In this report, we document the data preprocessing steps undertaken on a dataset containing beneficiary summary information. The dataset was read from a CSV file, and various cleaning and transformation operations were performed to prepare it for further analysis and modeling.

**1. Data Reading**

The first step involved downloading the dataset from the source in the form of zipfile and extracted as .csv. The data was fetched from the following URL:

[CMS 2008-2010 Data Entrepreneurs’ Synthetic Public Use File (DE-SynPUF) | CMS](https://www.cms.gov/data-research/statistics-trends-and-reports/medicare-claims-synthetic-public-use-files/cms-2008-2010-data-entrepreneurs-synthetic-public-use-file-de-synpuf)

And used below file.

[DE1.0Sample 1](https://www.cms.gov/data-research/statistics-trends-and-reports/medicare-claims-synthetic-public-use-files/cms-2008-2010-data-entrepreneurs-synthetic-public-use-file-de-synpuf/de10-sample-1" \o "CMS 2008-2010 Data Entrepreneurs Synthetic PUF Download Page)

**2. Identifying Significant Null Columns**

Upon loading the data, we performed an initial assessment to identify columns with a significant number of missing values. We found that one column, "BENE\_DEATH\_DT," had a high percentage of missing data, specifically 98.44%. Given this high level of missing information, it was decided that this column was not significant for our modeling purposes, and thus, it was removed from the dataset.

**3. Data Cleansing - Handling Date Format**

We also noticed that the "BENE\_BIRTH\_DT" column was not in a proper date-time format. As an example, one of the first dates in this column was printed to 19230501. This is the format for all the values in the give column. To address this, we converted the "BENE\_BIRTH\_DT" column to a proper date-time format using the "%Y%m%d" format.

**4. Removing Unnecessary Columns**

Furthermore, the "DESYNPUF\_ID" column was identified as a unique and non-null primary key. Given that this column is unlikely to provide significant information for our modeling purposes, it was removed from the dataset to reduce complexity.

**5. Handling Categorical Data**

The column "BENE\_ESRD\_IND" had two distinct values, '0' and 'Y'. To facilitate modeling and analysis, we converted these values to integers, where '0' was retained as '0', and 'Y' was converted to '1' to represent the absence or presence of end-stage renal disease (ESRD).

**6. Further Data Exploration**

Lastly, we explored the "BENE\_BIRTH\_DT" column's day component, and it was observed that all values were '1'. Given this uniformity, the day component was found to be non-significant and, therefore, did not provide valuable information for modeling.

**Diabetes Analysis by Birth Date**

In this section of our analysis, we delve into the relationship between the 'SP\_DIABETES' variable and beneficiaries' birth dates. We aim to understand how the distribution of individuals with different values of 'SP\_DIABETES' changes across birth dates.

**1. Data Preparation**

To begin, we divided the dataset into two groups based on the 'SP\_DIABETES' variable. Those with 'SP\_DIABETES' equal to 1 were grouped as "oned," and those with 'SP\_DIABETES' equal to 2 were grouped as "twod." We then grouped these datasets by 'BENE\_BIRTH\_DT' and summed the 'SP\_DIABETES' values for each birth date. The results were sorted in descending order and labeled as "1" and "2" respectively.

**2. Visualizing the Data**

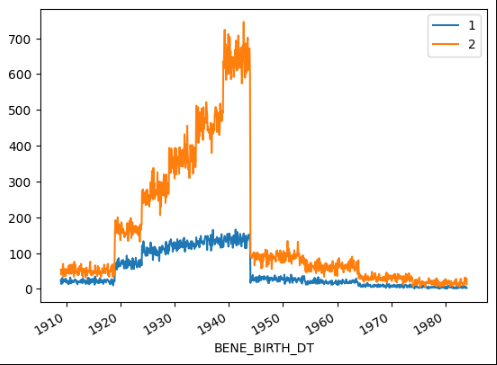
After preparing the data, we created line plots to visualize the trends. Both groups were plotted together, with a legend to distinguish between 'SP\_DIABETES' values 1 and 2. The x-axis represented birth dates, and the y-axis represented the sum of 'SP\_DIABETES' values for each date.

**3. Interpretation**

The resulting plot provides insight into the relationship between birth dates and the 'SP\_DIABETES' values. Notably, we observed a substantial drop in counts for 'SP\_DIABETES' value 2 after the year 1945. A similar but less pronounced trend was observed for 'SP\_DIABETES' value 1.

In summary, this analysis reveals that the distribution of beneficiaries with 'SP\_DIABETES' values 1 and 2 varies significantly based on birth dates. The drop in counts after 1945, especially for 'SP\_DIABETES' value 2, may be indicative of a change in the diagnosis or reporting of diabetes cases, warranting further investigation.

This visualization provides valuable insights for understanding the relationship between birth dates and the prevalence of diabetes among beneficiaries.



**Correlation Analysis**

In this section of our analysis, we explore the relationships between different variables in our dataset by generating a correlation heatmap. This heatmap visually represents the correlation coefficients between various pairs of attributes.

**1. Data Visualization**

To create the correlation heatmap, we used the 'matplotlib' and 'seaborn' libraries. The heatmap was rendered with a larger figure size (25x25) to ensure clarity. Each cell in the heatmap represents the correlation between two variables, with the intensity of the color indicating the strength of the correlation.

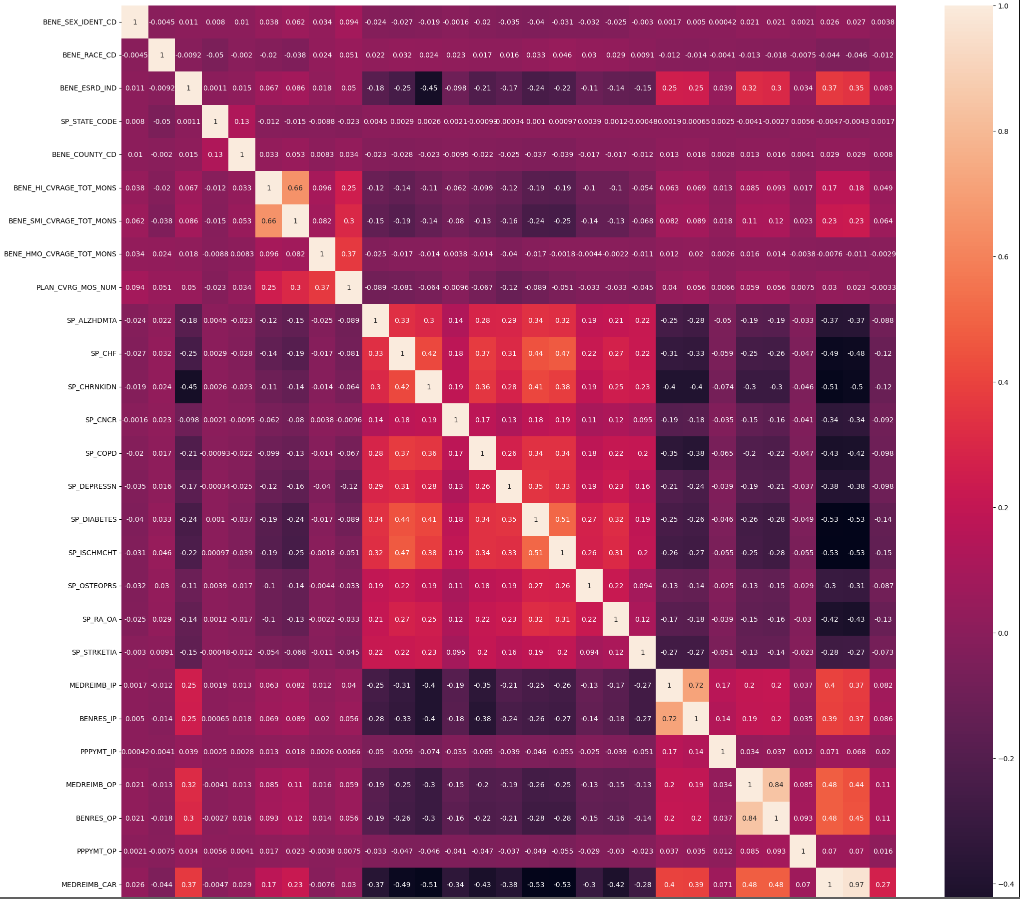
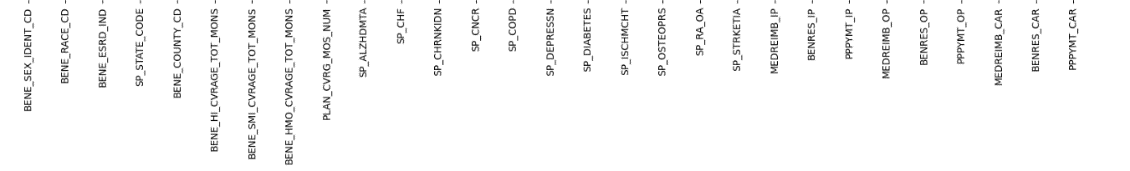
**2. Interpretation**

The heatmap, with annotations enabled, reveals important insights about the dataset. Correlation values range from -1 (perfect negative correlation) to 1 (perfect positive correlation).

* **Positive Correlation**: In our analysis, we identified that the variable 'SP\_DIABETES' is positively correlated with 'SP\_ISCHMCHT' with a maximum correlation coefficient of 0.51. This indicates that as 'SP\_DIABETES' increases, 'SP\_ISCHMCHT' tends to increase as well.
* **Negative Correlation**: We also observed a negative correlation between 'SP\_DIABETES' and two other variables, 'MEDREIMB\_CAR' and 'BENRES\_CAR,' with a minimum correlation coefficient of -0.53 for both. This implies that as 'SP\_DIABETES' increases, 'MEDREIMB\_CAR' and 'BENRES\_CAR' tend to decrease.

The correlation heatmap serves as a valuable tool for identifying potential relationships between variables, which can be crucial for subsequent analysis, modeling, and decision-making.

In summary, this analysis provides a visual representation of the interdependencies within the dataset, shedding light on the relationships between 'SP\_DIABETES' and other attributes.



**Feature Selection**

In this section of our analysis, we focus on feature selection, where we carefully choose the most relevant attributes from our dataset for further analysis and modeling. The selected features will be essential in building predictive models and understanding the factors that influence our target variable, 'SP\_DIABETES.'

**Selected Features**

The following features have been carefully selected from the dataset for their potential importance in our analysis:

1. **BENE\_SEX\_IDENT\_CD**: Beneficiary gender identification code.
2. **BENE\_RACE\_CD**: Beneficiary race code.
3. **BENE\_ESRD\_IND**: End-stage renal disease indicator.
4. **SP\_STATE\_CODE**: State code.
5. **BENE\_COUNTY\_CD**: County code.
6. **BENE\_HI\_CVRAGE\_TOT\_MONS**: Total months of Medicare Part A coverage.
7. **BENE\_SMI\_CVRAGE\_TOT\_MONS**: Total months of Medicare Part B coverage.
8. **BENE\_HMO\_CVRAGE\_TOT\_MONS**: Total months of HMO coverage.
9. **PLAN\_CVRG\_MOS\_NUM**: Number of months covered by a plan.
10. **SP\_ALZHDMTA**: Alzheimer's or dementia indicator.
11. **SP\_CHF**: Congestive heart failure indicator.
12. **SP\_CHRNKIDN**: Chronic kidney disease indicator.
13. **SP\_CNCR**: Cancer indicator.
14. **SP\_COPD**: Chronic obstructive pulmonary disease (COPD) indicator.
15. **SP\_DEPRESSN**: Depression indicator.
16. **SP\_ISCHMCHT**: Ischemic heart disease indicator.
17. **SP\_OSTEOPRS**: Osteoporosis indicator.
18. **SP\_RA\_OA**: Rheumatoid arthritis or osteoarthritis indicator.
19. **SP\_STRKETIA**: Stroke or transient ischemic attack indicator.
20. **MEDREIMB\_IP**: Medicare reimbursement for inpatient care.
21. **BENRES\_IP**: Beneficiary responsibility for inpatient care.
22. **PPPYMT\_IP**: Per-patient-per-year payment for inpatient care.
23. **MEDREIMB\_OP**: Medicare reimbursement for outpatient care.
24. **BENRES\_OP**: Beneficiary responsibility for outpatient care.
25. **PPPYMT\_OP**: Per-patient-per-year payment for outpatient care.
26. **MEDREIMB\_CAR**: Medicare reimbursement for carrier claims.
27. **BENRES\_CAR**: Beneficiary responsibility for carrier claims.
28. **PPPYMT\_CAR**: Per-patient-per-year payment for carrier claims.
29. **SP\_DIABETES**: Our target variable, the diabetes indicator.

By selecting these features, we aim to focus on the most relevant factors that may influence the occurrence of diabetes ('SP\_DIABETES') among beneficiaries. These features will be used in subsequent modeling and analysis to build predictive models and gain deeper insights into the dataset.

**Addressing Class Imbalance**

In the course of our analysis, we observed a class imbalance issue within our dataset. Specifically, the target variable, 'SP\_DIABETES,' exhibits a disparity in the distribution of its classes. To rectify this imbalance and ensure our modeling is accurate and robust, we will employ the Synthetic Minority Over-sampling Technique (SMOTE).

**Class Imbalance Overview**

The class imbalance issue is reflected in the distribution of the 'SP\_DIABETES' variable. The 'value\_counts()' function revealed the following distribution:

SP\_DIABETES

2 72292

1 44060

As is evident, there is a significant disparity between the two classes, which could lead to biased predictions when building a predictive model. To address this issue, we will apply the SMOTE oversampling technique.

**SMOTE (Synthetic Minority Over-sampling Technique)**

SMOTE is a technique commonly used to alleviate class imbalance. It works by generating synthetic samples of the minority class to balance the dataset. This artificial oversampling ensures that the model does not favor the majority class and can better generalize to predict the minority class.

By applying SMOTE, we aim to balance the distribution of 'SP\_DIABETES' to ensure that both positive and negative cases are adequately represented in our dataset. This approach will enhance the model's ability to make accurate predictions, especially for the minority class.

SP\_DIABETES

2 72292

1 72292

In summary, addressing the class imbalance is a critical step in our analysis, and SMOTE will be utilized to create a more balanced dataset for our modeling and predictive tasks.

**Data Preprocessing and Model Selection**

In this section, we outline the steps taken to prepare the data and the selection of classification models for predicting the 'SP\_DIABETES' variable.

**Data Preprocessing**

1. **Feature Normalization:** Given the dataset's potential variations in the scale of different features, we employed feature normalization using the **Normalizer** from scikit-learn. This process scales the features to ensure that they all have the same weight in the modeling process, contributing to more accurate and consistent results.
2. **Addressing Class Imbalance:** As previously noted, the dataset suffers from a class imbalance, particularly concerning the 'SP\_DIABETES' variable. To rectify this, we utilized the Synthetic Minority Over-sampling Technique (SMOTE) from the **imblearn** library. SMOTE generates synthetic samples of the minority class to balance the dataset, ensuring a more equitable representation of both positive and negative diabetes cases.

**Model Selection**

For building predictive models, we have chosen a set of classification algorithms:

1. **Logistic Regression:** A fundamental algorithm for binary classification tasks.
2. **Gaussian Naive Bayes:** A probabilistic classification algorithm based on Bayes' theorem.
3. **Decision Tree Classifier:** A non-linear model that uses a tree-like structure for classification.
4. **K-Nearest Neighbors (KNN):** A model that classifies data points based on their proximity to other data points.
5. **XGBoost Classifier:** An ensemble algorithm known for its high performance.
6. **Random Forest Classifier:** Another ensemble algorithm that builds multiple decision trees.
7. **Linear Support Vector Classifier (SVC):** A linear classification algorithm that aims to find the optimal hyperplane for separation.

These models were selected for their diversity and applicability to the task of diabetes prediction. By utilizing a variety of algorithms, we aim to assess which model performs best on our balanced dataset.

The combination of data preprocessing and model selection is a crucial step in our analysis, as it lays the foundation for building predictive models that can accurately classify diabetes cases based on the dataset's features.

**Model Evaluation and Comparison**

In this section, we evaluate and compare the performance of various classification models that were trained on our balanced dataset for predicting 'SP\_DIABETES.' The models considered include Logistic Regression, Naive Bayes, Decision Tree, K-Nearest Neighbors (KNN), XGBoost, Random Forest, and Linear Support Vector Classifier (SVC).

**Model Performance Evaluation**

Each model was assessed for its predictive accuracy using cross-validation. The accuracy of each model was recorded, and the results are presented in a tabular format.

| **Model** | **Accuracy** |
| --- | --- |
| XGBoost | 0.824407 |
| Random Forest | 0.811127 |
| Linear Support Vector | 0.797959 |
| K-Nearest Neighbors | 0.792564 |
| Naive Bayes | 0.792481 |
| Decision Tree | 0.773247 |
| Logistic Regression | 0.704663 |

As evident from the table, the XGBoost Classifier exhibited the highest accuracy as 82.44%, surpassing all other models. This makes it the top-performing model for predicting diabetes based on our dataset and the chosen features.

The comparison of models provides a clear indication of which classification algorithm is best suited for this specific task. The results affirm the effectiveness of XGBoost as a robust and accurate predictive model for diabetes classification.