Module 4 exercises: Preprocessing Data for Analysis

Load in the dataset `renfe_trains.csv`

Initial data inspection

1. Inspect the columns of the DataFrame. Specifically, consider the type of each column and whether it seems reasonable. If not, investigate why.
2. It seems like we have some bad values in the price column with the value 'price'.

   You can see them by using the method .value_counts().

   Inspect the specific rows where this is the case.
3. It looks like some sort of error has meant the column names have been fed into the data in intervals. Let's drop these rows as they are clearly an accident.
4. We can now represent price using the appropriate type. Convert it to the appropriate data type.

Missing values

1. Identify whether there are missing values in the DataFrame.
2. Which columns are they in?
3. Inspect some rows which contain them.
4. Drop all rows which have missing `vehicle_class` and `price` and `fare` (i.e. a value of NaN for all of them). Hint: how='all'
5. Run the below code. What does it suggest about ticket price with respect to vehicle_class and fare?

   df[['vehicle_class', 'fare', 'price']].groupby(['vehicle_class', 'fare']).mean()
6. Fill the remaining missing price values with the mean of all the prices.
   a. In the extension, you can try to tackle this more appropriately (and trickily!).
7. Check you have gotten rid of all NaN values in df.

Deduplication

Use `duplicated` to see whether the dataset contains any duplicated rows.

As the dataset constitutes ticket price search results, there's a good chance duplication has come about due to the data collection method. For example, there are many tickets available on each train.

We would want to investigate this further, but to use the functionality, let's get rid of these duplicate rows.

Outliers

Identify outliers in the price column. A common measure used to determine outliers is 1.5 * IQR above the upper quartile (Q3) or below the lower quartile (Q1)

Examine these outliers. Do they appear to be erroneous or is there a reason that they exist?

Training, testing, validation

Split the dataset into training and testing sets, assuming you are trying to predict `price.`

Scaling

Using `scikit-learn`'s `StandardScaler`, scale the price column.

Encoding

Appropriately encode the destination column.

Stretch exercises

As it appears `price` depends upon `vehicle_class` and `fare`, we choose to replace missing `price` values with the average for their `vehicle_class` and `fare` category. Write some code which does this.