NAME : RETHINAGIRI G

ROLL NO : 225229130

COURSE TITLE : PRACTICAL MACHINE LEARNING LAB

LAB.10 Patient Physical Activities Prediction using Boosting2

In [1]:
```
import pandas as pan
```

Step-I

In [2]:
```
human=pan.read_csv("C:\\Users\\user\\Downloads\\dataset_pml\\Human_Activity_Data.csv")
```

In [3]:
```
human.head()
```

Out[3]:

	tBodyAcc-mean()-X	tBodyAcc-mean()-Y	tBodyAcc-mean()-Z	tBodyAcc-std()-X	tBodyAcc-std()-Y	tBodyAcc-std()-Z	tBodyAcc-mad()-X	tBodyAcc-mad()-Y	tBodyAcc-mad()-Z	tBodyAcc-max()-X	...	fBodyBodyGyroJerkMag-skewness()
0	0.288585	-0.020294	-0.132905	-0.995279	-0.983111	-0.913526	-0.995112	-0.983185	-0.923527	-0.934724	...	-0.298676
1	0.278419	-0.016411	-0.123520	-0.998245	-0.975300	-0.960322	-0.998807	-0.974914	-0.957686	-0.943068	...	-0.595051
2	0.279653	-0.019467	-0.113462	-0.995380	-0.967187	-0.978944	-0.996520	-0.963668	-0.977469	-0.938692	...	-0.390748
3	0.279174	-0.026201	-0.123283	-0.996091	-0.983403	-0.990675	-0.997099	-0.982750	-0.989302	-0.938692	...	-0.117290
4	0.276629	-0.016570	-0.115362	-0.998139	-0.980817	-0.990482	-0.998321	-0.979672	-0.990441	-0.942469	...	-0.351471

5 rows × 562 columns

In [4]:
```
human.shape
```

Out[4]:
```
(10299, 562)
```

In [5]:
```
human.columns
```

Out[5]:
```
Index(['tBodyAcc-mean()-X', 'tBodyAcc-mean()-Y', 'tBodyAcc-mean()-Z',
       'tBodyAcc-std()-X', 'tBodyAcc-std()-Y', 'tBodyAcc-std()-Z',
       'tBodyAcc-mad()-X', 'tBodyAcc-mad()-Y', 'tBodyAcc-mad()-Z',
       'tBodyAcc-max()-X',
       ...
       'fBodyBodyGyroJerkMag-skewness()', 'fBodyBodyGyroJerkMag-kurtosis()',
       'angle(tBodyAccMean,gravity)', 'angle(tBodyAccJerkMean),gravityMean)',
       'angle(tBodyGyroMean,gravityMean)',
       'angle(tBodyGyroJerkMean,gravityMean)', 'angle(X,gravityMean)',
       'angle(Y,gravityMean)', 'angle(Z,gravityMean)', 'Activity'],
      dtype='object', length=562)
```

In [6]:
```
human.dtypes
```

Out[6]:
```
tBodyAcc-mean()-X                     float64
tBodyAcc-mean()-Y                     float64
tBodyAcc-mean()-Z                     float64
tBodyAcc-std()-X                      float64
tBodyAcc-std()-Y                      float64
                                       ...
angle(tBodyGyroJerkMean,gravityMean)  float64
angle(X,gravityMean)                  float64
angle(Y,gravityMean)                  float64
angle(Z,gravityMean)                  float64
Activity                               object
Length: 562, dtype: object
```

In [7]:
```
human.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10299 entries, 0 to 10298
Columns: 562 entries, tBodyAcc-mean()-X to Activity
dtypes: float64(561), object(1)
memory usage: 44.2+ MB
```

In [8]:
```
human.value_counts
```

Out[8]:
```
<bound method DataFrame.value_counts of       tBodyAcc-mean()-X  tBodyAcc-mean()-Y  tBodyAcc-mean()-Z  \
0                0.288585          -0.020294          -0.132905
1                0.278419          -0.016411          -0.123520
2                0.279653          -0.019467          -0.113462
3                0.279174          -0.026201          -0.123283
4                0.276629          -0.016570          -0.115362
...                   ...                ...                ...
10294            0.310155          -0.053391          -0.099109
10295            0.363385          -0.039214          -0.105915
10296            0.349966           0.030077          -0.115788
10297            0.237594           0.018467          -0.096499
10298            0.153627          -0.018437          -0.137018

       tBodyAcc-std()-X  tBodyAcc-std()-Y  tBodyAcc-std()-Z  tBodyAcc-mad()-X  \
0             -0.995279         -0.983111         -0.913526         -0.995112
1             -0.998245         -0.975300         -0.960322         -0.998807
2             -0.995380         -0.967187         -0.978944         -0.996520
3             -0.996091         -0.983403         -0.990675         -0.997099
4             -0.998139         -0.980817         -0.990482         -0.998321
...                 ...               ...               ...               ...
10294         -0.287866         -0.140589         -0.215088         -0.356083
10295         -0.305388          0.028148         -0.196373         -0.373540
10296         -0.329638         -0.042143         -0.250181         -0.388017
10297         -0.323114         -0.229775         -0.207574         -0.392380
10298         -0.330046         -0.195253         -0.164339         -0.430974

       tBodyAcc-mad()-Y  tBodyAcc-mad()-Z  tBodyAcc-max()-X  ...  \
0             -0.983185         -0.923527         -0.934724  ...
1             -0.974914         -0.957686         -0.943068  ...
2             -0.963668         -0.977469         -0.938692  ...
3             -0.982750         -0.989302         -0.938692  ...
4             -0.979672         -0.990441         -0.942469  ...
...                 ...               ...               ...  ...
10294         -0.148775         -0.232057          0.185361  ...
10295         -0.030036         -0.270237          0.185361  ...
10296         -0.133257         -0.347029          0.007471  ...
10297         -0.279610         -0.289477          0.007471  ...
10298         -0.218295         -0.229933         -0.111527  ...

       fBodyBodyGyroJerkMag-skewness()  fBodyBodyGyroJerkMag-kurtosis()  \
0                            -0.298676                        -0.710304
1                            -0.595051                        -0.861499
2                            -0.390748                        -0.760104
3                            -0.117290                        -0.482845
4                            -0.351471                        -0.699205
...                                ...                              ...
10294                        -0.376278                        -0.750809
```

```
10295                       -0.320418                       -0.700274
10296                       -0.118854                       -0.467179
10297                       -0.205445                       -0.617737
10298                       -0.072237                       -0.436940

       angle(tBodyAccMean,gravity)  angle(tBodyAccJerkMean),gravityMean)  \
0                        -0.112754                             0.030400
1                         0.053477                            -0.007435
2                        -0.118559                             0.177899
3                        -0.036788                            -0.012892
4                         0.123320                             0.122542
...                            ...                                  ...
10294                    -0.337422                             0.346295
10295                    -0.736701                            -0.372889
10296                    -0.181560                             0.088574
10297                     0.444558                            -0.819188
10298                     0.598808                            -0.287951

       angle(tBodyGyroMean,gravityMean)  angle(tBodyGyroJerkMean,gravityMean)  \
0                             -0.464761                             -0.018446
1                             -0.732626                              0.703511
2                              0.100699                              0.808529
3                              0.640011                             -0.485366
4                              0.693578                             -0.615971
...                                 ...                                   ...
10294                          0.884904                             -0.698885
10295                         -0.657421                              0.322549
10296                          0.696663                              0.363139
10297                          0.929294                             -0.008398
10298                          0.876030                             -0.024965

       angle(X,gravityMean)  angle(Y,gravityMean)  angle(Z,gravityMean)  \
0                 -0.841247              0.179941             -0.058627
1                 -0.844788              0.180289             -0.054317
2                 -0.848933              0.180637             -0.049118
3                 -0.848649              0.181935             -0.047663
4                 -0.847865              0.185151             -0.043892
...                     ...                   ...                   ...
10294             -0.651732              0.274627              0.184784
10295             -0.655181              0.273578              0.182412
10296             -0.655357              0.274479              0.181184
10297             -0.659719              0.264782              0.187563
10298             -0.660080              0.263936              0.188103

              Activity
0             STANDING
1             STANDING
2             STANDING
```

```
3               STANDING
4               STANDING
...                  ...
10294  WALKING_UPSTAIRS
10295  WALKING_UPSTAIRS
10296  WALKING_UPSTAIRS
10297  WALKING_UPSTAIRS
10298  WALKING_UPSTAIRS

[10299 rows x 562 columns]>
```

Step-2 [Build a small dataset]

In [9]:
```
human["Activity"].unique()
```

Out[9]:
```
array(['STANDING', 'SITTING', 'LAYING', 'WALKING', 'WALKING_DOWNSTAIRS',
       'WALKING_UPSTAIRS'], dtype=object)
```

In [10]:
```
df = human.loc[human["Activity"].isin(["WALKING", "SITTING", "LAYING"])]
```

In [11]:
```
walk= human.loc[human["Activity"] == "WALKING"].sample(500)
sit= human.loc[human["Activity"] == "SITTING"].sample(500)
lay=human.loc[human["Activity"] == "LAYING"].sample(500)
```

In [12]:
```
nd= pan.concat([walk, sit, lay])
```

In [13]:
```
nd.to_csv("Human.csv",index=False)
```

Step-3 [Build GradientBoostingClassifier]

In [14]:
```
Data=pan.read_csv("Human.csv")
Data.head()
```

Out[14]:

	tBodyAcc-mean()-X	tBodyAcc-mean()-Y	tBodyAcc-mean()-Z	tBodyAcc-std()-X	tBodyAcc-std()-Y	tBodyAcc-std()-Z	tBodyAcc-mad()-X	tBodyAcc-mad()-Y	tBodyAcc-mad()-Z	tBodyAcc-max()-X	...	fBodyBodyGyroJerkMag-skewness()
0	0.223498	-0.004764	-0.054825	-0.298798	-0.115901	-0.317805	-0.336247	-0.124901	-0.333654	0.022884	...	0.550369
1	0.331556	-0.037826	-0.125532	-0.399544	-0.254734	-0.151447	-0.391599	-0.291021	-0.124841	-0.399624	...	-0.159240
2	0.249343	-0.003411	-0.056042	-0.399476	-0.137845	-0.461271	-0.437246	-0.198605	-0.453340	-0.179696	...	-0.495646
3	0.271651	-0.015662	-0.144156	-0.350667	-0.231905	-0.325115	-0.397051	-0.240116	-0.328589	-0.137797	...	-0.326996
4	0.315267	0.016745	-0.138095	-0.382439	-0.157401	-0.563687	-0.403496	-0.147394	-0.568735	-0.228827	...	-0.575934

5 rows × 562 columns

In [15]:
```
Data.shape
```

Out[15]:
```
(1500, 562)
```

In [16]:
```
Data.dtypes
```

Out[16]:
```
tBodyAcc-mean()-X                       float64
tBodyAcc-mean()-Y                       float64
tBodyAcc-mean()-Z                       float64
tBodyAcc-std()-X                        float64
tBodyAcc-std()-Y                        float64
                                          ...
angle(tBodyGyroJerkMean,gravityMean)    float64
angle(X,gravityMean)                    float64
angle(Y,gravityMean)                    float64
angle(Z,gravityMean)                    float64
Activity                                 object
Length: 562, dtype: object
```

In [17]:
```
Data.value_counts
```

Out[17]:
```
<bound method DataFrame.value_counts of       tBodyAcc-mean()-X  tBodyAcc-mean()-Y  tBodyAcc-mean()-Z  \
0              0.223498          -0.004764          -0.054825
1              0.331556          -0.037826          -0.125532
2              0.249343          -0.003411          -0.056042
3              0.271651          -0.015662          -0.144156
4              0.315267           0.016745          -0.138095
...                 ...                ...                ...
1495           0.284841          -0.019201          -0.107611
1496           0.278778           0.001278          -0.129886
1497          -0.503823          -0.594207           0.264804
1498           0.273700          -0.012533          -0.114728
1499           0.264384          -0.017022          -0.107920

      tBodyAcc-std()-X  tBodyAcc-std()-Y  tBodyAcc-std()-Z  tBodyAcc-mad()-X  \
0            -0.298798         -0.115901         -0.317805         -0.336247
1            -0.399544         -0.254734         -0.151447         -0.391599
2            -0.399476         -0.137845         -0.461271         -0.437246
3            -0.350667         -0.231905         -0.325115         -0.397051
4            -0.382439         -0.157401         -0.563687         -0.403496
...                ...               ...               ...               ...
1495         -0.972128         -0.995757         -0.991906         -0.972685
1496         -0.993938         -0.957732         -0.975728         -0.994663
1497         -0.703402          0.672487         -0.464985         -0.704548
1498         -0.983734         -0.970662         -0.981009         -0.986951
1499         -0.949239         -0.971667         -0.971883         -0.947305

      tBodyAcc-mad()-Y  tBodyAcc-mad()-Z  tBodyAcc-max()-X  ...  \
0            -0.124901         -0.333654          0.022884  ...
1            -0.291021         -0.124841         -0.399624  ...
2            -0.198605         -0.453340         -0.179696  ...
3            -0.240116         -0.328589         -0.137797  ...
4            -0.147394         -0.568735         -0.228827  ...
...                ...               ...               ...  ...
1495         -0.994660         -0.992572         -0.912239  ...
1496         -0.952886         -0.972024         -0.934344  ...
1497          0.967664         -0.401827         -0.948813  ...
1498         -0.974966         -0.981606         -0.908698  ...
1499         -0.968391         -0.970534         -0.904388  ...

      fBodyBodyGyroJerkMag-skewness()  fBodyBodyGyroJerkMag-kurtosis()  \
0                            0.550369                         0.344301
1                           -0.159240                        -0.604960
2                           -0.495646                        -0.808660
3                           -0.326996                        -0.706172
4                           -0.575934                        -0.884190
...                               ...                              ...
1495                        -0.851367                        -0.981048
```

```
1496                        -0.640985                        -0.894815
1497                        -0.412991                        -0.718527
1498                         0.617445                         0.439463
1499                        -0.284005                        -0.700522

      angle(tBodyAccMean,gravity)  angle(tBodyAccJerkMean),gravityMean)  \
0                        0.476859                             0.083179
1                       -0.645232                            -0.647860
2                        0.258529                             0.038298
3                       -0.047375                             0.927682
4                       -0.069911                            -0.250392
...                           ...                                  ...
1495                     0.205729                            -0.211422
1496                     0.027342                             0.020652
1497                     0.249525                             0.440171
1498                     0.043347                            -0.532228
1499                     0.002422                            -0.365710

      angle(tBodyGyroMean,gravityMean)  angle(tBodyGyroJerkMean,gravityMean)  \
0                            -0.093855                             -0.775732
1                             0.489015                             -0.798114
2                             0.904697                             -0.421875
3                             0.933117                              0.216304
4                            -0.412534                              0.603081
...                                ...                                   ...
1495                          0.395310                             -0.229736
1496                          0.110563                              0.167558
1497                         -0.007508                              0.343715
1498                          0.542719                             -0.023971
1499                          0.554906                              0.262370

      angle(X,gravityMean)  angle(Y,gravityMean)  angle(Z,gravityMean)  \
0                -0.721141             0.265173             0.116227
1                -0.675771             0.231274             0.203654
2                -0.670009             0.325729             0.004459
3                -0.716721             0.235446             0.161681
4                -0.750986             0.257155            -0.049705
...                    ...                  ...                  ...
1495              0.539980            -0.488183            -0.517488
1496              0.407937            -0.315549            -0.673453
1497              0.683948            -0.505728            -0.495238
1498              0.612304            -0.363663            -0.648593
1499              0.545731            -0.234862            -0.780867

      Activity
0      WALKING
1      WALKING
2      WALKING
```

```
3      WALKING
4      WALKING
...        ...
1495    LAYING
1496    LAYING
1497    LAYING
1498    LAYING
1499    LAYING

[1500 rows x 562 columns]>
```

In [18]:
```
Data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1500 entries, 0 to 1499
Columns: 562 entries, tBodyAcc-mean()-X to Activity
dtypes: float64(561), object(1)
memory usage: 6.4+ MB
```

In [19]:
```
Data.columns
```

Out[19]:
```
Index(['tBodyAcc-mean()-X', 'tBodyAcc-mean()-Y', 'tBodyAcc-mean()-Z',
       'tBodyAcc-std()-X', 'tBodyAcc-std()-Y', 'tBodyAcc-std()-Z',
       'tBodyAcc-mad()-X', 'tBodyAcc-mad()-Y', 'tBodyAcc-mad()-Z',
       'tBodyAcc-max()-X',
       ...
       'fBodyBodyGyroJerkMag-skewness()', 'fBodyBodyGyroJerkMag-kurtosis()',
       'angle(tBodyAccMean,gravity)', 'angle(tBodyAccJerkMean),gravityMean)',
       'angle(tBodyGyroMean,gravityMean)',
       'angle(tBodyGyroJerkMean,gravityMean)', 'angle(X,gravityMean)',
       'angle(Y,gravityMean)', 'angle(Z,gravityMean)', 'Activity'],
      dtype='object', length=562)
```

In [20]:
```
from sklearn.model_selection import train_test_split as tts
from sklearn.ensemble import GradientBoostingClassifier as GBC
from sklearn.metrics import*
```

In [21]:
```
X=Data.drop("Activity",axis=1)
y=Data["Activity"]
```

In [22]:
```
X_train,X_test,y_train,y_test=tts(X,y,test_size=.3,random_state=42)
len(X_train)
len(y_train)
```

Out[22]:
```
1050
```

In [23]:
```
gbc=GBC(n_estimators=100, learning_rate=0.1, max_depth=3,random_state=42)
```

In [24]:
```
gbc.fit(X_train,y_train)
```

Out[24]:
```
GradientBoostingClassifier(random_state=42)
```

In [25]:
```
y_pred=gbc.predict(X_test)
```

In [26]:
```
accuracy_score(y_test,y_pred)
```

Out[26]:
```
1.0
```

In [27]:
```
print(classification_report(y_test,y_pred),'/n')
```

```
              precision    recall  f1-score   support

      LAYING       1.00      1.00      1.00       161
     SITTING       1.00      1.00      1.00       141
     WALKING       1.00      1.00      1.00       148

    accuracy                           1.00       450
   macro avg       1.00      1.00      1.00       450
weighted avg       1.00      1.00      1.00       450
 /n
```

Step-4 [Find Best no. of Trees and Best Learning Rate using Grid Search and Cross Validation]

In [28]:
```
# Create GridSearchCV model with GradientBoostingClassifier

from sklearn.model_selection import GridSearchCV as gsc
```

In [29]:
```
#parameters

param_grid={'n_estimators':[50,100,200,400],'learning_rate':[0.1,0.01]}
```

In [30]:
```
grid= gsc(gbc, param_grid, cv=5,verbose=3, n_jobs=-1)
```

In [31]:
```
# Fit the GridSearchCV model on the training data
grid.fit(X_train, y_train)
```

```
Fitting 5 folds for each of 8 candidates, totalling 40 fits
```

Out[31]:
```
GridSearchCV(cv=5, estimator=GradientBoostingClassifier(random_state=42),
             n_jobs=-1,
             param_grid={'learning_rate': [0.1, 0.01],
                         'n_estimators': [50, 100, 200, 400]},
             verbose=3)
```

In [32]:
```
#predict the data

y1_pred=grid.predict(X_test)
```

In [33]:
```
accuracy_score(y_test,y1_pred)
```

Out[33]:
```
1.0
```

In [34]:
```
print(classification_report(y_test,y1_pred),'/n')
```

```
              precision    recall  f1-score   support

      LAYING       1.00      1.00      1.00       161
     SITTING       1.00      1.00      1.00       141
     WALKING       1.00      1.00      1.00       148

    accuracy                           1.00       450
   macro avg       1.00      1.00      1.00       450
weighted avg       1.00      1.00      1.00       450
 /n
```

In [35]:
```
# Retrieve the best hyperparameters
best_params = grid.best_params_
print("Best hyperparameters:", best_params)
```

```
Best hyperparameters: {'learning_rate': 0.01, 'n_estimators': 50}
```

Step-5 [Build AdaBoostClassifier]

Create AdaBoostClassifier with DecisionTreeClassifier

In [58]:
```
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import AdaBoostClassifier
```

In [59]:
```
param_grid = {'n_estimators': [50, 10, 200, 400],'learning_rate': [0.1, 0.01]}
```

In [60]:
```
ada= AdaBoostClassifier(base_estimator=DecisionTreeClassifier())
```

In [62]:
```
abc= gsc(estimator=ada, param_grid=param_grid, cv=5)
```

In [63]:
```
# Fit the AdaBoostClassifier model on the training data
abc.fit(X_train, y_train)
```

Out[63]:
```
GridSearchCV(cv=5,
             estimator=AdaBoostClassifier(base_estimator=DecisionTreeClassifier()),
             param_grid={'learning_rate': [0.1, 0.01],
                         'n_estimators': [50, 10, 200, 400]})
```

In [64]:
```
# Use the AdaBoostClassifier model to make predictions on the testing data
y2_pred = abc.predict(X_test)
```

In [65]:
```
accuracy_score(y_test,y2_pred)
```

Out[65]:
```
1.0
```

In [66]:
```
print(classification_report(y_test,y2_pred),'/n')
```

```
              precision    recall  f1-score   support

      LAYING       1.00      1.00      1.00       161
     SITTING       1.00      1.00      1.00       141
     WALKING       1.00      1.00      1.00       148

    accuracy                           1.00       450
   macro avg       1.00      1.00      1.00       450
weighted avg       1.00      1.00      1.00       450
 /n
```

In [70]:
```
best_params = abc.best_estimator_
print("Best hyperparameters:", best_params)
```

```
Best hyperparameters: AdaBoostClassifier(base_estimator=DecisionTreeClassifier(), learning_rate=0.01,
                   n_estimators=10)
```

Step-6 [Build LogisticRegressionCV classifer]

In [48]:
```
from sklearn.linear_model import LogisticRegressionCV as lr
```

In [73]:
```
clf =lr(Cs=5, cv=4, penalty='l2')
clf.fit(X_train, y_train)
y_pred = clf.predict(X_test)
```

...

In [74]:
```
print(classification_report(y_test, y_pred))
```

```
              precision    recall  f1-score   support

      LAYING       1.00      1.00      1.00       161
     SITTING       1.00      0.99      1.00       141
     WALKING       0.99      1.00      1.00       148

    accuracy                           1.00       450
   macro avg       1.00      1.00      1.00       450
weighted avg       1.00      1.00      1.00       450
```

Step-7[Build VotingClassifier]

In [108]:
```
from sklearn.ensemble import  VotingClassifier as vc
```

In [109]:
```
gbc = GBC(n_estimators=100, learning_rate=0.1, random_state=42)
lrc = LogisticRegression(penalty='none',solver='saga')
vc = vc(estimators=[('gbc', gbc), ('lrc', lrc)], voting='soft')
```

In [110]:
```
vc.fit(X_train, y_train)
```

```
C:\Users\user\anaconda3\lib\site-packages\sklearn\linear_model\_sag.py:352: ConvergenceWarning: The max_iter was reach
ed which means the coef_ did not converge
  warnings.warn(
```

Out[110]:
```
VotingClassifier(estimators=[('gbc',
                              GradientBoostingClassifier(random_state=42)),
                             ('lrc',
                              LogisticRegression(penalty='none',
                                                 solver='saga'))],
                 voting='soft')
```

In [111]:
```
vc_pred=vc.predict(X_test)
```

In [113]:
```
print("Accuracy_Score",accuracy_score(y_test,vc_pred))
print("VotingClassifier:")
print(classification_report(y_test, vc_pred))
```

```
Accuracy_Score 0.9977777777777778
VotingClassifier:
              precision    recall  f1-score   support

      LAYING       1.00      1.00      1.00       161
     SITTING       1.00      0.99      1.00       141
     WALKING       0.99      1.00      1.00       148

    accuracy                           1.00       450
   macro avg       1.00      1.00      1.00       450
weighted avg       1.00      1.00      1.00       450
```

Step-8[Interpret your result]

Analyze your results

```
The models we have built have achieved decent accuracy and classification reports.
```