IDENTIFY CRITICAL PATIENTS PREDICTION MODEL

PREPARED BY:

- Darshan Jayantbhai Bhatt
- Harsh Chaudhary
- Mohini Pujari
- Pranit Jadhav
- Sujitha L

BUSINESS PROBLEMS

- 1.DEVELOP AND VALIDATE A PREDICTION MODEL
- 2.CONTAINS INFORMATION ABOUT ADMITTED PATEINTS .
- 3.FOR THE VARIOUS PURPOSES SUCH AS LEARNING , RESEARCH AND APPLICATION
- 4. PREDICTING PATEINT CONDITION BASED ON FACTORS RECORDED DURING HOSPITALIZATION . BY ANALYSUIIING DATR WE CAN UNDERSTAND WHERE SPEACIAL ATTENTION IS NEEDED

TECHNOLOGICAL PROBLEM

DEVELOPING A MACHINE LEARNING MODEL TO PREDICT PATEINT SURVIVAL INVOLVES VARIOUS DATA SOURCES , INCLUDING ELECTRONIC HEALTH RECORDS , MEDICAL HISTORY ,DIAGNOSTIC TESTS AND TREATMENT INFORMATION .THE CHALLENGE IS TO PROCESS AND ANALYZE THIS VAST AND COMPLEX DATASET TO BUILD AN ACCURATE PREDICTIVE MODEL

IMPORTANCE

Reduced Healthcare Costs

Medical Research

Reduced Healthcare Costs

Challenges in Healthcare

IImproved Patient Care

VALUE ADDITION

Data-Driven Healthcare

Resource Optimization

Interdisciplinary Collaboration

Continous Monitoring

SUGGESTED SOLUTION

Resource Optimization: Hospitals often face resource constraints, such as a limited number of intensive care unit (ICU) beds or specialized medical staff. Predictive models can help allocate these resources effectively, reducing strain on healthcare facilities.

Data-Driven Healthcare: The healthcare industry is increasingly adopting data-driven approaches to enhance patient care. Developing predictive models for patient conditions is in line with this trend, promoting more efficient and effective healthcare delivery.

Identifying Critical Patients dataset is collected.From hospitals across U.S regarding patients admitted for various reasons.
EDA DONE IS -

```
]: df.shape
]: (91713, 84)
```

Remove id columns and unnamed column

```
df.drop('Unnamed: 83',axis=1,inplace=True)
```

```
dtypes: float64(70), int64(7), object(7)
memory usage: 58.8+ MB
```

```
df.isnull().sum().sum()
196333
```

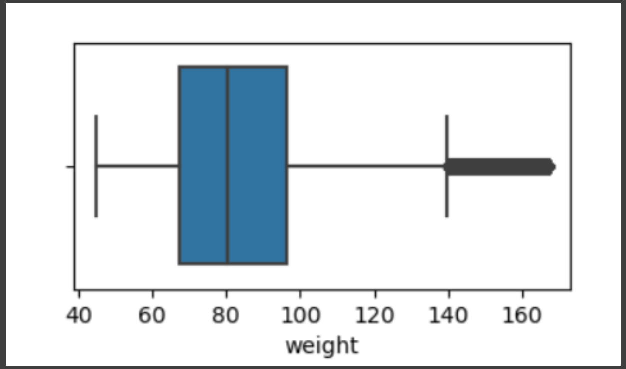
```
]: df_cat=df.select_dtypes(include=object)
df_cat.columns
```

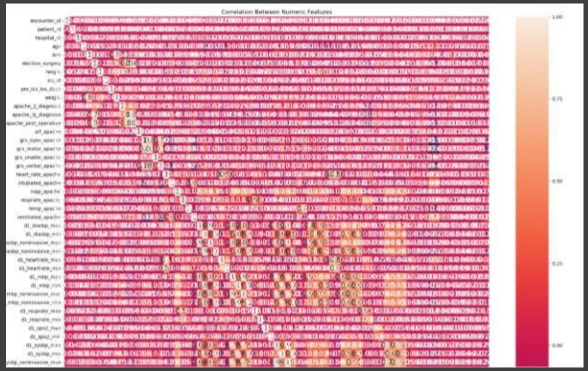
```
df_num_only.fillna(df_num_only.median(),inplace=True,axis=0)
df_num_only.isnull().sum()
```

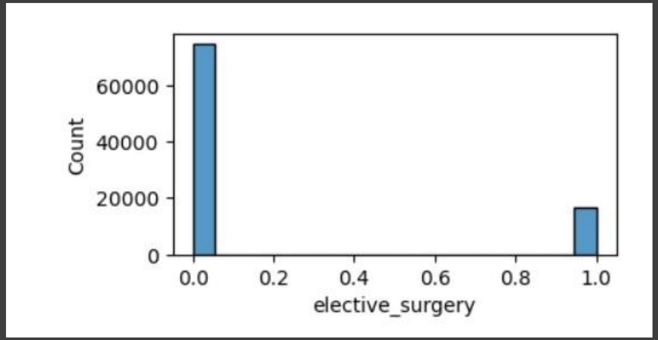
```
: df_num_only.fillna(df_num_only.median(),inplace=True,axis=0)
df_num_only.isnull().sum()
```

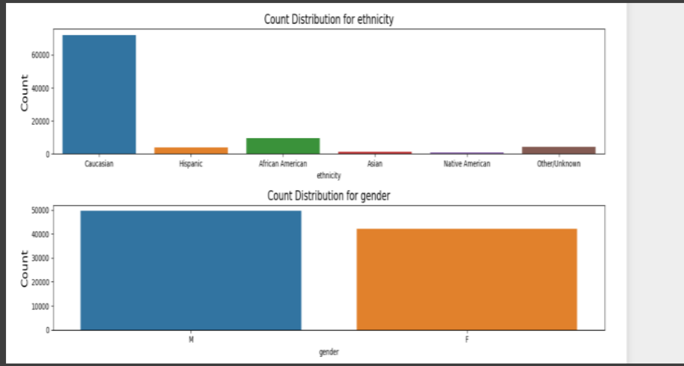
```
df.describe()
```

	encounter_id	patient_id	hospital_id	age	bmi	elective_surgery	height	icu_id	pre_icu_los_days	weight
count	91713.000000	91713.000000	91713.000000	91713.000000	91713.000000	91713.000000	91713.000000	91713.000000	91713.000000	91713.000000
mean	65606.079280	65537.131464	105.669262	62.435881	29.101322	0.183736	169.648256	508.357692	0.841238	83.848824
std	37795.088538	37811.252183	62.854406	16.387168	8.002081	0.387271	10.716719	228.989661	2.481865	24.000953
min	1.000000	1.000000	2.000000	18.000000	14.844926	0.000000	137.200000	82.000000	0.000000	45.000000
25%	32852.000000	32830.000000	47.000000	53.000000	23.787760	0.000000	162.560000	369.000000	0.035417	67.300000
50%	65665.000000	65413.000000	109.000000	65.000000	27.654655	0.000000	170.100000	504.000000	0.138889	80.300000
75%	98342.000000	98298.000000	161.000000	75.000000	32.653061	0.000000	177.800000	679.000000	0.409028	96.300000
max	131051.000000	131051.000000	204.000000	89.000000	63.000000	1.000000	195.590000	927.000000	159.090972	167.000000









- **correlation between numeric columns**
- **heat map of numeric columns**
- Bivariate analysis - boxplot for categorical variables and target column
- Check outliers in the numerical column
- box plot of categorical columns for checking outliers
- Univariate analysis - Histogram plots
- Findings -
- High correlation between noninvasive and invasive measurements of blood pressure. With correlation coefficient bet 0.7 and 0.9.
- eg 1.)d1_diasbp_min and d1_diasbp_max are correlated with d1_diasbp_noninvasive_min and d1_diasbp_noninvasive_max with correlation coefficient as 1.
- 2.) d1_mbp max highly correlated with d1_mbp_noninvasive_max with correlation coefficient as 0.98. Similarly d1_mbp min highly correlated with d1_mbp_noninvasive_min with correlation coefficient as 1.

CHALLENGES

- apache_4a_icu_death_prob, apache_4a_hospital_death_prob, and pre_icu_los_days had negative values values, so we removed those rows.
- Unnamed: 83 columns was a redundant columns with 100% null values, so we dropped that column.
- Out of all numerical columns none of them are normally distributed.
- Many of the columns show huge spike at one place eg age column indicate 65 as highest number of population
- Data imbalance in the dataset

Hypothesis testing

performing chi-contineqency test on age and target variable

```
Chi-square statistic: 1203.131695692268
P-value: 4.3986226670688454e-204
There is  significant association between 'age ' and 'hospital_death'
```

- performing chi-contineqency test on gender and target variable

```
Chi-square statistic: 4.205486236689309
P-value: 0.040293421082557034
There is  significant association between 'gender ' and 'hospital_death'
```

- performing chi-contineqency test on bmi and target variable

```
Chi-square statistic: 35221.10746789586
P-value: 0.10323036630688469
 There is  significant association between 'bmi ' and 'hospital_death'
```

performing chi-contineqency test on ethinicity and all the disease variable

```
Chi-square statistic for 'ethnicity' and 'aids': 57.197820387430525
P-value: 4.603756741047879e-11
There is significant association between 'ethnicity' and 'aids'
========================================

Chi-square statistic for 'ethnicity' and 'cirrhosis': 286.76088200110155
P-value: 7.019396933504672e-60
There is significant association between 'ethnicity' and 'cirrhosis'
========================================

Chi-square statistic for 'ethnicity' and 'diabetes_mellitus': 244.8298824906402
P-value: 7.067935701744601e-51
There is significant association between 'ethnicity' and 'diabetes_mellitus'
========================================

Chi-square statistic for 'ethnicity' and 'hepatic_failure': 234.32186334813406
P-value: 1.2668936853455252e-48
There is significant association between 'ethnicity' and 'hepatic_failure'
========================================

Chi-square statistic for 'ethnicity' and 'immunosuppression': 12.25661629339409
P-value: 0.03143580136822404
There is significant association between 'ethnicity' and 'immunosuppression'
========================================

Chi-square statistic for 'ethnicity' and 'leukemia': 6.4632513452456175
P-value: 0.263714113944348
There is no significant association between 'ethnicity' and 'leukemia'
Chi-square statistic for 'ethnicity' and 'lymphoma': 17.40054942884735
P-value: 0.0037995751257215216
There is significant association between 'ethnicity' and 'lymphoma'
========================================

Chi-square statistic for 'ethnicity' and 'solid_tumor_with_metastasis': 24.10924158091857
P-value: 0.00020686537511766326
There is significant association between 'ethnicity' and 'solid_tumor_with_metastasis'
========================================
```

Algorithms

- Logisitic regression -
- Naïve bais -
- K – nearest neighbour

```
y_predict = model.predict(X_test)
print(metrics.classification_report(y_test, y_predict)) #logistic regression
```

	precision	recall	f1-score	support
0	0.92	1.00	0.96	32373
1	0.62	0.05	0.09	3067
accuracy			0.92	35440
macro avg	0.77	0.52	0.52	35440
weighted avg	0.89	0.92	0.88	35440

class 1 has low scores because of imbalanced data.

```
print(metrics.classification_report(expected, predicted)) #naive bais
```

	precision	recall	f1-score	support
0	0.95	0.88	0.92	32373
1	0.30	0.52	0.38	3067
accuracy			0.85	35440
macro avg	0.63	0.70	0.65	35440
weighted avg	0.90	0.85	0.87	35440

```
In [301]: # summarize the fit of the model
          print(metrics.classification_report(y_test, predicted_labels)) #k nearest neighbour
```

	precision	recall	f1-score	support
0	0.91	0.98	0.94	32373
1	0.10	0.03	0.05	3067
accuracy			0.89	35440
macro avg	0.51	0.50	0.49	35440
weighted avg	0.84	0.89	0.87	35440

Algorithms with scaling

- K- NEAREST NEIGBOUR WITH SCALING
- DECISION TREE
- RANDOM FOREST
- ENSEMBLE
- BAGGING

```
]: # summarize the fit of the model
print(metrics.classification_report(y_test, predicted_labels)) #k nearest ne
```

	precision	recall	f1-score	support
0	0.93	0.99	0.96	32373
1	0.55	0.19	0.28	3067
accuracy			0.92	35440
macro avg	0.74	0.59	0.62	35440
weighted avg	0.90	0.92	0.90	35440

```
In [311]: print(metrics.classification_report(y_test,y_pred)) # decision tree
```

	precision	recall	f1-score	support
0	0.93	0.99	0.96	24251
1	0.69	0.18	0.29	2329
accuracy			0.92	26580
macro avg	0.81	0.59	0.62	26580
weighted avg	0.91	0.92	0.90	26580

	precision	recall	f1-score	support
0	0.93	0.99	0.96	24251
1	0.71	0.23	0.35	2329
accuracy			0.92	26580
macro avg	0.82	0.61	0.66	26580
weighted avg	0.91	0.92	0.91	26580

	precision	recall	f1-score	support
0	0.93	0.99	0.96	24251
1	0.72	0.17	0.27	2329
accuracy			0.92	26580
macro avg	0.82	0.58	0.62	26580
weighted avg	0.91	0.92	0.90	26580

	precision	recall	f1-score	support
0	0.93	0.99	0.96	24251
1	0.71	0.19	0.30	2329
accuracy			0.92	26580
macro avg	0.82	0.59	0.63	26580
weighted avg	0.91	0.92	0.90	26580

	precision	recall	f1-score	support
0	0.93	0.99	0.96	24251
1	0.69	0.18	0.29	2329
accuracy			0.92	26580
macro avg	0.81	0.59	0.62	26580
weighted avg	0.91	0.92	0.90	26580

	precision	recall	f1-score	support
0	0.91	0.98	0.94	24251
1	0.09	0.03	0.04	2329

BOOSTING

In [330]:
```
print(metrics.classification_report(y_train,y_trained_pred)) #adaboosting
```

	precision	recall	f1-score	support
0	0.94	0.99	0.96	56692
1	0.67	0.30	0.42	5326
accuracy			0.93	62018
macro avg	0.80	0.64	0.69	62018
weighted avg	0.91	0.93	0.91	62018

In [331]:
```
y_test_pred=adaboost_model.predict(X_test)
```

In [332]:
```
print(metrics.classification_report(y_test,y_test_pred)) #adaboosting
```

	precision	recall	f1-score	support
0	0.94	0.99	0.96	24251
1	0.66	0.29	0.40	2329
accuracy			0.92	26580
macro avg	0.80	0.64	0.68	26580
weighted avg	0.91	0.92	0.91	26580

In [321]:
```
print(metrics.classification_report(y_test,y_pred)) #gradient boosting test
```

	precision	recall	f1-score	support
0	0.94	0.99	0.96	24251
1	0.68	0.29	0.41	2329
accuracy			0.93	26580
macro avg	0.81	0.64	0.68	26580
weighted avg	0.91	0.93	0.91	26580

In [322]:
```
y_train_pred=gb_model.predict(X_train)
```

In [326]:
```
print(metrics.classification_report(y_train,y_train_pred)) #gradient boosting train
```

	precision	recall	f1-score	support
0	0.94	0.99	0.96	56692
1	0.75	0.33	0.46	5326
accuracy			0.93	62018
macro avg	0.84	0.66	0.71	62018
weighted avg	0.92	0.93	0.92	62018

- GRADIENT BOOSTING
- ADA- BOOSTING

BALANCING DONE

- LOGISTIC REGRESSION
- RANDOM FOREST
- ENSEMBLE
- BAGGING
- DECISION TREE
- K-NEARST NEIOGBOUR

```
In [336]: # summarize the fit of the model
y_predict = model.predict(X_test)
print(metrics.classification_report(y_test, y_predict)) #log
```

	precision	recall	f1-score	support
0	0.91	1.00	0.95	24251
1	0.66	0.01	0.03	2329
accuracy			0.91	26580
macro avg	0.79	0.51	0.49	26580
weighted avg	0.89	0.91	0.87	26580

	precision	recall	f1-score	support
0	0.91	0.98	0.94	24251
1	0.09	0.03	0.04	2329
accuracy			0.89	26580
macro avg	0.50	0.50	0.49	26580
weighted avg	0.84	0.89	0.86	26580

```
y_predict=clf.predict(X_test)
print(metrics.classification_report(y_test,y_predict)) #
```

	precision	recall	f1-score	support
0	0.93	0.99	0.96	24251
1	0.71	0.23	0.35	2329
accuracy			0.92	26580
macro avg	0.82	0.61	0.66	26580
weighted avg	0.91	0.92	0.91	26580

	precision	recall	f1-score	support
0	0.93	0.99	0.96	24251
1	0.73	0.18	0.28	2329
accuracy			0.92	26580
macro avg	0.83	0.58	0.62	26580
weighted avg	0.91	0.92	0.90	26580

	precision	recall	f1-score	support
0	0.93	0.99	0.96	24251
1	0.71	0.21	0.32	2329
accuracy			0.92	26580
macro avg	0.82	0.60	0.64	26580
weighted avg	0.91	0.92	0.90	26580

	precision	recall	f1-score	support
0	0.94	0.99	0.96	24251
1	0.68	0.29	0.41	2329
accuracy			0.93	26580
macro avg	0.81	0.64	0.68	26580
weighted avg	0.91	0.93	0.91	26580

PARAMETER TUNNING

RANDOM FOREST

```
In [347]: print(metrics.classification_report(y_test,y_pred))  #best model after parameter tuning Random forest.
```

	precision	recall	f1-score	support
0	0.93	0.99	0.96	24251
1	0.72	0.24	0.37	2329
accuracy			0.93	26580
macro avg	0.83	0.62	0.66	26580
weighted avg	0.91	0.93	0.91	26580

CONCLUSION

- SUNCE THE DATA IS IMBALANCED. HENCE IT IS NOT GET GOOD PRECISION AND RECALL SCORE FOR TARGET CLASS.
- BALANCING WE PERFORM SMOTE BALANCING PERCISION OF THE TARGET CLASS IMPROVED BUT THE RECALL SCORE IS LOW.
- WE ALSO PERFORMED PARAMETER TUNNUNING AND USING RANDOM FOREST MODEL WE GOT GOOD PRECSION AND RECALL SCORE.