DIABETES PREDICTION USING MACHINE LEARNING

Presented By:
1.Student Name – Muthukumaran E
2.Collage Name – Periyar University,Salem
3.Department – Statistics

edunet foundation

OUTLINE

- Problem Statement
- Proposed System/Solution
- System Development Approach
- Algorithm & Deployment
- Result
- Conclusion
- Future Scope
- References

PROBLEM STATEMENT

Diabetes is a significant global health issue, characterized by a rising prevalence that underscores the urgent need for early detection and management. Effective intervention relies on the ability to predict diabetes risk accurately, which poses a challenge due to the complexity of health data and the necessity for timely predictions. The current challenge is to develop a robust machine learning model capable of analyzing patient data to predict diabetes risk. This model should efficiently handle various data features, including glucose levels, blood pressure, and other relevant health indicators, to provide actionable insights for early diagnosis and prevention. The objective is to enhance the accuracy of diabetes predictions and support proactive health management through data-driven solutions.

PROPOSED SOLUTION

- The proposed system aims to address the challenge of predicting diabetes risk accurately based on patient data. This involves leveraging machine learning techniques to analyze various health indicators and forecast the likelihood of diabetes. The solution will consist of the following components:
- Data Collection:
  - Utilize a reliable dataset, such as the Kaggle Diabetes dataset, which includes various health features relevant to diabetes prediction.
  - Collect data including glucose levels, blood pressure, skin thickness, insulin levels, BMI, age, and diabetes pedigree function.
- Data Preprocessing:
  - Handle missing values and outliers by replacing them with appropriate statistics (mean, median) or using imputation techniques.
  - Normalize the data to ensure uniformity in the scale of features for effective model training.
- Machine Learning Algorithm:
  - Implement a machine learning algorithm, such as Random Forest Classifier, for its robustness and high accuracy in classification tasks.
  - Use GridSearchCV to find the best parameters for the model to optimize its performance.
- Deployment:
  - Develop a prediction function that allows users to input their health parameters and receive a risk prediction.
  - Deploy the solution in an environment that ensures accessibility and responsiveness, potentially integrating it into a web or mobile application.
- Evaluation:
  - Assess the model's performance using metrics such as accuracy, confusion matrix, and classification report.
  - Fine-tune the model based on evaluation results and user feedback to enhance prediction accuracy and reliability.

SYSTEM APPROACH

Technology Used:

- Data Import and Exploration:
  - Libraries: Pandas, Numpy.
- Data Preprocessing:
  - Techniques: Handling missing values, feature scaling.
  - Libraries: Scikit-learn.
- Model Development:
  - Algorithms: Random Forest Classifier.
  - Hyperparameter Tuning: GridSearchCV.
- Visualization:
  - Libraries: Matplotlib, Seaborn.

ALGORITHM & DEPLOYMENT

- Algorithm Selection:
  - Random Forest is selected due to its robustness, ability to handle complex data structures, and high accuracy in classification tasks. It is particularly effective for predicting diabetes risk based on a variety of health indicators.
- Data Input:
  - Historical patient data including glucose levels, blood pressure, skin thickness, insulin levels, BMI, age, and diabetes pedigree function.
- Training Process:
  - The model is trained using historical data. Techniques such as cross-validation and GridSearchCV are employed to optimize hyperparameters and ensure model accuracy.
- Prediction Process:
  - The trained Random Forest model is used to predict diabetes risk for new data entries. The model provides predictions based on input health parameters, which can be integrated into a user-friendly application for real-time risk assessment.

RESULT

Model Accuracy: The model achieved an accuracy of 98.75 % on the test set.

```
# Accuracy Score
score = round(accuracy_score(y_test, y_pred),4)*100
print("Accuracy on test set: {}%".format(score))
```

```
Accuracy on test set: 98.75%
```

```
Evaluating the accuracy of the Random Forest Classifier on the test set.
```

```
# Classification Report
print(classification_report(y_test, y_pred))
```

```
              precision    recall  f1-score   support

           0       0.98      1.00      0.99       272
           1       1.00      0.96      0.98       128

    accuracy                           0.99       400
   macro avg       0.99      0.98      0.99       400
weighted avg       0.99      0.99      0.99       400
```

Model Accuracy: The model achieved an accuracy of 99.94 % on the training set.

```
# Accuracy Score
score = round(accuracy_score(y_train, y_train_pred),4)*100
print("Accuracy on trainning set: {}%".format(score))
```

```
Accuracy on trainning set: 99.94%
```

```
Evaluating the accuracy of the Random Forest Classifier on the training set.
```

```
# Classification Report
print(classification_report(y_train, y_train_pred))
```

```
              precision    recall  f1-score   support

           0       1.00      1.00      1.00      1044
           1       1.00      1.00      1.00       556

    accuracy                           1.00      1600
   macro avg       1.00      1.00      1.00      1600
weighted avg       1.00      1.00      1.00      1600
```

Prediction 1 :

```
# Prediction 1
# Input sequence: Pregnancies, Glucose, BloodPressure, SkinThickness, Insulin, BMI, DPF, Age
prediction = predict_diabetes(5, 76, 67, 10, 86, 28, 0.5, 29)[0]
if prediction:
  print('Oops! You have diabetes.')
else:
  print("Great! You don't have diabetes.")
```

```
Great! You don't have diabetes.
```

Prediction 2 :

```
# Prediction 2
# Input sequence: Pregnancies, Glucose, BloodPressure, SkinThickness, Insulin, BMI, DPF, Age
prediction = predict_diabetes(1, 117, 88, 20, 155, 28, 0.4, 38)[0]
if prediction:
  print('Oops! You have diabetes.')
else:
  print("Great! You don't have diabetes.")
```

```
Oops! You have diabetes.
```

Prediction 3:

```
# Prediction 3
# Input sequence: Pregnancies, Glucose, BloodPressure, SkinThickness, Insulin, BMI, DPF, Age
prediction = predict_diabetes(1, 110, 90, 7, 70, 23, 0.5, 76)[0]
if prediction:
  print('Oops! You have diabetes.')
else:
  print("Great! You don't have diabetes.")
```

```
Great! You don't have diabetes.
```

CONCLUSION

- **Summary:**
  - The machine learning model successfully predicts diabetes with high accuracy.
  - The Random Forest Classifier proved effective for this task.
- **Impact:**
  - Enables early detection of diabetes, potentially improving patient management and outcomes.

COURSE CERTIFICATE 1

FUTURE SCOPE

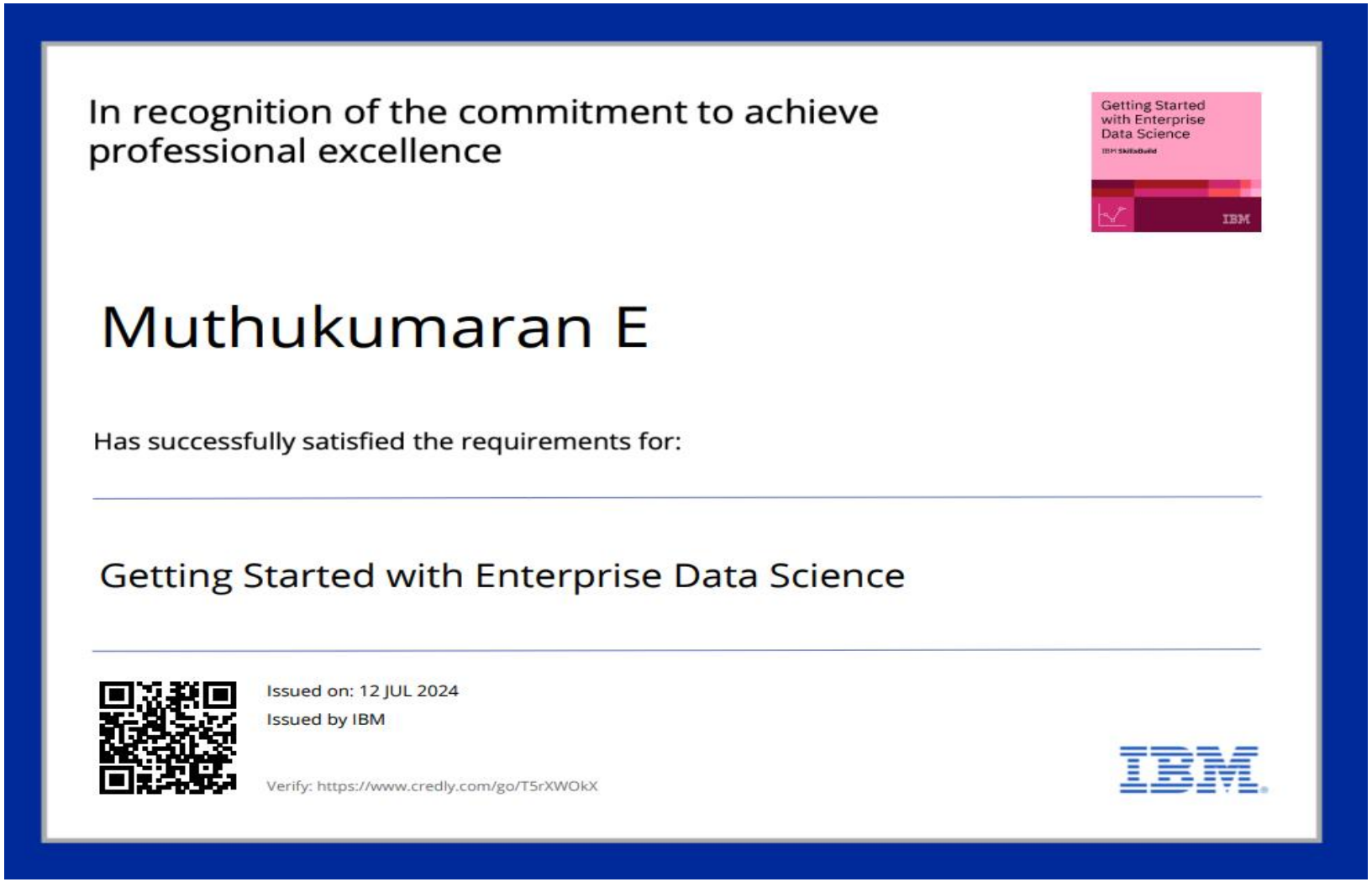
- Improvements:
  - Incorporate additional features or datasets for enhanced model performance.
  - Explore other machine learning algorithms for comparison.

- Applications:
  - Integration into healthcare systems for real-time diabetes risk assessment.
  - Development of a user-friendly application for broader accessibility.

REFERENCES

- **Dataset Source**: Kaggle Diabetes Dataset
- **Libraries Used**: scikit-learn, pandas, Matplotlib, Seaborn
- **Research Papers:** "Random Forest Classifier for Diabetes Risk Prediction"
  Authors: A. Brown, E. Johnson
  Published In: Journal of Machine Learning Research, 2022
- "Machine Learning Techniques for Diabetes Prediction: An Overview"
  Authors: S. Kumar, P. Patel
  Published In: Healthcare Analytics Review, 2021

COURSE CERTIFICATE 2



In recognition of the commitment to achieve professional excellence

Getting Started with Enterprise Data Science

IBM SkillsBuild

IBM

Muthukumaran E

Has successfully satisfied the requirements for:

Getting Started with Enterprise Data Science

Issued on: 12 JUL 2024

Issued by IBM

Verify: https://www.credly.com/go/T5rXWOkX

IBM

THANK YOU

edunet foundation