Homework 1: Aircraft Inventory Analysis

Written answers

1. Investigate the missing data in this dataset. Specifically, for each of the following variables that have missing data, decide if any imputation is possible. Give your reasoning and code if you decide to impute missing values.

   - Columns for investigation: CARRIER, CARRIER_NAME, MANUFACTURE_YEAR, NUMBER_OF_SEATS, CAPACITY_IN_POUNDS, and AIRLINE_ID.
   - For example, watch out for "North American Airlines" aircrafts. Are the CARRIER/UNIQUE_CARRIER column *really* missing?

After investigating the missing data in this dataset, I chose to use different imputation strategies for each of the columns that needed to be investigated. For CARRIER, I looked at a special case where “North American Airlines” had the carrier code “NA” leading to being misinterpreted as null values. I recognized that the CARRIER/UNIQUE_CARRIER columns were not *really* missing. To fix this, I found the rows where CARRIER_NAME contains “North American” and the CARRIER is missing and then manually filled in “NA” for these values in the CARRIER column. For the columns CARRIER_NAME and CARRIER, I constructed one-to-one mappings to fill in the missing values both ways (from CARRIER to CARRIER_NAME and vice versa). I imputed missing CARRIER_NAME using data from both UNIQUE_CARRIER_NAME and CARRIER columns. I used the same method of creating one-to-one mappings to impute AIRLINE_ID using data from the CARRIER column. For MANUFACTURE_YEAR, I assumed that the same models of aircrafts are manufactured during a close time range and imputed missing values by taking the median manufacture year for each aircraft MODEL. I used K-nearest neighbors imputation for imputing the numeric columns CAPACITY_IN_POUNDS and NUMBER_OF_SEATS since it uses the similarity between values to fill in missing values and would be more accurate than mean or median.

2. Inspect the columns MANUFACTURER, MODEL, AIRCRAFT_STATUS, and OPERATING_STATUS. Decide, for each column, if transformation or standardization of data are required. Give your reasoning and code if you decide to transform the data.
   Hints:

   - For very messy data like manufacturer/model names, give your best attempt. It is okay to not catch them all.
   - Use value_counts() to identify "big wins".
   - Break down into multiple steps, instead of having one line of code to do them all.

After inspecting the required columns, I found that there was a lot of mismatch in the names of the MANUFACTURER column. Uppercase and lowercase versions of the same MANUFACTURER were treated as distinct values. To fix this, I performed standardization – converted everything to uppercase and removed leading and trailing white spaces. For instance, I changed all the versions of Boeing to be “BOEING”. I also identified a list of the most common MANUFACTURER names and used vectorization in combination with nearest neighbors using cosine similarity to group them (cosine similarity threshold of 0.3). Something to note here is that this method may not be the most effective since the list of manufacturer names was not guaranteed to be exhaustive. Having said that, this method got rid of 30 manufacturer names by transforming them to their canonical form, improving the consistency of the data.

I chose not to transform the MODEL column because I did not want to oversimplify and wrongly group different models together. I cleaned the AIRCRAFT_STATUS column by converting everything to lowercase and once again removing leading and trailing white spaces. For the OPERATING_STATUS column, I converted everything to lowercase, removed leading and trailing whitespaces, and chose to map the "y" and "n" to "1" and "0" respectively for readability and clarity purposes. In general, I chose to standardize or transform data to remove inconsistencies and make the data more usable.

3. Remove data rows that still have missing values. Report the amount of remaining data you obtained.

   After removing the rows that still had missing values, I got the following result:

   Number of rows originally: 132313

   Remaining rows: 101275

   Rows removed: 31038

4. Transformation and derivative variables

   - For the columns NUMBER_OF_SEATS and CAPACITY_IN_POUNDS, check the skewness in the variable and plot a histogram for each variable.
   - The Box-Cox transformation (scipy.stats.boxcox) is one possible way to transform variables into a "more-normal-like" variable. Apply the Box-Cox transformation for these two columns and save them as new columns, i.e. XXXXXXXXX_BOXCOX.
   - Plot a histogram for each transformed variable.
   - Describe what you observe before and after transformation.

Both the columns NUMBER_OF_SEATS and CAPACITY_IN_POUNDS showed strong left skewness before transformation meaning that there were some really small values. CAPACITY_IN_POUNDS showed stronger left skewness before transformation.

After Box-Cox transformation, both NUMBER_OF_SEATS and CAPACITY_IN_POUNDS became more bell-shaped and symmetric, resembling normal distributions. Box-Cox transformation made decreased skewness by stabilizing the variance.

5. Feature engineering

   - Create a new column SIZE by the quartiles of NUMBER_OF_SEATS
     - below 25% percentile: SMALL
     - 25% - 50% percentile: MEDIUM
     - 50% - 75% percentile: LARGE
     - above 75% percentile: XLARGE
   - For each size group, provide and plot the proportions of aircrafts that are operating versus not (OPERATING_STATUS).
   - For each size group, provide and plot the proportions of aircrafts belonging to each aircraft status group (AIRCRAFT_STATUS).
   - Provide a written summary of your findings.

After plotting the proportions belonging to each size that are operating versus not OPERATING_STATUS, we found that the XLARGE, LARGE, and SMALL aircrafts had almost equal proportions of active operating status – higher than that for MEDIUM aircrafts. MEDIUM aircrafts had a higher proportion of inactive entries. This may imply that MEDIUM aircrafts are less economical and therefore not prioritized.

The same plots for each AIRCRAFT_STATUS resulted in very similar proportions for XLARGE, LARGE, and SMALL aircrafts where status "a" was the highest proportion followed be status "b", "o", and "l" in decreasing order. MEDIUM aircrafts had a different proportion with status "b" being the highest followed by status "o", "a", and "l" in decreasing order. This may imply that MEDIUM aircrafts have lower demand and are not as central to the operations.

Generative AI Disclosure

(1) I used GenAI for help with syntax for sklearn and pandas operations, and in depth help specifically for a word similarity function.

(2) ChatGPT 4o.

(3) The prompts I used to get the results are as follows:
What sklearn function can I use for embeddings and string similarity? I'm looking to group together similar kinds of words based on cosine similarity.