A Mid-Term Report

on

The Healthcare Sector: Evolution from Blackbox Models to SHAP and LIME in Explainable AI

carried out as part of the course CSE CS3270 Submitted by

Ritvik Chawla

209301056

VI CSE-E

in partial fulfilment for the award of the degree

of

BACHELOR OF TECHNOLOGY

In

Computer Science & Engineering

MANIPAL UNIVERSITY
JAIPUR

INSPIRED BY LIFE

Department of Computer Science & Engineering,
School of Computer Science & Engineering,
Manipal University Jaipur,
March 2022

Acknowledgement

This project would not have completed without the help, support, comments, advice, cooperation and coordination of various people. However, it is impossible to thank everyone individually; I am hereby making a humble effort to thank some of them.

I acknowledge and express my deepest sense of gratitude of my internal supervisor **Dr Rishi Gupta** for his/her constant support, guidance, and continuous engagement. I highly appreciate his technical comments, suggestions, and criticism during the progress of this project "**The Healthcare Sector: Evolution from Blackbox Models to SHAP and LIME in Explainable AI**.".

I owe my profound gratitude to **Dr**. **Neha Chaudhary**, Head, Department of CSE, for his valuable guidance and facilitating me during my work. I am also very grateful to all the faculty members and staff for their precious support and cooperation during the development of this project.

Finally, I extend my heartfelt appreciation to my classmates for their help and encouragement.

Registration No.: 209301056
Student Name: Ritvik Chawla

MANIPAL UNIVERSITY
JAIPUR
(University under Section 2(f) of the UGC Act)

INSPIRED BY LIFE

Department of Computer Science and Engineering

School of Computer Science and Engineering

Date: 21.03.2023

CERTIFICATE

This is to certify that the project entitled "_**The Healthcare Sector: Evolution from Blackbox Models to SHAP and LIME in Explainable AI**." is a bonafide work carried out as ***Minor Project Midterm Assessment (Course Code: CS3270)*** in partial fulfillment for the award of the degree of Bachelor of Technology in Computer Science and Engineering, under my guidance by ***Ritvik Chawla*** bearing registration number, **209301056**, during the academic semester *VI of year 2022-23.*

Place: Manipal University Jaipur, Jaipur

Name of the project guide: Dr Rishi Gupta

Signature of the project guide: ______________________

LIST OF TABLES

LIST OF FIGURES

Contents

1. Introduction

Artificial intelligence (AI) models have been used to automate decision-making in a variety of industries, from business to more important ones that significantly influence people's lives, like healthcare. There is a growing movement seeking to construct medically comprehensible AI systems, even though a large percentage of these suggested AI systems are regarded as black box models with no explainability. If people can understand how an AI system arrived at its conclusion, the system is said to be explainable. There is a discussion of several XAI-driven healthcare techniques and how well they performed in the present research. The current paper discusses the development tools utilised in local and global post hoc explainability as well as the many explainability methodologies relevant to logical, statistics, and operational explainability.

The Local Interpretable Model-Agnostic Explanations and Shapley Additive Explanations are used to enforce the explainability of the artificially intelligent system in the medical sector for a stronger comprehension of the internal functionalities of the classic AI models and the similarity among the number of features that impacts the decision of the model.

The state-of-the-art XAI-based methods of today and those of the future are documented on scientific studies in many implementation elements, encompassing research difficulties and model constraints. It is addressed how XAI might be used in the healthcare industry for everything from disease detection to prior illness prediction. Along with several explainability tools, the parameters considered in assessing the model's explainability are offered. For better comprehension, three scenarios regarding using XAI in the healthcare industry are included with their results. The potential of XAI in healthcare will help researchers gain insight into the subject.

1.1 Motivation

I have undertaken this research project as I am interested in Machine Learning, Data Science and Artificial Intelligence. I have been learning and exploring these domains since my fresher year at Manipal University, Jaipur and I feel I can research more in these fields through this opportunity in my 6th-semester minor project. A major industry where explainable AI is being applied today is the healthcare sector, where extensive research is being done for finding and refining techniques with more explainability. I want to research this domain since I have the skillset required to understand the practical functionalities of the explainable AI models and what could lie ahead. This will help me access further opportunities in the domain of explainable AI and will get me to explore the ongoing advancements in the medical sector since in the coming time, the health index will play a major role to check any community or nation’s development. Wherever people are healthy, there will be progress and to meet such standards, technology needs to advance. Therefore, explainable AI and its toolkits will play an enormous role in the rising development of such standards.

2. Literature Review

(i) **Explainable AI: A review of the empirical literature**

The article discusses how AI deep learning models have improved in capability while simultaneously becoming more complicated and difficult to understand. So-called explainable AI (XAI) research has become increasingly popular in response to this trend. This study examines the empirical research on XAI that is based on studies using human subjects. It organises current research according to several technical and experimental parameters. Our findings imply that self-reported comprehension and trust in AI are enhanced by explainable AI. Yet, this seldom ever results in better human performance on activities with incentives when supported by AI. Regarding the application of explainable AI in human-computer interaction, we offer several consequences of these findings.

(ii) **The Third Wave of AI: Fraud Detection via Knowledge Graphs within Explainable AI**

They narrow our attention to the subject of finance in this essay. As most operations in the financial sector now take place online, more innovative financial technology development is required to offer financial products and services that utilise ICT (ICT). Hence, any output from AI that is introduced into such a mission-critical industry surely needs a very strong justification. The adoption of AI in the financial sector as well as adjacent mission-critical industries including aerospace, security, defence, and ground transportation has been hampered by the absence of a very persuasive justification. Hence, explainable AI must be used to ensure equity, responsibility, dependability, openness, and privacy/security.

(iii) **From Blackbox to Explainable AI in Healthcare: Existing Tools and Case Studies**

Artificial intelligence (AI) models have been used to automate decision-making in a variety of industries, from commerce to more important ones that directly impact people's lives, including healthcare. There is a growing trend of trying to develop medically explicable Artificial Intelligence (XAI) systems employing strategies like attention mechanisms and surrogate models, even though the great majority of these suggested AI systems are regarded as black box models with no explainability. If people can understand how an AI system arrived at its choice, the system is considered to be explainable. There is the discussion of several XAI-driven healthcare techniques and how well they performed in the present research. The current paper discusses the toolkits utilised in local and global post hoc explainability as well as the many explainability methodologies relevant to the rational, data, and performance explainability. The Local Interpretable Model-Agnostic Explanations and Shapley Additive Explanations are used to enforce the explainability of the a.i. model in the health sector for greater understanding of the internal functional blocks of the original AI models and the correlation among the extracted features that impacts decision-making of the model. The state-of-the-art XAI-based technologies of today and those of the future are presented on study results in many implementation elements, including research difficulties and model constraints. It is addressed how XAI might be used in the healthcare industry for everything from disease detection to earlier sickness prediction. Together with several explainability tools, the metrics taken into account in assessing the model's explainability are offered.

(iv) <u>Explainable AI – Requirements, Use Cases and Solutions</u>

In 2025, it is anticipated that Germany alone would see earnings from services and goods based on the usage of artificial intelligence (AI) total 488 billion euros, or 13% of Germany's GDP. In significant application areas, the users' acceptance, approval and certification processes, or compliance with the GDPR's transparency requirements all depend on the judgements made by AI. So, at least in the European environment, one of the most crucial commercial success elements for AI products is their explainability. The underlying AI models are always at the heart of AI-based applications, by which we effectively mean machine learning applications in this context. The two classifications of these are white-box models and black-box models. White-box models that use understandable input variables, such decision trees, enable a rudimentary understanding of underlying algorithmic linkages. As a result, both their decision-making and processes of action are self-explanatory. Due to the interconnection and multi-layered structure of black-box models like neural networks, it is typically no longer feasible to comprehend how the model functions. Nevertheless, additional explanatory tools can be utilised in order to later improve comprehensibility, at least for the explanation of individual decisions (local explainability). AI developers may rely on well-known explanation tools, such as LIME, SHAP, Integrated Gradients, LRP, DeepLift, or GradCAM, depending on the particular requirements, but these tools demand specialist expertise. There are now just a handful decent tools available for casual AI users that give simple to comprehend decision justifications (saliency maps, counterfactual explanations, prototypes or surrogate models). About equal amounts of popular white-box models (statistical/probabilistic models, decision trees) and black-box models (neural networks) are used today by participants in the survey that was done as part of this study. Nonetheless, the poll indicates that a growing usage of black-box models, particularly neural networks, is anticipated in the future. This means that while explanatory techniques are already a crucial part of many AI applications today, their significance will only grow in the future.

2.3 Outcome of Literature Review

The literature review gives a brief understanding of the models we are trying to undertake and what all pros are there as well as what all cons need to be addressed in the research.

2.4 Problem Statement

The problem we are trying to address through this minor project is so understand the reason why models like SHAP and LIME which are the focal points of explainable AI give greater efficiency in predicting outcomes in the health sector for several applications compared to blackbox models of before.

2.5 Research Objectives.

The current study will examine various explainable perspectives, tools, and research reports that would help people in explaining the models employed in the health sector. When it comes to medical implementations, there is a meaningful relationship between the efficiency of treatment or therapy and the acknowledgement of the approach used. The following is an overview of the studies' total contributions:

- Explaining the several domains in which XAI models and systems can be included making the models noticeable.
- Discuss the different systems, such as explainable models based on LIME and SHAP.
- Talk about the many toolkits that are offered to make the model explainable under different explainable variables.
- Present the numerous explainability categories connected to the decision models in intricate detail.
- The study would be easier to understand as healthcare-related case studies and statistical analyses are presented.

PROS and CONS of black box models, LIME and SHAP

PROS	CONS
• the LIME’s basic operating premise is to evaluate the model for local transparency and understandability. To assess the local transparency of the model, the characteristics used throughout the prediction phase are crucial. The local explainability does, however, increase the prediction process' vigilance.	• While the process’s vigilance does increase in a LIME model’s basic operation premise, it may or may not fit the model globally.
• Utilizing XAI technology, gene expression variations are being studied. Since rule-based techniques are particularly well adapted for empirical	• Numerous reliable methods have been utilised as "black boxes," providing no information on the usage of certain evaluation, classification, and prediction

confirmation of the predictions generated from gene analysis, XAI was developed as a solution to this problem.	techniques. Although consumers' potential to use the tools or apps that these models equip may not be immediately impacted by this lack of transparency, professionals may nevertheless be able to understand their structure.
• To approximate the coherence and accuracy of the local model, the Shapy values in the SHAP model provide a distinct additive feature set. Both model-specific and model-independent justifications may be used with SHAP successfully.	• For a comprehensive understanding of diverse challenges, collecting and examining useful but outdated functionalities may be necessary. In several disciplines of healthcare informatics, determining the kind of illness and future model risk analysis requires the application of data collecting, preprocessing, preparation, modelling, and visualisation.
• In order to replace the originally utilised linear models with more complicated and conceivably more reliable models, interpretable artificial intelligent models and methodologies for error handling, description, and fairness may aid in the spread of and trust in newer or more stringent machine learning techniques.	

3. Methodology and Framework

3.1 System Architecture

Google Collaboratory/ Kaggle/ Jupyter Notebook: Used for a minor application project that will help explain the XAI concepts and toolkits being researched and discussed in this paper. Machine learning and artificial intelligence algorithms can be implemented on such platforms to get certain statistical conclusions and parameters, which will help consolidate the research.

3.2 Algorithms, Techniques etc.

The algorithms and techniques used for the minor project are a variety of machine learning libraries which have been implemented on a Kaggle notebook where the data has been imported. Libraries such as Numpy, Pandas, Seaborn, Matplotlib, sklearn, catboost, itertools, shap, etc have also been imported to give the project implementation more substantial outcomes and better predictions.

- **Numpy**: NumPy can be used **to perform a wide variety of mathematical operations on arrays**. It adds powerful data structures to Python that guarantee efficient calculations with arrays and matrices, and it supplies an enormous library of high-level mathematical functions that operate on these arrays and matrices

- **Pandas**: Pandas is a Python library used for working with data sets. It has functions for analyzing, cleaning, exploring, and manipulating data.

- **Seaborn**: Seaborn is a library for **making statistical graphics in Python**. It builds on top of matplotlib and integrates closely with pandas data structures. Seaborn helps you explore and understand your data.

- **Matplotlib**: Matplotlib is a comprehensive library for **creating static, animated, and interactive visualizations in Python**. Matplotlib makes easy things easy and hard things possible. Create publication-quality plots. Make interactive figures that can zoom, pan, update, etc.

- **Sklearn**: Scikit-learn (Sklearn) is the most useful and robust library for machine learning in Python. It provides a selection of efficient tools for machine learning and statistical modelling including classification, regression, clustering and dimensionality reduction via a consistence interface in Python.

- **Catboost**: CatBoost is an open-source software library developed by Yandex. It provides a gradient boosting framework which among other features attempts to solve for Categorical features using a permutation-driven alternative compared to the classical algorithm.

- **Itertools**: Python's Itertool is a module that provides various functions that work on iterators to produce complex iterators. This module works as a fast, memory-efficient tool that is used either by themselves or in combination to form **iterator algebra**.

- **Shap**: SHAP is **the most powerful Python package for understanding and debugging your models**. It can tell us how each model feature has contributed to an individual prediction. By aggregating SHAP values, we can also understand trends across multiple predictions.

3.3 Detailed Design Methodologies (as applicable)

1. The paper will start with first defining the problem statement related to the topic chosen
2. Next, we need to explain the current and prerequisite knowledge about the problem statement we are trying to solve. This includes the theoretical knowledge about the previous black box models, the innovations that took place with them, and all problem statements they helped tackle.
3. We shall also include flow charts and diagrams which help the reader understand the concepts we are trying to explain as well as the features of the problem statement through such visuals.

4. A few examples are:

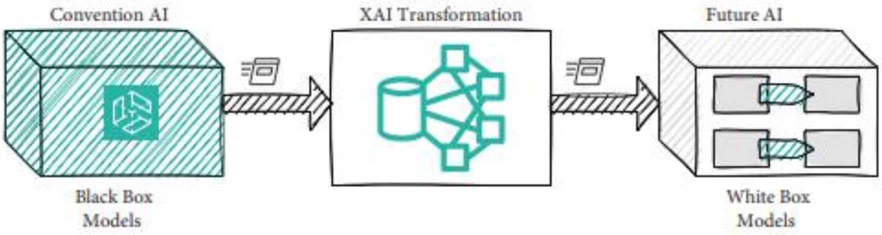
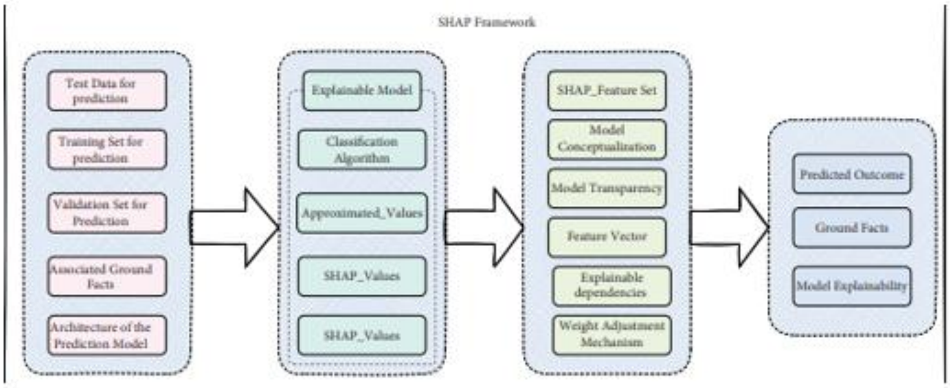


Image representing the transformation of XAI



SHAP framework for the explainable model.

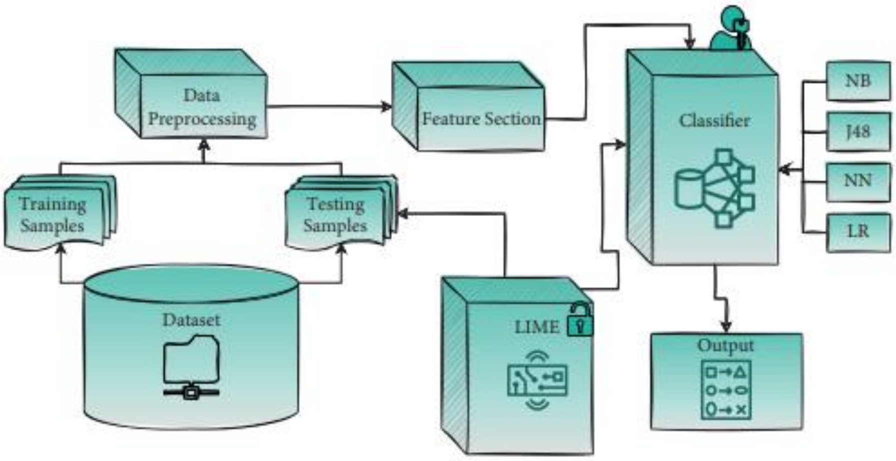


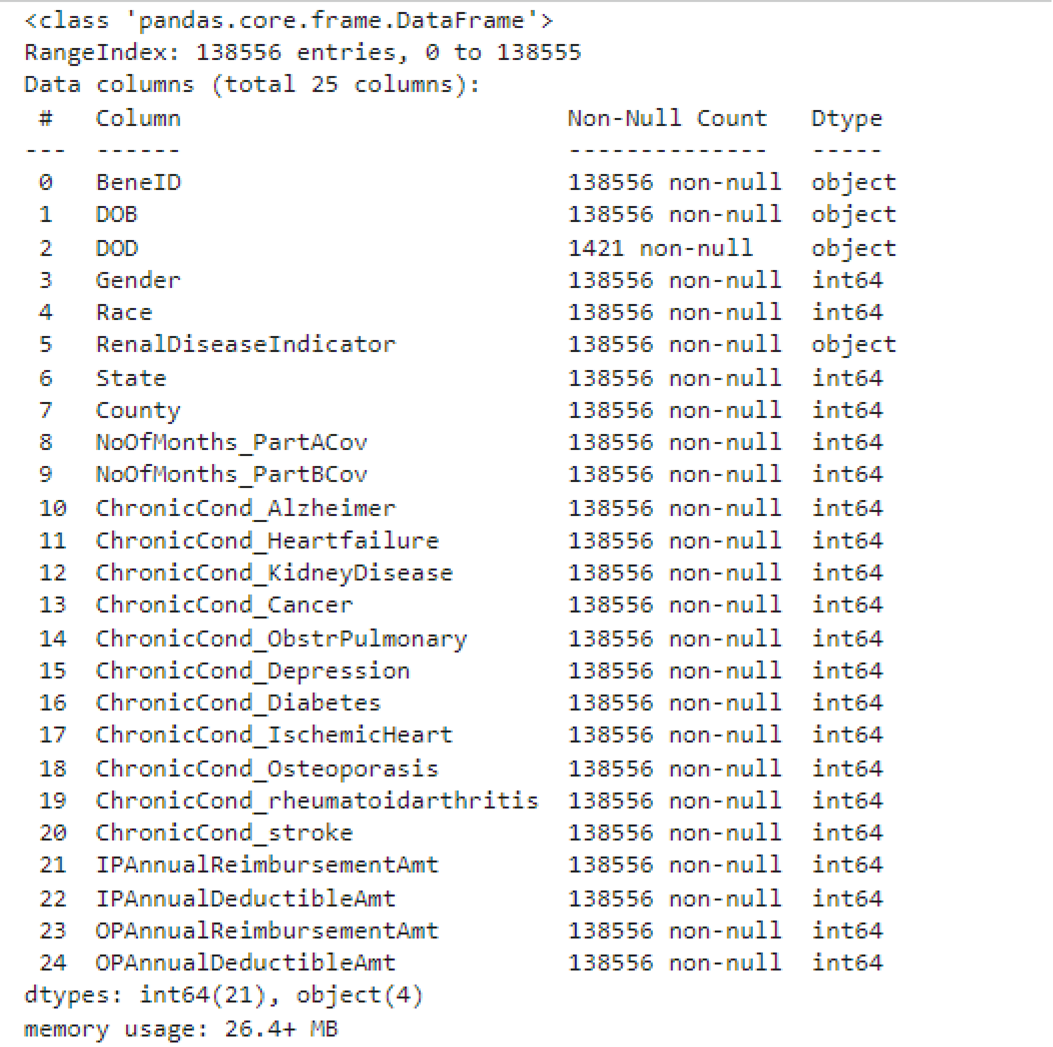
Image representing the block diagram of the LIME-based prediction model.

5. Try out a minor project on specific platforms like Google Collaboratory, Kaggle, Jupyter Notebook, etc where we can get results and outputs corresponding to the problem statement.
6. Credit the sites using the bibliography through which the content for your research came.
7. Upload the project to GitHub and get the research paper published.

4. Work Done

4.1 Details as required.

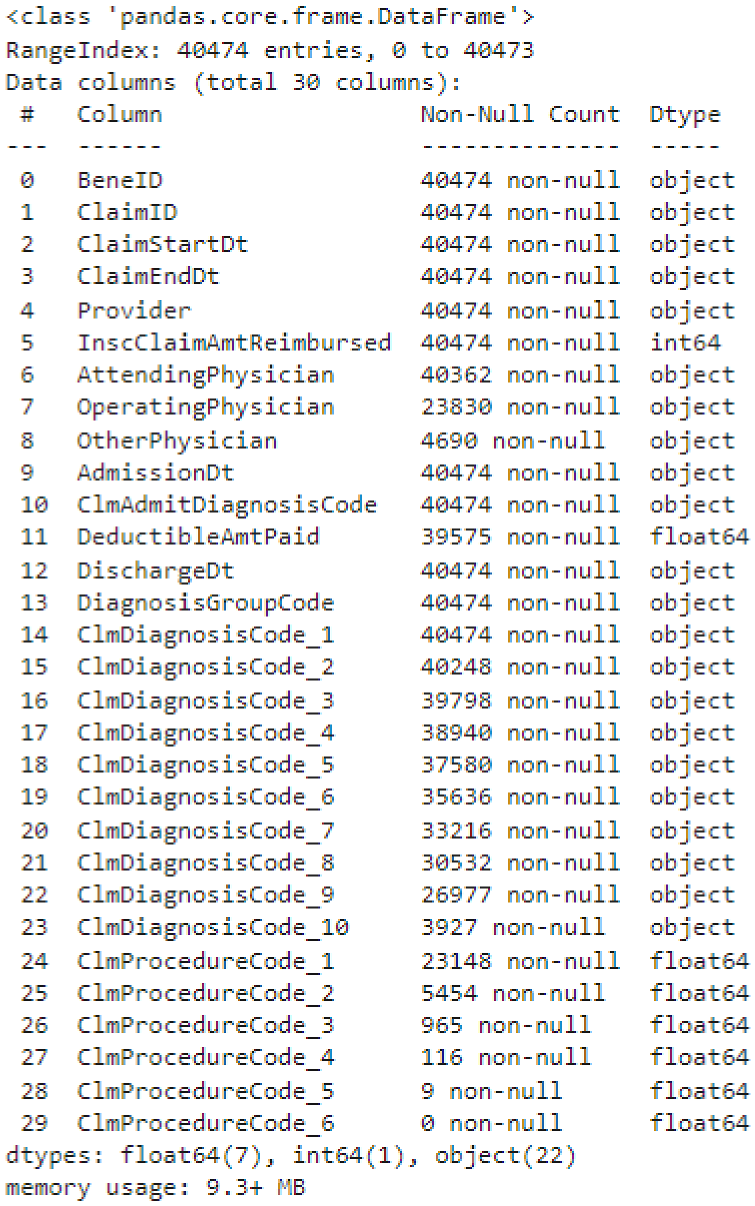
- Relevant articles were searched on various sources on the internet which will be able to give the appropriate information needed to build an understanding of the concept of black box models and explainable AI models like LIME and SHAP along with their effect on the healthcare sector.
- Took out appropriate and relevant content which can help explain the problem statement and give the reader a detailed idea of the background of the topic. Using the case studies which I saw as references, I was able to explain the requirement for systems with explicit AI, Explainability on a local vs. global scale, agnostic and model-specific models, XAI model characteristics, locally interpretable and model-independent justifications, Additive explanations by Shapley, frameworks, and toolkits, etc.
- Using GitHub, found medical data on medical fraud detection including beneficiary data which had features like:



```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 138556 entries, 0 to 138555
Data columns (total 25 columns):
 #   Column                           Non-Null Count   Dtype
---  ------                           --------------   -----
 0   BeneID                           138556 non-null  object
 1   DOB                              138556 non-null  object
 2   DOD                              1421 non-null    object
 3   Gender                           138556 non-null  int64
 4   Race                             138556 non-null  int64
 5   RenalDiseaseIndicator            138556 non-null  object
 6   State                            138556 non-null  int64
 7   County                           138556 non-null  int64
 8   NoOfMonths_PartACov              138556 non-null  int64
 9   NoOfMonths_PartBCov              138556 non-null  int64
 10  ChronicCond_Alzheimer            138556 non-null  int64
 11  ChronicCond_Heartfailure         138556 non-null  int64
 12  ChronicCond_KidneyDisease        138556 non-null  int64
 13  ChronicCond_Cancer               138556 non-null  int64
 14  ChronicCond_ObstrPulmonary       138556 non-null  int64
 15  ChronicCond_Depression           138556 non-null  int64
 16  ChronicCond_Diabetes             138556 non-null  int64
 17  ChronicCond_IschemicHeart        138556 non-null  int64
 18  ChronicCond_Osteoporasis         138556 non-null  int64
 19  ChronicCond_rheumatoidarthritis  138556 non-null  int64
 20  ChronicCond_stroke               138556 non-null  int64
 21  IPAnnualReimbursementAmt         138556 non-null  int64
 22  IPAnnualDeductibleAmt            138556 non-null  int64
 23  OPAnnualReimbursementAmt         138556 non-null  int64
 24  OPAnnualDeductibleAmt            138556 non-null  int64
dtypes: int64(21), object(4)
memory usage: 26.4+ MB
```

Features of beneficiary dataset using the .info() function

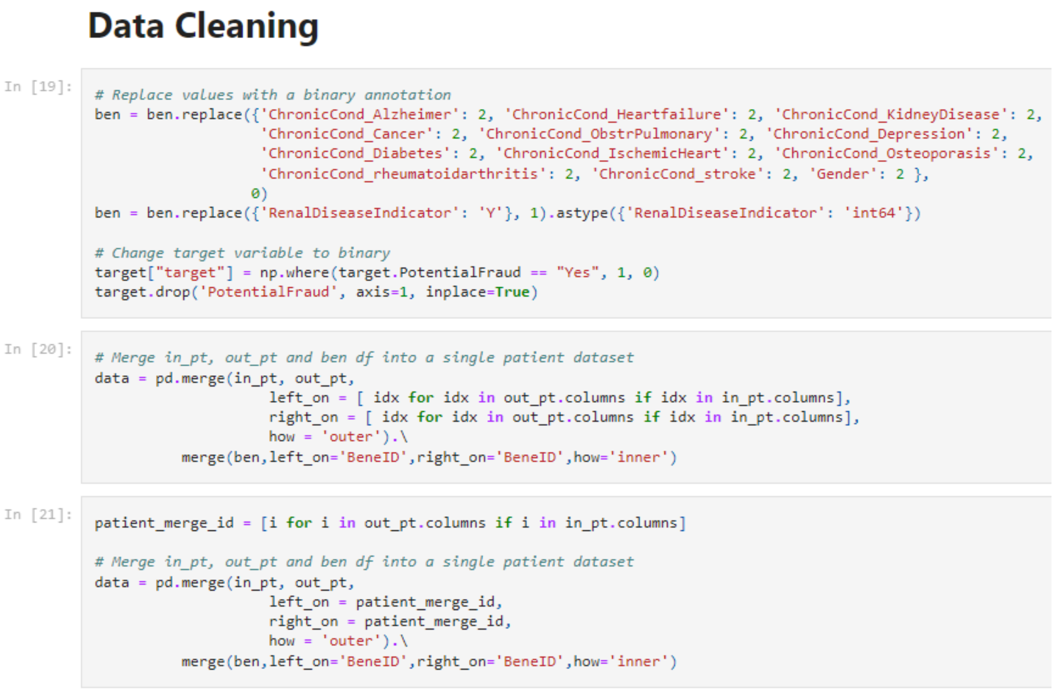
It had inpatient and outpatient data also which roughly had similar features. It looked something like this:



```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 40474 entries, 0 to 40473
Data columns (total 30 columns):
 #   Column                  Non-Null Count  Dtype  
---  ------                  --------------  -----  
 0   BeneID                  40474 non-null  object 
 1   ClaimID                 40474 non-null  object 
 2   ClaimStartDt            40474 non-null  object 
 3   ClaimEndDt              40474 non-null  object 
 4   Provider                40474 non-null  object 
 5   InscClaimAmtReimbursed  40474 non-null  int64  
 6   AttendingPhysician      40362 non-null  object 
 7   OperatingPhysician      23830 non-null  object 
 8   OtherPhysician          4690 non-null   object 
 9   AdmissionDt             40474 non-null  object 
 10  ClmAdmitDiagnosisCode   40474 non-null  object 
 11  DeductibleAmtPaid       39575 non-null  float64
 12  DischargeDt             40474 non-null  object 
 13  DiagnosisGroupCode      40474 non-null  object 
 14  ClmDiagnosisCode_1      40474 non-null  object 
 15  ClmDiagnosisCode_2      40248 non-null  object 
 16  ClmDiagnosisCode_3      39798 non-null  object 
 17  ClmDiagnosisCode_4      38940 non-null  object 
 18  ClmDiagnosisCode_5      37580 non-null  object 
 19  ClmDiagnosisCode_6      35636 non-null  object 
 20  ClmDiagnosisCode_7      33216 non-null  object 
 21  ClmDiagnosisCode_8      30532 non-null  object 
 22  ClmDiagnosisCode_9      26977 non-null  object 
 23  ClmDiagnosisCode_10     3927 non-null   object 
 24  ClmProcedureCode_1      23148 non-null  float64
 25  ClmProcedureCode_2      5454 non-null   float64
 26  ClmProcedureCode_3      965 non-null    float64
 27  ClmProcedureCode_4      116 non-null    float64
 28  ClmProcedureCode_5      9 non-null      float64
 29  ClmProcedureCode_6      0 non-null      float64
dtypes: float64(7), int64(1), object(22)
memory usage: 9.3+ MB
```

Features of inpatient and outpatient datasets using the .info() function

- This data will help perform data cleaning, data mining, exploratory data analysis (EDA), pre-processing, feature engineering, balancing the data, and model building using the SHAP library in the catboost module to show the enhanced prediction percentage result of SHAP compared to black box models.



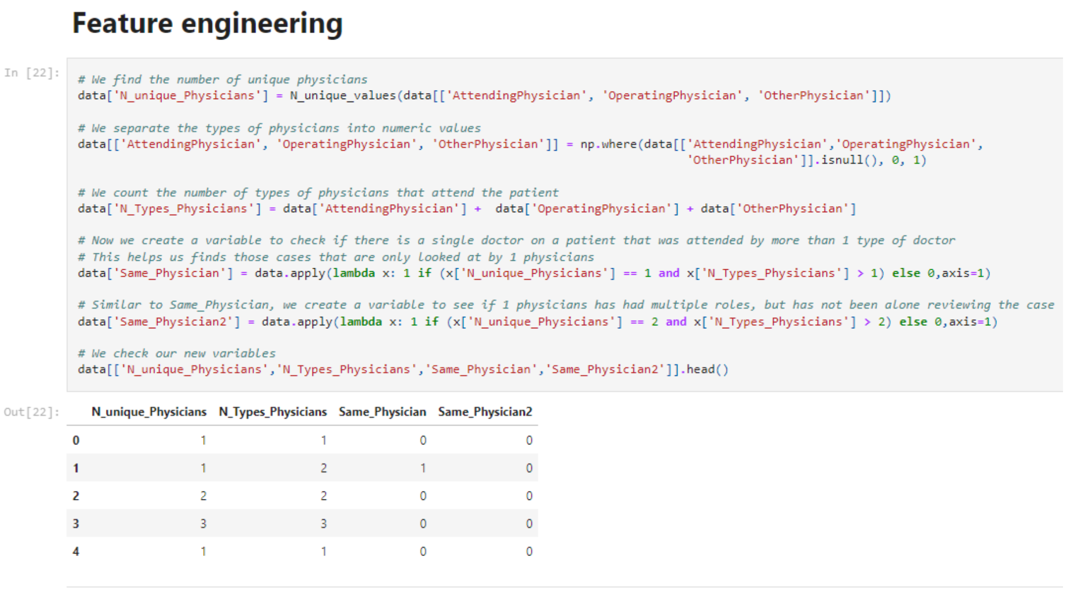
Data Cleaning

```
In [19]:
# Replace values with a binary annotation
ben = ben.replace({'ChronicCond_Alzheimer': 2, 'ChronicCond_Heartfailure': 2, 'ChronicCond_KidneyDisease': 2,
                   'ChronicCond_Cancer': 2, 'ChronicCond_ObstrPulmonary': 2, 'ChronicCond_Depression': 2,
                   'ChronicCond_Diabetes': 2, 'ChronicCond_IschemicHeart': 2, 'ChronicCond_Osteoporasis': 2,
                   'ChronicCond_rheumatoidarthritis': 2, 'ChronicCond_stroke': 2, 'Gender': 2 },
                  0)
ben = ben.replace({'RenalDiseaseIndicator': 'Y'}, 1).astype({'RenalDiseaseIndicator': 'int64'})

# Change target variable to binary
target["target"] = np.where(target.PotentialFraud == "Yes", 1, 0)
target.drop('PotentialFraud', axis=1, inplace=True)
```

```
In [20]:
# Merge in_pt, out_pt and ben df into a single patient dataset
data = pd.merge(in_pt, out_pt,
                    left_on = [ idx for idx in out_pt.columns if idx in in_pt.columns],
                    right_on = [ idx for idx in out_pt.columns if idx in in_pt.columns],
                    how = 'outer').\
          merge(ben,left_on='BeneID',right_on='BeneID',how='inner')
```

```
In [21]:
patient_merge_id = [i for i in out_pt.columns if i in in_pt.columns]

# Merge in_pt, out_pt and ben df into a single patient dataset
data = pd.merge(in_pt, out_pt,
                    left_on = patient_merge_id,
                    right_on = patient_merge_id,
                    how = 'outer').\
          merge(ben,left_on='BeneID',right_on='BeneID',how='inner')
```

Data Cleaning being performed on the data sets

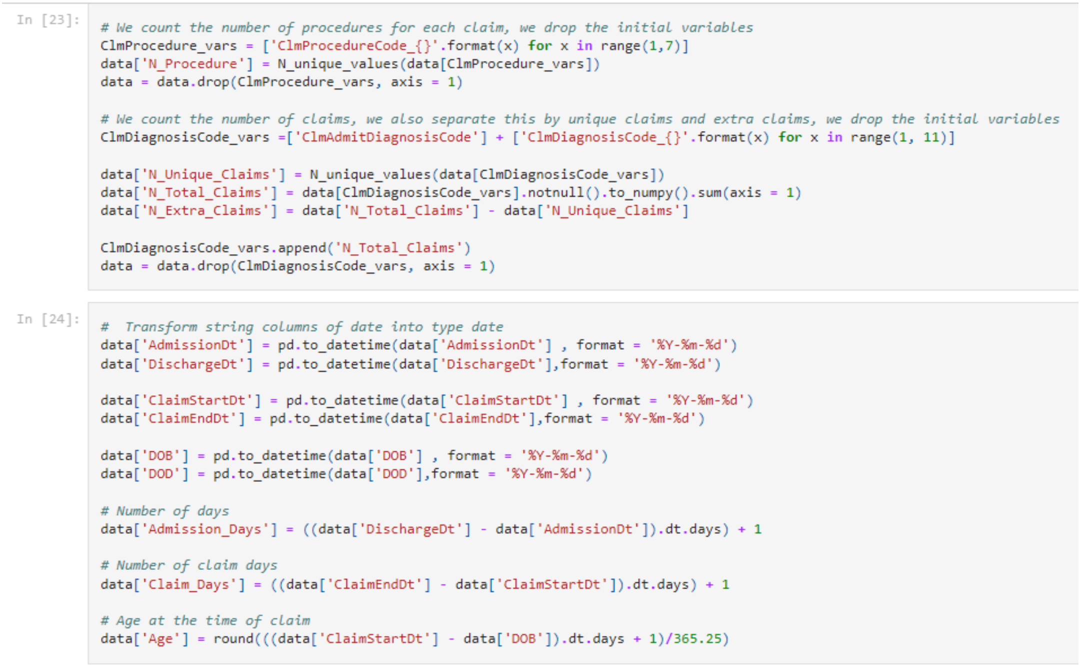


Feature engineering

```
In [22]:
# We find the number of unique physicians
data['N_unique_Physicians'] = N_unique_values(data[['AttendingPhysician', 'OperatingPhysician', 'OtherPhysician']])

# We separate the types of physicians into numeric values
data[['AttendingPhysician', 'OperatingPhysician', 'OtherPhysician']] = np.where(data[['AttendingPhysician','OperatingPhysician',
                                                                                  'OtherPhysician']].isnull(), 0, 1)

# We count the number of types of physicians that attend the patient
data['N_Types_Physicians'] = data['AttendingPhysician'] +  data['OperatingPhysician'] + data['OtherPhysician']

# Now we create a variable to check if there is a single doctor on a patient that was attended by more than 1 type of doctor
# This helps us finds those cases that are only looked at by 1 physicians
data['Same_Physician'] = data.apply(lambda x: 1 if (x['N_unique_Physicians'] == 1 and x['N_Types_Physicians'] > 1) else 0,axis=1)

# Similar to Same_Physician, we create a variable to see if 1 physicians has had multiple roles, but has not been alone reviewing the case
data['Same_Physician2'] = data.apply(lambda x: 1 if (x['N_unique_Physicians'] == 2 and x['N_Types_Physicians'] > 2) else 0,axis=1)

# We check our new variables
data[['N_unique_Physicians','N_Types_Physicians','Same_Physician','Same_Physician2']].head()
```

```
Out[22]:
   N_unique_Physicians  N_Types_Physicians  Same_Physician  Same_Physician2
0                    1                   1               0                0
1                    1                   2               1                0
2                    2                   2               0                0
3                    3                   3               0                0
4                    1                   1               0                0
```

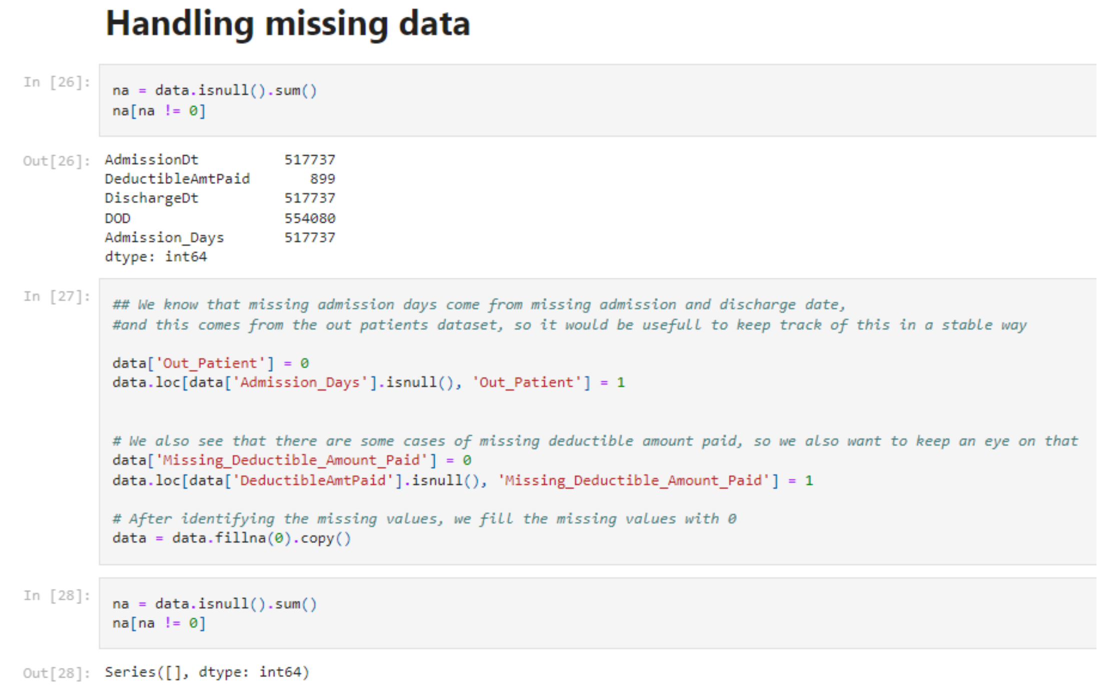
Feature Engineering pt. 1



```
In [23]:
# We count the number of procedures for each claim, we drop the initial variables
ClmProcedure_vars = ['ClmProcedureCode_{}'.format(x) for x in range(1,7)]
data['N_Procedure'] = N_unique_values(data[ClmProcedure_vars])
data = data.drop(ClmProcedure_vars, axis = 1)

# We count the number of claims, we also separate this by unique claims and extra claims, we drop the initial variables
ClmDiagnosisCode_vars =['ClmAdmitDiagnosisCode'] + ['ClmDiagnosisCode_{}'.format(x) for x in range(1, 11)]

data['N_Unique_Claims'] = N_unique_values(data[ClmDiagnosisCode_vars])
data['N_Total_Claims'] = data[ClmDiagnosisCode_vars].notnull().to_numpy().sum(axis = 1)
data['N_Extra_Claims'] = data['N_Total_Claims'] - data['N_Unique_Claims']

ClmDiagnosisCode_vars.append('N_Total_Claims')
data = data.drop(ClmDiagnosisCode_vars, axis = 1)
```

```
In [24]:
#  Transform string columns of date into type date
data['AdmissionDt'] = pd.to_datetime(data['AdmissionDt'] , format = '%Y-%m-%d')
data['DischargeDt'] = pd.to_datetime(data['DischargeDt'],format = '%Y-%m-%d')

data['ClaimStartDt'] = pd.to_datetime(data['ClaimStartDt'] , format = '%Y-%m-%d')
data['ClaimEndDt'] = pd.to_datetime(data['ClaimEndDt'],format = '%Y-%m-%d')

data['DOB'] = pd.to_datetime(data['DOB'] , format = '%Y-%m-%d')
data['DOD'] = pd.to_datetime(data['DOD'],format = '%Y-%m-%d')

# Number of days
data['Admission_Days'] = ((data['DischargeDt'] - data['AdmissionDt']).dt.days) + 1

# Number of claim days
data['Claim_Days'] = ((data['ClaimEndDt'] - data['ClaimStartDt']).dt.days) + 1

# Age at the time of claim
data['Age'] = round(((data['ClaimStartDt'] - data['DOB']).dt.days + 1)/365.25)
```

Feature Engineering pt. 2



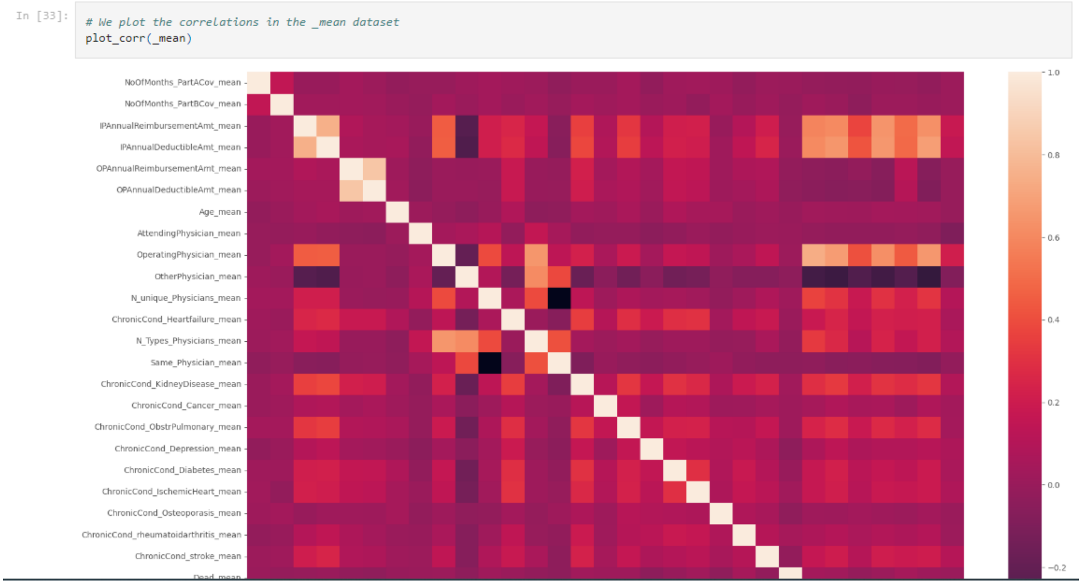
Handling missing data

```
In [26]:
na = data.isnull().sum()
na[na != 0]

Out[26]:
AdmissionDt          517737
DeductibleAmtPaid       899
DischargeDt          517737
DOD                  554080
Admission_Days       517737
dtype: int64
```

```
In [27]:
## We know that missing admission days come from missing admission and discharge date,
#and this comes from the out patients dataset, so it would be usefull to keep track of this in a stable way

data['Out_Patient'] = 0
data.loc[data['Admission_Days'].isnull(), 'Out_Patient'] = 1


# We also see that there are some cases of missing deductible amount paid, so we also want to keep an eye on that
data['Missing_Deductible_Amount_Paid'] = 0
data.loc[data['DeductibleAmtPaid'].isnull(), 'Missing_Deductible_Amount_Paid'] = 1

# After identifying the missing values, we fill the missing values with 0
data = data.fillna(0).copy()
```

```
In [28]:
na = data.isnull().sum()
na[na != 0]

Out[28]: Series([], dtype: int64)
```

Missing values either replaced or removed.

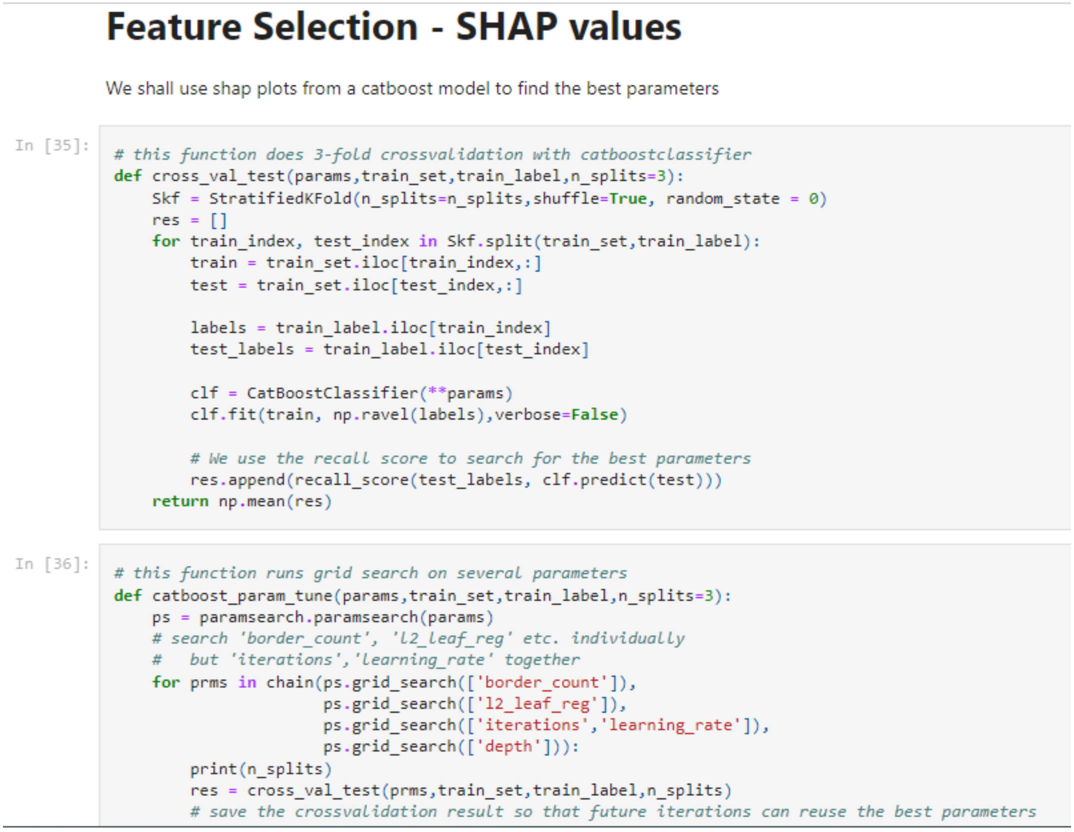


In [33]:

```python
# We plot the correlations in the _mean dataset
plot_corr(_mean)
```

NoOfMonths_PartACov_mean
NoOfMonths_PartBCov_mean
IPAnnualReimbursementAmt_mean
IPAnnualDeductibleAmt_mean
OPAnnualReimbursementAmt_mean
OPAnnualDeductibleAmt_mean
Age_mean
AttendingPhysician_mean
OperatingPhysician_mean
OtherPhysician_mean
N_unique_Physicians_mean
ChronicCond_Heartfailure_mean
N_Types_Physicians_mean
Same_Physician_mean
ChronicCond_KidneyDisease_mean
ChronicCond_Cancer_mean
ChronicCond_ObstrPulmonary_mean
ChronicCond_Depression_mean
ChronicCond_Diabetes_mean
ChronicCond_IschemicHeart_mean
ChronicCond_Osteoporasis_mean
ChronicCond_rheumatoidarthritis_mean
ChronicCond_stroke_mean
Dead_mean
1.0
0.8
0.6
0.4
0.2
0.0
−0.2

One of the few correlations charts



Feature Selection - SHAP values

We shall use shap plots from a catboost model to find the best parameters

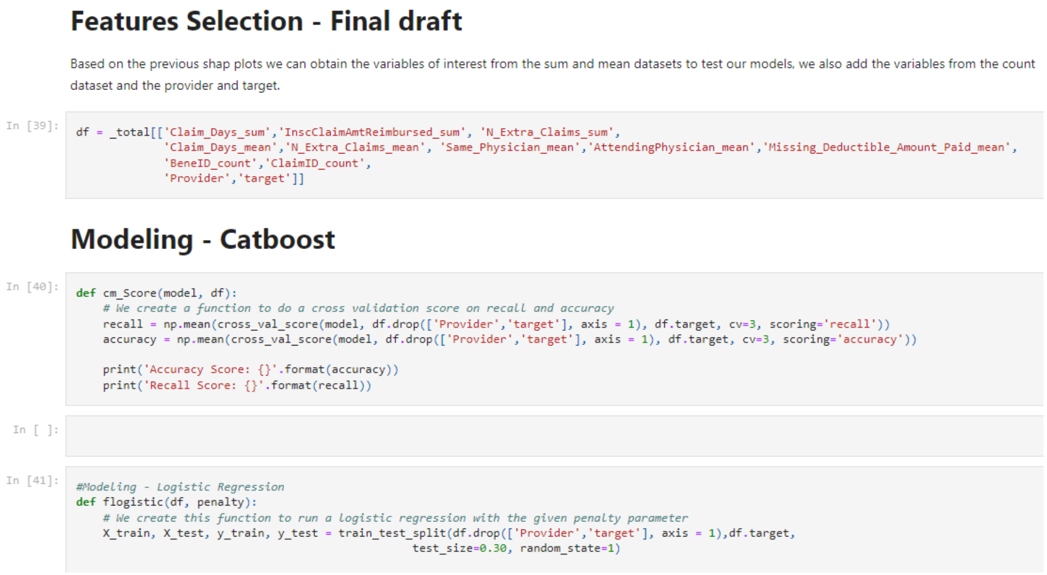
In [35]:

```python
# this function does 3-fold crossvalidation with catboostclassifier
def cross_val_test(params,train_set,train_label,n_splits=3):
    Skf = StratifiedKFold(n_splits=n_splits,shuffle=True, random_state = 0)
    res = []
    for train_index, test_index in Skf.split(train_set,train_label):
        train = train_set.iloc[train_index,:]
        test = train_set.iloc[test_index,:]

        labels = train_label.iloc[train_index]
        test_labels = train_label.iloc[test_index]

        clf = CatBoostClassifier(**params)
        clf.fit(train, np.ravel(labels),verbose=False)

        # We use the recall score to search for the best parameters
        res.append(recall_score(test_labels, clf.predict(test)))
    return np.mean(res)
```

In [36]:

```python
# this function runs grid search on several parameters
def catboost_param_tune(params,train_set,train_label,n_splits=3):
    ps = paramsearch.paramsearch(params)
    # search 'border_count', 'l2_leaf_reg' etc. individually
    #   but 'iterations','learning_rate' together
    for prms in chain(ps.grid_search(['border_count']),
                      ps.grid_search(['l2_leaf_reg']),
                      ps.grid_search(['iterations','learning_rate']),
                      ps.grid_search(['depth'])):
        print(n_splits)
        res = cross_val_test(prms,train_set,train_label,n_splits)
        # save the crossvalidation result so that future iterations can reuse the best parameters
```

SHAP library using catboost classifier implementation.



Features Selection - Final draft

Based on the previous shap plots we can obtain the variables of interest from the sum and mean datasets to test our models, we also add the variables from the count dataset and the provider and target.

```
In [39]:
df = _total[['Claim_Days_sum','InscClaimAmtReimbursed_sum', 'N_Extra_Claims_sum',
             'Claim_Days_mean','N_Extra_Claims_mean', 'Same_Physician_mean','AttendingPhysician_mean','Missing_Deductible_Amount_Paid_mean',
             'BeneID_count','ClaimID_count',
             'Provider','target']]
```

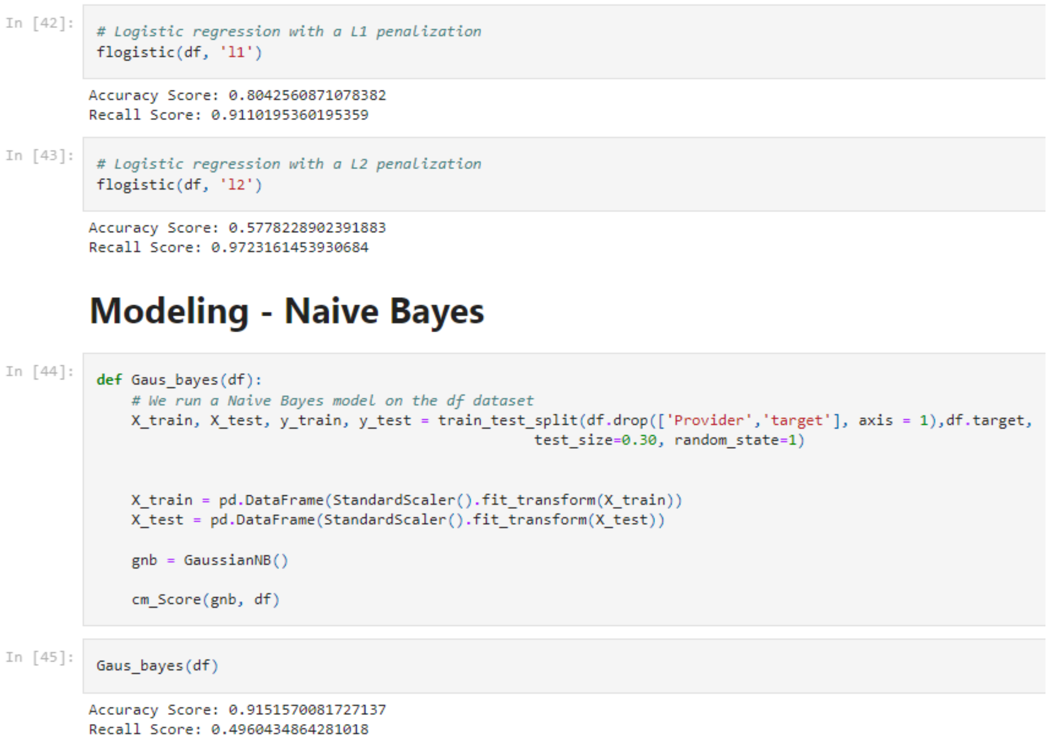
Modeling - Catboost

```
In [40]:
def cm_Score(model, df):
    # We create a function to do a cross validation score on recall and accuracy
    recall = np.mean(cross_val_score(model, df.drop(['Provider','target'], axis = 1), df.target, cv=3, scoring='recall'))
    accuracy = np.mean(cross_val_score(model, df.drop(['Provider','target'], axis = 1), df.target, cv=3, scoring='accuracy'))

    print('Accuracy Score: {}'.format(accuracy))
    print('Recall Score: {}'.format(recall))
```

```
In [ ]:
```

```
In [41]:
#Modeling - Logistic Regression
def flogistic(df, penalty):
    # We create this function to run a logistic regression with the given penalty parameter
    X_train, X_test, y_train, y_test = train_test_split(df.drop(['Provider','target'], axis = 1),df.target,
                                                        test_size=0.30, random_state=1)
```

Model building



```
In [42]:
# Logistic regression with a L1 penalization
flogistic(df, 'l1')

Accuracy Score: 0.8042560871078382
Recall Score: 0.9110195360195359
```

```
In [43]:
# Logistic regression with a L2 penalization
flogistic(df, 'l2')

Accuracy Score: 0.5778228902391883
Recall Score: 0.9723161453930684
```

Modeling - Naive Bayes

```
In [44]:
def Gaus_bayes(df):
    # We run a Naive Bayes model on the df dataset
    X_train, X_test, y_train, y_test = train_test_split(df.drop(['Provider','target'], axis = 1),df.target,
                                                        test_size=0.30, random_state=1)


    X_train = pd.DataFrame(StandardScaler().fit_transform(X_train))
    X_test = pd.DataFrame(StandardScaler().fit_transform(X_test))

    gnb = GaussianNB()

    cm_Score(gnb, df)
```

```
In [45]:
Gaus_bayes(df)

Accuracy Score: 0.9151570081727137
Recall Score: 0.4960434864281018
```

Accuracy score and recall score of logistic regression and nave bayes algorithm.

- Compare the accuracy of various algorithms and constantly upload work on GitHub to keep the progress saved.

5. Conclusion and Future Plan

- Few conclusions from the work done yet are that firstly, models like LIME and SHAP have immensely changed the healthcare sector for the better. New treatments can be implemented just because there is an assurance of a very high accuracy whose functioning can be explained to people during its implementation which is the main objective of explainable AI.

- Secondly, black box models may give high accuracy which will be implemented in the small capstone project done using the lordosis data set, but we don't have a mechanism to explain why it is getting higher accuracy compared to SHAP.
- For the future, a similar dataset will be taken for LIME implantation and high accuracy will be the aim to achieve. The data will be visualized to show how LIME models work and which features will best suit them.
- The observations and conclusions will be described and told in detail in the minor project content which will help the reader give a detailed overview of the work done on the topic.
- Getting the paper published using the IEEE website.

References

[1] Parvathaneni Naga Srinivasu , 1N. Sandhya , 1RutvijH. Jhaveri , 2 and Roshani Raut 13 June 2022 "From Blackbox to Explainable AI in Healthcare: Existing Tools and Case Studies"

[2] 1,2 Ugochukwu Onwudebelu 1LIPN, Université Sorbonne Paris Nord, 99, avenue jeanbaptiste clément, 93430 Villetaneuse, Paris & 2Department of Computer Science Alex-Ekwueme Federal University of Ndufu-Alike Ikwo (AE-FUNAI), Abakaliki Ebonyi State, Nigeria, September 2021 "The Third Wave of AI: Fraud Detection via Knowledge Graphs within Explainable AI"

[3] SERHIY KANDUL, University of Zurich, Switzerland VINCENT MICHELI, University of Geneva, Switzerland JULIANE BECK, University of St. Gallen, Switzerland MARKUS KNEER, University of Zurich, Switzerland THOMAS BURRI, University of St. Gallen, Switzerland FRANÇOIS FLEURET, University of Geneva, Switzerland MARKUS CHRISTEN, University of Zurich, Switzerland, January 2023, "Explainable AI: A review of the empirical literature"

[4] Dr. Tom Kraus Lene Ganschow Marlene Eisenträger Dr. Steffen Wischmann, April 2022, "Explainable AI - Requirements, Use Cases and Solutions"

[5] dancor7- GitHub account, July 22, 2020,
https://github.com/dancor7/Medical_Fraud_Detection_With_SHAP_Values/blob/master/data.rar