Posture Reconstruction

Aayush Gupta - 2017125
Bhavya Srivastava - 2017037
Raghav Gupta - 2017178

Introduction & Related Work

1. Predict posture using tags attached to ankles, chest and belt.
2. Can be used to check if a person is in danger or not.
3. Useful for elderly and handicapped people

Related work:
Paper - Classification of posture reconstruction with univariate time series data type (2018 international conference on SIET). It used kNN.
In this, k=5 based on similarity measurement using euclidean distance.
Accuracy achieved = 99.5%

Dataset and Evaluation

The dataset has 164860 instances, and 8 attributes (including the output class).
Thus, there are 7 features:
1) Sequence Name: 5 sequences (01, 02, 03, 04, 05) for 5 people (A, B, C, D, E)
2) Tag identificator: To identify different parts of the body
3) timestamp (Numeric) all unique
4) date, FORMAT = dd.MM.yyyy HH:mm:ss:SSS (Date)
5) x coordinate of the tag (Numeric)
6) y coordinate of the tag (Numeric)
7) z coordinate of the tag (Numeric)

The number of output classes is 11, which are as follows:
Walking, falling, lying down, lying, sitting down, sitting, standing up from lying, on all fours, sitting on the ground, standing up from sitting, standing up from sitting on the ground.

Evaluation metrics:
We are going to use Confusion matrices, ROC curves, F1 score, Accuracy, precision, recall etc. as evaluation metrics.

Analysis and Progress

The challenges faced by us so far are:

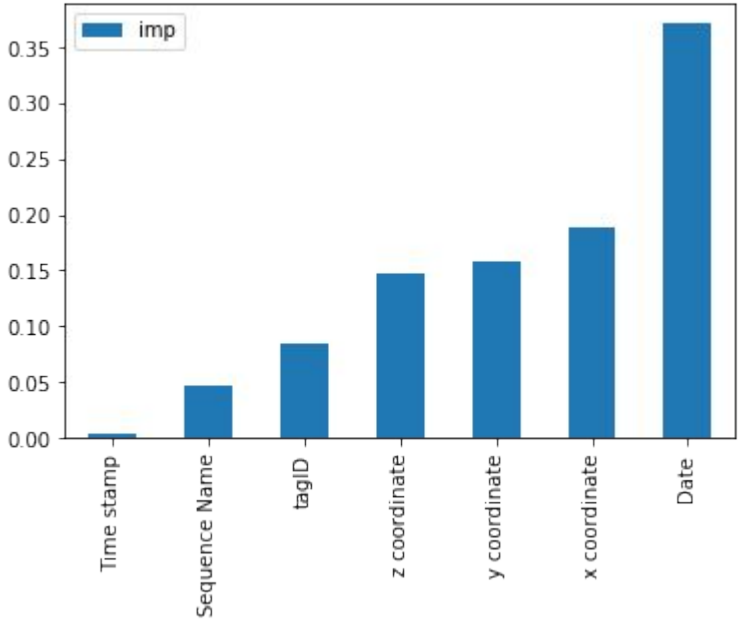
1) Accounting for the time series nature of the data
2) Applying new techniques previously not known

Learning methods chosen and their accuracies:

1) Logistic Regression - 41%
2) Decision Tree - 85%
3) Random Forest - 92%
4) Support Vector Machines -
5) K-Nearest-Neighbours - 99.6%
6) MLP - 77.19%
7) Gradient Boosting Classifier - 92.1%

Accuracy for K-NN is 99.6%, which is the higher than all other models, and is comparable to other state-of-the-art-solutions that are much more complex.

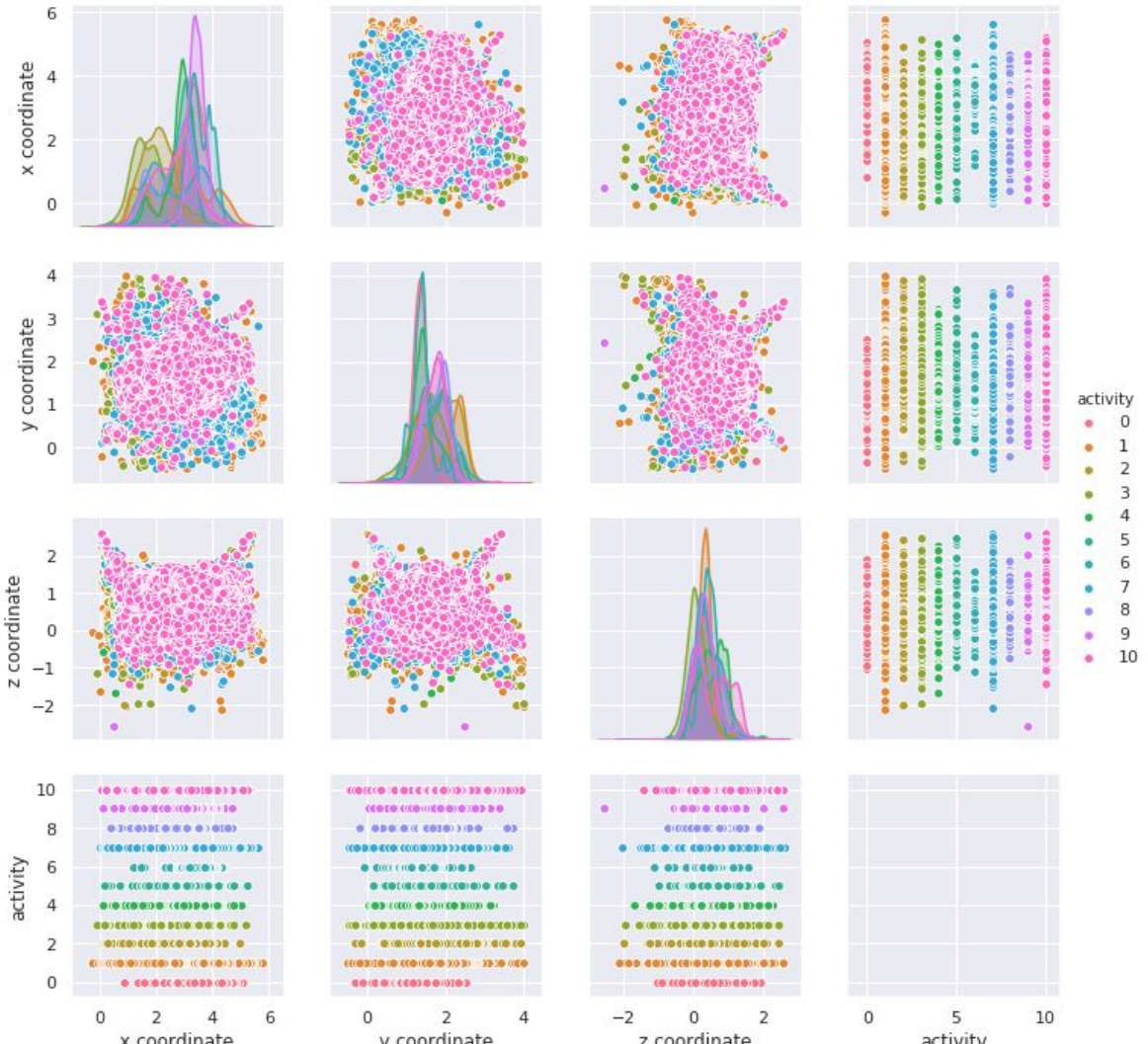
Feature Importances



Based on the feature importance we decided to drop the timestamp field from our dataset.

Since the problem is a time series problem, we see that the feature importance for date, which is the time-series factor, is the highest.

PAIRPLOT


x coordinate
y coordinate
z coordinate
activity
x coordinate
y coordinate
z coordinate
activity
activity
0
1
2
3
4
5
6
7
8
9
10

Heat Map


Correlation between different features
Sequence Name 1.00 -0.06 -0.67 -0.68 0.04 0.05 -0.02 0.08
tagID -0.06 1.00 0.04 0.04 -0.07 -0.03 0.47 -0.00
Time stamp -0.67 0.04 1.00 0.99 -0.08 -0.02 0.01 -0.05
Date -0.68 0.04 0.99 1.00 -0.10 -0.02 0.00 -0.06
x coordinate 0.04 -0.07 -0.08 -0.10 1.00 -0.35 0.03 -0.02
y coordinate 0.05 -0.03 -0.02 -0.02 -0.35 1.00 0.03 -0.07
z coordinate -0.02 0.47 0.01 0.00 0.03 0.03 1.00 0.21
activity 0.08 -0.00 -0.05 -0.06 -0.02 -0.07 0.21 1.00
Sequence Name
tagID
Time stamp
Date
x coordinate
y coordinate
z coordinate
activity
0.9
0.6
0.3
0.0
−0.3
−0.6

Performance Metrics for the KNN Classifier

Confusion matrix:

```
[[ 578    7    0    0    0    0    4    0    0    0    7]
 [   8 10963    6    5    0    0    0   16    0    0    0]
 [   0    5 1220    5    0    0    0    0    0    0   10]
 [   0    4    3 1046    0    0    0    0    0    0    0]
 [   4    0    0    0 5391    5    0    0    2    0    0]
 [   0    0    0    0    7  325    0    0    0    0    2]
 [   1    0    0    0    0    0 2273    0    0    1    0]
 [   0   16    0    0    0    0    0 3547    0    0   12]
 [   0    0    0    0    3    0    0    0  280    0    1]
 [   0    0    0    0    0    0    4    0    0  572    2]
 [   5    0   12    0    0    8    0   17    2    1 6592]]
```

KNN Classification Report

	precision	recall	f1-score	support
0	0.97	0.97	0.97	596
1	1.00	1.00	1.00	10998
2	0.98	0.98	0.98	1240
3	0.99	0.99	0.99	1053
4	1.00	1.00	1.00	5402
5	0.96	0.97	0.97	334
6	1.00	1.00	1.00	2275
7	0.99	0.99	0.99	3575
8	0.99	0.99	0.99	284
9	1.00	0.99	0.99	578
10	0.99	0.99	0.99	6637
accuracy			0.99	32972
macro avg	0.99	0.99	0.99	32972
weighted avg	0.99	0.99	0.99	32972

Performance Metrics for the Gradient Boosting

Confusion matrix:

```
[[ 176 203   0   0  68   0  42  13   0   0  94]
 [  6 9931   4  75 271   3  61 235   1   0 411]
 [  0 379 299  32  47   1   2 176   0   0 304]
 [  0 135   1 698  26   0  13  40   0   0 140]
 [  3 402   3   1 4259  3  35  57   1   0 638]
 [  1  22   0   0  71 137   1   7   0   0  95]
 [  1  91   0   0  26   1 2110  21   0   3  22]
 [  8 1105  14  43 129   0  95 1407  0   9 765]
 [  0  13   0   0  44   0   0   2 159   0  66]
 [  1  51   0   0  24   0  71  36   0 325  70]
 [  7 508   6  21 422   4  39 255   2   3 5370]]
```

	precision	recall	f1-score	support
0	0.95	0.81	0.88	596
1	0.90	0.97	0.94	10998
2	0.98	0.72	0.83	1240
3	0.98	0.97	0.98	1053
4	0.97	0.98	0.97	5402
5	0.93	0.92	0.92	334
6	0.98	1.00	0.99	2275
7	0.89	0.71	0.79	3575
8	0.99	0.99	0.99	284
9	0.99	0.98	0.99	578
10	0.88	0.91	0.90	6637
accuracy			0.92	32972
macro avg	0.95	0.90	0.92	32972
weighted avg	0.92	0.92	0.92	32972