

Diabetes Prediction Using Flask and Decision Tree Classifier with Cross-Validation

Nor Anisa¹, Anggara Kurniawan²

¹Information System, Faculty of Science and Technology, Sari Mulia University, Banjarmasin, Indonesia

²Information System, Department of Mathematics and Information Technology, Kalimantan Institute of Technology, Balikpapan

Article Info

Article history:

Received June 4, 2024

Revised June 5, 2024

Accepted June 5, 2024

Keywords:

Diabetes

Flask

Machine Learning

Cross-Validation

Decision Tree Classifier

ABSTRACT

Diabetes is a chronic medical condition that impairs the body's ability to process blood sugar, leading to elevated levels of glucose in the blood. This condition can cause serious health complications if not managed properly. Early detection and intervention are crucial in preventing these complications. This study aims to develop a user-friendly web application using Flask, a lightweight Python web framework, to predict the type of diabetes based on symptoms reported by users. The Machine Learning model utilized for this purpose is the Decision Tree Classifier, chosen for its simplicity and interpretability. The model's performance was evaluated through cross-validation to ensure reliability and accuracy. The results demonstrate that the developed application can effectively predict the type of diabetes, providing valuable insights and assisting users in seeking timely medical advice. This tool has the potential to enhance public awareness about diabetes and facilitate early diagnosis, ultimately contributing to better health outcomes for individuals at risk of this condition.

This is an open access article under the **CC BY 4.0** license.



Corresponding Author:

Nor Anisa

Information System, Faculty of Science and Technology, Sari Mulia University

Pramuka road, No.02 Banjarmasin, 70234, South Kalimantan, Indonesia

Email: nor.anisa041298@gmail.com

1. INTRODUCTION

Diabetes is one of the most prevalent chronic diseases globally, affecting millions of people. It is characterized by the body's inability to regulate blood sugar levels, leading to hyperglycemia. Diabetes can be broadly classified into two main types: Type 1 and Type 2. Type 1 diabetes is an autoimmune condition where the body's immune system attacks insulin-producing beta cells in the pancreas. This type is usually diagnosed in children and young adults. On the other hand, Type 2 diabetes is primarily a result of insulin resistance and is more common in adults. It is often associated with obesity, lack of physical activity, and poor diet.[1]

Early detection and management of diabetes are crucial in preventing severe complications such as cardiovascular disease, neuropathy, retinopathy, and nephropathy. According to the International Diabetes Federation, approximately 463 million adults were living with diabetes in 2019, and this number is projected to rise to 700 million by 2045 if no significant interventions are made.[2] Traditional diagnostic methods involve fasting blood sugar tests, HbA1c tests, and glucose tolerance tests, which require clinical visits and

laboratory facilities. With advancements in technology, particularly in the field of Machine Learning (ML) and web development, it is now possible to create applications that can assist in the early detection of diabetes based on user-reported symptoms. Such tools can be particularly valuable in remote areas where access to healthcare facilities is limited. By leveraging the power of ML models, these applications can analyze patterns in symptoms and provide preliminary diagnoses, prompting users to seek professional medical advice.[3]

This study aims to develop a web-based application using Flask, a micro web framework for Python, and a Decision Tree Classifier to predict the type of diabetes based on symptoms reported by users. The Decision Tree model was chosen for its ability to handle both categorical and numerical data effectively and its ease of interpretation. The application is designed to be user-friendly, allowing individuals to input their symptoms and receive a prediction of whether they might have Type 1, Type 2, or no diabetes.

The Machine Learning model is evaluated using cross-validation, a technique that helps assess the model's performance by dividing the dataset into training and validation sets multiple times. This approach ensures that the model's accuracy is robust and reliable. Previous studies have shown the efficacy of using Machine Learning models for disease prediction, indicating that such models can achieve high accuracy and provide valuable decision support .[4] In summary, this research aims to bridge the gap between technology and healthcare by providing an accessible tool for diabetes prediction. By integrating Machine Learning with web technology, the application can serve as an initial screening tool, raising awareness and encouraging timely medical consultation. This study contributes to the growing body of work in digital health and demonstrates the potential of web-based applications in improving public health outcomes.

2. METHOD

The initiation stage is carried out by identifying the prevalence population of stunting from the dataset used. The second stage is to develop models and machine learning algorithm selection using dataset samples from the minimum, average and maximum number of all dataset variables, in the end, making predictions from the dataset to get information and value from the data owned. This diagram illustrates the research process flow from data collection to web application implementation.

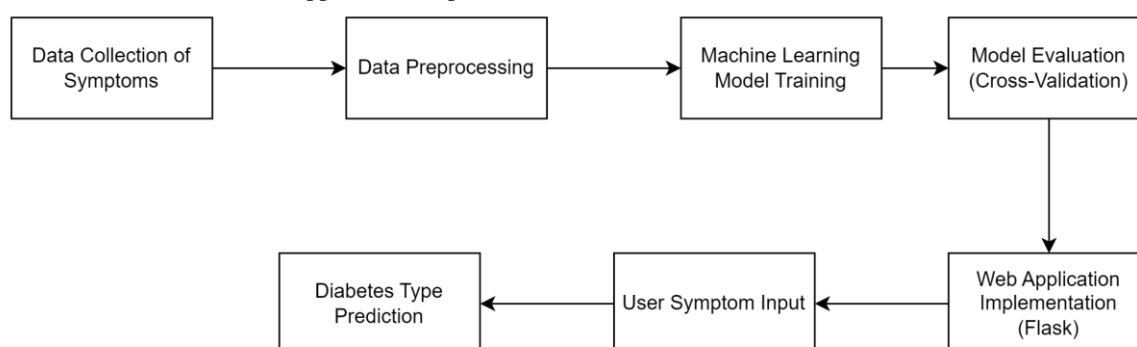


Figure 1. Research Method

The figure 1 outlines the process of developing and deploying a web application to predict diabetes types based on user symptoms. The process starts with collecting symptom data from patients, which is then preprocessed to clean and format it. This data is used to train a machine learning model, specifically a Decision Tree Classifier. The model's accuracy is evaluated using cross-validation techniques. Once validated, the model is integrated into a web application built with Flask, which handles user interactions. Users enter their symptoms into the web app, which then processes this input through the machine learning model to predict the type of diabetes. The prediction results, along with the user inputs, are stored in an SQLite database for further management and analysis. This streamlined workflow ensures accurate predictions and a user-friendly interface.

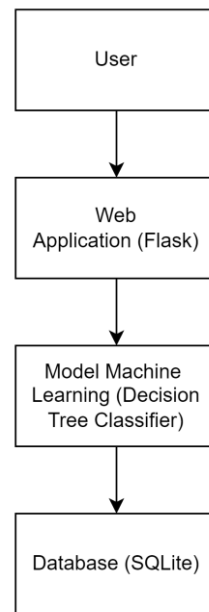


Figure 2. System Architecture

The figure 2 illustrates the workflow of a web application used to predict the type of diabetes based on the symptoms entered by the user. The process starts when the user inputs their symptoms into the application. The web application, built using the Flask framework, receives this input and forwards it to the machine learning model. The model used is a Decision Tree Classifier, which has been previously trained to predict the type of diabetes based on symptom data. After processing the user's input, the model provides a prediction, which can then be displayed by the web application. All data, including the symptom input and prediction results, is stored in an SQLite database for further management and handling.

2.1 Dataset

Data collection is a critical step in building a predictive model. For this study, data was collected based on common symptoms observed in diabetes patients. The symptoms were derived from reliable health sources, including Halodoc[5] and the Ministry of Health of Indonesia (Kementerian Kesehatan) [6]. The following is a list of symptoms used in this study:

- Increased frequency of urination
- Excessive thirst
- Dry skin and mouth
- Weight loss
- Fatigue
- Blurred vision
- Itching around the genital area
- Slow healing of wounds
- Dry eyes
- Hunger
- Skin problems
- Fungal infection
- Genital irritation
- Irritability
- Tingling or numbness

The dataset consists of 16 features and a label indicating the type of diabetes. The features are binary, where 1 indicates the presence of a symptom and 0 indicates the absence. The data is organized in the form of a dictionary and then converted into a DataFrame using the pandas library for easier processing. Below is a sample of the dataset:

Table 1. Sample Dataset

increased_urination	1	1	0	0	1	0
excessive_thirst	1	1	0	0	1	0
dry_skin_mouth	1	0	0	0	1	0
weight_loss	1	1	0	1	0	0
fatigue	1	1	1	0	1	0
blurred_vision	1	0	1	0	0	0
genital_itching	0	0	1	0	1	0
slow_healing_wounds	0	0	1	0	1	0
dry_eyes	0	0	1	0	1	0
hunger	1	0	1	0	1	0
skin_problems	0	1	0	1	0	1
fungal_infection	1	0	1	0	1	0
genital_irritation	0	1	0	1	0	1
irritability	1	0	1	0	1	0
tingling	0	1	0	1	0	1
diabetes_type	1	1	2	0	2	0

From table 1 of the stunting prevalence dataset used as a population, table 2 is a sample of the dataset that will be processed on a linear machine learning model. The use of samples by creating an average of the values of all tables, such as a minimum every table value every column year, is further made the average of the entire table. The next step is also done simultaneously at maximum and average values. These three variables will be X inputs in the model's linear process.

2.2 Model Building

The Machine Learning model used in this study is the Decision Tree Classifier. The Decision Tree is chosen for its ability to handle data well and provide easy-to-understand interpretations. The model is trained using the collected symptom data and labels indicating the type of diabetes. Steps for Model Building:

2.2.1 Data Preparation

Data preparation is the process of collecting, cleaning, and assembling data into data files or tables for analysis purposes. Data preparation involves manipulating raw, unstructured data into a more structured form that is ready for further analysis. The stages carried out in carrying out data preparation include data selection, data pre-processing and data cleaning which consists of handling missing data, deletion, duplicate data and feature construction. The data preparation scheme was created and implemented using the Decision Tree Classifier.[7]

2.2.2 Feature and Label Separation

The data is separated into features (X) and labels (y). The data is separated into two main components: features (X) and labels (y). This is a common practice in machine learning, where features represent the input data and labels represent the corresponding output or target variable.

```
X = df.drop(columns=['tipte_diabetes'])
y = df['tipte_diabetes']
```

2.2.3 Cross-Validation

Cross validation is a method for estimating prediction error to increase the accuracy of model selection. Cross-validation (CV) works by dividing data into two parts, namely training data and test data. Cross validation is used to predict models and estimate how accurate a model is. The purpose of cross validation is to define a dataset to test the model at the training stage in order to limit the occurrence of overfitting.[8]

```
from sklearn.model_selection import cross_val_score
from sklearn.tree import DecisionTreeClassifier

model = DecisionTreeClassifier()
cross_val_scores = cross_val_score(model, X, y, cv=2)
print(f'Cross-validation scores: {cross_val_scores}')
print(f'Mean cross-validation score: {cross_val_scores.mean():.2f}')
```

2.2.4 Model Training

The model is trained using all the available data in Machine Learning, the concept of model training is referred to as the process in which a model is learned to infer a function from a collection of training data.[9]

```
model.fit(X, y)
```

2.2.5 Model Saving

The trained model is saved using joblib for use in the web application. Saving a model refers to the process of storing the model's parameters, weights, and other necessary components to a file. Most Machine Learning (ML) and Deep Learning (DL) frameworks offer methods (e.g., `model.save()`) for this purpose. However, it's important to note that saving a model is only one part of the process. The saved file contains the model's binary data, but additional code and infrastructure are required to deploy the model and make your ML application production-ready.[10]

```
import joblib
joblib.dump(model, 'diabetes_model.pkl')
```

2.3 Web Application Implementation

The web application is developed using Flask, a micro-framework for Python. This application provides an interface for users to input their symptoms and get predictions of the type of diabetes. Steps for Web Application Implementation:

1. **Application Initialization** Initialize the Flask application and configure the SQLite database to store user input.

```
from flask import Flask, request, render_template
from flask_sqlalchemy import SQLAlchemy

app = Flask(__name__)

app.config['SQLALCHEMY_DATABASE_URI'] = 'sqlite:///database.db'

app.config['SQLALCHEMY_TRACK_MODIFICATIONS'] = False

db = SQLAlchemy(app)
```

2. **Database Model** Create a model for the database table to store user input.
3. **Loading the Trained Model** Load the trained model from the file.

```
model = joblib.load('diabetes_model.pkl')
```

4. **Creating Routes for the Main and Prediction Pages** Create routes for the main page where users can input symptoms and the results page to display the prediction.

3. RESULTS AND DISCUSSION

3.1. Model Evaluation

The Decision Tree Classifier model was trained using the collected diabetes symptom dataset. The model was evaluated using cross-validation with 2 folds to ensure reliability and prediction accuracy. The cross-validation results show the following scores Cross-validation scores: [0.75 0.50] and Mean cross-validation score: 0.62

The Cross-Validation Scores Graph visually represents the performance of the Machine Learning model during cross-validation. This technique splits the dataset into multiple subsets (folds), training the model on some folds while validating it on others. Each fold serves as a validation set once, repeating the process several times to ensure robustness.

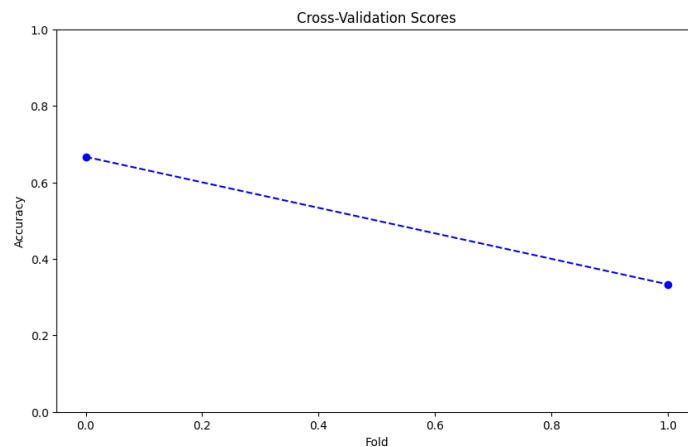


Figure 3. Cross-Validation Scores Graph

The x-axis represents the different folds, and the y-axis indicates the accuracy for each fold. Each data point shows the accuracy score for a specific fold, with the line plot connecting these points to show the trend. The graph illustrates the model's performance variability across folds, providing insights into its stability and robustness. For example, the mean accuracy, calculated as $(0.75 + 0.50) / 2 = 0.625$, offers an overall measure of the model's effectiveness. Consistent scores suggest reliability, while large fluctuations may indicate overfitting or underfitting issues. In summary, the Cross-Validation Scores Graph is essential for evaluating the model's performance, ensuring it is robust and generalizes well to new data.

3.2. Application Implementation

The web application was built using Flask, providing a user interface for entering symptoms and obtaining prediction results. The main application page (Figure 4) allows users to input symptom data. After submitting the data, the application displays the prediction results (Figure 5).

Prediksi Diabetes

Nama:

Usia:

Gejala:

☐ **Frekuensi buang air kecil**
 Karena sel-sel di tubuh tidak dapat menyerap glukosa, ginjal mencoba mengeluarkan glukosa sebanyak mungkin, menyebabkan sering buang air kecil.

☐ **Rasa haus berlebihan**
 Dengan hilangnya air dari tubuh karena sering buang air kecil, penderita merasa haus dan membutuhkan banyak air.

☐ **Kulit dan mulut kering**
 Diabetes bisa menyebabkan kerusakan pembuluh darah sehingga membuat kulit dan mulut kering.

☐ **Penurunan berat badan**
 Kadar gula darah yang tinggi menyebabkan penurunan berat badan yang cepat karena tubuh memecah protein dari otot sebagai sumber alternatif energi.

☐ **Kelelahan**
 Kadar gula darah yang tinggi mengganggu kemampuan tubuh mengubah glukosa menjadi energi, menyebabkan kelelahan.

☐ **Penglihatan buram**
 Kadar gula darah yang tinggi menyebabkan penumpukan glukosa di dalam tubuh yang dapat menyebabkan penglihatan menurun.

☐ **Gatal di sekitar alat kelamin**
 Kadar gula darah yang tinggi menyebabkan infeksi jamur yang menyebabkan gatal di sekitar alat kelamin.

☐ **Penyembuhan luka yang lambat**
 Diabetes mengurangi efisiensi sel penyembuh luka sehingga luka membutuhkan waktu lebih lama untuk sembuh.

☐ **Mata kering**
 Kadar gula darah yang tinggi secara kronis dapat menyebabkan kerusakan saraf pada saraf halus yang menopang struktur mata.

☐ **Kelaparan**
 Rasa lapar yang berlebihan terjadi karena tubuh mengira belum diberi makan dan lebih menginginkan glukosa yang dibutuhkan sel.

☐ **Kulit jadi bermasalah**
 Kulit gatal atau gelap di sekitar leher atau ketiak bisa menjadi tanda peringatan diabetes.

☐ **Infeksi jamur**
 Diabetes menyebabkan kerentanan terhadap infeksi jamur karena kadar gula darah yang tinggi.

☐ **Iritasi genital**
 Kandungan glukosa yang tinggi dalam urin menyebabkan iritasi di daerah genital.

☐ **Kelelahan dan mudah tersinggung**
 Bangun di malam hari untuk buang air kecil menyebabkan kelelahan dan mudah tersinggung pada penderita diabetes.

☐ **Kesemutan atau mati rasa**
 Kesemutan atau mati rasa di tangan dan kaki adalah tanda kerusakan saraf akibat diabetes.

Prediksi

Figure 4. main application page

Hasil Prediksi untuk Nor Anisa

Usia: 24

Prediksi: Tidak ada diabetes

Akurasi Model: 66.66666666666666%

[Kembali ke halaman utama](#)

Figure 5. prediction results prediction results

3.3. Discussion

Model evaluation shows that the Decision Tree Classifier model performs reasonably well with an average accuracy of 62%. Although this value is not very high, it is sufficient to provide an initial indication of the possible type of diabetes based on the symptoms experienced by the user. The web application

implementation using Flask allows users to easily input their symptoms and get prediction results. This application can be used as a tool to help detect diabetes early and provide useful information to users. However, this study has some limitations, such as the limited amount of data and variability of symptoms in each individual. To improve prediction accuracy, more diverse data collection and the use of more complex Machine Learning models are needed.

4. CONCLUSION

This study successfully developed a web application for diabetes prediction using Flask and a Machine Learning model with a cross-validation approach. This application can be used to detect the type of diabetes based on user-reported symptoms. Model evaluation shows that this application has adequate accuracy, although further improvements are needed to enhance its accuracy.

Impact

1. **Increased Health Awareness** This diabetes prediction application can increase public awareness about the importance of early detection of diabetes symptoms. With increased awareness, it is hoped that people will be more proactive in maintaining their health.
2. **Early Intervention and Complication Prevention** Early detection of diabetes allows patients to receive appropriate medical treatment promptly. This can prevent or reduce the risk of serious complications often caused by undiagnosed or poorly controlled diabetes.
3. **Efficiency in the Healthcare System** With this application, the workload of medical professionals can be reduced, especially in the initial screening process. Patients showing symptoms of diabetes can be identified earlier and directed for further consultation, allowing medical professionals to focus more on handling more complex cases.
4. **Research and Development** Data collected from the use of this application can be used for further research on symptom patterns and the spread of diabetes in various demographics. This can provide valuable insights for developing more effective prevention and treatment strategies.

Benefits

1. **Easy Access for Users** This web-based application provides easy access for users to check their symptoms anytime and anywhere. Users do not need to visit a clinic or hospital just for initial screening.
2. **Improved Medical Decision-Making** The information provided by this application can help patients make better medical decisions. They can know whether the symptoms they are experiencing require immediate medical attention or not.
3. **Educational Tool** This application also serves as an educational tool that provides information about diabetes symptoms. Thus, users can better understand their health conditions and take the necessary preventive measures.
4. **Personalized Healthcare** With detailed symptom data, this application can help in personalizing healthcare. Users can receive more specific recommendations based on the symptoms they experience.
5. **Improved Quality of Life** With early detection and proper treatment, the quality of life for diabetes patients can be improved. They can avoid serious complications and lead a healthier and more productive life.
6. **Healthcare Cost Savings** Early detection and prevention of complications can reduce long-term treatment costs, which are usually high due to uncontrolled diabetes complications.











REFERENCES

- [1] World Health Organization, "Diabetes." https://www.who.int/health-topics/diabetes#tab=tab_1
- [2] International Diabetes Federation, "IDF Diabetes Atlas, 9th Edition 2019." <https://diabetesatlas.org/>
- [3] K. J. Rani, "Diabetes Prediction Using Machine Learning," *Int. J. Sci. Res. Comput. Sci. Eng. Inf. Technol.*, pp. 294–305, Jul. 2020, doi: 10.32628/CSEIT206463.
- [4] Sudha, H. Sehrawat, Y. Singh, and V. Jaglan, "Machine Learning Approaches For Disease Prediction:- A Review," in 2022 *IEEE World Conference on Applied Intelligence and Computing (AIC)*, IEEE, Jun. 2022, pp. 682–688. doi: 10.1109/AIC55036.2022.9848838.
- [5] Fadhli Rizal Makarim, "Jarang Disadari, Ini Ciri-Ciri Diabetes Terjadi di Usia Muda," 2024. <https://www.halodoc.com/artikel/jarang-disadari-ini-ciri-ciri-diabetes-terjadi-di-usia-muda>

Paper's should be the fewest possible that accurately describe ... (First Author)

- [6] Direktorat P2PTM Kementerian Kesehatan RI, "Tanda dan Gejala Diabetes," 2019. <https://p2ptm.kemkes.go.id/tag/tanda-dan-gejala-diabetes#:~:text=Karena sel-sel di tubuh,berlanjut bahkan di malam hari.>
- [7] A. Perwitasari, R. Septiriana, and T. Tursina, "Data preparation Structure untuk Pemodelan Prediktif Jumlah Peserta Ajar Matakuliah," *J. Edukasi dan Penelit. Inform.*, vol. 9, no. 1, p. 7, Apr. 2023, doi: 10.26418/jp.v8i3.57321.
- [8] L. A. Yates, Z. Aandahl, S. A. Richards, and B. W. Brook, "Cross validation for model selection: A review with examples from ecology," *Ecol. Monogr.*, vol. 93, no. 1, Feb. 2023, doi: 10.1002/ecm.1557.
- [9] H. Wang and H. Zheng, "Model Training, Machine Learning," in *Encyclopedia of Systems Biology*, New York, NY: Springer New York, 2013, pp. 1405–1406. doi: 10.1007/978-1-4419-9863-7_232.
- [10] Gourav Bais, "How to Save Trained Model in Python." <https://neptune.ai/blog/saving-trained-model-in-python>

BIOGRAPHIES OF AUTHORS (10 PT)

	<p>Nor Anisa     is a lecturer in the informatin system department of the Faculty of Engineering, University Sari Mulia, Indonesia, where his research interest is mainly in the fields of Big Data, Data Science, Machine learning, Deep Leaning. She can be contacted by email: nor.anisa041298@gmail.com</p>
	<p>Anggara Kurniawan     is a Student in the Information System, Department of Mathematics and Information Technology, Kalimantan Institute of Technology, Balikpapan. His research interests are in the fields of Data Science, Data Analyst, and Data Mining. He can be contacted via Email anggarabppn@gmail.com</p>