CMP 2023 Data Mining Project Report

Dan Kha PHAM
xnf21ktu@uea.ac.uk

March 2022

1 Introduction

To start accessing data about diabetes, I applied the path in road map learned in the module. The goal of the project is to predict the correct patient is positive with diabetes mellitus. The data set contains a lot of medical indexes. Therefore, the first thing to do is to understand the data. Also, I looking for similar researches that has been done to see how they are deliberately carried out, to understand the procedure of proceeding the project (Aljumah et al. 2013), (Yang et al. 2021),(Kavakiotis et al. 2017). The first step is to do data exploration, get descriptive statistics of the variables in the data set, and do some initial research on missing values.

2 Data Exploration, Cleaning and Pre-processing

2.1 Loading Data and Describe Dataset

In the first step of Data Exploration, after loading the dataset and conducting descriptive statistics. I confirmed that the dataset has all 79,159 fields and 88 variables. In which only 3 variables are objects including "ethnicity", "gender", "icu-type". The rest are all numerical variables. I proceed to filter the list of variables that have more than 60% of missing value, the list includes 38 numerical variables. I decided to remove all of these variables, because of the lack of too many lines of data. If kept, these variables will cause the model to be skewed and biased. The remaining dataset consists of 79159 x 50, total cells include 3,957,950, total missing 468,325, missing data rate is 11.83% (Figure 1). I tried understand to basic lab indicators in the data set (Ong & Penm 2019).

2.2 Data Cleaning

2.2.1 Numerical Data

Next, I proceed doing further data cleaning. By dropping all the age variable with value equal 0, working on the relation between BMI, height and weight. Then drop all the fields that both height and weight are missing. I also found that BMI are incorrect computed in approximately 3000 fields, hence I recomputed it based on the given height and weight.

2.2.2 Categorical Data

Since the missing percentage in "Gender" is just 0.04%, I decided to drop all the null rows in it. For "Ethnicity" variable, after plotted the distribution. I decided to put all the Null value of "Ethnicity" to "Other/Unknown" category.

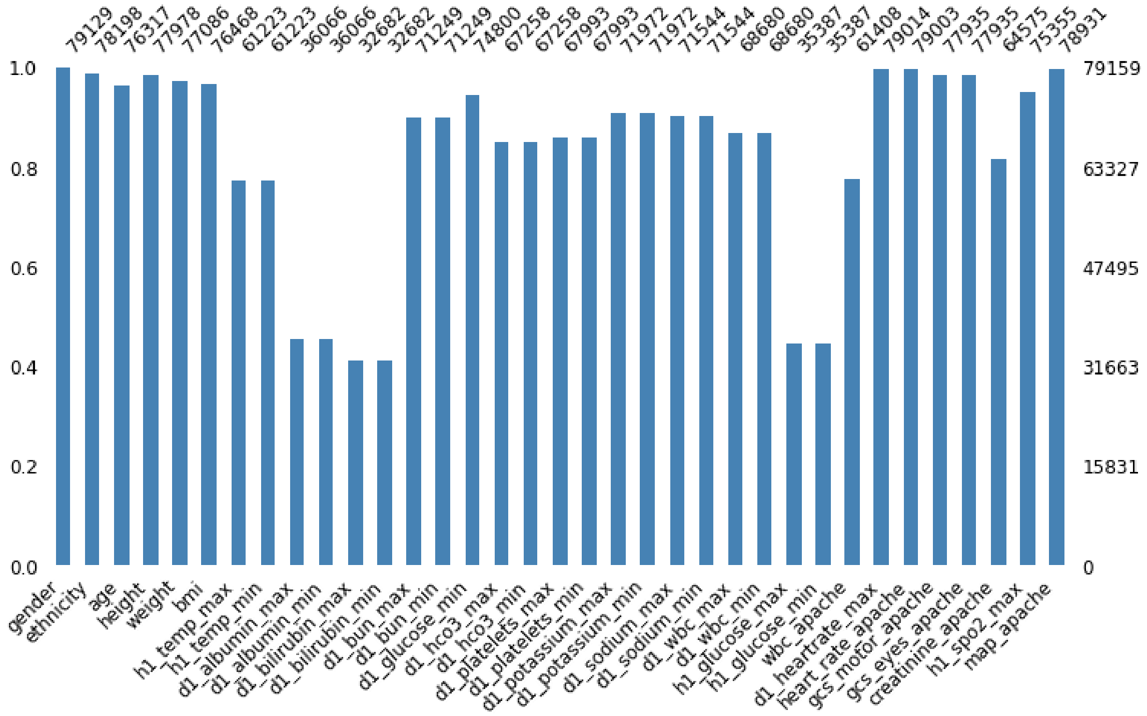


Figure 1: Missing plot

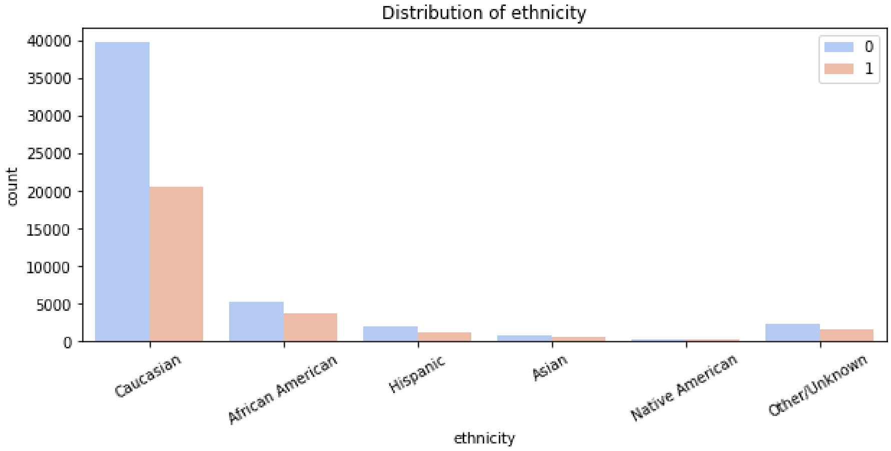


Figure 2: Ethnicity Distribution

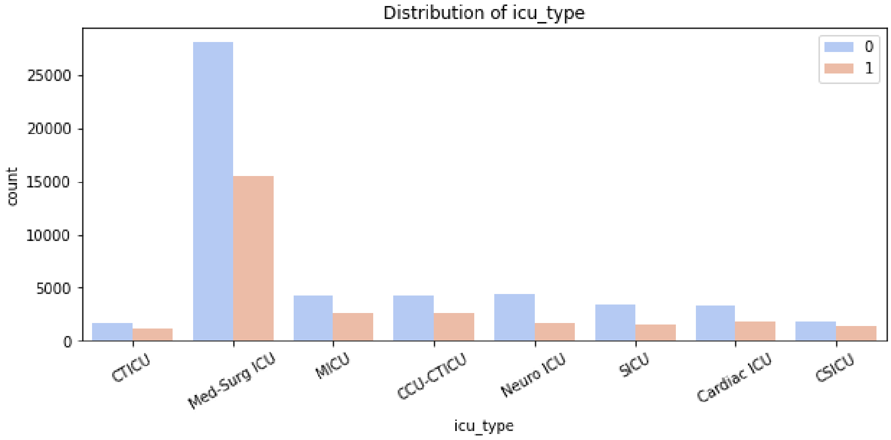


Figure 3: ICU_type Distribution

2.2.3 Handle Correlate Variables and Missing Model

After done dealing with object columns, I proceed finding duplicate columns, there is no Duplicate column in the data frame. I proceed handle the correlate variables, I run correlate statistics for the data set and find out that 10 pairs of labs indicators are high correlated to each others (all of those are between max's and min's), "Readmission_status" seems having no involve to the data-set (Figure 5), and GCS Score can be the combination of both Eye and Motor in our dataset (Enriquez et al. 2019). Therefore, I decided to take the average of the two Max and Min variables, to replace 2 of them for all 10 pairs variables, take the sum of 2 GCS variables to formulate one single GCS Score column and drop the irrelevant variables. After that, realizing there are d1_albumin, d1_bilirubin and h1_glucose which are accounted for more than 50%. Since these missing are important for target variable, and they are not Missing Complete at Random (MCAR) (Figure 4). I'm try making model for the missingness of the 5 highest missing percent variables. Which then added 5 more columns. After all, I concatenated the missingness model to the Data Frame make it finally become 78570 rows and 43 columns.

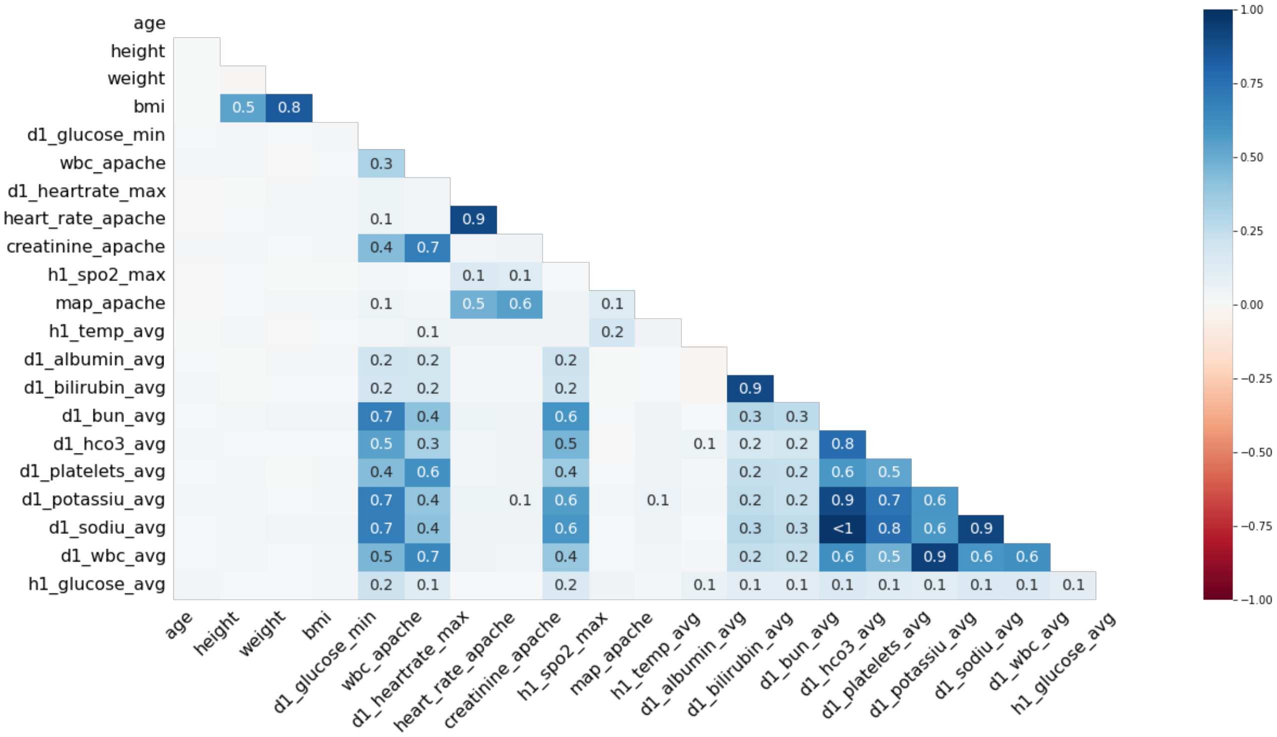


Figure 4: Missingness Correlation

From here I think about whether or not this data is vital to determine our target variable, which is to determine whether or not the patient is having diabetes mellitus. Therefore, I divided the data set into 2 groups, 1 group of patients with diabetes, the other group of patients without diabetes. Although, according to the graphs, Platelets variables does not seems to convey significant information, I found this research article about its importance (Santilli et al. 2015), I decided to keep all the outliers the same they are. Because, the outliers may contain the information the ones who is diabetes. And vice versa, those patients' data become outliers because of their diabetes situation. It is not optimised to drop all the outliers away from this data set. Since we are looking for the abnormal to distinguish the unique features of diabetes mellitus. By that thinking, I plot KDE for all variables to distinguish which variables important than others (Figure 6).

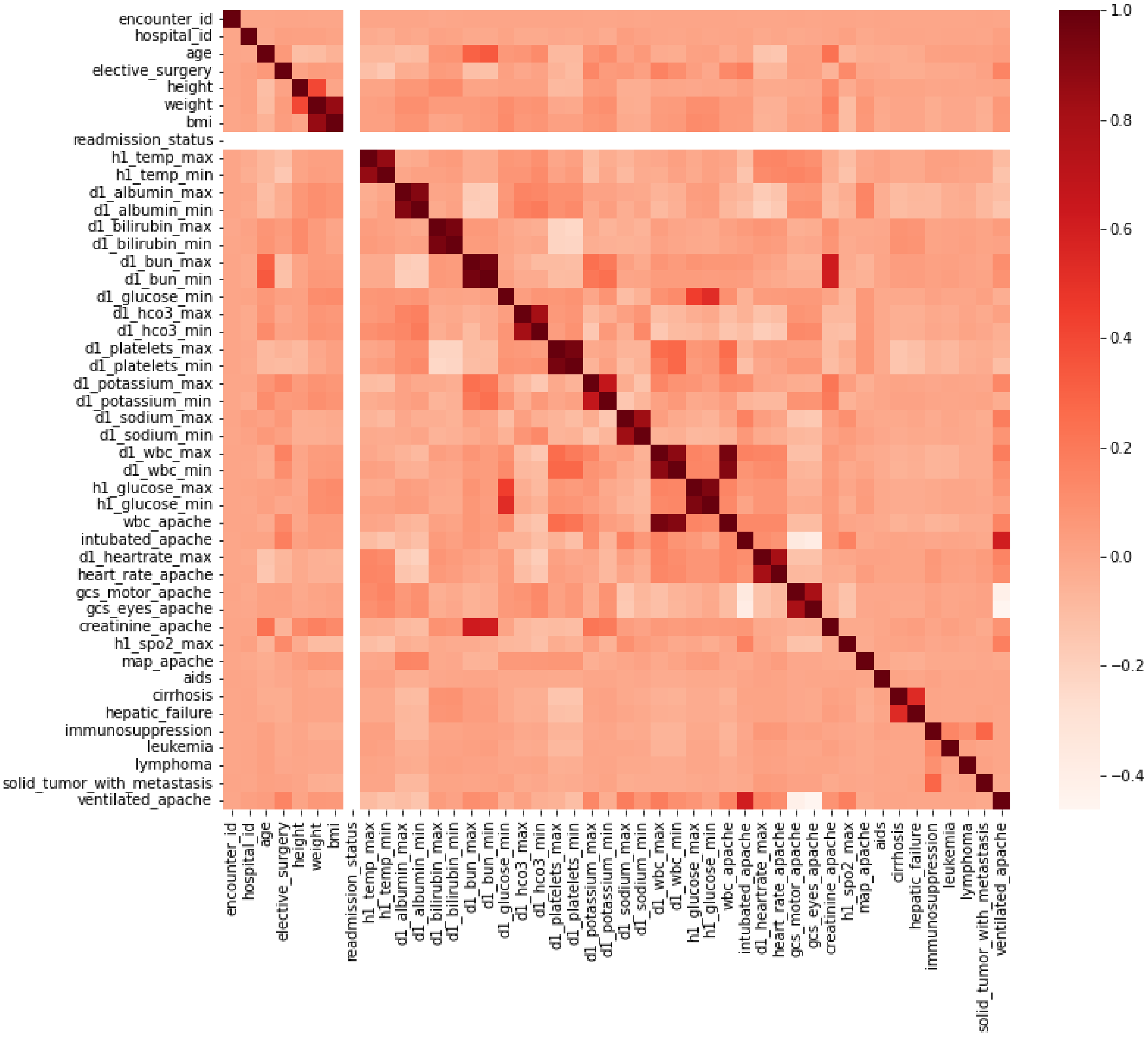


Figure 5: Correlation of all variables

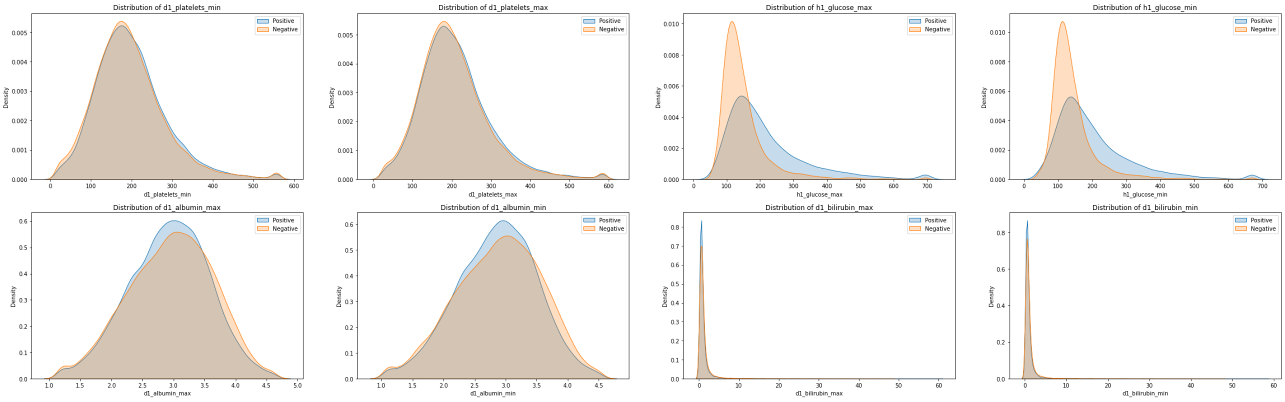


Figure 6: KDE plot 1

2.3 Data Pre-Processing

2.3.1 Encode Categorical Data

Then I proceed on encode binary for ”Gender” and One Hot Encoder for ”Ethnicity” and ”ICU_Type”. I took the Encode columns to replace those Categorical object columns. At this stage, the whole data set contains only numerical data types.

2.3.2 Split the Data set

At this point, the data is well cleaned and ready to handle the missingness either by drop all the NaN or by imputation. However, in order to keep the truthfulness and unbiased of test data. I split the data set into Train set and Test set with 67% and 33% respectively.

2.3.3 Train and Test data set Imputation

At this stage, there are decisions to impute the missing data or to drop all of them. But drop all the N/A would make the data set lost too many information. Therefore, I chose to impute all the missing data at once based on MICE method (Austin et al. 2021) (Jakobsen et al. 2017).

After imputation, the data set is varied from the original (Schweizer et al. 2020). Although provide more distribute to the mean area by imputed missing data, it mostly keep the same distribution as like the original data, (Figure 7). I leave the Test set untouched until using it for Supervised Learning.

2.3.4 Outliers Detection

Then I proceeding handle the outliers for the train set only. First by using Boxplot IQR method to get the total percentage of the outliers, which are 1.83%. I will use this rate to fill in the contamination parameters of Local Outlier Factor (LOF) and Isolation Forest methods.

For the safeness of keeping genuine data, I would remove only those data which seems really unrealistic (Torkey et al. 2021)(Maniruzzaman et al. 2018). The results by using DBScan, LOF and Isolation Forest method is shown in Figure 9 where the green dots are the outliers. DBScan give the result of just 33 outliers, preserve the data set the most out of the 3 methods. Therefore, I would chose to use data set of DBScan to keep proceeding on Feature Selection.

3 Supervised Learning

3.1 Feature Selection

To begin with Classification, I run Feature selection to see the importance of the variables, then I will keep only those which have high distinguish impact only.

For my data set, the results return with high F-Scores give more confidence for the distinguish factors (Figure 10). I then decided to choose up to 20 features from both F_Regression and Mutual Info Classif Scores to assure do not miss any significant variables. Then I proceed to the correlation between those selected features, and decided to remove the features that has correlation higher than 0.60, except for feature ”d1_bun_avg” and ”creatinine_apache” since those features is importance for classify the target variable (Mukaka 2012). The final result appears the data set is now ready for Low level tasks. I repeat the same process for the Test Set, to make sure both Data set have the same variables structure. (Figure 11).

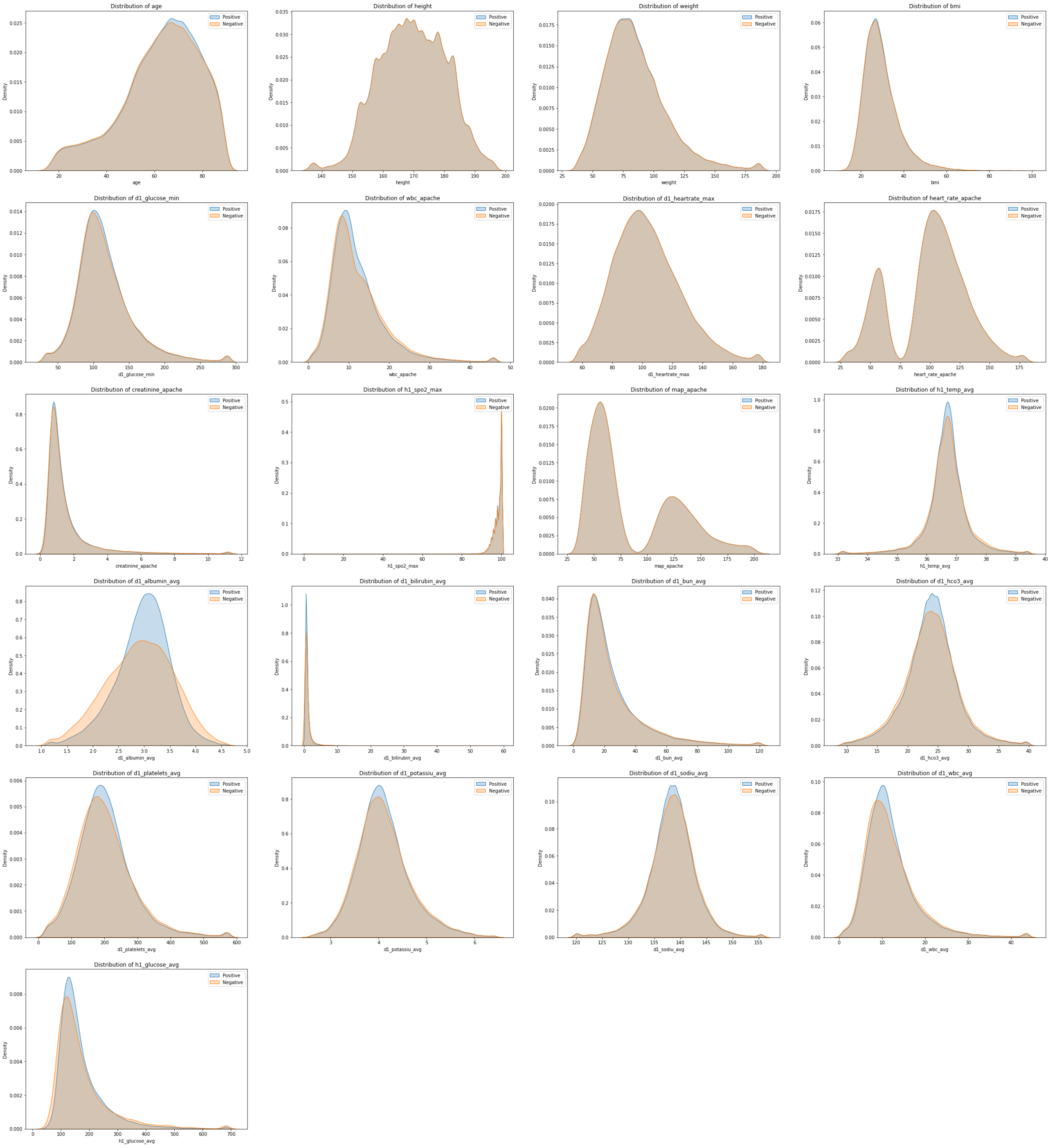

Distribution of age
Distribution of height
Distribution of weight
Distribution of bmi
Distribution of d1_glucose_min
Distribution of wbc_apache
Distribution of d1_heartrate_max
Distribution of heart_rate_apache
Distribution of creatinine_apache
Distribution of h1_spo2_max
Distribution of map_apache
Distribution of h1_temp_avg
Distribution of d1_albumin_avg
Distribution of d1_bilirubin_avg
Distribution of d1_bun_avg
Distribution of d1_hco3_avg
Distribution of d1_platelets_avg
Distribution of d1_potassiu_avg
Distribution of d1_sodiu_avg
Distribution of d1_wbc_avg
Distribution of h1_glucose_avg

Figure 7: Imputed Result with Blue means Imputed data, Orange means Old data

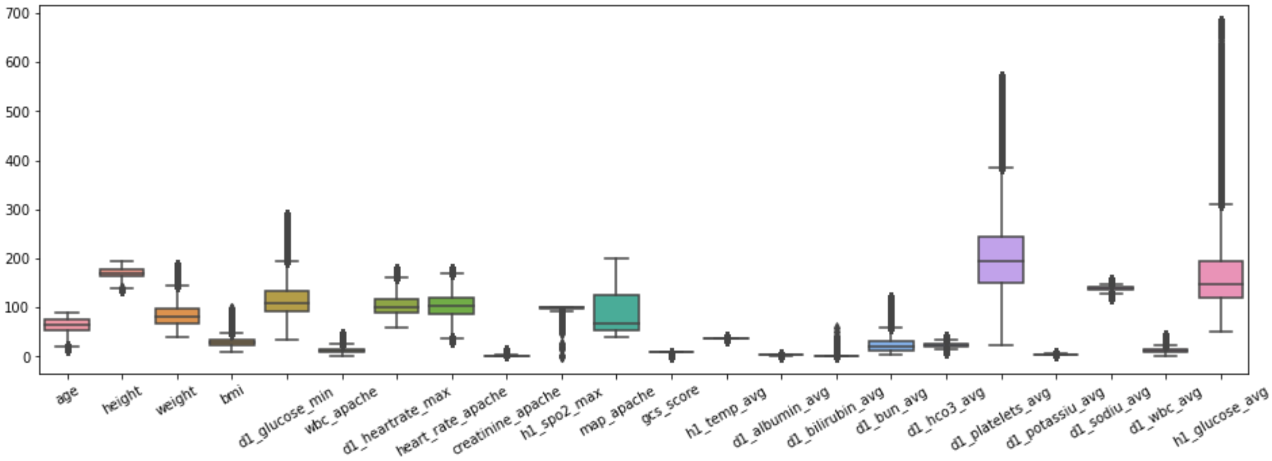


Figure 8: Boxplot outliers data

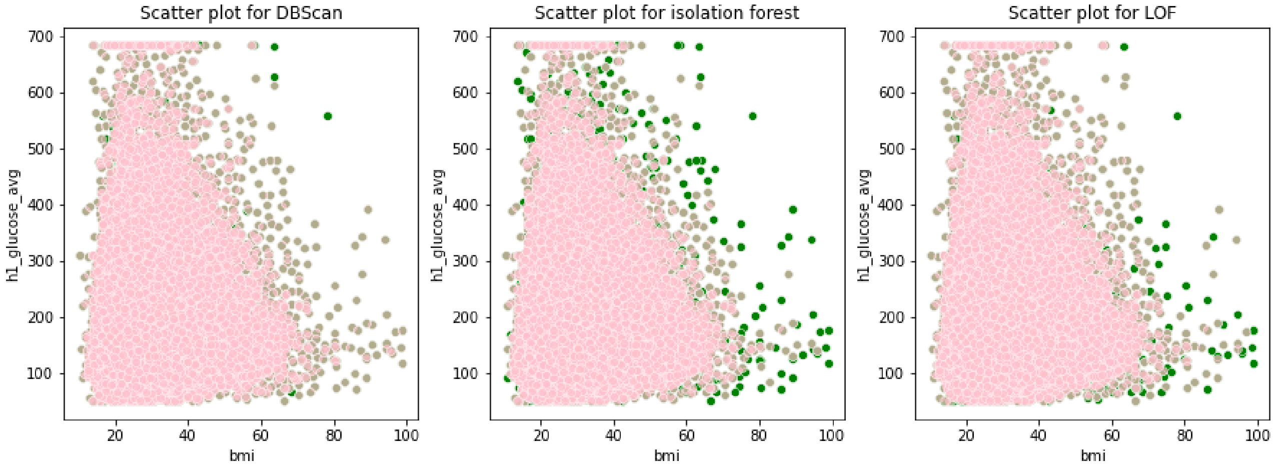
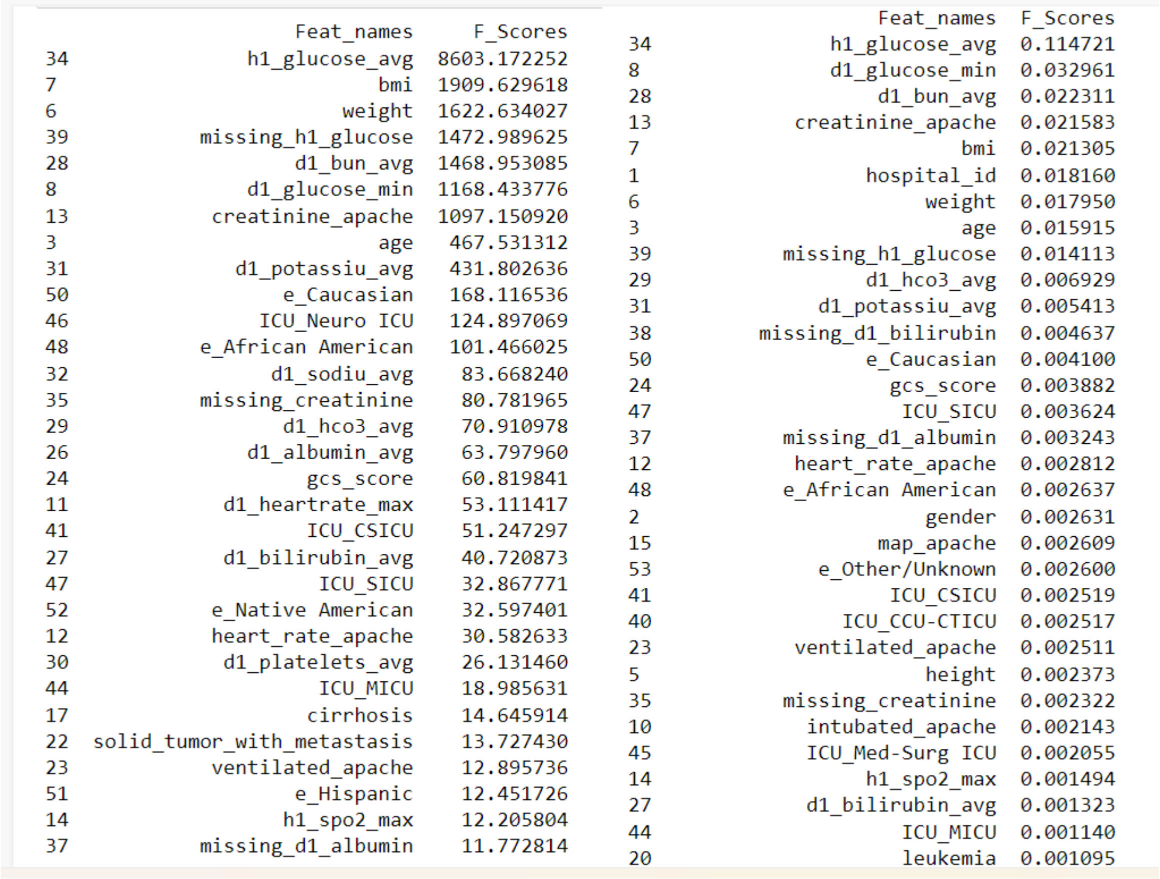


Figure 9: Outliers Detection



```
                   Feat_names      F_Scores
34               h1_glucose_avg  8603.172252
7                           bmi  1909.629618
6                        weight  1622.634027
39           missing_h1_glucose  1472.989625
28                   d1_bun_avg  1468.953085
8                d1_glucose_min  1168.433776
13             creatinine_apache  1097.150920
3                           age   467.531312
31              d1_potassiu_avg   431.802636
50                  e_Caucasian   168.116536
46                 ICU_Neuro ICU   124.897069
48           e_African American   101.466025
32                 d1_sodiu_avg    83.668240
35           missing_creatinine    80.781965
29                  d1_hco3_avg    70.910978
26               d1_albumin_avg    63.797960
24                    gcs_score    60.819841
11             d1_heartrate_max    53.111417
41                     ICU_CSICU    51.247297
27             d1_bilirubin_avg    40.720873
47                      ICU_SICU    32.867771
52            e_Native American    32.597401
12             heart_rate_apache    30.582633
30             d1_platelets_avg    26.131460
44                      ICU_MICU    18.985631
17                     cirrhosis    14.645914
22  solid_tumor_with_metastasis    13.727430
23            ventilated_apache    12.895736
51                   e_Hispanic    12.451726
14                  h1_spo2_max    12.205804
37           missing_d1_albumin    11.772814
```

```
            Feat_names  F_Scores
34        h1_glucose_avg  0.114721
8         d1_glucose_min  0.032961
28            d1_bun_avg  0.022311
13      creatinine_apache  0.021583
7                    bmi  0.021305
1            hospital_id  0.018160
6                 weight  0.017950
3                    age  0.015915
39    missing_h1_glucose  0.014113
29           d1_hco3_avg  0.006929
31       d1_potassiu_avg  0.005413
38  missing_d1_bilirubin  0.004637
50           e_Caucasian  0.004100
24             gcs_score  0.003882
47              ICU_SICU  0.003624
37    missing_d1_albumin  0.003243
12     heart_rate_apache  0.002812
48    e_African American  0.002637
2                 gender  0.002631
15            map_apache  0.002609
53       e_Other/Unknown  0.002600
41             ICU_CSICU  0.002519
40          ICU_CCU-CTICU  0.002517
23     ventilated_apache  0.002511
5                 height  0.002373
35    missing_creatinine  0.002322
10      intubated_apache  0.002143
45     ICU_Med-Surg ICU  0.002055
14           h1_spo2_max  0.001494
27      d1_bilirubin_avg  0.001323
44              ICU_MICU  0.001140
20              leukemia  0.001095
```

Figure 10: Feature Selection Result

3.2 Balancing Train set

After split, the y_train data has 33 thousand of value 0 and 18 thousand of value 1, this make the data imbalance. Downsample would make us lose the information of the non target variable, but Upsample would make it replicate of the target data, making bias in prediction. Therefore I chose to downsample the data.

3.3 Classification

At this stage, I run pipeline with option to normalised the data of all classifiers I can use, which are Logistic Regression, SVC, Multi-layer Perceptron, KNeighbors, Decision Tree, Random Forest and Gradient Boosting. Of all classification methods, I let the algorithm to run with default parameters. I got the results in Figure 12. As of all methods, the Random Forest return the highest test_score (approx 90.9%) with acceptable standard deviation.

Therefore, I decided to choose Random Forest to be the Classification method for this problems. At this stage, I would tune the best parameters for Random Forest classifiers. Using RandomizedSearchCV and GridSearchCV with pipeline of selectKbest, tranforms the data, all together to find out the best procedure for classifying the target result. In transformers method, I added StandardScaler, MinMaxScaler, Normalizer, PowerTransformer and QuantileTransformer. Through using SearchCV algorithm, I have cross-validation the result on the data set. The best combination of parameters for this algorithm is shown in Figure 13.

However, on the Test set, the Precision and Recall is just 57% and 84%, this appears incorrectness in the predict ability. But in the perception of detecting patient with False Positive is better than False Negative, I decided to try other Classifiers that having more accuracy in both precision and recall for positive detected. After many tests and trials, I finally

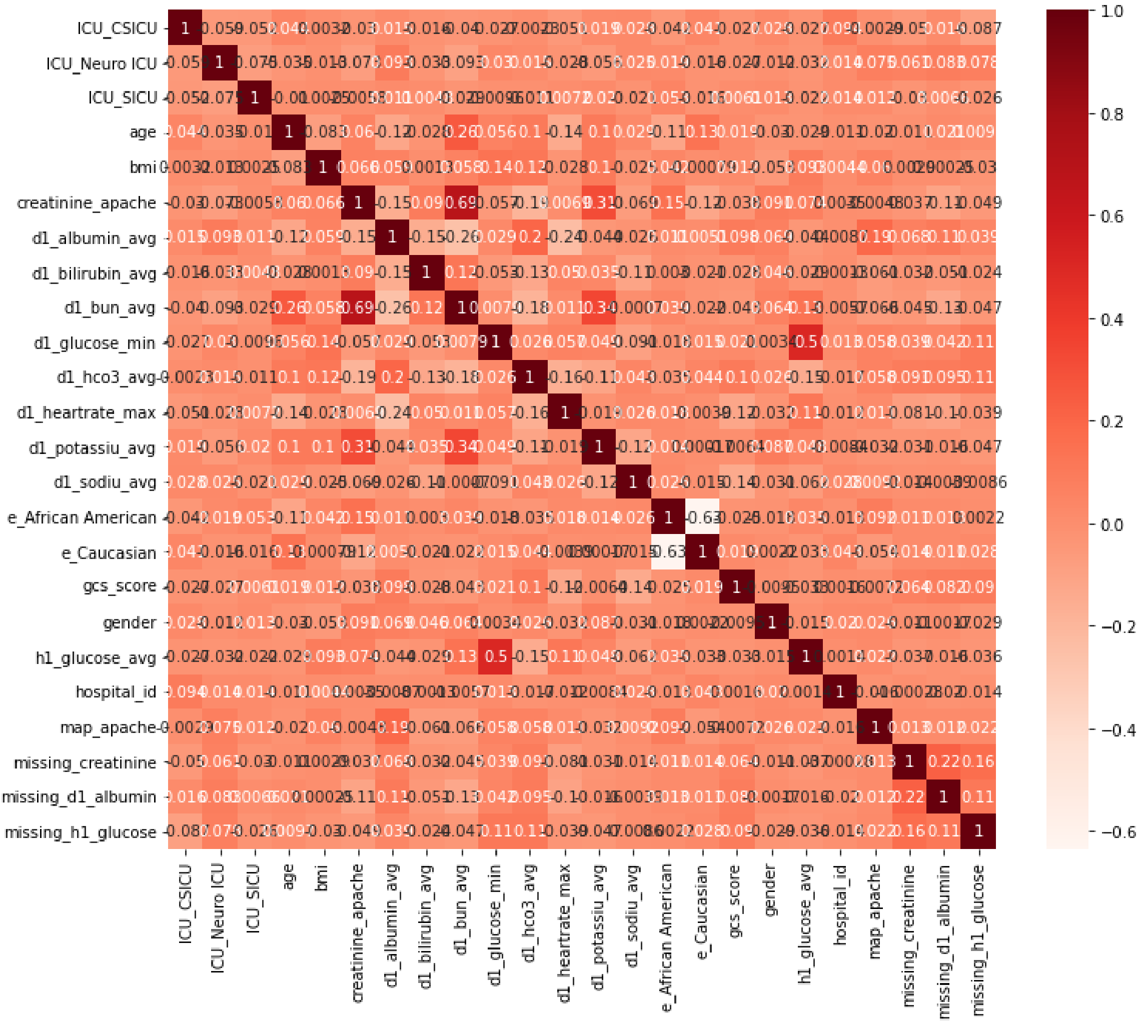
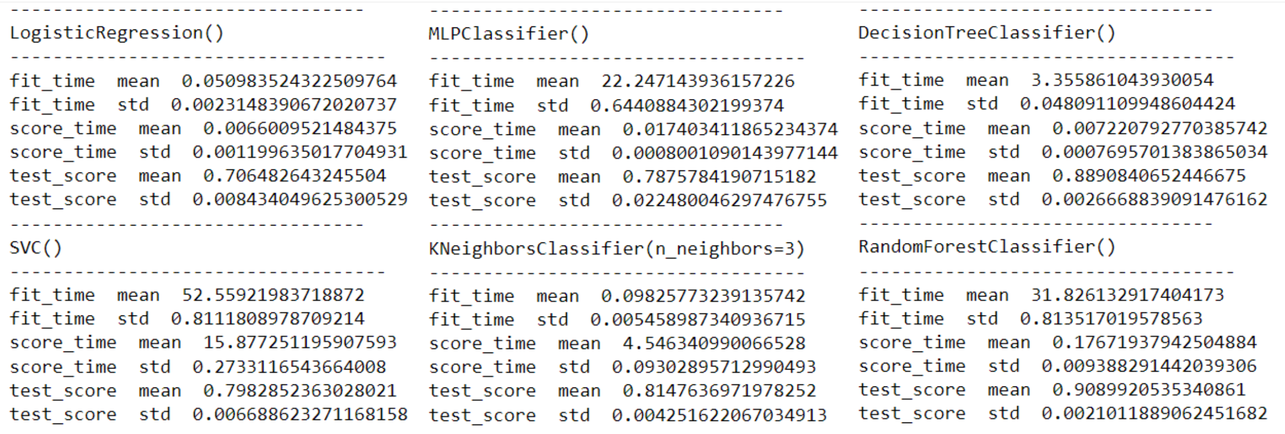


Figure 11: Correlation of Selected Features

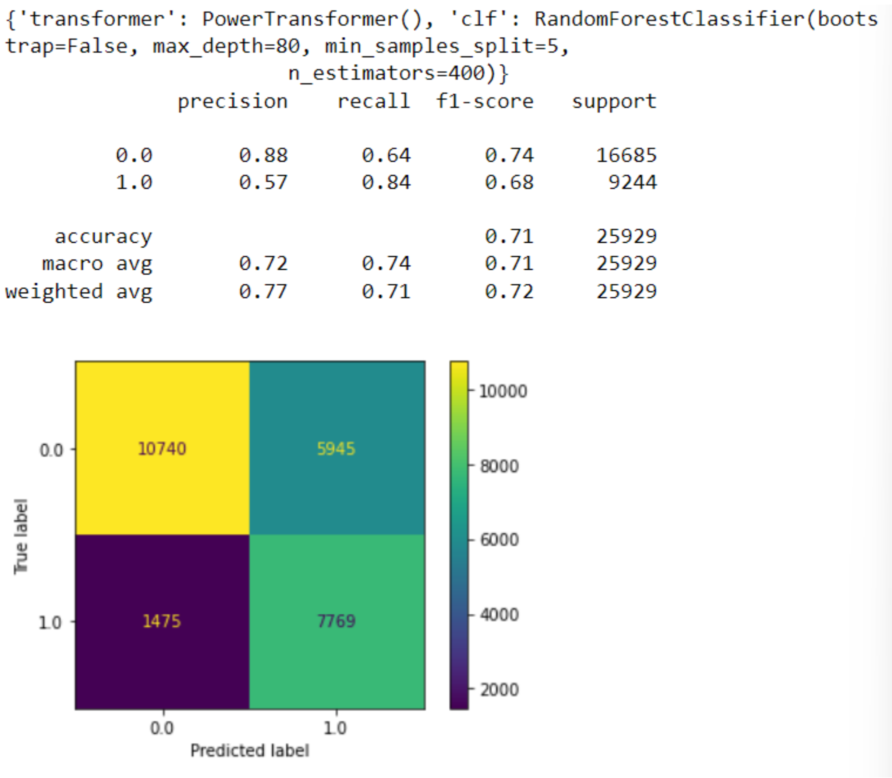


```
-----------------------------------
LogisticRegression()
-----------------------------------
fit_time  mean  0.050983524322509764
fit_time  std  0.0023148390672020737
score_time  mean  0.0066009521484375
score_time  std  0.0011996350177704931
test_score  mean  0.706482643245504
test_score  std  0.008434049625300529
-----------------------------------
SVC()
-----------------------------------
fit_time  mean  52.55921983718872
fit_time  std  0.8111808978709214
score_time  mean  15.877251195907593
score_time  std  0.2733116543664008
test_score  mean  0.7982852363028021
test_score  std  0.006688623271168158

-----------------------------------
MLPClassifier()
-----------------------------------
fit_time  mean  22.247143936157226
fit_time  std  0.6440884302199374
score_time  mean  0.017403411865234374
score_time  std  0.0008001090143977144
test_score  mean  0.7875784190715182
test_score  std  0.022480046297476755
-----------------------------------
KNeighborsClassifier(n_neighbors=3)
-----------------------------------
fit_time  mean  0.09825773239135742
fit_time  std  0.005458987340936715
score_time  mean  4.546340990066528
score_time  std  0.09302895712990493
test_score  mean  0.8147636971978252
test_score  std  0.004251622067034913

-----------------------------------
DecisionTreeClassifier()
-----------------------------------
fit_time  mean  3.355861043930054
fit_time  std  0.048091109948604424
score_time  mean  0.007220792770385742
score_time  std  0.0007695701383865034
test_score  mean  0.8890840652446675
test_score  std  0.0026668839091476162
-----------------------------------
RandomForestClassifier()
-----------------------------------
fit_time  mean  31.826132917404173
fit_time  std  0.813517019578563
score_time  mean  0.17671937942504884
score_time  std  0.009388291442039306
test_score  mean  0.9089920535340861
test_score  std  0.0021011889062451682
```

Figure 12: Classifiers' Result



```
{'transformer': PowerTransformer(), 'clf': RandomForestClassifier(boots
trap=False, max_depth=80, min_samples_split=5,
                       n_estimators=400)}
              precision    recall  f1-score   support

         0.0       0.88      0.64      0.74     16685
         1.0       0.57      0.84      0.68      9244

    accuracy                           0.71     25929
   macro avg       0.72      0.74      0.71     25929
weighted avg       0.77      0.71      0.72     25929
```

Figure 13: RandomForest Result on Test set

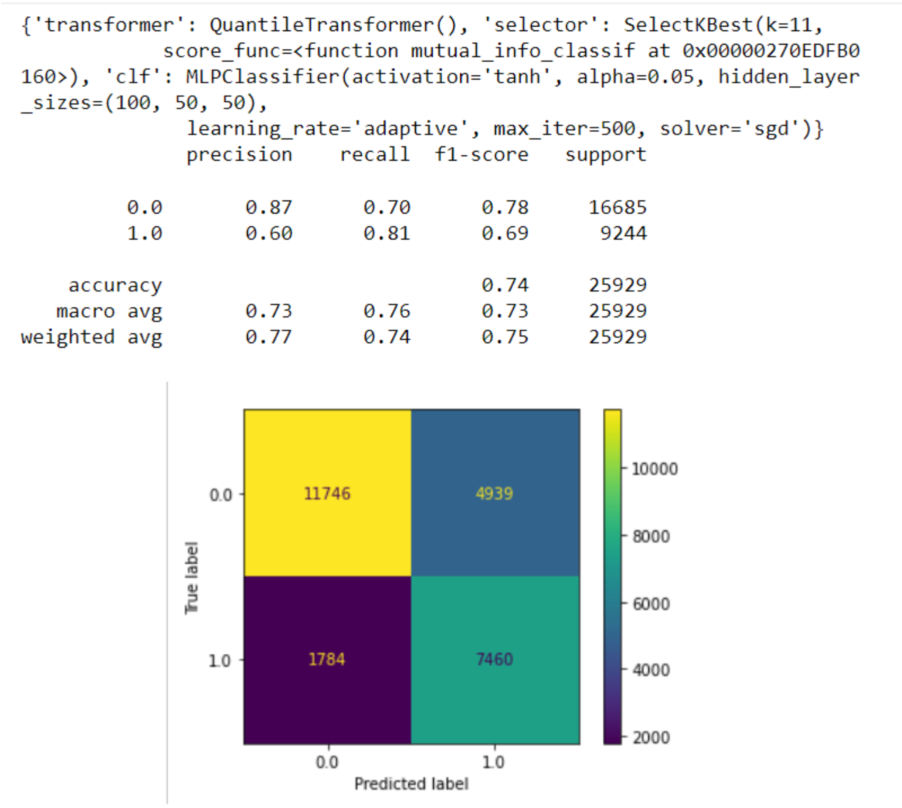
tuned MLPClassifier's hyperparameters to return good results for the meters that I am looking for as shown in Figure 14.

This time the overall Accuracy is higher than RandomForest method (approx 74%) comes with the compromise of losing 3% in the Recall meter. To judge which classification method should be used for this Supervised Learning Project. I would prefer Recall meter than the overall accuracy, in the sense that RandomForest Classifier would safer in real life diagnose purpose. Despite the fact that it performance is poor on Precision of positive patients, when facing emergency cases in reality, the option to take safer procedure is always better as the cautiousness is always needed in medical field since every one minor mistake could terribly aggravate the patient's condition.

4 Unsupervised Learning

As doing Unsupervised Learning means we do not give labels for the algorithm learn from that, but to let it decide by itself. Hence, the predicted outcome seems to be not promising. I used whole data set after dropped the target variable to apply clustering. Using Silhouette score to choose number of clusters to be 2, I have tried using Feature Agglomeration to reduce the dimension of the data set to 4 features. Then using adjusted mutual info score and adjusted rand score to measure the accuracy. The scores are 8.4% and 6.1% respectively.

Then I tried using PCA (Ding & He 2004) to reduce the dimension of the data set as in Figure 16. I chose to keep 80% of the explained variance, hence 4 components. Then proceeding on choose the number of clusters, the same Silhouette scores give the result of 2 clusters (Figure 17). At this stage, I proceed Clustering with KMeans and PCA, the illustration result is in Figure 18. However, The result with PCA method is similar to Agglomeration method.



```
{'transformer': QuantileTransformer(), 'selector': SelectKBest(k=11,
            score_func=<function mutual_info_classif at 0x00000270EDFB0
160>), 'clf': MLPClassifier(activation='tanh', alpha=0.05, hidden_layer
_sizes=(100, 50, 50),
              learning_rate='adaptive', max_iter=500, solver='sgd')}
              precision    recall  f1-score   support

         0.0       0.87      0.70      0.78     16685
         1.0       0.60      0.81      0.69      9244

    accuracy                           0.74     25929
   macro avg       0.73      0.76      0.73     25929
weighted avg       0.77      0.74      0.75     25929
```

Figure 14: MLPClassifier Result Test set

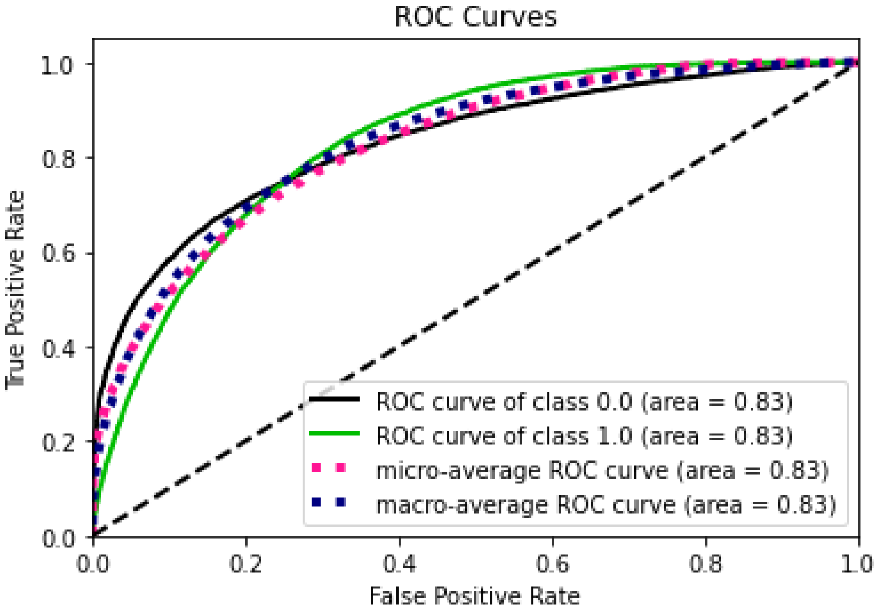


Figure 15: AUROC Curve of MLP Classifier

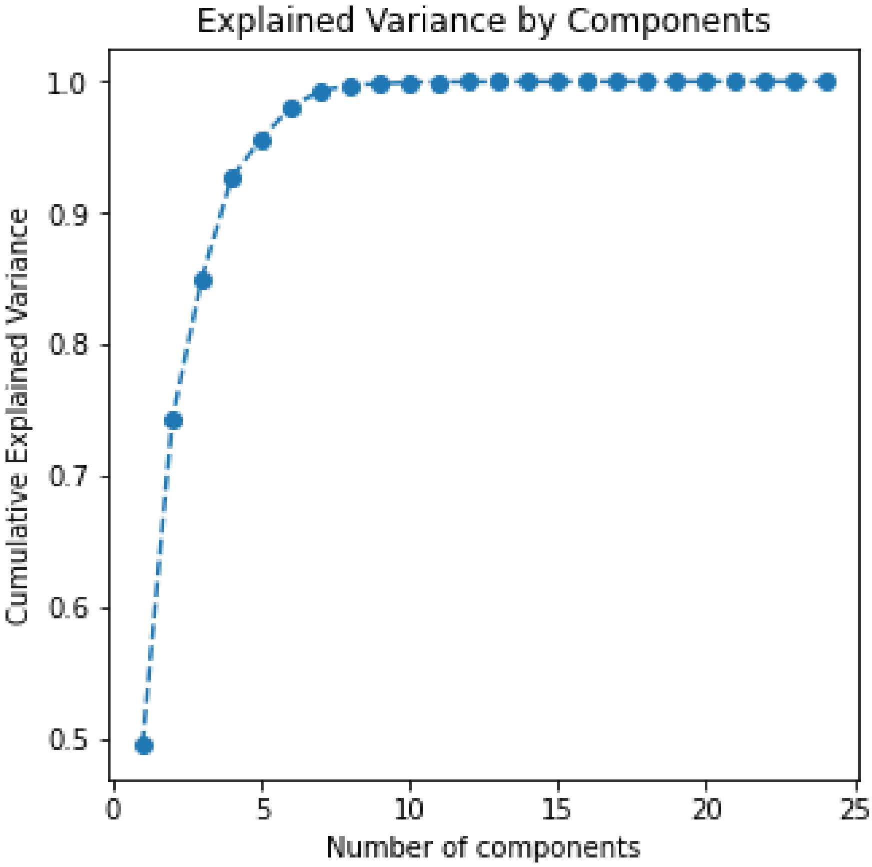


Figure 16: PCA Components

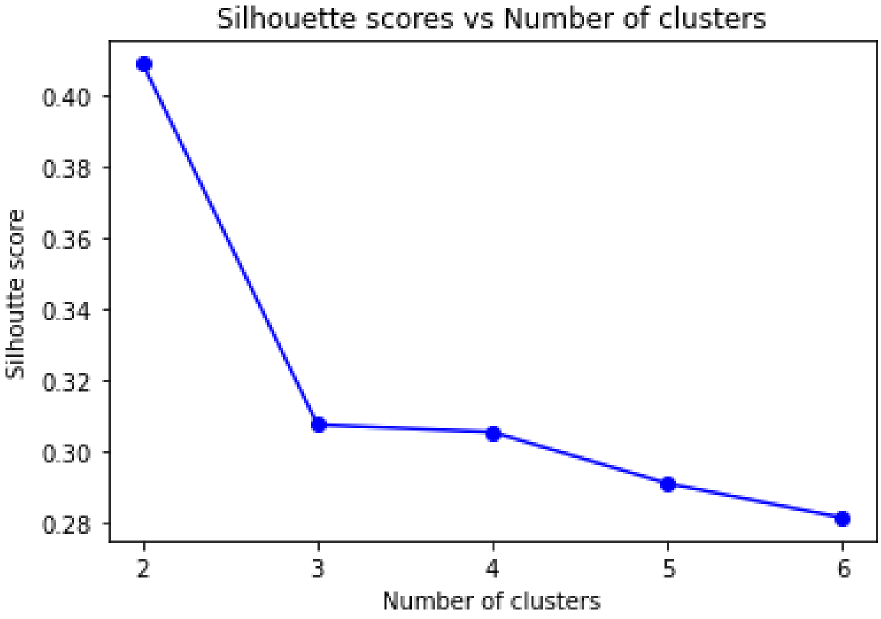


Figure 17: Silhouette Scores

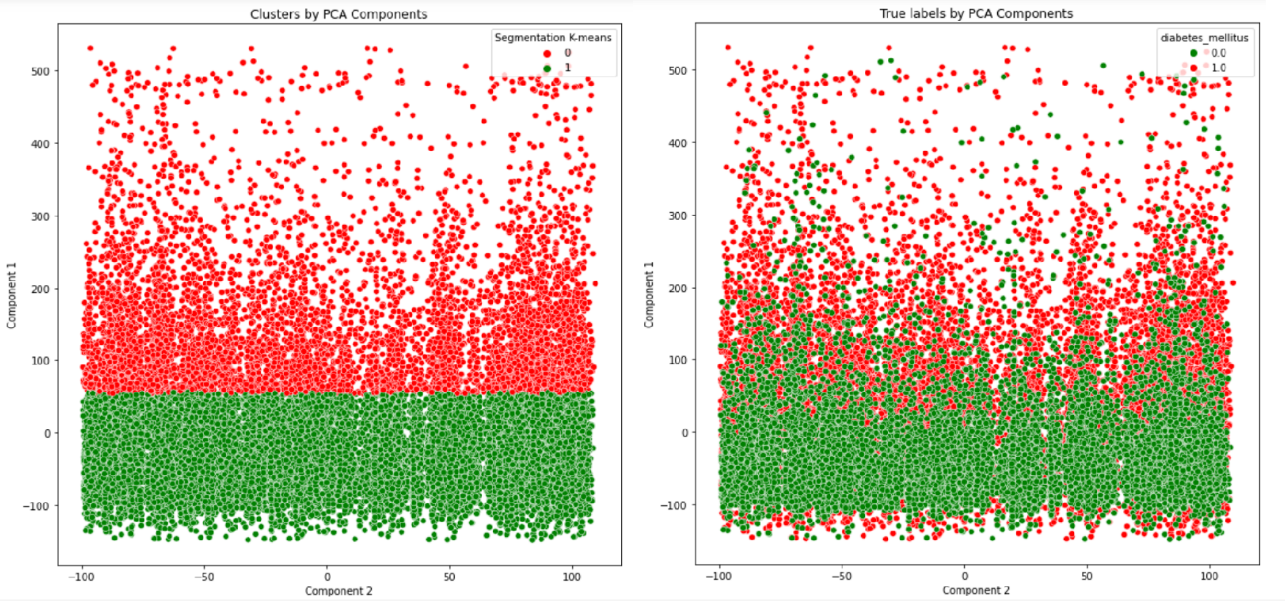


Figure 18: PCA Result on 2D dimension

5 Conclusion

Through the whole process of doing this project, follow the outlined road map, starting from understanding the project purpose, understanding the data set in hand, conducting Exploration Data Analysis (EDA), Data Cleaning and Pre-processing, then finally proceed Supervised and Unsupervised Learning. I have found that EDA processing takes the most time and effort. From there, we can have a nice set of data, ready to use for the next low-level tasks. It is also a process that greatly affects the output of algorithms in Machine Learning.

Just a small decision about whether to keep or change an element affects the authenticity of the data. This is very dangerous and difficult, since the original data set has 40% missing data. Because of the cautiousness when dealing with missingness and outliers, I use an AGILE structured workflow, which involves a lot of trial and error, to finally having the most appropriate data set. That is the reason why my Jupyter Notebook scripts is not best organised. By the time finished handling the entire Pre-Processing process, we are considered to have completed 50% of the project, the rest is to run algorithmic models and optimize them.

For this project, it is obvious to utilised Supervised than Unsupervised Learning. In Supervised Learning classification, after tuning algorithm models, the result of Random Forest and Multi-layer Perceptron return the best amongst all methods I tried, in a sense of accuracy and run time. As discussed in the above section, I would prefer Recall meter over overall accuracy since False positive in this matter is safer than False negative (Anand et al. 2018).

References

Aljumah, A. A., Ahamad, M. G. & Siddiqui, M. K. (2013), ‘Application of data mining: Diabetes health care in young and old patients’, *Journal of King Saud University - Computer and Information Sciences* **25**(2), 127–136.
URL: *https://www.sciencedirect.com/science/article/pii/S1319157812000390*

Anand, R., Stey, P., Jain, S., Biron, D., Bhatt, H., Monteiro, K., Feller, E., Ranney, M., Sarkar, I. & Chen, E. (2018), ‘Predicting mortality in diabetic icu patients using machine learning and severity indices', *AMIA Joint Summits on Translational Science proceedings. AMIA Joint Summits on Translational Science* **2017**, 310–319.

Austin, P. C., White, I. R., Lee, D. S. & van Buuren, S. (2021), 'Missing data in clinical research: A tutorial on multiple imputation', *Canadian Journal of Cardiology* **37**(9), 1322–1331.
URL: *https://www.sciencedirect.com/science/article/pii/S0828282X20311119*

Ding, C. & He, X. (2004), 'K-means clustering via principal component analysis', *Proceedings, Twenty-First International Conference on Machine Learning, ICML 2004* **1**.

Enriquez, C., Chisholm, K., Madden, L., Larsen, A., Longpré, T. & Stannard, D. (2019), 'Glasgow coma scale: Generating clinical standards', *Journal of Neuroscience Nursing* **51**, 142–146.

Jakobsen, J., Gluud, C., Wetterslev, J. & Winkel, P. (2017), 'When and how should multiple imputation be used for handling missing data in randomised clinical trials - a practical guide with flowcharts', *BMC Medical Research Methodology* **17**.

Kavakiotis, I., Tsave, O., Salifoglou, A., Maglaveras, N., Vlahavas, I. & Chouvarda, I. (2017), 'Machine learning and data mining methods in diabetes research', *Computational and Structural Biotechnology Journal* **15**, 104–116.
URL: *https://www.sciencedirect.com/science/article/pii/S2001037016300733*

Maniruzzaman, M., Rahman, M., Hasan, M. A., Suri, H., Abedin, M., El-Baz, A. & Suri, J. (2018), 'Accurate diabetes risk stratification using machine learning: Role of missing value and outliers', *Journal of Medical Systems* **42**.

Mukaka, M. (2012), 'Statistics corner: A guide to appropriate use of correlation coefficient in medical research.', *Malawi medical journal : the journal of Medical Association of Malawi* **24 3**, 69–71.

Ong, J. & Penm, J. (2019), 'How to understand and interpret clinical data', *The Pharmaceutical Journal* **303**.
URL: *https://pharmaceutical-journal.com/article/ld/how-to-understand-and-interpret-clinical-data*

Santilli, F., Simeone, P., Liani, R. & Davì, G. (2015), 'Platelets and diabetes mellitus', *Prostaglandins Other Lipid Mediators* **120**, 28–39. Eicosanoids and related compounds.
URL: *https://www.sciencedirect.com/science/article/pii/S1098882315000544*

Schweizer, K., Gold, A., Krampen, D. & Wang, T. (2020), 'On modeling missing data of an incomplete design in the cfa framework', *Frontiers in Psychology* **11**.
URL: *https://www.frontiersin.org/article/10.3389/fpsyg.2020.581709*

Torkey, H., Ibrahim, E., Hemdan, E. E.-D., El-Sayed, A. & Shouman, M. (2021), 'Diabetes classification application with efficient missing and outliers data handling algorithms', *Complex Intelligent Systems* .

Yang, H., Luo, Y., Ren, X., Wu, M., He, X., Peng, B., Deng, K., Yan, D., Tang, H. & Lin, H. (2021), 'Risk prediction of diabetes: Big data mining with fusion of multifarious physical examination indicators', *Information Fusion* **75**, 140–149.
URL: *https://www.sciencedirect.com/science/article/pii/S1566253521000397*