**Missing Data Walkthrough**

Notebook Guide

This guide is intended to be supplemental to the “Missing Data Walkthrough” Jupyter Notebook.

**Step 1:**

**Download the (3) files** needed for following along with the guide and ensure that they are saved in the same directory on your local machine:

1. “Missing Data Walkthrough.ipynb” – the Jupyter Notebook
2. raw\_titanic\_data.csv – the complete Titanic dataset (i.e., without missing values)
3. missing\_titanic\_data.csv – the incomplete Titanic dataset (i.e., with missing values)

**Step 2:**

Open the “Missing Data Walkthrough.ipynb” and **review the dependencies** listed in the first cell. You’ll need to ensure that you **have the full list of packages installed** on your machine before proceeding. All packages should be candidates for a simple – “pip install” – meaning that there are no major hurdles to cross (other than ensuring each package is installed; and being mindful of versions)

**Step 3:**

Enter your local working directory (i.e., the folder location of the files you downloaded and saved in Step 1 above) in **1.1 - Set Your Working Directory** (image below; simply type in your folder location using the “text box” preceded by the “Directory:” label)

A screenshot of a cell phone

Description automatically generated

**Step 4:**

If you’ve correctly entered in your directory in Step 3 and have saved the (3) input files to the directory entered, you should be able to run each cell with the default inputs until you reach the header labeled **1.3 - Select a Data Frame.** The first input box will be preselected for you, but you’ll need to select which variables to include from the selected dataframe in the next cell (image below). Highlight all variables **excluding the “name” variable** at this time.

A screenshot of a cell phone

Description automatically generated

**Step 5:**

Continue running each subsequent cell using the default inputs until you reach the cell with the header **2.1 - Select Your Target Variable** where you’ll need to make (2) selections:

1. Choose a “Target” variable from the list of columns available – **choose the “survived” variable**
2. Choose a “Target-type” from the list (either Continuous or Categorical) – **choose the “Categorical” label**

**Step 6:**

Initiate H2O by running the cell just below those shown in Step 5. If you’ve successfully launched h2o, you should see the following output in your notebook:



**Step 7:**

Run the subsequent cells using the default settings until you reach the header **2.3 - Configure Models**, where you’ll see a wide range of interactive inputs. Note: when running this walkthrough the first time, you can simply change the “Project:” label and use all default settings (ex., if you’d like to return to this later, you can play with the “run time” and even select more algorithms to include). Image below:

A screenshot of a cell phone

Description automatically generated

A screenshot of a cell phone

Description automatically generated

**Step 8:**

Run the next several cells to start the “autoML” process and to generate your “leaderboard” (which is the table that stores the evaluation results for each of your trained models). When you get to the cell shown below, choose the input: “varimp\_plot” label from the list of “methods” to generate the variable importance plot:

A screenshot of a social media post

Description automatically generated

**Step 9:**

Run the next several cells to apply your scored model to the full dataset before moving on to the cell with the header **3.1 - Upload Your Data (Excel and CSV files)**, which will look very similar to the steps you followed at the beginning of the guide. In this sequence, we’ll be loading in the complete Titanic dataset (with no missing values).



**Step 10:**

Once you’ve ensured the “File Name:” from Step 9 has loaded, you’ll run the cells shown in the image below:

A screenshot of a cell phone

Description automatically generated

**Step 11:**

At this point in the guide, you’re ready to switch from the Jupyter Notebook to h2o.ai’s Flow browser-based tool. You should be able to click:

<http://127.0.0.1:54321>

Which will bring you to a browser that should look like this:

A screenshot of a social media post

Description automatically generated

**Step 12:**

From the toolbar at the top of the page, click the down caret next to “Data” and select “List All Frames”

A screenshot of a cell phone

Description automatically generated

**Step 13:**

Look for a “Frame” in the list that appears that resembles the one shown below. This is the “leaderboard” and it contains the full list of trained models from your Jupyter Notebook. Click the “Inspect” button just below its name.

A screenshot of a cell phone

Description automatically generated

The “Inspect” button above reveals the following table:

**A screenshot of a cell phone

Description automatically generated**

**Step 14:**

Click on the “model\_id” label (blue-text) which is shown in the “COLUMN SUMMARIES” table to reveal your ranked leaderboard (and the model\_ids):

A screenshot of a cell phone

Description automatically generated

**Step 15:**

Now that you have your ranked list of models. Find the first one listed (ex., “GBM\_grid\_1\_AutoML……\_model\_19”) and take note of its name. From the toolbar at the top of the page, click the down caret next to “Model” and select “List All Models”

A screenshot of a cell phone

Description automatically generated

**Step 16:**

Find the model from Step 15 in the list that appears and click on its name (blue-text):



Which will reveal the Model summary (and the trained model’s parameters – these are the inputs you’ll use to train another model on the complete Titanic dataset):

A screenshot of a social media post

Description automatically generated

**Step 17:**

Now that you have the parameters for the model trained on the Titanic dataset with missing values, you need to bring in the complete Titanic dataset (i.e., the one without any missing values). Go back to the top of your Flow (or click the down caret next to “Files” and select “List All Frames”) and find the following data frame:



Click on the “Inspect” button below the frame (as shown above)

**Step 18:**

In the table that appears, find the row containing the label “survived” and click on the “Convert to enum” blue-text shown in the “Actions” column for that row

A screenshot of a cell phone

Description automatically generated

Once the “Summary” cell displays (after taking the action above), return to the view as shown in the image above and click on “Build Model…” just beneath the “non\_missing\_training\_data” title

A screenshot of a social media post

Description automatically generated

**Step 19:**

Once the “Build a Model” section appears, ensure that your inputs match the ones shown in the image below (exactly):

A screenshot of a cell phone

Description automatically generated

**Step 20:**

Refer back to the list of parameters from Step 16 and make the necessary adjustments to the remaining “Build a Model” parameters (in the example below, ntrees was changed from the default value of 50 to 10,000):



When you’re ready, click on the “Build Model” button as shown below:



**Step 21:**

Once your “Job” has finished (meaning the model has been built), click on the “View” button:

A close up of a logo

Description automatically generated

**Step 22:**

You can now review the model’s performance on the complete dataset. In the images below, you’ll first see the AUC (ROC Curve) for the complete dataset’s Training set and Testing set (which returned an AUC of 0.902 and 0.891, respectively) followed by the “Variable Importances” (ranked by scaled importance). Compare these values to those found in the summary of the model trained on the incomplete dataset (which can be observed in the last (3) images below; returning an AUC of 0.921 and 0.870 on the training and testing sets, with a slight variance to the order of important variables).

A screenshot of a cell phone

Description automatically generated

A close up of a map

Description automatically generated

A screenshot of a cell phone

Description automatically generated

A close up of a map

Description automatically generated

A close up of a map

Description automatically generated

A screenshot of a cell phone

Description automatically generated

**Final Comments:**

While the original model’s Training AUC of 0.921 (from the dataset with missing values) is greater than the Training AUC of 0.902 (from the dataset without missing values), the original model’s Test AUC of 0.870 is lower than the other Test AUC of 0.891. This result is anticipated, when using the complete dataset, the algorithm was able to train a model that generalized better across both datasets (i.e., the training and testing set) as shown by the narrower range of AUC values (0.902 and 0.891). When the model was trained on the dataset with missing values, the range between the AUCs was wider (0.921 and 0.870). Additionally, while the first (3) variables ranked in the “Variable Importance” plot remained constant between both models, the model trained on the complete dataset found that the fourth most important variable was “total\_passengers” while the model trained on the dataset with missing values listed the fourth variable as “passenger\_class.”

If you’d like to experiment further on the impact of missing values with this dataset, simply open up the “missing\_titanic\_data.csv” and randomly delete additional cell values from each of the columns. After doing so, simply rerun the sequence of steps listed in this guide and compare the model performances from each run.

Lastly, this guide provides a simple approach to building a baseline model without removing or imputing missing values. Knowing when to delete or impute data is often times more of an “art than a science” approach. When in doubt, start by building a baseline model – identify highly correlated variables, evaluate the important variables, and make iterative changes so you can compare your results against your baseline. Lather-rinse-repeat until you’ve plateaued or developed a “useful” model.