WHITE PAPER

PREDICTIVE MODELS FOR CONNECTED PRODUCTS

Mariana Osorio – Application Engineer – Data Science, Altair / February 27, 2023

This white paper outlines the steps necessary to implement machine learning predictive models for connected products using the Altair® RapidMiner® data analytics platform.

Introduction

Digital technology has changed the landscape of manufacturing and product creation: Internet of Things (IoT), artificial intelligence (AI), and data analytics are connecting organizations, generating data, driving more intelligent operations, and unlocking potential like never before. New skills are needed in the world of connected products and the success of innovation will depend on companies' digital capabilities. This is why the investments geared toward adoption of digital technologies, products, and services that allow companies to thrive in the fast-evolving economic environment is growing.

Manufacturers need to capitalize on the data generated by connected products in their transition to selling services, as this data offers very valuable information. Through its analysis one can gain insight into how to improve designs, find the cause of failures, perform timely maintenance, and optimize warranty strategies. On the other hand, users get extra functionality and a significant improvement in their product and related service.

Using predictive models to foresee product failures, estimate their remaining life, and detect unusual patterns lets companies enact preventative and corrective actions. As a result, companies can obtain more information about the root causes of failures to provide correct maintenance, and they can identify patterns of anomalies that lead to failures so they can avoid them altogether.

Altair RapidMiner enables an end-to-end analytical process that uses machine learning (ML) to create predictive models for connected products and deploys them in real-time dashboards. As technology rewrites business paradigms, organizations can remain competitive by making well-informed, insightful decisions using their data resources in an agile, collaborative, and customer-centric way. This white paper outlines the steps to increase value and reduce risk through analytic processes for connected products.

Data Analytics for Connected Products

ML technology that leverages historical and real-time data from sensors in connected products brings value to a number of different applications. Failure prediction, anomaly detection, remaining useful life and performance prediction, parameter optimization, and quality detection are just some of the areas where machine learning models show superior performance.

Every use case will have a specific workflow, but the general steps that comprise the data analytics workflow for connected products are shown in Figure 1. This paper talks through each step and exemplifies it through the use case of predictive models for failure in connected home appliances.

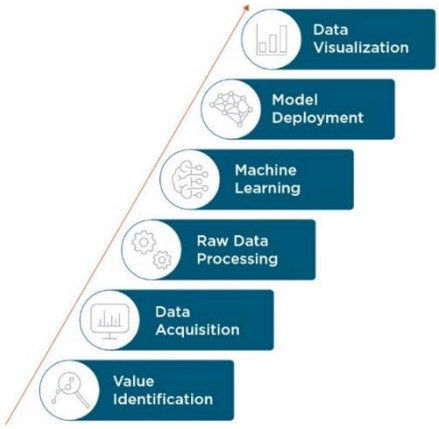


Figure 1 – The Data Analytics Workflow

Value Identification

The engineering team of a home appliances manufacturer sought to prevent failures on their connected refrigerators and improve their design. For this they created a failure detection solution using Altair RapidMiner. This solution created predictive models for refrigerator failures and implemented these models in a real-time display dashboard of the refrigerator performance with its failure probability. It facilitated investigative, preventive, and corrective actions for the operation of connected fridges. The team used two Altair RapidMiner applications to process the data, build predictive models, and deploy real time dashboards: Altair® Knowledge Studio® to build the ML models and Altair Panopticon™ visual analytics and stream processing software.

Data Acquisition

To build the solution, we first need to connect to the product's data. Each connected refrigerator has sensors that constantly send information to the cloud. This data is stored on a live SQL database. We need access to the historical records of the connected products within Knowledge Studio to build the models.

Knowledge Studio is an easy-to-use ML and predictive analytics desktop/server solution that lets the user visualize and profile data, use different modeling techniques to build ML models, and generate explainable results. Within Knowledge Studio we have an open canvas that builds analytic processes through a drag-and-drop interface (Figure 2). Here, we will build the model for the connected product solutions, starting with the data acquisition stage.

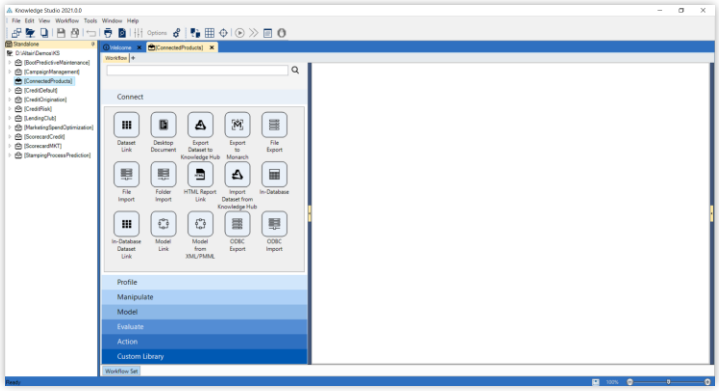


Figure 2 – Knowledge Studio is a wide-open canvas where we can connect wizard-driven nodes to build analytical solutions.

To connect to the historical database of the connected products we use Knowledge Studio's ODBC connection node in the Connect palette (Figure 3) to pull a copy of the refrigerators' history into the platform, where we are to pre-process and model the data. Knowledge Studio also supports in-database analytics, which allows users to perform data mining tasks directly in the source database without importing tables. Double-clicking on the ODBC node in the canvas will open its wizard (Figure 4), where we can connect to the connected product's historical database through different options (Figure 5).

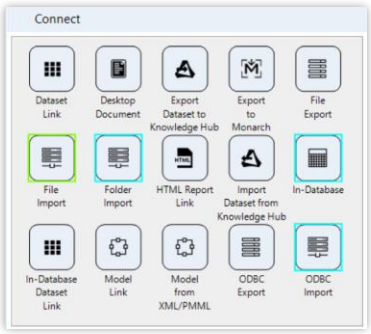


Figure 3 – Within Knowledge Studio we can connect to different data-sources through the Connect nodes palette.

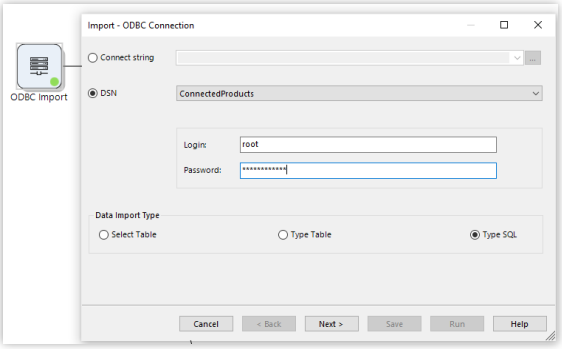


Figure 4 – To perform an action in Knowledge Studio we simply need to drag and drop the node onto the canvas and follow the instructions in the node's wizard.

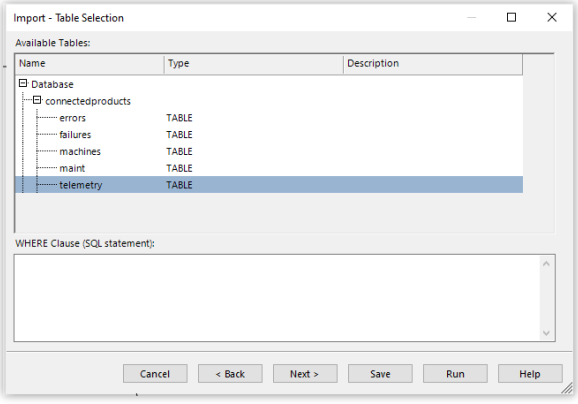


Figure 5 – The ODBC connection allows us to directly select a table in the database and bring it whole, bring it with filters and/or perform a direct query over the database to bring in the needed data.

Once we have defined the query to bring in our time history for the model, we can visually explore the data inside Knowledge Studio with its different profiling capabilities (Figure 6). Once we have imported and explored the raw data, it's time to move to the next stage.

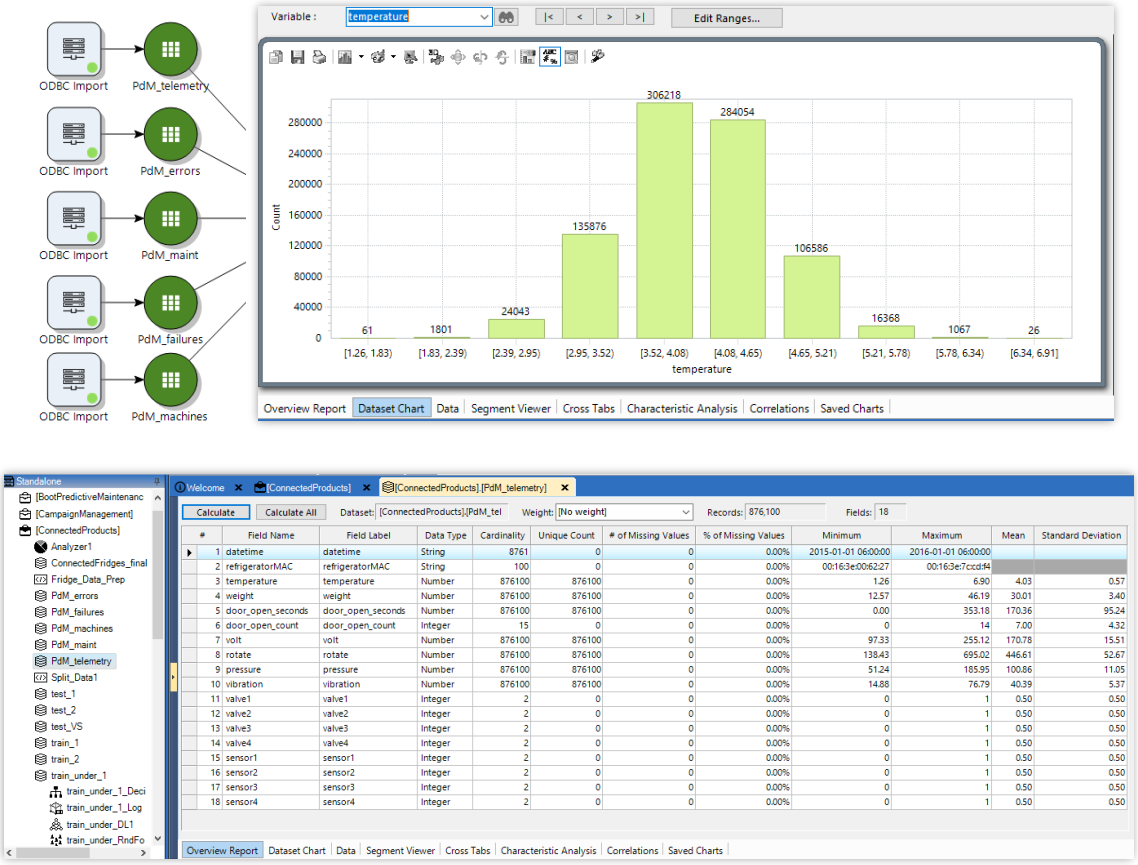


Figure 6 – When we bring in the data, we can see a green node that represents the dataset; opening the node allows the use of Knowledge Studio's multiple data profiling capabilities to explore and profile the data.

Raw Data Processing and Feature Engineering

One of the most important stages for the failure detection solution was the data pre-processing and feature engineering. The connected refrigerators generate signals for 15 sensors and software requests at different frequencies, which results in a large dataset with high numbers of missing data; additionally, we have the appliance's metadata, error data, and maintenance data that we must combine into a single dataset.

The raw data pre-processing entailed a careful profiling of the different variables closely with the engineering team responsible for the product. The first step was identifying variables of high predictive power and importance from an engineering standpoint. Afterwards, there was a careful design of the transformation process that converted the previously defined variables into relevant features for a ML model. This stage also required a treatment of null values, reduction of the data frequency, creation of rolling statistical aggregates (such as the mean of the refrigerator's weight every 24 hours), creation of more complex features (such as the time it takes to reach the target temperature), and splitting the data into training and validation subsets.

In failure prediction for connected products, very imbalanced datasets are often the case: a lot of data for correct performance and very little failure data. This is unhelpful while trying to build ML models, as learning from mostly no fail cases may lead to a poor classifier. The data for the connected refrigerators suffered from this imbalance. There are multiple techniques that alleviate it, and we chose *random undersampling*. Random undersampling eliminates random instances of the majority class. This technique improves the model's accuracy, but it may increase the variance of the classifier and potentially discard important samples.

Knowledge Studio's connection to Python was greatly leveraged for the raw data processing stage in this solution. Natively, Knowledge Studio has nodes to perform data transformation (Figure 7), however, more complex data processing techniques are available through its connection to Python and R code. We used Python's Pandas and NumPy packages to create the engineered features. We also leveraged scikit-learn, a machine learning package, to create stratified splits with random undersampling for the training and validation samples. After pulling the historical data into Knowledge Studio, we used custom Python nodes to process, prepare, and split this data to leave it ready for modeling (Figure 8). After this step, we were left with a dataset that contained 18 statistically important features to be used inside the ML models.

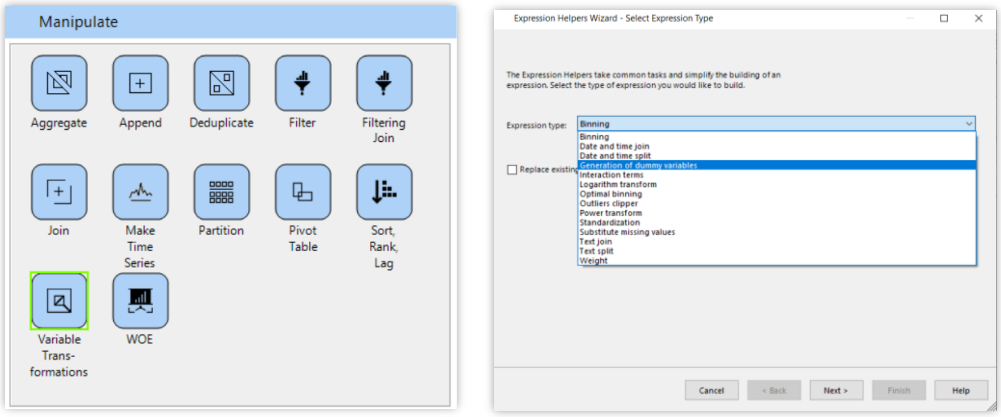


Figure 7 – Knowledge Studio has plenty of nodes to manipulates and pre-process raw data, they are in the Manipulate palette and they allow us to combine different datasets, and do feature engineering such as binning, standardization, and generation of dummy variables.

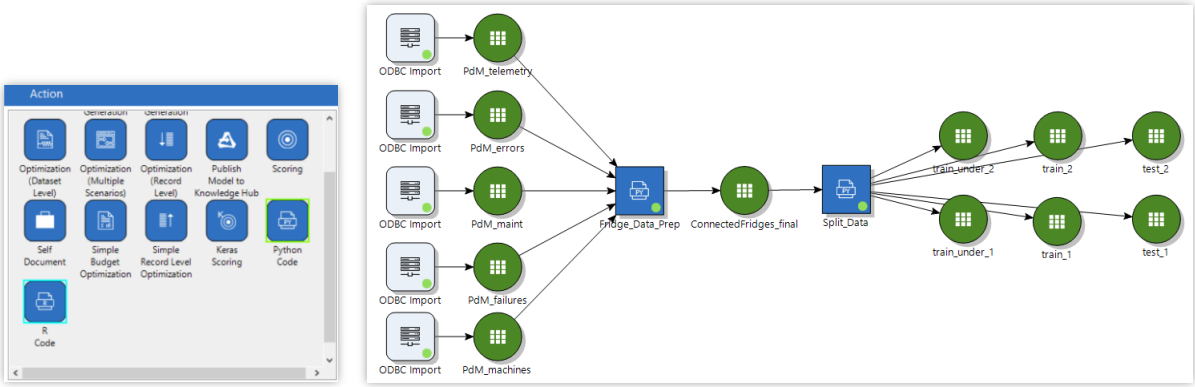


Figure 8 – Knowledge Studio can further expand the data preparation capabilities through its connection to R and Python transformations. For the complex data processing needed we used the Fridge_Data_Prep python node and to split the data we used the Split_Data python node.

Machine Learning and Model Creation

The failure detection solution for refrigerators generated a binary classification model for each failure type. This was due to the fact that each class of failure related differently to the sensor variables (for example, the voltage failure was closely related to high voltage deviations due to the home's electric connection, unlike one of the use failures for which the sensors of the opening and closings of the fridge's doors were highly important) and each failure class was highly unbalanced. Therefore, it was not possible to create a single multi-class classifier for all the different types of failure.

Training models inside Knowledge Studio is easy. Knowledge Studio allows us to choose what type of modeling technique to use and to define every parameter in the model and tune it to fit the needs of the application through no-code wizards (Figure 9). Details about how a model is configured and what the model's output means can be shown using explainable AI. Along with AutoML Knowledge Studio's approach towards responsible AI means all users of a model's output can be confident in making decisions, knowing how and why a prediction was made.

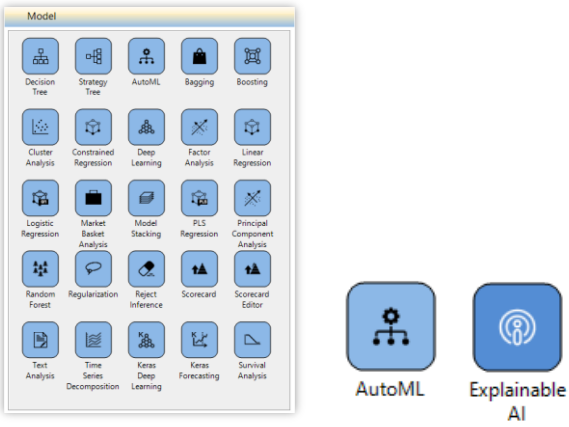


Figure 9 – Knowledge Studio generates insights by using a wide range of modeling techniques and algorithms, including decision trees, regression models, and deep learning (neural networks).

To train the models for our application we simply drag and drop our different choices into the canvas and define the variables and parameters for training (Figure 10). We performed these steps for several models to build our different choices (Figure 11), each model built can be validated through evaluation nodes on a test dataset and we can see how well it performs on new data. Inside Knowledge Studio, we can also build interactive decision trees to analyze the impact of different variables for the final predictions (Figure 12).

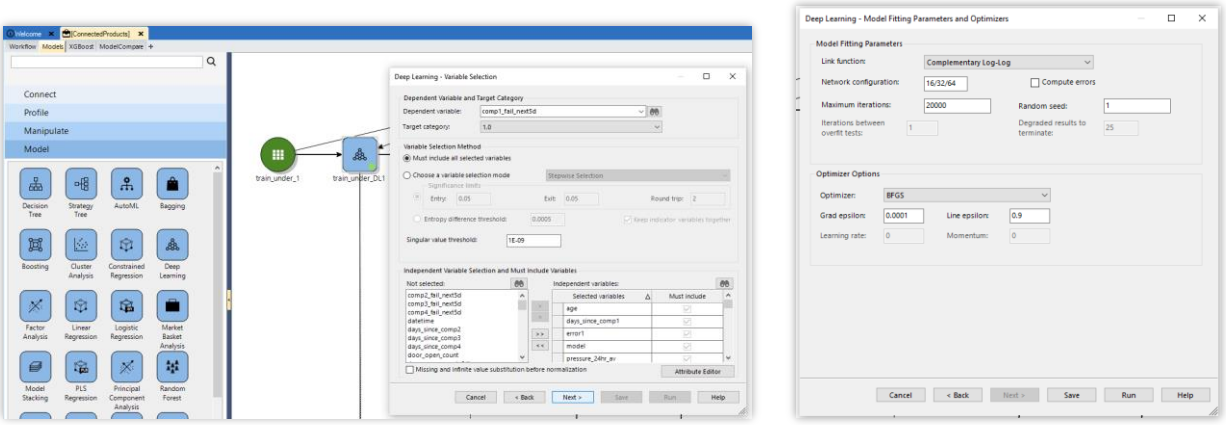


Figure 10 – Deep learning wizard for training a three layered dense neural network with 16-32-64 neurons in the respective layers. Knowledge Studio lets the user configure the training's variable selection method and model and optimizer parameters.

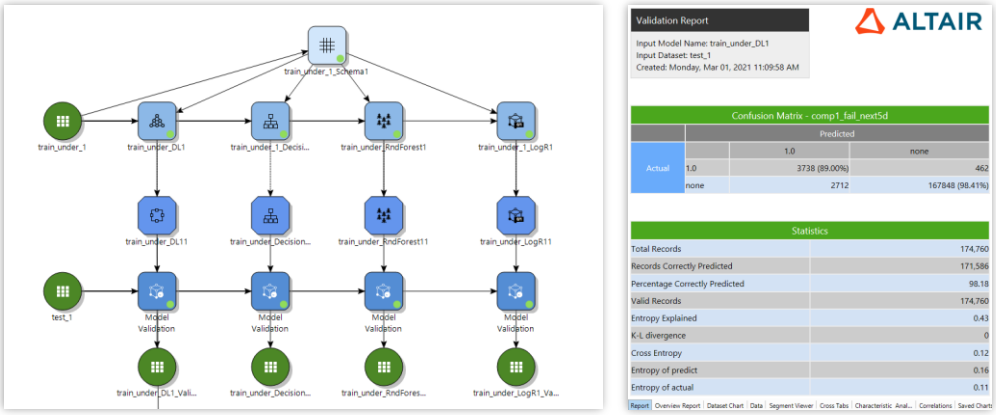


Figure 11 – After training several different models on the under-sampled training data, the user can validate the results on the test data and see the performance of each model.

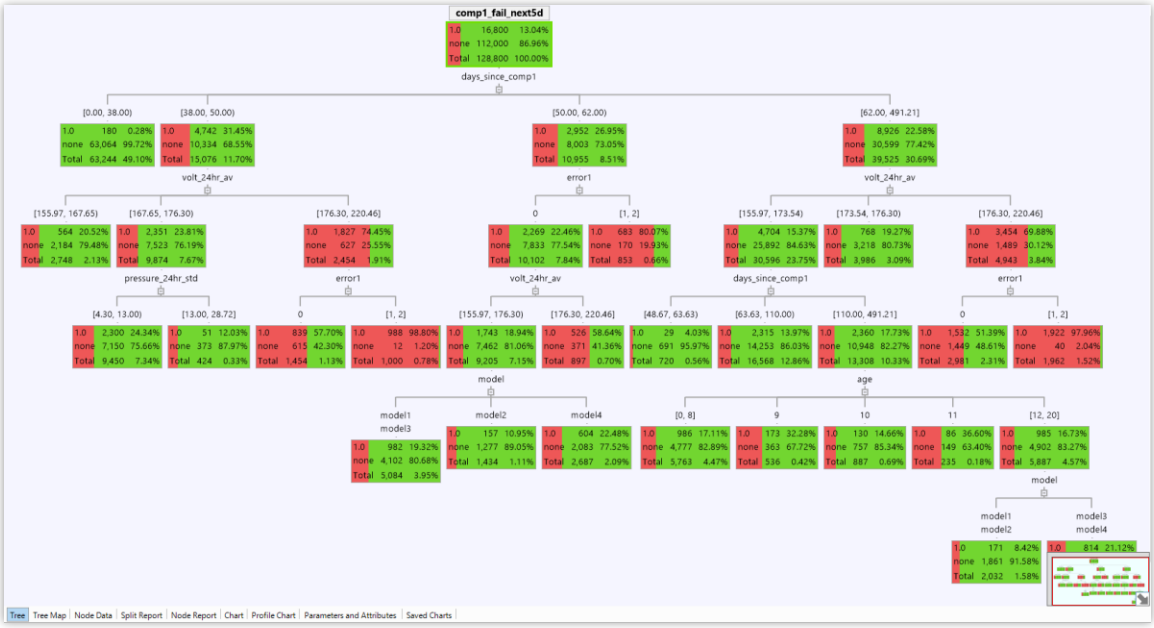
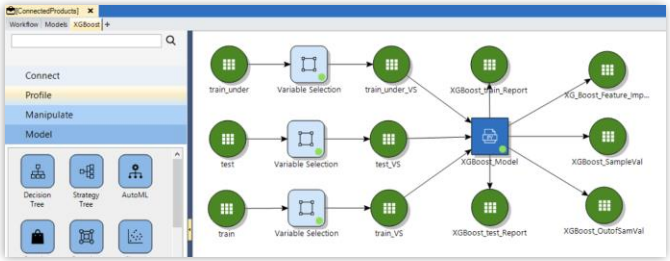


Figure 12 – Decision trees are visual models and the user can see the splits and rules that govern the model's predictions. Knowledge Studio has interactive decision trees, so the data science team can modify the bins and splits to incorporate business knowledge.

Inside Knowledge Studio, we can also leverage Python for modeling, not just pre-processing, and we can implement other types of models that aren't natively supported by Knowledge Studio. For this example, we used a custom node for an XGBoost implementation (Figure 13). Running the code inside Knowledge Studio makes it possible to compare all models inside the application and to leverage our visual capabilities. So finally, we used the validation and model analyzer nodes to compare all the trained models (Figure 14).



	Actual_val	Predicted_0	Predicted_1	Predicted_0_%	Predicted_1_%
1	0	154651	15909	0.90672490619137	0.0932750938086304
2	1	19	4181	0.00452380952380952	0.995476190476191

Figure 13 – Aside from Knowledge Studio's native modeling techniques, we can implement models in Python or R through the code nodes. Here is an example of an XGBoost implementation in Python for the connected products use case.

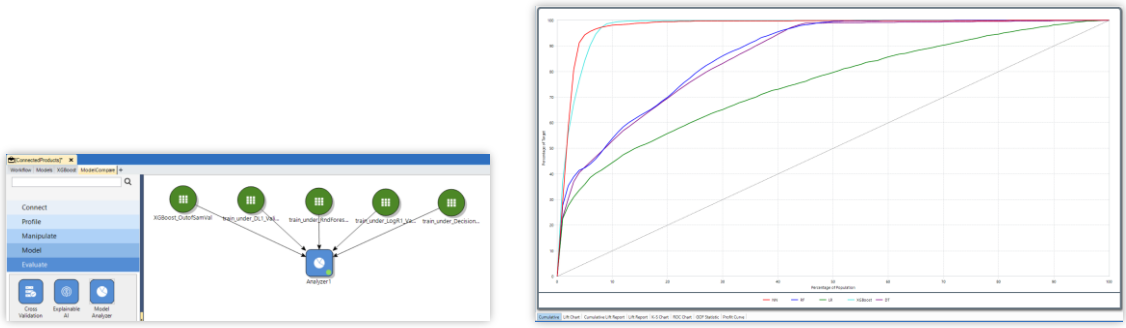


Figure 14 – Knowledge Studio also lets you validate and analyze the different trained models to choose which one is best. On the left is the cumulative chart generated by the model analyzer for the trained models.

After this process, it was concluded that the best models for failure classification in the validation stage were the dense neural networks and the XGboost models. The final step is to export the trained models to python formats to be deployed in Panopticon (Figure 15).

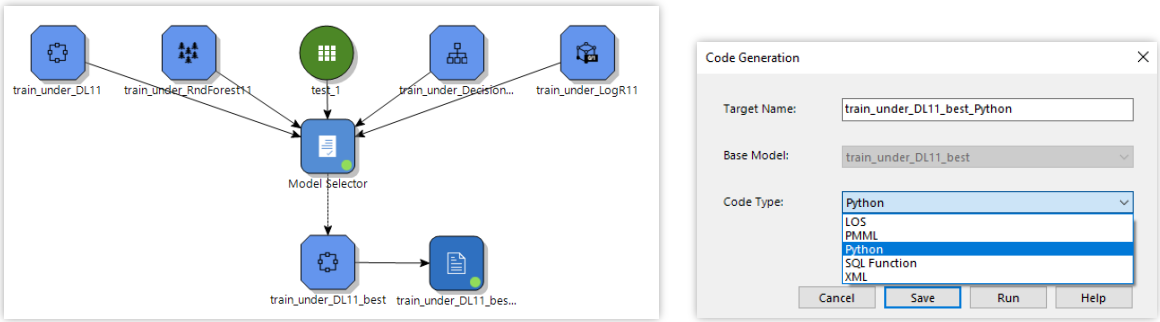


Figure 15 – The last step before deploying the model is exporting it into a format that can be used in the platform where it will be implemented. Knowledge Studio can export models into several languages, and in this case, we exported it to Python to use it inside Panopticon.

Model Deployment and Data Visualization

After developing the models, the *Failure Detection Solution* for refrigerators was deployed on a set of visual dashboards in Panopticon that allowed the following:

1. Dashboard to view the fridge's data in real time and score its failure probability, most granular data.
2. Dashboard for an aggregated view of all the refrigerators, their failure probabilities, and the possibility to highlight groups that may be more prone to fail.
3. Dashboard to consult the model's agnostic interpretation (explainable AI) to make a root cause analysis of each of the failure predictions to be able to take corrective action and understand the errors fully.

Panopticon is a server-based application that lets business users and engineers build, modify, and deploy sophisticated streaming analytics and data visualization applications using a fully drag-and-drop interface (Figure 16). They can connect to virtually any data source, develop complex stream processing programs, and design visual user interfaces that give them the perspectives they need to make insightful, fully informed decisions based on massive amounts of fast-changing data. It can also interact with Python and R to implement ML models in the visual dashboards.

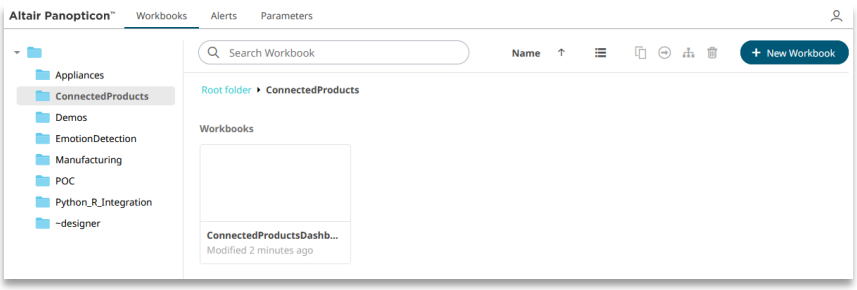


Figure 16 – Panopticon's web-authoring interface.

To connect to the live database with the data we used Panopticon's out of the box connections that allow for subscriptions to streaming sources and loading databases on a set frequency. This way we were able to connect to the product's data through the JDBC connector, so engineers and analysts can enter the dashboards and see the real time performance and predictions for failure (Figure 17).

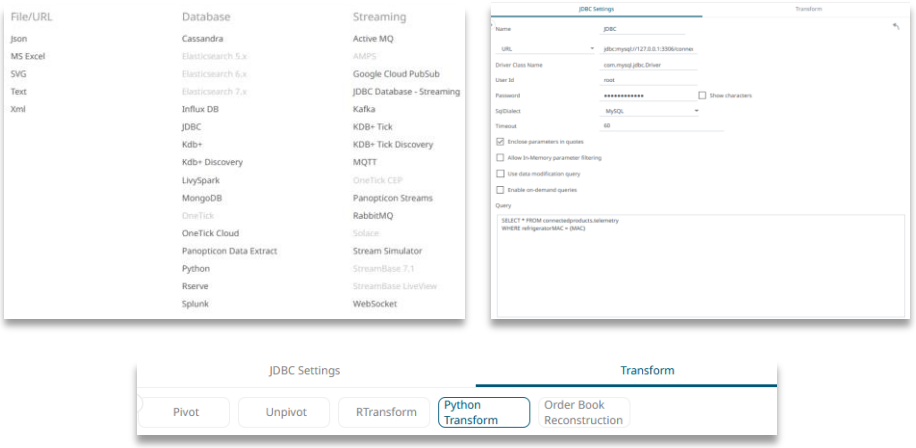


Figure 17 – Panopticon's JDBC connection. Each data source can be transformed through different techniques, the Python transform allows us to use Knowledge Studio's previously trained model to score new data.

As with Knowledge Studio, Panopticon can also connect to Python and R to transform the data to be visualized. Since we build the predictive models on clean data, we need the data in the live dashboards to have the same format, so we can use the models to score it. Inside Panopticon, we used the same Python transformations that were used on Knowledge Studio to process the data and we can also perform the joins on the SQL tables to have the complete picture of all the data. After these transformations, we have a clean dataset that can be scored with the previously trained predictive model (Figure 18).

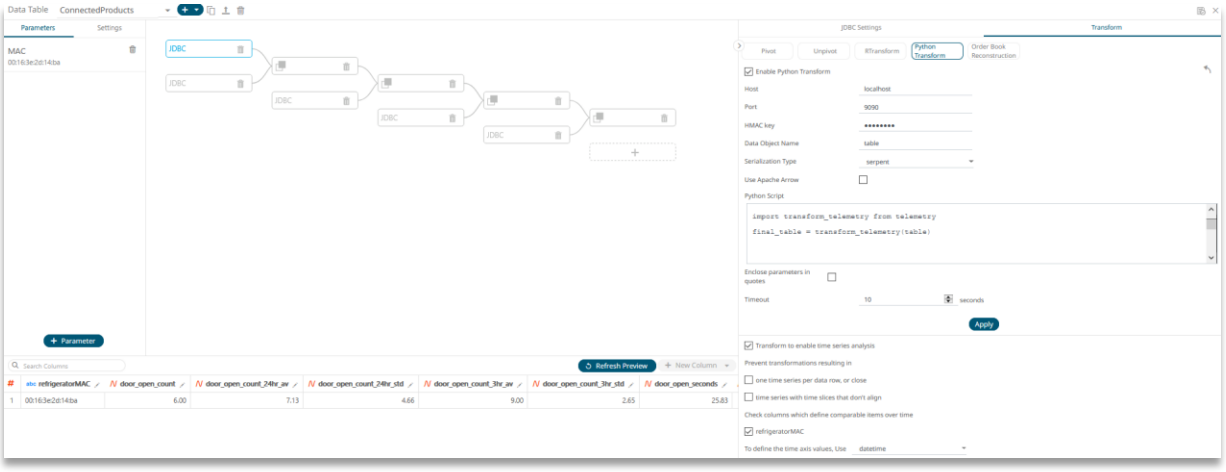


Figure 18 – For this application we connect to 5 different databases in a SQL server. After transforming each data base through python to apply the data pre-processing defined in Knowledge Studio, we transform the five tables to timeseries and join them, finally we score the live data to generate live model predictions.

Once the data source is set, we can start building the visual dashboard as needed. This just means creating graphs, tables and buttons inside the application and using the variables from the connected product's data (Figure 19). These dashboards can be as customized as the user needs and they can even incorporate interactivity, choosing certain parameters, and exploring what if scenarios.

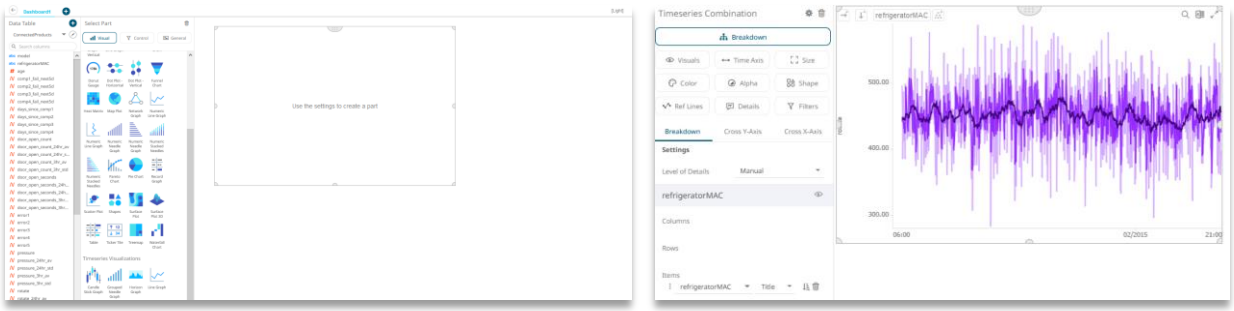


Figure 19 – Once the data source is set, the user can start to build the visual dashboards with the graphs and variables as needed.

Finally, after the process of designing the dashboards, we have the complete visual dashboard implementation of the refrigerator's data with the predictive model's scores (Figure 20). The engineers can use this to prevent failures, explore the reasons behind them and generate powerful, visual insights from the data, all of which leads to future improvements in the product design.

Figure 20 – Final visual dashboard with the sensor data and the failure probabilities from the predictive model.

Conclusion

Low-cost sensors and new wireless connectivity tools let businesses employ digital analytics more effectively than ever. With the right tools, they can create products that are interconnected, generate data, and ensure intelligent performance and functionality. Consequently, it has become imperative to cleanse, process, analyze, and draw insights from these massive amounts of data that cover the product's life.

Using ML, we can leverage technology to deepen our understanding of what's affecting our connected products and create strategies to predict and prevent failure. By analyzing the product's data, we can build models to predict future failures of the whole product or its components to prevent them from happening and accelerate and optimize the product service/maintenance actions. We can estimate the remaining useful life of the product to adequately design the warranty strategy, create maintenance suggestions, and optimize the component replacement's inventory. We can also create anomaly detection models that detect unusual behavior and may prevent major operational issues.

Through a six step process, the most important stages of building an analytic solution for connected products were highlighted: identifying value, acquiring the necessary data, processing, and turning it into valuable features, training and validating ML models, deploying the models, and monitoring the live data and model predictions to ensure functionality. The example of a home appliance manufacturer that successfully built a connected products' analytic solution to predict failure in its refrigerators was shown in detail to outline the steps for these types of applications and the simplicity of building them inside Altair's RapidMiner platform.

Altair's predictive analytics solution makes it easy for people of different skillsets to build analytical applications or augment analytics into existing applications to use smart data for insightful, informed decision making. Leveraging 30+ years' experience in manufacturing, product design, and machine learning, Altair RapidMiner is an automated, repeatable, and sustainable solution that's easy to deploy and supports the complete data lifecycle. Engineers can architect a complete end-to-end analytic process pipeline that supports a data-driven enterprise operation for connected products.

Great product quality wins new customers and keeps existing customers coming back. Consumers and business buyers expect high build quality, long mean times before repair, and long product life – Altair's RapidMiner ensures connected products meet these expectations.