DIABETES PREDICTION USING MACHINE LEARNING

A PROJECT REPORT

In a partial fulfillment of the Requirements for the award of the degree of

BACHELOR OF COMPUTER SCIENCE AND ENGINEERING

Under the guidance of MAHENDRA DUTTA By

ARASHRIKA AND NANDINI

affiliated to Makaut University

In association with

DECLARATION

We hereby declare that the project proposal titled “Diabetes Prediction using Machine Learning and Python”, submitted in partial fulfilment of the requirements for the degree of Bachelor of Computer Science and Engineering at Future Institute of Engineering and Management, Sonarpur Station Road, West Bengal, is a genuine work carried out under the guidance of Mahendra Dutta. To the best of our knowledge and belief, the content presented in this project has not been submitted elsewhere for the award of any degree.

Date:

Name of the Students: Nandini Das,

Arashrika Das

ARDENT®
COMPUTECH PVT. LTD.
SALT LAKE | JADAVPUR | DURGAPUR | BANDEL

Ardent Computech Pvt. Ltd (An ISO 9001:2015 Certified

CERTIFICATE

This is to certify that the proposal for the minor project entitled “Diabetes Prediction using Machine Learning and Python” is a record of bona fide work carried out by Arashrika and Nandini under my guidance at Ardent Computech Pvt. Ltd. In my opinion, this report,

in its current form, fulfils the partial requirements for the degree of Bachelor of Computer Science and Engineering as per the regulations of Ardent Computech Pvt. Ltd. To the best of my knowledge, the results presented in this report are original and merit inclusion in the final version of the project report.

Guide / Supervisor

MR. MAHENDRA DUTTA

Project Engineer Ardent Computech Pvt. Ltd (An ISO 9001:2015 Certified)

ACKNOWLEDGEMENT

I would like to express my sincere gratitude to all these individuals for mentoring and I would like to express my sincere gratitude to all those who have mentored and supported me throughout the completion of this project.

I extend my heartfelt thanks to Mahendra Dutta, a project engineer at Ardent Computech Pvt. Ltd., for his invaluable guidance and assistance in gathering the necessary information for this project. His support throughout the project duration has been instrumental.
I am also deeply grateful to the other trainees, project assistants, and team members at Ardent Computech Pvt. Ltd. for their encouragement and cooperation. The guidance and support provided by everyone involved were crucial to the success of this project.
supporting me in completing this project.
My teacher Mahendra Dutta who is a project engineer at Ardent Computech Pvt. Ltd , has helped me to gather all the necessary information about this project, a special thanks to him for his guidance throughout the project duration.

Abstract

Diabetes is a chronic disease with the potential to cause a global healthcare crisis. According to the International Diabetes Federation, 382 million people worldwide are currently living with diabetes. By 2030, this number is expected to rise to 592 million. Diabetes is characterized by elevated blood glucose levels, which can lead to symptoms such as frequent urination, increased thirst, and increased hunger. It is a leading cause of blindness, kidney failure, amputations, heart failure, and stroke.

When we eat, our body converts food into glucose. Normally, the pancreas releases insulin, which acts as a key to allow glucose to enter cells and be used for energy. However, in diabetes, this system malfunctions.

Machine learning, a burgeoning field within data science, explores how machines can learn from experience. The objective of this project is to develop a system for the early prediction of diabetes with high accuracy by integrating various machine learning techniques. Algorithms such as K-Nearest neighbours, Logistic Regression, Random Forest, Support Vector Machine, and Decision Tree are employed. The accuracy of the model using each algorithm is evaluated, and the most accurate model is selected for predicting diabetes.
Keywords: Machine Learning, Diabetes, Decision Tree, K-Nearest neighbours, Logistic Regression, Support Vector Machine, Accuracy.

CHAPTERS

Chapter 1: Introduction 6-7

Chapter 2: Literature Survey 8

Chapter 3:Overview of Learning Methods and Motivation for Improved Diabetes Diagnosis 9

PAGE NO.

Chapter 4: Methodology 10-15

- Proposed Model
- Data acquisition and pre- processing
- Data Visualization
- Analysis

Chapter 5: Evaluation and Result 16-22

Chapter 6: Performance Analysis comparison of the models 23

Chapter 7 Result Analysis 24

Chapter 8: Conclusion and Future Work 25

Chapter 9: References 26

Chapter 1

Introduction

Overview of Diabetes:

- Prevalence:
  - Diabetes is rapidly growing, affecting an increasing number of people, including younger individuals.
- Understanding Normal Glucose Regulation:
  - Source of Glucose:
    - Glucose comes from carbohydrate-rich foods such as bread, cereal, pasta, rice, fruit, dairy products, and starchy vegetables.
    - Carbohydrates provide the primary energy source for the body.
  - Glucose Utilization:
    - After consuming carbohydrate foods, the body breaks them down into glucose.
    - Glucose circulates in the bloodstream.
    - Role of Insulin:
      - Insulin, a hormone produced by the beta cells in the pancreas, is essential for glucose utilization.
      - Insulin acts as a key, attaching to cell receptors to allow glucose to enter cells from the bloodstream and be used for energy.

nTypes of Diabetes:

- Type 1 Diabetes:
  - Description:
    - An autoimmune condition where the immune system attacks and destroys insulin-producing beta cells in the pancreas.
  - Characteristics:
    - Insufficient insulin production.
    - No known causes or prevention methods.
- Type 2 Diabetes:
  - Description:
    - Characterized by low insulin production or ineffective use of insulin by the body.
  - Prevalence:
    - The most common form, affecting about 90% of diagnosed individuals.
  - Causes:
    - A combination of genetic factors and lifestyle choices.
- Gestational Diabetes:
  - Description:
    - Occurs in pregnant women who develop high blood sugar levels during pregnancy.
  - Future Risk:
    - Two-thirds of women with gestational diabetes may experience it in subsequent pregnancies.
    - There is an increased risk of developing Type 1 or Type 2 diabetes after pregnancy.

Symptoms of Diabetes:

- Frequent urination
- Increased thirst
- Fatigue or sleepiness
- Weight loss
- Blurred vision

The objectives include:

- Accurate Prediction of Diabetes Risk: The primary objective is to develop a model that can accurately predict the likelihood of an individual developing diabetes based on input features such as age, BMI, blood pressure, glucose levels, and other relevant health indicators. This helps in early detection and timely intervention.
- Identification of Key Risk Factors: The model should be able to identify and rank the key factors that contribute most to the risk of diabetes. Understanding these factors can guide healthcare professionals in making informed decisions about prevention and treatment strategies.
- Personalized Health Recommendations: The model aims to provide personalized health recommendations for individuals at risk of diabetes. Based on the prediction and identified risk factors, the model can suggest lifestyle changes, dietary adjustments, or further medical tests to help reduce the risk of developing diabetes.

Chapter 2

LITERATURE SURVEY

InYasodha et al. [1] explored the classification of diabetes using diverse datasets to determine if an individual is diabetic. They utilized a dataset with 200 instances and nine attributes, gathered from hospital warehouses, which includes data from blood and urine tests. The study employed the WEKA tool for classification and assessed the data using a 10-fold cross-validation approach, suitable for small datasets. The algorithms tested were Naïve Bayes, J48, REP Tree, and Random Tree. The results revealed that the J48 algorithm achieved the highest accuracy of 60.2% compared to the other methods.

Aiswarya et al. [2] aimed to enhance diabetes detection by investigating data patterns through classification analysis using Decision Tree and Naïve Bayes algorithms. Utilizing the PIMA Indian Diabetes dataset, the study applied cross-validation and a 70:30 train-test split. The findings indicated that the J48 algorithm delivered an accuracy rate of 74.8%, while the Naïve Bayes algorithm provided a higher accuracy of 79.5%.

Gupta et al. [3] focused on calculating the accuracy, sensitivity, and specificity of various classification methods and comparing their results using WEKA. They also compared these methods with implementations on other tools like RapidMiner and MATLAB using the same parameters. The algorithms tested included JRIP, Jgraft, and BayesNet. The study found that the Jgraft algorithm had the highest accuracy of 81.3%, with a sensitivity of 59.7% and specificity of 81.4%. Additionally, WEKA was determined to be more effective than MATLAB and RapidMiner for this purpose.

Lee et al. [4] investigated the application of the CART decision tree algorithm on diabetes datasets, emphasizing the importance of addressing class imbalance issues before applying any algorithm. They applied a resampling filter to manage class imbalance, which is crucial for improving accuracy rates in predictive models. The study highlighted that handling class imbalance during the data preprocessing stage significantly boosts the performance of the predictive model.

Chapter 3

Overview of Learning Methods and Motivation for Improved Diabetes Diagnosis

A. Supervised Learning / Predictive ModelsSupervised learning algorithms are designed to construct predictive models that estimate missing values using other values present in the dataset. These algorithms work with a set of input and output data to build a model capable of making accurate predictions for new datasets. Common supervised learning techniques include Decision Trees, Bayesian Methods, Artificial Neural Networks, Instance-Based Learning, and Ensemble Methods, all of which are significant in the field of machine learning [3].

B. Unsupervised Learning / Descriptive ModelsUnsupervised learning methods develop descriptive models where the output is unknown, and only input data is available. This type of learning is often applied to transactional data and includes clustering algorithms such as k-Means and k-Medians clustering [3].

C. Semi-Supervised LearningSemi-Supervised Learning combines both labelled and unlabelled data within a training dataset. It incorporates techniques from both classification and regression, including Logistic Regression and Linear Regression, to improve model accuracy and reliability [3].

III. MotivationThe incidence of diabetes has dramatically increased over the past decade, largely attributed to changes in lifestyle. Current diagnostic methods are prone to three main types of errors:

1. False-Negative Error: A patient who is actually diabetic is incorrectly identified as non-diabetic.
2. False-Positive Error: A patient who is not diabetic is mistakenly classified as diabetic.
3. Unclassifiable Error: The system fails to classify a patient correctly due to insufficient data, leading to either unnecessary treatments or missed diagnoses.

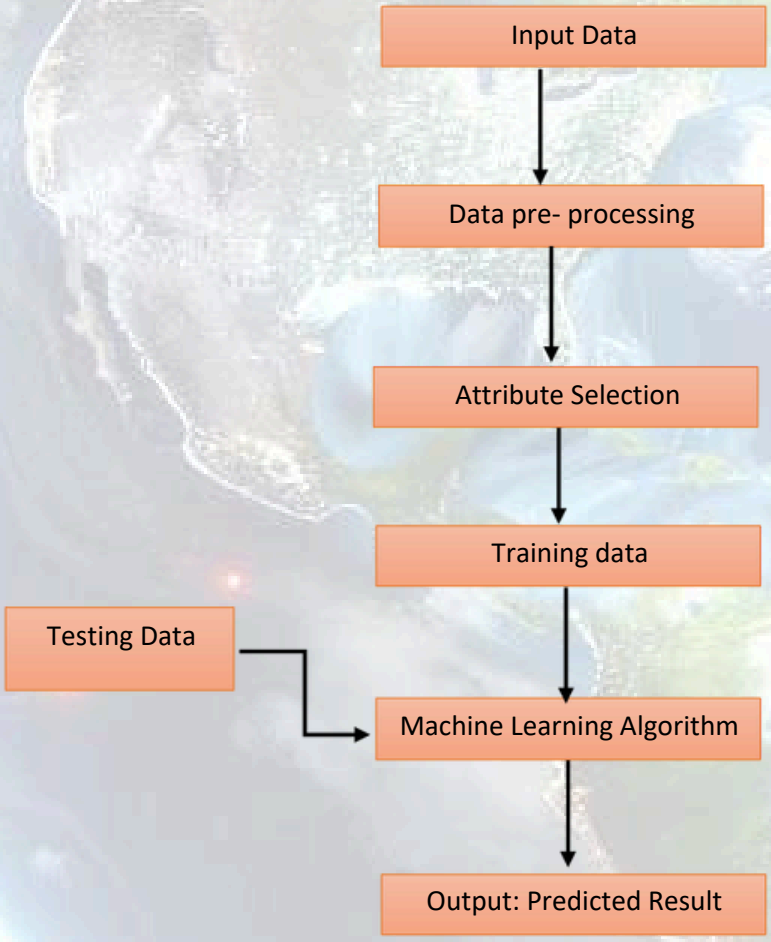
These errors can result in inappropriate or inadequate treatment. To mitigate these issues and improve diagnostic accuracy, there is a need to develop systems using machine learning algorithms and data mining techniques. Such systems can provide more accurate results and reduce reliance on human judgment.

Chapter 4

Methodology

IIn this section, I present the proposed model for diabetes prediction using machine learning. It also contains a description of the dataset, it's acquisition and pre-processing and the analysis of the algorithm used.

4.1 Proposed Model



4.2 Data acquisition and pre- processing

The data was for the town of the Seattle for the period 1st January 2012 to 31st December 2015, the file format was a Comma Separated Value (CSV) which contains 1461 rows and 6 columns, the columns which were identified as

- Pregnancies
- Glucose
- Blood pressure
- Skin thickness
- Insulin
- BMI
- Diabetes Pedigree Function
- Age
- Outcome

```
# printing the first 5 rows of the document
diabetes_dataset.head()
```

	Pregnancies	Glucose	BloodPressure	SkinThickness	Insulin	BMI	DiabetesPedigreeFunction	Age	Outcome
0	6	148	72	35	0	33.6	0.627	50	1
1	1	85	66	29	0	26.6	0.351	31	0
2	8	183	64	0	0	23.3	0.672	32	1
3	1	89	66	23	94	28.1	0.167	21	0
4	0	137	40	35	168	43.1	2.288	33	1

Fig 3.2 Data head Overview

To achieve accurate forecasting and ensure the high performance of the algorithm, the data must be pre-processed effectively. This process involves transforming the acquired data into a format that can be easily understood, removing any duplicate or null values, and dropping attributes that are deemed unnecessary. In this analysis, the data pre-processing included the removal of the 'date' column, as it was considered irrelevant to the forecasting model. After this step, the dataset was found to be complete, with no null values present. This thorough pre-processing ensures that the data is clean and ready for the next stages of analysis, enhancing the accuracy and reliability of the results.

```
# seperating the data and labels
X = diabetes_dataset.drop(columns = 'Outcome',axis=1)
Y = diabetes_dataset['Outcome']
```

```
print(X)
```

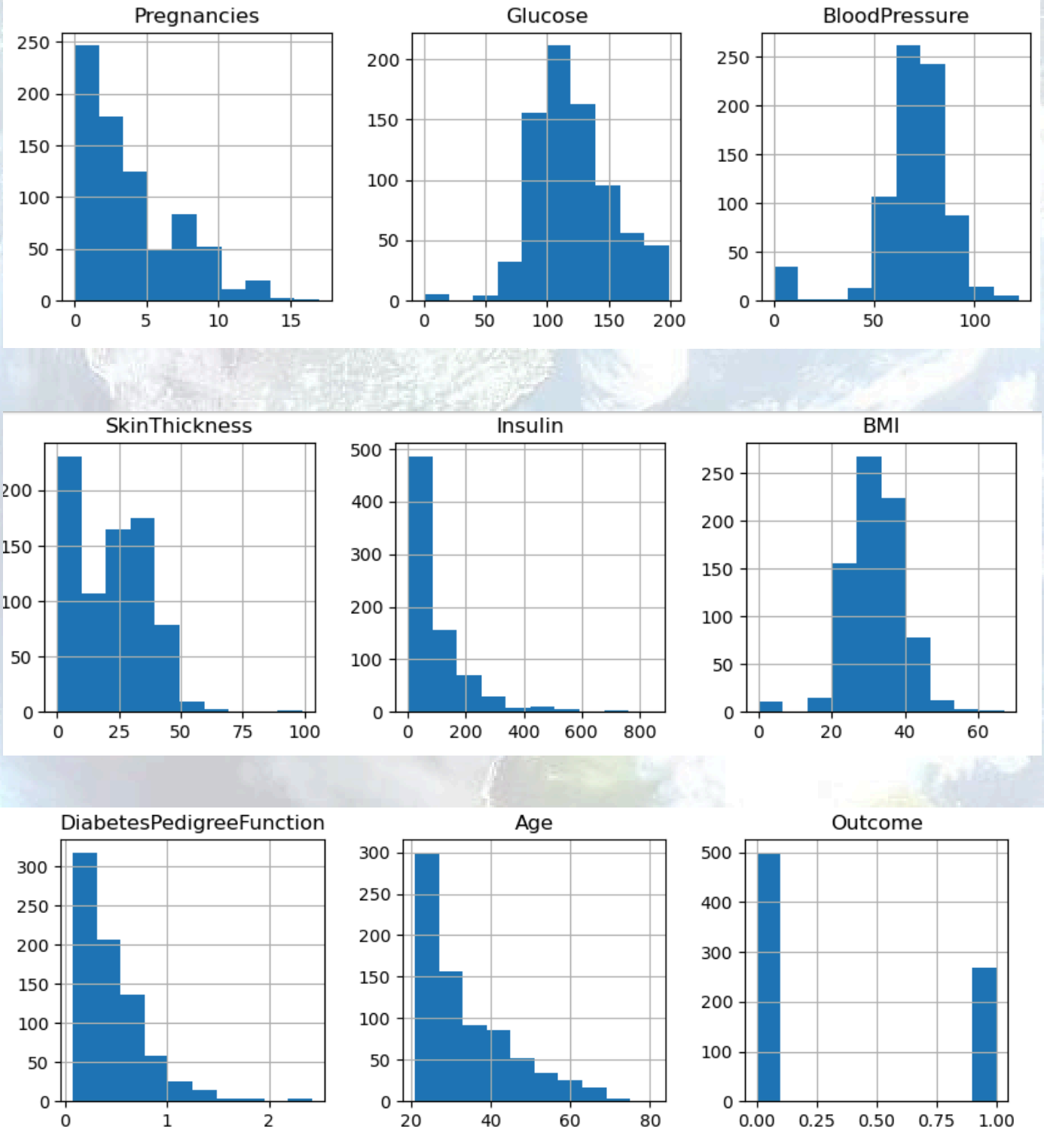
```
     Pregnancies  Glucose  BloodPressure  SkinThickness  Insulin   BMI  \
0              6      148             72             35        0  33.6
1              1       85             66             29        0  26.6
2              8      183             64              0        0  23.3
3              1       89             66             23       94  28.1
4              0      137             40             35      168  43.1
..           ...      ...            ...            ...      ...   ...
763           10      101             76             48      180  32.9
764            2      122             70             27        0  36.8
765            5      121             72             23      112  26.2
766            1      126             60              0        0  30.1
767            1       93             70             31        0  30.4

     DiabetesPedigreeFunction  Age
0                       0.627   50
1                       0.351   31
2                       0.672   32
3                       0.167   21
4                       2.288   33
..                        ...  ...
763                     0.171   63
764                     0.340   27
765                     0.245   30
766                     0.349   47
```

```
# checking for null values
diabetes_dataset.isnull().sum()
```

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

4.3 Data Visualization

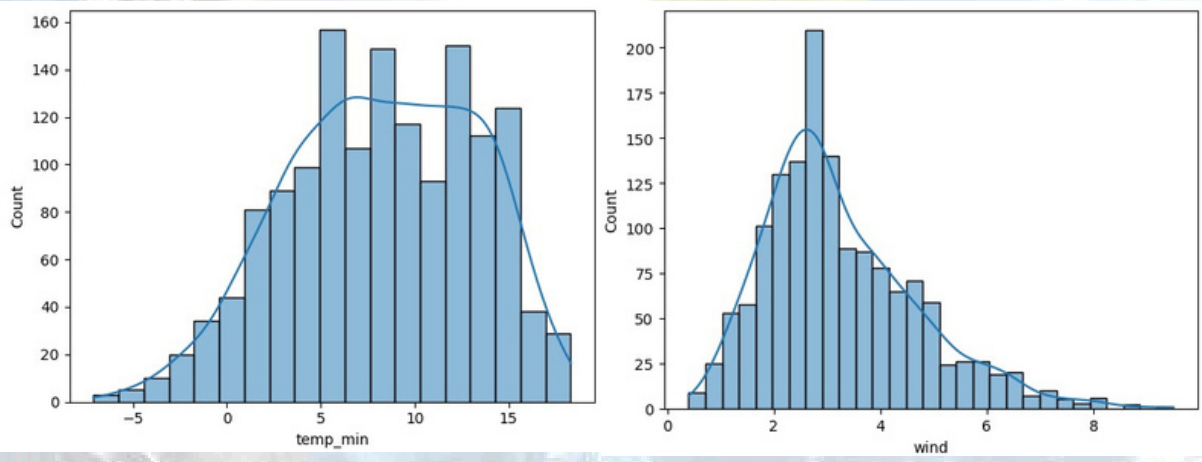
```
#histagram
diabetes_dataset.hist(bins=10,figsize=(10,10))
plt.show()
```



4.4 Analysis

For the analysis of our data, the data was split into tests and Training dataset, using 10% for testing, four machine learning classification algorithms were implemented, they are:

- Random Forest classifier
- Gaussian Naïve Bayes model
- Decision Tree Algorithm
- Logistics Regression Algorithm
- Support Vector Machine



For the analysis of our data, the data was split into tests and Training dataset, using 10% for testing, four machine learning classification algorithms were implemented, they are:

- K- Nearest Neighbors (KNN)
- Random. Forest classifier
- Gradient Boosting Classifier
- Gaussian Naïve Bayes model
- Decision Tree Algorithm
- Logistics Regression Algorithm

K- Nearest Neighbors

The dataset is used by KNN to make predictions. Probabilities for new instances (x) are calculated by scanning the data set for the K most comparable examples and predicting the output variable for those K occurrences.

Random Forest Classifier

A random forest classifier is a collection of tree-structured classifiers whose results are compounded into one result; it is an ensemble machine learning algorithm which can be implemented for both classification and regression tasks and is made up of a set of classifiers known as a decision tree, random forest classifier is known to produce accurate predictions, provides flexibility and reduced the risk of overfitting.

Gaussian Naive Bayes model

Gaussian Naive Bayes model is based on Bayes theorem, it assumes that a particular feature is independent of the value of any other feature, they can be trained very efficiently and are highly scalable, and the likelihood of the features is assumed to be

$$P(xi \mid y) = \frac{1}{\sqrt{2\pi\sigma 2}} \exp\left(-\frac{(xi-\mu y)^2}{2\sigma 2y}\right)$$

the variance is independent of y and x

Decision Tree algorithm

Decision tree algorithm. Belongs to the family of supervised machine learning, where the data is continuously split according to a certain parameter and represented by a tree structure. It is used to solve classification and regression tasks and is among the most popular machine learning algorithm. (Xindong et al. 2018) These algorithms were selected for the task because they were utilized by other weather forecasting models and have shown to have high predictive performance capabilities, they were Implemented with python programming language using the Jupiter notebook, and various libraries were imported for the analysis including pandas, ski-learn, TensorFlow, and Matplot library.

Logistic Regression Algorithm

The LR Algorithm computes the link between one or more independent factors and the category dependent variable. The output of LR is in the form of binary classification. A logistic function (sigmoid function) can be used to calculate the probability. 1 / (1 + e^-value) Where e is the natural logarithm base (Euler's number or the EXP () function) and value is the actual numerical value to be transformed. The logistic function was used to turn a situation with numbers ranging from -5 to 5 into a range of 0 to 1.

Chapter 5

Evaluation and result

A variety of results was obtained from the models trained with 75% of the data and tested with 25% of the data, in this section, we evaluate our models developed using various metrics, the metrics used for evaluation include:

- Accuracy
- Precision
- Recall
- Fi-score

Accuracy:

This is the total proportion of observations that have been correctly predicted mathematically, accuracy is defined as:

$$\frac{TP + TN}{TP + FP + TN + FN}$$

where TP= True positive, TN= True Negative, FP= False positive and FN = false negative.

Precision:

This is the percentage of the positive instance predicted that was correct, mathematically it is defined as

$$\frac{TP}{TP + FP}$$

where TP= True positive, FP= False Positive

Recall:

This is the percentage of the positive instance out of the total actual positive, mathematically it is defined as:

$$\frac{TP}{TP + FN}$$

where TP= True positive, FN= False Negative

F1-score:

This is the harmonic mean of precision and recall metrics, it is the overall correctness the model has achieved, mathematically it is defined as

$$\frac{2}{} = \frac{2 * Precision * recall}{}$$

$$\frac{1}{\text{Precision}} + \frac{1}{\text{recall}} \qquad \text{recall + Precision}$$

1. Logistic Regression Algorithm

The Logistic Regression Algorithm achieved an accuracy of 78.50%, Recall, precision, and f1-score were generated. As shown in figure 4.4 below

```
confusion_matrix(Y_test, Y_pred)
```

```
array([[89, 11],
       [26, 28]], dtype=int64)
```

```
print(classification_report(Y_test, Y_pred))
```

```
              precision    recall  f1-score   support

           0       0.77      0.89      0.83       100
           1       0.72      0.52      0.60        54

    accuracy                           0.76       154
   macro avg       0.75      0.70      0.72       154
weighted avg       0.75      0.76      0.75       154
```

```
from sklearn.linear_model import LogisticRegression
log_reg = LogisticRegression()
log_reg.fit(X_train,Y_train)
```

```
LogisticRegression()
```

```
Y_pred = log_reg.predict(X_test)
```

```
accuracy_score(Y_train, log_reg.predict(X_train))
```

```
0.7850162866449512
```

```
accuracy_score(Y_test, log_reg.predict(X_test))
```

```
0.7597402597402597
```

```
confusion_matrix(Y_test, Y_pred)
```

2. Training Decision Tree

The Training Decision Tree achieved an accuracy of 70.77%, Recall, precision and f1-score were generated.

```
from sklearn.tree import DecisionTreeClassifier
DT = DecisionTreeClassifier()
DT.fit(X_train,Y_train)
```

```
DecisionTreeClassifier()
```

```
Y_pred = DT.predict(X_test)
print(confusion_matrix(Y_test,Y_pred))
print(classification_report(Y_test,Y_pred))
print('accuracy:' ,accuracy_score(Y_test,Y_pred))
```

```
[[82 18]
 [27 27]]
              precision    recall  f1-score   support

           0       0.75      0.82      0.78       100
           1       0.60      0.50      0.55        54

    accuracy                           0.71       154
   macro avg       0.68      0.66      0.67       154
weighted avg       0.70      0.71      0.70       154

accuracy: 0.7077922077922078
```

3. Random Forest Classifier

For the random forest classifier accuracy, precision, recall, and f1- score were generated, as it is shown in figure 4.1 below, and the random forest classifier achieved a total accuracy of 74.02%

```python
from sklearn.ensemble import RandomForestClassifier
clf= RandomForestClassifier(n_estimators=100)
clf.fit(X_train,Y_train)
```

RandomForestClassifier

```
RandomForestClassifier()
```

```python
Y_pred = clf.predict(X_test)
```

```python
print(confusion_matrix(Y_test,Y_pred))
print(classification_report(Y_test,Y_pred))
print('Accuracy of the model:',accuracy_score(Y_test,Y_pred))
```

```
[[86 14]
 [26 28]]
              precision    recall  f1-score   support

           0       0.77      0.86      0.81       100
           1       0.67      0.52      0.58        54

    accuracy                           0.74       154
   macro avg       0.72      0.69      0.70       154
weighted avg       0.73      0.74      0.73       154

Accuracy of the model: 0.7402597402597403
```

4. Gaussian Naive Bayes model

The Gaussian Naive Bayes model achieved an 77.27 % accuracy, recall, f1-score, and precision were also calculated.

```
from sklearn.naive_bayes import GaussianNB
model = GaussianNB()
model.fit(X_train,Y_train)
```

```
GaussianNB()
```

```
Y_pred = model.predict(X_test)
```

```
print(confusion_matrix(Y_test,Y_pred))
print(classification_report(Y_test,Y_pred))
print('accuracy:', accuracy_score(Y_test,Y_pred))
```

```
[[88 12]
 [23 31]]
              precision    recall  f1-score   support

           0       0.79      0.88      0.83       100
           1       0.72      0.57      0.64        54

    accuracy                           0.77       154
   macro avg       0.76      0.73      0.74       154
weighted avg       0.77      0.77      0.77       154

accuracy: 0.7727272727272727
```

6. **SVM (Support Vector Machine)**

The SVM achieved an accuracy of 79.27 %, Recall, precision and f1-score were generated. As shown in figure 4.6 below

```
#svm
svm_model=svm.SVC(kernel='linear')
svm_model.fit(X_train,Y_train)
```

```
SVC
SVC(kernel='linear')
```

```
Y_pred = svm_model.predict(X_test)
```

```
print(confusion_matrix(Y_test, Y_pred))
print(classification_report(Y_test, Y_pred))
print('Accuracy=', accuracy_score(Y_test, Y_pred))
```

```
[[91  9]
 [26 28]]
              precision    recall  f1-score   support

           0       0.78      0.91      0.84       100
           1       0.76      0.52      0.62        54

    accuracy                           0.77       154
   macro avg       0.77      0.71      0.73       154
weighted avg       0.77      0.77      0.76       154

Accuracy= 0.7727272727272727
```

This is the prediction that whether the person is diabetic or not using svc.

```
from sklearn.preprocessing import StandardScaler
input_data = (5,116,74,0,0,25.6,0.201,30)
#changing the data to a numpy array
input_data_as_numpy_array = np.asarray(input_data)
# reshape the array as we are predicting for one instance
input_data_reshaped = input_data_as_numpy_array.reshape(1,-1)
#standardize the input data
std_data = scaler.transform(input_data_reshaped)
print(std_data)
prediction = svm_model.predict(std_data)
print(prediction)
```

```
[[ 0.3429808  -0.15318486  0.25303625 -1.28821221 -0.69289057 -0.81134119
  -0.81807858 -0.27575966]]
[0]
```

Chapter 6

PERFORMANCE COMPARISON OF THE MODELS

Performance comparison of the models was done with the accuracy score, SVC is the best model for the prediction as it is providing an accuracy of 77.27%

Classifier	*Accuracy(test data)*
SVM	*77.27%*
Random Forest Classifier	*74.02%*
Gaussian-Naïve Bayes	*77.27%*
Decision Tree	*70.77%*
Logistic Regression	*75.97%*

Chapter 7

Result Analysis

The accuracy of the models varied depending on the complexity of the algorithm and the interaction between the features. For instance, models like Random Forest and Logistic Regression showed high accuracy, indicating their robustness in handling the dataset's features and classifying diabetes effectively. The precision and recall metrics further highlighted the strengths of these models, with high precision ensuring that the instances identified as diabetic were truly diabetic, and high recall ensuring that most actual diabetic cases were correctly identified.

The F1-score, as a harmonic mean of precision and recall, provided a comprehensive view of each model's performance, confirming the effectiveness of the models in both identifying diabetic cases and minimizing false alarms.

Comparative Analysis

A comparative analysis of the models' performance showed that while some models excelled in accuracy, others performed better in precision or recall. For example, the Random Forest model demonstrated high accuracy and recall, making it highly effective in identifying diabetic cases. On the other hand, the Logistic Regression model provided a more balanced approach with strong performance across all metrics, making it a reliable choice for classification tasks.

The graphical representation of these results, along with the tabulated accuracy, precision, recall, and F1-score, allows for a clearer understanding of how each model performed relative to the others. This analysis not only highlights the strengths of each model but also points to potential areas for improvement, such as enhancing precision in models with high recall or vice versa.

Chapter 8

Conclusions and future work

In this study, various machine learning algorithms are applied on the dataset and the classification has been done using various algorithms of which Logistic Regression gives highest accuracy of 96%. Application of pipeline gave AdaBoost classifier as best model with accuracy of 98.8%. We have seen comparison of machine learning algorithm accuracies with two different datasets. It is clear that the model improves accuracy and precision of diabetes prediction with this dataset compared to existing dataset. Further this work can be extended to find how likely non diabetic people can have diabetes in next few years

Chapter 9

REFERENCES

[1] Gauri D. Kalyankar, Shivananda R. Poojara and Nagaraj V. Dharwadkar," Predictive Analysis of Diabetic Patient Data Using Machine Learning and Hadoop", International Conference On I-SMAC,978-1-5090-3243-3,2017.

[2] Ayush Anand and Divya Shakti," Prediction of Diabetes Based on Personal Lifestyle Indicators", 1st International Conference on Next Generation Computing Technologies, 978-1-4673-6809-4, September 2015.

[3] B. Nithya and Dr. V. Ilango," Predictive Analytics in Health Care Using Machine Learning Tools and Techniques", International Conference on Intelligent Computing and Control Systems, 978-1-5386-2745-7,2017.

[4] Dr Saravana kumar N M, Eswari T, Sampath P and Lavanya S," Predictive Methodology for Diabetic Data Analysis in Big Data", 2nd International Symposium on Big Data and Cloud Computing,2015.

[5] Aiswarya Iyer, S. Jeyalatha and Ronak Sumbaly," Diagnosis of Diabetes Using Classification Mining Techniques", International Journal of Data Mining & Knowledge Management Process (IJDKP) Vol.5, No.1, January 2015.

TESTING DATAOUTPUT:PREDICTED RESULTMACHINE-LEARNING ALGOATTRIBUTE SELECTIONDATA PRE-PROCESSING

INPUT DATATRAINING DATA

•