Data Wrangling Report

Background

The purpose of this project is to predict the quantitative target variable, price, of an Uber or Lyft ride based on Boston Uber/Lyft ride data and weather data taken from a real-time API, ranging from November 26 to December 18, 2018. To achieve this, supervised regression methods will be employed on a clean dataset containing the weather data appended to the ride data, along with a dummy variable column representing the occurrence of a sports game/event that occurred within one hour of the game's start time and end time. This section details the methodology involved in the data wrangling process, cleaning all datasets, taking care of any missing values, and merging each dataset together.

Dataset Description

Real time ride data, collected using Uber and Lyft API queries, containing 10 features including: ride application used (Uber, Lyft), distance between pickup and dropoff, timestamp of ride, pickup location of ride, destination of ride, price of ride, surge multiplier of ride (over much price was increased), specific type of ride (Uber Black, Lyft Lux XL), and corresponding id is included. There are over 690000 ride data instances recorded for this dataset.

Weather data, containing shared features with the ride dataset such as the location of the ride pickup, at timestamps ranging between November - December, contains unique features such as rain, clouds, humidity, wind, and pressure measurements.

Sports data, which was manually created, contains features for NBA Celtics and NHL Bruins games occurring between November 26 to December 18, 2018. Features include location of the game's stadium (e.g. North Station), start time, stop time. This data was collected from NBA and NHL 2018-2019 schedule.

Ride and Weather data provided from:
https://www.kaggle.com/ravi72munde/uber-lyft-cab-prices and is relatively clean.

Methodology

To prepare for the data wrangling process, both the ride and weather data, from Kaggle, were saved as a local copy, in csv file format, for reference. The ride and weather data were read in, using the pandas package in Python, as two separate dataframes. The timestamp values, or columns, for the rides and weather dataframes were in epoch time format, where ride timestamps were in milliseconds and weather timestamps were in seconds, so they were converted into datetime objects. Both dataframes were then sorted by their timestamps, in which the column name of the location in the weather data was renamed to 'source' to match the source column (representing the pickup location of the ride). The rides dataset contained approximately 50000 missing values for the price column (representing the target variable price of the ride). All of

these instances with the missing price values come from an Uber ride type: Taxi. Because there is not any recorded price for the Taxi ride type, these ride instances were removed from the dataframe.

Once all of the missing values were filled in, the weather dataframe was then merged, using the merge_asof method in pandas, into the ride dataframe, matching weather rows to ride rows by the 'source' columns on the 'time_stamp' column. A timedelta tolerance of one hour is specified, which will match weather data occurring within one hour of the ride pickup time. The merged dataframe now contains missing weather column values due no weather data occurring within 1 hour of the ridetime. For example, multiple ride instances on November 26th, 2018 occur at 2:30 A.M., whereas the weather data begins at 3:40 A.M on the same day, yielding instances on this day with null weather values. To deal with this, these null weather value ride instances are dropped, for a final dataframe containing 633,296 ride instances and their appropriate weather data appended. A local copy of the dataframe is saved by creating a csv file containing all of the instances.

A dummy variable column is created representing an occurence of a sports game, such as an NHL game by the Boston Bruins, or an NBA game by the Boston Celtics, occuring at the TD Garden stadium, located in the vicinity of North Station. A separate csv file was manually made to display the location of the game (ie. North Station, Boston University, etc.), the type of game (i.e. NHL/Bruins, NBA/Celtics), start date-time and stop date-time. This file is read in, along with the merged ride and weather data, into pandas dataframes. The time_stamp column in the ride data and the start and stop times in the game data is converted to datetime objects in order to merge both dataframes. The location column in the game data is renamed to source to merge the game data with the ride data, by the 'source' with a tolerance of 2 hours on the start time in order to determine the game's effect on ride prices. Another merge is performed on the stop time of the game, since game attendees would require transportation after the game ends. Two more merges are performed to match the location of the game to the destination of the ride for both the start and stop time. Once all of the game data has merged, the 'type' column values are replaced as 1 if a ride occurred within two hours of the start or stop time of the game. All other null values, in which a ride did not occur at the location or within 2 hours of the start or stop time of the game, are filled in with the value 0. One type column is renamed to 'Sports Occurrence' which will hold the values of 1 or 0, and all other 'type' columns are deleted. The final merged dataframe is saved as a csv file ready for exploratory data analysis.