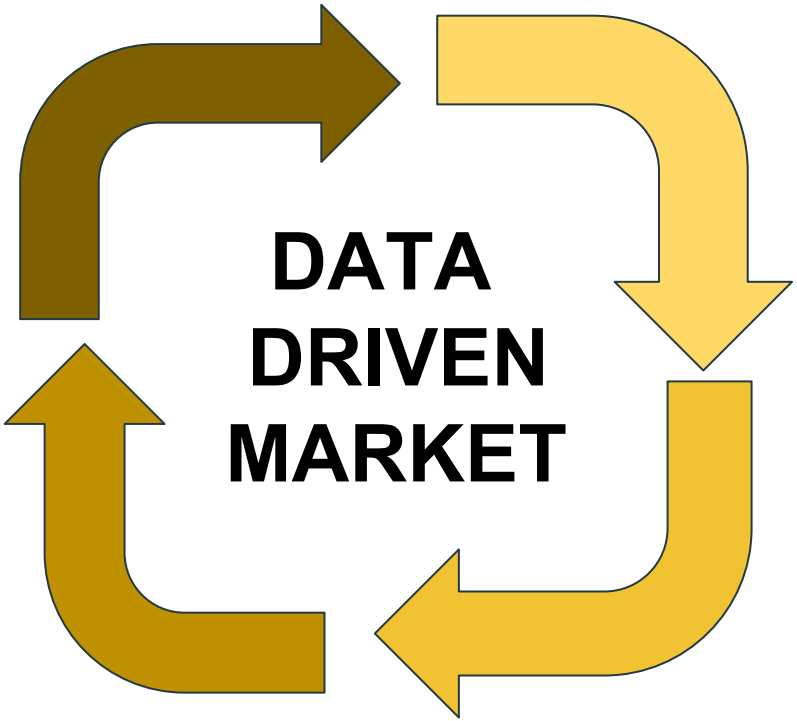
Airfare Data Scraper

Deep Dave
David Lin
Sharon Ng
Vanessa Salas
Alexandre Vincent

Web Scraping Applications

Record Results



SCRAPE DATA:

Stock Prices
Housing Market
Energy Consumption
Consumer Market

Analyze Data:

Find Trends
When is the best time to buy?
Increase Profit Margin

Make Data Driven Decisions

The Original Problem:

When is the cheapest time of day to buy plane tickets?

Upon weeks of searching for the right data set and different projects to pivot towards, there are 2 main problems:

Kaggle data sets only had flight times

Dept. of Transportation only had yearly average flight prices.

What is the Project?

Objective: Scrape Google flights on a daily basis to create a workable and shareable flight price data set

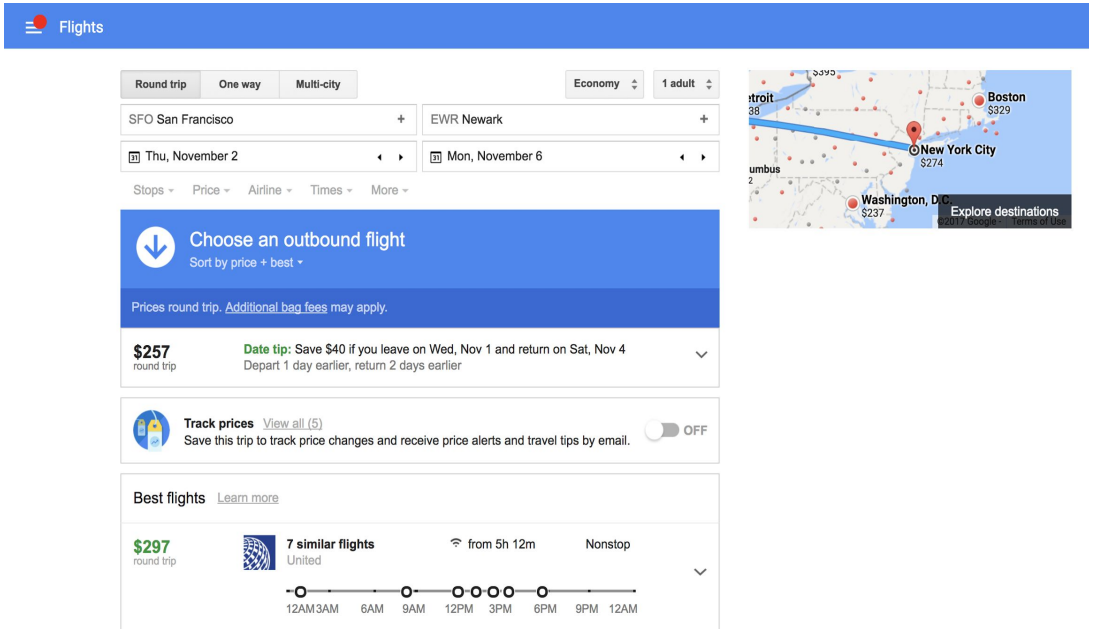
How:

1. Write a program that scrapes data on airfare prices for certain flights each day on an hourly basis

2. Collect the data into a CSV file or database

3. Share with the world

User Interface

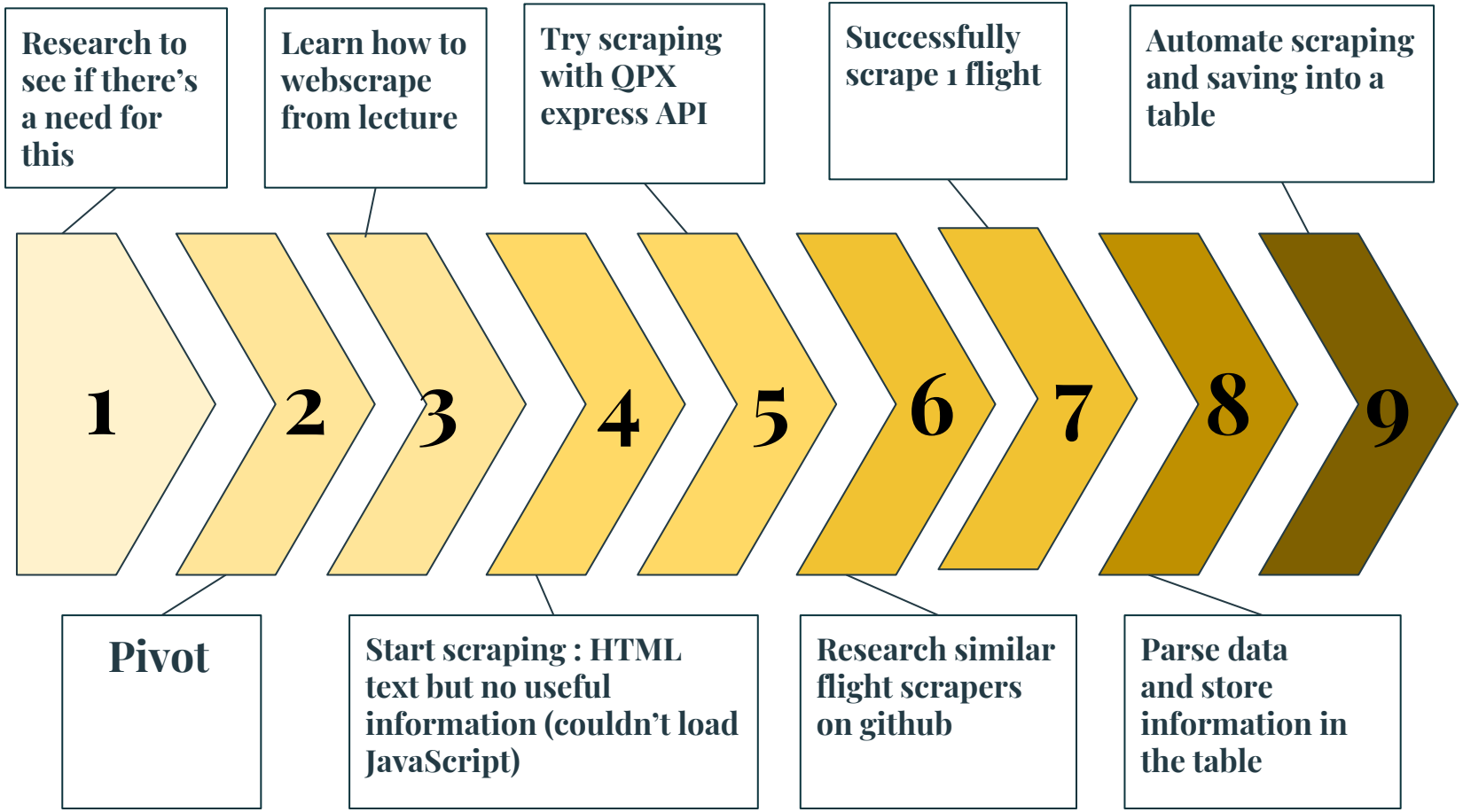


Inputs: Flight ID

Output: CSV file with the prices and the dates associated

Top 3 User Requirements:

1. Thorough datasets with many flights available
2. Presentation (clean data)
3. More features in the dataset


Research to see if there's a need for this
Learn how to webscrape from lecture
Try scraping with QPX express API
Successfully scrape 1 flight
Automate scraping and saving into a table
1
2
3
4
5
6
7
8
9
Pivot
Start scraping : HTML text but no useful information (couldn't load JavaScript)
Research similar flight scrapers on github
Parse data and store information in the table

Phase 1 - Research

Google flights, Scott's Cheap Flights and Skiplagged all have algorithms to detect the lowest prices. Therefore, it must be possible to find a dataset with hourly flight prices so we can do some analysis ourselves

We looked into various factors and trends that could determine airfare prices including:

- Least popular travel days (Tuesday, Wednesday, Saturday)
- Empty middle seats
- Flight distance
- Aviation turbine fuel (ATF) accounts for ⅓ of an airline's total operating expenses

Phase 2 - Scraping

After learning about web scraping in class, we took matters into our own hands. We tried:

1. **Basic scraping with BeautifulSoup and requests**

PROS	CONS
Scraping was possible because we got all the HTML text with formatting	But, we didn't get any relevant or important information because it's loaded in Javascript

2. **Use Google API Developer method**

PROS	CONS
Established library and easy to use commands	QPX express not assigned to project, 50 query quota per day, unavailable after 4/2018

3. **Use Selenium + PhantomJS**

PROS	CONS
IT WORKS!!	Difficult to set up for different computers

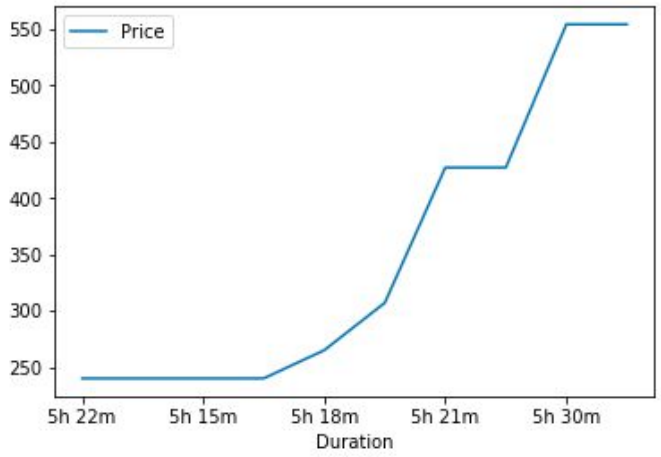
Phase 3 - Automate the scrape

1. **Proof of concept**
   a. Scraped several flights from the same day
   b. Able to parse the data with specific characters
   c. Wrote definitions to get that information into a dataframe

2. **Automation**
   a. Creating a tool that would run the program at specific intervals
   b. Writing a for loop statement for certain URLs
   c. Appending each URL's information to the rest of the data set

Analysis:

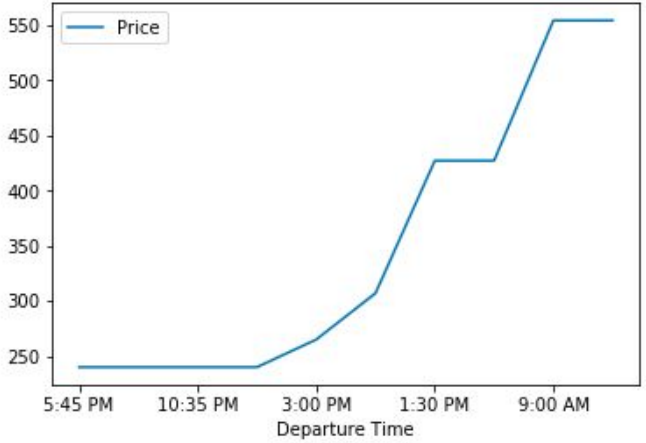
```
flights.plot('Duration', 'Price')
plt.show()
```



From our data set, we verified that the longer the flight, the higher the price, likely due to jet fuel prices. However, this could use additional consumer research because travellers typically wouldn't pay more for longer flights.

This plot makes sense because the earlier you travel in the day, the more time you have to spend at your destination. Therefore, it makes sense that it is more expensive to purchase plane tickets earlier in the day.

```
flights.plot('Departure Time', 'Price')
plt.show()
```



Google Flights URL

https://www.google.com/flights/#search;f=SFO;t=EWR;d=2018-04-01;tt=o;a=UA;s=0

f = From this airport (ex. SFO)

t = To this airport (ex.JFK)

d = Date of flight (ex YYYY-MM-DD)

tt = Travel type (ex. O for one-way, m for multi-city)

a = airline (ex. UA for United)

s = # of stops (ex. 0 for nonstop)

Architecture Layout

Input: Google Flights URL
https://www.google.com/flights/#search;
f=SFO;
t=EWR;
d=2017-11-02;
r=2017-11-06

Back-end: Scraping
- Price
- Flight time

Variables
- By airline
- By time of day

Calculations/Analysis
- Days away
- Near holiday

Output: Clean, New Data!
Columns:
- From city
- To city
- Date
- Specific Time
- Length of flight

Track specific flight cost over time and build long term dataset

Applications

Round trip | One way | Multi-city

Economy | 1 adult

SFO San Francisco +

EWR Newark +

Thu, November 2

Add return date

Stops Price Airline Times More

Choose a flight

Sort by price + best

Prices one way. Additional bag fees may apply.

$129
one way

Date tip: Save $20 if you leave on Wed, Nov 1
Depart 1 day earlier

Track prices View all (5)
Save this trip to track price changes and receive price alerts and travel tips by email.

OFF

Best flights Learn more

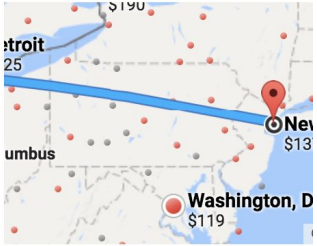
$149
one way

7 similar flights
United

from 5h 12m

Nonstop

12AM 3AM 6AM 9AM 12PM 3PM 6PM 9PM 12AM



Prices will likely increase in 5 days. · Show less

Historically, 90% of the time the cheapest price on this route increased 11 days before departure by at least $8.

https://github.com/lin-david/airfare-scraping-dataset

	Unnamed: 0	Airline	Departure City	Arrival City	Date	Departure Time	Arrival Time	Price	Duration	Stops
0	0	United	SFO	EWR	2018-04-01	5:45 PM	2:07 AM+1	240	5h 22m	Nonstop
1	1	United	SFO	EWR	2018-04-01	8:40 PM	5:02 AM+1	240	5h 22m	Nonstop
2	2	United	SFO	EWR	2018-04-01	10:35 PM	6:50 AM+1	240	5h 15m	Nonstop
3	3	United	SFO	EWR	2018-04-01	11:55 PM	8:15 AM+1	240	5h 20m	Nonstop
4	4	United	SFO	EWR	2018-04-01	3:00 PM	11:18 PM	265	5h 18m	Nonstop
5	5	United	SFO	EWR	2018-04-01	8:10 AM	4:26 PM	307	5h 16m	Nonstop
6	6	United	SFO	EWR	2018-04-01	1:30 PM	9:51 PM	427	5h 21m	Nonstop
7	7	United	SFO	EWR	2018-04-01	2:30 PM	10:57 PM	427	5h 27m	Nonstop
8	8	United	SFO	EWR	2018-04-01	9:00 AM	5:30 PM	554	5h 30m	Nonstop
9	9	United	SFO	EWR	2018-04-01	12:30 PM	8:47 PM	554	5h 17m	Nonstop

Future Work

-Data set with many different destinations, airlines and price variations for many more time intervals

-Automate program to run on specific intervals

-Make a user interface, for consumers

References

https://www.crummy.com/software/BeautifulSoup/bs4/doc/

https://www.dataquest.io/blog/web-scraping-tutorial-python/

https://github.com/ikhlaqsidhu/data-x/tree/master/03-tools-webscraping-crawling_api_afo

Headless Selenium testing with Python and PhantomJS:

https://realpython.com/blog/python/headless-selenium-testing-with-python-and-phantomjs/

Setting PhantomJS user agent string:

https://coderwall.com/p/9jgaeq/set-phantomjs-user-agent-string

Another helpful reference for airfare scraping:

https://github.com/hakanmhmd/air-fare-scraper/blob/master/flight_price_scrape.ipynb