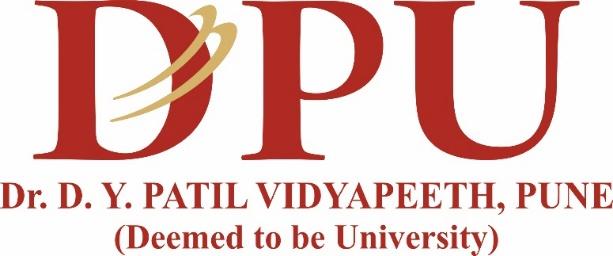
****

**MEDICAL DIAGNOSIS**

**DR. D. Y. PATIL SCHOOL OF SCIENCE AND TECHNOLOGY**

**TATHAWADE, PUNE**

**A Mini- Project Report on**

**MEDICAL DIAGNOSIS WITH MACHINE LEARNING**

**SUBMITTED BY:**

**NAME OF STUDENT ROLL NUMBER**

**1. PUNIT JAIN BTAI - 2314**

**2. JANHVI PATIL BTAI - 2315**

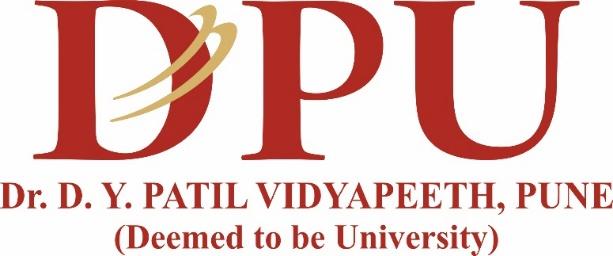
**3. SAHIL SINGH BTAI - 2344**

**GUIDED BY:**

**Dr. Mily Lal**

**ARTIFICIAL INTELLIGENCE & DATA SCIENCE**

**ACADEMIC YEAR 2024-2025**

****

**DR. D. Y. PATIL SCHOOL OF SCIENCE AND TECHNOLOGY**

**TATHAWADE, PUNE**

**CERTIFICATE**

**This is to certify that the Mini- Project Report entitled**

**MEDICAL DIAGNOSIS WITH MACHINE LEARNING**

is a bonafide work carried out by **Mr. PUNIT JAIN** under the supervision of Mrs. Mili Lal and it is submitted towards the partial fulfillment of the requirement of **Fundamentals of Data Science.**

**Dr. Mily Lal**  **Dr. Manisha Bhende**

**Project Guide Director I/C**

**ARTIFICIAL INTELLIGENCE & DATASCIENCE**

**ACADEMIC YEAR 2024-2025**

**ABSTRACT**

Medical assessments are physician-dependent, and human error or limitations can impact accuracy. The project’s goal is to develop machine learning models designed to analyze patient data for diagnostic purposes. Using advanced algorithms, the model will be trained to predict medical conditions with the goal of improving the accuracy and availability of diagnostic equipment, especially in underserved areas. This approach can reduce the risk of human error, increase diagnostic accuracy, and improve clinical decision-making. Future work will include integrating this methodology with clinical data and expanding its applicability to a variety of clinical settings. The project aims to harness machine learning models for enhanced diagnostic accuracy by analyzing patient data, addressing the limitations of physician-dependent assessments. By employing advanced algorithms, these models will predict medical conditions, significantly reducing the risk of human error and improving diagnostic reliability, particularly in underserved areas. This approach not only increases the availability of accurate diagnostic tools but also enhances clinical decision-making. Future efforts will focus on integrating these models with diverse clinical data, broadening their applicability across various healthcare settings to further support clinicians and improve patient outcomes.

**Keywords**: Medical diagnosis, machine learning, predictive modeling, healthcare automation, K-Nearest Neighbors.

**TABLE OF CONTENTS**

|  |  |  |
| --- | --- | --- |
| CHAPTER NO. | TOPIC | PAGE NO |
| 1 | INTRODUCTION | 6 |
| 1.1 Problem Statement | 7 |
| 1.2 Objectives | 7 |
|  | 1.3 Scope | 8 |
|  | 1.4 System Architecture | 9-10 |
| 2 | DATA COLLECTION & PREPROCESSING | 11 |
| 2.1 Dataset | 11 |
| 2.2 Data Preprocessing | 12 |
| 3 | MODEL SELECTION & TRAINING | 13 |
|  | 3.1 Feature Engineering | 13 |
|  | 3.2 Machine Learning Model | 21 |
| 4 | MODEL EVALUATION & VALIDATION | 27 |
|  | 4.1 Performance Matrix | 27 |
| 5 | CONCLUSION | 34 |
|  | 5.1 Conclusion | 34 |
|  | 5.2 Future Scope | 34 |
| 6 | REFERENCES | 35 |
|  | 6.1 References | 35-36 |

|  |  |  |
| --- | --- | --- |
| **SR.NO** | **FIGURE NO.** | **PAGE NO** |
| **1** | SYSTEM ARCHITECTURE | **10** |
| **2** | DATASET | **11** |
| **3** | MISSING VALUES | **13** |
| **4** | REPLACIG MISING VALUES USING MEAN, MEDIAN | **14** |
| **5** | APPLYING STANDARDSCAKER AND PIPELINE | **15** |
| **6** | TRAINING SCORES | **16** |
| **7** | TESTING SCORES | **17** |
| **8** | BOXPLOT | **17** |
| **9** | SCATTERPLOT | **19** |
| **10** | HEATMAP | **20** |
| **11** | SGD CLASSIFIER | **21** |
| **12** | LOGISTIC REGRESSION | **22** |
| **13** | SVM | **23** |
| **14** | SGD CLASSIFIER MODEL EVALUATION | **24** |
| **15** | SGD CLASSIFIER MODEL ACCURACY | **24** |
| **16** | LOGISTIC REGRESSION MODEL EVALUATION | **24** |
| **17** | LOGISTIC REGRESSION MODEL ACCURACY | **25** |
| **18** | SVM MODEL EVALUATION | **25** |
| **19** | SVM MODEL ACCURACY | **26** |
| **20** | HEATMAP | **29** |
| **21** | MODEL COMPARISON | **31** |
|  |  |  |
|  |  |  |
|  |  |  |

**LIST OF FIGURES**

**LIST OF TABLES**

|  |  |  |
| --- | --- | --- |
| **SR NO.** | **TABLE NO.** | **PAGE NO.** |
| **1** | MODEL COMPARISON | **27** |
| **2** | STACKING/VOTING | **33** |

**CHAPTER 1**

**INTRODUCTION**

Medical diagnosis identifies the disease or conditions that explain a person’s symptoms and signs. Typically, diagnostic information is gathered from the patient’s history and physical examination[1]. It is frequently difficult due to the fact that many indications and symptoms are ambiguous and can only be diagnosed by trained health experts[2]. Therefore, countries that lack enough health professionals for their populations, such as developing countries like Bangladesh and India, face difficulty providing proper diagnostic procedures for their maximum population of patients [3]. Traditional diagnosis processes are costly, time-consuming, and often require human intervention[4]. While the individual’s ability restricts traditional diagnosis techniques, ML-based systems have no such limitations, and machines do not get exhausted as humans do[5]. As a result, a method to diagnose disease with outnumbered patients’ unexpected presence in health care may be developed.[6] In contrast, machine learning (ML)-based diagnostic systems offer a promising alternative. These systems can process vast amounts of data quickly and efficiently, without the limitations that human practitioners face, such as fatigue and time constraints[7]. By leveraging ML technology, it becomes possible to develop innovative diagnostic solutions that can accommodate the high volume of patients in resource-limited settings[8]. This shift toward automated and intelligent diagnostic approaches holds the potential to significantly improve healthcare accessibility and quality in underserved regions[9].

**1.1 – PROBLEM STATEMENT**

Accurate medical diagnosis remains one of the most challenging aspects of healthcare, often hindered by human error, variability in physician assessments, and limited access to specialists. To address this, the project aims to develop machine learning models that enhance diagnostic accuracy by analyzing patient data, thereby reducing dependence on subjective evaluations and increasing the availability of reliable diagnostic tools.

**1.2 – OBJECTIVE**

* **Loading and Finding the Data from Dataset :-**

Load the diabetes dataset and find the information about the dataset like no. of null values, type of data, etc.

* **Feature Scaling :-**

Standardize the feature values using StandardScaler to ensure that they are on the same scale for the use in prediction.

* **Model Training :-**

Train a model using KNN, LR, SGD, SVM to predict and give output about the Diabetes Dataset which is to be taken.

* **Public Awareness :-**

Through a survey of Diabetes Program a public awareness in formed in peoples to also check , about the Diabetes and other diseases.

* **Visualization of Data :-**

Visualize the data through graphs[histogram, bar graph, etc.] and predict the relationship between different measures like BMI, Blood Group, Glucose Level, etc.

**1.3 – SCOPE**

This project focuses on the development and evaluation of machine learning models for predicting diabetes based on medical diagnostic data. The scope includes data preprocessing, exploratory data analysis, feature scaling, model training, hyperparameter tuning, and evaluation using various algorithms such as Logistic Regression, Support Vector Machine (SVM), Stochastic Gradient Descent (SGD), and K-Nearest Neighbors (KNN).

**LIMITATIONS –**

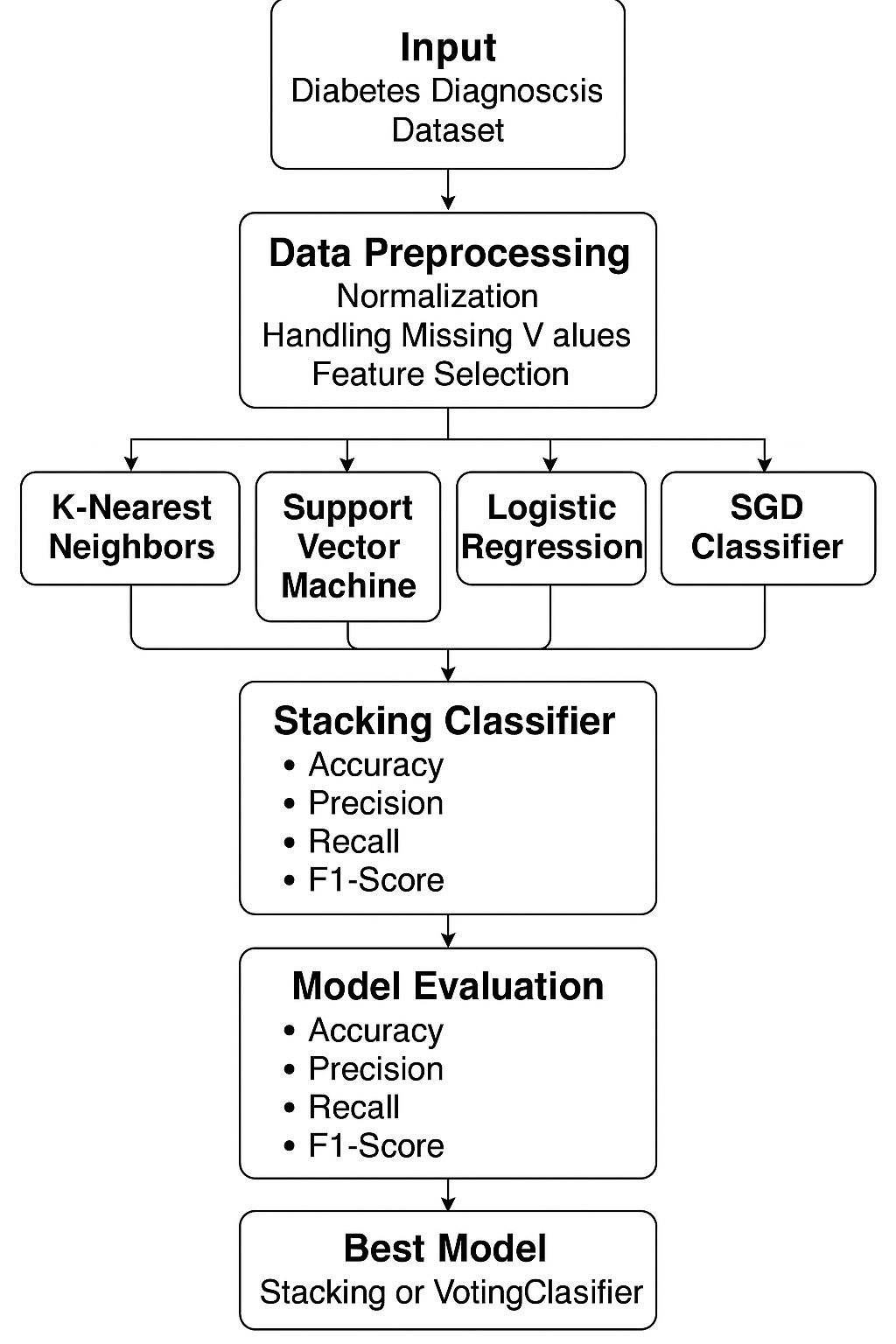
The dataset used is static and limited to the Pima Indians Diabetes Database, which may not generalize to broader populations with different genetic, demographic, or lifestyle profiles. Real-time prediction using user input is simulated and does not include validation against a live medical diagnosis. The project does not incorporate time-series data or continuous monitoring, which are increasingly relevant in modern healthcare diagnostics.

**INTENDED AUDIENCE –**

* Data Science Students and Practitioners : To understand the implementation of multiple machine learning models for binary classification problems.
* Medical Researchers and Healthcare Analysts : Interested in exploring how machine learning can assist in early diagnosis of diabetes.
* Educators and Trainers : Who require a comprehensive example for teaching data preprocessing, visualization, and classification techniques.
* Developers : Seeking to build or extend predictive healthcare applications using standard Python libraries and machine learning frameworks.

**1.4 – SYSTEM ARCHITECTURE**

The flowchart presents a machine learning pipeline for diabetes diagnosis. It begins with inputting the dataset, followed by preprocessing steps such as normalization, handling missing values, and feature selection. Multiple algorithms—K-Nearest Neighbors, Support Vector Machine, Logistic Regression, and SGD Classifier—are trained individually. Their outputs are then combined using a stacking classifier, evaluated based on accuracy, precision, recall, and F1-score. A final model evaluation helps determine the most effective model. The best-performing approach, either stacking or a voting classifier, is selected. This pipeline aims to enhance diagnostic accuracy and reduce physician-dependent errors using ensemble machine learning techniques.

****

**Fig No. :- 1 :- SYSTEM ARCHITECTURE**

**CHAPTER 2**

**DATA COLLECTION & PREPROCESSING**

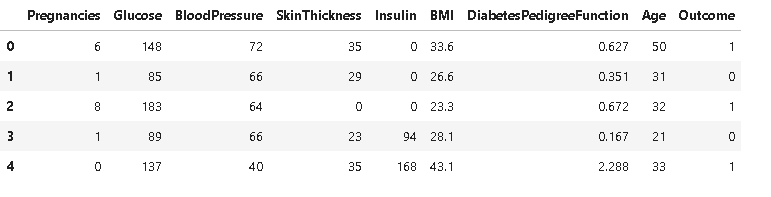
**2.1 – DATASET**

The dataset used in this project is the Pima Indians Diabetes Database, which is publicly available from the UCI Machine Learning Repository and also on KAGGLE. It contains medical records of female patients from the Pima Indian heritage, with attributes such as glucose levels, blood pressure, skin thickness, insulin levels, BMI, age, and family history of diabetes. The dataset is comprised of 768 samples, each representing an individual’s health metrics and diabetes status. The dataset's attributes, including glucose levels, BMI, and age, are known risk factors for diabetes, providing a rich source of information to train a machine learning model. Its relevance lies in the fact that it directly correlates to the condition being studied by making it ideal for testing the effectiveness of predictive algorithms in a healthcare context.

* **DATASET :-** Loading the dataset using .read\_csv function and displaying first 5 rows using .head() function.

**DIABETES\_df = pd.read\_csv('C:/Users/punit/Downloads/DIABETESDIAGNOSIS/DIABETES.csv')**

**DIABETES\_df.head()**

****

**Fig No. :- 2 :- DATASET**

**2.2 – DATA PREPROCESSING**

In this project, **data preprocessing** was essential to ensure accurate and reliable predictions. Initially, **missing or invalid values**—particularly zeros in features like **Glucose, BloodPressure, SkinThickness, Insulin, and BMI**—were identified and treated as nulls. These were **imputed using mean or median values** depending on their distribution. The dataset was **cleaned and visualized** using **histograms, boxplots, and scatter plots** to understand feature distributions and detect outliers. A **correlation matrix heatmap** helped assess relationships between features. The **target variable (Outcome)** was separated from the feature set, and the data was **split into training and testing sets with stratification** to preserve class balance. **Feature scaling** was applied using **StandardScaler, Min-Max Scaler, and Robust Scaler**, with StandardScaler used primarily for model training. A **pipeline** was built to **automate preprocessing and model fitting**, ensuring consistent transformation across cross-validation. These steps collectively prepared the dataset for **efficient and accurate modeling** using various classification algorithms.

**CHAPTER 3**

**MODEL SELECTION & TRAINING**

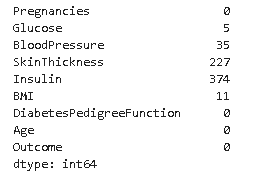
**3.1 – FEATURE ENGINEERING**

* **Finding Missing Values :** Finding missing values using .isnull().sum() function**.** Here we can see the null values by replacing zero with npNaN in each column of the dataset which will help us to change these values to make our dataset more efficient.

DIABETES\_df\_copy = DIABETES\_df.copy(deep = True)

DIABETES\_df\_copy[['Glucose', 'BloodPressure', 'SkinThickness', 'Insulin', 'BMI']] = DIABETES\_df\_copy[['Glucose', 'BloodPressure', 'SkinThickness', 'Insulin', 'BMI']].replace(0,np.NaN)

print(DIABETES\_df\_copy.isnull().sum())

****

**Fig No. :- 3 :- MISSING VALUES**

* **Replacing Missing Values using mean(), median() :** Replacing missing values using .mean(), .median(). By using mean and median we can simply fill those missing values which are affecting the dataset and the change will result in increment of the overall performance of the model.

DIABETES\_df\_copy['Glucose'].fillna(DIABETES\_df\_copy['Glucose'].mean(), inplace = True)

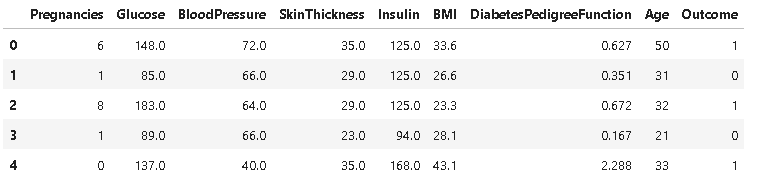
DIABETES\_df\_copy['BloodPressure'].fillna(DIABETES\_df\_copy['BloodPressure'].mean(), inplace = True)

DIABETES\_df\_copy['SkinThickness'].fillna(DIABETES\_df\_copy['SkinThickness'].median(), inplace = True)

DIABETES\_df\_copy['Insulin'].fillna(DIABETES\_df\_copy['Insulin'].median(), inplace = True)

DIABETES\_df\_copy['BMI'].fillna(DIABETES\_df\_copy['BMI'].median(), inplace = True)

DIABETES\_df\_copy.head()

****

**Fig No. :- 4 :- REPLACING MISSING VALUES WITH MEAN, MEDIAN**

* **Standard Scaler :** Defining training and testing data using StandardScaler and train\_test\_split. By using StandardScaler we ensure that all the data in the dataset must be in the same scale as every feature has its own range or scale so it is necessary to make all the features work properly, and further using train\_test\_split to get the testing data.

X = DIABETES\_df.drop(columns = 'Outcome', axis = 1)

y = DIABETES\_df['Outcome']

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X,y,test\_size = 1/3, random\_state = 42, stratify = y)

print(X.shape, X\_train.shape, X\_test.shape)

train\_df, test\_df = train\_test\_split(DIABETES\_df\_copy, test\_size=0.1, random\_state=42)

train\_df\_labels = train\_df["Outcome"].copy()

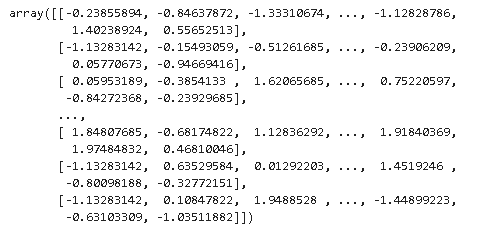
train\_df= train\_df.drop("Outcome", axis=1)

num\_pipeline = Pipeline([('std\_scaler', StandardScaler()), ])

num\_pipeline

train\_prepared = num\_pipeline.fit\_transform(train\_df)

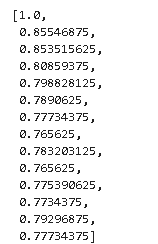
train\_prepared

****

**Fig No. :- 5 :- APPLYING STANDARDSCALER AND PIPELINE**

* **Training Scores :** Displaying training data. Here we can see the training scores of our dataset that we had excluded from train\_test\_split function.

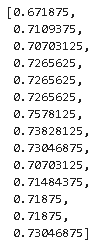
train\_scores

****

**Fig No. :- 6 :- TRAINING SCORES**

* **Testing Scores :** Displaying testing data. Here we can see the testing scores of our dataset that we had excluded from train\_test\_split function.

test\_scores

****

**Fig No. :- 7 :-TESTING SCORES**

* **DATA VISUALZATION :-**
* **BOXPLOT :-** Displaying boxplot and visualizing the outliers in the dataset. Boxplot is used to see the outliers occurring in the dataset. Here we can see that maximum no. f outliers are occurring in Insulin and it may affect the overall performance of the model. We can limit the Insulin range to see that is it affecting the performance or not.

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

df = DIABETES\_df

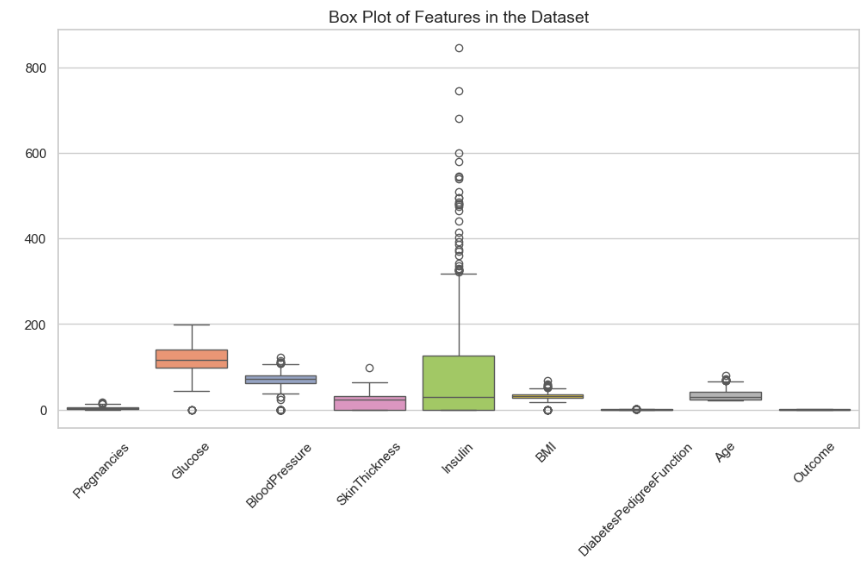
plt.figure(figsize=(12, 6))

sns.boxplot(data=df, palette="Set2")

plt.title("Box Plot of Features in the Dataset", fontsize=14)

plt.xticks(rotation=45)

plt.show()

****

**Fig No. :- 8 :- BOXPLOT**

* **SCATTERPLOT :-** Displaying scatterplot and visualizing the spread of Diabetic and Non-Diabetic patients. Scatterplot is used to see and analyze the spread of the outcome and also the outliers. Here we can see that out data is not spreaded in small clusters in different areas as this will affect the outcome. Our data is clustered in on area and the outer points are defining the outliers.

import pandas as pd

import matplotlib.pyplot as plt

import seaborn as sns

df = pd.read\_csv('C:/Users/punit/Downloads/DIABETESDIAGNOSIS/DIABETES.csv')

plt.figure(figsize=(8, 6))

sns.scatterplot(x=df["Glucose"], y=df["BMI"], hue=df["Outcome"], palette="coolwarm", alpha=1)

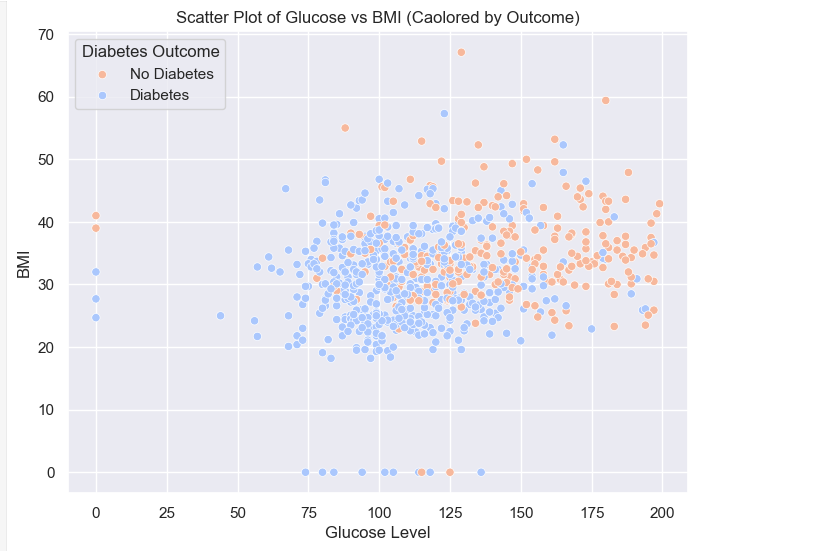
plt.xlabel("Glucose Level")

plt.ylabel("BMI")

plt.title("Scatter Plot of Glucose vs BMI (Caolored by Outcome)")

plt.legend(title="Diabetes Outcome", labels=["No Diabetes", "Diabetes"])

plt.show()

****

**Fig No. :- 9 :- SCATTERPLOT**

* **HEATMAP :-** Displaying heatmaps and visualizing the correlation between different features included in the prediction. Heatmap is used to analyze the correlation between the features of the dataset. Glucose, BMI, Age, and Pregnancies show the strongest positive correlation with diabetes outcome, with Glucose being the most significant. Most features have low inter-correlation, indicating minimal multicollinearity and good feature independence.

import pandas as pd

import seaborn as sns

import matplotlib.pyplot as plt

df =DIABETES\_df

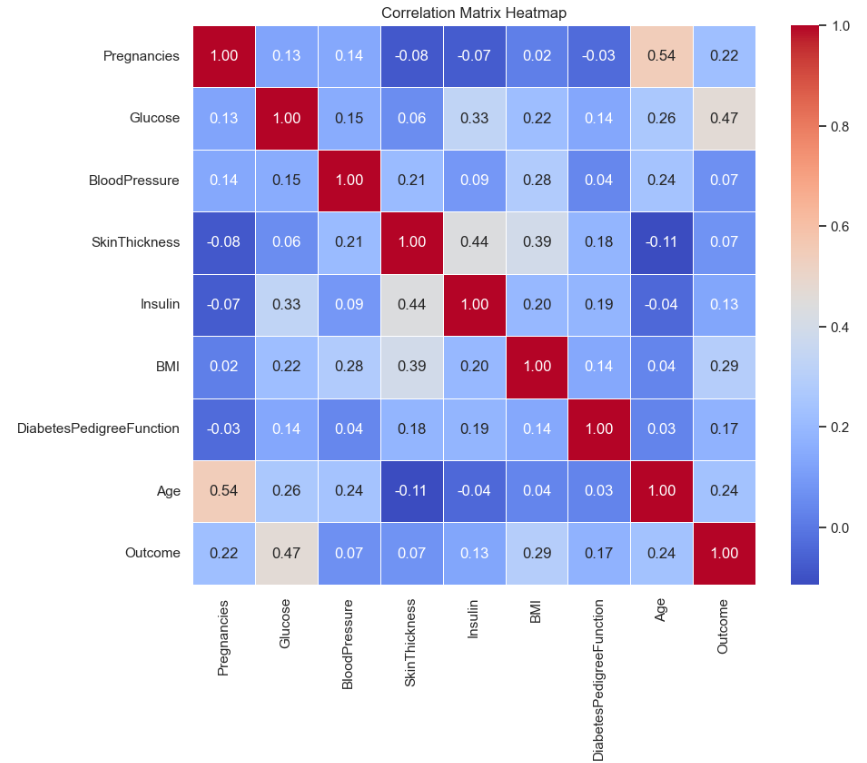
corr\_matrix = df.corr()

plt.figure(figsize=(10, 8)

sns.heatmap(corr\_matrix, annot=True, cmap="coolwarm", fmt=".2f", linewidths=0.5)

plt.title("Correlation Matrix Heatmap")

plt.show()

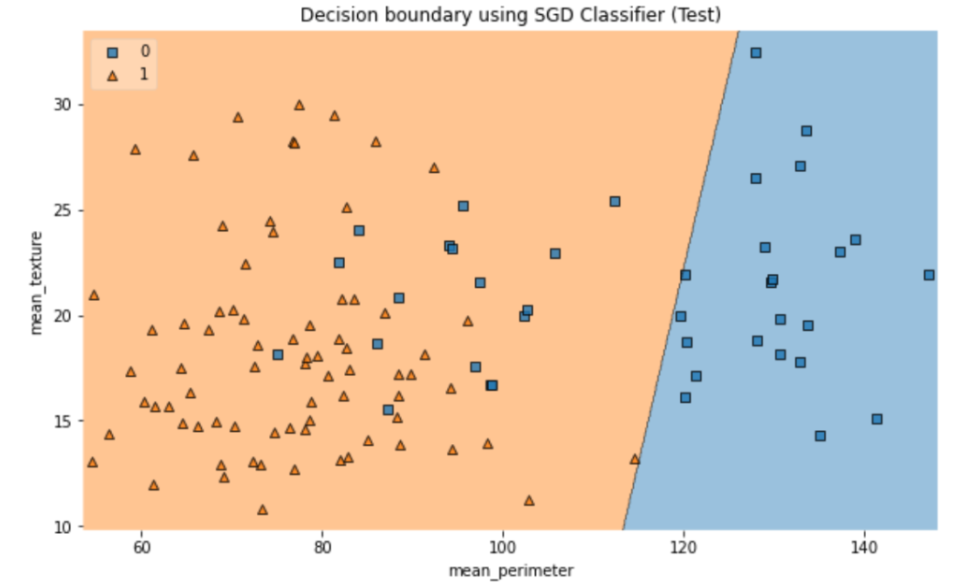
****

**Fig No. :- 10 :- HEATMAP**

**3.2 – MACHINE LEARNING MODEL**

* **SGD CLASSIFIER :-**

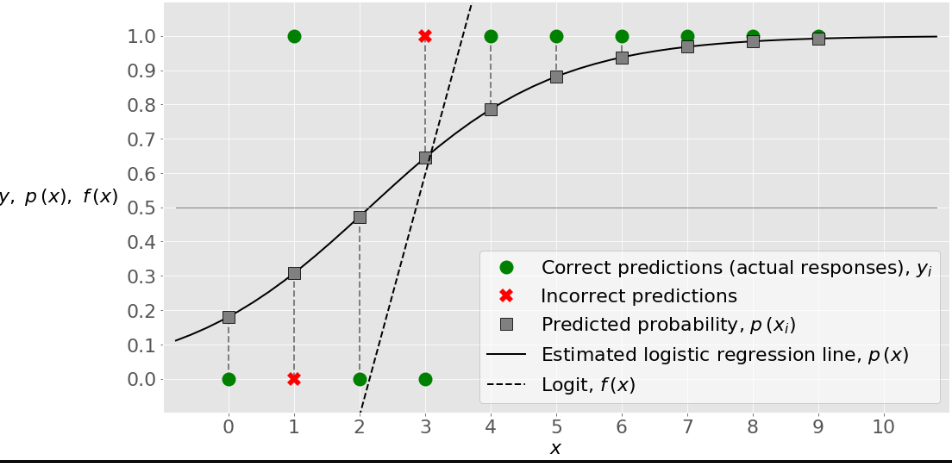
This estimator implements regularized linear models with stochastic gradient descent (SGD) learning: the gradient of the loss is estimated each sample at a time and the model is updated along the way with a decreasing strength schedule (aka learning rate). SGD allows minibatch (online/out-of-core) learning via the partial\_fit method. For best results using the default learning rate schedule, the data should have zero mean and unit variance.

****

**Fig No. :- 11 :- SGD CLASSIFIER**

* **LOGISTIC REGRESSION :-**

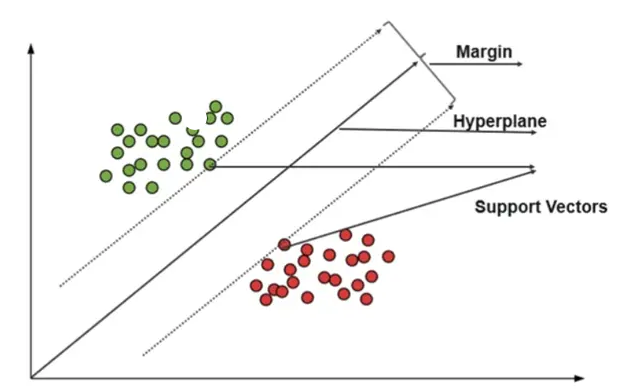
In logistic regression, a logit transformation is applied on the odds—that is, the probability of success divided by the probability of failure.

****

**Fig No. :- 12 :- LOGISTIC REGRESSION**

* **SVM AND SVC :-**

A Support Vector Machine (SVM) is a powerful machine learning algorithm widely used for both linear and nonlinear classification, as well as regression and outlier detection tasks. SVMs are highly adaptable, making them suitable for various applications such as text classification, image classification, spam detection, handwriting identification, gene expression analysis, face detection, and anomaly detection. SVMs are particularly effective because they focus on finding the maximum separating hyperplane between the different classes in the target feature, making them robust for both binary and multiclass classification.



**Fig No. :- 13 :- SVM/SVC**

* **MACHINE LEARNING MODEL DISPLAY :-**
* **SGD CLASSSIFIER :-**

**train\_df, test\_df = train\_test\_split(DIABETES\_df\_copy, test\_size=0.1, random\_state=42)**

**train\_df\_labels = train\_df["Outcome"].copy()**

**train\_df= train\_df.drop("Outcome", axis=1)**

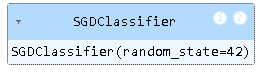
**num\_pipeline = Pipeline([('std\_scaler', StandardScaler()), ])**

**train\_prepared = num\_pipeline.fit\_transform(train\_df)**

**train\_prepared**

**sgd\_clf = SGDClassifier(random\_state=42)**

**sgd\_clf.fit(train\_prepared, train\_df\_labels)**

****

**Fig No. :- 14 :- SGD CLASSIFIER MODEL EVALUATION**

**cross\_val\_score(sgd\_clf, train\_prepared, train\_df\_labels, cv= 3, scoring='accuracy')**

**prediction = sgd\_clf.predict(train\_prepared)**

**print("SGD Accuracy of Classifier: ", sgd\_clf.score(train\_prepared, train\_df\_labels))**

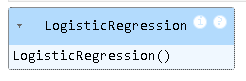
****

**Fig No. :- 15 :- SGD CLASSIFIER MODEL ACCURACY**

* **LOGISTIC REGRESSION :-**

**model = LogisticRegression()**

**model.fit(train\_prepared, train\_df\_labels)**

****

**Fig No. :- 16 :- LOGISTIC REGRESSION MODEL EVALUATION**

**from sklearn.model\_selection import cross\_val\_score**

**cross\_val\_score(model, train\_prepared, train\_df\_labels, cv= 3, scoring='accuracy')**

**prediction = model.predict(train\_prepared)**

**print("LR Accuracy of Classifier: ", model.score(train\_prepared, train\_df\_labels))**

****

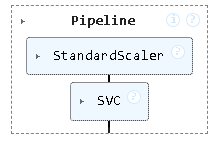
**Fig No. :- 17 :- LOGISTIC REGRESSION MODEL ACCURACY**

* **SVM/SVC :-**

**poly\_kernel\_svm\_clf = Pipeline([ ("scaler", StandardScaler()),**

**("svm\_clf", SVC(kernel="poly", degree=3, coef0=1, C=5))])**

**poly\_kernel\_svm\_clf.fit(train\_prepared, train\_df\_labels)**

****

**Fig No. :- 18 :- SVM MODEL EVALUATION**

**print("SVM Accuracy of Classifier: ", poly\_kernel\_svm\_clf.score(train\_prepared, train\_df\_labels))**

****

**Fig No. :- 19 :- SVM MODEL ACCURACY**

**CHAPTER 4**

**MODEL EVALUATION & VALIDATION**

**4.1 – PERFORMANCE MATRIX**

In this project, we evaluated model performance using key classification metrics: Accuracy, Precision, Recall, and F1-Score. These metrics were derived from the confusion matrix and help assess both the correctness and reliability of predictions, especially in an imbalanced dataset. Accuracy measures the overall correctness, Precision evaluates true positive predictions among all positive predictions, Recall captures how well actual positives were identified, and F1-Score balances Precision and Recall.

We applied and compared multiple models: SGD Classifier, Logistic Regression, SVM with Polynomial Kernel, and K-Nearest Neighbors (KNN). Cross-validation was used for fair evaluation. The best accuracy was observed with KNN (k=11) and SVM, showing high potential for accurate diabetes prediction.

**Tab. No. :- 1 :- MODEL COMPARISON**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **MODEL** | **ACCURACY** | **PRECESION** | **RECALL** | **F1 – SCORE** |
| **SGD Classifier** | **80.76%** | **0.67** | **0.81** | **0.73** |
| **Logistic Regression** | **80.76%** | **0.73** | **0.65** | **0.69** |
| **SVM** | **72.30%** | **0.57** | **0.62** | **0.60** |
| **KNN (K = 11)** | **78.46%** | **0.69** | **0.62** | **0.65** |

* **HEATMAP :-** Displaying heatmaps and visualizing the correlation between different features included in the prediction. Heatmap is used to analyze the correlation between the features of the dataset. Glucose, BMI, Age, and Pregnancies show the strongest positive correlation with diabetes outcome, with Glucose being the most significant. Most features have low inter-correlation, indicating minimal multicollinearity and good feature independence.

import pandas as pd

import seaborn as sns

import matplotlib.pyplot as plt

df =DIABETES\_df

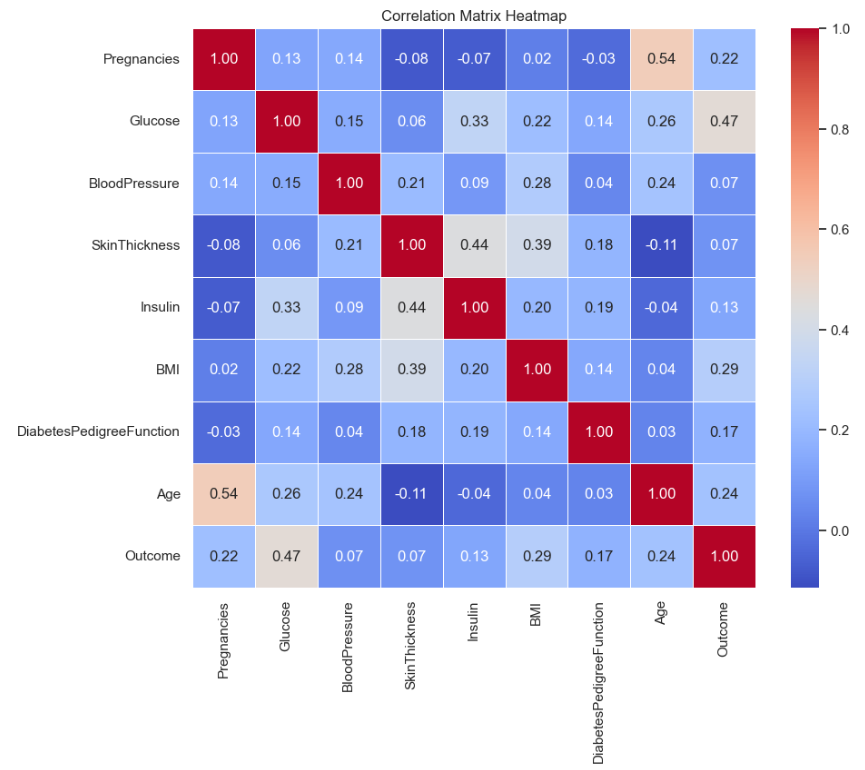
corr\_matrix = df.corr()

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

sns.heatmap(corr\_matrix, annot=True, cmap="coolwarm", fmt=".2f", linewidths=0.5)

plt.title("Correlation Matrix Heatmap")

plt.show()



**Fig No. :- 20 :- HEATMAP**

* **MODEL COMPARISON :-** Comparing models [SGD Classifier, Logistic Regression, SVM, KNN] and extracting insights that will help to give the best model’s Accuracy, Precesion, Recall, F1-Score. Here we have used the analysis of scatterplot about the insulin outliers and set the insulin range from 0 – 846 to 16 – 846. This will help in performance of models and also improve accuracy of the overall prediction.
* df = pd.read\_csv('C:/Users/punit/Downloads/DIABETESDIAGNOSIS/DIABETES.csv')

df = df[df["Insulin"] >= 16]

X = df.drop(['Outcome', 'Insulin'], axis=1)

y = df['Outcome']

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.33, random\_state=42, stratify=y)

scaler = StandardScaler()

X\_train\_scaled = scaler.fit\_transform(X\_train)

X\_test\_scaled = scaler.transform(X\_test)

models = {

"SGD Classifier": SGDClassifier(random\_state=42),

"Logistic Regression": LogisticRegression(),

"SVM (Poly Kernel)": SVC(kernel='poly', degree=3, coef0=1, C=5),

"KNN (k=11)": KNeighborsClassifier(n\_neighbors=11)

}

results = []

for name, model in models.items():

model.fit(X\_train\_scaled, y\_train)

y\_pred = model.predict(X\_test\_scaled)

acc = accuracy\_score(y\_test, y\_pred)

prec = precision\_score(y\_test, y\_pred)

rec = recall\_score(y\_test, y\_pred)

f1 = f1\_score(y\_test, y\_pred)

results.append([name, acc, prec, rec, f1])

results\_df = pd.DataFrame(results, columns=["Model", "Accuracy", "Precision", "Recall", "F1-Score"])

print("\nModel Comparison:\n")

print(results\_df)

plt.figure(figsize=(12, 6))

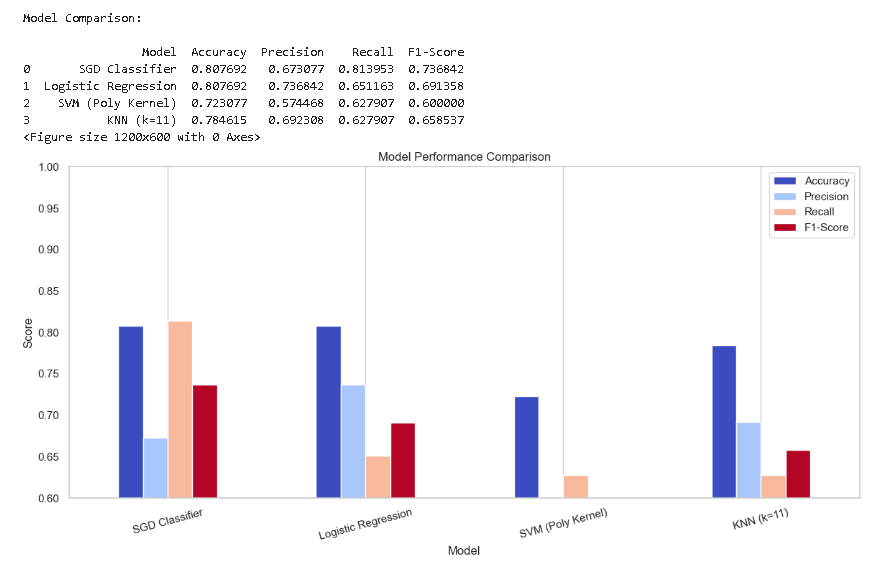
results\_df.set\_index("Model")[["Accuracy", "Precision", "Recall", "F1-Score"]].plot(kind='bar', figsize=(12, 6), ylim=(0.6, 1.0), colormap='coolwarm')

plt.title("Model Performance Comparison")

plt.ylabel("Score")

plt.xticks(rotation=15)

plt.grid(axis='y')

plt.tight\_layout()

plt.show()

**Fig No. :- 21 :- MODEL COMPARISON**

* **COMBINED MODEL COMPARISON :-** Here we had used both model combination techniques to ensure the best model for the prediction by analyzing their accuracy and overall performance. By using Stacking and Voting we are looking for the best from the both in terms of accuracy, precesion, recall, f1-score.

models = [

('lr', LogisticRegression()),

('knn', KNeighborsClassifier(n\_neighbors=11)),

('svm', SVC(kernel='poly', degree=3, coef0=1, C=5, probability=True))

]

stacking = StackingClassifier(estimators=models, final\_estimator=LogisticRegression())

stacking.fit(X\_train\_scaled, y\_train)

y\_pred\_stack = stacking.predict(X\_test\_scaled)

voting = VotingClassifier(estimators=models, voting='soft')

voting.fit(X\_train\_scaled, y\_train)

y\_pred\_vote = voting.predict(X\_test\_scaled)

def evaluate\_model(name, y\_true, y\_pred):

return {

"Model": name,

"Accuracy": accuracy\_score(y\_true, y\_pred),

"Precision": precision\_score(y\_true, y\_pred),

"Recall": recall\_score(y\_true, y\_pred),

"F1-Score": f1\_score(y\_true, y\_pred)

}

results = [

evaluate\_model("Stacking Classifier", y\_test, y\_pred\_stack),

evaluate\_model("Voting Classifier", y\_test, y\_pred\_vote)

]

results\_df = pd.DataFrame(results)

print(results\_df.sort\_values(by="F1-Score", ascending=False))

**Tab. No. :- 2 :- STACKING/VOTING**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Model** | **Accuracy** | **Precision** | **Recall** | **F1-Score** |
| Stacking Classifier | 0.815385 | 0.743590 | 0.674419 | 0.707317 |
| Voting Classifier | 0.800000 | 0.742857 | 0.604651 | 0.666667 |

**CHAPTER 5**

**5.1 – CONCLUSION**

In this study, several machine learning models were applied and compared for predicting diabetes based on patient health data. Accuracy—defined as the ratio of correctly predicted instances to the total number of predictions—served as a key metric in evaluating model performance. Among the individual classifiers, Logistic Regression and the SVM with a polynomial kernel performed well; however, ensemble methods like the **Voting Classifier** and **Stacking Classifier** outperformed them, with the **Stacking Classifier achieving the highest accuracy of 81.53%.** This superior performance underscores the value of combining multiple models to improve prediction reliability. Furthermore, strategic preprocessing, such as excluding uninformative insulin values and applying feature scaling, significantly contributed to model efficiency and accuracy. These findings suggest that ensemble approaches are particularly well-suited for medical prediction tasks like diabetes diagnosis.

**5.2 – FUTURE SCOPE**

To enhance the effectiveness of diabetes prediction systems, incorporating additional features such as lifestyle and genetic information—like diet, physical activity, and family history—can lead to more holistic and accurate predictions. Leveraging advanced techniques, including deep learning models and neural networks, particularly with larger datasets, has the potential to significantly improve performance. Integrating real-time monitoring by connecting the system with wearable devices could allow for continuous diabetes tracking and timely interventions. Furthermore, feature engineering, such as creating new variables like BMI categories or age groups, and applying feature selection techniques, can boost model efficiency and relevance. Finally, ensuring explainability through model interpretability tools such as SHAP or LIME is crucial, especially for building trust and facilitating adoption in clinical settings.

**CHAPTER 6**

**6.1 – REFERENCES**

1.<https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8950225/#B3-healthcare-10-00541>

2. <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8950225/>

3 <https://www.mdpi.com/search?q=Machine+Learning>

4.  <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8754556/#CR112>

5. Stafford I., Kellermann M., Mossotto E., Beattie R., MacArthur B., Ennis S. A systematic review of the applications of artificial intelligence and machine learning in autoimmune diseases. *NPJ Digit. Med.*2020;3:1–11. doi: 10.1038/s41746-020-0229-3. [[PMC free article](https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7062883/)] [[PubMed](https://pubmed.ncbi.nlm.nih.gov/32195365)] [[CrossRef](https://doi.org/10.1038%2Fs41746-020-0229-3)] [[Google Scholar](https://scholar.google.com/scholar_lookup?journal=NPJ+Digit.+Med.&title=A+systematic+review+of+the+applications+of+artificial+intelligence+and+machine+learning+in+autoimmune+diseases&author=I.+Stafford&author=M.+Kellermann&author=E.+Mossotto&author=R.+Beattie&author=B.+MacArthur&volume=3&publication_year=2020&pages=1-11&pmid=31934645&doi=10.1038/s41746-020-0229-3&)]

6. . Fix E., Hodges J.L. Discriminatory analysis. Nonparametric discrimination: Consistency properties. *Int. Stat. Rev. Int. De Stat.*1989;57:238–247. doi: 10.2307/1403797. [[CrossRef](https://doi.org/10.2307%2F1403797)] [[Google Scholar](https://scholar.google.com/scholar_lookup?journal=Int.+Stat.+Rev.+Int.+De+Stat.&title=Discriminatory+analysis.+Nonparametric+discrimination:+Consistency+properties&author=E.+Fix&author=J.L.+Hodges&volume=57&publication_year=1989&pages=238-247&doi=10.2307/1403797&)]

7. .  Balogh E.P., Miller B.T., Ball J.R. *Improving Diagnosis in Health Care.* National Academic Press; Washington, DC, USA: 2015. [[PubMed](https://pubmed.ncbi.nlm.nih.gov/26803862)] [[CrossRef](https://doi.org/10.17226%2F21794)] [[Google Scholar](https://scholar.google.com/scholar_lookup?title=Improving+Diagnosis+in+Health+Care&author=E.P.+Balogh&author=B.T.+Miller&author=J.R.+Ball&publication_year=2015&)]

8. American Diabetes Association. Diagnosis and classification of diabetes mellitus. Diabetes Care*.*2014;37(Supplement 1):S81–S90. doi: 10.2337/dc14-s081. [[PubMed](https://pubmed.ncbi.nlm.nih.gov/24357215)] [[CrossRef](https://doi.org/10.2337%2Fdc14-s081)] [[Google Scholar](https://scholar.google.com/scholar_lookup?journal=Diabetes+Care&title=Diagnosis+and+classification+of+diabetes+mellitus&volume=37&issue=Supplement+1&publication_year=2014&pages=S81-S90&pmid=24357215&doi=10.2337/dc14-s081&)]

9. Srinivasu P.N., SivaSai J.G., Ijaz M.F., Bhoi A.K., Kim W., Kang J.J. Classification of skin disease using deep learning neural networks with MobileNet V2 and LSTM. *Sensors.*2021;**21**:2852. doi: 10.3390/s21082852. [[PMC free article](https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8074091/)] [[PubMed](https://pubmed.ncbi.nlm.nih.gov/33919583)] [[CrossRef](https://doi.org/10.3390%2Fs21082852)] [[Google Scholar](https://scholar.google.com/scholar_lookup?journal=Sensors&title=Classification+of+skin+disease+using+deep+learning+neural+networks+with+MobileNet+V2+and+LSTM&author=P.N.+Srinivasu&author=J.G.+SivaSai&author=M.F.+Ijaz&author=A.K.+Bhoi&author=W.+Kim&volume=21&publication_year=2021&pages=2852&pmid=33919583&doi=10.3390/s21082852&)] [[Ref list](https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8659829/#B4-sensors-21-08095)]

10. Davies M. J., D’Alessio D. A., Fradkin J., et al. Management of hyperglycaemia in type 2 diabetes, 2018. a consensus report by the American diabetes association (ada) and the european association for the study of diabetes (easd) Diabetologia*.*2018;61(12):2461–2498. doi: 10.1007/s00125-018-4729-5. [[PubMed](https://pubmed.ncbi.nlm.nih.gov/30288571)] [[CrossRef](https://doi.org/10.1007%2Fs00125-018-4729-5)] [[Google Scholar](https://scholar.google.com/scholar_lookup?journal=Diabetologia&title=Management+of+hyperglycaemia+in+type+2+diabetes,+2018.+a+consensus+report+by+the+American+diabetes+association+(ada)+and+the+european+association+for+the+study+of+diabetes+(easd)&author=M.+J.+Davies&author=D.+A.+D%E2%80%99Alessio&author=J.+Fradkin&volume=61&issue=12&publication_year=2018&pages=2461-2498&pmid=30288571&doi=10.1007/s00125-018-4729-5&)]

11. Choubey D. K., Kumar M., Shukla V., Tripathi S., Dhandhania V. K. Comparative analysis of classification methods with PCA and LDA for diabetes. Current Diabetes Reviews*.*2020;16(8):833–850. doi: 10.2174/1573399816666200123124008. [[PubMed](https://pubmed.ncbi.nlm.nih.gov/31971112)] [[CrossRef](https://doi.org/10.2174%2F1573399816666200123124008)] [[Google Scholar](https://scholar.google.com/scholar_lookup?journal=Current+Diabetes+Reviews&title=Comparative+analysis+of+classification+methods+with+PCA+and+LDA+for+diabetes&author=D.+K.+Choubey&author=M.+Kumar&author=V.+Shukla&author=S.+Tripathi&author=V.+K.+Dhandhania&volume=16&issue=8&publication_year=2020&pages=833-850&pmid=31971112&doi=10.2174/1573399816666200123124008&)]

12. Kumari S., Kumar D., Mittal M. An ensemble approach for classification and prediction of diabetes mellitus using soft voting classifier. International Journal of Cognitive Computing in Engineering*.*2021;2 doi: 10.1016/j.ijcce.2021.01.001. [[CrossRef](https://doi.org/10.1016%2Fj.ijcce.2021.01.001)] [[Google Scholar](https://scholar.google.com/scholar_lookup?journal=International+Journal+of+Cognitive+Computing+in+Engineering&title=An+ensemble+approach+for+classification+and+prediction+of+diabetes+mellitus+using+soft+voting+classifier&author=S.+Kumari&author=D.+Kumar&author=M.+Mittal&volume=2&publication_year=2021&doi=10.1016/j.ijcce.2021.01.001&)]

13.  Baik I. Response: projection of diabetes prevalence in Korean adults for the year 2030 using risk factors identified from national data (diabetes metab J 2019;43:90-6) Diabetes & Metabolism Journal*.*2019;43(2):244–245. doi: 10.4093/dmj.2019.0060. [[PMC free article](https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6470096/)] [[PubMed](https://pubmed.ncbi.nlm.nih.gov/30993941)] [[CrossRef](https://doi.org/10.4093%2Fdmj.2019.0060)] [[Google Scholar](https://scholar.google.com/scholar_lookup?journal=Diabetes+&+Metabolism+Journal&title=Response:+projection+of+diabetes+prevalence+in+Korean+adults+for+the+year+2030+using+risk+factors+identified+from+national+data+(diabetes+metab+J+2019;43:90-6)&author=I.+Baik&volume=43&issue=2&publication_year=2019&pages=244-245&pmid=30993941&doi=10.4093/dmj.2019.0060&)]

14. Ampofo A. G., Emmanuel B. B. Beyond 2020: modelling obesity and diabetes prevalence. Diabetes Research and Clinical Practice*.*2020;167108362 [[PubMed](https://pubmed.ncbi.nlm.nih.gov/32758618)] [[Google Scholar](https://scholar.google.com/scholar_lookup?journal=Diabetes+Research+and+Clinical+Practice&title=Beyond+2020:+modelling+obesity+and+diabetes+prevalence&author=A.+G.+Ampofo&author=B.+B.+Emmanuel&volume=167&publication_year=2020&)]

15. Zhang Y., Lin Z., Kang Y., Ning R., Meng Y. A feed-forward neural network model for the accurate prediction of diabetes mellitus. International Journal of Scientific and Technology Research*.*2018;7(8):151–155. [[Google Scholar](https://scholar.google.com/scholar_lookup?journal=International+Journal+of+Scientific+and+Technology+Research&title=A+feed-forward+neural+network+model+for+the+accurate+prediction+of+diabetes+mellitus&author=Y.+Zhang&author=Z.+Lin&author=Y.+Kang&author=R.+Ning&author=Y.+Meng&volume=7&issue=8&publication_year=2018&pages=151-155&)]