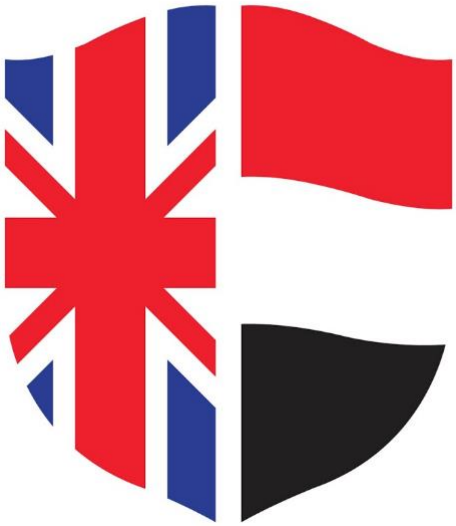

24CSAI03H
Machine Learning

Project 1 Report

Sameh218767

Overview

This project focuses on analyzing and processing a diabetes dataset derived from the Behavioral Risk Factor Surveillance System (BRFSS) survey conducted by the CDC. The Behavioral Risk Factor Surveillance System (BRFSS) is a health-related telephone survey that is collected annually by the CDC. Each year, the survey collects responses from over 400,000 Americans on health-related risk behaviors, chronic health conditions, and the use of preventative services. It has been conducted every year since 1984. For this project, a csv of the dataset available on Kaggle for the year 2015 was used. This dataset contains 253,680 survey responses to the CDC's BRFSS2015. The target variable Diabetes_012 has 3 classes. 0 is for no diabetes or only during pregnancy, 1 is for prediabetes, and 2 is for diabetes. There is class imbalance in this dataset. This dataset has 21 feature variables. The motivation behind this dataset is that early diagnosis by public health officials through predictive and classification models for diabetes can lead to lifestyle changes and more effective treatment.

Source: https://www.kaggle.com/datasets/alexteboul/diabetes-health-indicators-dataset/data

Dataset Attributes:

1. **Diabetes_012**: 0 = no diabetes, 1 = prediabetes, 2 = diabetes. (Target Class)
2. **HighBP**: 0 = no high blood pressure, 1 = high blood pressure.
3. **HighChol**: 0 = no high cholesterol, 1 = high cholesterol.
4. **CholCheck**: 0 = no cholesterol check in past 5 years, 1 = checked.
5. **BMI**: Body Mass Index (numeric value).
6. **Smoker**: 0 = never smoked, 1 = smoked at least 100 cigarettes.
7. **Stroke**: 0 = no history of stroke, 1 = history of stroke.
8. **HeartDiseaseorAttack**: 0 = no heart disease/attack, 1 = history of coronary heart disease or heart attack.
9. **PhysActivity**: 0 = no physical activity, 1 = active in past 30 days (excluding job).
10. **Fruits**: 0 = does not consume fruit daily, 1 = consumes fruit daily.
11. **Veggies**: 0 = does not consume vegetables daily, 1 = consumes vegetables daily.
12. **HvyAlcoholConsump**: 0 = not a heavy drinker, 1 = heavy drinker (men >14, women >7 drinks/week).
13. **AnyHealthcare**: 0 = no healthcare coverage, 1 = has healthcare coverage.

14. **NoDocbcCost**: 0 = no cost barriers to doctor, 1 = avoided doctor due to cost.
15. **GenHlth**: General health rating, 1 (excellent) to 5 (poor).
16. **MentHlth**: Days of poor mental health in past 30 days (1-30).
17. **PhysHlth**: Days of poor physical health in past 30 days (1-30).
18. **DiffWalk**: 0 = no difficulty walking, 1 = serious difficulty.
19. **Sex**: 0 = female, 1 = male.
20. **Age**: Age category, 1 (18-24) to 13 (80+).
21. **Education**: Education level, 1 (no school) to 6 (college graduate).
22. **Income**: Income level, 1 (<$10,000) to 8 ($75,000+).

Data Exploration

The dataset is first explored by checking the basic structure and statistics such as the number of rows and columns, datatypes, number of unique values per class, number of nulls and class distribution of the target. Initial analysis shows that the data includes some missing values, especially in the minority classes (1 and 2 for prediabetes and diabetes), and most importantly there is a major class imbalance in the distribution of the target variable, with class 0 (no diabetes) being the majority and class 1 (prediabetes being the minority as shown below:

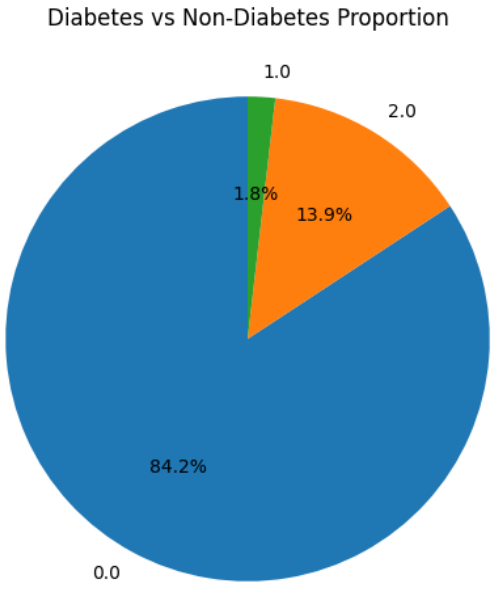

Figure 1: Target class distribution

Data Visualization

Several general visualizations were done in this section to further explore and create initial hypotheses how some features might correlate with others and the target classes. Among the most notable is a heatmap of the missing values in the dataset, a plot distribution of diabetes across different age categories to see any resemblance, the distribution of BMI to identify skewness, a heatmap to show correlation between features, and finally, a pie chart of the class distribution of the target variable with emphasis on the class imbalance shown earlier.

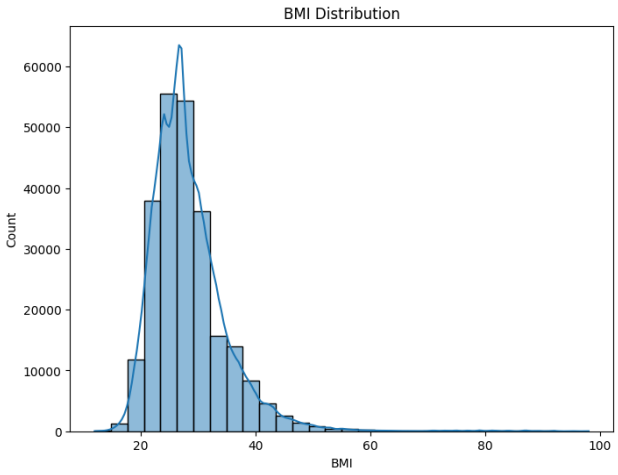

Figure 2: BMI Distribution (Positive Skew)

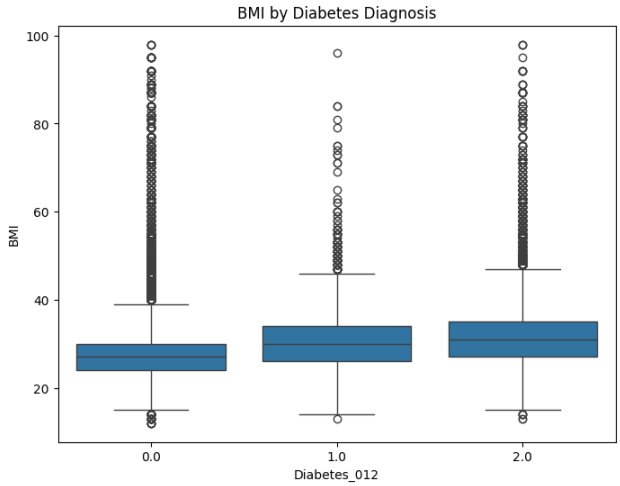

Figure 3: BMI Distribution Across Target Classes

Data Preprocessing

The preprocessing stages were done and modified in accordance with earlier findings as well as future operations that required redoing of certain processes or changing their order. A rigorous methodology was thus followed starting with categorical features being converted into appropriate data types to properly get processed in the imputation and resampling stages later. This is followed closely by dealing with missing data through mean, median and mode imputation according to the datatype, which proved effective and far superior in evaluation afterwards to KNN imputation despite expecting otherwise. After handling the missing values, the dataset is further cleaned by dropping duplicates in order to prevent overfitting. Lastly, BMI, considered the only continuous feature, has its positive skewness fixed and then discretized into meaningful categories based on domain knowledge as shown below:

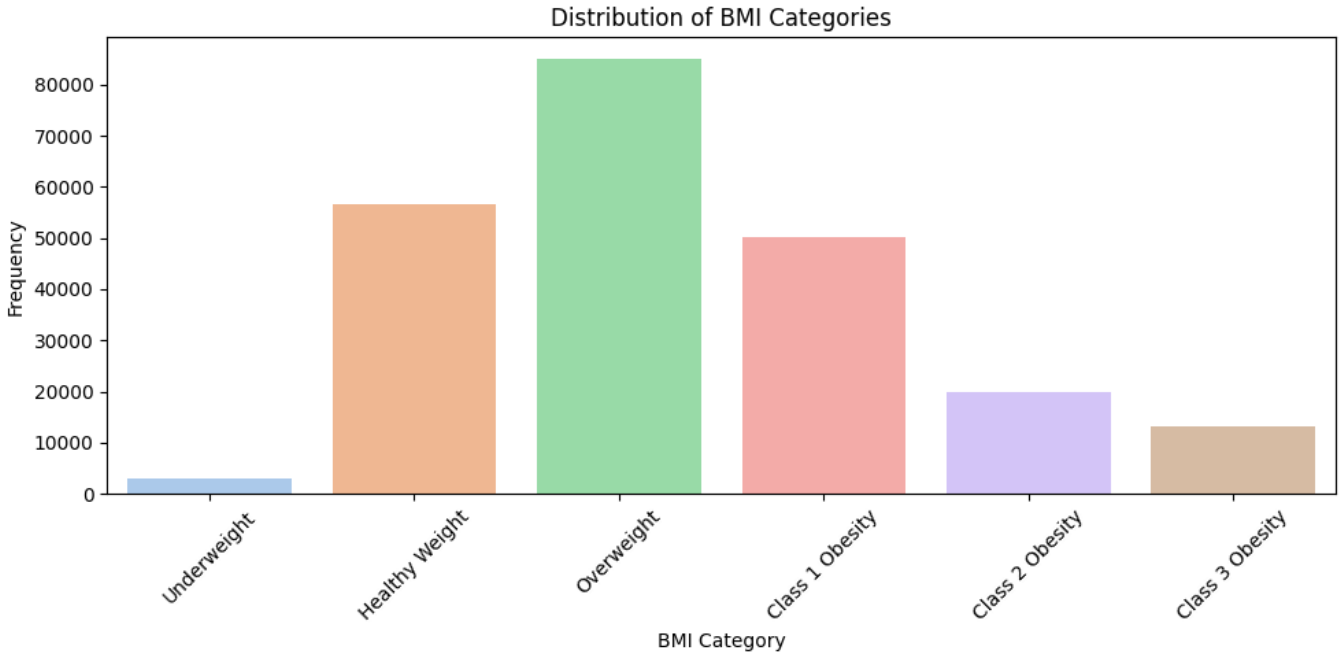

Figure 4: BMI Distribution After Discretization

Resampling

This is where we take the most crucial step for this dataset - resampling. Since we have 3 classes, class 0 being a majority at 83% and class 1 being a minority at 1.8%, a decision was made to first under sample the majority class to an acceptable level and oversample the minority class to be close to total representation of the middle class, class 2.

The under-sampling process was rather easy and straightforward, just randomly reducing the size of the class to a specific number. The number chosen was 150,000 after various experimentation with the models later.

Afterwards, SMOTENC was employed for its ability to work with categorical data to oversample the minority class as well as the middle class to also reach the same 150,000 size, ensuring that all three target classes are more equally represented.

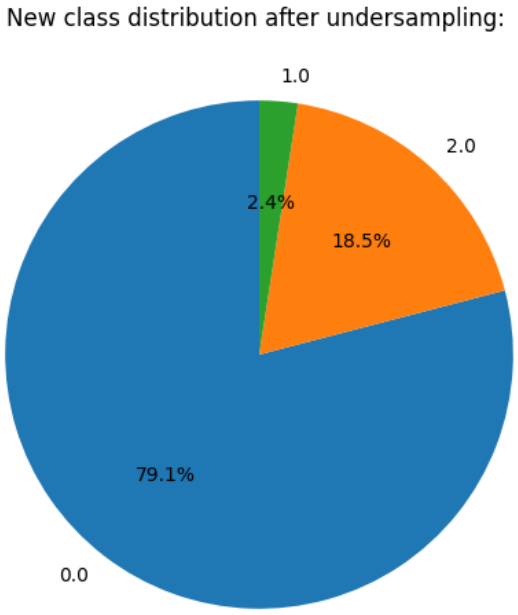

Figure 5: Target class distribution (After Underdamping Majority)

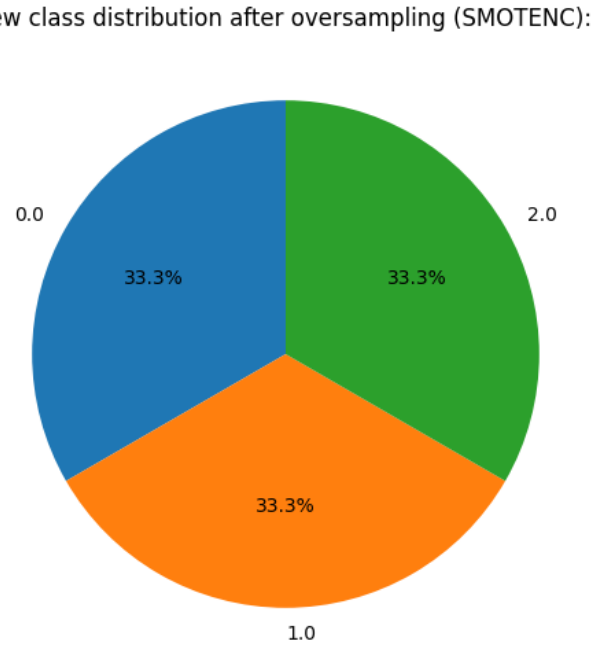

Figure 6: Target class distribution (After Oversampling Minority classes)

Outlier Handling

Not a lot of outliers were technically found in this section, but several features tended to have a class imbalance within them. Although there are ways to deal with it, future processes dealt with it just as effectively and seamlessly, such as the following section about feature selection.

Feature Selection

In this section, we prepare the dataset for modeling by selecting the most relevant features to predict the outcome effectively without overfitting the model. To achieve this, we use different feature selection methods, comparing their results to find the best set of features.

We employed the following methods:

1. **Filter method:** Correlation-based Feature Selection
2. **Filter method:** Chi-Squared Test (for categorical data)
3. **Embedded method:** Random Forest

The first step in feature selection involved examining correlations between the features. High correlations between certain features might indicate redundancy. To visualize this, we computed and plot the correlation matrix. However, none of these correlations were strong enough to deem the

information redundant as shown below.

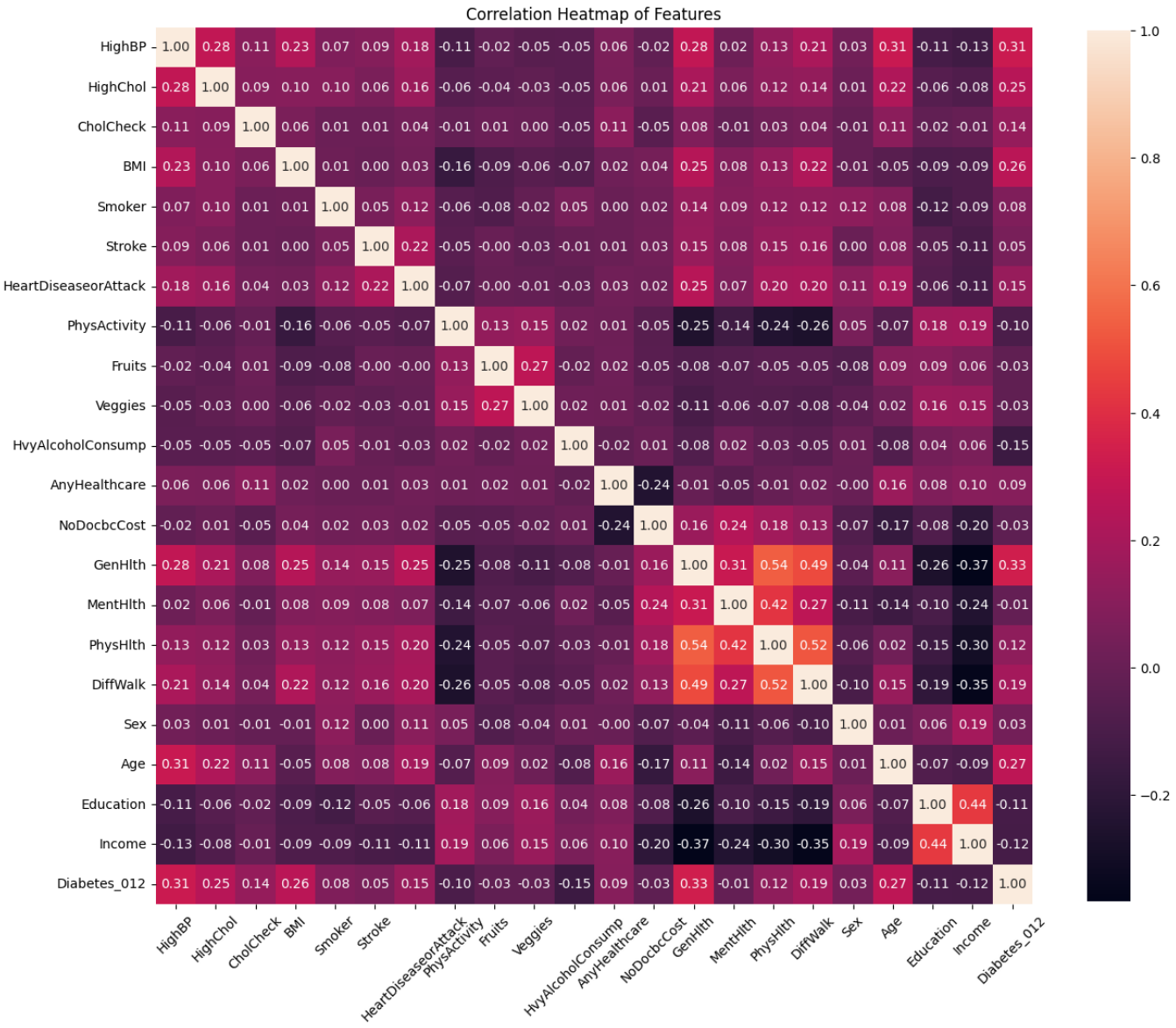

Figure 7: Correlation Matrix Across All Features

A Chi-Squared test was performed in order to find out how each feature correlated with the target variable. It checked if the distribution of values for each feature differs significantly from what would be expected, assuming that the feature is independent of the target. The Chi-Squared scores and p-values for each feature were calculated, applying the Chi-Squared test. The features were then ranked according to the Chi-Squared score, whereby the features with the highest scores were considered most relevant for prediction. The top 20 features were taken for further analysis.

For the embedded method, Random Forest was used in order to estimate the feature importance while training the models. One of the strengths of using Random Forest is the ready-made way of ranking features based on their usefulness in prediction of the target variable. By training a Random Forest model and inspecting the feature importances, one could find out which features contributed most to the model's predictions. The top 17 features were selected based on their importance scores.

Finally, we combine the features selected by both the Chi-Squared test and Random Forest to identify the most commonly selected features. The intersection of these two sets gives us the final set of features for model training.

Model Training and Evaluation

As we finally arrive at the meat and bone of our work, we need to address a couple of things to justify what comes later. The domain of this dataset is medical—one where there is a diagnosis of a disease that can put lives in danger if misdiagnosed. As such, the goal of this training is to increase the recall (False Negatives) for classes 1 and 2, representing "prediabetes" and "diabetes" respectively.
The following sequence of methodologies was used to achieve this goal:

- Implement different models: Random Forest, Decision Tree, LightGBM, Logistic Regression
- Hyperparameter Tuning: Perform grid search or random search for hyperparameter optimization
- Display Learning Curve for each to detect bias/variance
- Generate 3 versions of each model to show performance change:
    - Without feature selection and hyperparameters
    - Without feature selection, but with hyperparameters
    - With feature selection and hyperparameters
- Evaluate performance: accuracy, precision, recall, F1-score, and ROC-AUC
- Confusion Matrix: Goal to have minimal false negatives

Each model was trained and evaluated in three distinct configurations:

1. Baseline Model: Trained without feature selection or hyperparameters to provide a performance benchmark.
2. With Hyperparameter Tuning: Trained without feature selection but optimized to improve model performance.
3. With Feature Selection and Hyperparameters: Uses selected features and tuned parameters to maximize predictive accuracy and efficiency.

Each setup shows how feature selection and tuning impact performance. This approach reveals the importance of identifying the best model setup that maximizes predictive accuracy while minimizing false negatives for critical medical diagnoses, especially in a medical dataset where precision and recall are critical.

In this section, there were a variety of other models that I experimented with that didn't make it here due to their inherit limitations, or limitations in the context of my dataset, which made them challenging to use for this project.

The list includes:

- SVMs: They were too time-consuming. Fitting the model and experimenting with hyperparameter tuning took excessive time and were computationally expensive.
- KNNs: Similar to SVMs, KNNs work well on smaller datasets, which was not the case here. The choice was between a smaller k value, yielding poor results, or a larger k value, which took hours to display in the learning curve or even fit.
- Linear Regression with SGD: This model is too simple to capture meaningful insights. No matter how much effort went into adding complexity, a linear model cannot yield valuable results on a non-linear dataset.
- XGBoost: Although it should theoretically yield good results, the dataset's size made it difficult to complete. It was replaced by LightGBM, which performed better in this context.

Comparison of Models

After cleaning, resampling, and training various models, we can now compare their performance across different metrics. Each model is evaluated based on accuracy, precision, recall, F1-score, and ROC-AUC, which provides a comprehensive view of their predictive capabilities. Starting with the most notable graph that compares accuracy for all models of different configurations:

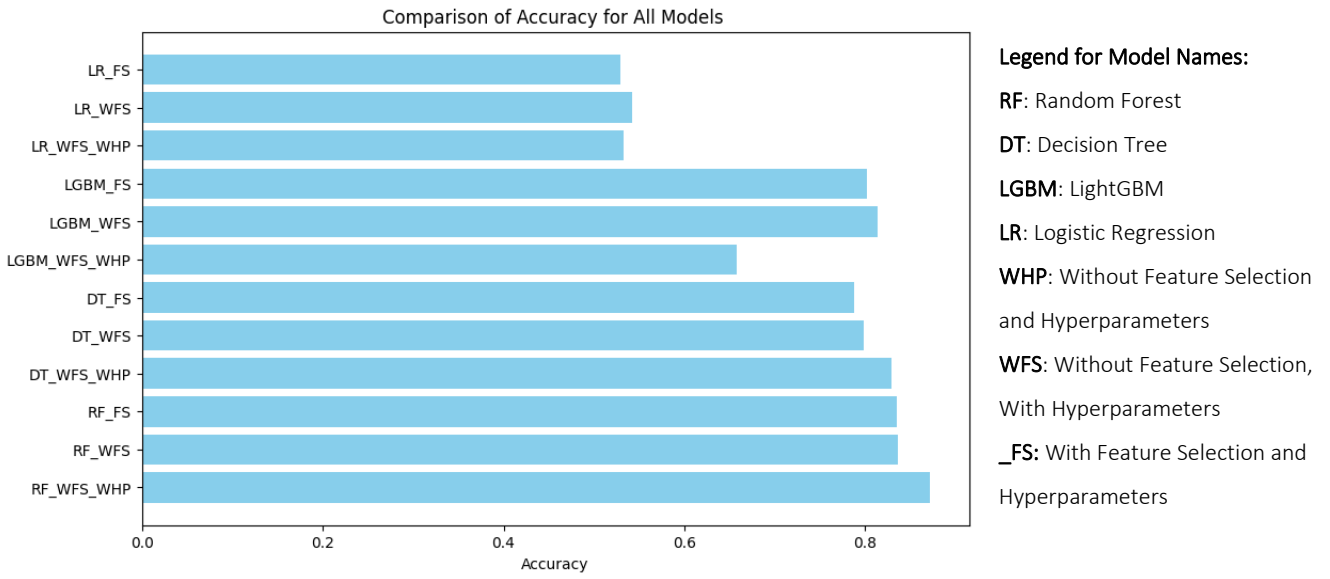

Figure 8: Accuracy Comparison Across All Models

This trend can also be clearly seen in the precision, recall and f1-scores accordingly. Indicating that if a model performs well in one metric, it well performs just as good in another. Perhaps the greatest save by grace was the LightGBM model. As seen above, it seems as though it is the only model that benefitted the most from the feature selection and hyperparameter tuning and went on to achieve comparable, if not greater results, to the baseline Decision Tree. This, along other trends in this graph, can also be further illustrated in the following ROC curves:

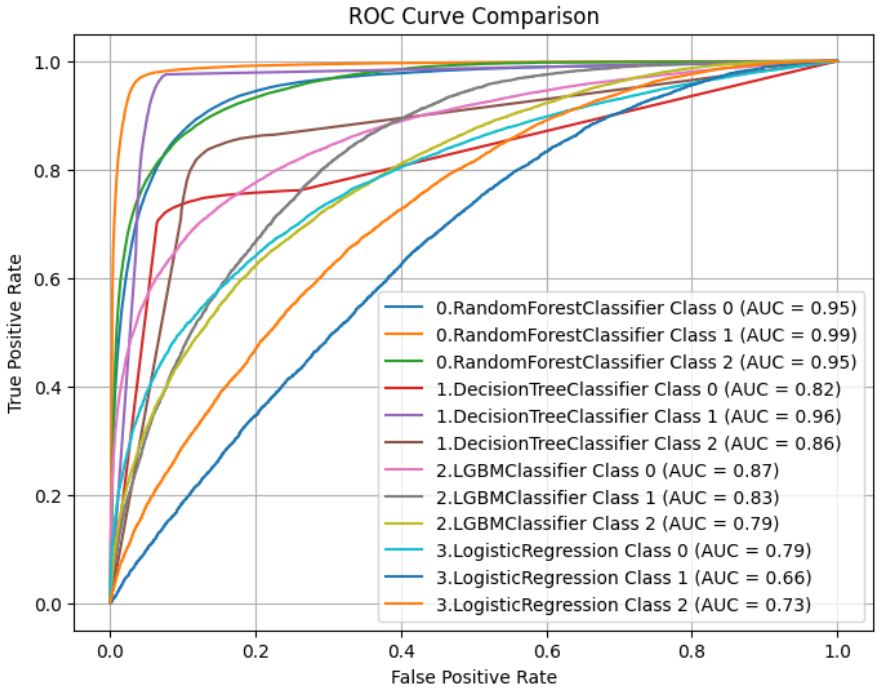

Figure 9: ROC Curve Comparison Between Baseline Models

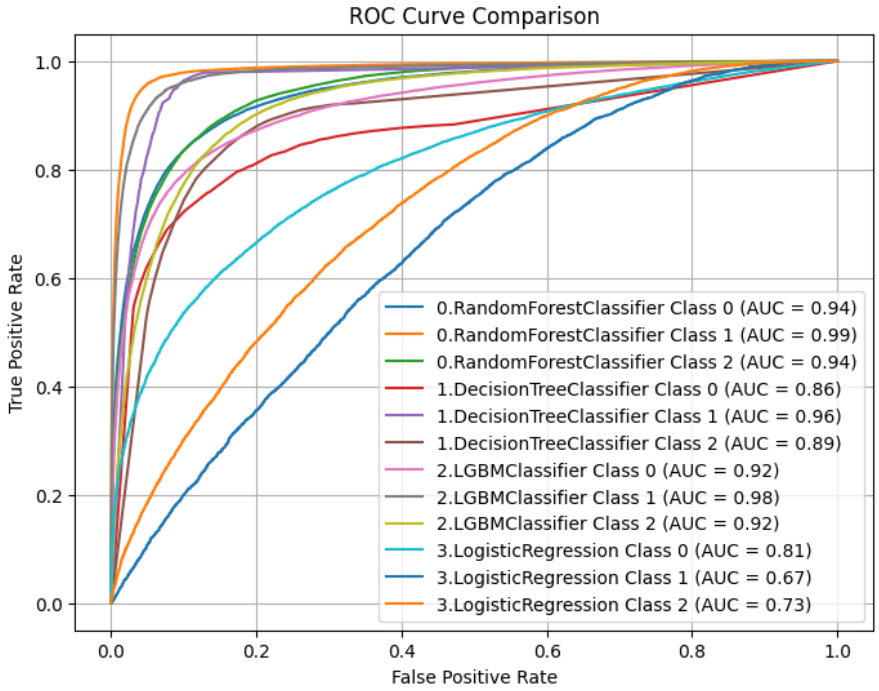

Figure 10: ROC Curve Comparison Between Models
(With feature selection and hyperparameter tuning)

Conclusion

After several trials and assessments, Random Forest turned out to be the top-performing model in predicting the diagnosis of diabetes. It has outperformed all other models in a number of metrics, including recall, which is a very key metric in this scenario, because it reduces false negatives. The inclusion of feature selection and hyperparameter tuning does not enhance the model's performance in the metric but does achieve comparable levels of efficiency with lower computational power. Thus, we conclude that Random Forest is the most suitable model for predicting diabetes in this dataset, offering the best trade-off between accuracy, recall, and interpretability

.