Answer 1 : Exploratory Data Analysis :

Step 1 : Reading data and getting summary of each column.

Summary of the data :

Column name	Min	1st Quartile	Median	Mean	3rd quartile	Max	Missing values
User_id	-	-	-	-	-	-	9000
Age	14.00	58.00	83.00	83.17	108.00	186.00	9002
Start_bmi	7.40	30.00	33.10	33.52	36.30	99.80	10471
Activity_factor	1.200	1.200	1.375	1.374	1.550	1.725	7

Note : In the above table User_id contains "-" for some places where it does not make any sense, since I am not going to include this column for training my model. Also the rest of features are categorical features and they don't fit in this summary and none of them contains any missing values.

After removing all rows where user_id was missing we got following summary.

Column name	Min	1st Quartile	Median	Mean	3rd quartile	Max	Missing values
User_id	-	-	-	-	-	-	0
Age	14.00	58.00	83.00	83.17	108.00	186.00	2
Start_bmi	7.40	30.00	33.10	33.52	36.30	99.80	1471
Activity_factor	1.200	1.200	1.375	1.374	1.550	1.725	7

We still have some missing values for age, start_bmi and activity_factor column. I have handled that in next step.

Step 2 : Data quality issue

1. Missing values
2. Class imbalance problem

Missing values

- Missing values in user_id column :
    a. This column represents the user id which does not contributes in predicting dependent variable. So there is no point of including this column for building our model. So no need to handle missing values of this column. Just simply delete the rows where user_id had missing values.

- Missing values in age column :
    a. Simple approach [Might result in poor performance]
        Simple approach is to replace missing values by just average of the age column.
    b. By building a linear relationship between age and start_bmi. But start_bmi column also contains some missing values. So for replacing missing values of start_bmi we can replace missing values by mean of start_bmi column and the we can build an age as a linear function of start_bmi. Note that now start_bmi does not contain any missing values, since we have already replaced missing values by its mean.

- Missing values in start_bmi column :
    a. Simplest approach :
        Replace by its mean.
    b. By building linear relationship between start_bmi and age, as we discussed in previous point.

- Missing values in activity_factor :
    a. This column contains only 7 missing values. So just replace missing values by its mean. This will not affect much since there are only 7 missing values. If we will build linear relationship for this column also then there will be no point in running long code for replacing just 7 missing value. It will be extra overhead only for 7 values.

Class Imbalance problem

- dependent variable is not evenly distributed in the dataset. In the training dataset out of 63586 observations only 646 observations has class 1 and rest has class 0. So there Is clearly class imbalance problem.

Train dataset summary of dependent variable :

class	0	1
Number of observations	63586	646

Probability :

class	0	1
Number of observations	0.98952811 (98.95%)	0.01047189 (1.04%)

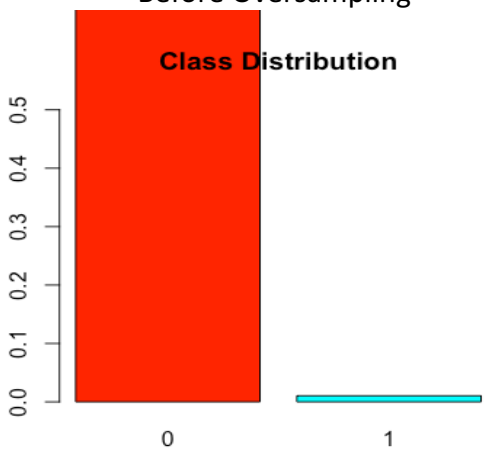
- Solution to the class imbalance problem :
    - Oversampling and Undersampling

Oversampling :
Increasing the number of observations of class having less number of observations. So here observations of class 1 is randomly added to maintain the balance of class distribution equally. So if class 0 contains 63586 observations then in total 63586*2 observations would be there.

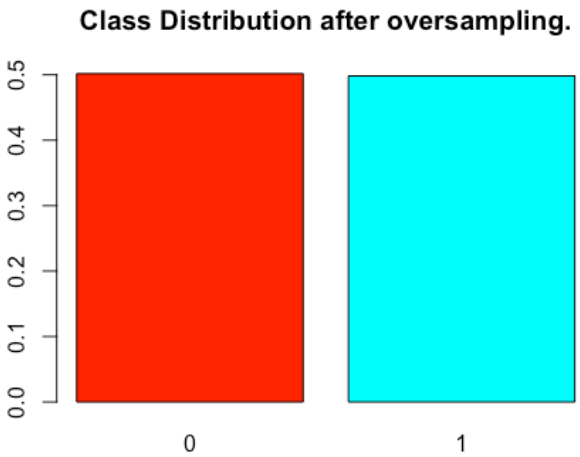
Undersampling :
In undersampling we follow the reverse procedure of oversampling. We reduce the number of observations to the class having more number of observations. So here class 0 has more number of observations so we eliminate some of the observations from class 0 to make equal observations from both class.

Here I have used oversampling and below are some charts which explains the oversampling results.

Before Oversampling



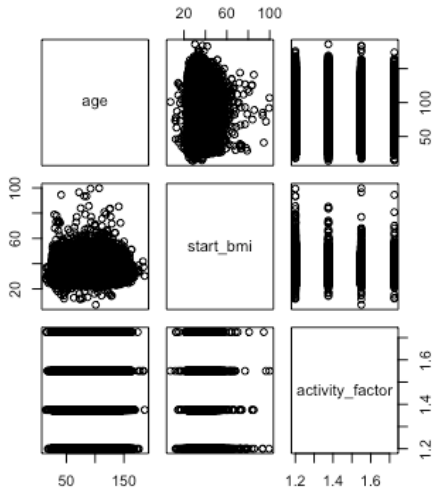
After oversampling



Step 3 : Data representation

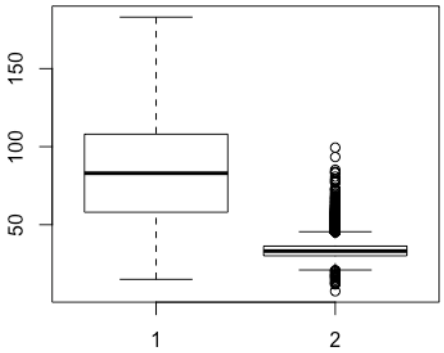
- Correlation between numerical features :

	Age	Start_bmi	Activity_factor
Age	1.000000	0.0289097686	0.0002877679
Start_bmi	0.0289097686	1.000000	-0.0346975235
Activity_factor	0.0002877679	-0.03469752	1.000000

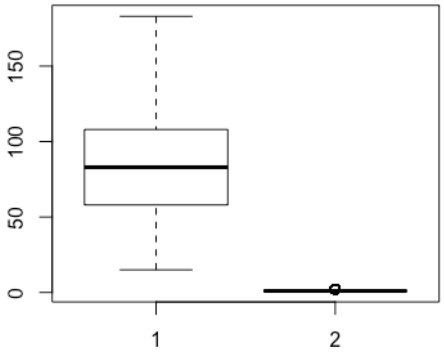


Correlation between numerical features.

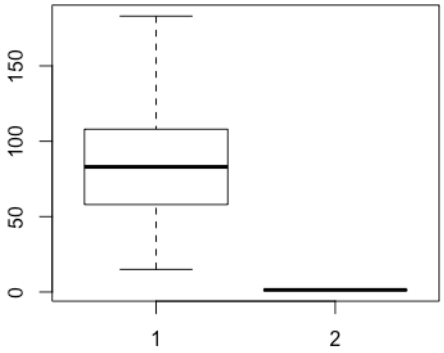
- Box plots of numerical variables with dependent variable :



Box plot of start_bmi with dependent variable.



Box plot of age with dependent variable.



Box plot of activity_factor with dependent variable

Answer 2 : Model building

Step 4 : Training a model :

- This is a classification problem and I have used svm classifier to train a model. So for this step I have divided the data in to two parts, training and testing. Will train our model on training data and test on testing data.
- I have excluded index, user_id and connected_pedometer column, because these columns does not contribute to predict dependent variable.

Building model by following command :

```
model <-
svm(paid~gender+age+start_bmi+activity_factor+OS+hypothyroid+diabetes+pcos+physical+hypertension+high_blood_pressure+cholesterol+medical_conditions+devicebrand, data = over)
```

where I am using oversampled data.

Here is a model summary :

Parameters	:	SVM-Type: C-classification
SVM-Kernel	:	radial
Cost	:	1
Gamma	:	0.005988024
Number of Support Vectors	:	119732 (59855 59877)
Number of Classes	:	2

Confusion matrix :

	0	1
0	18102	223
1	9215	93

Accuracy : 0.6584519
Misclassification error : 0.3415481

Step 5 : More improvements in results

There are several techniques by which we can improve accuracy. Some of them are below :

- Replacing missing values more sophistically. Like we can replace missing values by building some predictive model which will predict the value for missing value.
- Ensemble methods to get goodness of each classification algorithms.
- By doing feature analysis and by adding relevant derived features from the available features.