**FINAL REPORT**

MACHINE LEARNING MODEL FOR 3D HAND POSTURE RECOGNITION

*(Dataset:Mo-Cap Hand Postures Data Set UCI Repository)*

Problem Statement: Developing a Machine Learning model aimed at recognition of hand postures from the 3-Dimensional data points corresponding to reflective markers on a left handed glove

**Mentor Submitted by**

Mr. Naveen Divyam Tripathi

Shreya Goyal

Shweta Chopra Shahrukh Khan

Yash Bagaria

**Premable**

**Hand Posture Recognition:**

Today Hand posture recognition is gradually becoming a well-known means of Human Computer Interaction (HCI). It is efficiently making our live simpler and effective by achieving “hands free” interaction through eliminating the need to hold or press the device. The most realistic examples can be using hand postures to operate our smart phones, to play videogames like Xbox and to operate highly secure facilities. Industries like Automotive sector, Consumer electronics sector, Gaming sector are flourishing by making use of hand posture recognition daily.

To get the domain knowledge on how these various sectors are developing their hand recognition technique, we obtained data which involved users and their different hand postures which were captured using a Vicon motion capture camera system.

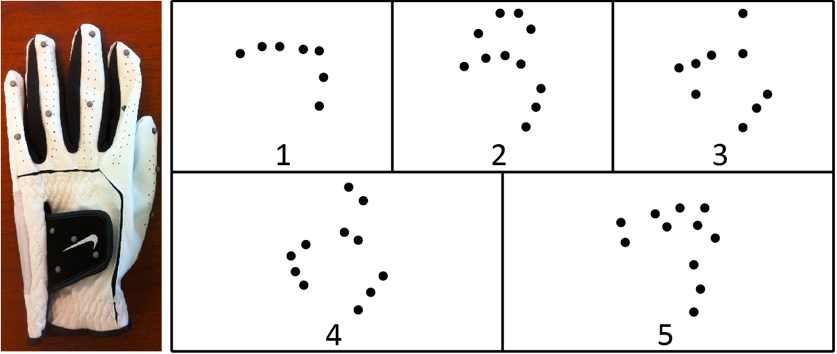
**

Figure 1: The glove used to capture data along with a sample from each class of posture projected onto the local XY plane. The classes are fist (1), stop (2), point with one finger (3), point with two fingers (4), and grab (5)

**Data Monography:**

As already mentioned A Vicon motion capture camera system was used to record 12 users performing 5 hand postures with markers attached to a left-handed glove.

A rigid pattern of markers on the back of the glove was used to establish a local coordinate system for the hand, and 11 other markers were attached to the thumb and fingers of the glove. 3 markers were attached to the thumb with one above the thumbnail and the other two on the knuckles. 2 markers were attached to each finger with one above the fingernail and the other on the joint between the proximal and middle phalanx. Their positions were not explicitly tracked.

Consequently, there is no a priori correspondence between the markers of two given records. In addition, due to the resolution of the capture volume and self-occlusion due to the orientation and configuration of the hand and fingers, many records have missing markers. Extraneous markers were also possible due to artifacts in the Vicon software’s marker reconstruction/recording process and other objects in the capture volume. As a result, the number of visible markers in a record varied considerably.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Data Set Characteristics:** | Multivariate | **Number of Instances:** | 78095 | **Area:** | Computer |
| **Attribute Characteristics:** | Real | **Number of Attributes:** | 38 | **Date Donated** | 2017-01-27 |
| **Associated Tasks:** | Classification, Clustering | **Missing Values?** | Yes | **Number of Web Hits:** | 26195 |

Dataset link:

<https://archive.ics.uci.edu/ml/datasets/Motion+Capture+Hand+Postures>

**Data Pre-Processing:**

The data available here is already pre-processed in the following ways:

* All markers were transformed to the local coordinate system of the record containing them.
* Each transformed marker with a norm greater than 200 mm was removed.
* Any record that contained fewer than 3 markers was removed and each record has at most 12 markers.

**Attributes’ Information:**

1. **Classes:**

5 types of hand postures from 12 users were recorded. Each class is recorded as an integer. The class ID of the given record ranges from 1 to 5 with :

* 1. Fist (with thumb out)
  2. Stop (hand flat)
  3. Point1 (point with pointer finger)
  4. Point2 (point with pointer and middle fingers)
  5. Grab (fingers curled as if to grab)



1. **Features:**

As there were 12 different people whose postures were recorded, the ID of the user that contributed the record is represented in the user column. Other features are X, Y, Z coordinates of all the markers used. No meaning other than as an identifier. Each record is a set.

• ‘Xi’ – Real Number. The x-coordinate of the i-th marker position. ‘i’ ranges from 0 to 11.

• ‘Yi’ - Real Number. The y-coordinate of the i-th marker position. ‘i’ ranges from 0 to 11.

• ‘Zi’ - Real Number. The z-coordinate of the i-th marker position. ‘i’ ranges from 0 to 11.

 The i-th marker of a given record does not necessarily correspond to the i-th marker of a different record. One may randomly permute the visible (i.e. not missing) markers of a given record without changing the set that the record represents. For the sake of convenience, all visible markers of a given record are given a lower index than any missing marker. A class is not guaranteed to have even a single record with all markers visible.

**METHODOLOGY:**

I Data Cleaning

II Exploratory Data Analysis

III Modelling Technique

I Data Cleaning:-

There is a large number of null values present in the dataset represented by ‘?’ symbol(markers not captured in a particular frame (row) due to occlusion were given a ‘?’ value).These ‘?’were converted to ‘0’ value because markers were not occluded from the camera.

**Percentage of Null Values:**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Feature** | **%Null** | **Feature** | **%Null** | **Feature** | **%Null** |
| X0 | 0.00 | X1 | 0.00 | X2 | 0.00 |
| Y0 | 0.00 | Y1 | 0.00 | Y2 | 0.00 |
| Z0 | 0.00 | Z1 | 0.00 | Z2 | 0.00 |
| X3 | 0.88 | X4 | 3.99 | X5 | 16.67 |
| Y3 | 0.88 | Y4 | 3.99 | Y5 | 16.67 |
| Z3 | 0.88 | Z4 | 3.99 | Z5 | 16.67 |
| X6 | 33.09 | X7 | 50.13 | X8 | 60.86 |
| Y6 | 33.09 | Y7 | 50.13 | Y8 | 60.86 |
| Z6 | 33.09 | Z7 | 50.13 | Z8 | 60.86 |
| X9 | 69.31 | X10 | 81.11 | X11 | 99.96 |
| Y9 | 69.31 | Y10 | 81.11 | Y11 | 99.96 |
| Z9 | 69.31 | Z10 | 81.11 | Z11 | 99.96 |

**Inferences:-**

1. Features X8 onwards have a large number of NULL values( more than 50% ) but we will not drop these because of the vital information loss it may cause for a particular class.
2. Instead, we will replace NULL with 0, inherently transferring all the NULL data points(a trio of Xi,Yi, Zi) to origin of the local coordinate axes, nullifying the effect without any information loss.
3. Rest of the values are pre-processed and hence clean enough to perform EDA and modelling on.
4. Scaling of data was not done because each of the features was already on a similar scale.

II Exploratory Data Analysis:-

The data set was loaded in a pandas DataFrame object and a thorough investigation was done to unearth vital information about the data.Then we proceeded with the Univariate analysis on features (performed on the trio trio of Xi,Yi, Zifor a better understanding). Few of the plots generated are presented.

**Univariate Analysis:-**

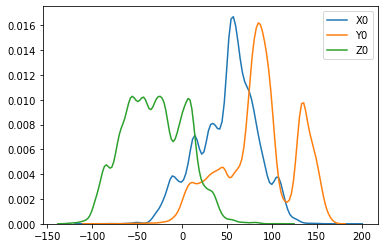


Figure 2: kde plot of the features X0, Y0 and Z0

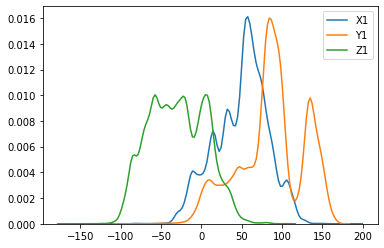


Figure 3: kde plot of the features X1, Y1 and Z1

**Inferences:-**

* The X0 feature has values in the middle range i.e. it ranges from -50 to 150.
* The Y0 feature has values in the highest range i.e. it ranges from -50 to 200.
* The Z0 feature has values in the lowest range i.e. it ranges from -150 to 100.(It has high number of negative values)
* When compared with the above graph which shows the distribution of X0,Y0,Z0 features we can come to a conclusion that (Xi,Yi,Zi) features are very similar.

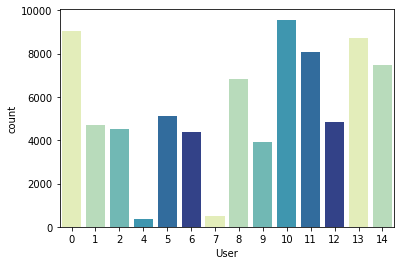
****

Figure 4: Distribution of User using Countplot

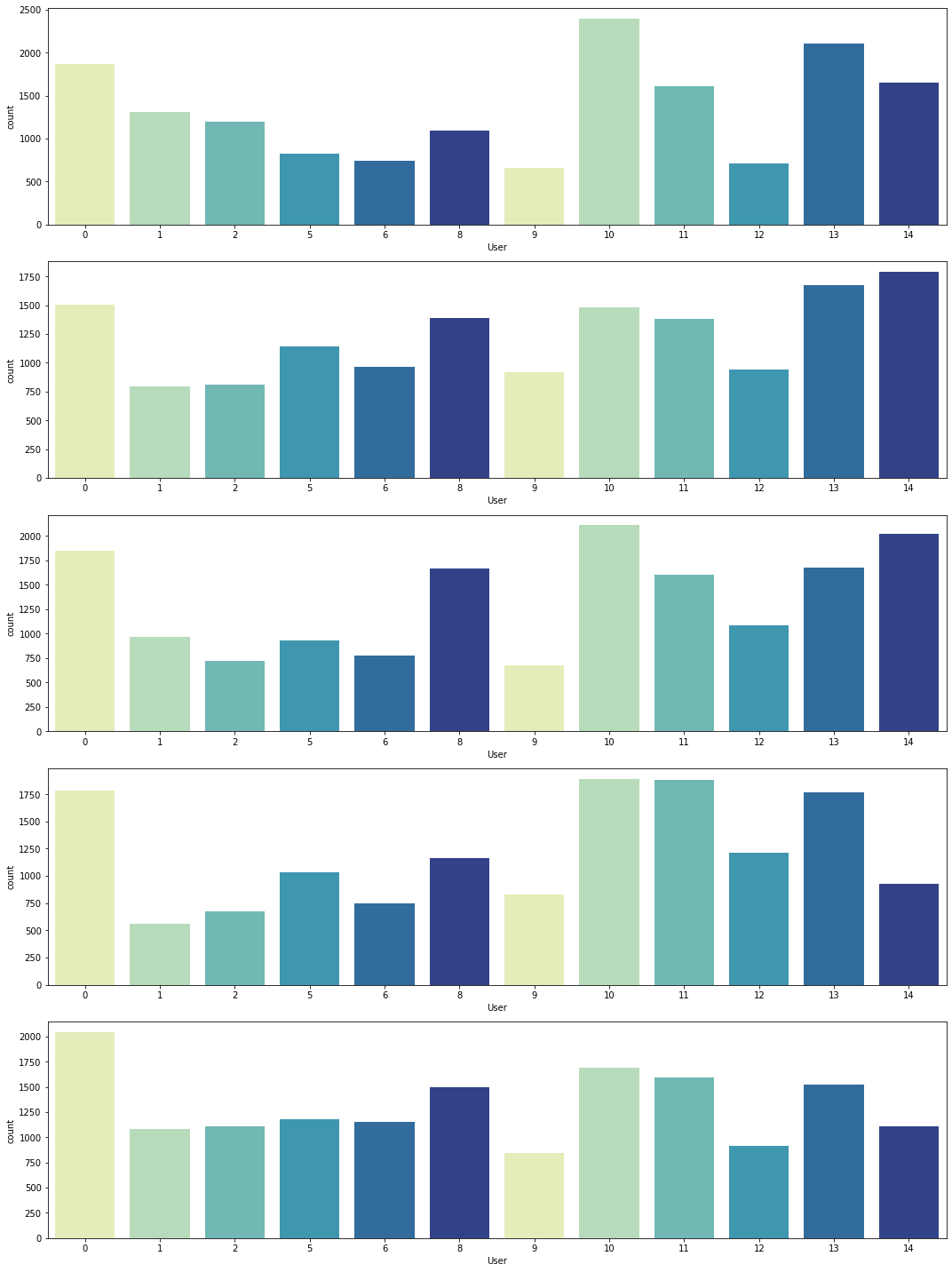
**Bivariate Analysis:-**

Figure 5:Class-Wise Distribution of User

**Inferences:-**

* As it is evident from the subplots above that Users 7 and 4 have only 4 postures namely 1,2,3,4 and 5th posture is missing.
* Also User 7 and User 4 have very less amount of values as compared to others.So we will drop them.

III Modelling Technique:-

Our main objective is to predict the hand postures using classification.

In all we have used 3 approaches to split the train and test.

**A. Our first approach was to split the data into traditional train and test.**

We applied Logistic Regression(One Vs Rest Technique) because our data had 5 classes . Reason behind working with Logistic Regression was that we wanted to keep things simple and clear initially just like Occam Razor suggests.

After noting down the results of LR model ,we went ahead with Decision Tree because it is capable to handle both unordered and unlabelled data efficiently.

Here F1\_score as our evaluation parameter.

**Interpretation:** On using the above mentioned models we found that we get considerably good score.

**Challenge Anticipated :** It raised a doubt in our mind that if we do train-test split in the traditional way there are possibilities that when our model is tested on an unknown data set , it might not perform well. This gave raise to our second approach.

**B. The second approach was to split the data into train and test using one user out method.**

In-order to overcome the above issue we decided to do the train-test split on the basis of one-user out.

We have tested our model by using Logistic Regression(One Vs Rest Technique) and Decision Tree algorithm.

**Interpretation:** On comparing the F1\_score we found that our data was performing well and decision tree gave more efficient results which is why we decided to keep Decision Tree as our base model.

**Challenge Anticipated:** While brain storming, we thought of changing the data upside down and this led to our 3rd approach.

**C. The third approach was to split the data into train and test using one user out method but we transformed the data.**

We created 12 records in transformed data set for 1 row in original data set.

We captured X,Y,Z coordinates along with respective class and row\_label .

We trained our model and made predictions on transformed data set .

Then for each 12 records in transformed data set we took voting measure(mode) for final prediction of original dataset but it was giving unsatisfactory results .

So we stick up with our last approach of leave one user out with missing value imputation as 0.

**Challenge Anticipated:** We anticipated that how is model learning the raw data as the data is unlabelled and unordered. This led to the idea of feature extraction mentioned below.

IV Feature Engineering:-

We gradually realized that since the data is unlabelled and unordered , it does not make any sense to apply machine learning algorithm directly on raw data because model will assume that each point has a supposed correspondence and leads to confusion(garbage-in,garbage-out).

Therefore we created a set of aggregate features in order to avoid above correspondence problem.We extracted the general information about each observation from the dataset and then applied machine learning algorithms on it. Following features were created :-

**Number of Markers:** By exploring the dataset we noticed that number of markers are somehow associated with marker classes as some classes appear to have more missing points than others.

**Mathematics used:**

**Step 1:** Let No. of Non-null values in each row be N.

**Step 2:** So Number of Markers = (N-2)/3

**Note:**

We subtracted N from 2 and divided the result by 3 because we have two columns Class and User which needs to be separated. And then for each marker we have 3 coordinates namely (x,y,z) so we divided it by 3.

**Mean, Variance and Co-variance:** We expect the mean, variance and covariance to display systematically different patterns across different classes according to the shape of each type of hand posture.

**Mathematics used:**

**Mean:** µ = (Sum of terms) / (Number of terms)

**Mean** in each row:-

|  |
| --- |
| X\_mean= mean of all X axes (X0,X1,X2…….X12) |
| Y\_mean=mean of all Y axes (Y0,Y1,Y2……Y12) |
| Z\_mean=mean of all X axes (Z0,Z1,Z2……….Z12) |

**Variance:** σ2 = ( Sum of Squared distance of each term from mean) / Number of terms)

**Variance** in each row:-

|  |
| --- |
| X\_var= variance of all X axes (X0,X1,X2…….X12) |
| Y\_var= variance of all Y axes (Y0,Y1,Y2……Y12) |
| Z\_var=variance of all X axes (Z0,Z1,Z2……….Z12) |

**Co-variance:**  Cov(x,y) = ∑(xi - x\_mean)(yi – y\_mean) / Number of terms)

**Co-variance** in each row:-

|  |
| --- |
| XY\_cov= Cov(X axes values ,Y axes values) |
| YZ\_ cov = Cov(Y axes values, Z axes values) |
| XZ\_ cov =Cov (X axes values,Z axes values) |

**Dimension of the bounding box :** The logic behind creating this feature was that we wanted to measure the degree of expansion that varies between various types of hand postures .This feature is determined by the 1-D distance between the two farthest points in each dimension.(i.e. Range)

**Dimension** in each row:-

|  |
| --- |
| X\_dim= max(X axes values) - min(Y axes values) |
| Y\_ dim = max(Y axes values) - min(Y axes values) |
| Z\_ dim = max(Z axes values) - min(Z axes values) |

**Aggregate Data:** Representation of the dataset after performing Feature Engineering.

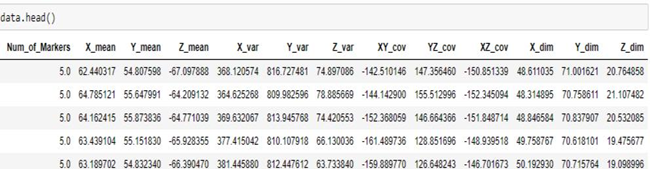


Figure 6: Dataframe First 5 rows

V Data Preparation:-

* We dropped all the records where number of markers where less than 4 as it a very small unit to make any classification.
* Since data was highly unscaled, it was scaled down using standard scaler.

VI Exploratory Data Analysis on Aggregrated Data:-

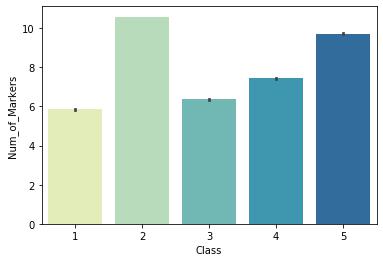
**Bivariate Analysis**

Figure 7: Class Vs Num\_of\_Markers

**Inference:**

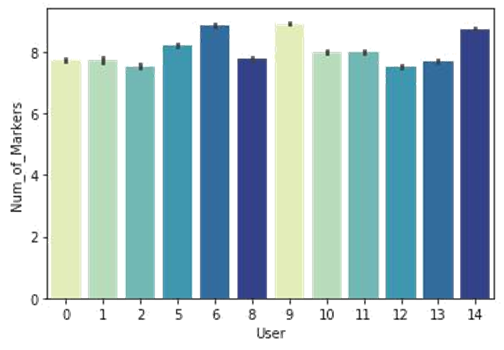
* We noticed here that Class 2(flat hand) is having highest number of visible markers and second highest is Class 5 (grab) which is also having significantly higher number of markers as compared to others.

Figure 8: User Vs Num\_of\_Markers

**Inference:**

* We noticed here that the number of markers were almost similar for most of the users.

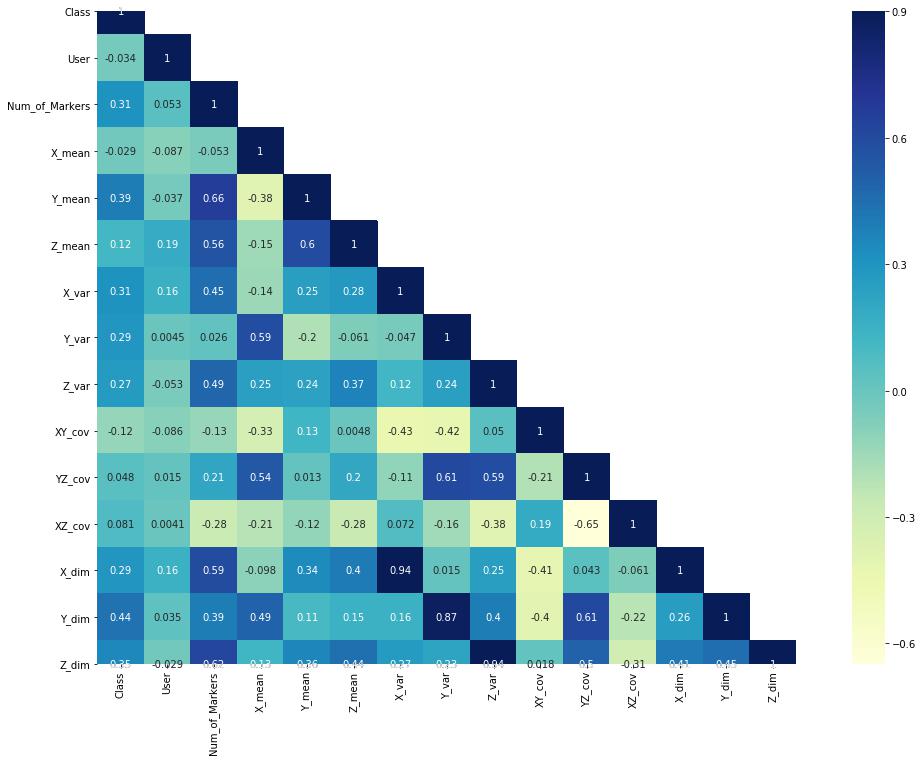


Figure 9: Correlation Graph

**Inference:**

* Highest Correlated feature is Y\_dim and least correlated feature is XZ\_cov with Class.
* There is multicollinearity in our data and we will move forward with it using complex models.

VII Statistical Tests:-

**Variance Inflation Factor(VIF):-**

We applied VIF test to check if there is multicollinearity within features and found that the features are highly correlated but we decided to keep all the features as they contained crucial information and dropping any one of them resulted into the decrease in model’s efficiency.

**Normality Test:-**

We applied Anderson darling test to check whether the data is normal and found that all the features are not normal.

**Significance Test:-**

We applied kruskal test to check the significance of each feature on the tagert and found that all the features are significant.

VIII Application of models:-

Results when we trained and tested using traditional train-test split.

|  |  |
| --- | --- |
| Model Name | Test Scores |
| Logistic Regression(One Vs Rest) | 0.7913384484759948 |
| Logistic Regression(One Vs One) | 0.7733239469428259 |
| Decision Tree | 0.9645208270771177 |
| Random Forest | 0.9911936273822417 |

**Inference:**

We get a very high score of 99%.Possibly the reason can be - We have similar values of different class of different Users on Train Data as well as on the Test Data that's why our Test result is 99%.

To overcome this situation we need to make a methodology that can tell whether the model working good on a unseen data or not. To achieve this we made a user defined function that will make use of User column and will evaluate the model on One-User-Out- Basis.

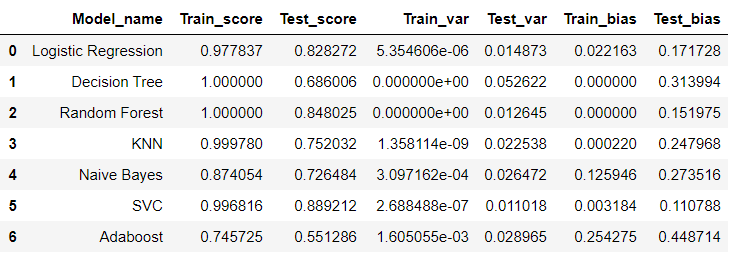
Base Results when we trained and tested using One-User-Out Basis

Figure 10: Result Table

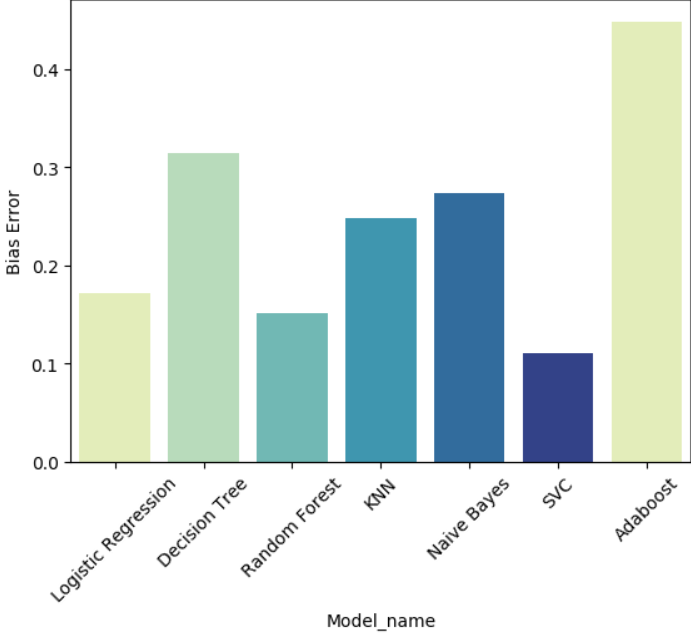


Figure 11: Chart to represent Bias Error for Models

**Inference:-** Min is better : With Default Parameters SVC, RF and LR are working best as compared to others.

Now we will tune the above more models to get the parameters that work the best for each model and hence find the score that we get when the models are run with optimized parameters.

Tunned Results when we trained and tested using One-User-Out Basis

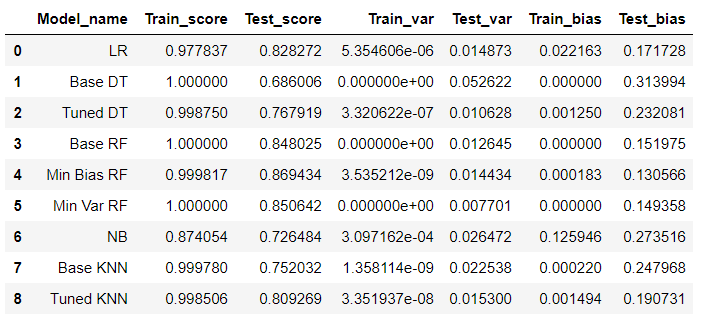
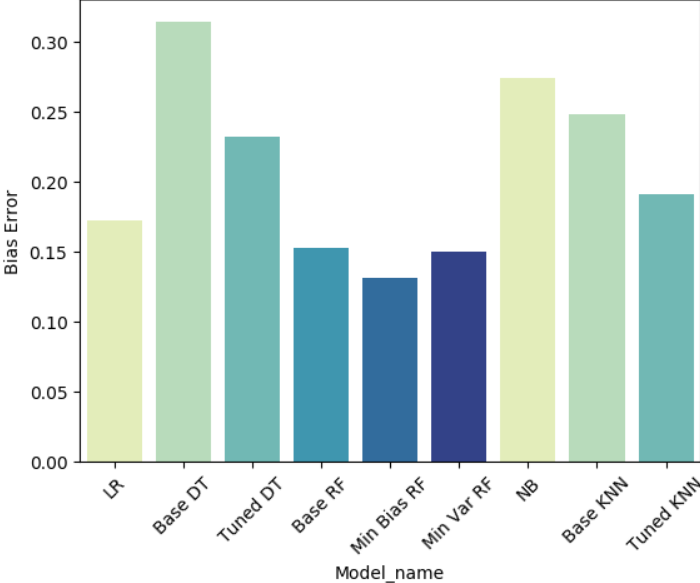
 Figure 12: Tuned Results

Figure 13: Bias Error Comparison for Tuned Models

**Boosting:-**

Now we will perform **Boosting** on the models and see if we can see an improvement in the score.

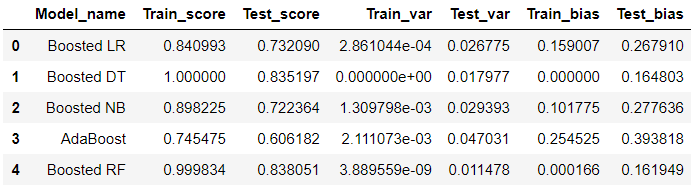


Figure 14: Boosting Results for Models

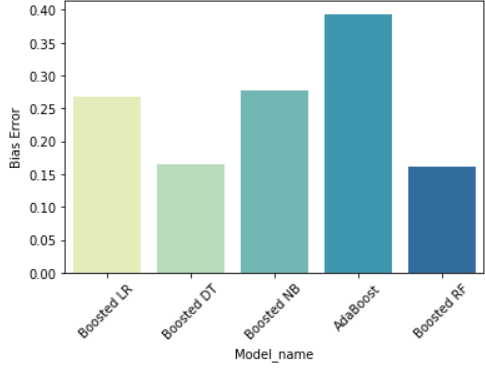


Figure 15: Comparison of Boosted Results for Models

**Bagging:-**

Now we will do bagging on our models to see if performing bagging improves

our scores or not.

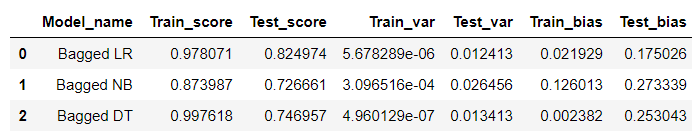


Figure 16: Bagging Results for Models

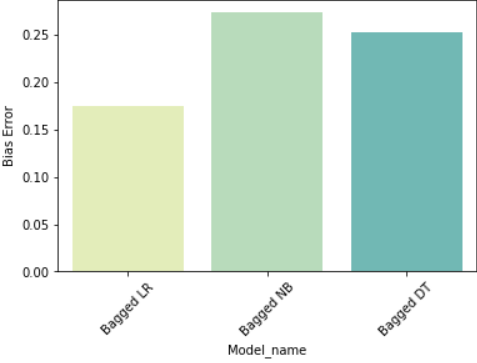


Figure 17:Comparison of Bagging Results for Models

Now we can go ahead and select the best performing models from above.

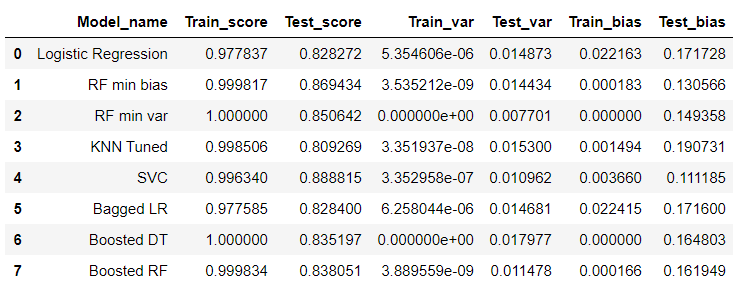
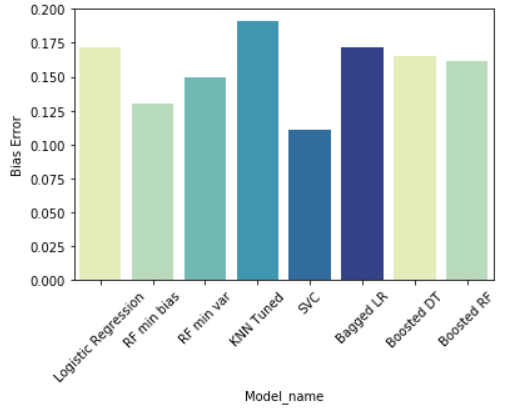


Figure 18:Models with 80+ F1-Score

 Figure 18:Models Comparison with 80+ F1-Score

After trying the One-User-Out-Metric for modeling we see that our is model overfitting a lot. We need to take care of this factor.

So we can try another approach where we further generalise the train test split.Hence we came with the similar metric to One-User-Out train test split called Three-User-Out metric where we keep 3 users to evaluate our model.

Results for Three-User-Out-Metric

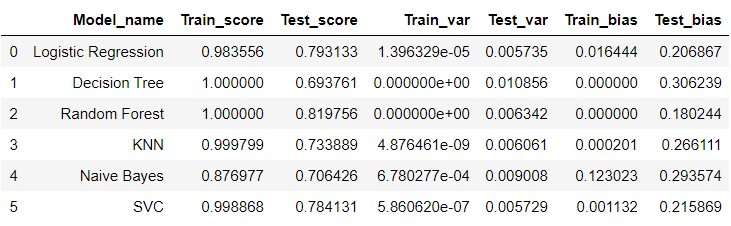


Figure 19:Base Model Results for Three-User-Out

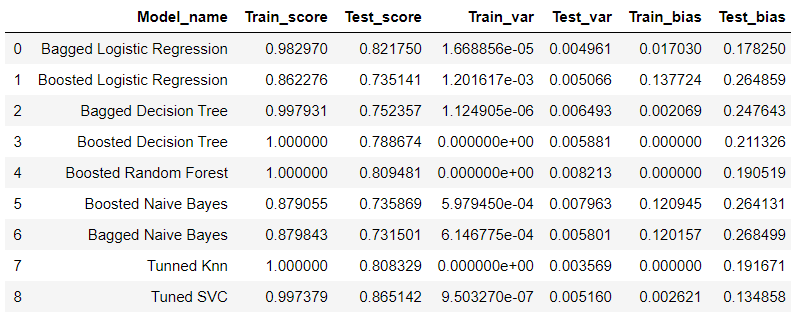
Further expanding on three-user-out by applying bagging and boosting on the models to see if we can reduce overfitting.

Figure 20: Model Results for Three-User-Out

As we can see from the above results that our models are overfitting. We tried different ways to reduce the overfitting but the models tends to overfit and we do not get satisfactory results.Moving forward we will take the best three models that works somewhat better on both the metrics of one-user-out and three-user-out and make a voting-classifier model out of it.

The three based performing algorithms for our data are:

* Logistic Regression
* Tuned Random Forest
* Tuned SVC

Before jumping on creating a voting classifier for the above 3 models we will try training the model using the stratified train test split to see if we get any further good performing models so that we can include in our voting classifier.

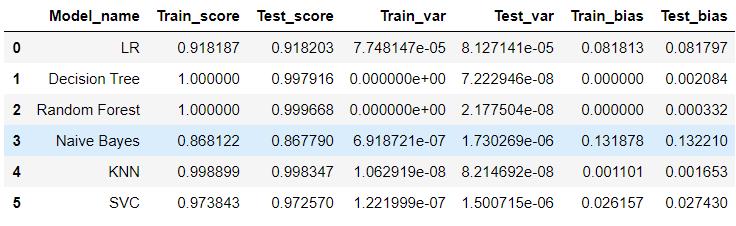
Results for Stratified train-test split

Figure 21: Model Results for Stratified TrainTest Split

We can infer here that our model performs well when we use the stratified train-test split. But there is a problem, that being our model is very generalised i.e. the samples that are in training can also be present in the test set that can result our models to perfom well.

Hence we move forwad with creating a voting classifier with the models that were mentioned abov.

We even consider making the confusion matrix for One-User-Out-Metric because it was in our scope.

Confusion Matrix for Logistic Regression(One-User-Out)

Confusion Matrix

[[13885 48 140 23 25]

[ 76 12656 25 459 1539]

[ 790 126 13856 352 64]

[ 2225 974 1300 9758 216]

[ 687 2003 57 704 12235]]

**Confusion Matrix for Tunned Random Forest (One-User-Out)**

Confusion Matrix

[[12539 441 640 29 472]

[ 9 13881 29 54 782]

[ 315 144 13816 237 676]

[ 30 642 1309 11462 1030]

[ 50 1924 36 21 13655]]

**Confusion Matrix for Tunned SVC (One-User-Out)**

Confusion Matrix

[[13838 49 153 23 58]

[ 57 13260 76 584 778]

[ 312 0 13638 1030 208]

[ 133 5 2626 11434 275]

[ 2 883 591 447 13763]]

Now let’s look at the results for voting-classifiers for both one-user-out metric and three-user-out-metric.

Voting Classifier One-User-Out-Metric Results

Confusion Matrix

[[13849 48 152 23 49]

[ 70 13731 49 70 835]

[ 384 140 14438 121 105]

[ 1374 978 1236 10671 214]

[ 518 829 60 352 13927]]

Train Score 0.997597754162526 Train var 2.314563332967544e-07

Test Score 0.8905947257937088 Test var 0.00723563553193754

Voting Classifier Three-User-Out-Metric Results

Train Score 0.997879994033291 Train var 7.611079867838138e-07

Test Score 0.8554069798417429 Test var 0.0033582142652505676

From the above results we can infer that our model continues to overfit and we have to live with overfit results in this model because of the following reasons:-

* With this type of data and feature engineering we would have obtained excellent results if the data had been more ordered and labeled.
* Statistically, it would have been better if we had more that 30 or at least 30 users instead to get a generalized result.
* Also since it is an image recognition problem, information granules would have helped but here we don’t have the luxury.

Our best model that we will fix for deployment would be the Voting Classifier One-User-Out-Metric because one-user-out-metric always gave us more promising results.

**IX Evaluation Methodology:-**

* **Stratified Train-Test Split:-** This evaluation methodology is better to use in closed environment where we’re finding hand pose recognition for limited group of people.
* **One-User-Out and Three-User-Out:-** This evaluation methodology is better to use where we try to create a more generalized model where there is scope to find pose estimation for new users.

**X Future Scope**:-

To improve current model results:

* Having consistency in the markers data is important so that we’d have more specific data at least at markers level. Currently, we’re used mode generic features of mean, variance, co-variance etc.
* For 3-users out evaluation, we’re currently having 9 users for training and 3 users for testing. A general large sample statistically requires at least 30. Dataset with at least 30 users may help in getting better results.
* If we have more information at pixel-level, instead of only 11-markers for pose estimation, along with that if we have sufficient number of images for training object-detection models (CNN models), we’ll have better scope for identifying the pose even more accurately.

X Relevant Papers:-

A. Gardner, J. Kanno, C. A. Duncan, and R. Selmic. 'Measuring distance between unordered sets of different sizes,' in 2014 IEEE Conference on Computer Vision and Pattern Recognition(CVPR), June 2014, pp. 137-143.

A. Gardner, C. A. Duncan, J. Kanno, and R. Selmic. '3D hand posture recognition from small unlabeled point sets,' in 2014 IEEE International Conference on Systems, Man and Cybernetics (SMC), Oct 2014, pp. 164-169.