Documentation

# Introduction

This Document defines the over-all flow of predicting and categorizing the risk probability associated with a patient operating during pre-op stage using different machine learning algorithms and data analysis. This document also serves as an outline for understanding different transformations applied on different set of geographical and clinical features of a patient to make the data more understandable and scalable. It will also help on identifying key features and values of probabilities threshold and categorizing the patient based on it.

# Technology stack

Right Now we have fetching all our data from Excel files and have written our codes in python to check accuracy and consistency of the system. Going forward we will have Database as our source of data.

**Big Question: - Why Python?**

We have converted all the existing R code to Python code to achieve below mentioned advantages’: -

1. The strength of its core libraries (NumPy, SciPy, pandas, matplotlib)
2. High Productivity for prototyping
3. Useful for building scalable and reusable system and easy interactions with Big Data (Hadoop, Spark) and both relational & non-relational databases.
4. Runs pretty fast.

In order to create a bigger ecosystem of forecasting risk of a patients using medical and Surgeon knowledge in real time above factors played an important role. Thus to maintain dynamic and fast result prediction we are using python.

# Description

Sequencing of process end to end: -

1. Data Extraction and feature selection sub-setting
2. Data Cleaning
3. Data Preprocessing
4. Splitting into Training Data and Test Data
5. Applying gam model and Predicting risk probabilities
6. Categorizing patients based on probabilities

**Data Extraction and feature selection sub-setting**

We are collecting data from EMR of a patient and dumping it to excel files. This data includes 285 demographic, socio-economic, administrative, clinical, pharmacy and laboratory variables as the superset contributing as the features for prediction.

Number of input features is quite high so we subset it using by picking out predictors which have more than 2% statistical significance.

Features like (icancer, CKD\_cor, esrd\_corr etc.) are removed by using above criteria

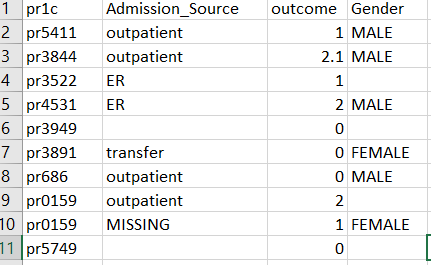
**Data Cleaning & Pre-Processing**

Missing observations were imputed before model building using automated algorithm.

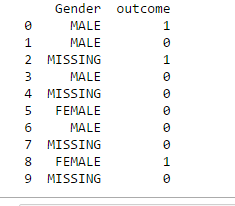
Function Name**: - Preprocessing.clean\_categorical**

For nominal variables with missing entries, a distinct “missing” category was created.

For Example, **Gender** Column below**, Input: -**



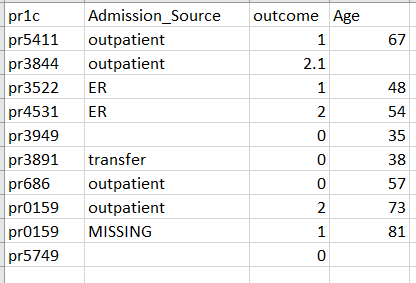
**Output will be: -**

****

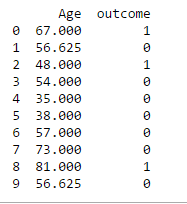
Function Name**: - Preprocessing.** **outlier\_detect**

For continuous variables, the mean value for a given variable was used for imputation.

For example, **Age** column for below table



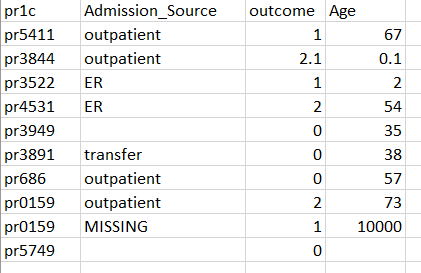
The Output of above input will be: -



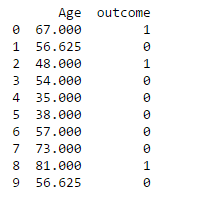
For all variables we developed set of automatic rules for the removal of outliers that were considered unreasonable observations by medical experts. For continuous variables, observations that fell in the top and bottom 1% of the distribution were considered as outliers.

Identified outliers are being treated as missing values and are were imputed using the algorithm. (R function, data\_pre\_processing)

For Example, we have outliers in below **Age** variables: -



The Output for above will be: -

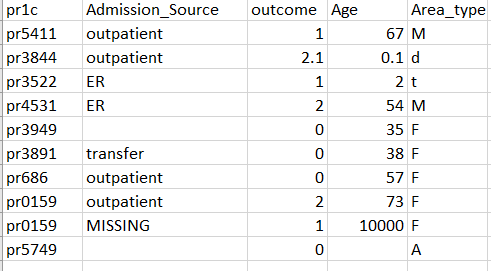


**Optimization of Categorical Features**

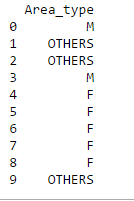
Function Name**: - Preprocessing.train\_categorical\_feature**

For nominal variables and categorical features with more than two levels (Table 1), if the count of each category is less than the threshold value then the algorithms redefines it to ‘OTHERS’ Category.

For Example, **Area\_type** column in below input

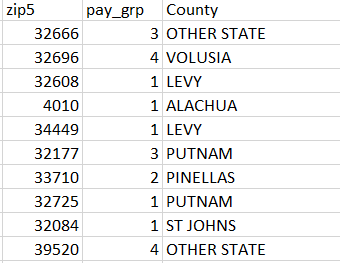


Output of the same will be: -

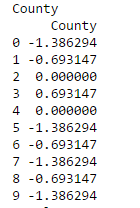


We also log transform the categorical data in pre-processing stage to get a numerical number from each category type of that particular column.

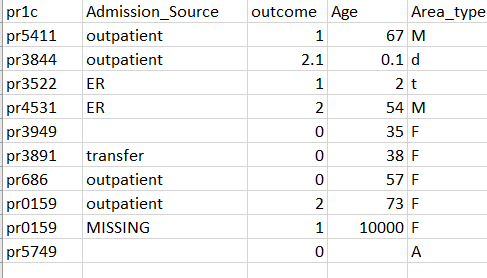
For Example, **Country** feature in below input: -



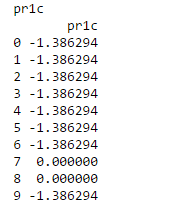
Will be get converted into following output: -



We also similarly transformed the ‘procedure’ data which is also categorical using log transformation mentioned in the technical specification document. For example, **‘pr1c’** feature of below input



Will be get converted into output: -



**Splitting into Training Data and Test Data: -**

The original data is split into 70% train data and 30% test data. We used the 70% train data to our train our GAM model by tuning the parameters of the model using 10 fold cross-validation to get a proper balance between bias and variance.

**GAM MODEL: -**

The pre-processed data is passed to **Gam**\_**Model**\_**Run** for the building the model from the training data and predicting the risk probabilities associated with each patient.

**Categorizing patients based on probabilities: -**

For **“**cv\_comp\_new”

|  |  |  |
| --- | --- | --- |
| Risk Group | Low\_Threshold\_Probability | High\_Threshold\_Probability |
| Low | 0.0 | 0.07 |
| Medium | 0.07 | 0.35 |
| High | 0.35 | 1 |

For **“**MV\_comp”

|  |  |  |
| --- | --- | --- |
| Risk Group | Low\_Threshold\_Probability | High\_Threshold\_Probability |
| Low | 0.0 | 0.14 |
| Medium | 0.14 | 0.45 |
| High | 0.45 | 1 |

For **“**30-day mortality\_comp”

|  |  |  |
| --- | --- | --- |
| Risk Group | Low\_Threshold\_Probability | High\_Threshold\_Probability |
| Low | 0.0 | 0.04 |
| Medium | 0.04 | 0.15 |
| High | 0.15 | 1 |

For **“**ICU\_comp”

|  |  |  |
| --- | --- | --- |
| Risk Group | Low\_Threshold\_Probability | High\_Threshold\_Probability |
| Low | 0.0 | 0.30 |
| Medium | 0.30 | 0.60 |
| High | 0.6 | 1 |