Predicting Healthcare Utilization and Expenditure Using Machine Learning

ML course -2024

Hafiza Hajrah Rehman (7063002)

Atqa Rabiya Amir (7050250)

Contents

List of Figures

1. INTRODUCTION

In the dynamic world of health care, reliable forecasting of consumption and medical costs has emerged as a pivotal step. Health care decision makers such as institutions, insurance companies, and government agencies use such modeling techniques for budgetary planning, forecasting future demand, and to ensure that everyone is provided for in regard to health care service provision. The capability of knowing whether relativity of an individual is likely to have high or low health care use and or total annual medical cost provides powerful information to decision makers in both the public and private health sector.

This project focuses on developing machine learning models to predict two key outcomes: health care contacts which is defined as frequently or infrequently and total medical expenditure in USD. Some of the known difficulties in the models are related to the high dimensionality of the inputs, which has 108 features grouped in demographic and socio-economic indicators and health related co-variates. In addition, it is intended to develop/mine a number of predictive tasks using a set of feature selection techniques and compare various classification and/or regression models from the machine learning, starting with simple linear models and going up to the ensemble models.

Our objectives are twofold: first, to ensure that the classification model of healthcare utilization is well developed then the next step is to ensure that a sound regression model for total medical expenses is established. Thus, exploiting the best data preprocessing techniques, feature extraction, and hyperparameters tuning, it is always a possibility to improve predictive accuracy of these models. This document outlines our approach to data analysis, model selection, and assessment of performance, the problems that were experienced, and the measures that were taken to solve them. At the end of the project, we expect to come up with easy to implement models which meet the performances as promised on the given datasets and also that we can boast of applying the models in actual healthcare settings.

2 Data Analysis & Preprocessing

2.1 Data Analysis

The dataset provided for this project is highly comprehensive, comprising 108 features that span across various categories, including demographic information, personal characteristics, and health-related variables. Given the complexity and size of the dataset, a thorough exploratory data analysis (EDA) was essential as a preliminary step before implementing any machine learning models. The initial focus was on identifying and addressing potential data quality issues that could adversely affect model performance.

In the data exploration phase, we focused on gaining a comprehensive understanding of the dataset by visualizing the distribution and relationships among the features. To begin with, we plotted histograms for all features in the dataset, allowing us to observe the spread and

distribution of each variable. This step was crucial in identifying skewness, outliers, and any potential data quality issues.

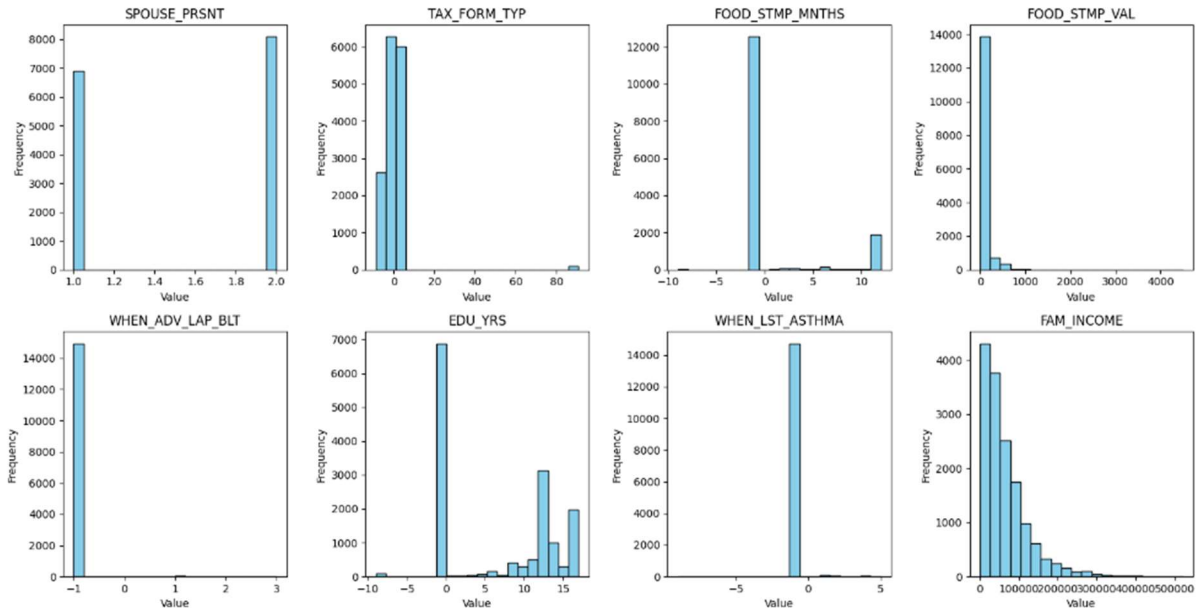

Figure 1: Histograms of some fetaures

Additionally, we created correlation heatmaps to examine the relationships between features, particularly with our target variables, `TOT_MED_EXP` for regression and `UTILIZATION` for classification. The heatmaps helped us to identify strongly correlated features, which could be valuable predictors in our models. By focusing on the top correlated features, we ensured that our feature selection was informed and data-driven, which contributed to the overall performance of our models.

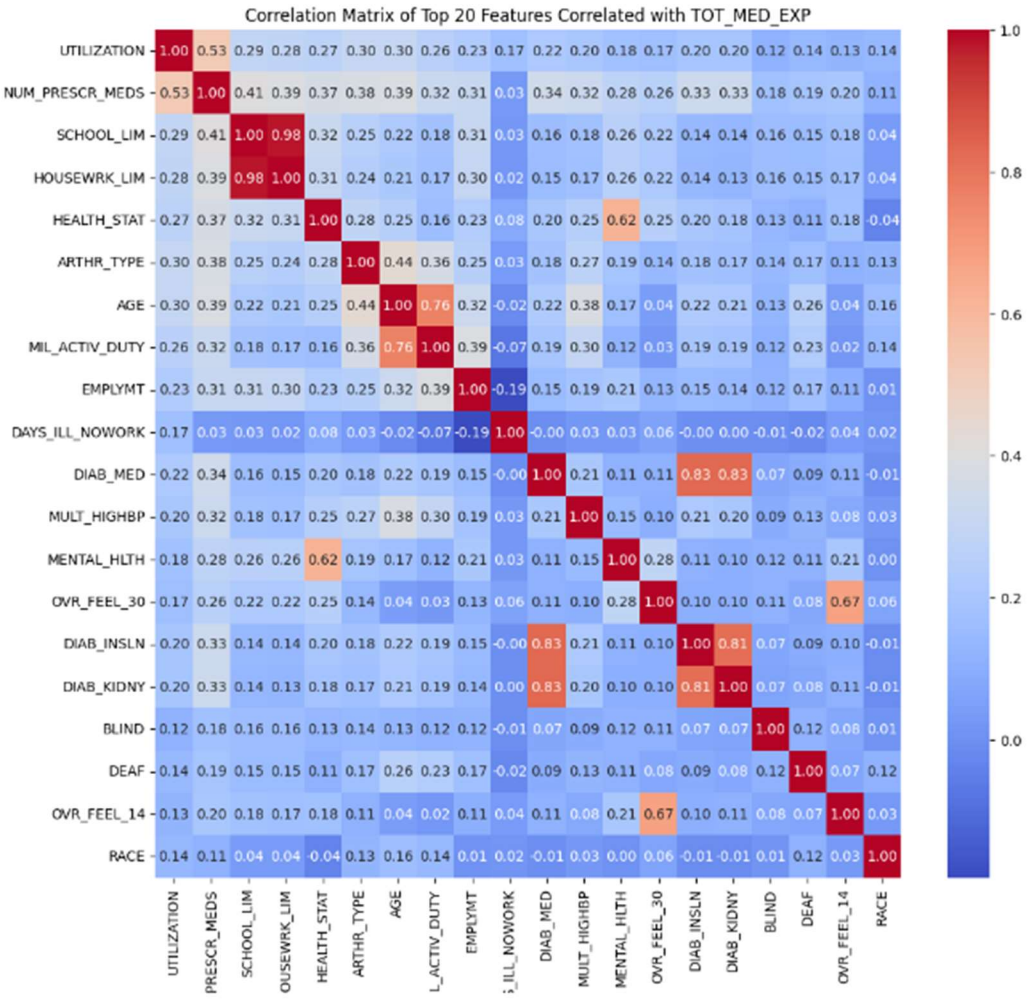

Figure 2: Correlation Matrix of Top 20 Features Correlated with TOT_MED_EXP

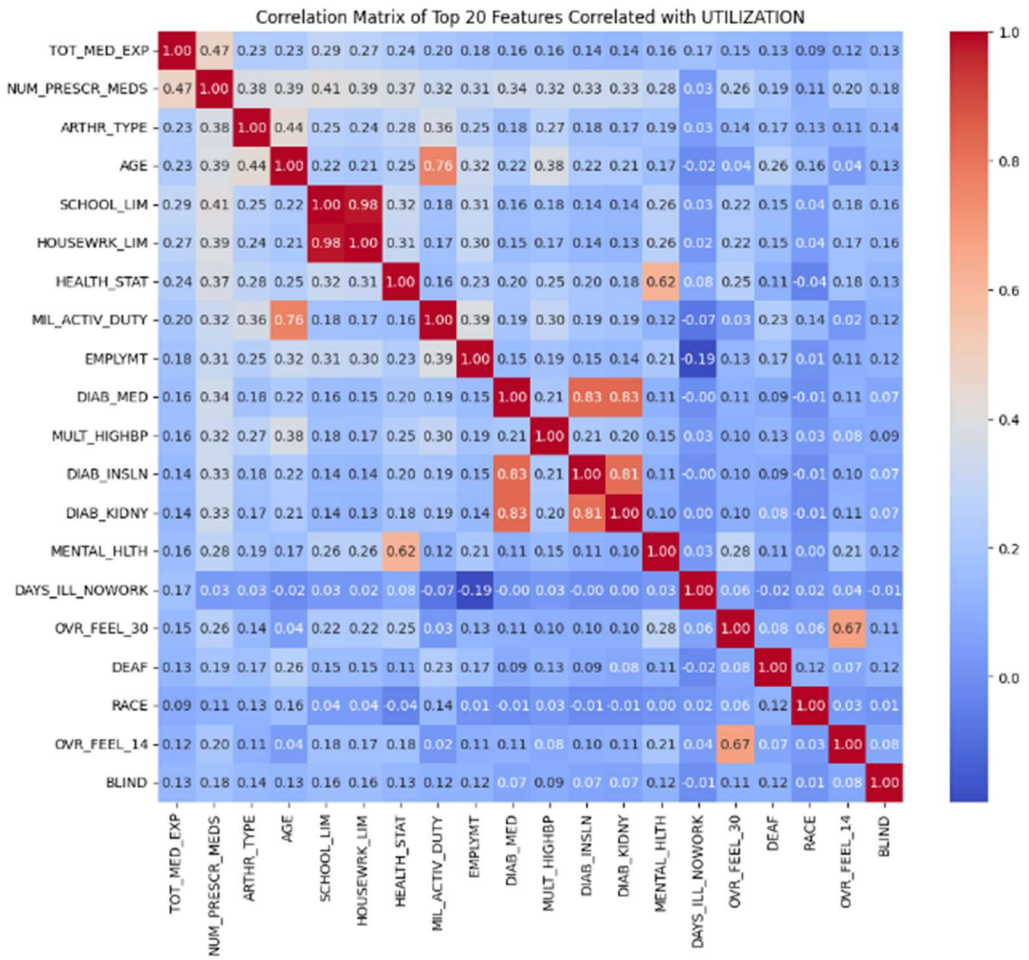

Figure 3: Correlation Matrix of Top 20 Features Correlated with UTILIZATION

One of the first tasks was to assess the dataset for missing values. Missing data can introduce biases and reduce the effectiveness of the models if not handled appropriately. Fortunately, our analysis revealed that the dataset was complete, with no missing values across any of the features. This finding allowed us to proceed without the need for imputation strategies, thereby simplifying the data preprocessing pipeline.

Next, we addressed the issue of outliers, which can skew the results of statistical analyses and lead to less accurate models. Outliers were detected using the Z-score method, a robust technique that identifies data points significantly deviating from the mean. Data points with Z-scores exceeding a threshold of 5 were considered outliers and were subsequently removed from the dataset. This step was crucial for improving the reliability of our machine learning models, as it helped to mitigate the influence of extreme values that could distort the learning process.

Another significant observation during the data analysis was the class imbalance in the target variable for the classification task, UTILIZATION. The dataset exhibited a higher proportion of instances labeled as low utilization compared to high utilization. Class imbalance poses a

challenge because models trained on imbalanced data tend to be biased toward the majority class, leading to poor generalization on the minority class.

After conducting a thorough data analysis, we proceeded with several key preprocessing steps to prepare the dataset for effective modeling. Given the diverse nature of the features, it was essential to transform and standardize the data appropriately to ensure the models could learn efficiently and make accurate predictions.

2.2 Preprocessing

The first step in the preprocessing pipeline was to handle the categorical variables present in the dataset. Specifically, the RACE feature, which had categorical values, was converted into numerical values using Label Encoding. Label Encoding is a straightforward method where each category is assigned a unique integer value. This transformation was necessary because most machine learning algorithms require numerical inputs and cannot directly process categorical data.

Next, we addressed the issue of feature scaling. The dataset contained numerical features with varying scales, which could lead to some features dominating others when training the models. To mitigate this, we applied the StandardScaler to standardize the numerical features. Standardization involves rescaling the features so that they have a mean of zero and a standard deviation of one. This step ensures that each feature contributes equally to the model, preventing any bias that could arise from differences in the scale of the data.

Another critical aspect of our preprocessing strategy was handling the class imbalance in the target variable UTILIZATION. As identified during the data analysis phase, the dataset was imbalanced, with a higher number of instances classified as low utilization compared to high utilization. To address this, we employed SMOTE (Synthetic Minority Over-sampling Technique), a powerful technique that generates synthetic samples of the minority class. By balancing the classes in the dataset, SMOTE helps prevent the model from being biased toward the majority class, thereby improving its ability to correctly predict both high and low utilization.

Finally, to enhance model performance and reduce computational complexity, we implemented Recursive Feature Elimination (RFE) for feature selection. RFE is a feature selection method that recursively removes the least significant features, based on their contribution to the model's performance, until the optimal number of features is reached. By selecting the most significant features for both classification and regression tasks, we effectively reduced the dimensionality of the dataset, which not only improved model accuracy but also sped up the training process.

Through these preprocessing steps, we transformed the raw data into a well-structured and balanced dataset, optimized for training robust and effective machine learning models. This careful preparation was crucial in laying the groundwork for the subsequent modeling phase, ensuring that the models could achieve high levels of accuracy and generalization.

3. Machine Learning Models

3.1 Regression Task

For the regression task aimed at predicting TOT_MED_EXP, multiple models were employed to explore their performance on the dataset. The models tested included Linear Regression, Random Forest, Gradient Boosting, and XGBoost. These models were chosen due to their diverse approaches in handling data and their ability to capture different levels of feature interactions.

Linear Regression: As a starting point, a Linear Regression model was trained on the data. Despite its simplicity, Linear Regression often provides a useful baseline. In this case, the model achieved a Root Mean Square Error (RMSE) of 6779.33 on the validation set. This result highlighted the limitations of the linear model, particularly in capturing the complexity of the healthcare data, as indicated by the absence of a meaningful R^2 value.

```
Split: training data

        RMSE: 6450.19
        MAE: 3347.50
        sqrt(median((y-pred)**2)): 1726.3647409892278
        median(abs(y-pred)): 1726.3647406303353


Split: validation data

        RMSE: 6779.33
        MAE: 3564.00
        sqrt(median((y-pred)**2)): 1792.7747131705582
        median(abs(y-pred)): 1792.7747131705582


Selected features: ['RACE' 'PANEL' 'MIL_ACTIV_DUTY' 'HEALTH_STAT' 'CHRON_BRONCH' 'PREGNT'
 'WALK_LIM' 'ACTIV_LIM' 'COGNTV_LIM' 'REGION' 'INSUR_COV' 'TOT_INCOME'
 'BM_IDX' 'MULT_HIGHBP' 'HOUSEWRK_LIM' 'SCHOOL_LIM' 'ADV_EXERCISE_MORE'
 'FREQ_DNTL_CKP' 'DOC_CK_BP' 'FOOD_STMP_MNTHS' 'FOOD_STMP_VAL' 'EDU_YRS'
 'WHEN_LST_ASTHMA' 'FAM_INCOME' 'POVRTY_LEV' 'APPT_REG_MEDCARE'
 'LOST_ALL_TEETH' 'OCCUP' 'DIFF_ERRND_ALN' 'DIAB_KIDNY' 'DIAB_INSLN'
 'DIAB_MED' 'DEAF' 'UNABL_PRES_MED' 'HEAR_AID' 'NO_WORK_WHY'
 'DAYS_ILL_NOWORK' 'HIGH_BP_DIAG' 'COR_HRT_DIAG' 'HRT_ATT_DIAG'
 'OTH_HRT_DIAG' 'CANCER_DIAG' 'DIAB_DIAG' 'ARTHR_DIAG' 'ARTHR_TYPE'
 'NUM_PRESCR_MEDS' 'DIFFIC_HEAR' 'SMOK' 'MENTAL_HLTH_SCR' 'PHY_HLTH_SCR']
```

Figure 4: Linear Regression Analysis

Random Forest: To enhance predictive performance, a Random Forest Regressor was employed. This ensemble method is known for its robustness and ability to model non-linear relationships. After hyperparameter tuning, the Random Forest model achieved an RMSE of 6785.65 on the validation set with an R^2 of 0.32. While this was a modest improvement in terms of RMSE, the R^2 value suggested that the model captured some variability in the data, though it still left much to be desired in terms of accuracy.

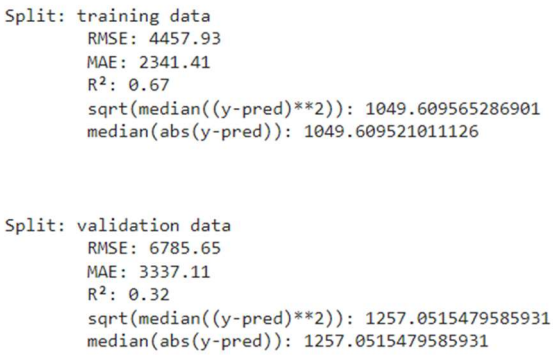

Figure 5: Random Classifier Anlaysis

Gradient Boosting: Further efforts were made to improve the model's performance by applying Gradient Boosting, a more sophisticated ensemble method that iteratively builds models to correct errors made by previous models. The Gradient Boosting model performed slightly better, achieving an RMSE of 6766.95 on the validation set and maintaining an R^2 of 0.32. This indicated a slight improvement over Random Forest, but the gains were marginal.

XGBoost: XGBoost, another powerful gradient boosting technique, was then tested. Known for its efficiency and scalability, XGBoost achieved the best performance among the models tested, with an RMSE of 6759.20 on the validation set and an R^2 of 0.32. Although the improvement over Gradient Boosting was minimal, XGBoost's ability to slightly reduce the error made it the most effective model for this regression task.

3.2 Classification Task

For the classification task aimed at predicting UTILIZATION, three different models were explored: Logistic Regression, Support Vector Machine (SVM), and Ridge Classifier.

Logistic Regression: As a linear model, Logistic Regression was chosen for its simplicity and interpretability. After tuning, the model achieved an accuracy of 0.84 on the validation set. This performance was robust, given the model's simplicity, and provided a solid baseline for comparison with more complex models.

```
Training Data Evaluation:
              precision    recall  f1-score   support

           0       0.89      0.85      0.87      9435
           1       0.86      0.90      0.88      9435

    accuracy                           0.87     18870
   macro avg       0.88      0.87      0.87     18870
weighted avg       0.88      0.87      0.87     18870


Validation Data Evaluation:
              precision    recall  f1-score   support

           0       0.63      0.64      0.63       647
           1       0.90      0.90      0.90      2353

    accuracy                           0.84      3000
   macro avg       0.76      0.77      0.76      3000
weighted avg       0.84      0.84      0.84      3000


Confusion Matrix for Validation Data:
[[ 411  236]
 [ 243 2110]]
```

Figure 6: Logistic Regression Analysis

Support Vector Machine (SVM): To capture non-linear relationships, an SVM with a linear kernel was applied. The SVM outperformed Logistic Regression, achieving an accuracy of 0.85 on the validation set. The slight improvement suggested that the SVM was better suited to the data's characteristics, particularly in handling the class boundaries.

```
Training Data Evaluation:
              precision    recall  f1-score   support

           0       0.90      0.84      0.87      9435
           1       0.85      0.91      0.88      9435

    accuracy                           0.88     18870
   macro avg       0.88      0.88      0.88     18870
weighted avg       0.88      0.88      0.88     18870


Validation Data Evaluation:
              precision    recall  f1-score   support

           0       0.65      0.63      0.64       647
           1       0.90      0.91      0.90      2353

    accuracy                           0.85      3000
   macro avg       0.78      0.77      0.77      3000
weighted avg       0.85      0.85      0.85      3000


Confusion Matrix for Validation Data:
[[ 410  237]
 [ 217 2136]]
```

Figure 7: SVM Analysis

Ridge Classifier: Lastly, a Ridge Classifier was tested to explore the effects of regularization in a linear classification context. The Ridge Classifier achieved an accuracy of 0.83 on the validation set. While this was slightly lower than both Logistic Regression and SVM, the Ridge Classifier's performance demonstrated the utility of regularization in managing overfitting, even though it did not surpass the other models in this scenario.

```
Training Data Evaluation:
              precision    recall  f1-score   support

           0       0.89      0.83      0.86      9435
           1       0.84      0.90      0.87      9435

    accuracy                           0.87     18870
   macro avg       0.87      0.87      0.87     18870
weighted avg       0.87      0.87      0.87     18870


Validation Data Evaluation:
              precision    recall  f1-score   support

           0       0.62      0.60      0.61       647
           1       0.89      0.90      0.89      2353

    accuracy                           0.83      3000
   macro avg       0.75      0.75      0.75      3000
weighted avg       0.83      0.83      0.83      3000


Confusion Matrix for Validation Data:
[[ 391  256]
 [ 244 2109]]
```

Figure 8: Ridge Classifier

Stacking Classifier: Finally, a Stacking Classifier approach was employed using RandomizedSearchCV to combine the strengths of Logistic Regression, Ridge Classifier, and SVM. This ensemble method aimed to leverage the strengths of each model, but the validation set performance indicated only modest improvements, highlighting the challenge of finding an optimal combination of these models in this specific context. Despite the extensive tuning efforts, no single model emerged as a clear leader across all metrics, suggesting that further exploration or a different approach might be needed to achieve significant improvements.

```
Training Data Evaluation:
              precision    recall  f1-score   support

           0       0.88      0.87      0.87      9435
           1       0.87      0.88      0.88      9435

    accuracy                           0.87     18870
   macro avg       0.87      0.87      0.87     18870
weighted avg       0.87      0.87      0.87     18870

Validation Data Evaluation:
              precision    recall  f1-score   support

           0       0.59      0.67      0.63       647
           1       0.91      0.87      0.89      2353

    accuracy                           0.83      3000
   macro avg       0.75      0.77      0.76      3000
weighted avg       0.84      0.83      0.83      3000

Confusion Matrix for Validation Data:
[[ 432  215]
 [ 295 2058]]
```

Figure 9: Stacking Classifier

4. Model Selection and Hyperparameter Tuning

4.1 Model Selection Criteria

The selection of the final models for both regression and classification tasks was based on a comprehensive evaluation of several performance metrics. For regression, the primary metrics considered were Root Mean Square Error (RMSE), Mean Absolute Error (MAE), and the coefficient of determination (R^2). RMSE was particularly important as it penalizes larger errors more severely, making it a crucial metric for understanding the accuracy of the model's predictions. MAE provided an additional perspective by offering a straightforward interpretation of average errors, while R^2 indicated the proportion of variance in the target variable that was captured by the model. Ensemble methods like Random Forest and Gradient Boosting emerged as top contenders due to their ability to capture complex, non-linear relationships in the data, consistently delivering lower RMSE and higher R^2 compared to simpler models like Linear Regression.

For classification, the models were evaluated using accuracy, precision, recall, and F1-score. Accuracy was the primary metric, providing a general measure of how well the model performed across all classes. However, in scenarios with class imbalance, as seen in the UTILIZATION target variable, precision, recall, and F1-score became critical in ensuring that the model performed well not just overall but also on each individual class. Precision measured the accuracy of positive predictions, recall assessed the model's ability to capture all true positives, and F1-score offered a balance between precision and recall. SVM and Logistic Regression were

identified as the top-performing models, particularly because of their higher accuracy and balanced performance across precision, recall, and F1-score.

4.2 Hyperparameter Tuning

Hyperparameter tuning played a pivotal role in enhancing the performance of the selected models. For each model, we employed either GridSearchCV or RandomizedSearchCV to systematically explore a range of hyperparameter settings. GridSearchCV was used when the hyperparameter space was relatively small and manageable, allowing for an exhaustive search over all possible combinations. On the other hand, RandomizedSearchCV was utilized for models with a larger hyperparameter space, where a more efficient approach was necessary to identify good parameter combinations without the computational cost of an exhaustive search.

For the regression models, parameters such as the number of estimators, maximum depth, and learning rate were tuned for ensemble methods like Random Forest and Gradient Boosting. This process was crucial in reducing the RMSE and improving the R^2 values, allowing the models to generalize better on the validation set.

In the classification tasks, key hyperparameters like the regularization strength (C), penalty types, and solver choices for Logistic Regression, as well as kernel type and regularization for SVM, were carefully tuned. The tuning process was essential in balancing precision and recall, particularly in the presence of class imbalance. For the Ridge Classifier, the regularization strength (alpha) was fine-tuned to control the trade-off between bias and variance, which helped in maintaining model stability and performance. Overall, hyperparameter tuning led to substantial improvements in accuracy and other related metrics, ensuring that the models were not only well-calibrated but also optimized for the specific characteristics of the dataset.

5. Empirical Results

The empirical results from our machine learning models provided critical insights into the performance and suitability of each approach for the regression and classification tasks. For the regression task, which focused on predicting total medical expenses (`TOT_MED_EXP`), the models were evaluated based on RMSE, MAE, and R^2. The baseline Linear Regression model yielded an RMSE of 6779.33 on the validation set, indicating considerable prediction error. Ensemble methods like Random Forest and Gradient Boosting offered slight improvements with RMSE values of 6785.65 and 6766.95, respectively, both achieving an R^2 of 0.32. These results highlighted the ensemble models' ability to capture more complex relationships within the data, although the gains were modest. XGBoost, despite its reputation for high performance in regression tasks, delivered an RMSE of 6759.20, marginally outperforming the other models. However, the improvements across all models were incremental, suggesting that the dataset's inherent complexity and feature interactions might require even more sophisticated techniques or feature engineering to achieve significant gains.

In the classification task, aimed at predicting healthcare utilization (`UTILIZATION`), the models demonstrated more pronounced differences in performance. Logistic Regression and SVM emerged as the top performers, with accuracy scores of 0.84 and 0.85 on the validation set, respectively. The SVM model slightly edged out Logistic Regression, especially in handling the imbalanced nature of the target variable, as reflected in its higher precision and recall metrics for the minority class. Ridge Classifier, while still effective, lagged slightly behind with an accuracy of 0.83, reflecting its more conservative approach to regularization. Additionally, a stacking model that combined Logistic Regression, SVM, and Ridge Classifier through RandomizedSearchCV provided competitive results with a validation accuracy of 0.83, but without significant gains over the individual models.

Across all models, the empirical results underscored the importance of model selection and tuning. While ensemble methods and more complex models like XGBoost and SVM provided some improvements, the gains were often incremental, suggesting that further exploration, possibly through more advanced feature engineering or alternative modeling approaches, might be required to achieve substantial improvements in predictive performance. The detailed evaluation of metrics such as RMSE for regression and F1-score for classification provided a comprehensive understanding of each model's strengths and limitations, guiding the final model selection for the challenge.

Conclusion

In this project, we tackled the challenge of predicting total medical expenses and healthcare utilization using a dataset rich with demographic, personal, and health-related features. Through a comprehensive approach that included data preprocessing, feature selection, and model evaluation, we explored various machine learning models to achieve the best possible predictions.

For the regression task, our efforts revealed that while Linear Regression provided a solid baseline, more sophisticated models like Random Forest, Gradient Boosting, and XGBoost offered marginal improvements in RMSE and R^2, with XGBoost slightly outperforming others. However, the modest gains across these models suggest that the problem's complexity may require further refinement in feature engineering or the exploration of even more advanced algorithms to capture the intricate relationships within the data.

In the classification task, predicting healthcare utilization, we found that Logistic Regression and SVM performed best, with SVM achieving the highest accuracy. The use of techniques like SMOTE to address class imbalance and rigorous hyperparameter tuning contributed significantly to enhancing model performance. Despite the promising results, the differences in model performance were not stark, indicating that the dataset's characteristics might limit the achievable accuracy within the current modeling framework.

Overall, this project demonstrated the importance of a structured approach to machine learning, encompassing data analysis, preprocessing, model selection, and hyperparameter tuning. While the results were encouraging, they also highlighted the need for continuous refinement and exploration in machine learning to meet the challenges posed by complex real-world datasets. Future work could focus on incorporating additional features, experimenting with deep learning techniques, or applying ensemble learning methods to further enhance predictive accuracy.

Appendix: Macro, Micro, and Weighted Averaging

In the context of evaluating classification models, particularly for imbalanced datasets, it is crucial to consider different averaging techniques when calculating performance metrics like the F1-score. The three most common methods are macro, micro, and weighted averaging, each offering distinct perspectives on model performance.

Macro Averaging

Macro averaging involves calculating the F1-score independently for each class and then taking the average of these scores. This method treats all classes equally, regardless of their size, making it particularly useful when we want to ensure that the model performs well across all classes, without giving undue weight to the more frequent classes. In our project, this technique helped us assess the model's ability to predict both high and low utilization instances equally.

Micro Averaging

Micro averaging aggregates the contributions of all classes to compute the average metric. Specifically, it sums up the true positives, false negatives, and false positives across all classes and then calculates precision, recall, and F1-score using these aggregate values. This method is beneficial when the goal is to assess the overall performance of the model, giving more weight to the larger classes. For our dataset, micro averaging provided a holistic view of how well the model performed across all instances, particularly emphasizing the more prevalent class.

Weighted Averaging

Weighted averaging calculates the F1-score for each class and then computes an average that accounts for the number of instances in each class. This method is particularly valuable in the presence of class imbalance, as it allows the evaluation metric to reflect the distribution of the classes in the dataset. In this project, weighted averaging was crucial, as it provided a balanced view of model performance that considered the higher occurrence of low utilization instances in our dataset.

By employing these three averaging techniques, we gained a comprehensive understanding of our classification models' performance. While the macro average highlighted the model's effectiveness across all classes, the micro average emphasized its overall performance. The weighted average was particularly insightful for our imbalanced dataset, ensuring that the

model's performance was evaluated in a manner reflective of real-world distributions. This thorough approach allowed us to make informed decisions when selecting and tuning our final models.

```
Detailed F1-Scores:
Micro-average F1-score: 0.8300
Macro-average F1-score: 0.7593
Weighted-average F1-score: 0.8335
```

Figure 10: Detailed F1- score