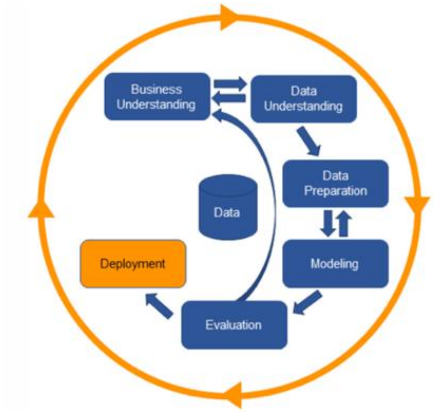
Submitted by:
Piu Mallick (pim16@pitt.edu) & Debdas Ghosh (deg107@pitt.edu)

INFSCI 2160 – Data Mining
Python Lab

AUC Train: 0.87 AUC Test: 0.84 Best Modeling Technique/Algorithm chosen: LightGBM

A predictive analysis algorithm will be developed using **CRISP-DM**, which is a robust and well-proven methodology. It stands for **C**ross-**I**ndustry **P**rocess for **D**ata **M**ining and it provides a structured approach to planning a data mining project. Various stages of **CRISP-DM** are discussed below:

1. Business Understanding



A. **Background**: A real and anonymized dataset has been used provided by UPMC. It is regarding surgical procedures between June 2017 and June 2018. The data is already divided into training and test populations. A data dictionary is also provided along with the dataset.

B. **Goal**: The goal of the **Python Lab** is to develop an algorithm that can accurately predict patients that are at risk of a '**long**' length of a stay post-surgery.
An abnormally long **LOS** or **Length of Stay** can be defined in numerous ways. However, for prediction purpose, a **long LOS** can be defined as **greater than 5 days post-surgery**.
Hence, turning our predictive analysis into a **binary classification problem**, which defines **long LOS** as **1** or otherwise **0**.

2. Data Understanding

A. Dataset is loaded and its shape is determined – it has **80000** rows and **292** columns/features.
B. The statistics of all the numeric values in the dataset are checked.
C. Distribution of NULL values are checked throughout the database. Most of the columns have NULL values.
D. The datatypes of all the features are checked and it is seen that they are a combination of numeric of numeric and object datatypes.

3. Model Selection

Since, this is a **Classification problem** and we have **a sparse matrix data**, it is **best to try out gradient boosted trees models** for prediction. So, the obvious choices are **XGBoost**, **CatBoost** and **LightGBM** as these are the most popular gradient boosting trees algorithms. However, for prediction purpose, all the datatypes should be converted to numeric. This will be discussed in the '**Data Preparation**' section.

4. Data Preparation

A. A copy of the original dataset is made, in case if there is a need to look back to the original data.
B. All the object datatypes have been converted to numerical values using **LabelEncoder**.

C. The target variable, which is of numeric datatype, is converted into binary – **1** as **long LOS** (LOS > 5), or otherwise **0**.

D. The dataset is then split into **train** (**80%** of the data) and **test** (**20%** of the data). Train dataset further split into validation set in 80:20.

5. Modeling

As already mentioned, given the dataset and datatypes of the features in the dataset, we thought of implementing the following **Gradient Boost Decision Tree - modeling techniques** for the prediction-analysis purpose:

A. CatBoost

Reasons for choosing CatBoost:

CatBoost can be used for solving problems, such as regression, classification, multi-class classification and ranking. It has pre-build metrics to measure the accuracy of the model. It is an innovative algorithm for processing categorical features. For datasets with categorical features, the accuracy would be better in comparison to other algorithms; hence this is one of the reasons for choosing this algorithm for the prediction purpose.

B. XGBoost

Reasons for choosing XGBoost:

"XGBoost can be termed as a scalable and accurate implementation of gradient boosting modeling techniques and it has proven to push the limits of computing power for boosted trees algorithms as it was built and developed for the sole purpose of model performance and computational speed." It is a decision-tree based ensemble ML algorithm that uses a gradient boosting framework. One of the main reasons for choosing this algorithm is because of the dataset size – for small-to-medium structured/tabular data, decision tree-based algorithms are considered best.

C. LightGBM

Reasons for choosing LightGBM:

LightGBM can be defined as a gradient boosting framework that uses tree-based learning algorithm. It grows tree leaf-wise i.e. it grows vertically as compared to other algorithms which grow horizontally (level-wise growth). Although leaf-wise growing is more prone to overfitting (in case of smaller datasets) that's why it is advised to use LightGBM for large datasets (and, the UPMC's dataset is comparatively bigger). Also, while growing leaf-wise the loss can be reduced more effectively. Light GBM can handle the large size of data and takes lower memory to run. Another reason of Light GBM is because it focuses on accuracy of results, also supports GPU learning.

6. Evaluation

Evaluation of the 3 models based on **AUC accuracy** is summarized below:

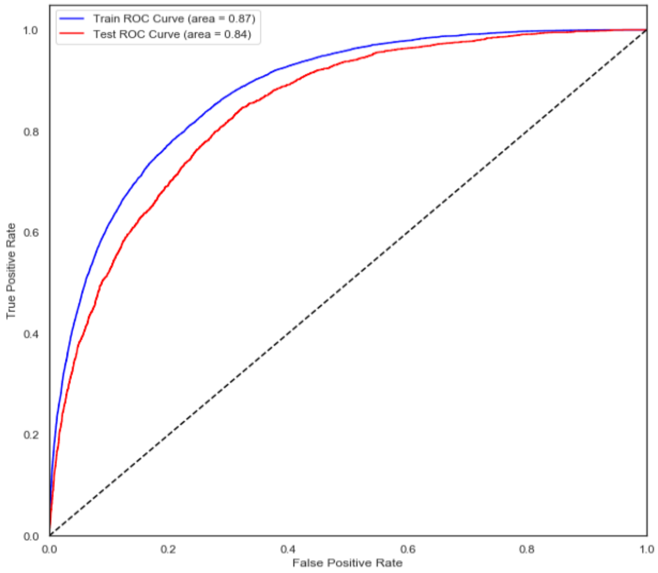
Models	**Train AUC**	**Test AUC**	**Train-test AUC Difference**	**Evaluation**
CatBoost	**0.7694**	**0.7633**	**0.0061**	Fit-fine
XGBoost	**0.85349**	**0.82854**	**0.02495**	Fit-fine
LightGBM	**0.8749**	**0.8407**	**0.0342**	Fit-fine

Hence, among the three chosen **Gradient Boost Decision Trees**, **LightGBM** gives **the best accuracy on AUC** and that too with a difference of **0.03**, which makes the model '**fit-fine**'.

However, both **XGBoost** and **CatBoost** have lower train-test accuracy difference, we decided to choose **LightGBM** as it has **highest AUC.**

Interpretation of raw results:

The **AUC-ROC** curve for **LightGBM** model is shown, which gives the best result among the 3 chosen models.



Confusion Matrix for **LightGBM** model has also been created:

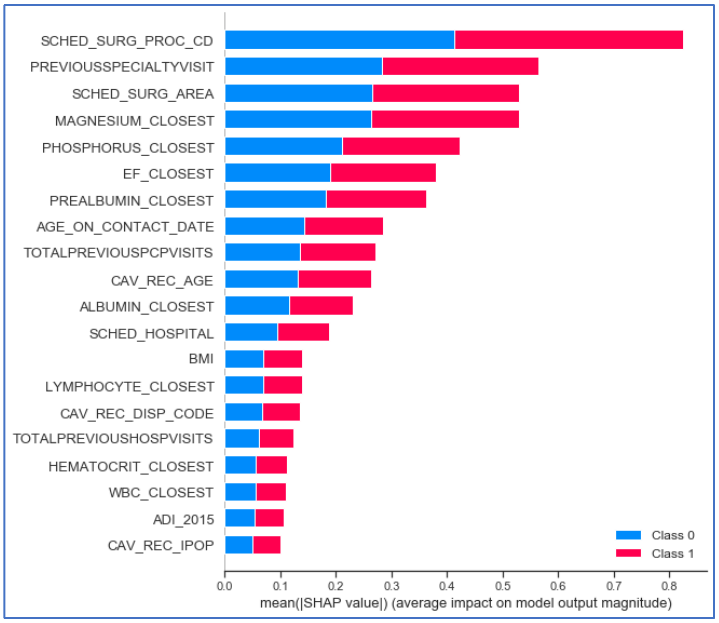
N = 12800	**Predicted 0**	**Predicted 1**
Actual 0	9410	607
Actual 1	1636	1147

True Positives = 1147
True Negatives = 9410
False Positives = 607
False Negatives = 1636

Feature Selection Importance:

"**SHAP** (**SHapley Additive exPlanations**) is used to explain the prediction of an instance x by computing the contribution of each feature to the prediction. The SHAP explanation method computes Shapley values from coalitional game theory. The feature values of a data instance act as players in a coalition. Shapley values tell us how to fairly distribute the "payout" (= the prediction) among the features."

SHAP is used in our best model to achieve the aforesaid purpose:



The predictions are in equally distributed among class 0 (i.e. length of stay <=5) and 1(i.e. length of stay > 5). Hence, we can conclude that there is **equal confidence** in the models for class 0 and class 1. Most important variable that impacted our model is **SCHED_SURG_PROC_CD**.

7. Deployment

We would definitely choose and most likely proceed for deployment with **LightGBM model** as we got the **best score** from that model. We would follow the following steps as the part of the integration process:

- Validation of the model
- Choosing appropriate model
- Scoring the model
- Following necessary steps for integration with the application
- Monitoring the application
- Get feedback from various surveys and observations
- Finding ways for improvising the model accuracy or finding other models which could produce more accuracy.

Nevertheless, since this model determines the criticality of the patient's condition post-surgery (based on LOS <=5 or >5), we are dubious about patient's life-risk. In that case, if given any opportunity, we would like to divide the risk factor as **Low**, **High** and **Medium**.

8. Proof of Deployment

We have built a **CSV** file containing **ID1** and **LOS_PROB** columns, where **ID1** is the unique identifier of unseen cases given in the "**PYTHON_LAB_DF_TEST_2.csv**" file. **LOS_PROB** is the probability of **LOS > 5** for each record. For very high probability values like greater than or equal to **0.5**, we may consider the patient might not have recovered enough to get discharged and probably will stay longer in the hospital post their surgery. The data may be verified from the train data set for the accuracy.