
- **Foot Progression Angle** The angle between the longitudinal axis of the foot and the line of gait progression



Source:https://pubmed.ncbi.nlm.nih.gov/32580330/

2 KINETICS FEATURES

- The kinetics analysis (or dynamics of gait) is the study of the forces and their effect on motion.
- The common forces represented by the ground reaction forces (GRF) on the hip, knee and ankle points calculate on the sagittal plane.
- GRF works when feet are in contact with the ground and represent the effect of gravity on body which are counterbalance by ground contact and limb muscular movement.

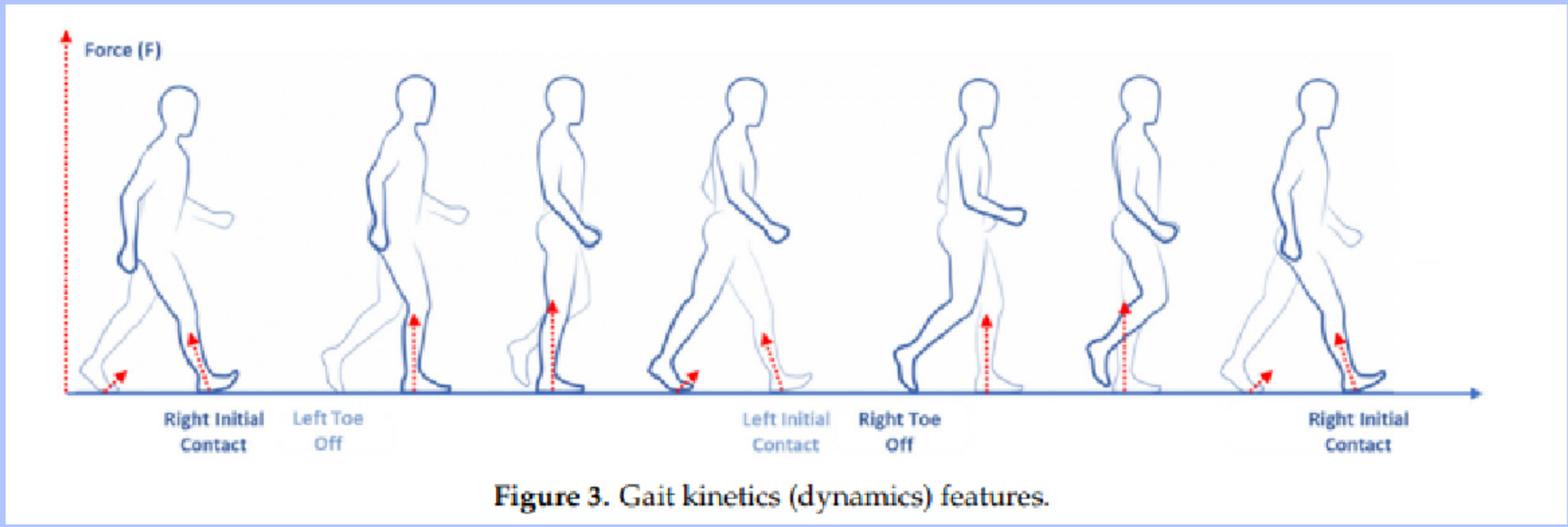


Figure 3. Gait kinetics (dynamics) features.

3

Kinematics Feature

- Kinematics analysis could describe gait features based on the sagittal, horizontal or frontal plane for several body areas and joints such as the ankle, knee, hip and pelvis.
- Kinematic features extrapolated from the stance and swing phases.
- Angular kinematics objectively quantify (as degrees) the joint's motion around axes in different gait phases.

METHODOLOGY

- Finding FOG episodic event in a individual, suffering from Parkinson disease is a multilabel classification problem.
- To find whether a patient under medication state has FOG provoking episode or not.
- We are using LGBM(Light Gradient Boosting Machine) multi-classification model.

Tasks

1. **4-meter walk test**
2. **Timed Up & Go (TUG)**
3. **Timed Up & Go (TUG) – Dual-task** (subtracting numbers while performing the TUG test)
4. **Turning task with alternating directions** (performing 4 x 360 degrees turns, each time alternating the rotation direction).
5. **Turning task – Dual-task** (same as before, but with additional number subtraction task).
6. **Hotspot Door –** A walking trial that involves opening a door, entering another room, turning, and returning to the start point.
7. **Personalized Hotspot -** walking through an area in the house that the subject describes as FoG provoking.

Dataset

DeFOG

Data collected in Subject’s home environment🏘️.

Both in “ON” and “OFF” medication state.

tDCS FOG

Data collected in the LAB🥼

Both in “ON” and “OFF” medication state.

@misc{tlvmc-parkinsons-freezing-gait-prediction, author = {Addison Howard, amit salomon, eran gazit, HCL-Jevster, Jeff Hausdorff, Leslie Kirsch, Maggie, Pieter Ginis, Ryan Holbrook, Yasir F Karim}, title = {Parkinson's Freezing of Gait Prediction}, publisher = {Kaggle}, year = {2023}, url = {https://kaggle.com/competitions/tlvmc-parkinsons-freezing-gait-prediction} }

Dataset Info()

- The tDCS FOG (tdcsfog) dataset, comprising data series collected in the lab, as subjects completed a FOG-provoking protocol

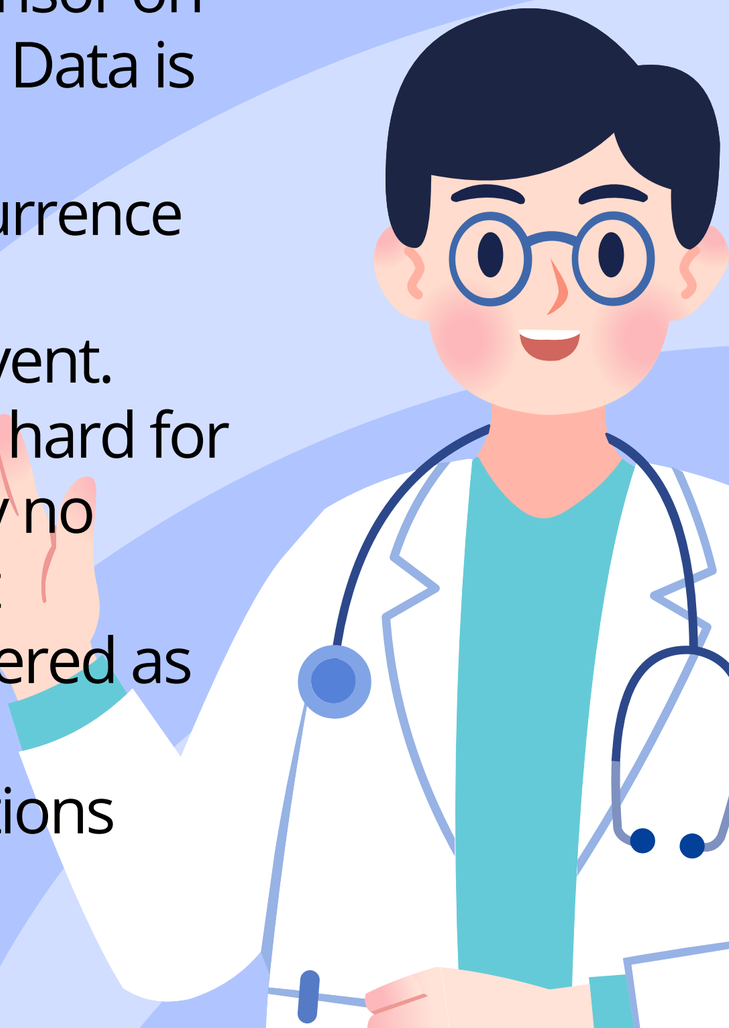
- The DeFOG (defog) dataset, comprising data series collected in the subject's home,therefore there are two additioanl columns (valid & test) to check the quality of data.

- Remove invalid data from defog dataset.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3594191 entries, 0 to 3594190
Data columns (total 9 columns):
 #   Column           Dtype
---  ------           -----
 0   Time             int64
 1   AccV             float64
 2   AccML            float64
 3   AccAP            float64
 4   StartHesitation  int64
 5   Turn             int64
 6   Walking          int64
 7   Valid            bool
 8   Task             bool
dtypes: bool(2), float64(3), int64(4)
memory usage: 198.8 MB
```

Attribute Information

- **Time :** An integer timestep.
- **AccV, AccML, and AccAP -:** Acceleration from a lower-back sensor on three axes: V - vertical, ML - mediolateral, AP - anteroposterior. Data is in units of m/s^2 for tdcsfog and g for defog and notype
- **StartHesitation, Turn, Walking Indicator** variables for the occurrence of each of the event types.
- **Event** Indicator variable for the occurrence of any FOG-type event.
- **Valid** There were cases during the video annotation that were hard for the annotator to decide if there was an Akinetic (i.e., essentially no movement) FoG or the subject stopped voluntarily. Only event annotations where the series is marked true should be considered as unambiguous.
- **Task Series** were only annotated where this value is true. Portions marked false should be considered unannotated.



DATA PREPROCESSING

- Merged several files to form our Dataset.
- We had drop the columns that are not of use as a part of feature selection.
- Extraction of Attributes while training the Dataset.
- Finding the Unique values.

Merge & Drop of Column

In [6]:

```
defog_m= defog_metadata.merge(defog, how = 'inner', left_on = 'Id', right_on = 'file')
defog_m.drop(['file','Valid','Task'], axis = 1, inplace = True)
defog_m
```

Out[6]:

	Time	AccV	AccML	AccAP	StartHesitation	Turn	Walking	Valid	Task	file
0	0	-1.002697	0.022371	0.068304	0	0	0	False	False	be9d33541d
1	1	-1.002641	0.019173	0.066162	0	0	0	False	False	be9d33541d
2	2	-0.999820	0.019142	0.067536	0	0	0	False	False	be9d33541d
3	3	-0.998023	0.018378	0.068409	0	0	0	False	False	be9d33541d
4	4	-0.998359	0.016726	0.066448	0	0	0	False	False	be9d33541d
...	...	...	...	...	...	...	...	...	...	...
109120	109120	-0.939241	0.031564	-0.394737	0	0	0	False	False	06414383cf
109121	109121	-0.941096	0.031582	-0.392626	0	0	0	False	False	06414383cf
109122	109122	-0.940131	0.029092	-0.394385	0	0	0	False	False	06414383cf
109123	109123	-0.939872	0.028058	-0.398664	0	0	0	False	False	06414383cf
109124	109124	-0.939006	0.026628	-0.398454	0	0	0	False	False	06414383cf

13525702 rows × 10 columns

defog

	Id	Subject	Visit	Medication
0	02ab235146	e1f62e	2	on
1	02ea782681	ae2d35	2	on
2	06414383cf	8c1f5e	2	off
3	092b4c1819	2874c5	1	off
4	0a900ed8a2	0e3d49	2	on
...	...	...	...	...
132	f3a921edee	1a778d	1	off
133	f40e8c6ebe	575c60	1	off
134	f8ddbdd98d	107712	1	on
135	f9efef91fb	5d9cae	2	off
136	f9fc61ce85	040587	1	on

137 rows × 4 columns

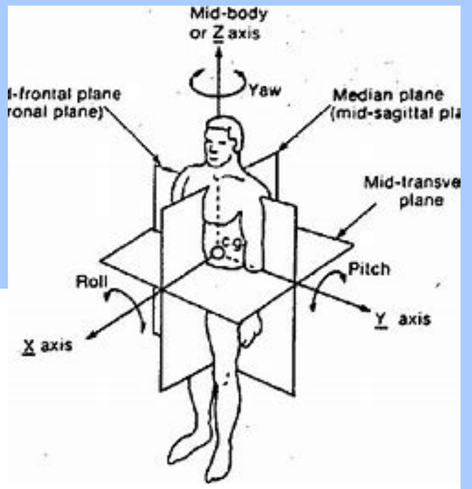
defog-metadata

Description of Data

```
# summary table function
def summary(df):
    print(f'data shape: {df.shape}')
    summ = pd.DataFrame(df.dtypes, columns=['data type'])
    summ['#missing'] = df.isnull().sum().values * 100
    summ['%missing'] = df.isnull().sum().values / len(df)
    summ['#unique'] = df.nunique().values
    desc = pd.DataFrame(df.describe(include='all').transpose())
    summ['min'] = desc['min'].values
    summ['max'] = desc['max'].values
    summ['first value'] = df.loc[0].values
    summ['second value'] = df.loc[1].values
    summ['third value'] = df.loc[2].values

    return summ
```

Summary




```
data shape: (4090530, 11)
```

Out[14]:

	data type	#missing	%missing	#unique	min	max	first value	second value	third value
Id	object	0	0.0	91	NaN	NaN	02ea782681	02ea782681	02ea782681
Subject	object	0	0.0	38	NaN	NaN	ae2d35	ae2d35	ae2d35
Visit	int64	0	0.0	2	1.0	2.0	2	2	2
Medication	object	0	0.0	2	NaN	NaN	on	on	on
Time	int64	0	0.0	338197	1000.0	414387.0	1000	1001	1002
AccV	float64	0	0.0	3485229	-6.024701	4.458365	-0.970018	-0.984375	-0.984375
AccML	float64	0	0.0	3539384	-2.115008	4.524038	0.061626	0.044497	0.029016
AccAP	float64	0	0.0	3437121	-5.11865	4.388132	-0.265625	-0.265625	-0.265625
StartHesitation	int64	0	0.0	2	0.0	1.0	0	0	0
Turn	int64	0	0.0	2	0.0	1.0	0	0	0
Walking	int64	0	0.0	2	0.0	1.0	0	0	0

Feature Engineering

	Id	Subject	Visit	Medication	Time	AccV	AccML	AccAP	StartHesitation	Turn	Walking	event
0	02ea782681	ae2d35	2	on	1000	-0.970018	0.061626	-0.265625	0	0	0	Normal
1	02ea782681	ae2d35	2	on	1001	-0.984375	0.044497	-0.265625	0	0	0	Normal
2	02ea782681	ae2d35	2	on	1002	-0.984375	0.029016	-0.265625	0	0	0	Normal
3	02ea782681	ae2d35	2	on	1003	-0.984375	0.015625	-0.265625	0	0	0	Normal
4	02ea782681	ae2d35	2	on	1004	-0.984670	0.015330	-0.265625	0	0	0	Normal
												...
												Normal
												Normal
												Normal
												Normal
												Normal

In [16]:

```
conditions = [
    (defog_m['StartHesitation'] == 1),
    (defog_m['Turn'] == 1),
    (defog_m['Walking'] == 1)]
choices = ['StartHesitation', 'Turn', 'Walking']
defog_m['event'] = np.select(conditions, choices, default='Normal')
```

Train-Test Split

In [31]:
```
X
```

Out[31]:

	AccV	AccML	AccAP
0	-0.970018	0.061626	-0.265625
1	-0.984375	0.044497	-0.265625
2	-0.984375	0.029016	-0.265625
3	-0.984375	0.015625	-0.265625
4	-0.984670	0.015330	-0.265625
...	...	...	...
4090525	-0.961216	0.142428	-0.289655
4090526	-0.960343	0.142836	-0.290506
4090527	-0.957958	0.145494	-0.290007
4090528	-0.960616	0.145839	-0.291527
4090529	-0.967076	0.144342	-0.292384

4090530 rows × 3 columns

```
y
```

```
0          0
1          0
2          0
3          0
4          0
          ..
4090525    0
4090526    0
4090527    0
4090528    0
4090529    0
Name: target, Length: 4090530, dtype: int64
```

```
y.unique()
```

```
array([0, 2, 3, 1])
```

```
y.value_counts()
```

```
0    3404683
2     586829
3      98518
1        500
Name: target, dtype: int64
```

```
# spliting dataset into training and test set
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=1004)
```

Classifiers

1. **LGBM**
2. **Naive Bayes**
3. **Random Forest**
4. **XG Boost**
5. **Decision Tree**
6. **KNN**

LGBM

LightGBM is a gradient boosting framework that uses tree based learning algorithms. It is designed to be distributed and efficient with the following advantages:

- Faster training speed and higher efficiency
- Lower memory usage.
- Better accuracy.
- Support of parallel and GPU learning.
- Capable of handling large-scale data.
- Light GBM can handle the large size of data and takes lower memory to run.

Control Parameters of LGBM

- **max_depth** : Describes maximum depth of tree. Handle model overfitting.
- **min_data_in_leaf** : Default value is 20, optimum value. It is also used to deal with overfitting.
- **feature_fraction**: Used when your boosting is random forest. 0.8 feature fraction means LightGBM will select 80% of parameters randomly in each iteration for building trees.
- **bagging_fraction** : specifies the fraction of data to be used for each iteration and is generally used to speed up the training and avoid overfitting.
- **early_stopping_round** : This parameter can help you speed up your analysis. Model will stop training if one metric of one validation data doesn’t improve in last early_stopping_round rounds. This will reduce excessive iterations.
- **lambda** : lambda specifies regularization. Typical value ranges from 0 to 1.

Using LGBM Model

```
params={}
params['learning_rate']=0.03
params['boosting_type']='gbdt' #GradientBoostingDecisionTree
params['objective']='multiclass' #Multi-class target feature
params['metric']='multi_logloss' #metric for multi-class
params['max_depth']=7
params['num_class']=4 #no.of unique values in the target class not inclusive of the end value
params['verbose']=-1
#training the model
clf_2=lgb.train(params,d_train,1000)  #training the model on 1,000 epocs
#prediction on the test dataset
y_pred_2=clf_2.predict(X_test)
```

LGBM

```
Accuracy: 0.8336955520841227
Confusion Matrix:
[[1017220       0    3799       0]
 [    152       0       0       0]
 [ 170414       0    5857       0]
 [  29701       0      16       0]]
Classification Report:
              precision    recall  f1-score   support

           0       0.84      1.00      0.91   1021019
           1       0.00      0.00      0.00       152
           2       0.61      0.03      0.06    176271
           3       0.00      0.00      0.00     29717

    accuracy                           0.83   1227159
   macro avg       0.36      0.26      0.24   1227159
weighted avg       0.78      0.83      0.77   1227159
```

KNN

```python
# KNN
from sklearn.neighbors import KNeighborsClassifier

# Initialize the KNN classifier
knn_classifier = KNeighborsClassifier(n_neighbors=5)  # You
(n_neighbors) based on your preference

# Train the classifier
knn_classifier.fit(X_train, y_train)

# Make predictions on the test set
y_pred = knn_classifier.predict(X_test)
```

```
Accuracy: 0.8303520570683994
Confusion Matrix:
[[972169      4  47315   1531]
 [   141      2      9      0]
 [130247      0  45794    230]
 [ 27661      0   1047   1009]]
Classification Report:
              precision    recall  f1-score   support

           0       0.86      0.95      0.90   1021019
           1       0.33      0.01      0.03       152
           2       0.49      0.26      0.34    176271
           3       0.36      0.03      0.06     29717

    accuracy                           0.83   1227159
   macro avg       0.51      0.31      0.33   1227159
weighted avg       0.79      0.83      0.80   1227159
```

Naive Bayes

```
# Naive Bayes Classifier
from sklearn.naive_bayes import GaussianNB

# Initialize the Naive Bayes Classifier
clf = GaussianNB()

# Train the classifier
clf.fit(X_train, y_train)

# Make predictions on the test set
y_pred = clf.predict(X_test)
```

```
# Evaluate metrics
accuracy = accuracy_score(y_test, y_pred)
conf_matrix = confusion_matrix(y_test, y_pred)
classification_rep = classification_report(y_test, y_pred)
```

Decision Tree

```
# Decision tree classifier
from sklearn.tree import DecisionTreeClassifier
from sklearn.metrics import accuracy_score, classification_report, confusion_matrix

# Initialize the Decision Tree Classifier
clf = DecisionTreeClassifier(random_state=42)

# Train the classifier
clf.fit(X_train, y_train)

# Make predictions on the test set
y_pred = clf.predict(X_test)
```

```
# Evaluate metrics
accuracy = accuracy_score(y_test, y_pred)
conf_matrix = confusion_matrix(y_test, y_pred)
classification_rep = classification_report(y_test, y_pred)
```

Decision Tree

Accuracy: 0.7644649144894834

Confusion Matrix:

```
[[876620    134 118662  25603]
 [   124      5     20      3]
 [113695     24  58809   3743]
 [ 23441      2   3588   2686]]
```

Classification Report:

	precision	recall	f1-score	support
0	0.86	0.86	0.86	1021019
1	0.03	0.03	0.03	152
2	0.32	0.33	0.33	176271
3	0.08	0.09	0.09	29717
accuracy			0.76	1227159
macro avg	0.33	0.33	0.33	1227159
weighted avg	0.77	0.76	0.77	1227159

Naive Bayes

Accuracy: 0.8320144333374893

Confusion Matrix:

```
[[1021014       0       5       0]
 [    152       0       0       0]
 [ 176271       0       0       0]
 [  29717       0       0       0]]
```

Classification Report:

	precision	recall	f1-score	support
0	0.83	1.00	0.91	1021019
1	0.00	0.00	0.00	152
2	0.00	0.00	0.00	176271
3	0.00	0.00	0.00	29717
accuracy			0.83	1227159
macro avg	0.21	0.25	0.23	1227159
weighted avg	0.69	0.83	0.76	1227159

XG Boost

```
# XG boost
from xgboost import XGBClassifier

# Initialize the XGBoost classifier
xgb_classifier = XGBClassifier(n_estimators=100, learning_rate=0.1, max_depth=3, random_state
=42)

# Train the classifier
xgb_classifier.fit(X_train, y_train)

# Make predictions on the test set
y_pred = xgb_classifier.predict(X_test)
```

Random Forest

```
# Make predictions on the test set
y_pred = rf_classifier.predict(X_test)

# Evaluate metrics
accuracy = accuracy_score(y_test, y_pred)
conf_matrix = confusion_matrix(y_test, y_pred)
classification_rep = classification_report(y_test, y_pred)

# Print the metrics
print(f'Accuracy: {accuracy}')
print(f'Confusion Matrix:\n{conf_matrix}')
print(f'Classification Report:\n{classification_rep}')
```

Random Forest

```
Accuracy: 0.8394780138515058
Confusion Matrix:
[[987819      1  32449    750]
 [   139      0     13      0]
 [134598      0  41523    150]
 [ 28305      0    581    831]]
Classification Report:
              precision    recall  f1-score   support

           0       0.86      0.97      0.91   1021019
           1       0.00      0.00      0.00       152
           2       0.56      0.24      0.33    176271
           3       0.48      0.03      0.05     29717

    accuracy                           0.84   1227159
   macro avg       0.47      0.31      0.32   1227159
weighted avg       0.81      0.84      0.81   1227159
```

XG Boost

```
Accuracy: 0.8333907831014563
Confusion Matrix:
[[1019172       0    1847       0]
 [    152       0       0       0]
 [ 172740       0    3531       0]
 [  29713       0       4       0]]
Classification Report:
              precision    recall  f1-score   support

           0       0.83      1.00      0.91   1021019
           1       0.00      0.00      0.00       152
           2       0.66      0.02      0.04    176271
           3       0.00      0.00      0.00     29717

    accuracy                           0.83   1227159
   macro avg       0.37      0.25      0.24   1227159
weighted avg       0.79      0.83      0.76   1227159
```

TDCS FOG(🥼)

	event
Normal	4871262
Turn	1678782
StartHesitation	304790
Walking	207838

	Id	Subject	Visit	Test	Medication	Time	AccV	AccML	AccAP	StartHesitation	Turn	Walking
0	003f117e14	4dc2f8	3	2	on	0	-9.533939	0.566322	-1.413525	0	0	0
1	003f117e14	4dc2f8	3	2	on	1	-9.536140	0.564137	-1.440621	0	0	0
2	003f117e14	4dc2f8	3	2	on	2	-9.529345	0.561765	-1.429332	0	0	0
3	003f117e14	4dc2f8	3	2	on	3	-9.531239	0.564227	-1.415490	0	0	0
4	003f117e14	4dc2f8	3	2	on	4	-9.540825	0.561854	-1.429471	0	0	0
...	...	...	...	...	...	...	...	...	...	...	...	...
7062667	ffda8fadfd	7fcee9	20	1	off	4220	-9.403467	0.089003	-3.220304	0	0	0
7062668	ffda8fadfd	7fcee9	20	1	off	4221	-9.404246	0.090531	-3.216584	0	0	0
7062669	ffda8fadfd	7fcee9	20	1	off	4222	-9.405770	0.084380	-3.224039	0	0	0
7062670	ffda8fadfd	7fcee9	20	1	off	4223	-9.403579	0.084236	-3.236686	0	0	0
7062671	ffda8fadfd	7fcee9	20	1	off	4224	-9.405036	0.082027	-3.234458	0	0	0

7062672 rows × 12 columns

Summary

```
data shape: (7062672, 12)
```

:

	data type	#missing	%missing	#unique	min	max	first value	second value	third value
Id	object	0	0.0	833	NaN	NaN	003f117e14	003f117e14	003f117e14
Subject	object	0	0.0	62	NaN	NaN	4dc2f8	4dc2f8	4dc2f8
Visit	int64	0	0.0	7	2.0	20.0	3	3	3
Test	int64	0	0.0	3	1.0	3.0	2	2	2
Medication	object	0	0.0	2	NaN	NaN	on	on	on
Time	int64	0	0.0	97077	0.0	97076.0	0	1	2
AccV	float64	0	0.0	7027490	-35.521119	20.906953	-9.533939	-9.53614	-9.529345
AccML	float64	0	0.0	7030366	-26.164398	27.484719	0.566322	0.564137	0.561765
AccAP	float64	0	0.0	7028071	-47.829639	30.337694	-1.413525	-1.440621	-1.429332
StartHesitation	int64	0	0.0	2	0.0	1.0	0	0	0
Turn	int64	0	0.0	2	0.0	1.0	0	0	0
Walking	int64	0	0.0	2	0.0	1.0	0	0	0

LGBM

Accuracy: 0.7232521672029366

Confusion Matrix:

```
[[924192     40  50372    192]
 [ 44500     91  15740    453]
 [238070     61  97022    305]
 [ 28600     80  12503    314]]
```

Classification Report:

	precision	recall	f1-score	support
0	0.75	0.95	0.84	974796
1	0.33	0.00	0.00	60784
2	0.55	0.29	0.38	335458
3	0.25	0.01	0.01	41497
accuracy			0.72	1412535
macro avg	0.47	0.31	0.31	1412535
weighted avg	0.67	0.72	0.67	1412535

XG Boost

Accuracy: 0.719957381587005

Confusion Matrix:

```
[[924753      2  49871    170]
 [ 45221      0  15131    432]
 [243224      0  91902    332]
 [ 29006      1  12180    310]]
```

Classification Report:

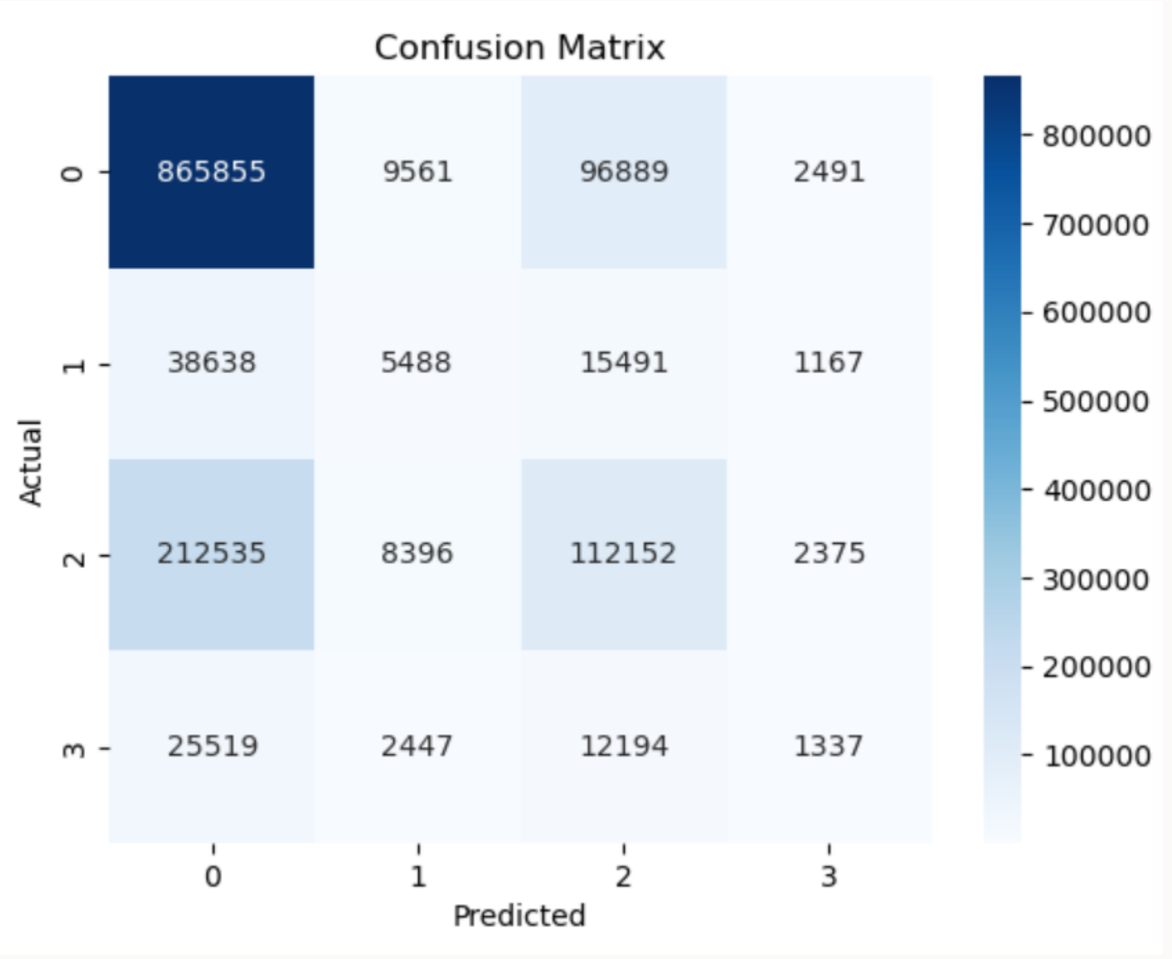
	precision	recall	f1-score	support
0	0.74	0.95	0.83	974796
1	0.00	0.00	0.00	60784
2	0.54	0.27	0.36	335458
3	0.25	0.01	0.01	41497
accuracy			0.72	1412535
macro avg	0.38	0.31	0.30	1412535
weighted avg	0.65	0.72	0.66	1412535

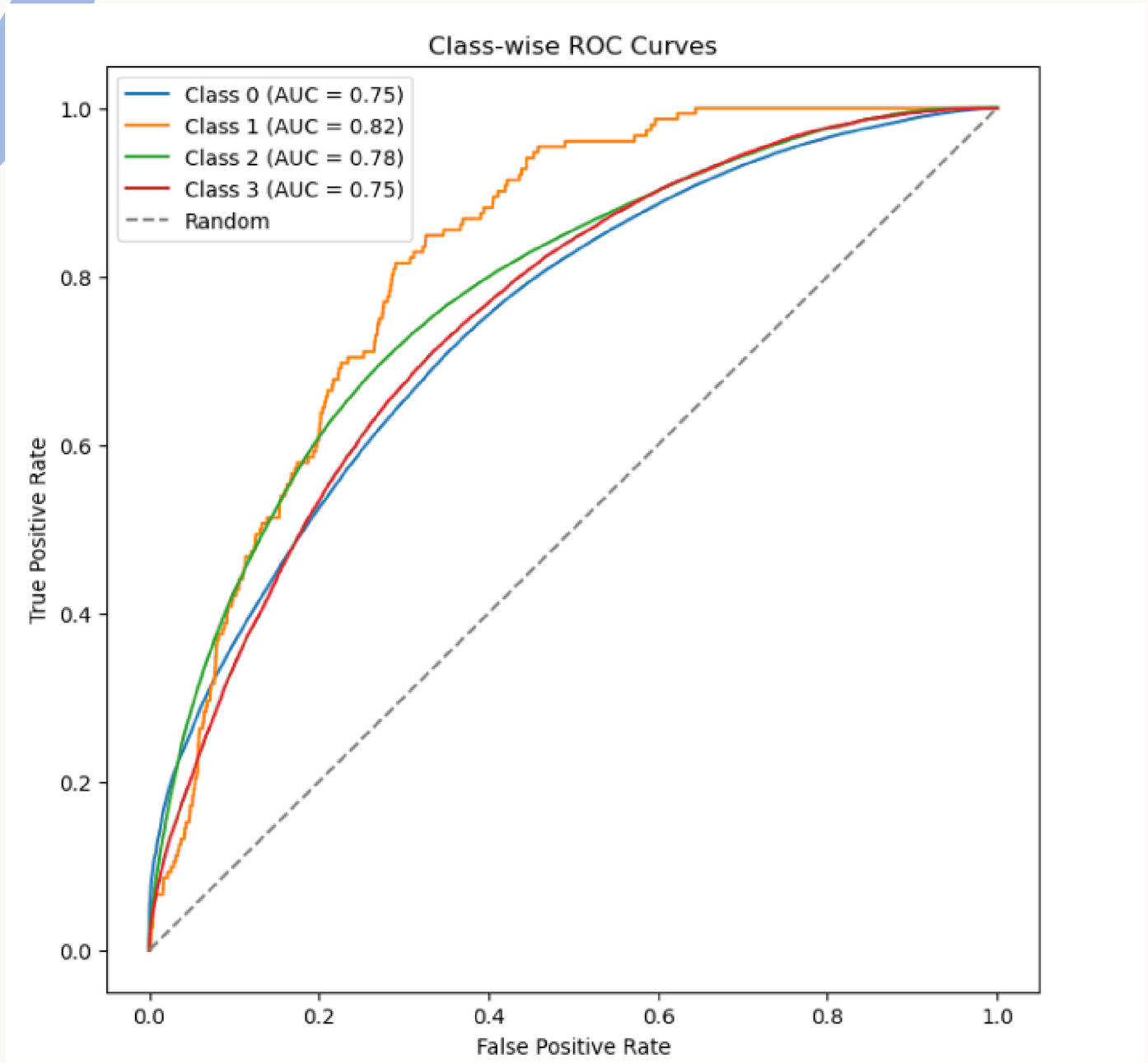
KNN

```
Accuracy: 0.6972089187170584
Confusion Matrix:
[[865855   9561  96889   2491]
 [ 38638   5488  15491   1167]
 [212535   8396 112152   2375]
 [ 25519   2447  12194   1337]]
Classification Report:
              precision    recall  f1-score   support

           0       0.76      0.89      0.82    974796
           1       0.21      0.09      0.13     60784
           2       0.47      0.33      0.39    335458
           3       0.18      0.03      0.05     41497

    accuracy                           0.70   1412535
   macro avg       0.41      0.34      0.35   1412535
weighted avg       0.65      0.70      0.66   1412535
```





CLASS-WISE ROC CURVES

In this analysis, we generated class-wise ROC curves to evaluate the performance of our multi-class classification model. Each curve represents the Receiver Operating Characteristic (ROC) for a specific class, indicating the trade-off between True Positive Rate and False Positive Rate.

REFERENCE

- https://pubmed.ncbi.nlm.nih.gov/32580330/
- https://www.sciencedirect.com/science/article/abs/pii/S0952197622002433
- https://pubmed.ncbi.nlm.nih.gov/37634658/
- https://www.kaggle.com/code/prashant111/lightgbm-classifier-in-python
- https://www.kaggle.com/code/pppppppooijh/lgbm-for-both-defog-tdcsfog/edit

THANKYOU