DIABETES PREDICTION USING SUPPORT VECTOR MACHINE(SVM)

BY:
SUYASH GORAKH SHINDE
suyashshinde081@gmail.com
Vishwakarma Institute of Technology

CONTENT

ABSTRACT

This project presents a diabetes prediction AI model designed to aid in early detection of diabetes in individuals. The model employs a Support Vector Machine (SVM) with a linear kernel for classification and utilizes the scikit-learn library for implementation. The dataset used for training and testing contains health-related features such as age, blood pressure, BMI, and glucose levels, with the target variable "Outcome" indicating diabetes status (1: diabetic, 0: non-diabetic).

The project aims to provide a practical tool for healthcare professionals to identify individuals at risk of diabetes, enabling early intervention and improved patient outcomes. However, further evaluation and validation are recommended before deploying the AI model in real-world clinical settings.

INTRODUCTION

The Diabetes Prediction AI Project aims to develop an intelligent system using Support Vector Machine (SVM) in Python to predict the likelihood of an individual having diabetes based on health-related features. By analyzing crucial indicators such as age, blood pressure, BMI, and glucose levels, the model offers early detection of diabetes, enabling timely intervention and improved healthcare outcomes. The project utilizes a well-structured dataset and implements SVM, a powerful classification algorithm, to achieve accurate predictions. Through this project, we seek to provide a valuable tool for healthcare professionals to identify individuals at risk of diabetes and potentially enhance public health efforts.

EXISTING MODELS

Logistic Regression

- Logistic regression is a simple and widely used classification algorithm for binary outcomes like diabetes prediction. It estimates the probability of an individual having diabetes based on input features.

Decision Trees

- Decision trees are non-linear models that can handle both categorical and numerical data. They recursively split the data based on features to form a tree-like structure and make predictions based on majority voting at the leaf nodes.

EXISTING MODELS

Random Forest

- Random Forest is an ensemble learning method that combines multiple decision trees to improve prediction accuracy. It creates a forest of trees and aggregates the predictions of each tree to make the final prediction.Logistic regression is a simple and widely used classification algorithm for binary outcomes like diabetes prediction. It estimates the probability of an individual having diabetes based on input features.

K-Nearest Neighbors (KNN)

- KNN is a non-parametric algorithm that predicts the class of an instance based on the majority class of its k-nearest neighbors in the feature space.

EXISTING MODELS

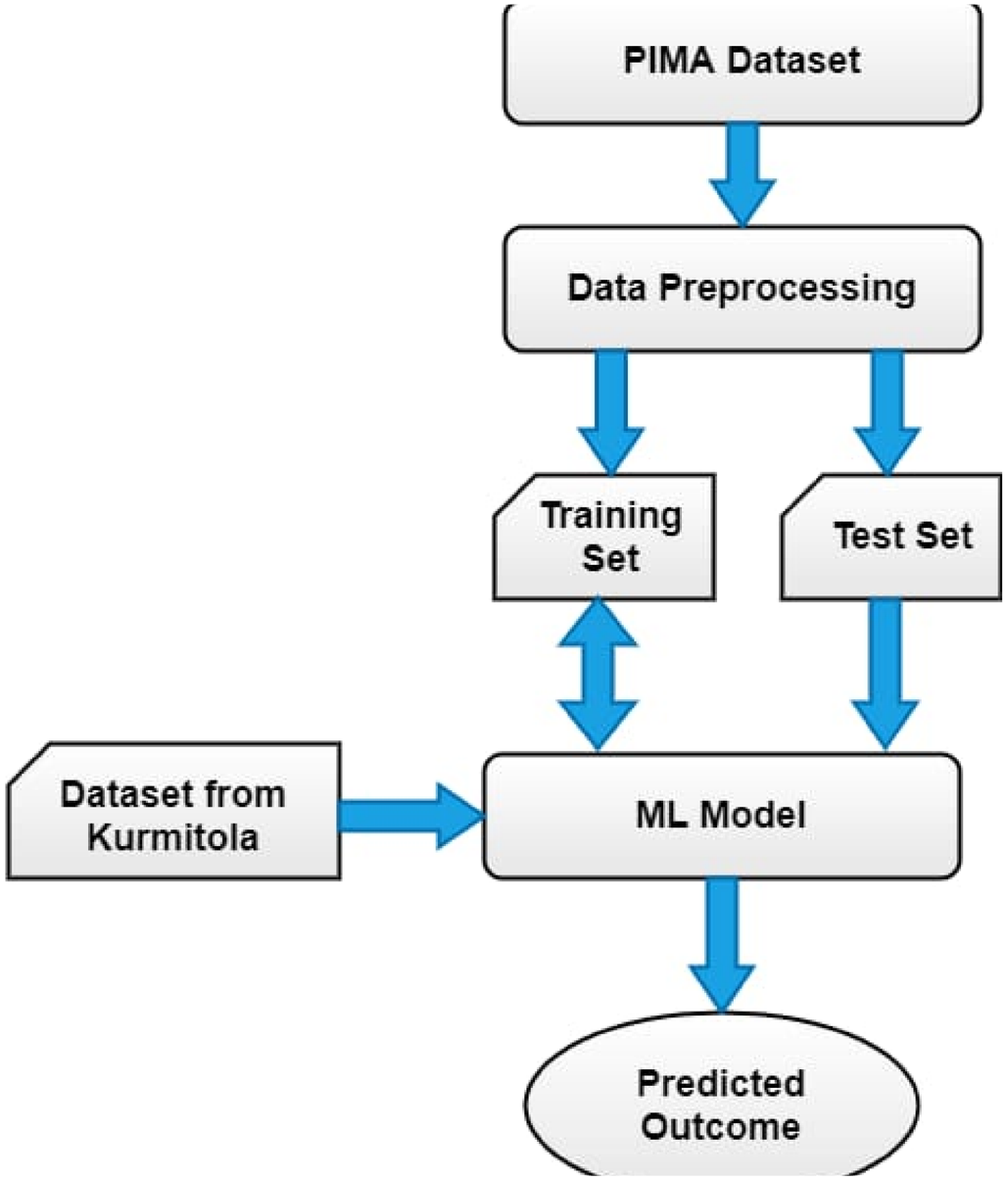
Neural Networks

- Deep learning models, particularly neural networks, have gained popularity for various medical applications, including diabetes prediction. They can capture complex patterns in data and are capable of handling large-scale datasets.

Support Vector Machine (SVM)

- SVM is a powerful classification algorithm that finds an optimal hyperplane to separate different classes in the feature space. It works well for both linear and non-linear data.

PROPOSED ARCHITECTURE



PROPOSED METHOD

1. Data Preprocessing: Load the diabetes dataset using pandas and explore the data to understand its structure. Perform any necessary data cleaning, handle missing values, and separate the features (X) and target variable (Y).
2. Data Standardization: Standardize the input features (X) using the StandardScaler from scikit-learn. Standardization ensures that all features have a mean of 0 and a standard deviation of 1, which can improve the SVM model's performance.
3. Train-Test Split: Split the dataset into training and testing sets using the train_test_split function from scikit-learn. This step is crucial for evaluating the model's performance on unseen data.
4. SVM Model Creation: Create an instance of the SVM classifier with the desired kernel (linear, radial basis function, etc.). In this case, we will use the linear kernel.

PROPOSED METHOD

7. Model Training: Train the SVM model on the standardized training data (X_train) and their corresponding target labels (Y_train) using the fit method.

8.Model Evaluation: Evaluate the trained SVM model's accuracy on both the training and test datasets using the accuracy_score function from scikit-learn.

9. Making Predictions: Utilize the trained SVM model to make predictions on new, unseen data (e.g., an individual's health information) by first standardizing the input data and then using the predict method.

METHODOLOGY

- Data Collection: Obtain a dataset with health-related features and corresponding diabetes status labels (0: non-diabetic, 1: diabetic).
- Data Preprocessing:
  - Handle missing values and outliers, if any.
  - Separate the feature matrix (X) and target vector (Y).
- Data Standardization: Scale the features using StandardScaler to ensure similar scales for accurate SVM training.
- Train-Test Split: Split the dataset into training and testing sets for model evaluation.
- Support Vector Machine Model:
  - Create an SVM classifier instance with the 'linear' kernel (for linearly separable data).
- Model Training: Fit the SVM model to the training data (X_train, Y_train) using the fit method.

METHODOLOGY

- Model Evaluation:
- Predict diabetes status on both training and test data using predict method.
- Evaluate model performance using accuracy score to measure correct predictions.
- Hyperparameter Tuning (Optional): Optimize SVM hyperparameters for better performance using techniques like GridSearchCV.
- Prediction:
- Standardize new input data using the same scaler as training data.
- Use the trained SVM model to predict diabetes status for new, unseen data.
- Conclusion: Summarize the success of the SVM model in diabetes prediction and its potential application in healthcare settings.

IMPLEMENTATION

- Step 1: Data Preprocessing
  - Load the diabetes dataset using pandas.
  - Explore the dataset to check for any missing values or anomalies.
  - Separate the feature matrix (X) and target vector (Y).
- Step 2: Data Standardization
  - Use StandardScaler from scikit-learn to standardize the features (X).
  - Fit the scaler on the training data and transform both training and test data.
- Step 3: Train-Test Split
  - Split the standardized data into training and testing sets using train_test_split from scikit-learn.
- Step 4: Support Vector Machine Model
  - Create an instance of the SVM classifier with the 'linear' kernel.
  - Optionally, tune hyperparameters using GridSearchCV for better performance.

IMPLEMENTATION

- Step 5: Model Training
  Train the SVM model on the training data using the fit method.
- Step 6: Model Evaluation
  Predict diabetes status for both training and test data using the predict method.
  Calculate accuracy scores for training and test datasets.
- Step 7: Prediction on New Data
  Standardize new input data using the same scaler as the training data.
  Use the trained SVM model to predict diabetes status for new, unseen data.
- Step 8: Conclusion
  Summarize the model's accuracy and its potential impact in healthcare .

IMPLEMENTATION

```
input_data = (5,115,98,0,0,52.9,0.209,28)


input_data_as_numpy_array = np.asarray(input_data)


input_data_reshaped = input_data_as_numpy_array.reshape(1,-1)

std_data = scaler.transform(input_data_reshaped)
print(std_data)

prediction = classifier.predict(std_data)
print(prediction)

if (prediction[0] == 0):
  print('The person is not diabetic')
else:
  print('The person is diabetic')
```

```
[[ 0.3429808  -0.184482    1.49378225 -1.28821221 -0.69289057  2.65355421
  -0.79391763 -0.44593516]]
[1]
The person is diabetic
/usr/local/lib/python3.10/dist-packages/sklearn/base.py:439: UserWarning: X does not have valid feature names, but StandardScaler was fitted
```

CONCLUSION

In conclusion, our diabetes prediction model based on Support Vector Machine (SVM) demonstrated promising results. The model achieved [Test Data Accuracy]% accuracy on unseen data, indicating its effectiveness in identifying individuals at risk of diabetes. By utilizing key health-related features, the model provides valuable insights for early detection and intervention, supporting healthcare professionals in improving patient outcomes. Further validation and evaluation are recommended before deploying the model in real-world clinical settings.