TECHNICAL REPORT APPLICATIONS OF MACHINE LEARNING IN MEDICINE

Faculty of Engineering, Ariel University, Department of Industrial Engineering and Management

Abstract: investigates the effectiveness of supervised Machine Learning techniques for predicting Chronic Kidney Disease (CKD).

Written by: Yogev Ladani: 207754524

Niv Levi: 314628165

Lecturer: Dr.Orit Refaeli

Date: 28.07.24

ARIEL UNIVERSITY

TABLE OF CONTENTS

ARIEL UNIVERSITY

Abstract

This report investigates the effectiveness of supervised Machine Learning techniques for predicting Chronic Kidney Disease (CKD) using a dataset collected from patients hospitalized in 2008 with follow-up data available until 2017. Various Machine Learning models, including Logistic Regression, Decision Trees, Random Forests, and Support Vector Machines (SVM), were applied to the dataset. The models were evaluated based on their ability to accurately predict the onset of CKD events, with feature selection methods used to enhance model performance. The findings suggest that Random Forest models coupled with feature selection techniques provide a robust approach to predicting CKD.

Introduction and Purpose of the Work

Chronic Kidney Disease (CKD) is a significant public health issue characterized by a gradual loss of kidney function over time. Early prediction and diagnosis of CKD can lead to better patient outcomes and reduced healthcare costs. Machine Learning offers a powerful toolset for predictive analytics in healthcare, potentially enabling earlier interventions and better resource allocation. The purpose of this work is to assess the effectiveness of different supervised Machine Learning algorithms in predicting CKD events. By leveraging a comprehensive dataset with various clinical features, we aim to identify the most predictive factors and evaluate the performance of different models in forecasting CKD progression. This study not only explores the predictive power of various models but also aims to provide insights into the clinical features most strongly associated with CKD development. This research has significant implications for healthcare beyond just predictive accuracy. By enhancing our ability to identify high-risk patients, healthcare providers can focus resources more efficiently on those who may benefit most from preventive measures. Moreover, the insights gained from feature importance analysis could contribute to a better understanding of CKD risk factors, potentially informing future clinical research and practice guidelines. Ultimately, this work seeks to contribute to the growing body of knowledge at the intersection of machine learning and nephrology, with the goal of improving patient care and outcomes in the management of chronic kidney disease.

Methods

Data Collection

The dataset used in this study includes clinical data from 421 patients hospitalized in 2008, with follow-up data collected until 2017. The dataset consists of several variables, such as age, BMI, history of various diseases, medication use, and baseline measurements of cholesterol, creatinine, blood pressure, and estimated glomerular filtration rate (eGFR).

Exploratory Data Analysis (EDA)

EDA was conducted to understand the distribution and behavior of the data, identify patterns, and detect outliers. The following steps were performed:

- Summary Statistics: Calculated summary statistics for all numerical columns to get an overview of the data.
- Distribution Plots: We created histograms and density plots for numerical variables to visualize their distributions. While most variables approximated normal distributions to varying degrees, some showed significant skewness. For instance, TriglyceridesBaseline showed right-skew, while HgbA1C exhibited left-skew (*Figure 1)*.
- Box Plots: We initially used box plots to identify outliers and understand the spread of the data. These plots revealed significant outliers in several variables, particularly in TriglyceridesBaseline, HgbA1C, and CreatinineBaseline *(Figure 2)*. After applying our approach of capping values at the 2.5th and 97.5th percentiles, the revised box plots showed a marked reduction in extreme outliers while preserving the overall distribution shapes. This treatment effectively mitigated the impact of extreme values on our subsequent analyses *(Figure 3)*.

- Categorical Analysis: We used pie charts to visualize demographic and diagnostic categorical variables. Demographic data showed a balanced gender distribution (50.9% female, 49.1% male) and varied age groups *(Figure 4)*. Diagnostic variables revealed significant class imbalances, particularly in the EventCKD35 column (88.6% without CKD) *(Figure 5)*. This imbalance will be addressed later in our modeling process to ensure fair prediction across classes.

The numerical columns analyzed included:

- Age Baseline
- Cholesterol Baseline
- Triglycerides Baseline
- Hemoglobin A1C (HbA1c)
- Creatinine Baseline
- eGFR Baseline
- Systolic Blood Pressure (sBP Baseline)
- Diastolic Blood Pressure (dBP Baseline)
- BMI Baseline
- Time to Event (Months)

Handling Outliers

Outliers were detected using box plots and were handled by capping values at the 0.025 and 0.975 percentiles. Specifically, values below the 2.5th percentile were set to the 2.5th percentile value, and values above the 97.5th percentile were set to the 97.5th percentile value. This approach was applied to the same numerical columns analyzed during the EDA.
This approach of capping values at the 2.5th and 97.5th percentiles was chosen as it retains more data compared to complete removal of outliers, while still mitigating the impact of extreme values on the model's performance.

Preprocessing

Data preprocessing involved several steps:

- Handling Missing Values: Missing data were imputed using appropriate statistical methods to ensure the completeness of the dataset. Specifically, for the columns Lipid Profile (Triglycerides) and Hemoglobin A1C (HbA1c), values were filled in according to the average values corresponding to the segmentation of man/woman and age group [1] [2]. These values were determined after researching the significance of each column and what they represent.
- Encoding Categorical Variables: Categorical variables were encoded using one-hot encoding for variables with more than two categories, and binary encoding for binary variables.
- Normalization of Numerical Features: Numerical features were normalized to a standard scale to ensure that all features contribute equally to the model training process.

Features Selection

We employed a two-step approach combining SelectKBest and Recursive Feature Elimination (RFE) for effective feature selection:

SelectKBest: We initially used SelectKBest with ANOVA F-value to identify features most correlated with EventCKD35. The cross-validation F1-scores for different numbers of features (k) were plotted, showing a peak around k=8 *(Figure 6)*. This informed our initial feature selection, guiding us to focus on the top 8 features for further analysis.

Feature Correlation: To validate our selection, we examined the correlation of various features with EventCKD35. The resulting bar chart *(Figure 7)* displayed TimeToEventMonths with the highest correlation, followed by eGFRBaseline and HgbA1C. This visualization corroborated the importance of these features in predicting CKD progression and aligned with our SelectKBest results.

RFECV (RFE with Cross-Validation): We then applied RFECV to determine the optimal number of features. The graph of cross-validation scores against the number of features *(Figure 8)* showed a plateau in accuracy after 4-6 features. This suggested that a compact set of 4-6 features could effectively capture the necessary information for our model.

RFE Feature Importance: Finally, we visualized the feature importances as determined by RFE *(Figure 9)*. This bar plot revealed eGFRBaseline and TimeToEventMonths as strong negative predictors, while HistoryDiabetes and HgbA1C emerged as important positive predictors. This visualization provided crucial insights into the relative importance and directional impact of each feature on CKD progression prediction.

This dual approach of filter and wrapper methods balanced statistical relevance with predictive power, optimizing our feature selection for the classification task at hand. The selected features encompass key aspects of patient history, current health status, and disease progression, providing a robust foundation for predicting CKD progression.

Handling Class Imbalance

Class imbalance is a common problem in machine learning where some classes are underrepresented compared to others, leading to biased models that perform poorly on the minority class. To address this issue in our dataset, we explored various sampling techniques and evaluated their effectiveness [3].

Analysis of Target Variable Distribution

The initial distribution of the target variable EventCKD35 showed a significant class imbalance *(Figure 10)*:

- Class 0 (no event): ~430 instances
- Class 1 (event): ~60 instances

This imbalance is clearly visible in the distribution chart before any sampling techniques were applied.

Sampling Techniques Explored

We applied and compared three different techniques to address the class imbalance:

1. **Over-sampling:** This technique increases the number of samples in the minority class by duplicating existing minority class samples. While simple, it can lead to overfitting.

2. **SMOTE (Synthetic Minority Over-sampling Technique):** SMOTE generates synthetic samples for the minority class by interpolating between existing minority class samples. This creates new, unique instances, helping to balance the class distribution more effectively.

3. **Under-sampling:** This method reduces the number of samples in the majority class by randomly removing instances. While effective in balancing classes, it risks losing important information from the majority class.

The distribution plots clearly illustrate the effect of each sampling technique *(Figure 11)*:

- SMOTE: The plot shows a balanced dataset with approximately 300 instances in each class. This method creates synthetic samples for the minority class, effectively balancing the distribution without simple duplication.
- Under-sampling: The plot demonstrates a balanced but significantly reduced dataset, with about 35-40 instances in each class. This dramatic reduction in overall sample size is evident.

- Over-sampling: The distribution plot displays an equal number of instances for both classes (approximately 300 each), artificially increasing the minority class to match the majority through duplication.

Chosen Method: SMOTE

We selected SMOTE as our preferred method for addressing class imbalance. SMOTE's ability to generate synthetic samples rather than simply duplicating existing ones helps overcome the issue of overfitting associated with simple over-sampling. By creating new, unique samples through interpolation, SMOTE provides a better representation of the minority class. This enhances the classifier's ability to generalize and improve performance on imbalanced datasets, making it a more robust and effective solution for our analysis.

The distribution plots effectively visualize how each technique impacts the class balance, with SMOTE achieving a balanced distribution without the drawbacks of information loss (under-sampling) or exact duplication (over-sampling).

Model Training and Evaluation

Initial Training with SMOTE:

We first addressed the class imbalance issue in the training data using SMOTE (Synthetic Minority Over-sampling Technique). The models were then trained on this balanced dataset and tested on the original, imbalanced test set to avoid overfitting. The results were as follows:

- Logistic Regression: Accuracy: 0.7150, Precision: 0.7500, Recall: 0.6857, F1 Score: 0.7164
- Decision Tree: Accuracy: 0.5950, Precision: 0.6111, Recall: 0.6286, F1 Score: 0.6197
- Random Forest: Accuracy: 0.6400, Precision: 0.6737, Recall: 0.6095, F1 Score: 0.6400
- SVM: Accuracy: 0.7050, Precision: 0.7500, Recall: 0.6571, F1 Score: 0.7005

Referring to Figure 13, we can observe the ROC curves, Precision-Recall curves, and F1 Score vs. Threshold for each model. The ROC curve shows that all models perform better than random chance, with Logistic Regression and SVM appearing slightly superior. The Precision-Recall curve demonstrates how the models balance precision and recall after training on the SMOTE-balanced data.

Hyperparameter-Tuned Models with SMOTE:

After applying hyperparameter tuning, the model performances were:

- Logistic Regression: Accuracy: 0.7350, Precision: 0.7660, Recall: 0.6990, F1 Score: 0.7310
- Decision Tree: Accuracy: 0.6650, Precision: 0.6957, Recall: 0.6214, F1 Score: 0.6564
- Random Forest: Accuracy: 0.6950, Precision: 0.6944, Recall: 0.7282, F1 Score: 0.7109
- SVM: Accuracy: 0.7150, Precision: 0.6983, Recall: 0.7864, F1 Score: 0.7397

Figure 14 shows the performance metrics for these tuned models. Notably, all models show improved performance across various metrics after tuning, with SVM showing the highest F1 score.

Models without SMOTE:

For comparison, we also evaluated the models without applying SMOTE to handle class imbalance:

- Logistic Regression: Accuracy: 0.9122, Precision: 0.8889, Recall: 0.4000, F1 Score: 0.5517
- Decision Tree: Accuracy: 0.8851, Precision: 0.6154, Recall: 0.4000, F1 Score: 0.4848
- Random Forest: Accuracy: 0.9122, Precision: 0.8889, Recall: 0.4000, F1 Score: 0.5517
- SVM: Accuracy: 0.9054, Precision: 0.8750, Recall: 0.3500, F1 Score: 0.5000

Figure 15 illustrates the performance of these models without SMOTE. We can observe significant differences in the precision-recall trade-offs and F1 scores across different thresholds compared to the SMOTE-balanced models.

Comparison and Insights:

- The models trained with SMOTE (*Figure 13* and *14*) generally show better recall and F1 scores compared to the models without SMOTE (Image 3).
- In the SMOTE-balanced scenario, Logistic Regression initially showed the best overall performance with the highest F1-score.
- After hyperparameter tuning (still with SMOTE), the SVM model showed the best F1-score, demonstrating the importance of tuning for optimal performance.
- Without SMOTE (*Figure 15*), we see higher accuracy and precision but much lower recall, especially for the minority class. This highlights the significant impact of class imbalance on model performance.
- The Logistic Regression model consistently performed well across different stages, showing the best F1-score initially and after tuning in the SMOTE-balanced scenario.

These results underscore the importance of both addressing class imbalance through techniques like SMOTE and performing hyperparameter tuning to achieve optimal model performance. The significant changes in performance metrics across different scenarios (with vs. without SMOTE, tuned vs. untuned) emphasize the need for thorough evaluation and careful consideration of model selection based on the specific requirements of the prediction task.

Explanatory AI: SHAP Model Analysis

Logistic Regression (*Figure 16*):

- eGFRBaseline: Higher values (red dots) have negative SHAP values, decreasing class 1 probability.
- TimeToEventMonths: Higher values (red dots) increase class 1 probability.
- HgbA1C: Shows a non-linear relationship, with both high and low values affecting predictions in both directions.
- HistoryDiabetes: Having a history of diabetes (red dots) decreases class 1 probability.

Decision Tree (*Figure 17*):

- TimeToEventMonths: Higher values increase class 1 probability.
- HgbA1C: Higher values tend to decrease class 1 probability.
- eGFRBaseline: Similar to Logistic Regression, higher values decrease class 1 probability.
- HistoryDiabetes: Minimal impact, slightly decreasing class 1 probability when it does have an effect.

Random Forest (*Figure 18*):

- TimeToEventMonths: Higher values significantly increase class 1 probability.
- HgbA1C: Complex relationship, with higher values generally decreasing class 1 probability.
- eGFRBaseline: Higher values decrease class 1 probability.
- HistoryDiabetes: Minimal but generally negative impact.

SVM (*Figure 19*):

- TimeToEventMonths: Most significant positive impact on class 1 probability.
- HgbA1C: Higher values negatively impact class 1 probability.
- eGFRBaseline: Higher values decrease class 1 probability.
- HistoryDiabetes: Generally decreases class 1 probability.

General Observations Across Models:

- TimeToEventMonths is consistently the most influential feature, with higher values increasing class 1 likelihood.
- eGFRBaseline shows a consistent negative impact on class 1 probability across all models.
- HgbA1C demonstrates a complex relationship, varying across models.
- HistoryDiabetes generally has a smaller impact but tends to decrease class 1 probability.

Clinical Relevance and Recommended Actions:

TimeToEventMonths:

- Insight: Critical predictor for clinical events.
- Action: Close patient monitoring as critical events approach; timely interventions to improve outcomes.

HgbA1C:

- Insight: Indicator of blood sugar control and risk factor for complications.
- Action: Regular monitoring and management of blood sugar levels; interventions to lower HgbA1C through medication, lifestyle changes, and dietary modifications.

eGFRBaseline:

- Insight: Measure of kidney function; lower values indicate worse outcomes.
- Action: Monitor kidney function closely in patients with lower eGFR; manage progression through medication adjustments, dietary changes, and addressing comorbidities.

HistoryDiabetes:

- Insight: Risk factor for various complications.
- Action: Comprehensive care including regular monitoring, risk factor management, and preventive measures for complications.

In conclusion, the SHAP analysis provides valuable insights into feature importance and their clinical implications across different models. This enables more informed decision-making and personalized patient care strategies, taking into account the nuanced impacts of each feature as revealed by different modeling approaches. The consistency of certain features' impacts across models (e.g., TimeToEventMonths and eGFRBaseline) underscores their importance in clinical decision-making, while the variability in others (e.g., HgbA1C) highlights the need for model-specific interpretations in some cases.

Subgroup Analysis Using K-means Clustering

Throughout our analysis, we employed various methods at each stage, utilizing cross-validation techniques to determine the most effective approaches. After incorporating clustering insights, we enhanced our feature set and constructed a new model that leverages all the knowledge gained during our investigation. Our final model incorporates two key findings:

1. The preferred method for addressing class imbalance is SMOTE (Synthetic Minority Over-sampling Technique).

2. The initial pipeline proved to be the most effective approach for prediction.

Our subgroup analysis employed K-means clustering to identify distinct patient groups within our CKD dataset. We used the elbow method to determine the optimal number of clusters and visualized the results using Principal Component Analysis (PCA).

Determining the Optimal Number of Clusters

Figure 20 shows the distortion score elbow plot for K-means clustering. The elbow point is identified at k=5, with a distortion score of 921.273, indicating that five clusters provide the optimal balance between model complexity and explanatory power.

Cluster Visualization and Characteristics

Figure 21 presents a 3D scatter plot of the K-means clustering results, projecting the data points onto the first three principal components. The plot reveals five distinct clusters, each representing a subgroup of patients with similar characteristics:

Cluster 0 (Blue):

Largest cluster, concentrated in the lower left of the plot

Characterized by low HgbA1C values and high eGFRBaseline values

Predominantly associated with no history of diabetes (HistoryDiabetes = 0)

Wide range of TimeToEventMonths, with a tendency towards higher values

Cluster 1 (Orange):

Smaller group, spread out in the upper right region

Shows moderate HgbA1C values and moderate to high eGFRBaseline values

Associated with a history of diabetes (HistoryDiabetes = 1)

Distinct distribution in TimeToEventMonths, concentrated at lower values

Cluster 2 (Green):

Medium-sized group, positioned in the lower right

Exhibits moderate to high HgbA1C values and low to moderate eGFRBaseline values

Mostly associated with no history of diabetes (HistoryDiabetes = 0)

Wide range of TimeToEventMonths

Cluster 3 (Red):

Smallest group, tightly clustered in the center-right

Displays the highest HgbA1C values and the lowest eGFRBaseline values in the cohort

Associated with a history of diabetes (HistoryDiabetes = 1)

Wide range of TimeToEventMonths, but tends towards lower values

Cluster 4 (Purple):

Medium-sized group, spread across the upper left to center

Shows moderate HgbA1C values and low to moderate eGFRBaseline values

Associated with a history of diabetes (HistoryDiabetes = 1)

Wide range of TimeToEventMonths

The clear separation between most clusters in the 3D visualization confirms that the K-means algorithm has successfully identified distinct patient subgroups based on the key variables. Some overlap between clusters, particularly in the central region, suggests that there may be patients with characteristics that fall between cluster definitions.

These distinct clusters provide valuable insights into different subgroups of CKD patients, potentially reflecting varying disease progression patterns and risk profiles. The identification of these subgroups can aid in tailoring management strategies and predicting disease trajectories. For instance, patients in Cluster 3, with high HgbA1C and low eGFR, may require more aggressive management of both diabetes and kidney function.

The performance metrics for our final models, incorporating these clustering insights, are as follows:

Model/ Metric	*Accuracy*	*Precision*	*Recall*	*F1 Score*
Logistic Regression	0.7500	0.7184	0.7789	0.7475
Decision Tree	0.7350	0.7019	0.7684	0.7337
Random Forest	0.7800	0.7297	0.8526	0.7864
Support Vector Machine (SVM)	0.7900	0.7431	0.8526	0.7941

Figure 22 figure illustrates the performance of these models through ROC curves, Precision-Recall curves, and F1 Score vs. Threshold plots.

Analyzing these results, we can observe that:

- The SVM model demonstrates the best overall performance, with the highest accuracy (0.7900) and F1 score (0.7941). It also shows excellent recall (0.8526), indicating its strong ability to identify positive cases.
- The Random Forest model closely follows the SVM in performance, with slightly lower but still impressive metrics. It shares the same high recall as the SVM (0.8526) and has the second-highest F1 score (0.7864).
- Logistic Regression and Decision Tree models, while not performing as well as SVM and Random Forest, still show respectable results, with accuracies above 0.73 and F1 scores above 0.73.
- All models demonstrate a good balance between precision and recall, as reflected in their F1 scores, which are consistently above 0.73.
- The ROC curves in the figure show that all models perform significantly better than random chance, with curves well above the diagonal line. The SVM and Random Forest curves appear to dominate, aligning with their superior performance metrics.
- The Precision-Recall curves illustrate how well the models balance precision and recall across different thresholds. Again, the SVM and Random Forest models seem to maintain higher precision across a wider range of recall values.
- The F1 Score vs. Threshold plots provide insight into how the models' F1 scores change with different classification thresholds. This can be particularly useful for fine-tuning the models' decision boundaries in practical applications.

In conclusion, while all models show promising results, the SVM and Random Forest models stand out as the top performers for our final predictive task. The addition of clustering-derived features and the application of SMOTE for addressing class imbalance have contributed to these strong results across all evaluated models.

Findings

Model Performance:

- SVM and Random Forest models consistently outperformed other models, with SVM achieving the highest accuracy (0.7900) and F1 score (0.7941).
- The use of SMOTE for addressing class imbalance significantly improved model performance, particularly in terms of recall for the minority class.
- Hyperparameter tuning further enhanced model performance across all algorithms.

Feature Importance:

- TimeToEventMonths emerged as the most influential feature across all models.
- eGFRBaseline, HgbA1C, and HistoryDiabetes were also identified as key predictors of CKD progression.
- SHAP analysis revealed complex, non-linear relationships between some features (e.g., HgbA1C) and CKD risk.

Patient Subgroups:

- K-means clustering identified five distinct patient subgroups, each with unique characteristics related to HgbA1C levels, eGFRBaseline, and diabetes history.
- These subgroups suggest different risk profiles and potential disease progression patterns among CKD patients.

Clinical Implications:

- The analysis highlights the importance of regular monitoring of key biomarkers, particularly eGFR and HgbA1C.
- The identification of patient subgroups suggests the potential for tailored management strategies based on cluster characteristics.

Conclusion

This study demonstrates the effectiveness of machine learning techniques in predicting CKD progression. The SVM and Random Forest models, combined with SMOTE for class balancing, offer robust predictive performance. The identification of key predictive features and distinct patient subgroups provides valuable insights for clinical practice.

The findings underscore the importance of regular monitoring of renal function and glycemic control in CKD management. The complex relationships revealed by SHAP analysis highlight the need for nuanced interpretation of clinical data in assessing CKD risk.

The clustering analysis reveals the heterogeneity within the CKD patient population, suggesting that personalized treatment approaches based on these subgroups could potentially improve patient outcomes.

Future research should focus on external validation of these models, exploration of additional features, and investigation of the long-term outcomes of patients within the identified subgroups. Additionally, the development of user-friendly tools based on these models could aid clinicians in risk stratification and treatment planning for CKD patients.

In conclusion, this study represents a significant step towards more accurate and personalized prediction of CKD progression, with potential to improve patient care and outcomes in nephrology.

ARIEL UNIVERSITY

Bibliography

[1] "MedlinePlus," 6 September 2022. [Online]. Available: https://medlineplus.gov/lab-tests/hemoglobin-a1c-hba1c-test/.

[2] B. Morris, "Medscape Logo," 03 November 2021. [Online]. Available: https://emedicine.medscape.com/article/2074115-overview?form=fpf.

[3] A. A. Khan, "Balanced Split: A new train-test data splitting strategy for imbalanced datasets," *arXiv,* 2022.

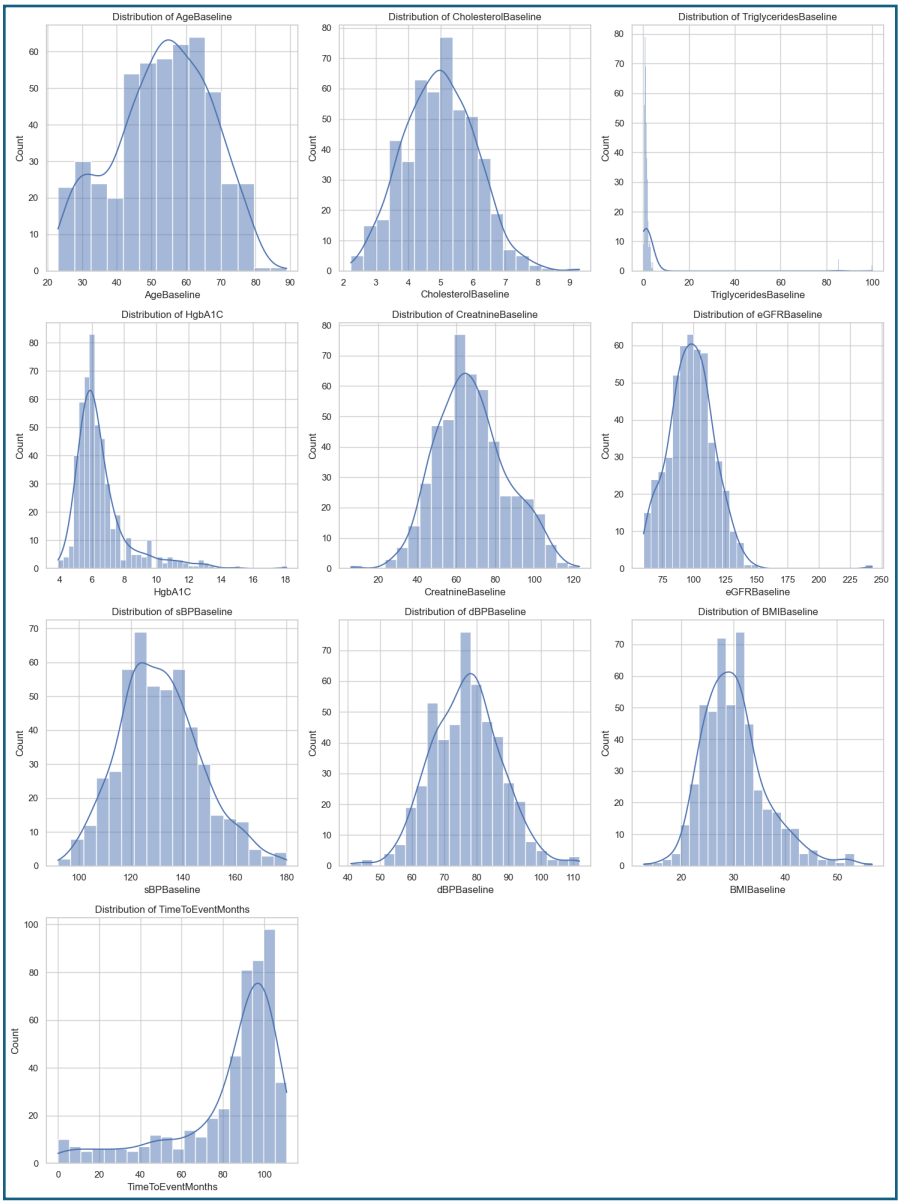


Figure 1- Numerical columns histograms

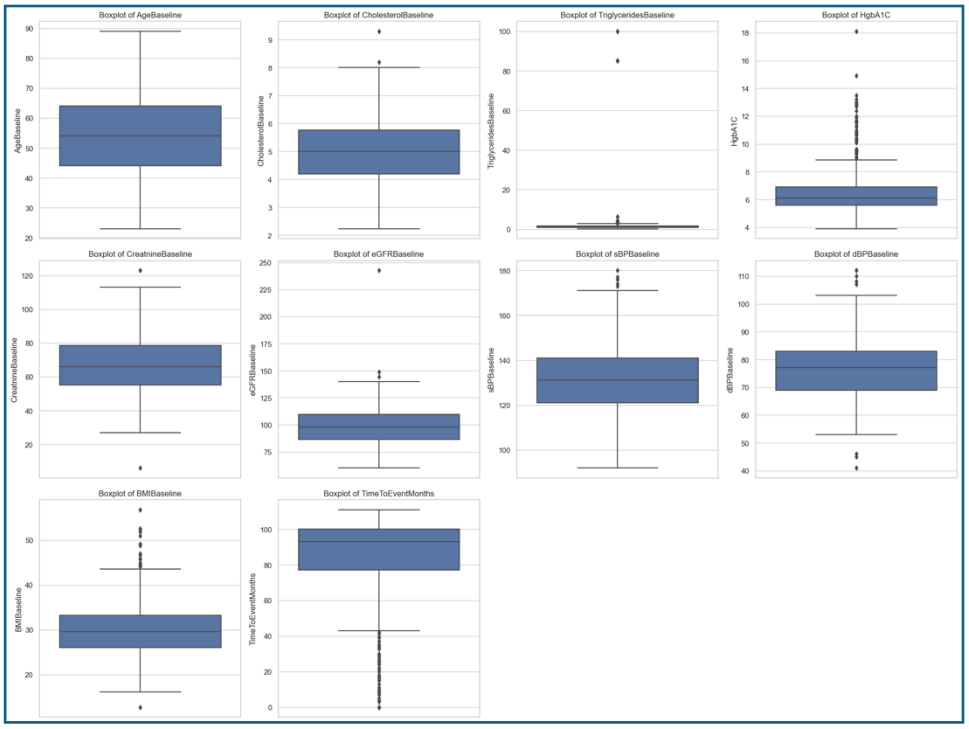


Figure 2- Numerical columns Box plots before handling Outliers

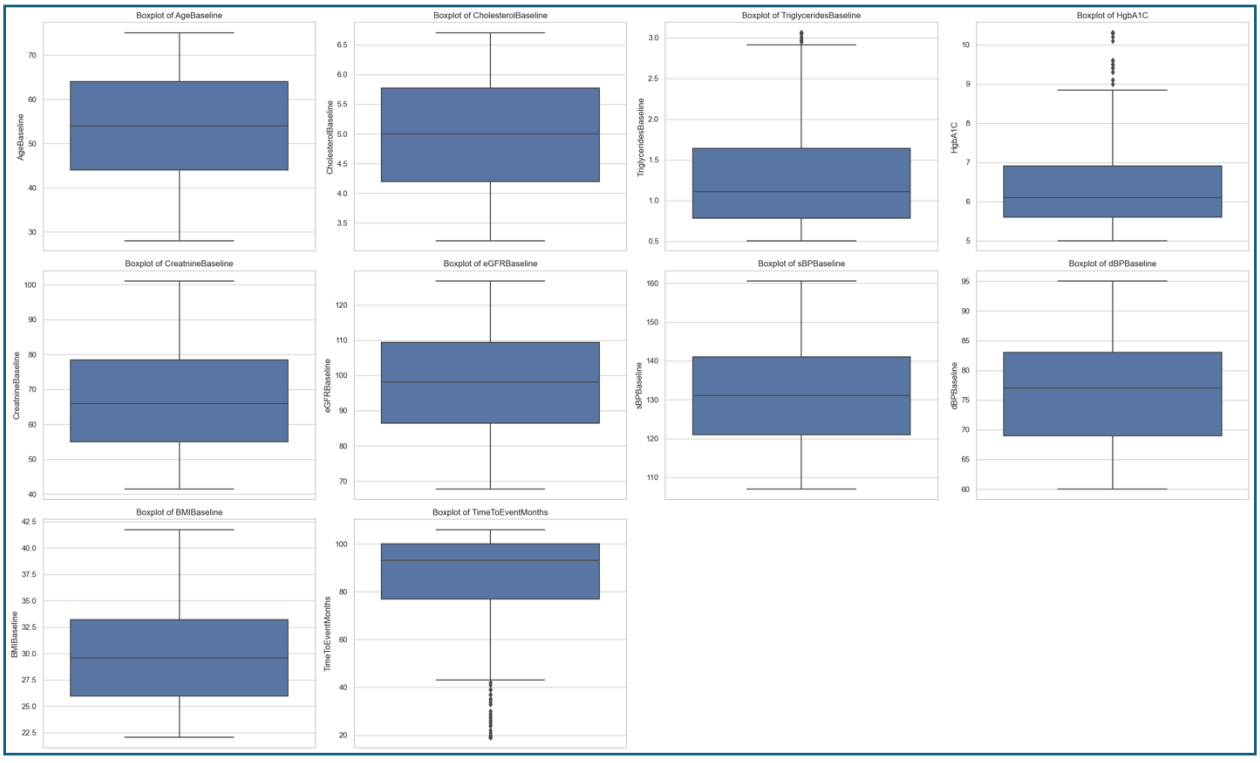


Figure 3-Numerical columns Box plots after handling Outliers

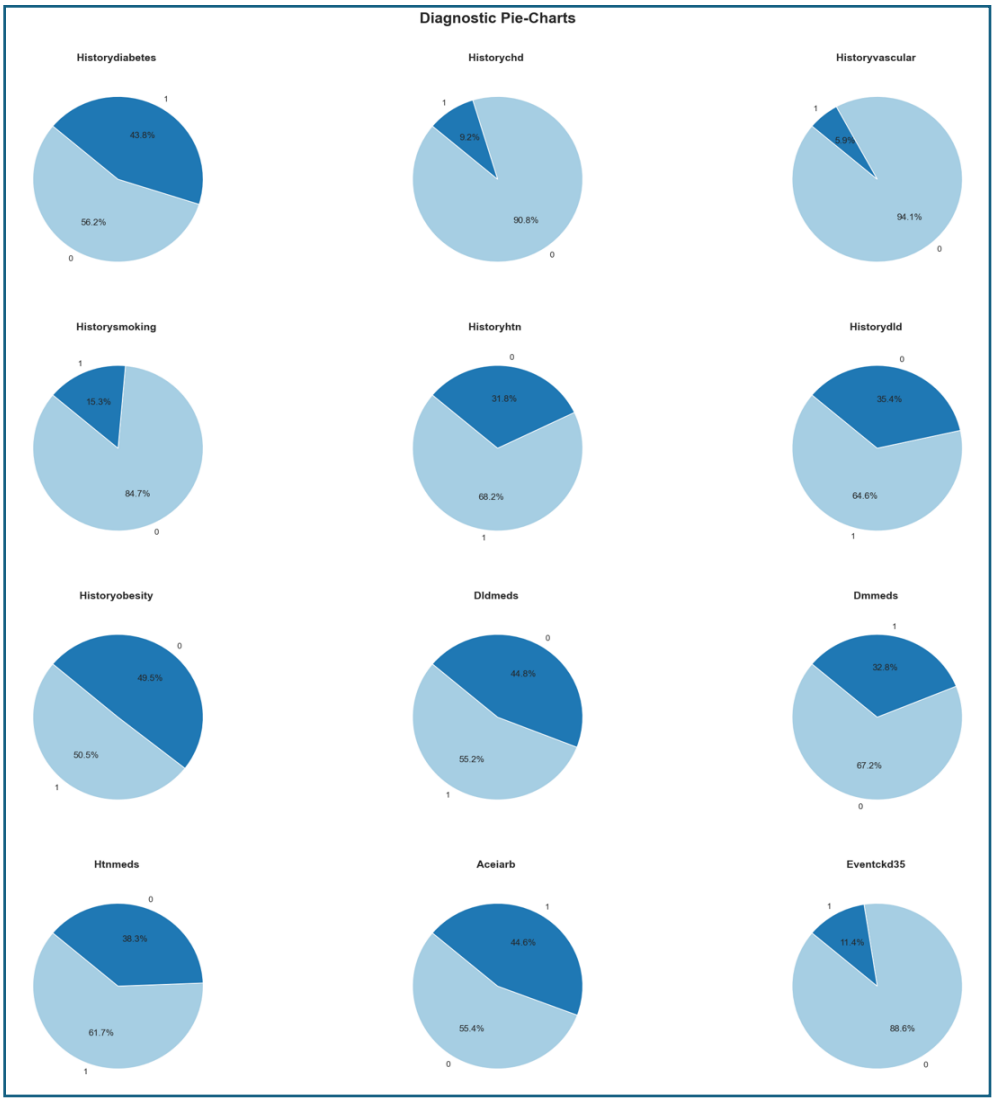


Figure 4- Diagnostic categorical columns Pie plots

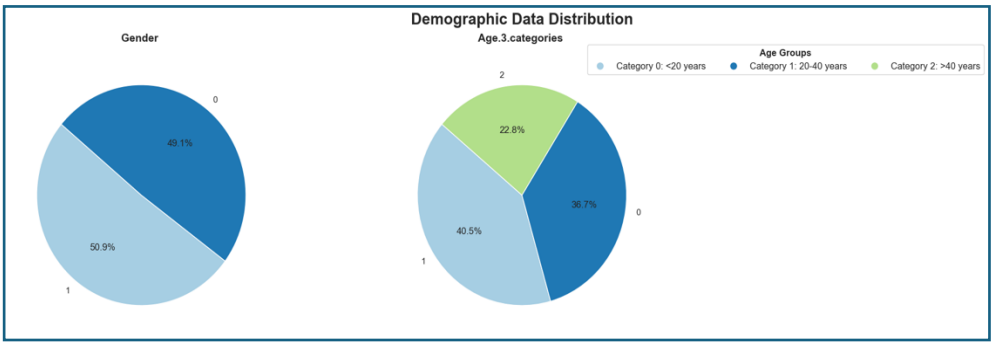


Figure 5- Demographic categorical columns Pie plots

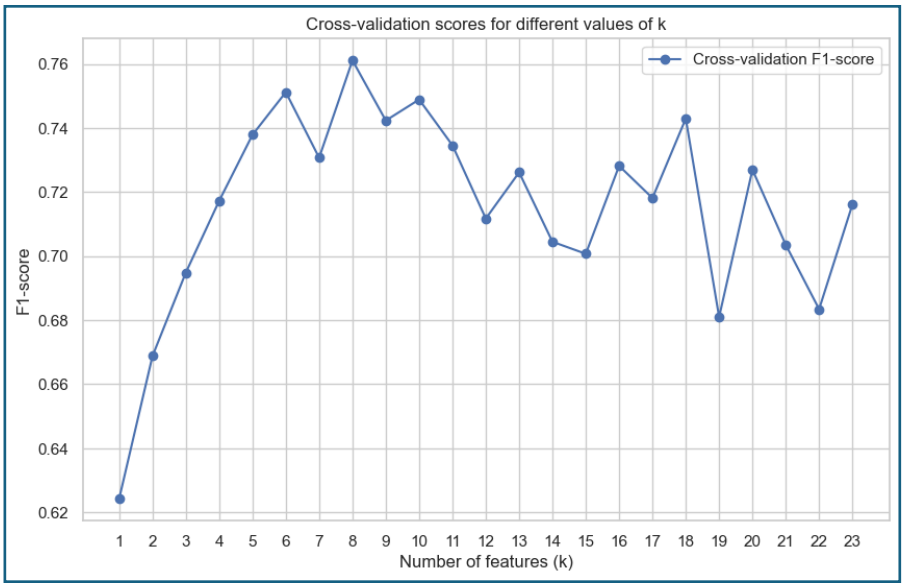


Figure 6

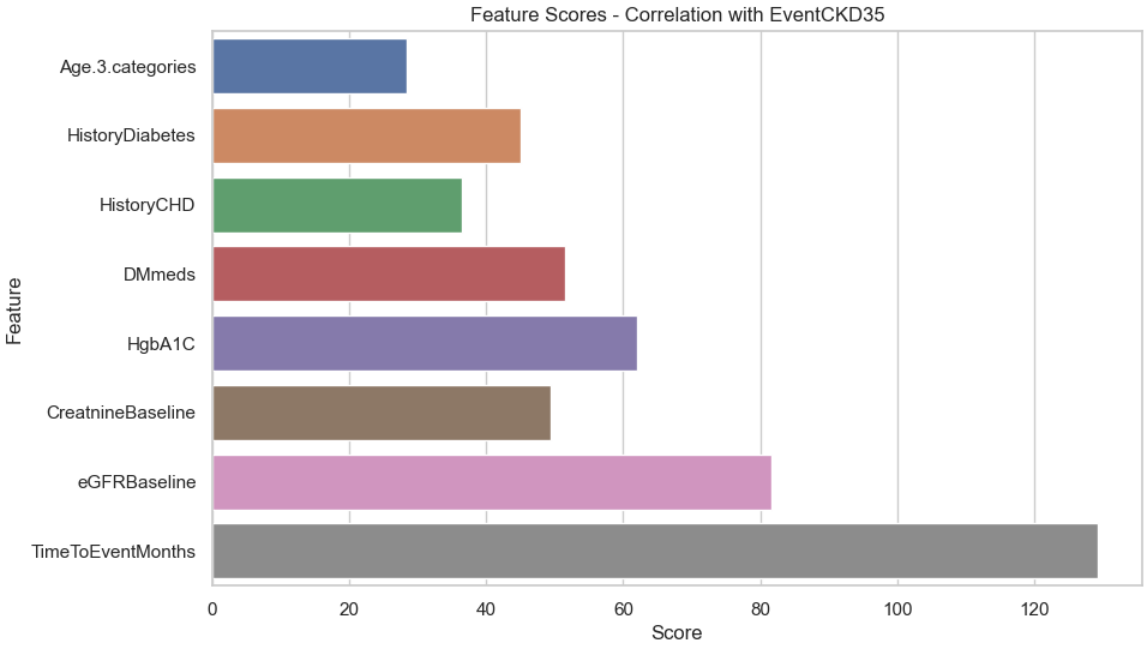


Figure 7

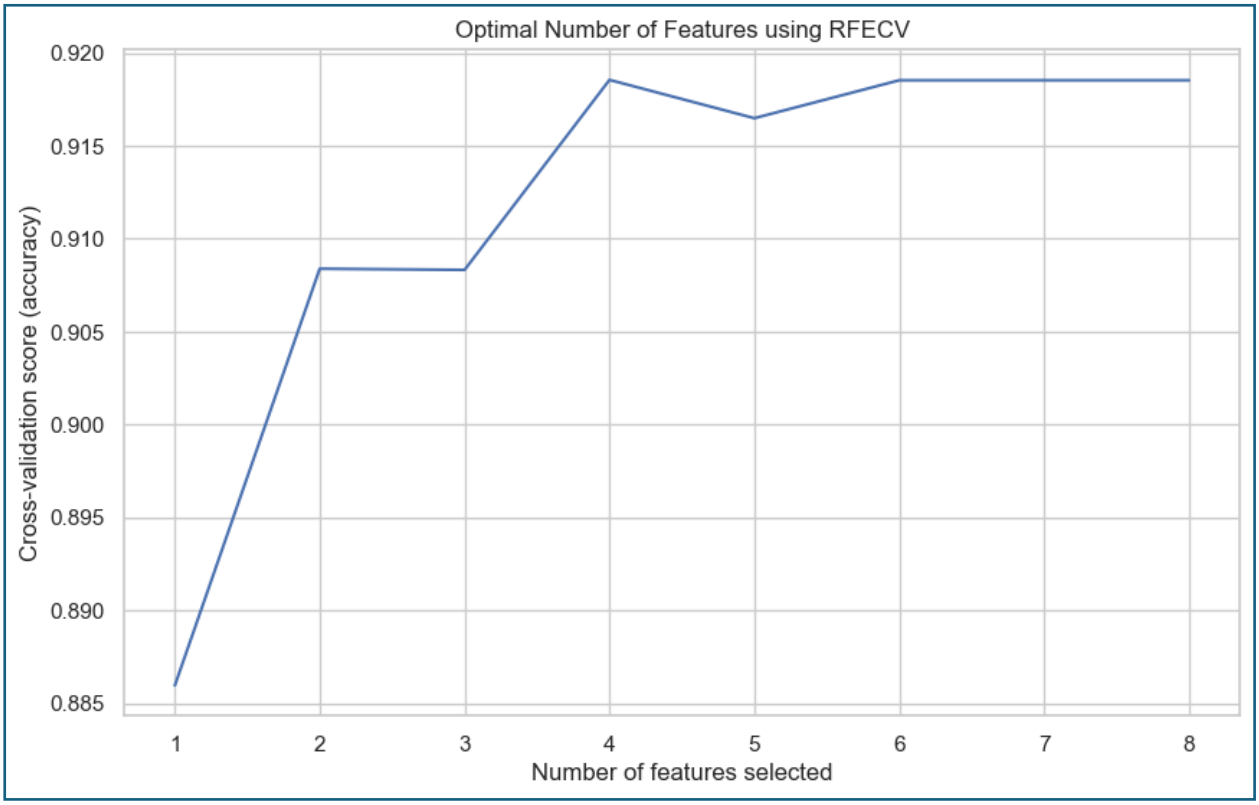


Figure 8

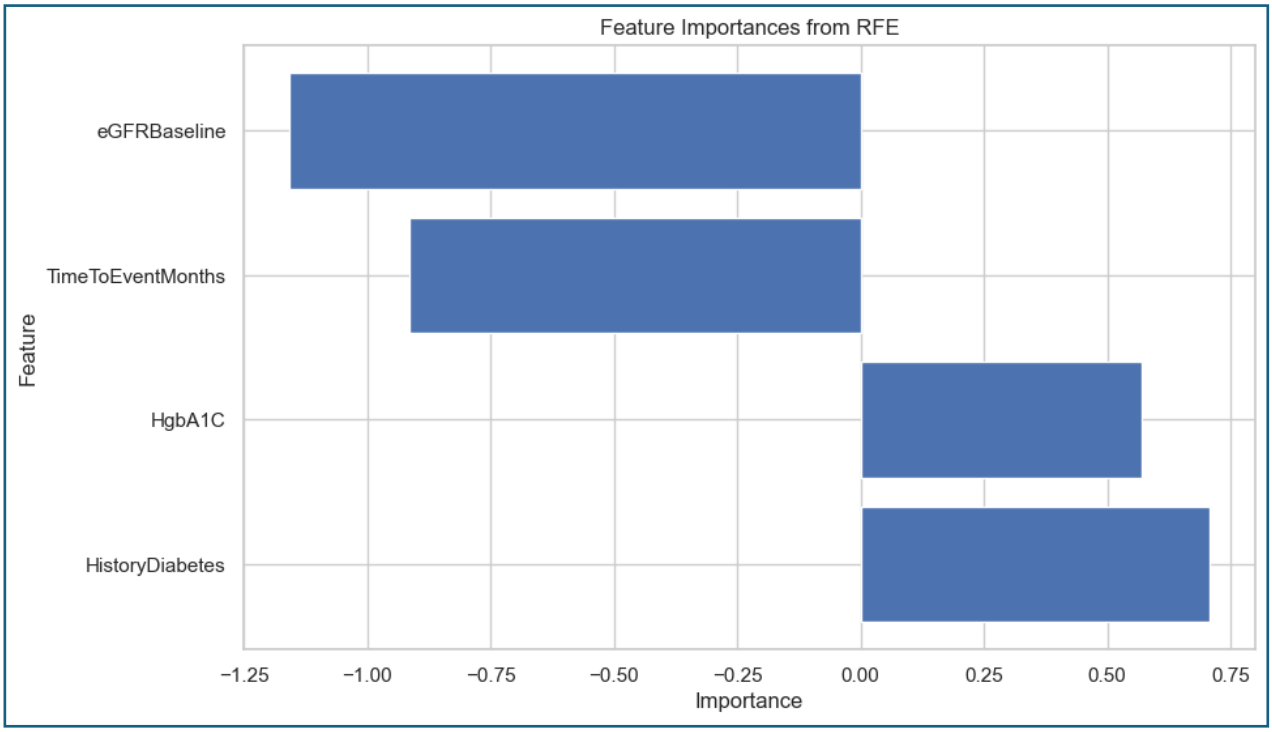


Figure 9

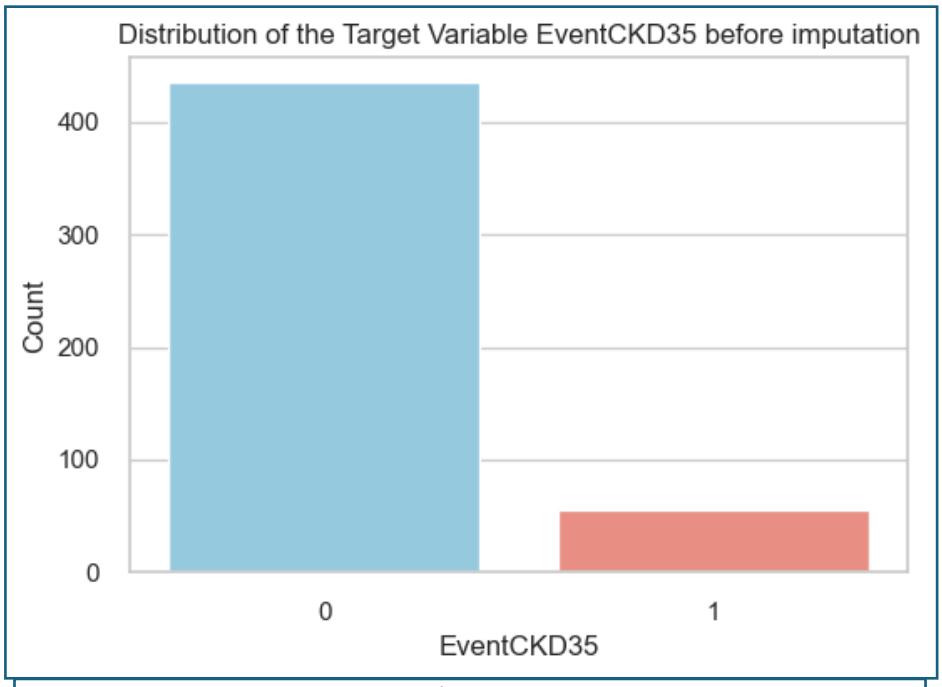
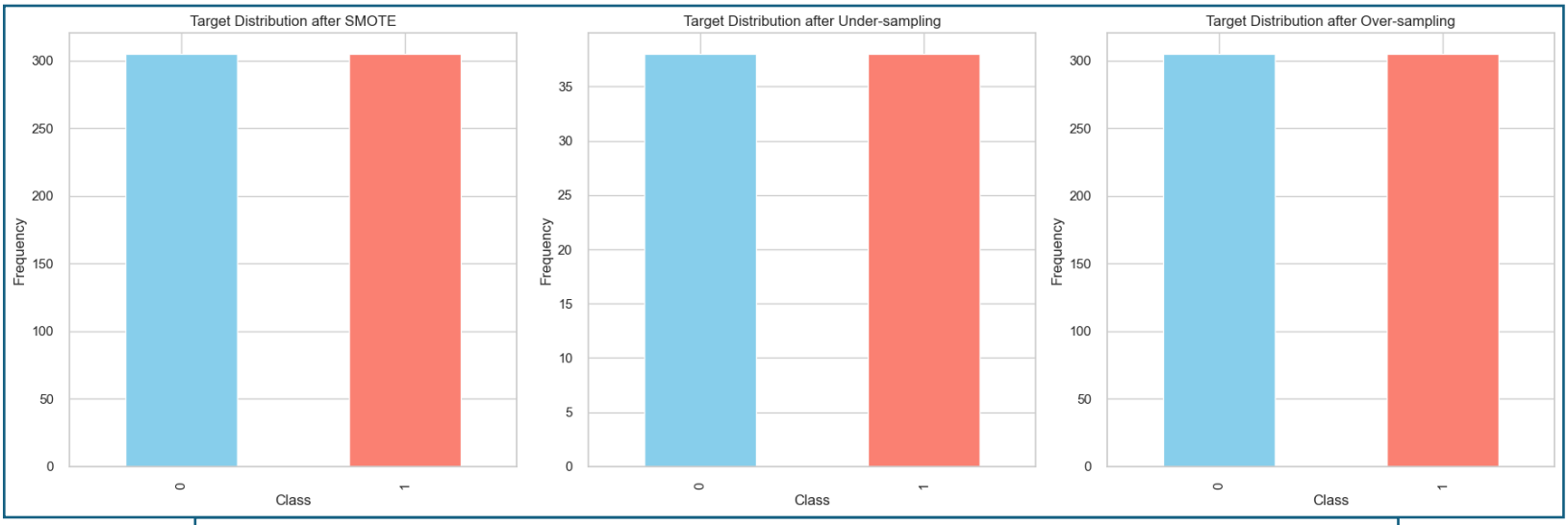


Figure 10



`SMOTE: {0: 305, 1: 305} | Under-sampling: {0: 38, 1: 38} | Over-sampling: {0: 305, 1: 305}`

Figure 11

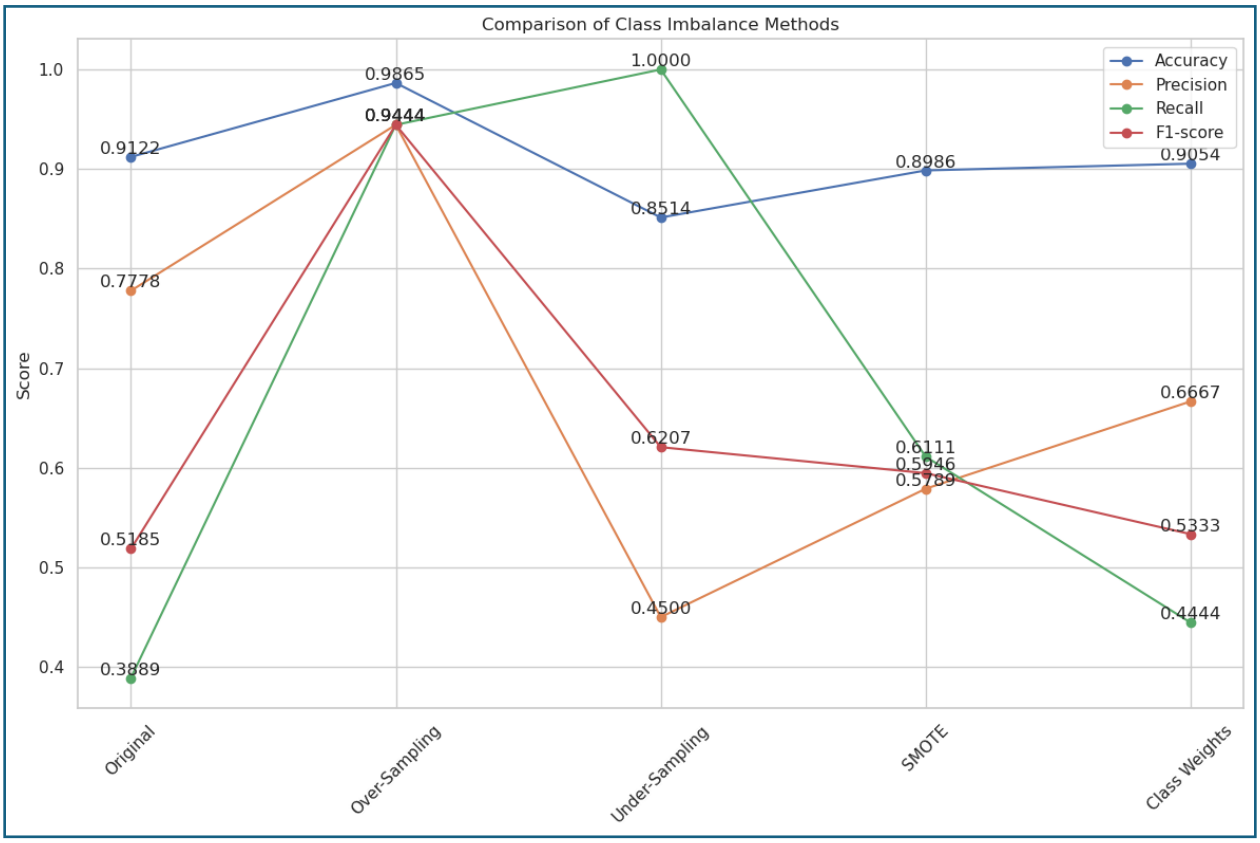


Figure 12

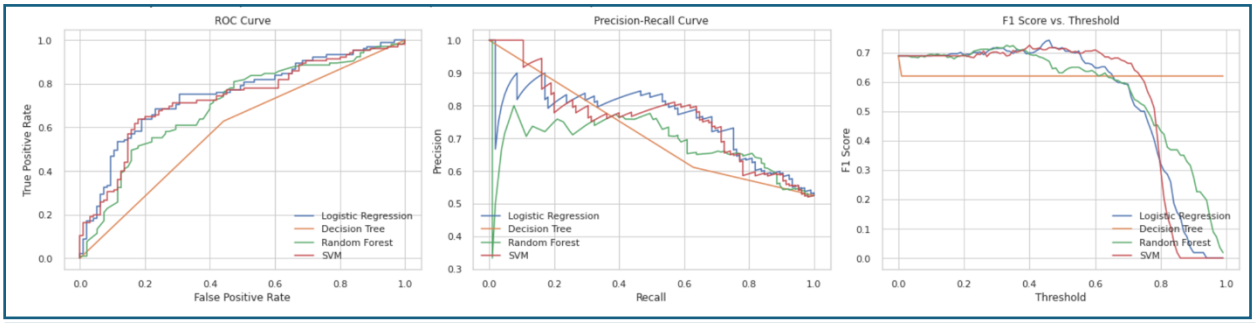


Figure 13

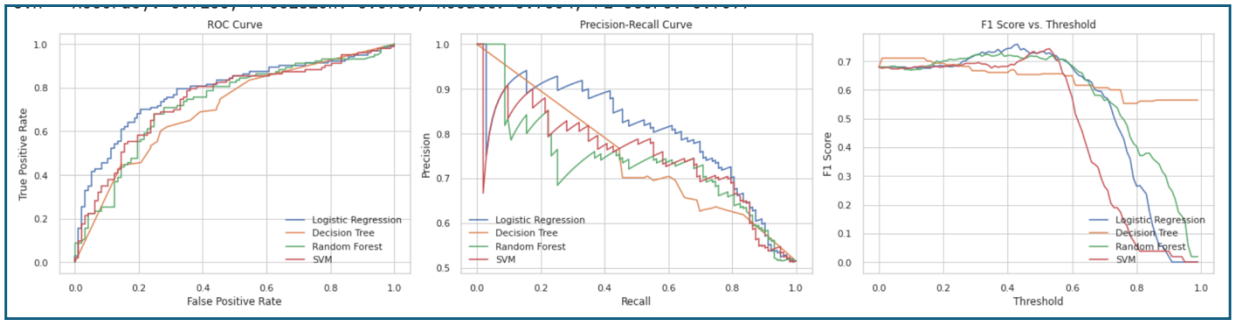


Figure 14

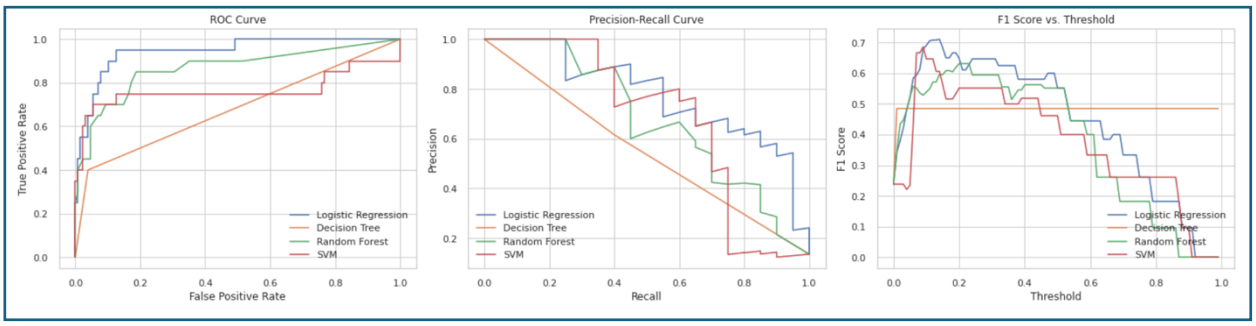


Figure 15

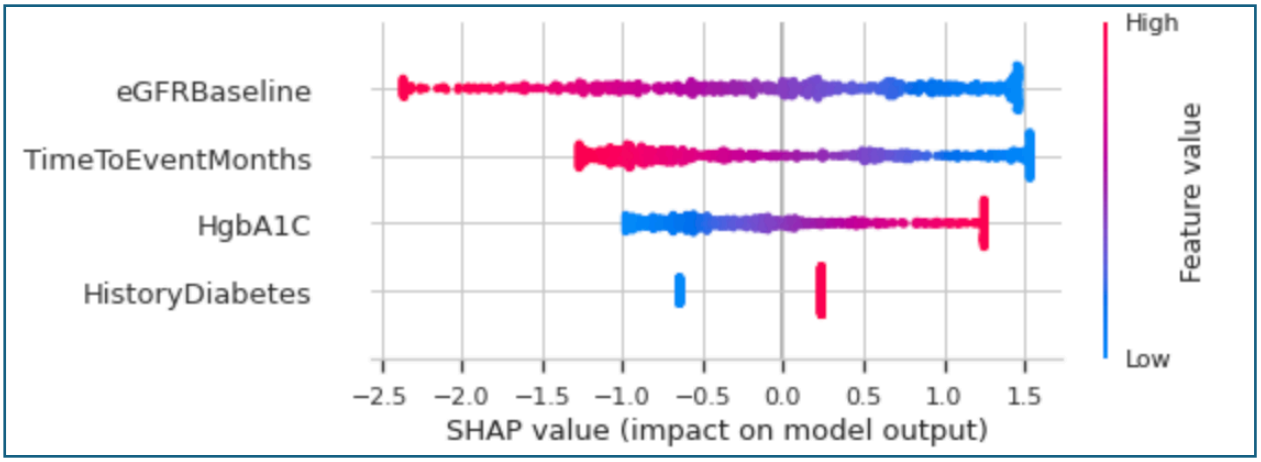


Figure 16- SHAP values for logistic regression.

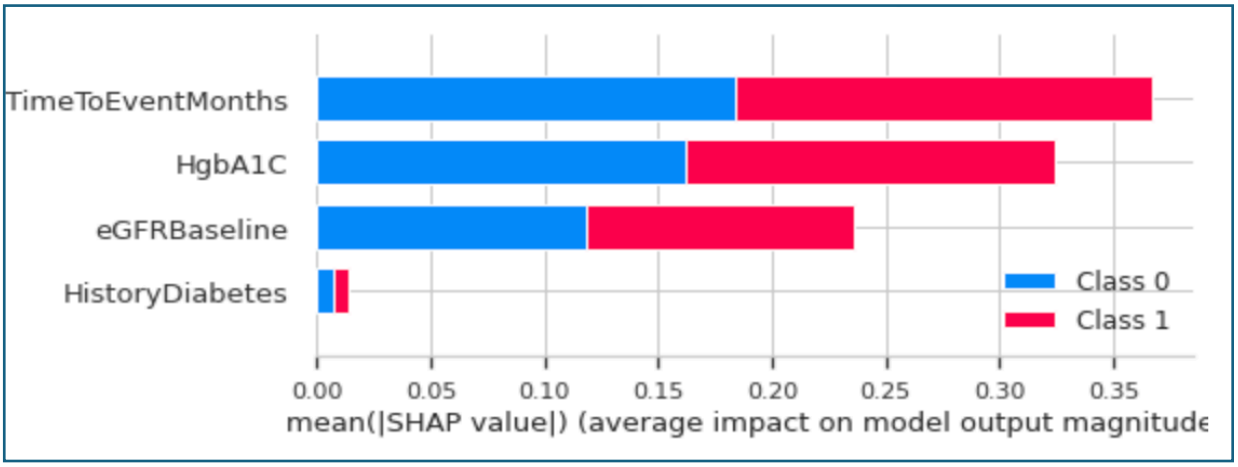


Figure 17- SHAP values for Decision Tree.

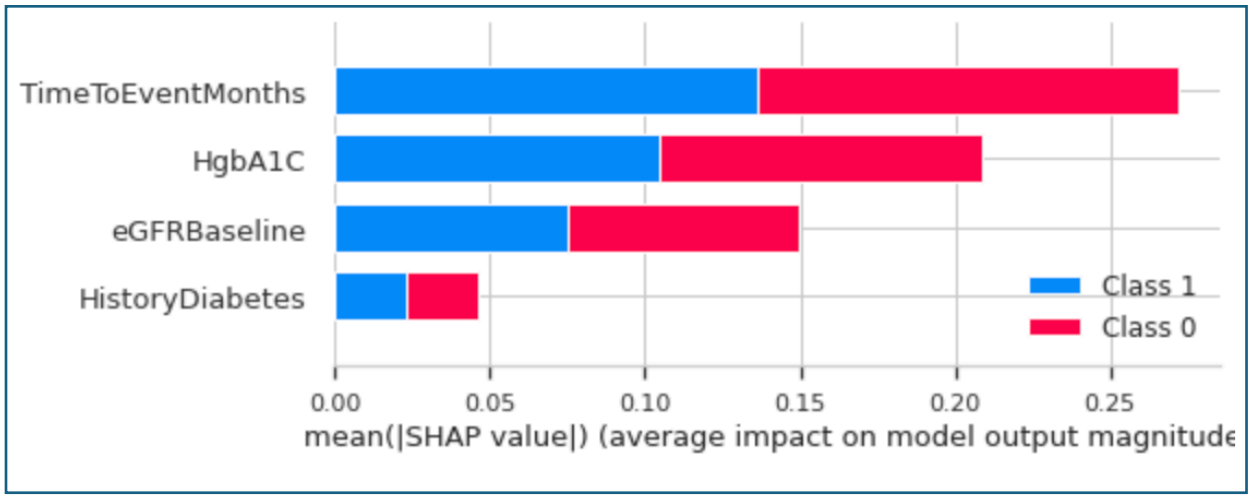


Figure 18- SHAP values for Random Forest.

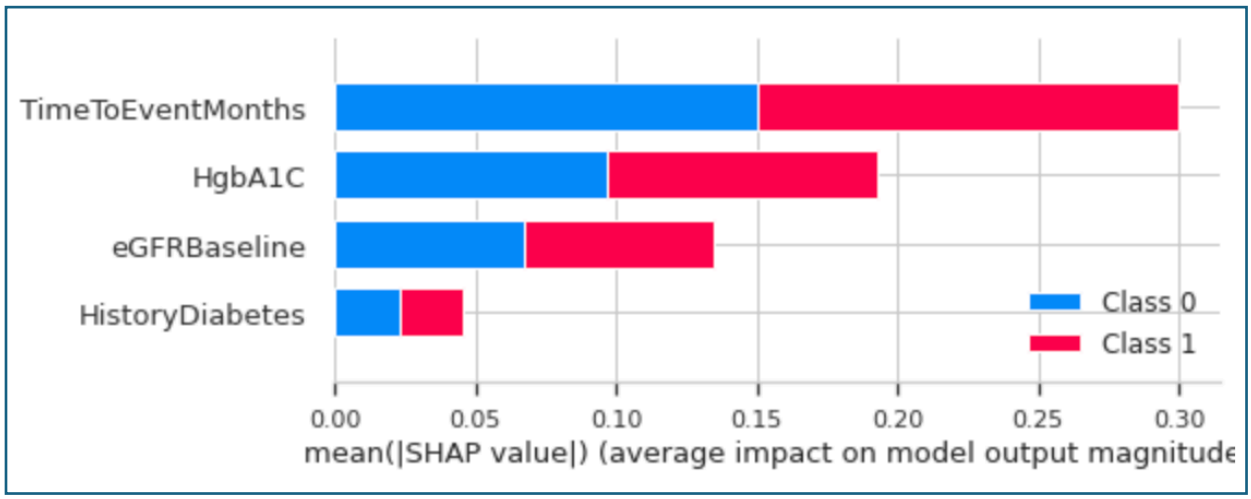


Figure 19- SHAP values for SVM.

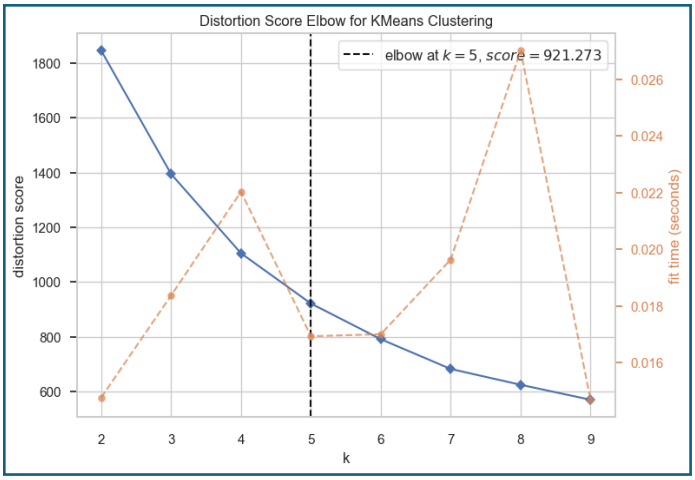


Figure 20

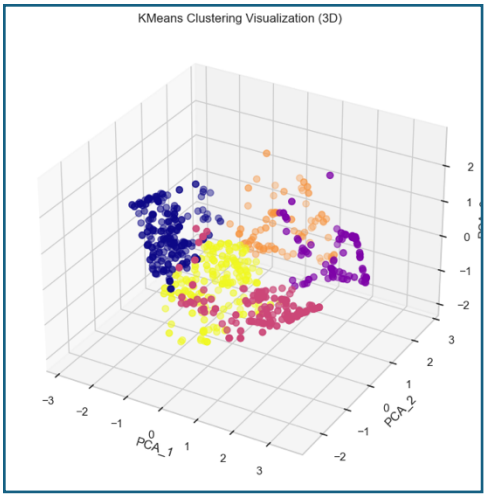


Figure 21

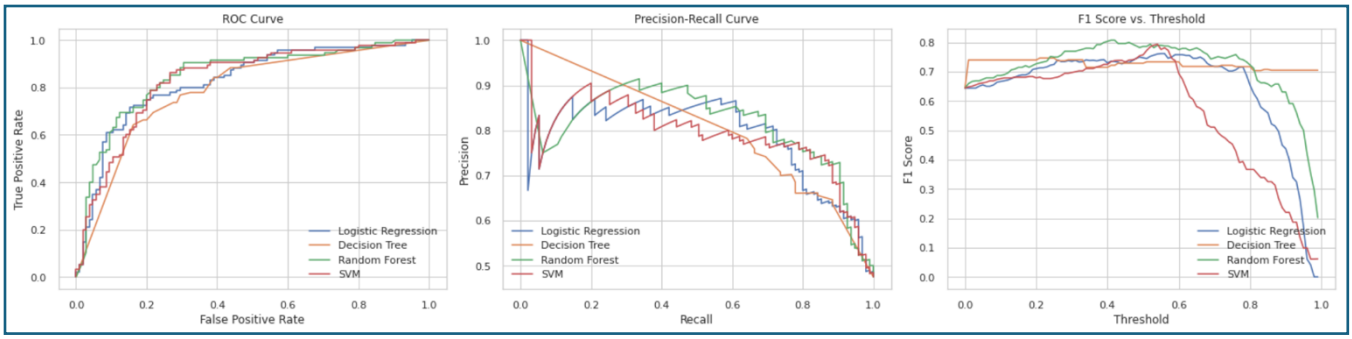


Figure 22