**Readme**

1. Make sure we have Python 2.7 edition and Anaconda is installed in the machine to execute the codes.
2. Install Rpy2 package by running the below script from the Anaconda command prompt line: -

“conda install -c joshadel rpy2=2.3.9”

1. Copy all the files from source folder to the machine
2. Open “**read\_data.py**” file in Notepad++ and go the function read\_data function and change the **‘path’** variable to the location where you have copied the Excel files containing the data and associated features\_lists.
3. Execute the “Main file” to run the prediction algorithm.

**Files Name**

1. **read\_data.py**: - It contains functions which will read the data from the mentioned folder location and stored it as a data frame and impute the data based on some pre-defined medical conditions.
2. **preprocessing**.**py: -** It contains functions which are used to pre-process the raw data and making it suitable to train the model which will help for prediction. The following functions are implemented under this file

**gen\_proc\_data ():** It call the suitable functions based on the feature type of each column of raw data.

**Outlier\_detect ():-** Detect the outliers in the data and impute it with mean values using some formulas.

**Clean\_categorical (): -** Transform categorical data and impute missing values with ‘MISSING’ and use the concept of minimum thresholding for count for each categorical value.

**TrainProcedureFeature ():-** Use Log transformation for converting procedure features to numeric form.

**TrainCategoricalFeature ():-** Use Log Transformation for converting categorical features to numeric form.

**Get\_procedure\_feature:** - to obtain the dictionary which consist the mapping of each procedure type to its corresponding log transformation performed in function Trainprocedurefeature().

**Get\_Categorical\_feature:** - to obtain the dictionary which consist the mapping of each category type to its corresponding log transformation performed in function Traincategoricalfeature().

1. **run\_gams\_model.py**: - The different functions inside this file are used to generate gam models and get the best model based on different metric. The functions are: -**gams\_for\_Individual\_run (): -** This function splits the data into training and test data in ratio of 70:30 and generate the GAM model using training dataset. This model is then tested on test data set to get the accuracy and AUC of the model GAM models.

**get\_all\_gams\_model ():** - This function is used to 50 different GAM model by shuffling the dataset before splitting it and to store corresponding AUC of each run.

**get\_best\_model ()**: This function returns the best model out of all generated 50 models based on maximum AUC value.

**ROC\_parameters ()-** This function calculates the metrics Accuracy, PPV, NPV, Sensitivity, Specificity, Yoden Index for a particular threshold value

**calculate metric ()-** Calculates metric corresponding to different threshold value ranging from 0.01 to 0.98

1. **main**.**py**: - This is the Entry point of the Entire code and will call functions which is present in other python files.

**PS:-**Further Details about the function can be found in the below document

**FLOW DIAGARAM**

Run\_gams\_model.py

Preprocessing.py

Main.py

Read\_Data.py