DATA ENGINEERING AND MODELING

Data description:

The data was collected from physionet.org from the cardiology computing department.

Six descriptors are collected at the time the patient is admitted to the ICU. Their associated time-stamps are set to 00:00 (thus they appear at the beginning of each patient's record).

- RecordID (a unique integer for each ICU stay)
- Age (years)
- Gender (0: female, or 1: male)
- Height (cm)
- ICUType (1: Coronary Care Unit, 2: Cardiac Surgery Recovery Unit, o 3: Medical ICU, or 4: Surgical ICU)
- Weight (kg)

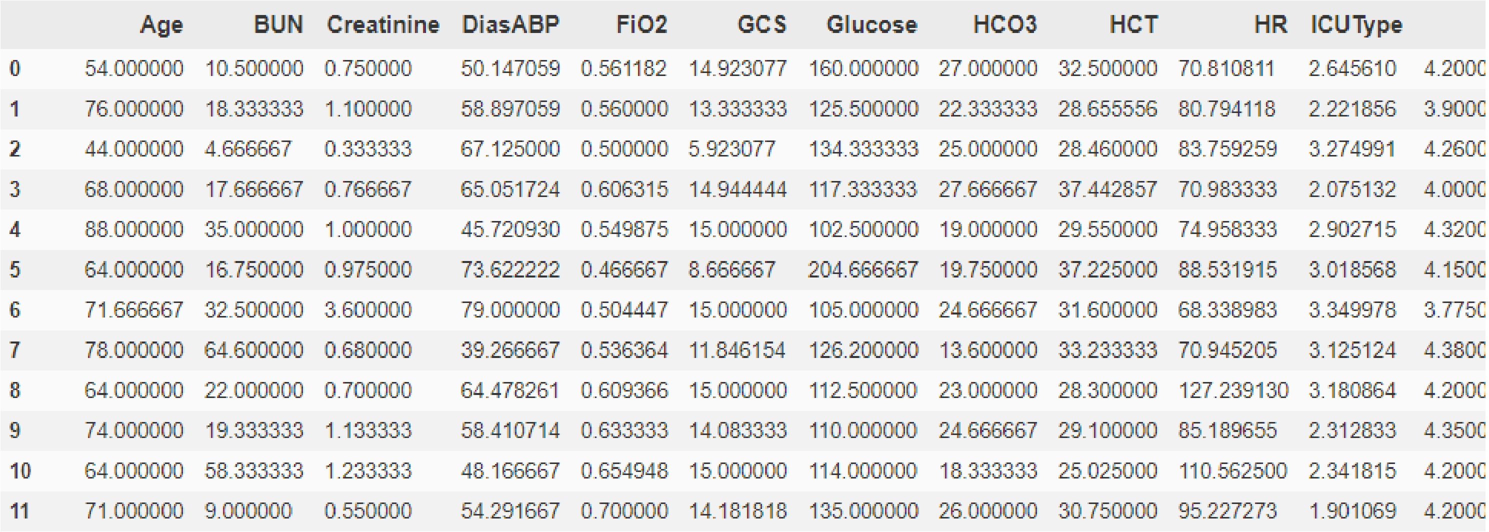
Given the data of the patients admitted in an ICU, at least a few of the following variables are recorded once a day during their stay.

- Albumin (g/dL)
- ALP [Alkaline phosphatase (IU/L)]
- ALT[Alaninetransaminase (IU/L)]
- AST [Aspartate transaminase (IU/L)]
- Bilirubin (mg/dL)
- BUN [Blood urea nitrogn (mg/dL)]
- Cholesterol (mg/dL)
- Creatinine [Serum creatinine (mg/dL)]
- DiasABP [Diastolic arterial blood pressure]
- FiO2 [Fractional inspired O2 (0-1)]
- GCS [Glasgow Coma Score (3-15)]
- Glucose [Serum glucose (mg/dL)]
- HCO3 [Serum bicarbonate (mmol/L)]
- HR [Heart rate (bpm)]
- K [Serum potassium (mEq/L)]
- Lactate (mmol/L)
- HCT [Hematocrit (%)]
- Mg [Serum magnesium (mmol/L)]
- MAP [mean arterial blood pressure]
- Na [Serum sodium (mEq/L)]
- PaCO2 [partial pressure of arterial CO2]
- SysABP [Systolic arterial blood pressure]
- pH [Arterial pH (0-14)]
- Platelets (cells/nL)
- RespRate [Respiration rate (bpm)]
- Urine [Urine output (mL)]

- SaO2 [O2 saturation in hemoglobin (%)]
- WBC [White blood cell count (cells/nL)]
- Temp [Temperature (°C)]
- TropT [Troponin-T (μg/L)

Target Variables:

- <u>In-hospital Mortality Class:</u> (0: survivor, or 1: died in-hospital).
- <u>Length of Stay</u>: It is the number of days between the patient's admission to the ICU and the end of hospitalization.



	Age	BUN	Creatinine	DiasABP	FiO2	GCS	Glucose	HCO3	HCT	HR	ICUType	
0	54.000000	10.500000	0.750000	50.147059	0.561182	14.923077	160.000000	27.000000	32.500000	70.810811	2.645610	4.2000
1	76.000000	18.333333	1.100000	58.897059	0.560000	13.333333	125.500000	22.333333	28.655556	80.794118	2.221856	3.9000
2	44.000000	4.666667	0.333333	67.125000	0.500000	5.923077	134.333333	25.000000	28.460000	83.759259	3.274991	4.2600
3	68.000000	17.666667	0.766667	65.051724	0.606315	14.944444	117.333333	27.666667	37.442857	70.983333	2.075132	4.0000
4	88.000000	35.000000	1.000000	45.720930	0.549875	15.000000	102.500000	19.000000	29.550000	74.958333	2.902715	4.3200
5	64.000000	16.750000	0.975000	73.622222	0.466667	8.666667	204.666667	19.750000	37.225000	88.531915	3.018568	4.1500
6	71.666667	32.500000	3.600000	79.000000	0.504447	15.000000	105.000000	24.666667	31.600000	68.338983	3.349978	3.7750
7	78.000000	64.600000	0.680000	39.266667	0.536364	11.846154	126.200000	13.600000	33.233333	70.945205	3.125124	4.3800
8	64.000000	22.000000	0.700000	64.478261	0.609366	15.000000	112.500000	23.000000	28.300000	127.239130	3.180864	4.2000
9	74.000000	19.333333	1.133333	58.410714	0.633333	14.083333	110.000000	24.666667	29.100000	85.189655	2.312833	4.3500
10	64.000000	58.333333	1.233333	48.166667	0.654948	15.000000	114.000000	18.333333	25.025000	110.562500	2.341815	4.2000
11	71.000000	9.000000	0.550000	54.291667	0.700000	14.181818	135.000000	26.000000	30.750000	95.227273	1.901069	4.2000

Data Preprocessing:

Data was sparse, with imbalanced class distribution in In-hospital mortality variable.

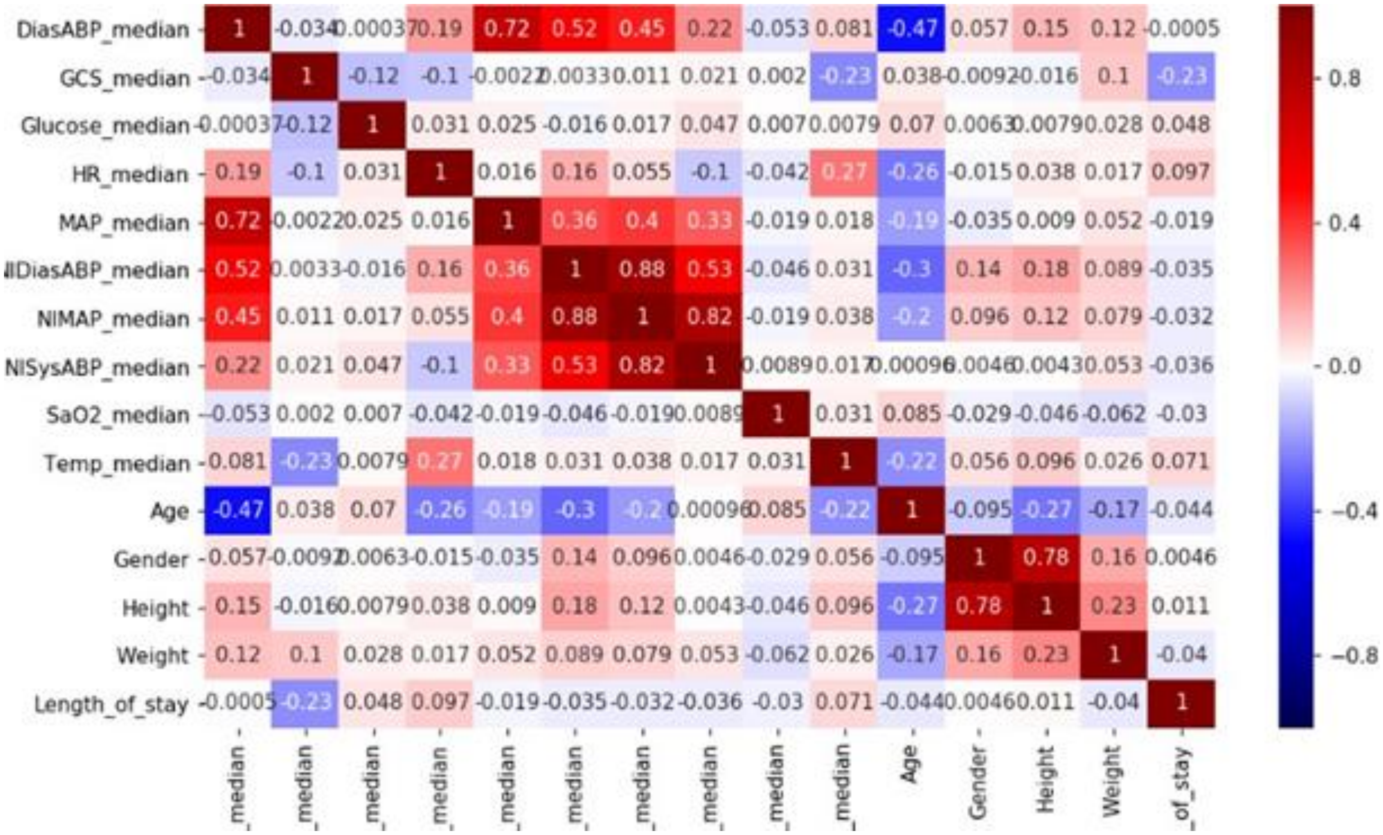
We used MICE imputation in sklearn, with Iterative Imputation approach.

There was a total of 12000 datapoint of the patients which was read in form of txt files.

We followed a Standard Scaler preprocessing technique for normalizing the data, since we want a data matrix with 0 mean and a variance of 1.

We pickled out an imputation module, to be applied on the fresh data points when new patient enters. [File name: impute.pkl]

Highly Correlated features were removed to avoid overfitting.



Correlation Matrix of Temporal Features

The above correlation matrix was used to remove highly correlated features. Here for eg: NIMAP is correlated to DiasABP and NISysABP etc.

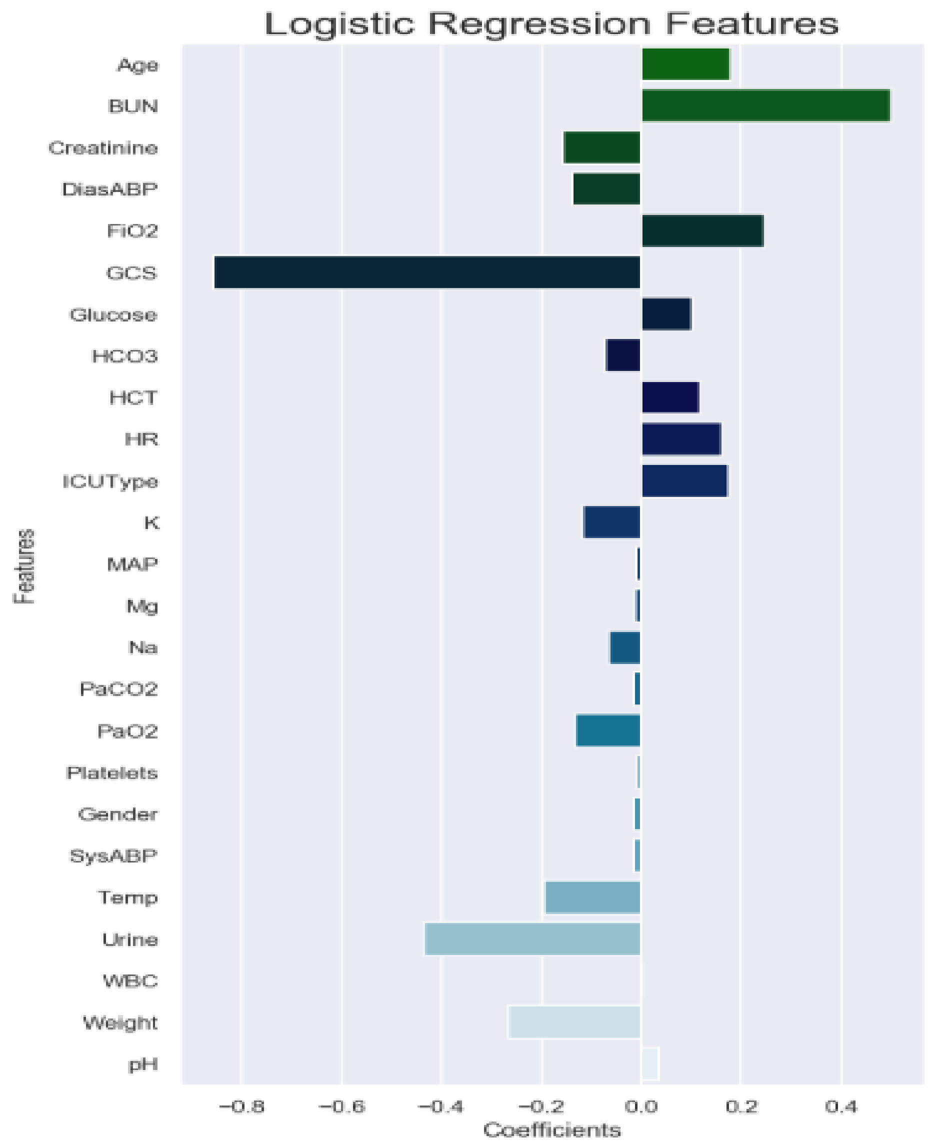
The median values of temporal features were only taken since it oscillated between the first and the last time stamps and this also allowed the reduction of highly correlated features.

Feature Selection:

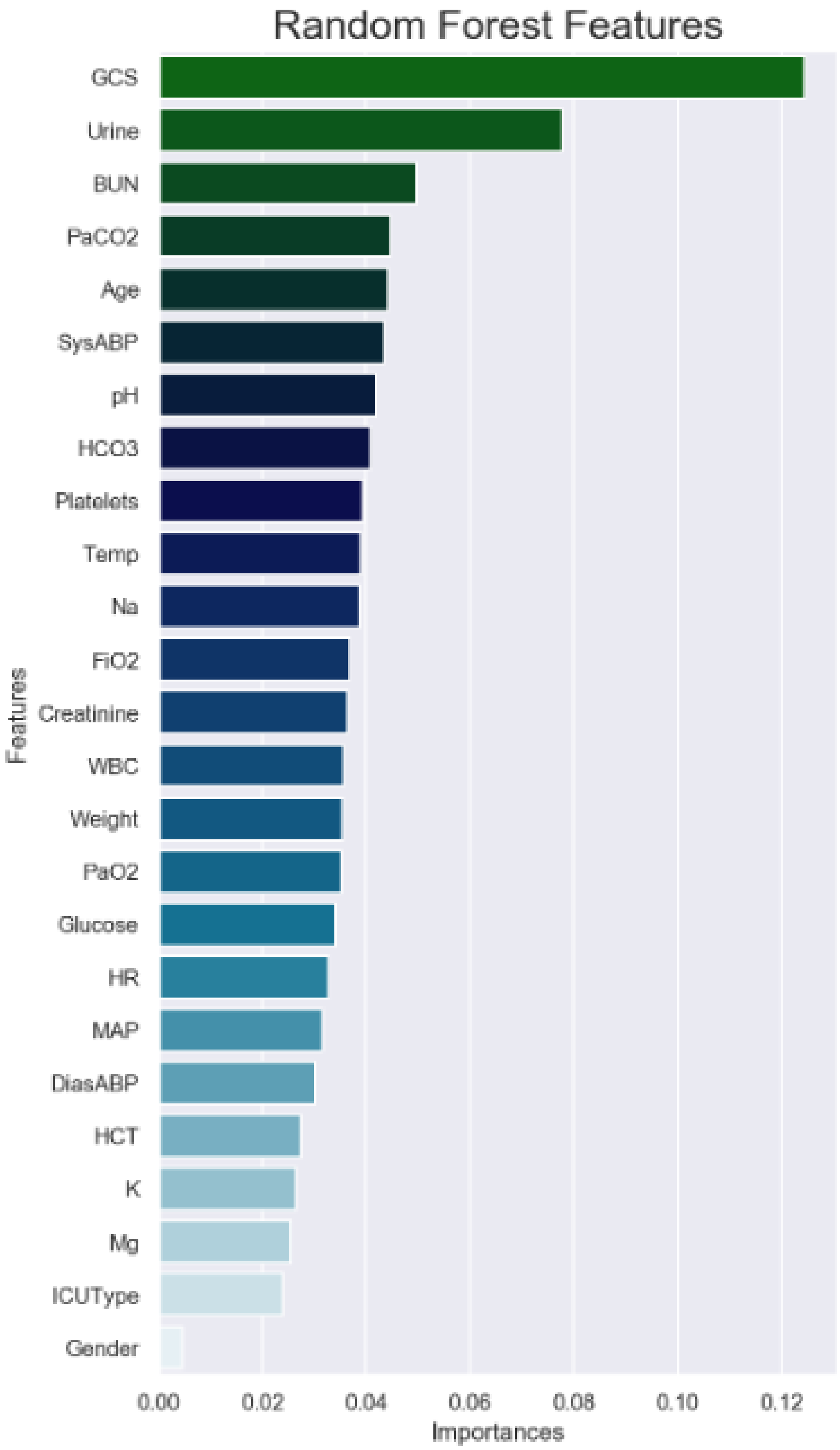
First we tried using Recursive Elimination technique of sklearn, to obtain reducing the feature space to 14 features. But this technique was not as beneficial as manual feature selection from OLS and Lasso Regression parameter coefficients and it provided better results on the test set.

OLS Regression Results

Dep. Variable:	Outcome	**R-squared:**	0.252
Model:	OLS	**Adj. R-squared:**	0.216
Method:	Least Squares	**F-statistic:**	6.972
Date:	Tue, 05 Nov 2019	**Prob (F-statistic):**	1.23e-22
Time:	03:52:30	**Log-Likelihood:**	-288.67
No. Observations:	609	**AIC:**	635.3
Df Residuals:	580	**BIC:**	763.3
Df Model:	28		
Covariance Type:	nonrobust		



Random Forest Feature Importance



Model Selection:

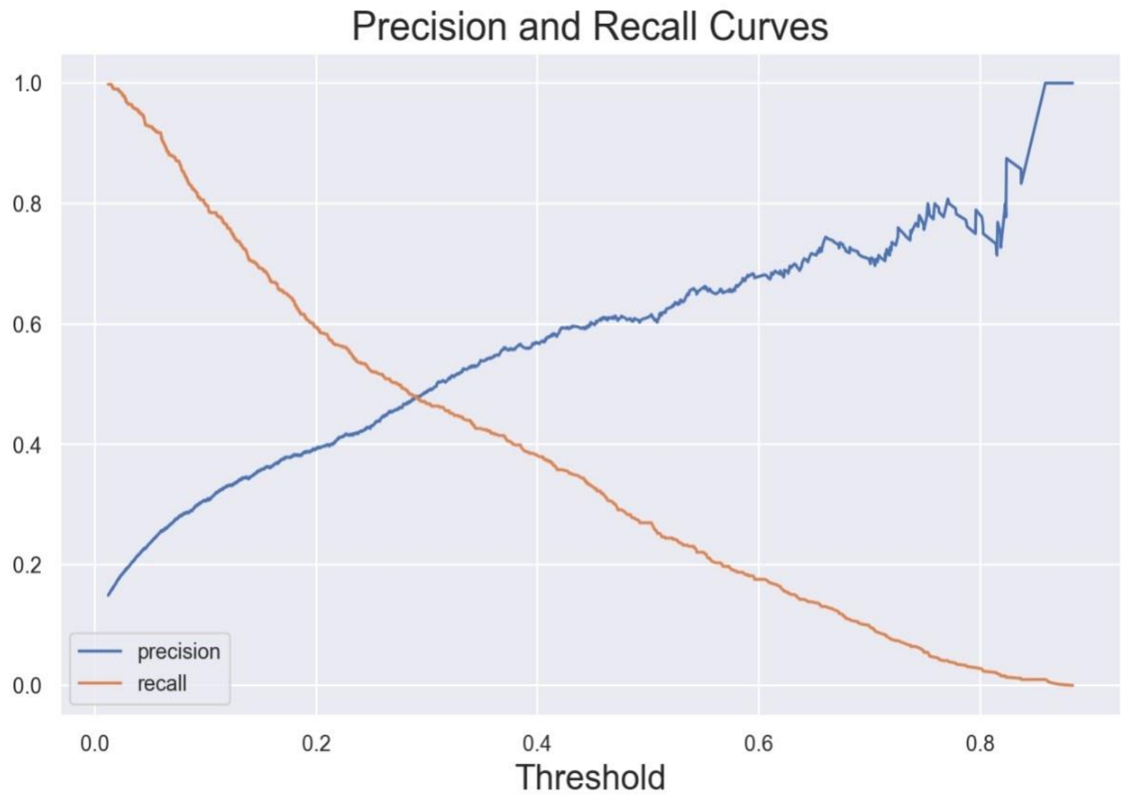
Our baseline model included Logistic Regression, Bernoulli's Naïve Bayes showing satisfying results on validation set. Further improving the model for suitable results, we found best models with XGBoost and Random forest (hyperparameter tuned).

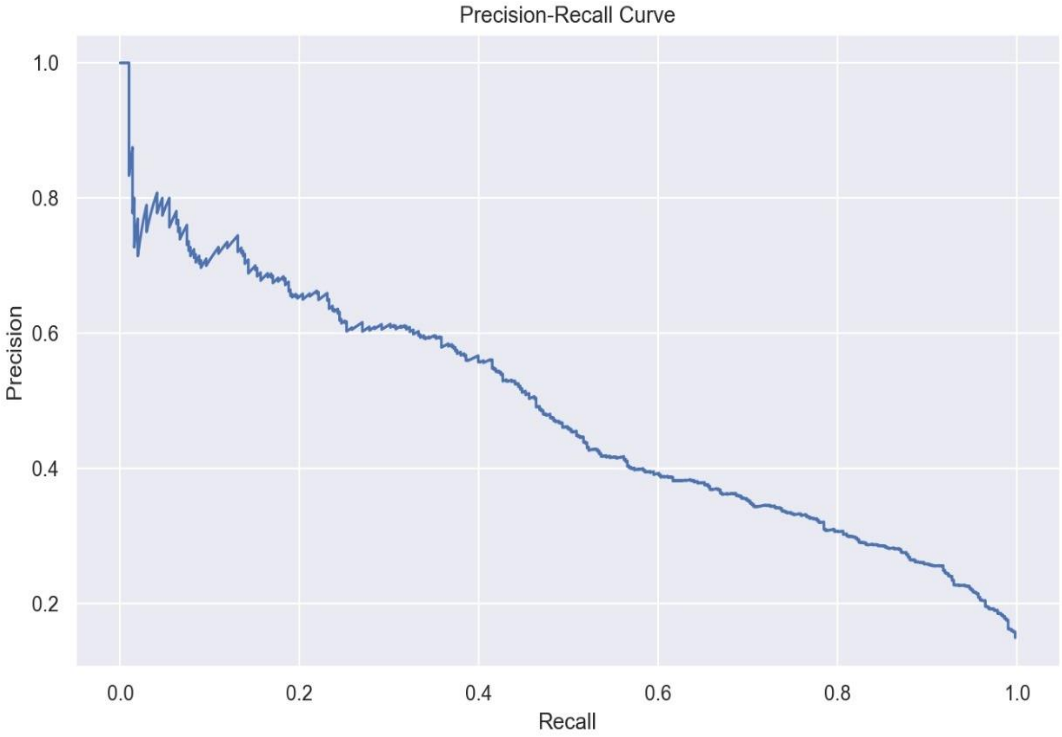
Our major concern for building a survival prediction model, was to consider the sensitivity metrics because of the kind of problem we are tackling. The Recall value can help us understand how many patient's who actually have high chances of not surviving (1: Deceased class) are being classified correctly. Since, if we classify them as (0: Surviving), we would not be doing justice to our ranking system, and patient who is not so much into the critical stage will end up ranking higher. Other concern was the reduction of False Positive Rate which ensures better user (hospital) experience.

[For viewing code, you can log on to our webapp deployed under the ***Model Button***:

http://ec2-100-27-2-78.compute-1.amazonaws.com:5000/]

PR Curve of Random Forest: Threshold = 0.311

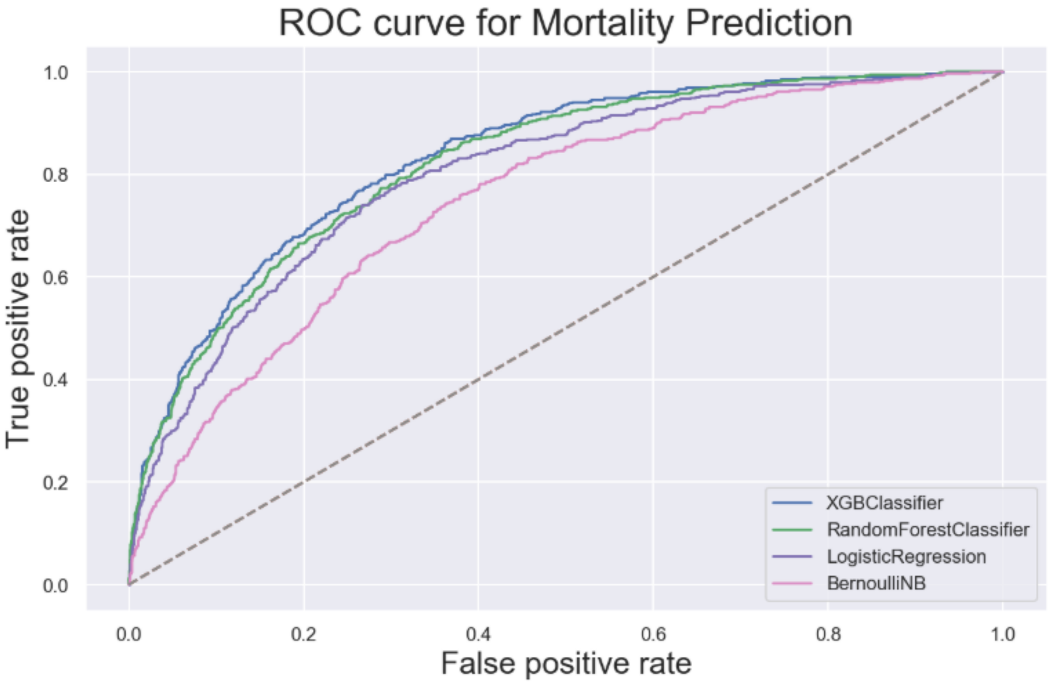


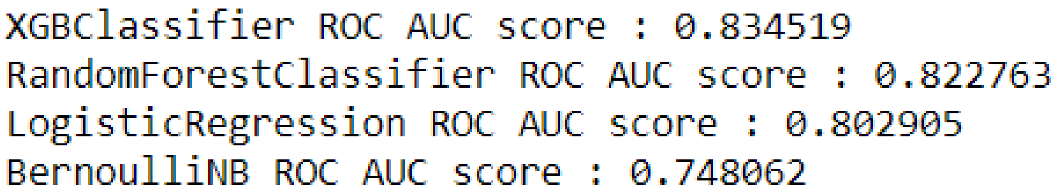


Precision v/s Recall of Random Forest

As we can see, a steady decline in the Recall Value as precision goes down

ROC Curve of all our models:





```
XGBClassifier ROC AUC score : 0.834519
RandomForestClassifier ROC AUC score : 0.822763
LogisticRegression ROC AUC score : 0.802905
BernoulliNB ROC AUC score : 0.748062
```

ROC-AUC Score for XGBoost rank the highest, following Random Forest.

Hence winner in a landslide is XGBOOST

Deployment

Further we built a **Flask** app, which satisfactorily predicted the probability of survival and the length of stay for each patient’s ID.

We then went on to deploy it on AWS EC2 server, powered by Amazon Web Services.

You can view our web app here:

http://ec2-100-27-2-78.compute-1.amazonaws.com:5000/

[For more details of the web app, please refer to execution_guide.pdf]