Assignment 2

Total points: 100

Deadline: 04/10/2025

Building Machine Learning Models for Diabetes Health Indicators

In this assignment, you will perform exploratory data analysis (EDA) on the given dataset and build machine learning models for classifying diabetes health indicators using Python programming language. You will explore several machine learning techniques and evaluate their effectiveness on the dataset. You can use existing packages/libraries to implement the assignment, but make sure to thoroughly comment on the code clearly explaining what you did. Follow the instructions in JupyterNotebook installation guide on pilot for setting up the environment. Analysis and Interpretation are crucial aspects of the assignment, so try to answer the question in more detail to receive full credit. Create a PDF format report answering the questions in detail and attach any graphs/charts/visualizations you may have. Don't directly copy paste cell outputs, create tables for confusion matrices, classification reports etc. Don't just write code, comment your code wherever necessary, to help navigate.

The given dataset contains various health indicators which can be used to classify the presence of diabetes. The Behavioral Risk Factor Surveillance System (BRFSS) is a health-related telephone survey that is collected annually by the Centers for Disease Control and Prevention (CDC). The dataset consists of 253,680 survey responses to the CDC's BRFSS2015. The target variable Diabetes_binary has 2 classes. 0 is for no diabetes, and 1 is for prediabetes or diabetes. This dataset has 21 feature variables.

1. Exploratory Data Analysis [20 points]

a. Load the dataset, explore its dimensions and handle any missing values [2]
b. Calculate the balance of the target variable (Diabetes_binary) and comment on it [4]
c. Visualize the class distribution using appropriate plots (e.g., bar plots) [4]
d. Explore descriptive statistics and create a table [4]
e. Calculate the correlation matrix and generate heatmaps to visualize correlations. Comment on any multicollinearity, if exists. [6]

For implementing machine learning models, it is necessary to separate features from the predictor variable and split data into training and test sets (sometimes validation set as well). Here, for the assignment, split the data into train and test sets. Load the required model, train it and predict the output variable. Evaluate model performance using metrics such as F1-score, accuracy, precision, recall etc. This is the backbone of a machine learning model implementation. Based on the distribution of data and other constraints, several other components can be added to help deal with them. For example, we employ stratification and cross-validation to handle class imbalance and feature scaling to normalize the data etc. Likewise, regularization helps in handling overfitting of the models.

2. Logistic Regression [25 points]

a. Implement logistic regression to predict the presence of Diabetes [target variable (Diabetes_binary)] using all the features in the dataset without regularization and evaluate model performance [5]
b. Implement logistic regression with L1 (Lasso) regularization and evaluate model performance [5]
c. Implement logistic regression with L2 (Ridge) regularization and evaluate model performance [5]

d. Discuss the impact of regularization on model performance and what difference did you notice from 2a, 2b and 2c? [10]

3. Support Vector Machine (SVM) [25 points]

a. Implement SVM with a linear kernel. [5]
b. Implement SVM with a polynomial kernel. [5]
c. Implement SVM with an RBF (Radial Basis Function) kernel. [5]
d. Compare the performance of each kernel using metrics such as accuracy, precision, recall, F1-score and confusion matrix. Include any visualizations if applicable. Summarize which SVM performed best and why. Comment on class distribution affecting model performance [10]

To implement neural networks, feel free to use pytorch or tensorflow libraries.

- Use 'Adam' as optimizer while training the models
- Experiment with different values of learning rate, batch size, number of neurons per layer, activation functions, and dropout rates

4. Neural Networks [25 points]

a. Design and test two different neural network architectures, one being a simple feedforward neural network (e.g., 1-2 hidden layers) and the other being a deeper neural network (e.g., 3+ hidden layers, dropout for regularization) [15]
b. Compare the performance of these architectures and provide a summary of the architectures, training process, and evaluation metrics (accuracy, loss, precision, recall, F1-score, confusion matrix etc.) [10]

5. Test with Different Cross-Validation Folds [5 points]

a. Implement k-fold cross-validation on Logistic regression and SVM models with different values of k (e.g., 5, 10) and provide your observations [2]
b. Test cross-validation with and without stratification and provide your observations. Analyze the impact of the number of folds and stratification on model performance [3]

References

- Sample Logistic Regression implementation https://www.geeksforgeeks.org/ml-logistic-regression-using-python/
- Sample SVM implementation https://www.geeksforgeeks.org/support-vector-machine-algorithm/
- Sample Neural network implementation https://www.geeksforgeeks.org/feedforward-neural-network/

Submission Instructions

You can use NumPy, pandas, Matplotlib, scikit-learn, pytorch, tensorflow etc required libraries for your assignment, make sure to understand it conceptually and answer the questions with proper analysis

- Please upload a zipped file named 'Assignment-2_YourName' to Dropbox. The zip file should contain the following items: a dataset, code file(s) (preferably .ipynb format) (please use relative paths when reading/importing the dataset), a PDF-format report, and a README.txt.

- Please ensure that your code is error-free; there should be no errors or warnings when running it on my machine.
- If you are using a programming language other than Python, please provide a README.txt file explaining how to run your code to obtain the result. Mention any libraries or dependencies that need to be installed before running the code, if necessary.

Academic Integrity

Discussion of course contents with other students is an important part of the academic process and is encouraged. However, it is expected that course programming assignments, homework assignments, and other course assignments will be completed on an individual basis (unless specified otherwise). Students may discuss general concepts with one another, but may not, under any circumstances, work together on the actual implementation of any course assignment. If you work with other students on “general concepts” be certain to acknowledge the collaboration and its extent in the assignment. Unacknowledged collaboration will be considered dishonest. “Code sharing” (including code from previous quarters) is strictly disallowed. “Copying” or significant collaboration on any graded assignments will be considered a violation of the university guidelines for academic honesty.
If the same work is turned in by two or more students, all parties involved will be held equally accountable for violation of academic integrity. You are responsible for ensuring that other students do not have access to your work: do not give another student access to your account, do not leave printouts in the recycling bin, pick up your printouts promptly, do not leave your workstation unattended, etc. If you suspect that your work has been compromised notify me immediately. If you have any questions about collaboration or any other issues related to academic integrity, please contact me immediately for clarification. In addition to the policy stated in this syllabus, students are expected to comply with the Wright State University Code of Student Conduct (http://www.wright.edu/students/judicial/conduct.html) and the portions Pertaining to Academic Integrity http://www.wright.edu/students/judicial/integrity.html) at all times. Note: In cases where there is suspicion of academic dishonesty, the professor and teaching assistant reserve the right to address the matter by calling in the student for an in-person question and answer session.