Ryan Pregitzer

ECON 9000

April 23, 2019

Problem Set 1: Dark Sky API

Project Overview

This project utilizes data from the Dark Sky API ([Powered by Dark Sky](https://darksky.net/poweredby/)), a weather forecasting application and website, to predict the temperature at any given time given several other known variables and trained through a linear regression machine learning model. I evaluated several variables, which I describe in depth later, and attempted to produce a model accurate in predicting my final variable, current temperature. My model predicted current temperature with upwards of 80% accuracy. Let’s begin by looking at the data I had to work with.

Description of the Data

The Dark Sky API provided me with a variety of weather variables and forecast information, some of which I employed in my project, and some of which I did not. My final analysis only included nine of at least 20 variables provided for me in my json files. I omitted some that had little to no effect on my predictions or far too much effect on my predictions (“feels like” temperature is clearly correlated with temperature, and was thus omitted). Each json file includes the latitude and longitude location along with a time zone, and is then broken up into four primary sections: currently, minutely, hourly, and daily. Other sections including flags and warnings are at the end, but were of no relevance for this project. ‘Currently’ provides the user with weather attributes for the current time stamp; ‘minutely’ gives forecast info each minute for the next hour; ‘hourly’ provides forecast information for the next 24 or 48 hours each hour; and finally, ‘daily’ provides forecast information for the next week by day.

My work focused solely on data from ‘currently,’ just for simplicity. An expansion of this project would involve comparing my predictions—particularly for precipitation—with those from Dark Sky by minute, hour, or day. Here are the nine variables I pulled from Dark Sky for my analysis, however, with brief descriptions. I added spaces and capitalization for readability:

Time: UNIX time for the beginning of the data point

Summary: a brief description of the weather (i.e. ‘clear’ or ‘heavy rain’)

Temperature: current temperature in Fahrenheit

Nearest Storm Distance: distance in kilometers to the nearest storm

Precipitation Probability: current probability of precipitation

Pressure: sea-level air pressure in millibars

Humidity: relative humidity, between 0 and 1

Wind Speed: wind speed in miles per hour

Cloud Cover: percentage of the sky obscured by clouds, between 0 and 1

Further descriptions can be found in the Dark Sky API documentation [here](https://darksky.net/dev/docs#response-headers), including those for the many other variables included in my json files. This documentation provides a wealth of other information as well to anyone interested in working with the Dark Sky API.

Dark Sky Analysis

I created a relatively simple model to predict temperature. Though I pulled all of the variables listed above from my data files, I only used ‘nearest storm distance,’ ‘precipitation probability,’ ‘pressure,’ ‘humidity,’ ‘wind speed,’ and ‘cloud cover’ to try to build my temperature prediction model. I then created a data set, which had these variables, and a target set, which had temperature. I then pulled 75% of the data set into a data training set, with the other 25% in a data test set. I performed a similar operation for the target data set. After fitting the data to a linear regression model, I predicted values based off the data test set and compared it to the target test set (actuals). My predictions matched the actual temperature values with 81.87% accuracy. I scattered these two to put together a visual of my work as well. This visual is included in my Github repository, saved as “scatter\_test\_prediction.png”. In terms of expansion, I mentioned earlier that it would be valuable to test my predictions with those of Dark Sky and compare each of them with actuals, but time did not allow me that luxury for this project. It would also be interesting to test my model by predicting the icon provided for ‘currently’ to see how well I can predict it. Either would be a feasible and interesting addition to the project.

Project Notes

The labels for my variables in my data table start with the digits 0 through 9 because I had difficulty sorting them other than alphabetically. This method ensured that the labels remained with the correct values, as unaesthetic as it turned out to be.