ICU SURVIVAL ANALYSIS

Yuling Gu

Alice Guo

Tam Nguyen

OUTLINE

- Business Use Case
- Dataset
- EDA & Approach
- Modeling
- Model Comparison
- Conclusion
- Lesson Learned

BUSINESS USE CASE

- In clinical practice, estimates of mortality risk can be useful in triage and resource allocation
- Help hospital to:
  - determine appropriate levels of care
  - prepare discussions with patients and their families around expected outcomes
- Help payers to know how the health outcomes of their policyholders will be affected, so that payers can identify useful policies

PROBLEM STATEMENTS

- MIT's GOSSIS community initiative is seeking an efficient way to address the problems with existing severity of illness systems:
  - They often lack generalizability beyond the patients on whom the models were developed, and
  - The models are often proprietary, costly to use (APACHE scoring system…), and suffer from opaque algorithms.

GOSSIS

OBJECTIVES

- Create a model that uses data from the first 24 hours of intensive care to predict patient survival with:
  - Better prediction probability of death (as compared to apache_4a_icu_prob, apache_4a_hospital_prob)
  - Minimize apache features
  - Transparent (easy to explain)
  - Generalizability
  - Less complexity

https://gossis.mit.edu/about/

DATA DESCRIPTION

DATA

- SOURCE
  - GOSSIS & WDIS
  - GOSSIS
  - WOMEN IN DATA SCIENCE @STANFORD UNIVERSITY
  - https://gossis.mit.edu/about/
- SHAPE
  - **90K patients**
  - **186 features**
  - **147 hospitals**
  - **6 countries**
- CATEGORY
  - Demographic (18 features)
  - Vitals (52 features)
  - Labs (76 features)
    - Vitals and Labs: First hour; First 24 hours
  - Apache (40 features)
    - Covariate (28 features)
    - Commodity (12 features)
- MISSING VALUES
  - First hour 78%
  - First 24 hour 50%

EDA & APPROACH

- Data Cleaning & Feature Engineering
- Initial Findings
- Challenges
- Approach
- Assumption

DATA CLEANING | DEMOGRAPHIC

FEATURES

	percent_missing
hospital_admit_source	23.3
age	4.6
bmi	3.7
weight	3.0
ethnicity	1.5
height	1.5
icu_admit_source	0.1
gender	0.0
elective_surgery	0.0
hospital_death	0.0
hospital_id	0.0
patient_id	0.0
icu_id	0.0
icu_stay_type	0.0
icu_type	0.0
pre_icu_los_days	0.0
readmission_status	0.0
encounter_id	0.0

CLEANING & IMPUTE

1

- **Drop features:**
  - that add no value to the model with std = 0: readmission_status
  - encounter id (repeat with patient id)

2

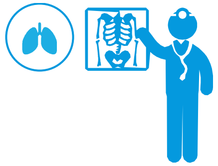
- **Replace negative values with 0:**
  - pre_icu_los_days (the length of stay (days) of the patient between hospital admission and unit admission)

3

- **Impute missing values**: (Mice imputer & most frequent)
  - Mice imputer *: age, height, weight
    - calculate BMI based on height, weight and impute missing value for BMI
  - Most frequent value:
    - ethnicity
    - Impute either hospital_admit_source or icu_admit_source, based on the other: most frequent category for each group.

(*) https://stats.stackexchange.com/questions/421545/multiple-imputation-by-chained-equations-mice-explained

DATA CLEANING | VITALS - LABS



FEATURES

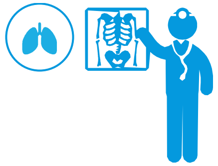
VITALS

	percent_missing
h1_diasbp_invasive_max	81.7
h1_diasbp_invasive_min	81.7
h1_sysbp_invasive_min	81.7
h1_sysbp_invasive_max	81.7
h1_mbp_invasive_min	81.6
h1_mbp_invasive_max	81.6
d1_diasbp_invasive_min	74.1
d1_diasbp_invasive_max	74.1
d1_sysbp_invasive_max	74.1
d1_sysbp_invasive_min	74.1
d1_mbp_invasive_max	73.9
d1_mbp_invasive_min	73.9
h1_temp_max	23.7
h1_temp_min	23.7
h1_mbp_noninvasive_max	9.9
h1_mbp_noninvasive_min	9.9
h1_diasbp_noninvasive_max	8.0
h1_diasbp_noninvasive_min	8.0
h1_sysbp_noninvasive_min	8.0
h1_sysbp_noninvasive_max	8.0
h1_mbp_max	5.1
h1_mbp_min	5.1
h1_resprate_max	4.8
h1_resprate_min	4.8
h1_spo2_max	4.6
h1_spo2_min	4.6
h1_diasbp_max	3.9
h1_diasbp_min	3.9
h1_sysbp_max	3.9
h1_sysbp_min	3.9
h1_heartrate_max	3.0
h1_heartrate_min	3.0
d1_temp_min	2.5
d1_temp_max	2.5
d1_mbp_noninvasive_max	1.6
d1_mbp_noninvasive_min	1.6
d1_diasbp_noninvasive_min	1.1
d1_diasbp_noninvasive_max	1.1
d1_sysbp_noninvasive_min	1.1
d1_sysbp_noninvasive_max	1.1
d1_resprate_max	0.4
d1_resprate_min	0.4
d1_spo2_max	0.4
d1_spo2_min	0.4
d1_mbp_max	0.2
d1_mbp_min	0.2
d1_diasbp_max	0.2
d1_diasbp_min	0.2
d1_sysbp_max	0.2
d1_sysbp_min	0.2
d1_heartrate_min	0.2
d1_heartrate_max	0.2

LABS

	percent_missing
h1_bilirubin_max	92.3
h1_bilirubin_min	92.3
h1_lactate_max	92.0
h1_lactate_min	92.0
h1_albumin_min	91.4
h1_albumin_max	91.4
h1_pao2fio2ratio_min	87.4
h1_pao2fio2ratio_max	87.4
h1_arterial_ph_min	83.3
h1_arterial_ph_max	83.3
h1_hco3_max	83.0
h1_hco3_min	83.0
h1_arterial_pco2_max	82.8
h1_arterial_pco2_min	82.8
h1_wbc_max	82.8
h1_wbc_min	82.8
h1_arterial_po2_max	82.8
h1_arterial_po2_min	82.8
h1_calcium_min	82.7
h1_calcium_max	82.7
h1_platelets_min	82.5
h1_platelets_max	82.5
h1_bun_min	81.9
h1_bun_max	81.9
h1_creatinine_min	81.7
h1_creatinine_max	81.7
h1_hematocrit_min	80.1
h1_hematocrit_max	80.1
h1_hemaglobin_max	79.7
h1_sodium_max	79.2
h1_sodium_min	79.2
h1_potassium_max	78.6
h1_potassium_min	78.6
d1_lactate_min	74.6
d1_lactate_max	74.6
d1_pao2fio2ratio_max	72.0
d1_pao2fio2ratio_min	72.0
d1_arterial_ph_min	65.6
d1_arterial_ph_max	65.6
d1_arterial_pco2_max	64.6
d1_arterial_pco2_min	64.6
d1_arterial_po2_max	64.6
d1_arterial_po2_min	64.6
d1_inr_min	63.2
d1_inr_max	63.2
h1_inr_max	63.2
h1_inr_min	63.2
d1_bilirubin_min	58.5
d1_bilirubin_max	58.5
h1_glucose_max	57.4
h1_glucose_min	57.4
d1_albumin_min	53.5
d1_albumin_max	53.5
d1_hco3_min	16.4
d1_hco3_max	16.4
d1_platelets_min	14.7
d1_platelets_max	14.7
d1_wbc_max	14.4
d1_wbc_min	14.4
d1_calcium_min	14.2
d1_calcium_max	14.2
d1_hemaglobin_max	13.2
d1_hemaglobin_min	13.2
d1_hematocrit_max	12.7
d1_hematocrit_min	12.7
d1_bun_max	11.5
d1_bun_min	11.5
d1_sodium_max	11.1
d1_sodium_min	11.1
d1_creatinine_max	11.1
d1_creatinine_min	11.1
d1_potassium_min	10.5
d1_potassium_max	10.5
d1_glucose_max	6.3
d1_glucose_min	6.3

DATA CLEANING | VITALS - LABS



CLEANING & IMPUTE

1

Min - Max problem (max < min)

Consistent values

	d1_bun_max	d1_bun_min
5678	4.0	113.1
13466	4.0	113.1
17084	4.0	53.0
19002	4.0	76.0
23600	4.0	113.1

Replace with nan

Different values by patients

	d1_resprate_max	d1_resprate_min
52067	14.0	20.0
71776	92.0	96.0
72737	14.0	25.0
73863	92.0	100.0
81314	14.0	18.0

Flip the columns

2

Impute & add in new features

- **Impute** missing values for:
  - d1 features (max and min) based on the most frequent values of patients in the same apache_3j_bodysystem group
- **Add features:**
  - calculated the difference between:
    - max and min value for every indicator, i.e. : *'diff_sodium_d1'='d1_sodium_max' - 'd1_sodium_min';*
    - 1^{st} hour and 1^{st} 24 hours, i.e.: *'diff_max_sodium_1hr_24hr' , 'diff_min_sodium_1hr_24hr'*
  - pulse pressure = sysbp - diasbp *(systolic blood pressure) - (diastolic blood pressure)*
  - the severity of patients: based on the number of missing features

3

Drop features

- Multiple measurements for same indicator , i.e. 'mbp': mean blood pressure.
  - ('d1_mbp_max','d1_mbp_min') ; ('d1_mbp_invasive_max','d1_mbp_invasive_min'); ('d1_mbp_noninvasive_max','d1_mbp_noninvasive_min')
- Drop first hour data (more than 80% of missing values)

DATA CLEANING | APACHE (ACUTE PHYSIOLOGY AND CHRONIC HEALTH EVALUATION)

FEATURES

	percent_missing
pao2_apache	77.3
fio2_apache	77.3
ph_apache	77.3
paco2_for_ph_apache	77.3
paco2_apache	77.3
bilirubin_apache	63.4
albumin_apache	59.3
urineoutput_apache	53.4
wbc_apache	24.0
hematocrit_apache	21.7
bun_apache	21.0
creatinine_apache	20.6
sodium_apache	20.3
glucose_apache	12.0
apache_4a_icu_death_prob	8.7
apache_4a_hospital_death_prob	8.7
temp_apache	4.5
gcs_verbal_apache	2.1
gcs_eyes_apache	2.1
gcs_motor_apache	2.1
apache_3j_bodysystem	1.8
apache_2_bodysystem	1.8
apache_2_diagnosis	1.8
resprate_apache	1.3
apache_3j_diagnosis	1.2
gcs_unable_apache	1.1
map_apache	1.1
heart_rate_apache	1.0
immunosuppression	0.8
intubated_apache	0.8
solid_tumor_with_metastasis	0.8
lymphoma	0.8
leukemia	0.8
cirrhosis	0.8
hepatic_failure	0.8
diabetes_mellitus	0.8
aids	0.8
arf_apache	0.8
ventilated_apache	0.8
apache_post_operative	0.0

APACHE: severity score and mortality estimation tool in US

CLEANING & IMPUTE

1

- **Drop features:**
  - apache_2_diagnosis (the APACHE II diagnosis for the ICU admission)
  - apache_2_bodysystem (Admission diagnosis group for APACHE II), due to high correlation between apache II and apache III.

2

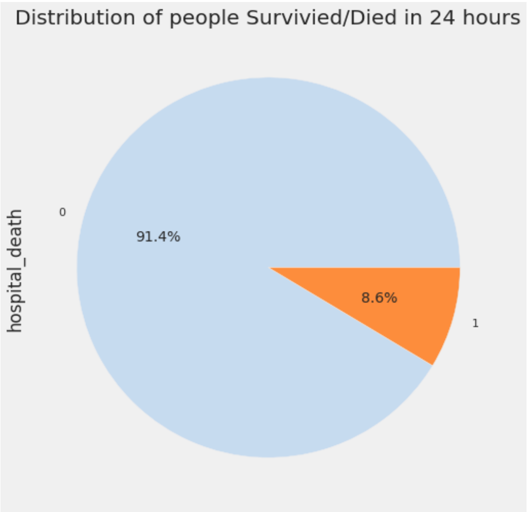
- **Impute:**
  - apache score by using 1st 24 hours min, max values for the same measurements
  - Replace negative value of apache_icu_death_prob with 0
- **Create** 'Undefined' category for missing values in apache_3j_bodysystem

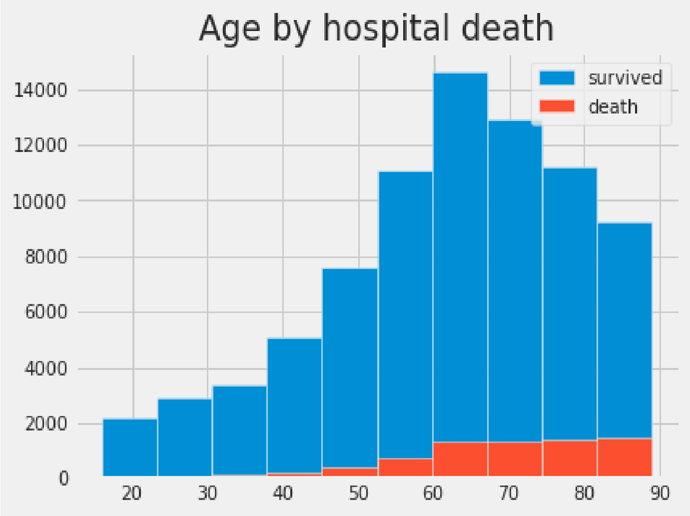
3

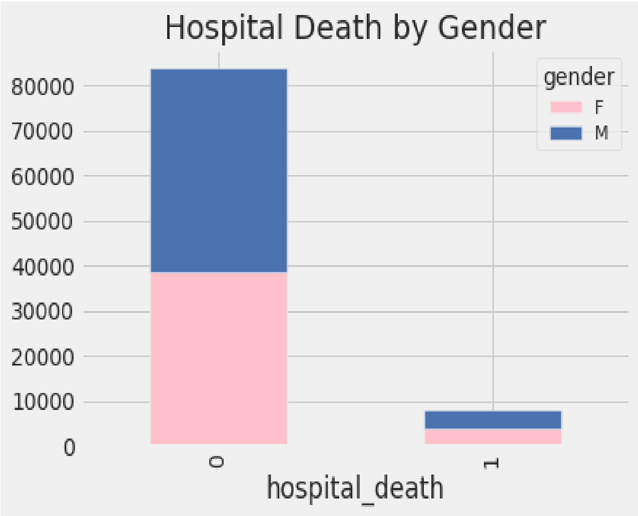
- **Encode:**
  - apache_3j_diagnosis: The APACHE III-J sub-diagnosis code which best describes the reason for the ICU admission, i.e.
    - 203: Aspiration pneumonia
    - '203.01': Arrest, respiratory (without cardiac arrest)

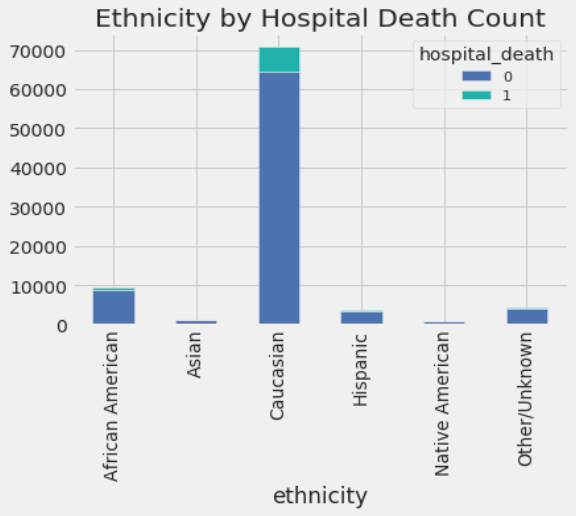
INITIAL FINDINGS (1)

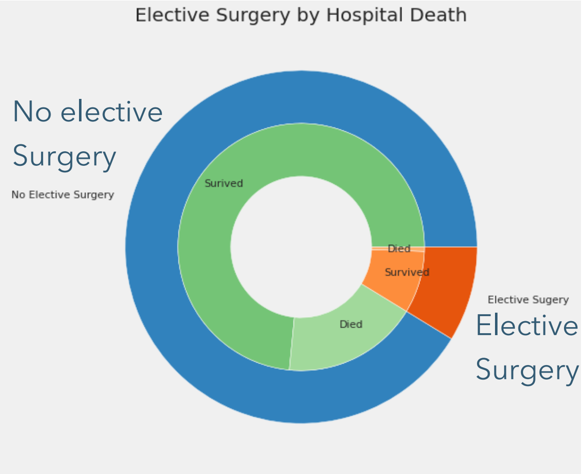
Age, Gender, Ethnicity distribution with Hospital Death









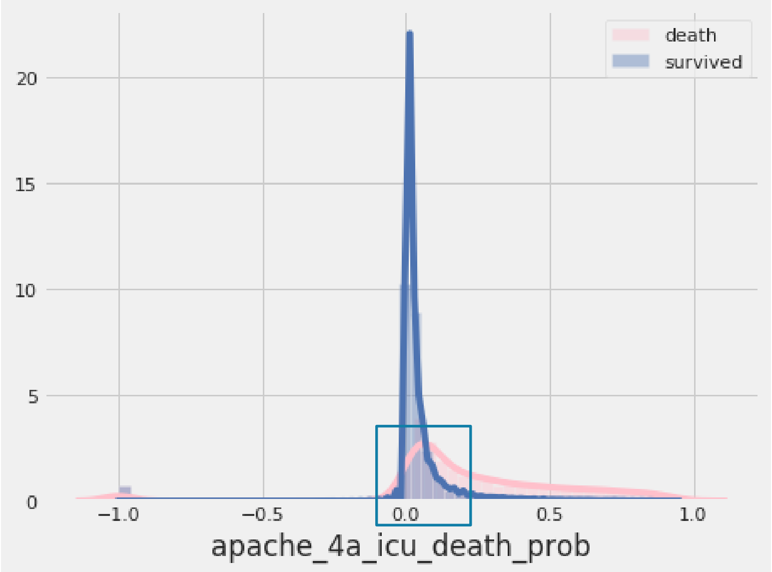


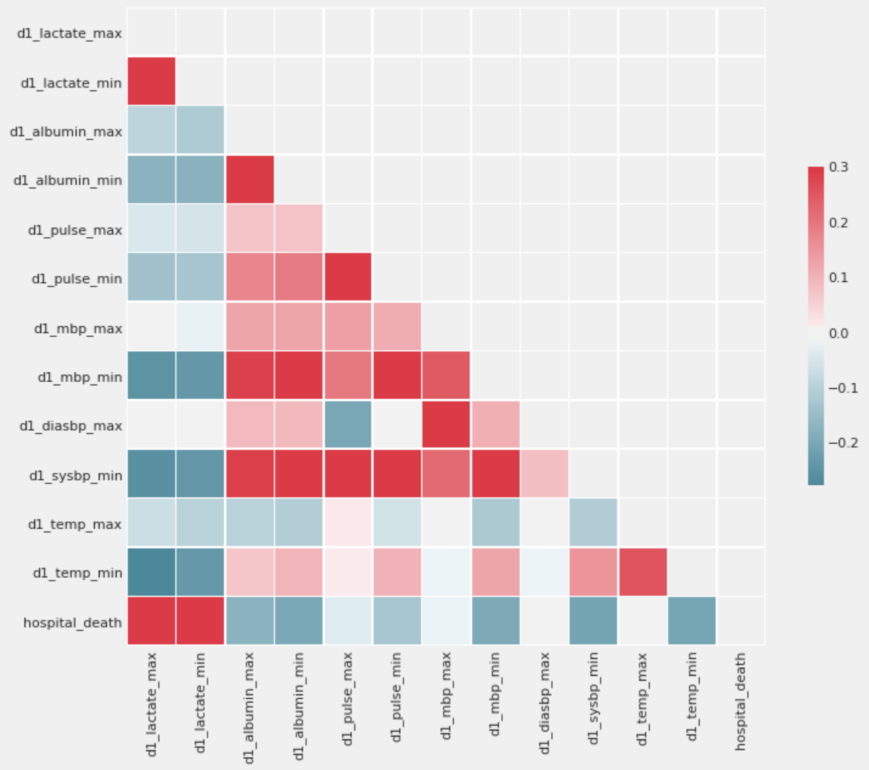
	icu_admit_source	count	death_rate
0	Other ICU	859	14.4
1	Other Hospital	2358	13.4
2	Floor	15611	13.4
3	Accident & Emergency	54060	8.6
4	Operating Room / Recovery	18713	3.7

	hospital_admit_source	count	death_rate
0	Step-Down Unit (SDU)	1131	18.8
1	Other ICU	233	15.0
2	Other	7	14.3
3	Floor	8055	13.9
4	Other Hospital	1641	13.5
5	Acute Care/Floor	1910	10.5
6	Direct Admit	6441	10.3
7	Emergency Department	36962	8.7
8	ICU	35	8.6
9	ICU to SDU	45	6.7
10	Chest Pain Center	134	6.0
11	Recovery Room	2896	3.6
12	Operating Room	9787	3.5
13	PACU	1017	3.0
14	Observation	10	0.0

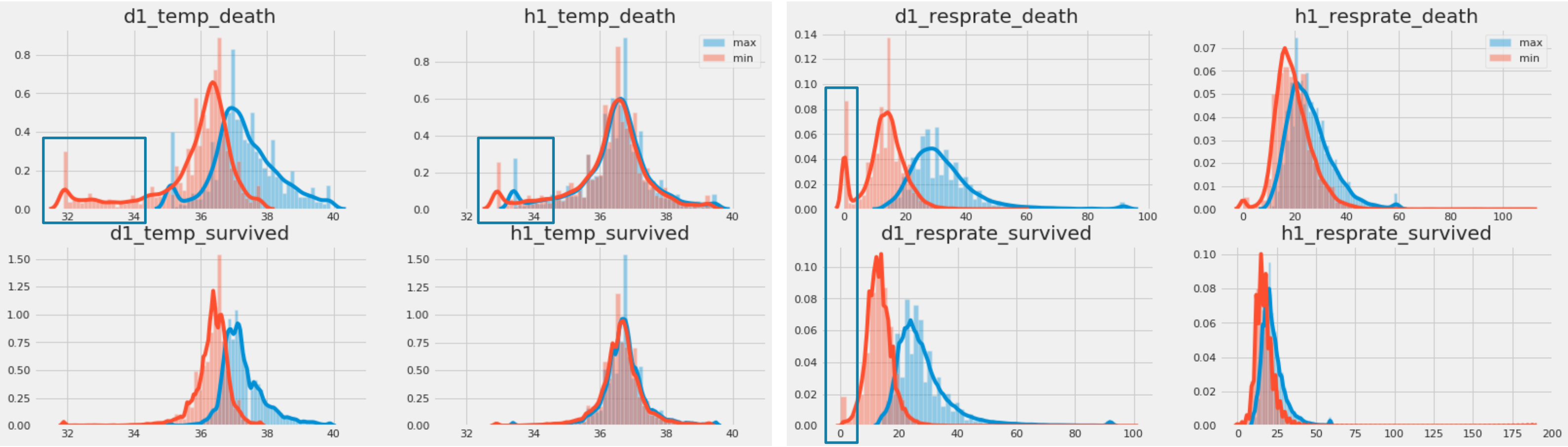
INITIAL FINDINGS (2)

	no_missingFeatures	total_patients	survived	death	death_rate
0	0.0	25.0	14.0	11.0	44.0
1	10.0	882.0	681.0	201.0	22.8
2	20.0	3673.0	3004.0	669.0	18.2
3	30.0	8196.0	6944.0	1252.0	15.3
4	40.0	13790.0	11688.0	2102.0	15.2
5	50.0	22863.0	19432.0	3431.0	15.0
6	60.0	35578.0	30762.0	4816.0	13.5
7	70.0	58977.0	52744.0	6233.0	10.6
8	80.0	77748.0	70732.0	7016.0	9.0
9	90.0	83446.0	76116.0	7330.0	8.8
10	100.0	87658.0	80057.0	7601.0	8.7
11	110.0	91101.0	83241.0	7860.0	8.6



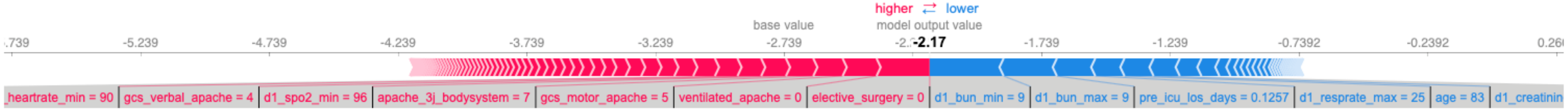


apache_3j_bodysystem	count	death_rate
Cardiovascular	29999	8.0
Neurological	11896	7.9
Sepsis	11740	15.8
Respiratory	11609	11.2
Gastrointestinal	9026	7.4
Metabolic	7650	1.5
Trauma	3842	6.7
Genitourinary	2172	6.2
Musculoskeletal/Skin	1166	4.7
Hematological	638	9.1
Gynecological	313	0.6

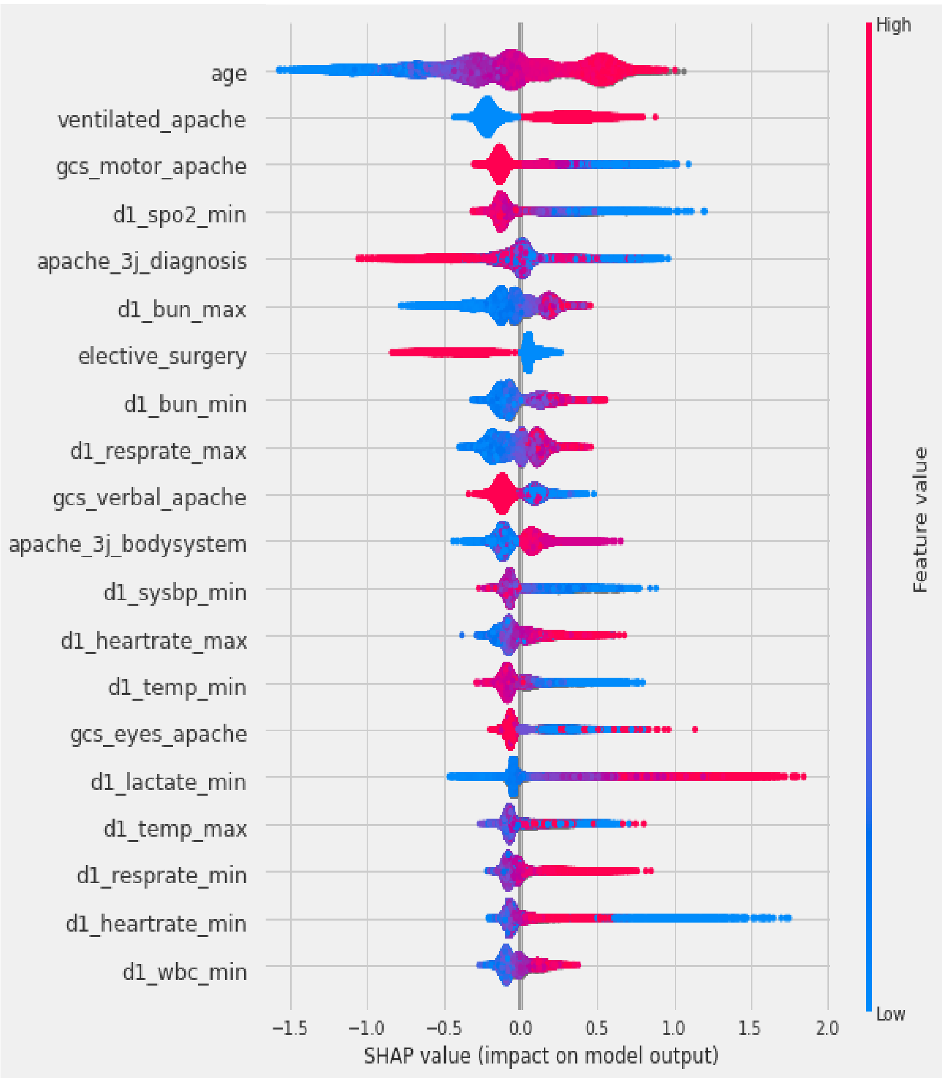


CHALLENGES

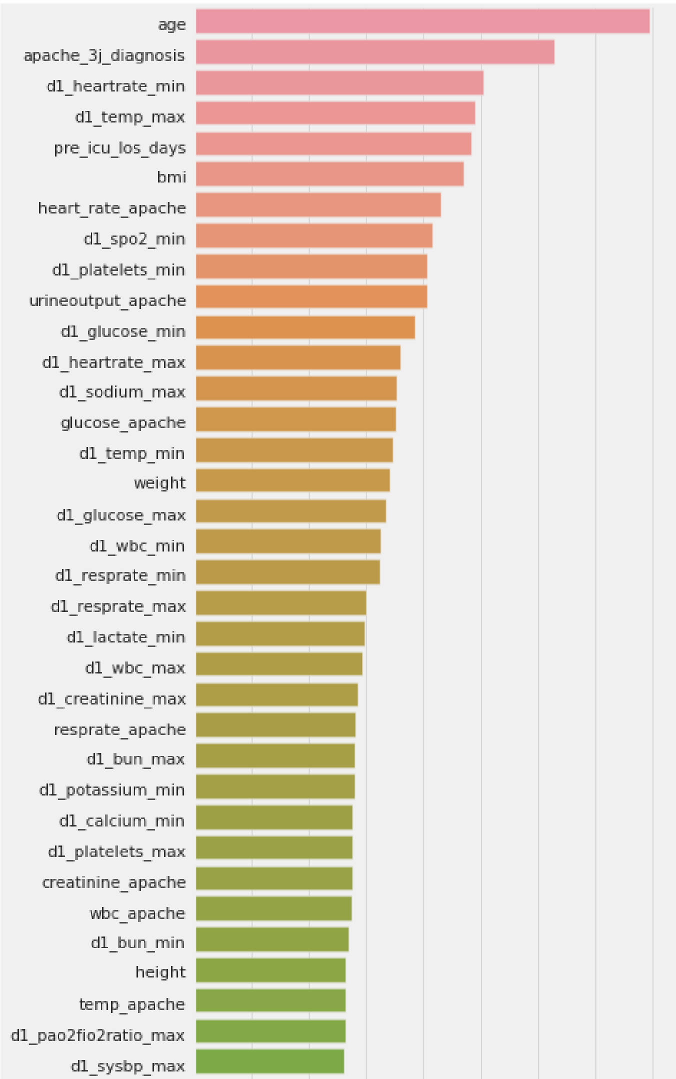
1. Minimum domain knowledge
2. High dimensional data
3. Plenty of missing variables, hard to impute
4. Highly imbalanced dataset



Shapley Explanator



LGB Feature Importance



INITIAL APPROACH

- BASELINE MODEL - Light gradient boost
  - to have a better understanding of the dataset - using feature importance & Shapley Explanator
  - can deal with missing data
- Oversampling – SMOTE to address imbalanced data problem

SMOTE – COMPUTATION EXPENSIVE BUT DOES NOT RESOLVE PROBLEM!

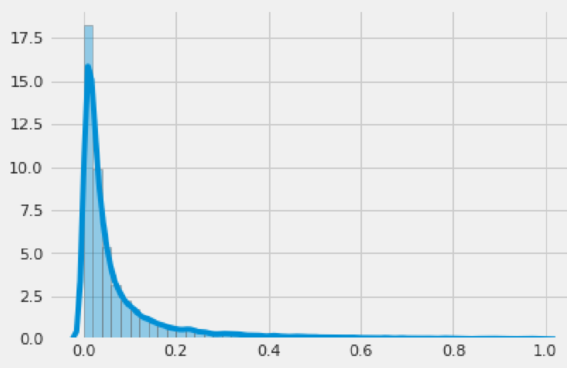
Impute Approach: Logistic Regression

Without SMOTE

```
ROC_AUC_test:  0.8586259061433837
Brier_Score_test:  0.061777675845410746
Accuracy Score_Test: 0.92
Classification Report_Test:
              precision    recall  f1-score   support

           0       0.93      0.99      0.96     20950
           1       0.60      0.21      0.31      1979

    accuracy                           0.92     22929
   macro avg       0.77      0.60      0.63     22929
weighted avg       0.90      0.92      0.90     22929
```

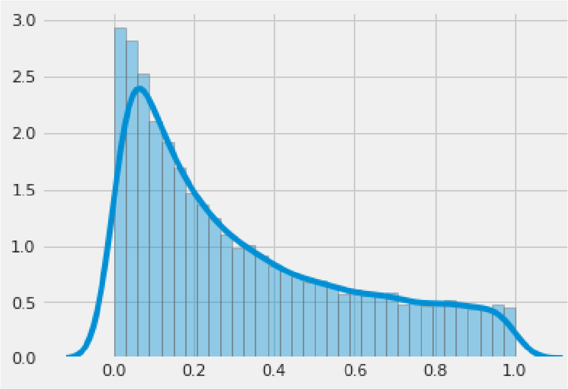


With SMOTE

```
ROC_AUC_test:  0.8551542991385682
Brier_Score_test:  0.15200817887032306
Accuracy Score_Test: 0.782
Classification Report_Test:
              precision    recall  f1-score   support

           0       0.97      0.78      0.87     20950
           1       0.25      0.76      0.38      1979

    accuracy                           0.78     22929
   macro avg       0.61      0.77      0.62     22929
weighted avg       0.91      0.78      0.83     22929
```



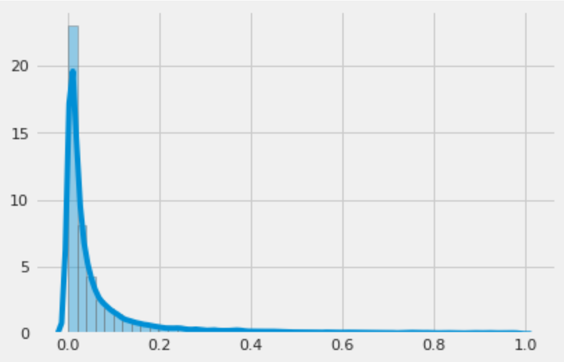
Binning Approach: Logistic Regression

Without SMOTE

```
ROC_AUC:  0.8829942317966332
Brier_Score:  0.0571302837125455
Accuracy Score_Test: 0.926
Classification Report_Test:
              precision    recall  f1-score   support

           0       0.94      0.98      0.96     20950
           1       0.65      0.30      0.41      1979

    accuracy                           0.93     22929
   macro avg       0.79      0.64      0.69     22929
weighted avg       0.91      0.93      0.91     22929
```

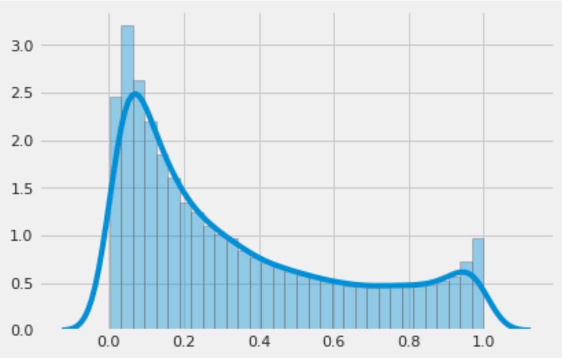


With SMOTE

```
ROC_AUC:  0.8650211227434602
Brier_Score:  0.16096862158562905
Accuracy Score_Test: 0.778
Classification Report_Test:
              precision    recall  f1-score   support

           0       0.98      0.78      0.86     20950
           1       0.25      0.80      0.38      1979

    accuracy                           0.78     22929
   macro avg       0.61      0.79      0.62     22929
weighted avg       0.91      0.78      0.82     22929
```



TRADE OF BETWEEN PRECISION – RECALL; HIGHER BRIER SCORE WHILE USING SMOTE

ASSUMPTION & SOLUTION

ASSUMPTIONS

BUSINESS RELATED

- Apache score is specialized to the US's patients; therefore, it might not be appropriate measurements for patient from outside of the US.
- Keep minimum features without losing accuracy

Solutions:

- Try 2 different approaches:
  - keep provided apache score for modeling , and compare against
  - models that remove almost apache score having similar feature measurements to labs and vitals.

MODEL BIAS

- Any features that makes the model biased and less generalizable should be dropped
- Our model only considers patient's health, severity instead of hospital or ICU quality, level of care, etc.

Solutions:

- Drop features:
  - hospital_id, icu_id,
  - apache_4a_hospital_death_prob
  - apache_4a_icu_death_prob
  - gender, ethnicity

BALANCED DATA

- SMOTE is not applicable for this data
- Adjust probability instead of trying to balance the data

Solutions:

- Adjust prediction probability to classify target variable based on quantile probability - 90% quantile
- Using other metrics to evaluate the model instead of accuracy, i.e.: AUC, precision-recall, Brier score, …

MISSING VALUES

- Patients with high missing features has lower survival rate overall
- Assuming missing measurement as people who falls into normal range of the test results.

Solutions:

- Binning dataset (vitals | labs)
  - Bin it into 5 categories based on quantile
  - Treat missing value as another category (normal range)
- Impute missing value using apache3j bodysystem
  - Features that has more than 50% missing values, fillna by 99

SOLUTIONS

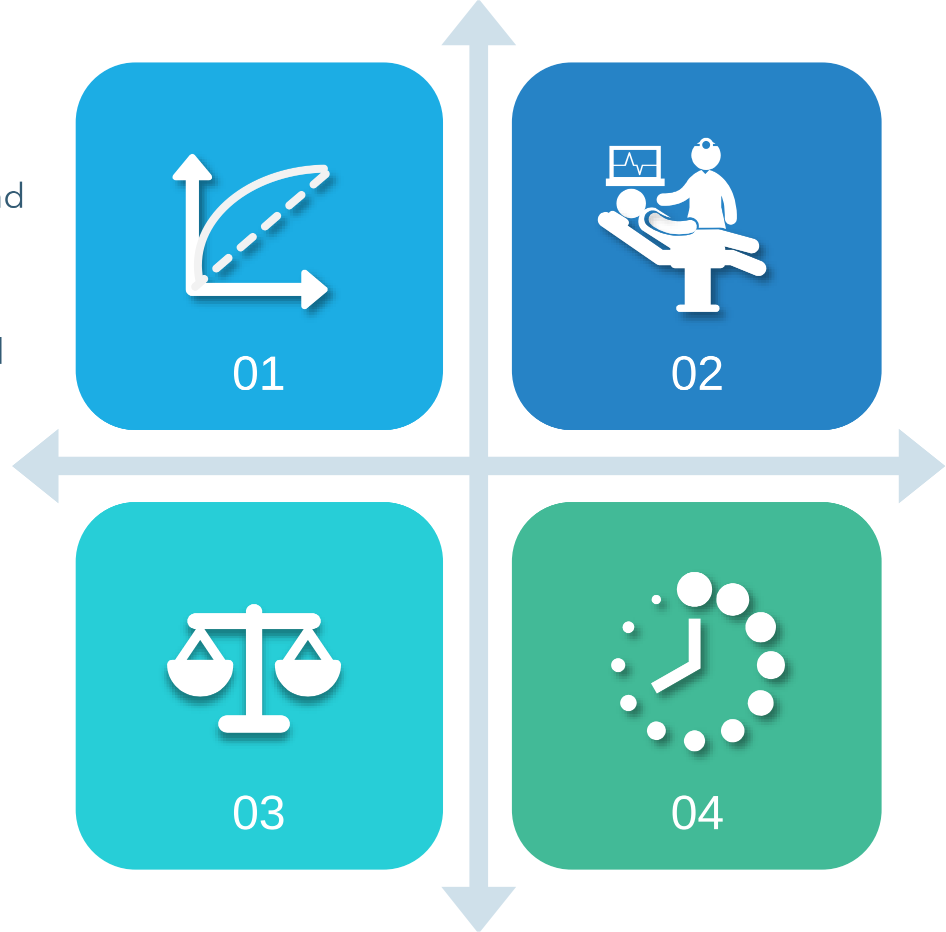
MODELING

- Assessment Criteria
- Models
  - Logistic Regression
  - Random Forest
  - Light Gradient Boosting
  - CatBoost
  - Neural Network
- Model Selection

MODEL ASSESSMENT CRITERIA

1 AUC score

- Imbalanced dataset - False positive rate and true positive rate are more important than accuracy
- Higher the AUC score, means better model results



2 Brier Score

- The lower Bier Score, the better model results

3 Precision- Recall

- More balanced between two scores, means better model results

4 Complexity

- Time takes model to run
- Number of features
- Less time and features make a good model

MODELING: USING STRATIFY SAMPLING ON 'HOSPITAL'_DEATH' TO HAVE SAME PROPORTION OF BOTH TRAIN AND TEST

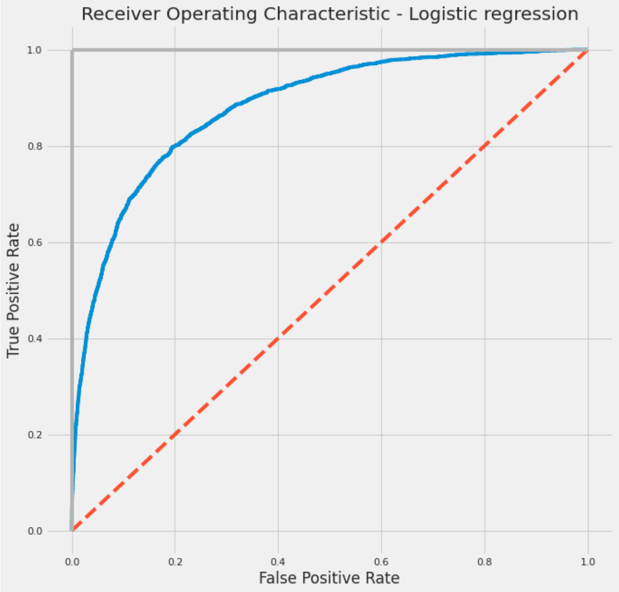
LOGISTIC REGRESSION (1) – **BINNING**

AUC

Train	Test
0.894	**0.880**

Brier Score

Train	Test
0.0549	**0.0575**



Classification Report_Train

	precision	recall	f1-score
0	0.96	0.94	0.95
1	0.49	0.56	0.52

Classification Report_Test

	precision	recall	f1-score
0	0.96	0.94	0.95
1	0.46	0.54	0.50

Complexity

- Total number of features using: **463 features**
- Time running models: **13.4s** *(Colab)*

Number of models tried: 10

Feature Importance

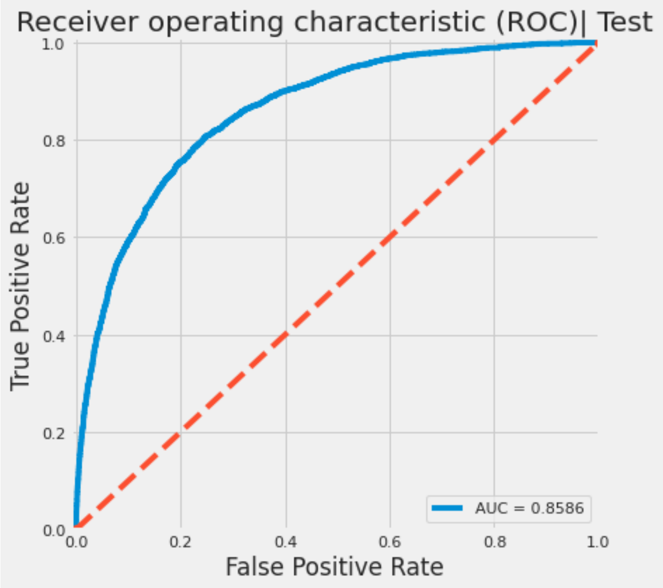
	Coefficient	Importance
apache_3j_bodysystem_Metabolic	1.0	1.5
bin_d1_lactate_min_100_percentile	1.0	1.5
elective_surgery_0	0.7	1.0
bin_d1_heartrate_max_100_percentile	0.7	1.0
bin_d1_creatinine_max_100_percentile	0.6	0.9
gcs_motor_apache_2.0	0.6	0.8
bin_d1_inr_max_100_percentile	0.5	0.8
age	0.5	0.8
bin_d1_temp_max_60_percentile	0.5	0.8
apache_3j_bodysystem_Neurological	0.5	0.8
bin_d1_hemaglobin_max_80_percentile	0.5	0.7
bin_d1_temp_max_100_percentile	0.5	0.7
gcs_motor_apache_6.0	0.5	0.7
bin_d1_wbc_min_100_percentile	0.5	0.7
solid_tumor_with_metastasis_0.0	0.5	0.7
bin_d1_temp_max_80_percentile	0.5	0.7
bin_d1_pao2fio2ratio_max_80_percentile	0.5	0.7
bin_urineoutput_apache_80_percentile	0.5	0.7
solid_tumor_with_metastasis_1.0	0.4	0.7
bin_d1_h1_min_hco3_(0.0, 32.0]	0.4	0.7

COMMENTS: No signs of overfitting

LOGISTIC REGRESSION (2) – **IMPUTING**

AUC

	Train	Test
AUC	0.869	0.859
Brier Score	0.0596	0.0617



Feature Importance

	Coefficient	Importance
apache_3j_bodysystem_Metabolic	1.3	4.6
apache_3j_bodysystem_Hematological	0.7	2.6
diff_max_platelets_24hr_1hr	0.7	2.5
elective_surgery_1	0.6	2.2
apache_3j_bodysystem_Genitourinary	0.6	2.2
icu_admit_source_Operating Room / Recovery	0.6	2.2
diff_min_platelets_24hr_1hr	0.6	2.2
gcs_motor_apache_6.0	0.6	2.1
ventilated_apache_0.0	0.6	2.0
age	0.5	1.9
gcs_motor_apache_2.0	0.5	1.7
solid_tumor_with_metastasis_0.0	0.5	1.7
gcs_motor_apache_5.0	0.4	1.6
diabetes_mellitus_1.0	0.4	1.5
hospital_admit_source_Step-Down Unit (SDU)	0.4	1.5
hospital_admit_source_Operating Room	0.4	1.5
gcs_motor_apache_1.0	0.4	1.5
icu_type_CSICU	0.4	1.4
apache_3j_bodysystem_Gynecological	0.4	1.3
apache_3j_bodysystem_Musculoskeletal/Skin	0.3	1.2

Classification Report_Train

	precision	recall	f1-score
0	0.95	0.94	0.95
1	0.45	0.49	0.47

Classification Report_Test

	precision	recall	f1-score
0	0.95	0.94	0.95
1	0.43	0.46	0.45

Complexity

- Total number of features using: **177 features**
- Time running models: **9.85s** *(Colab)*

Number of models tried: 2

COMMENTS: No signs of overfitting

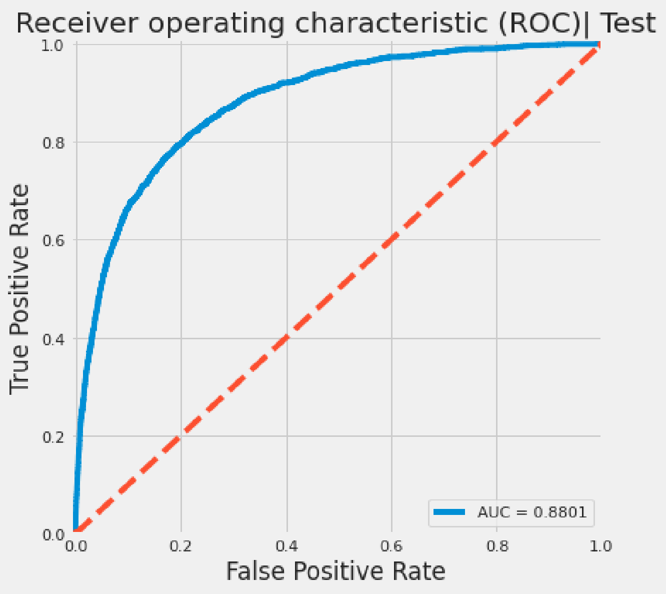
LOGISTIC REGRESSION (3) – **PCA**

AUC

Train	Test
0.891	**0.880**

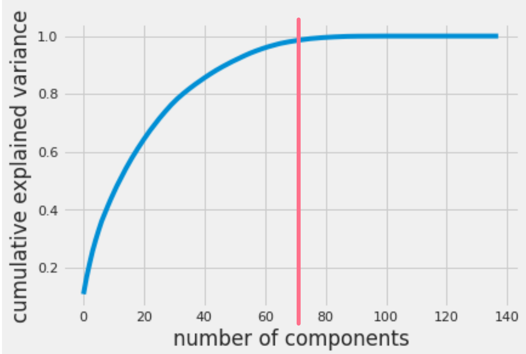
Brier Score

Train	Test
0.0559	**0.0579**



Feature Importance

	Coefficient	Importance
apache_3j_diagnosis_1207.01	2.2	2.0
apache_3j_diagnosis_703.03	1.6	1.5
apache_3j_diagnosis_702.01	1.6	1.4
apache_3j_diagnosis_1502.02	1.3	1.2
apache_3j_diagnosis_1501.01	1.3	1.1
apache_3j_diagnosis_1206.03	1.2	1.1
apache_3j_diagnosis_202.01	1.1	1.0
apache_3j_diagnosis_306.01	1.0	0.9
apache_3j_diagnosis_401.01	0.9	0.8
apache_3j_diagnosis_601.01	0.9	0.8
apache_3j_diagnosis_102.01	0.9	0.8
apache_3j_diagnosis_402.02	0.9	0.8
apache_3j_diagnosis_402.01	0.8	0.8
missing_features_104	0.8	0.7
apache_3j_diagnosis_403.01	0.8	0.7
apache_3j_diagnosis_201.01	0.7	0.7
apache_3j_diagnosis_301.01	0.7	0.6
es_105	0.7	0.6
sis_409.02	0.7	0.6
sis_209.01	0.6	0.6
sis_211.1	0.6	0.6
es_102	0.6	0.6



Using 70 out of 138 principle components

Classification Report_Train

	precision	recall	f1-score
0	0.96	0.94	0.95
1	0.48	0.55	0.51

Classification Report_Test

	precision	recall	f1-score
0	0.96	0.94	0.95
1	0.47	0.54	0.51

Complexity

- Total number of features using: **712 features**
- Time running models: **22.6s** *(Colab)*

Number of models tried: 2

COMMENTS: No signs of overfitting, most of the top feature importance are binary data

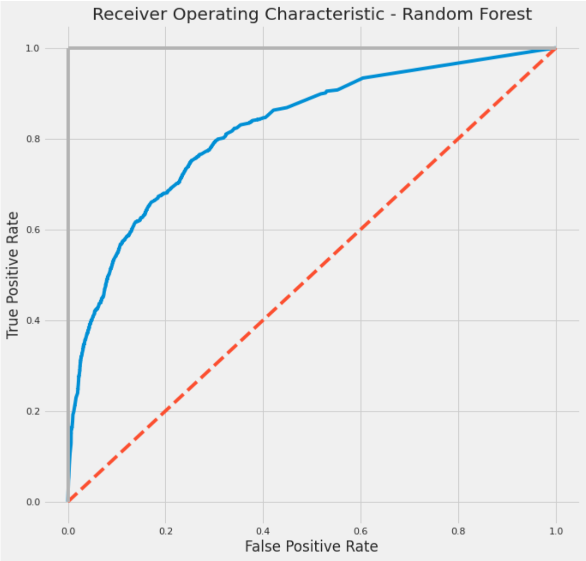
RANDOM FOREST – **BINNING**

AUC

	Train	Test
AUC	**0.823**	**0.820**

Brier Score

	Train	Test
Brier Score	**0.0645**	**0.0644**



Classification Report_Train

	precision	recall	f1-score
0	0.94	0.97	0.95
1	0.49	0.35	0.41

Classification Report_Test

	precision	recall	f1-score
0	0.94	0.97	0.95
1	0.49	0.35	0.41

Complexity

- Total number of features using: **34 features**
- Time running models: **8.48s** *(Colab)*

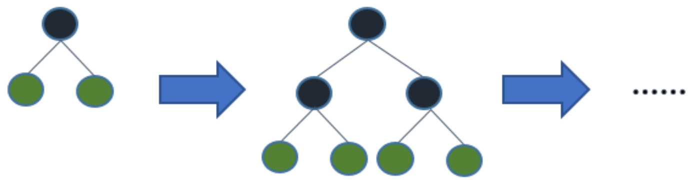
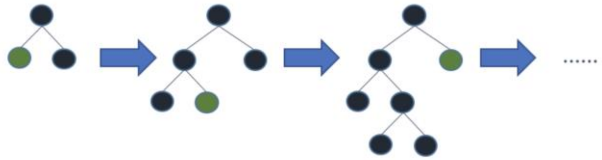
Number of models tried: 2

Feature Importance

	Importance
gcs_eyes_apache_1.0	0.26
gcs_motor_apache_6.0	0.24
bin_d1_spo2_min_(-1.0, 89.0]	0.13
ventilated_apache_0.0	0.08
icu_admit_source_Operating Room / Recovery	0.05
elective_surgery_0	0.03
bin_d1_lactate_max_100_percentile	0.03
bin_d1_creatinine_max_Normal	0.03
bin_d1_max_min_spo2_(10.0, 100.0]	0.03
bin_d1_creatinine_min_Normal	0.02
gcs_motor_apache_1.0	0.02
ventilated_apache_1.0	0.01
bin_d1_lactate_max_80_percentile	0.01
bin_d1_creatinine_max_100_percentile	0.01
apache_3j_bodysystem_Cardiovascular	0.01
bin_fio2_apache_(0.8, 1.0]	0.01
bin_d1_arterial_po2_max_Normal	0.01
bin_d1_hematocrit_max_Normal	0.00
bin_d1_arterial_pco2_max_Normal	0.00
pulse_pressure_min	0.00
bin_d1_max_min_pco2_Normal	0.00
bin_d1_sodium_min_Normal	0.00
bin_d1_hematocrit_min_Normal	0.00
bin_d1_arterial_pco2_min_Normal	0.00
apache_3j_bodysystem_Metabolic	0.00
age	0.00
bin_d1_wbc_max_Normal	0.00
bin_d1_lactate_min_100_percentile	0.00
missing_count	0.00

COMMENTS: No signs of overfitting, but poor performance on recall

WHY WE USE LGB AND CATBOOST OVER XBG?

<table>
<tr><th>Function</th><th>XGB</th><th>CATBOOST</th><th>LGB</th></tr>
<tr><td>Categorical variable</td><td>▪ Can not handle categorical variable; only accept numerical variables</td><td>▪ Can handle categorical variable automatically (one-hot max size encode)</td><td>▪ Can handle categorical variable: binning continuous variable to discrete variable based on histogram</td></tr>
<tr><td>Tree growth</td><td colspan="2">Level-wise tree growth

▪ uses pre-sorted algorithm & Histogram-based algorithm for computing the best split</td><td>Leaf-wise tree growth

▪ filter out the data instances for finding a split value; can reduce more loss than the level-wise algorithm, resulting in much better accuracy which can rarely be achieved by any of the existing boosting algorithms</td></tr>
<tr><td>Time complexity</td><td>▪ Take more time to run model, especially on high dimensional data</td><td>▪ The algorithm reduce time for hyper-parameter tuning and and lower the chances of overfitting also which leads to more generalized models</td><td>▪ Compatibility with large data set:
▪ Reduce significant training time as compared to XGBOOST</td></tr>
</table>

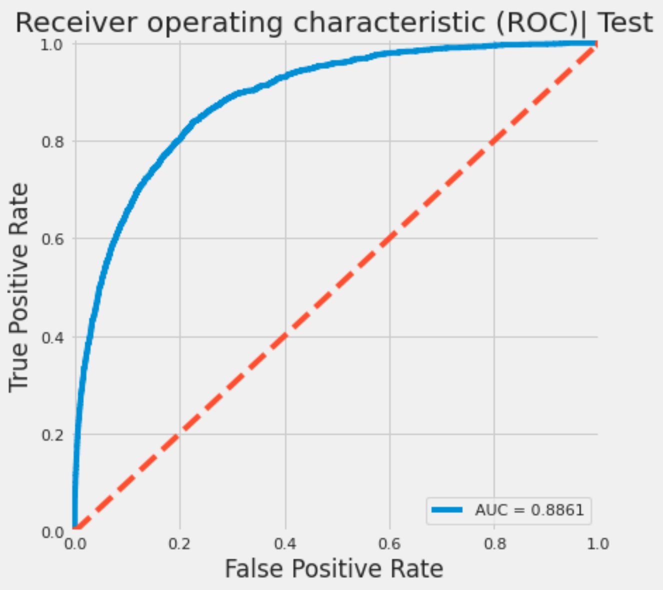
https://towardsdatascience.com/catboost-vs-light-gbm-vs-xgboost-5f93620723db

https://www.analyticsvidhya.com/blog/2017/06/which-algorithm-takes-the-crown-light-gbm-vs-xgboost/

LGB – **IMPUTING**

AUC

	Train	Test
AUC	**0.947**	**0.886**
Brier Score	**0.0417**	**0.0566**



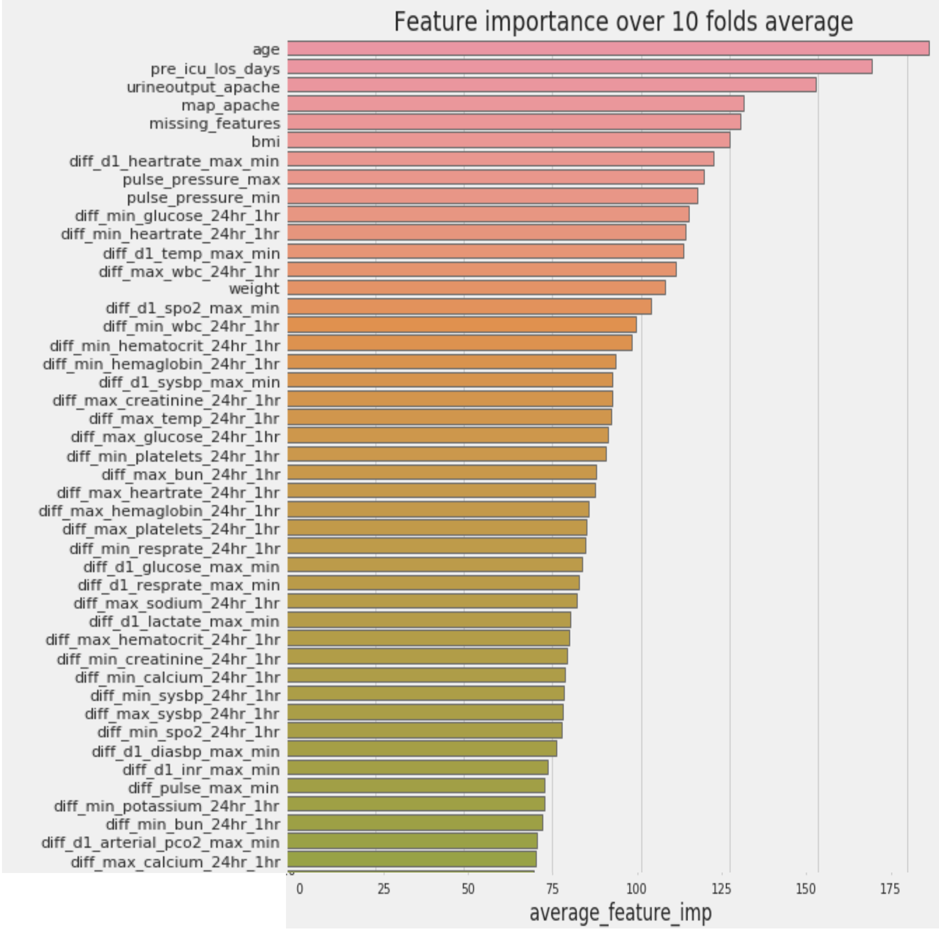
Classification Report_Train

	precision	recall	f1-score
0	0.97	0.95	0.96
1	0.60	0.73	0.66

Classification Report_Test

	precision	recall	f1-score
0	0.96	0.94	0.95
1	0.47	0.54	0.50

Feature Importance



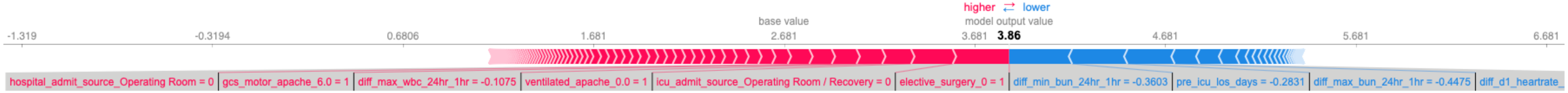
Complexity

- Total number of features using: **177 features**
- Time running models: **8.15 mins** *(Colab)* *With StratifiedKFold: 10 folds*

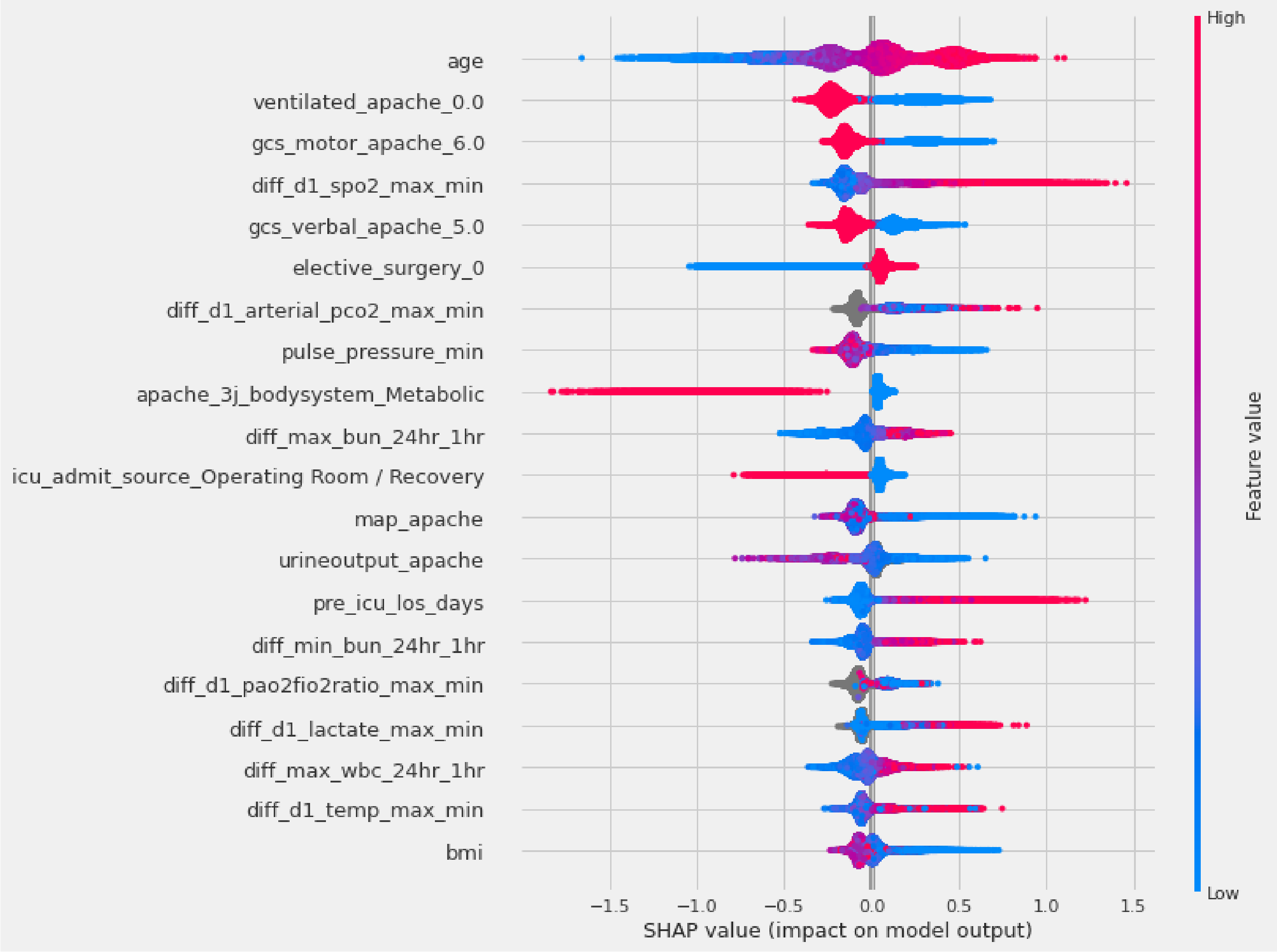
Number of models tried: 4

COMMENTS: Signs of **overfitting**! Big gap between train and test on AUC score

LGB – SHAPLEY



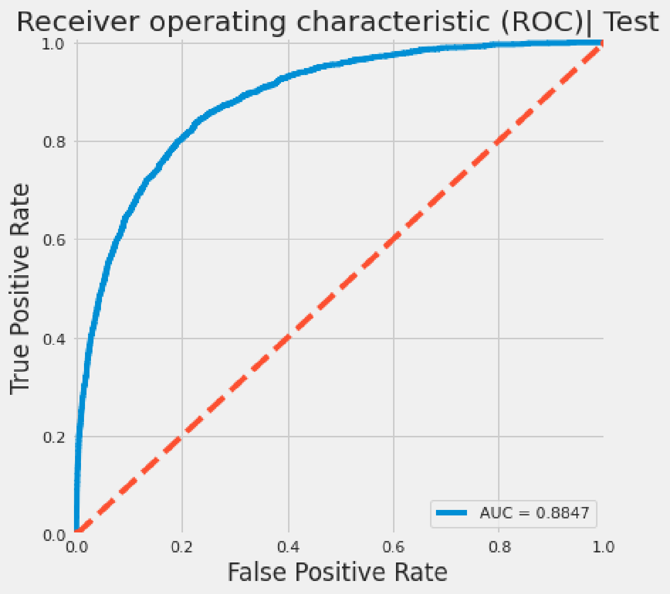
Feature Importance



CATBOOST – **IMPUTING**

AUC	Train	Test
AUC	0.834	0.884

Brier Score	Train	Test
Brier Score	0.0765	0.0567



Classification Report_Train

	precision	recall	f1-score
0	0.97	0.75	0.84
1	0.22	0.75	0.34

Classification Report_Test

	precision	recall	f1-score
0	0.96	0.94	0.95
1	0.47	0.54	0.50

Complexity

- Total number of features using: **107 features**
- Time running models: **8 mins** *(Colab)* + ***1hr*** - *GridSearch*

Grid Search Parameters:

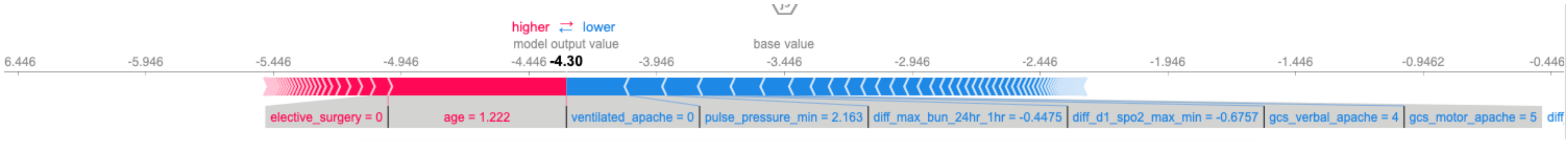
- *Learning rate: 0.04*
- *Depth: 7*

Feature Importance

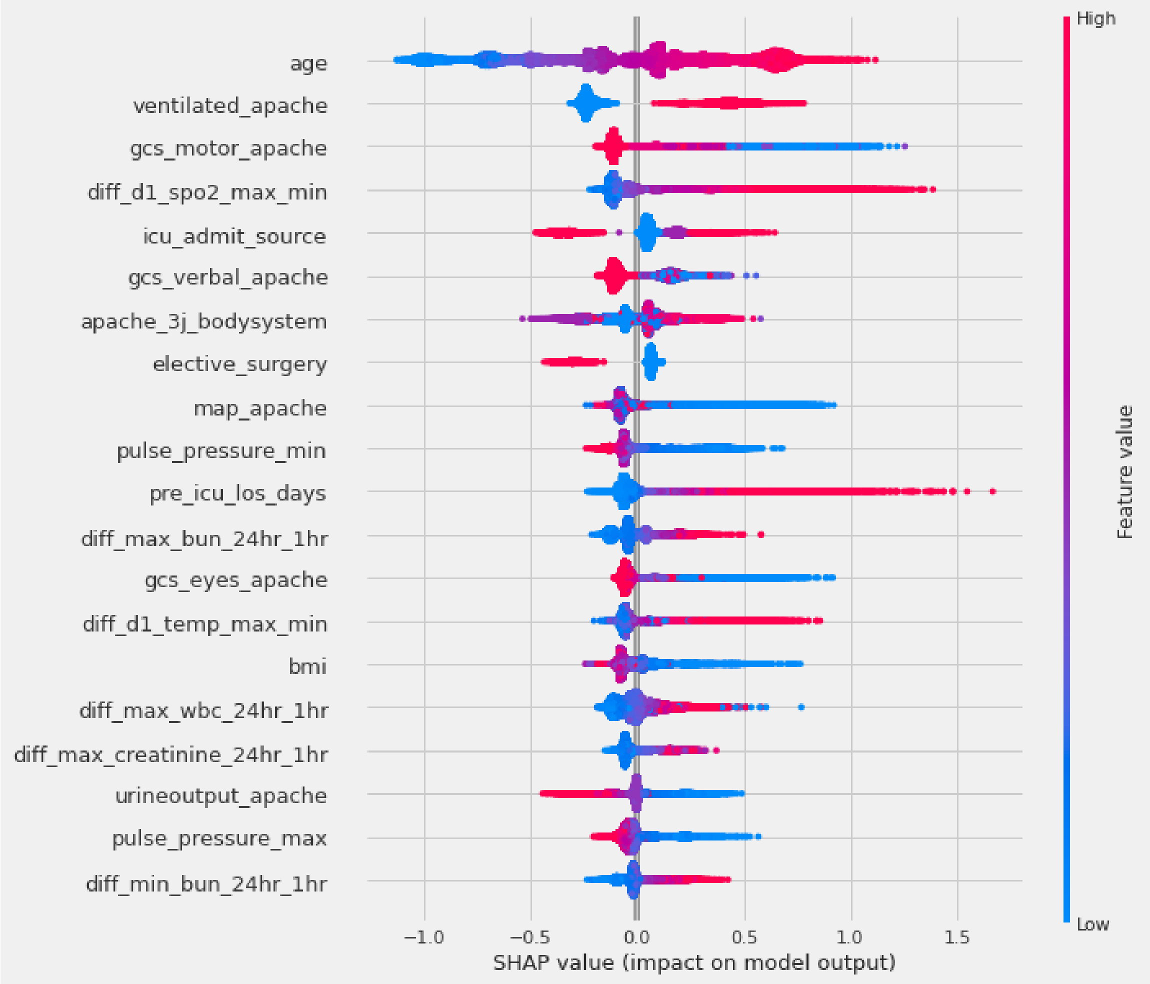
feature_names	feature_importances
age	7.4
ventilated_apache	4.6
gcs_motor_apache	3.7
diff_d1_spo2_max_min	3.2
apache_3j_bodysystem	3.2
gcs_verbal_apache	3.0
icu_admit_source	2.8
map_apache	2.4
elective_surgery	2.4
pre_icu_los_days	2.2
gcs_eyes_apache	2.0
diff_d1_temp_max_min	2.0
diff_max_wbc_24hr_1hr	1.9
missing_features	1.8
pulse_pressure_min	1.8
diff_max_bun_24hr_1hr	1.7
diff_min_bun_24hr_1hr	1.5
urineoutput_apache	1.5
diff_max_sodium_24hr_1hr	1.4
pulse_pressure_max	1.4

COMMENTS: The model fits test better than on train. Regularization help removes overfitting (poorer on train) but more **generalize** on test

CATBOOST - SHAPLEY

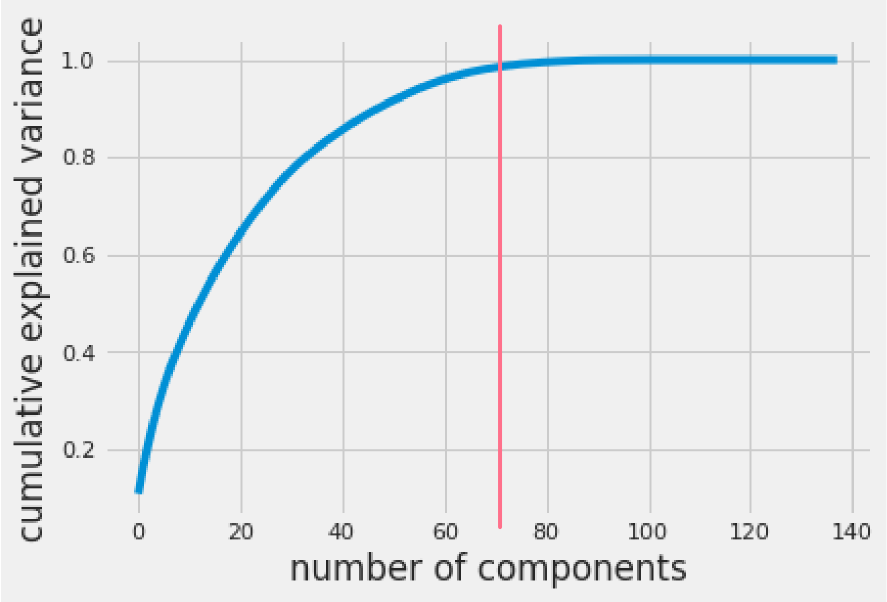


Feature Importance



PCA + NEURAL NETWORK (1)

Using 70 out of 138 principle components



- **Why using PCA for Neural Network?**
  - Reduces computation complexity by reducing the size of the network, amount of data needed to train
  - Reduce overfitting
  - However, discriminative information that distinguishes the class might be in low variance components.

Neural Network Model

```
def create_model(input_dim):
    input_layer = Input(shape=(input_dim, ))
    classifier = Dense(256, activation='relu')(input_layer)
    classifier = Dense(128, activation='relu')(input_layer)
    classifier = Dropout(0.5)(classifier)
    classifier = Dense(1, activation='sigmoid')(classifier)
    classModel = Model(inputs=input_layer, outputs=classifier)
    classModel.compile(optimizer='adam', loss='mean_squared_error')
    return classModel
```

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
input_5 (InputLayer)         (None, 712)               0
_________________________________________________________________
dense_10 (Dense)             (None, 128)               91264
_________________________________________________________________
dropout_3 (Dropout)          (None, 128)               0
_________________________________________________________________
dense_11 (Dense)             (None, 1)                 129
=================================================================
Total params: 91,393
Trainable params: 91,393
Non-trainable params: 0
```

```
nb_epoch = 50
batch_size = 128
adam = Adam(lr=0.0005)

earlystopping = EarlyStopping(monitor='val_loss', patience=4, verbose=0)
checkpointer = ModelCheckpoint(filepath="classifier.h5",
                               verbose=0,
                               save_best_only=True)

%time class_history = classModel.fit(X_train, y_train, epochs=nb_epoch, batch_size=batch_size,
                                     shuffle=True,validation_data=(X_test, y_test),
                                     verbose=1,callbacks=[earlystopping, checkpointer]).history
```

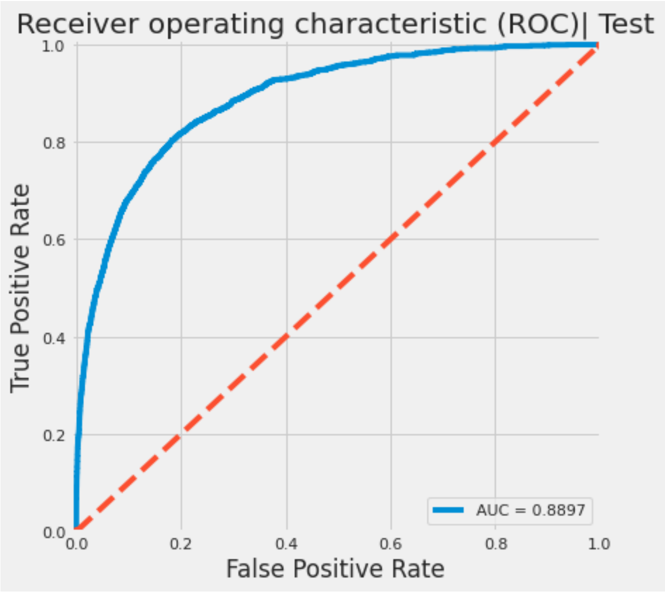
PCA + NEURAL NETWORK (2)

AUC

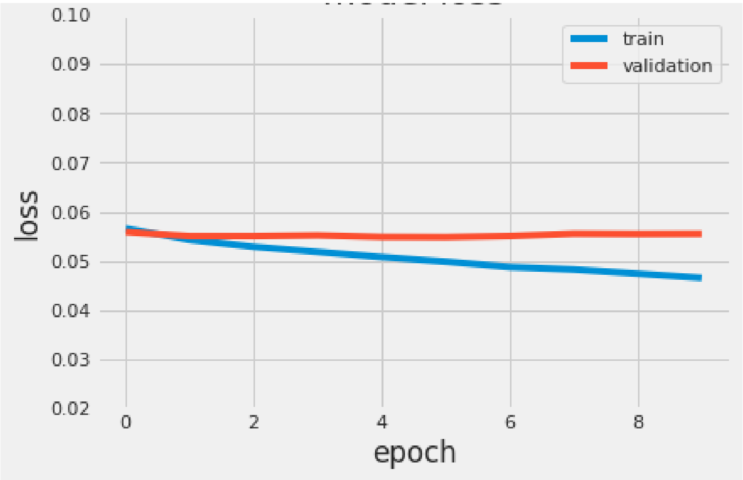
	Train	Test
AUC	0.912	0.890

Brier Score

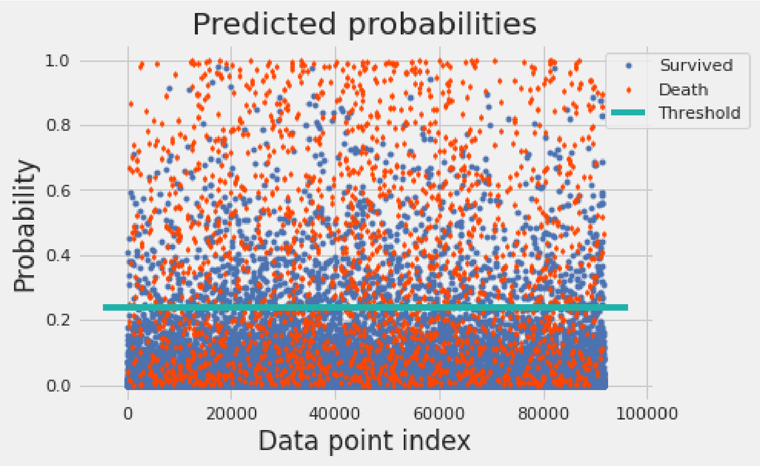
	Train	Test
Brier Score	0.0463	0.0546



Train vs. Validation loss



Sparse – probability for both death and survived



Classification Report_Train

	precision	recall	f1-score
0	0.97	0.95	0.96
1	0.56	0.64	0.60

Classification Report_Test

	precision	recall	f1-score
0	0.96	0.94	0.95
1	0.48	0.56	0.52

Complexity

- Total number of features using: **91,393 features**
- Time running models: **38.5s** *(Colab)*

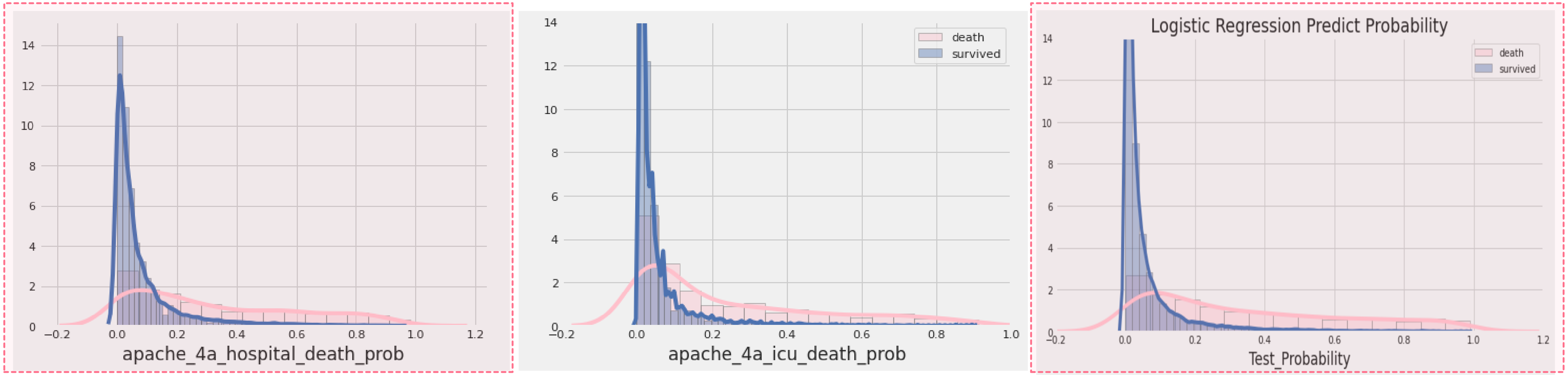
COMMENTS: No signs of overfitting

MODEL COMPARISION & SELECTION

No	Model	AUC	Brier Score	Precision	Recall	Run Time	No. features	Overfitting
1	Logistic Reg - Binning	**0.880**	**0.0575**	**0.46**	**0.54**	**13.4s**	463	No
2	Logistic Reg - Imputing	0.859	0.0617	0.43	0.46	9.85s	177	No
3	Logistic Reg - PCA	0.880	0.0579	0.47	0.54	22.6s	712	No
4	Random Forest - Binning	0.820	0.0644	0.49	0.35	8.48s	34	No
5	LGB - Imputing	0.886	0.0566	0.47	0.54	**8.15 mins**	177	**Yes**
6	CatBoost	0.884	0.0567	0.47	0.54	**8 mins**	107	No
7	Neural Network - PCA	**0.890**	**0.0546**	**0.48**	**0.56**	38.5s	91,939	No

- Best performance in terms of AUC, Brier Score, Precision, Recall
- Business related (less complexity + generalizable)

COMPARE THE WINNING MODEL WITH ACTUAL APACHE_4A_PROBABILITY



AUC

Apache_4a: **0.794**

Logistic - Binning: **0.880**

Brier Score

Apache_4a: **0.0656**

Logistic - Binning: **0.0575**

Classification Report_Logistic-Binning

	precision	recall	f1-score
0	0.96	0.94	0.95
1	0.46	0.54	0.50

Classification Report_apache_4a

	precision	recall	f1-score
0	0.95	0.92	0.93
1	0.36	0.52	0.43

The winner

Logistic_Binning

SUMMARY

- Conclusion
- Lesson Learned
- Future Work

CONCLUSION

- Logistic regression is a good model for this type of dataset
- SMOTE does not help in this case since:
  - it does not take into account neighboring examples can be from other classes, introducing additional noise
  - is not very practical for high dimensional data
- Binning: works well for extreme values, that shows importance in the model

LESSON LEARNT

- Should not drop features or observations with high percentage of missing values without having a basic domain knowledge
- Trade-off between explainability and interpretability. The best performance model does not have to be the one that being used in practical
- Be creative! Our call to adjust the threshold instead of using the original probability threshold: 0.5 for imbalanced data
- Try to use functions to utilize code

FUTURE WORK

- Have a better understanding about the features (domain knowledge)
- Collect more data: Using GAN to generate more data instead of using SMOTE
- Improve model prediction ability by:
  - Learning the key features importance of each model and try to combine such features
  - Applying Autoencoder to get a higher level of understanding the characteristics of patients who were misclassified with 'death' or 'survived'

THANK YOU!

APPENDIX

DICTIONARY

Features	Definition
ventilated_apache	Whether the patient was invasively ventilated at the time of the highest scoring arterial blood gas using the oxygenation scoring algorithm, including any mode of positive pressure ventilation delivered through a circuit attached to an endo-tracheal tube or tracheostomy
urineoutput_apache	The total urine output for the first 24 hours
map_apache	The mean arterial pressure measured during the first 24 hours which results in the highest APACHE III score
intubated_apache	Whether the patient was intubated at the time of the highest scoring arterial blood gas used in the oxygenation score
apache_post_operative	The APACHE operative status; 1 for post-operative, 0 for non-operative
arf_apache	Whether the patient had acute renal failure during the first 24 hours of their unit stay, defined as a 24 hours urine output <410ml, creatinine >=133 micromol/L and no chronic dialysis
gcs_unable_apache	Whether the Glasgow Coma Scale was unable to be assessed due to patient sedation
apache_3j_diagnosis	The APACHE III-J sub-diagnosis code which best describes the reason for the ICU admission