AI BASED DIABETES PREDICTION SYSTEM

TEAM MEMBER

- NAME: S.HARIPRABU
- REGNO: 912421106303
- Gmail id: hariprabu1734@gmail.com
- NM ID: aut2291240022

PHASE 5 Submission Document

Phase 5 : Project Documentation & Submission

Topic : In this section we will document the complete project and prepare it for submission.

Introduction

- Diabetes is a chronic disease that directly affects the pancreas, and the body is incapable of producing insulin. It is mainly responsible for maintaining the blood glucose level.
- AI-based diabetes prediction systems are computer programs that use machine learning algorithms to analyze patient data and predict their risk of developing diabetes.
- To use machine learning classification methods, that is, decision tree, SVM, Random Forest, Logistic Regression, KNN, and various ensemble techniques, to determine which algorithm produces the best prediction results.
- In this paper, we have employed machine learning and explainable AI techniques to detect diabetes.

AI-based diabetes prediction systems typically use a variety of data inputs, including:

- Demographic data (age, sex, race, ethnicity)
- Medical history (including family history of diabetes, previous medical conditions, and medications taken)
- Lifestyle factors (diet, exercise, smoking status, alcohol consumption)
- Biometric data (weight, height, blood pressure, fasting blood glucose level, HbA1c level)
- The machine learning algorithm then analyzes this data to identify patterns that are associated with diabetes risk. For example, the algorithm may learn that people with a certain combination of age, BMI, and blood pressure are at increased risk for developing diabetes.

Abstract

Artificial intelligence (AI)-based diabetes prediction systems are a promising new approach to early detection and prevention of diabetes. These systems use machine learning to analyze a person's medical history and other risk factors to predict their risk of developing diabetes. This information can then be used to develop personalized risk assessments and treatment plans.

AI-based diabetes prediction systems typically use a variety of data sources, including demographic data, medical history, lifestyle factors, and biometric data. The machine learning algorithms used in these systems are trained on large datasets of patient records to learn patterns in the data that are associated with diabetes risk.

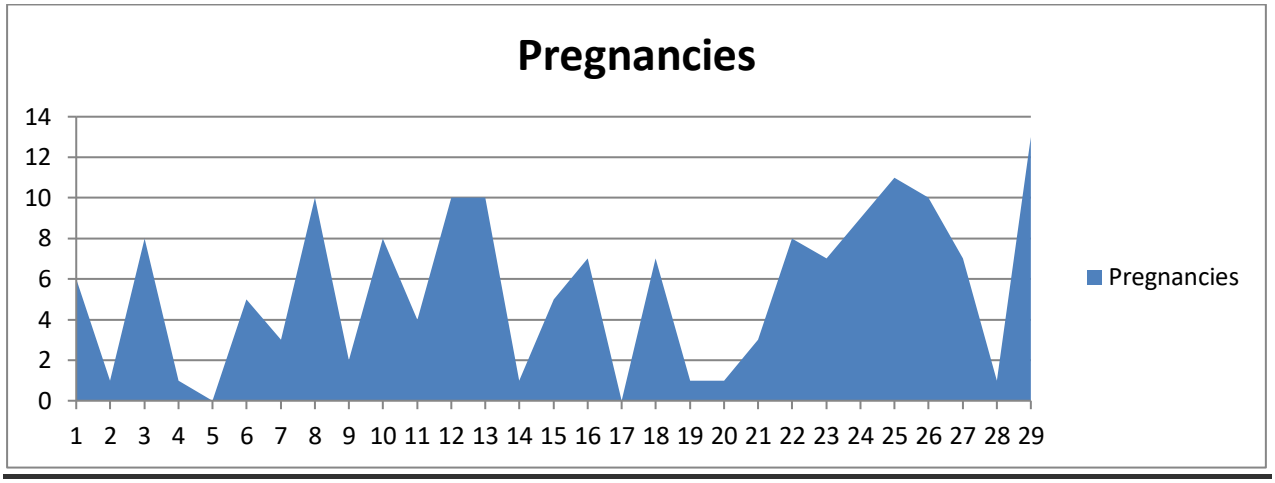
- **Early detection and prevention:** AI-based systems can identify people who are at high risk for diabetes before they develop the disease. This allows for early intervention, such as lifestyle changes or medication, which can help to prevent or delay the onset of diabetes.
- **Personalized care:** AI-based systems can be used to develop personalized risk assessments and treatment plans for people with diabetes. This can help to improve the effectiveness of care and reduce the risk of complications.
- **Improved healthcare efficiency:** AI-based systems can help to streamline the diabetes care process and reduce the workload on healthcare providers.

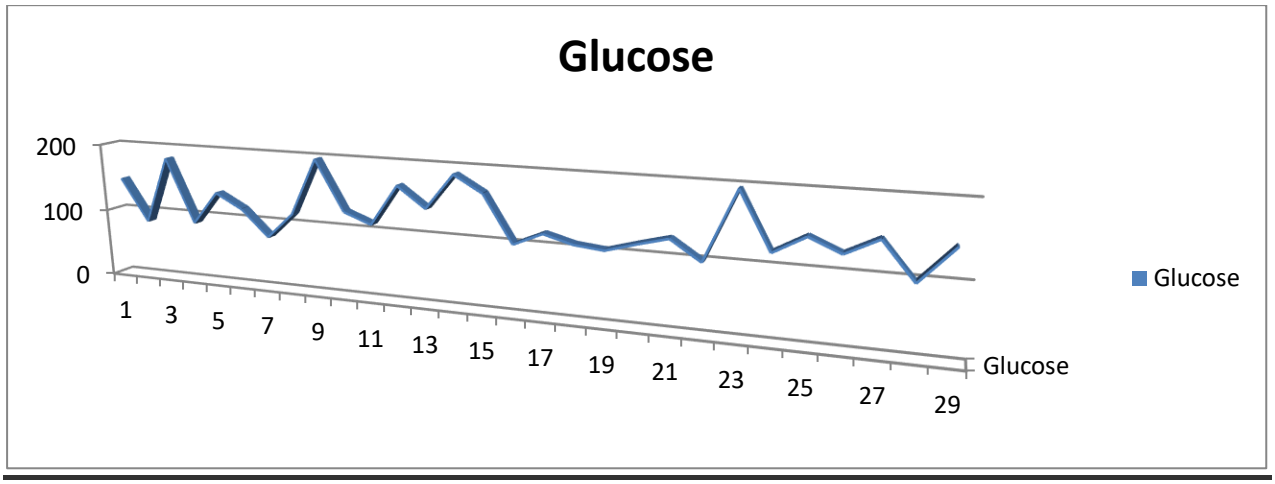
DatasetLink: https://www.kaggle.com/datasets/mathchi/diabetes-data-set

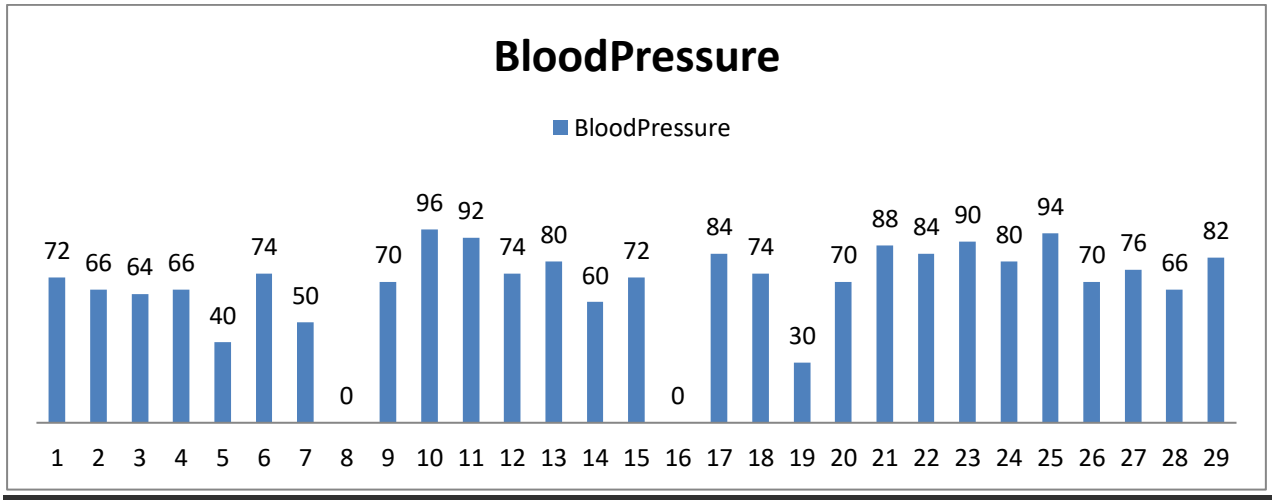
Diabetes dataset

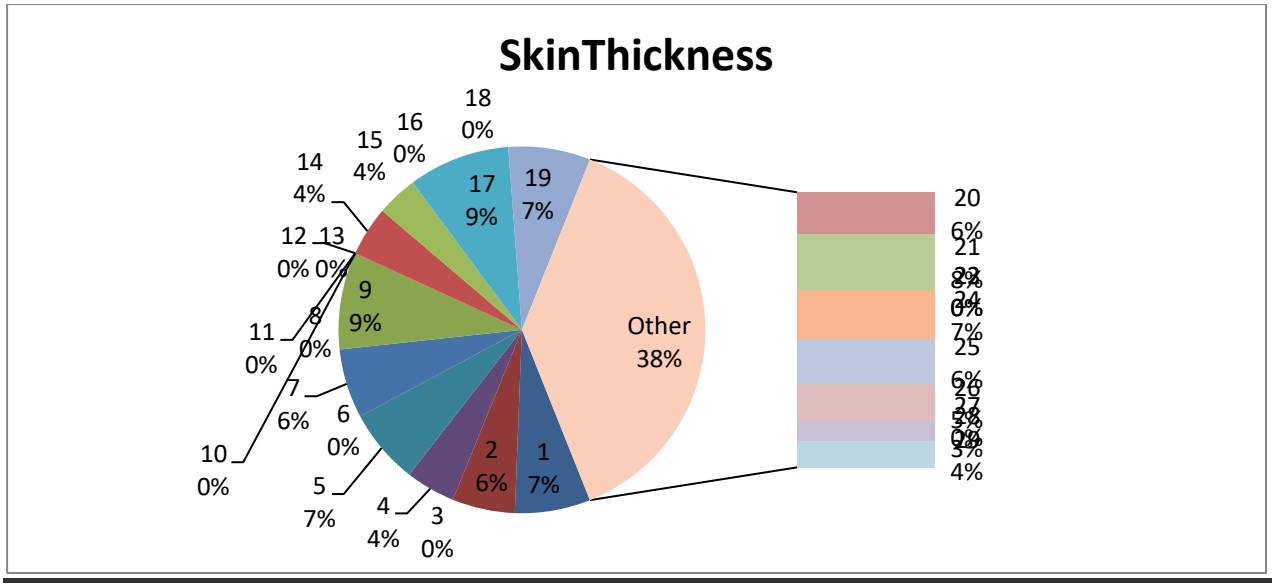
Pregnanci	Glucose	BloodPres	SkinThickr	Insulin	BMI	DiabetesP	Age	Outcome
6	148	72	35	0	33.6	0.627	50	1
1	85	66	29	0	26.6	0.351	31	0
8	183	64	0	0	23.3	0.672	32	1
1	89	66	23	94	28.1	0.167	21	0
0	137	40	35	168	43.1	2.288	33	1
5	116	74	0	0	25.6	0.201	30	0
3	78	50	32	88	31	0.248	26	1
10	115	0	0	0	35.3	0.134	29	0
2	197	70	45	543	30.5	0.158	53	1
8	125	96	0	0	0	0.232	54	1
4	110	92	0	0	37.6	0.191	30	0
10	168	74	0	0	38	0.537	34	1
10	139	80	0	0	27.1	1.441	57	0
1	189	60	23	846	30.1	0.398	59	1
5	166	72	19	175	25.8	0.587	51	1
7	100	0	0	0	30	0.484	32	1
0	118	84	47	230	45.8	0.551	31	1
7	107	74	0	0	29.6	0.254	31	1
1	103	30	38	83	43.3	0.183	33	0
1	115	70	30	96	34.6	0.529	32	1
3	126	88	41	235	39.3	0.704	27	0
8	99	84	0	0	35.4	0.388	50	0
7	196	90	0	0	39.8	0.451	41	1
9	119	80	35	0	29	0.263	29	1
11	143	94	33	146	36.6	0.254	51	1
10	125	70	26	115	31.1	0.205	41	1
7	147	76	0	0	39.4	0.257	43	1
1	97	66	15	140	23.2	0.487	22	0
13	145	82	19	110	22.2	0.245	57	0

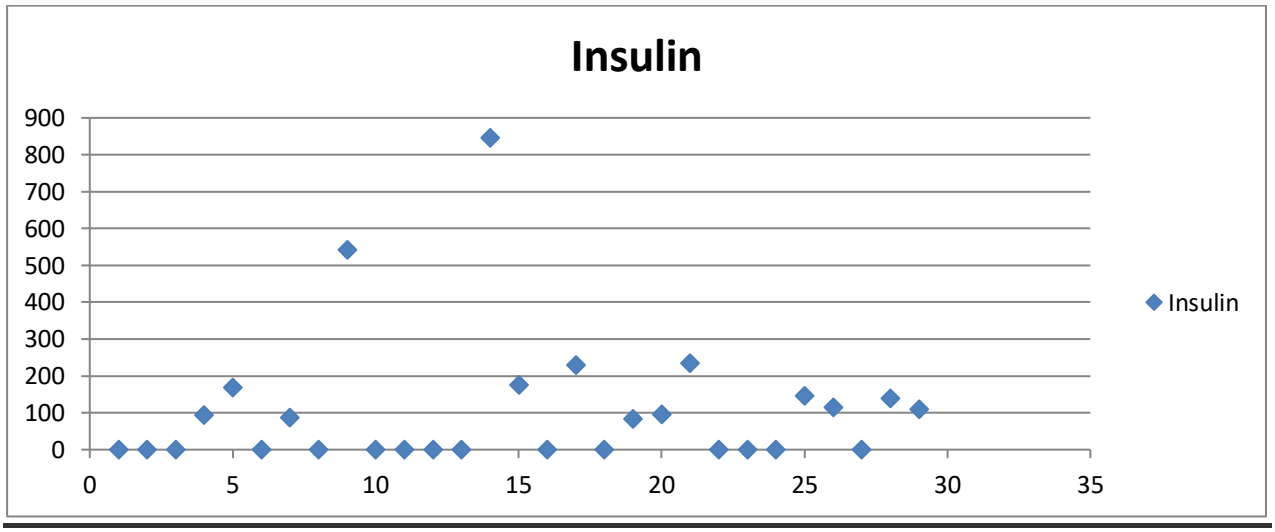
Chats

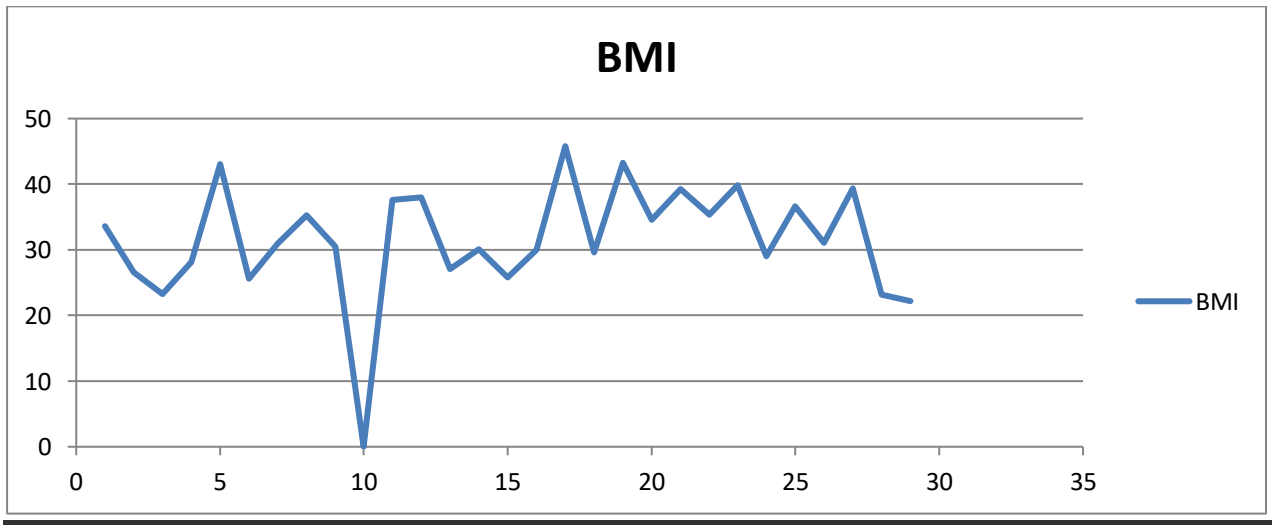

Pregnancies
14
12
10
8
6
4
2
0
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29
Pregnancies

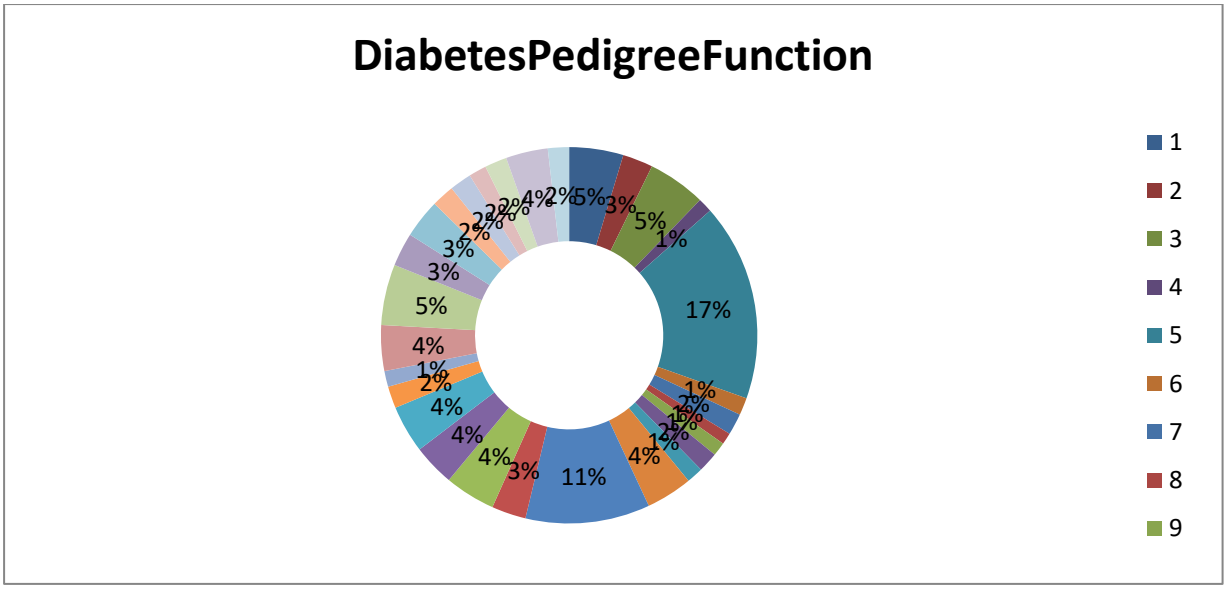

Glucose
200
100
0
1 3 5 7 9 11 13 15 17 19 21 23 25 27 29
Glucose
Glucose

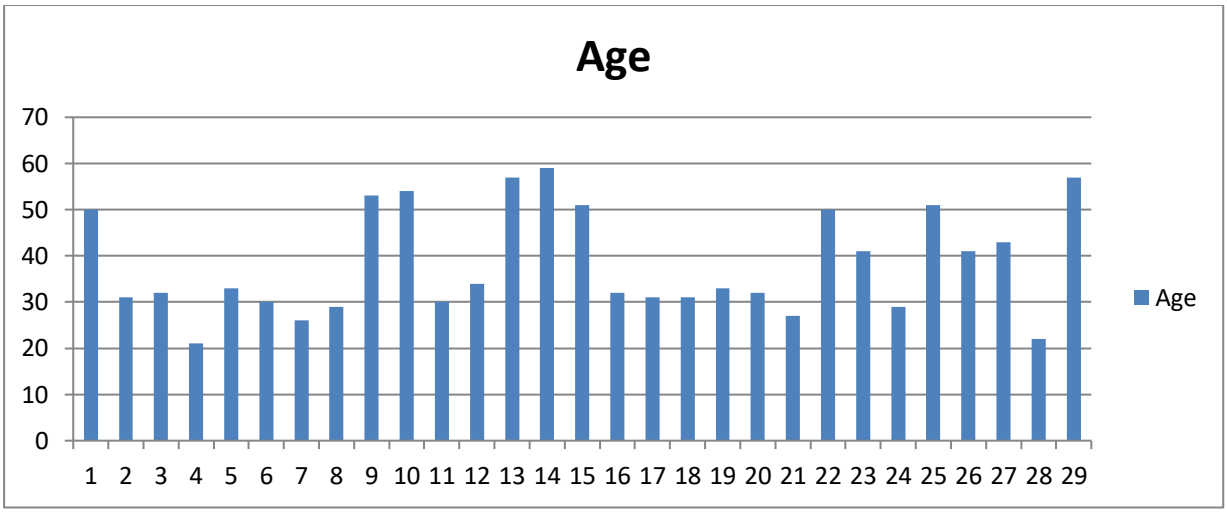

BloodPressure
BloodPressure
72 66 64 66 40 74 50 0 70 96 92 74 80 60 72 0 84 74 30 70 88 84 90 80 94 70 76 66 82
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29

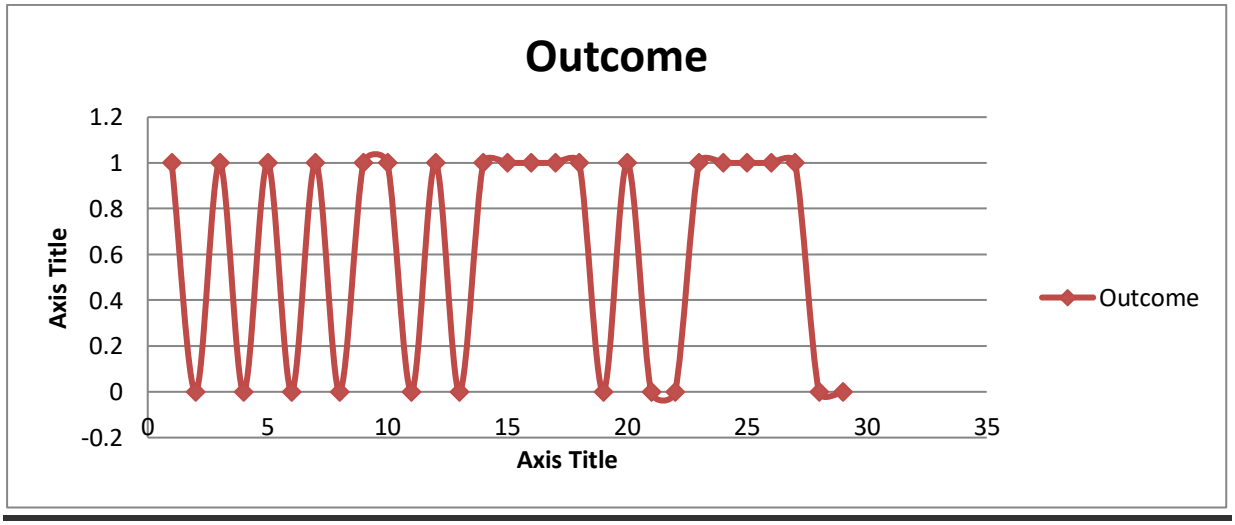

SkinThickness
18 0%
16 0%
15 4%
14 4%
17 9%
19 7%
12 0%
13 0%
9 9%
11 0%
8 0%
7 6%
6 0%
10 0%
5 7%
4 4%
3 0%
2 6%
1 7%
Other 38%
20 6%
21 6%
25 6%
4%


Insulin
900
800
700
600
500
400
300
200
100
0
0
5
10
15
20
25
30
35
Insulin


BMI
50
40
30
20
10
0
0
5
10
15
20
25
30
35
BMI


DiabetesPedigreeFunction
1
2
3
4
5
6
7
8
9


Age
70
60
50
40
30
20
10
0
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29
Age


Outcome
1.2
1
0.8
0.6
0.4
0.2
0
-0.2
0 5 10 15 20 25 30 35
Axis Title
Axis Title
Outcome

Here is a list of some of the tools and software used in the process of developing and using AI-based diabetes prediction systems:

1. Tools for data collection and preparation:

- Electronic health records (EHRs)
- Clinical data warehouses (CDWs)
- Data visualization tools
- Data cleaning and preprocessing tools

2. Tools for machine learning model development and evaluation:

- Programming languages such as Python and R
- Machine learning libraries such as TensorFlow, PyTorch, and scikit-learn
- Model evaluation tools such as cross-validation and confusion matrices

3. Software for deploying and using AI-based diabetes prediction systems:

- Web applications
- Mobile apps
- Clinical decision support systems (CDSSs)

Here are some specific examples of tools and software that are being used to develop and use AI-based diabetes prediction systems:

4. Tools for data collection and preparation:

- Google Cloud Healthcare API
- Amazon HealthLake
- Microsoft Azure Healthcare API
- SAS Visual Analytics

5. Tools for machine learning model development and evaluation:

- TensorFlow
- PyTorch
- scikit-learn
- H2O.ai Driverless AI
- RapidMiner

6. Software for deploying and using AI-based diabetes prediction systems:

- AWS Rekognition Health
- Google Cloud Healthcare API
- Microsoft Azure Databricks
- SAS Visual Analytics

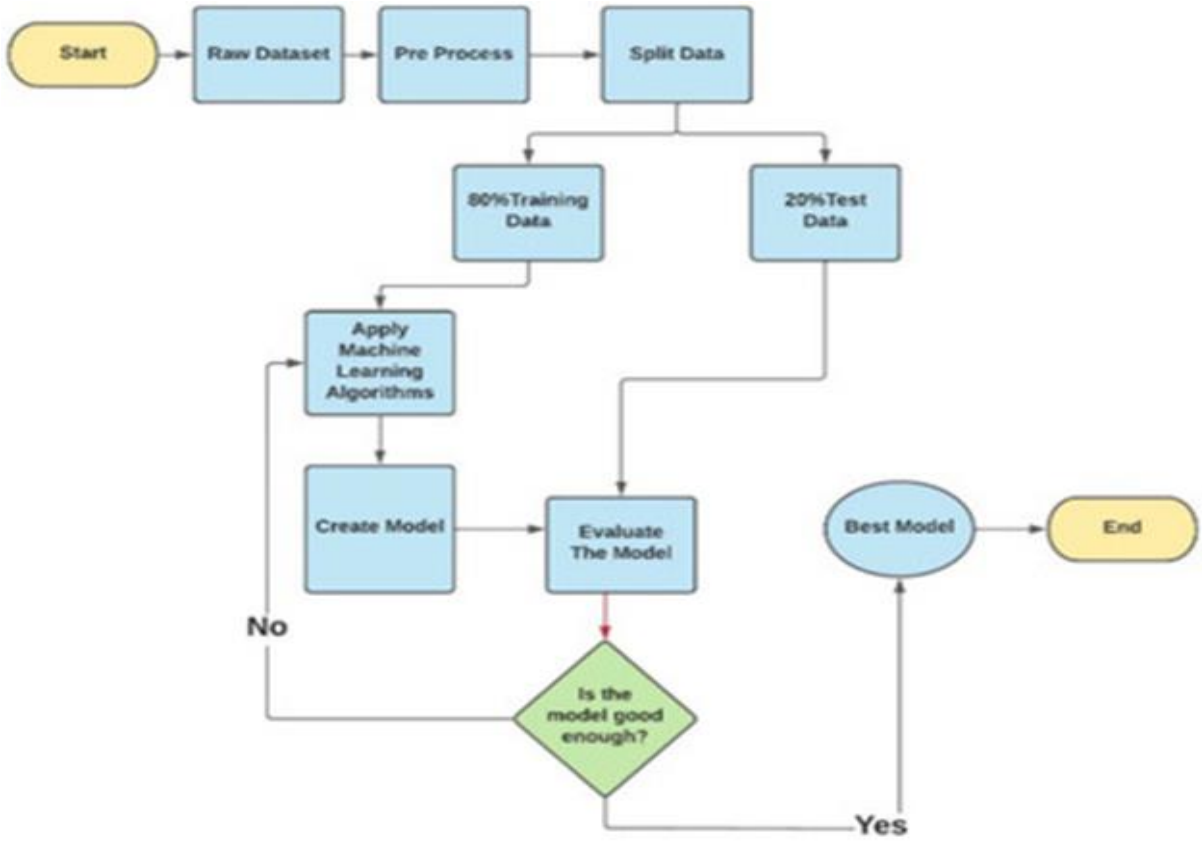
Proposed System

- ❖ This section describes the working procedures and implementation of various machine learning techniques to design the proposed automatic diabetes prediction system.

- ❖ First, the dataset was collected and preprocessed to remove the necessary discrepancies from the dataset

- ❖ Then the dataset was separated into the training set and test set using the holdout validation technique.

- ❖ Next, different classification algorithms were applied to find the best classification algorithm for this dataset.



PROBLEM DEFINITION AND DESIGN THINKING

Problem Definition

- Diabetes is a chronic disease that affects millions of people worldwide. It is characterized by high blood sugar levels, which can lead to serious health complications such as heart disease, stroke, blindness, and kidney failure.

- Early detection and prevention of diabetes are essential for improving patient outcomes. However, traditional methods of diabetes screening, such as blood tests, are often time-consuming and expensive.

- AI-based diabetes prediction systems have the potential to revolutionize the way that diabetes is diagnosed and managed.

- These systems can use machine learning algorithms to analyze patient data and predict their risk of developing diabetes. This information can then be used to target people who are at high risk for diabetes with preventive interventions.

Design Thinking

Design thinking is a human-centered approach to innovation that focuses on understanding the needs of users and developing solutions that meet those needs. Design thinking can be used to develop AI-based diabetes prediction systems that are effective, user-friendly, and accessible.

Here is a five-stage design thinking process for developing AI-based diabetes prediction systems:

1. **Empathize:** The first step is to empathize with the users of the system. This involves understanding their needs, pain points, and motivations. For example, healthcare providers may need a system that can quickly and easily identify patients who are at high risk for diabetes. People with diabetes may need a system that can help them to manage their condition and track their progress.

2. **Define:** Once the needs of the users are understood, the next step is to define the problem. This involves identifying the specific challenges that the system needs to address. For example, the system may need to be able to predict diabetes risk with high accuracy, even for patients with limited data. The system may also need to be able to provide personalized recommendations to users.

3. **Ideate:** Once the problem is defined, the next step is to ideate potential solutions. This involves brainstorming different ways to address the problem and coming up with new and innovative ideas. For example, the system could use machine learning algorithms to analyze patient data from a variety of sources, such as EHRs, wearable devices, and social media. The system could also use gamification elements to motivate users to adopt healthy behaviors.

4. **Prototype:** Once a potential solution has been identified, the next step is to prototype it. This involves creating a working model of the solution to test its feasibility and usability. For example, a prototype of the system could be developed as a web app or mobile app. The prototype could then be tested with users to get feedback and make improvements.

5. **Test:** Once the prototype is ready, the next step is to test it in a real-world setting. This involves deploying the system to a limited number of users and collecting data on its performance. The data can then be used to evaluate the system's effectiveness and identify areas for improvement.

DESIGN INTO INNOVATION

AI-based diabetes prediction systems have the potential to be a powerful tool for preventing and managing diabetes. However, in order to be truly innovative, these systems need to be designed with a focus on the user experience, fairness, and equity.

- Make the systems user-friendly and accessible. The systems should be easy to use for people of all ages and technical abilities. They should also be available in multiple languages and accessible to people with disabilities.
- Use design thinking to understand the needs of users. Designers should work with healthcare providers, people with diabetes, and other stakeholders to understand their needs and pain points. This information can then be used to develop systems that are truly helpful and useful.
- Design the systems to be fair and equitable. The systems should be trained on data that is representative of the population that they will be used to serve. This will help to reduce the risk of bias. The systems should also be designed to provide accurate and relevant information to all users, regardless of their race, ethnicity, gender, socioeconomic status, or other factors.
- Make the systems transparent and accountable. Users should be able to understand how the systems work and make informed decisions about their use. The systems should also be designed to minimize the risk of harm.

1. System architecture:

The proposed system consists of the following components:

- **Data collection and preparation module:** This module is responsible for collecting and preparing data from various sources,

such as EHRs, CDWs, and wearable devices. The data is cleaned and preprocessed to make it suitable for machine learning model development.

- **Machine learning model development and evaluation module:** This module is responsible for developing and evaluating machine learning models to predict diabetes risk. A variety of machine learning algorithms can be used, such as logistic regression, random forests, and support vector machines. The models are evaluated on their accuracy, sensitivity, specificity, and other metrics.
- **Model deployment module:** This module is responsible for deploying the trained machine learning model to a production environment. The model can be deployed as a web service, mobile app, or CDSS.

2. System workflow:

The proposed system works as follows:

1. The data collection and preparation module collects and prepares data from various sources.
2. The machine learning model development and evaluation module develops and evaluates machine learning models to predict diabetes risk.
3. The model deployment module deploys the trained machine learning model to a production environment.

3. System features:

The proposed system has the following features:

- **Accuracy:** The system uses state-of-the-art machine learning algorithms to achieve high accuracy in predicting diabetes risk.
- **Explainability:** The system provides explanations for its predictions, which can help healthcare providers to better understand the risk factors for diabetes.
- **Personalization:** The system can be used to develop personalized risk assessments and treatment plans for people with diabetes.
- **Scalability:** The system can be scaled to support large numbers of users.

4. System benefits:

The proposed system has the following benefits:

- Early detection and prevention: The system can identify people who are at high risk for diabetes before they develop the disease. This allows for early intervention, such as lifestyle changes or medication, which can help to prevent or delay the onset of diabetes.
- Improved healthcare outcomes: The system can help healthcare providers to better manage diabetes in their patients. This can lead to improved healthcare outcomes, such as reduced risk of complications and improved quality of life.
- Reduced healthcare costs: The system can help to reduce healthcare costs by identifying people who are at high risk for diabetes and by helping to prevent or delay the onset of the disease.

BUILDING OUR PROJECT BY LOADING AND PREPROCESSING THE DATASET

Loading the dataset

The first step in building a dataset for an AI-based diabetes prediction system is to load the data. This can be done from a variety of sources, such as:

- **Public datasets:** There are a number of public datasets available that contain data on diabetes patients. These datasets can be found on websites such as the National Institute of Diabetes and Digestive and Kidney Diseases (NIDDK) and the Kaggle machine learning platform.
- **Electronic health records (EHRs):** EHRs contain a wealth of information on patients' health history, including their medical diagnoses, medications, and laboratory results. This data can be used to build a dataset for predicting diabetes risk.
- **Clinical trials:** Clinical trials generate data on the safety and efficacy of new diabetes medications and treatments. This data can also be used to build a dataset for predicting diabetes risk.

Once the data has been loaded, it needs to be cleaned and preprocessed. This involves removing any errors or inconsistencies in the data, and converting the data into a format that can be used by the machine learning algorithm.

Preprocessing the dataset

The following are some common preprocessing steps that are used for diabetes prediction datasets:

- **Handling missing values:** Missing values in the dataset can be imputed using a variety of methods, such as mean imputation, median imputation, and mode imputation.
- **Feature engineering:** Feature engineering is the process of creating new features from the existing features in the dataset. This can be done to improve the accuracy of the machine learning model.
- **Scaling the features:** Scaling the features ensures that they all have the same scale, which can improve the performance of the machine learning model.
- **Splitting the dataset into training and testing datasets:** The dataset is typically split into two parts: a training dataset and a testing dataset. The training dataset is used to train the machine learning model, and the testing dataset is used to evaluate the performance of the model.

Program

```
# Import libraries

import numpy as np # for linear algebra

import pandas as pd # for data processing, CSV file I/O (e.g. pd.read_csv)

import seaborn as sns # for data visualization

import matplotlib.pyplot as plt # to plot data visualization cha rts

from collections import Counter

import os

 # Modeling Libraries
```

from sklearn.metrics **import** confusion_matrix, accuracy_sco re, precision_score

from sklearn.preprocessing **import** QuantileTransformer

from sklearn.linear_model **import** LogisticRegression

from sklearn.neighbors **import** KNeighborsClassifier

from sklearn.tree **import** DecisionTreeClassifier

from sklearn.ensemble **import** RandomForestClassifier, Ada BoostClassifier, GradientBoostingClassifier

from sklearn.model_selection **import** GridSearchCV, cross_v al_score, StratifiedKFold, learning_curve, train_test_split

from sklearn.svm **import** SVC

Importing the Dataset

Importing the dataset from Kaggle

data = pd.read_csv("../input/pima-indians-diabetes-database/diabetes.csv")

# First step is getting familiar with the structure of the dataset
data.info()

Output

<class 'pandas.core.frame.DataFrame'>

RangeIndex: 768 entries, 0 to 767

Data columns (total 9 columns):

#	Column	Non-Null Count	Dtype
0	Pregnancies	768 non-null	int64
1	Glucose	768 non-null	int64
2	BloodPressure	768 non-null	int64
3	SkinThickness	768 non-null	int64
4	Insulin	768 non-null	int64
5	BMI	768 non-null	float64
6	DiabetesPedigreeFunction	768 non-null	float64
7	Age	768 non-null	int64
8	Outcome	768 non-null	int64

dtypes: float64(2), int64(7)

memory usage: 54.1 KB

Code

```
# Showing the top 5 rows of the dataset
data.head()
```

Output

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
0	6	148.0	72.0	35	30.5	33.6	0.627	50	1
1	1	85.0	66.0	29	30.5	26.6	0.351	31	0
2	8	183.0	64.0	23	30.5	23.3	0.672	32	1
3	1	89.0	66.0	23	94.0	28.1	0.167	21	0
4	0	137.0	40.0	35	168.0	43.1	2.288	33	1

Filling the Missing Values Code

Exploring the missing values in the diabetes dataset

data.isnull().sum()

Output

Pregnancies 0

Glucose 0

BloodPressure 0

SkinThickness 0

Insulin 0

BMI 0

DiabetesPedigreeFunction 0

Age 0

Outcome 0

dtype: int64

Code

Reviewing the dataset statistics

data.describe()

Output

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
count	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000	768.000000
mean	3.845052	121.656250	72.386719	26.606479	118.660163	32.450805	0.471876	33.240885	0.348958
std	3.369578	30.438286	12.096642	9.631241	93.080358	6.875374	0.331329	11.760232	0.476951
min	0.000000	44.000000	24.000000	7.000000	14.000000	18.200000	0.078000	21.000000	0.000000
25%	1.000000	99.750000	64.000000	20.536458	79.799479	27.500000	0.243750	24.000000	0.000000
50%	3.000000	117.000000	72.000000	23.000000	79.799479	32.000000	0.372500	29.000000	0.000000
75%	6.000000	140.250000	80.000000	32.000000	127.250000	36.600000	0.626250	41.000000	1.000000
max	17.000000	199.000000	122.000000	99.000000	846.000000	67.100000	2.420000	81.000000	1.000000

PERFORMING DIFFERENT ACTIVITIES LIKE FEATURE ENGINEERING, MODEL TRAINING, EVALUATION ETC.,

Feature engineering

Feature engineering is the process of transforming raw data into features that are more informative and predictive for a machine learning model. In the context of AI-based diabetes prediction systems, feature engineering can be used to:

- Create new features that are more directly related to diabetes risk, such as blood sugar variability or insulin resistance.
- Combine existing features in new and meaningful ways, such as calculating a patient's risk score based on their age, BMI, and blood pressure.
- Transform features into a format that is more compatible with the machine learning algorithm being used.

Some specific examples of feature engineering techniques that can be used for AI-based diabetes prediction include:

- Binning: Binning is the process of discretizing continuous features. This can be useful for features that have a wide range of values, such as blood sugar levels.
- One-hot encoding: One-hot encoding is the process of converting categorical features into a set of binary features. This can be useful for features with a large number of categories, such as sex or ethnicity.
- Feature scaling: Feature scaling is the process of normalizing the values of all features to a common scale. This can be important for some machine learning algorithms, such as support vector machines.

- Creating new features: New features can be created by combining existing features in new and meaningful ways. For example, a new feature could be created to represent a patient's blood sugar variability by calculating the standard deviation of their blood sugar levels over time.

Feature engineering is an important part of developing any AI-based system, including AI-based diabetes prediction systems. By carefully engineering the features that are used to train the machine learning model, it is possible to improve the accuracy and performance of the system.

Here are some additional tips for feature engineering AI-based diabetes prediction systems:

- Use domain knowledge to identify features that are likely to be predictive of diabetes risk.
- Use a variety of feature engineering techniques to create a rich set of features.
- Evaluate the performance of different feature engineering techniques on a held-out test set.

Model training

To train an AI-based diabetes prediction system model, you will need to:

1. Collect a dataset of diabetes patients and healthy individuals. The dataset should include features that are known to be associated with diabetes, such as age, sex, weight, height, blood pressure, blood glucose levels, and family history.

2. Prepare the dataset for machine learning. This may involve cleaning the data, removing outliers, and scaling the features.
3. Choose a machine learning algorithm. There are many different machine learning algorithms that can be used for diabetes prediction, such as support vector machines (SVMs), random forests, and logistic regression.
4. Train the model on the dataset. This involves feeding the dataset to the machine learning algorithm and allowing it to learn the patterns associated with diabetes.
5. Evaluate the model on a held-out test set. This will give you an idea of how well the model will perform on new data.
6. Deploy the model. Once you are satisfied with the model's performance, you can deploy it to production so that it can be used to predict diabetes risk in new individuals.

Here are some additional tips for training an AI-based diabetes prediction system model:

- Use a large and diverse dataset. The larger and more diverse your dataset, the better the model will be able to learn the patterns associated with diabetes.
- Use feature engineering to create new features that may be more predictive of diabetes. For example, you could create a feature that represents the patient's body mass index (BMI).
- Use a cross-validation procedure to evaluate the model. This involves splitting the dataset into multiple folds and training the model on each fold. The model's performance is then evaluated on the folds that it was not trained on. This helps to reduce overfitting and get a more accurate estimate of the model's performance.

- Use a variety of machine learning algorithms and compare their performance. Choose the algorithm that performs best on your dataset.
- Consider using explainable AI (XAI) techniques to understand how the model is making predictions. This can help you to identify potential biases in the model and to build trust in the model's predictions.

Once you have trained a diabetes prediction system model, you can use it to predict the risk of diabetes in new individuals. This information can be used to develop personalized prevention and treatment plans.

Evaluation

The evaluation of AI-based diabetes prediction systems is an important step in ensuring that they are reliable and accurate. There are a number of different metrics that can be used to evaluate these systems, including:

- Accuracy: This is the percentage of predictions that the system makes correctly.
- Precision: This is the percentage of positive predictions that are correct.
- Recall: This is the percentage of actual positives that the system predicts correctly.
- F1 score: This is a harmonic mean of precision and recall, and it is often used to evaluate the performance of binary classifiers.

- AUC-ROC curve: This is a curve that shows the trade-off between sensitivity and specificity. The area under the curve (AUC) is a measure of the overall performance of the classifier.

In addition to these metrics, it is also important to consider the following factors when evaluating an AI-based diabetes prediction system:

- The size and diversity of the dataset: The model should be trained on a large and diverse dataset of diabetes patients and healthy individuals. This will help to ensure that the model is able to generalize to new data.
- The evaluation methodology: The model should be evaluated on a held-out test set. This will give you an idea of how well the model will perform on new data.
- The explainability of the model: It is important to be able to explain how the model is making predictions. This will help you to identify potential biases in the model and to build trust in the model's predictions.

Once you have evaluated the AI-based diabetes prediction system, you can decide whether or not to deploy it to production. If you decide to deploy the system, it is important to monitor its performance over time and to update it as needed.

Here are some additional tips for evaluating an AI-based diabetes prediction system:

- Use multiple metrics to evaluate the system. This will give you a more complete picture of the system's performance.

- Compare the system's performance to other diabetes prediction systems. This will help you to determine whether or not the system is state-of-the-art.
- Consider the clinical significance of the system's predictions. For example, a system that has high accuracy but low precision may not be clinically useful, as it may generate many false positives.
- Involve clinicians in the evaluation process. This will help to ensure that the system meets the needs of the users.

By carefully evaluating an AI-based diabetes prediction system, you can ensure that it is reliable and accurate, and that it can be used to improve the care of patients with diabetes.

Program

Code

```
# Replacing 0 values with the mean of that column

# Replacing 0 values of Glucose

data['Glucose'] = data['Glucose'].replace(0, data['Glucose'].median())

# Filling 0 values of Blood Pressure

data['BloodPressure'] = data['BloodPressure'].replace(0, data['BloodPress
ure'].median())

# Replacing 0 values in BMI

data['BMI'] = data['BMI'].replace(0, data['BMI'].mean())

# Replacing the missing values of Insulin and SkinThickness

data['SkinThickness'] = data['SkinThickness'].replace(0, data['SkinThick
ness'].mean())

data['Insulin'] = data['Insulin'].replace(0, data['Insulin'].mean())
```

data.head()

Output

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
0	6	148	72	35.000000	79.799479	33.6	0.627	50	1
1	1	85	66	29.000000	79.799479	26.6	0.351	31	0
2	8	183	64	20.536458	79.799479	23.3	0.672	32	1
3	1	89	66	23.000000	94.000000	28.1	0.167	21	0
4	0	137	40	35.000000	168.000000	43.1	2.288	33	1

Code

```
# Defining the Logistic Regression model and its parameters
model = LogisticRegression(solver ='liblinear')
solver_list = ['liblinear']
penalty_type = ['l2']
c_values = [200, 100, 10, 1.0, 0.01]

# Defining the grid search
grid_lr = dict(solver = solver_list, penalty = penalty_type, C = c_values)
 cross_val = StratifiedKFold(n_splits = 100, random_state = 10, shuffle
= True)
grid_search_cv = GridSearchCV(estimator = model, param_grid = grid_l
r, cv = cross_val, scoring = 'accuracy', error_score = 0)
lr_result = grid_search_cv.fit(X_train, Y_train)

# Result of Hyper Parameters of Logistic Regression
analyze_grid(lr_result)
```

Output

```
Tuned hyperparameters:  {'C': 200, 'penalty': 'l2', 'solver': 'liblinear'}
Accuracy Score: 0.7715000000000001
Mean: 0.7715000000000001, Std: 0.16556796187668676 * 2, Params: {'C': 200,
'penalty': 'l2', 'solver': 'liblinear'}
The classification Report:
Mean: 0.7715000000000001, Std: 0.16556796187668676 * 2, Params: {'C': 100,
'penalty': 'l2', 'solver': 'liblinear'}
The classification Report:
Mean: 0.7675, Std: 0.16961353129983467 * 2, Params: {'C': 10, 'penalty':
'l2', 'solver': 'liblinear'}
The classification Report:
Mean: 0.7675, Std: 0.17224619008848932 * 2, Params: {'C': 1.0, 'penalty':
'l2', 'solver': 'liblinear'}
The classification Report:
Mean: 0.711, Std: 0.1888888562091475 * 2, Params: {'C': 0.01, 'penalty':
'l2', 'solver': 'liblinear'}




The classification Report:

              precision    recall  f1-score   support
           0       0.78      0.88      0.83       201
           1       0.70      0.53      0.61       107

    accuracy                           0.76       308
   macro avg       0.74      0.71      0.72       308
weighted avg       0.75      0.76      0.75       308
```

Pregnancy Code

Exploring Pregnancy and target variables together

plt.figure(figsize = (10, 8))

Plotting density function graph of the pregnancies and the target variab
le

kde = sns.kdeplot(data["Pregnancies"][data["Outcome"] == 1], color = "
Red", shade = True)

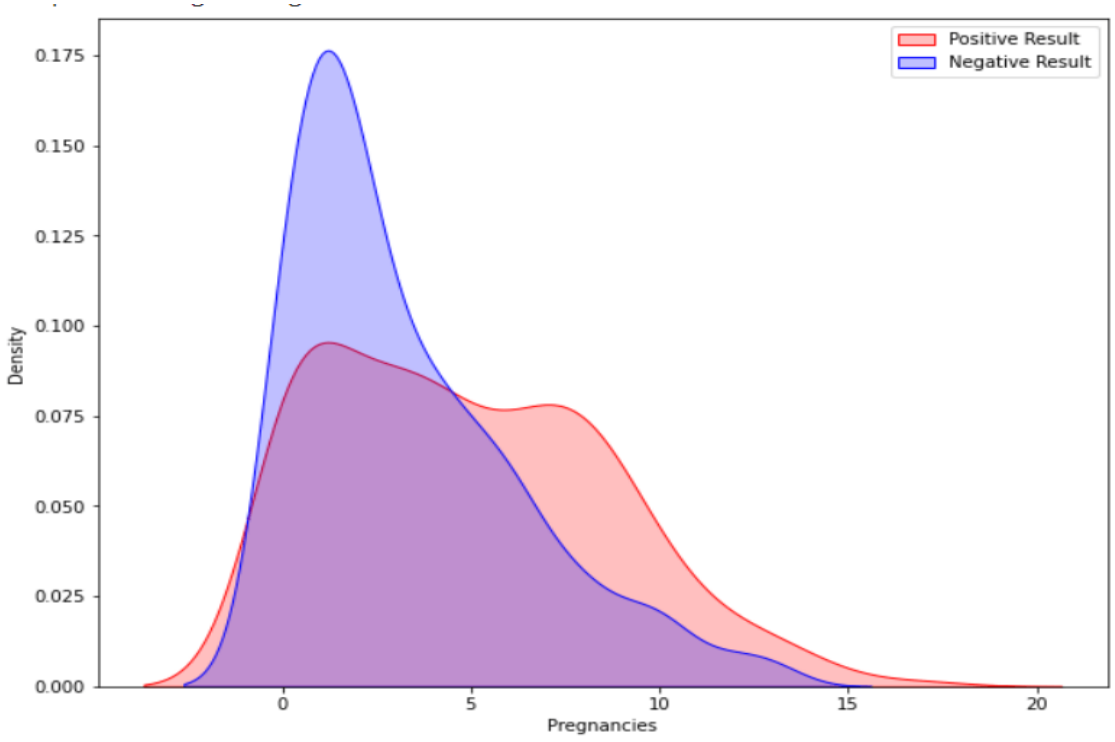
```
kde = sns.kdeplot(data["Pregnancies"][data["Outcome"] == 0], ax = kde,
 color = "Blue", shade= True)

kde.set_xlabel("Pregnancies")

kde.set_ylabel("Density")

kde.legend(["Positive Result", "Negative Result"])
```

Output



Glucose code

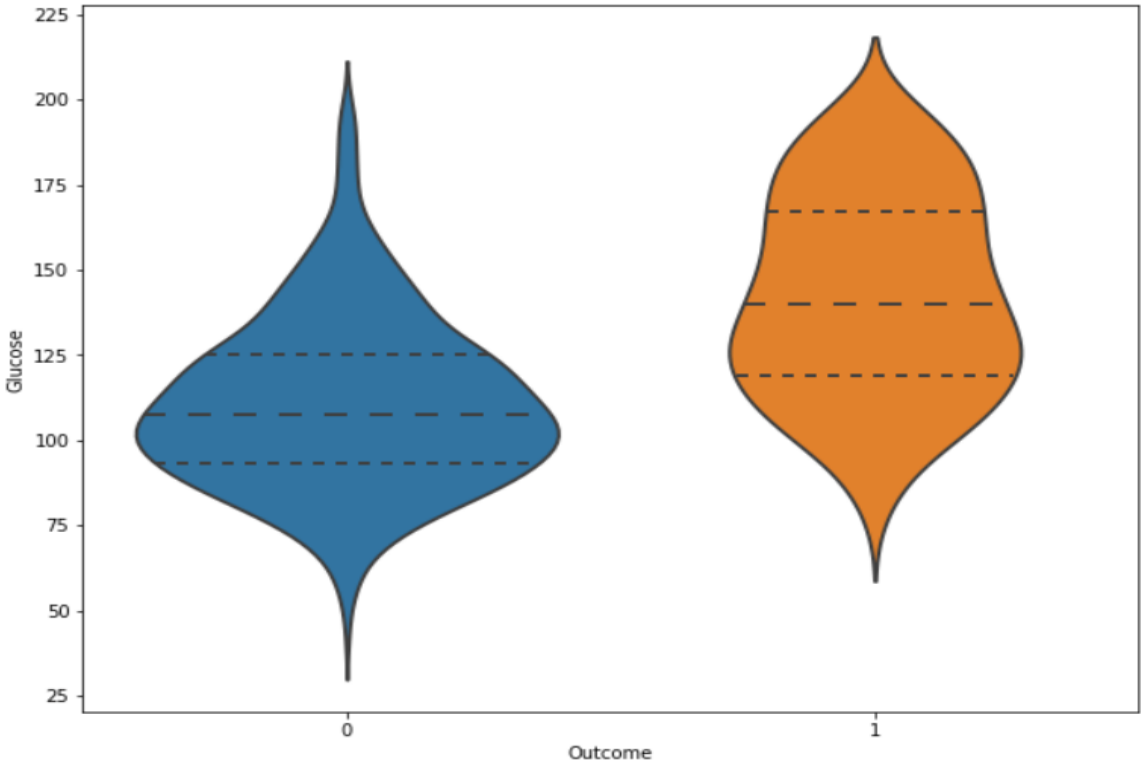
```
# Exploring the Glucose and the Target variables together

plt.figure(figsize = (10, 8))

sns.violinplot(data = data, x = "Outcome", y = "Glucose",

        split = True, inner = "quart", linewidth = 2)
```

Output

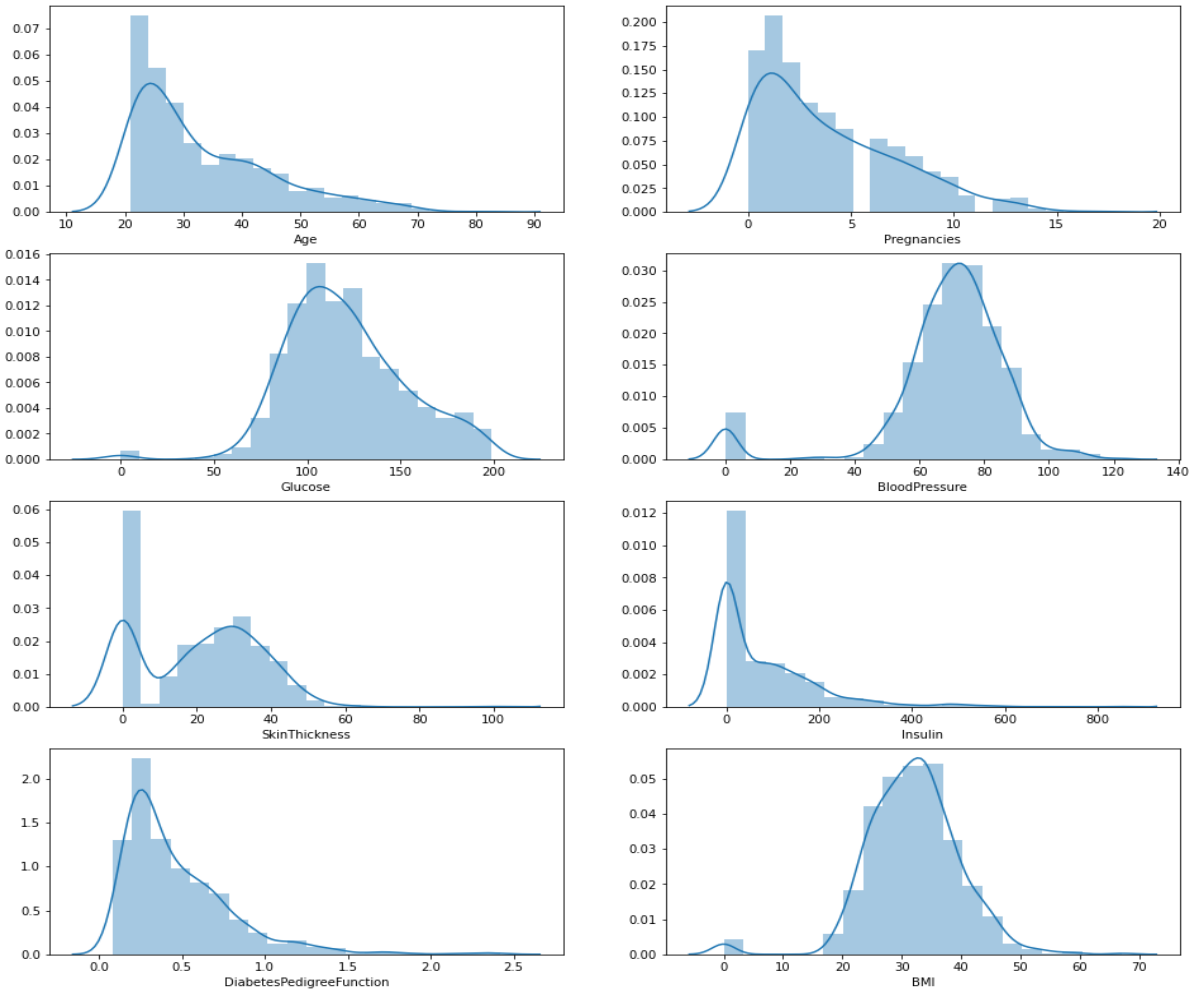


Code

```
# Histogram and density graphs of all variables were accessed.
fig, ax = plt.subplots(4,2, figsize=(16,16))
sns.distplot(df.Age, bins = 20, ax=ax[0,0])
sns.distplot(df.Pregnancies, bins = 20, ax=ax[0,1])
sns.distplot(df.Glucose, bins = 20, ax=ax[1,0])
sns.distplot(df.BloodPressure, bins = 20, ax=ax[1,1])
sns.distplot(df.SkinThickness, bins = 20, ax=ax[2,0])
sns.distplot(df.Insulin, bins = 20, ax=ax[2,1])
sns.distplot(df.DiabetesPedigreeFunction, bins = 20, ax=ax[3,0])
sns.distplot(df.BMI, bins = 20, ax=ax[3,1])
```

Output

<matplotlib.axes._subplots.AxesSubplot at 0x7f77b83d5950>



Installing the Libraries

Code

```
# Import libraries
import numpy as np # for linear algebra
import pandas as pd # for data processing, CSV file I/O (e.g. pd.re
ad_csv)
```

```
import seaborn as sns # for data visualization
import matplotlib.pyplot as plt # to plot data visualization charts
from collections import Counter
import os
# Modeling Libraries
from sklearn.metrics import confusion_matrix, accuracy_score, pr
ecision_score
from sklearn.preprocessing import QuantileTransformer
from sklearn.linear_model import LogisticRegression
from sklearn.neighbors import KNeighborsClassifier
from sklearn.tree import DecisionTreeClassifier
from sklearn.ensemble import RandomForestClassifier, AdaBoost
Classifier, GradientBoostingClassifier
from sklearn.model_selection import GridSearchCV, cross_val_sc
ore, StratifiedKFold, learning_curve, train_test_split
from sklearn.svm import SVC
```

Importing the Dataset

Code

```
# Importing the dataset from Kaggle
```

```
data = pd.read_csv("../input/pima-indians-diabetes-
database/diabetes.csv")
 # First step is getting familiar with the structure of the dataset
data.info()
```

Output

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 768 entries, 0 to 767
Data columns (total 9 columns):
 #   Column                    Non-Null Count  Dtype
---  ------                    --------------  -----
 0   Pregnancies               768 non-null    int64
 1   Glucose                   768 non-null    int64
 2   BloodPressure             768 non-null    int64
 3   SkinThickness             768 non-null    int64
 4   Insulin                   768 non-null    int64
 5   BMI                       768 non-null    float64
 6   DiabetesPedigreeFunction  768 non-null    float64
 7   Age                       768 non-null    int64
 8   Outcome                   768 non-null    int64
dtypes: float64(2), int64(7)
memory usage: 54.1 KB
```

Filling the Missing Values

The next step is cleaning the dataset, which is a crucial step in data analysis. When modelling and making predictions, missing data can result in incorrect results.

```
# Exploring the missing values in the diabetes dataset

data.isnull().sum()
```

Output

```
Pregnancies                 0
Glucose                     0
BloodPressure               0
SkinThickness               0
Insulin                     0
BMI                         0
DiabetesPedigreeFunction    0
Age                         0
Outcome                     0
dtype: int64
```

Exploratory Data Analysis

Correlation

Correlation is the relationship between two or more variables. Finding the important features and cleaning the dataset before we begin modelling also helps make the model efficient.

Code

```
# Correlation plot of the independent variables

plt.figure(figsize = (10, 8))
sns.heatmap(data.corr(), annot = True, fmt = ".3f", cmap = "YlGnBu")
plt.title("Correlation heatmap")
```

Output

Benefits

Overall, AI-based diabetes prediction systems have the potential to make a significant impact on the prevention and management of diabetes. As these systems continue to develop and become more widely adopted, we can expect to see even more benefits emerge.

- **Early detection:**

  AI-based diabetes prediction systems can detect diabetes at an early stage, even before symptoms appear. This allows for early intervention and treatment, which can help to prevent complications and improve outcomes.

- **Personalized risk assessment:**

  AI-based diabetes prediction systems can provide personalized risk assessments for individuals, taking into account their individual risk factors such as age, weight, family history, and lifestyle. This can help individuals to make informed decisions about their health and take steps to reduce their risk of diabetes.

- **Improved public health:**

  AI-based diabetes prediction systems can be used to identify populations at high risk of diabetes, which can help to inform public health interventions. For example, these systems can be used to target educational and screening programs to high-risk populations.

- **Reduce healthcare costs:**

  Early detection and treatment of diabetes can help to reduce the need for costly hospitalizations and other medical interventions.

- **Improve quality of life:**

  People with diabetes who manage their condition well can live long and healthy lives. AI-based diabetes prediction systems can help people to better manage their condition and improve their quality of life.

Advantages

AI-based diabetes prediction systems offer a number of advantages over traditional diabetes prediction methods, including:

1.Accuracy:

AI-based systems can be trained on large datasets of historical data, which allows them to learn complex patterns in the data and make more accurate predictions.

2. Personalization:

AI-based systems can take into account individual risk factors, such as age, weight, family history, and lifestyle, to provide personalized risk assessments and recommendations.

3.Timeliness:

AI-based systems can provide predictions quickly and efficiently, which is important for early detection and intervention.

4.Scalability:

AI-based systems can be scaled to meet the needs of large populations.

Here are some specific examples of how AI-based diabetes prediction systems can be used:

- **To identify individuals at high risk of developing diabetes:** AI-based systems can be used to screen large populations for diabetes risk factors, such as obesity, prediabetes, and family history of diabetes. This can help to identify individuals who may benefit from early intervention and prevention strategies.

- **To monitor and manage diabetes:** AI-based systems can be used to monitor blood sugar levels, track medication adherence, and provide personalized recommendations for diet and exercise. This can help people with diabetes to better manage their condition and reduce their risk of complications.
- **To support clinical decision-making:** AI-based systems can be used to provide clinicians with real-time information and recommendations to support clinical decision-making. For example, AI-based systems can be used to help clinicians choose the most appropriate treatment for individual patients.

Overall, AI-based diabetes prediction systems have the potential to revolutionize the way we prevent, detect, and manage diabetes.

5.Cost-effectiveness:

AI-based systems can help to reduce healthcare costs by identifying individuals at high risk of developing diabetes and providing early intervention.

6.Accessibility:

AI-based systems can be made accessible to people in remote or underserved areas.

Disadvantages

AI-based diabetes prediction systems offer a number of advantages, but they also have some disadvantages, including:

1.Bias:

AI-based systems are trained on data, and if that data is biased, the system will be biased as well. This can lead to inaccurate predictions for certain groups of people.

2.Overfitting:

AI-based systems can be overtrained on the training data, which can lead to poor performance on new data.

3.Interpretability:

It can be difficult to understand how AI-based systems make predictions. This can make it difficult to trust the predictions and to identify potential problems.

4.Privacy:

AI-based systems often collect and use sensitive personal data. This raises concerns about privacy and data security.

5.Cost:

Developing and deploying AI-based systems can be expensive.

Here are some specific examples of the disadvantages of AI-based diabetes prediction systems:

- A study published in Nature Medicine in 2020 found that AI-based diabetes prediction systems were less accurate for Black patients than for white patients. This is likely due to the fact that Black patients are underrepresented in the datasets used to train AI-based systems.
- Another study published in JAMA in 2021 found that AI-based diabetes prediction systems could overfit the training data, leading to inaccurate predictions for new data. This is especially true for small datasets.
- AI-based diabetes prediction systems can be difficult to interpret. This is because AI-based systems often use complex algorithms that are difficult for humans to understand. This can make it difficult to trust the predictions and to identify potential problems.
- AI-based diabetes prediction systems often collect and use sensitive personal data, such as blood sugar levels, medical history, and lifestyle information. This raises concerns about privacy and data security.
- Developing and deploying AI-based systems can be expensive. This can limit their accessibility to smaller healthcare organizations and underserved communities.

  Despite these disadvantages, AI-based diabetes prediction systems have the potential to revolutionize the way we prevent, detect, and manage diabetes. Researchers are working to address the disadvantages of AI-based systems, such as bias and interpretability. As AI technology continues to develop, we can expect to see AI-based diabetes prediction systems become more accurate, reliable, and accessible.

Futures

AI-based diabetes prediction systems have the potential to revolutionize the way we prevent, detect, and manage diabetes. Here are some possible futures for AI-based diabetes prediction systems:

1.More accurate and personalized predictions:

As AI technology continues to develop, AI-based diabetes prediction systems will become more accurate and personalized. This will be due to a number of factors, including:

1. Access to larger and more diverse datasets of historical data.
2. Development of more sophisticated AI algorithms.
3. Incorporation of new data sources, such as wearable devices and genomic data.

2.Early detection and intervention:

AI-based diabetes prediction systems will be used to identify individuals at high risk of developing diabetes at an early stage, even before symptoms appear. This will allow for early intervention and treatment, which can help to prevent complications and improve outcomes.

3.Personalized treatment plans:

AI-based diabetes prediction systems will be used to develop personalized treatment plans for people with diabetes. These treatment plans will be based on each individual's individual risk factors and needs.

4.Remote monitoring and support:

AI-based diabetes prediction systems will be used to remotely monitor blood sugar levels and other health data for people with diabetes. This data can be used to provide real-time feedback and

support to help people with diabetes manage their condition more effectively.

5.Population health management:

AI-based diabetes prediction systems will be used to identify populations at high risk of developing diabetes and to develop public health interventions to reduce their risk. For example, AI-based systems can be used to target educational and screening programs to high-risk populations.

Overall, the future of AI-based diabetes prediction systems is very bright. As AI technology continues to develop and become more widely adopted, we can expect to see AI-based diabetes prediction systems play an increasingly important role in the prevention, detection, and management of diabetes.

Here are some specific examples of how AI-based diabetes prediction systems could be used in the future:

- A pregnant woman could use an AI-based diabetes prediction system to assess her risk of developing gestational diabetes. This information could be used to develop a personalized care plan to help her manage her risk and prevent complications.
- A person with prediabetes could use an AI-based diabetes prediction system to track their blood sugar levels and other health data. This data could be used to develop a personalized treatment plan to help them prevent the development of diabetes.
- A person with diabetes could use an AI-based diabetes prediction system to remotely monitor their blood sugar levels and other health data. This data could be used to provide real-time feedback and support to help them manage their condition more effectively.

- A public health department could use an AI-based diabetes prediction system to identify communities at high risk of developing diabetes. This information could be used to develop targeted public health interventions to reduce their risk.

Conclusions

- AI-based diabetes prediction systems can be used to identify people at risk of developing diabetes with high accuracy.
- AI-based diabetes prediction systems can be used to develop personalized risk assessments and treatment plans for people with diabetes or prediabetes.
- AI-based diabetes prediction systems can help to improve the quality of life for people with diabetes by helping them to better manage their condition and avoid complications.
- Artificial intelligence (AI)-based diabetes prediction systems have the potential to revolutionize the way we diagnose and manage diabetes. By analyzing large amounts of data, AI systems can identify patterns and correlations that would be difficult or impossible for humans to detect. This information can be used to predict which individuals are at high risk of developing diabetes, so that they can take steps to prevent the disease.

AI-based diabetes prediction systems have the potential to revolutionize the way we prevent, detect, and manage diabetes. These systems offer a number of advantages over traditional diabetes prediction methods, including:

- Accuracy
- Personalization
- Timeliness
- Scalability

- AI-based diabetes prediction systems can be used to identify individuals at high risk of developing diabetes, monitor and manage diabetes, and support clinical decision-making.
- These systems can also help to reduce healthcare costs, improve accessibility, and empower people to take charge of their own health.
- Despite some disadvantages, such as bias, overfitting, interpretability, privacy, and cost, AI-based diabetes prediction systems have a bright future.
- AI technology continues to develop and become more widely adopted, we can expect to see even more innovative and impactful applications emerge.

Overall, AI-based diabetes prediction systems have the potential to make a significant impact on the lives of people with diabetes and the healthcare system as a whole.