- Analytics-Templates -
A Python Framework based on Palantir Foundry

Master's Thesis of Mike Siefert

Abstract

- The aim of the thesis is the design and the implementation of a framework that enables engineers and researchers (i.e. data scientists) to work together and provide quality ensured reusable code artifacts. The analytics-templates (AT) framework represents one common API in Palantir Foundry that encourages the collaboration between engineers and researchers.
- In order to demonstrate the applicability on data driven use cases, we implement a reusable component as a Palantir Foundry Code Workbook Template which solves a regression problem based on a defined configuration. The proof-of-concept based on a pricing use case in the life science industry that targets the discount prediction for a specific product. In order to evaluate the framework, a survey with 12 participants was conducted.
- Results: The survey shows an average grade of AT of 2.25 (1=best, 6=worst). 91% of those questioned will use the AT whereas 100% will contribute to the framework. Moreover, 91% of the participants believe that AT will reduce time to deliver of data products in a long run.

Motivation

- Data driven companies are dedicated to deliver data products with high quality standards.
- In order to reduce the time to deliver of data products, reusing of existing code artifacts (i.e. pieces of code that has been successfully applied and tested) can improve the development process.
- Especially the machine learning (ML) system anti-pattern *Glue Code* and *Pipeline Jungles* are considered as influence factors for technical debt in ML systems. In order to tackle these anti-patterns, one framework must be designed that encourages the collaboration between researchers and engineers.

Research Question

The target of the thesis is the design of a framework that supports the development of data driven analytical use cases. Therefore we need to investigate which specific tasks in the ML workflow are potential candidates to design as a reusable code artifact. Therefore, **is it possible to create a ML workflow with reusable code artifacts of that designed framework in Palantir Foundry?**

Architecture of framework

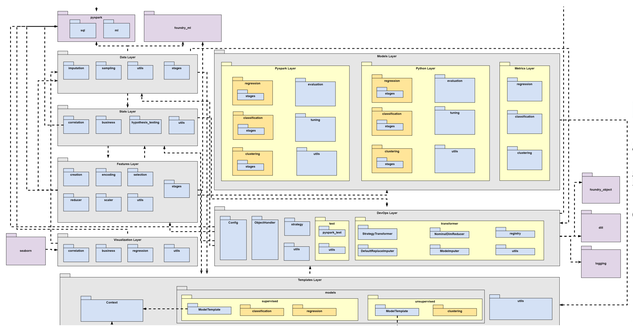

Figure 1:Process flow architecture

The architecture based on seven layers.

- **Data**: Data preparation tasks such as sampling or imputation.
- **Features**: Feature engineering tasks such as feature encoding, feature selection or feature simplification.
- **Models**: Different kind of ML models such as LinearRegression or RandomForest.
- **Visualization**: Predefined data visualizations such as barcharts or scatterplots.
- **Stats**: Calculations of business KPIs and additional statistical metrics.
- **Templates**: Workflow orchestration.
- **DevOps**: Development support for testing, configuration management and model registry.

Implementation of reusable component

The reusable component is implemented as Palantir Code Workbook Template that based on the analytics-template framework. The workflow implementation is shown in Fig.1.

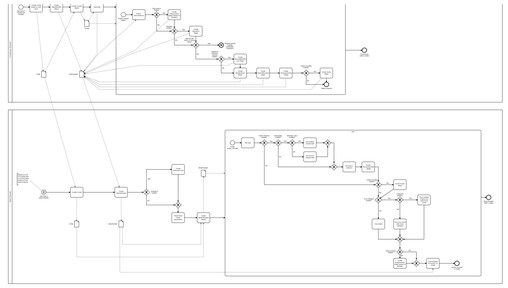

Figure 2:ML Workflow based on two templates that provide a simple user interface. Context template captures meta information about data and business context. Model template captures the model parameter configuration.

Application of reusable component

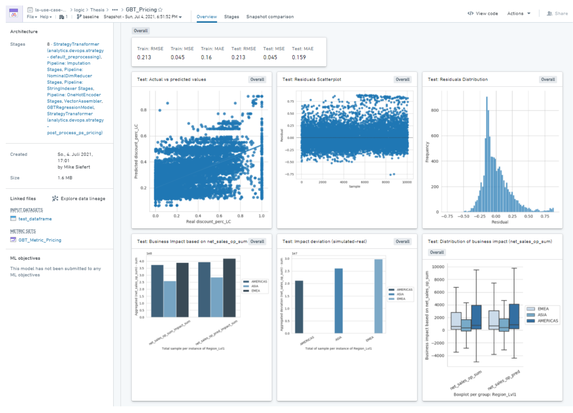

Figure 3:Example model output that shows six visualizations that show the model performance. (1) shows predicted vs. actual values. (2+3) show the residual distribution. (4) shows actual vs. simulated business net sales based on the given prediction. (5) shows the net sales deviation between simulated and actual net sales. (6) shows boxplots of the actual and the simulated net sales.

Survey Results

The survey was conducted after a training lab. The survey contained 26 questions which are grouped into subcategories *personal information*, *general*, *architecture*, *training*, *documentation*. **Participants**: Data engineers, analytics engineers and data scientists with different years of experiences.

- 11 out of 12 understand the purpose behind AT.
- 11 out of 12 think it will reduce the time to deliver data products in a long run.
- 11 out of 12 will use AT with a confidence level of medium to high.
- 12 out of 12 will contribute to AT and 91.6% are confident with at least a medium confidence level.
- 12 out of 12 like the architectural design, but only 83.3% find the reference architecture transparent.
- 12 out of 12 are able to reproduce the steps from the training lab.
- 9 out of 12 are not able to understand the field description of the templates if the meaning of a field is unknown.
- 8 out of 12 are able to contribute to AT with the given material.

Conclusion & Future Work

Finally, we conclude that it is possible to create a ML workflow with reusable code artifacts of the AT framework in Palantir Foundry. Although the number of participants in the survey is low, we see that we need to provide further guidance to help users contributing to the framework. In addition, following points need to be addressed in future work.

1. Investigation of alternatives to pyspark.ml in order to improve the run-time performance.
2. Evaluation of the usage of serverless architectures (e.g. AWS Lambda) that empowers the usage of additional resources without affecting the shared resources of the Palantir Foundry Platform.
3. Further development documentation and refinement of template field descriptions.