Hospital Mortality Prediction
(Milestone 4)

After feature engineering in milestone 3, we added four new features to our dataset:

1. 'anion_bicc'
2. 'ren_un'
3. 'leuko_neutri_baso_lympho'
4. 'age_bin'

Now we proceed by removing 'ID' column from this data-set as this feature will not add any value in training our model.

So our final revised data-set will look something like:

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1177 entries, 0 to 1176
Data columns (total 56 columns):
 #   Column                  Non-Null Count  Dtype
---  ------                  --------------  -----
 0   group                   1177 non-null   int64
 1   outcome                 1177 non-null   int64
 2   age                     1177 non-null   int64
 3   gendera                 1177 non-null   int64
 4   BMI_cat                 1177 non-null   int64
 5   hypertensive            1177 non-null   int64
 6   atrialfibrillation      1177 non-null   int64
 7   CHD with no MI          1177 non-null   int64
 8   diabetes                1177 non-null   int64
 9   deficiencyanemias       1177 non-null   int64
 10  depression              1177 non-null   int64
 11  Hyperlipemia            1177 non-null   int64
 12  Renal failure           1177 non-null   int64
 13  COPD                    1177 non-null   int64
 14  heart rate at           1177 non-null   int64
 15  Pulse rate cat          1177 non-null   int64
 16  Sys_cat                 1177 non-null   int64
 17  Diastolic               1177 non-null   int64
 18  respiratory cat         1177 non-null   int64
 19  temp_cat                1177 non-null   int64
 20  SP O2                   1177 non-null   int64
 21  urine_cat               1177 non-null   int64
 22  hemocrit_cat            1177 non-null   int64
 23  RBC_Cat                 1177 non-null   int64
 24  mch_cat                 1177 non-null   int64
 25  mchc_Cat                1177 non-null   int64
 26  mcv_cta                 1177 non-null   int64
 27  rdw_cat                 1177 non-null   int64
 28  leukocytes_cat          1177 non-null   int64
 29  platelets_cat           1177 non-null   int64
 30  neutriphil_cat          1177 non-null   int64
 31  Basophil_cat            1177 non-null   int64
 32  Lympho_cat              1177 non-null   int64
 33  PT_cat(sec)             1177 non-null   int64
 34  INR_cat                 1177 non-null   int64
 35  NT_cat                  1177 non-null   int64
 36  CK_cat                  1177 non-null   int64
 37  Creatinine_cat          1177 non-null   int64
 38  UN_cat                  1177 non-null   int64
 39  Glu_cat                 1177 non-null   int64
 40  potas_cat               1177 non-null   int64
 41  sodium_cat              1177 non-null   int64
 42  cal_cat                 1177 non-null   int64
 43  chloride_cat            1177 non-null   int64
 44  anion_cat               1177 non-null   int64
 45  Mag_cat                 1177 non-null   int64
 46  ph_cat                  1177 non-null   int64
 47  Biccarbon_cat           1177 non-null   int64
 48  metcat                  1177 non-null   int64
 49  lactic_cat              1177 non-null   int64
 50  pco2_cat                1177 non-null   int64
 51  ef_cat                  1177 non-null   int64
 52  anion_bicc              1177 non-null   int64
 53  ren_un                  1177 non-null   int64
 54  leuko_neutri_baso_lympho 1177 non-null  int64
 55  age_bin                 1177 non-null   float64
dtypes: float64(1), int64(55)
memory usage: 515.1 KB
```

Now we proceed with handling the imbalanced data-set.

We used **SMOTE(**Synthetic Minority Over-sampling Technique) to over sample the minority class by generating synthetic examples.

Before applyting SMOTE

```
y_train.value_counts()

0    771
1    111
Name: outcome, dtype: int64
```

After applying SMOTE

```
y_train.value_counts()

0    771
1    771
Name: outcome, dtype: int64
```

Now we apply standard scaler to our data and proceed by fitting multiple models to it.

There are various metrics which can measure the performance of a classification model.

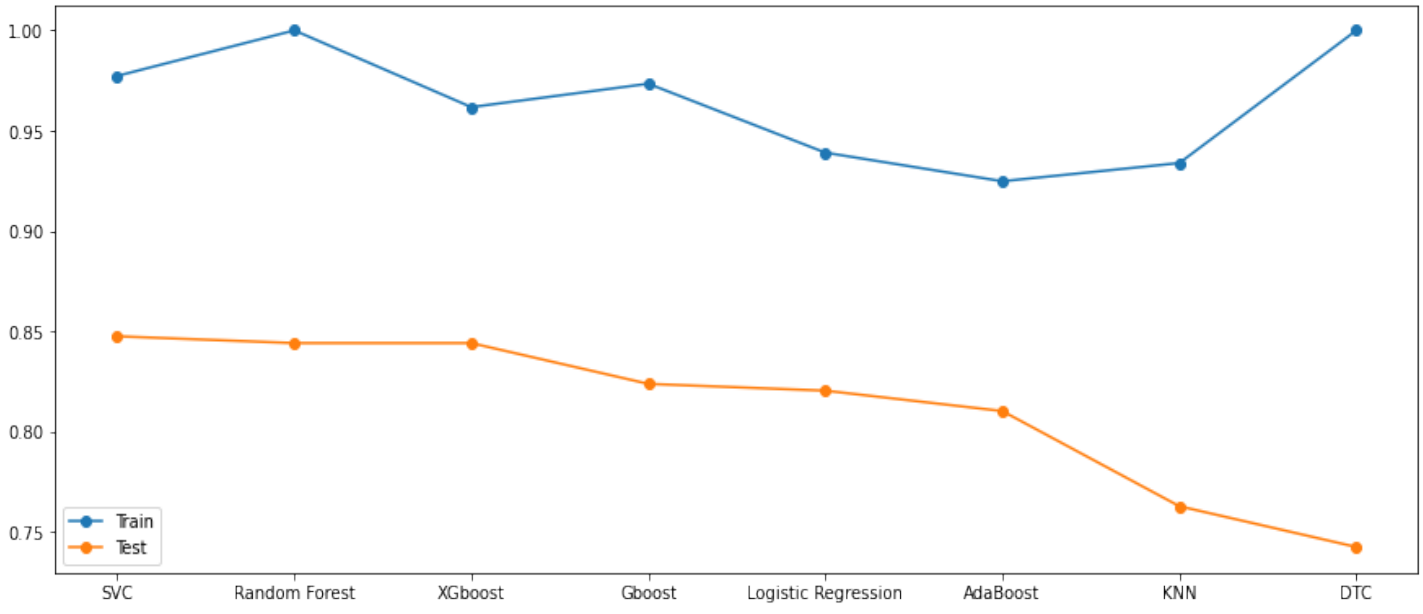
1. Accuracy
2. F1-score
3. ROC-AUC score
4. Precision
5. Recall

Here we will be using Accuracy.

Displaying the result in descending order of Accuracy.

	Model Name	Accuracy	F1-Score	ROC-AUC Score	Precision	Recall	Train Acc	Test Acc
2	SVC	0.847458	0.366197	0.615174	0.565217	0.270833	0.977302	0.847458
3	Random Forest	0.844068	0.323529	0.596365	0.550000	0.229167	1.000000	0.844068
6	XGboost	0.844068	0.477273	0.680288	0.525000	0.437500	0.961738	0.844068
5	Gboost	0.823729	0.422222	0.651358	0.452381	0.395833	0.973411	0.823729
0	Logistic Regression	0.820339	0.417582	0.649334	0.441860	0.395833	0.939040	0.820339
7	AdaBoost	0.810169	0.416667	0.651653	0.416667	0.416667	0.924773	0.810169
4	KNN	0.762712	0.453125	0.698844	0.362500	0.604167	0.933852	0.762712
1	DTC	0.742373	0.355932	0.619560	0.300000	0.437500	1.000000	0.742373

Plot of accuracy of different models:

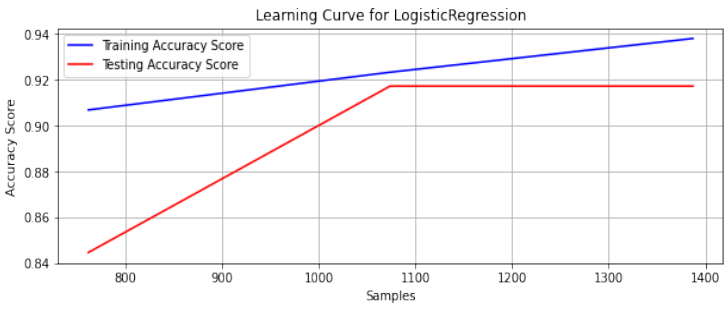


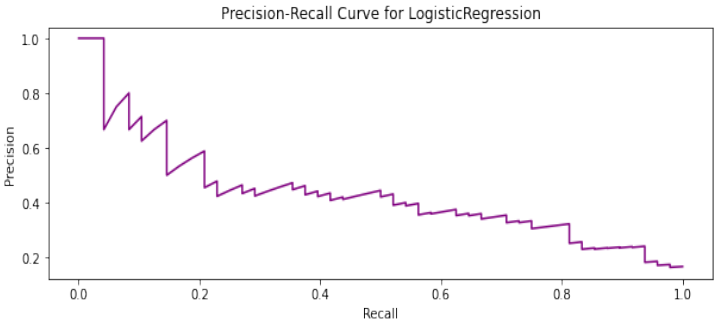
Performing cross validation:

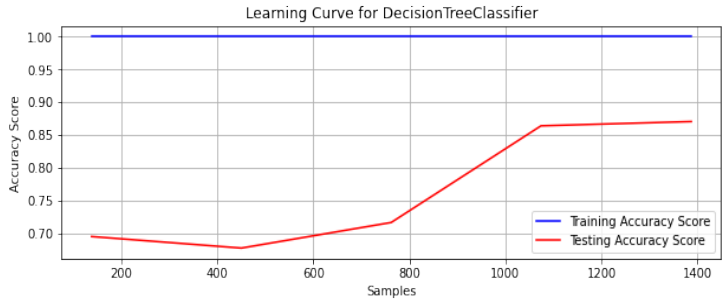
	Model Name	CV Accuracy	CV STD
3	Random Forest	0.946992	0.073904
2	SVC	0.933452	0.104271
5	Gboost	0.932111	0.086895
6	XGBoost	0.924964	0.083950
0	Logistic Regression	0.918513	0.097897
7	AdaBoost	0.906824	0.091499
4	KNN	0.905325	0.020328
1	DTC	0.872451	0.085592

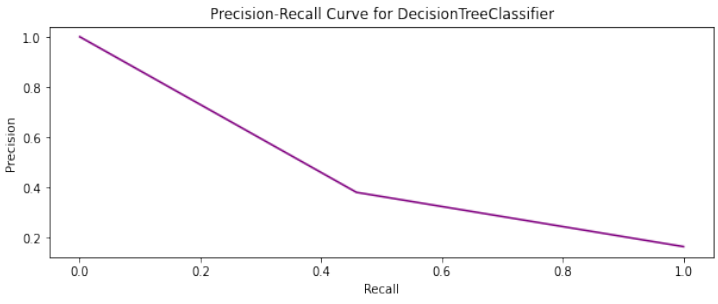
Plot of accuracy score on train and test data to see for overfitting and underfitting:

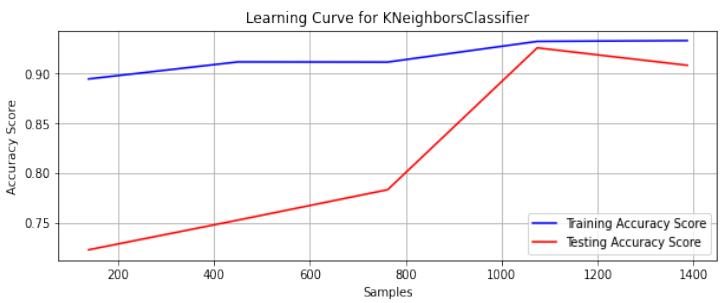
Plot of precision-recall curve for all the models

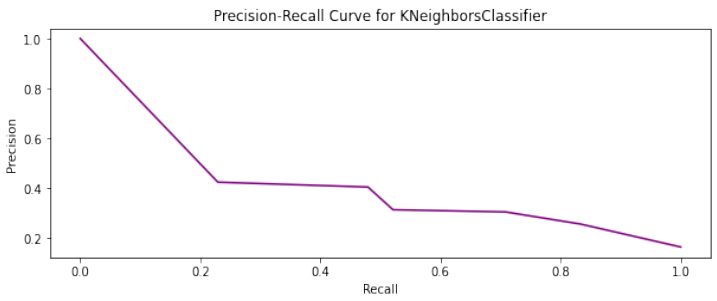


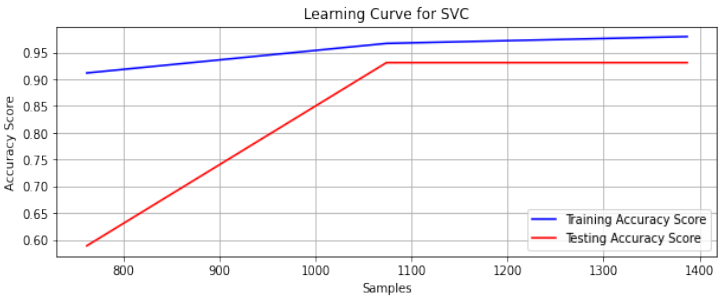


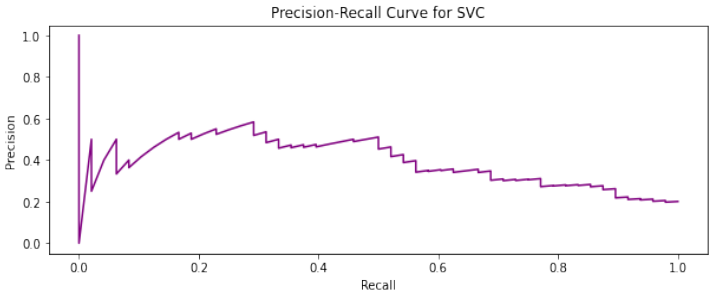


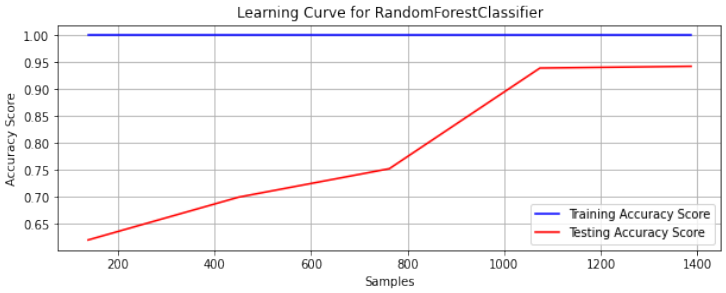


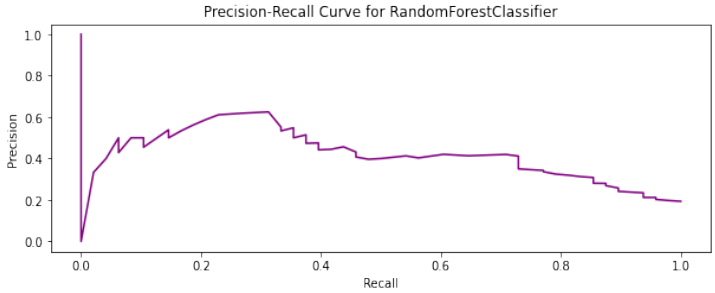


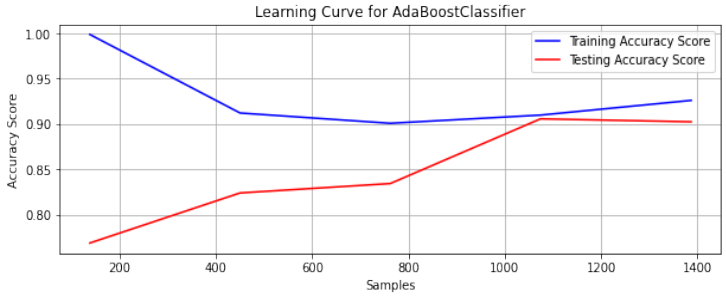


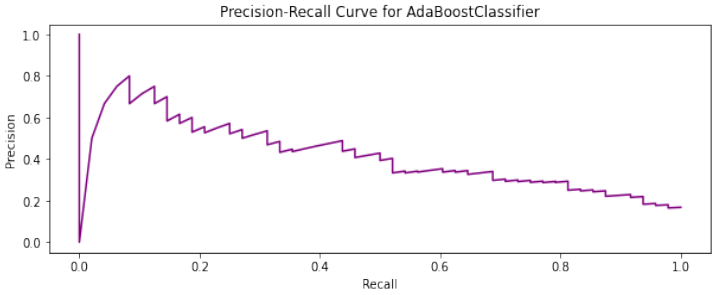


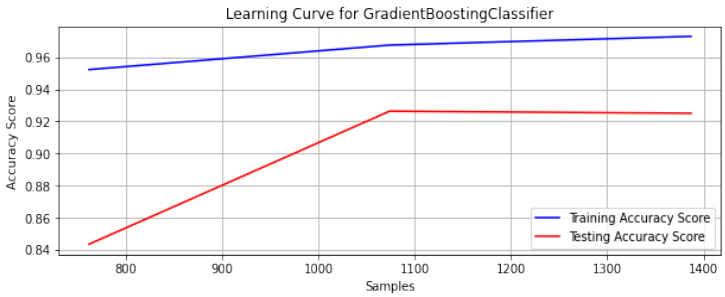


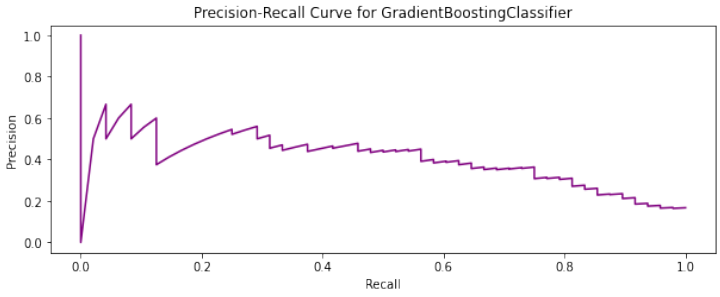


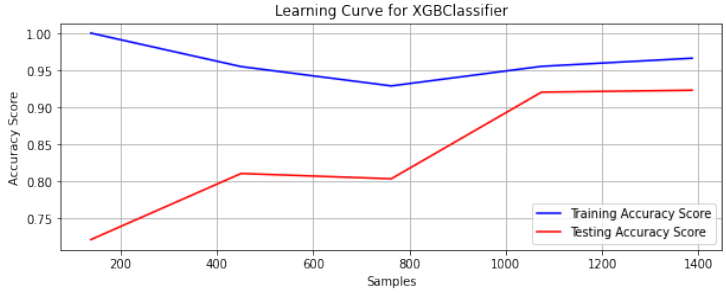


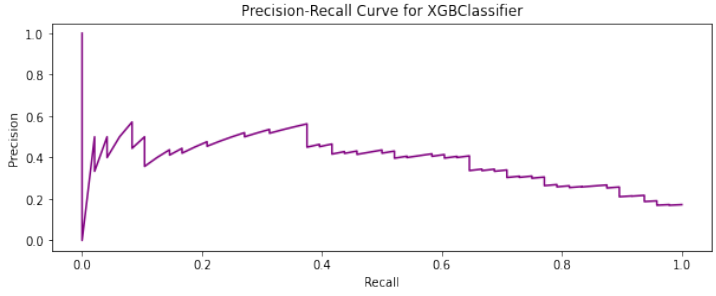












Top 3 best performing models:

1. Random Forest Classifier
2. SVC
3. Gradient Boosting Classifier

Applying Grid Search CV on top 3 models and getting the best parameters:

```
models: SVC
best parameters : {'C': 0.5, 'gamma': 'scale'}
models: GBC
best parameters : {'learning_rate': 0.1, 'max_depth': 9, 'n_estimators': 250}
models: RFC
best parameters : {'max_depth': 14, 'max_features': 4}
```

Feeding the parameters and getting the optimal output:

	Model Name	CV Accuracy	CV STD
2	Random Forest	0.950888	0.076642
1	Gboost	0.943104	0.073611
0	SVC	0.934755	0.108172

Finding top 15 important features with Random Forest model:

```
ren_un                      0.060440
leukocytes_cat              0.045026
anion_bicc                  0.044097
anion_cat                   0.040983
cal_cat                     0.040936
rdw_cat                     0.040808
age                         0.039150
age_bin                     0.038277
Lympho_cat                  0.036392
leuko_neutri_baso_lympho    0.028874
chloride_cat                0.028637
Creatinine_cat              0.025725
deficiencyanemias           0.024890
sodium_cat                  0.023671
respiratory cat             0.022231
dtype: float64
```

Visualization of top 15 important features with Random Forest model: