From Data to Diagnosis: Utilizing Machine Learning for Accurate Diabetes Prediction

Priyam Deepak Choksi

Northeastern University, Boston, MA, USA
choksi.pr@northeastern.edu

Diabetes is a chronic disease with significant health implications worldwide. Early diagnosis and prediction of diabetes are essential for effective management and prevention of complications. This paper presents the development of a web-based application designed to predict the likelihood of diabetes using various machine learning algorithms, with a focus on Logistic Regression. The paper discusses the rationale behind algorithm selection, model training, performance evaluation, and the development of a user-friendly interface. Additionally, we explore alternative algorithms and compare their performance to justify the choice of Logistic Regression for this application.

***Index Terms*—Diabetes Prediction, Logistic Regression, Machine Learning, Web Application, Exploratory Data Analysis (EDA).**

I. Introduction

DIABETES affects millions globally, and its prevalence is rising. Accurate prediction of diabetes based on clinical and demographic data can enable early intervention, thereby reducing the risk of severe health outcomes. This project developed a comprehensive web application to predict diabetes likelihood using several machine learning algorithms, with a final choice of Logistic Regression. The application also features Exploratory Data Analysis (EDA) to provide insights into the dataset.

II. Methodology

A. Dataset

The dataset utilized for this project comprises 100,000 rows of clinical data, including features such as age, BMI, glucose levels, HbA1c levels, gender, race, smoking history, hypertension, and heart disease. The target variable is binary, indicating whether the individual has diabetes. This dataset was sourced from anonymized medical records, ensuring data privacy and security.

B. Mathematical Foundation of Algorithms

1) Logistic Regression

Logistic Regression is a linear model for binary classification, where the probability of the dependent variable belonging to a particular class (in this case, diabetic or non-diabetic) is modeled as a logistic function (sigmoid function) of a linear combination of the independent variables.

The probability that a given data point belongs to class 1 (diabetic) is given by:

$$P(y = 1 \mid X) = \sigma(X\beta) = \frac{1}{1 + e^{-X\beta}} \tag{1}$$

where X represents the input features, β is the vector of coefficients (weights), and σ is the sigmoid function.

The decision boundary is determined by the equation:

$$X\beta = \log\left(\frac{P(y = 1)}{1 - P(y = 1)}\right) \tag{2}$$

This model is particularly suitable for binary classification due to its interpretability. The coefficients β_i represent the log-odds of the outcome (diabetes) for a one-unit increase in the corresponding feature X_i.

2) Support Vector Machines (SVM)

SVM constructs a hyperplane in a high-dimensional space that separates the data into two classes. The algorithm maximizes the margin between the closest data points (support vectors) from both classes.

The SVM optimization problem is formulated as:

$$\min_{\mathbf{w},b} \frac{1}{2}\|\mathbf{w}\|^2 \quad \text{subject to} \quad y_i(\mathbf{w} \cdot \mathbf{x}_i + b) \geq 1, \forall i \tag{3}$$

where $\mathbf{w}$ is the weight vector and b is the bias.

3) Random Forest

Random Forest is an ensemble learning method that constructs multiple decision trees during training and outputs the mode of the classes for classification.

$$h(X) = \text{mode}\{h_1(X), h_2(X), \ldots, h_n(X)\} \tag{4}$$

where $h_i(X)$ represents the prediction from the i-th tree.

4) K-Nearest Neighbors (KNN)

KNN is a non-parametric algorithm that classifies a data point based on the majority class of its k-nearest neighbors in the feature space.

$$y = \text{mode}\{y_1, y_2, \ldots, y_k\} \tag{5}$$

where y_i represents the class of the i-th nearest neighbor.

5) Gradient Boosting

Gradient Boosting models, such as XGBoost, build trees sequentially, with each tree attempting to correct the errors of the previous one. The algorithm minimizes a differentiable loss function using gradient descent.

$$\min \sum_{i=1}^{n} L(y_i, F_m(x_i)) + \sum_{m=1}^{M} \Omega(f_m) \tag{6}$$

where L is the loss function, F_m is the prediction from the m-th model, and Ω is a regularization term to prevent overfitting.

C. Model Selection and Training

1) Data Preprocessing

Numerical features (e.g., age, BMI) were standardized using StandardScaler. Categorical features (e.g., gender, smoking history) were encoded using OneHotEncoder.

2) Train-Test Split

The dataset was split into an 80% training set and a 20% testing set to evaluate model performance.

3) Model Training

The Logistic Regression model was trained using the maximum likelihood estimation (MLE) method to estimate the coefficients β. The optimization problem solved was:

$$\min_{\beta} \sum_{i=1}^{n} [y_i \log(\sigma(X_i\beta)) + (1 - y_i) \log(1 - \sigma(X_i\beta))] \tag{7}$$

Regularization (L2 norm) was used to prevent overfitting, modifying the objective function to include a penalty term:

$$\min_{\beta} \left\{ \sum_{i=1}^{n} [y_i \log(\sigma(X_i\beta)) + (1 - y_i) \log(1 - \sigma(X_i\beta))] + \lambda \|\beta\|^2 \right\} \tag{8}$$

where λ is the regularization parameter that controls the trade-off between fitting the training data and keeping the model parameters small.

4) Model Evaluation

The model was evaluated using accuracy and ROC AUC score:

$$\text{Accuracy} = \frac{\text{Number of correct predictions}}{\text{Total number of predictions}} \tag{9}$$

$$\text{ROC AUC} = \frac{1}{2} \sum_{i=1}^{n} \left(\frac{TPR_i + TPR_{i-1}}{FPR_i - FPR_{i-1}} \right) \tag{10}$$

where TPR is the true positive rate and FPR is the false positive rate. The model achieved:

- **Accuracy:** 0.9597
- **ROC AUC:** 0.9587

III. Exploratory Data Analysis (EDA)

Exploratory Data Analysis (EDA) was conducted to understand the dataset better, identify correlations, and detect any potential issues such as outliers or missing values. Key visualizations and statistical summaries were generated using Matplotlib and Plotly. This process was crucial for feature selection and understanding the data distribution, which directly influenced model selection and training.

IV. Model Selection

Initially, various machine learning algorithms were considered for predicting diabetes, including:

- Logistic Regression
- Support Vector Machines (SVM)
- Random Forest
- K-Nearest Neighbors (KNN)
- Gradient Boosting

Each of these algorithms was evaluated based on the following criteria:

1) **Performance on Binary Classification:** The nature of the target variable (diabetic or non-diabetic) makes binary classification algorithms suitable.
2) **Interpretability:** For clinical applications, model interpretability is crucial. It is essential to understand how each feature contributes to the prediction.
3) **Scalability:** Given the large dataset, the algorithm must efficiently scale to handle substantial data.
4) **Ease of Deployment:** The chosen model should be easy to deploy in a web application environment.

A. Algorithm Comparison

- **Logistic Regression:** Logistic Regression was selected for its simplicity, interpretability, and effectiveness in binary classification problems. It allows for easy interpretation of feature importance through coefficients and is computationally efficient, making it suitable for large datasets.
- **Support Vector Machines (SVM):** SVMs were considered due to their robustness in classification tasks, especially with high-dimensional data. However, SVMs are less interpretable, and their performance gains over Logistic Regression were marginal, making them less desirable for this application.
- **Random Forest:** Random Forest provides high accuracy and handles non-linear relationships well. However, it is less interpretable, with feature importance being more complex to understand than Logistic Regression's coefficients. Additionally, Random Forests require more computational resources, which could impact the application's performance.
- **K-Nearest Neighbors (KNN):** KNN is simple to understand and implement. However, it is computationally expensive for large datasets and lacks interpretability. Its performance also significantly depends on the choice of 'k', and it struggles with imbalanced data.
- **Gradient Boosting:** Gradient Boosting techniques like XGBoost were considered for their high accuracy and

ability to handle complex relationships. However, similar to Random Forest, these models are less interpretable and require careful tuning of hyperparameters, making them more complex to implement and maintain.

B. Model Training

Given the considerations above, Logistic Regression was chosen as the primary model. The training process involved the following steps:

- **Data Preprocessing:** The dataset was preprocessed using a combination of scaling and encoding techniques. Numerical features (e.g., age, BMI) were scaled using StandardScaler, while categorical features (e.g., gender, smoking history) were encoded using OneHotEncoder.
- **Train-Test Split:** The dataset was split into training and testing sets, with 80% used for training and 20% for testing.
- **Model Training:** Logistic Regression was trained on the processed data with a maximum iteration of 1000 to ensure convergence. The model's coefficients were examined to understand the impact of each feature on the prediction.
- **Model Evaluation:** The model's performance was evaluated using accuracy and ROC AUC scores. The results were as follows:
  - Accuracy: 0.9597
  - ROC AUC: 0.9587

  These results indicate a high level of accuracy and a robust model capable of predicting diabetes effectively.

C. Alternative Algorithms

- **Support Vector Machines (SVM):** The SVM model performed similarly to Logistic Regression in terms of accuracy but was less interpretable. It was therefore not chosen despite its robustness.
- **Random Forest:** Random Forest provided slightly higher accuracy but was more challenging to interpret, making it less suitable for clinical applications where understanding feature importance is crucial.
- **Gradient Boosting:** Gradient Boosting models like XGBoost offered excellent accuracy but at the cost of increased complexity and reduced interpretability. Given the marginal performance gain over Logistic Regression, it was not selected.
- **K-Nearest Neighbors (KNN):** KNN's performance was lower than the other models, and it was computationally intensive, especially with the large dataset, leading to its exclusion.

V. Predictive Model Integration

The trained Logistic Regression model was integrated into a web application using Streamlit. The application features a user-friendly interface where users can input health metrics such as age, BMI, and glucose levels. The model then provides real-time diabetes risk assessment.

A. Mathematical Integration

The application calculates the predicted probability using the logistic function:

$$P(y = 1 \mid X) = \sigma(X\beta) \tag{11}$$

This probability is then used to classify the user as diabetic or non-diabetic. The application also visualizes the input data against standard health metrics, providing context for the predictions.

VI. Results and Discussion

The Logistic Regression model, supported by rigorous mathematical reasoning, proved to be the most appropriate algorithm for this diabetes prediction application. Its high accuracy, coupled with clear interpretability, makes it a reliable tool for early diabetes diagnosis. While alternative algorithms offered slight performance improvements, their complexity and lack of transparency outweighed the benefits.

VII. User Interface

The web application was built using Streamlit, chosen for its simplicity and quick deployment capabilities. The application interface is designed to be user-friendly, allowing users to input health metrics through sliders and radio buttons. The application immediately calculates the probability of diabetes and provides visual feedback on how the user's metrics compare to standard health ranges.

VIII. Predictive Analysis

The application uses the trained Logistic Regression model to predict diabetes probability. The predictions are displayed in real-time, with a visual indicator showing whether the user is likely diabetic or not based on their input. The application also provides insights into how each input metric influences the prediction, leveraging the model's coefficients.

IX. Exploratory Data Analysis (EDA)

The EDA component allows users to explore the underlying dataset used for model training. Various visualizations provide insights into the relationships between different health metrics and diabetes. This feature helps users understand the data and the model's decision-making process.

X. Results and Discussion

The Logistic Regression model outperformed other algorithms in terms of simplicity, interpretability, and ease of deployment. While alternative models like Random Forest and Gradient Boosting provided marginally higher accuracy, their complexity and lack of transparency made them less suitable for this application. The application's real-time prediction capability, coupled with EDA, makes it a powerful tool for early diabetes detection and awareness.

XI. Conclusion

This research demonstrated the effectiveness of Logistic Regression for predicting diabetes in a clinical dataset. The choice of Logistic Regression was justified by its balance between accuracy, interpretability, and ease of use. The web application developed offers a valuable tool for users to assess their diabetes risk and gain insights into their health metrics. Future work could involve integrating additional features such as diet and physical activity levels, or exploring ensemble methods to improve accuracy further.

XII. Future Work

Further enhancements could include:

- Integrating additional features like genetic data or lifestyle factors.
- Exploring ensemble methods to combine the strengths of different algorithms.
- Expanding the application to include more detailed health analysis and recommendations.

References

[1] G. James, D. Witten, T. Hastie, and R. Tibshirani, *An Introduction to Statistical Learning: With Applications in R*, Springer, 2013.
[2] W. McKinney, *Hands-On Exploratory Data Analysis with Python*, Packt Publishing Ltd, 2017.
[3] F. Pedregosa, et al., "Scikit-learn: Machine Learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.
[4] J. Friedman, T. Hastie, and R. Tibshirani, *The Elements of Statistical Learning: Data Mining, Inference, and Prediction*, Springer, 2001.
[5] Streamlit Documentation. Available: https://docs.streamlit.io
[6] Matplotlib Documentation. Available: https://matplotlib.org
[7] Plotly Documentation. Available: https://plotly.com