# Data Manipulation and Feature Engineering

**Ishan Ambike**

**2/18/2022**

## Introduction:

This problem focuses on Data Manipulation and Feature Engineering. The data for this problem is available on Kaggle. Heritage Health Prize is a competition organized on Kaggle with data provided by Heritage Provider Network (HPN). According to the problem description on Kaggle “More than 71 million individuals in the United States are admitted to hospitals each year, according to the latest survey from the American Hospital Association. Studies have concluded that in 2006 well over $30 billion was spent on unnecessary hospital admissions. Is there a better way? Can we identify earlier those most at risk and ensure they get the treatment they need?”. So, the purpose of the challenge is to predict how many days a person will spend in the hospital next year. With this prediction, the cost of hospitalization should reduce and patients will receive treatment before they are needed to be hospitalized.

## The Dataset:

The data for this problem consists of seven tables. The tables are as follows:

- **Members:** Contains information about the sex and age of a patient.
- **Claims:** Contains detailed information about past hospitalizations, treatments, and medical issues about a member.
- **LabCount:** Contains certain details of lab tests provided to members.
- **DrugCount:** Contains certain details of prescriptions filled by members.
- **DaysInHospital Y2:** Contains claims truncated and days in hospital spent in year 2 by a member.
- **DaysInHospital Y3:** Contains claims truncated and days in hospital spent in year 3 by a member.
- **Target:** Contains claims truncated for year 4. Days in hospital are to be predicted for year 4.

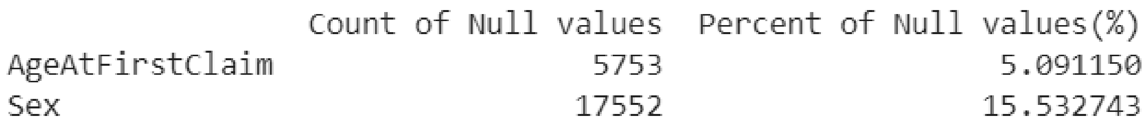
Detailed information about the data can be found on the competition page: https://www.kaggle.com/c/hhp/data

## Preparing the data:

As expected for any real-world data, this data has many missing values. The data also needs to be transformed so that it can be used for prediction. Hence ways should be found to make data suitable for prediction using regression. Feature engineering would be needed to get data that would help create a regression model with the most accurate results. Let’s perform feature engineering on each data table and transform the data to be used for regression.

## Members Table

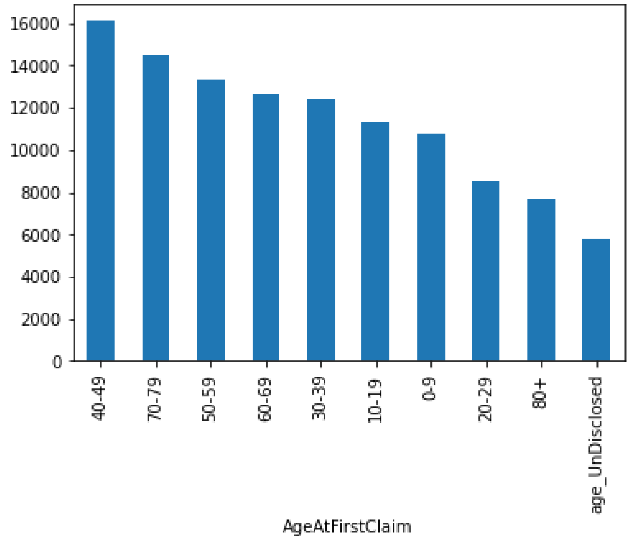
The Members table consists of three columns: MemberID, AgeAtFirstClaim, and Sex. MemberID is unique for each member and does not have missing values. AgeAtFirstClaim and Sex have missing values in them. Let’s see how many values are missing from the entire data.



```
                 Count of Null values  Percent of Null values(%)
AgeAtFirstClaim                  5753                   5.091150
Sex                             17552                  15.532743
```

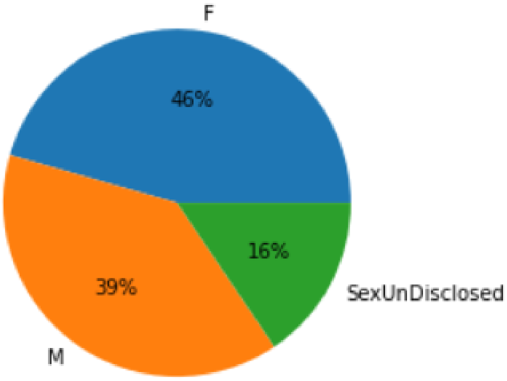
So AgeAtFirstClaim column has 5% values missing and Sex column has 15.5% values missing.

**Transforming AgeAtFirstClaim column**:
Missing values are replaced with the “age_UnDisclosed” value. The frequency of each category is:



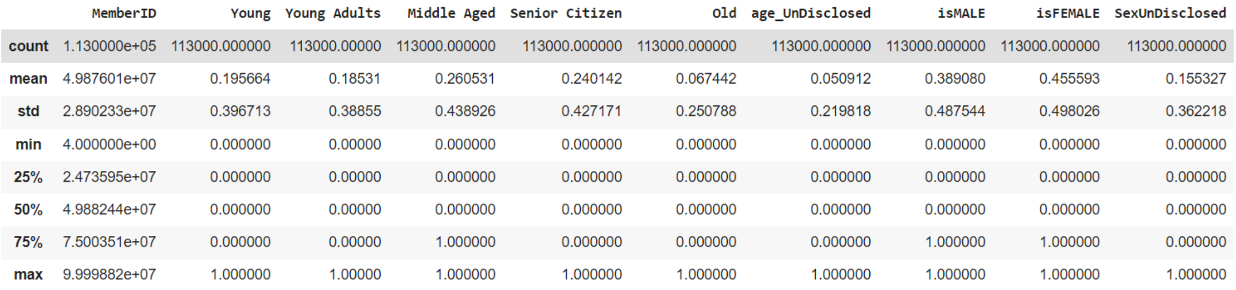
Category “40-49” has the highest occurrence. As the AgeAtFirstClaim has categorical values, they need to transform them to numerical values to be used for prediction. So I have created 5 new columns which will have binary values for age. A column will have 1 if age is within the interval of the column.  The 5 columns are as follows: **Young** (age 0 - 19), **Young Adults** (age 20 - 39), **Middle Aged** (age 40 - 59), **Senior Citizen** (age 60 - 79), **Old** (age 80 +) ,**age_UnDisclosed** (age not available).

**Transforming Sex column**:
The proportion of Sex is:



For sex also I have created 3 columns that would contain binary values. The columns are as follows: **isMALE** (if the member is male), **isFEMALE** (if the member is female), **SexUnDisclosed** (is sex data not available).

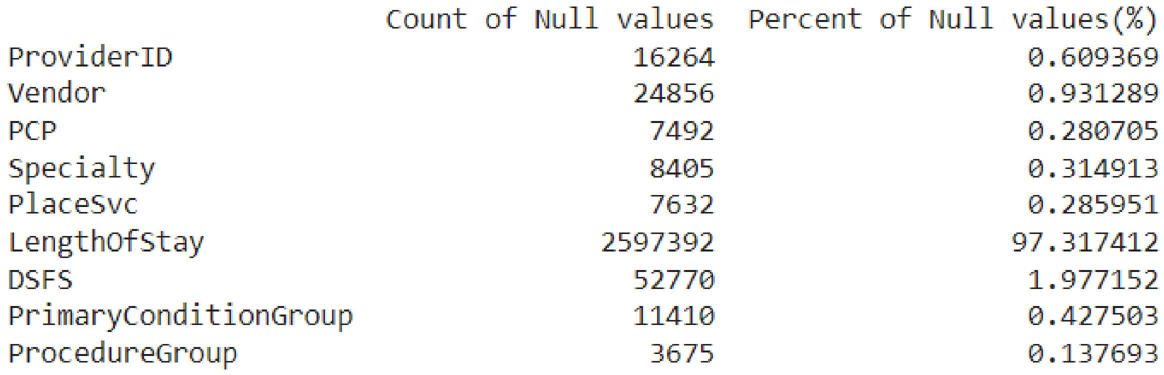
Now the Members table is transformed and does not have any missing value. The table has 113,000 rows and 10 columns.



| | MemberID | Young | Young Adults | Middle Aged | Senior Citizen | Old | age_UnDisclosed | isMALE | isFEMALE | SexUnDisclosed |
|---|---|---|---|---|---|---|---|---|---|---|
| **count** | 1.130000e+05 | 113000.000000 | 113000.00000 | 113000.000000 | 113000.000000 | 113000.000000 | 113000.000000 | 113000.000000 | 113000.000000 | 113000.000000 |
| **mean** | 4.987601e+07 | 0.195664 | 0.18531 | 0.260531 | 0.240142 | 0.067442 | 0.050912 | 0.389080 | 0.455593 | 0.155327 |
| **std** | 2.890233e+07 | 0.396713 | 0.38855 | 0.438926 | 0.427171 | 0.250788 | 0.219818 | 0.487544 | 0.498026 | 0.362218 |
| **min** | 4.000000e+00 | 0.000000 | 0.00000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| **25%** | 2.473595e+07 | 0.000000 | 0.00000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| **50%** | 4.988244e+07 | 0.000000 | 0.00000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 | 0.000000 |
| **75%** | 7.500351e+07 | 0.000000 | 0.00000 | 1.000000 | 0.000000 | 0.000000 | 0.000000 | 1.000000 | 1.000000 | 0.000000 |
| **max** | 9.999882e+07 | 1.000000 | 1.00000 | 1.000000 | 1.000000 | 1.000000 | 1.000000 | 1.000000 | 1.000000 | 1.000000 |

## Claims Table

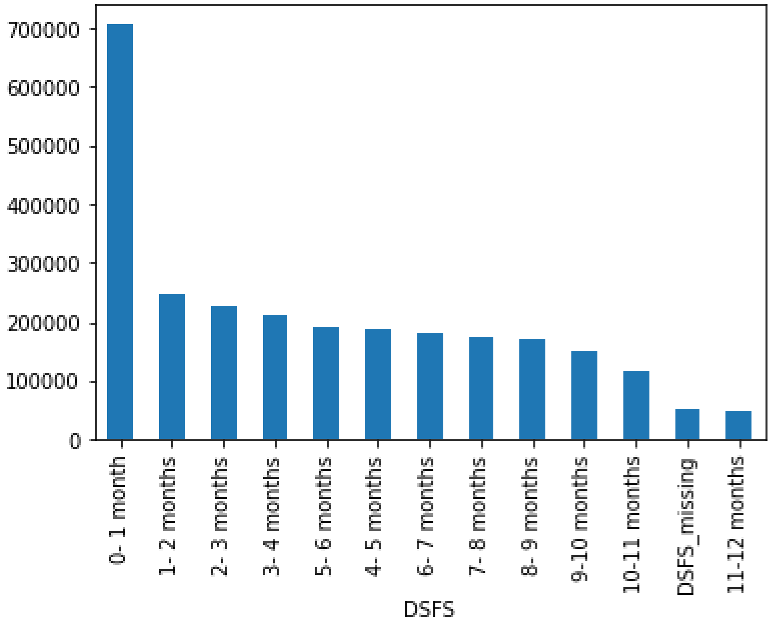
The Claims table has 14 columns. This table too has many missing values. The count and percent of missing values in each column are:



```
                       Count of Null values  Percent of Null values(%)
ProviderID                            16264                   0.609369
Vendor                                24856                   0.931289
PCP                                    7492                   0.280705
Specialty                              8405                   0.314913
PlaceSvc                               7632                   0.285951
LengthOfStay                        2597392                  97.317412
DSFS                                  52770                   1.977152
PrimaryConditionGroup                 11410                   0.427503
ProcedureGroup                         3675                   0.137693
```

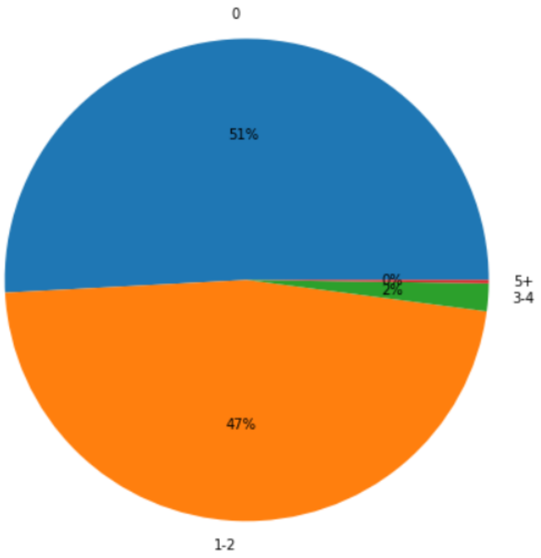
**Transforming PayDelay column**:
The PayDelay column is categorical and has values from 1 to 162+. To convert this column to be used as numeric I have changed rows with values of 162+ to 162 as the values beyond 162 are not present.

**Transforming DSFS column**:
DSFS stands for Days Since First Service(for the year). It contains a time range of months.



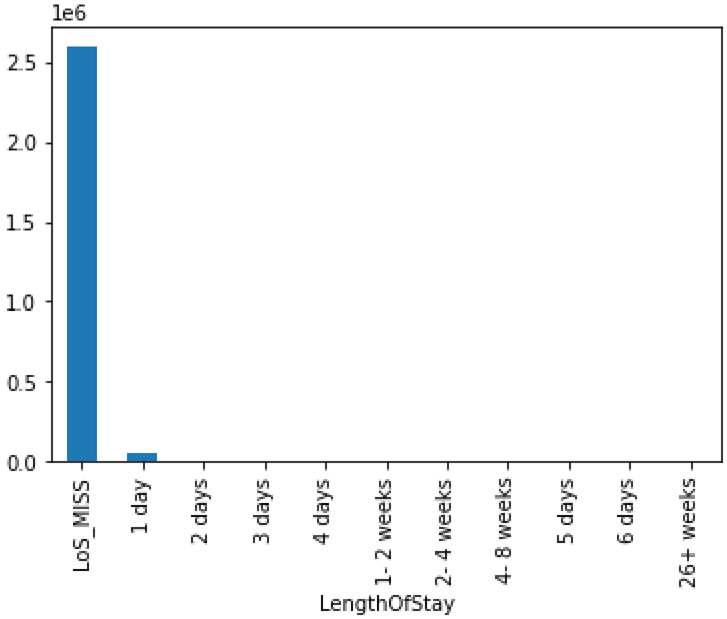
0-1 month is the most common value. To convert it into a numerical value I have considered the upper value of the month. For example, 1-2 months will be stored as 2. As there is no pattern observed I have filled missing values with 0 as it seems that values were left missing for 0 days since service.

**Transforming CharlsonIndex column**:
CharlsonIndex column does not have missing values.
Pie-chart of Charlson index:

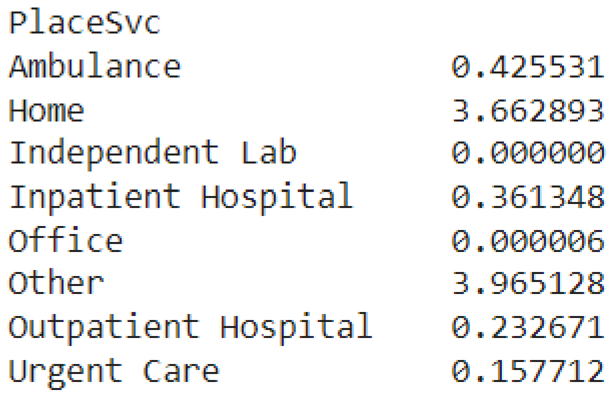


I have changed values from categorical to numeric. For range values, the higher value is considered. Rows having value “5+” are replaced with value 5.

**Transforming LengthOfStay column**:
LengthOfStay column has 97% values missing.



To transform LengthOfStay from categorical to numeric values I have converted values in days. For values mentioned in days, I have just stored numeric values of days. For values mentioned in the range of weeks, I have multiplied the upper limit value by 7 to get the number of days. For example “1-2 weeks” will be stored as 14 (2*7). I have handled missing values by filling 0 as from the data it seems that if less than one day was spent then the value was left missing. By analyzing data it seems that on average only the home and other places of service have high values.

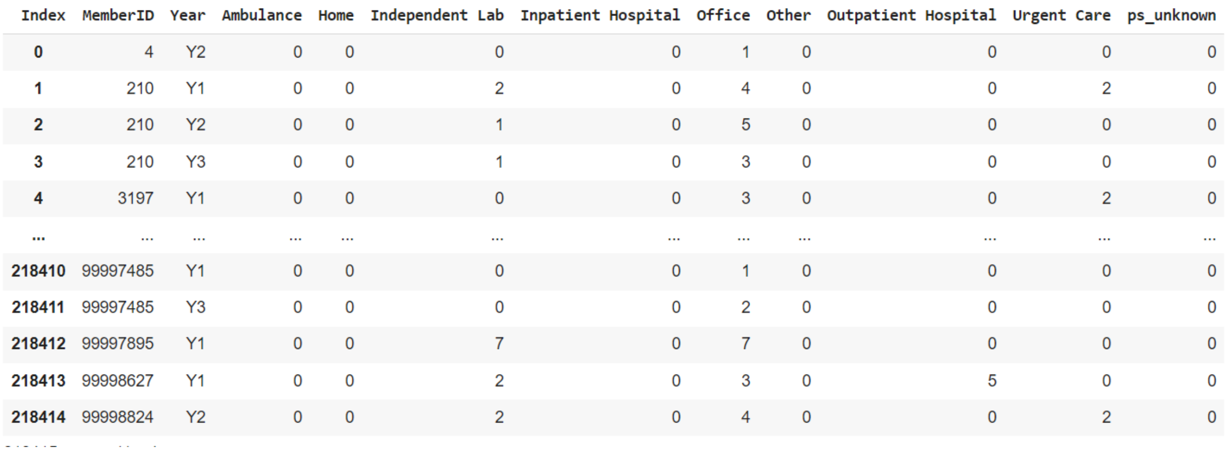


```
PlaceSvc
Ambulance              0.425531
Home                   3.662893
Independent Lab        0.000000
Inpatient Hospital     0.361348
Office                 0.000006
Other                  3.965128
Outpatient Hospital    0.232671
Urgent Care            0.157712
```

### Transforming PrimaryConditionGroup, Specialty, ProcedureGroup, and PlaceSvc columns

All these columns contain categorical data. To transform them to numeric I have created columns for each category and filled them with the sum of instances of that category for each unique MemberID and year pair. I have replaced missing values with “unknown” for each column. So a total of 4 new tables was created, one for each type of variable. These will be merged later on into a single one.

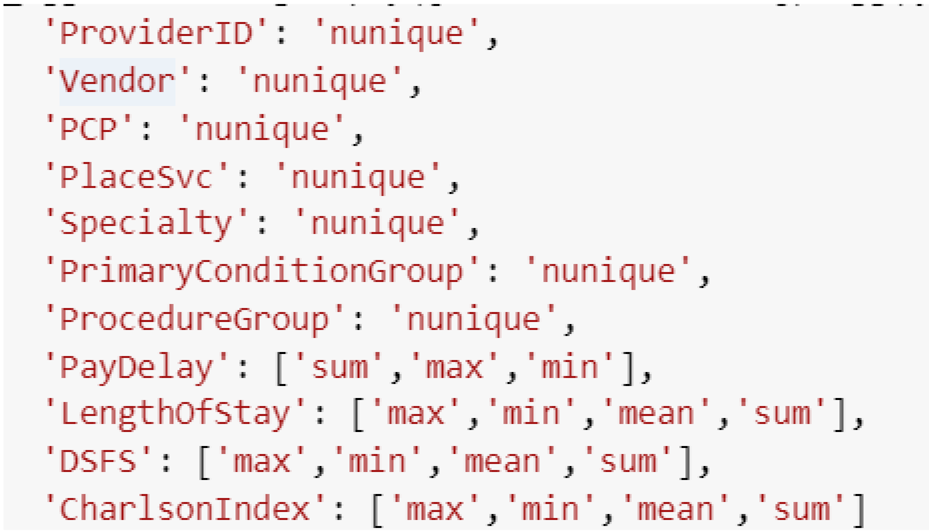
For example, transformed PlaceSvc table:



| Index | MemberID | Year | Ambulance | Home | Independent Lab | Inpatient Hospital | Office | Other | Outpatient Hospital | Urgent Care | ps_unknown |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 4 | Y2 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| 1 | 210 | Y1 | 0 | 0 | 2 | 0 | 4 | 0 | 0 | 2 | 0 |
| 2 | 210 | Y2 | 0 | 0 | 1 | 0 | 5 | 0 | 0 | 0 | 0 |
| 3 | 210 | Y3 | 0 | 0 | 1 | 0 | 3 | 0 | 0 | 0 | 0 |
| 4 | 3197 | Y1 | 0 | 0 | 0 | 0 | 3 | 0 | 0 | 2 | 0 |
| ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| 218410 | 99997485 | Y1 | 0 | 0 | 0 | 0 | 1 | 0 | 0 | 0 | 0 |
| 218411 | 99997485 | Y3 | 0 | 0 | 0 | 0 | 2 | 0 | 0 | 0 | 0 |
| 218412 | 99997895 | Y1 | 0 | 0 | 7 | 0 | 7 | 0 | 0 | 0 | 0 |
| 218413 | 99998627 | Y1 | 0 | 0 | 2 | 0 | 3 | 0 | 5 | 0 | 0 |
| 218414 | 99998824 | Y2 | 0 | 0 | 2 | 0 | 4 | 0 | 0 | 2 | 0 |

### Aggregating values

Finally, I have used aggregated values like sum, unique count, min, max, mean of columns to convert into numeric and be used as predictor variables during regression. The aggregate function done on columns are:



```
  'ProviderID': 'nunique',
  'Vendor': 'nunique',
  'PCP': 'nunique',
  'PlaceSvc': 'nunique',
  'Specialty': 'nunique',
  'PrimaryConditionGroup': 'nunique',
  'ProcedureGroup': 'nunique',
  'PayDelay': ['sum','max','min'],
  'LengthOfStay': ['max','min','mean','sum'],
  'DSFS': ['max','min','mean','sum'],
  'CharlsonIndex': ['max','min','mean','sum']
```

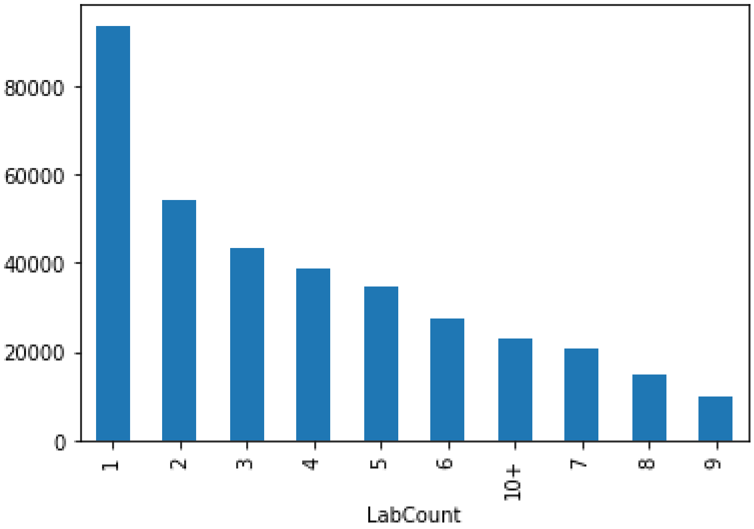
The output is stored in a new table.

**Merging all the subtables created from the Claims table**
Finally, all 5 tables are merged on unique MemberID and Year column pairs. This merged table will be used further. The final claims merged table has 110 columns.

## LabCount table

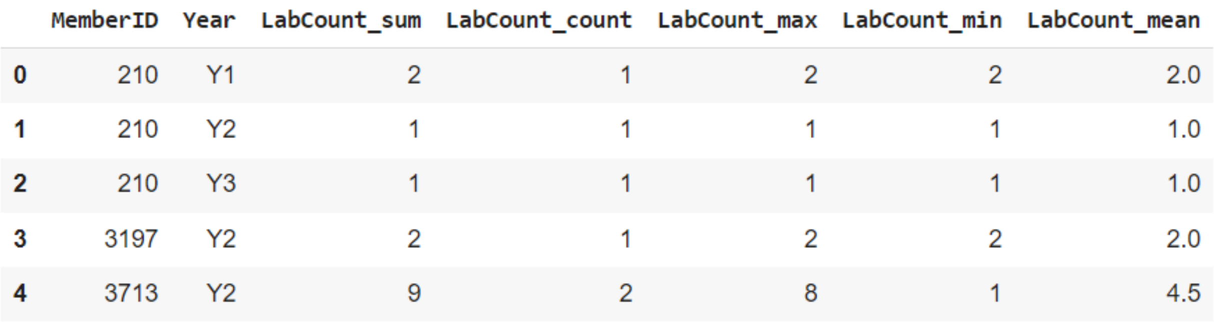
LabCount table has MemberID, Year, DSFS, and LabCount columns. There are no null values in this table. As DSFS is already present in the Claims table there is no need to transform it again as all the individual tables would be transformed in the end.

**Transforming LabCount column**:



I have converted values with “10+” as 10 to transform them into numeric values. I have then calculated aggregated measures like sum, count, max, min, and mean of the LabCount column for unique MemberID and Year column pairs.

Sample of transformed LabCount table:

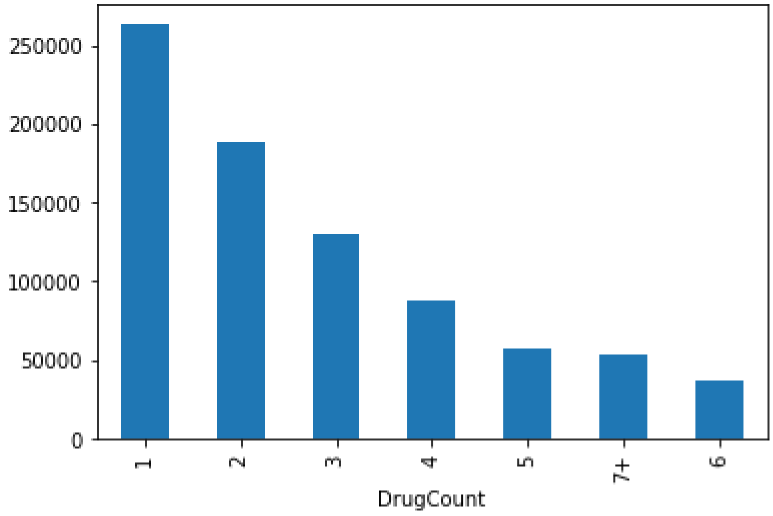


|  | MemberID | Year | LabCount_sum | LabCount_count | LabCount_max | LabCount_min | LabCount_mean |
|---|---|---|---|---|---|---|---|
| **0** | 210 | Y1 | 2 | 1 | 2 | 2 | 2.0 |
| **1** | 210 | Y2 | 1 | 1 | 1 | 1 | 1.0 |
| **2** | 210 | Y3 | 1 | 1 | 1 | 1 | 1.0 |
| **3** | 3197 | Y2 | 2 | 1 | 2 | 2 | 2.0 |
| **4** | 3713 | Y2 | 9 | 2 | 8 | 1 | 4.5 |

## DrugCount table

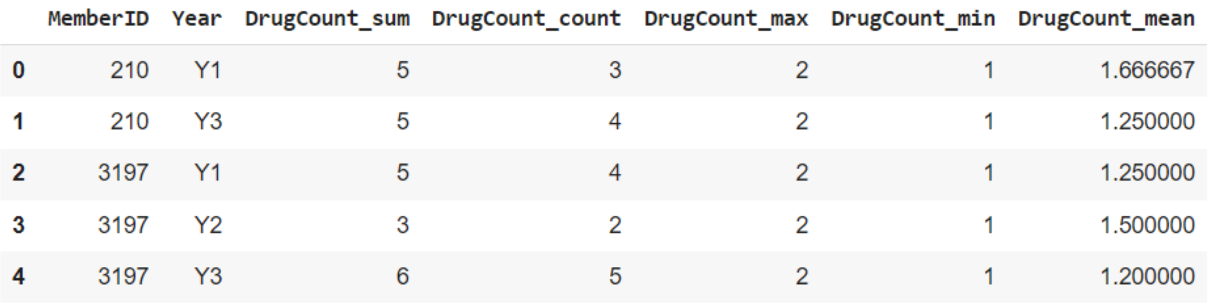
DrugCount table has MemberID, Year, DSFS, and DrugCount columns. There are no null values in this table. As DSFS is already present in the Claims table there is no need to transform it again as all the individual tables would be transformed in the end.

**Transforming DrugCount column**:



I have converted values with “7+” as 7 to transform them into numeric values. I have then calculated aggregated measures like sum, count, max, min, and mean of the DrugCount column for unique MemberID and Year column pairs.

Sample of transformed DrugCount table:



| | MemberID | Year | DrugCount_sum | DrugCount_count | DrugCount_max | DrugCount_min | DrugCount_mean |
|---|---|---|---|---|---|---|---|
| **0** | 210 | Y1 | 5 | 3 | 2 | 1 | 1.666667 |
| **1** | 210 | Y3 | 5 | 4 | 2 | 1 | 1.250000 |
| **2** | 3197 | Y1 | 5 | 4 | 2 | 1 | 1.250000 |
| **3** | 3197 | Y2 | 3 | 2 | 2 | 1 | 1.500000 |
| **4** | 3197 | Y3 | 6 | 5 | 2 | 1 | 1.200000 |

## Final Data Preparation

I have then merged transformed Members, Claims, LabCount, and DrugCount tables. This merged table contains data for all members for all years.
I have then loaded the DaysInHospital_Y2, DaysInHospital_Y3, and Target tables.
I have then filtered data for each year from the merged data. Post this Y1 data is merged with DaysInHospital_Y2. Similarly, Y2 and Y3 data are merged with DaysInHospital_Y3 and Target tables respectively. These 3 tables will be used for regression analysis.

## Regression Analysis

Values of DaysInHospital are to be predicted for year 4. For regression, I have used data of Y1 and Y2. DaysInHospital will be the dependent variable and all the variables except MemberID and Year would be used as predictors. I have used 30% of the data as the test data and the rest as train data. I have created multiple linear regression models and compared them based on Root Mean Squared Logarithmic Error (RMSLE) and R- squared values. The below are definitions:

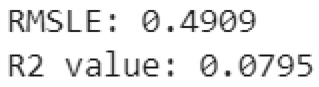
**Root Mean Squared Logarithmic Error(RMSLE):** It is the Root Mean Squared Error of the log-transformed predicted and log-transformed actual values. Lower the RMSLE, better the model.

Formula:

$$RMSLE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(log(\hat{y}_i + 1) - log(y_i + 1))^2}$$

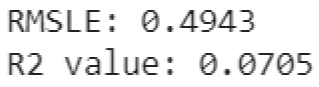
**R- squared:** It is a statistical measure that represents the proportion of the variance for a dependent variable that's explained by an independent variable or variables in a regression model. Higher the R- squared value, the better the model.

**Model 1**:
For this model, I have used all the variables generated to predict DaysInHospital. So there is a total of 128 predictors in this model. After creating the model and evaluating it. Below are the RMSLE and R-squared values:



```
RMSLE: 0.4909
R2 value: 0.0795
```

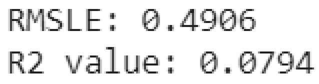
**Model 2:**
To prevent problems like overfitting and multicollinearity I have used Recursive Feature Elimination (RFE) to select the most useful variables for regression. I have first tried RFE to get 64(half) columns. On getting the best 64 columns from RFE I have created a new regression model using these features. On evaluating the model I got the following:



```
RMSLE: 0.4943
R2 value: 0.0705
```

After reducing the variables, RMSLE has increased and R-squared has decreased. Hence Model 1 is better than Model 2.

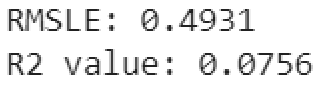
**Model 3:**
In this model, I have used RFE to get the 100 best features. Using these 100 features I have again created a new regression model. On evaluating the model I got the following:



```
RMSLE: 0.4906
R2 value: 0.0794
```

By reducing to 100 features, we are getting the least RMSLE and almost the same R-squared as Model 1. Hence, Model 3 is the best model among the three.

**Model 4:**
Trying to reduce the features to 90. On evaluating the model I got the following:



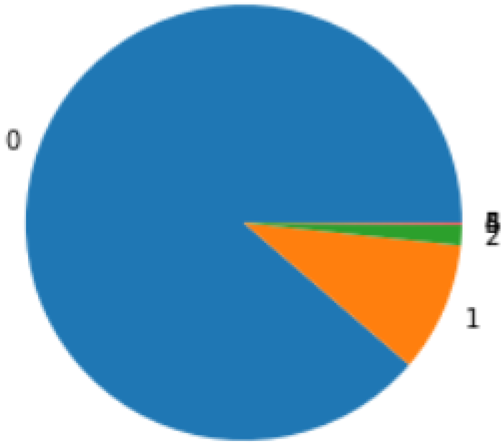
```
RMSLE: 0.4931
R2 value: 0.0756
```

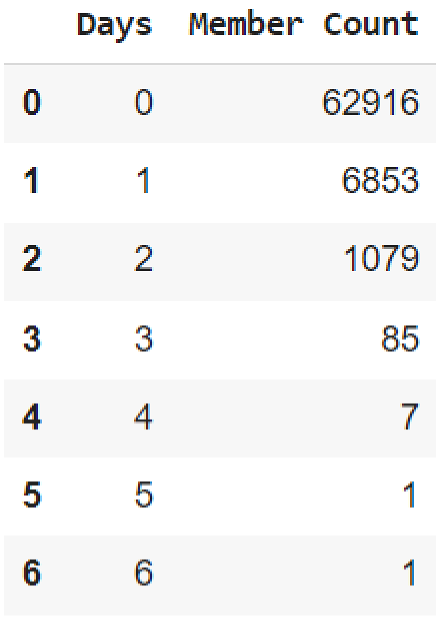
RMSLE has increased compared to Model 3 and R-squared has also decreased.

Among the four models, Model 3 is the best and will be used for predicting DaysInHospital for the Target file.

## Final Prediction:

Using the selected model I have predicted days in the hospital for the members. Below are the results:





|   | Days | Member Count |
|---|---|---|
| **0** | 0 | 62916 |
| **1** | 1 | 6853 |
| **2** | 2 | 1079 |
| **3** | 3 | 85 |
| **4** | 4 | 7 |
| **5** | 5 | 1 |
| **6** | 6 | 1 |

The majority of the members won’t require hospitalization. The maximum number of days of hospitalization required is 6. As days of hospitalization are increasing, member count is decreasing.

## Conclusion:

The purpose of the assignment was feature engineering and data manipulation. The given data was spread across multiple tables and required a lot of transformations. So the data was first transformed into a format usable for prediction using regression. After transforming all the data was merged to be used further for regression. Then regression model was created and using Recursive Feature Elimination best features were selected. Finally, the days in the hospital for year 4 were predicted for all the members. Using the predictions, healthcare professionals can make changes to their system. This prediction model is not perfect and using more sophisticated algorithms and better data transformation the predictions can be made more accurately.