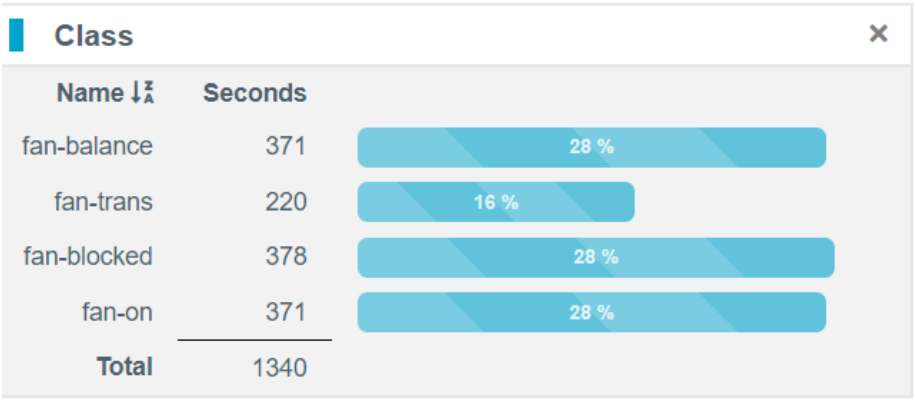
Reality AI Tools Tutorial

Using Vibration Sensor Dataset

Description

This document covers various functionality of Renesas's Reality AI Tools portal. The portal is useful to generate AI models for non-visual sensing application. We will use vibration dataset -> collected using a 3-axis accelerometer for this tutorial. The dataset was created to identify different conditions (image below) of a fan stack:



Below are the sections we will go over:

Lab Section	Reality AI Tools Features
Section 1	Creating Project and Uploading Dataset
Section 2	Curating and Preprocessing Dataset
Section 3	Analyzing Data and Creating Initial Models
Section 4	Testing the Models
Section 5	Exporting the Model
Section 6	BoM Optimization

Lab Objectives

- Gain hands-on experience on Reality AI Tools using vibration dataset.
- Learn how to use analyse and create ML models for non-visual sensing applications.

Lab Materials

- Please verify you have the following materials at your lab station.
  - PC or MacBook with Chrome or Firefox Installed.
  - Tutorial data (if made available) → 21 CSV files containing vibration data from a 3-axis accelerometer. **Note: you don't need this data if you have the Explorer Tier/ Freemium access.**

Time

- 90 minutes

Lab Sections

© 2024 Renesas Electronics. All rights reserved. Download and disclosure restricted by Reality AI Tools® Terms & Conditions.

1 Creating Project and Uploading Dataset 3
2 Curating and Preprocessing Dataset 8
3 Analyzing Data and Creating Initial Models 12
4 Testing the Model 20
5 Exporting the Model 24
6 BoM Optimization 30

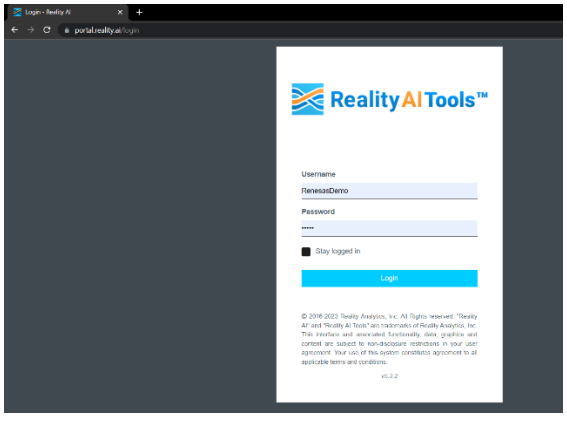
© 2024 Renesas Electronics. All rights reserved. Download and disclosure restricted by Reality AI Tools® Terms & Conditions.

1 Creating Project and Uploading Dataset

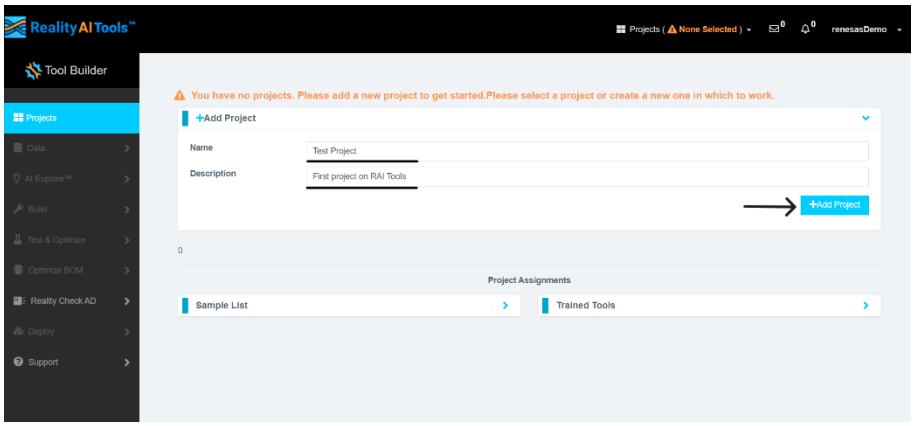
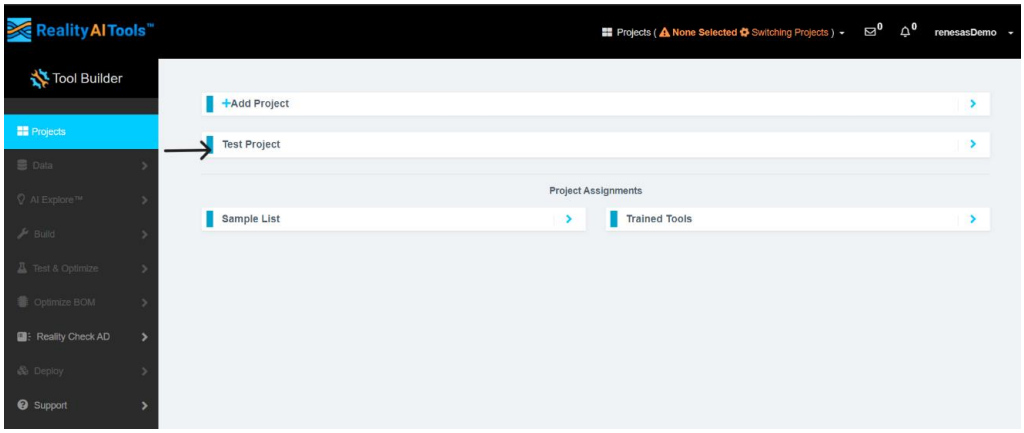
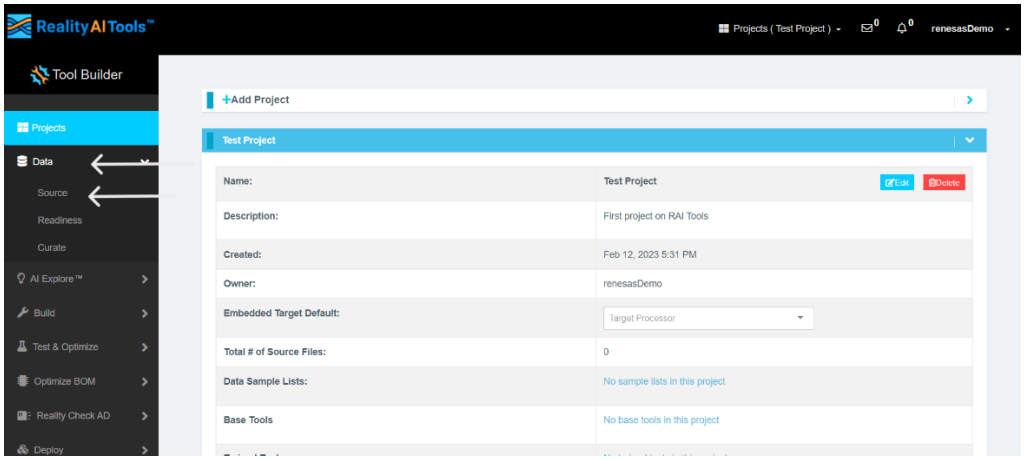
Overview

In this section, we will setup a project, upload and format the dataset on Reality AI Tools.

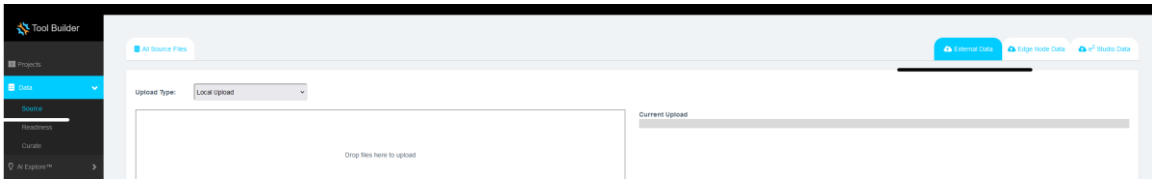
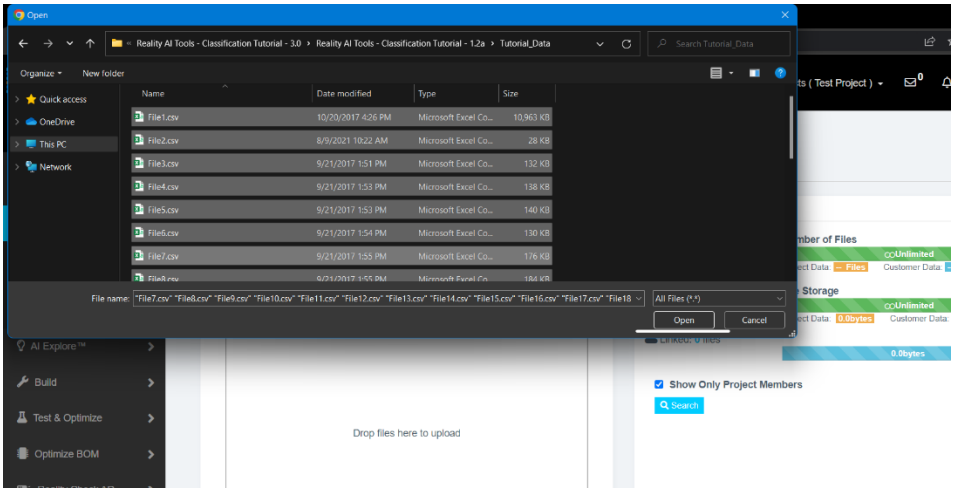
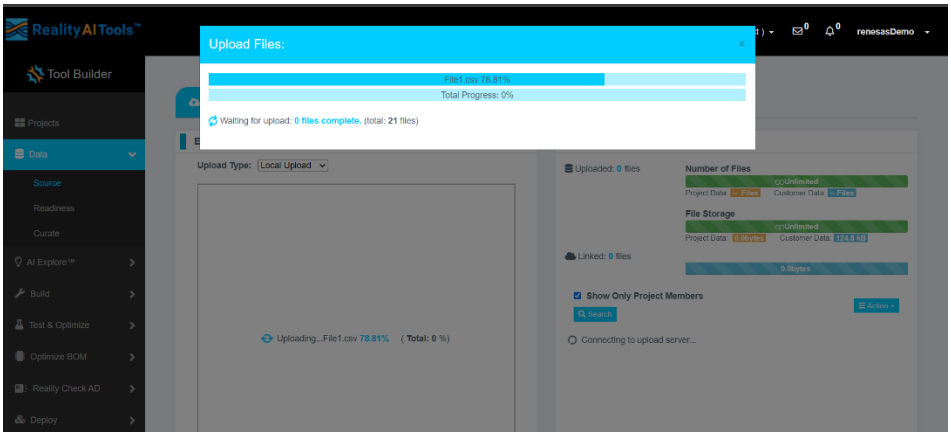
Procedural Steps

1.1	Note: If you got the Reality AI Tools credentials via the Explorer Tier/ Freemium access route, this section is for information purpose only --> Mainly because your Tools account already has example projects with data preloaded to experiment on. The data used in this section is part of the **Fan Condition Monitoring** project. As Freemium user, please review this section for information purposes (you are welcome to use your own data to upload and experiment upon based on the below outlined steps. However, if you want to focus your experimentation on just the tutorial datasets, please navigate to the next section). Finger Friction using RA6T2 board + E2 Studio Integration (ms4DE) Motion Detection using AIK + E2 Studio Integration (ms4DE) Asset Tracking using CK - RA6M5 + E2 Studio Integration (ms4DE) Pump Condition Monitoring (AhoNN) Activity Detection - running vs walking (AhoNN) Urban Scene Classification (AhoNN) Fan Condition Monitoring (AhoNN) If you are a Tools customer via subscription/ Get Started Package, use the below steps. Unzip Files from the tutorial package → "TutorialData.zip". Verify that there are 21 CSV data files in the folder.
1.2	Open your browser and navigate to the URL: **portal.reality.ai** Enter your login credentials (provided via an email) on the page. Click **Login**. 

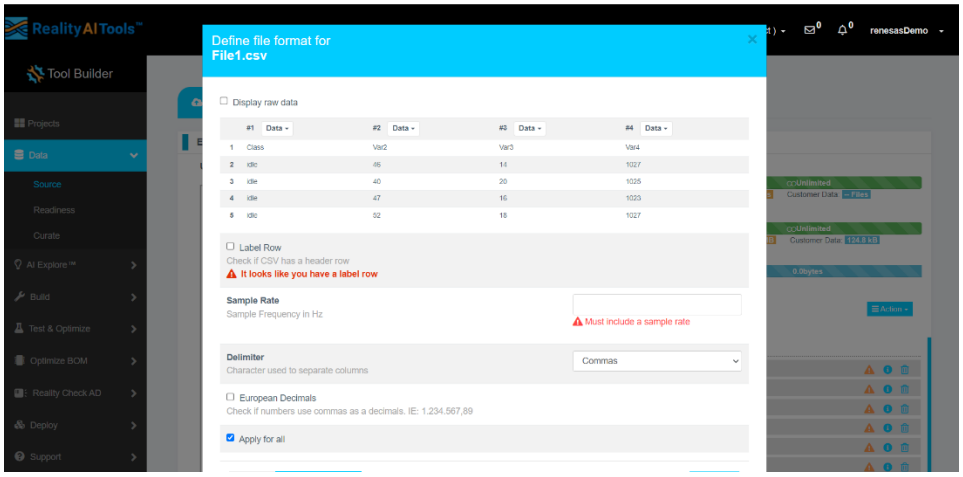
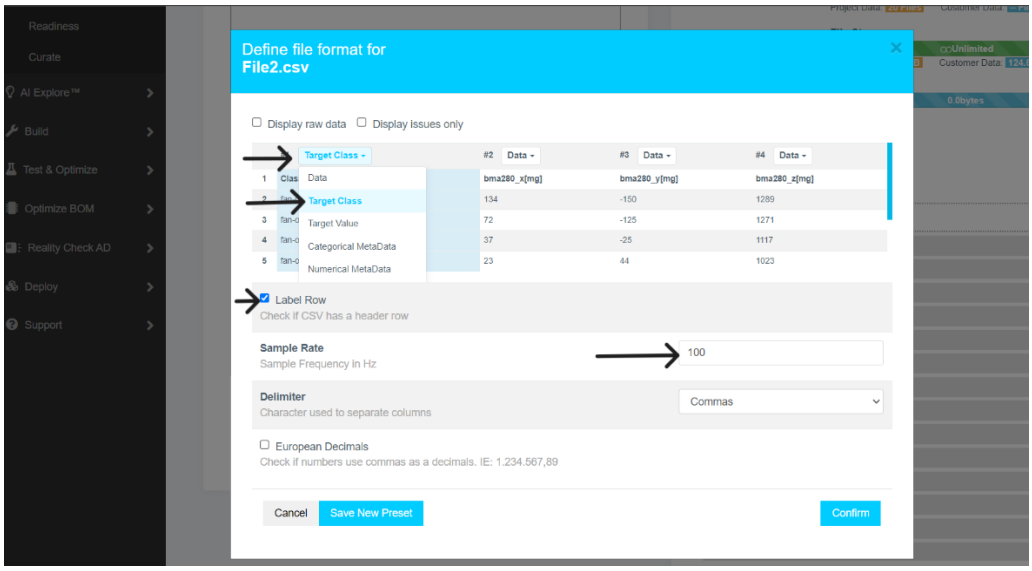
© 2024 Renesas Electronics. All rights reserved. Download and disclosure restricted by Reality AI Tools® Terms & Conditions.

1.3	After logging in, you will see the portal with no projects. As a first step, • we will create a project (this will contain our dataset + all the analysis we perform in future). • Enter a Project Name • Description (Optional) click **Add Project** as highlighted by the arrow. 
1.4	After creating the project, click on the newly created project tab (as shown by the arrow). This will select that project and allow us to perform various functions in it. 
1.5	On the left menu, click on **Data** and the **Source** options. 

© 2024 Renesas Electronics. All rights reserved. Download and disclosure restricted by Reality AI Tools® Terms & Conditions.

1.6	Click on **External Data** then **Drop file here to upload** to upload CSV file on the portal. 
1.7	Navigate to the TutorialData folder. Select all the file and click on **Open**. 
1.8	You can see the progress of your uploads on the page file-by-file. 

© 2024 Renesas Electronics. All rights reserved. Download and disclosure restricted by Reality AI Tools® Terms & Conditions.

1.9	Below is the first page that will pop-up to format the files. You will have to assign **Target Class** and **Sample Rate** (next step). Note: This formatting will be applied to all the 21 files that were uploaded. 
1.10	Click on Data dropdown and select **Target Class** to assign lable to the first column of the data file. • Select Label Row checkbox. • Assign sample rate (**100**). Click **Confirm** button to finalize formatting. 
	Why Target Class? Class represents labels for classification problems in Machine Learning. This is a type of data labelling used when we want to analyze and predict categorical events. In this case, it's the state/ condition of the fan: Fan-On, fan-Off, etc. In another case, it could just be a normal vs abnormal state of a machine. Target Value labelling would represent an integer or float value. The Tools only cares about 3 types of options from the above drop-down: Target Class OR Target Value for labelling and Data for signifying raw sensor data. Everything else in the list (time, date, numeric metadata) just represents additional information which is useful for sorting of the results. That information does not effect the model training/ generation process.
1.11	**You have completed this section.**

© 2024 Renesas Electronics. All rights reserved. Download and disclosure restricted by Reality AI Tools® Terms & Conditions.

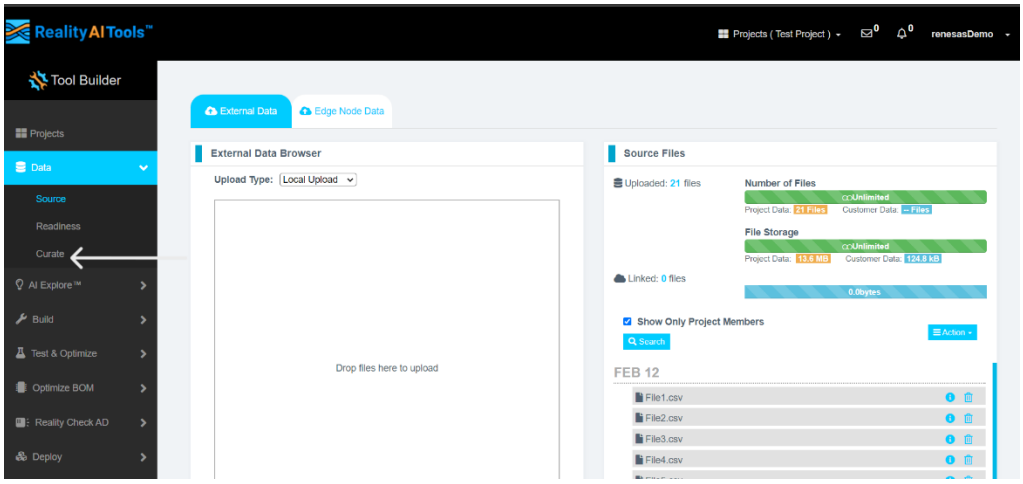
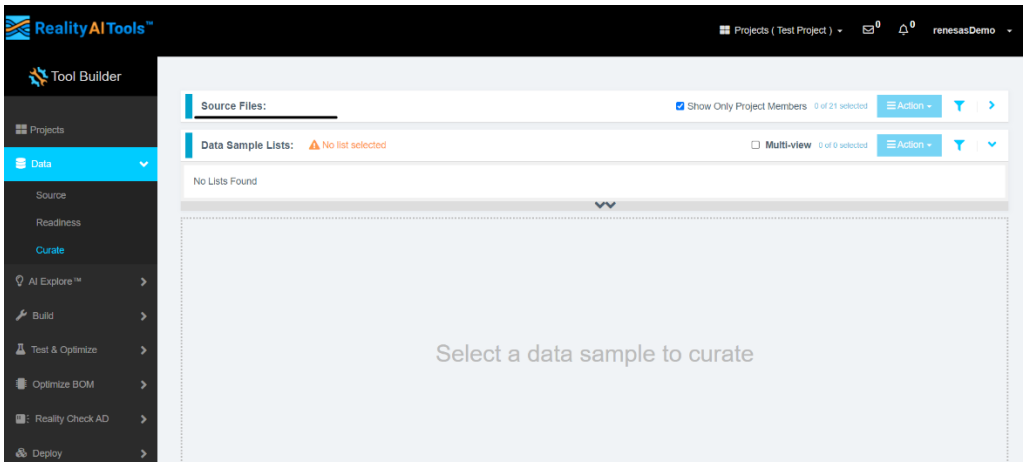
© 2024 Renesas Electronics. All rights reserved. Download and disclosure restricted by Reality AI Tools® Terms & Conditions.

2 Curating and Preprocessing Dataset

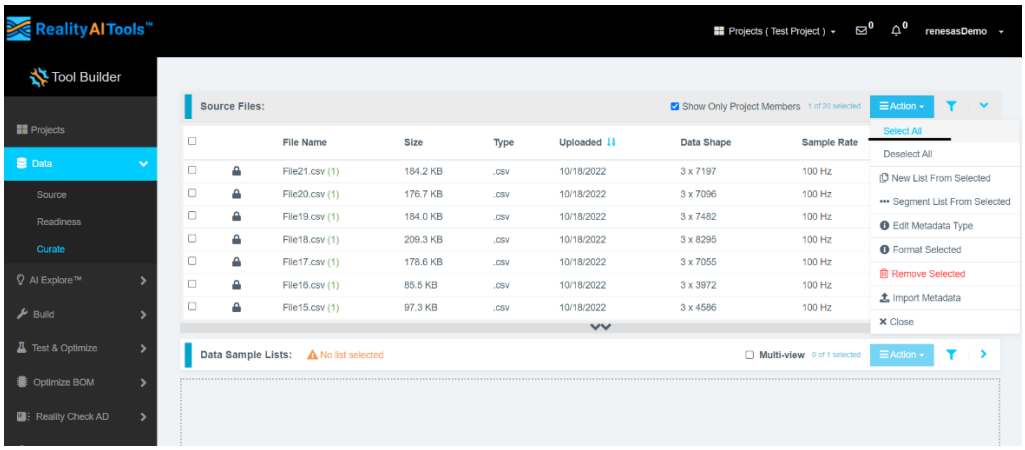
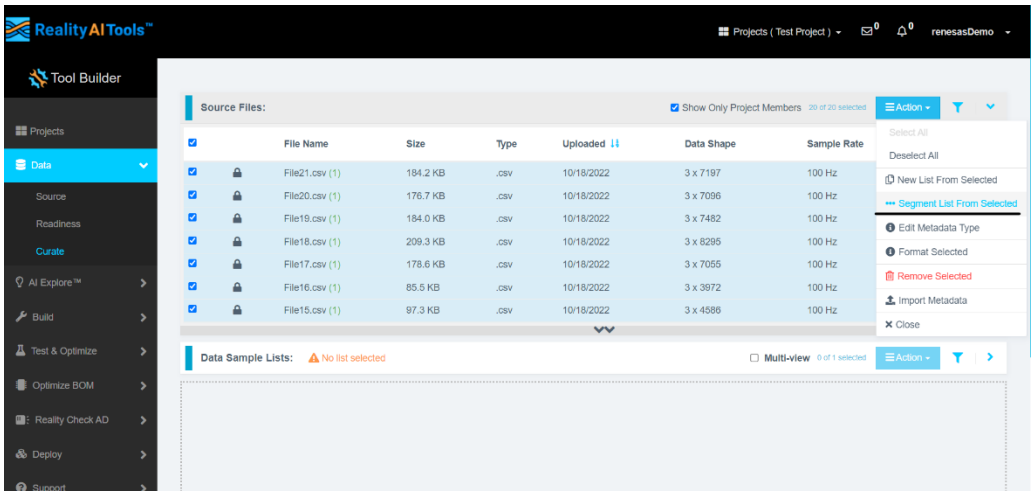
Overview

In this section, we will preprocess the data files by breaking them down into smaller blocks/ pieces/ segments. Breaking down the data files is important as these blocks will be the input to our ML model. For example, if your data files is 10 seconds long → we can break it down into 1 second long 10 segments. We can choose the length of these segments based on prior knowledge of a use-case or start with a 1 second window length and experiment from there. As part of this tutorial, we will use a 1 second window length.

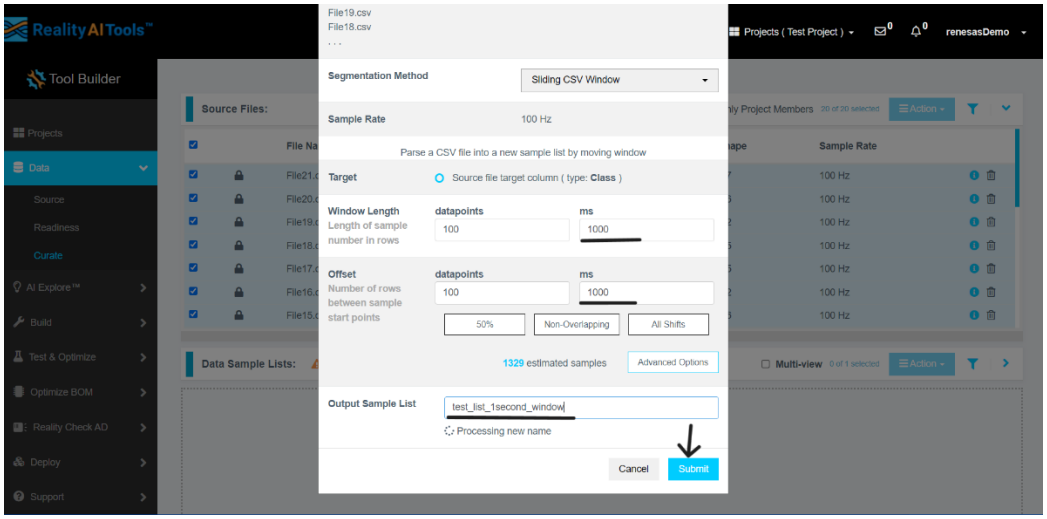
Procedural Steps

2.1	On the left menu, click on **Data** and the **Curate** options. 
2.2	Click on the **Source Files** tab to expand the section (highlighted by the black line). 

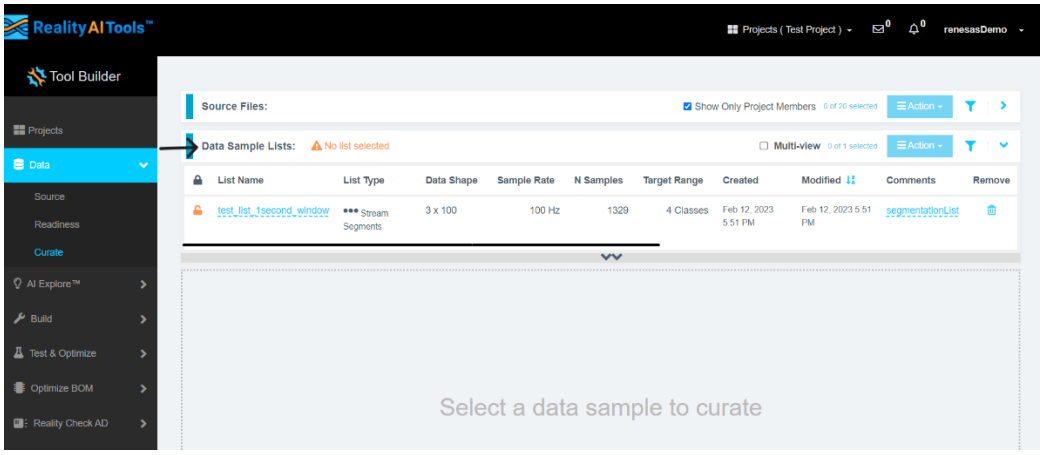
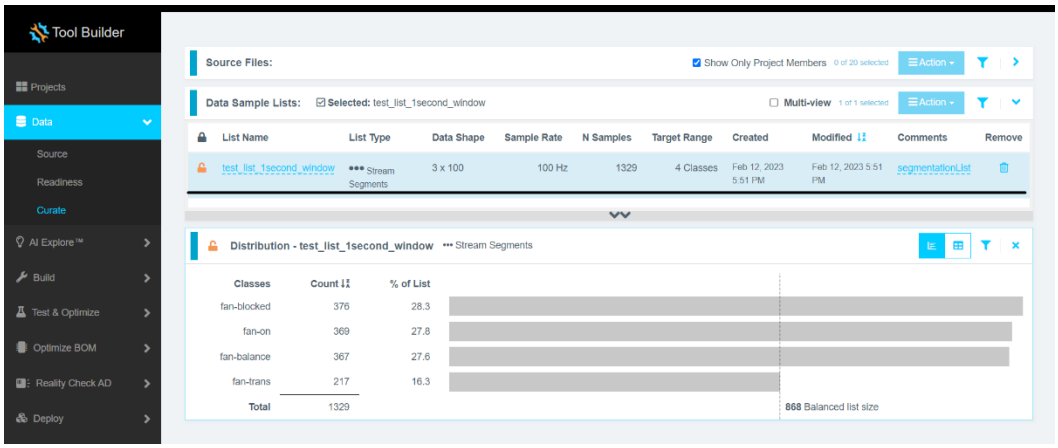
© 2024 Renesas Electronics. All rights reserved. Download and disclosure restricted by Reality AI Tools® Terms & Conditions.

2.3	Once you see al the uploaded CS files, click on **Action** button and then **Select All**. This will select all the CSV files for segmentation process. 
2.4	Click on **Segment List from Selected** to bring up the segmentation window. 
	Why Segmentation?: One of the main purposes of generating models from Tools is to deploy them to a variety of Renesas MCU's. To that effect, these models have to process live data within a resource constrained environment. So for practical purposes, a model might be looking at 1 second or 500 ms or even a smaller window to make quick predictions. As opposed to a few seconds (or minutes or hours) long data stream. To mimic that effect, we break down the raw training data and feed that to the model generation engine to start learning what its gonna see in a live (production) setting.

© 2024 Renesas Electronics. All rights reserved. Download and disclosure restricted by Reality AI Tools® Terms & Conditions.

2.5	Once segmentation window pops-up, you will need to enter a few parameters to break down the data into smaller segments/ blocks. Parameters: **Window length**: Enter 1000 miliseconds. **Offset**: Enter 1000 miliseconds. After you are done with setting the parameters, enter a name and click **Submit**. 
	Additional Information: **Window Length:** The window length determines how much data will be considered by the AI to decide on a classification. You may want to experiment with different window lengths to see what works best in future. **Offset:** The offset determines how far ahead from the start of the last window the parser moves before creating a new window. **General Guide**: **50% Overlap** is usually a good compromise between covering starting-point variations in the data and too much redundancy. Use **non-overlapping** windows when you have a great deal of data, perhaps with longer offsets. Typically you'll do initial exploration and training on a subset of the available data. Use **All Shifts** (offset = 1) for testing after you have a good candidate classifier and you want to simulate performance on a stream of new data arbitrarily sampled. **Limitation**: Freemium/ Explorer Tier user have limitation of creating segmented lists with size < 7000 samples.

© 2024 Renesas Electronics. All rights reserved. Download and disclosure restricted by Reality AI Tools® Terms & Conditions.

2.6	After clicking on Submit, wait for a ~30 seconds and refresh the page. After refreshing the page, click on the **Data Sample List** tab. After clicking on the tab, you should be able to see the finished processed list. Note: It takes < 1min to segment/ break down this dataset. However, the duration of segmentation depends on the size of the dataset. For example: A 1 GB file may take 5-10 minutes to segment. 
2.7	Click anywhere on the newly created segmented list → This list contains the blocks/ windows of segments. You should be able to see a histogram of the data. **Note**: You may have to refresh the page if histogram doesn't pop-up immediately after segmentation. 
2.8	**You have completed this section.**

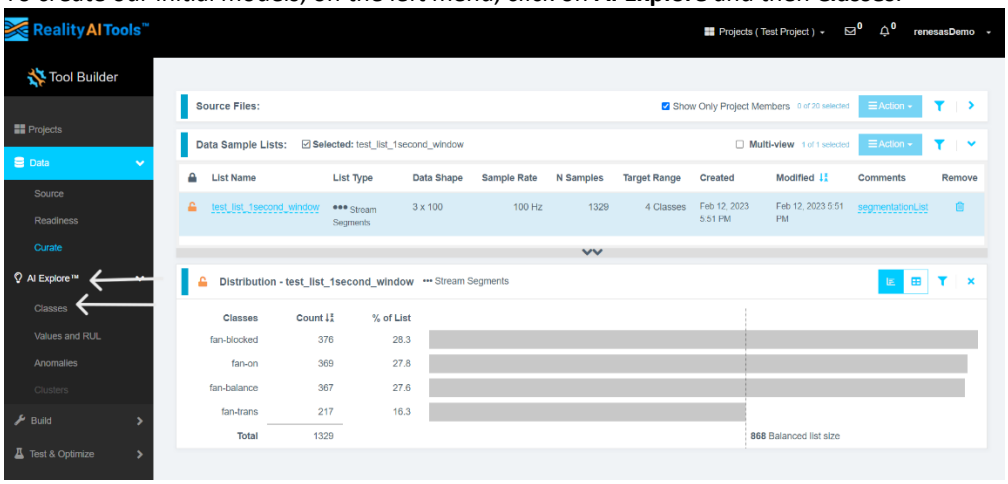
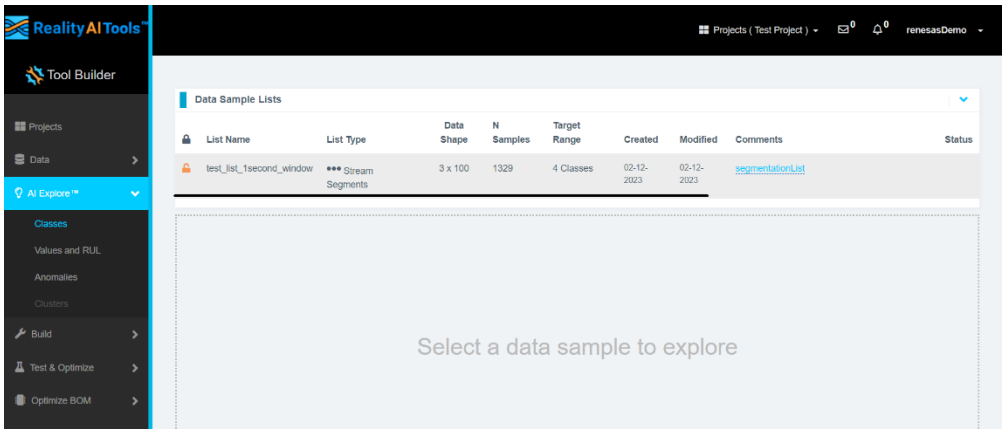
© 2024 Renesas Electronics. All rights reserved. Download and disclosure restricted by Reality AI Tools® Terms & Conditions.

3 Analyzing Data and Creating Initial Models

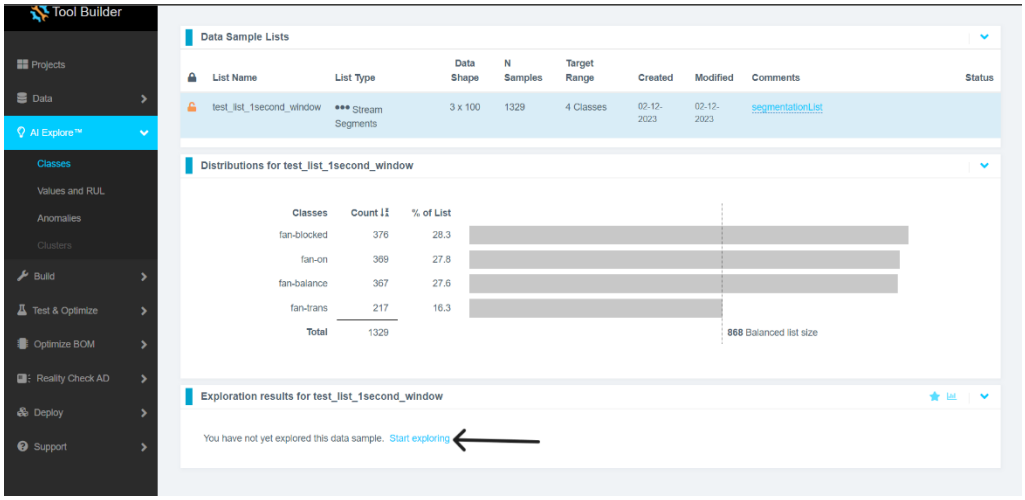
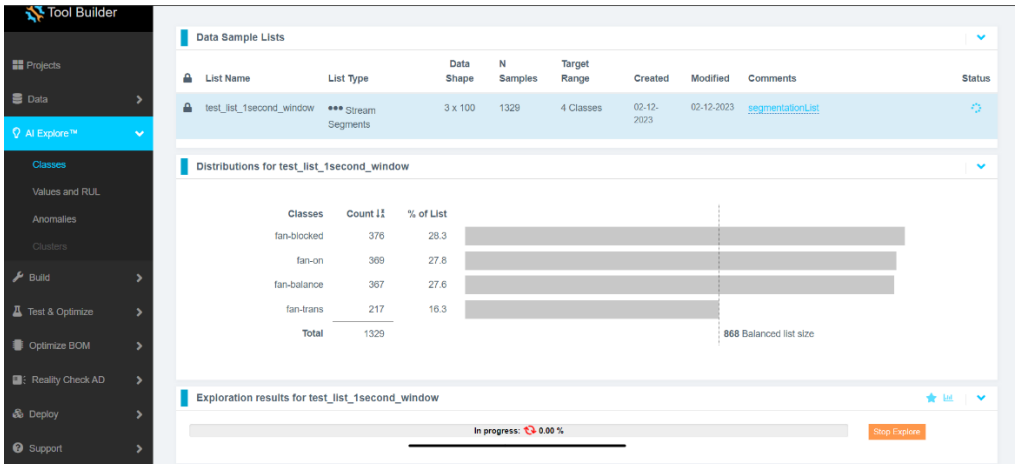
Overview

In this section, we will add create AI models on the uploaded dataset and look at various statistics for those models.

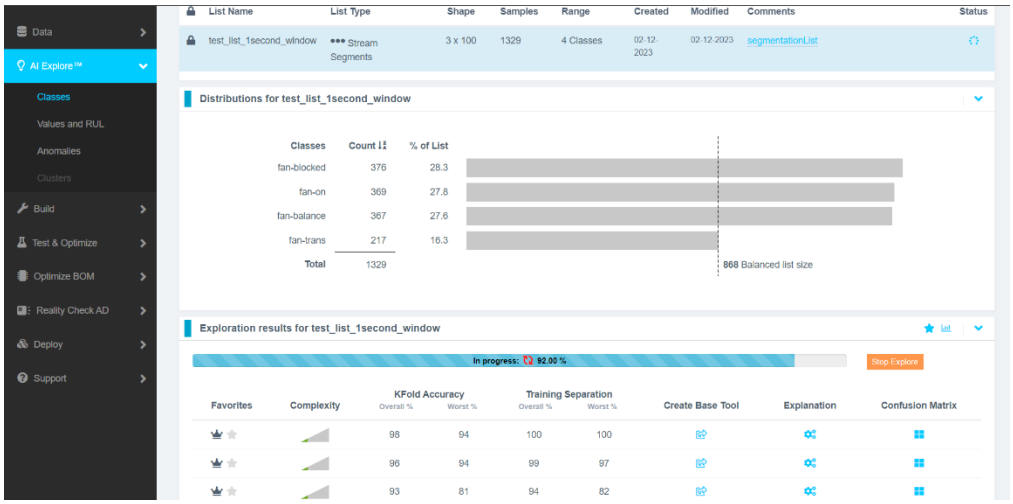
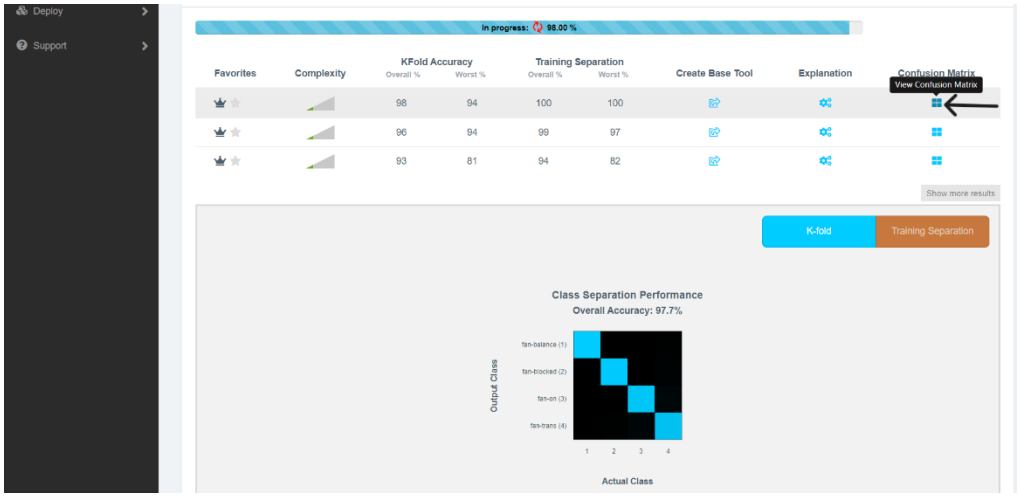
Procedural Steps

3.1	To create our initial models, on the left menu, click on **AI Explore** and then **Classes**. 
	Reality AI Tools has options for creating 3 types of AI Models: • **Classes:** When we have labeled categorical data (**we will use this option in the tutorial**). You might have noticed that we uploaded perfectly labeled data in section 1. This is needed for classification models as we would be performing supervised learning. • **Values:** When we have descrete int or float values instead of categories. (Example: What is the exact temperature of a machine? Or What is the exact value of a tire pressure of a car?). This is also supervised learning. • **Anomalies:** This is Anomaly Detection module. It's a semi-supervised model where you only need examples of Normal data to create a baseline model.
3.2	Click on the **segmented list** once you are on Classes page. 

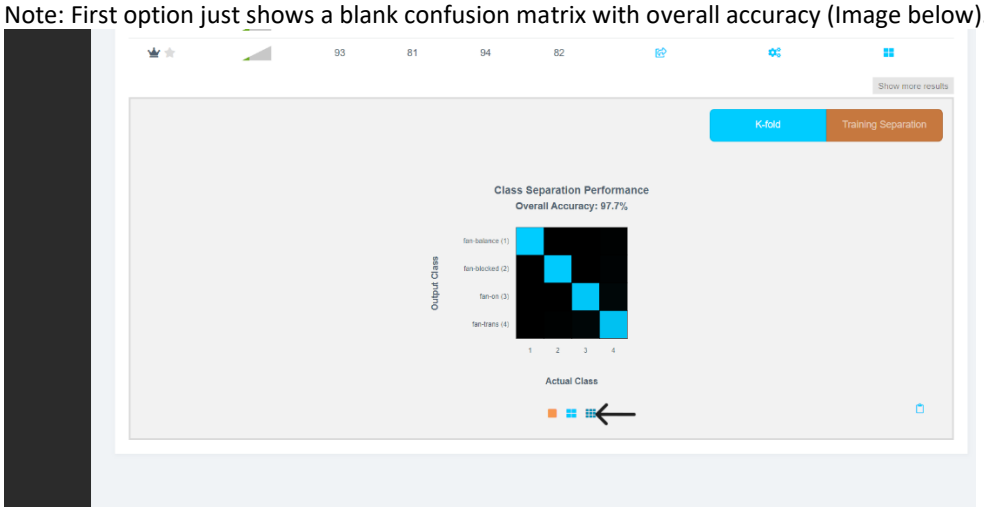
© 2024 Renesas Electronics. All rights reserved. Download and disclosure restricted by Reality AI Tools® Terms & Conditions.

3.3	After clicking on the list, click on **Start Exploring** to generate models. 
3.4	You should be able to see progress of your model generation process. 
	What is happening in the background: Clicking on Start exploring button will turn the Reality AI algorithm loose on your data. It will create a number of optimized feature sets and machine learning models that best fit the classification problem, and show you the ones that perform best. **What is the algorithm doing**: AI Explore first creates a **balanced subset** of the sample list (meaning a list in which each class has equal numbers). If the sample list is very large, it may also sub-sample for a shorter processing time. Explore then runs through a procedure in which the Reality AI algorithms attempt to discover the best possible set of features and machine learning parameters for separating the different training classes represented in the data. The feature sets that are most promising are then used to construct machine learning models, which are trained on the sub-sample and put thru a Kfold validation. Only the best performing results are displayed. Several hundred different feature sets and machine learning models are compared in a typical AI Explore run. **What is K-Fold validation?** K-Fold is where a given data set is split into a K number of sections/folds where each fold is used as a testing set at some point. Lets take the scenario of 10-Fold cross validation(K=10). Here, the data set is split into 10 folds. In the first iteration, the first fold is used to test the model and the rest are used to train the model. In the second iteration, 2nd fold is used as the testing set while the rest serve as the training set. This process is repeated until each fold of the 5 folds have been used as the testing set. **We use K=10 in the AI Explore page.**

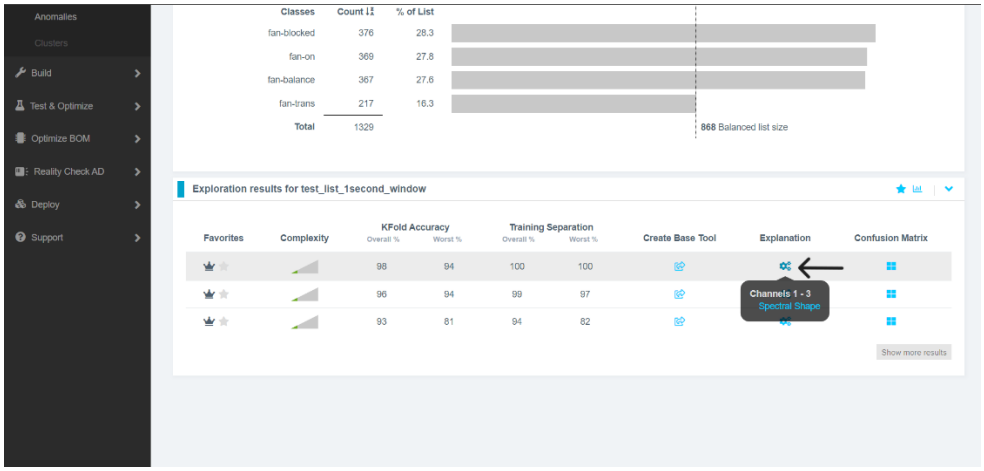
© 2024 Renesas Electronics. All rights reserved. Download and disclosure restricted by Reality AI Tools® Terms & Conditions.

3.5	As the system starts going through feature-spaces, you should be able to see pregress and top performing models listed on the page. 
3.6	In the Exploration results section, each row represents a different model based on different features identified during the AI Explore run. To look at detailed accuracy statistics, you can click on the **Confusion Matrix** option. 

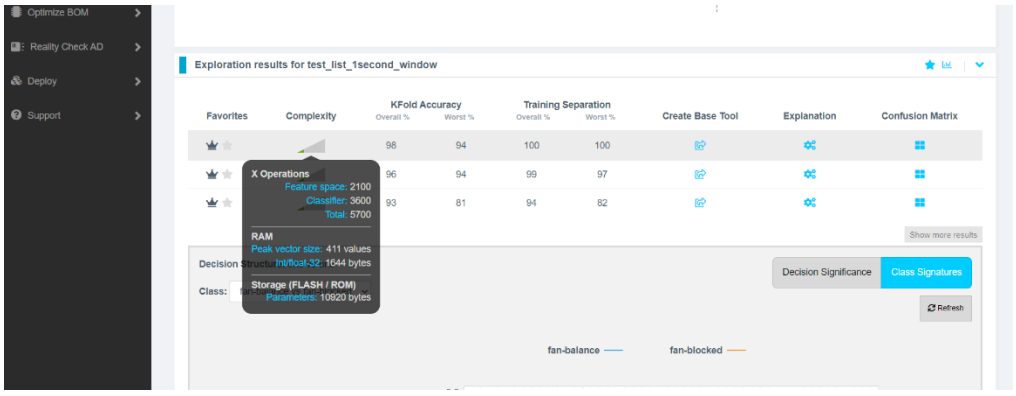
© 2024 Renesas Electronics. All rights reserved. Download and disclosure restricted by Reality AI Tools® Terms & Conditions.

3.7	Once a blank Confusion Matrix pop-up, click on the **third option** at the bottom to expand on detailed stats. Note: First option just shows a blank confusion matrix with overall accuracy (Image below).  Second option shows accuracy for each class/ category. But it won't show you errors percentage/ confusions and detailed precision and recall numbers. Use second option to get class accuracy summary only.
3.8	Below is the Confusion Matrix (Actual vs Predicted) for the first model in the list. 

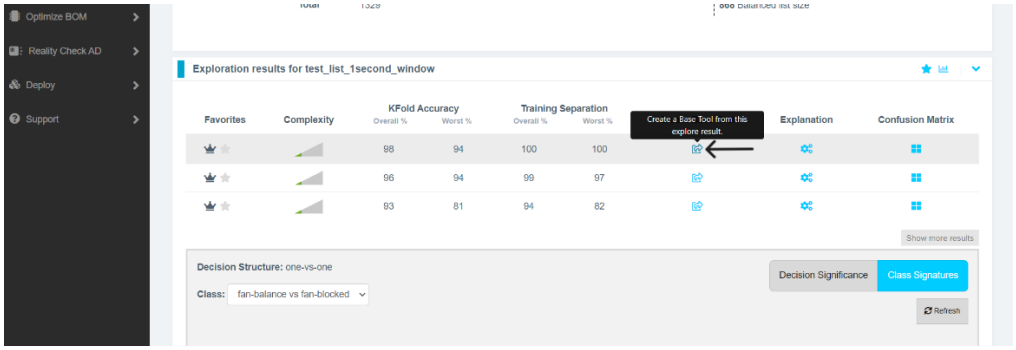
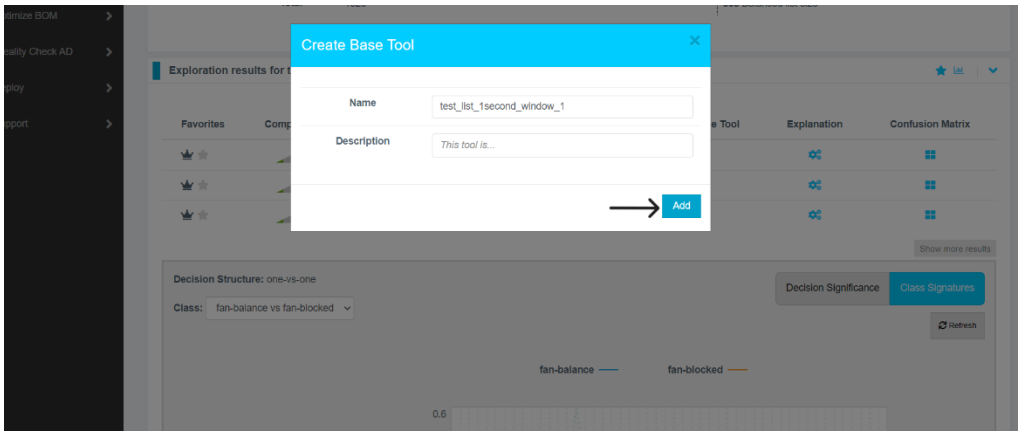
© 2024 Renesas Electronics. All rights reserved. Download and disclosure restricted by Reality AI Tools® Terms & Conditions.

	How to read a Confusion Matrix: Rows are the output class – what the trained classifier predicted a sample would be**. Columns are the actual class** – how the sample was labeled in the source data.Diagonals are true positives and true negatives shaded in blue. A **true positive** is an outcome where the model correctly predicts the positive class. Similarly, a **true negative** is an outcome where the model correctly predicts the negative class. The bottom row shows "class marginals" – accuracy (**green**) and error (**red**) rates specific to each class. The lower right corner shows overall accuracy in green, and overall error rate in red. **White cells off the diagonal indicate false-positives and false-negatives**. In a multi-class model, these cells indicate specific class confusions. Note: Numbers on the X-axis correlate to the numbers assigned to each class on the Y-axis. For example: **fan-balance (1)** on Y-axis is the same label for **1** on the X-axis. Since we are using a balanced subset of the list (histogram displayed in section 3.4 & 3.6), each number in the cell belongs to a segment/ block of data from the histogram. **Note A good confusion matrix will have least possible numbers in the white cells (confusions or wrong predictions).**
3.9	To know which feature space was selected, click on the option in the **Explaination** tab. You should be able to see the **name of the feature space (Spectral Shape in this case) pop-up**. Further clicking on the button will give detailed plots on that feature space. 

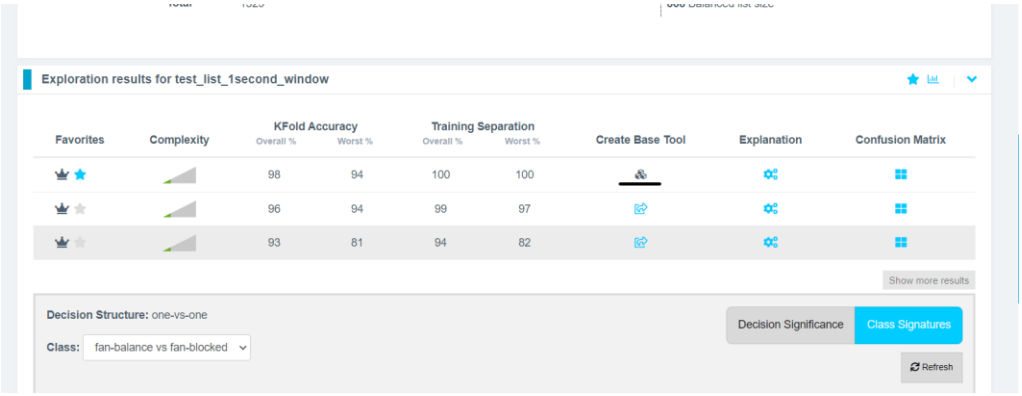
© 2024 Renesas Electronics. All rights reserved. Download and disclosure restricted by Reality AI Tools® Terms & Conditions.

	Below is an example plot for Spectral Shape feature space (click on the option highlighted in section 3.9). The bottom plot represents Decision Significance of each frequency band/ which frequency band is important (and how much) in making a distinction between class fan-balance and fan-blocked. The higher the square (representing each frequency band), higher its importance. If it's a shade of light blue and lower on Y-axis → less important. If its dark blue and middle of Y-axis → moderate importance. If it's a shade of orange and high in Y-axis → high importance. **Importance**: Imagine a scale of 0-1. Low importance would assign a score closer to 0. Moderate importance would assign a score closer to 0.5 and high importance would assign a score closer to 1 to each frequency band. Top plot shows mean of the signal in solid lines and standard deviation of signal in dotted lines. If a frequency band is high on the decision significance plot, it is contributing the most in differentiating between fan-balance vs fan-blocked. 
3.10	You should also be able to see ballpark number for resource consumption for this model. Just **hover the mouse around the complexity ta**b to view the numbers. X operations indicate the multiplication operations for the model It is further broken down into Feature Space (indicating the multiplication operations for the feature calculation), Classifier (indicating the multiplication operation needed for classifier model to runa prediction) and Total. 

© 2024 Renesas Electronics. All rights reserved. Download and disclosure restricted by Reality AI Tools® Terms & Conditions.

	How should I interpret the complexity numbers? Complexity numbers can help you guage the resource consumption of the model on the MCU/ MPU's. These number are not final as the model can still be optimized. For example: In the previous section 3.9, we see that most of the important frequency bands reside in range < 50 Hz on the decision significance plot. Knowing this, we can use a filter to only use frequencies < 50 Hz in feature computation and save resources (multiplication operations). Also note: If you preselected the target processor on the project creation stage, then the top models are going to be the ones that can fit on that processor → meaning the resource consumption of the models would fall within range of available resources for that target processor.
3.11	Finally, after you have looked at all the statistics, click on the highlighted option to deploy the model for further tesing and export. 
3.12	Give exported model a name and click **Add**. 

© 2024 Renesas Electronics. All rights reserved. Download and disclosure restricted by Reality AI Tools® Terms & Conditions.

3.13	The icon for the deployed model should change (highlighted) to indicate a succesful deployement. If you hover on the new icon, you should see the given name from section 3.14. Anything else means the deployement was unsuccessful → contact customer support to report the issue. 
3.14	**You have completed this section.**

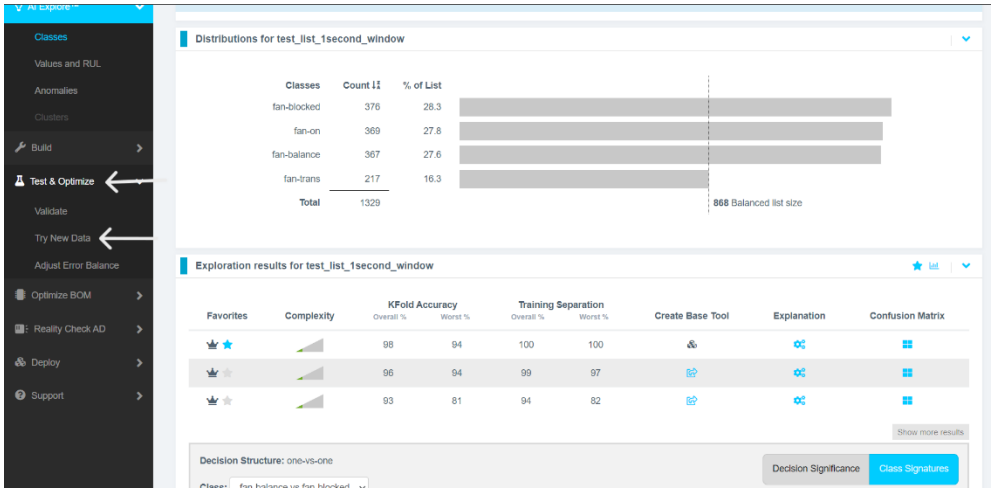
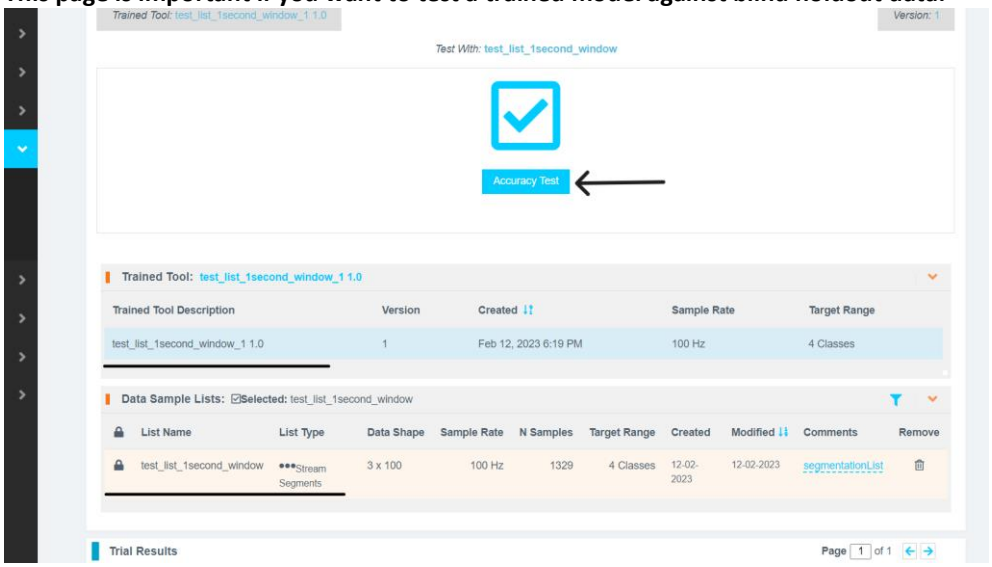
© 2024 Renesas Electronics. All rights reserved. Download and disclosure restricted by Reality AI Tools® Terms & Conditions.

4 Testing the Model

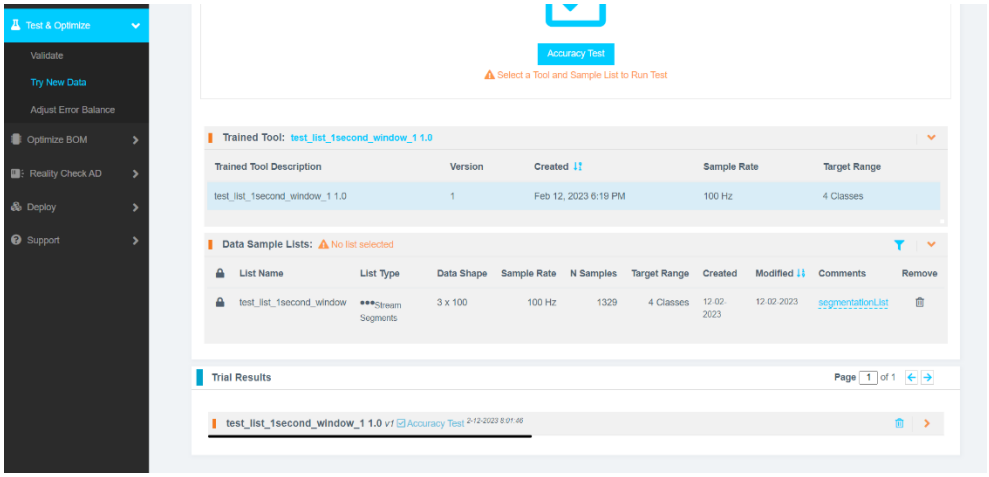
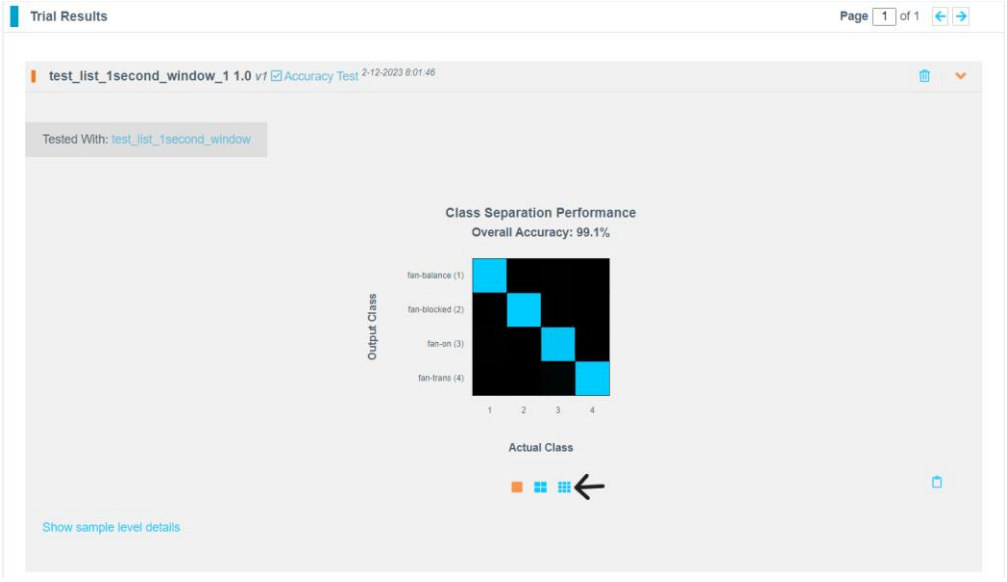
Overview

In this section, we will go over how to test a model against some data and generate + export the results.

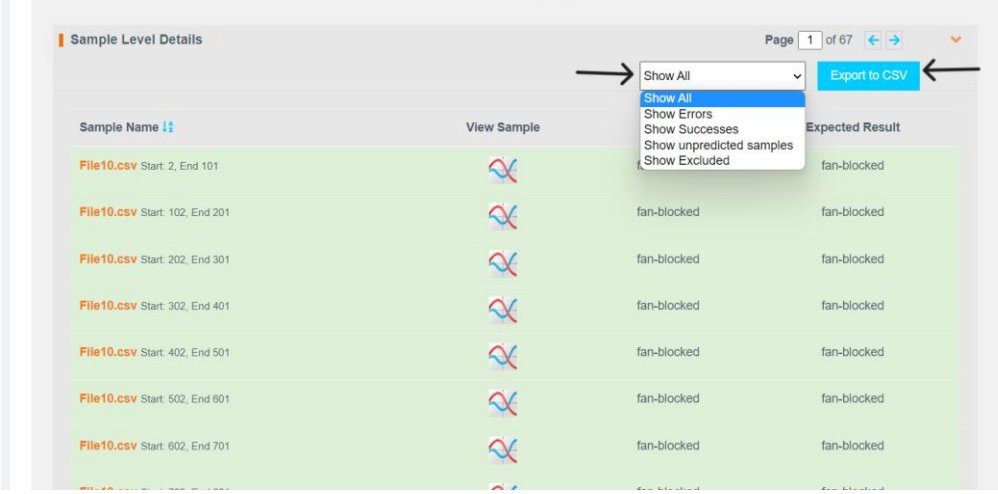
Procedural Steps

4.1	Using the menu on the left, navigate to **Test & Optimize** and **Try New Data**. This section will be used to test the models. 
4.2	**Select the model (or Trained Tool → As shown in the image below)and the Data Sample List** you want to test against (**underlined**). Then click on **Accuracy Test** button to start a test job. Note: If you want to test the model against any blind dataset, you would need to upload those data files → Segment them (follow steps in Section 1 & 2) and navigate to this section to run a test. For this tutorial, we will just test the model against the list we used to train it (for simplicity). **This page is important if you want to test a trained model against blind holdout data.** 

© 2024 Renesas Electronics. All rights reserved. Download and disclosure restricted by Reality AI Tools® Terms & Conditions.

4.3	Once an accuracy test job is triggered, wait for 2-3 minutes and refresh the page. After refreshing the page, click on the Trial Results tab (underlined in the screenshot below). Note: The duration for the test job execution could vary based on the size of the dataset. For example, a dataset of 1 GB could take ~10 min to finish a test job. 
4.4	Click on the third option below Confusion Matrix to expand for more statistics. Check section 3.8 on how to read a confusion matrix. 

© 2024 Renesas Electronics. All rights reserved. Download and disclosure restricted by Reality AI Tools® Terms & Conditions.

4.5	You should now be able to statistics in similar format as ***AI Explore*** page. Note: A Confusion Matrix is produced because the test data is labelled similar to training data. However, if no labelled data is provided for this test job, no confusion matrix will be produced. You can still export the raw results in a csv file (next step). **To export results in a CSV file, click on Show Sample level Details** (underlined botton left). 
4.6	You can view just the errors or just the correct results in the GUI. To export results/ predictions in a CSV file, click on **Export to CSV** button. **Results**: What Reality AI Algorithm predicted. **Expected Results**: Labels provided and assigned as part of the data upload/ ground truths. 
	Note: If you want to look at the waveform of each block of data/ segment from the file, you can click on the waveform in **View Sample** tab. This is useful if you want to compare the waveform/ plot of successful vs unsuccessful predictions.
4.7	**You have completed this section.**

© 2024 Renesas Electronics. All rights reserved. Download and disclosure restricted by Reality AI Tools® Terms & Conditions.

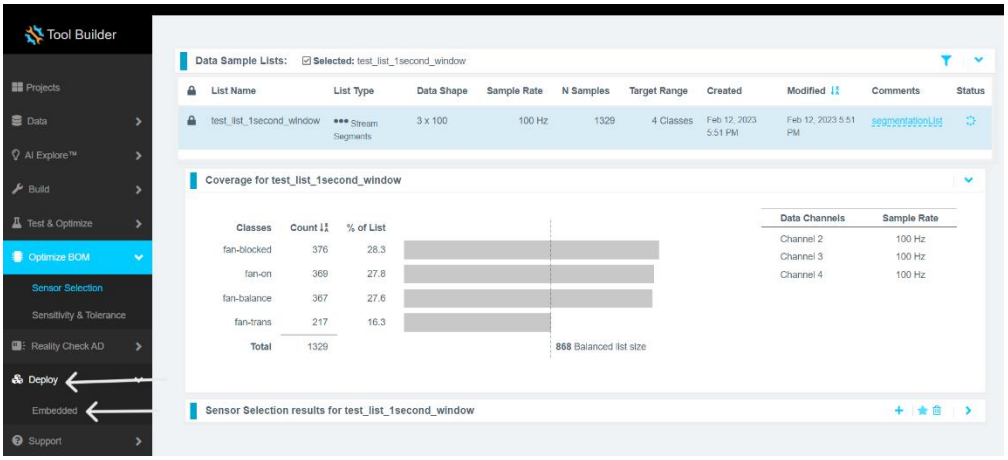
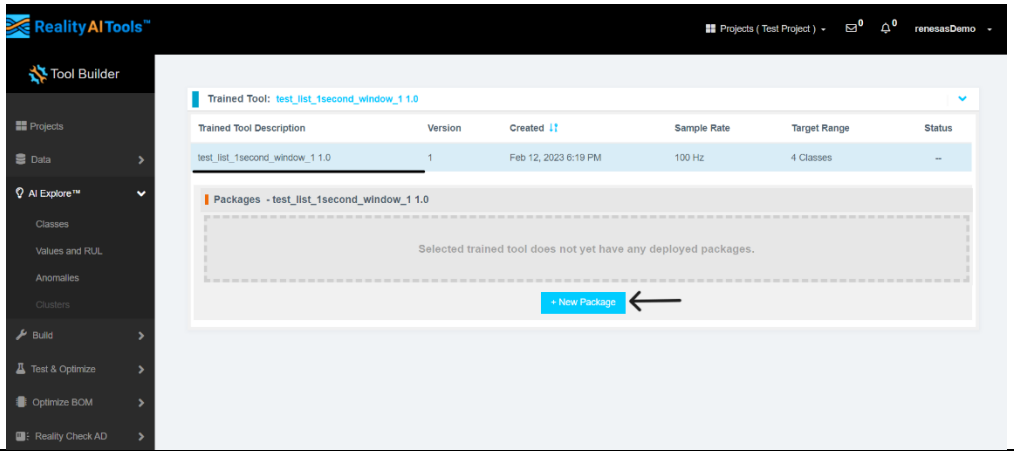
© 2024 Renesas Electronics. All rights reserved. Download and disclosure restricted by Reality AI Tools® Terms & Conditions.

5 Exporting the Model

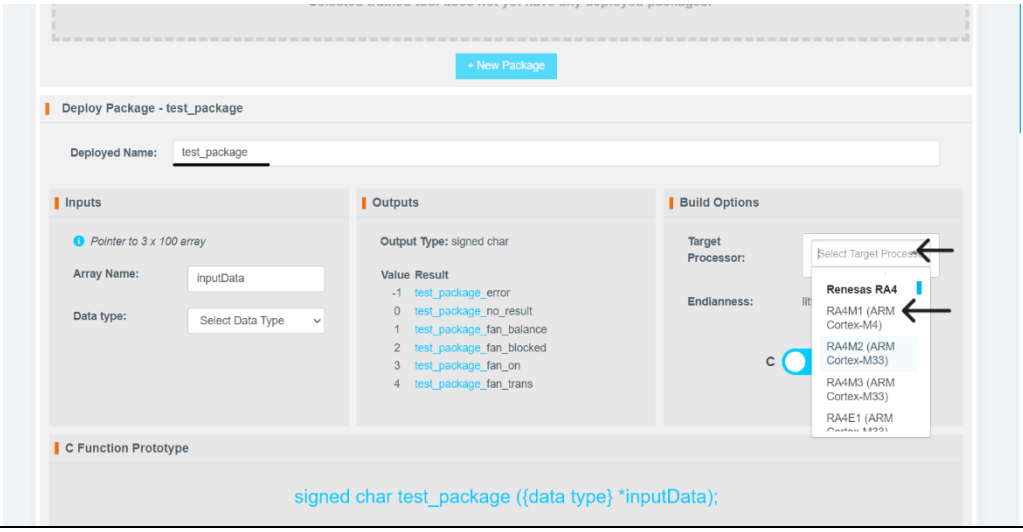
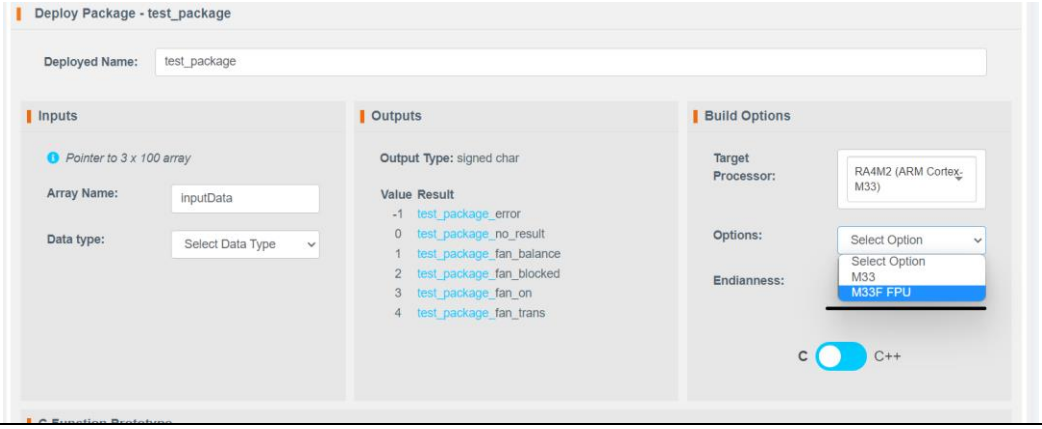
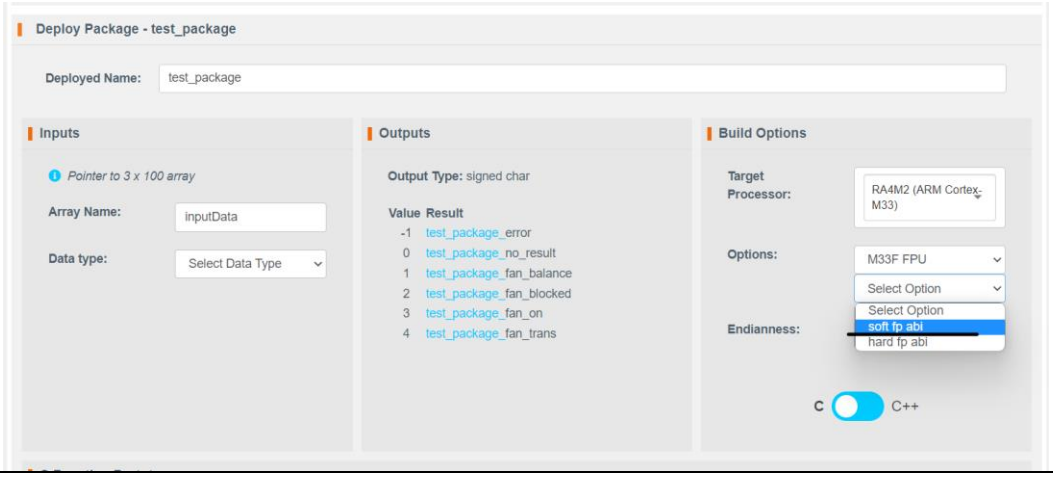
Overview

In this section, we will export the created model to be deployed to a target embedded board.

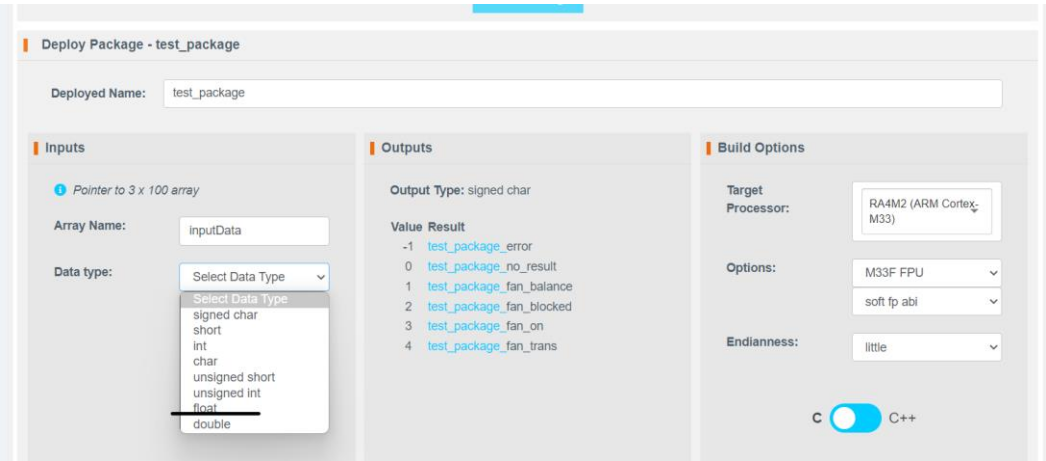
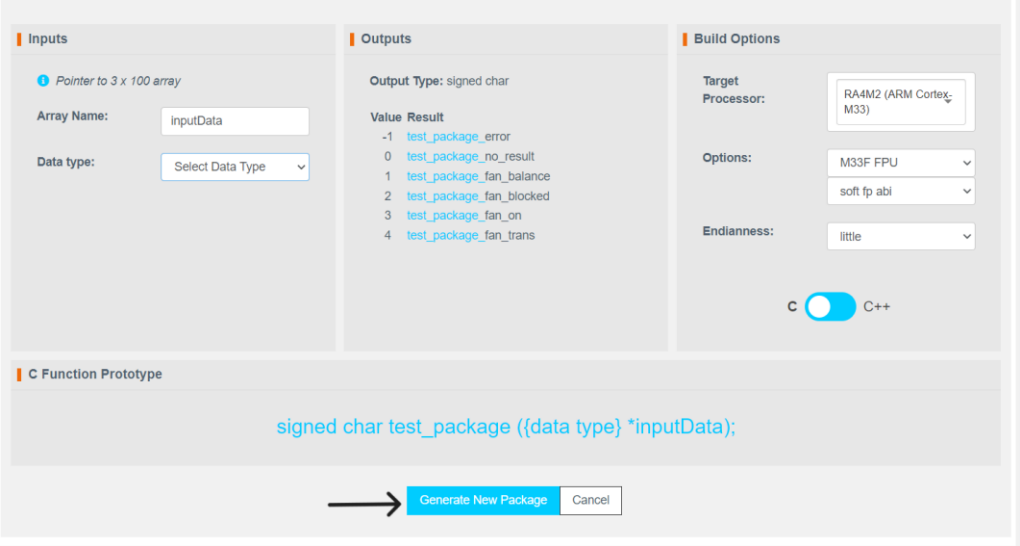
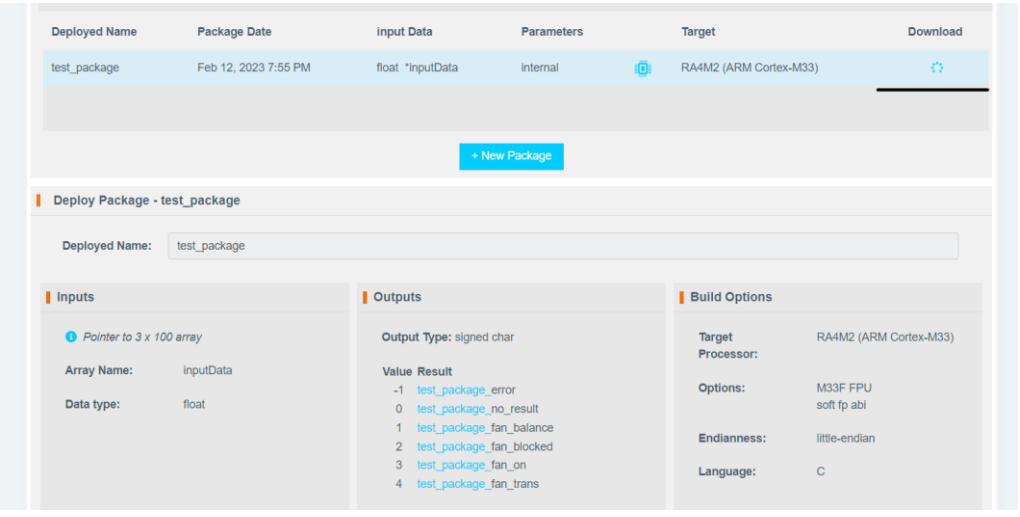
Procedural Steps

5.1	We will first navigate to **Deploy** and **Embedded** using the menu on the left. 
5.2	Click on the model you have already exported (from Section 3) and click on **+New Package**. 

© 2024 Renesas Electronics. All rights reserved. Download and disclosure restricted by Reality AI Tools® Terms & Conditions.

5.3	Give the exported package a name and select a processor from the list of available Renesas processors. 
5.4	Make additional selections for FPU vs non-FPU variant. 
5.5	Select soft vs hard fp abi. 

© 2024 Renesas Electronics. All rights reserved. Download and disclosure restricted by Reality AI Tools® Terms & Conditions.

5.6	Select target data type (we will use float for this tutorial). 
5.7	Click on **Generate New Package**. 
5.8	Wait for 10-15 min for the processing to finish. Once its done (the underlined buffering should turn to a download option), you should be able to download a zip file of the model deployement package. 

© 2024 Renesas Electronics. All rights reserved. Download and disclosure restricted by Reality AI Tools® Terms & Conditions.

5.9	Below are the contents of the downloaded zip file: 
5.10	To check the function call of the model, open **examplemain.c**. **size_report.txt** will give you exact resource consumption numbers for the processor selected during the export stage. Integrate all these files into your project on the IDE to integrate the model with your project.

© 2024 Renesas Electronics. All rights reserved. Download and disclosure restricted by Reality AI Tools® Terms & Conditions.

5.11

Example code loop **(open: examplemain.c to view**)that needs to be integrated withing your IDE project. The model is essentially just a function call on-loop → running predictions on every window of data.

Note: The model will take the data in the same size window length/ data points as we specified in Section 2.5 (1 second in this case or 100 data points).

```
int main() {
  /* Init Your board and hardware here. */

  /* Example code loop
   *  Enter an infinite loop, simply generates a random vector of input and then
   * attempt to classify it, printing the result. Replace this with your actual
   * data read and windowing process.
   */

  while (1) {
    /* Fetch next data input window from your device  */
    /* Sensor data is simulated in this loop. Multi-channel windows
        are constructed for single event classification. */
    /* Expected format of sensor data stream: ch1.1,ch2.1,ch1.2,ch2.2,... */
    /* REPLACE THIS BLOCK WITH INPUT FROM YOUR DEVICE  */
    for (int sample = 0; sample < test_package2_SAMPLES_PER_CHANNEL; sample++) {
      for (int channel = 0; channel < test_package2_NUM_CHANNELS; channel++) {
        float r = (rand() % 64000) - 32000;
        input_vector[channel * test_package2_SAMPLES_PER_CHANNEL + sample] =
            (float)r;
      }
    }

    /* Classify the input data */
    test_package2_AICLASS AIC =
        (test_package2_AICLASS)test_package2_classifier_prediction(
            input_vector);

    /* Act on the output using a switch statement againt the enumerated, named
     * classes. */

    switch (AIC) {
    case (test_package2_fan_balance):
      printf("Detected class: test_package2_fan_balance\n");
      break;
```


© 2024 Renesas Electronics. All rights reserved. Download and disclosure restricted by Reality AI Tools® Terms & Conditions.

```
      case (test_package2_fan_blocked):
        printf("Detected class: test_package2_fan_blocked\n");
        break;
      case (test_package2_fan_on):
        printf("Detected class: test_package2_fan_on\n");
        break;
      case (test_package2_fan_trans):
        printf("Detected class: test_package2_fan_trans\n");
        break;
      case (test_package2_0):
        printf("Detected class: test_package2_0\n");
        break;
      }

      // delay(1000);
    }
    return 0;
}
```

5.12	**You have completed this section.**

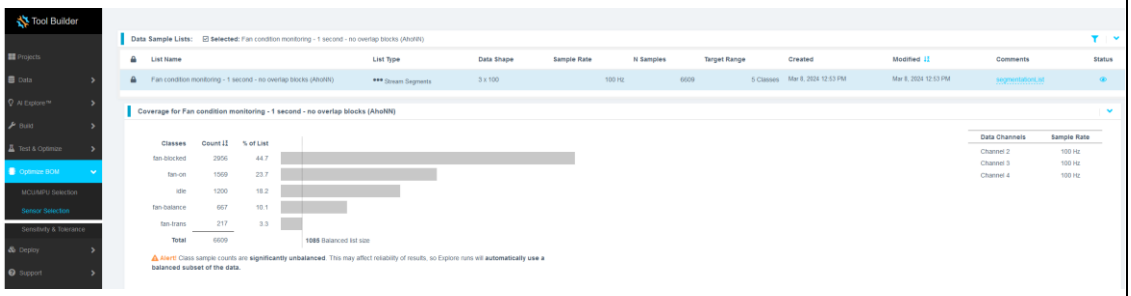
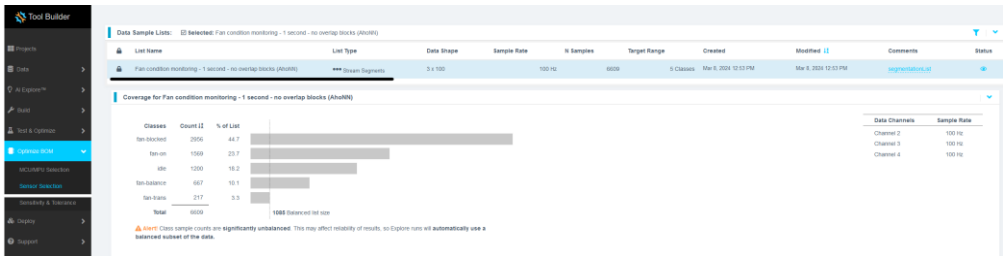
© 2024 Renesas Electronics. All rights reserved. Download and disclosure restricted by Reality AI Tools® Terms & Conditions.

6 BoM Optimization

Overview

In this section, we will explore the Bill of Material optimization section. Navigate to **Optimize BoM** menu on the left. There will be two major parts of it, **Sensor Selection** & **Sensor Sensitivity** (MPU/ MCU Selection is in Beta right now).

Procedural Steps

6.1	We will first navigate to **Optimize BoM** and **Sensor Selection** using the menu on the left. 
6.2	Select the Segmented List you have already worked on in the previous sections (highlighted in black line).  Note: If your account is part of Freemium/ Explorer Tier, then you will see a sensor selection job already run on the account. There will be an eye symbol indicating this. Move to next step to view the results. For normal accounts, after selecting the list, click on Start Sensor Selection and select the number of channels you want to analyze.
	Sensor Selection: This utility is useful to analyse how many of the existing channels are actually needed to solve the problem. For real world usecases, you can have questions like -->* *Do I really need 3 sensors to solve this problem?* *Do I need all 3 channels of accelerometer to solve this problem?* *Which location of the sensor is best to solve the problem?* Sensor Selection module answers these questions by automatically running Explore jobs on multiple permutation & combinations of the channel configurations.
6.3	Once the analysis is finished, you will see the results:

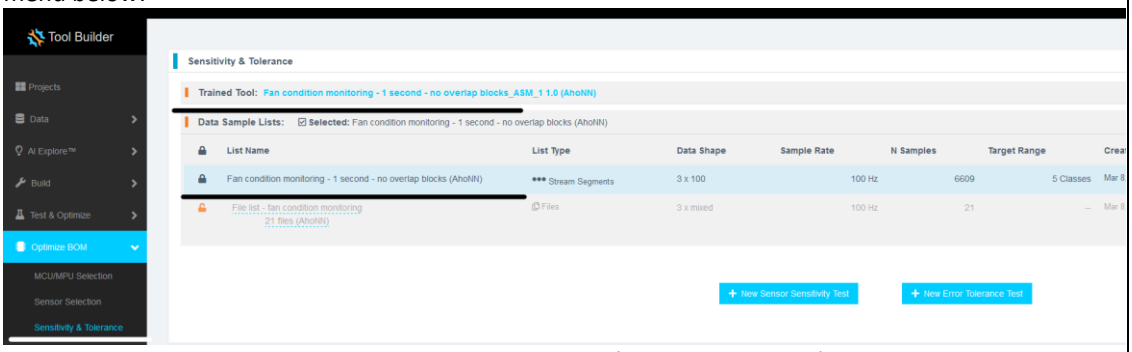
© 2024 Renesas Electronics. All rights reserved. Download and disclosure restricted by Reality AI Tools® Terms & Conditions.

Sensor Selection results for Fan condition monitoring - 1 second - no overlap blocks (AhoNN)

Favorites	Sensor Channels	Overall Accuracy (kfold)	fan-balance		fan-blocked		fan-on		fan-trans		idle	
			Accuracy %	Precision %	Accuracy %	Precision %	Accuracy %	Precision %	Accuracy %	Precision %	Accuracy %	Precision %
★ ☆	Channel 2 Channel 3 Channel 4	96	100	100	90	97	94	87	95	97	100	100
★ ☆	Channel 3 Channel 4	94	100	100	89	96	92	85	91	94	99	97
★ ☆	Channel 2	91	99	98	81	87	85	87	89	84	100	97
☆	Channel 2 Channel 3	95	100	100	89	97	89	85	95	92	100	100
☆	Channel 2 Channel 4	95	100	99	92	90	88	89	94	95	100	100
☆	Channel 4	92	100	100	85	90	87	87	90	87	100	98
☆	Channel 3	88	100	100	86	95	84	84	72	79	95	81

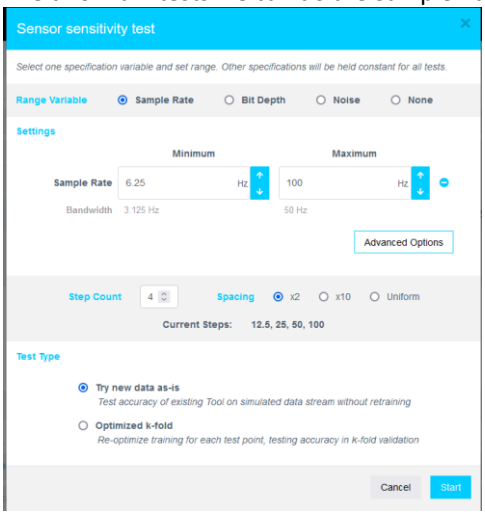
If you like a result, you can go reformat the data using Data --> Curate --> Select files --> Action (in source tab) --> Format Selected --> and re-format the data by only including the channels/ sensors that gives the optimized performance.
You can then run an AI Explore on it & export the model (same steps as previous sections).

6.4 Now Lets navigate to Sensor Sensitivity module of Optimize BoM. This is the second option in the menu below:



Then select the list & model you want to experiment on (show by black line).
Click on **+New Sensor Sensitivity Test**

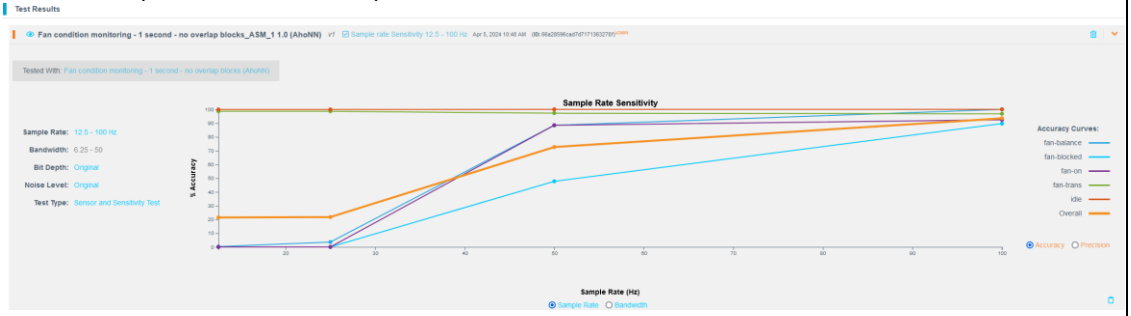
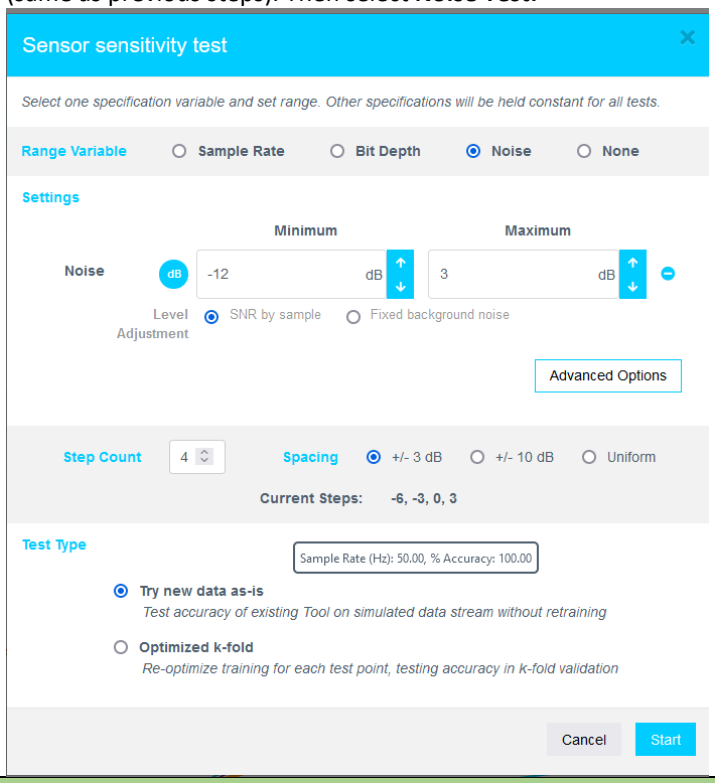
6.5 The two main tests we can do are Sample Rate & Noise. Lets do Sample Rate first:



Here you can define the range of sample rates and the step count. You can keep settings as default and click **Start.**

Sensor Sensitivity:
Here our goal is to see how low we can go in sample rate and still get good results. Remember, lower the sample rate, lower resources the model will consume. On Reality AI Tools, we automatically downsample the sampling rate to specified vaues (specirfied in **Step Count** in the previous step). The we recalculate the model parameters and generate accuracy statistics.

© 2024 Renesas Electronics. All rights reserved. Download and disclosure restricted by Reality AI Tools® Terms & Conditions.

6.6	Below is the plot of a finished sample rate test:  Here, X-axis is Sample Rate and Y-axis is accuracy. Lines in different colors represent different classes. This plot will give the user an idea about how low we can go in sampling rate and still get decent accuracy across the board.
6.7	Next, lets look at Noise test. Click on **+New Sensor Sensitivity Test** after selecting the model and list (same as previous steps). Then select **Noise Test.** 
	Noise Test: Noise test is useful if a user wants to check how susecptible to noise their current setup is. Maybe the current sensor is not good for noisy conditions? On Reality AI Tools, we inject your data with artifically generated white gaussian noise at different ranges. User can specify mimimum and maximum of these noise ranges and start the test.
6.8	One the test finishes, you will get a plot like this:  Here, X-axis is scaled noise. 0 on X-axis represents equal ratio of signal and noise. Moving right --> portion of signal increases in the ratio. Moving left of 0 --> portion of signal increases in the ratio.

© 2024 Renesas Electronics. All rights reserved. Download and disclosure restricted by Reality AI Tools® Terms & Conditions.

	Y-axis represents accuracy at each SNR (Signal to Noise Ratio) level. That's why you see for certain classes the accuracy goes down when moving from right to left on the plot. Now, remember it's natural for certain conditions and classes to go down in accuracy. What you are trying to find out is how much? The more susceptible a class is to noise, the more you would need to pay attention to it for future testing. Based on the conclusions of Sample rate and Noise test, you can guess the changes needed to be made in any data collection setup for future improvements.
6.9	**You have completed this section.**

END OF LAB

THANK YOU

© 2024 Renesas Electronics. All rights reserved. Download and disclosure restricted by Reality AI Tools® Terms & Conditions.