Comprehensive Report on AI Coursework: Team Contributions and Individual Model Extensions

Anndischeh Mostafavi

October 2024

Dataset Link

Link

1 Introduction

Diabetes is a chronic and widespread health condition affecting millions of individuals globally, with significant implications for public health and economic burden. In the United States, diabetes and its precursor, prediabetes, pose a major challenge, impacting 34.2 million Americans and putting an additional 88 million at risk. The prevalence of diabetes varies by factors such as age, socioeconomic status, and lifestyle choices. Early detection and intervention are critical for managing the disease and reducing associated complications, including heart disease, kidney failure, and vision loss. Predictive modeling of diabetes risk is a powerful tool for identifying high-risk individuals and informing public health strategies.

This project aims to predict diabetes status (a **classification** task) using health indicators from the ”diabetes _012_health_indicators_BRFSS20151” dataset, part of the BRFSS survey. The project involves importing necessary libraries, performing exploratory data analysis (EDA), visualizing key features, and preprocessing the data to handle missing values, outliers, and class imbalance. Several machine learning models, including random forest, decision tree, and K-nearest neighbors, will be trained and evaluated using accuracy, precision, recall, F1-score, and ROC-AUC metrics to predict diabetes status effectively Teboul (2015).

2 Description of data set

This study utilizes the Diabetes 012 Health Indicators BRFSS 2015 dataset, derived from the Behavioral Risk Factor Surveillance System (BRFSS) survey conducted by the Centers for Disease Control and Prevention (CDC). The BRFSS is an annual, large-scale health-related survey designed to gather data on health behaviors, chronic diseases, and preventive practices. The dataset, available on Kaggle, includes 253,680 responses and provides a rich set of 21 features capturing key health indicators such as BMI, physical activity, smoking status, and age Centers for Disease Control and Prevention (2015).

3 Exploratory Data Analysis (EDA)

plot 1(a) shows the distribution of target column”Diabetes_012” which contains 3 classes [0: No diabetes or only during pregnancy, 1: Prediabetes, 2: Diabetes]. This plot tells dataset is not balanced, too many “zero class” but a few “one class”. To handle this class imbalance I used SMOTE which created artificial samples for the minority classes that solved our problem of class imbalance. Inplot 1(c)Correlation Matrix and Correlation Heatmap works base on the correlation between each pairs of the columns in dataset, the hottest colors show higher correlation, and neutral colors show near zero correlation. I realize GenHealth and diff walk and MentHealth have a higher correlation(positive), income and Genhealth higher negative correlation, and Diabetes(target) is not significantly correlated to specific columns.

As shown in plot 1(b) Proportion of HighBP and HighCol is shown by the status of diabetes, in order (0,1,2 class) have higher value of HighBP and HighCol.

As shown in Plot1(d), the BMI distribution for classes 0, 1, and 2 reveals that the highest density of BMI values is observed in individuals without diabetes (class 0.0), peaking between BMI 20 and 30. This group also has the largest population, as reflected by the high density and histogram frequency. The BMI density plot provided valuable insights into the data distribution, revealing skewness and the presence of outliers, particularly in the “BMI” feature. These outliers prompted the decision to apply the IQR-based outlier removal (Interquartile Range Outlier Removal) method during preprocessing,

ensuring that extreme values did not skew the analysis or negatively impact the model's performance.
The Box Plot 1(e) of Age based on the diabetes category shows although age is not significantly effective but on average the Age of people with diabetes is higher.

4 Data Preprocessing

4.1 Handling Missing Values

I identified missing values in the dataset and removed any rows containing them. Although handling missing values can involve imputation or advanced techniques, I opted for complete-case analysis because the dataset had relatively few missing values. Removing these rows ensured simplicity without compromising the quality of the data.

- Missing values before cleaning: **0** (no missing values were present).
- Missing values after cleaning: **0** (the dataset remained complete).

4.2 Handling Outliers

I used the Interquartile Range (IQR) method to identify and remove outliers from the **BMI** column using Equation (1) and Equation (2).
Outliers were defined as values falling outside:

$$\text{Lower Bound} = Q1 - 1.5 \times \text{IQR} \tag{1}$$

$$\text{Upper Bound} = Q3 + 1.5 \times \text{IQR} \tag{2}$$

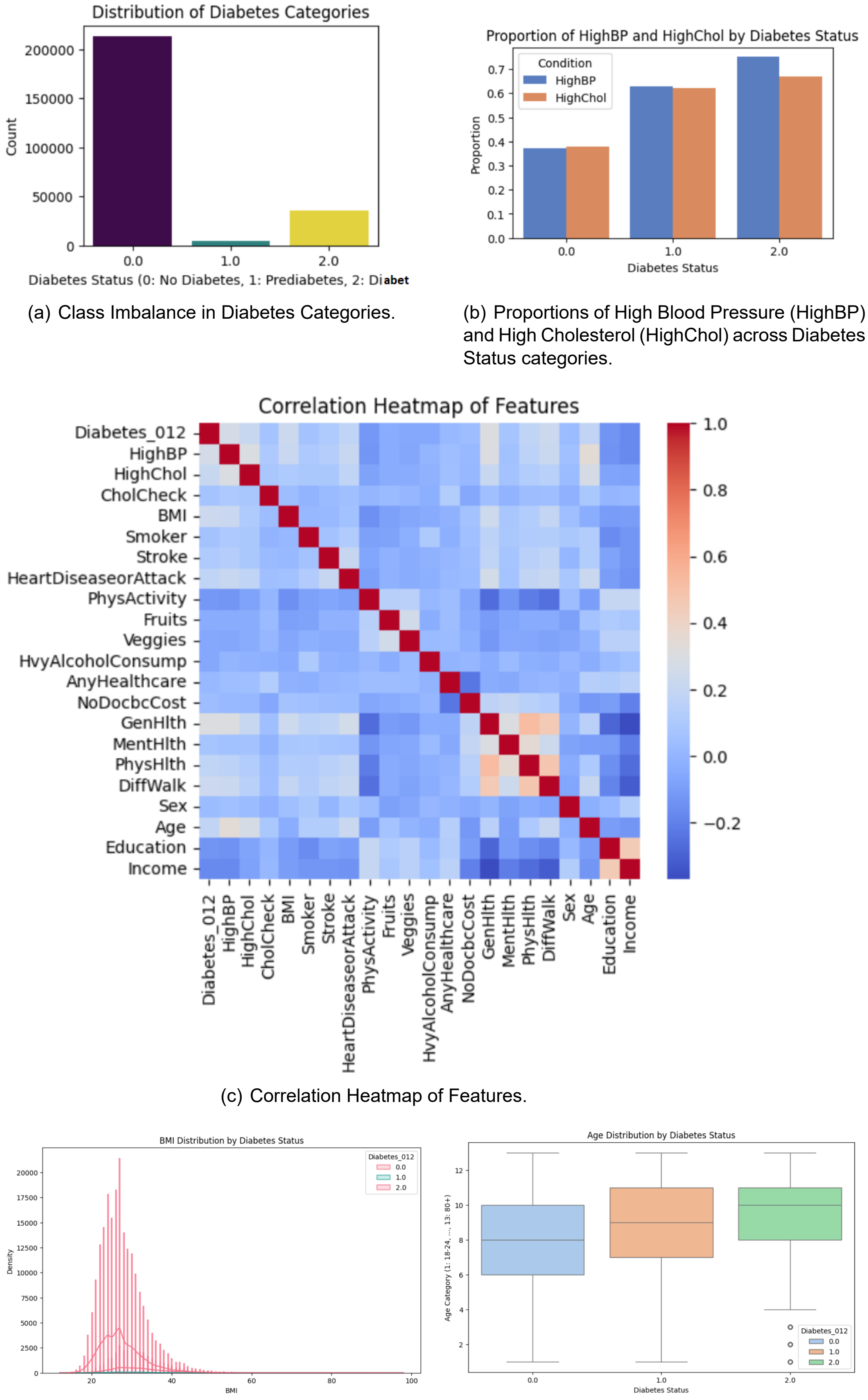
Rows with BMI values beyond these bounds were excluded.
Outliers can disproportionately influence models, particularly those sensitive to extreme values. By removing outliers, I ensured that the data distribution was more representative of typical cases, which improved model stability and accuracy.

4.3 Encoding Categorical Variables

I did not perform additional encoding since all categorical variables in the dataset were already encoded numerically.
Further encoding was unnecessary, as the dataset's current format was sufficient for machine learning models.

4.4 Scaling Numeric Features

I standardized numeric features (BMI, MentHlth, PhysHlth, Age) using 'StandardScaler' to transform these features to have a mean of 0 and a standard deviation of 1. Many machine learning models are sensitive to feature scales, and standardizing ensures that all numeric features contribute equally to the model training process. This is particularly important for gradient-based methods like logistic regression and support vector machines.



(a) Class Imbalance in Diabetes Categories.

(b) Proportions of High Blood Pressure (HighBP) and High Cholesterol (HighChol) across Diabetes Status categories.

(c) Correlation Heatmap of Features.

(d) BMI distribution across Diabetes Status categories, highlighting differences with density curves.

(e) Boxplot showing age distribution across Diabetes Status categories, with age represented in grouped categories (1: 18-24, ..., 13: 80+)..

Fig. 1: Data visualization plots showing the distribution of diabetes categories, correlations between features, and the relationship between diabetes status and various health conditions such as high blood pressure, high cholesterol, BMI, and age Mostafavi (2024).

4.5 Addressing Class Imbalance

I used the Synthetic Minority Oversampling Technique (SMOTE) to address the class imbalance in the target variable (Diabetes_012).

- Original class distribution: {0.0: 207,514, 2.0: 31,979, 1.0: 4,340}.

- After applying SMOTE: {0.0: 207,514, 2.0: 207,514, 1.0: 207,514}.

Class imbalance can cause models to favor the majority class, leading to poor generalization for minority classes. By oversampling minority classes, I ensured the model learned equally from all classes, improving predictive performance across the dataset.

4.6 Splitting the Data

I split the preprocessed dataset into training and testing sets using an 80-20 ratio. Splitting the data allows us to evaluate model performance on unseen data. Using a consistent split ensures fair comparison across all models and prevents data leakage.

5 Method

5.1 Logistic Regression

Logistic Regression is a linear model widely used for classification problems Russell & Norvig (2016). It predicts class probabilities using the logistic function and applies L2L_2-regularization to control overfitting.
The model was initialized with a regularization strength (C=1C = 1) and the liblinear solver, suitable for small datasets. I trained it on XtrainX_train and ytrainy_train, iterating up to a maximum of 200 steps to ensure convergence. Logistic regression served as our baseline for comparison due to its simplicity and interpretability.

Why Use Logistic Regression?
It is a simple and interpretable model, which is quick to train and test. It serves as a strong baseline model due to its efficiency and clarity in feature interpretation. Often, logistic regression performs well with linearly separable data, which is quite common in many classification tasks Russell & Norvig (2016), Learning (2020).

5.2 Random Forest

Random Forest is an ensemble method combining multiple decision trees to improve accuracy and reduce overfitting. It handles non-linear relationships and automatically captures feature importance Chicharro Raventos (2024), Russell & Norvig (2016).

Implementation:
I performed hyperparameter tuning using GridSearchCV to optimize the number of estimators (n_estimatorsn_estimators) in the forest. A parameter grid was defined, and cross-validation (cv=3cv=3) was used to ensure robust selection. The best model was extracted and trained on the dataset.

Why Use Random Forest?
This method is well-suited for complex datasets with non-linear relationships. It typically offers high predictive performance while maintaining a balance of interpretability through feature importance analysis. Random Forest is less prone to overfitting compared to single decision trees due to the ensemble effect of aggregating multiple trees.

5.3 Decision Tree

Decision Tree classifiers split the data recursively based on feature values, creating a tree structure for classification. This model is interpretable and can handle non-linear decision boundaries Chicharro Raventos (2024), Bengio et al. (2017).

Implementation:
I tuned the splitting criterion (gini or entropy) through a grid search with 5-fold cross-validation. The best decision tree model was selected and trained on the dataset. Its visual structure helped interpret decision-making processes.

Why Use a Decision Tree?
It provides easily interpretable decision-making rules, which are useful in explaining the model's behavior. While simple and transparent, it can capture non-linear relationships in the data. Decision trees can serve as a good baseline for comparison against more complex ensemble models like Random Forest.

5.4 K-Nearest Neighbors (KNN)

KNN is a non-parametric model that predicts the class of a sample based on the majority class among its kk-nearest neighbors. It is simple yet effective for datasets with well-separated classes Chicharro Raventos (2024), Russell & Norvig (2016).

Implementation:
I tuned the number of neighbors (n_neighborsn_neighbors) using grid search and divided XtrainX_train into training and validation sets to evaluate during tuning. The best model was chosen and retrained on the training data.

Why Use KNN?
KNN is simple and effective for small datasets with well-separated classes. It serves as a strong non-parametric baseline, especially useful when no assumptions about data distribution can be made. KNN can capture local patterns in the data but is slower in large datasets due to its computational complexity.

5.5 Stacked Model (Ensemble)

Stacking is an ensemble learning method combining predictions of multiple base models through a meta-model. It leverages the strengths of diverse algorithms to achieve higher accuracy Chicharro Raventos (2024).

Implementation:
Base models included Random Forest, Decision Tree, and SVM. Logistic Regression was selected as the meta-model for its interpretability. Cross-validation (cv=3cv=3) was applied during the stacking process to ensure robustness. The ensemble demonstrated how combining predictions from different models could improve overall classification accuracy.

Why Use Stacking?
This approach combines the strengths of multiple classifiers, improving overall predictive performance. Stacking often results in better performance than any individual classifier, as it leverages diverse models to capture different patterns in the data.
It provides a robust comparison framework, ensuring that all models are evaluated together for optimal performance selection.

Tab. 1: Performance metrics of various machine learning models on the diabetes dataset, including accuracy, ROC-AUC, and F1 Score Mostafavi (2024).

Model	Train Accuracy (%)	Test Accuracy (%)	ROC-AUC	F1 Score
Logistic Regression	52.58	52.63	0.7183	0.5140
Random Forest	66.98	66.40	0.8453	0.6629
Decision Tree	95.10	86.66	0.9320	0.8660
K-Nearest Neighbors	93.97	89.41	0.9558	0.8906
Multi-Layer Perceptron	71.01	70.59	0.8753	0.7014
Nested-Model	99.58	94.72	0.9937	0.9473
Genetic Algorithm	52.86	52.85	0.7205	0.5219

5.6 Genetic Algorithm

GA is a heuristic optimization technique inspired by natural selection. It evolves a population of solutions (hyperparameters) across generations to find the optimal set Chicharro Raventos (2024), Russell & Norvig (2016).

Implementation: First, I Randomly create hyperparameter combinations. Then I evaluate each combination's performance using model accuracy on validation data. I choose the best candidates based on fitness. Creates new candidates by combining/mutating hyperparameters of selected parents. Repeats for multiple generations, returning the best-performing hyperparameters.

Why Use GA?
Effective for complex, non-linear hyperparameter spaces. Adaptable and faster than exhaustive search when parameter space is large. Leverages parallelism for efficient computation.

5.7 Multi-Layer Perceptron

Grid Search systematically tests all combinations of hyperparameters within a predefined grid using cross-validation, selecting the best-performing model configuration Chicharro Raventos (2024), Bengio et al. (2017).

Implementation:
Defines hyperparameter options for MLP (e.g., activation functions). Evaluates each configuration on different data splits. Finds and trains the best model using accuracy as the scoring metric.

Why Grid Search?
MLP can model complex, non-linear relationships in data due to its multiple layers and non-linear activation functions. It can adapt to various tasks, such as image recognition, text classification, or general tabular data prediction, making it highly flexible. Hidden layers in MLP extract hierarchical feature representations, improving predictive performance.

6 Result

6.1 Logistic Regression

Logistic Regression performed poorly, with low precision, recall, and F1 scores. The results suggest it struggles to capture the underlying patterns in the data due to its linear decision boundaries (As shown in Fig.3 and Tab.1 . The model is insufficient for handling

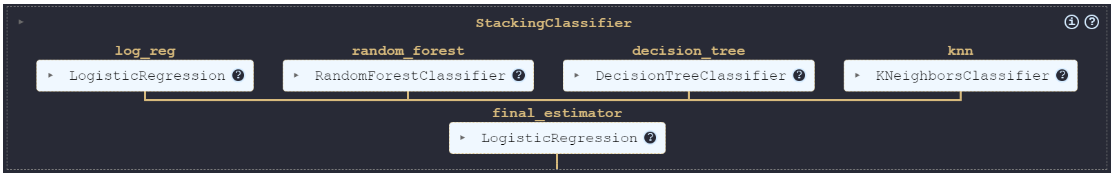


Fig. 2: Stacking Classifier: Combining multiple base models (Logistic Regression, Random Forest, Decision Tree, and K-Nearest Neighbors) with a Logistic Regression meta-model for improved predictions.Mostafavi (2024).

complex relationships and class imbalance, as evidenced by its lower performance for class ‘1.0‘ compared to ‘0.0‘ and ‘2.0‘ Chicharro Raventos (2024).

6.2 Random Forest

Random Forest shows a significant improvement over Logistic Regression via Fig.3 and Tab.1. Its ability to handle non-linear relationships allows for better predictions across all classes, though class ‘2.0‘ is still challenging. Robust to non-linear patterns and moderately handles class imbalance. While better than Logistic Regression, the model's reliance on majority voting and splitting criteria can lead to issues with minority classes or overlapping data.

6.3 Decision Tree

The Decision Tree model performs well on the test set but exhibits overfitting due to its high training accuracy as showes in Fig.3 and Tab.1. The simplicity of the tree structure can overly fit the noise in the training data. Effective for capturing patterns in small datasets or simple features. Susceptible to overfitting, especially on imbalanced datasets or noisy data.

6.4 K-Nearest Neighbors (KNN)

KNN outperforms Decision Tree in terms of generalization. According to Fig.3 and Tab.1the high precision and recall reflect its capacity to handle multi-class classification effectively. Non-parametric and adapts well to complex boundaries in the data.Computationally expensive with increasing data size and sensitive to noisy or irrelevant features.

6.5 Multi-Layer Perceptron (MLP)

MLP demonstrates balanced performance across classes, leveraging its ability to model complex patterns as . However, it does not outperform KNN and Decision Trees due to potential limitations in hyperparameter tuning. Effective for non-linear relationships and robust to class imbalance. Requires extensive tuning and is computationally intensive. shown in Fig.3 and Tab.1

6.6 Nested-Model

This model achieves the highest accuracy and AUC, demonstrating excellent generalization and class balance (as shown in Fig.2). Its ability to fine-tune multiple layers or sub-models likely explains its superior performance. Highly adaptive and captures intricate relationships in the data. The risk of overfitting if not regularized, and computational complexity is high.

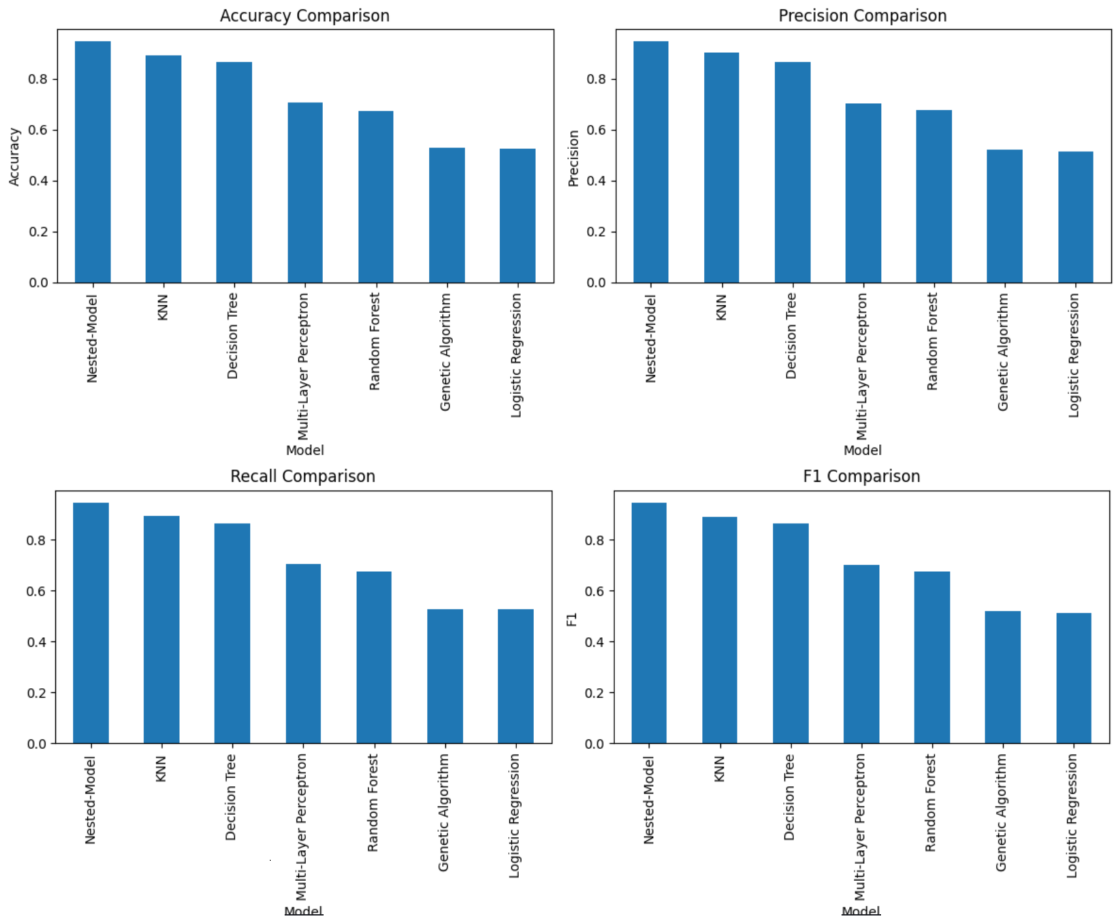


Fig. 3: Comparison of models based on Accuracy, Precision, Recall, and F1-scoreMostafavi (2024).

6.7 Genetic Algorithm

Despite the advanced optimization nature of Genetic Algorithms, this model underperforms (regarding to Fig.3 and Tab.1). It demonstrates limited improvement over Logistic Regression, likely due to challenges in feature selection or parameter tuning. Useful for feature selection and optimization. Computationally expensive and sensitive to initialization.

7 Conclusion

Random Forest provides a good balance between interpretability and robust performance, making it a reliable choice for most applications.
Nested Model demonstrates superior performance but requires significant computational resources, making it ideal for scenarios where accuracy is paramount and resources are not constrained.
Fine-tuning the hyperparameters of the **Multi Layer Perceptron (MLP)** or employing ensemble techniques with **Decision Trees** could further improve model performance, leveraging their strengths for complex patterns.

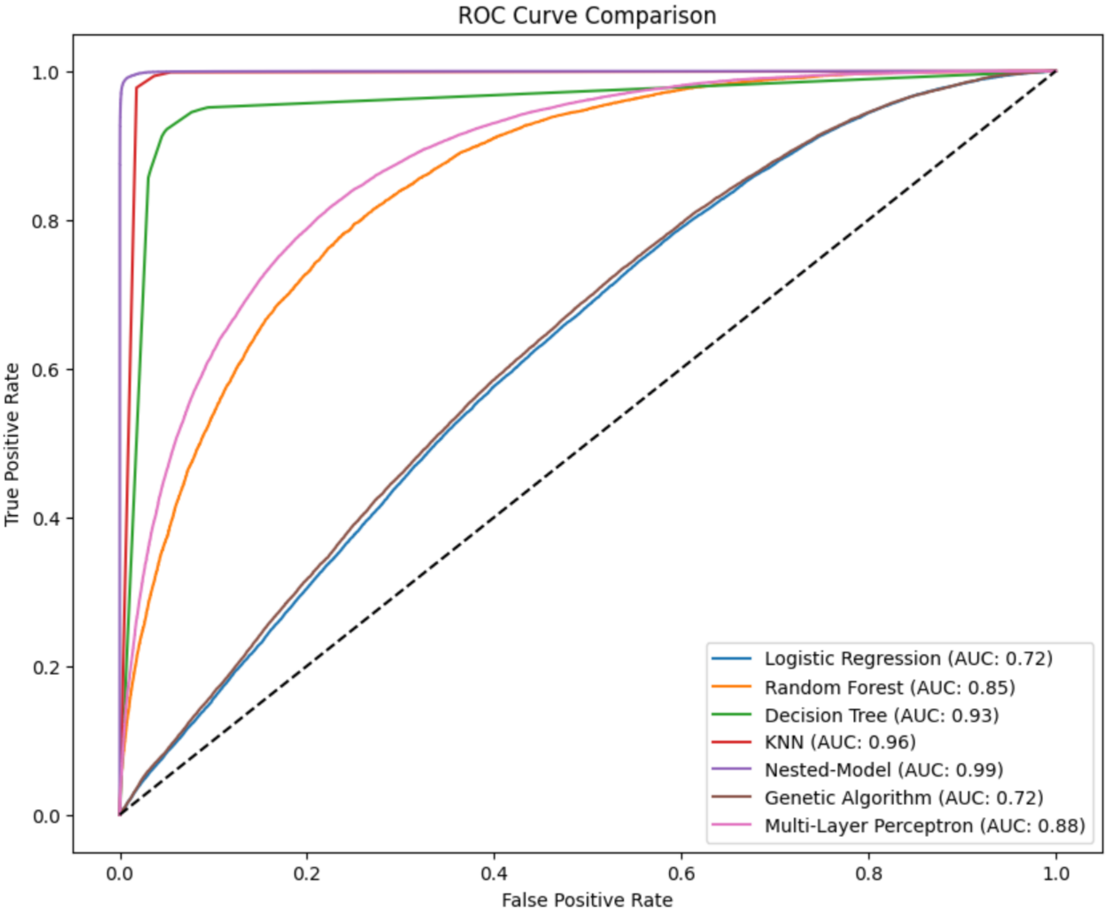


Fig. 4: ROC curves for the machine learning models, illustrating their performance in distinguishing between classes based on true positive and false positive ratesMostafavi (2024).

References

Bengio, Y., Goodfellow, I. & Courville, A. (2017), *Deep learning*, Vol. 1, MIT press Cambridge, MA, USA.

Centers for Disease Control and Prevention (2015), ‘Diabetes health indicators dataset’, `https://www.kaggle.com/datasets/alexteboul/diabetes-health-indicators-dataset`.

Chicharro Raventos, D. (2024), ‘Introduction to ai module’, University Lecture.

Learning, P. (2020), ‘Pattern recognition and machine learning| christopher bishop’.

Mostafavi, A. (2024), ‘Github repository: Introduction to ai coursework’, `https://github.com/Anndischeh/Introduction-to-AI-Coursework`.

Russell, S. J. & Norvig, P. (2016), *Artificial intelligence: a modern approach*, Pearson.

Teboul, A. (2015), ‘Kaggle dataset: Diabetes health indicators’, `https://www.kaggle.com/datasets/alexteboul/diabetes-health-indicators-dataset`. Accessed on: [2014].