Application Programming interfaces are a mainstay of modern data usage. It is common to hear and see advertising for rest APIs. This week we have had an opportunity to learn the power or utilizing APIs to obtain data easily, while then being able to manipulate and clean up tables and databases using MongoDb and Python Pandas. APIs are an important tool that most all organizations are utilizing to obtain data and communicate effectively. The ability to understand the how’s and the whys of implementing API interconnectivity is not only an important but could be considered a necessary skill for the workplace of a data centric role.

What is an API, and what can we use them for?

API stands for Application Programming Interface; these are tools that allow a user to interface with a website and consume data to then be filtered into a digestible format for your computer. A computer can then edit data, without the manual approach of a person copying and pasting data from various sources. API integration is the process in which two systems are linked up, the server with the client which is a program “that knows what data is available through the API, and can manipulated it, typically at the request of the user.”Coosey (2014). APIs are helpful tools to scrape data from various sources, it takes what could be a manual process and creates an interface to obtain and transform data as the end user requires. Amazon rolled out their API to allow web developers could access their product information more easily, with the intention of allowing third party websites to post direct links to Amazon products and pricing with the option to buy now. APIs are generally used in similar scenarios to that of Amazon, it provides a 3rd party interface that allows an app to utilize their data.

Cooksey, Brian. April 22, 2014.An Introduction to APIs. Retrieved from <https://zapier.com/learn/apis/chapter-1-introduction-to-apis/>

Roos, Dave. How to Leverage an API for Conferencing. <https://money.howstuffworks.com/business-communications/how-to-leverage-an-api-for-conferencing1.htm>

When should we consider putting API-fetched data in SQL vs a NoSQL database?

API data is generally structured with a few general characteristics; high frequency, payload sizes, and data structure. The payload sizes can differ based on whether it is a request or a response payload, getting or providing data. The determination of using SQL vs. NoSQL can be made based on the some of the general guiding principles for deciding for NoSQL vs. SQL, this includes the data growth and anticipated archival of data – a data growth model will be better suited to NoSQL. Predetermined schema is better suited towards a SQL database; price and scalability – higher cost vertically scalable servers for SQL and lower cost horizontally scalable servers in the NoSQL option. If we know that our query structure will be unstandardized, then we would gravitate to SQL. The density of the analytics needed will also be a consideration in the choice of database, deep analytics are better suited to NoSQL whereas light analysis can be conducted in SQL.

Joshi, Vaneet. Relational vs. NoSQL Database for API Traffic. June 7, 2016. <https://blog.cloud-elements.com/relational-vs.-nosql-databases>

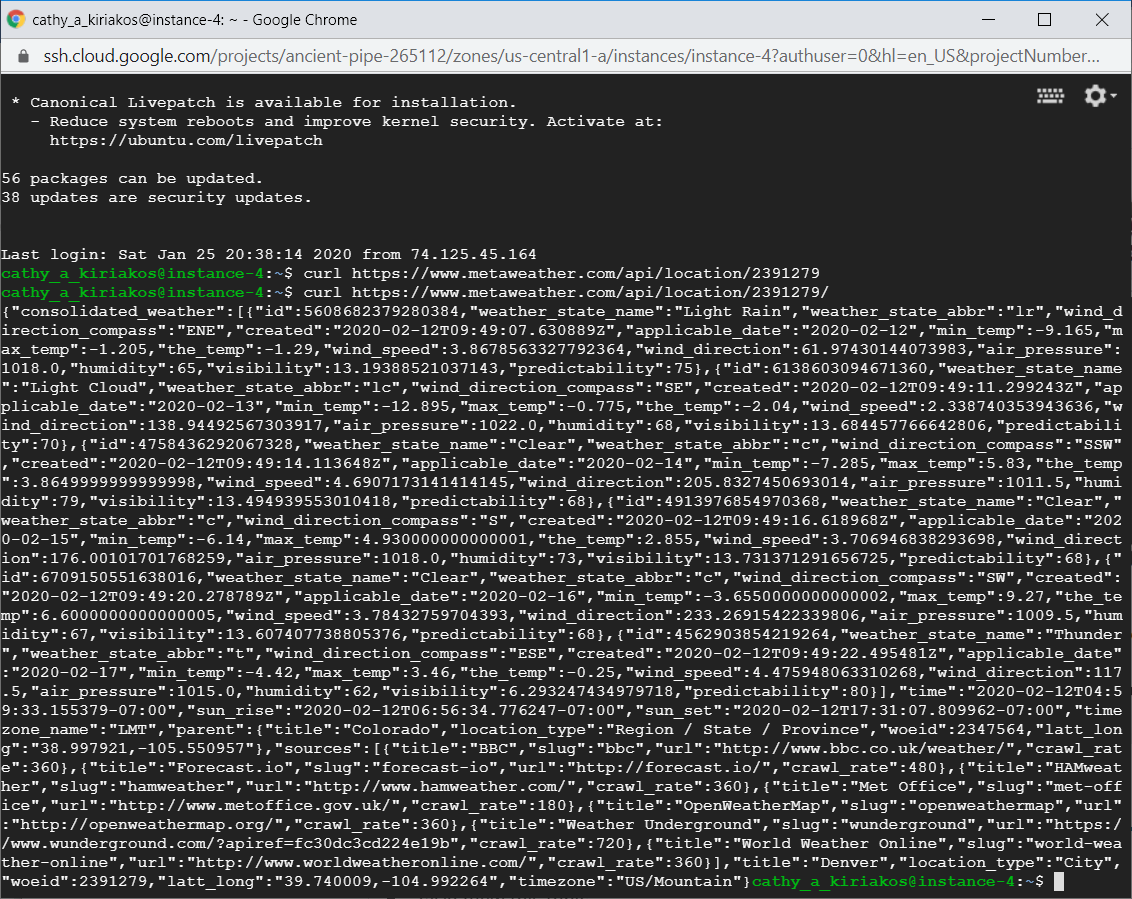
What was challenging about using APIs?

Application programming interfaces allow different applications and platforms to communicate and share data together. They’re popular because they’re fairy simple to implement, therefore less of a barrier to entry and work to provide a platform for business objects to communicate with one another. Despite all of the benefits of using APIs, there are still some downfalls or challenges. The first one noted in the blog post, was quality standards that vary across the marketplace leaving vulnerability to hackers. The second noted vulnerability involves is the simplicity of commands, and there are challenges of implementing AI into the applications so at the time of this article the intelligence was lacking. From a first time user, ensuring that you have the proper applications installed in the appropriate locations was a challenge, in my scenario I had some difficulty installing the yahoo finance, as there was a change/deprecation in the name of the programs to be installed. I was able to update my pip installation, pyHamcrest which was needed for the yfinance to work, and then change the name of the installation to get it running properly when getting a different API going to pull the S&P 500 data from Yahoo Finance. Additionally, when pulling weather data via the API, I had to ignore the verification process; which I could see as potentially creating a vulnerability when utilizing the weather data.

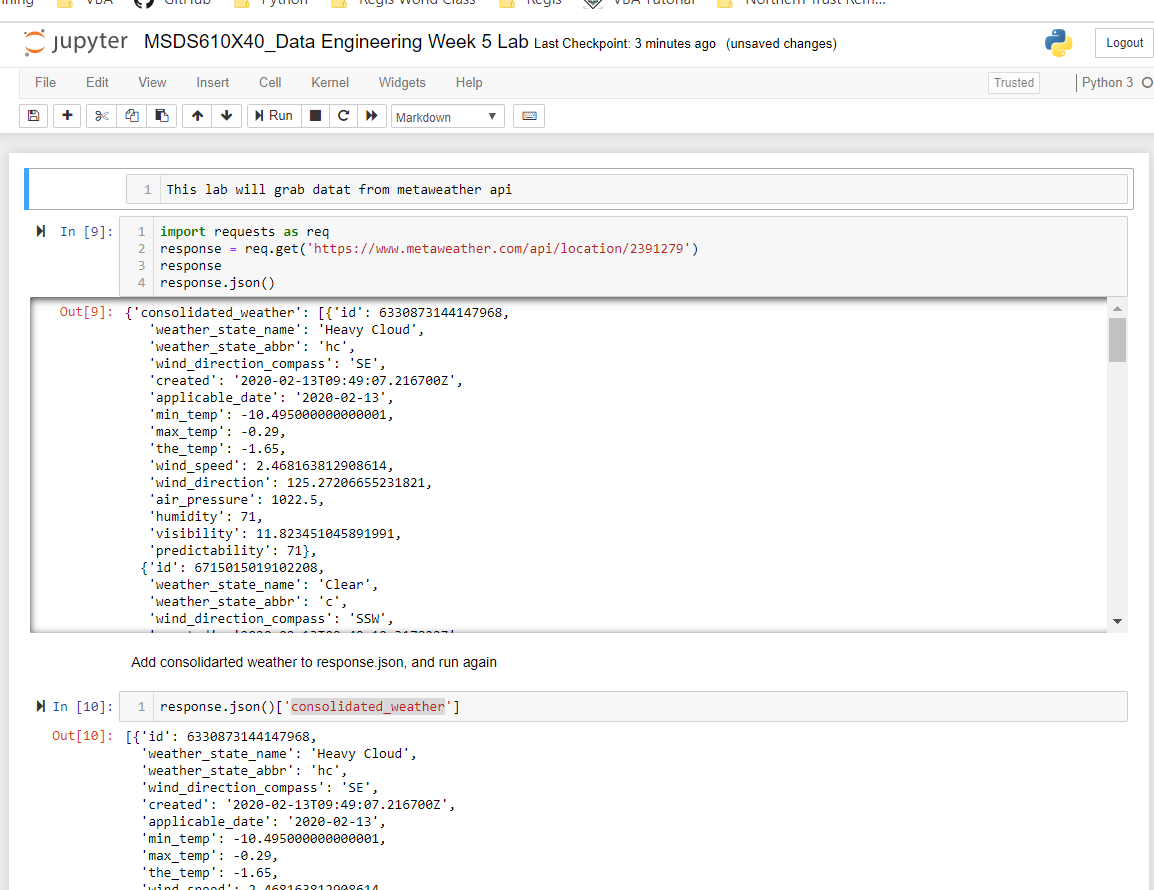
What are APIs and what challenges do APIs face? November 9, 2019 https://blog.signaturit.com/en/what-are-apis-and-what-challenges-do-they-face

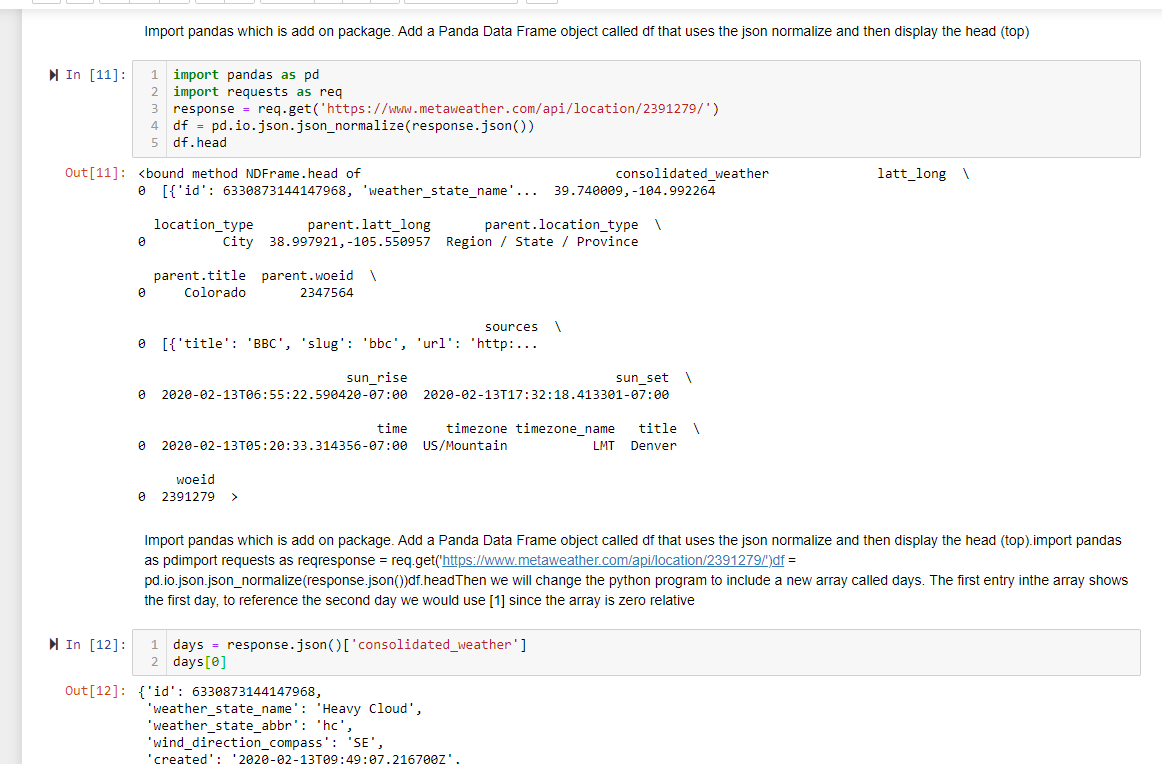
Technical Lab

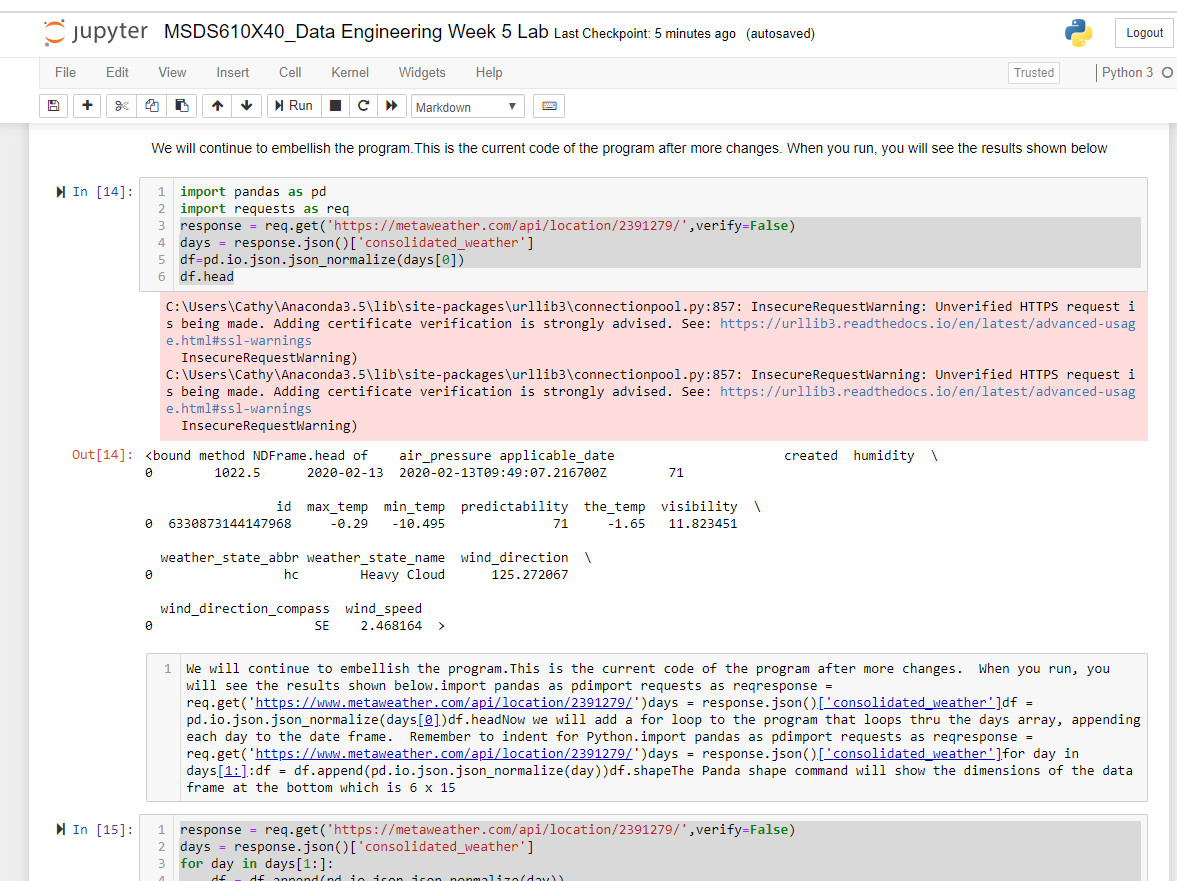
Below is summary of the calls for the technical portion of the API lab, below you will find a snippet from getting the cgp API calls

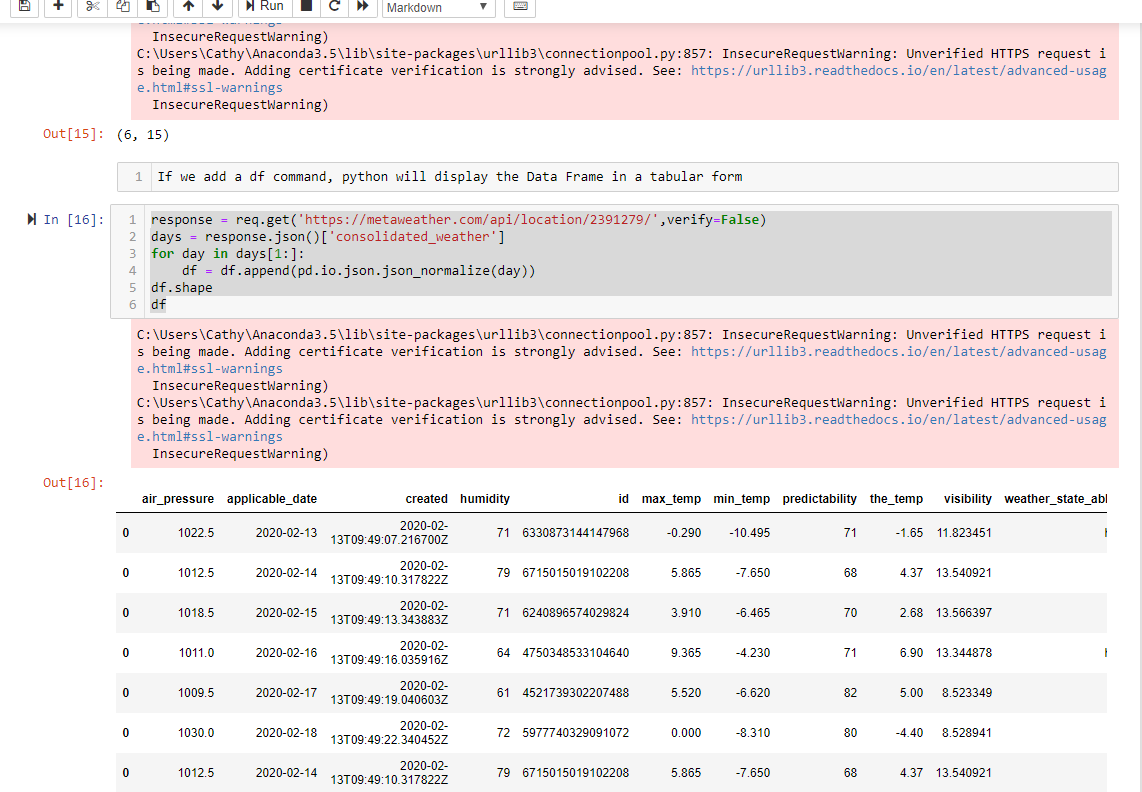


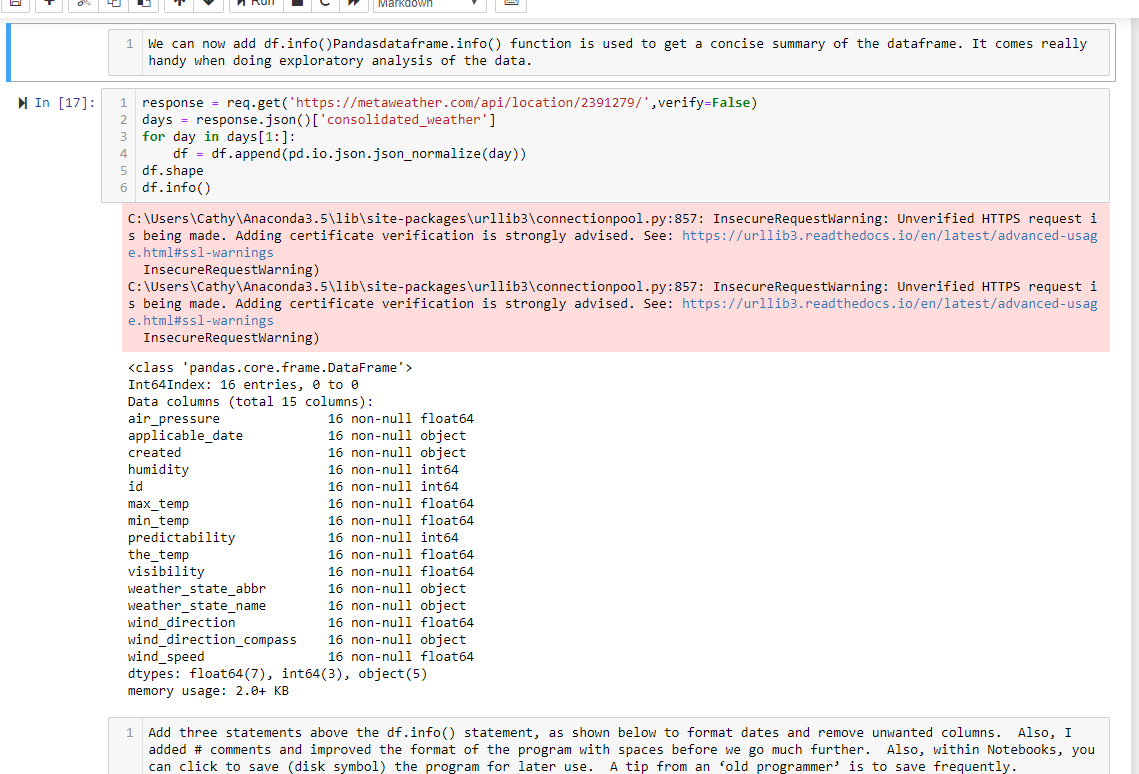
Moving on to the Jupyter notebook piece of the lab, below you can see the python calls – that pulled the Weather data into a data base using pandas. I had commented out the processes in the markdown cells shown in the snips below:

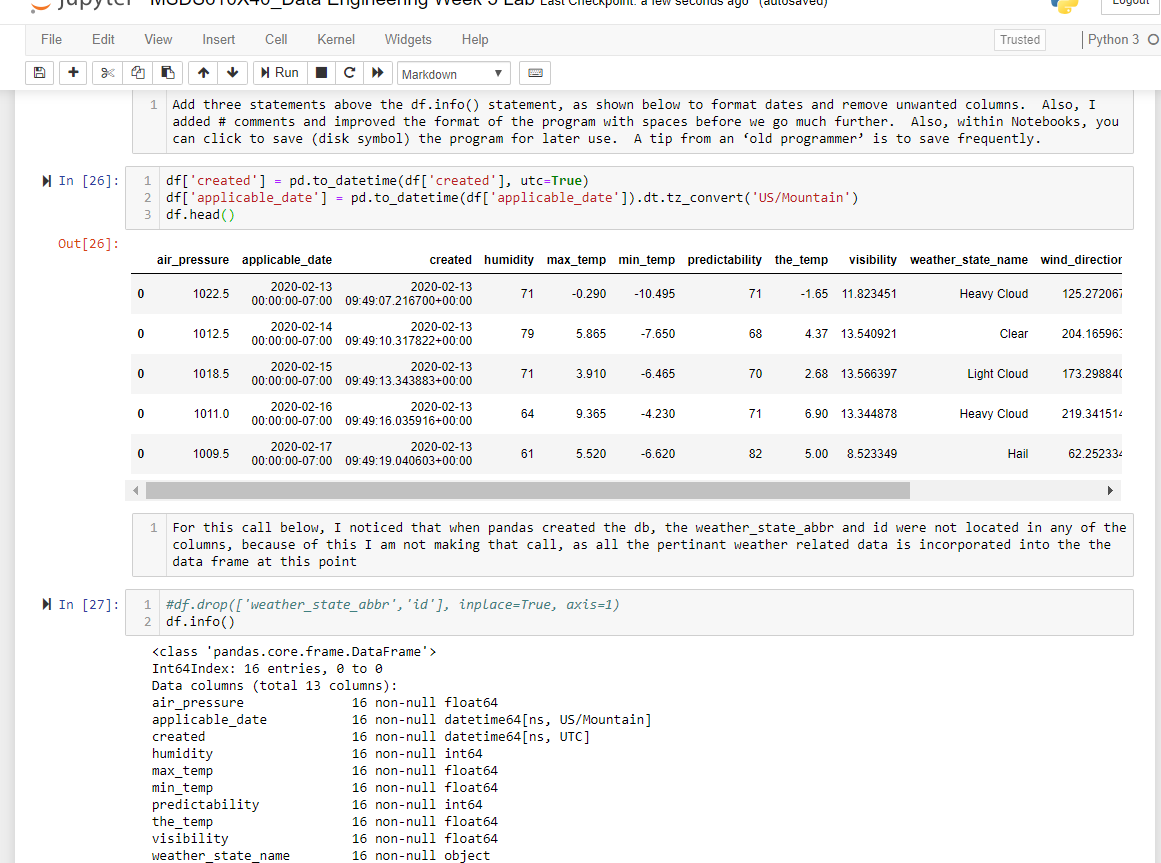


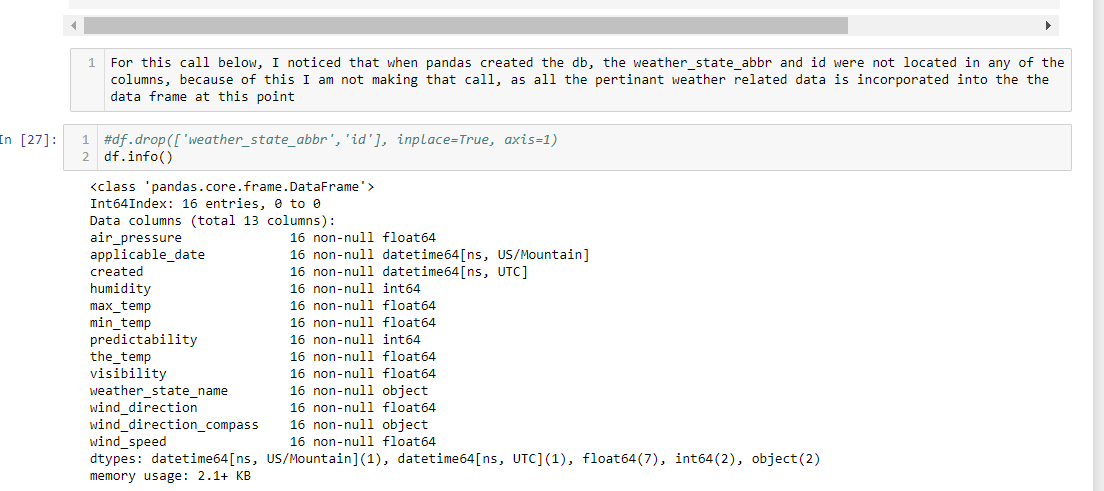




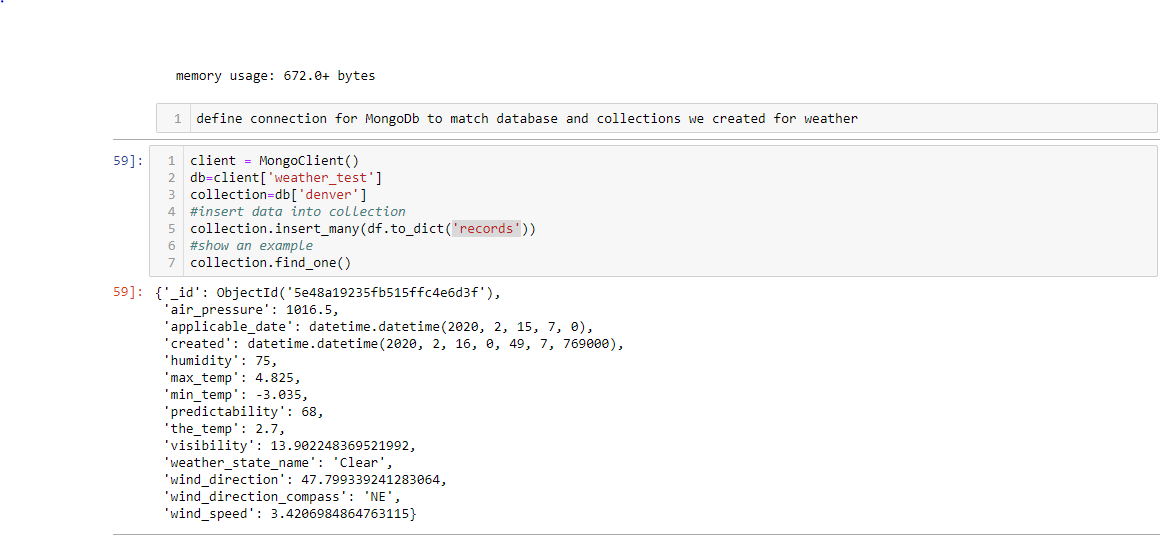


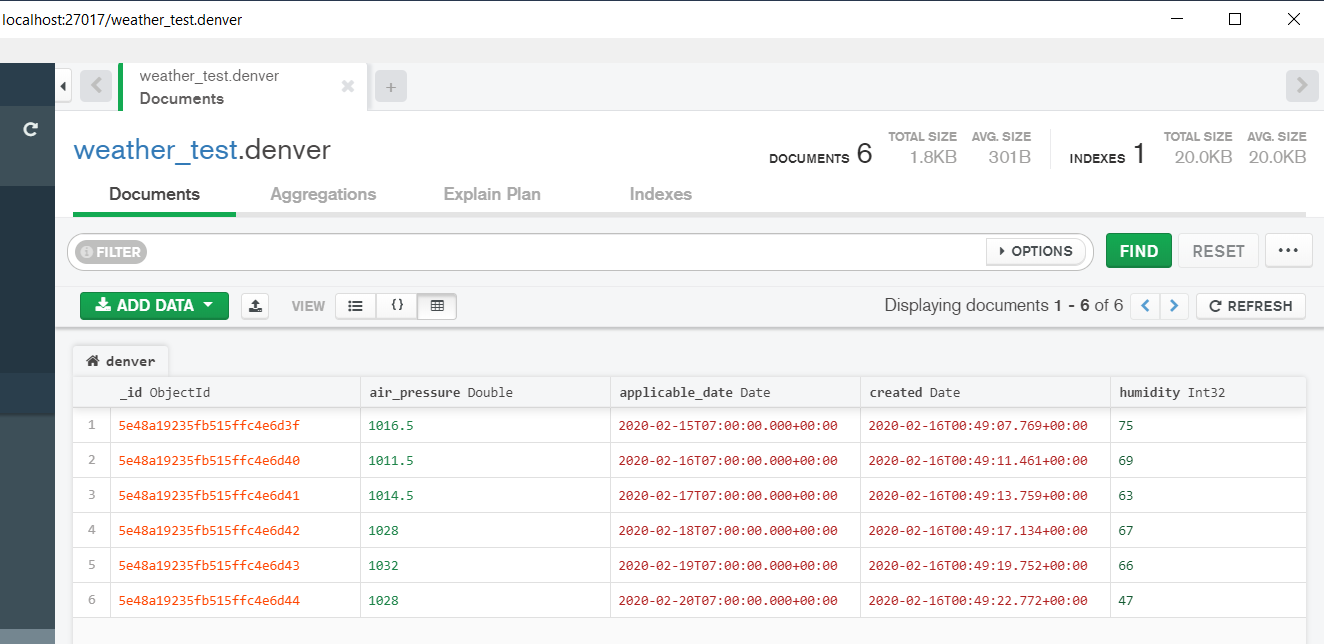




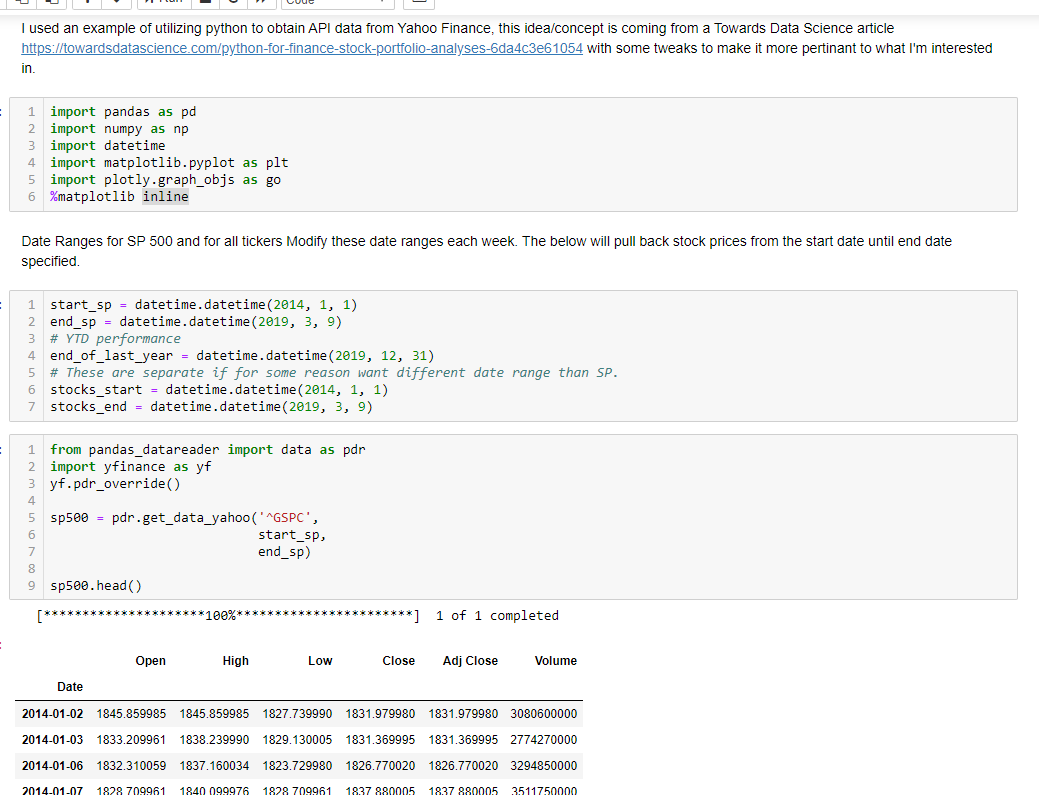


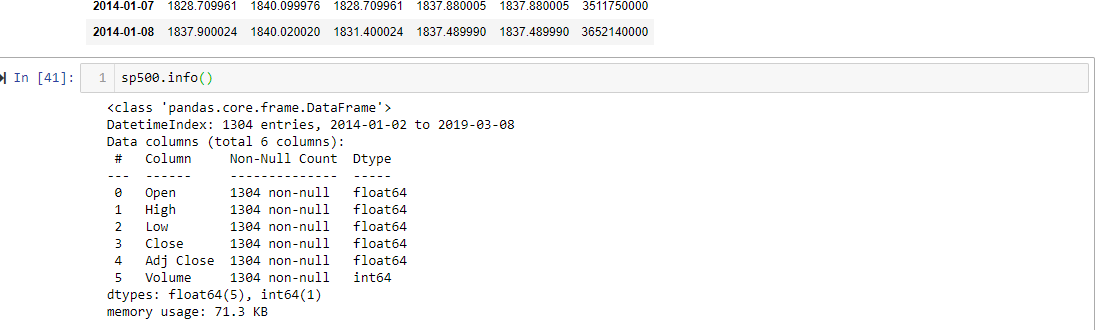
Now on to creating a new database in MongoDB for the weather data frame that was built in python:





Creating my own API, I followed an example that I found at Towards Data Science, utilizing data from the Yahoo Finance API. Below is a summary showing Jupyter Notebook utilizing pandas\_datareader to obtain data from the S&P 500 Index for a 2014-2019 time period:





Below is a plot of the S&P information pulled from the API

