Feasibility Study

Comparative Evaluation of Concept Drift Detection Algorithms: A Benchmarking Study

10/07/2024

Mentor : Dr. Uthayasanker Thayasivam
TA: Mr. Dileesha Kannangara
Team members: 210386A - Mendis P.H.M.N
210401T - Nadawa R.M.N
210204F - Nanayakkara A.H.M

TABLE OF CONTENTS

1. Introduction

1.1 Overview of the Project

The project involves developing a comprehensive benchmarking system for evaluating concept drift detection algorithms specifically tailored for regression datasets. The system aims to identify and compare key algorithms to determine their effectiveness in maintaining the accuracy and reliability of regression models in dynamic environments. The benchmarking process will use both real and synthetic datasets to provide a robust experimental design. This evaluation will benefit organizations by enhancing the robustness of their predictive models, thereby ensuring better decision-making and increased trust in AI systems.

1.2 Objectives of the Project

The objectives of this project are to:

- Design and implement a benchmarking system for concept drift detection algorithms in regression contexts.
- Provide a comparative analysis of various concept drift detection algorithms for regression datasets.
- Automate the evaluation process to identify the most effective algorithms for specific regression scenarios.
- Enhance the robustness and reliability of regression models in dynamic data environments.

1.3 The Need for the Project

In dynamic environments, the performance of regression models can degrade over time due to changes in the relationship between input data and model targets, known as concept drift. Concept drift reflects the evolution of the underlying problem statement or process over time. Effective detection of concept drift is crucial for maintaining the accuracy and reliability of these models. This project addresses the need for a systematic evaluation of concept drift detection algorithms tailored to regression datasets, ensuring that organizations can choose the best methods for maintaining model performance and trustworthiness.

1.4 Overview of Existing Systems and Technologies

Existing systems for concept drift detection include various algorithms and tools designed to identify changes in the relationship between input data and model targets. Popular algorithms include DDM (Drift Detection Method), EDDM (Early Drift Detection Method), Page Hinkley, and ADWIN (Adaptive Windowing). Tools like the Frouros library and DeepChecks package provide functionalities for implementing and evaluating these algorithms. However, a comprehensive benchmarking system that systematically evaluates these methods under different conditions specific to regression datasets is lacking. This project aims to fill this gap by providing a rigorous evaluation framework for concept drift detection algorithms tailored to regression models, ensuring organizations can make informed decisions to maintain model accuracy and reliability in dynamic environments.

1.5 Scope of the Project

The project will involve the following user roles and functionalities:

- **Data Scientists**: Will use the benchmarking system to evaluate concept drift detection algorithms and choose the most effective ones for detecting changes in the relationship between input data and model targets in regression models.
- **Machine Learning Engineers**: Will integrate the selected algorithms into their regression pipelines to monitor and adapt to concept drift, ensuring ongoing model performance and accuracy.
- **Researchers/Reviewers:** Responsible for analytical research and documentation, they will contribute to evaluating algorithm performance and documenting findings throughout the project.
- **Organizations**: Will rely on the insights provided by the system to maintain the robustness and reliability of their regression models in dynamic environments, thereby enhancing decision-making processes and fostering trust in AI systems.

1.6 Deliverables

The main outputs of the project will be:

- A comprehensive benchmarking system for evaluating data drift detection algorithms tailored for regression datasets.
- Functional Python scripts for replicating the experiments.
- A detailed report explaining the findings, including comparative studies, discussions, and conclusions.
- A dashboard showcasing the results and insights visually.

2. Feasibility Study

2.1 Financial Feasibility

2.1.1 Projected Costs:

Software and Tools:

- Computing Resources: Cloud services (e.g., AWS, Google Cloud, Azure) for data processing and storage. Estimated cost: None, Since the project is only focused on dashboarding.
- No personnel and miscellaneous costs.

2.1.2 Projected Benefits:

Enhanced Understanding of Drift Detection Algorithms:

- Potential to publish research papers and contribute to academic knowledge, potentially leading to grants and funding opportunities.
- Improved algorithms can be commercialized or implemented in industry, leading to potential revenue streams.

Skill Development:

- Team members will gain valuable experience and skills in data science, machine learning, and software engineering, enhancing their professional value.

Reputation and Networking:

- Completion and publication can enhance the reputation of the individuals and institutions involved, leading to future collaborations and projects.

2.2 Technical Feasibility

This section discusses the technical aspects of system development, outlining the planned tools and technologies, and assessing the project's technical feasibility.

2.2.1 Tools and Technologies

Programming Languages:

- Python:- Python will be the primary language for data analysis, algorithm implementation, and visualization, utilizing libraries such as pandas, NumPy, SciPy, Scikit-learn, and Matplotlib.

Libraries and Frameworks:

- Frouros:- Frouros, a Python library for drift detection, will be used as a core component of the project.
- Scikit-Multiflow:- Scikit-Multiflow, another Python library, will be employed for learning from data streams and drift detection.
- River:- River, a Python library for online machine learning, will be used for its capabilities in generating synthetic datasets, handling data streams, and performing real-time drift detection.

Data Sets:

- Friedman Data Set:- The Friedman data set will be used as synthetic data for initial testing and validation.
- NYC Taxi Data:- The NYC taxi data will serve as real-world data for testing and benchmarking the drift detection algorithms.

Development and Collaboration Tools:

- GitHub:- GitHub will be used for version control, project management, and collaboration. It will ensure that code is systematically managed, reviewed, and shared among team members.
- Google Colab:- Google Colab will be utilized for interactive coding, sharing notebooks, and running experiments in a cloud-based environment without the need for local computational resources.

Computing Resources:

- Cloud Services: AWS, Google Cloud, or Azure for scalable computing and storage solutions.

2.2.2 Data Availability and Quality:

The selected data sets (Friedman and NYC Taxi) are publicly available and well-documented, ensuring high data quality and ease of access.

2.2.3 Scalability and Flexibility:

- Cloud Computing:- Utilizing cloud services ensures that the project can scale as needed, with the flexibility to adjust computing resources based on project demands.
- Open-Source Libraries:- The use of open-source libraries and frameworks allows for customization and adaptation to specific project requirements.

2.3 Resource and Time Feasibility

The well-planned phased approach ensures efficient use of resources and time.

- literature review - 1st week
- Experimental Framework development - 2nd week
- Carrying out experiments - 3rd to 5th weeks
- Observations and Analyzing - 6th week
- Documentation - 7th weeks
- dashboard and tool development - 8th to 11th weeks

Access to synthetic datasets and real-world datasets from sources like River and Kaggle provides a streamlined process for data acquisition.

The structured timeline and defined objectives indicate that with a dedicated team, the project can be completed within a reasonable timeframe, ensuring resource and time efficiency.

2.4 Risk Feasibility

Potential risks include

- ineffectiveness of certain algorithms in specific scenarios
- data quality issues
- integration challenges with existing systems.

To mitigate these risks, the project will employ a robust testing framework, regular performance assessments, and comprehensive documentation.

This proactive risk management strategy ensures that any issues are identified and resolved promptly, minimizing the impact on the project's overall success.

2.5 Social/Legal Feasibility

By using publicly available datasets and synthetic data, the project ensures compliance with data privacy regulations and ethical standards.

The development of tools that improve the reliability and adaptability of machine learning models addresses a critical need in various industries, from finance to healthcare, thus providing significant social value.

The project's commitment to ethical considerations and legal compliance ensures that it meets societal expectations while contributing positively to the broader community.

3. Considerations

Performance:

- The system must be capable of accurately and promptly detecting shifts in data distributions to ensure timely adjustments and maintain model accuracy.

- This requires efficient processing of data streams and robust evaluation metrics to assess the performance of different algorithms under various conditions.
- Benchmarking the performance of selected algorithms will provide insights into their strengths and limitations, guiding the selection of the most effective methods for real-world applications.

Security:

- Ensuring data privacy and protection against unauthorized access is essential. The system should incorporate strong encryption protocols and access controls to safeguard data.
- Additionally, the development of the library and application must adhere to best practices in software security, preventing vulnerabilities that could be exploited by malicious actors.
- Regular security audits and compliance with relevant data protection regulations will further enhance the system's trustworthiness.

Usability:

- The system should feature an intuitive interface that allows users to easily configure experiments, monitor data streams, and interpret results.
- Comprehensive documentation and user guides will facilitate understanding and efficient use of the tool.
- User feedback during development will be crucial to refining the interface and ensuring that it meets the needs of its target audience.

Ease of Use:

- The tool should be designed to integrate seamlessly with existing machine learning workflows, minimizing the need for extensive modifications or additional training.
- Automation of repetitive tasks, such as data preprocessing and algorithm configuration, will enhance user efficiency.
- Providing pre-configured templates and examples can help users quickly get started with the tool, reducing the learning curve and enabling them to focus on analyzing and addressing data drift.

4. References

1. Thomas Lambart. "Concept Drift Detection: An Overview." Medium. Available at: https://medium.com/@thomaslambart/concept-drift-detection-an-overview-d087feea9676 . Accessed on 26 June 2024.
2. DataCamp. "Understanding Data Drift & Model Drift." DataCamp Tutorial. Available at: https://www.datacamp.com/tutorial/understanding-data-drift-model-drift. Accessed on 27 June 2024.
3. Jie Lu, Anjin Liu, Fan Dong, Feng Gu, Joao Gama, and Guangquan Zhang. "Learning under Concept Drift: A Review." IEEE Transactions on Knowledge and Data Engineering. Accessed on 29 June 2024.

4. Lucas Baier, Marcel Hofmann, Niklas Kühl, Marisa Mohr, and Gerhard Satzger. "Handling Concept Drifts in Regression Problems – the Error Intersection Approach." Accessed on 1 July 2024.
5. Marília Lima, Manoel Neto, Telmo Silva Filho, and Roberta A. de A. Fagundes. "Learning Under Concept Drift for Regression—A Systematic Literature Review." IEEE Transactions on Neural Networks and Learning Systems. Accessed on 6 July 2024.
6. Frouros library. GitHub repository. Available at: https://github.com/IFCA-Advanced-Computing/frouros. Accessed on 8 July 2024.
7. DeepChecks package. GitHub repository. Available at: https://github.com/deepchecks/deepchecks. Accessed on 8 July 2024.
8. Productivity Commission of Australia. "Regulation Benchmarking Feasibility Study." Available at: https://www.pc.gov.au/inquiries/completed/regulation-benchmarking-feasibility/report. Accessed on 9 July 2024.
9. Evidently AI. "Evidently: Tools for Machine Learning Model Monitoring." Available at: https://www.evidentlyai.com/. Accessed on 9 July 2024.
10. Asana. "Feasibility Study Resources." Available at: https://asana.com/resources/feasibility-study. Accessed on 10 July 2024