Python types and typecasting

Integer Any whole number including 0 and negative numbers.

Boolean This type can be either one of two values: true or false. Can be transformed from a column that has 0s and 1s. It is written as True / False in python.

Datetime A special type that recognizes the day, month, year, hour, minute and second information. Helps us compare the dates and times with each other.

Float Decimal numbers. They are written with a decimal point. Includes 0 (written as 0.0) and negative values.

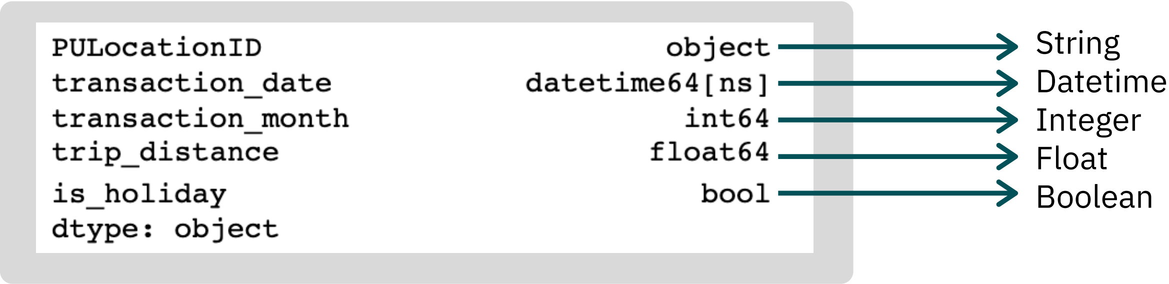
String Text values. Written in either single (') or double quotes("). Stated as "object" in the list of column types.

Figuring out types of columns

To figure out types of columns use the '.dtypes' function. Like this:

```
my_favorite_df.dtypes
```

And you will get a response that will look like this:



```
PULocationID                  object
transaction_date      datetime64[ns]
transaction_month              int64
trip_distance                float64
is_holiday                      bool
dtype: object
```

Typecasting

You can transform types into each other. This is called typecasting. Here is how some of these casts would look like:

5 → '5'
Integer to String

'5' → 5.0
String to Float

'2019-01-01 00:46:40' → Timestamp('2019-01-01 00:46:40')
String to DateTime

1.0 → **True**
Float to Boolean

False → 0
Boolean to Integer

'True' → **True**
String to Boolean

Why do we typecast?

Your data might be incorrectly imported. For example, most likely, when you first import your data to your notebook, the date column will be imported as String and not datetime. Thus, you need to cast the column to datetime in order to be able to use that column efficiently.

You might have a column that has integer values that represent different categories. If you feed your data to your model as it is, your model will assume that category number 3 is bigger than category number 2. Even though there is no direct relationship between those two categories. That's why you want to cast your column to String so that the model will see them as separate categories and not entities that relate to each other.

Make sure to check your data column's types at least once before training a model to make sure they make sense and they are the type you want them in.