Collaborative Machine Learning Project with Jupyter Notebook

F. Saïd

1. Objectives

This project is designed to give you practical experience with machine learning techniques, from initial data preprocessing to the final steps of model evaluation and interpretation. You will select a dataset of interest, conduct exploratory data analysis, prepare the data for modeling, build and tune machine learning models using two different types of learning approaches, and evaluate their performance. This project should be completed in pairs, promoting collaboration and collective problem-solving.

2. Project Description

In pairs, you are tasked with completing a machine learning project using Jupyter Notebook as your development environment. The project should encompass the following components:

2.1 Team Formation:

- Pair up with a classmate. Each pair will work together throughout the project.
- Collaborative efforts and peer programming are encouraged to leverage diverse perspectives and expertise.

2.2 Data Selection:

- Choose a dataset that both partners agree upon and find intriguing. It could be from sectors like finance, healthcare, education, social media, or technology.
- The chosen dataset should have sufficient complexity to construct an insightful machine learning model.
- You are encouraged to explore various sources to find a dataset that fits the project's objectives. Here are some known sources where you can find a wide range of datasets:
  - UCI Machine Learning Repository: A collection of databases, domain theories, and data generators widely used by the machine learning community.
  - Kaggle Datasets: A platform for predictive modeling and analytics competitions that hosts datasets in various domains.
  - Google Dataset Search: A tool that enables the discovery of datasets stored across the web.

- AWS Public Datasets: Amazon Web Services provides a range of public datasets that anyone can access.
- Government Databases: For US-related projects, this site provides a wealth of government-related data.
- Additionally, review the datasets that I have proposed in our learning platform, which have been selected to offer a variety of challenges suitable for your projects.

When selecting a dataset, ensure that you have the right to use it and that it does not contain sensitive or private information unless it has been anonymized and is compliant with privacy laws.

2.3 Data Preprocessing:

- Address any issues of missing values, duplicates, and outliers.
- Engage in necessary feature engineering and data scaling.

2.4 Exploratory Data Analysis (EDA):

- Develop visualizations to comprehend data distributions and variable interrelations.
- Compile a summary of the insights obtained from the EDA.

2.5 Model Selection:

- Select at least two distinct types of machine learning algorithms (e.g., one supervised and one unsupervised, or one regression and one classification).
- Provide a rationale for your algorithm choices.

2.6 Model Training and Tuning:

- Divide the data appropriately into training and testing sets.
- Implement cross-validation to refine model parameters.

2.7 Model Evaluation:

- Utilize suitable metrics to assess and contrast your models' performance, reflecting on the different learning approaches.
- Discuss outcomes and possibilities for model enhancement.

3. Documentation

- Create comprehensive documentation within the Jupyter Notebook, detailing code, methodologies, and analyses.
- Utilize comments and markdown cells for clarity on the undertaken steps and gathered insights.

4. Presentation

- Assemble a conclusive report that encapsulates the project's methods, analysis, findings, and any conclusions or advice.
- The report should be in the form of a Jupyter Notebook (.ipynb) with all code cells executed and outputs displayed.

5. Deliverables

- A shared Jupyter Notebook containing the complete project with exhaustive comments and markdown elucidations. Ensure that your notebook is well-organized, with code, methodologies, and analyses clearly documented and explained.
- A succinct presentation (10-15 slides) that outlines the primary facets of your project.

6. Submission Instructions

- Please submit your final Jupyter Notebook and presentation slides through the Project Report Repository area on Moodle platform.
- Ensure that both partners' names are included in the submission and that the files are appropriately named to reflect the content and authors

  (e.g.,ML_Project_Lastname1_Lastname2.ipynb).
- All submissions must be made before the deadline: February 16th, 2024, at 23:59. Late submissions may not be accepted or could be subject to a penalty, as per the course policies.

7. Evaluation Criteria

- Thoroughness in data preprocessing and EDA.
- Judicious selection and deployment of two varied types of machine learning algorithms.
- The caliber of code and accompanying documentation : quality and standard of the programming code written for the project as well as the quality of the documentation that explains and supports the code.
  - Caliber of Code: This pertains to how well the code is written. It includes considerations like:
    * Readability: Is the code easy to read and understand?
    * Efficiency: Does the code run efficiently without unnecessary computational overhead?
    * Robustness: Is the code stable, and does it handle errors or unexpected inputs gracefully?

    - Maintainability: Can the code be easily maintained and updated by someone other than the original author?
    - Best Practices: Are programming best practices followed, such as using meaningful variable names, commenting code, and adhering to style conventions?
  - Accompanying Documentation: This refers to the written explanations and descriptions that accompany the code. Good documentation is crucial for understanding what the code does, how it does it, and why certain decisions were made. This includes:
    - Comments in the Code: Inline explanations that clarify why specific lines or blocks of code are there.
    - README Files: A document that provides an overview of the codebase, how to set it up, and how to use it.
    - Developer Guides: More detailed explanations that may include the logic behind the code, the structure of the software, and how different parts interact.
    - API Documentation: If the code has an API, clear documentation of the API endpoints, parameters, expected inputs, and outputs.

- The effectiveness of the model evaluation and comparative analysis.
- The comprehensiveness and clarity of the final report and presentation.