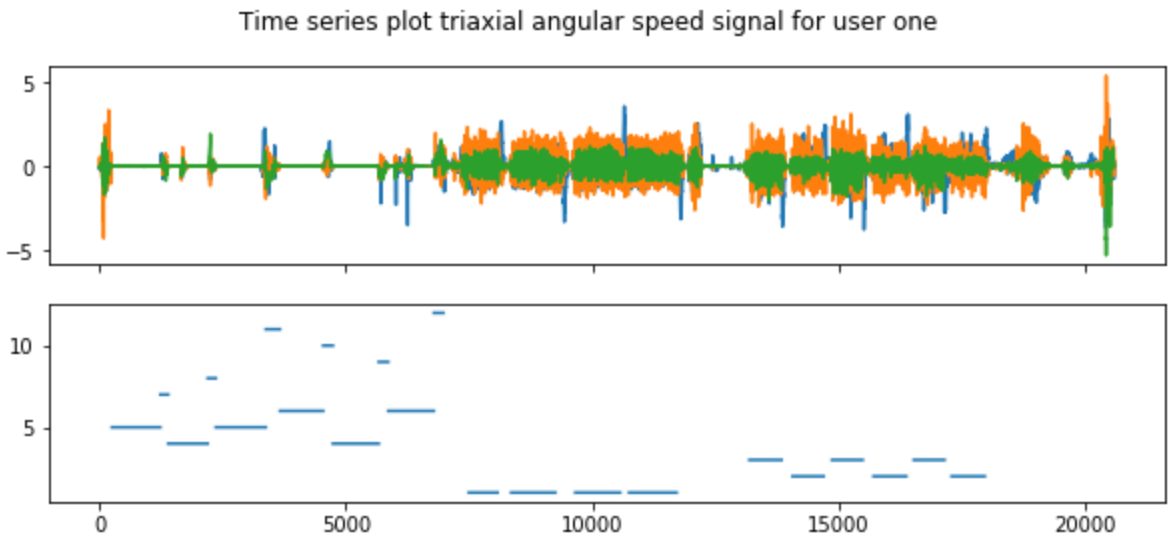
Multi Classification model of Human activity prediction for smartphone data sets

Problem:
Nowadays a lot of people wear smart devices, thosegnal data collected from sensors can be used to predict human activity(walking,sitting,lying,walking stairs and also pose transition activities). In some situations based on their pattern, we may even be able to identify different age group activities, or further the individual participants from their activity styles , which may help to detect early signals of sickness.and guide people to live in a healthy style. In this Analysis ,we only try to resolve the first step, predict human activity using the signal data. Below is the dataset we are going to use.

Data: UCI
http://archive.ics.uci.edu/ml/datasets/Smartphone-Based+Recognition+of+Human+Activities+and+Postural+Transitions
Sensor signals from smartphones of a group of 30 people performed a protocol of activities(such as walking,sitting ,lying and walking stairs) were collected and pre-processed.

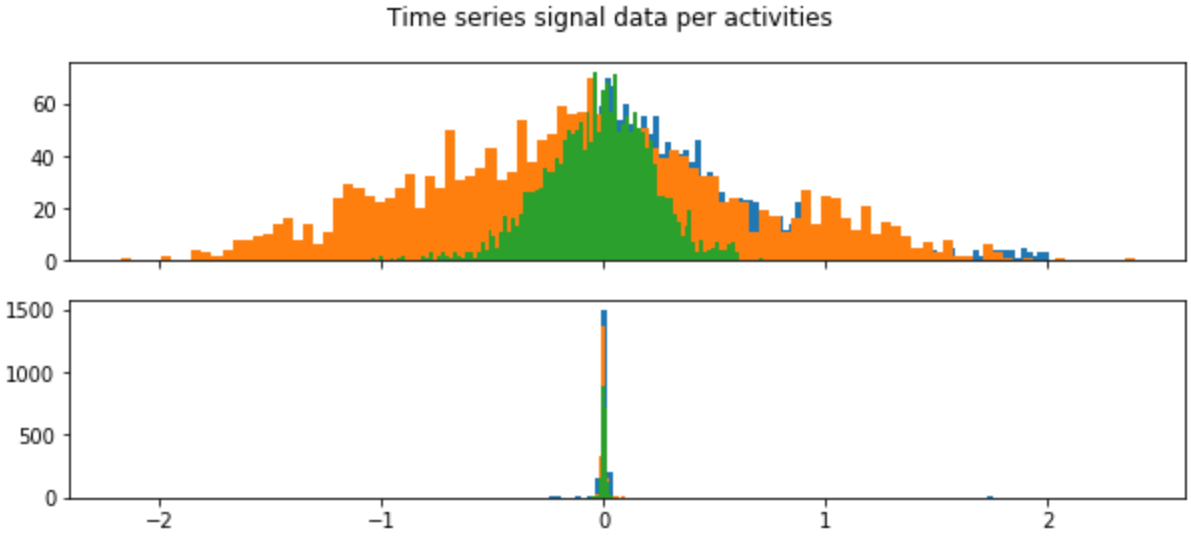
1.Data wrangling:
This datasets has two separate datasets,one is for time series models, one is for regular models. Let's explore both datasets a little bit.
Dataset1: Raw triaxial signals from the accelerometer and gyroscope of all the trials with participants
a)Plot Time Series Data for One Subject:

The raw triaxial angular speed signal for the experiment number 01 and associated to the user number 01. Every point is one angular velocity sample (three axis) captured at a frequency of 50Hz. The second plot is the class of activity it associates, from the plot we can see that walking(upstair and downstair) involved intense signal, the transition pose or laying pose, signal is tend to 0.



b)Plot Histograms Per Activity

Below is the distribution plot of gyro signal per activity. Compare walking upstair and laying activity,we can see that although they all follow normal distribution ,but walking upstair has a larger standard deviation



Dataset2: Records of activity windows. Each one composed of:

- A 561-feature vector with time and frequency domain variables.
- Its associated activity label.
- An identifier of the subject who carried out the experiment

In this project we will build two classification on dataset2 using random forest and logistic regression using Dataset2

1)Load Data

we will download the data from the above link,used pandas function pd.read_csv to load it

```
In [40]: X_train.head()
```

Out[40]:

	tBodyAcc-Mean-1	tBodyAcc-Mean-2	tBodyAcc-Mean-3	tBodyAcc-STD-1	tBodyAcc-STD-2	tBodyAcc-STD-3	tBodyAcc-Mad-1	tBodyAcc-Mad-2	tBodyAcc-Mad-3	tBodyAcc-Max-1	...	fBodyGyroJerkMag-MeanFreq-1	fBodyGyroJ Ske
0	0.043580	-0.005970	-0.035054	-0.995381	-0.988366	-0.937382	-0.995007	-0.988816	-0.953325	-0.794796	...	-0.012236	-(
1	0.039480	-0.002131	-0.029067	-0.998348	-0.982945	-0.971273	-0.998702	-0.983315	-0.974000	-0.802537	...	0.202804	-(
2	0.039978	-0.005153	-0.022651	-0.995482	-0.977314	-0.984760	-0.996415	-0.975835	-0.985973	-0.798477	...	0.440079	-(
3	0.039785	-0.011809	-0.028916	-0.996194	-0.988569	-0.993256	-0.996994	-0.988526	-0.993135	-0.798477	...	0.430891	-(
4	0.038758	-0.002289	-0.023863	-0.998241	-0.986774	-0.993115	-0.998216	-0.986479	-0.993825	-0.801982	...	0.137735	-(

5 rows × 561 columns

X_train has 7767 records and 561 columns, y_train has 12 different labels

```
In [41]: y_label=pd.read_csv('activity_labels.txt',header=None)
         y_label
```

Out[41]:

	0
0	1 WALKING
1	2 WALKING_UPSTAIRS
2	3 WALKING_DOWNSTAIRS
3	4 SITTING
4	5 STANDING
5	6 LAYING
6	7 STAND_TO_SIT
7	8 SIT_TO_STAND
8	9 SIT_TO_LIE
9	10 LIE_TO_SIT
10	11 STAND_TO_LIE
11	12 LIE_TO_STAND

2)Missing data

The null analysis shows there is no missing data for this dataset,so we don’t need to fill the missing data

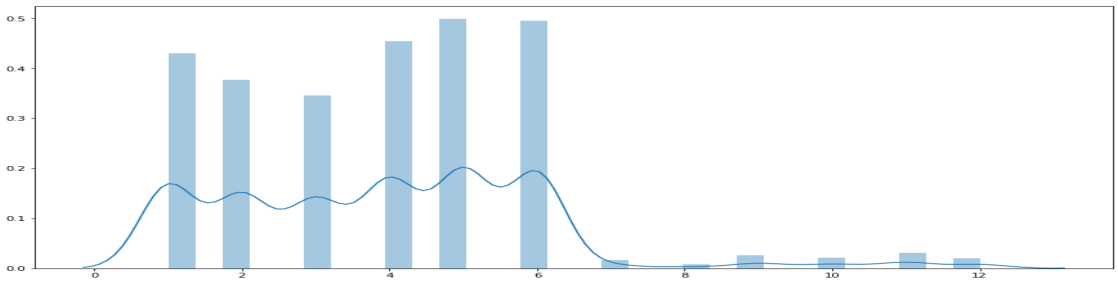
```
In [4]: total=dataset.isnull().sum().sort_values(ascending=False)
        percent = (dataset.isnull().sum()/dataset.isnull().count()).sort_values(ascending=False)
        missing_data = pd.concat([total, percent], axis=1, keys=['Total', 'Percent'])
        missing_data.head()
```

Out[4]:

	Total	Percent
560	0	0.0
183	0	0.0
189	0	0.0
188	0	0.0
187	0	0.0

3) **Data balance**

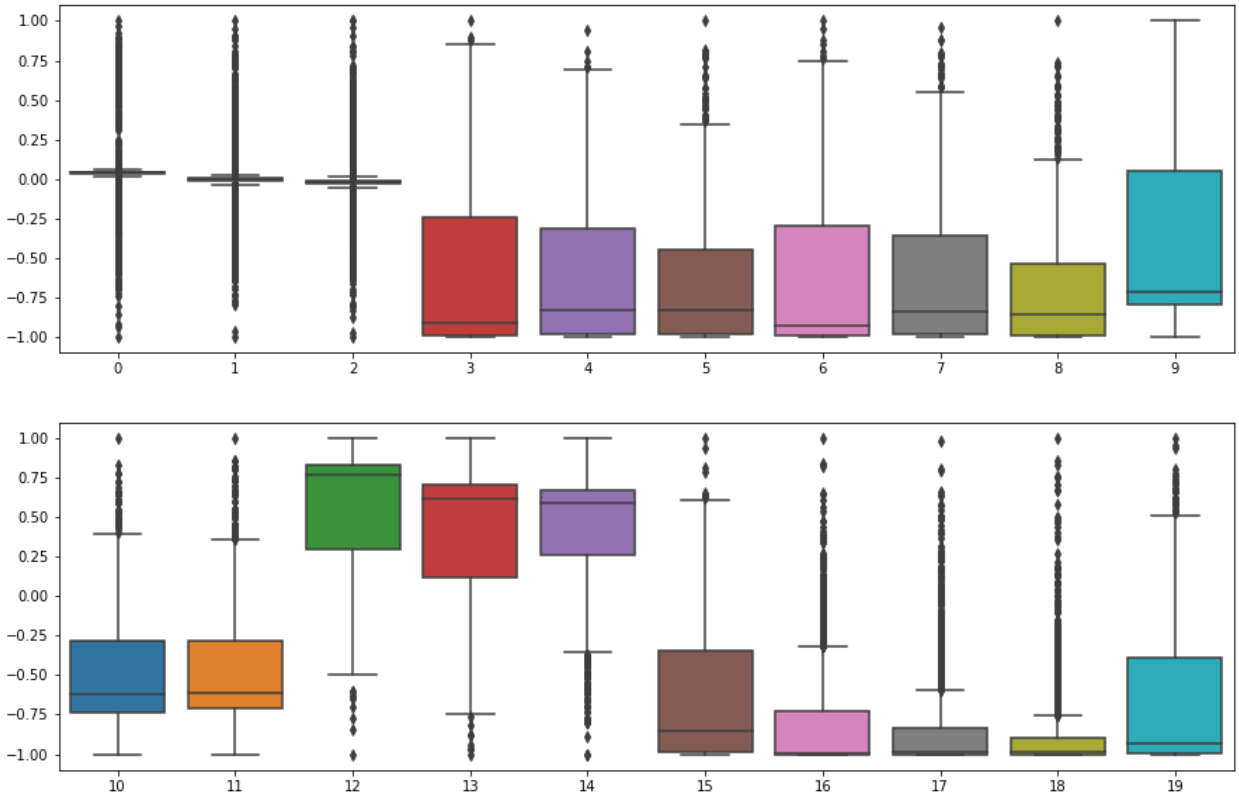
The barplot shows our dataset has 6 major classes and 6 minor classes, it is unbalanced, so we applied SMOTE to resample the data



4) Outlier

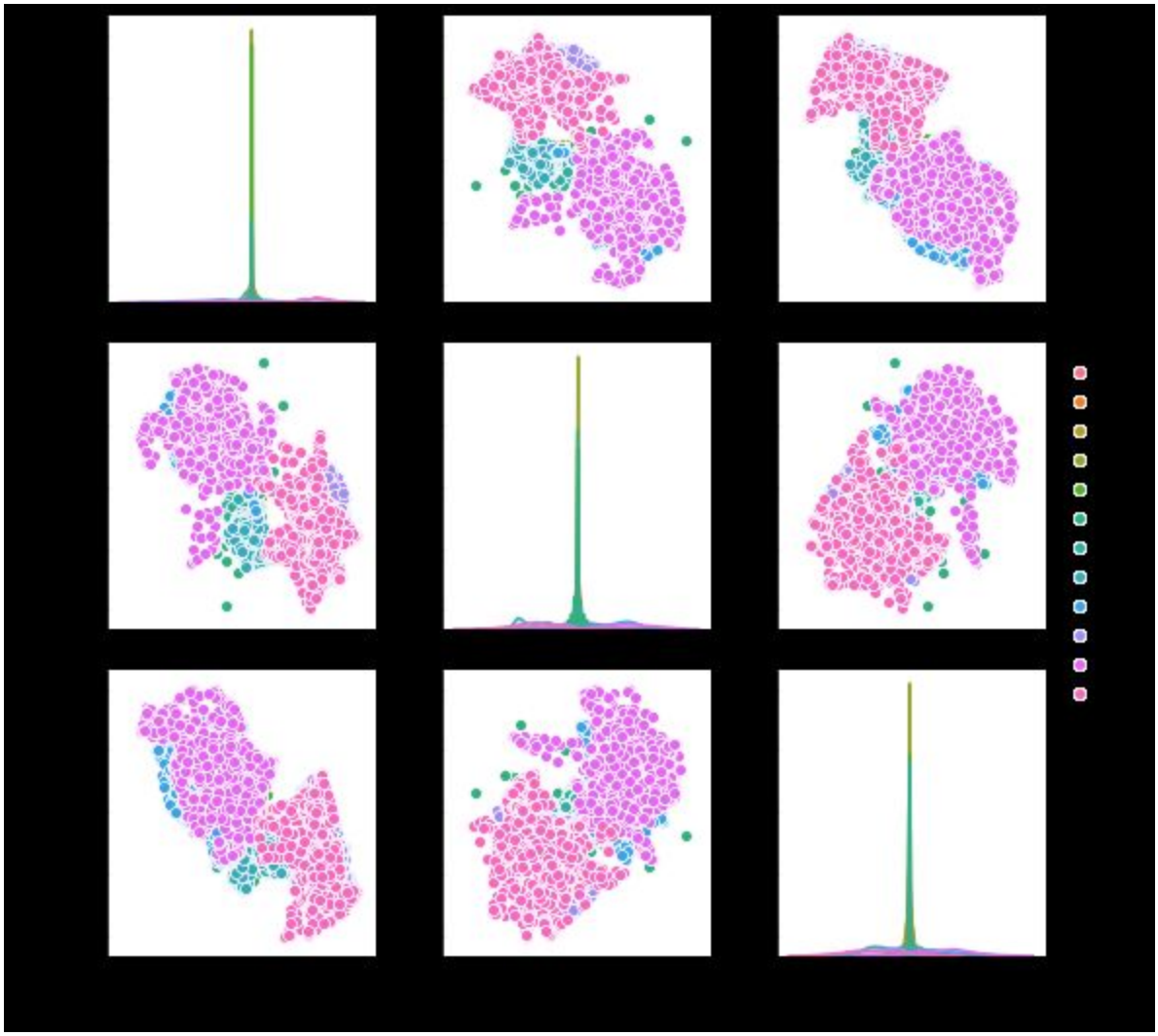
According to the boxplot , we can see that there are a lot of features having outliers

For example feature 8 has a lot of larger outliers,we refill the outlier with 75% value ,feature 13 has smaller outliers ,we will refill those with 25% value.



5)**Correlation**

The results of plot pairs of the first 3 numeric variables shows some variables are correlated with each other , for example feature 0 and feature 1



6)Null hythophis

There is no significent difference between below two sets:

Set a) feature 2 value larger than 0.5, y belong to class 4 (standing pose)

Set b) feature 2 value less than 0.5, y not belong to class 4(standing pose)

```
new_dfyy=len(new_df[(new_df[2]>0.5)&(new_df['0_y']==4)])
new_dfyn=len(new_df[(new_df[2]>0.5)&(new_df['0_y']!=4)])
new_dfny=len(new_df[(new_df[2]<0.5)&(new_df['0_y']==4)])
new_dfnn=len(new_df[(new_df[2]<0.5)&(new_df['0_y']!=4)])
```

```
import scipy.stats as stats
oddsratio, pvalue = stats.fisher_exact([[new_dfyy,new_dfyn], [new_dfny,
new_dfnn]])
pvalue
```

Out[79]:

7.363006721969925e-25

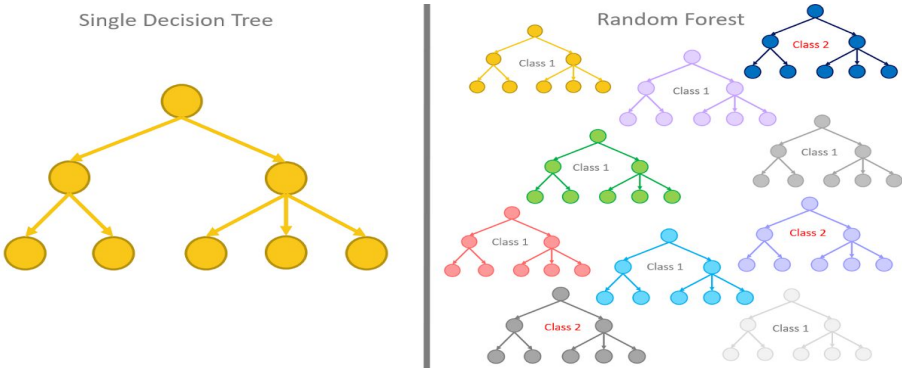
The pvalue is 7.363006721969925e-25 < 0.05, so the null hythophis is rejected, there is a big chance if feature 2>0.5 ,it will belong to class 4.

There is no missing data for our dataset, and it is all numeric fields. We resample it into a more balanced dataset using smoke technique. We also refill the outlier with 25% or 75% value. The pairbox shows some features are correlated with each other. When feature 2>0.5, there is a big chance ,the final class will be 4.The data is already scaled,now it is ready for our model

2.Phase two:in-depth Analysis(machine learning)
1)Model selection:

Random Forest is an ensemble of randomized decision trees model: simple,efficient,versatile and also helps prevent overfitting

- Random Forests are less influenced by outliers ,can handle noisy data
- no assumptions about the underlying distribution of data, and can implicitly handle collinearity in features
- Random Forests can be used for feature selection
- "robust": can work with practically any kind of data, when mixing categorical and numerical features, or mixing completely different ranges of values, no need to scaling



Our dataset has a lot of outliers , and it has 561 features, some features seem lineally related to each other. The classes are imbalanced, So Random Forest could be a good fit even without data processing

2)Baseline Model:Multinomial logistic regression

3)Train the Model:

- Import randomforestClassifier from sklearn
- Fit the train data for the classifier
- Predict the test data

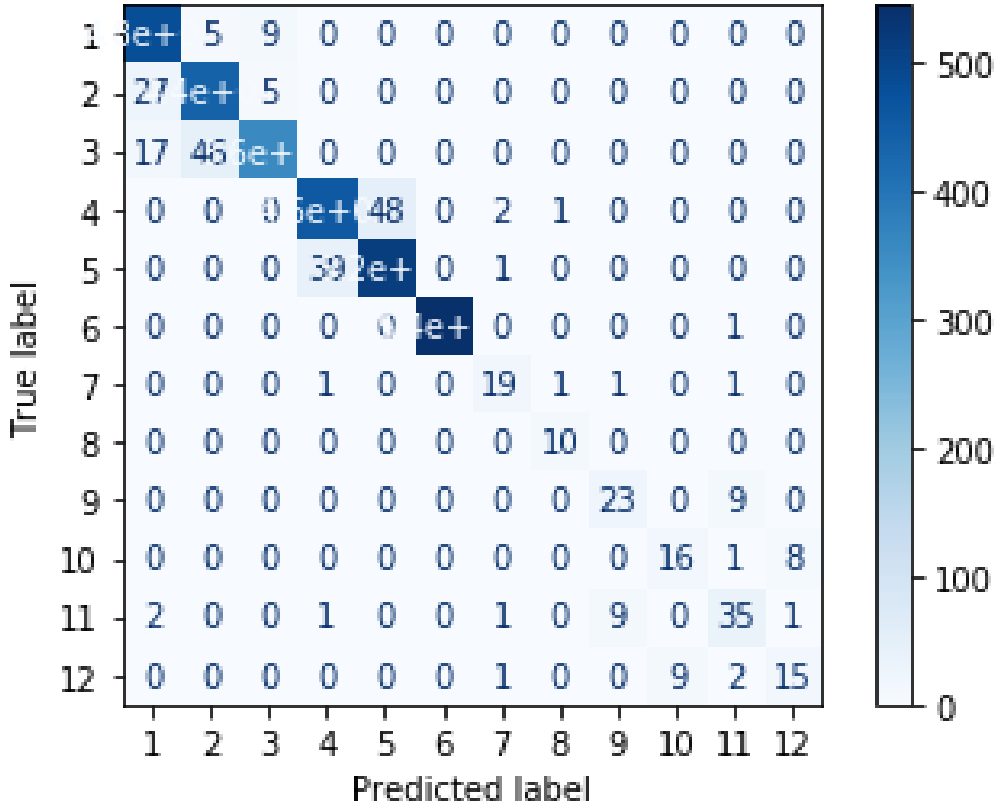
4) Model Optimization: the number of decision trees in the forest and the number of features considered by each tree when splitting a node was tuned using **GridSearchCV()** which combine gridsearch and K-folder cross validation to select best parameters and avoid overfitting .

The best value we get for (max_depth,n_estimators,max_feature) are (18,360,10)

5)Evaluation Metrics:

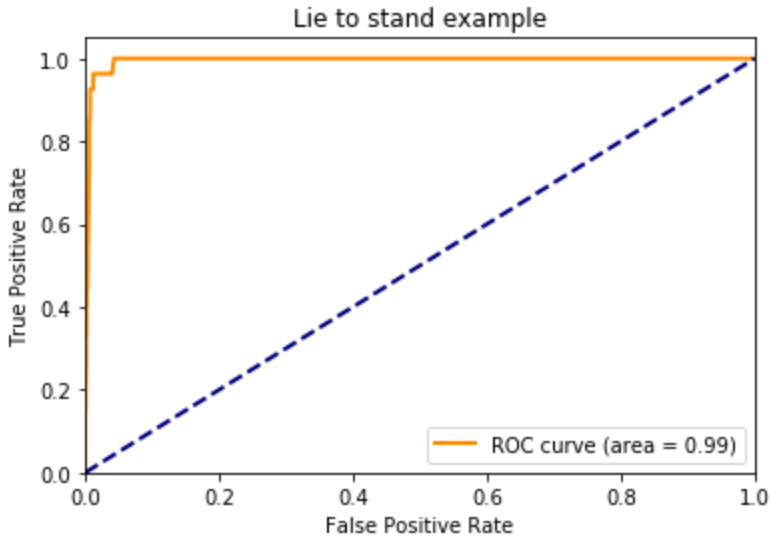
Accuracy_score: good for uniformly distributed class ,after resample our data,may still be a good evaluation metrics

Confusion Matrix: Below confusion Matrix shows that the classifier has very good predictions on the 6 Major class(walking,walking_upstairs,walking_downstairs,sitting, laying), has less performance on the transition poses. both false positive and false negative case are exist, especially for class 10 and 12, it hard to seperate 'lie to sit' from 'lie to stand' which make sense



Roc_auc_score: Compute Area Under the Receiver Operating Characteristic Curve (ROC AUC) from prediction scores. By 'ovr' mode , we got score value `0.995625`

ROC curves: Below is the ROC for last class ‘lie to stand’ transition pose, we can see it still has good AUC score.



Classification report:

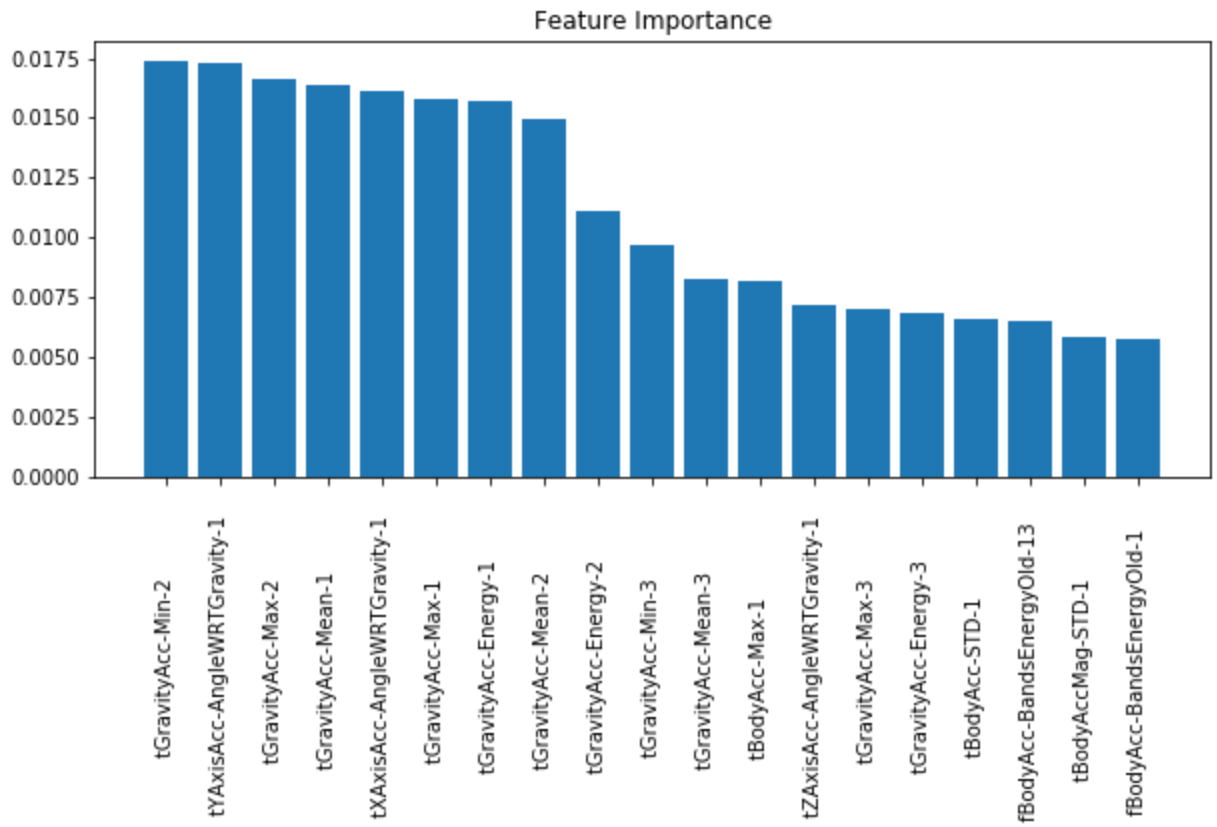
	precision	recall	f1-score	support
1	0.90	0.98	0.94	496
2	0.89	0.93	0.91	471
3	0.96	0.84	0.89	420
4	0.96	0.88	0.92	508
5	0.90	0.97	0.94	556
6	1.00	1.00	1.00	545
7	0.74	0.74	0.74	23
8	0.82	0.90	0.86	10
9	0.64	0.72	0.68	32
10	0.65	0.68	0.67	25
11	0.70	0.57	0.63	49
12	0.67	0.52	0.58	27
accuracy			0.92	3162
macro avg	0.82	0.81	0.81	3162
weighted avg	0.92	0.92	0.92	3162

We can see for class 6, its prediction is 100%, the classfiler is good to predict laying pose. Standing pose has good recall than precision, in opposite, walking downstairs has good precision than recall, for transition poses both are not good enough.

6) Feature selection

PCA:Since our data has 560 features, PCA analysis shows that if we can reduce the dimension to 14 and we still can keep 80% varierious

Feature_importantce: below is top 20 important features, it sees tgr features matters



Final thoughts: Both LR and RF are good classifiers for our data, Logistic Regression seems better, that might because the dataset be normal or binomial distribution and features are mostly independent. We may tune LR as our final model. We also can build

deep learning models on Dataset1(time series data), it will be interesting to compare those results.