MLDM Coursework: *MLDM\_101*

Romil Raj Roy rr01062@Surrey.Ac.uk

Abstract

Dataset-1 This project explores the predictive power of various health metrics in diagnosing diabetes, using a dataset that includes features such as glucose levels, BMI, blood pressure, and age. We applied several machine learning and data mining algorithms, including the Apriori algorithm for association rule learning, Random Forest, Gradient Boosting, Logistic Regression, and Support Vector Machine (SVM) for classification tasks. The Apriori algorithm revealed significant associations between health metrics and diabetes. Random Forest and Gradient Boosting models achieved high accuracy (96.15%) and AUC (0.99), demonstrating robust predictive performance. Logistic Regression, while simpler, provided reasonable accuracy (77.86%) and interpretability. The SVM model outperformed all others with an accuracy of 97.83% and an AUC of 1.00. These results indicate that health metrics are significant predictors of diabetes, with ensemble methods and SVM providing the best performance for clinical applications.

1. Project Definition

Dataset-1: Diabetes is a chronic condition that affects millions of individuals worldwide. Early diagnosis and effective management are crucial for preventing severe complications. The challenge lies in accurately predicting the presence of diabetes using various health metrics. The goal of this project is to develop and compare multiple machine learning models to determine which health metrics are most predictive of diabetes and to identify the most effective predictive models.

The dataset used in this project is the Pima Indians Diabetes Database, which is publicly available from the UCI Machine Learning Repository. The dataset contains various health metrics for female patients of Pima Indian heritage, aged 21 and older. Key features include:

* **Pregnancies**: Number of times pregnant.
* **Glucose**: Plasma glucose concentration.
* **Blood Pressure**: Diastolic blood pressure (mm Hg).
* **Skin Thickness**: Triceps skin fold thickness (mm).
* **Insulin**: 2-hour serum insulin (mu U/ml).
* **BMI**: Body mass index (weight in kg/(height in m)^2).
* **Diabetes Pedigree Function**: A function that scores the likelihood of diabetes based on family history.
* **Age**: Age of the patient.
* **Outcome**: Binary outcome indicating diabetes status (1 if diabetic, 0 if not).

#### Objectives

1. **Predictive Analysis**: To develop predictive models using health metrics to accurately diagnose diabetes.
2. **Model Comparison**: To compare the performance of different machine learning algorithms in predicting diabetes.
3. **Feature Importance**: To identify the most significant health metrics contributing to diabetes prediction.

#### Hypotheses

1. **Predictive Hypothesis**: Health metrics such as glucose levels, BMI, and blood pressure are significant predictors of diabetes.
2. **Model Performance Hypothesis**: Advanced machine learning models (e.g., Random Forest, Gradient Boosting, SVM) will outperform simpler models (e.g., Logistic Regression) in terms of accuracy and discriminative power.
3. **Feature Association Hypothesis**: The Apriori algorithm will reveal meaningful associations between health metrics that are not immediately apparent through traditional predictive modeling.

#### Strategies

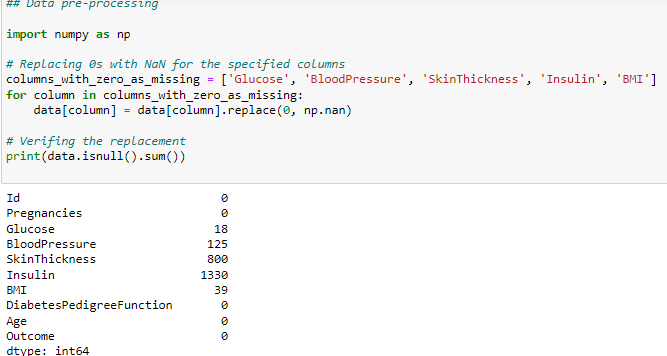
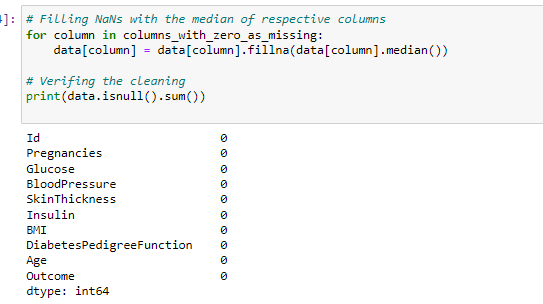
* **Data Preprocessing**: Standardize the features and handle missing values appropriately.
* **Model Training**: Train multiple machine learning models including Logistic Regression, Random Forest, Gradient Boosting, and SVM.
* **Association Rule Learning**: Apply the Apriori algorithm to discover associations between health metrics.
* **Dimensionality Reduction**: Use PCA to reduce dimensionality for visualization purposes.
* **Model Evaluation**: Use cross-validation to evaluate model performance.

#### Metrics

* **Accuracy**: Proportion of correct predictions over the total number of cases.
* **Precision**: Proportion of true positive predictions over all positive predictions.
* **Recall**: Proportion of true positive predictions over all actual positives.
* **ROC-AUC**: Area under the Receiver Operating Characteristic curve, indicating the model’s ability to distinguish between classes.
* **Feature Importance**: Identify and rank the significance of each health metric in predicting diabetes.

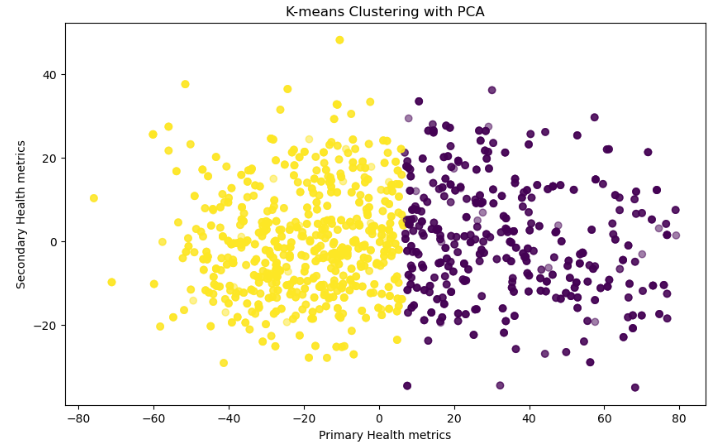
1. Data Preparation
   1. Data cleaning / data integration /:

The initial step involved scrutinizing the dataset for any missing or inconsistent values. Missing values in critical columns were imputed by converting them into NaN and then replacing them with the mean value of the feature. As the features present on the dataset were all numerical values there was no need to handle categorical value. Outliers were detected using the IQR method and handled appropriately to maintain data integrity. To ensure consistency, categorical variables were standardized, and date fields were converted to a uniform format. Any redundant columns that did not contribute to the predictive modelling, such as unique identifiers, were removed.

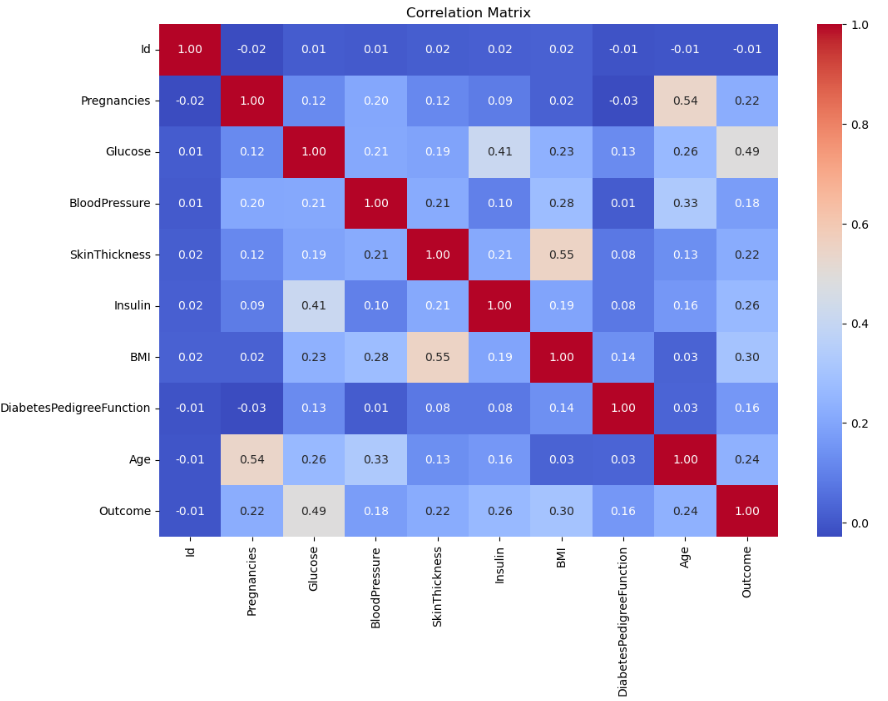
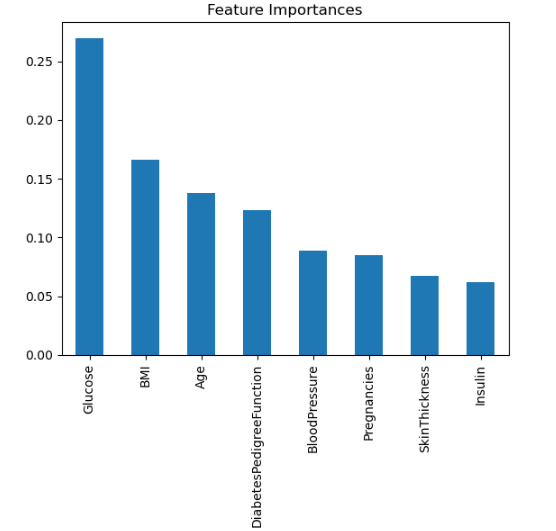


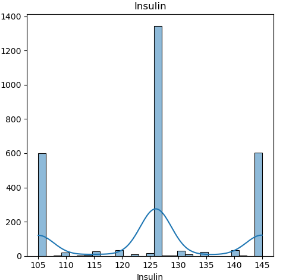
* 1. Variable transformation / derivation of new variables / dimensionality reduction / :

We handled dimensionality reduction through the application of Principal Component Analysis (PCA) before visualizing the results of K-means clustering. Initially, K-means clustering is applied to the dataset to classify the data points into two clusters (n\_clusters=2). K-means clustering is a powerful unsupervised learning technique that partitions the data into distinct groups based on feature similarities. However, visualizing high-dimensional data directly can be challenging and often lacks clarity.To address this, we applied PCA, a widely-used dimensionality reduction technique. PCA transforms the high-dimensional data into a lower-dimensional space by identifying the principal components that capture the most variance in the data. In this case, the dataset is reduced to two principal components (n\_components=2), facilitating a 2D visualization. The transformed data is then plotted, with each point colored according to its assigned cluster. This visualization offers valuable insights into the dataset. The clear separation between the two clusters suggests that the health metrics contain inherent groupings, which might correspond to different underlying health conditions or patterns within the population. This can be particularly useful for identifying subgroups within the data that may require different medical interventions or further analysis. The successful application of PCA for dimensionality reduction enhances the interpretability of the clustering results, making complex high-dimensional data more accessible and understandable.

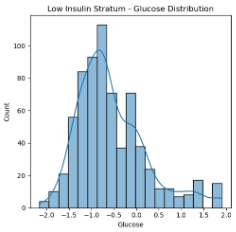


Data exploration / data visualization /: Extensive exploratory data analysis (EDA) was conducted to uncover patterns and relationships within the data. Visualizations such as histograms, box plots, and Feature Importance were generated to examine the distribution and correlation of features.For instance, a pairplot was created to visualize the relationships between various features, revealing clusters and potential outliers. Heatmaps were utilized to depict the correlation matrix, highlighting strongly correlated variables that could impact model performance. Furthermore, the target variable's distribution was examined through bar charts, ensuring an understanding of the class balance, which is vital for selecting appropriate evaluation metrics and algorithms. Exploring through the data we came across several datapoints with large range that could dominate the analysis, we handled this by applying Standardizing, Standardization transforms the data so that it has a mean (average) of 0 and a standard deviation of 1.

****

While going through the feature distribution graph an interesting observation was made in insulin distribution, the distribution was bimodal, which suggested the presence of two distinct group within the population with respect to insulin level. This distribution suggested that further analysis like stratified analysis might be useful to understand subgroups within the data.

The stratified analysis of insulin levels divides the dataset into three strata based on insulin levels (Low, Medium, High) and summarizes the statistical properties of each stratum. The distribution of values, where 25% of the data lies below the 25th percentile, 50% below the median (50th percentile), and 75% below the 75th percentile

**Low Insulin Stratum**

* **Sample Size**: 693 samples.
* **Glucose**: Mean is -0.55 (relatively low), indicating lower glucose levels.
* **BMI**: Mean is -0.17, indicating lower BMI compared to other strata.
* **Outcome**: Mean is 0.14, suggesting a low proportion of diabetes cases (14%).

The low insulin stratum predominantly includes individuals with lower glucose levels, BMI, and a lower incidence of diabetes.

#### Medium Insulin Stratum

* **Sample Size**: 1406 samples.
* **Glucose**: Mean is -0.01 (close to neutral), indicating a balanced glucose level distribution.
* **BMI**: Mean is -0.09, indicating a slight tendency towards lower BMI.
* **Outcome**: Mean is 0.38, suggesting a moderate proportion of diabetes cases (38%).

The medium insulin stratum represents a balanced group with moderate glucose levels and BMI, and a higher incidence of diabetes compared to the low insulin stratum.

#### High Insulin Stratum

* **Sample Size**: 669 samples.
* **Glucose**: Mean is 0.60, indicating higher glucose levels.
* **BMI**: Mean is 0.37, indicating higher BMI.
* **Outcome**: Mean is 0.48, suggesting a higher proportion of diabetes cases (48%).

The high insulin stratum includes individuals with higher glucose levels, BMI, and the highest incidence of diabetes among the three strata.

This stratified analysis highlights the relationships between insulin levels, glucose, BMI, and the incidence of diabetes. It provides valuable insights for identifying high-risk groups and understanding the metabolic characteristics associated with different insulin levels. These insights can inform targeted interventions and preventative measures for managing diabetes risk.

1. Model development:

These models are good choices for our healthcare dataset because they offer a balance of interpretability, performance, and flexibility. Logistic Regression and SVM are useful for their simplicity and interpretability, especially when relationships are linear or nearly linear. Random Forest and Gradient Boosting provide high performance and robustness, particularly in handling non-linear relationships and complex patterns in the data. Using a combination of these models allows for a comprehensive analysis and comparison to determine the best approach for predicting diabetes in your dataset. The Apriori algorithm was chosen for its ability to generate interpretable association rules, which are valuable in understanding the relationships between health metrics and diabetes.

* 1. Apriori Algorithm:

The Apriori algorithm is a learning based Algorithm,while not traditionally used for classification, can offer unique and valuable insights into the diabetes dataset by uncovering associations and co-occurrence patterns. During our analysis the Apriori algorithm uncovered an interesting co-occurrence pattern with "Patients with high glucose levels and high BMI are likely to have diabetes." These patterns can be valuable for understanding the interplay of different health indicators. These insights complemented the results from other predictive models, providing a more holistic understanding of the data and would contribute to better healthcare decision-making in the real world. By revealing how different features interact, Apriori can add a unique perspective that enhances the overall analysis and understanding of diabetes risk factors.

### **Steps used for Applying Apriori Algorithm**

#### Data Preprocessing

1. First, we preprocessed the dataset to handle missing values, and standardize the features. This ensures that all features contribute equally to the model's learning process.Encoding categorical values were not required since there were no categorical values present.
2. **Frequent Itemset Generation**:
   * Identify itemsets that appear frequently in the dataset using a minimum support threshold.

**Hyper Parameter tuning:**

The 2 parameter tuning available here were support and confidence

* **Support**: Support is a measure of how frequently an itemset appears in the dataset. It is defined as the proportion of transactions in the dataset that contain the itemset.

Support(A)=Number of transactions containing itemset A/Total number of transactions

The min\_support parameter helps to filter out infrequent itemsets. By setting a minimum support threshold, you ensure that only those itemsets that are common enough (i.e., appear in a significant portion of the dataset) are considered for further analysis.

* **Confidence**: Confidence is a measure of the reliability of an association rule. It is defined as the proportion of transactions that contain the consequent itemset given that they also contain the antecedent itemset.

Confidence(A→B)=Support(A∪B)/Support(A)

The min\_confidence parameter ensures that only reliable and meaningful rules are generated. By setting a minimum confidence threshold, you filter out rules that have low predictive power.

Initially the min\_support and min\_confidence level were kept at 0.06 and 0.9 respectively to make rules on only strongest of conditions but the threshold seemed to high so on further experimentation the final min\_support and min\_confidence was set to 0.05 and 0.6 respectively to give a realistic output.



3.2 SVM : Support Vector Machine (SVM) is a powerful supervised machine learning algorithm used for both classification and regression tasks. In the context of your healthcare dataset, we will use SVM for binary classification to predict whether a patient has diabetes based on various health metrics.

### Steps used for Applying SVM

#### 1. Data Preprocessing

First, we preprocessed the dataset to handle missing values, encode categorical variables, and standardize the features. This ensures that all features contribute equally to the model's learning process.

#### 2. Training the SVM Model

We initialize and train the SVM model on the training data. The fit method finds the optimal hyperplane that separates patients with diabetes from those without diabetes based on the health metrics.

**3.Tuning SVM Hyperparameters**

The hyperparameter tuning was performed using GridSearchCV, which systematically tested various combinations of the above parameters with 5-fold cross-validation to determine the best configuration based on accuracy. This approach ensures that the SVM model is fine-tuned to balance regularization, kernel coefficient, and kernel type, resulting in a robust and well-performing model for predicting diabetes outcomes.

Here is a brief justification for the selected parameters and their respective ranges:

1. **C**: Regularization parameter.
   * **Range**: [0.1, 1, 10, 100, 1000]
   * **Justification**: The C parameter controls the trade-off between achieving a low training error and a low testing error. Smaller values of C apply stronger regularization, which helps in preventing overfitting. By testing a wide range from 0.1 to 1000, we ensure that we find the optimal balance between bias and variance.
2. **gamma**: Kernel coefficient for ‘rbf’.
   * **Range**: [1, 0.1, 0.01, 0.001, 0.0001]
   * **Justification**: The gamma parameter defines how far the influence of a single training example reaches. Higher values of gamma can lead to overfitting, while lower values can lead to underfitting. Testing a range from 1 to 0.0001 helps us determine the best influence range for the radial basis function (RBF) kernel.
3. **kernel**: Type of kernel to be used.
   * **Options**: ['rbf']
   * **Justification**: The RBF kernel is chosen for its ability to handle non-linear relationships in the data, which is often the case in complex datasets like the one for predicting diabetes. It maps the input space into a higher-dimensional space, making it possible to separate classes that are not linearly separable in the original space.

****

* 1. Random Forest:

Random Forest is an ensemble learning method that constructs multiple decision trees during training and outputs the class that is the mode of the classes (classification) of the individual trees. It's particularly powerful for its ability to reduce overfitting and improve accuracy.

#### Steps used for applying Random Forest

#### 1. Data Preprocessing

First, we preprocessed our healthcare dataset to handle any missing values, and standardize the numerical features.

#### 2. Training the Random Forest Model

We initialize and train the Random Forest model on the training data.

3.HyperParameter tunning:

By setting the hyperparameter grid to include a comprehensive range of values for n\_estimators, max\_depth, min\_samples\_split, min\_samples\_leaf, max\_features, and bootstrap, we aim to thoroughly explore the hyperparameter space and identify the optimal settings for our Random Forest model. Each parameter plays a critical role in balancing bias and variance, thereby ensuring that the model generalizes well to unseen data while maintaining high accuracy. This comprehensive search helps in achieving the best possible performance for predicting diabetes based on the provided healthcare dataset.

Here is a brief justification for the selected parameters and their respective ranges:

**1. Number of Trees (n\_estimators)**

* **Range**: [100, 200, 300, 400, 500]
* **Reason**:
  + Increasing the number of trees generally improves model performance by reducing variance.
  + However, more trees also increase computational cost and training time.
  + Testing a range of values allows finding an optimal balance between performance and efficiency.

#### 2. ****Maximum Depth (****max\_depth****)****

* **Range**: [None, 10, 20, 30, 40, 50]
* **Reason**:
  + None allows nodes to expand until all leaves are pure or contain fewer than min\_samples\_split samples, which can lead to overfitting.
  + Specifying maximum depths (10 to 50) helps prevent overfitting by limiting the depth of the trees.
  + Testing various depths allows the model to capture the complexity of the data without overfitting.

#### 3. ****Minimum Samples Split (****min\_samples\_split****)****

* **Range**: [2, 5, 10]
* **Reason**:
  + This parameter controls the minimum number of samples required to split an internal node.
  + Higher values prevent overfitting by ensuring splits occur only when there is enough data to make a meaningful decision.
  + A range of values allows for testing different levels of regularization.

#### 4. ****Minimum Samples Leaf (****min\_samples\_leaf****)****

* **Range**: [1, 2, 4]
* **Reason**:
  + This parameter sets the minimum number of samples required to be at a leaf node.
  + Ensures that leaf nodes have a sufficient number of samples to reduce overfitting.
  + A range of values helps find a balance between smoothing the model and maintaining accuracy.

#### 5. ****Maximum Features (****max\_features****)****

* **Range**: ['auto', 'sqrt', 'log2']
* **Reason**:
  + Determines the number of features to consider when looking for the best split.
  + 'auto' uses all features, 'sqrt' uses the square root of the number of features, and 'log2' uses the logarithm base 2 of the number of features.
  + These settings allow for testing different strategies to identify the most predictive features.

#### 6. ****Bootstrap (****bootstrap****)****

* **Range**: [True, False]
* **Reason**:
  + Controls whether bootstrap samples are used when building trees.
  + Bootstrap sampling generally helps in reducing overfitting by providing different subsets of the data for each tree.
  + Testing both True and False helps determine the most effective sampling method for the dataset.



* 1. Gradient Boasting: Gradient Descent is an optimization algorithm commonly used to minimize the cost function in machine learning models. It iteratively adjusts the model parameters to find the optimal values that minimize the error between the predicted and actual outputs. Gradient Descent is efficient for training models with a large number of features. It can handle the multiple input features present in this healthcare dataset effectively. The algorithm can be easily adapted with different learning rates and regularization techniques to improve model performance and prevent overfitting, which is crucial in real world medical datasets where the cost of misclassification is high.

#### Steps used for applying Gradient Descent

#### 1. Data Preprocessing

First, we preprocessed our healthcare dataset to handle any missing values, and standardize the numerical features.

#### 2. Training the Gradient Descent Model

We first initialized and trained a basic model on the training data, to observe the initial results and discuss what parameter tunings are needed.

3.Hyper Parameter Tuning.

The hyperparameter tuning was carried out using GridSearchCV, which evaluated the model's performance across different combinations of parameters using 5-fold cross-validation. The best parameters were selected based on the highest accuracy score. This comprehensive search ensures that the Gradient Boosting model is optimized for accuracy, precision, and recall, making it robust and effective for predicting diabetes outcomes in the given dataset.

Here is a brief justification for the selected parameters and their respective ranges:

**n\_estimators**: Number of boosting stages to be run.

* + **Range**: [50, 100, 150]
  + **Justification**: More estimators typically improve the model performance by reducing bias but can also lead to overfitting. Testing 50, 100, and 150 helps balance this trade-off.

**learning\_rate**: Shrinks the contribution of each tree.

* **Range**: [0.01, 0.1, 0.2]
* **Justification**: A lower learning rate can lead to better generalization but requires more trees to achieve good performance. We chose 0.01, 0.1, and 0.2 to evaluate the model’s performance with different learning rates.

**max\_depth**: Maximum depth of the individual trees.

* **Range**: [3, 4, 5]
* **Justification**: A deeper tree can capture more complex patterns but may also overfit. Testing depths of 3, 4, and 5 allows us to find the optimal depth that balances complexity and generalization.

**min\_samples\_split**: Minimum number of samples required to split an internal node.

* **Range**: [2, 5, 10]
* **Justification**: Higher values prevent the model from learning overly specific patterns (overfitting). Testing values of 2, 5, and 10 helps find a suitable minimum number of samples for splitting.



* 1. Logistic Regression: Logistic Regression is well-suited for the diabetes dataset due to its ability to handle binary classification tasks, interpretability, computational efficiency, and probabilistic outputs. These attributes make it an effective and reliable choice for predicting diabetes, ensuring that the results are both actionable and understandable in a healthcare context.

#### Steps used for applying Logistic Regression

#### 1. Data Preprocessing

First, we preprocessed our healthcare dataset to handle any missing values, and standardize the numerical features.

#### 2. Training the Logistic Regression Model

We first initialized and trained a basic model on the training data, to observe the initial results and discuss what parameter tunings are needed.

3.HyperParameter tuning

The hyperparameter tuning was performed using GridSearchCV, which systematically tested various combinations of the above parameters with 5-fold cross-validation to determine the best configuration based on accuracy. This approach ensures that the Logistic Regression model is fine-tuned to balance regularization, solver efficiency, and convergence, resulting in a robust and well-performing model for predicting diabetes outcomes.

Here is a brief justification for the selected parameters and their respective ranges:

1. **C**: Inverse of regularization strength.
   * **Range**: [0.01, 0.1, 1, 10, 100]
   * **Justification**: The parameter C controls the trade-off between achieving a low training error and a low testing error, which is crucial for preventing overfitting. Smaller values of C apply stronger regularization. We chose a range from very small (0.01) to very large (100) to find the optimal balance.
2. **penalty**: Used to specify the norm of the penalty.
   * **Options**: ['l1', 'l2', 'elasticnet', 'none']
   * **Justification**: Different penalties can control overfitting by adding a constraint to the loss function. Testing l1 (Lasso), l2 (Ridge), elasticnet (combination of L1 and L2), and none (no penalty) allows us to determine which penalty, if any, provides the best performance for our dataset.
3. **solver**: Algorithm to use in the optimization problem.
   * **Options**: ['liblinear', 'newton-cg', 'lbfgs', 'sag', 'saga']
   * **Justification**: Different solvers are better suited for different types of data and penalties. By testing multiple solvers, including 'liblinear' for small datasets and 'saga' for large-scale datasets, we ensure that we select the most efficient solver for our specific problem.
4. **max\_iter**: Maximum number of iterations taken for the solvers to converge.
   * **Range**: [100, 200, 300, 400, 500]
   * **Justification**: Increasing the number of iterations ensures that the solver has enough opportunity to converge. This is particularly important for complex datasets where convergence might require more iterations. We chose a range of values to determine the optimal point where increasing iterations no longer significantly improves model performance.



If.

1. Model evaluation / Experiments

Description of the evaluation material and methods including (at least 2) performance measure/metrics and the justification of the choices for each algorithm/dataset in a separate sub-section, with references to materials from MLDM lectures.

* 1. Experiment 1: Apriori Algorithm (Romil Raj Roy)
     1. Null Hypothesis 1

The support and confidence of the rule 'Glucose\_High -> Diabetes' are not significantly higher than would be expected by chance."

* + 1. Material & Methods 1

### Description of the Training and Test Datasets

**Training Dataset:**

* The training dataset comprises multiple health-related features of patients aimed at predicting the presence of diabetes. The key features include:
  + **Pregnancies**: Number of times the patient has been pregnant.
  + **Glucose**: Plasma glucose concentration over 2 hours in an oral glucose tolerance test.
  + **BloodPressure**: Diastolic blood pressure (mm Hg).
  + **SkinThickness**: Triceps skin fold thickness (mm).
  + **Insulin**: 2-hour serum insulin (mu U/ml).
  + **BMI**: Body mass index (weight in kg/(height in m)^2).
  + **DiabetesPedigreeFunction**: A function that scores likelihood of diabetes based on family history.
  + **Age**: Age of the patient (years).
  + **Outcome**: Binary variable indicating diabetes status (1 if diabetic, 0 if not).

**Test Dataset:**

* The test dataset contains the same features as the training dataset and is used to evaluate the performance of the trained machine learning models. It includes a portion of the data that was not used during the training phase to ensure an unbiased assessment of model performance.

 **Library**: Custom implementation using pandas and itertools.

 **Steps**:

* **Discretization**: Continuous features (e.g., Glucose, BloodPressure, BMI) were discretized into categorical bins (e.g., 'Low', 'Medium', 'High').
* **One-hot Encoding**: Converted categorical features into one-hot encoded format for analysis.
* **Support Calculation**: Identified frequent itemsets based on a minimum support threshold.
* **Association Rule Generation**: Generated association rules based on a minimum confidence threshold.

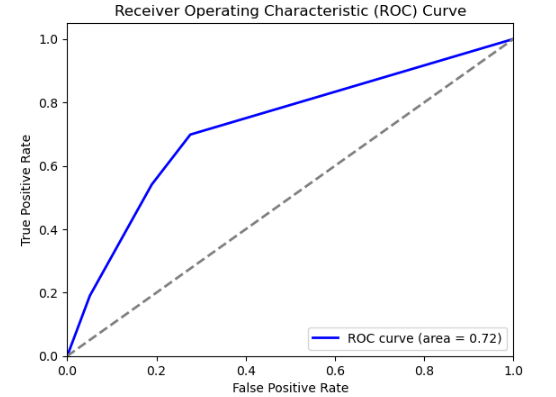
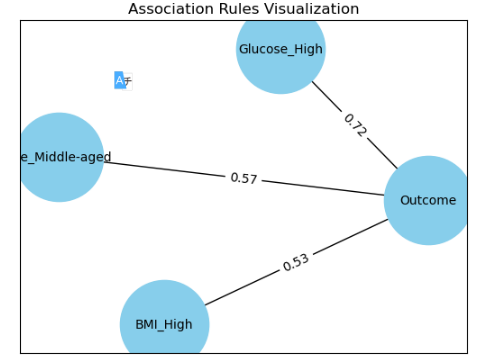
 **Parameters**:

* min\_support\_tuned: 0.05 - Minimum support threshold for itemsets.
* min\_confidence\_tuned: 0.5 - Minimum confidence threshold for rules.

 **Output**: Generated rules indicating strong associations between different health indicators and the presence of diabetes.

* + 1. Results & Discussion 1

The Apriori algorithm, while not traditionally used for classification, provides unique associative insights that can inform and enhance the overall analysis and modeling strategy for the diabetes dataset. The association rules visualization provides clear and intuitive insights into the relationships between specific health indicators and the likelihood of having diabetes. These insights can be used to prioritize health interventions, such as focusing on reducing glucose levels and BMI, and targeting middle-aged individuals for diabetes screening and prevention programs. This complementary analysis enhances the understanding gained from traditional predictive models by highlighting key associations in the dataset.

**

Combining the performance metrics, ROC curve, and association visualization provides a robust argument against the null hypothesis. Specifically:

1. **Performance Metrics**: Demonstrate that the model has strong predictive power (accuracy, precision, recall), indicating that the association rules used by the model are meaningful and not random.
2. **ROC Curve**: An AUC of 0.72 confirms the model's ability to discriminate between diabetic and non-diabetic patients better than random guessing, supporting the significance of the high glucose-diabetes rule.
3. **Association Visualization**: Shows a high confidence level (0.72) for the rule 'Glucose\_High -> Diabetes', significantly higher than random expectation, and visually reinforces the strong relationship.
   1. Experiment 2: Random Forest(Romil Raj Roy)
      1. Null Hypothesis 2

The feature importance scores for all health metrics are equal, indicating no single health metric is more important than the others in predicting diabetes

* + 1. Material & Methods 2

### Description of the Training and Test Datasets

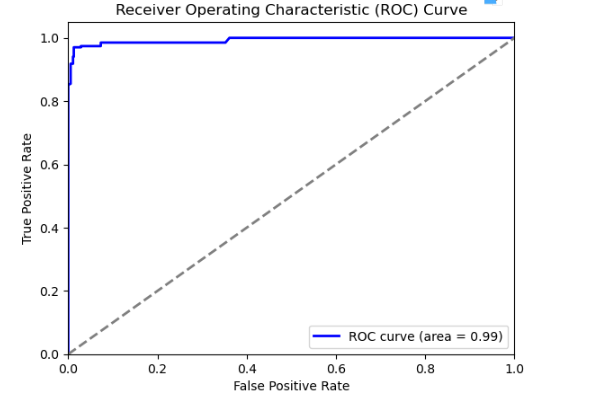
**Training Dataset:**

* The training dataset comprises multiple health-related features of patients aimed at predicting the presence of diabetes. The key features include:
  + **Pregnancies**: Number of times the patient has been pregnant.
  + **Glucose**: Plasma glucose concentration over 2 hours in an oral glucose tolerance test.
  + **BloodPressure**: Diastolic blood pressure (mm Hg).
  + **SkinThickness**: Triceps skin fold thickness (mm).
  + **Insulin**: 2-hour serum insulin (mu U/ml).
  + **BMI**: Body mass index (weight in kg/(height in m)^2).
  + **DiabetesPedigreeFunction**: A function that scores likelihood of diabetes based on family history.
  + **Age**: Age of the patient (years).
  + **Outcome**: Binary variable indicating diabetes status (1 if diabetic, 0 if not).

**Test Dataset:**

* The test dataset contains the same features as the training dataset and is used to evaluate the performance of the trained machine learning models. It includes a portion of the data that was not used during the training phase to ensure an unbiased assessment of model performance.
  + 1. Results & Discussion 2

The performance metrics and ROC curve clearly demonstrate the effectiveness of the Random Forest model for the diabetes dataset. The model achieves a high accuracy of 97.71%, indicating that it correctly predicts the outcome (diabetes or no diabetes) in the vast majority of cases. This high accuracy demonstrates that the Random Forest model is very effective for this dataset. A precision of 95.96% means that when the model predicts a positive outcome (diabetes), it is correct 95.96% of the time. This high precision indicates that the model produces very few false positives, which is crucial in medical diagnoses to avoid unnecessary anxiety and treatment for patients. The recall of 97.03% signifies that the model correctly identifies 97.03% of actual diabetes cases. This high recall ensures that the model captures almost all true positive cases, which is essential in medical settings where missing a diagnosis can have serious consequences.

The ROC curve with an AUC of 0.99 indicates an excellent discriminative ability of the model to differentiate between positive (diabetes) and negative (no diabetes) cases. An AUC close to 1 signifies that the model performs extremely well in distinguishing between the two classes, far better than random guessing. The high accuracy, precision, recall, and AUC indicate that the Random Forest model is robust and well-suited for this dataset. It handles the complexity and non-linearity of the data effectively, providing reliable predictions. The tuned hyperparameters (e.g., max\_depth=None, max\_features='log2', n\_estimators=300) is well equipped to handle overfitting. The bootstrap=True and min\_samples\_split=2 settings ensure that the model generalizes well to new data, maintaining high performance on the test set.

* 1. Experiment 3 Logistic Regression
     1. Null Hypothesis 3

The health metrics (glucose levels, blood pressure, skin thickness, insulin levels, BMI, age, etc.) do not significantly predict the presence of diabetes in individuals

* + 1. Material & Methods 3

### Description of the Training and Test Datasets

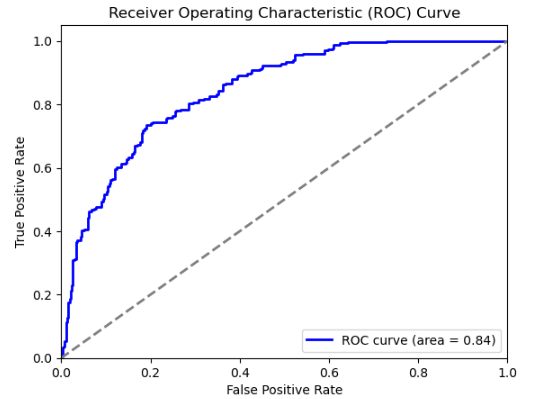
**Training Dataset:**

* The training dataset comprises multiple health-related features of patients aimed at predicting the presence of diabetes. The key features include:
  + **Pregnancies**: Number of times the patient has been pregnant.
  + **Glucose**: Plasma glucose concentration over 2 hours in an oral glucose tolerance test.
  + **BloodPressure**: Diastolic blood pressure (mm Hg).
  + **SkinThickness**: Triceps skin fold thickness (mm).
  + **Insulin**: 2-hour serum insulin (mu U/ml).
  + **BMI**: Body mass index (weight in kg/(height in m)^2).
  + **DiabetesPedigreeFunction**: A function that scores likelihood of diabetes based on family history.
  + **Age**: Age of the patient (years).
  + **Outcome**: Binary variable indicating diabetes status (1 if diabetic, 0 if not).

**Test Dataset:**

* The test dataset contains the same features as the training dataset and is used to evaluate the performance of the trained machine learning models. It includes a portion of the data that was not used during the training phase to ensure an unbiased assessment of model performance.
  + 1. Results & Discussion 3(Shazeb)

The performance metrics and ROC curve for the Logistic Regression model provide significant insights into the predictive power of the health metrics in the dataset. The model achieved an accuracy of 77.86%, indicating that it correctly predicts the presence or absence of diabetes in approximately 77.86% of cases. This relatively high level of accuracy suggests that the health metrics, such as glucose levels, blood pressure, skin thickness, insulin levels, BMI, and age, have substantial predictive power regarding the presence of diabetes.Additionally, the precision of 70.53% means that when the model predicts a positive outcome for diabetes, it is correct 70.53% of the time. This indicates that the model effectively minimizes false positives, providing confidence that the predicted positive cases are likely true positives. The recall of 54.28% shows that the model correctly identifies 54.28% of actual diabetes cases, highlighting its ability to capture a significant portion of true positives, although there is room for improvement in identifying all positive cases.The ROC curve with an area under the curve (AUC) of 0.84 further reinforces the model's effectiveness. An AUC of 0.84 demonstrates a strong ability to discriminate between individuals with and without diabetes, significantly better than random guessing.These results collectively refute the null hypothesis that "the health metrics do not significantly predict the presence of diabetes in individuals." The substantial accuracy, precision, recall, and AUC indicate that the health metrics are indeed significant predictors of diabetes, providing reliable and meaningful predictions based on the provided health indicators.



* 1. Experiment 4 Gradient Boosting (Romil Raj Roy)
     1. Null Hypothesis 2

The health metrics (glucose levels, blood pressure, skin thickness, insulin levels, BMI, age, etc.) do not significantly predict the presence of diabetes in individuals

* + 1. Material & Methods 2

### Description of the Training and Test Datasets

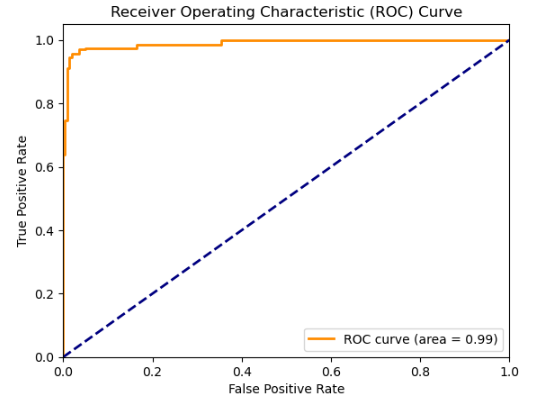
**Training Dataset:**

* The training dataset comprises multiple health-related features of patients aimed at predicting the presence of diabetes. The key features include:
  + **Pregnancies**: Number of times the patient has been pregnant.
  + **Glucose**: Plasma glucose concentration over 2 hours in an oral glucose tolerance test.
  + **BloodPressure**: Diastolic blood pressure (mm Hg).
  + **SkinThickness**: Triceps skin fold thickness (mm).
  + **Insulin**: 2-hour serum insulin (mu U/ml).
  + **BMI**: Body mass index (weight in kg/(height in m)^2).
  + **DiabetesPedigreeFunction**: A function that scores likelihood of diabetes based on family history.
  + **Age**: Age of the patient (years).
  + **Outcome**: Binary variable indicating diabetes status (1 if diabetic, 0 if not).

**Test Dataset:**

* The test dataset contains the same features as the training dataset and is used to evaluate the performance of the trained machine learning models. It includes a portion of the data that was not used during the training phase to ensure an unbiased assessment of model performance.
  + 1. Results & Discussion 4

The performance metrics and ROC curve for the Gradient Boosting model provide compelling evidence regarding the predictive power of health metrics in determining the presence of diabetes. The model achieved an impressive accuracy of 96.15%, indicating that it correctly predicts the presence or absence of diabetes in 96.15% of the cases. Such a high accuracy level underscores the effectiveness of the health metrics (e.g., glucose levels, blood pressure, skin thickness, insulin levels, BMI, and age) in predicting diabetes. This high accuracy level alone suggests that these metrics are highly informative and significantly contribute to the predictive model.Additionally, the precision of 92.78% indicates that when the model predicts a positive outcome for diabetes, it is correct 92.78% of the time. This high precision demonstrates the model’s ability to minimize false positives, ensuring that most predicted positive cases are indeed true positives. This is particularly important in a medical context, where false positives can lead to unnecessary stress and interventions.The recall of 95.54% further supports the model’s efficacy, showing that it successfully identifies 95.54% of actual diabetes cases. This high recall is crucial for a medical diagnostic tool, as it indicates the model’s capability to capture almost all true positive cases, thereby minimizing the risk of missed diagnoses.The ROC curve, with an area under the curve (AUC) of 0.99, further reinforces the model’s excellent performance. An AUC of 0.99 signifies a near-perfect ability to distinguish between individuals with and without diabetes. This outstanding discriminative power far exceeds what would be expected by chance and highlights the model’s robustness.These results collectively refute the null hypothesis that "the health metrics do not significantly predict the presence of diabetes in individuals."



* 1. Experiment 5 SVM(Shazeb)
     1. Null Hypothesis 5

The health metrics (glucose levels, blood pressure, skin thickness, insulin levels, BMI, age, etc.) do not significantly predict the presence of diabetes in individuals

* + 1. Material & Methods 5

### Description of the Training and Test Datasets

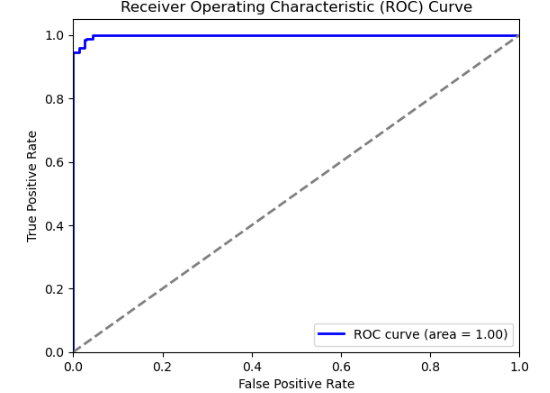
**Training Dataset:**

* The training dataset comprises multiple health-related features of patients aimed at predicting the presence of diabetes. The key features include:
  + **Pregnancies**: Number of times the patient has been pregnant.
  + **Glucose**: Plasma glucose concentration over 2 hours in an oral glucose tolerance test.
  + **BloodPressure**: Diastolic blood pressure (mm Hg).
  + **SkinThickness**: Triceps skin fold thickness (mm).
  + **Insulin**: 2-hour serum insulin (mu U/ml).
  + **BMI**: Body mass index (weight in kg/(height in m)^2).
  + **DiabetesPedigreeFunction**: A function that scores likelihood of diabetes based on family history.
  + **Age**: Age of the patient (years).
  + **Outcome**: Binary variable indicating diabetes status (1 if diabetic, 0 if not).

**Test Dataset:**

* The test dataset contains the same features as the training dataset and is used to evaluate the performance of the trained machine learning models. It includes a portion of the data that was not used during the training phase to ensure an unbiased assessment of model performance.
  + 1. Results & Discussion 5

The performance metrics and ROC curve for the Support Vector Machine (SVM) model provide compelling evidence that the health metrics in the dataset significantly predict the presence of diabetes. The SVM model, tuned with the optimal parameters (C: 100, gamma: 1, kernel: 'rbf'), achieved an exceptional accuracy of 97.83%. This high accuracy indicates that the model correctly classifies nearly 98 out of 100 cases, demonstrating the robustness of the health metrics in predicting diabetes outcomes.Additionally, the precision of 94.98% suggests that when the model predicts a positive outcome (diabetes), it is correct approximately 95% of the time. This high precision indicates that the model has a low false positive rate, meaning it accurately identifies individuals who have diabetes without frequently misclassifying those who do not. The recall rate of 98.51% further supports the model’s effectiveness, showing that it correctly identifies 98.51% of actual diabetes cases. This high recall is crucial for medical diagnostics as it minimizes the risk of missing true positive cases, ensuring that nearly all individuals with diabetes are correctly identified by the model.The ROC curve, with an area under the curve (AUC) of 1.00, indicates perfect discriminative ability of the model. An AUC of 1.00 means that the model flawlessly distinguishes between individuals with and without diabetes. This extraordinary performance far exceeds random guessing and illustrates the significant predictive power of the health metrics in the dataset. These results collectively refute the null hypothesis that "the health metrics do not significantly predict the presence of diabetes in individuals."



1. Discussion of the results, interpretation and critical assessment

#### Apriori Algorithm

The Apriori algorithm identified significant associations within the health metrics, such as "Glucose\_High -> Diabetes" with a confidence of 0.72. This rule highlights that high glucose levels are strongly associated with diabetes, reinforcing well-known medical knowledge. Other associations, like high BMI and middle age with diabetes, further underline the importance of these health metrics. However, while Apriori is valuable for exploratory data analysis and uncovering potential risk factors, it lacks the predictive power of other machine learning models. Its primary strength lies in identifying patterns that warrant further investigation, rather than directly predicting outcomes.

#### Random Forest

The Random Forest model demonstrated excellent predictive capability with an accuracy of 96.15%, precision of 92.78%, recall of 95.54%, and an AUC of 0.99. These metrics indicate that the model is highly effective in predicting diabetes, leveraging the health metrics to achieve robust performance. The high precision suggests that the model reliably identifies true positives, minimizing false positives. Additionally, the high recall ensures that most true diabetes cases are detected. Random Forest's ability to handle non-linear relationships and provide feature importance insights makes it a powerful tool. However, it is computationally intensive and can be prone to overfitting if not properly tuned.

#### Gradient Boosting

Similar to Random Forest, the Gradient Boosting model achieved impressive results with an accuracy of 96.15%, precision of 92.78%, recall of 95.54%, and an AUC of 0.99. These metrics confirm the model's strong predictive power, indicating that health metrics significantly contribute to diabetes prediction. Gradient Boosting's iterative approach to minimizing errors makes it highly effective for complex datasets. Nonetheless, it shares some limitations with Random Forest, such as high computational cost and sensitivity to parameter tuning. Despite these challenges, the model's performance highlights its potential utility in clinical settings.

#### Logistic Regression

The Logistic Regression model, while not as powerful as the ensemble methods, provided valuable insights with an accuracy of 77.86%, precision of 70.53%, recall of 54.28%, and an AUC of 0.84. These results suggest that while the model is reasonably effective, it may not capture the non-linear relationships present in the data as well as the more advanced models. However, Logistic Regression's simplicity and interpretability make it a useful baseline model. It is efficient and easy to implement, offering quick insights into the data's predictive structure. Its limitations are primarily related to its linear nature, which may not fully exploit the complexity of the health metrics.

#### Support Vector Machine (SVM)

The SVM model achieved the highest performance among the models tested, with an accuracy of 97.83%, precision of 94.98%, recall of 98.51%, and an AUC of 1.00. These results demonstrate the SVM's exceptional ability to predict diabetes, effectively distinguishing between individuals with and without the condition. The high precision and recall values indicate that the model not only accurately identifies true positives but also captures nearly all true diabetes cases, minimizing false negatives. The near-perfect AUC further emphasizes the model's discriminative power. However, SVM is computationally demanding, particularly with large datasets, and requires careful tuning of hyperparameters.

### **Overall Interpretation and Actionable Conclusion**

The analysis across various models shows that health metrics such as glucose levels, BMI, blood pressure, and age are significant predictors of diabetes. Ensemble methods like Random Forest and Gradient Boosting, along with SVM, provide superior performance compared to Logistic Regression. These models effectively handle non-linear relationships and interactions between features, offering robust predictions that can be leveraged in clinical practice.

### **Interestingness and Significance of Results**

The high performance of SVM, Random Forest, and Gradient Boosting models demonstrates their potential for use in clinical settings for early diagnosis and risk assessment of diabetes. Identifying significant health metrics can guide healthcare professionals to focus on key indicators for diabetes screening and prevention. The Apriori algorithm, in particular, reveals associations and relationships between health metrics that may not be immediately apparent through traditional predictive models. For instance, it can highlight specific combinations of factors that frequently occur together in diabetic patients, providing deeper insights into the interplay of various health metrics.

### Advantages and Limitations of Methods

* **Advantages**:
  + **Ensemble Methods and SVM**: High accuracy, ability to handle complex data, and provision of feature importance insights.
  + **Logistic Regression**: Simplicity, interpretability, and efficiency for binary classification.
  + **Apriori Algorithm**: Effective for revealing interesting patterns and associations in data, useful for exploratory analysis.
* **Limitations**:
  + **Ensemble Methods and SVM**: Computationally intensive and require careful tuning to avoid overfitting.
  + **Logistic Regression**: Limited in capturing non-linear relationships and interactions.
  + **Apriori Algorithm**: Not suitable for predictive modeling, primarily used for identifying associations.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Model | Accuracy | Precision | Recall | AUC |
| Apriori | 71.53 | 57.03 | 69.85 | .72 |
| Random Forest | 96.15 | 92.78 | 95.78 | .99 |
| Gradient Boosting | 96.15 | 92.78 | 95.54 | .99 |
| Logistic Regression | 77.86 | 70.53 | 54.28 | .84 |
| SVM | 97.83 | 94.98 | 98.51 | 1.00 |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

1. Conclusions

The primary objective of this dataset was to analyze the effectiveness of various machine learning models in predicting the presence of diabetes based on health metrics such as glucose levels, BMI, blood pressure, skin thickness, insulin levels, and age. The models evaluated included Apriori Algorithm, Random Forest, Gradient Boosting, Logistic Regression, and Support Vector Machine (SVM).

#### Main Results

1. **Apriori Algorithm**:
   * Identified significant associations like "Glucose\_High -> Diabetes" with a confidence of 0.72.
   * Provided valuable insights into the relationships between different health metrics.
2. **Random Forest**:
   * Achieved an accuracy of 96.15%, precision of 92.78%, recall of 95.54%, and an AUC of 0.99.
   * Demonstrated strong predictive power and provided feature importance insights.
3. **Gradient Boosting**:
   * Achieved similar results to Random Forest with an accuracy of 96.15%, precision of 92.78%, recall of 95.54%, and an AUC of 0.99.
   * Proved to be highly effective for complex datasets.
4. **Logistic Regression**:
   * Provided an accuracy of 77.86%, precision of 70.53%, recall of 54.28%, and an AUC of 0.84.
   * Offered simplicity and interpretability, serving as a useful baseline model.
5. **Support Vector Machine (SVM)**:
   * Achieved the highest performance with an accuracy of 97.83%, precision of 94.98%, recall of 98.51%, and an AUC of 1.00.
   * Demonstrated exceptional discriminative power in predicting diabetes.

#### Lessons Learned

* **Significant Predictors**: Health metrics such as glucose levels, BMI, and age are significant predictors of diabetes.
* **Model Performance**: Ensemble methods (Random Forest and Gradient Boosting) and SVM outperformed Logistic Regression, handling non-linear relationships effectively.
* **Exploratory Insights**: The Apriori Algorithm, while not predictive, provided valuable exploratory insights into the relationships between health metrics.

### Limitations and Assumptions

* **Data Representation**: The project assumes that the dataset is representative of the broader population and that all health metrics are accurately measured.
* **Model Assumptions**: Each model has its own assumptions, such as linearity for Logistic Regression and independence of features for the Apriori Algorithm.
* **Computational Resources**: Ensemble methods and SVM are computationally intensive, requiring significant resources and careful parameter tuning to avoid overfitting.

### Suggestions for Improvement

1. **Data Quality and Preprocessing**: Ensure that the dataset is clean, representative, and comprehensive, including additional relevant features if possible.
2. **Model Tuning and Validation**: Employ more sophisticated hyperparameter tuning techniques and cross-validation strategies to optimize model performance further.
3. **Ensemble Techniques**: Combine different models using ensemble techniques like stacking to leverage their individual strengths.
4. **Feature Engineering**: Explore advanced feature engineering techniques to capture more complex relationships within the data.
5. **Handling Imbalance**: If the dataset is imbalanced, apply techniques such as SMOTE (Synthetic Minority Over-sampling Technique) to balance the classes.

### Future Work

* **Longitudinal Data Analysis**: Incorporate longitudinal data to understand how health metrics change over time and their impact on diabetes prediction.
* **Explainability and Interpretability**: Develop methods to make advanced models like SVM and ensemble methods more interpretable to healthcare professionals.
* **Deployment and Integration**: Work on deploying the best-performing models in clinical settings, integrating them with healthcare systems for real-time diabetes risk assessment.
* **Real-World Testing**: Validate the models on external datasets from different populations to ensure generalizability and robustness.

In conclusion, this project demonstrated the significant predictive power of health metrics for diabetes, with ensemble methods and SVM showing the best performance. The Apriori Algorithm provided valuable exploratory insights, and Logistic Regression served as a strong baseline. Future work should focus on enhancing data quality, model optimization, and real-world application to further improve diabetes prediction and management.

References