WaveAccess

ValueX

Description of Value Xi Functionality

Phone: +1 866 311 2467

wave-access.com

Contents

Phone: +1 866 311 2467 **wave-access.com**

Overview of the Value Xi Product

Value Xi is a "low code" platform designed to automate the processing of structured data using machine learning. It is focused on processing and utilizing structured data to automate routine business operations. With its modular engine, Value Xi provides flexibility in customization and allows the platform to be tailored to meet the specific requirements of each process.

How Value Xi Works

Data Sets

1. Users can upload their data sets into the product independently. The data set must meet the following requirements:
    - Be *structured,* such as in a table format like CSV, XLS, XLSX, or a zip archive.
    - Be *labeled*, meaning it must contain historical values for the parameter that the product will evaluate during its operation;
2. The system checks whether the column types are correctly recognized, giving the user the option to select the target parameter that needs to be predicted for their task. Multiple target columns can be selected when training the model. If the chosen field does not meet the requirements for target values, the system will provide an explanation.
3. Immediately after uploading the data set, the user can perform a preliminary analysis and adjust as needed. The preliminary analysis includes:
    - General data overview,
    - Information on input variables,
    - Correlations,
    - Duplicate columns,
    - Missing values,
    - Data gaps,
    - Data sample preview.
4. The user can initiate data processing and model training on the configured data set. It's possible to run multiple training processes simultaneously (if the system is set up for this). If there are available virtual machines, the processes will run in parallel. If not, the processes will be queued. There's also an option to stop a process to free up a virtual machine.
5. Value Xi allows the use of a pre-configured data processing and transformation pipeline for training or prediction, with options to customize it via the UI or by uploading a pre-prepared pipeline description file. Value Xi tracks the pipeline execution and provides logs for review. The platform supports customizing existing

pipeline modules, as well as uploading custom ones tailored to specific process requirements, following the interaction template between pipeline elements.

6. After the model training is completed, the user receives metrics indicating the model's prediction accuracy. Detailed metrics from the training and testing phases are available on the model training results analysis screen.
7. All completed predictions are available for review and analysis.
8. The model analytics provide detailed information for each model, including the following charts: PR curve, ROC curve, and Confusion Matrix (for classification algorithms), as well as Predictions and Deviation Density (for regression algorithms).
9. Users can integrate the trained model into their systems via REST API. All necessary configurations are available in the user interface. Integration can be done in two modes: synchronous and asynchronous (the latter being more suitable for large datasets). Users can configure how the service operates depending on the volume of data.
10. Each model can be deployed on one or more workers depending on the expected load.

OCR Projects

For working with documents in Value Xi, users have the option to create another project type — OCR, which allows the uploading of scanned text document images and automatically converting them into DOCX format for further processing of the information contained within the documents.

- For OCR projects, images must be uploaded in the following formats: BMP, JPG, JPEG, PNG, GIF, PDF, or as a ZIP archive.

Large Language Models (LLM Models)

For tasks such as text generation, determining text similarity, text classification, extracting answers from text, summarizing documents while retaining key information, and text paraphrasing, large language models can be used. Value Xi allows users to select the most suitable LLM model from supported options (OpenAI GPT, Llama), considering the limitations of each model.

Chatbots (LLM Chats)

LLM models can be configured through a chat interface. Users can create chats by selecting the appropriate model, customize parameters, including the RAG option, manage their chats, and view chat history.

Knowledge Bases

When the RAG option is enabled, users can select a knowledge base for use in LLM chats. Using knowledge bases allows for more precise responses to specific topics

when querying LLM models. In Value Xi, users can create knowledge bases, upload files to them, and manage settings and access permissions.

REST API

1. Users can interact with Value Xi via REST API, offering functionalities for Projects, Datasets, Models, Predictions, OCR recognition settings and results, NER projects, Custom Modules, LLM Models, LLM Chats, and Knowledge Bases.
2. With the REST API, users can upload data into Value Xi from various external sources, such as messengers, email, task tracker JIRA, MS Excel, MySQL, SQLite, PostgreSQL, Oracle, Databricks, MongoDB, ODBC, JDBC, Clickhouse, and other sources supporting REST API queries.

SDK

Using the SDK, users can interact with Value Xi via Python.

Other Features

Value Xi also offers the following capabilities:

- Deployment in both cloud and local customer environments.
- Support text processing.
- Simultaneous multi-user access.
- Role-based access control for managing permissions across system sections, projects, knowledge bases, and models.
- Flexible user-specific and global usage restrictions, such as limiting access to LLM models.
- Comprehensive documentation, including detailed descriptions of the algorithms and modules used.

Technologies Used

Backend: Python 3.8 (Flask, Gunicorn, SQLAlchemy, Psycopg2, Msgpack, Dask, Alembic, Requests).
Frontend: Angular 15, JavaScript, TypeScript, HTML5, CSS3.
OS-level Virtualization Systems: Docker.

ML Libraries:

Fluent-logger, Psycopg2-binary, Kafka-python, Psutil, PyPika, Pympler, Pymorphy2, Gunicorn, Flask, Flask-Cors, Flask-restful / Flask_restx, Jinja2, Gevent, MarkupSafe, Flask-swagger-ui, Python-dateutil, Config-client, Requests, Werkzeug, Pydantic, Deprecation,

PyYAML, MinIO, Retry, Yoyo-migrations, Msgpack, Numpy, Matplotlib, Scipy, Pandas, Pandas-profiling, NLTK, Scikit-learn, PyArrow, LightGBM, Seaborn, OpenCV-python, Numba, Imbalanced-learn, XGBoost, Phik, Spacy (ru, en, da, de), Pint, Scikit-image, Pytesseract, Featuretools, CatBoost, Gensim, Sentence-transformers, Transformers, Lime, Shap, Dill, Prophet, Pystan, Statsmodels, Plotly, MMDet, MMCV-full, MMOCR, Torchvision, Layoutparser, Python-docx, Doc2pdf, Pdfminer.six, Pdf2image, Pillow, PyPDF4, Etna, Vosk, Docx2python, Elasticsearch, Dask-ml, S3fs, Cachey, Bokeh, Rapidfuzz, Tiktoken, Protobuf, Langchain, Pymilvus, Pypdf, Docx2txt, Sentencepiece.

ML Algorithms:

Regression and Classification: Random Forest Classification/Regression, XGBoost Classification/Regression.

Classification: LightGBM Classification, Logistic Regression Classification, SGD Classification, SVM Classification, MultiLayer Perceptron Classification, Multinomial Naive Bayes Classification, Linear SVC Classification, Major Class Classification, CatBoost Classification, Gaussian Naive Bayes Classification, KNN Classification, Bayesian ARD Classification.

Regression: Bayesian Ridge Regression, ElasticNet Regression, Lasso Regression, Linear Regression, Ridge Regression, ARD Regression, LGBMRegressor.

Forecasting: ARIMA, AutoARIMA, ExponentialSmoothing, Prophet.

Third-Party Services: MinIO, Zookeeper, PostgreSQL 14, Kafka, Fluentd, Loki, Prometheus, Grafana, Kafka UI.

Product Architecture Diagram