**INTERIM REPORT**



Made by :-

Moses Bothello

Mohan Kohad

Naseema Memon

Navika Sharma

Sahil Kedar

**Motion Capture Hand Dataset Information and Facts:**

1. A Vicon motion capture camera system was used to record **12 users performing 5 hand postures** with markers attached to a left-handed glove.
2. **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.
3. The 11 markers not part of the rigid pattern were unlabelled; 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 artefacts 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.
4. The data presented here is already partially pre-processed. First, all markers were transformed to the local coordinate system of the record containing them.

**Finally, any record that contained fewer than 3 markers was removed. The processed data has at most 12 markers per record and at least 3**.

**Attribute Information:**

1) Data is provided as a CSV file.

2) A header provides the name of each attribute.

3) A question mark '?' is used to indicate a missing value. A record corresponds to a single instant or frame as recorded by the camera system.   
  
4) 'Class' - 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. 'User' - Integer. The ID of the user that contributed the record. No meaning other than as an identifier.
2. 'Xi' - Real. The x-coordinate of the i-th unlabelled marker position. 'i' ranges from 0 to 11.   
   'Yi' - Real. The y-coordinate of the i-th unlabelled marker position. 'i' ranges from 0 to 11.   
   'Zi' - Real. The z-coordinate of the i-th unlabelled marker position. 'i' ranges from 0 to 11.
3. Each record is a set. 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

**Goal / Expected Outcomes of this Project:**

We are building a predictive classification model by using a set of recorded data for training to predict hand gestures with marker coordinates.

The data from sensors is used to make machine learn these postures (the patterns and variations in the data) and predict the class id which is actually a hand posture in reality.

These models can be fed into various hardware. Such projects and models can be critical for human use in daily life. Augmenting these models with artificial intelligence, various physical machines/devices can be built and our project work can form base for those physical machines/devices.

**Applications of hand gesture:**

Building robots for farming

• Developing aids for the hearing impaired

• Enabling very young children to interact with computer

• Designing techniques for forensic identification

• Recognizing sign language

• Medically monitoring patients

• Navigating or manipulating virtual environments

• Communicating in video conferencing

• Distance learning / tele-teaching assistance

• Graphic editor control

**Things to be noted before applying a modelling technique:**

Due to the manner in which data was captured, it is likely that for a given record and user there exists a near duplicate record originating from the same user. We therefore need to evaluate classification algorithms on a leave-one-user-out basis wherein each user is iteratively left out from training and used as a test set. One then tests the generalization of the algorithm to new users. A 'User' attribute is provided to accommodate this strategy.

Our initial analysis of the dataset:

As rightly mentioned above

1. there were lot of missing values in the dataset in the form ‘?’ mark.
2. On further probation, we note that these data points are not just numerical figures but coordinates of different images captured by camera.
3. So, this makes them significantly different than general cardinal data.
4. When an image is captured, it is transformed in the form of pixels and then different coordinates are calculated of the image captured from the markers.

**Data cleaning and Exploratory Data Analysis:**

1. After understanding the input data, we see that there is no need for processing of data.
2. For our analysis, we will change the data types of the columns from object to float values, since that is the most appropriate for such unique data.
3. But since there is presence of non-numerical data within the column in the form of missing data or irrelevant data, we proceed towards missing data cleaning, manipulation and EDA.
4. We drop those columns which have more than 50% of missing values. Since it is not very useful to impute those columns, which have, for ex: 96% of the column data values as missing values.
5. We replace ‘?’ with null values to reduce unwanted noise.
6. Also ‘?’ and nan values have to be dealt with to apply any Machine Learning Technique.
7. We then plot correlation heat maps, bar charts, and dendrograms to study and inspect the relationship between the missing data, observed data and the dependent/output variable (Class).
8. We then apply single imputation on the missing values by imputing them with the mean.

**Results of EDA and first application of ML Technique on Base Model:**

1. We are successfully able to apply ML technique as we have removed the irrelevant string data, nan values from the columns and are doing our modelling on the correct float data.
2. We apply Random Forest Classification Technique to gauge the performance of our first model.
3. We get the results as following:

Train Score: 0.9992865766655691

Test Score: 0.9526655000213411

Accuracy Score 0.9526655000213411

**Inference form the first model: *From the train and test scores, we see that there’s a case of overfitting since the test score is less than the training score. So, we have to further improvise our model. For this, we will again revise our EDA technique and missing value manipulation.***

Challenges faced during the analysis:

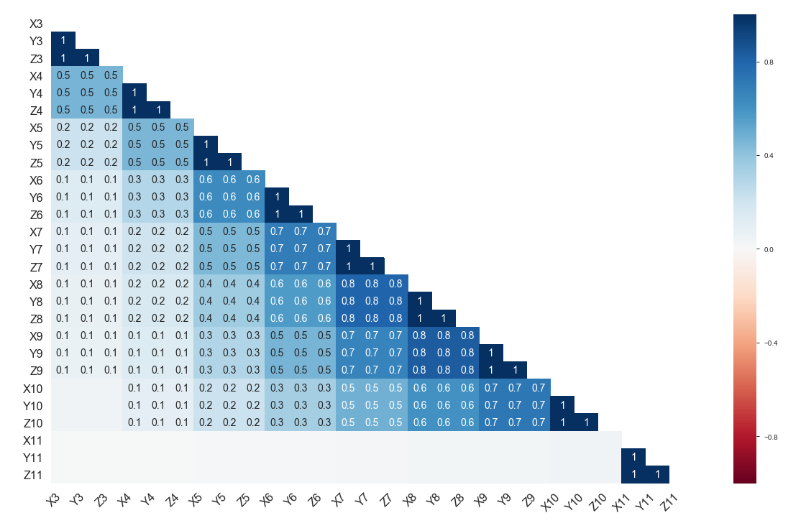
After the initial analysis and understanding the input variables, we realize that:

* As per our understanding of the data, a simple imputation for missing values wouldn’t be appropriate or make sense for this data.
* Our logic is, because the input variables are coordinates returned by electronic sensors, estimation of missing data is not possible and cannot be confirmed of its mathematical value. Even if we impute through traditional method, we would only introduce more bias.
* So, we decided to check correlation within the missing data and try to extract patterns or relations.
* We used library called missing no and built a heatmap and graphs to find correlation in between missing values.
* After that columns that have more than 50 percent missing values are dropped from consideration.

***After inferring the results from correlation lots and graphs between the missing data and the class id column, we note that there is a pattern between the number of missing values per column plus the number of columns having missing values, and the variation in the class id. The remaining columns with considerable missing value percentage but less than 50%, we decide to use the imputation method. We will consider other multiple imputation techniques such as MICE (Multiple Imputation by Chained Equations), KNN Imputation, Regression Imputation or Stochastic Regression Technique. We will constantly tweak our EDA and our Imputation methods accordingly, to get the best model and to see that the machine learns and follows the pattern of the missing and the non-missing data.***

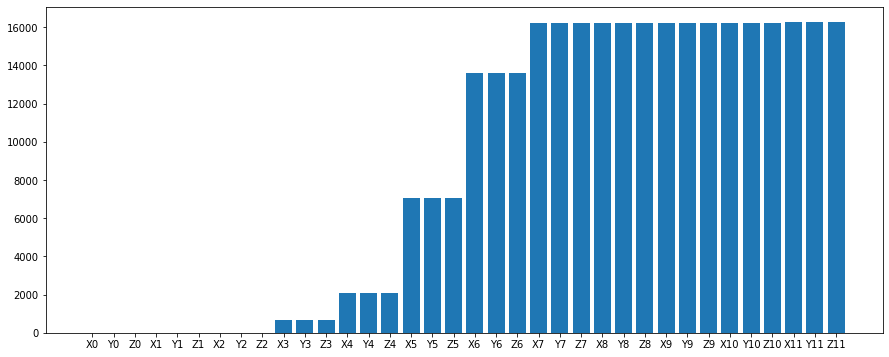
Graphical Patterns and Graphs to support our theory

1. **Correlation Heatmap including all the coordinates.**

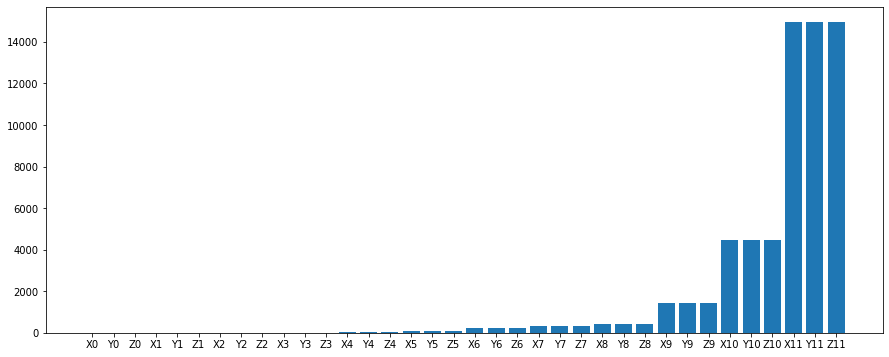


1. **Bar charts depicting the number of missing values across all columns per class id.**

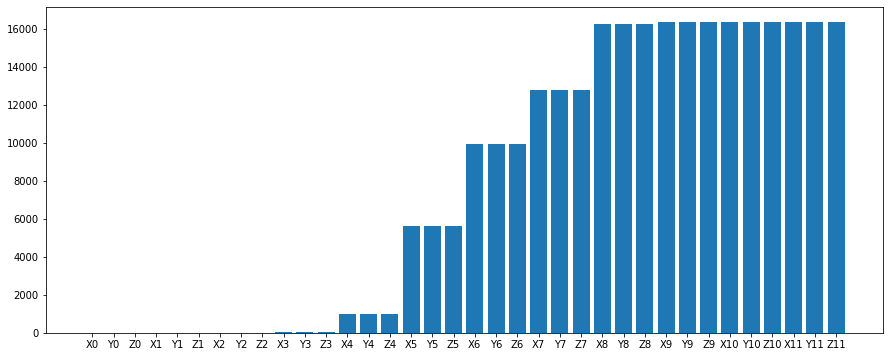
**Bar chart for missing values of class 1 or FIST posture**



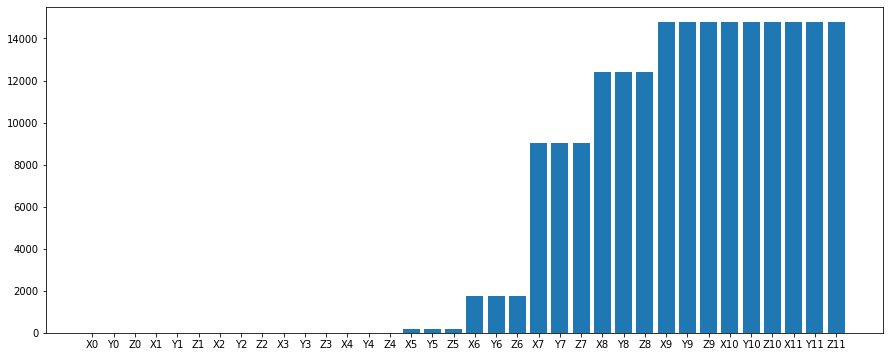
Bar chart for missing values of class 2 or STOP posture



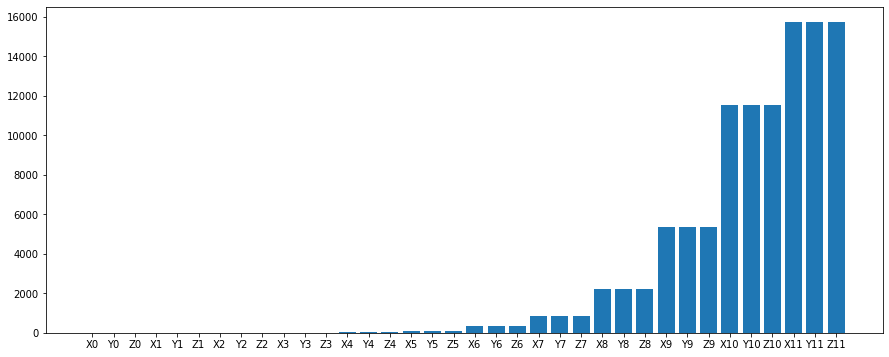
Bar chart for missing values of class 3 or POINT 1 posture



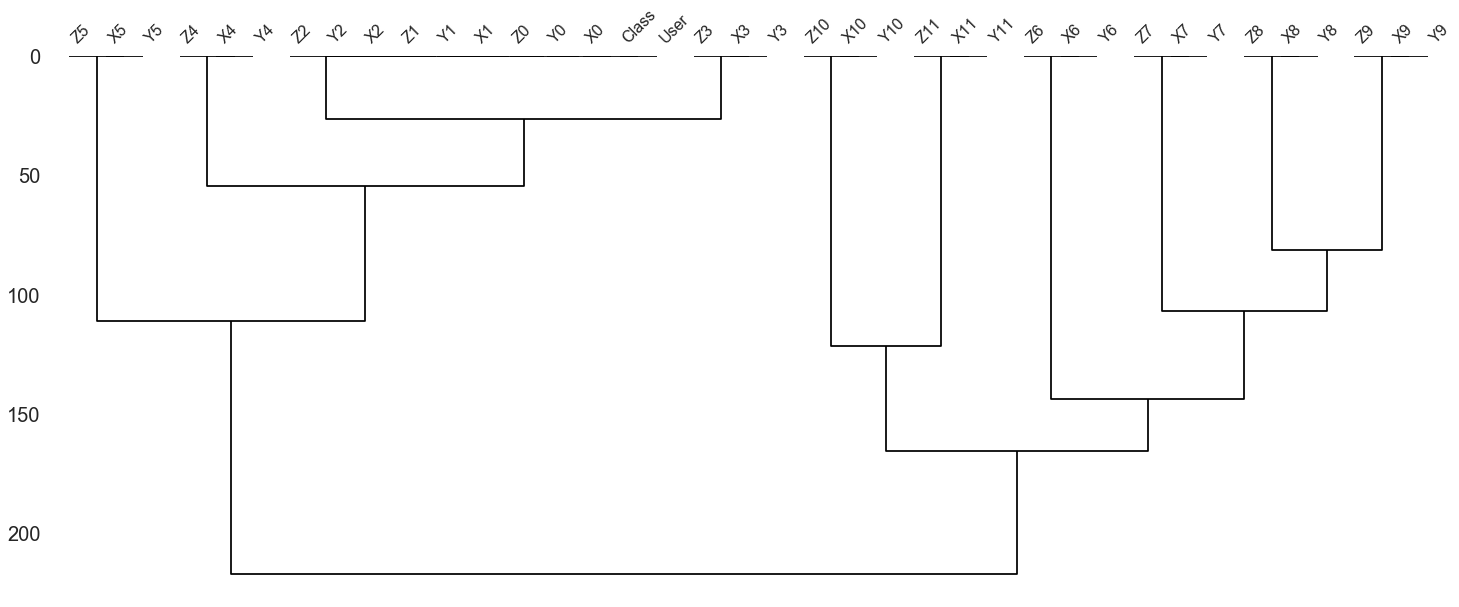
Bar chart for missing values of class 4 or POINT 2 posture



Bar chart for missing values of class 5 or GRAB posture



Dendrogram showing clusters of related data features



**Next steps to be** taken**:**

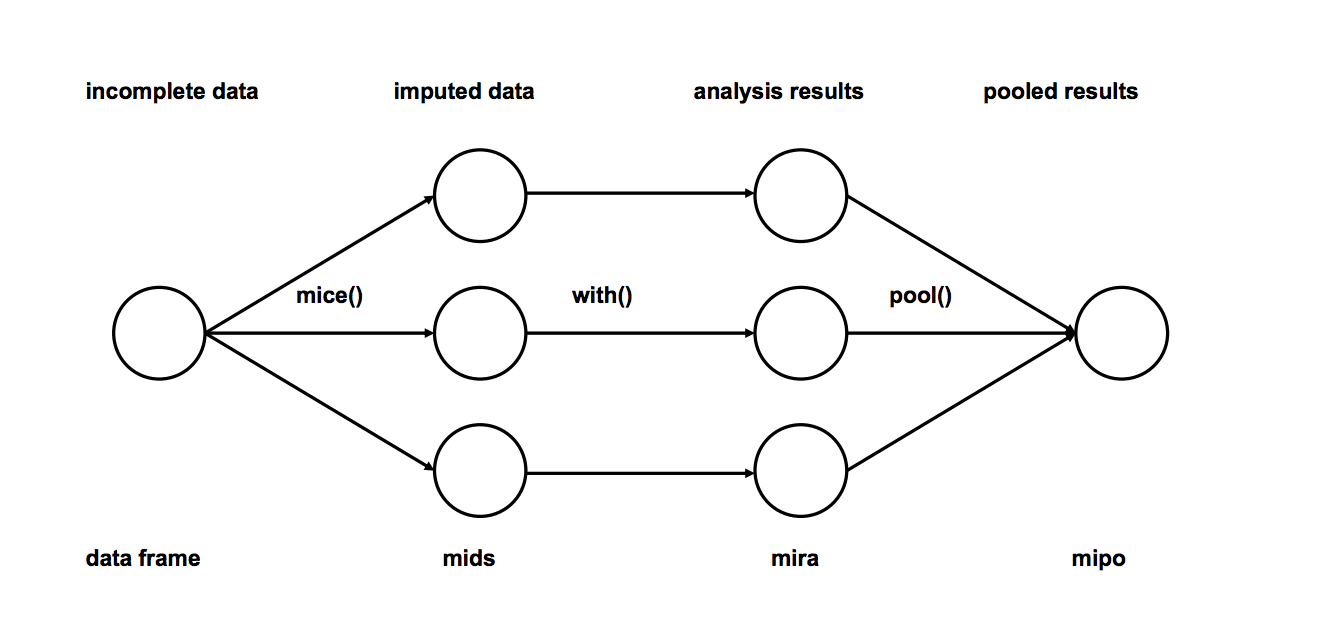
**Reasons for choosing MICE Imputation for our Missing Data Imputation:**

1. During treatment of missing data values, one always should look at the missing data mechanism; whether the data is Missing Completely at Random (MCAR), Missing at Random (MAR), or Missing Not at Random (MNAR).
2. According to our observation, the missing data mechanism follows Missing at Random.
3. Missing at Random, MAR, means there is a systematic relationship between the propensity of missing values and the observed data, but not the missing data.
4. Whether an observation is missing has nothing to do with the missing values, but it does have to do with the values of an individual’s observed variables. So, for example, if men are more likely to tell you their weight than women, weight is MAR.
5. The same way, when we tracked the missing values according to the class id, we saw that the number of columns having or not having any missing value varied accordingly.
6. Multiple Imputation methods assume that the data is missing at random.
7. A chi-square test to see whether the number of missing values depends on the class id or the posture can also further the proof of the propensity or behaviour of missing values.
8. **The limitation with single imputation is that because these techniques find maximally likely values, they do not generate entries which accurately reflect the distribution of the underlying data.**
9. We explore further and come across a multiple imputation technique which tries to follow the distribution of the underlying data. MICE is a multiple imputation method used to replace missing data values in a data set **under certain assumptions about the data missingness mechanism.**
10. MICE Imputation approaches use other variables in the data set to predict the missing value, and contain a random component. Using other variables preserves the relationships among variables in the imputations.
11. **The random component is important so that all missing values of a single variable are not all exactly equal.**
12. **In principle, MICE is be able to handle large amounts of missing data, as it runs multiple imputations on the same dataset to predict the most accurate estimate.** Variables with lots of missing data points would be expected to end up with larger error terms than those with fewer missing data points, so your ability to detect significant relations to those variables would be limited accordingly. That's an advantage of having multiple imputations and analysing results from all of the imputations.
13. For MICE method, a column with 80% missing value is no problem and we have column with more than 90% of missing data.
14. This technique is very flexible and can handle variables of varying types (e.g., continuous or binary).

**How does MICE Technique do its imputation ?**

The MICE algorithm works: by running multiple regression models and each missing value is modelled conditionally depending on the observed (non-missing) values.

1. **Imputation**: Impute the missing entries of the incomplete data sets m times
2. **Analysis**: Analyse each of the m completed data sets.
3. **Pooling**: Integrate the *m* analysis results into a final result

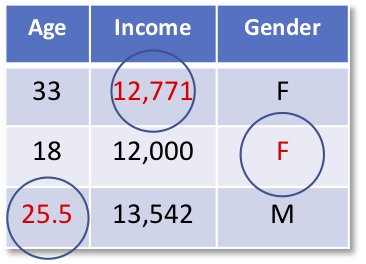


To sum up, MICE imputes missing values in the variables of a data set by using a divide and conquer approach - in other words, by focusing on one variable at a time. Once the focus is placed on one variable, MICE uses all the other variables in the data set to predict missingness in that variable.

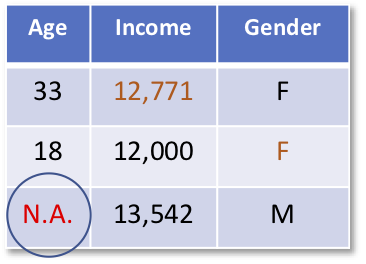
The prediction is based on a regression model, with the form of the model depending on the nature of the focus.

**Methodology of MICE Imputation technique:**

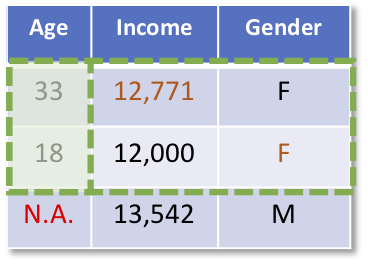
In step 1 of the MICE process, each variable would first be imputed using, e.g., mean imputation, temporarily setting any missing value equal to the mean observed value for that variable.



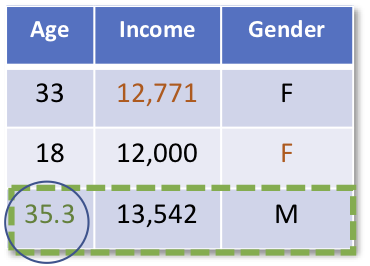
Then in the next step the imputed mean values of age would be set back to missing (N.A).



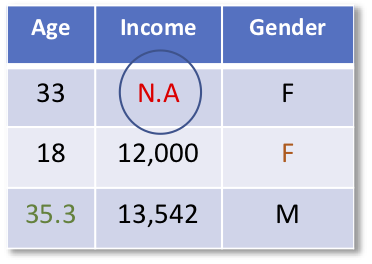
In the next step Bayesian linear regression of age predicted by income and gender would be run using all cases where age was observed.



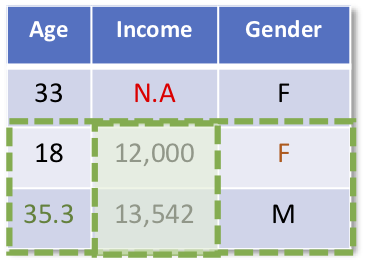
In the next step, prediction of the missing age value would be obtained from that regression equation and imputed. At this point, age does not have any missingness.



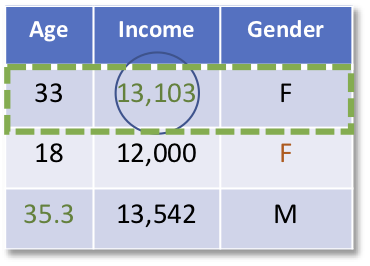
The previous steps would then be repeated for the income variable. The originally missing values of income would be set back to missing (N.A).



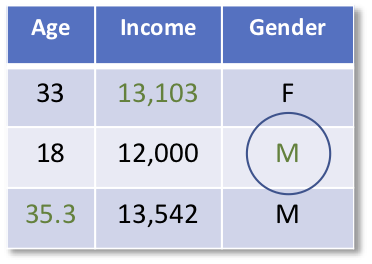
A linear regression of income predicted by age and gender would be run using all cases with income observed.



Imputations (predictions) would be obtained from that regression equation for the missing income value.



Then, the previous steps would again be repeated for the variable gender. The originally missing values of gender would be set back to missing and a logistic regression of gender on age and income would be run using all cases with gender observed. Predictions from that logistic regression model would be used to impute the missing gender values.



This entire process of iterating through the three variables would be repeated until some measure of convergence, where the imputations are stable; the observed data and the final set of imputed values would then constitute one “complete” data set.

We then repeat this whole process multiple times in order to get multiple imputations.

Machine Learning Techniques:

We are planning on again applying ensemble classification techniques like random forest, knn, etc. to see the accuracy score between the train, test data and the predicted values.

A model is trained several times on random sample of the dataset to achieve good prediction performance from the random forest algorithm. In this ensemble learning method, the output of all the decision trees in the random forest, is combined to make the final prediction. The final prediction of the random forest algorithm is derived by polling the results of each decision tree or just by going with a prediction that appears the most times in the decision trees.

**Summary:**

Random Forest are resistant to outliers, and are more robust to noise.

1. So, for the EDA part of our model, we will try to apply MICE technique to fill out missing values and notice the changes in the accuracy and overall fitting of the model.
2. After that we will apply Random Forest Classification Technique and check the accuracy score .
3. After checking the performance of our model, we will probe for overfitting or underfitting.
4. We may have to do hyperparameter tuning and cross validation to further combat overfitting, if there is any.
5. We may use other evaluation metrics such as a ROC curve, confusion matrix.

**Appendix:**

Loading the appropriate python libraries:

import pandas as pd

import numpy as np

import matplotlib.pyplot as plt

import seaborn as sbn

import warnings

warnings.filterwarnings('ignore')

import pandas\_profiling

from sklearn.model\_selection import train\_test\_split

from sklearn.metrics import mean\_absolute\_error, mean\_squared\_error, r2\_score

Reading the data and replacing the ‘?’ missing values with null values.

d=pd.read\_csv("Postures.csv")

d.replace('?',np.nan,inplace=True)

Checking the percentage of missing data and dropping the columns accordingly.

pandas\_profiling.ProfileReport(d)

d.drop(["User",'X7','Y7','Z7','X8','Y8','Z8','X9','Y9','Z9','X10','Y10','Z10','X11','Y11','Z11'],axis=1,inplace=True)

Applying single Imputation to columns:

dd=d.fillna(d.mean())

Preparing our dependent and independent and splitting them for training and test dataset:

x=df.drop('Class',axis=1)

y=df.Class

x\_train,x\_test,y\_train,y\_test=train\_test\_split(x,y,test\_size=0.30,random\_state=10)

Applying Random Forest classifier Technique on our first base model and checking the output:

rfc=RandomForestClassifier(random\_state=2).fit(x\_train,y\_train)

predict=rfc.predict(x\_test)

print('Train Score:',rfc.score(x\_train,y\_train))

print('Test Score:',rfc.score(x\_test,y\_test))

print('Accuracy Score',accuracy\_score(y\_test,predict))

**Output:**

Train Score: 0.9992865766655691

Test Score: 0.9526655000213411

Accuracy Score 0.9526655000213411