Healthcare Capstone Project

Prateek Gharat

S S K M Chaitanya PUSULURI

Project Objective

Hospital Compare includes information on over 100 quality measures and more than 4,000 hospitals. The primary objective of the Overall Hospital Quality Star Ratings project is to develop a statistically sound methodology for summarizing information from the existing measures on Hospital Compare in a way that is useful and easy to interpret for patients and consumers. Consistent with other CMS Star Rating programs, this methodology assigns each hospital between one and five stars, reflecting the hospital's overall performance on selected quality measures.

CMS intends for the Overall Hospital Quality Star Ratings to complement existing efforts, such as the Hospital Consumer Assessment of Healthcare Providers and Systems (HCAHPS) star rating (implemented in April 2015), and will continue to report individual quality measures for stakeholders seeking more detailed information.

Approach to Grouping Measures

To calculate the Overall Hospital Quality Star Rating, the measures are categorized into **seven mutually exclusive** groups and 4 main **domains**.

❖1. Outcomes – Mortality (7 measures)

❖2. Outcomes – Safety of Care (8 measures)

❖3. Outcomes – Readmissions (8 measures)

❖4. Patient Experience (11 measures)

❖5. Process – Effectiveness of Care (18 measures)

❖6. Process – Timeliness of Care (7 measures)

❖7. Efficiency – Outpatient Imaging Use (5 measures)

These seven groups generally align with the categories on the Hospital Compare website, the CMS Hospital Value-Based Purchasing (VBP) Program, and other national quality initiatives.

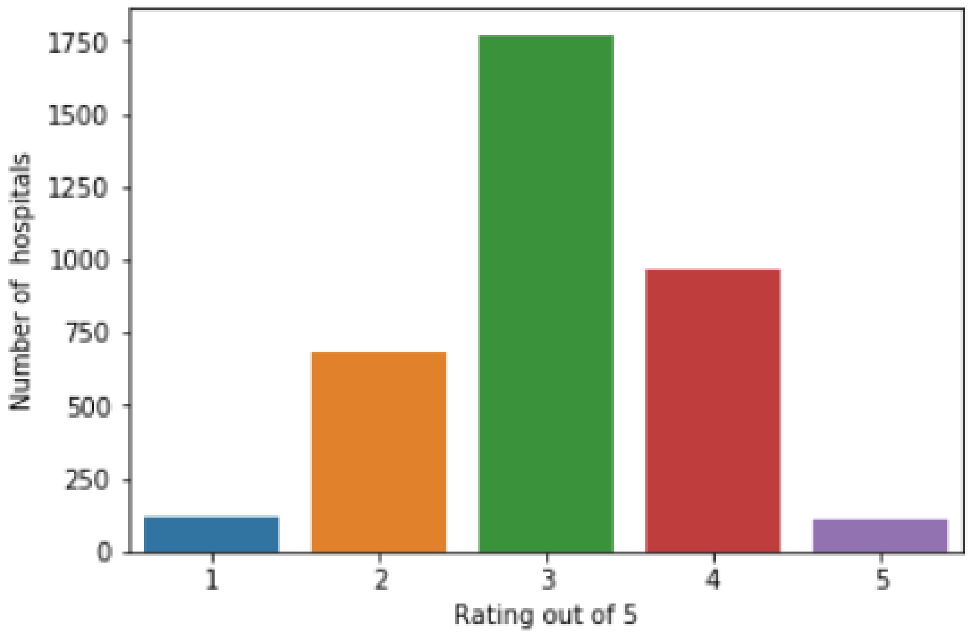
Measures Grouping details

Readmission Measures	Imaging Measures	Timelycare Measures	Effectivecare Measures	Mortality Measures	Safetycare Measures	Patient Experience Measures
READM_30_AMI	OP_8	ED_1b	CAC_3	MORT_30_AMI	COMP_HIP_KNEE	H_CLEAN_LINEAR_SCORE
READM_30_CABG	OP_10	ED_2b	IMM_2	MORT_30_CABG	HAI_1_SIR	H_COMP_1_LINEAR_SCORE
READM_30_COPD	OP_11	OP_3b	IMM_3_OP_27_FAC_ADHPCT	MORT_30_COPD	HAI_2_SIR	H_COMP_2_LINEAR_SCORE
READM_30_HF	OP_13	OP_5	OP_4	MORT_30_HF	HAI_3_SIR	H_COMP_3_LINEAR_SCORE
READM_30_HIP_KNEE	OP_14	OP_18b	OP_22	MORT_30_PN	HAI_4_SIR	H_COMP_4_LINEAR_SCORE
READM_30_PN		OP_20	OP_23	MORT_30_STK	HAI_5_SIR	H_COMP_5_LINEAR_SCORE
READM_30_STK		OP_21	OP_29	PSI_4_SURG_COMP	HAI_6_SIR	H_COMP_6_LINEAR_SCORE
READM_30_HOSP_WIDE			OP_30		PSI_90_SAFETY	H_COMP_7_LINEAR_SCORE
			PC_01			H_HSP_RATING_LINEAR_SCORE
			STK_1			H_QUIET_LINEAR_SCORE
			STK_4			H_RECMND_LINEAR_SCORE
			STK_6			
			STK_8			
			VTE_1			
			VTE_2			
			VTE_3			
			VTE_5			
			VTE_6			

Process and Methodology

- **Business Understanding:**
  - It deals with understanding the overall goal and input parameter functionality to achieve the desired output.
- **Data Understanding:**
  - Understanding the all the required files and extracting relevant columns to perform the model building and gain high accuracy.
  - Reading of multiple files including Hospital General Information file and performing basic operation like checking the data type, shape, summary, outliers, missing values, etc. from data frame.
  - Identifying the measure id and categorise as per the 7 mutually exclusive groups and creating a pivot table.
- **Data Cleaning:**
  - Replacement of Not Available values with nan and identifying the actual count of missing values from data.
  - Deriving a new column to with the count of missing values in a row.
  - Imputing the missing values with mean and checking for distribution of data.
- **Data pre-processing:**
  - Using quantile transformation to transforms the features to follow a uniform or a normal distribution. Also this method reduces the impact of (marginal) outliers.
- **Exploratory data analysis (EDA):**
  - Plotting the various graphs to understand the distribution of data like overall star rating, count of hospitals as per state, type and ownership, national comparison, count of measure id with measure type.
  - Normalization plots of measure groups to understand the overall behavior.
  - Checking for the correlation between measure id for measure group.
- **Model building and Evaluation:**
  - Splitting the dataset in train test and validation to perform model building.
  - Clustering the dataset as per the CMS standards.
  - Recommendations to improve the hospital star rating based modelling features.

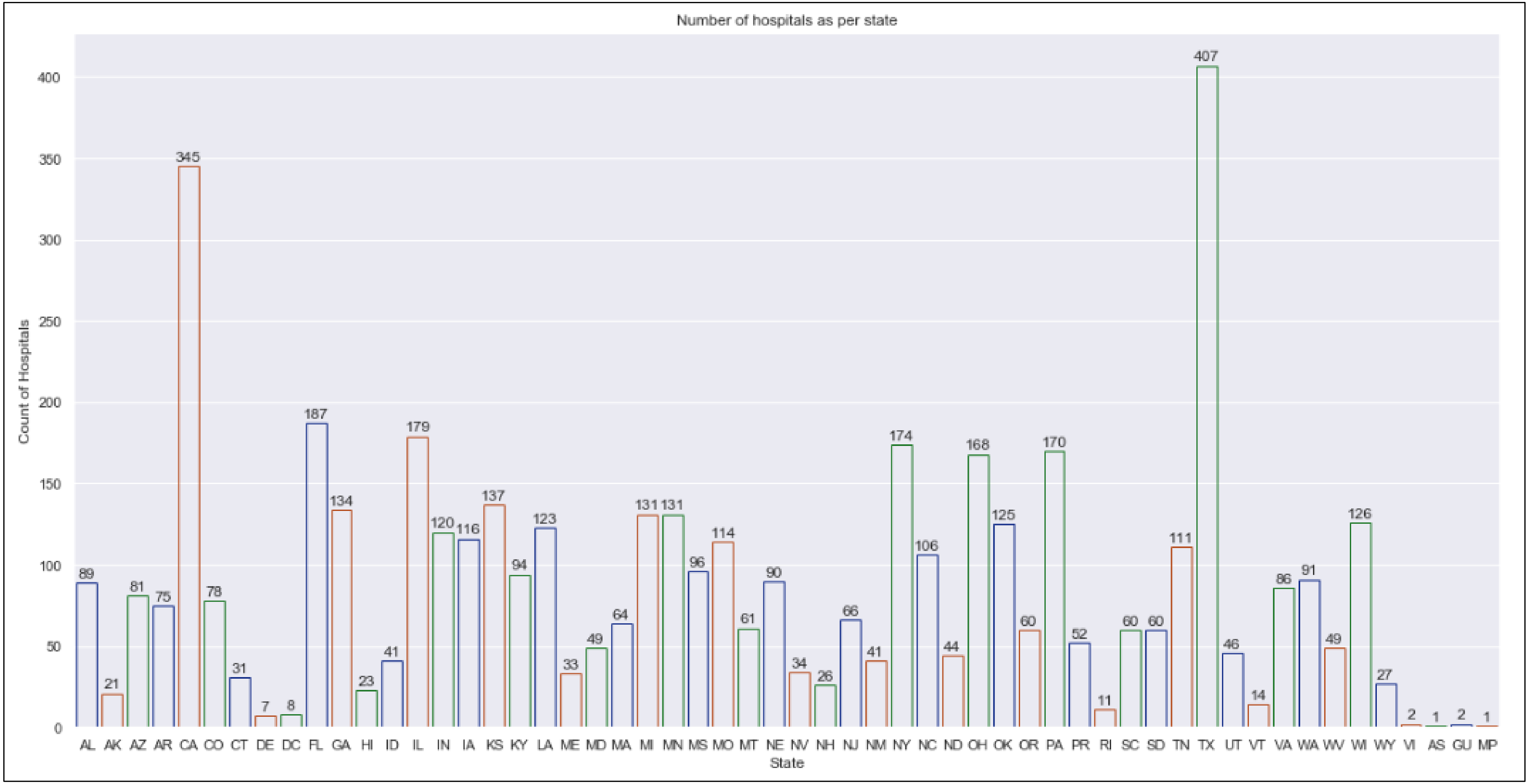
Star Rating



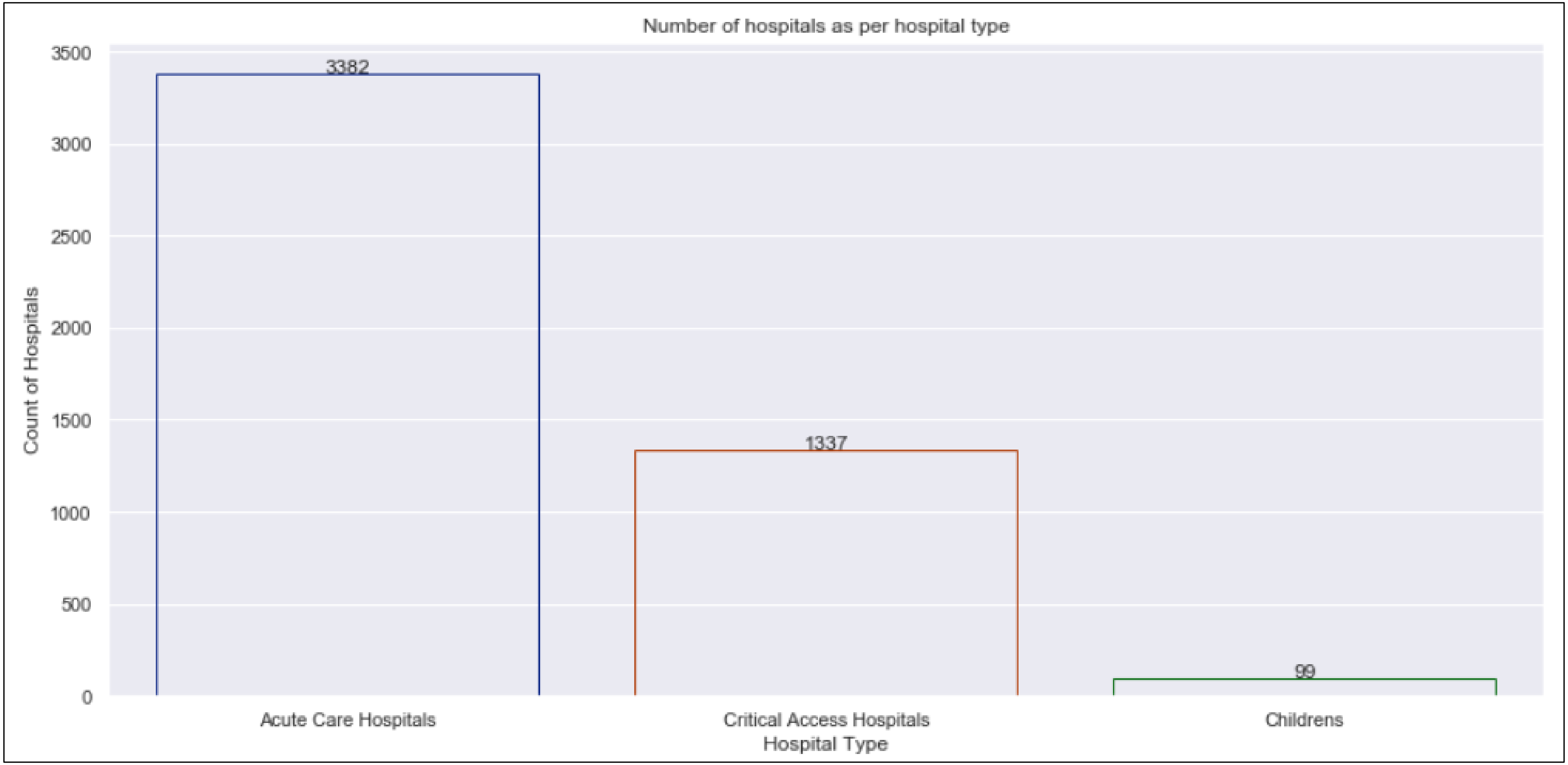
Rating	Count	Percentage
3	1772	36.78%
Not Available	1170	24.28%
4	964	20.01%
2	684	14.20%
1	117	2.43%
5	111	2.30%

- It was found that 24% of hospital data has no star rating due to various reasons mentioned in footnotes.
- Out of all 4818 hospital listed 36.78% of the hospital has star rating of 3.
- 117 hospitals which have star rating of 1 should emphasize on improving there star rating.
- At the same time 2.30% of hospitals i.e. 111 hospital received 5 star rating and other hospitals should understand the quality of services provided by these 5 star hospitals.

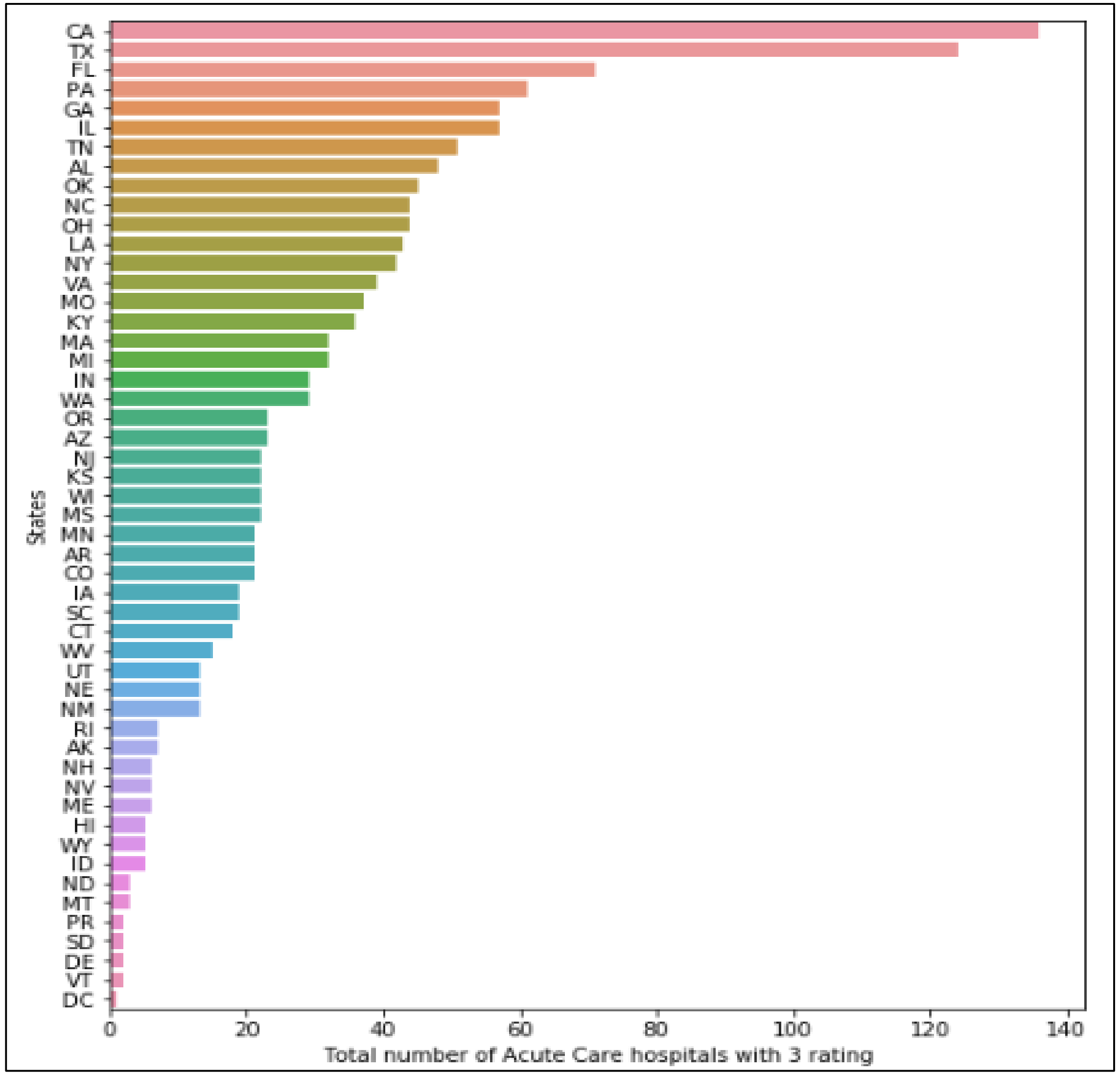
Hospitals as per State

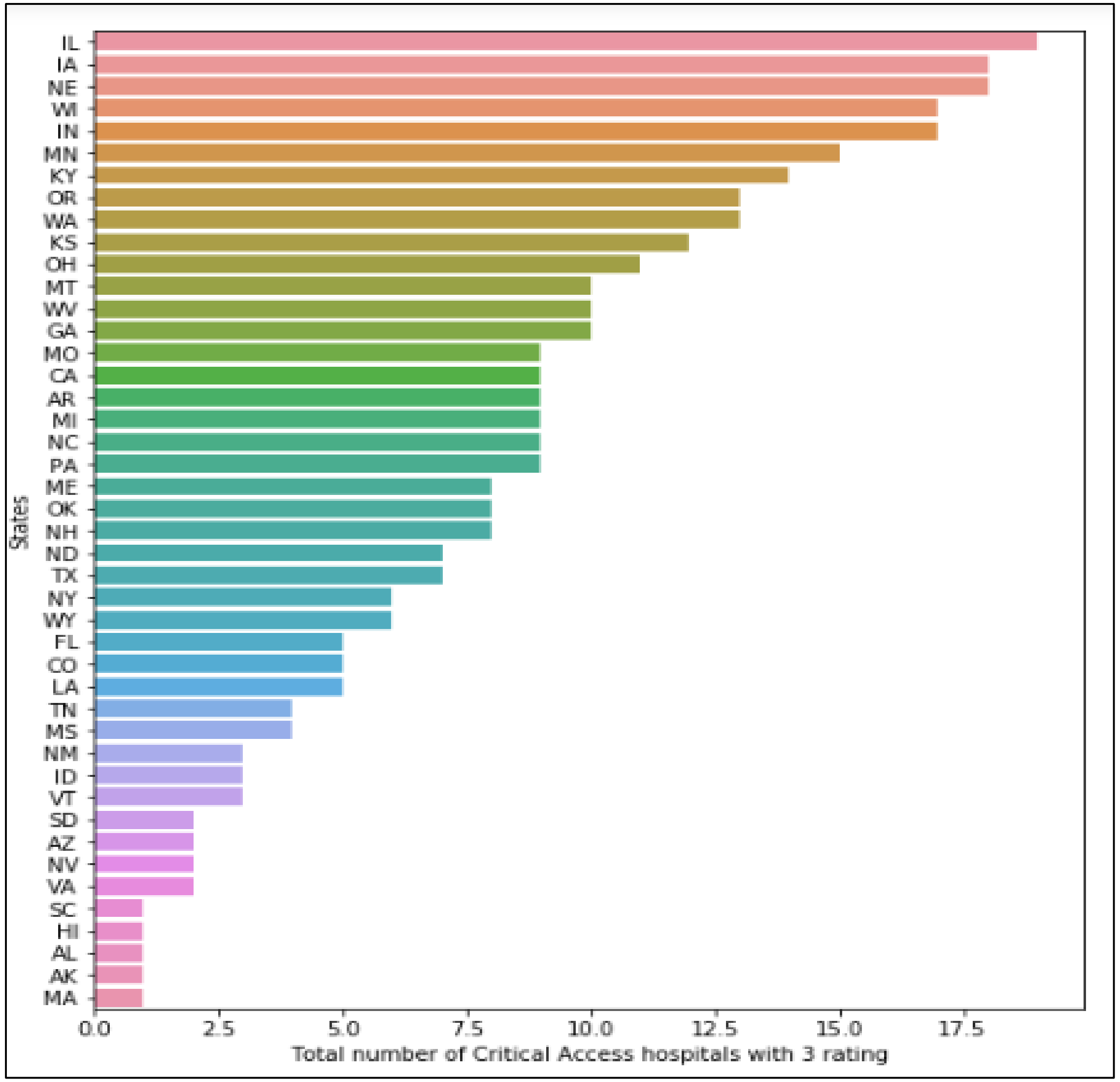


Hospitals As per Hospital Type



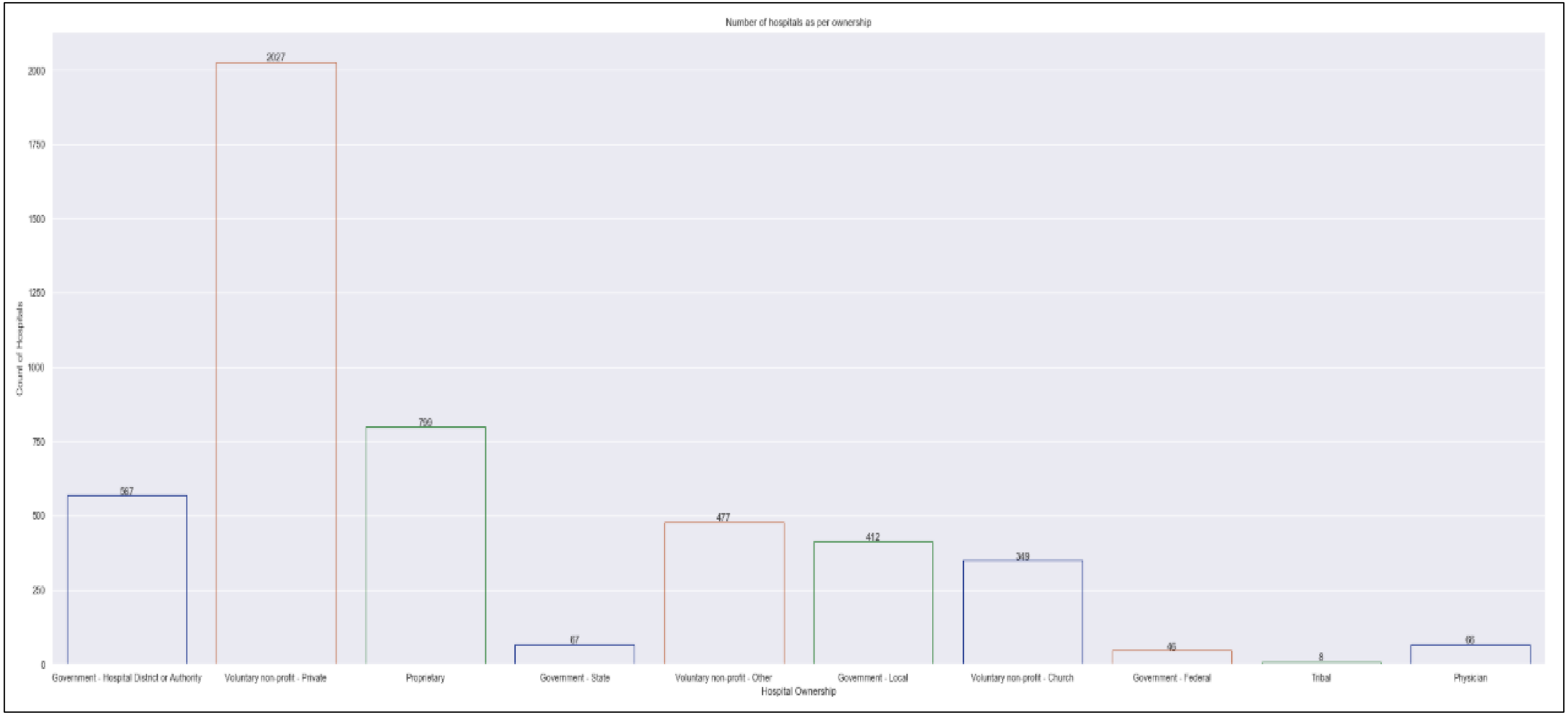
Hospital type with 3 rating



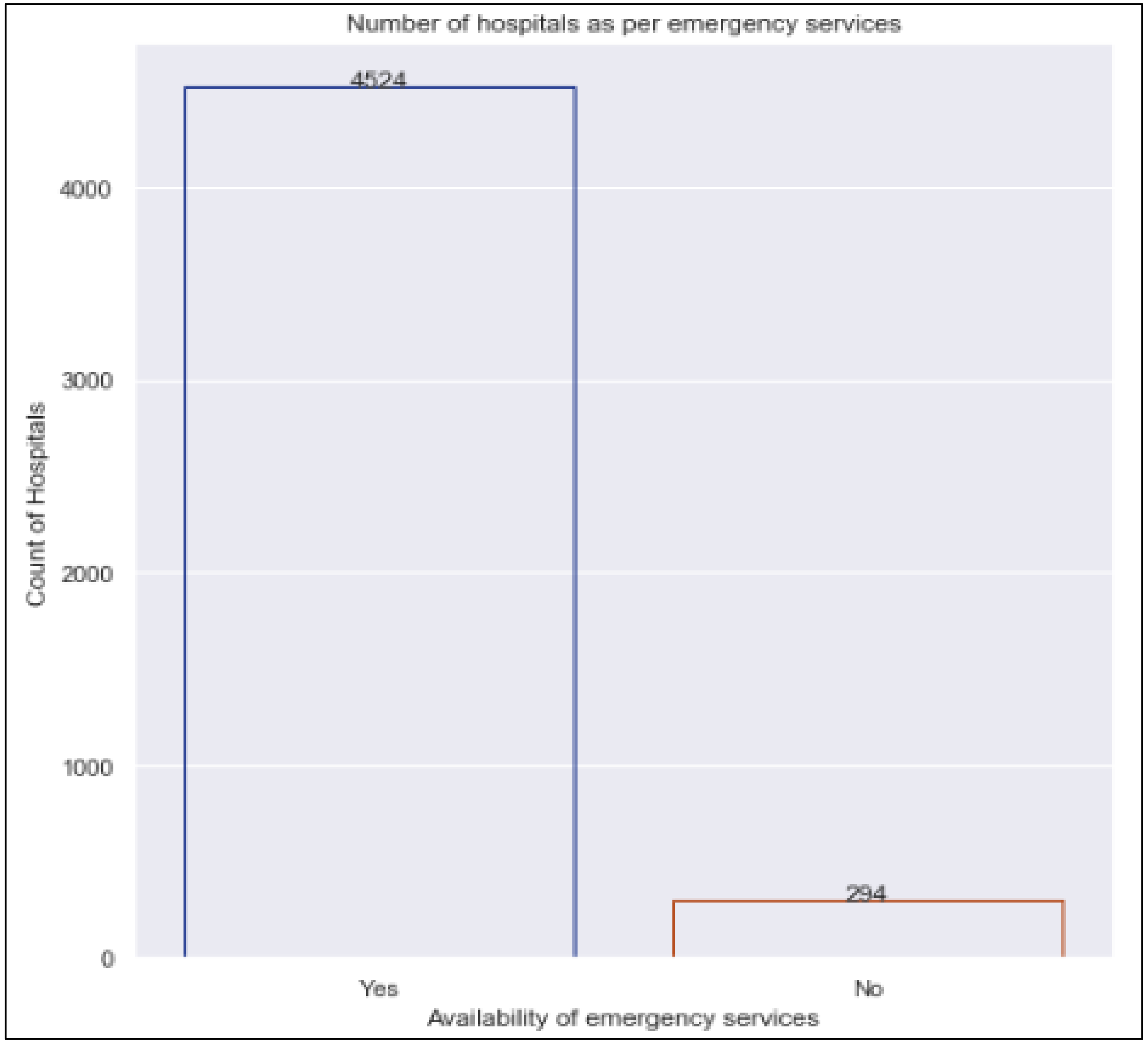


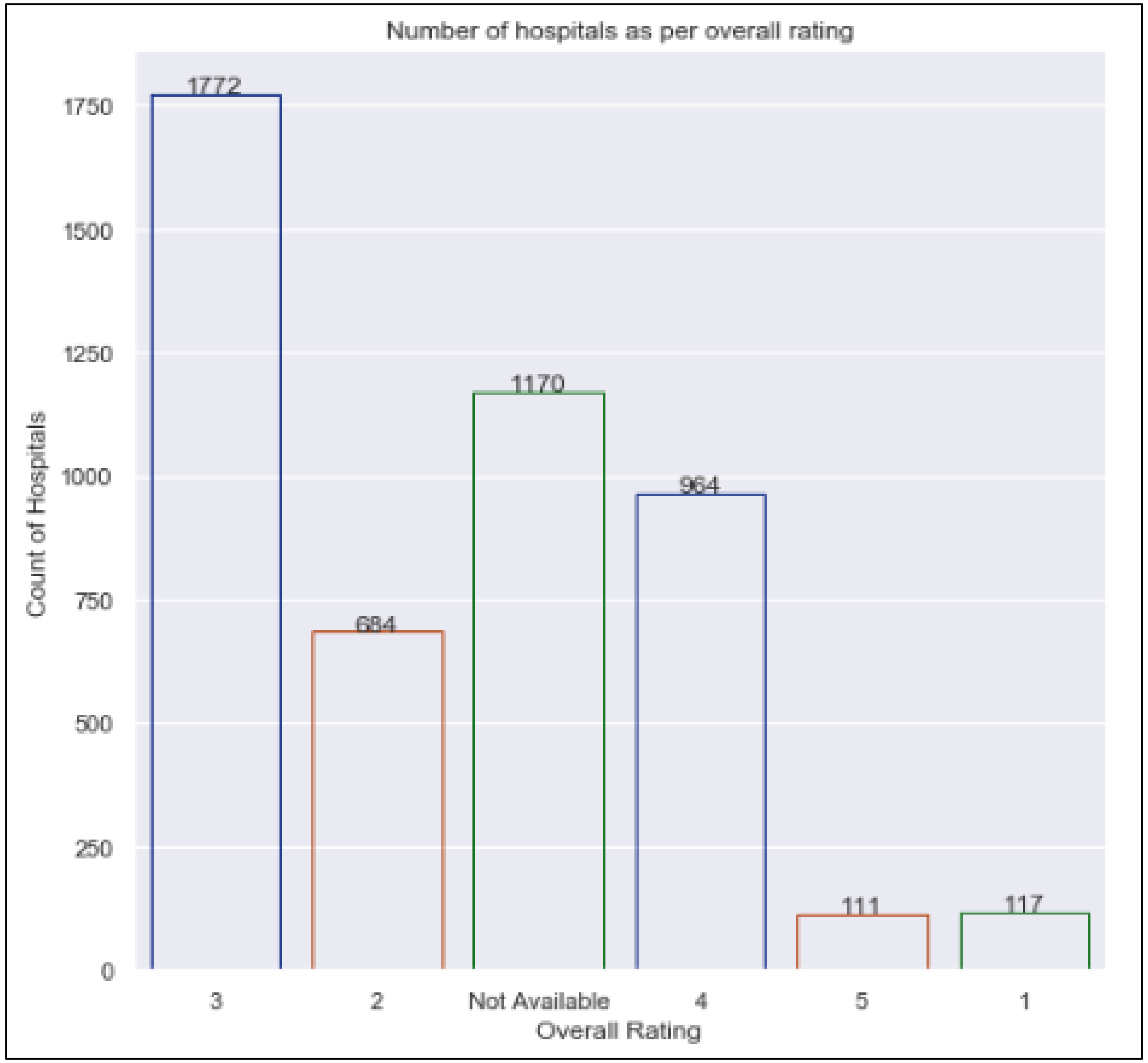
Note: Hospital type Childrens has no ratings

Hospitals As per ownership

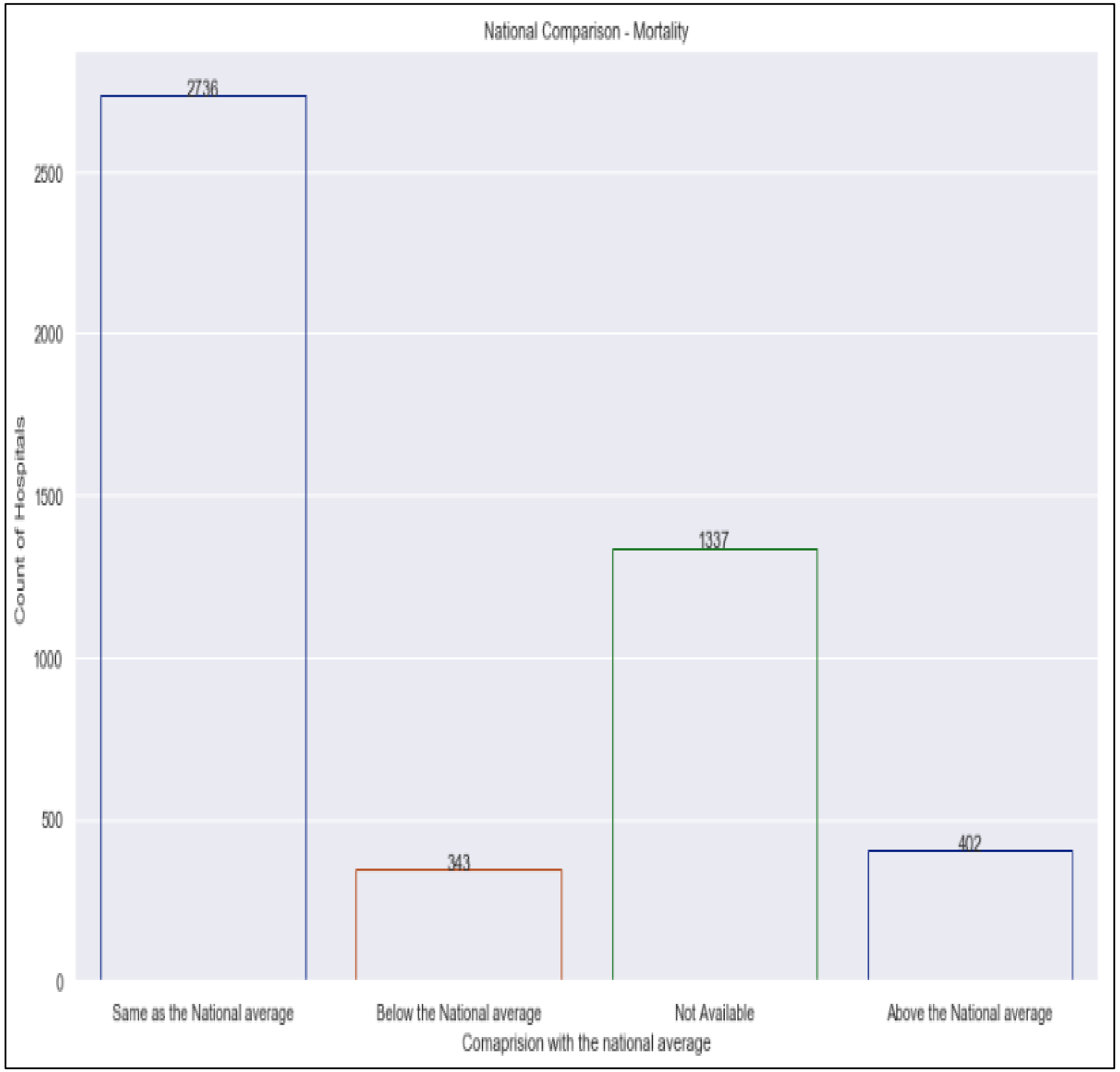


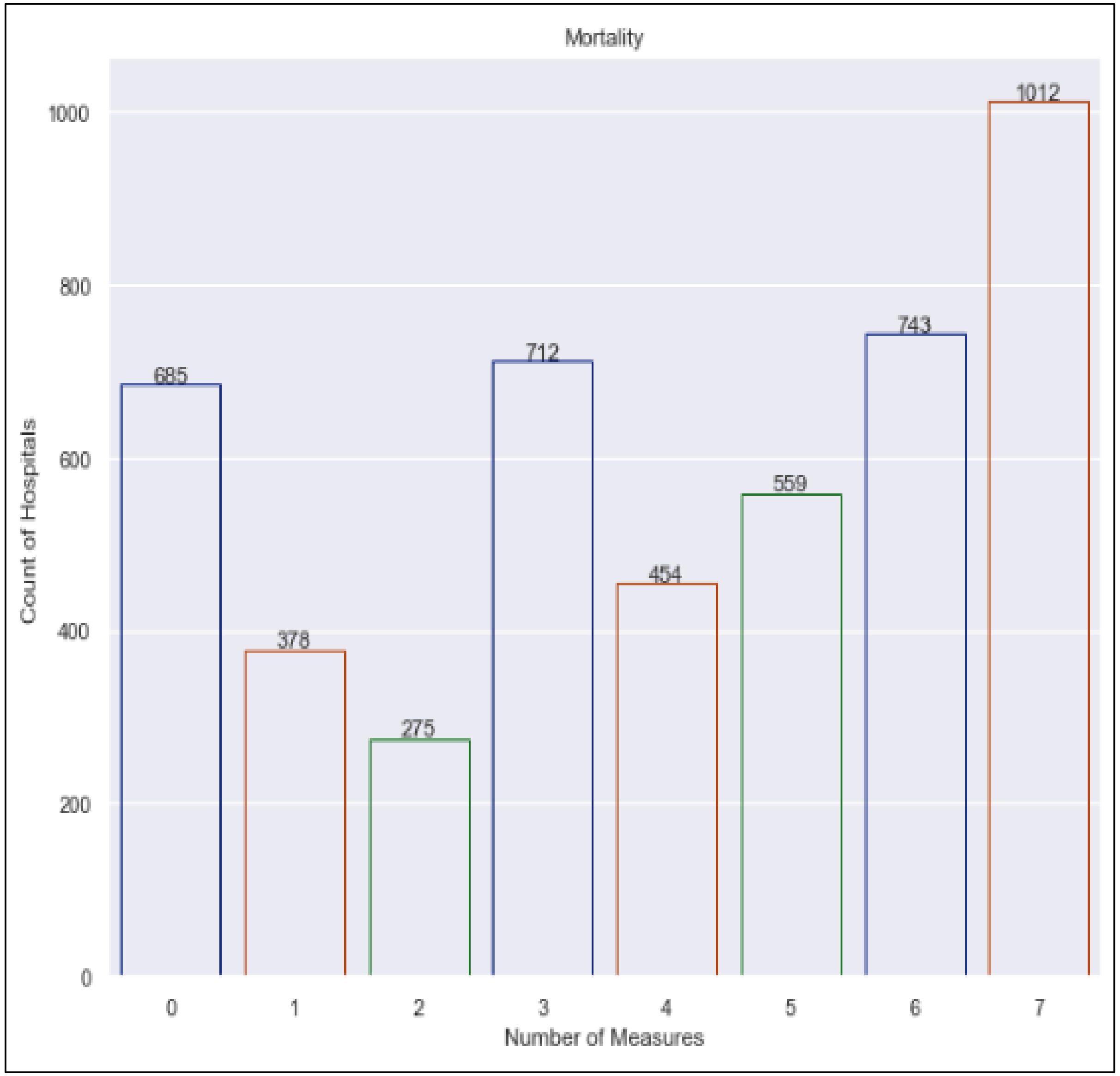
Hospitals As per availability of emergency services and overall rating



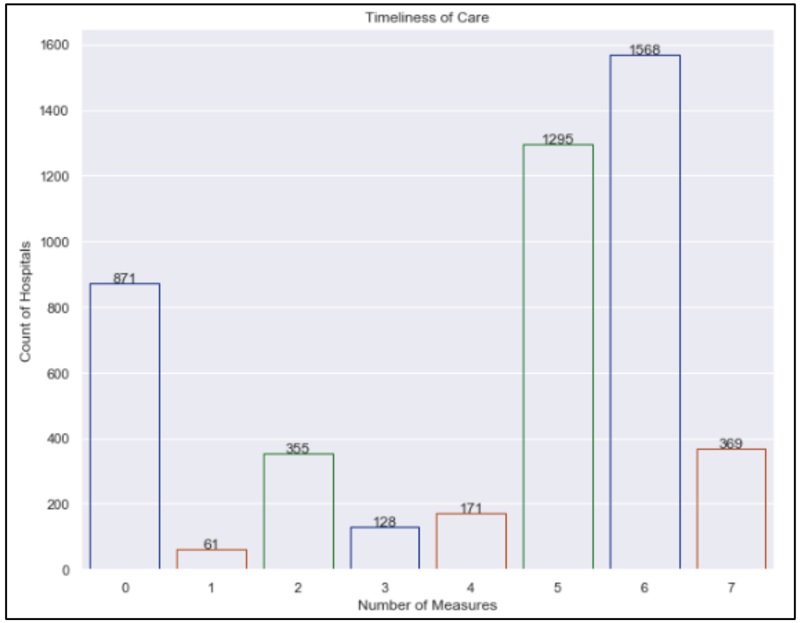


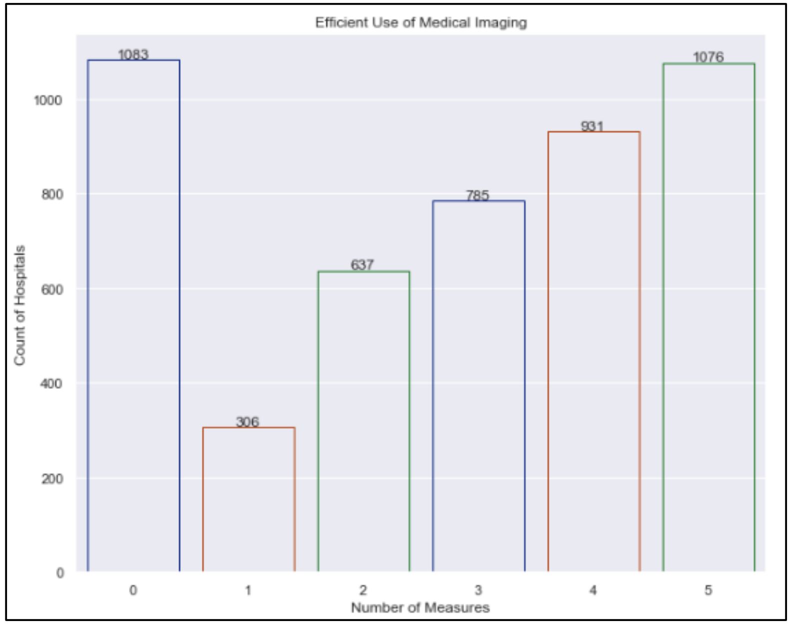
Hospitals As per National Comparison - Mortality

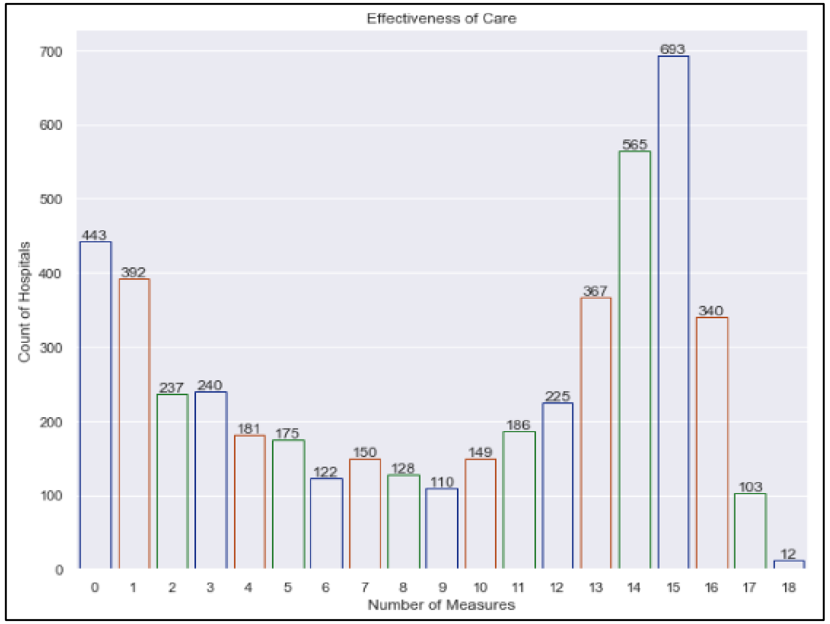


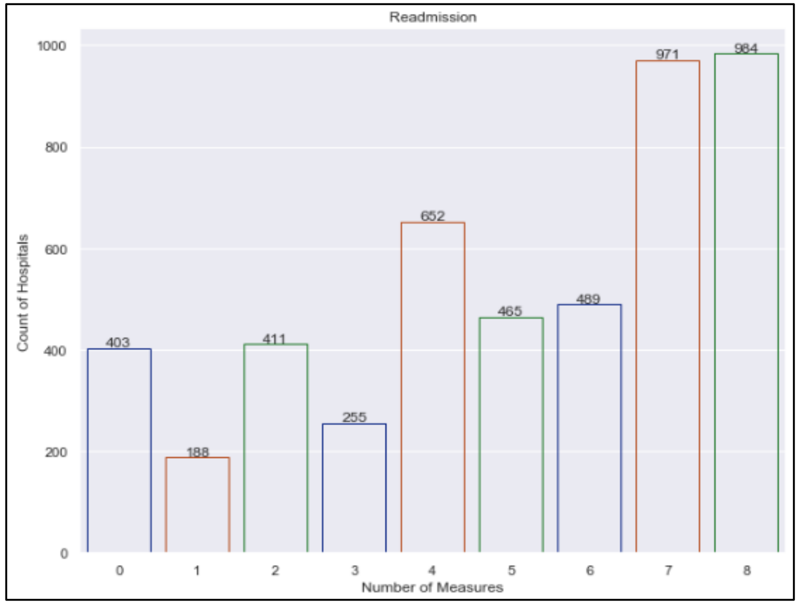


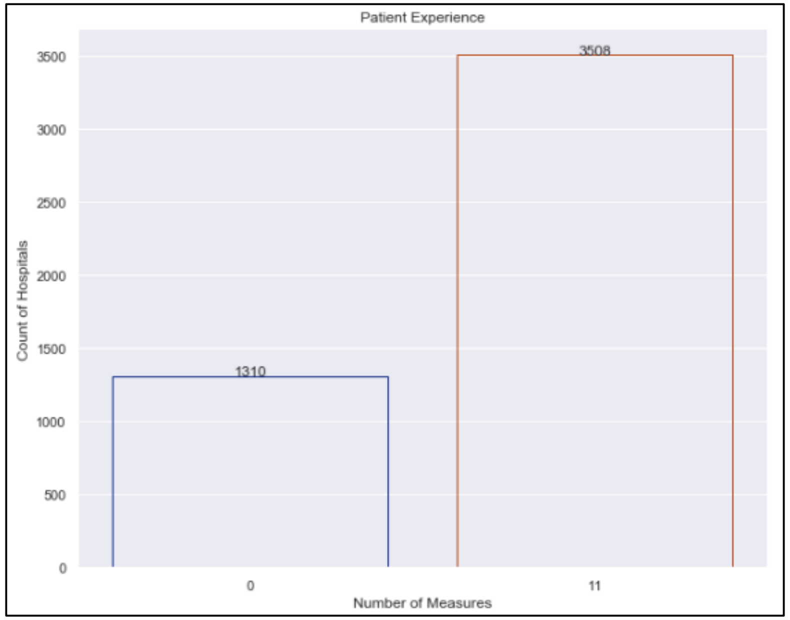
Hospitals Measure Id and Measure Groups

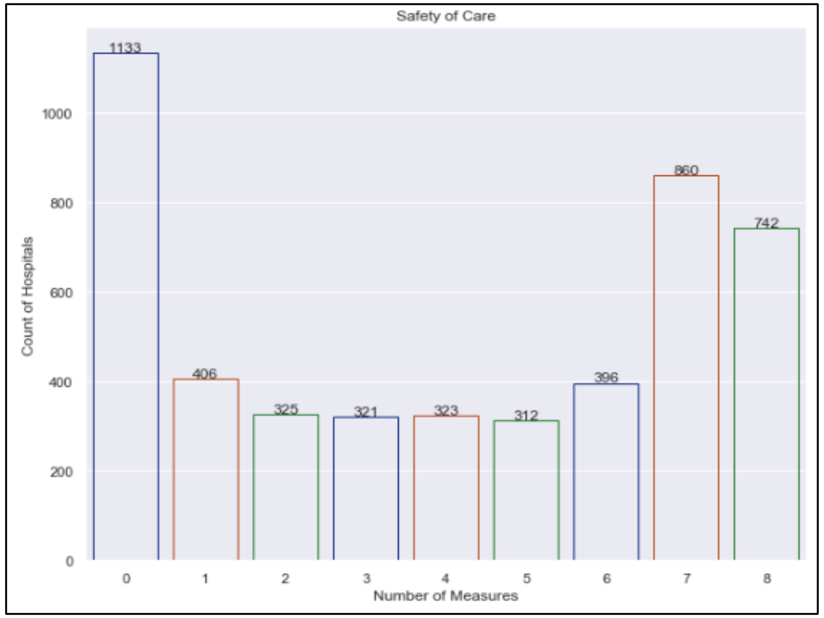




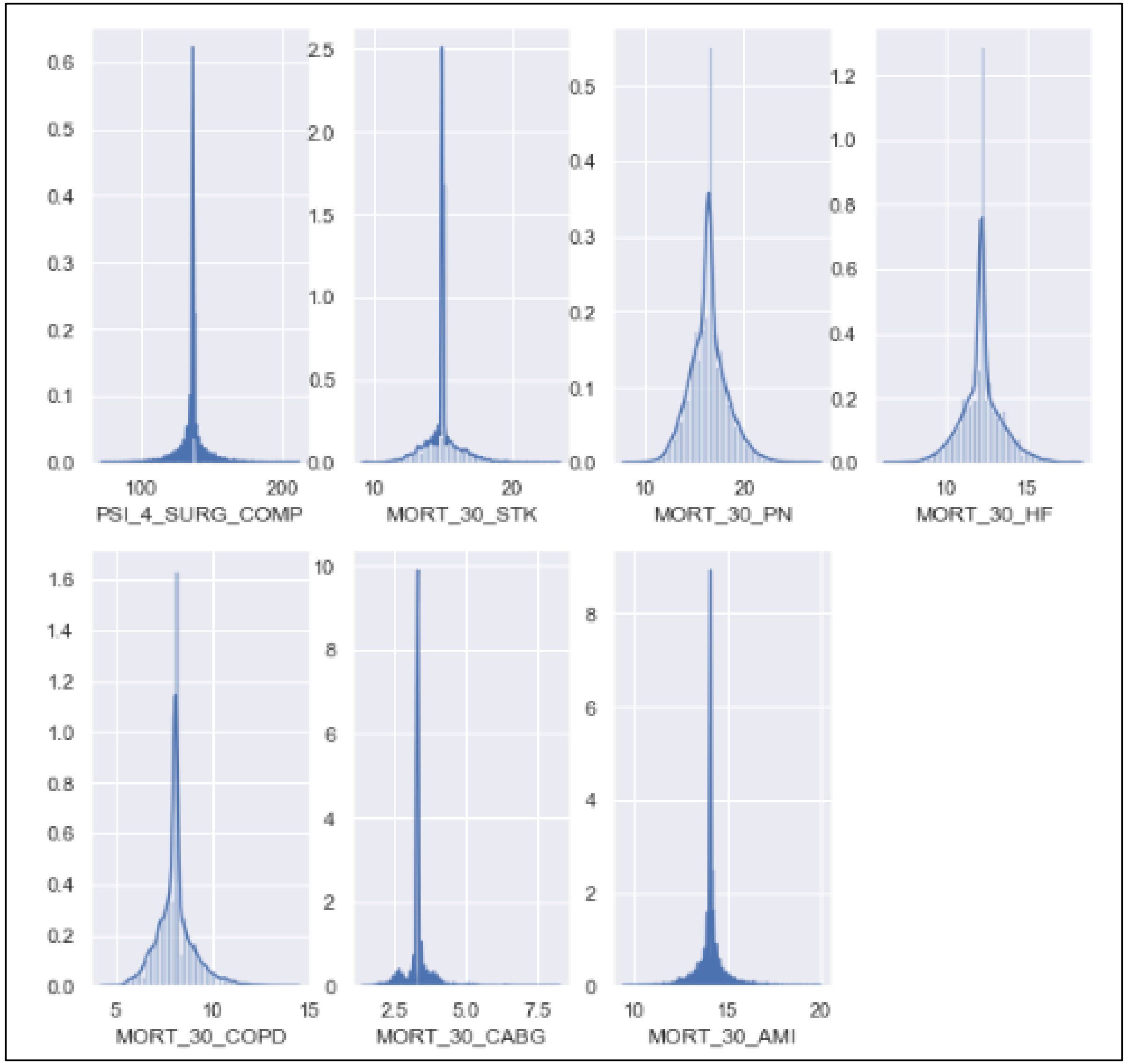


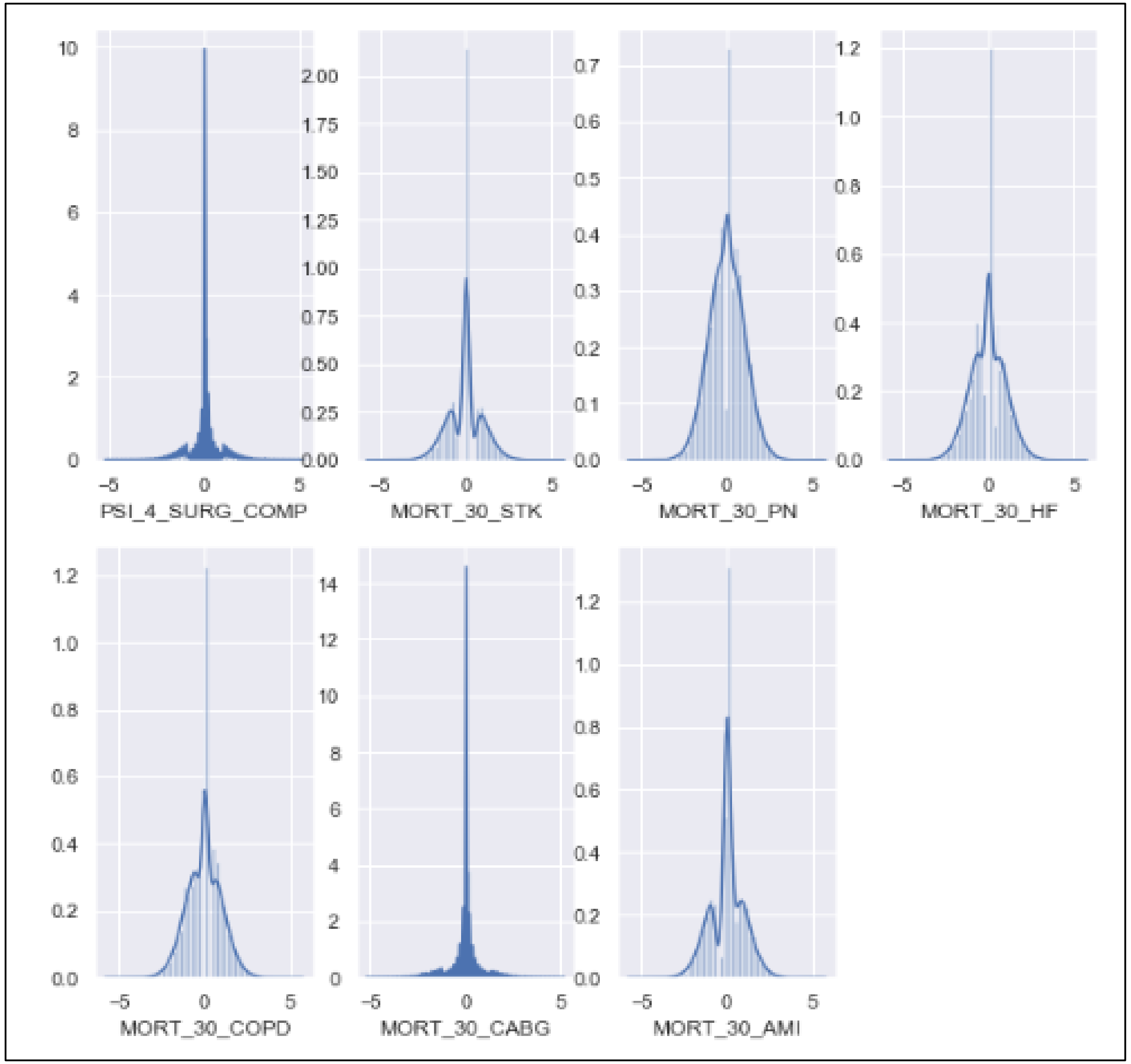




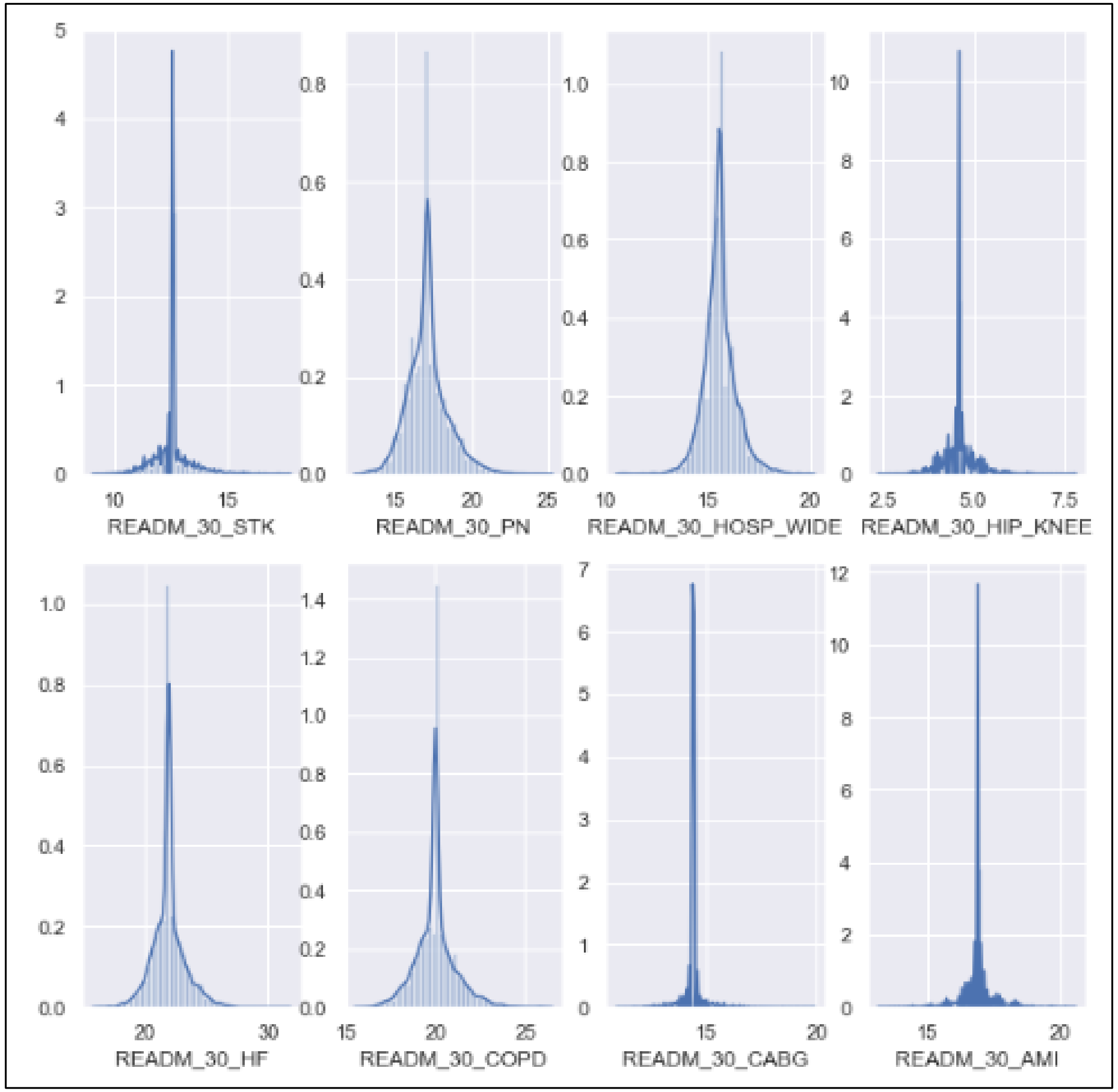


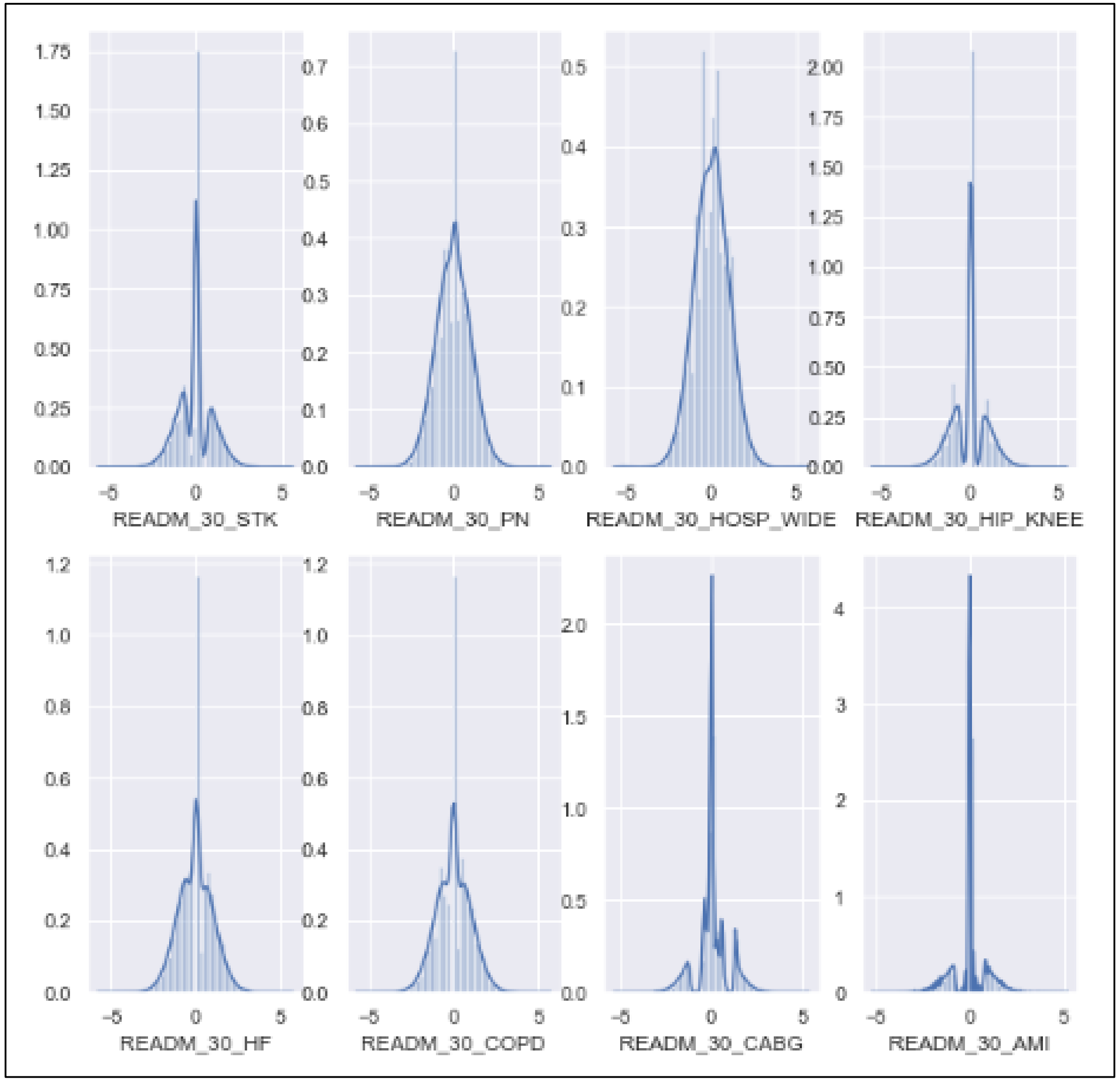
Distribution of Mortality Normalization



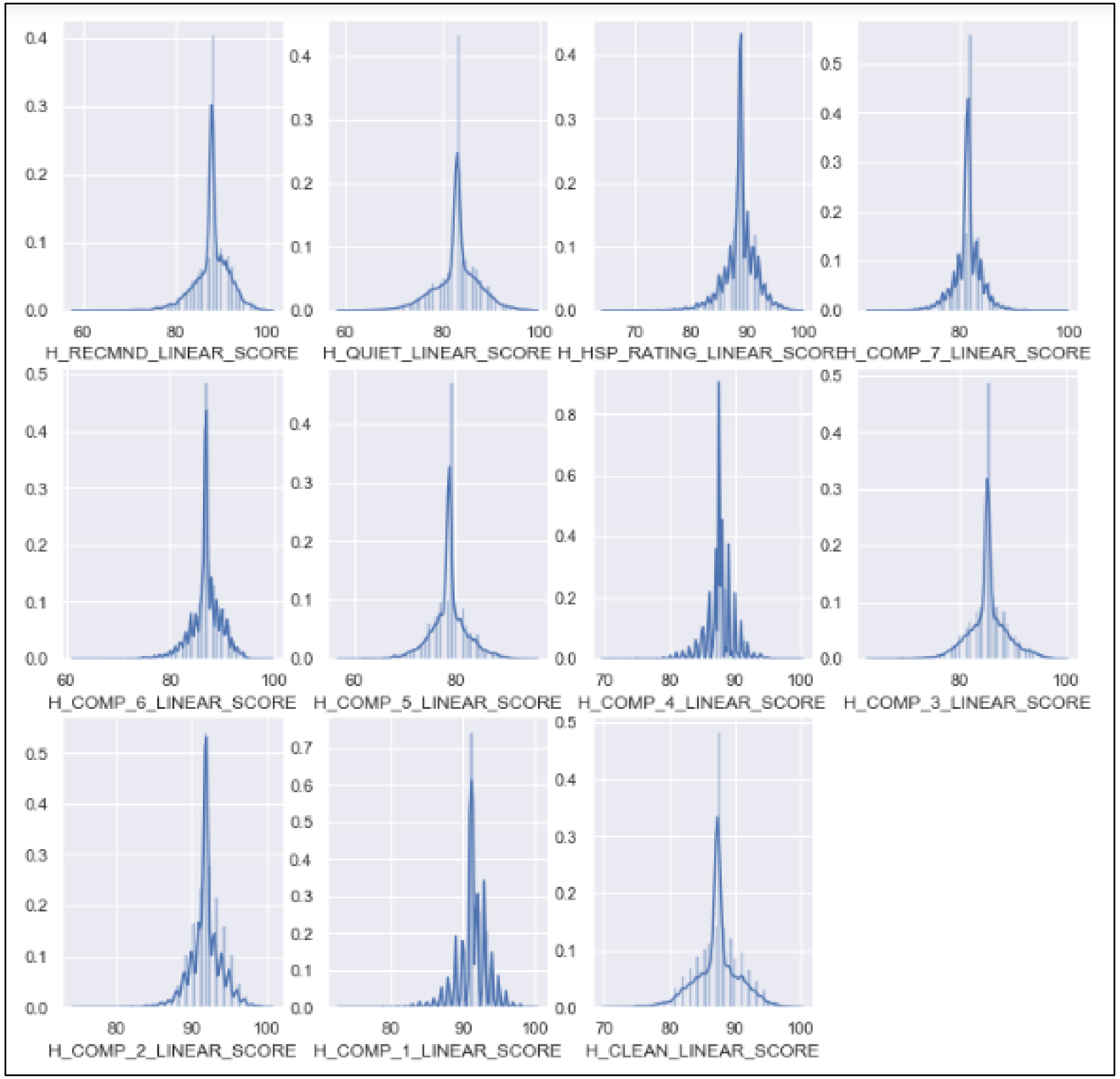


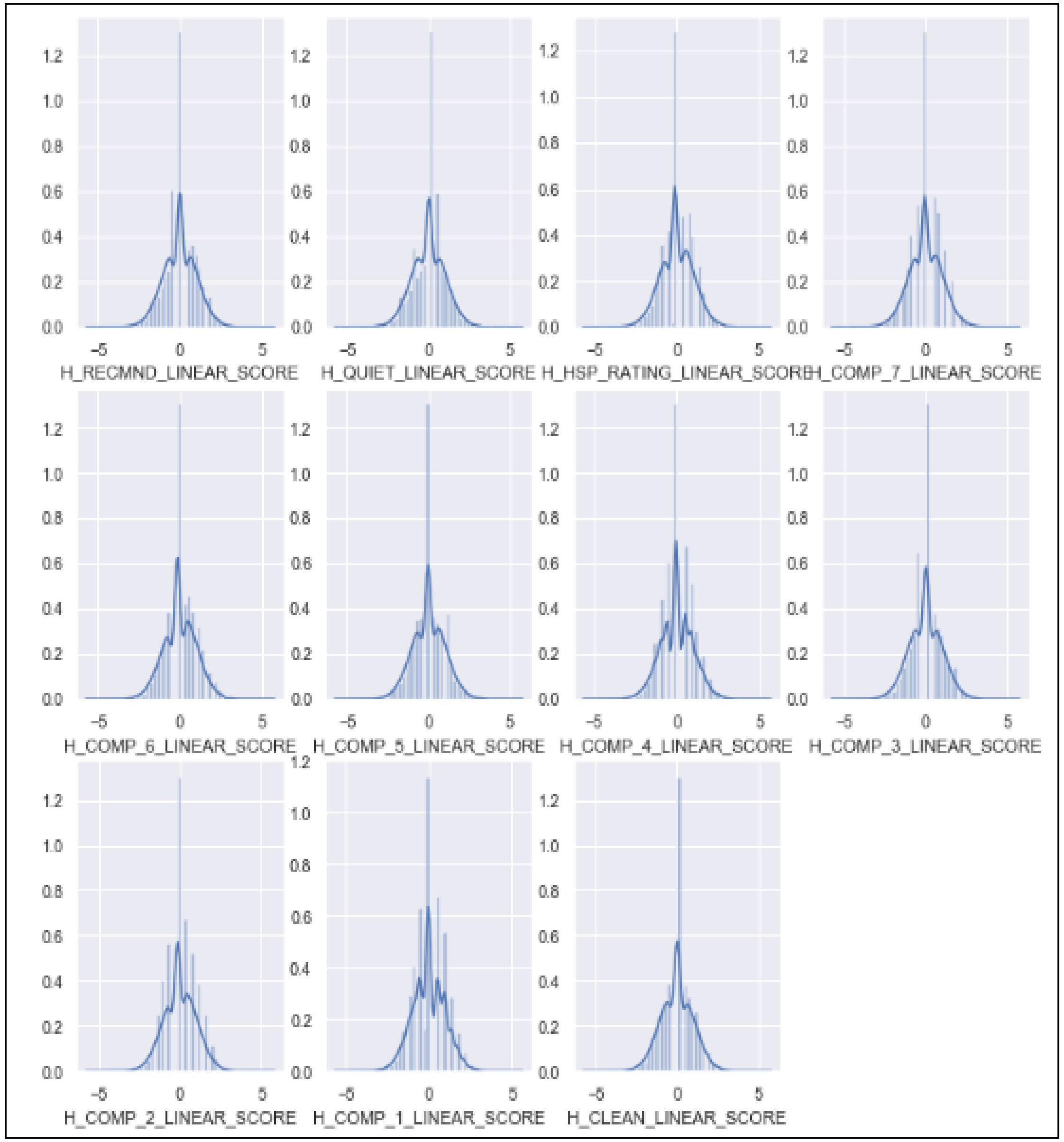
Distribution of Readmission Normalization



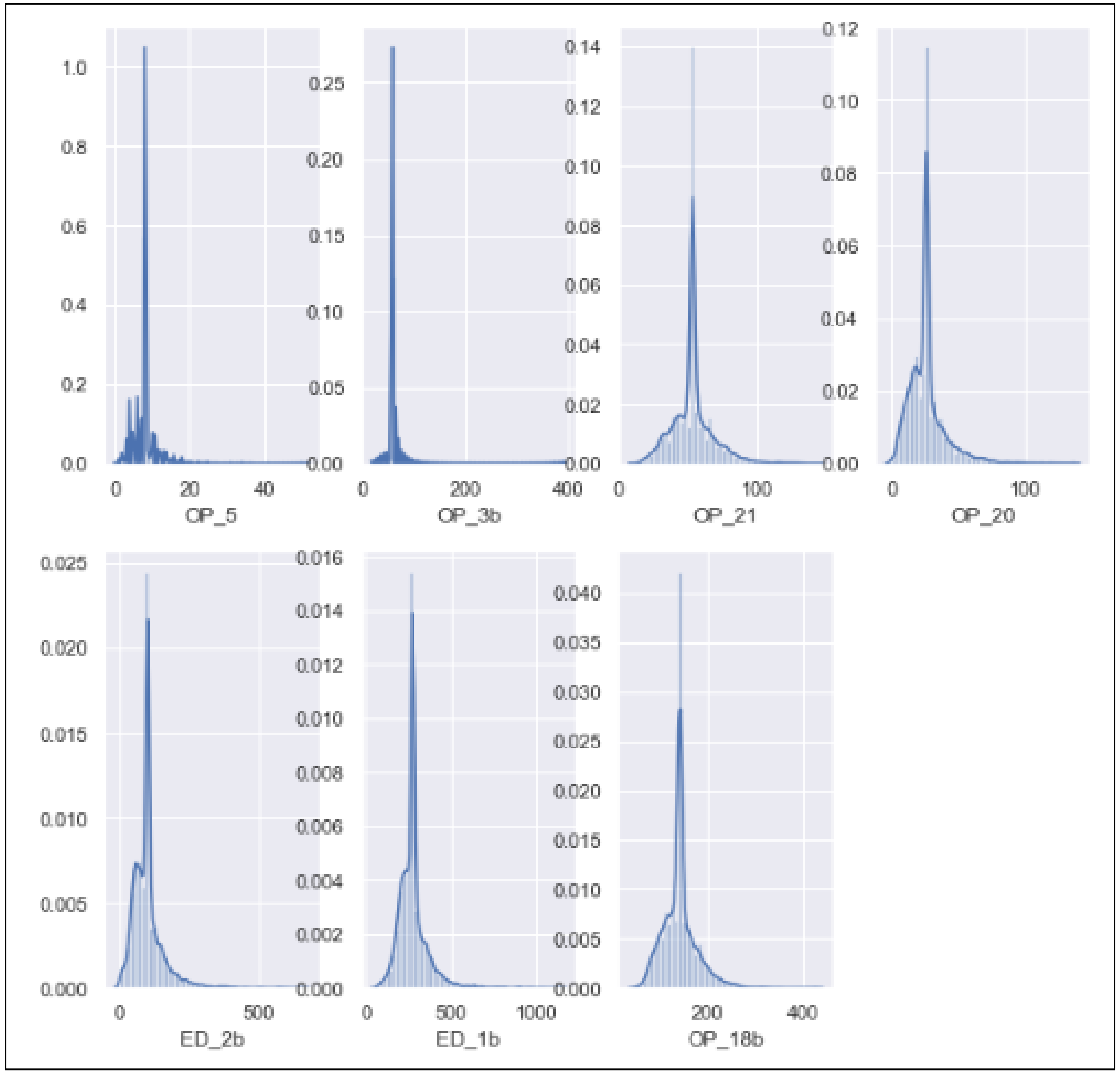


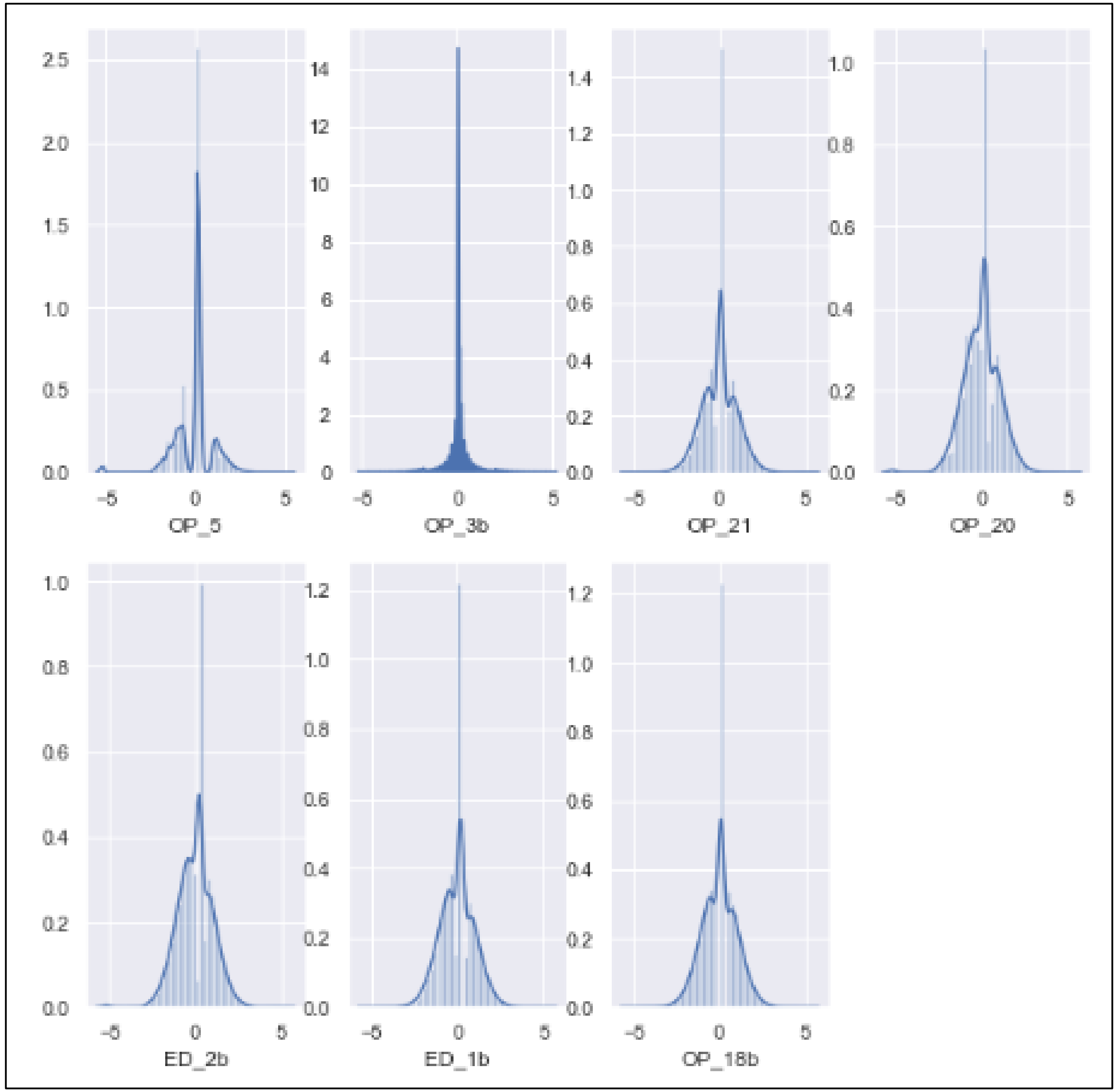
Distribution of Patient Experience Normalization



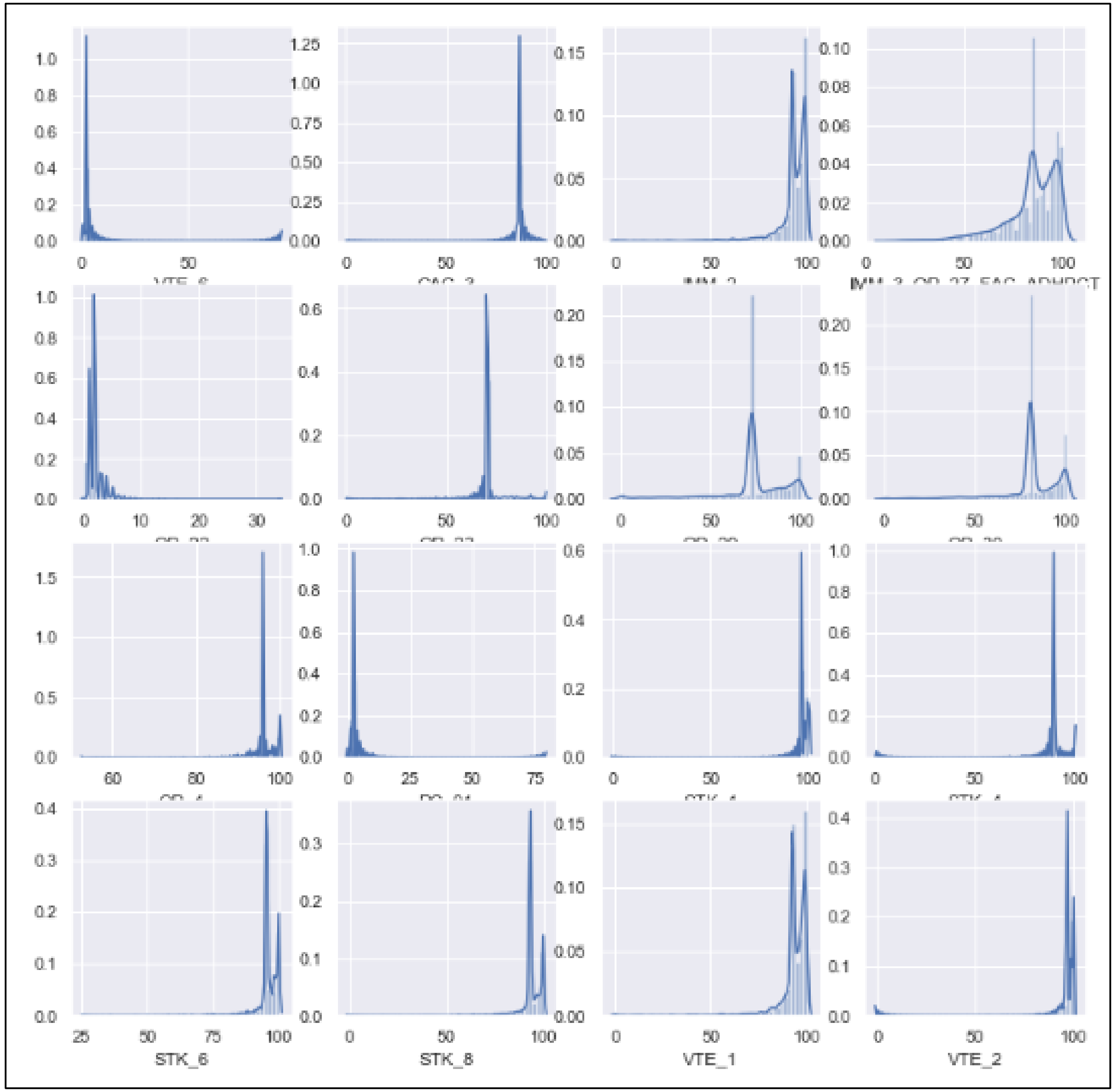


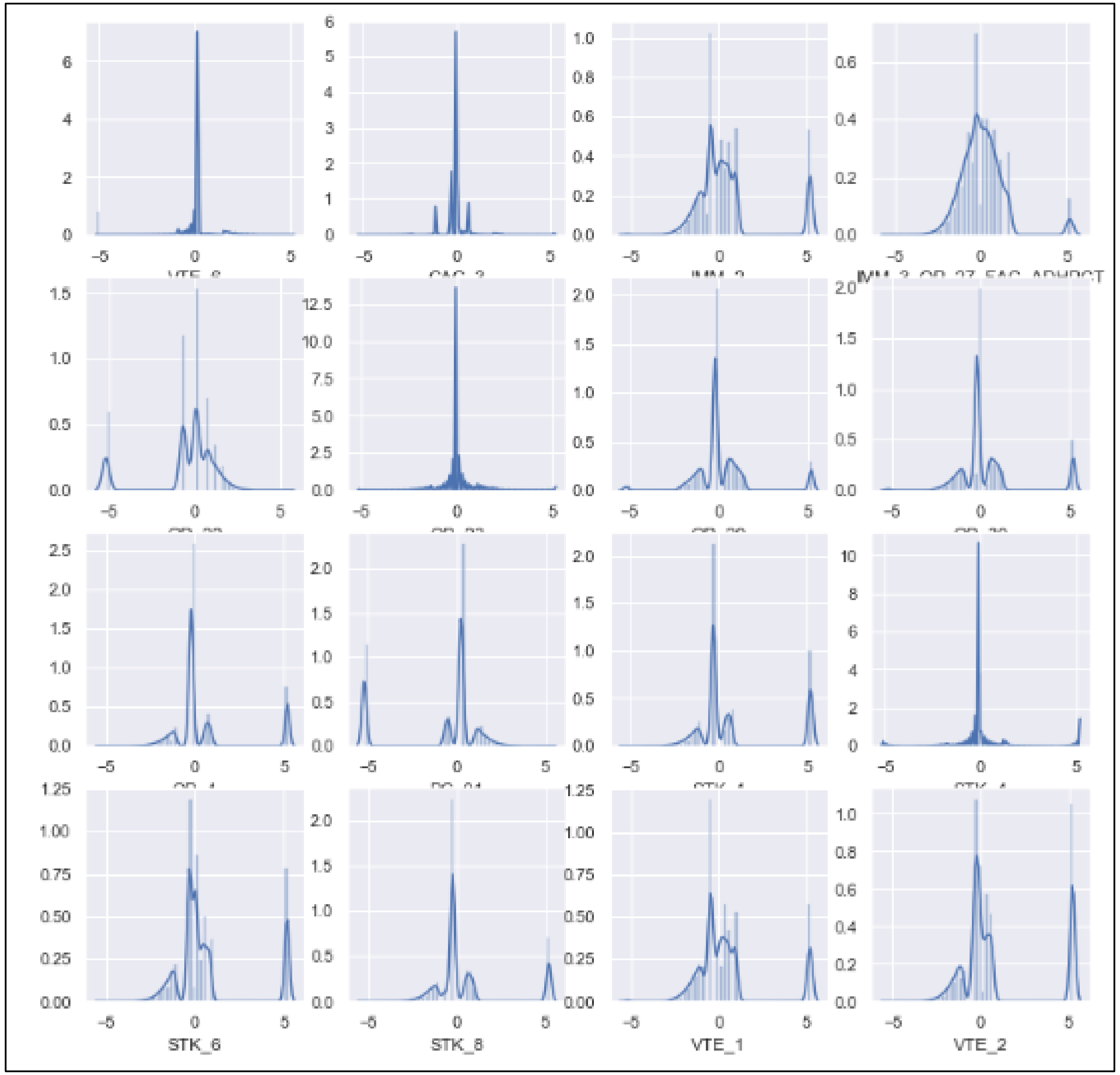
Distribution of Timeliness Normalization



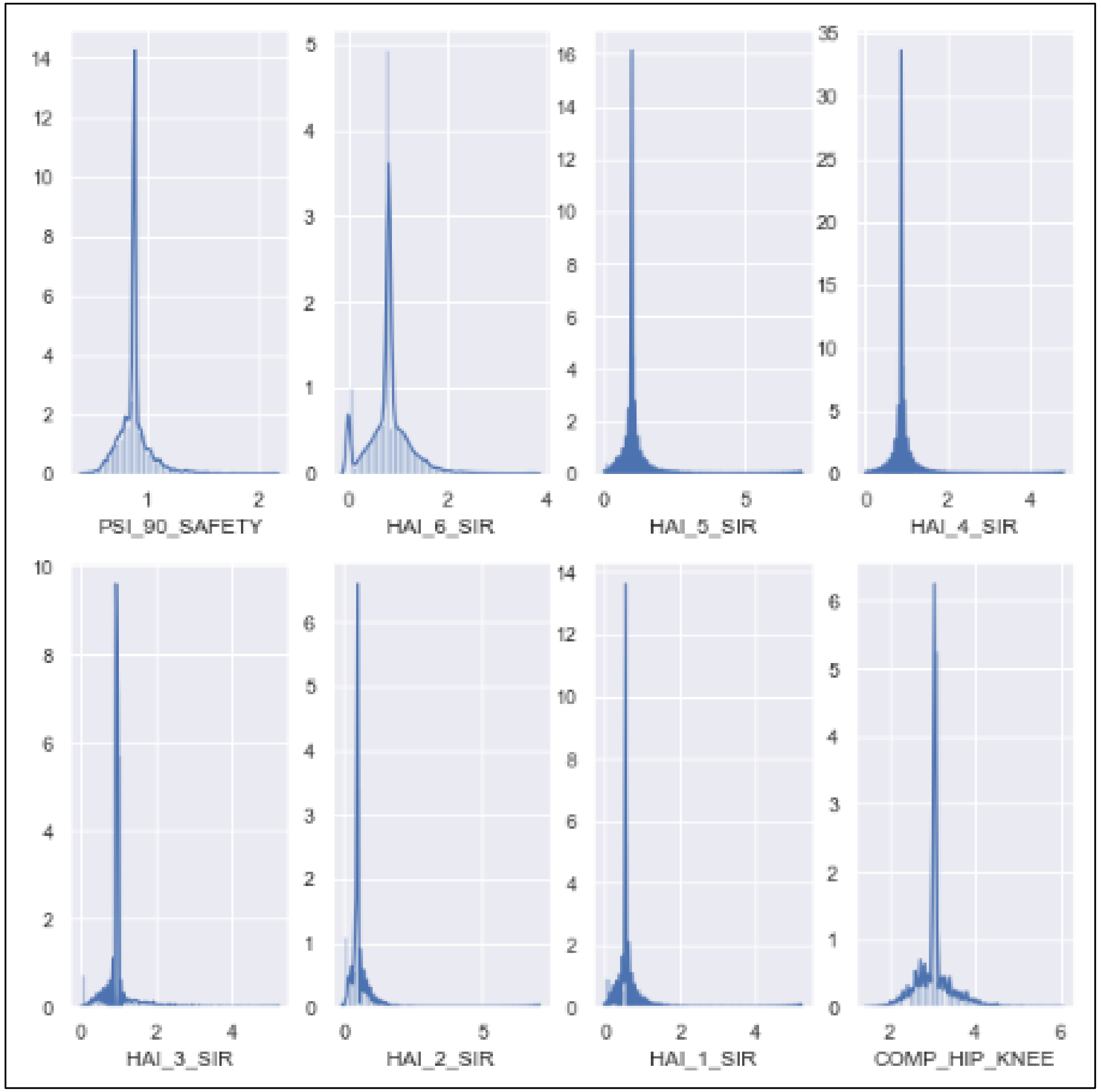


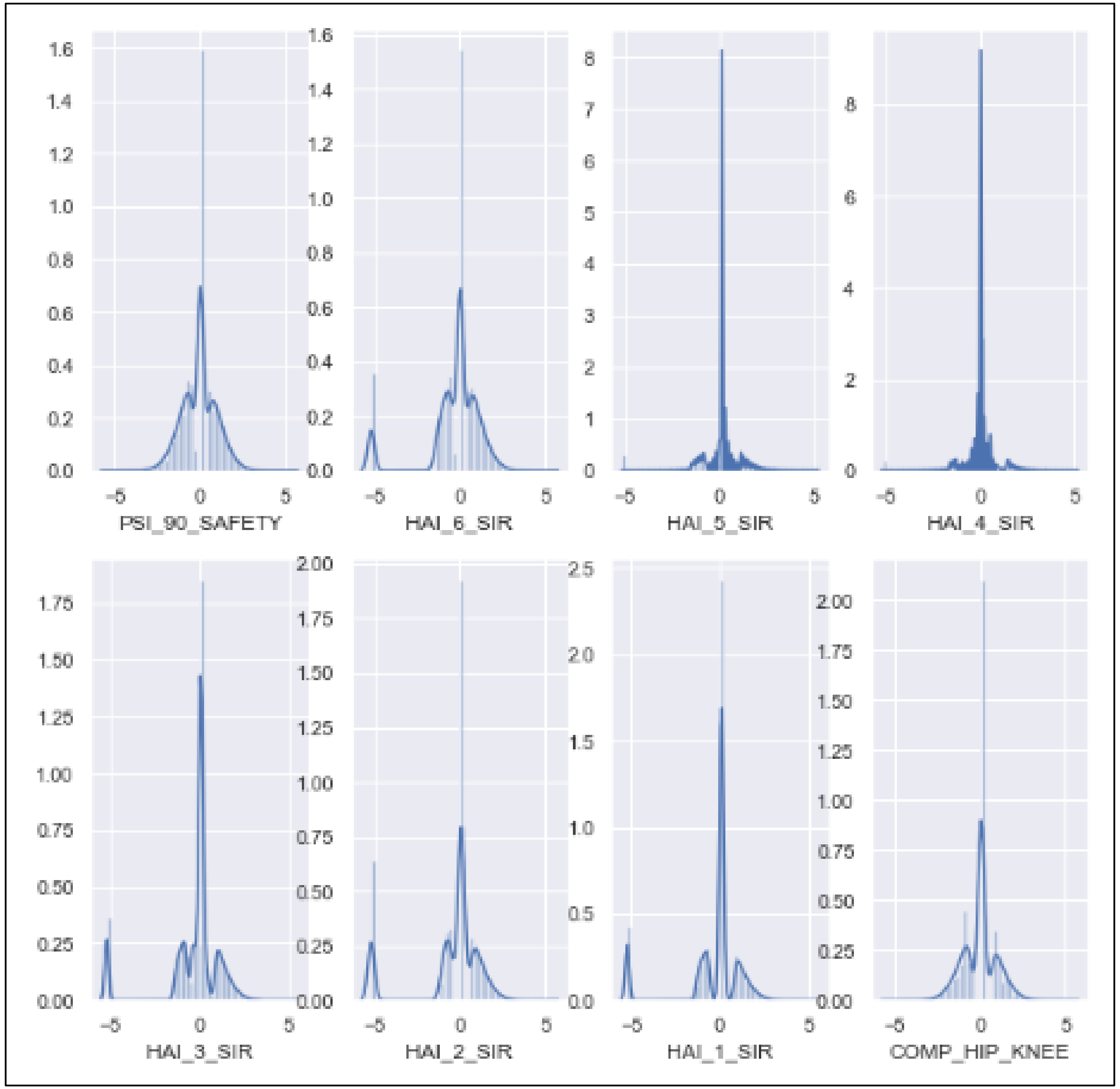
Distribution of Effective care Normalization



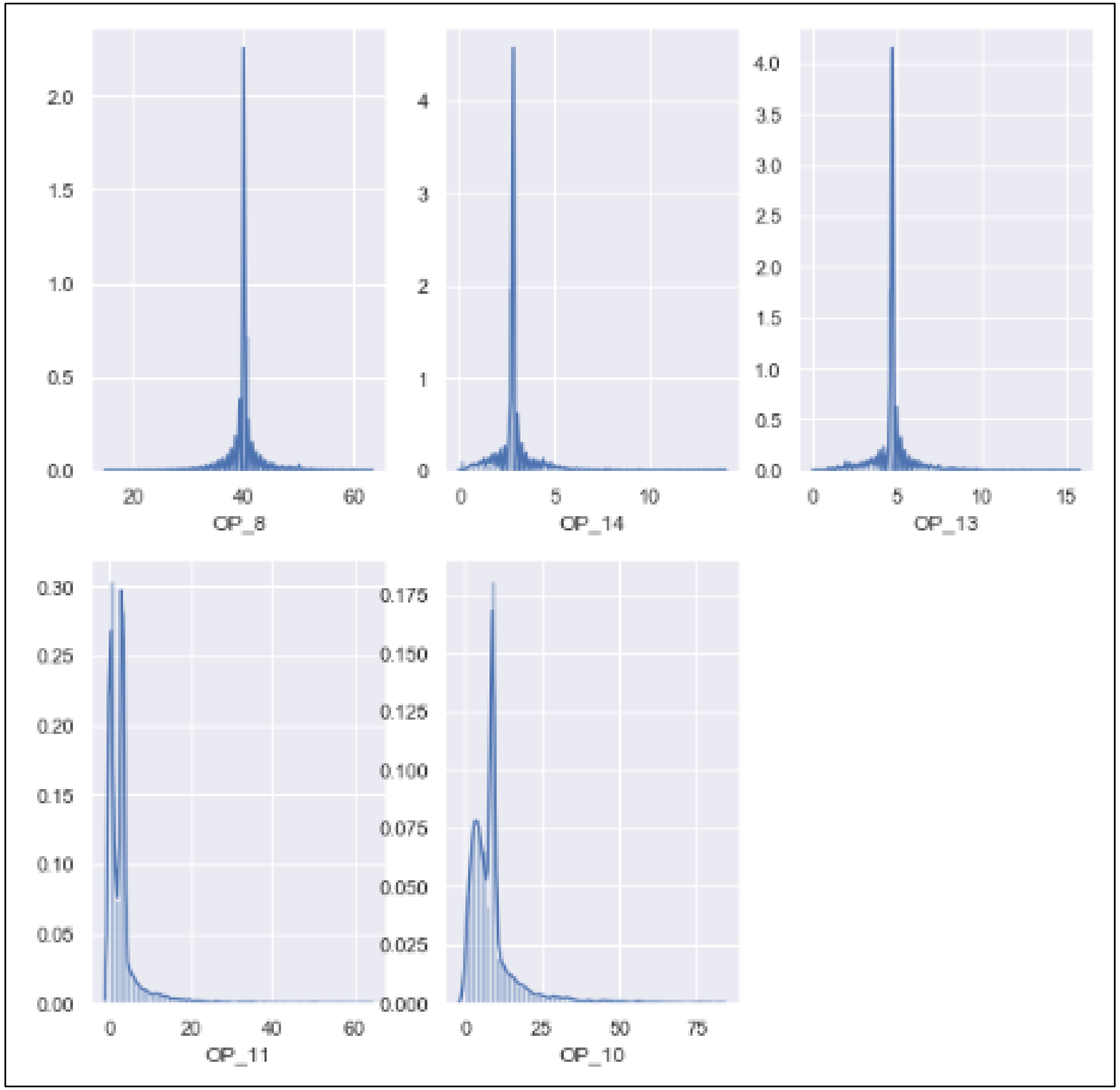


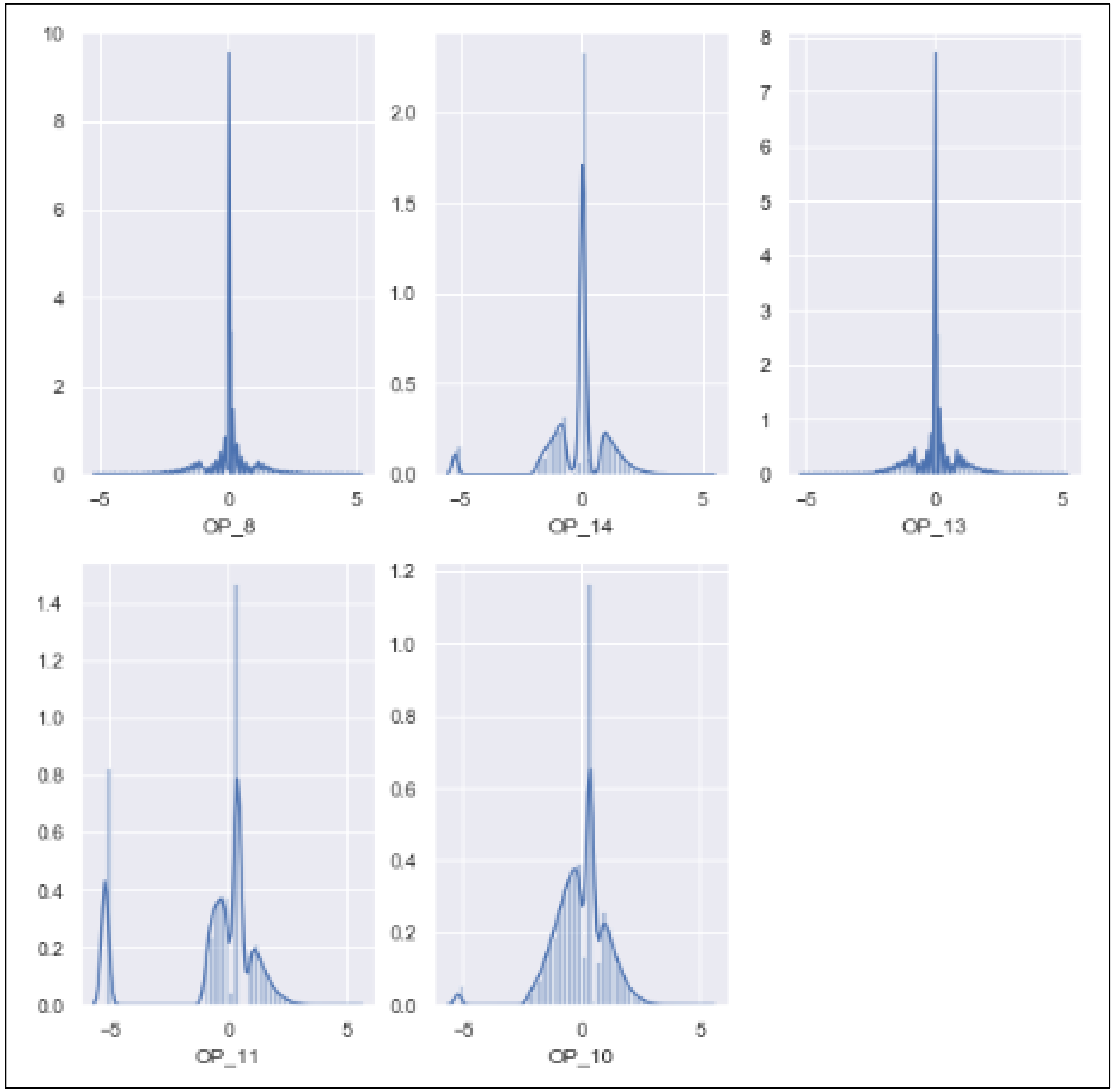
Distribution of Safety Care Normalization



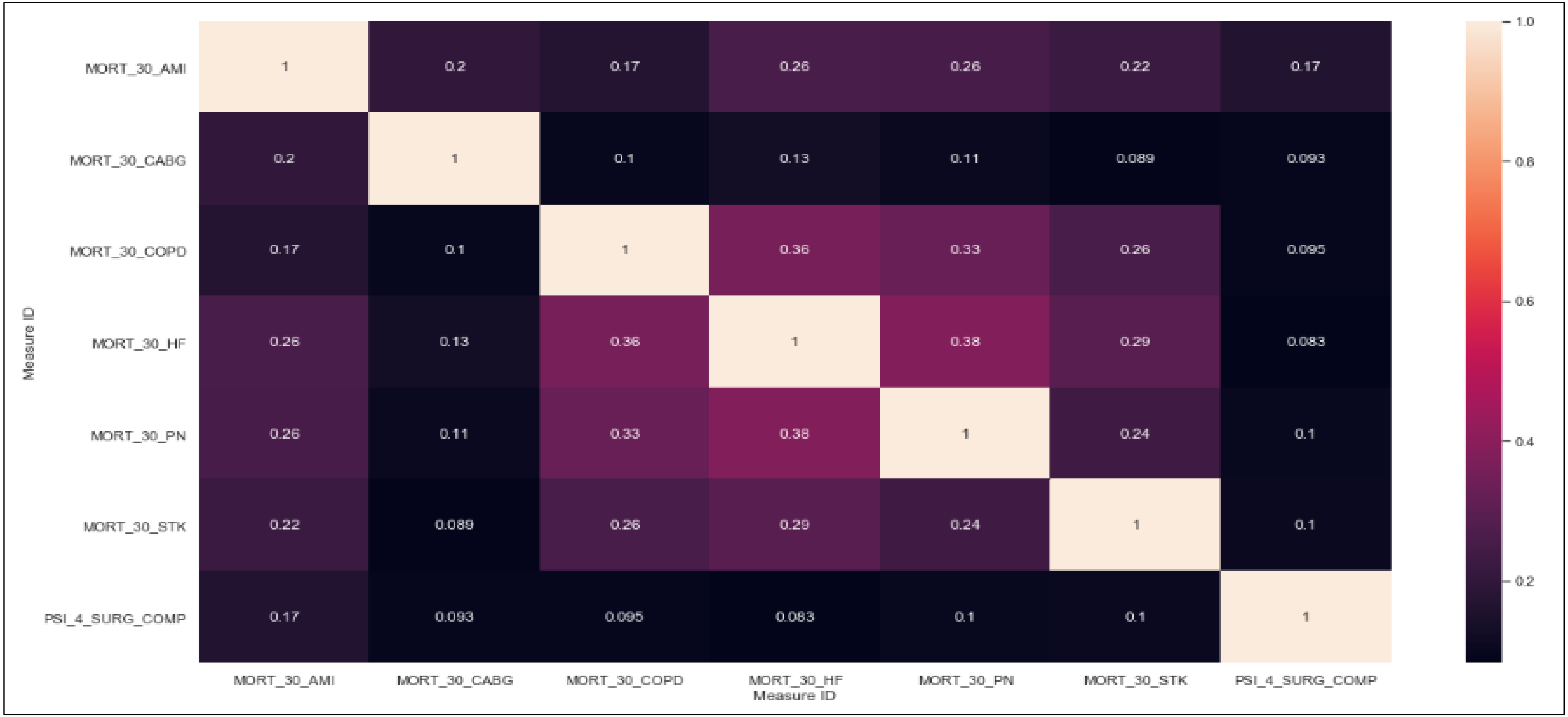


Distribution of Imaging Normalization

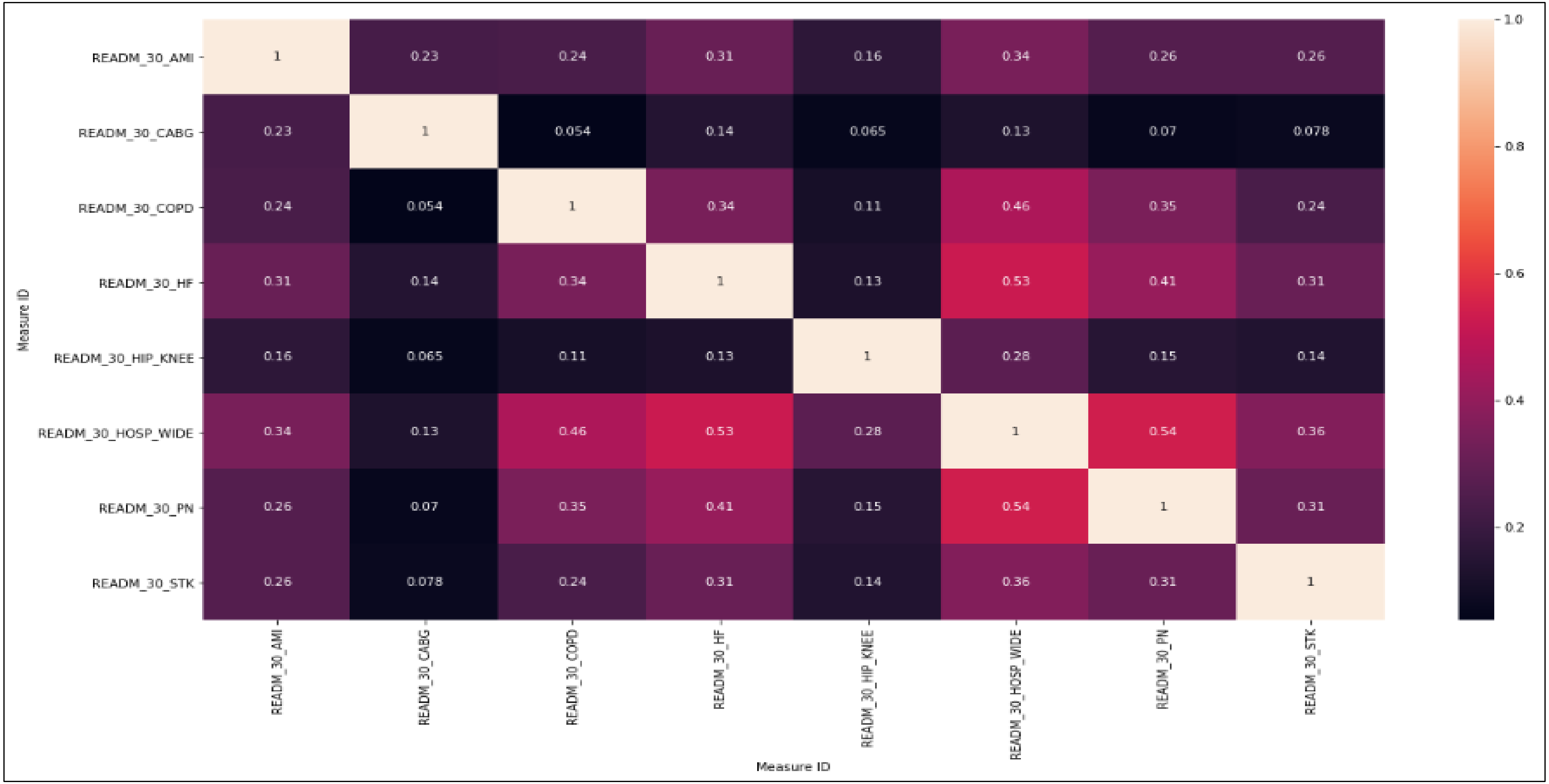




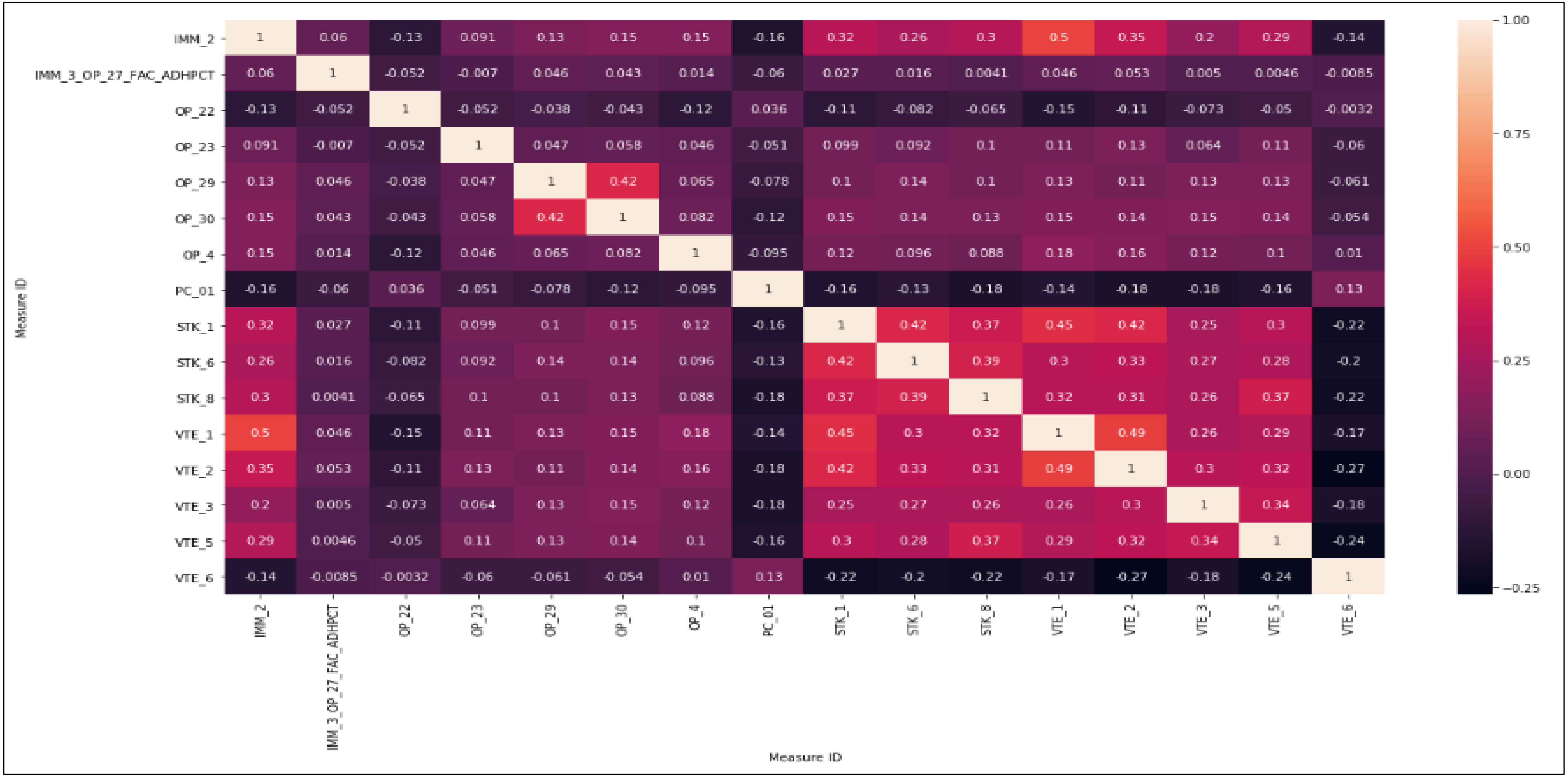
Correlations of Mortality Measure Id's



Correlations of Readmission Measure Id's

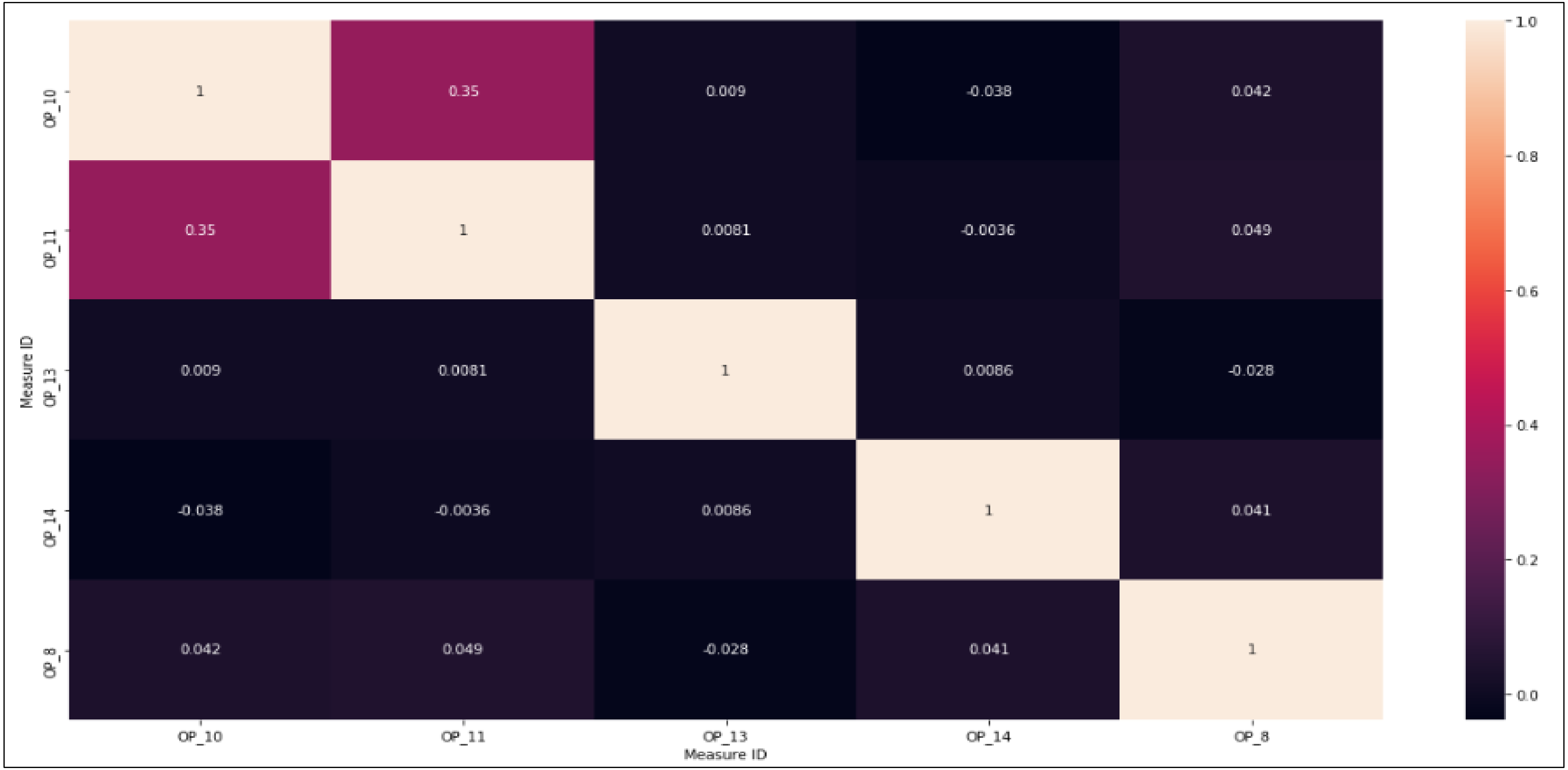


Correlations of Effective care Measure Id’s

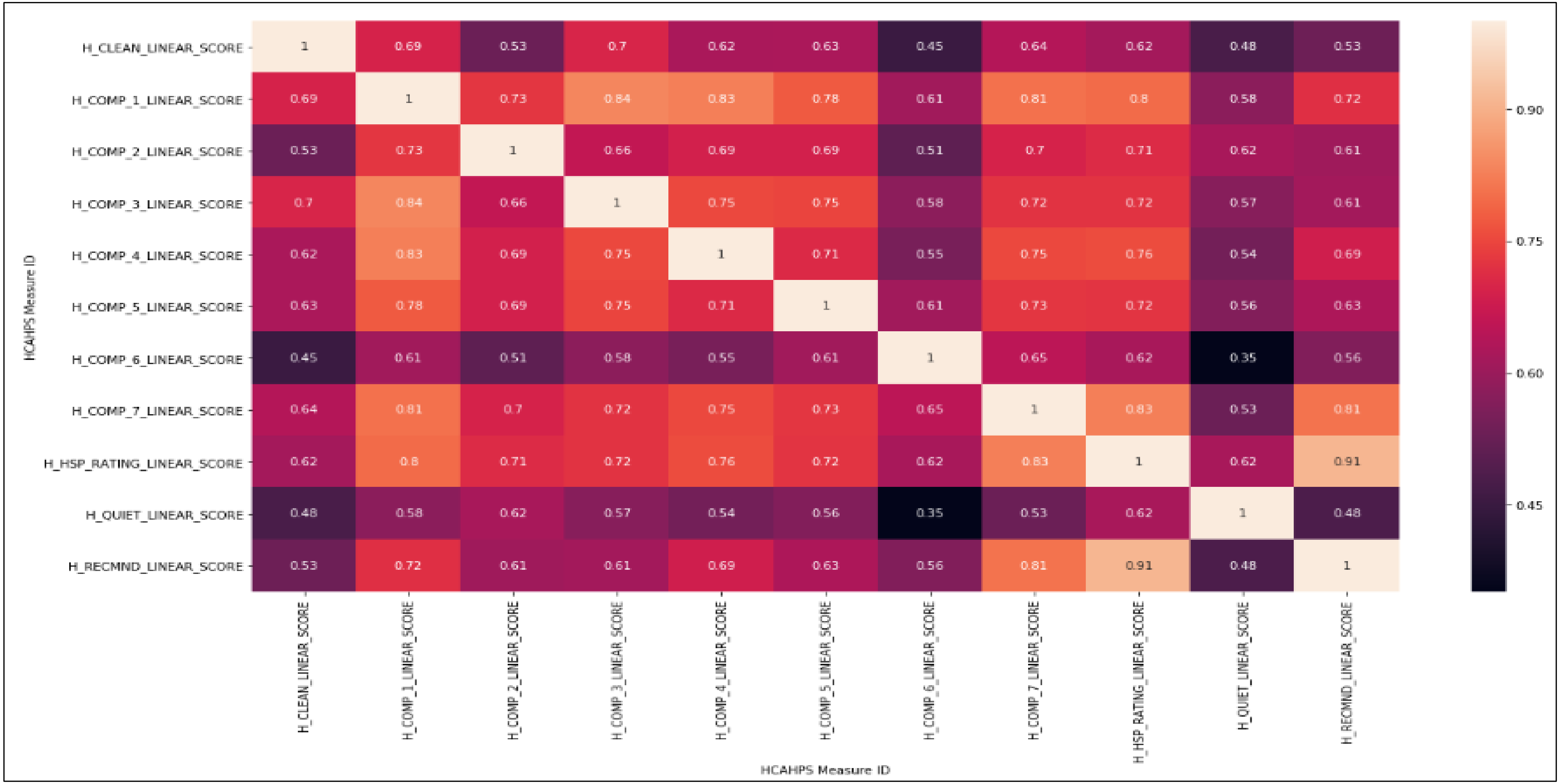


Correlations of Timely care Measure Id’s

Correlations of Imaging Measure Id's



Correlations of Patient Experience Measure Id’s



Modelling Process and Methodology

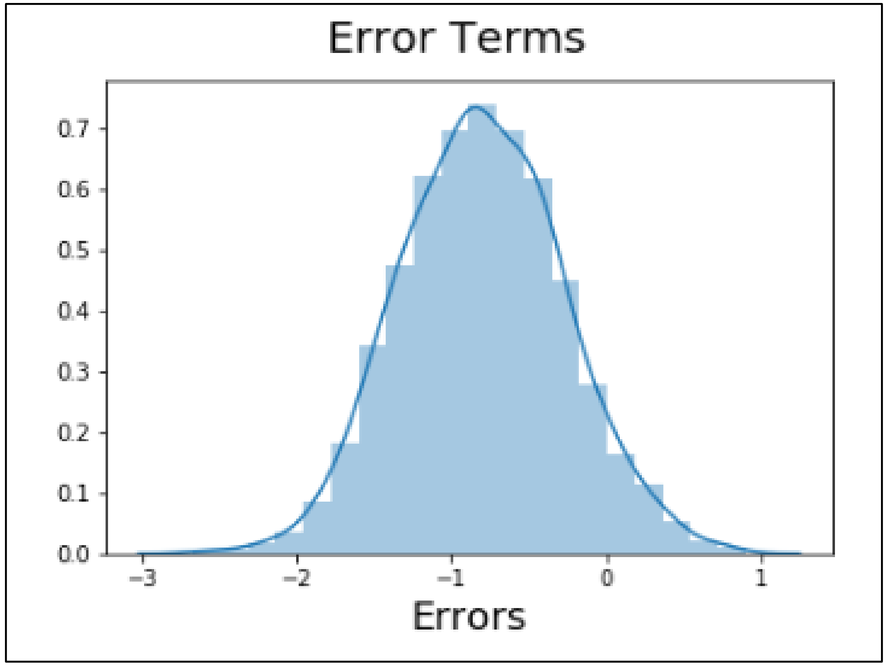
- **Data Preprocessing:**
  - Merging all the pivot table file to **Master Data frame** after dropping highly correlated variables to perform modelling.
  - Defining of **X features (Variables)** and **Y label (Output)** from Master Data frame with shape of (**3648, 57**).
  - Splitting the Master Data Frame in **Train and Test in 70:30** to perform training and evaluation of model.
- **Model Building 1 (Linear Regression):**
  - Using **Recursive feature elimination (RFE)** method that fits a model and removes the weakest feature (or features) until the specified number of features is reached.
  - Using the **statsmodels** performing various iterative model of linear regression till optimum accuracy is reached.
  - Checking for **p-value** and **Variance Inflation Factor (VIF)** for model evaluation.
  - Dropping the variables with **high VIF and p-value** and performing multiple iterations.
  - Evaluating the model on validation dataset and making the prediction and calculating **R- Square score** and **Root Mean Square Error (RMSE)** for the same.
- **Model Building 2 (Random Forest):**
  - Random Forest model building with **balanced class weight** and fitting.
  - Model Evaluation based on **Accuracy Score, Sensitivity, Specificity, False Positive Rate, Positive Predictive Value, Negative Predictive Value and Misclassification Rate.**
  - Hyper-tuning the model based on **max_depth, n_estimators, max_features, min_samples_leaf and min_samples_split** and getting optimal accuracy score.
  - Identifying the **feature importance** from Random Forest model.
- **Model Building 3 (KMeans clustering):**
  - Calculating the **Hopkins statistic** and scaling the data frame with **standard_scaler.fit_transform**.
  - Checking the **silhouette score** to identify the ideal number of clusters.
  - Use of **Factor Analysis** to assign the weights for variables using **Bartlett Test** and **KMO Test.**
  - Created different models with and without **weighted variables**.

Linear Regression Model

```
                            OLS Regression Results
==============================================================================
Dep. Variable:     Hospital overall rating   R-squared:                       0.647
Model:                                 OLS   Adj. R-squared:                  0.645
Method:                      Least Squares   F-statistic:                     309.9
Date:                     Mon, 23 Dec 2019   Prob (F-statistic):               0.00
Time:                             13:21:33   Log-Likelihood:                -1804.9
No. Observations:                     2553   AIC:                             3642.
Df Residuals:                         2537   BIC:                             3735.
Df Model:                               15
Covariance Type:                 nonrobust
==========================================================================================
                              coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------------------
const                      -1.6064      0.143    -11.234      0.000      -1.887      -1.326
READM_30_AMI                0.6378      0.100      6.375      0.000       0.442       0.834
READM_30_COPD               1.1120      0.098     11.341      0.000       0.920       1.304
READM_30_HF                 1.5108      0.103     14.698      0.000       1.309       1.712
READM_30_STK                0.7046      0.100      7.062      0.000       0.509       0.900
OP_10                       0.3924      0.090      4.346      0.000       0.215       0.569
VTE_1                      -0.1818      0.050     -3.608      0.000      -0.281      -0.083
ED_1b                       0.8168      0.088      9.277      0.000       0.644       0.989
MORT_30_AMI                 0.6359      0.097      6.575      0.000       0.446       0.826
MORT_30_COPD                0.8276      0.100      8.304      0.000       0.632       1.023
MORT_30_HF                  1.1045      0.104     10.591      0.000       0.900       1.309
MORT_30_STK                 0.5180      0.099      5.256      0.000       0.325       0.711
PSI_4_SURG_COMP             0.3306      0.098      3.361      0.001       0.138       0.523
COMP_HIP_KNEE               0.5209      0.094      5.557      0.000       0.337       0.705
PSI_90_SAFETY               2.7168      0.092     29.425      0.000       2.536       2.898
H_COMP_5_LINEAR_SCORE      -2.6951      0.084    -32.167      0.000      -2.859      -2.531
==============================================================================
Omnibus:                        3.371   Durbin-Watson:                   2.014
Prob(Omnibus):                  0.185   Jarque-Bera (JB):                3.364
Skew:                          -0.062   Prob(JB):                        0.186
Kurtosis:                       3.127   Cond. No.                         36.1
==============================================================================
```

Linear Regression final model output with **R-square** as **64.7%** and **Adjusted R-square** as **64.5%**

Linear Regression Model



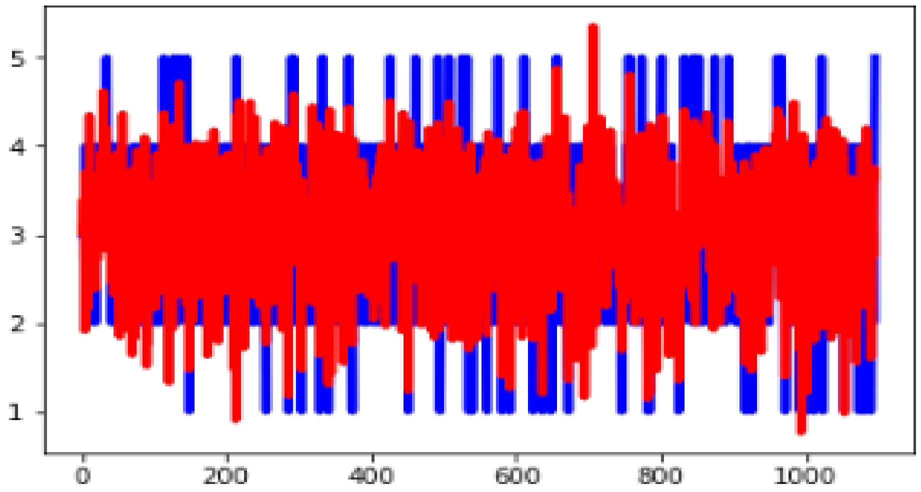
	Features	VIF
9	MORT_30_HF	29.15
2	READM_30_HF	28.35
3	READM_30_STK	27.00
0	READM_30_AMI	26.99
8	MORT_30_COPD	26.78
10	MORT_30_STK	26.03
1	READM_30_COPD	26.01
11	PSI_4_SURG_COMP	25.69
7	MORT_30_AMI	25.23
12	COMP_HIP_KNEE	22.73
13	PSI_90_SAFETY	21.70
4	OP_10	21.48
6	ED_1b	11.87
14	H_COMP_5_LINEAR_SCORE	11.14
5	VTE_1	5.91

Error distribution and **Variance Inflation Factor (VIF)** of Features

Linear Regression Model

```
c = [i for i in range(1,1096,1)]
fig = plt.figure()
plt.plot(c,y_test, color="blue", linewidth=3.5, linestyle="-")     #Plotting Actual
plt.plot(c,y_pred, color="red",  linewidth=3.5, linestyle="-")
```

```
[<matplotlib.lines.Line2D at 0x10689ba8>]
```




```
from sklearn.metrics import r2_score
r2_score(y_test, y_pred)
```

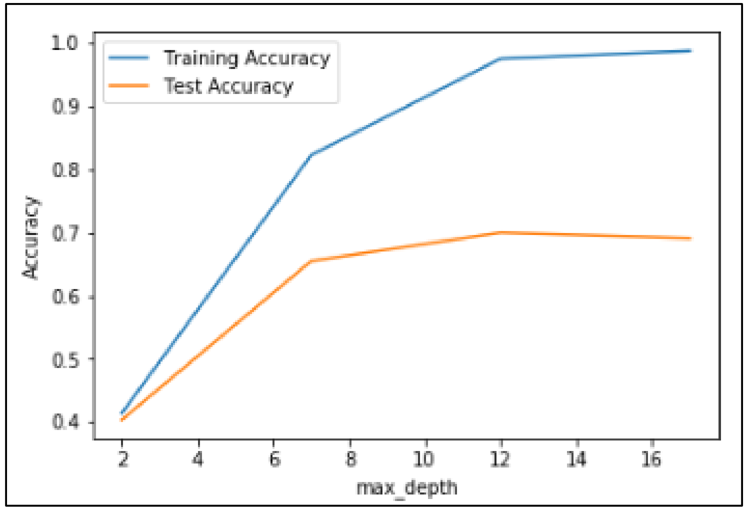
```
0.6794482776221095
```

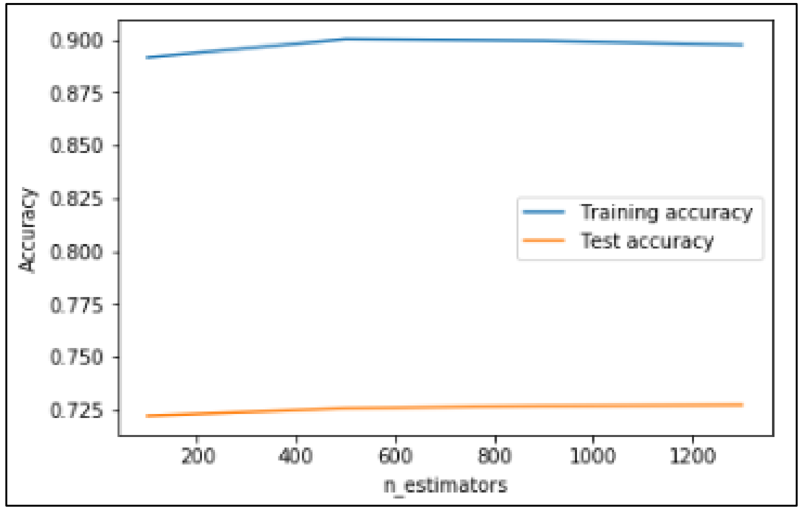
```
from sklearn import metrics
print('RMSE :', np.sqrt(metrics.mean_squared_error(y_test, y_pred)))
```

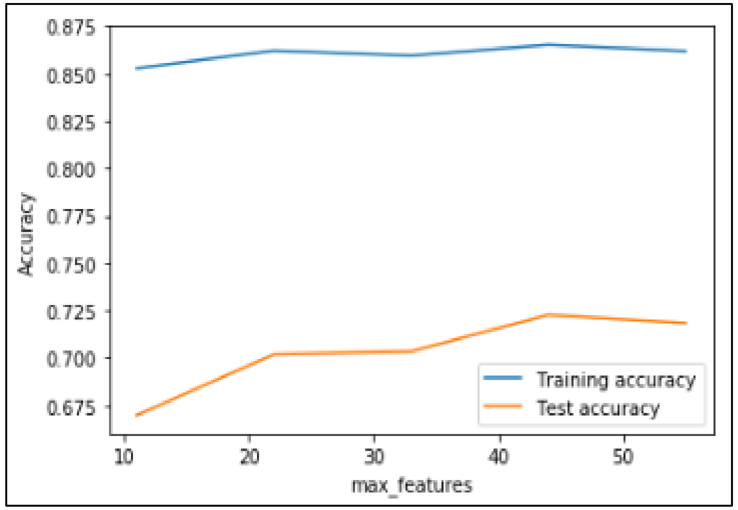
```
RMSE : 0.4829710956249968
```

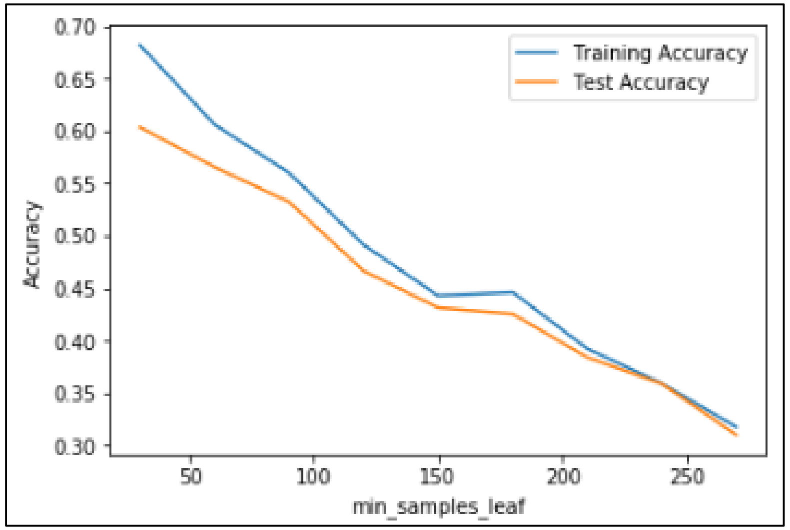
Prediction results with **R-square** and **RMSE value**

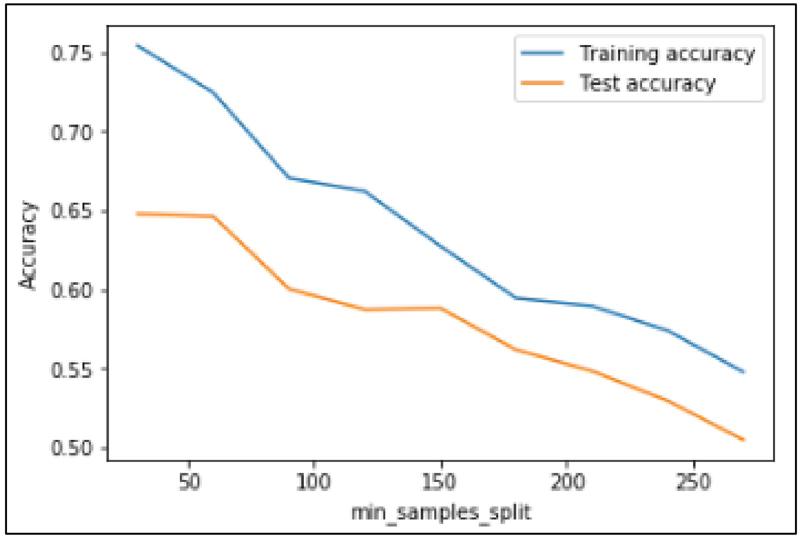
Random Forest Model











Random Forest hyper-tuning parameters **max_depth, n_estimators, max_features, min_samples_leaf and min_samples_split**

Random Forest Model

```
#Printing the optimal accuracy score and hyperparameters
print('The accuracy of',grid_searchRF.best_score_,'with',grid_searchRF.best_params_)
```

```
The accuracy of 0.713670191931O615 with {'max_depth': 10, 'max_features': 20, 'min_samples_leaf': 30, 'min_samples_split': 30,
'n_estimators': 500}
```

```
# Confusion Matrix
confusion_mat_RF_HP=confusion_matrix(y_test,y_pred_RandomForestHP)
confusion_mat_RF_HP
```

```
array([[ 28,   9,   0,   0,   0],
       [ 23, 175,  35,   2,   0],
       [  4,  73, 334, 100,   2],
       [  0,   2,  44, 199,  30],
       [  0,   0,   1,   7,  27]], dtype=int64)
```

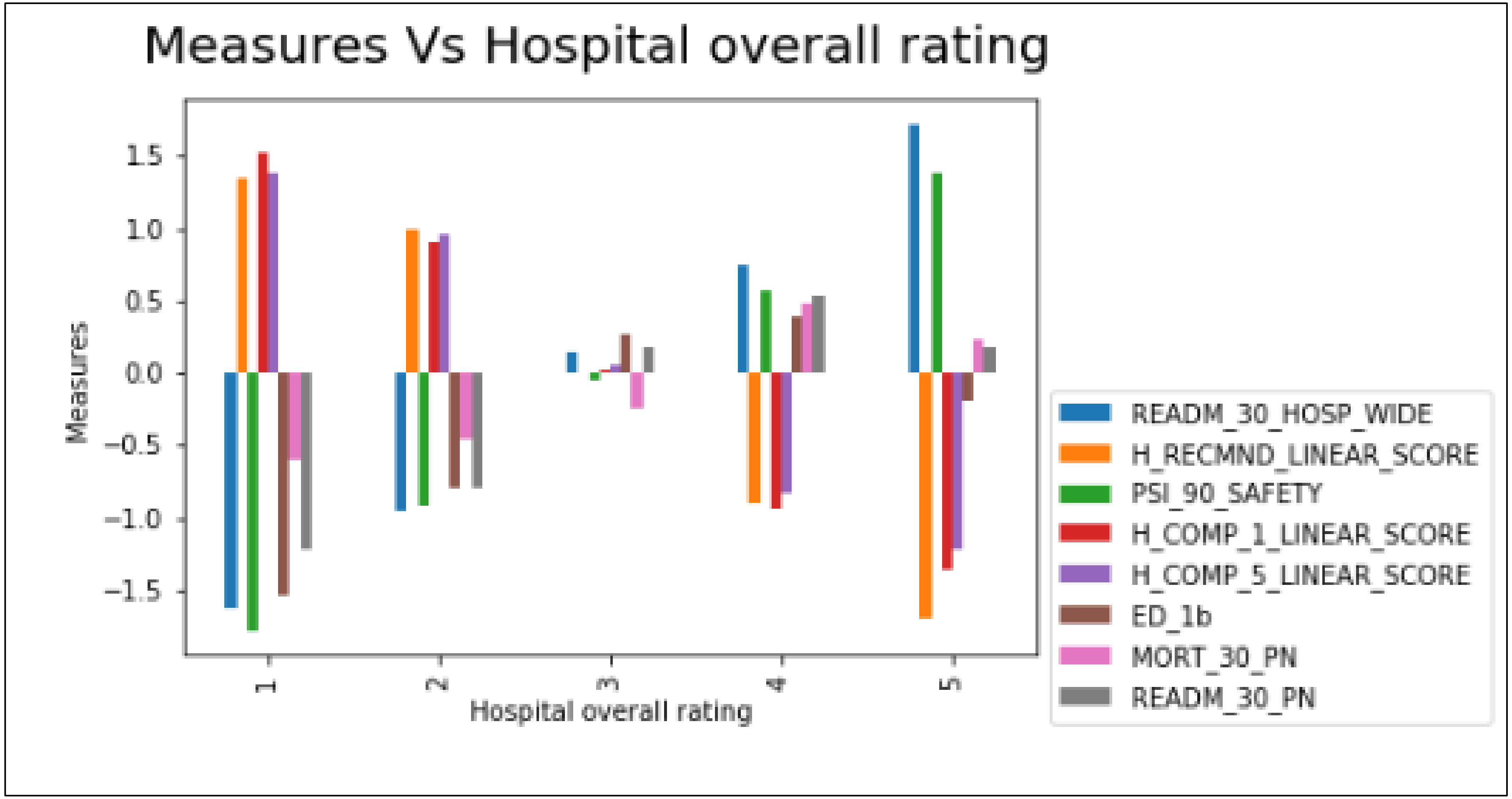
```
TP = confusion_mat_RF_HP[1,1] # true positive
TN = confusion_mat_RF_HP[0,0] # true negatives
FP = confusion_mat_RF_HP[0,1] # false positives
FN = confusion_mat_RF_HP[1,0] # false negatives

print('Accuracy Score on test data: ', accuracy_score(y_test,y_pred_RandomForestHP)*100)   #Accuracy
print('Sensitivity: ', TP / float(TP+FN)*100)  #Sensitivity
print('Specificity: ',TN / float(TN+FP)*100)   #Specificity
print('False Postive Rate: ',FP/ float(TN+FP)*100)  #FPR
print('Positive Predictive Value: ', TP / float(TP+FP)*100)  #PPV
print('Negative Predictive Value: ',TN / float(TN+ FN)*100)  #NPV
print('Misclassification Rate: ',(FN+FP)/(TP+TN+FP+FN)*100)  #Misclassification rate
```

```
Accuracy Score on test data:  69.68036529680364
Sensitivity:  88.38383838383838
Specificity:  75.67567567567568
False Postive Rate:  24.324324324324326
Positive Predictive Value:  95.1086956521739
Negative Predictive Value:  54.90196078431373
Misclassification Rate:  13.617021276595745
```

	Variable	Importance
0	READM_30_HOSP_WIDE	0.276851
1	H_RECMND_LINEAR_SCORE	0.137570
2	PSI_90_SAFETY	0.132495
3	H_COMP_1_LINEAR_SCORE	0.109687
4	H_COMP_5_LINEAR_SCORE	0.041054
5	ED_1b	0.034307
6	MORT_30_PN	0.031073
7	READM_30_PN	0.025867
8	H_CLEAN_LINEAR_SCORE	0.021725
9	MORT_30_HF	0.016775
10	H_COMP_2_LINEAR_SCORE	0.015666
11	READM_30_HF	0.013603
12	H_QUIET_LINEAR_SCORE	0.010463
13	READM_30_COPD	0.009389
14	H_COMP_6_LINEAR_SCORE	0.009112

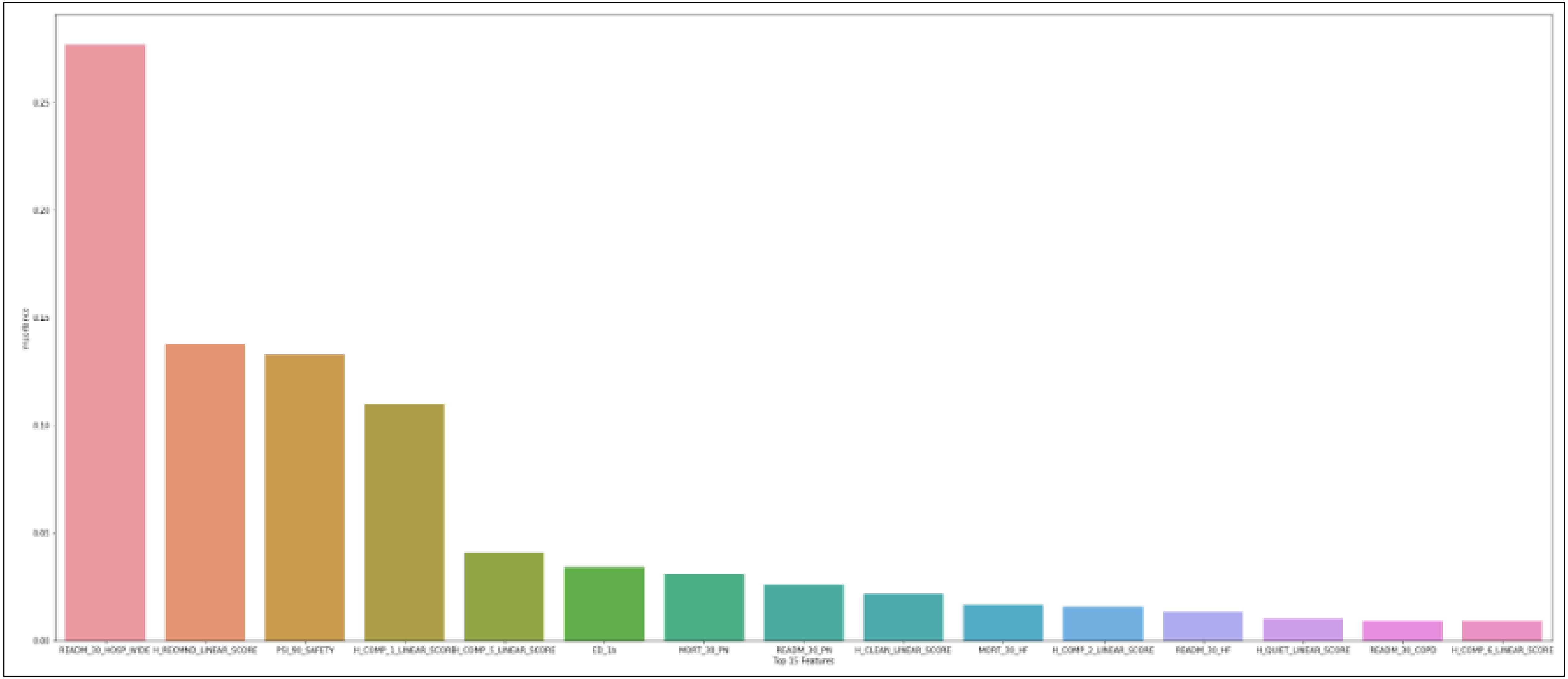
Optimal hyper parameter, confusion matrix and feature selection by random forest.

Random Forest Model



Distribution of important measures from Random forest.

Random Forest Model



Important feature selection by random forest.

K Means Clustering Model

```
# Create factor analysis object and perform factor analysis
fa_mor = FactorAnalyzer()
fa_mor.analyze(Mortality_final,1,rotation=None)
# Check Eigenvalues
ev_mor, v_mor = fa_mor.get_eigenvalues()

fa_mor.loadings
```

	Factor1
MORT_30_AMI	-0.443449
MORT_30_CABG	-0.235795
MORT_30_COPD	-0.529294
MORT_30_HF	-0.633378
MORT_30_PN	-0.578310
MORT_30_STK	-0.458511
PSI_4_SURG_COMP	-0.206983

```
fa_read = FactorAnalyzer()
fa_read.analyze(Readmission_final, 1, rotation=None)
# Check Eigenvalues
ev_read, v_read = fa_read.get_eigenvalues()

fa_read.loadings
```

	Factor1
READM_30_AMI	-0.467700
READM_30_CABG	-0.189586
READM_30_COPD	-0.529539
READM_30_HF	-0.640466
READM_30_HIP_KNEE	-0.276135
READM_30_HOSP_WIDE	-0.824128
READM_30_PN	-0.631045
READM_30_STK	-0.472005

```
fa_img = FactorAnalyzer()
fa_img.analyze(imaging_final,1,rotation=None)
# Check Eigenvalues
ev_img, v_img = fa_img.get_eigenvalues()

fa_img.loadings
```

	Factor1
OP_10	0.635959
OP_11	0.546981
OP_13	0.011016
OP_14	-0.032502
OP_8	0.073549

```
fa_eff = FactorAnalyzer()
fa_eff.analyze(Effective_care_final, 1, rotation=None)
# Check Eigenvalues
ev_eff, v_eff = fa_eff.get_eigenvalues()

fa_eff.loadings
```

	Factor1
IMM_2	-0.555497
IMM_3_OP_27_FAC_ADHPCT	-0.059156
OP_22	0.167919
OP_23	-0.177656
OP_29	-0.247803
OP_30	-0.281945
OP_4	-0.223637
PC_01	0.289350
STK_1	-0.625569
STK_6	-0.549787
STK_8	-0.561562
VTE_1	-0.653347
VTE_2	-0.638255
VTE_3	-0.467893
VTE_5	-0.538655
VTE_6	0.340633

```
fa_time = FactorAnalyzer()
fa_time.analyze(Timely_care_final, 1, rotation=None)
# Check Eigenvalues
ev_time, v_time = fa_time.get_eigenvalues()

fa_time.loadings
```

	Factor1
ED_1b	-0.672006
OP_18b	-0.788145
OP_20	-0.689165
OP_21	-0.604466
OP_5	-0.183985

```
fa_patient = FactorAnalyzer()
fa_patient.analyze(patient_experience_final,1, rotation=None)
# Check Eigenvalues
ev_patient, v_patient = fa_patient.get_eigenvalues()

fa_patient.loadings
```

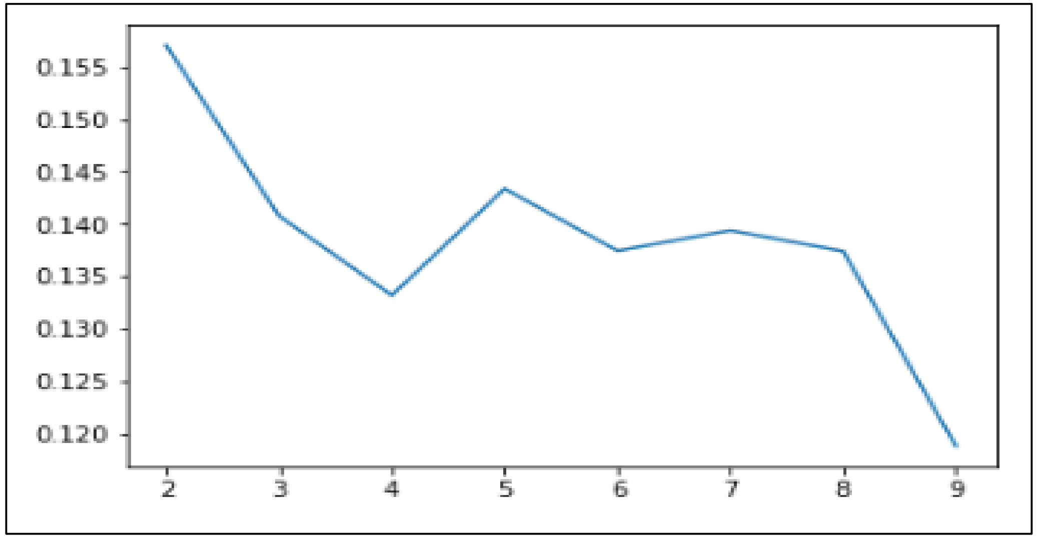
	Factor1
H_CLEAN_LINEAR_SCORE	-0.711852
H_COMP_1_LINEAR_SCORE	-0.920658
H_COMP_2_LINEAR_SCORE	-0.804131
H_COMP_5_LINEAR_SCORE	-0.860563
H_COMP_6_LINEAR_SCORE	-0.660535
H_QUIET_LINEAR_SCORE	-0.653838
H_RECMND_LINEAR_SCORE	-0.765614

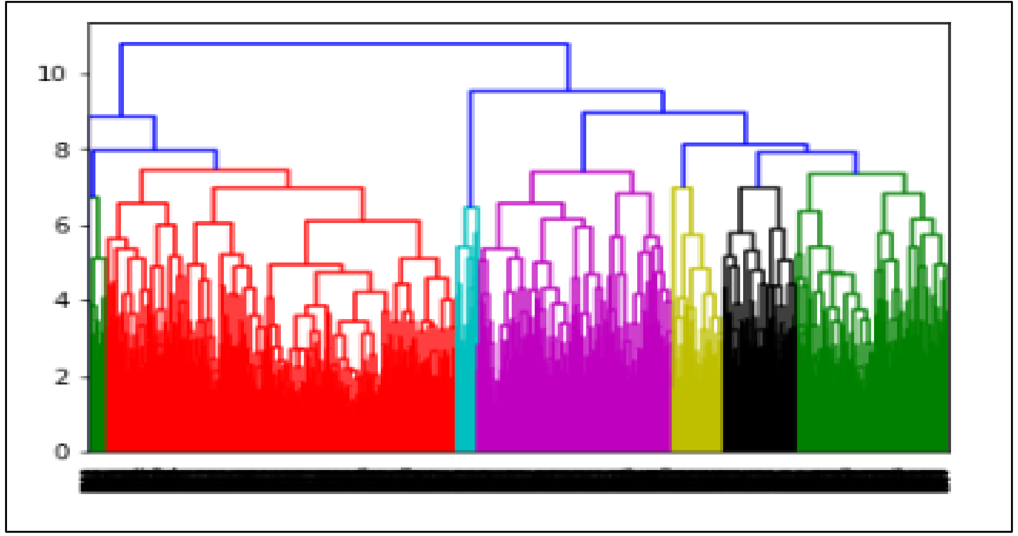
```
fa_safe = FactorAnalyzer()
fa_safe.analyze(Safety_care_final,1,rotation=None)
# Check Eigenvalues
ev_safe, v_safe = fa_safe.get_eigenvalues()

fa_safe.loadings
```

	Factor1
COMP_HIP_KNEE	0.007749
HAI_1_SIR	0.442793
HAI_2_SIR	0.543982
HAI_3_SIR	0.239016
HAI_5_SIR	0.127332
HAI_6_SIR	0.133557
PSI_90_SAFETY	0.212726

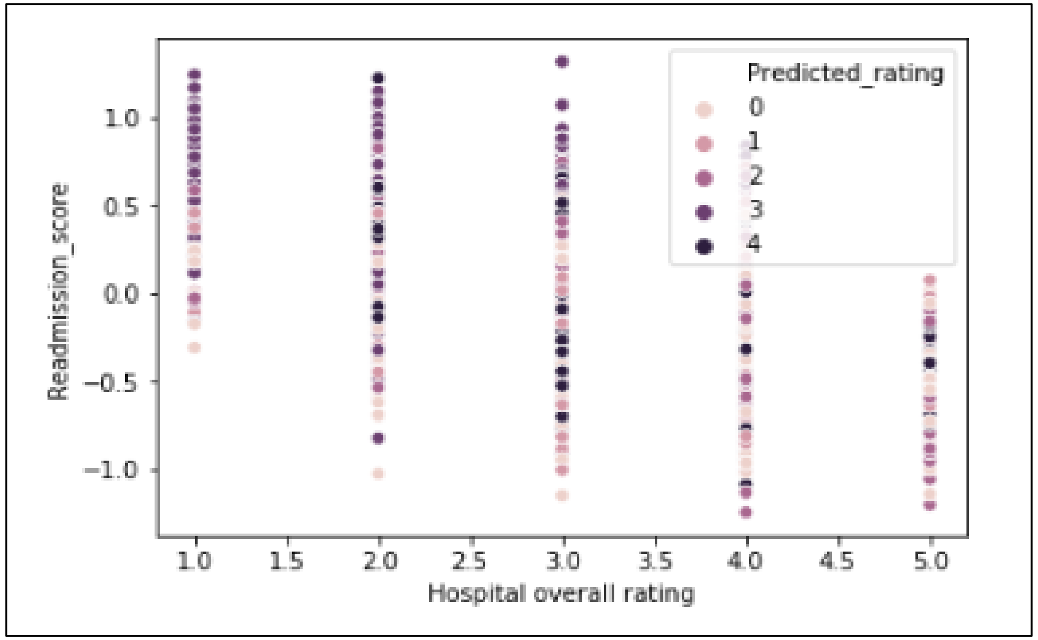
Factor Analysis for clustering.

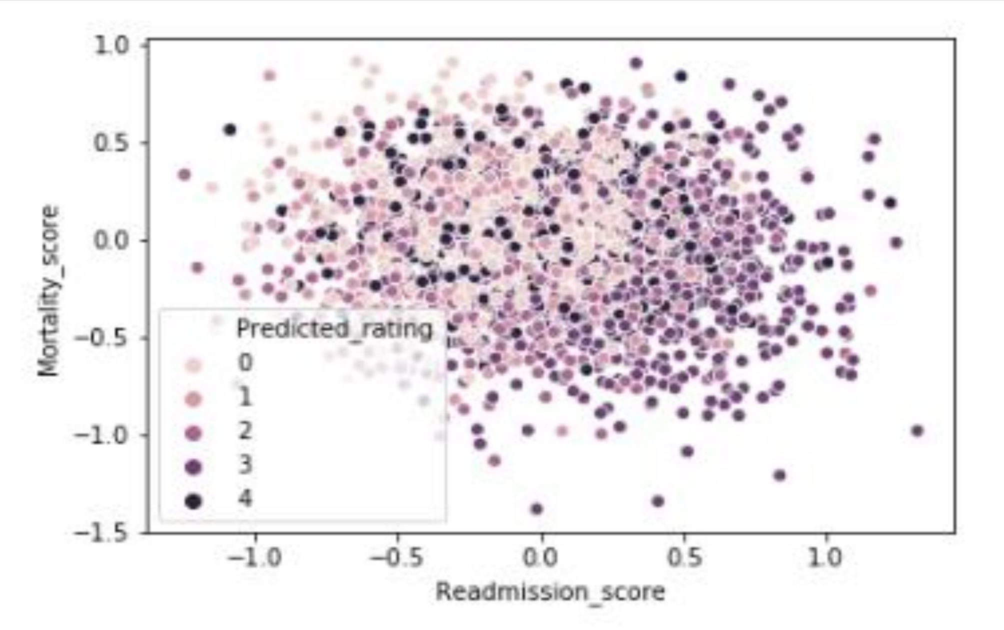
K Means Clustering Model





Error Sum of square, scatterplot of predicted rating, Dendrogram, cluster formation.





Recommendations

- **Recommendations for Hospital EVANSTON HOSPITAL(140010):-**
  - Readmission is lesser than overall average of the group, which means it is performing good in Readmissions.
  - Mortality should be lesser than overall group average and it's the same in our case.
  - Patient experience value should be above the Overall average but in our case it's lesser than the overall average of the group. So need an attention in improving patient experience.
  - Safety of care should be above overall group average for good hospital. In our case it's very less. So it needs an improvement.