CSI 4106 Introduction to Artificial Intelligence

Assignment 2: Machine Learning

Marcel Turcotte

Version: Oct 4, 2024 09:05

🎯 Learning Objectives

- **Conduct** a comprehensive exploratory data analysis
- **Apply** data preprocessing techniques effectively
- **Develop** and **evaluate** machine learning models
- **Optimize** hyperparameters and analyze model performance

📤 Submission

- **Deadline:**
  - Submit your notebook by October 20, 11 PM.
- **Individual or Group Assignment:**
  - This assignment may be undertaken either individually (group of 1 student) or collaboratively in pairs (group of 2 students)
  - A group must submit a single joint submission.
  - Prior to submitting, it is necessary to register your group on Brightspace.
  - To accommodate changes in group membership for each assignment, a new registration of groups is required for every assignment.
    - Thus, you must register anew for each assignment.
- **Submission Platform:**
  - Upload your submission to Brightspace under the Assignment section (Assignment 2).
- **Submission Format:**
  - Submit a copy of your notebook on Brightspace.

Important Notice: If the corrector cannot run your code, your submission will receive a mark of zero. It is your responsibility to ensure that your submission works from a different computer than your own and that all cells in your notebook are executable.

Requirements

1. Exploratory Analysis

Data Exploration

In this assignment, we will utilize the Diabetes Prediction Dataset, accessible via Diabetes Prediction Dataset. To mitigate the complexity associated with Kaggle's login requirement, the dataset has been made available on a public GitHub repository:

- github.com/turcotte/csi4106-f24/tree/main/assignments-data/a2

You can access and read the dataset directly from this GitHub repository in your Jupyter notebook.

(1) **Load the dataset and provide a summary of its structure**:

- Describe the features (columns), their data types, and the target variable.

(2) **Feature Distribution Analysis**:

- Examine the distribution of each feature using appropriate visualizations such as histograms and boxplots. Discuss insights gained, including the presence of outliers.

(3) **Target Variable Distribution**:

- Analyze the distribution of the target variable to identify class imbalances. Use bar plots to visualize the class frequencies.

(4) **Data Splitting**:

- Split the dataset into training (80%) and test (20%) sets using the holdout method.
- Ensure that this split occurs before any preprocessing to avoid data leakage.

Data Pre-Processing

(5) **Categorical Variable Encoding**:

- Encode any categorical variables. Justify the chosen method.

(6) **Normalization/Standardization of Numerical Features**:

- Normalize or standardize numerical features if necessary. Describe the technique used (e.g., Min-Max scaling, StandardScaler) and explain why it is suitable for this dataset.
- Ensure that this technique is applied only to the training data, with the same transformation subsequently applied to the test data without fitting on it.

Model Development & Evaluation

(7) **Model Development**:

- Implement the machine learning models covered in class: Decision Trees, K-Nearest Neighbors (KNN), and Logistic Regression. Use the default parameters of scikit-learn as a baseline for training each model.

(8) **Model Evaluation**:

- Use cross-validation to evaluate each model, justifying your choice of the number of folds.
- Assess the models using metrics such as precision, recall, and F1-score.

Hyperparameter Optimization

(9) **Exploration and Performance Evaluation:**

- Investigate the impact of varying hyperparameter values on the performance of each model.
- Focus on the following relevant hyperparameters for each model:
  - DecisionTreeClassifier: `criterion` and `max_depth`.
  - LogisticRegression: `penalty`, `max_iter`, and `tol`.
  - KNeighborsClassifier: `n_neighbors` and `weights`.

- Employ a grid search strategy or utilize scikit-learn's built-in methods to thoroughly evaluate all combinations of hyperparameter values. Cross-validation should be used to assess each combination.
- Quantify the performance of each hyperparameter configuration using precision, recall, and F1-score as metrics.
- Display the results in a tabular or graphical format (e.g., line charts, bar charts) to effectively demonstrate the influence of hyperparameter variations on model performance.
- Specify the default values for each hyperparameter tested.
- Analyze the findings and offer insights into which hyperparameter configurations achieved optimal performance for each model.

Analysis of Results

(10) **Model Comparison**:

- Compare the results obtained from each model.
- Discuss observed differences in model performance, providing potential explanations. Consider aspects such as model complexity, data imbalance, overfitting, and the impact of parameter tuning on overall results.
- Provide recommendations on which model(s) to choose for this task and justify your choices based on the analysis results.
- Train the recommended model(s) using the optimal parameter values identified from the parameter optimization step. Subsequently, apply the trained model to the test data. Document your observations comprehensively. Specifically, evaluate whether the results derived from cross-validation are consistent with those obtained from the test set.

2. Documentation of Exploratory Analysis

The report should comprehensively document the entire process followed during this assignment. The Jupyter Notebook must include the following:

- Your name(s), student number(s), and a report title.
- Explain how the tasks have been split between the members. How did you make sure that both students achieve the learning outcomes?
- A section for each step of the exploratory analysis, containing the relevant Python code and explanations or results.

  - For sections requiring Python code, include the code in a cell.
  - For sections requiring explanations or results, include these in a separate cell or in combination with code cells.

- Ensure logical separation of code into different cells. For example, the definition of a function should be in one cell and its execution in another. Avoid placing too much code in a single cell to maintain clarity and readability.
- The notebook you submit must include the results of the execution, complete with graphics, ensuring that the teaching assistant can grade the notebook without needing to execute the code.

Evaluation

- **Overall Effort in the Report (5%)**
- **Data Exploration (10%)**
- **Data Pre-Processing (20%)**
- **Model development & evaluation (20%)**
- **Parameter Optimization (30%)**
- **Analysis of Results (10%)**
- **Resources and References (5%)**

Resources

As previously mentioned, ensure that you cite any parts of your code that are derived from websites, textbooks, or other external resources.

Currently, many programmers leverage artificial intelligence to enhance their productivity, a trend that is likely to continue growing. To better prepare you for the job market, it is plausible to utilize these technologies. However, it is imperative that you fully understand the concepts upon which you are evaluated, as these tools will not be available during in-person evaluations.

If you do use AI assistance, thoroughly document your interactions. Include the tools and their versions in your report, along with a transcript of all interactions. Most AI assistants keep a record of your conversations. The recommended practice is to create a new conversation specifically for the assignment and consistently reuse this conversation throughout your work on the assignment. Ensure that this conversation is solely dedicated to the assignment. Submit this conversation transcript in the reference section of your Jupyter Notebook.

Questions

- You may ask your questions in the Assignment topic of the discussion forum on Brightspace.
- Alternatively, you can email one of the four teaching assistants. However, using the forum is strongly preferred, as it allows your fellow students to benefit from the questions and the corresponding answers provided by the teaching assistants.