IDVE_EXAM_Q2

December 10, 2021

Name: Riekert Holder Student Number: 2517888

[ ]:
```
from google.colab import drive
drive.mount('/content/drive')
```

```
Mounted at /content/drive
```

[ ]:
```
!pip install umap-learn minisom
```

```
Collecting umap-learn
  Downloading umap-learn-0.5.2.tar.gz (86 kB)
     || 86 kB 4.9 MB/s
Collecting minisom
  Downloading MiniSom-2.2.9.tar.gz (8.1 kB)
Requirement already satisfied: numpy>=1.17 in /usr/local/lib/python3.7/dist-
packages (from umap-learn) (1.19.5)
Requirement already satisfied: scikit-learn>=0.22 in /usr/local/lib/python3.7
/dist-packages (from umap-learn) (1.0.1)
Requirement already satisfied: scipy>=1.0 in /usr/local/lib/python3.7/dist-
packages (from umap-learn) (1.4.1)
Requirement already satisfied: numba>=0.49 in /usr/local/lib/python3.7/dist-
packages (from umap-learn) (0.51.2)
Collecting pynndescent>=0.5
  Downloading pynndescent-0.5.5.tar.gz (1.1 MB)
     || 1.1 MB 37.8 MB/s
Requirement already satisfied: tqdm in /usr/local/lib/python3.7/dist-
packages (from umap-learn) (4.62.3)
Requirement already satisfied: llvmlite<0.35,>=0.34.0.dev0 in
/usr/local/lib/python3.7/dist-packages (from numba>=0.49->umap-learn) (0.34.0)
Requirement already satisfied: setuptools in /usr/local/lib/python3.7/dist-
packages (from numba>=0.49->umap-learn) (57.4.0)
Requirement already satisfied: joblib>=0.11 in /usr/local/lib/python3.7/dist-
packages (from pynndescent>=0.5->umap-learn) (1.1.0)
Requirement already satisfied: threadpoolctl>=2.0.0 in /usr/local/lib/python3.7
/dist-packages (from scikit-learn>=0.22->umap-learn) (3.0.0)
Building wheels for collected packages: umap-learn, pynndescent, minisom
  Building wheel for umap-learn (setup.py) ... done
  Created wheel for umap-learn: filename=umap_learn-0.5.2-py3-none-any.whl
```

```
size=82709
sha256=07a14278ebda7412d6f86a2c26635408def09c968114245282083e8f98c29fb4
  Stored in directory: /root/.cache/pip/wheels/84/1b/c6/aaf68a748122632967cef4df
fef68224eb16798b6793257d82
  Building wheel for pynndescent (setup.py) ... done
  Created wheel for pynndescent: filename=pynndescent-0.5.5-py3-none-any.whl
size=52603
sha256=dbdc0d1d818c23191ee441e0fdce942388f0fd1127886ce6eb715719e93b670e
  Stored in directory: /root/.cache/pip/wheels/af/e9/33/04db1436df0757c42fda8ea6
796d7a8586e23c85fac355f476
  Building wheel for minisom (setup.py) ... done
  Created wheel for minisom: filename=MiniSom-2.2.9-py3-none-any.whl size=8594
sha256=c9e4dea7b977194e21e65b0fad09119daf645f14f78d0b42402758bd826d559b
  Stored in directory: /root/.cache/pip/wheels/3d/a1/10/f50b6f4865652eac239a2700
de411c3078c27e1318320e494c
Successfully built umap-learn pynndescent minisom
Installing collected packages: pynndescent, umap-learn, minisom
Successfully installed minisom-2.2.9 pynndescent-0.5.5 umap-learn-0.5.2
```

[ ]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.manifold import TSNE
from sklearn.preprocessing import StandardScaler, LabelEncoder
from matplotlib.cm import get_cmap
from matplotlib.colors import rgb2hex
from sklearn.ensemble import RandomForestClassifier
from sklearn.linear_model import LogisticRegression
from sklearn.svm import SVC
from sklearn.metrics import f1_score
from datetime import datetime
from sklearn.feature_selection import mutual_info_classif, f_classif
```

0.1 2.1 Exploration

0.1.1 2.1.1 Describe the Dataset

[ ]:
```
########################### READING DATA ###################################

########################### TRAIN DATA #####################################
#read the dataset and seperate data with delim_whitespace
X_train = pd.read_csv('/content/drive/MyDrive/IDVE_Exam/UCI DATASET/train/
↪X_train.txt', delim_whitespace=True, header=None)
#get the features from features.txt
feats = []
with open('/content/drive/MyDrive/IDVE_Exam/UCI DATASET/features.txt') as file:
```

```
    feats = [line.split()[1] for line in file.readlines()]
#add features
X_train.columns = [feats]
#add subject data
X_train['subject'] = pd.read_csv('/content/drive/MyDrive/IDVE_Exam/UCI DATASET/
 ↪train/subject_train.txt', header=None, squeeze=True)
#get y values and create y_train
y_train = pd.read_csv('/content/drive/MyDrive/IDVE_Exam/UCI DATASET/train/
 ↪y_train.txt', names=['Activity'], squeeze=True)
y_train_labels = y_train.map({1: 'WALKING', 2:'WALKING_UPSTAIRS',3:
 ↪'WALKING_DOWNSTAIRS',4:'SITTING', 5:'STANDING',6:'LAYING'})
#combine X and Y datasets to make overall training set
train = X_train
train['Activity'] = y_train
train['ActivityName'] = y_train_labels
train.to_csv('/content/drive/MyDrive/IDVE_Exam/UCI DATASET/train.csv',␣
 ↪index=False)
train = pd.read_csv('/content/drive/MyDrive/IDVE_Exam/UCI DATASET/train.csv')
############################################################################


########################### TEST DATA ######################################
#read the dataset and seperate data with delim_whitespace
X_test = pd.read_csv('/content/drive/MyDrive/IDVE_Exam/UCI DATASET/test/X_test.
 ↪txt', delim_whitespace=True, header=None)
X_test.columns = [feats]
#add subject data
X_test['subject'] = pd.read_csv('/content/drive/MyDrive/IDVE_Exam/UCI DATASET/
 ↪test/subject_test.txt', header=None, squeeze=True)

#get y values and create y_train
y_test = pd.read_csv('/content/drive/MyDrive/IDVE_Exam/UCI DATASET/test/y_test.
 ↪txt', names=['Activity'], squeeze=True)
y_test_labels = y_test.map({1: 'WALKING', 2:'WALKING_UPSTAIRS',3:
 ↪'WALKING_DOWNSTAIRS',4:'SITTING', 5:'STANDING',6:'LAYING'})

#combine X and Y datasets to make overall training set
test = X_test
test['Activity'] = y_test
test['ActivityName'] = y_test_labels

test.to_csv('/content/drive/MyDrive/IDVE_Exam/UCI DATASET/test.csv',␣
 ↪index=False)
test = pd.read_csv('/content/drive/MyDrive/IDVE_Exam/UCI DATASET/test.csv')
```

[ ]:
```
train.shape, test.shape
```

[ ]: ((7352, 564), (2947, 564))

[ ]:
```
train.describe()
```

[ ]:
```
       tBodyAcc-mean()-X  tBodyAcc-mean()-Y  ...      subject     Activity
count        7352.000000        7352.000000  ...  7352.000000  7352.000000
mean            0.274488          -0.017695  ...    17.413085     3.643362
std             0.070261           0.040811  ...     8.975143     1.744802
min            -1.000000          -1.000000  ...     1.000000     1.000000
25%             0.262975          -0.024863  ...     8.000000     2.000000
50%             0.277193          -0.017219  ...    19.000000     4.000000
75%             0.288461          -0.010783  ...    26.000000     5.000000
max             1.000000           1.000000  ...    30.000000     6.000000

[8 rows x 563 columns]
```

[ ]:
```
test.describe()
```

[ ]:
```
       tBodyAcc-mean()-X  tBodyAcc-mean()-Y  ...      subject     Activity
count        2947.000000        2947.000000  ...  2947.000000  2947.000000
mean            0.273996          -0.017863  ...    12.986427     3.577876
std             0.060570           0.025745  ...     6.950984     1.740348
min            -0.592004          -0.362884  ...     2.000000     1.000000
25%             0.262075          -0.024961  ...     9.000000     2.000000
50%             0.277113          -0.016967  ...    12.000000     4.000000
75%             0.288097          -0.010143  ...    18.000000     5.000000
max             0.671887           0.246106  ...    24.000000     6.000000

[8 rows x 563 columns]
```

TRAIN DATA: Our dataset has 7352 entries and 564 columns of data.The last 3 columns contain data regarding who the person is and the activities that were picked up.Since we have 564 columns of data it will take alot of typing to show the range in each feature and their mean's so the above output will be sufficient enoughThe min of all features are similar, which is -1.0 or in a range between -0.9 -> -1.0 and the maximums range from 0.9 - > 1.0The mean activity is 3.64 -> 4.0 which is sitting TEST DATA: The Test data has 2947 rows and 564 columns of data.The last 3 columns contain data regarding who the person is and the activities that were picked up.The ranges stated above of the train data is the same as the test data

0.1.2 2.1.2 Missing values or Duplicates

[ ]:
```
miss_vals = train.isna().sum().sum()
print(f"We have {miss_vals} missing values in our training dataset")
```

```
We have 0 missing values in our training dataset
```

[ ]:
```
dup_vals = train.duplicated().sum()
print(f"We have {dup_vals} duplicates in our training dataset")
```

```
We have 0 duplicates in our training dataset
```

We have no missing values or duplicates in our training dataset

0.1.3 2.1.3 Class and user visualizations

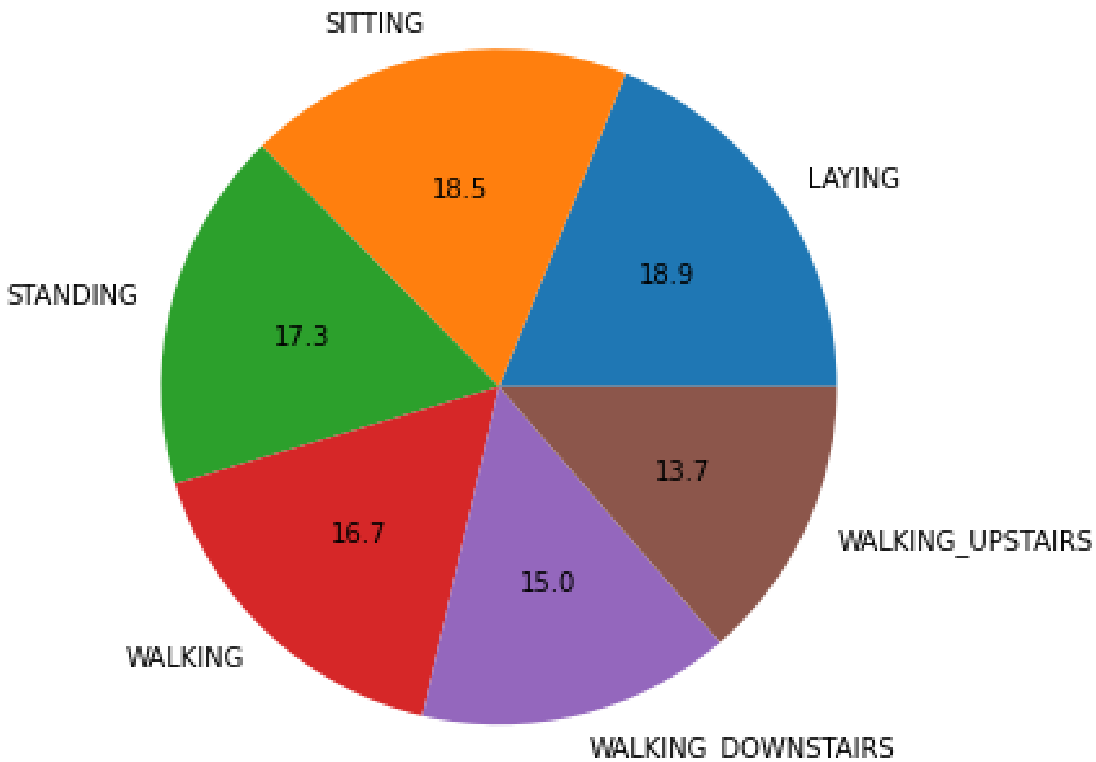
Since nothing was said whether we need to only explore the train or test data,I will assume we need to concatenate the data and explore it as one dataset

[ ]:
```
both_df = pd.concat([train, test], axis=0).reset_index(drop=True)
```

How many of each class:

[ ]:
```
plt.figure(figsize=(14,6))
plt.pie(np.array(both_df.iloc[:,-1].value_counts()),labels=sorted(both_df.iloc[:
↪,-1].unique()), autopct = '%0.1f')
```

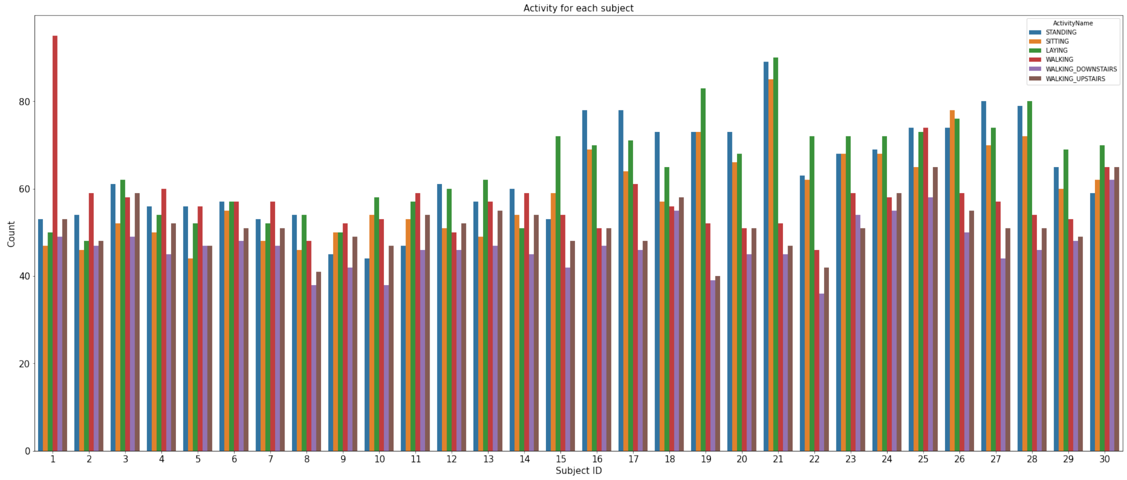
[ ]:
```
([<matplotlib.patches.Wedge at 0x7ff85e335f10>,
  <matplotlib.patches.Wedge at 0x7ff85e335810>,
  <matplotlib.patches.Wedge at 0x7ff85e32af90>,
  <matplotlib.patches.Wedge at 0x7ff85e31efd0>,
  <matplotlib.patches.Wedge at 0x7ff85e393cd0>,
  <matplotlib.patches.Wedge at 0x7ff85e386cd0>],
 [Text(0.912197696055357, 0.6147319442743304, 'LAYING'),
  Text(-0.21486667196758633, 1.0788106012074472, 'SITTING'),
  Text(-1.065601489034256, 0.2729349126952738, 'STANDING'),
  Text(-0.7531828672782088, -0.8016954337144352, 'WALKING'),
  Text(0.2636606321506688, -1.0679340199908935, 'WALKING_DOWNSTAIRS'),
  Text(1.000373282280183, -0.4574421232243195, 'WALKING_UPSTAIRS')],
 [Text(0.49756237966655836, 0.3353083332405438, '18.9'),
  Text(-0.1172000028914107, 0.5884421461131529, '18.5'),
  Text(-0.5812371758368668, 0.1488735887428766, '17.3'),
  Text(-0.41082701851538656, -0.43728841838969185, '16.7'),
  Text(0.14381489026400116, -0.5825094654495782, '15.0'),
  Text(0.5456581539710088, -0.24951388539508332, '13.7')])
```



Most participants in the study were laying, sitting and standing The ranking for the top 3 most activities are: 1. Laying - 18.9% 2. Sitting - 18.5% 3. Standing - 17.3%

How many for each user

[ ]:
```
fig = plt.figure(figsize = (25, 10))
ax = fig.add_axes([0,0,1,1])
ax.set_title("Activity for each subject", fontsize = 15)
plt.tick_params(labelsize = 15)
sns.countplot(x = "subject", hue = "ActivityName", data = both_df)
plt.xlabel("Subject ID", fontsize = 15)
plt.ylabel("Count", fontsize = 15)
plt.show()
```



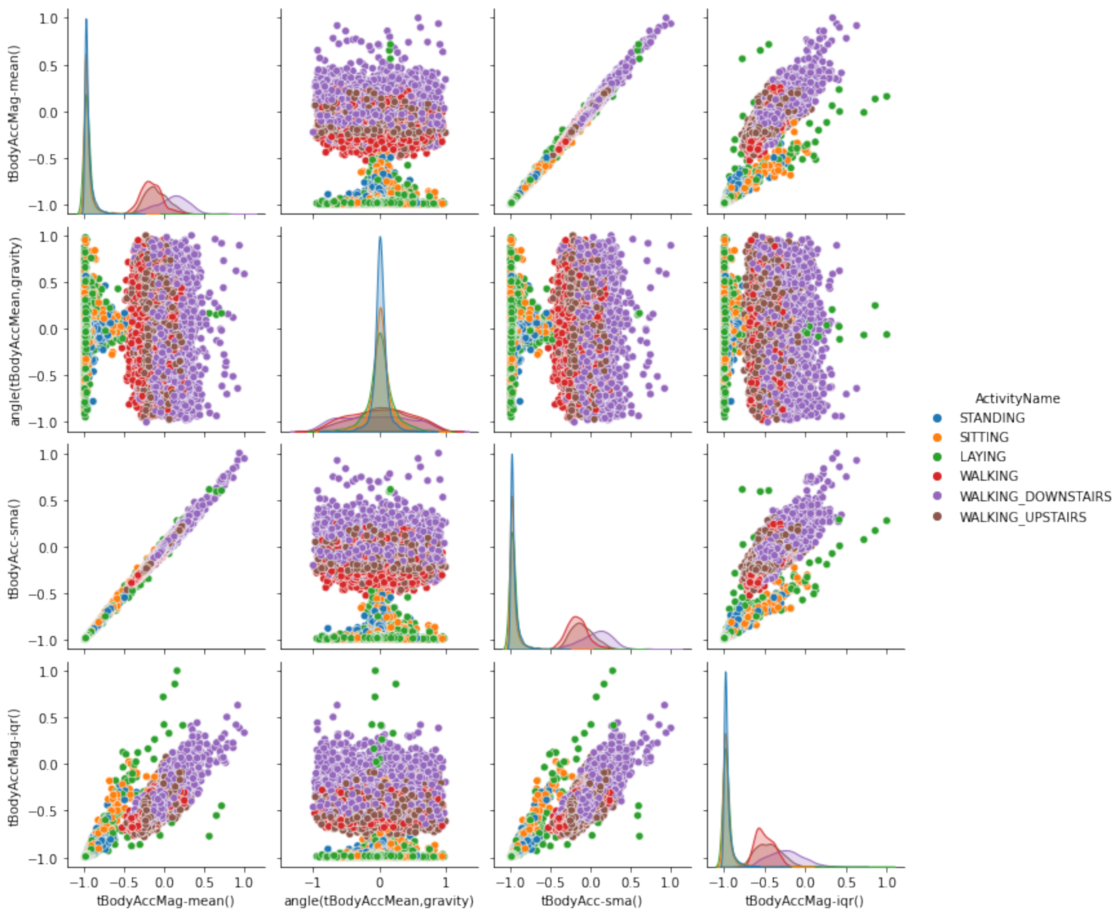
From the barplot above we can see the counts of the activities done by each subject.

0.1.4 2.1.4 Accelerometer readings for all classes

[ ]:
```
features =␣
↪['tBodyAccMag-mean()','ActivityName','angle(tBodyAccMean,gravity)','tBodyAcc-sma()','tBodyA
subset = both_df[features]
print(subset.shape)
sns.pairplot(subset, hue='ActivityName')
```

```
(10299, 5)
```

[ ]: <seaborn.axisgrid.PairGrid at 0x7ff85e2cc210>



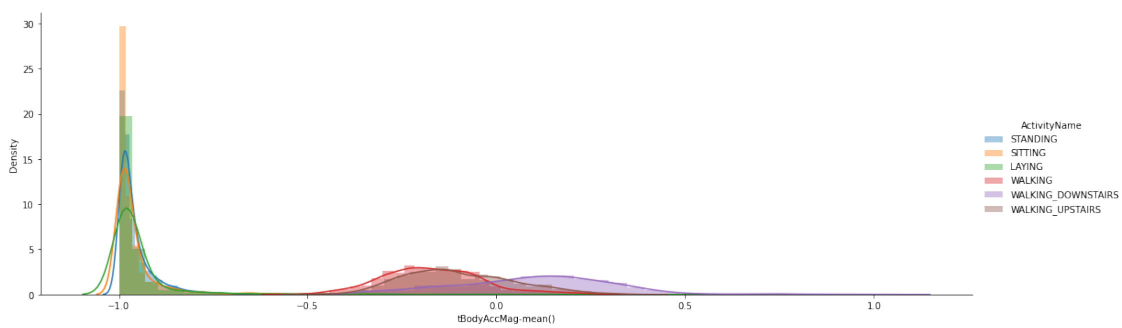
After testing some of the features, we found that the tBodyAccMag-mean() feature shows clear seperation between the accelorometer readings of the classes.The features tBodyAcc-sma() and tBodyAccMag-iqr() could also be used to show this seperation

[ ]:
```
g=sns.FacetGrid(both_df,hue='ActivityName',height=5,aspect=3)
g.map(sns.distplot,'tBodyAccMag-mean()').add_legend()
```

```
/usr/local/lib/python3.7/dist-packages/seaborn/distributions.py:2619:
FutureWarning: `distplot` is a deprecated function and will be removed in a
future version. Please adapt your code to use either `displot` (a figure-level
function with similar flexibility) or `histplot` (an axes-level function for
histograms).
  warnings.warn(msg, FutureWarning)
/usr/local/lib/python3.7/dist-packages/seaborn/distributions.py:2619:
FutureWarning: `distplot` is a deprecated function and will be removed in a
future version. Please adapt your code to use either `displot` (a figure-level
function with similar flexibility) or `histplot` (an axes-level function for
histograms).
  warnings.warn(msg, FutureWarning)
/usr/local/lib/python3.7/dist-packages/seaborn/distributions.py:2619:
```

```
FutureWarning: `distplot` is a deprecated function and will be removed in a
future version. Please adapt your code to use either `displot` (a figure-level
function with similar flexibility) or `histplot` (an axes-level function for
histograms).
  warnings.warn(msg, FutureWarning)
/usr/local/lib/python3.7/dist-packages/seaborn/distributions.py:2619:
FutureWarning: `distplot` is a deprecated function and will be removed in a
future version. Please adapt your code to use either `displot` (a figure-level
function with similar flexibility) or `histplot` (an axes-level function for
histograms).
  warnings.warn(msg, FutureWarning)
/usr/local/lib/python3.7/dist-packages/seaborn/distributions.py:2619:
FutureWarning: `distplot` is a deprecated function and will be removed in a
future version. Please adapt your code to use either `displot` (a figure-level
function with similar flexibility) or `histplot` (an axes-level function for
histograms).
  warnings.warn(msg, FutureWarning)
/usr/local/lib/python3.7/dist-packages/seaborn/distributions.py:2619:
FutureWarning: `distplot` is a deprecated function and will be removed in a
future version. Please adapt your code to use either `displot` (a figure-level
function with similar flexibility) or `histplot` (an axes-level function for
histograms).
  warnings.warn(msg, FutureWarning)
```

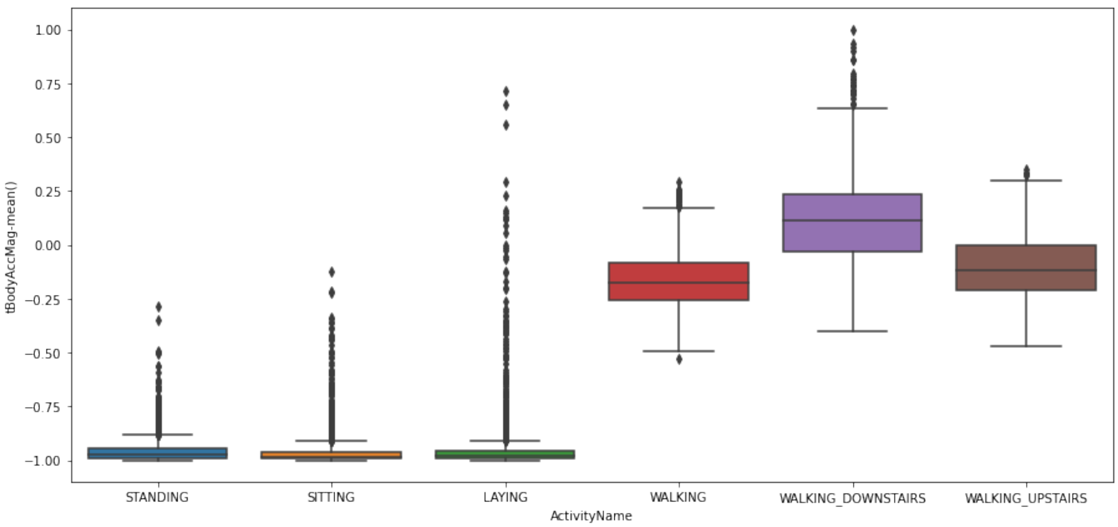
[ ]: <seaborn.axisgrid.FacetGrid at 0x7ff851854790>



Above is a plot that shows a clear static/dynamic separation, our dynamic accelerometer reading are on the left side of the plot and our static ones on the right hand side.When participants are moving the data is normally distributed with some long tail.

[ ]:
```python
plt.figure(figsize=(15,7))
sns.boxplot(x='ActivityName',y='tBodyAccMag-mean()',data=both_df)
```

[ ]: <matplotlib.axes._subplots.AxesSubplot at 0x7ff84ff41290>



Our boxplots also show a seperation between our dynamic and static data.Our dynamic data has a much higher BodyAccMag-mean() than the static data,and this is true since dynamic movements have a higher Body acceleration Mag.Our static categories also has more outliers than the dynamic categories

0.1.5 2.1.5 Rule(if/else)

- **If BodyACCMean < -0.75 then the activities are Standing, sitting or laying down**
- **If BodyACCMean > 0.60 then the activities are either Walking_Upstairs, Walking_Downstairs or Walking**
- **If BodyACCMean > 0 then the activity is Walking_Downstairs**

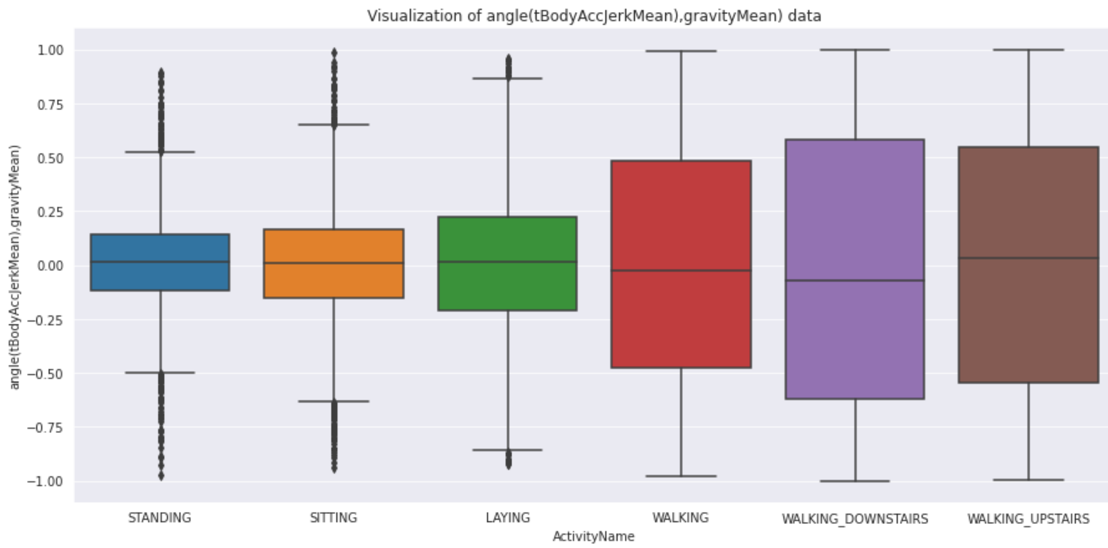
incorrect classification can occur if we classify activities like this, beacuse we do have outliers

0.1.6 2.1.6 Exploit Laying down class

One of the only ways i can think of that could exploit laying down to have more movement is with angles.So i will plot multiple barplots with angle data

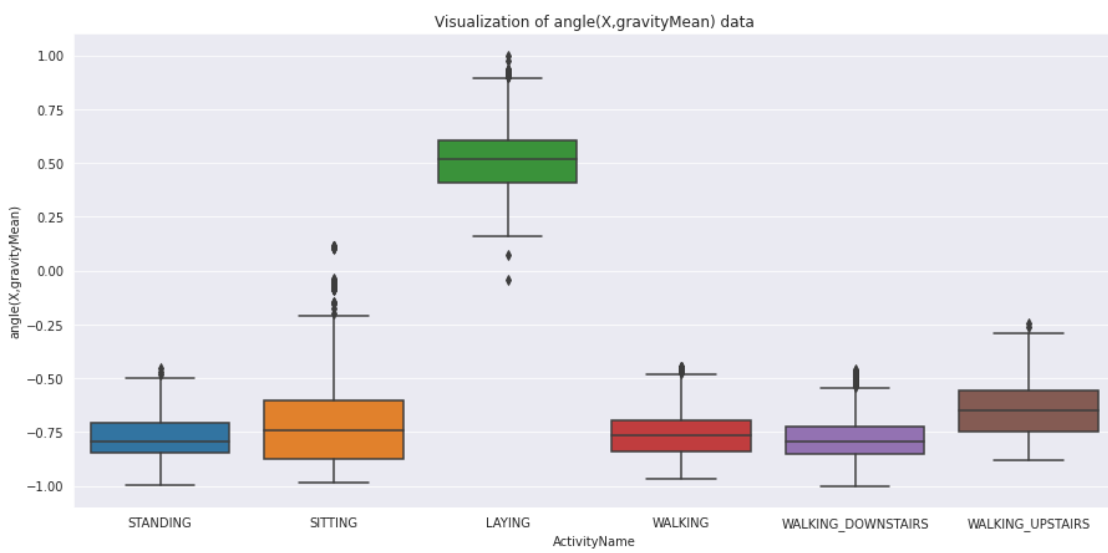
[ ]:
```
plt.figure(figsize=(15,7))
plt.title("Visualization of angle(tBodyAccJerkMean),gravityMean) data")
sns.
↪boxplot(x='ActivityName',y='angle(tBodyAccJerkMean),gravityMean)',data=both_df)
```

[ ]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7ff7dfb0d750>
```



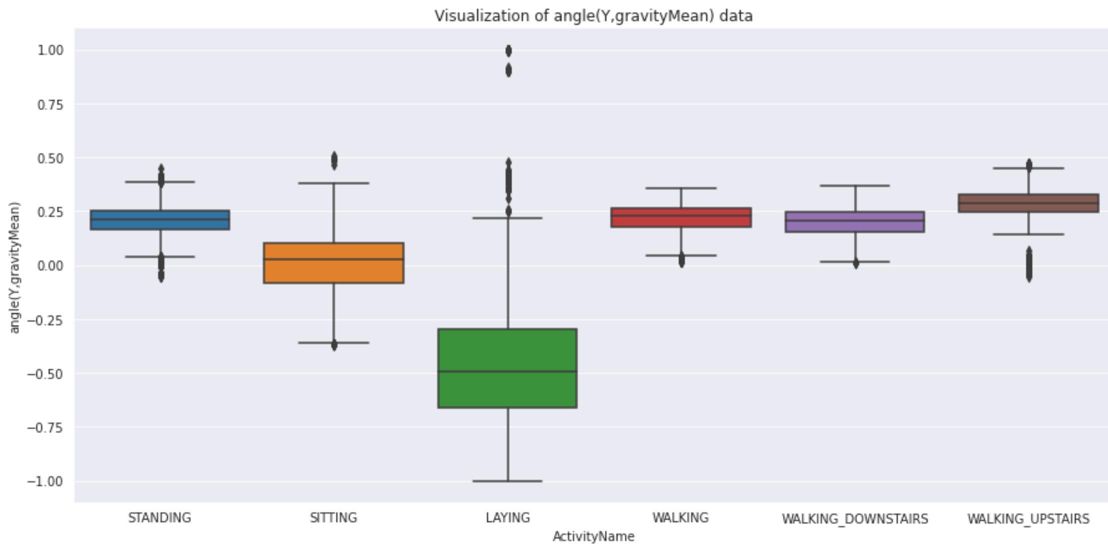
[ ]:
```
plt.figure(figsize=(15,7))
plt.title("Visualization of angle(X,gravityMean) data")
sns.boxplot(x='ActivityName',y='angle(X,gravityMean)',data=both_df)
```

[ ]: <matplotlib.axes._subplots.AxesSubplot at 0x7ff7dc2f9590>



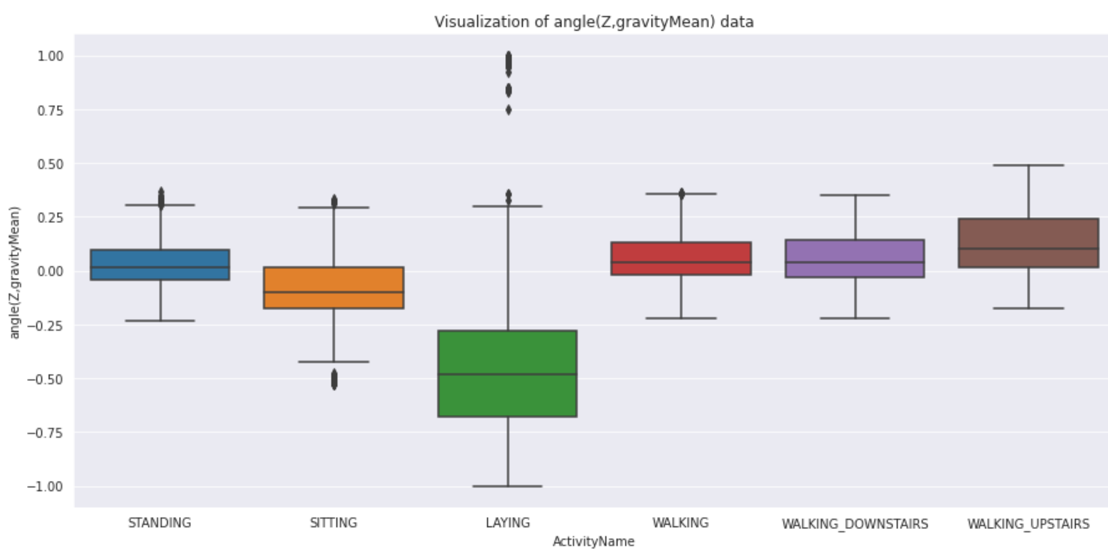
[ ]:
```
plt.figure(figsize=(15,7))
plt.title("Visualization of angle(Y,gravityMean) data")
sns.boxplot(x='ActivityName',y='angle(Y,gravityMean)',data=both_df)
```

[ ]: <matplotlib.axes._subplots.AxesSubplot at 0x7ff7db4afb90>



[ ]:
```
plt.figure(figsize=(15,7))
plt.title("Visualization of angle(Z,gravityMean) data")
sns.boxplot(x='ActivityName',y='angle(Z,gravityMean)',data=both_df)
```

[ ]: <matplotlib.axes._subplots.AxesSubplot at 0x7ff7dad4e5d0>



The second plotted boxplot clearly shows the exploiting, we will use this one

0.1.7 2.1.7 Rule that effectively separates LAYING from all other classes

We only need a if statemantIf angleX,gravityMean > 0 then Activity is Laying.

0.1.8 2.1.8 t-SNE visualization

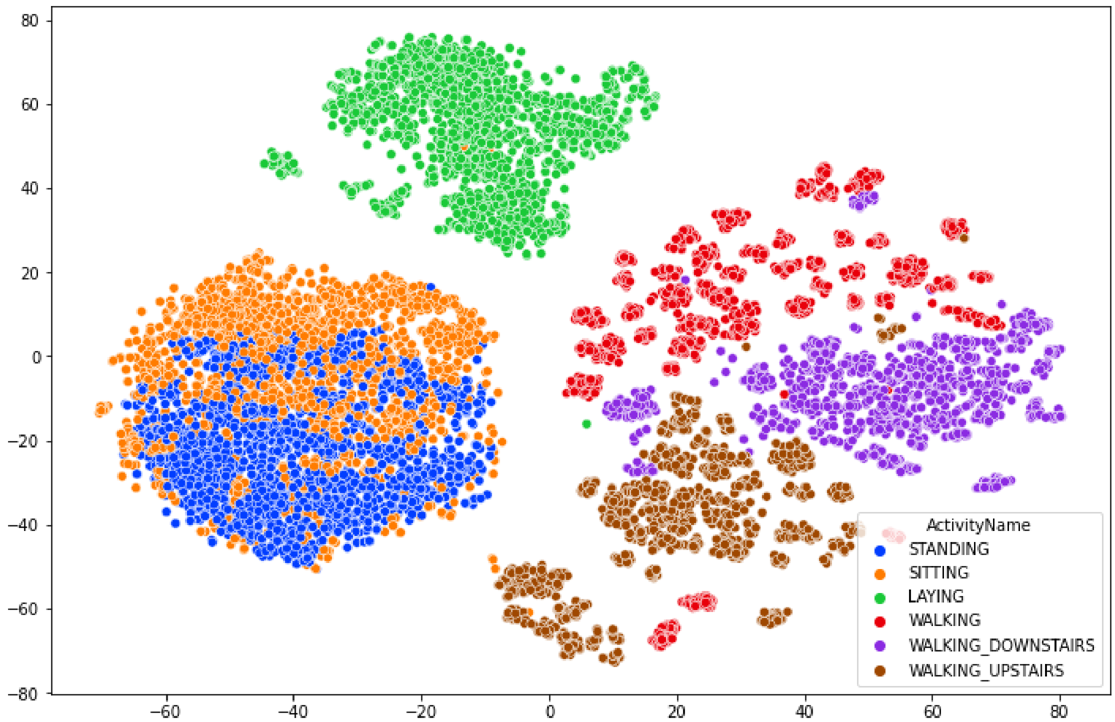
[ ]:
```
df_temp = both_df.copy()

train_tSNE = df_temp.drop(['Activity','subject','ActivityName'],axis=1)
tSNE = TSNE(random_state=42,␣
↪n_components=2,verbose=1,perplexity=50,n_iter=1000).fit_transform(train_tSNE)
```

```
/usr/local/lib/python3.7/dist-packages/sklearn/manifold/_t_sne.py:783:
FutureWarning: The default initialization in TSNE will change from 'random' to
'pca' in 1.2.
  FutureWarning,
/usr/local/lib/python3.7/dist-packages/sklearn/manifold/_t_sne.py:793:
FutureWarning: The default learning rate in TSNE will change from 200.0 to
'auto' in 1.2.
  FutureWarning,
```

```
[t-SNE] Computing 151 nearest neighbors...
[t-SNE] Indexed 10299 samples in 0.005s...
[t-SNE] Computed neighbors for 10299 samples in 4.047s...
[t-SNE] Computed conditional probabilities for sample 1000 / 10299
[t-SNE] Computed conditional probabilities for sample 2000 / 10299
[t-SNE] Computed conditional probabilities for sample 3000 / 10299
[t-SNE] Computed conditional probabilities for sample 4000 / 10299
[t-SNE] Computed conditional probabilities for sample 5000 / 10299
[t-SNE] Computed conditional probabilities for sample 6000 / 10299
[t-SNE] Computed conditional probabilities for sample 7000 / 10299
[t-SNE] Computed conditional probabilities for sample 8000 / 10299
[t-SNE] Computed conditional probabilities for sample 9000 / 10299
[t-SNE] Computed conditional probabilities for sample 10000 / 10299
[t-SNE] Computed conditional probabilities for sample 10299 / 10299
[t-SNE] Mean sigma: 1.385627
[t-SNE] KL divergence after 250 iterations with early exaggeration: 77.983643
[t-SNE] KL divergence after 1000 iterations: 1.505277
```

[ ]:
```
plt.figure(figsize=(12,8))
sns.scatterplot(x=tSNE[:,0],y=tSNE[:
↪,1],hue=both_df['ActivityName'],palette='bright')
```

[ ]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x7ff85db12610>
```



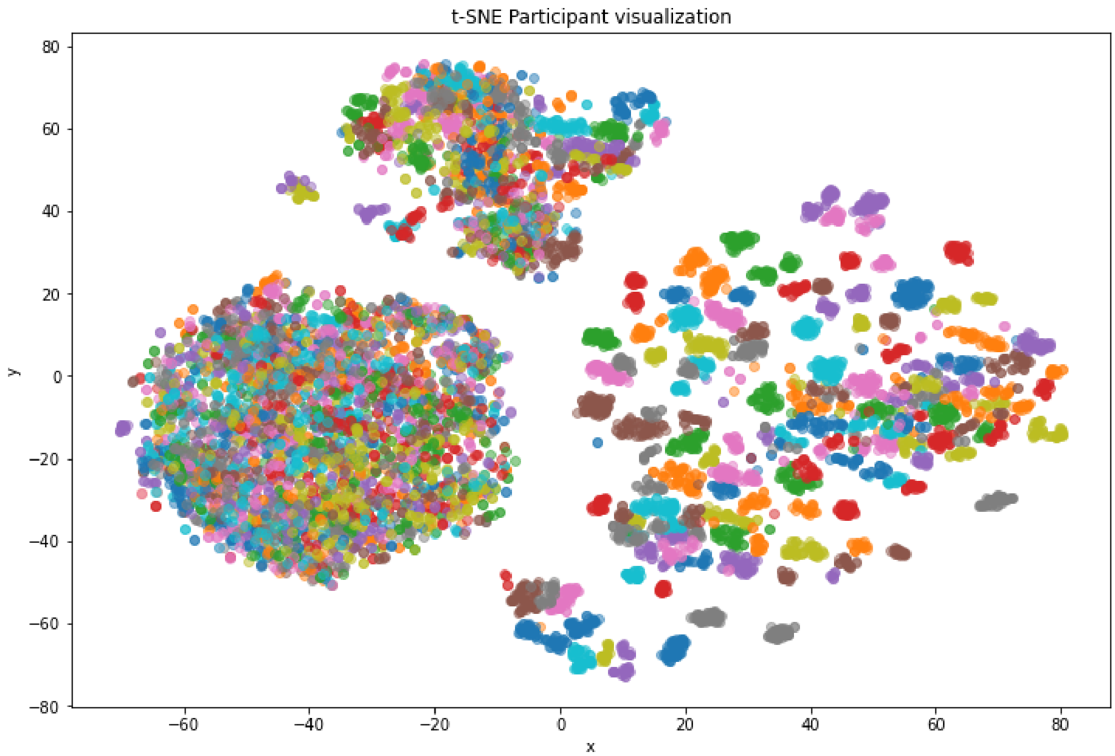
Our tSNE clusters work and show clear seperation except for the activities: Standing and Sitting.All our walking activities are located in the bottom right area of the plot.All laying activities are seperated clearly from the rest A resaon why standing and sitting are grouped tohgether and Laying down isnt with those groups could be that the angles of sitting and standing are similar too each other, and laying down has the oppsitite angles. Our model will probably be confused between standing and sitting data, but we can confidently predict our walking and laying down activities

0.1.9 2.1.9 t-SNE with userID visualization

[ ]:
```
dd = both_df.copy()
sub_data = dd.subject

label = dd.Activity
n = label.unique().shape[0]
colormap = get_cmap('viridis')
colors = [rgb2hex(colormap(col)) for col in np.arange(0, 1.01, 1/(n-1))]
```

[ ]:
```
plt.figure(figsize=(12,8))
plt.title("t-SNE Participant visualization")
plt.xlabel("x")
plt.ylabel("y")
for i, group in enumerate(sub_data.unique()):
    # Mask to separate sets
    mask = (sub_data==group).values
```

```
    plt.scatter(x=tSNE[mask][:,0], y=tSNE[mask][:,1], alpha=0.5, label=group)
plt.show()
```

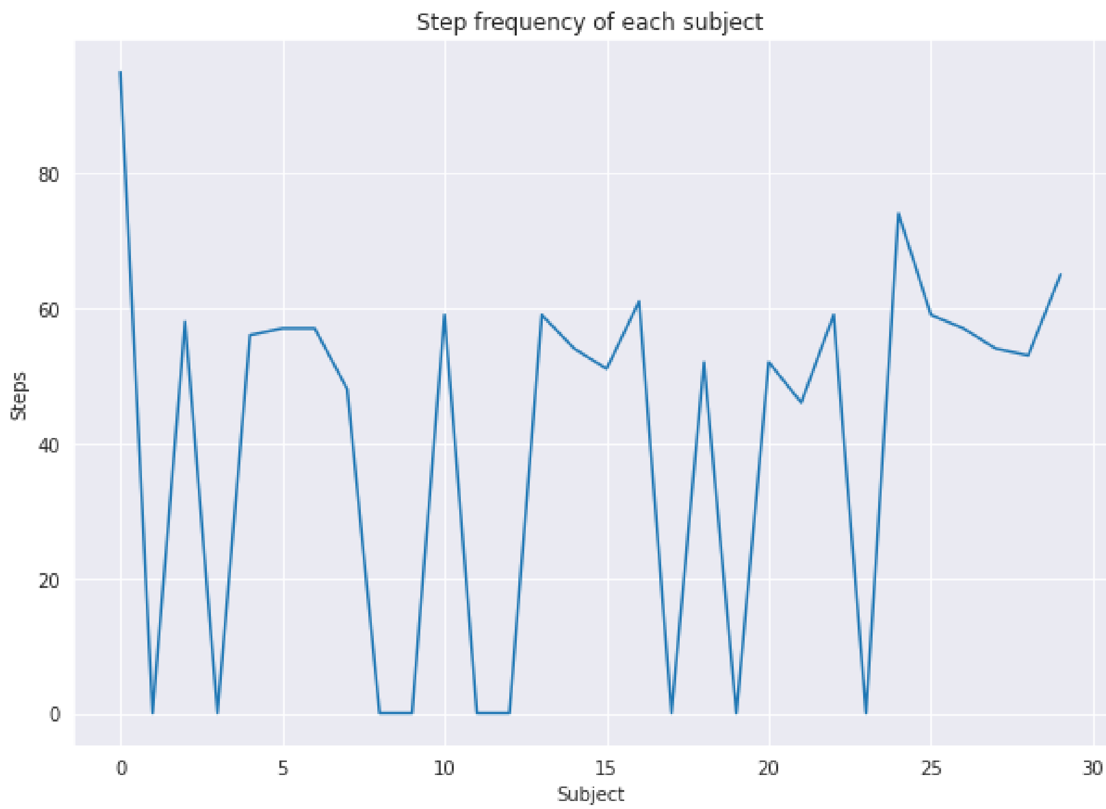


The plot above reveals every person's personal information, everybody has for example a unique walking style as seen in the bottom right area.We can detect what the participants are doing and also who is using the smartphone(only if theyre walking) We can also conclude that we cant really analyze the particpants in the standing and sitting clusters and the laying cluster.But from the 3 Walking clusters we can see groups of data which are clustered according to the userID

0.1.10 2.1.10 Investigate Walking

[ ]:
```
steps = [0 for i in range(30)]
for i in range(7352):
    for j in range(30):
        if((both_df.Activity[i]==1) and both_df.subject[i]==j+1):
            steps[j] += 1


sns.set_style("darkgrid")
plt.figure(figsize=(10,7))
plt.title('Step frequency of each subject')
plt.xlabel('Subject')
```

```
plt.ylabel('Steps')
plt.plot(steps)
plt.show()
```



Majority of all subjects had the same step count , with subject 1 having the most steps and subject 22 with the least amount of steps

0.1.11 2.1.11 Difference in Walking Speed among all participants

[ ]:

```

```

0.2 2.2 Baseline Models

0.2.1 2.2.1 baseline model predictions and F1 scores

[112]:

```
xTrain = train.drop(['subject', 'Activity', 'ActivityName'], axis=1)
yTrain = train.ActivityName

xTest = test.drop(['subject', 'Activity', 'ActivityName'], axis=1)
yTest = test.ActivityName
```

Random Forest Classifier

[ ]:
```
def RFC_model(xTrain, yTrain, xTest, yTest):
  rf = RandomForestClassifier(random_state=42)
  train_start_time = datetime.now()
  rf.fit(xTrain, yTrain)
  train_end_time = datetime.now()
  print("Build time:", train_end_time - train_start_time)
  rf_pred = rf.predict(xTest)
  accuracy_score_rf = f1_score(yTest, rf_pred, average='micro')*100
  print("F1 Score:",accuracy_score_rf)
```

Logistic Regression

[ ]:
```
def LR_model(xTrain, yTrain, xTest, yTest):
  lg = LogisticRegression(random_state=42)
  train_start_time = datetime.now()
  lg.fit(xTrain, yTrain)
  train_end_time = datetime.now()
  print("Build time:", train_end_time - train_start_time)
  lg_pred = lg.predict(xTest)
  accuracy_score_lf = f1_score(yTest, lg_pred, average='micro')*100
  print("F1 Score:",accuracy_score_lf)
```

SVC with rbf Kernel

[126]:
```
def SVC_model(xTrain, yTrain, xTest, yTest):
  svc = SVC(kernel='rbf',random_state=42)
  train_start_time = datetime.now()
  svc.fit(xTrain,yTrain)
  train_end_time = datetime.now()
  print("Build time:", train_end_time - train_start_time)
  svc_pred = svc.predict(xTest)
  accuracy_score_svc = f1_score(yTest, svc_pred, average='micro')*100
  print("F1 Score:",accuracy_score_svc)
```

Run all Models

[128]:
```
def run_models(xTrain, yTrain, xTest, yTest):
  print("RANDOM FOREST CLASSIFIER")
  print("======================================")
  RFC_model(xTrain, yTrain, xTest, yTest)
  print("======================================\n\n")

  print("LOGISTIC REGRESSION")
  print("======================================")
  LR_model(xTrain, yTrain, xTest, yTest)
  print("======================================\n\n")
```

```
  print("SVC")
  print("======================================")
  SVC_model(xTrain, yTrain, xTest, yTest)
  print("======================================\n\n")

run_models(xTrain, yTrain, xTest, yTest)
```

```
RANDOM FOREST CLASSIFIER
======================================
Build time: 0:00:12.871466
F1 Score: 92.60264675941634
======================================


LOGISTIC REGRESSION
======================================

/usr/local/lib/python3.7/dist-packages/sklearn/linear_model/_logistic.py:818:
ConvergenceWarning: lbfgs failed to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-
regression
  extra_warning_msg=_LOGISTIC_SOLVER_CONVERGENCE_MSG,

Build time: 0:00:03.577227
F1 Score: 95.89412962334578
======================================


SVC
======================================
Build time: 0:00:02.008686
F1 Score: 95.04580929759076
======================================
```

Findings: RFC -> F1 Score: 92.6 ; Build Time: 12ms LR -> F1 Score: 95.9 ; Build Time: 3ms SVC -> F1 Score: 95 ; Build Time: 1ms Our SVC model did the best in terms of the build time, and Logistic Regression(LR) got the highest F1 score

0.2.2 2.2.2 predict WALKING class

[162]:
```
train_2 = pd.read_csv('/content/drive/MyDrive/IDVE_Exam/UCI DATASET/train.csv')
test_2 = pd.read_csv('/content/drive/MyDrive/IDVE_Exam/UCI DATASET/test.csv')

train2 = train_2[train_2["ActivityName"] == "WALKING"]
test2 = test_2[test_2["ActivityName"] == "WALKING"]
```

[163]:
```
train2
```

[163]:
```
      tBodyAcc-mean()-X  tBodyAcc-mean()-Y  ...  Activity  ActivityName
78             0.282022          -0.037696  ...         1       WALKING
79             0.255841          -0.064550  ...         1       WALKING
80             0.254867           0.003815  ...         1       WALKING
81             0.343370          -0.014446  ...         1       WALKING
82             0.276240          -0.029638  ...         1       WALKING
...                 ...                ...  ...       ...           ...
7289           0.368741          -0.037037  ...         1       WALKING
7290           0.283921          -0.026589  ...         1       WALKING
7291           0.208795          -0.011955  ...         1       WALKING
7292           0.207863          -0.019810  ...         1       WALKING
7293           0.270378          -0.026488  ...         1       WALKING

[1226 rows x 564 columns]
```

[164]:
```
yTrain_2 = train2.subject
xTrain_2 = train2.drop(['subject', 'ActivityName'], axis=1)

yTest_2 = test2.subject
xTest_2 = test2.drop(['subject','ActivityName'], axis=1)
```

[165]:
```
xTest_2.shape,yTest_2.shape
```

[165]:
```
((496, 562), (496,))
```

[166]:
```
xTrain_2.shape , yTrain_2.shape
```

[166]:
```
((1226, 562), (1226,))
```

Random Forest Classifier

[167]:
```
xTrain_2
```

[167]:
```
      tBodyAcc-mean()-X  tBodyAcc-mean()-Y  ...  angle(Z,gravityMean)  Activity
78             0.282022          -0.037696  ...              0.044099         1
79             0.255841          -0.064550  ...              0.044638         1
80             0.254867           0.003815  ...              0.039417         1
81             0.343370          -0.014446  ...              0.039735         1
82             0.276240          -0.029638  ...              0.041412         1
...                 ...                ...  ...                   ...       ...
7289           0.368741          -0.037037  ...              0.000164         1
7290           0.283921          -0.026589  ...              0.004581         1
```

```
7291            0.208795          -0.011955  ...            0.005806         1
7292            0.207863          -0.019810  ...           -0.000384         1
7293            0.270378          -0.026488  ...           -0.012332         1

[1226 rows x 562 columns]
```

[169]:
```
run_models(xTrain_2, yTrain_2, xTest_2, yTest_2)
```

```
RANDOM FOREST CLASSIFIER
======================================
Build time: 0:00:02.078775
F1 Score: 0.0
======================================



LOGISTIC REGRESSION
======================================

/usr/local/lib/python3.7/dist-packages/sklearn/linear_model/_logistic.py:818:
ConvergenceWarning: lbfgs failed to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-
regression
  extra_warning_msg=_LOGISTIC_SOLVER_CONVERGENCE_MSG,

Build time: 0:00:01.600541
F1 Score: 0.0
======================================



SVC
======================================
Build time: 0:00:00.271260
F1 Score: 0.0
======================================
```

0.3 2.3 Feature Selection

0.3.1 2.3.1 Mutual information selection

[ ]:
```
from sklearn.preprocessing import LabelEncoder
from sklearn.feature_selection import SelectFromModel, SelectKBest


#prepare target
def prepare_targets(y_train, y_test):
    le = LabelEncoder()
    le.fit(y_train)
    y_train_enc = le.transform(y_train)
    y_test_enc = le.transform(y_test)
    return y_train_enc, y_test_enc

def select_features(X_train, y_train, X_test, kn):
    # configure to select 10 features
    fs = SelectKBest(score_func=mutual_info_classif, k=kn)
    fs.fit(X_train, y_train)
    # transform train input data
    X_train_fs = fs.transform(X_train)
    # transform test input data
    X_test_fs = fs.transform(X_test)
    return X_train_fs, X_test_fs, fs
```

[118]:
```
def getFeats(xTrain, xTest, yTrain, yTest, kn):
  y_train_enc, y_test_enc = prepare_targets(yTrain, yTest)
  _, _, fs = select_features(xTrain, y_train_enc, xTest, kn)
  # print(fs.get_support())
  chosen_feats = pd.DataFrame({ "Features": pd.DataFrame(xTrain).columns,
                         "Importances": fs.scores_, "Included": fs.
 ↪get_support()})
  chosen_feats.set_index('Importances')
  # sort in ascending order to better visualization.
  chosen_feats = chosen_feats[chosen_feats.Included == True].
 ↪sort_values('Importances')
  # what are scores for the feature
  for _, row in chosen_feats.iterrows():
    print('Feature %s: %f' % (row['Features'], row['Importances']))
  feats = chosen_feats.iloc[:,0]
  feats = np.array(feats)
  newXtrain = xTrain[feats]
  newXtest = xTest[feats]
  return newXtrain, newXtest
```

5 Features

[119]:
```
newXtrain, newXtest= getFeats(xTrain, xTest, yTrain, yTest, 5)
```

```
Feature tGravityAccMag-max(): 0.904660
Feature tGravityAcc-max()-Y: 0.928346
Feature tBodyAccJerk-max()-X: 0.937891
Feature tGravityAcc-min()-Y: 0.944264
Feature tBodyAcc-max()-X: 1.006971
```

[130]:
```
run_models(newXtrain, yTrain, newXtest, yTest)
```

```
RANDOM FOREST CLASSIFIER
======================================
Build time: 0:00:01.825583
F1 Score: 77.50254496097727
======================================



LOGISTIC REGRESSION
======================================

/usr/local/lib/python3.7/dist-packages/sklearn/linear_model/_logistic.py:818:
ConvergenceWarning: lbfgs failed to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-
regression
  extra_warning_msg=_LOGISTIC_SOLVER_CONVERGENCE_MSG,

Build time: 0:00:02.055965
F1 Score: 79.64031218187986
======================================



SVC
======================================
Build time: 0:00:00.620723
F1 Score: 79.91177468612148
======================================
```

10 Features

[131]:
```
newXtrain, newXtest= getFeats(xTrain, xTest, yTrain, yTest, 10)
```

```
Feature fBodyAcc-bandsEnergy()-1,8: 0.887428
Feature fBodyAcc-std()-X: 0.890934
Feature fBodyAcc-bandsEnergy()-1,16: 0.891218
Feature tBodyAccJerk-max()-Y: 0.892837
Feature tBodyAccMag-max(): 0.904660
Feature tGravityAccMag-max(): 0.904677
Feature tGravityAcc-max()-Y: 0.928346
Feature tBodyAccJerk-max()-X: 0.937809
Feature tGravityAcc-min()-Y: 0.944219
Feature tBodyAcc-max()-X: 1.006912
```

[133]:
```
run_models(newXtrain, yTrain, newXtest, yTest)
```

```
RANDOM FOREST CLASSIFIER
======================================
Build time: 0:00:01.846784
F1 Score: 77.50254496097727
======================================



LOGISTIC REGRESSION
======================================

/usr/local/lib/python3.7/dist-packages/sklearn/linear_model/_logistic.py:818:
ConvergenceWarning: lbfgs failed to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-
regression
  extra_warning_msg=_LOGISTIC_SOLVER_CONVERGENCE_MSG,

Build time: 0:00:02.181549
F1 Score: 79.64031218187986
======================================



SVC
======================================
Build time: 0:00:00.619853
F1 Score: 79.91177468612148
======================================
```

50 Features

[134]:
```
newXtrain, newXtest= getFeats(xTrain, xTest, yTrain, yTest, 50)
```

```
Feature tBodyAcc-iqr()-X: 0.816692
Feature fBodyAccMag-std(): 0.817165
Feature fBodyAccJerk-energy()-X: 0.817339
Feature tBodyAccJerk-std()-X: 0.818769
Feature fBodyAccJerk-bandsEnergy()-1,24: 0.818916
Feature tGravityAcc-min()-X: 0.822990
Feature tBodyGyro-min()-Z: 0.823055
Feature tBodyAccMag-std(): 0.826947
Feature tGravityAccMag-std(): 0.826981
Feature fBodyAccMag-energy(): 0.827403
Feature fBodyAcc-entropy()-X: 0.828912
Feature tBodyAcc-min()-Y: 0.833924
Feature tBodyGyroJerk-max()-Y: 0.834010
Feature tBodyGyroJerk-min()-Y: 0.835745
Feature tBodyAccJerkMag-min(): 0.839649
Feature fBodyAccMag-mad(): 0.839825
Feature fBodyAcc-mean()-X: 0.846683
Feature tBodyGyroJerk-min()-X: 0.847122
Feature tBodyGyroJerkMag-max(): 0.847799
Feature tBodyGyroJerkMag-min(): 0.847831
Feature tBodyAccJerk-min()-Z: 0.850914
Feature fBodyAccJerk-bandsEnergy()-1,16: 0.851100
Feature tBodyGyroJerk-min()-Z: 0.854550
Feature tBodyAccJerk-min()-Y: 0.854944
Feature angle(Y,gravityMean): 0.855138
Feature tBodyGyroJerk-max()-X: 0.855332
Feature tGravityAcc-mean()-Y: 0.856553
Feature tBodyAccJerk-max()-Z: 0.856927
Feature tBodyGyroJerk-max()-Z: 0.857461
Feature tBodyAccJerk-min()-X: 0.857462
Feature fBodyAcc-max()-X: 0.861385
Feature fBodyAcc-mad()-X: 0.863220
Feature fBodyAccJerk-bandsEnergy()-1,8: 0.867975
Feature tBodyAcc-mad()-X: 0.869326
Feature fBodyAcc-bandsEnergy()-1,24: 0.872864
Feature fBodyAcc-energy()-X: 0.873984
Feature tBodyAcc-std()-X: 0.874607
Feature tBodyAcc-energy()-X: 0.876395
Feature tBodyAcc-min()-X: 0.877175
Feature tBodyAccJerkMag-max(): 0.878197
Feature fBodyAcc-bandsEnergy()-1,8: 0.887236
Feature fBodyAcc-bandsEnergy()-1,16: 0.890220
Feature fBodyAcc-std()-X: 0.890948
Feature tBodyAccJerk-max()-Y: 0.892722
Feature tBodyAccMag-max(): 0.904660
```

```
Feature tGravityAccMag-max(): 0.904677
Feature tGravityAcc-max()-Y: 0.928346
Feature tBodyAccJerk-max()-X: 0.937842
Feature tGravityAcc-min()-Y: 0.944264
Feature tBodyAcc-max()-X: 1.006899
```

[135]:
```
run_models(newXtrain, yTrain, newXtest, yTest)
```

```
RANDOM FOREST CLASSIFIER
======================================
Build time: 0:00:03.737093
F1 Score: 85.74821852731591
======================================


LOGISTIC REGRESSION
======================================

/usr/local/lib/python3.7/dist-packages/sklearn/linear_model/_logistic.py:818:
ConvergenceWarning: lbfgs failed to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-
regression
  extra_warning_msg=_LOGISTIC_SOLVER_CONVERGENCE_MSG,

Build time: 0:00:01.274786
F1 Score: 87.64845605700712
======================================


SVC
======================================
Build time: 0:00:00.571578
F1 Score: 87.51272480488632
======================================
```

100 Features

[136]:
```
newXtrain, newXtest= getFeats(xTrain, xTest, yTrain, yTest, 100)
```

```
Feature tBodyAccJerk-sma(): 0.771934
Feature fBodyAccJerk-bandsEnergy()-9,16: 0.773844
```

```
Feature fBodyAccJerk-bandsEnergy()-1,24.1: 0.774854
Feature fBodyAccJerk-sma(): 0.775273
Feature tBodyAccJerkMag-sma(): 0.776041
Feature tBodyAccJerkMag-mean(): 0.776085
Feature tBodyGyroJerk-mad()-Z: 0.776583
Feature tBodyGyroJerk-mad()-X: 0.777748
Feature fBodyBodyAccJerkMag-mean(): 0.777820
Feature fBodyBodyAccJerkMag-max(): 0.777834
Feature fBodyBodyAccJerkMag-sma(): 0.777889
Feature fBodyBodyAccJerkMag-std(): 0.781514
Feature fBodyBodyAccJerkMag-mad(): 0.781670
Feature tGravityAccMag-sma(): 0.781832
Feature tBodyAccMag-mean(): 0.781894
Feature tGravityAccMag-mean(): 0.781894
Feature tBodyAccMag-sma(): 0.781923
Feature fBodyAccMag-iqr(): 0.782192
Feature tBodyGyro-max()-X: 0.782920
Feature fBodyAccJerk-max()-X: 0.783415
Feature tBodyAccJerkMag-energy(): 0.787081
Feature tBodyAcc-max()-Y: 0.787319
Feature fBodyAcc-sma(): 0.787826
Feature tBodyAcc-max()-Z: 0.792401
Feature fBodyBodyAccJerkMag-energy(): 0.795098
Feature tGravityAccMag-energy(): 0.795999
Feature tBodyAccMag-energy(): 0.796004
Feature tBodyAccJerkMag-entropy(): 0.796492
Feature fBodyAccMag-entropy(): 0.796881
Feature fBodyAccJerk-mad()-X: 0.797407
Feature tBodyAccJerk-mad()-X: 0.798905
Feature fBodyAccJerk-std()-X: 0.801580
Feature tBodyAccJerkMag-std(): 0.802109
Feature tBodyAccJerkMag-mad(): 0.802374
Feature tBodyAcc-min()-Z: 0.803398
Feature fBodyAcc-bandsEnergy()-9,16: 0.803816
Feature tBodyAccJerkMag-iqr(): 0.805221
Feature tBodyGyro-max()-Z: 0.805230
Feature tBodyGyro-min()-Y: 0.806147
Feature fBodyAccJerk-mean()-X: 0.807146
Feature tGravityAccMag-min(): 0.809351
Feature tBodyAccMag-min(): 0.809355
Feature tBodyGyro-min()-X: 0.810837
Feature tBodyGyroMag-max(): 0.811107
Feature tBodyAccJerk-energy()-X: 0.812069
Feature tBodyGyro-max()-Y: 0.812133
Feature tBodyAccMag-mad(): 0.812346
Feature tGravityAccMag-mad(): 0.812346
Feature fBodyAccMag-sma(): 0.813327
Feature fBodyAccMag-mean(): 0.813337
```

```
Feature tBodyAcc-iqr()-X: 0.816710
Feature fBodyAccJerk-energy()-X: 0.816839
Feature fBodyAccMag-std(): 0.817176
Feature fBodyAccJerk-bandsEnergy()-1,24: 0.818644
Feature tBodyAccJerk-std()-X: 0.818860
Feature fBodyAcc-entropy()-X: 0.819888
Feature tGravityAcc-min()-X: 0.822990
Feature tBodyGyro-min()-Z: 0.823078
Feature tBodyAccMag-std(): 0.826947
Feature tGravityAccMag-std(): 0.826991
Feature fBodyAccMag-energy(): 0.827533
Feature tBodyAcc-min()-Y: 0.833879
Feature tBodyGyroJerk-max()-Y: 0.833993
Feature tBodyGyroJerk-min()-Y: 0.835673
Feature tBodyAccJerkMag-min(): 0.839702
Feature fBodyAccMag-mad(): 0.839847
Feature fBodyAcc-mean()-X: 0.846695
Feature tBodyGyroJerk-min()-X: 0.847222
Feature tBodyGyroJerkMag-min(): 0.847758
Feature tBodyGyroJerkMag-max(): 0.847871
Feature fBodyAccJerk-bandsEnergy()-1,16: 0.850511
Feature tBodyAccJerk-min()-Z: 0.850910
Feature tBodyGyroJerk-min()-Z: 0.854525
Feature tBodyAccJerk-min()-Y: 0.854944
Feature angle(Y,gravityMean): 0.855172
Feature tBodyGyroJerk-max()-X: 0.855362
Feature tGravityAcc-mean()-Y: 0.856561
Feature tBodyAccJerk-max()-Z: 0.856972
Feature tBodyAccJerk-min()-X: 0.857447
Feature tBodyGyroJerk-max()-Z: 0.857472
Feature fBodyAcc-max()-X: 0.861417
Feature fBodyAcc-mad()-X: 0.863213
Feature fBodyAccJerk-bandsEnergy()-1,8: 0.868406
Feature tBodyAcc-mad()-X: 0.869377
Feature fBodyAcc-bandsEnergy()-1,24: 0.872411
Feature tBodyAcc-std()-X: 0.874610
Feature fBodyAcc-energy()-X: 0.874634
Feature tBodyAcc-energy()-X: 0.876179
Feature tBodyAcc-min()-X: 0.877181
Feature tBodyAccJerkMag-max(): 0.878195
Feature fBodyAcc-bandsEnergy()-1,8: 0.887239
Feature fBodyAcc-bandsEnergy()-1,16: 0.890237
Feature fBodyAcc-std()-X: 0.890966
Feature tBodyAccJerk-max()-Y: 0.892794
Feature tGravityAccMag-max(): 0.904677
Feature tBodyAccMag-max(): 0.904677
Feature tGravityAcc-max()-Y: 0.928346
Feature tBodyAccJerk-max()-X: 0.937790
```

```
Feature tGravityAcc-min()-Y: 0.944219
Feature tBodyAcc-max()-X: 1.006893
```

[137]:
```
run_models(newXtrain, yTrain, newXtest, yTest)
```

```
RANDOM FOREST CLASSIFIER
======================================
Build time: 0:00:05.389135
F1 Score: 88.49677638276214
======================================


LOGISTIC REGRESSION
======================================

/usr/local/lib/python3.7/dist-packages/sklearn/linear_model/_logistic.py:818:
ConvergenceWarning: lbfgs failed to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-
regression
  extra_warning_msg=_LOGISTIC_SOLVER_CONVERGENCE_MSG,

Build time: 0:00:01.156796
F1 Score: 89.54869358669833
======================================


SVC
======================================
Build time: 0:00:00.706230
F1 Score: 89.71835765184933
======================================
```

When we use less features our F1 score and build time both decreases The tradeoff for having more features gives our models a higher F1 score and higher build time, whereas a small amount of features will decrease our build time and F1 score

0.3.2 2.3.2 Mutual information selection: WALKING

[ ]:
```

```

0.4 2.4 Feature Extraction

0.4.1 2.4.1 2D plot of the embeddings/principal components

[ ]:
```
def plot2d(type, df):
  fig = plt.figure(figsize = (8,8))
  ax = fig.add_subplot(1,1,1)
  ax.set_xlabel('Component 1', fontsize = 15)
  ax.set_ylabel('Component 2', fontsize = 15)
  ax.set_title('2 component {0}'.format(type), fontsize = 20)
  targets = ['WALKING', 'WALKING_UPSTAIRS','WALKING_DOWNSTAIRS','SITTING',␣
↪'STANDING','LAYING']
  colors = ['r', 'g', 'b','c','m','y']
  for target, color in zip(targets,colors):
    indicesToKeep = df['ActivityName'] == target
    ax.scatter(df.loc[indicesToKeep, 'component 1']
               , df.loc[indicesToKeep, 'component 2']
               , c = color
               , s = 50)
  ax.legend(targets)
  ax.grid()
```

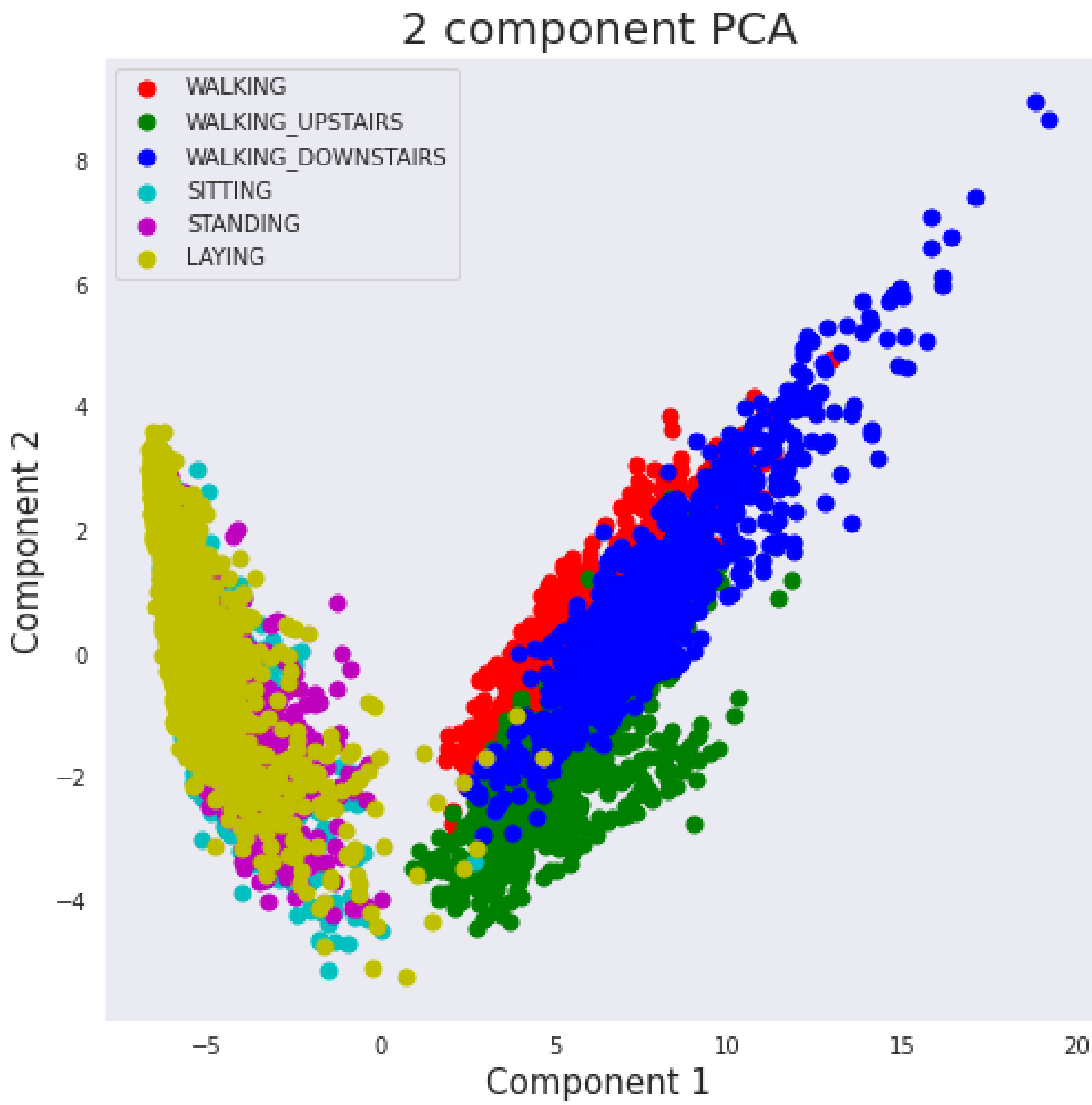
[ ]:
```
xTrain.shape
```

[ ]:
```
(7352, 561)
```

PCA

[ ]:
```
from sklearn.decomposition import PCA

def getPCA(xtrain, ytrain, xtest, ytest):
  pca = PCA(n_components=2,random_state=42)
  x_train = pca.fit_transform(xtrain)
  x_test = pca.transform(xtest)
  pcaDF_train = pd.DataFrame(data = x_train, columns = ['component 1',␣
↪'component 2'])
  PCA_vis_df = pd.concat([pcaDF_train, ytrain], axis = 1)
  plot2d("PCA", PCA_vis_df)
  return x_train, x_test, y_train, y_test
```

[ ]:
```
pca_xTrain, pca_xTest, pca_yTrain, pca_yTest = getPCA(xTrain, yTrain, xTest,␣
↪yTest)
```

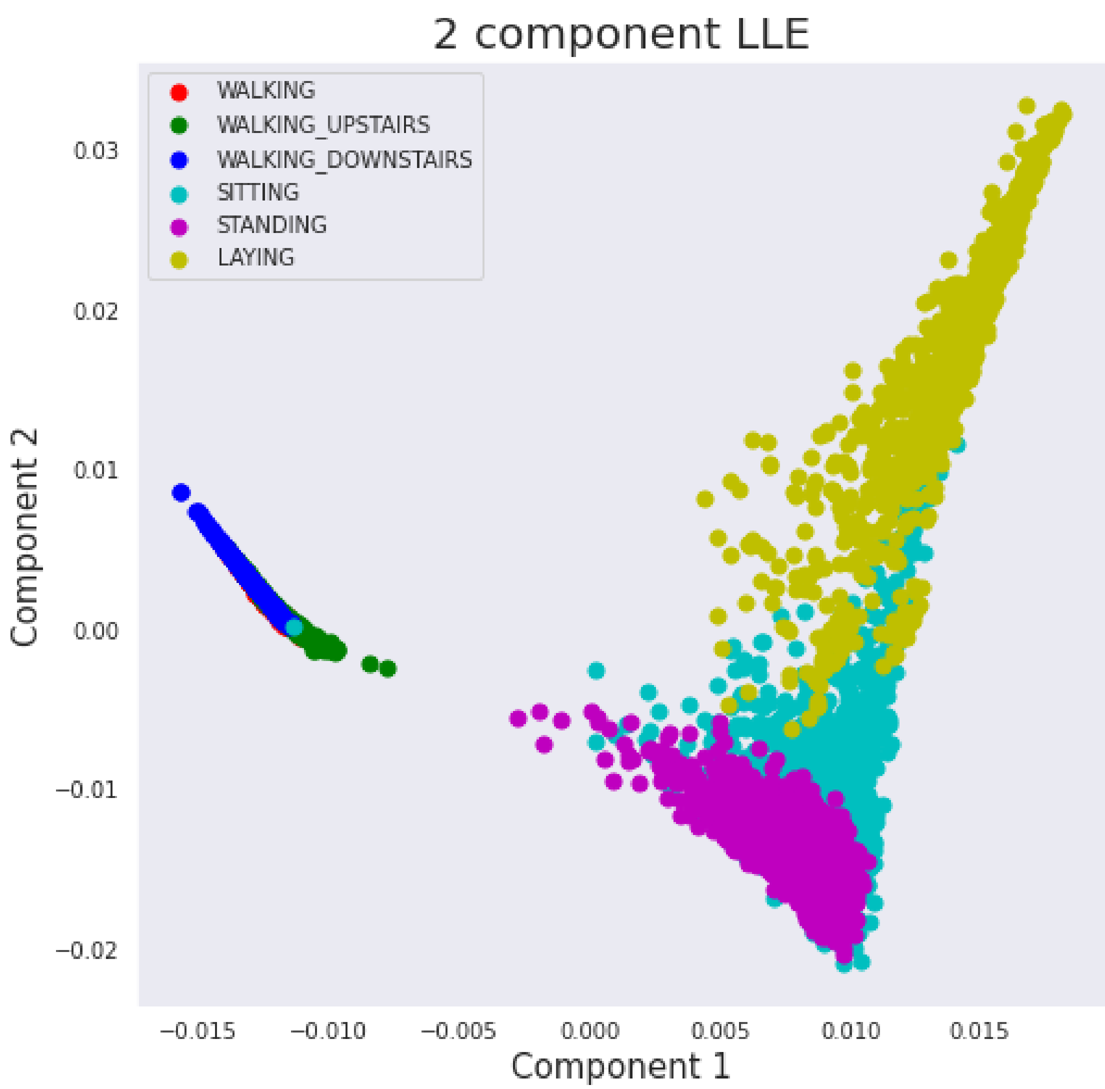


From the plot we can see that our model might struggle in classifying correctly because our clusters are close and on top of each other for each activity

LLE

[ ]:
```
from sklearn.manifold import LocallyLinearEmbedding as LLE # for LLE␣
↪dimensionality reduction

def getLLE(xtrain, ytrain, xtest, ytest):
  embed_lle = LLE(n_neighbors=65, n_components=2, method="standard")
  x_train = embed_lle.fit_transform(xtrain)
  x_test = embed_lle.transform(xtest)
  lleDF_train = pd.DataFrame(data = x_train, columns = ['component 1',␣
↪'component 2'])
```

```
  LLE_vis_df = pd.concat([lleDF_train, ytrain], axis = 1)
  plot2d("LLE", LLE_vis_df)
  return x_train, x_test, y_train, y_test

lle_xTrain, lle_xTest, lle_yTrain, lle_yTest = getLLE(xTrain, yTrain, xTest,␣
 ↪yTest)
```

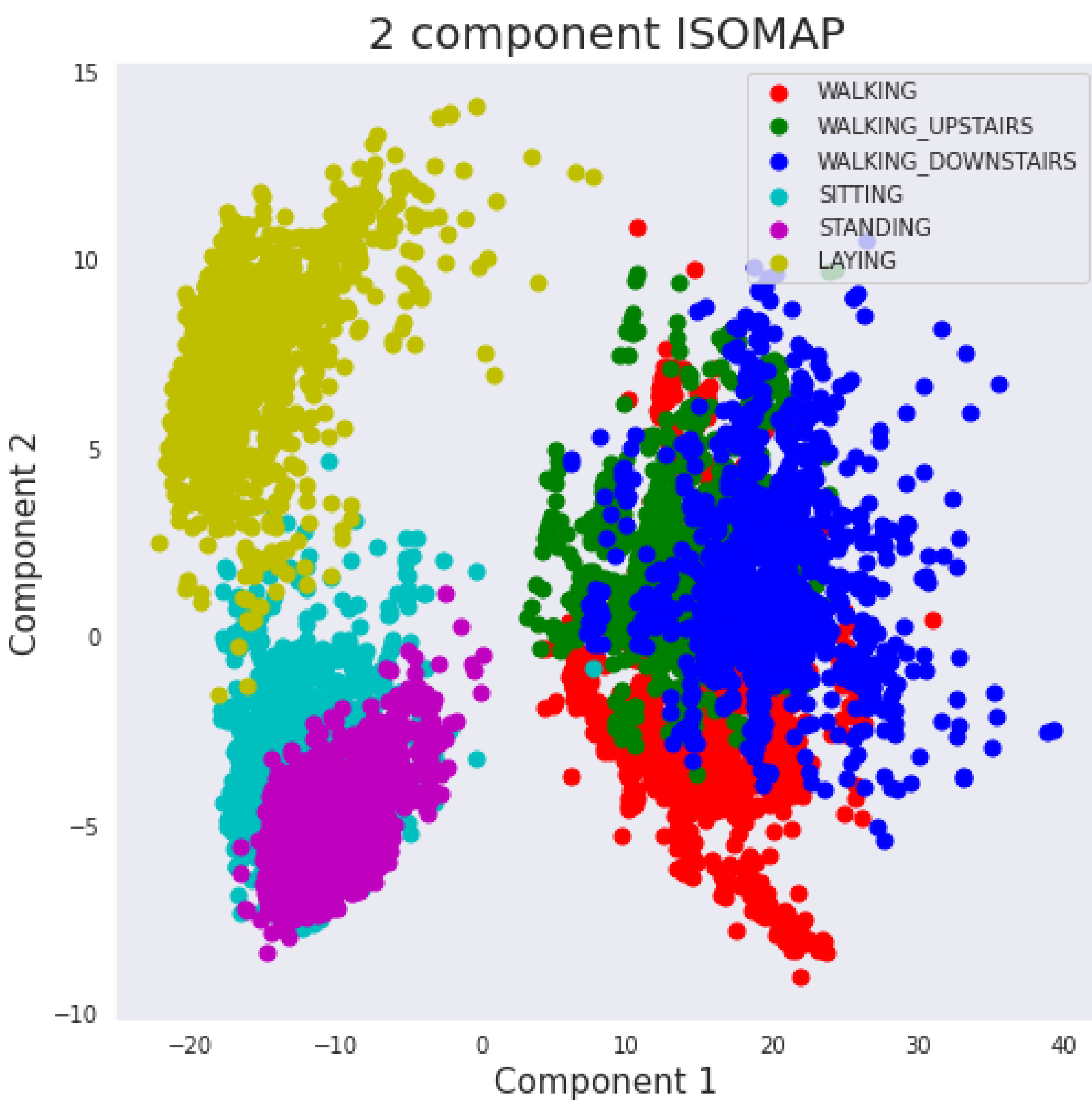


The plot tells us that our models will be able to predict our static activities confidently they are well clsutered but not optimal.Our models will struggle in predicting our dynamic activities because they're clustered on top of each other

ISOMAP

[ ]:
```
from sklearn.manifold import Isomap
```

```
def getISOMAP(xtrain, ytrain, xtest, ytest):
  iso = Isomap(n_components=2, n_neighbors=45)
  x_train = iso.fit_transform(xtrain)
  x_test = iso.transform(xtest)
  isoDF_train = pd.DataFrame(data = x_train, columns = ['component 1',␣
↪'component 2'])
  ISO_vis_df = pd.concat([isoDF_train, ytrain], axis = 1)
  plot2d("ISOMAP", ISO_vis_df)
  return x_train, x_test, y_train, y_test

iso_xTrain, iso_xTest, iso_yTrain, iso_yTest = getISOMAP(xTrain, yTrain, xTest,␣
↪yTest)
```

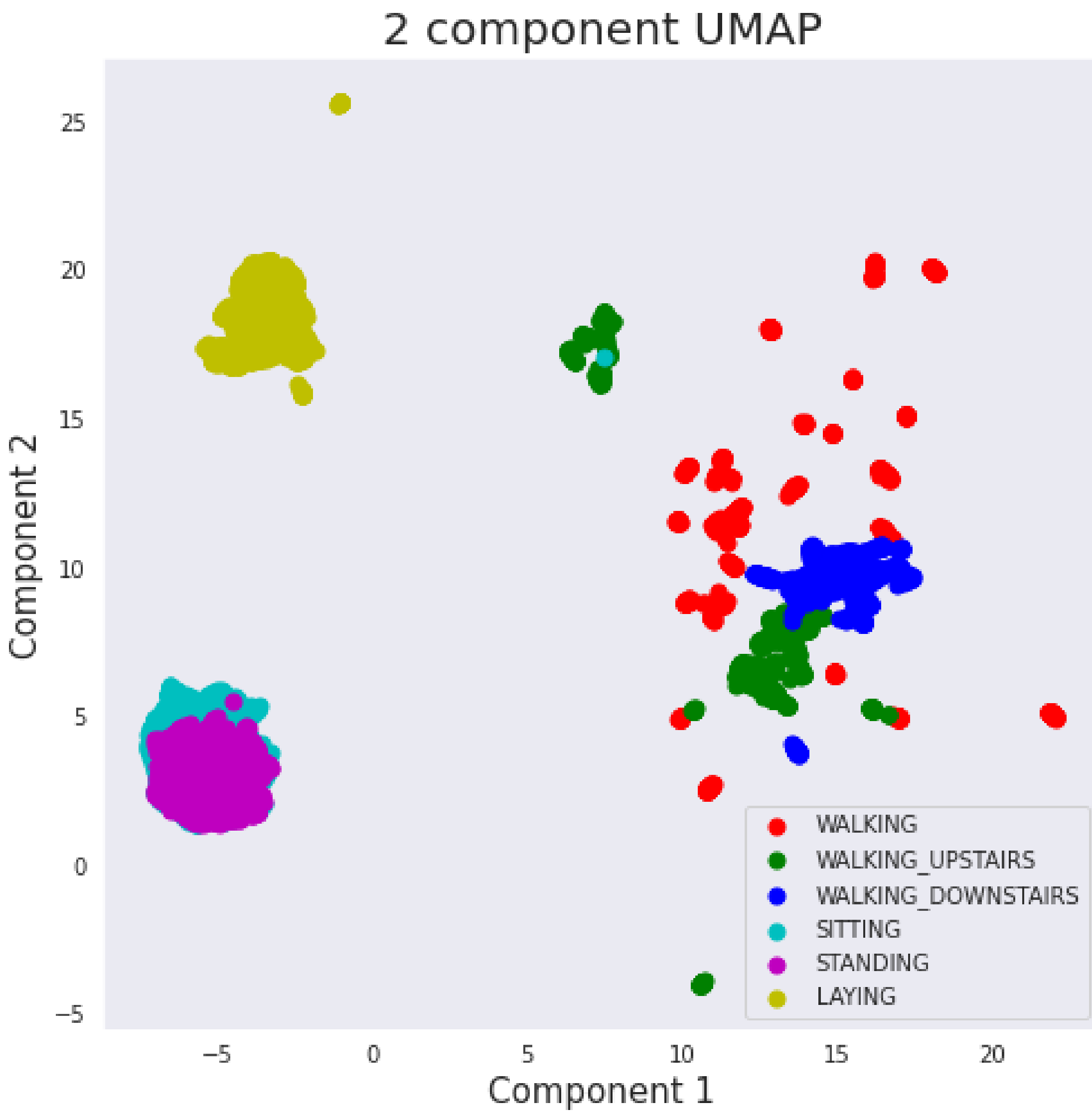


The plot tells us that our models will be able to predict our static activities confidently they are well clsutered but not optimal.Our models will struggle in predicting our dynamic activities because they're clustered on top of each other

UMAP

[ ]:
```
from umap import UMAP

 def getUMAP(xtrain, ytrain, xtest, ytest):
  umap_obj = UMAP(
      n_components=2,
      metric="euclidean",
      n_neighbors=20,
      min_dist=0.1,
      random_state=42
      )
  x_train = umap_obj.fit_transform(xtrain)
  x_test = umap_obj.transform(xtest)
  umapDF_train = pd.DataFrame(data = x_train, columns = ['component 1',␣
 ↪'component 2'])
  UMAP_vis_df = pd.concat([umapDF_train, ytrain], axis = 1)
  plot2d("UMAP", UMAP_vis_df)
  return x_train, x_test, y_train, y_test

umap_xTrain, umap_xTest, umap_yTrain, umap_yTest = getUMAP(xTrain, yTrain,␣
 ↪xTest, yTest)
```

```
/usr/local/lib/python3.7/dist-packages/numba/np/ufunc/parallel.py:363:
NumbaWarning: The TBB threading layer requires TBB version 2019.5 or later i.e.,
TBB_INTERFACE_VERSION >= 11005. Found TBB_INTERFACE_VERSION = 9107. The TBB
threading layer is disabled.
  warnings.warn(problem)
```



This plot shows the most confidence out of all our feauture extraction methods, we have well seperated clsuters far away from each other , with laying and standing activities that will be our only drwback when predicting activities because they are clsutered close to each other

0.4.2 2.4.2 Training embeddings on models

Models with PCA

[ ]:
```
run_models(pca_xTrain, pca_yTrain, pca_xTest, pca_yTest)
```

```
RANDOM FOREST CLASSIFIER
========================================
Build time: 0:00:01.252167
F1 Score: 54.90329148286393
```

```
========================================


LOGISTIC REGRESSION
========================================

/usr/local/lib/python3.7/dist-packages/sklearn/linear_model/_logistic.py:818:
ConvergenceWarning: lbfgs failed to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-
regression
  extra_warning_msg=_LOGISTIC_SOLVER_CONVERGENCE_MSG,

Build time: 0:00:00.450351
F1 Score: 58.33050559891415
========================================


SVC
========================================
Build time: 0:00:01.511016
F1 Score: 59.959280624363764
========================================
```

Models with LLE

[ ]:
```
run_models(lle_xTrain, lle_yTrain, lle_xTest, lle_yTest)
```

```
RANDOM FOREST CLASSIFIER
========================================
Build time: 0:00:01.387435
F1 Score: 70.71598235493722
========================================


LOGISTIC REGRESSION
========================================
Build time: 0:00:00.129042
F1 Score: 52.73159144893111
========================================


SVC
```

```
=======================================
Build time: 0:00:00.778993
F1 Score: 68.17102137767222
=======================================
```

Models with ISOMAP

[ ]:
```
run_models(iso_xTrain, iso_yTrain, iso_xTest, iso_yTest)
```

```
RANDOM FOREST CLASSIFIER
=======================================
Build time: 0:00:00.917097
F1 Score: 76.00950118764844
=======================================


LOGISTIC REGRESSION
=======================================

/usr/local/lib/python3.7/dist-packages/sklearn/linear_model/_logistic.py:818:
ConvergenceWarning: lbfgs failed to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-
regression
  extra_warning_msg=_LOGISTIC_SOLVER_CONVERGENCE_MSG,

Build time: 0:00:00.429028
F1 Score: 77.12928401764506
=======================================


SVC
=======================================
Build time: 0:00:00.580673
F1 Score: 78.45266372582287
=======================================
```

Models with UMAP

[ ]:
```
run_models(umap_xTrain, umap_yTrain, umap_xTest, umap_yTest)
```

```
RANDOM FOREST CLASSIFIER
========================================
Build time: 0:00:00.732559
F1 Score: 85.44282321004411
========================================


LOGISTIC REGRESSION
========================================

/usr/local/lib/python3.7/dist-packages/sklearn/linear_model/_logistic.py:818:
ConvergenceWarning: lbfgs failed to converge (status=1):
STOP: TOTAL NO. of ITERATIONS REACHED LIMIT.

Increase the number of iterations (max_iter) or scale the data as shown in:
    https://scikit-learn.org/stable/modules/preprocessing.html
Please also refer to the documentation for alternative solver options:
    https://scikit-learn.org/stable/modules/linear_model.html#logistic-
regression
  extra_warning_msg=_LOGISTIC_SOLVER_CONVERGENCE_MSG,

Build time: 0:00:00.502901
F1 Score: 71.56430268069222
========================================


SVC
========================================
Build time: 0:00:00.441893
F1 Score: 86.25721072276892
========================================
```

0.4.3 2.4.3 Findings on the trade-off between training time, performance, and interpretability

1. Interprability: All our plots show good interprebilty, we reduced more than 500+ features to only 2 features, and we could see how the different activities were clsutered , even though some were clustered on top of each other, those that were clsutered like this were from the same movement types(dynamic and static)
2. Training time:All our training times were much faster compared to the non feature extracted feautures due to the fact that all our models had to only fit to 2 features instead of 561.Overall our logistic regression had the fastest build times no matter the feature extraction technique used.
3. Performance: We did get lower F1 scores than if we didnt use feature extraction methods, but we still managed to get good F1 scores especially with UMAP and ISOMAP(UMAP the best)

Overall we were able to get better interpretability with feature extraction methods than if we didnt use them.We had a tradeoff in runtime , we were able to get better model building runtimes due to training with less features.Our performance wasnt as great as the models where we didnt apply feature extraction beacuase our models had less data to train on.But given the small amount of features we still were able to get good results and F1 scores.

[ ]:

```

```

IGNORE BELOW

[170]:

```
!pwd
```

```
/content
```

[176]:

```
%cd FOLDER
```

```
/content/drive/MyDrive/FOLDER
```

[177]:

```
!ls
```

```
IDVE_EXAM_Q1.ipynb  IDVE_EXAM_Q1.pdf  IDVE_EXAM_Q2.ipynb  IDVE_EXAM_Q3.ipynb
```

[ ]:

```
!sudo apt-get install texlive-xetex texlive-fonts-recommended␣
↪texlive-generic-recommended
```

```
Reading package lists... Done
Building dependency tree
Reading state information... Done
The following additional packages will be installed:
  fonts-droid-fallback fonts-lato fonts-lmodern fonts-noto-mono fonts-texgyre
  javascript-common libcupsfilters1 libcupsimage2 libgs9 libgs9-common
  libijs-0.35 libjbig2dec0 libjs-jquery libkpathsea6 libpotrace0 libptexenc1
  libruby2.5 libsynctex1 libtexlua52 libtexluajit2 libzzip-0-13 lmodern
  poppler-data preview-latex-style rake ruby ruby-did-you-mean ruby-minitest
  ruby-net-telnet ruby-power-assert ruby-test-unit ruby2.5
  rubygems-integration t1utils tex-common tex-gyre texlive-base
  texlive-binaries texlive-latex-base texlive-latex-extra
  texlive-latex-recommended texlive-pictures texlive-plain-generic tipa
Suggested packages:
  fonts-noto apache2 | lighttpd | httpd poppler-utils ghostscript
  fonts-japanese-mincho | fonts-ipafont-mincho fonts-japanese-gothic
  | fonts-ipafont-gothic fonts-arphic-ukai fonts-arphic-uming fonts-nanum ri
  ruby-dev bundler debhelper gv | postscript-viewer perl-tk xpdf-reader
  | pdf-viewer texlive-fonts-recommended-doc texlive-latex-base-doc
  python-pygments icc-profiles libfile-which-perl
  libspreadsheet-parseexcel-perl texlive-latex-extra-doc
  texlive-latex-recommended-doc texlive-pstricks dot2tex prerex ruby-tcltk
  | libtcltk-ruby texlive-pictures-doc vprerex
```

```
The following NEW packages will be installed:
  fonts-droid-fallback fonts-lato fonts-lmodern fonts-noto-mono fonts-texgyre
  javascript-common libcupsfilters1 libcupsimage2 libgs9 libgs9-common
  libijs-0.35 libjbig2dec0 libjs-jquery libkpathsea6 libpotrace0 libptexenc1
  libruby2.5 libsynctex1 libtexlua52 libtexluajit2 libzzip-0-13 lmodern
  poppler-data preview-latex-style rake ruby ruby-did-you-mean ruby-minitest
  ruby-net-telnet ruby-power-assert ruby-test-unit ruby2.5
  rubygems-integration t1utils tex-common tex-gyre texlive-base
  texlive-binaries texlive-fonts-recommended texlive-generic-recommended
  texlive-latex-base texlive-latex-extra texlive-latex-recommended
  texlive-pictures texlive-plain-generic texlive-xetex tipa
0 upgraded, 47 newly installed, 0 to remove and 37 not upgraded.
Need to get 146 MB of archives.
After this operation, 460 MB of additional disk space will be used.
Get:1 http://archive.ubuntu.com/ubuntu bionic/main amd64 fonts-droid-fallback
all 1:6.0.1r16-1.1 [1,805 kB]
Get:2 http://archive.ubuntu.com/ubuntu bionic/main amd64 fonts-lato all 2.0-2
[2,698 kB]
Get:3 http://archive.ubuntu.com/ubuntu bionic/main amd64 poppler-data all
0.4.8-2 [1,479 kB]
Get:4 http://archive.ubuntu.com/ubuntu bionic/main amd64 tex-common all 6.09
[33.0 kB]
Get:5 http://archive.ubuntu.com/ubuntu bionic/main amd64 fonts-lmodern all
2.004.5-3 [4,551 kB]
Get:6 http://archive.ubuntu.com/ubuntu bionic/main amd64 fonts-noto-mono all
20171026-2 [75.5 kB]
Get:7 http://archive.ubuntu.com/ubuntu bionic/universe amd64 fonts-texgyre all
20160520-1 [8,761 kB]
Get:8 http://archive.ubuntu.com/ubuntu bionic/main amd64 javascript-common all
11 [6,066 B]
Get:9 http://archive.ubuntu.com/ubuntu bionic-updates/main amd64 libcupsfilters1
amd64 1.20.2-0ubuntu3.1 [108 kB]
Get:10 http://archive.ubuntu.com/ubuntu bionic-updates/main amd64 libcupsimage2
amd64 2.2.7-1ubuntu2.8 [18.6 kB]
Get:11 http://archive.ubuntu.com/ubuntu bionic/main amd64 libijs-0.35 amd64
0.35-13 [15.5 kB]
Get:12 http://archive.ubuntu.com/ubuntu bionic/main amd64 libjbig2dec0 amd64
0.13-6 [55.9 kB]
Get:13 http://archive.ubuntu.com/ubuntu bionic-updates/main amd64 libgs9-common
all 9.26~dfsg+0-0ubuntu0.18.04.14 [5,092 kB]
Get:14 http://archive.ubuntu.com/ubuntu bionic-updates/main amd64 libgs9 amd64
9.26~dfsg+0-0ubuntu0.18.04.14 [2,265 kB]
Get:15 http://archive.ubuntu.com/ubuntu bionic/main amd64 libjs-jquery all
3.2.1-1 [152 kB]
Get:16 http://archive.ubuntu.com/ubuntu bionic-updates/main amd64 libkpathsea6
amd64 2017.20170613.44572-8ubuntu0.1 [54.9 kB]
Get:17 http://archive.ubuntu.com/ubuntu bionic/main amd64 libpotrace0 amd64
1.14-2 [17.4 kB]
```

```
Get:18 http://archive.ubuntu.com/ubuntu bionic-updates/main amd64 libptexenc1
amd64 2017.20170613.44572-8ubuntu0.1 [34.5 kB]
Get:19 http://archive.ubuntu.com/ubuntu bionic/main amd64 rubygems-integration
all 1.11 [4,994 B]
Get:20 http://archive.ubuntu.com/ubuntu bionic-updates/main amd64 ruby2.5 amd64
2.5.1-1ubuntu1.10 [48.6 kB]
Get:21 http://archive.ubuntu.com/ubuntu bionic/main amd64 ruby amd64 1:2.5.1
[5,712 B]
Get:22 http://archive.ubuntu.com/ubuntu bionic-updates/main amd64 rake all
12.3.1-1ubuntu0.1 [44.9 kB]
Get:23 http://archive.ubuntu.com/ubuntu bionic/main amd64 ruby-did-you-mean all
1.2.0-2 [9,700 B]
Get:24 http://archive.ubuntu.com/ubuntu bionic/main amd64 ruby-minitest all
5.10.3-1 [38.6 kB]
Get:25 http://archive.ubuntu.com/ubuntu bionic/main amd64 ruby-net-telnet all
0.1.1-2 [12.6 kB]
Get:26 http://archive.ubuntu.com/ubuntu bionic/main amd64 ruby-power-assert all
0.3.0-1 [7,952 B]
Get:27 http://archive.ubuntu.com/ubuntu bionic/main amd64 ruby-test-unit all
3.2.5-1 [61.1 kB]
Get:28 http://archive.ubuntu.com/ubuntu bionic-updates/main amd64 libruby2.5
amd64 2.5.1-1ubuntu1.10 [3,071 kB]
Get:29 http://archive.ubuntu.com/ubuntu bionic-updates/main amd64 libsynctex1
amd64 2017.20170613.44572-8ubuntu0.1 [41.4 kB]
Get:30 http://archive.ubuntu.com/ubuntu bionic-updates/main amd64 libtexlua52
amd64 2017.20170613.44572-8ubuntu0.1 [91.2 kB]
Get:31 http://archive.ubuntu.com/ubuntu bionic-updates/main amd64 libtexluajit2
amd64 2017.20170613.44572-8ubuntu0.1 [230 kB]
Get:32 http://archive.ubuntu.com/ubuntu bionic-updates/main amd64 libzzip-0-13
amd64 0.13.62-3.1ubuntu0.18.04.1 [26.0 kB]
Get:33 http://archive.ubuntu.com/ubuntu bionic/main amd64 lmodern all 2.004.5-3
[9,631 kB]
Get:34 http://archive.ubuntu.com/ubuntu bionic/main amd64 preview-latex-style
all 11.91-1ubuntu1 [185 kB]
Get:35 http://archive.ubuntu.com/ubuntu bionic/main amd64 t1utils amd64 1.41-2
[56.0 kB]
Get:36 http://archive.ubuntu.com/ubuntu bionic/universe amd64 tex-gyre all
20160520-1 [4,998 kB]
Get:37 http://archive.ubuntu.com/ubuntu bionic-updates/main amd64 texlive-
binaries amd64 2017.20170613.44572-8ubuntu0.1 [8,179 kB]
Get:38 http://archive.ubuntu.com/ubuntu bionic/main amd64 texlive-base all
2017.20180305-1 [18.7 MB]
Get:39 http://archive.ubuntu.com/ubuntu bionic/universe amd64 texlive-fonts-
recommended all 2017.20180305-1 [5,262 kB]
Get:40 http://archive.ubuntu.com/ubuntu bionic/universe amd64 texlive-plain-
generic all 2017.20180305-2 [23.6 MB]
Get:41 http://archive.ubuntu.com/ubuntu bionic/universe amd64 texlive-generic-
recommended all 2017.20180305-1 [15.9 kB]
```

```
Get:42 http://archive.ubuntu.com/ubuntu bionic/main amd64 texlive-latex-base all
2017.20180305-1 [951 kB]
Get:43 http://archive.ubuntu.com/ubuntu bionic/main amd64 texlive-latex-
recommended all 2017.20180305-1 [14.9 MB]
Get:44 http://archive.ubuntu.com/ubuntu bionic/universe amd64 texlive-pictures
all 2017.20180305-1 [4,026 kB]
Get:45 http://archive.ubuntu.com/ubuntu bionic/universe amd64 texlive-latex-
extra all 2017.20180305-2 [10.6 MB]
Get:46 http://archive.ubuntu.com/ubuntu bionic/universe amd64 tipa all 2:1.3-20
[2,978 kB]
Get:47 http://archive.ubuntu.com/ubuntu bionic/universe amd64 texlive-xetex all
2017.20180305-1 [10.7 MB]
Fetched 146 MB in 5s (30.3 MB/s)
debconf: unable to initialize frontend: Dialog
debconf: (No usable dialog-like program is installed, so the dialog based
frontend cannot be used. at /usr/share/perl5/Debconf/FrontEnd/Dialog.pm line 76,
<> line 47.)
debconf: falling back to frontend: Readline
debconf: unable to initialize frontend: Readline
debconf: (This frontend requires a controlling tty.)
debconf: falling back to frontend: Teletype
dpkg-preconfigure: unable to re-open stdin:
Selecting previously unselected package fonts-droid-fallback.
(Reading database ... 155222 files and directories currently installed.)
Preparing to unpack .../00-fonts-droid-fallback_1%3a6.0.1r16-1.1_all.deb ...
Unpacking fonts-droid-fallback (1:6.0.1r16-1.1) ...
Selecting previously unselected package fonts-lato.
Preparing to unpack .../01-fonts-lato_2.0-2_all.deb ...
Unpacking fonts-lato (2.0-2) ...
Selecting previously unselected package poppler-data.
Preparing to unpack .../02-poppler-data_0.4.8-2_all.deb ...
Unpacking poppler-data (0.4.8-2) ...
Selecting previously unselected package tex-common.
Preparing to unpack .../03-tex-common_6.09_all.deb ...
Unpacking tex-common (6.09) ...
Selecting previously unselected package fonts-lmodern.
Preparing to unpack .../04-fonts-lmodern_2.004.5-3_all.deb ...
Unpacking fonts-lmodern (2.004.5-3) ...
Selecting previously unselected package fonts-noto-mono.
Preparing to unpack .../05-fonts-noto-mono_20171026-2_all.deb ...
Unpacking fonts-noto-mono (20171026-2) ...
Selecting previously unselected package fonts-texgyre.
Preparing to unpack .../06-fonts-texgyre_20160520-1_all.deb ...
Unpacking fonts-texgyre (20160520-1) ...
Selecting previously unselected package javascript-common.
Preparing to unpack .../07-javascript-common_11_all.deb ...
Unpacking javascript-common (11) ...
Selecting previously unselected package libcupsfilters1:amd64.
```

```
Preparing to unpack .../08-libcupsfilters1_1.20.2-0ubuntu3.1_amd64.deb ...
Unpacking libcupsfilters1:amd64 (1.20.2-0ubuntu3.1) ...
Selecting previously unselected package libcupsimage2:amd64.
Preparing to unpack .../09-libcupsimage2_2.2.7-1ubuntu2.8_amd64.deb ...
Unpacking libcupsimage2:amd64 (2.2.7-1ubuntu2.8) ...
Selecting previously unselected package libijs-0.35:amd64.
Preparing to unpack .../10-libijs-0.35_0.35-13_amd64.deb ...
Unpacking libijs-0.35:amd64 (0.35-13) ...
Selecting previously unselected package libjbig2dec0:amd64.
Preparing to unpack .../11-libjbig2dec0_0.13-6_amd64.deb ...
Unpacking libjbig2dec0:amd64 (0.13-6) ...
Selecting previously unselected package libgs9-common.
Preparing to unpack .../12-libgs9-common_9.26~dfsg+0-0ubuntu0.18.04.14_all.deb
...
Unpacking libgs9-common (9.26~dfsg+0-0ubuntu0.18.04.14) ...
Selecting previously unselected package libgs9:amd64.
Preparing to unpack .../13-libgs9_9.26~dfsg+0-0ubuntu0.18.04.14_amd64.deb ...
Unpacking libgs9:amd64 (9.26~dfsg+0-0ubuntu0.18.04.14) ...
Selecting previously unselected package libjs-jquery.
Preparing to unpack .../14-libjs-jquery_3.2.1-1_all.deb ...
Unpacking libjs-jquery (3.2.1-1) ...
Selecting previously unselected package libkpathsea6:amd64.
Preparing to unpack .../15-libkpathsea6_2017.20170613.44572-8ubuntu0.1_amd64.deb
...
Unpacking libkpathsea6:amd64 (2017.20170613.44572-8ubuntu0.1) ...
Selecting previously unselected package libpotrace0.
Preparing to unpack .../16-libpotrace0_1.14-2_amd64.deb ...
Unpacking libpotrace0 (1.14-2) ...
Selecting previously unselected package libptexenc1:amd64.
Preparing to unpack .../17-libptexenc1_2017.20170613.44572-8ubuntu0.1_amd64.deb
...
Unpacking libptexenc1:amd64 (2017.20170613.44572-8ubuntu0.1) ...
Selecting previously unselected package rubygems-integration.
Preparing to unpack .../18-rubygems-integration_1.11_all.deb ...
Unpacking rubygems-integration (1.11) ...
Selecting previously unselected package ruby2.5.
Preparing to unpack .../19-ruby2.5_2.5.1-1ubuntu1.10_amd64.deb ...
Unpacking ruby2.5 (2.5.1-1ubuntu1.10) ...
Selecting previously unselected package ruby.
Preparing to unpack .../20-ruby_1%3a2.5.1_amd64.deb ...
Unpacking ruby (1:2.5.1) ...
Selecting previously unselected package rake.
Preparing to unpack .../21-rake_12.3.1-1ubuntu0.1_all.deb ...
Unpacking rake (12.3.1-1ubuntu0.1) ...
Selecting previously unselected package ruby-did-you-mean.
Preparing to unpack .../22-ruby-did-you-mean_1.2.0-2_all.deb ...
Unpacking ruby-did-you-mean (1.2.0-2) ...
Selecting previously unselected package ruby-minitest.
```

```
Preparing to unpack .../23-ruby-minitest_5.10.3-1_all.deb ...
Unpacking ruby-minitest (5.10.3-1) ...
Selecting previously unselected package ruby-net-telnet.
Preparing to unpack .../24-ruby-net-telnet_0.1.1-2_all.deb ...
Unpacking ruby-net-telnet (0.1.1-2) ...
Selecting previously unselected package ruby-power-assert.
Preparing to unpack .../25-ruby-power-assert_0.3.0-1_all.deb ...
Unpacking ruby-power-assert (0.3.0-1) ...
Selecting previously unselected package ruby-test-unit.
Preparing to unpack .../26-ruby-test-unit_3.2.5-1_all.deb ...
Unpacking ruby-test-unit (3.2.5-1) ...
Selecting previously unselected package libruby2.5:amd64.
Preparing to unpack .../27-libruby2.5_2.5.1-1ubuntu1.10_amd64.deb ...
Unpacking libruby2.5:amd64 (2.5.1-1ubuntu1.10) ...
Selecting previously unselected package libsynctex1:amd64.
Preparing to unpack .../28-libsynctex1_2017.20170613.44572-8ubuntu0.1_amd64.deb
...
Unpacking libsynctex1:amd64 (2017.20170613.44572-8ubuntu0.1) ...
Selecting previously unselected package libtexlua52:amd64.
Preparing to unpack .../29-libtexlua52_2017.20170613.44572-8ubuntu0.1_amd64.deb
...
Unpacking libtexlua52:amd64 (2017.20170613.44572-8ubuntu0.1) ...
Selecting previously unselected package libtexluajit2:amd64.
Preparing to unpack
.../30-libtexluajit2_2017.20170613.44572-8ubuntu0.1_amd64.deb ...
Unpacking libtexluajit2:amd64 (2017.20170613.44572-8ubuntu0.1) ...
Selecting previously unselected package libzzip-0-13:amd64.
Preparing to unpack .../31-libzzip-0-13_0.13.62-3.1ubuntu0.18.04.1_amd64.deb ...
Unpacking libzzip-0-13:amd64 (0.13.62-3.1ubuntu0.18.04.1) ...
Selecting previously unselected package lmodern.
Preparing to unpack .../32-lmodern_2.004.5-3_all.deb ...
Unpacking lmodern (2.004.5-3) ...
Selecting previously unselected package preview-latex-style.
Preparing to unpack .../33-preview-latex-style_11.91-1ubuntu1_all.deb ...
Unpacking preview-latex-style (11.91-1ubuntu1) ...
Selecting previously unselected package t1utils.
Preparing to unpack .../34-t1utils_1.41-2_amd64.deb ...
Unpacking t1utils (1.41-2) ...
Selecting previously unselected package tex-gyre.
Preparing to unpack .../35-tex-gyre_20160520-1_all.deb ...
Unpacking tex-gyre (20160520-1) ...
Selecting previously unselected package texlive-binaries.
Preparing to unpack .../36-texlive-
binaries_2017.20170613.44572-8ubuntu0.1_amd64.deb ...
Unpacking texlive-binaries (2017.20170613.44572-8ubuntu0.1) ...
Selecting previously unselected package texlive-base.
Preparing to unpack .../37-texlive-base_2017.20180305-1_all.deb ...
Unpacking texlive-base (2017.20180305-1) ...
```

```
Selecting previously unselected package texlive-fonts-recommended.
Preparing to unpack .../38-texlive-fonts-recommended_2017.20180305-1_all.deb ...
Unpacking texlive-fonts-recommended (2017.20180305-1) ...
Selecting previously unselected package texlive-plain-generic.
Preparing to unpack .../39-texlive-plain-generic_2017.20180305-2_all.deb ...
Unpacking texlive-plain-generic (2017.20180305-2) ...
```

[ ]:

```
!jupyter nbconvert --to pdf IDVE_EXAM_Q2.ipynb
```

[ ]:

```

```