Northeastern University
College *of* Engineering

Diabetes prediction models
Project report

DATA MINING
IE 7275
SPRING 2024

Laawanyaa Sai Thota
NU ID: 002208176

What is diabetes?

Diabetes is a chronic (long-lasting) health condition that affects how your body turns food into energy.

Your body breaks down most of the food you eat into sugar (glucose) and releases it into your bloodstream. When your blood sugar goes up, it signals your pancreas to release insulin. Insulin acts like a key to let the blood sugar into your body's cells for use as energy.

With diabetes, your body doesn't make enough insulin or can't use it as well as it should. When there isn't enough insulin or cells stop responding to insulin, too much blood sugar stays in your bloodstream. Over time, that can cause serious health problems, such as heart disease, vision loss, and kidney disease.

Problem definition

There isn't a cure yet for diabetes, but detecting it in early stage and taking proper medication and diet helps in keeping it under control. The intention of this project is to build supervised models like Logistic regression, K nearest neighbors, Random Forest and Decision trees and identifying the best algorithm which can predict diabetes for a patient with high accuracy.

Data source

The dataset for this project is obtained from centers for disease control and prevention (CDC) website. National center for health statistics(NCHS), as a part of National health and nutrition examination survey has submitted the dataset to CDC which can be assessed here

Data description

This dataset consists of 6643 records which is obtained by combining multiple datasets from the data source (lab data, diet data, questionnaire data). We have 27 attributes in the dataset and one target variable (diabetes). Below are the variables in the dataset.

S.No	Variable Name	Variable Type	Description
1	Gender	Categorical	Gender of the participant
2	Age	Numerical	Age of the participant in years
3	Age Months	Numerical	Age of the participant in months
4	Race	Categorical	Race of the participant
5	Veteran Status	Categorical	Veteran status of the participant
6	Marital Status	Categorical	Marital status of the participant
7	Pregnancy	Categorical	Pregnancy status of the participant
8	Ratio of Income to Poverty Guidelines	Numerical	Ratio of income to poverty guidelines of the participant
9	Weight	Numerical	Weight of the participant in pounds
10	Height	Numerical	Height of the participant in cms
11	BMI	Numerical	Body Mass Index of the participant
12	Pulse	Numerical	Pulse rate of the participant in beats per minute
13	Systolic Pressure	Numerical	Systolic blood pressure of the participant in mmHg
14	Diastolic Pressure	Numerical	Diastolic blood pressure of the participant in mmHg
15	Total Cholesterol	Numerical	Total cholesterol level of the participant in mg/dl
16	Glychohemoglobin	Numerical	Glycohemoglobin level of the participant
17	Albumin Creatinine level	Numerical	Albumin creatinine level of the participant
18	Water Level	Numerical	Water level of the participant
19	Insulin Level	Numerical	Insulin level of the participant in microunits/ml
20	Triglycerides	Numerical	Triglycerides level of the participant in mg/dl
21	Glucose	Numerical	Glucose level of the participant in mg/dl
22	Physical Activity	Categorical	Physical activity status of the participant
23	Fat Intake	Numerical	Fat intake of the participant in grams per day
24	Energy Intake	Numerical	Energy intake of the participant in kcal per day
25	Protein Intake	Numerical	Protein intake of the participant in grams per day
26	Carbs Intake	Numerical	Carbohydrate intake of the participant in grams per day
27	Alcohol Intake	Numerical	Alcohol intake of the participant in grams per day

Understanding the data

The obtained dataset consists of 6 categorical variables and 21 numerical variables. For instance, gender is given as a numerical variable where '1' represents male and '2' represents female. Diabetes is a number column where '1' indicates diabetic and '0' indicates non-diabetic. More details about the values in columns can be found in the documentation provided here.

Data pre-processing

As per the study done by Journal of diabetes investigation (JDI) glycated hemoglobin $\geq$ 6.5% indicates that the person has diabetes. The research paper can be accessed here. I have derived the diabetes attribute based on the above-mentioned study. If a person has glycated hemoglobin value $\geq$ 6.5% I have categorized them as '1' (Diabetic) and for those values which are less than 6.5%, I've categorized them as '0' (Non-Diabetic).

The dataset had columns with missing values (null values). Pregnancy attribute had around 80% null values which is mainly because pregnancy is observed only in females of ages 20-44. Rest all values were null. So, the null values are imputed with '2' (Not pregnant). Similarly, Marital status was null for age groups below 19 years. So, the null values are replaced by '5' (Never married). After these steps, the percentage of missing values in each column was calculated. Below are the values.

Gender	0.000000
Age	0.000000
Age Months	100.000000
Race	0.000000
Veteran Status	12.343821
Marital Status	0.000000
Pregnancy	0.000000
Ratio of Income to Poverty Guidelines	7.617040
Weight	0.948367
Height	1.008580
BMI	1.129008
Pulse	3.206383
Systolic Pressure	5.012795
Diastolic Pressure	5.012795
Total Cholestrol(mg/dl)	1.098901
Glychohemoglobin	0.000000
Albumin Creatinine level	1.580611
Water Level	81.695017
Insulin Level	53.469818
Triglycerides	52.671986
Glucose	64.729791
Physical Activity	0.331176
Fat Intake(gms in a day)	8.414873
Energy Intake(kcal in a day)	8.414873
Protein Intake(gms in a day)	8.414873
Carbs Intake(gms in a day)	8.414873
Alcohol Intake(gms in a day)	8.414873
Diabetes	0.000000

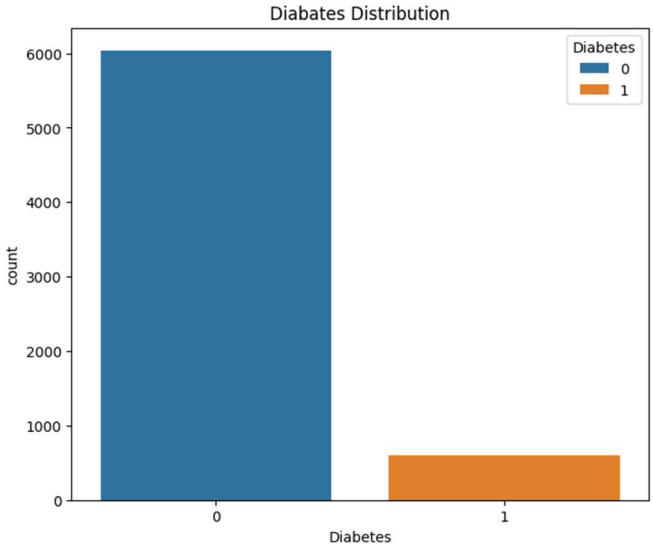
A threshold of 75% has been set, above which the whole column has been dropped from the analysis. After removing the unnecessary columns, the missing data in other columns has

been replaced by the mean of that particular age group people for all the columns. As a result, 6643 records and 26 attributes are left for the analysis.

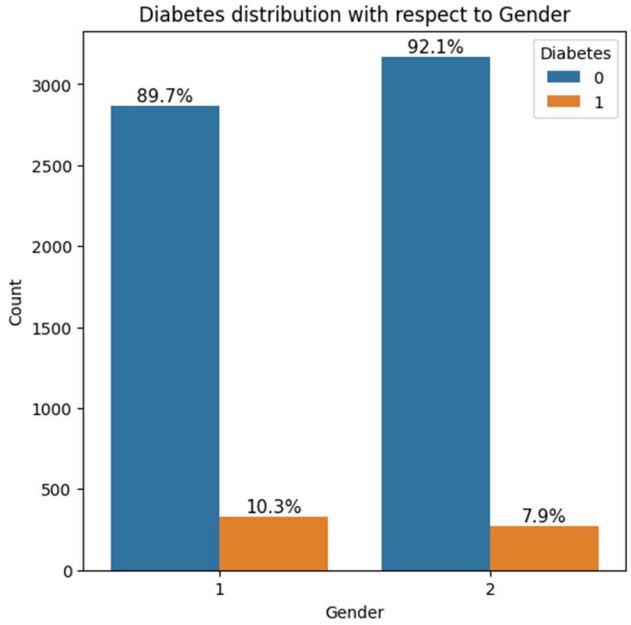
Exploratory data analysis

During the exploratory data analysis, the categorical variables and numeric variables were analyzed separately. Below are the findings from the analysis.

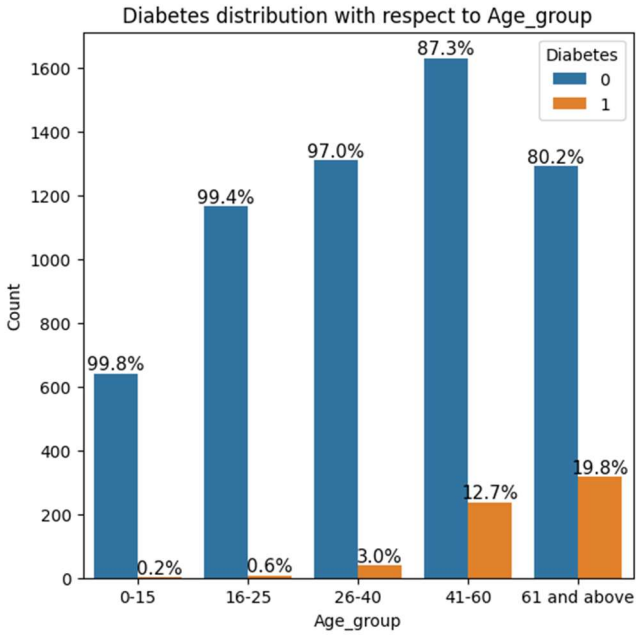
In the dataset there are 604 cases where the person is diabetic and 6039 cases where the person is not diabetic, indicating that we are dealing with an imbalanced dataset.



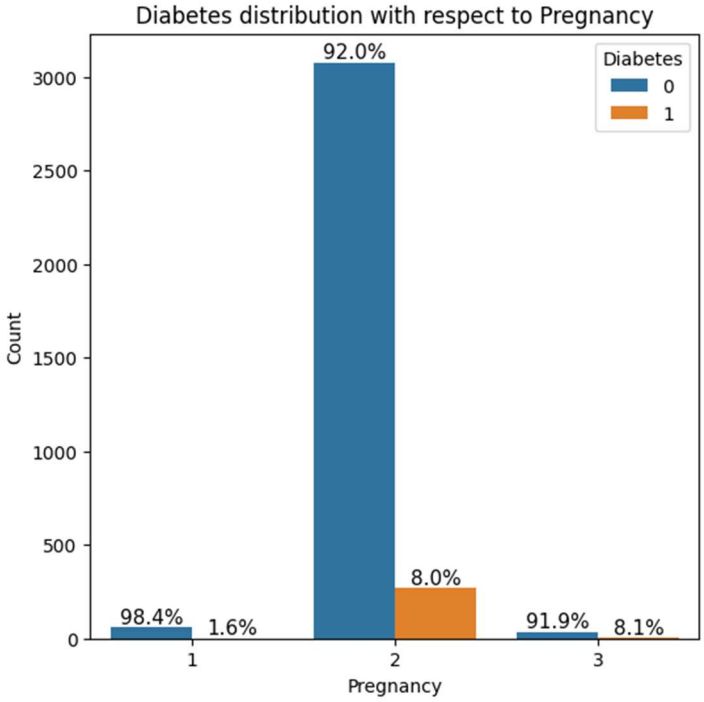
- Gender doesn't seem to have much impact on the diabetes patients distribution which can be observed from the below bar chart.



- As the age increases the chance of having diabetes seems to be increasing.

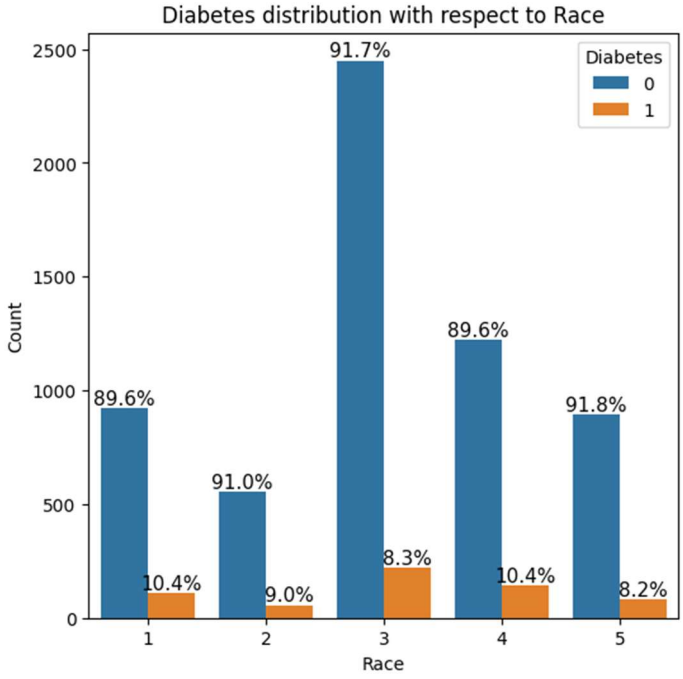


- It is observed that pregnancy women have a greater number of diabatic patient than non-pregnant women

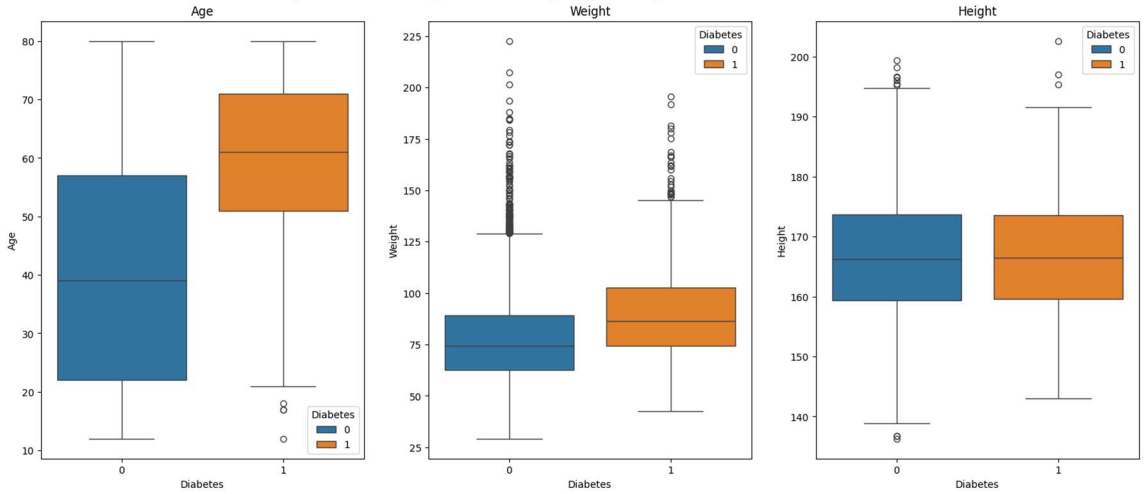


In the above graph category '1' indicates that the participant was not pregnant at the time of test. Category '2' indicates that the participant was pregnant at the time of exam. Category '3' are the participants that were not willing to answer.

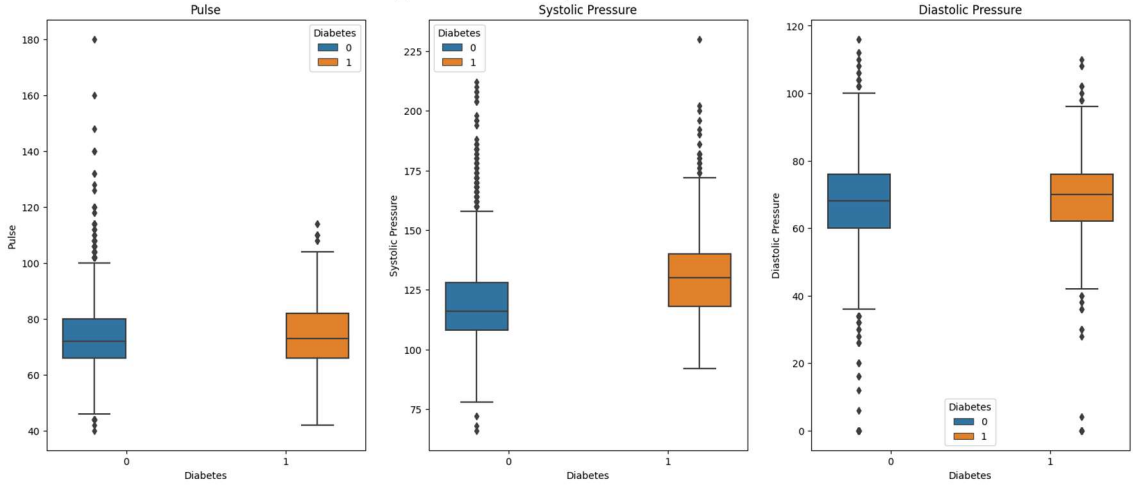
- Race has no significant impact on the share of diabetes patients and the share is almost same in all categories.



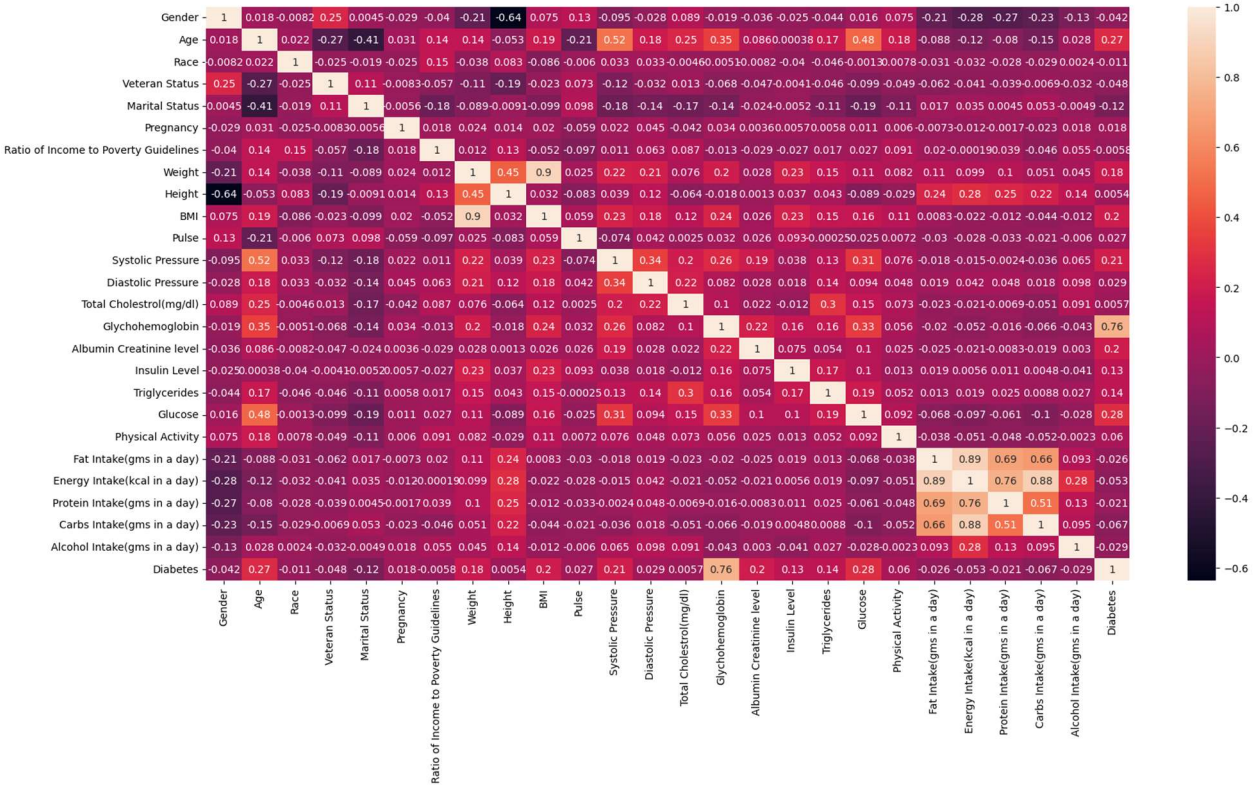
- It can be noticed that age and weight have good impact on diabetes.



- It is observed that the systolic pressure of diabetes patients is significantly higher than normal. Whereas, the diastolic pressure and pulse is similar to non-diabetes patients



Correlation analysis



From the heatmap, it is evident that fat, energy, carbohydrates, protein intake are heavily correlated

Data splitting

It is common practice to split the available data into separate sets for training and testing.
In this project, the data was split using the Hold out Method, with a 70:30 ratio (train size = 0.7) for the training and testing sets.
The predictor variables were represented by the variable "X" with indices 0 to 25,
The target variable "Diabetes" was represented by the variable "y" with an index of 26.
X_train and y_train represents the training datasets, which consisted of 4650 records and was used to train classification models.
X_test and y_test represents the testing set, which consists of 1993 records and was used to evaluate the performance of these models.

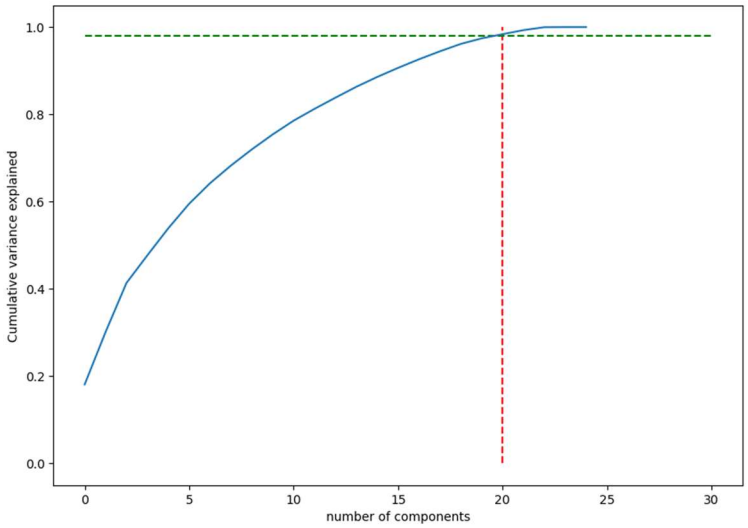
Balancing data using SMOTE

As mentioned above the dataset is imbalanced with only ~600 records of diabetes patients out of ~6600 records. To overcome this issue, I've applied SMOTE (Synthetic minority over sampling technique). Now the training dataset has 8448 records. Out of which 4224 are diabetic and 4224 are non-diabetic. Now the dataset had balanced distribution of minority and majority classes.

Principal Component analysis (PCA)

By identifying the primary components of the dataset, which are the directions in which the data varies the most, and projecting the data onto them, we can create a lower-dimensional representation that captures most of the variability in the original dataset. The shape of the

training data before applying PCA is (8448, 25) and trying to capture overall explained variance of 98%



The graph displays the relationship between the number of components (x-axis) and the cumulative frequency (y-axis). Based on the graph, we can determine the number of components that can explain a specific percentage of variance.
In this dataset 20 components leads to ~98% explained variance which is acceptable.

	Explained Variance	Proportion Variance	Cumulative Variance
PC1	5.331744	0.185598	0.180805
PC2	3.542801	0.123325	0.300945
PC3	3.315951	0.115428	0.413393
PC4	1.867932	0.065023	0.476736
PC5	1.826350	0.063575	0.538670
PC6	1.662017	0.057855	0.595031
PC7	1.387456	0.048297	0.642081
PC8	1.194272	0.041573	0.682580
PC9	1.091996	0.038012	0.719611
PC10	1.013285	0.035272	0.753972
PC11	0.928951	0.032337	0.785474
PC12	0.799172	0.027819	0.812575
PC13	0.754169	0.026253	0.838149
PC14	0.735762	0.025612	0.863100
PC15	0.665279	0.023158	0.885660
PC16	0.615447	0.021424	0.906531
PC17	0.579860	0.020185	0.926194
PC18	0.537818	0.018721	0.944432
PC19	0.500447	0.017421	0.961403
PC20	0.376708	0.013113	0.974178

Exploring supervised ML models

K-NN classifier

k-Nearest Neighbors (kNN) is a non-parametric statistical learning algorithm used for classification and regression. In kNN, the classification of a new instance is based on the classification of its k-nearest neighbors in the training set, determined by a distance metric such as Euclidean or Manhattan distance. The most common class among the k-nearest neighbors is assigned to the new instance.

Advantages of kNN:

- K-NN algorithm is very simple to understand and equally easy to implement. To classify the new data point K-NN algorithm reads through whole dataset to find out K nearest neighbors
- K-NN is a non-parametric algorithm which means there are assumptions to be met to implement K-NN. Parametric models like linear regression has lots of assumptions to be met by data before it can be implemented which is not the case with K-NN.
- K-NN does not explicitly build any model, it simply tags the new data entry based learning from historical data. New data entry would be tagged with majority class in the nearest neighbor.
- Given it's an instance-based learning; k-NN is a memory-based approach. The classifier immediately adapts as we collect new training data. It allows the algorithm to respond quickly to changes in the input during real-time use
- Most of the classifier algorithms are easy to implement for binary problems and needs effort to implement for multi class whereas K-NN adjust to multi class without any extra efforts
- One of the biggest advantages of K-NN is that K-NN can be used both for classification and regression problems

Disadvantages of kNN:

- kNN can be computationally expensive, particularly with large datasets.
- The optimal value of k can be difficult to determine and can impact the performance of the algorithm.
- k-NN doesn't perform well on imbalanced data. If we consider two classes, A and B, and the majority of the training data is labeled as A, then the model will ultimately give a lot of preference to A. This might result in getting the less common class B wrongly classified

Overall, kNN is a useful algorithm for classification tasks, particularly for small to medium-sized datasets, but its performance can be affected by the choice of distance metric, the value of k, and the presence of outliers.

Logistic Regression:

Logistic regression is a statistical learning algorithm used for binary classification problems. It models the probability of an instance belonging to a specific class, given its features. The goal of logistic regression is to estimate the coefficients of the features that maximize the likelihood of the observed data.

Advantages of logistic regression:

- It is a linear model that is computationally efficient and can be trained on large datasets
- Logistic regression provides interpretable results in terms of the coefficients of the features.
- It is less prone to overfitting compared to other complex models like neural networks.
- It can handle both continuous and categorical data and can be extended to multi-class classification problems.

Disadvantages of logistic regression:

- It assumes a linear relationship between the features and the log-odds of the response variable, which may not be the case in real-world problems.
- Logistic regression is sensitive to outliers and the presence of correlated features.

- It may not perform well when the decision boundary is nonlinear or complex.
- Logistic regression assumes that the errors are independent and identically distributed, which may not be the case in some datasets.

Overall, logistic regression is a useful algorithm for binary classification problems, particularly when the relationship between the features and the response variable is linear. However, it may not perform well in all scenarios, and its assumptions should be carefully considered when applied to real-world problems.

Decision Trees:

Decision trees are a non-parametric statistical learning algorithm used for classification and regression tasks. They model the relationship between the features and the response variable by recursively partitioning the data into smaller subsets based on the values of the features. The goal of a decision tree is to create a tree that maximizes the separation of the classes or minimizes the mean squared error in the case of regression.

Advantages of decision trees:

- Decision trees can handle both continuous and categorical data and do not require any assumptions about the underlying data distribution.
- They provide interpretable results in the form of a tree structure that can be visualized and understood by non-experts.
- Decision trees can handle interactions between the features and can identify important features for the classification task.
- They can be used for both binary and multi-class classification and can be extended to regression tasks.

Disadvantages of decision trees:

- Decision trees are prone to overfitting, especially when the tree is deep, or the data is noisy.
- They are sensitive to small variations in the data, which can result in different trees being generated for similar datasets.
- Decision trees can be biased towards features with more levels or high cardinality.
- They may not perform well on imbalanced datasets or when the decision boundary is nonlinear or complex.

Overall, decision trees are a useful algorithm for classification and regression tasks, particularly when the data is structured, and the goal is to understand the underlying relationships between the features and the response variable. However, their performance can be affected by overfitting, bias towards certain features, and sensitivity to variations in the data.

Random Forest:

Random forest is an ensemble learning method in statistics that combines multiple decision trees to create a more robust and accurate model. It is a non-parametric technique used for both classification and regression tasks.

Advantages of Random forest:

- High accuracy: Random forest produces highly accurate results as it combines multiple decision trees to make predictions.
- Robustness: Random forest is highly resistant to overfitting, noisy data, and outliers, making it suitable for complex and diverse datasets.

- Feature importance: Random forest provides a measure of feature importance that can be used to identify the most significant variables in a dataset.
- Easy to use: Random forest is relatively easy to use and implement, requiring minimal data preprocessing and feature engineering.

Disadvantages of Random forest:

- Computationally expensive: Building a random forest model can be computationally expensive, especially when dealing with large datasets and a large number of trees.
- Difficult to interpret: The results from a random forest model can be challenging to interpret, as the model's inner workings are not easily visible.
- Biased toward categorical variables: Random forest tends to be biased toward categorical variables with more levels or categories.
- Risk of overfitting: Although random forest is robust to overfitting, there is still a risk of overfitting when the number of trees is too high or when the dataset is imbalanced.

In conclusion, random forest is a powerful statistical technique with numerous advantages, including high accuracy, robustness, feature importance, and ease of use. However, it is not without its limitations, such as computational cost, interpretability, bias toward categorical variables, and the risk of overfitting.

Naive Bayes:

Naive Bayes is a probabilistic learning algorithm used for classification tasks. It is based on Bayes' theorem, which states that the probability of a hypothesis (class) given the observed evidence (features) is proportional to the product of the prior probability of the hypothesis and the likelihood of the evidence given the hypothesis. Naive Bayes assumes that the features are conditionally independent given the class, which means that the presence or absence of one feature does not affect the probability of the presence or absence of another feature.

Advantages of Naive Bayes:

- Naive Bayes is computationally efficient and can be trained on large datasets.
- It can handle both continuous and categorical data and can be extended to multi-class classification problems.
- Naive Bayes is a simple model that is easy to implement and interpret.
- It can handle irrelevant or redundant features by assigning them a low weight in the model.

Disadvantages of Naive Bayes:

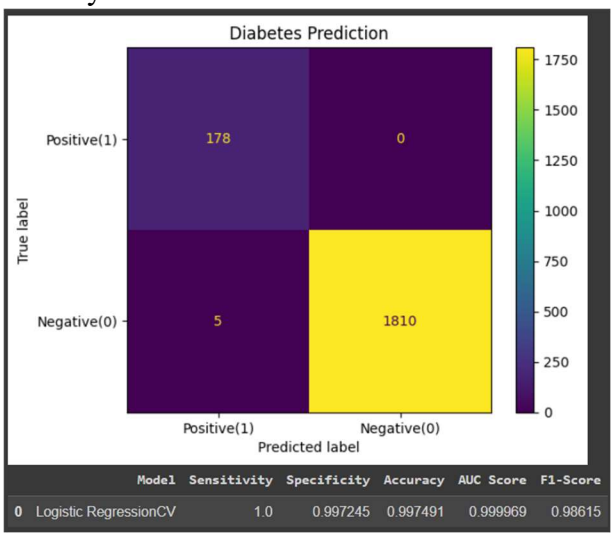
- Naive Bayes assumes that the features are conditionally independent given the class, which may not be the case in real-world problems.
- It may not perform well when the decision boundary is nonlinear or complex.
- Naive Bayes is sensitive to the presence of outliers or rare events in the data.
- It relies on the assumption of a strong prior distribution, which may not be available or accurate in some cases.

Overall, Naive Bayes is a useful algorithm for classification problems, particularly when the features are conditionally independent, and the problem is well-suited to the probabilistic framework. However, its performance can be affected by the independence assumption, the presence of outliers, and the quality of the prior distribution.

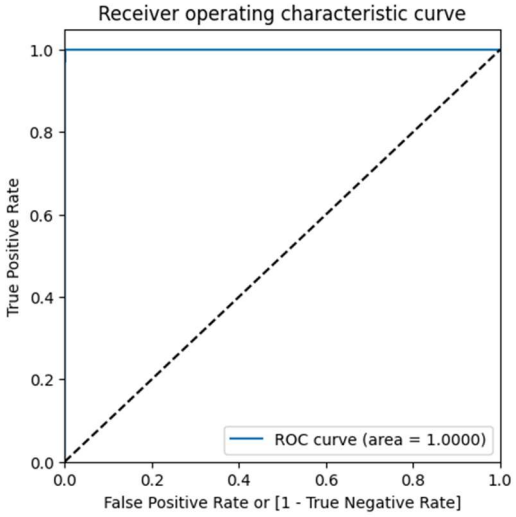
Performance Evaluation

1. Logistic Regression:

A logistic regression model was trained using cross-validation and grid search to optimize the hyperparameters. The model's performance was evaluated using metrics like sensitivity, specificity, accuracy, AUC score, and F1-score. The model predicted the outcomes with an accuracy of 0.997491. An F1 score of 0.98615 indicates that the model performance is good.



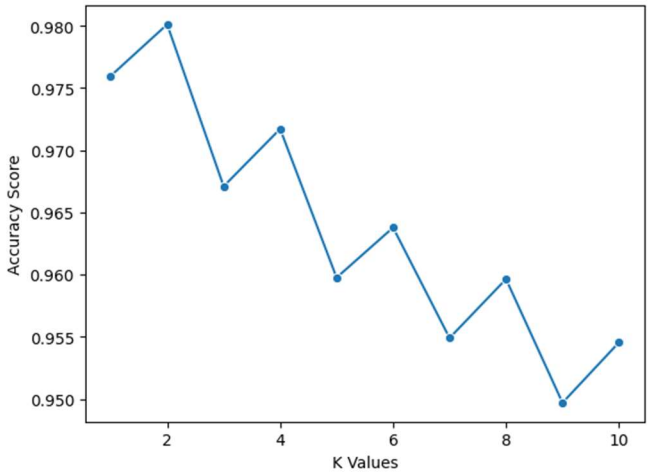
	Model	Sensitivity	Specificity	Accuracy	AUC Score	F1-Score
0	Logistic RegressionCV	1.0	0.997245	0.997491	0.999969	0.98615



Hyper parameter tuning was done using grid search with a five-fold cross-validation strategy, and the best value for the regularization parameter (c) was found to be 100, resulting in the highest accuracy of the model. The use of Stratified k-Fold with three splits ensured that the distribution of the target variable was maintained in each fold.

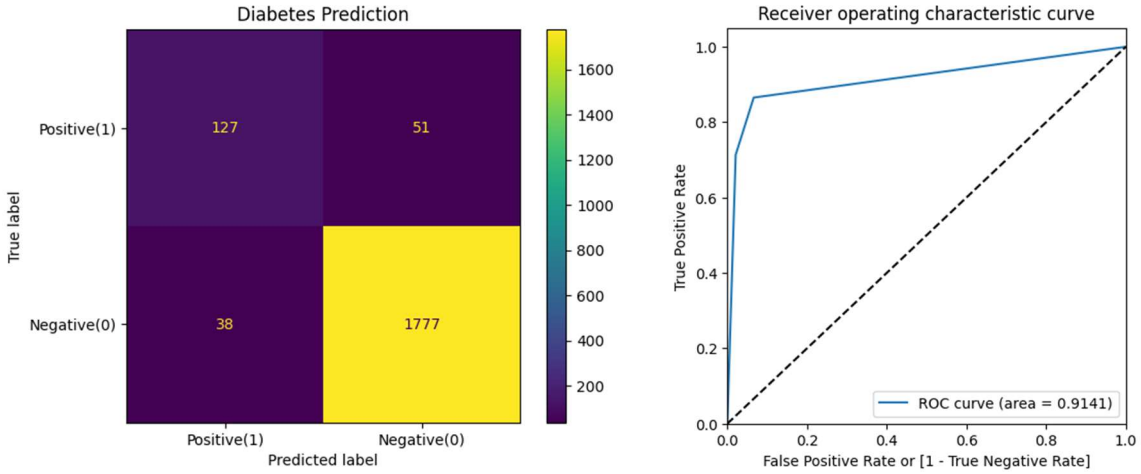
2. k-NN Classifier:

The best value for the hyperparameter k in KNN model was found to be 2 through cross validation. This suggests that the model may be underfitting when k is set to higher values, and a value of 2 provides the best balance between bias and variance in the model.



The model has an accuracy of 0.955344 indicating that the model classifies 95.5% of the data correctly. The sensitivity of the model is 0.713483, indicating that the model is able to classify 71.3% of the positive class instances correctly. The specificity of the model is 0.979063, indicating that the model is able to classify 97.9% of the negative class instances correctly. The AUC score of 0.914056 indicates that the model is able to distinguishing between positive and negative instances well. The F1-score of the model is 0.740525, which

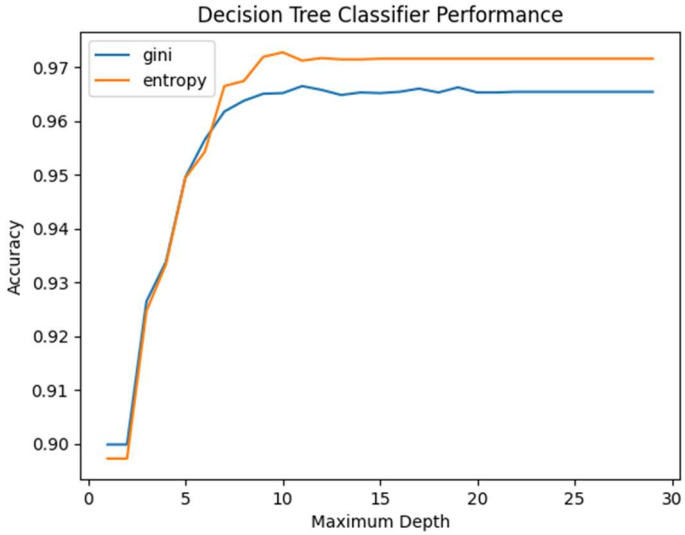
is a measure of the model's balance between precision and recall. A high F1-score indicates a model with high precision and recall.



	Model	Sensitivity	Specificity	Accuracy	AUC Score	F1-Score
0	KNN	0.713483	0.979063	0.955344	0.914056	0.740525

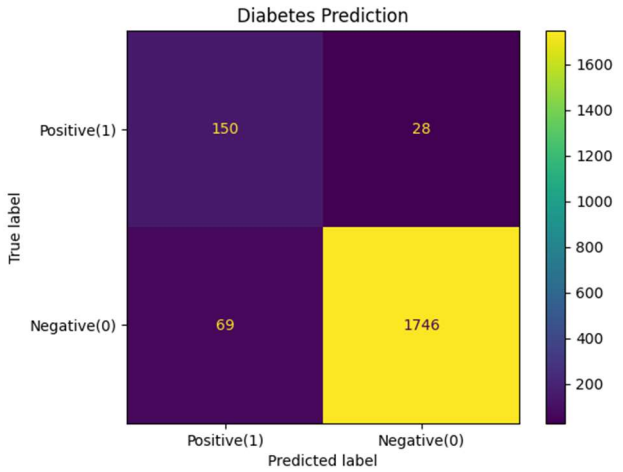
3. Decision trees:

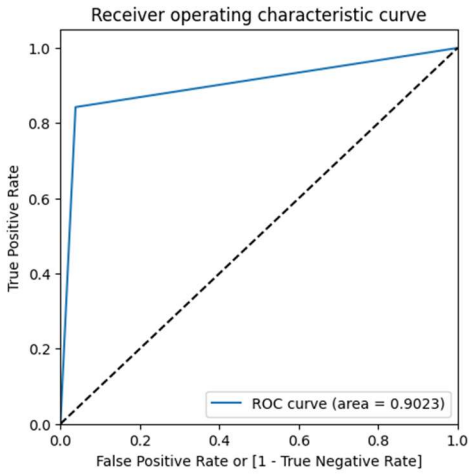
The hyperparameters max depth and impurity method were found to be optimized at 15 and entropy respectively, through cross-validation. This suggests that a deeper tree with the entropy impurity method provides the best balance between bias and variance in the model.



The following parameters are used for training the model, max_depth=21, criterion='entropy'.

The model has an accuracy of 0.95133 indicating that the model classifies 95.1% of the data correctly. The sensitivity of the model is 0.842697, indicating that the model is able to classify 84.3% of the positive class instances correctly. The specificity of the model is 0.961983, indicating that the model is able to classify 96.2% of the negative class instances correctly. The AUC score of 0.90234 indicates that the model is able to distinguishing between positive and negative instances well. The F1-score of the model is 0.755668, which is a measure of the model's balance between precision and recall. A high F1-score indicates a model with high precision and recall.

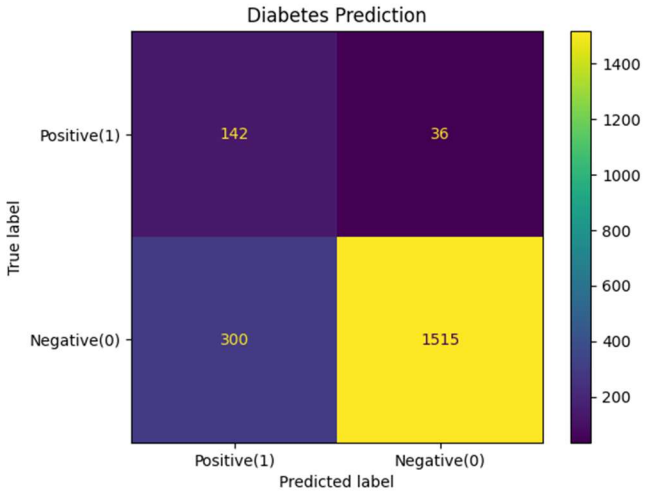


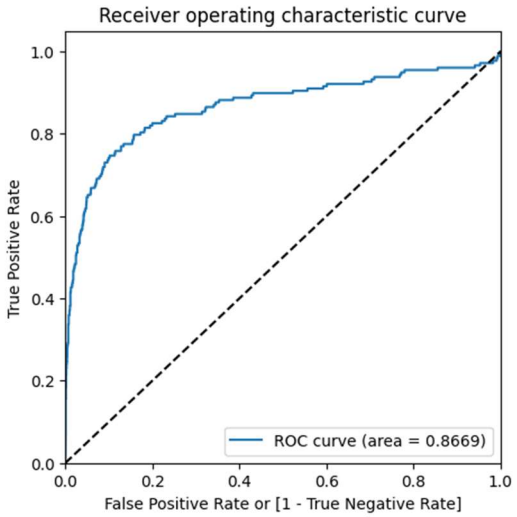


	Model	Sensitivity	Specificity	Accuracy	AUC Score	F1-Score
0	Decision Tree	0.842697	0.961983	0.95133	0.90234	0.755668

4. Naïve Bayes Classifier:

The model has an accuracy of 0.83141 indicating that the model classifies 83.1% of the data correctly. The sensitivity of the model is 0.797753, indicating that the model is able to classify 79.7% of the positive class instances correctly. The specificity of the model is 0.834711, indicating that the model is able to classify 83.5% of the negative class instances correctly. The AUC score of 0.866879 indicates that the model is able to distinguishing between positive and negative instances well. The F1-score of the model is 0.458065, which is a measure of the model's balance between precision and recall. A high F1-score indicates a model with high precision and recall.



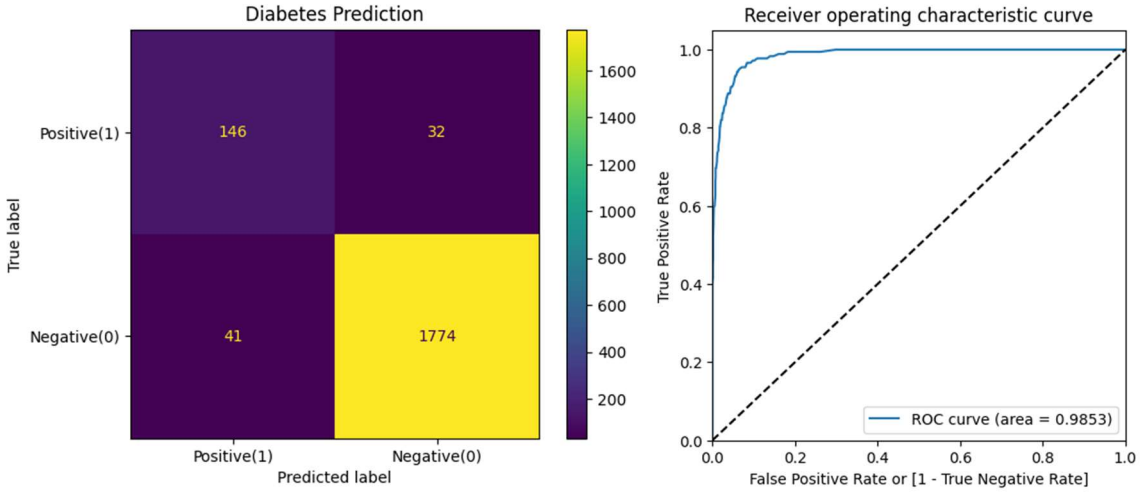


	Model	Sensitivity	Specificity	Accuracy	AUC Score	F1-Score
0	Naive Bayes	0.797753	0.834711	0.83141	0.866879	0.458065

5. Random Forest:

The model has an accuracy of 0.963372 indicating that the model classifies 96.3% of the data correctly. The sensitivity of the model is 0.820225, indicating that the model is able to classify 82% of the positive class instances correctly. The specificity of the model is 0.97741,

indicating that the model is able to classify 97.7% of the negative class instances correctly. The AUC score of 0.985274 indicates that the model is able to distinguishing between positive and negative instances well. The F1-score of the model is 0.8, which is a measure of the model's balance between precision and recall. A high F1-score indicates a model with high precision and recall.



	Model	Sensitivity	Specificity	Accuracy	AUC Score	F1-Score
0	Random Forest	0.820225	0.97741	0.963372	0.985274	0.8

Models Comparison

The performance of the selected 5 models (logistic regression, kNN, Decision tree, Naïve Bayes, Random forest) are compared using various performance metrics.

Model	Sensitivity	Specificity	Accuracy	AUC Score	F1-Score
Logistic RegressionCV	1.000000	0.997245	0.997491	0.999969	0.986150
KNN	0.713483	0.979063	0.955344	0.914056	0.740525
Decision Tree	0.842697	0.961983	0.951330	0.902340	0.755668
Naive Bayes	0.797753	0.834711	0.831410	0.866879	0.458065
Random Forest	0.820225	0.977410	0.963372	0.985274	0.800000

Logistic regression model has the highest accuracy of 99.7% indicating its ability to classify both positive and negative cases correctly. This model also has high sensitivity, specificity and AUC scores.

Decision Tree and Naïve Bayes has comparatively lower accuracy rate making it least favorable in this case.

kNN model has good accuracy (95.5%) but, the sensitivity and F1-scores are low making it less favorable compared to Logistic regression model

In conclusion, Logistic Regression model perform best in predicting diabetes using health related parameters, with high accuracy rates, sensitivity, specificity, AUC score, and F1-score. However, the other models also exhibit moderate to high accuracy rates and can be considered as alternatives in situations where these models are not applicable or suitable. Therefore, the choice of the best model depends on the specific context of the application and the desired trade-off between different performance metrics