Team 2 Time Series Repository
Software Requirements Specification

Madeline Porcaro, Luying Cai, Catherine Raj, Sam Schwartz
Team 2
https://github.com/orcap23/CS422Project.github.io
J. Flores - 10-12-2021

Table of Contents

1. SRS Revision History

Date	**Author**	**Description**
10-12-2022	Maddie P.	Created the SRS documents
10-15-2022	Maddie P.	Began ConOps
10-24-2022	Maddie P.	Further edited ConOps
10-30-2022	Maddie P.	Added section 2
10-30-2022	Luying C.	Added sections 3.3, 3., 3.5
10-30-2022	Maddie P.	Added section 3.1, 3.2
10-30-2022	Maddie P.	Added SRS revision history and finalized SRS

2. The Concept of Operations (ConOps)

2.1. Current System or Situation

With the rise of machine learning, data science and a society that is more reliant on data now more than ever, the need to have sources for data to access is becoming more a necessity. To help with the data accessibility, repositories have come up as a means of having open-access datasets to download for the use of data scientists, machine learning engineers or anyone who would like to try their hand learning on their own. Individuals will also have the ability to upload the datasets to have data error calculations be made for their datasets.

2.2. Justification for a New System

The current system of paywalls around data, regardless of the type can be a major deterrent to many when it comes to learning about machine learning or data science or engineers trying to find a set to train with without building one themselves. Data drives society as it is seen today and with more and more data being collected through various means. Data gives people a direction to think too, whether it be who to follow online, who to vote for, what to buy and from which brand based upon numerous variables. Data is a massive driver in decision making.

As of now, there are some time series repositories but what sets Team's 2 Time Series Repository meets the need for a digital warehouse for training sets that can be downloaded and used for training models so machine learning engineers and data scientists don't have to make their own sets. There are numerous repositories that have multiple categories out there but having a place designed only for time series datasets and not attempting to advertise anything or take away from user experience with added "noise" from other features.

2.3. Operational Features of the Proposed System

The new system, Team 2's Time Series Repository, is an application for machine learning engineers and data scientists to find time series datasets and only time series datasets to download and use to train their models. As models are tested, the datasets can change from the initial datasets that were originally downloaded. Machine learning engineers and data scientists can then upload their modified set and compare it to a base set that the training set is taken from. The file is uploaded and read and values that include but are not limited to, mean average error and correlation coefficient.

2.4. User Classes

For the user classes there are two major classes that overlap often: machine learning engineers and data scientists. Since the application is geared towards that particular group to begin with, it is assumed that users will know what a dataset is and what values are being projected upon uploading a file.

2.5. Modes of Operation

There is only one mode of operation, and it is the upload and download functionality for the users. This mode will let users be able to access training sets along with uploading modified sets to see the data modeling as mentioned in section 1.3. This is the only mode of operation.

2.6. Operational Scenarios (aka "Use Cases")

Use Case: Downloading a Dataset

Brief description: This use case describes how a user would download a dataset from the web application

Actors: A user

Preconditions:

1. The user has made a connection to the server.
2. The user has space on their hard drive of a device to store the dataset.
3. The user has properly set up the application on their local machine.
4. The user has internet access.

Steps to Complete the Task:

1. The user connects the application to the host address.
2. The user downloads a training set

Postconditions:

The user has successfully downloaded a training set from the repository. This will allow the user to use the training set to their discretion.

Use Case: Uploading a Dataset

Brief description: This use case describes how a user would upload a dataset to the repository and see the modeling data after an upload.

Actors: A user.

Preconditions:

1. The user has made a connection to the server.
2. The user has space on their hard drive of a device to store the dataset.
3. The user has properly set up the application on their local machine.
4. The user has internet access.
5. The user has a dataset to upload

Steps to Complete the Task:

1. The user connects the application to the host address.
2. The user navigates to the upload page
3. The user uploads a dataset to the application

Postconditions:

The user should have successfully uploaded a dataset and see it compared with the base set that is already in the application.

3. Specific Requirements

3.1. External Interfaces (Inputs and Outputs)

3.1.1. Downloading a dataset

1. Purpose: Giving a user access to training sets to be used in time series forecasting or modeling.
2. Source of input/output: Input will be a click from a user to start the download and the output is the downloaded dataset
3. Valid ranges of input/output:
4. Units of measure: N/A
5. Data formats: Downloaded file is a CSV file.

3.1.2. Uploading a dataset

1. Purpose: Giving user the functionality of uploading a dataset for comparison against a base set
2. Source of input/output: Input will be a file from the user and the output is the data modeling information
3. Valid ranges of input/output: Input should be a CSV, txt or xlxs file
4. Units of measure: N/A
5. Data formats: Uploaded file is a CSV, txt or xlxs file.

3.2. Functions

1. Validity checks on the inputs: when a user uploads their dataset it has to fit the required file type for uploading. This will prevent users from trying to upload incorrect file types to the database.
2. Sequence of operations in processing inputs: After a user uploads a dataset, the system will read the file and they can see the data modeling on the upload page.
3. Responses to abnormal situations, including error handling and recovery: if a user tries to upload an incorrect file type the user will be told the file cannot be uploaded and a message pops up to tell the user the correct file types for uploading.

3.3. Usability Requirements

Required:

Must-Have:

1. TS set/hierarchy metadata
2. TS metadata
3. TS data
4. Forecasting task
5. Training, Validation, and Test sets, MLE can download the training set and time series metadata and forecasting task.
6. MLE can upload their solutions to the website for model testing.
7. Model testing: repository can compare forecasting results regarding error measures with test set.

Should-Have:

1. Website can search for the specific time series.
2. Website can have user login system

3.4. Performance Requirements

Static:

1. .csv file of train set.
2. .html file for the front end. One page for MLE reviews the dataset and one page for description of dataset and evaluates the solutions uploaded from MLE.
3. .CSS file to beautify the webpage.

Dynamic:

1. When MLE clicks the download button, it will automatically download the metadata, forecasting task and train set.
2. When MLE clicks the upload button, they can upload their solutions to evaluate and can see the result after evaluations.

3.5. Software System Attributes

Software Attributes:

1. Reliability
2. Security
3. Privacy
4. Maintainability
5. Portability

Most Important Attributes:

1. Project must run in the Django framework with a pip virtual environment.
2. Python code must run in Python3.7 through 3.10.
3. Instruction must be provided for how to compile the code.
4. All project-development-related documents that are intended for human reading.

4. References

IEEE Std 1362-1998 (R2007). (2007). IEEE Guide for Information Technology–System Definition–Concept of Operations (ConOps) Document. https://ieeexplore.ieee.org/document/761853

IEEE Std 830-1998. (2007). IEEE Recommended Practice for Software Requirements Specifications. https://ieeexplore.ieee.org/document/720574

ISO/IEC/IEEE Intl Std 29148:2011. (2011). Systems and software engineering — Life cycle processes — Requirements engineering. https://ieeexplore.ieee.org/document/6146379

ISO/IEC/IEEE Intl Std 29148:2018. (2018). Systems and software engineering — Life cycle processes — Requirements engineering. https://ieeexplore.ieee.org/document/8559686

Faulk, Stuart. (2013). *Understanding Software Requirements*. https://projects.cecs.pdx.edu/attachments/download/904/Faulk_SoftwareRequirements_v4.pdf

Oracle. (2007). White Paper on “Getting Started With Use Case Modeling”. Available at: https://www.oracle.com/technetwork/testcontent/gettingstartedwithusecasemodeling-133857.pdf

van Vliet, Hans. (2008). *Software Engineering: Principles and Practice*, 3nd edition, John Wiley & Sons.

Work Breakdown Structures. In *Wikipedia*, n.d. https://en.wikipedia.org/wiki/Work_breakdown_structure.

N2 Charts. In *Wikipedia*, n.d. https://en.wikipedia.org/wiki/N2_chart.

Functional Flow Block Diagrams. In *Wikipedia*, n.d. https://en.wikipedia.org/wiki/Functional_flow_block_diagram.

Structure Chart. In *Wikipedia*, n.d. https://en.wikipedia.org/wiki/Structure_chart.

Data-flow Diagram. In *Wikipedia*, n.d. https://en.wikipedia.org/wiki/Data-flow_diagram.

Object Diagram. In *Wikipedia*, n.d. https://en.wikipedia.org/wiki/Object_diagram.

System Context Diagram. In *Wikipedia*, n.d. https://en.wikipedia.org/wiki/System_context_diagram.

Storyboard. In *Wikipedia*, n.d. https://en.wikipedia.org/wiki/Storyboard.

Entity Relationship Model. In *Wikipedia*, n.d. https://en.wikipedia.org/wiki/Entity%E2%80%93relationship_model.

5. Acknowledgements

The SRS template was provided by Juan Flores as a submission example from the Fetch Team from 2022.