IBM Developer
SKILLS NETWORK

Winning the Space Race with Data Science

Andre Moreira

June 21, 2023

Outline

- Executive Summary
- Introduction
- Methodology
- Results
- Conclusion
- Appendix

Executive Summary

The Project

- Provide competitive insights to SpaceY, a newcomer in the space transportation business
  - Find key success factors of the market leader, SpaceX's
  - Develop a predictive model for mission success

Methodology

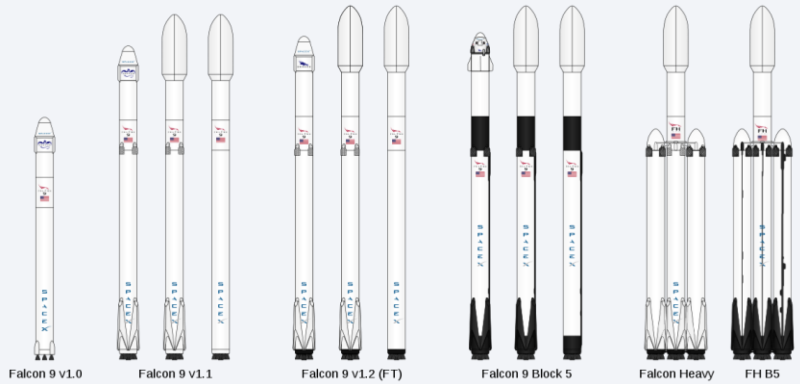
- Data collection (SpaceX APIs, web scraping) and wrangling
- EDA using visualization packages and SQL
- Interactive geospatial information using Folium
- Interactive dashboard using Dash and Plotly
- Develop a predictive outcome model using Machine Learning techniques for supervised classification models

Key Learnings & Results

- Between 2010 and 2020, Falcon 9 transported ~550 metric tons of material to the space
- During this period, the payload per launch increased by more that 5x (from ~2 to >10 tons on average)
- Falcon 9 booster recovery and re-use is a key success factor → reduces overall launch costs
- By 2020, the overall launch success rate was ~66% and ~40% of all launches used recovered boosters
- The company uses existing facilities to launch their vehicles, which reduces upfront investment level
- The best predictive model for launch outcomes is an optimized Decision Tree, with the highest accuracy and lowest False Positives

Introduction

- **SpaceY**
  - Founded by Allon Mask
  - Wants to enter the space transportation business
  - Requested a competitive intelligence project on the market leader (SpaceX)
- **SpaceX**
  - Founded in 2002 by Ellon Musk
  - Goal: *reduce space transportation costs and colonize Mars*
  - Commercial launches offered at 62 mUSD → ca. 1/3 of the price offered by other commercial operators
  - Falcon 9 is its “workhorse” and key player in its success story
- **This project**
  - Find the key factors driving SpaceX’s successful missions
  - Develop a predictive model for mission success



Section 1

Methodology

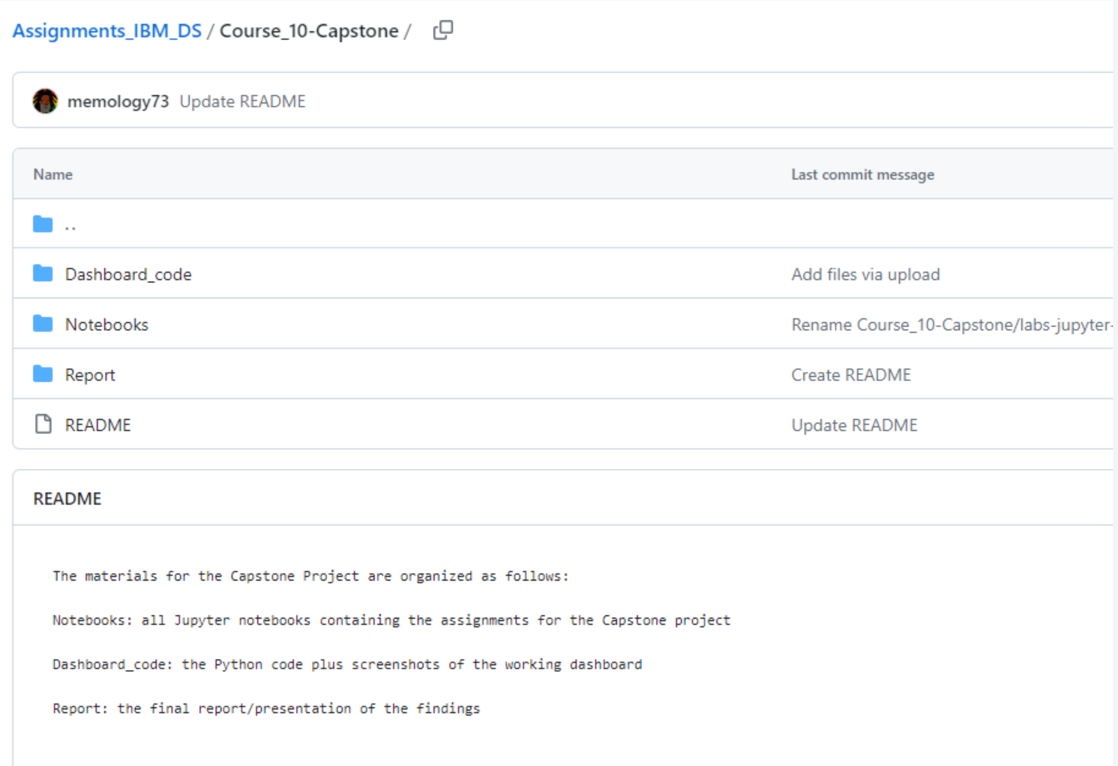
Before We Start: GitHub Links

All materials related to the Capstone Project are under the following link

https://github.com/memology73/Assignments_IBM_DS/tree/main/Course_10-Capstone

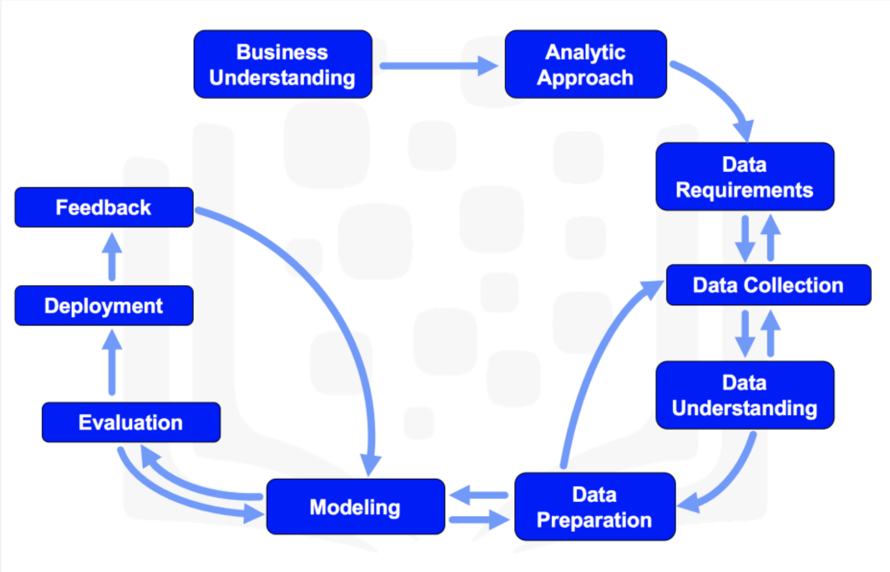
- Notebooks: all Jupyter notebooks containing the Capstone project assignments
- Dashboard_code: the Python code plus screenshots of the working dashboard
- Report: the final report/presentation of the findings (PDF)

GitHub links are also available at the bottom of several slides, corresponding the slide's content

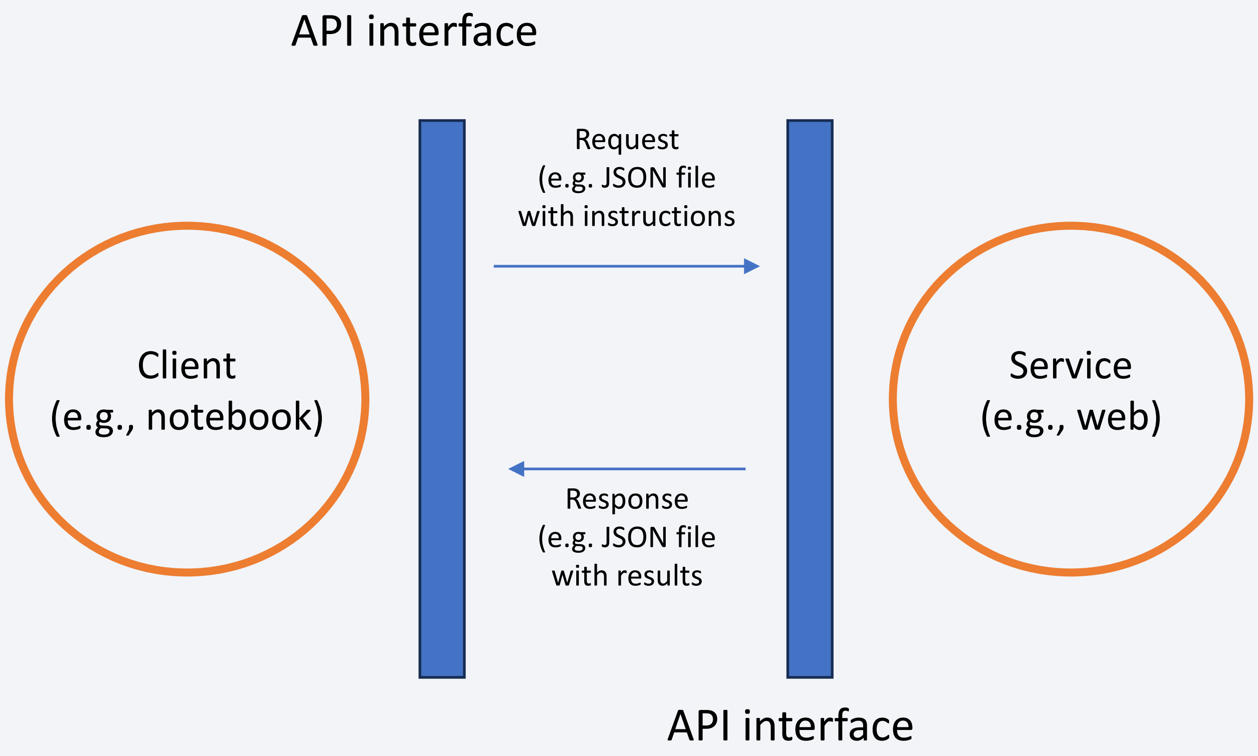


Methodology - Summary

- Data collection methodology
  - Collected data using API requests and web scraping
- Perform data wrangling
  - Substituted some missing data for column mean values, translated outcomes into success (“1”) and failure (“0”)
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Used advanced python packages to train and evaluate different classification models



Data Collection – SpaceX API



- Used an API interface to connect with a website and download JSON files
- Used the method .json_normalize() to transform the JSON content into a Pandas data frame
- Later in the process, we will use the data frame to manipulate and prepare the data for analysis

https://github.com/memology73/Assignments_IBM_DS/blob/main/Course_10-Capstone/Notebooks/1_jupyter-labs-spacex-data-collection-api_AM.ipynb

Data Collection – Scraping

Used HTTP GET method to request the Falcon9 Launch HTML page, as an HTTP response.

Used BeautifulSoup to parse and organize the data

Stored the data as a Pandas data frame that we can use to manipulate and organize

https://github.com/memology73/Assignments_IBM_DS/blob/main/Course_10-Capstone/Notebooks/2_jupyter-labs-webscraping_AM.ipynb

Data Wrangling

Load Space X dataset, from last section.

```
df=pd.read_csv("https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN-SkillsNetwork/datasets/data
df.head(10)
```

Date	BoosterVersion	PayloadMass	Orbit	LaunchSite	Outcome	Flights	GridFins	Reused	Legs	LandingPad	Block	ReusedCount	S
2010-06-04	Falcon 9	6104.959412	LEO	CCAFS SLC 40	None None	1	False	False	False	NaN	1.0	0	B
2012-05-22	Falcon 9	525.000000	LEO	CCAFS SLC 40	None None	1	False	False	False	NaN	1.0	0	B
2013-03-01	Falcon 9	677.000000	ISS	CCAFS SLC 40	None None	1	False	False	False	NaN	1.0	0	B
2013-09-29	Falcon 9	500.000000	PO	VAFB SLC 4E	False Ocean	1	False	False	False	NaN	1.0	0	B

```
# Create a dictionary that will translate the Out
otc_dict_0 = dict.fromkeys(bad_outcomes, 0)
otc_dict = dict.fromkeys(good_outcomes, 1)
otc_dict.update(otc_dict_0)
otc_dict
```

a

```
{'True RTLS': 1,
 'True Ocean': 1,
 'True ASDS': 1,
 'False ASDS': 0,
 'False Ocean': 0,
 'None None': 0,
 'False RTLS': 0,
 'None ASDS': 0}
```

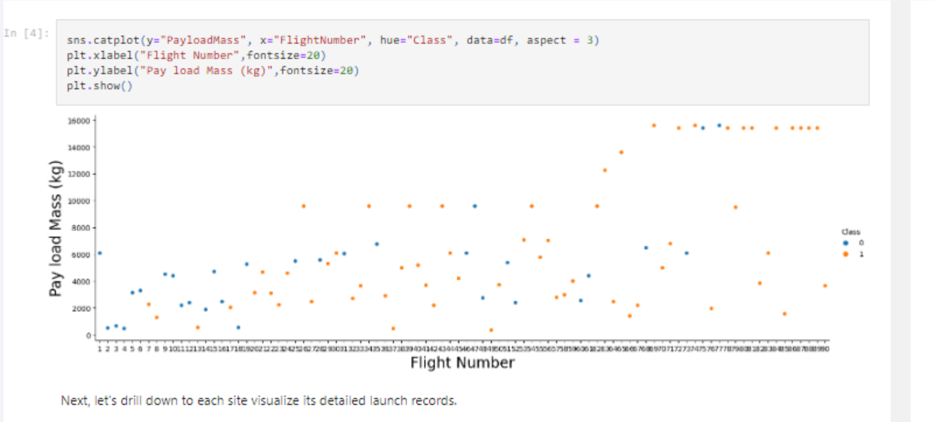
b

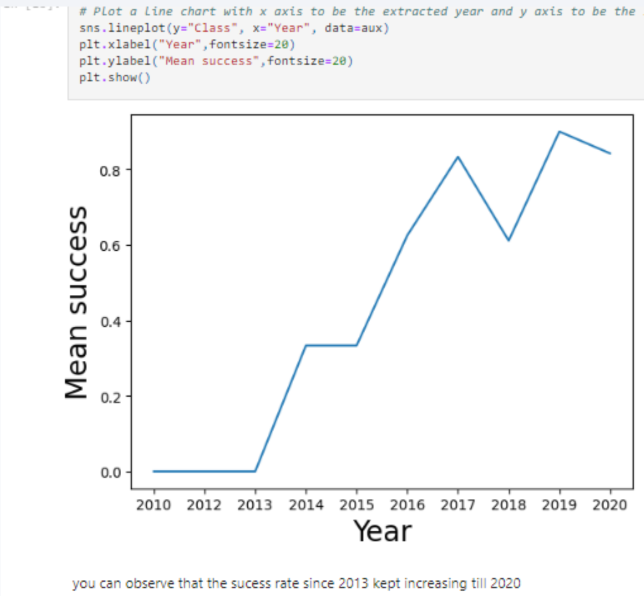
Block	ReusedCount	Serial	Longitude	Latitude	Class
1.0	0	B0003	-80.577366	28.561857	0
1.0	0	B0005	-80.577366	28.561857	0
1.0	0	B0007	-80.577366	28.561857	0
1.0	0	B1003	-120.610829	34.632093	0
1.0	0	B1004	-80.577366	28.561857	0

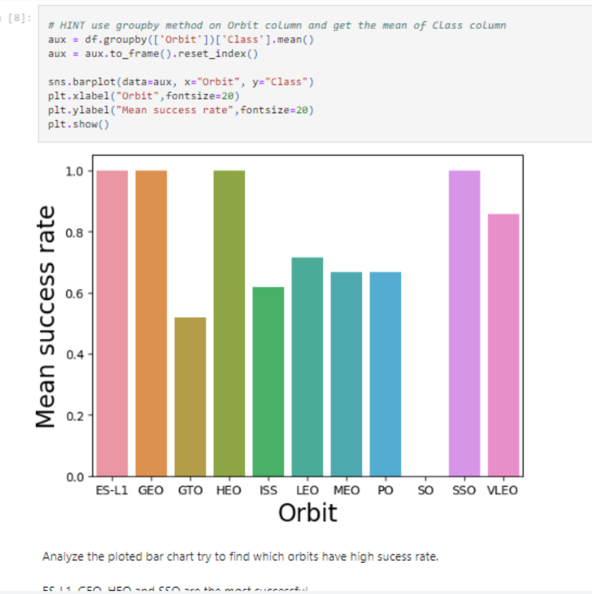
- Before we try to find the key factors driving SpaceX's successful missions, we need to "define success" and put it in a format that we can use for analysis.
- From the stored data frame, we created a dictionary (see picture, step a) that translated the outcomes into 1s – success and 0s – failed (step b)
- The column ('Class') will form the target variable for the predictive models that we will develop for this project

https://github.com/memology73/Assignments_IBM_DS/blob/main/Course_10-Capstone/Notebooks/3_labs-jupyter-spacex-Data%20wrangling_AM.ipynb

EDA with Data Visualization







- At this stage, we want to familiarize ourselves with the data and what it is telling us
- With the Seaborn package, we used various plots to look at different aspects of the data
  - How does the payload change over time?
  - How is the success rate changing over time?
  - Is there an obvious relationship between orbit and success rate?
  - etc.

https://github.com/memology73/Assignments_IBM_DS/blob/main/Course_10-Capstone/Notebooks/5_jupyter-labs-eda-dataviz_AM.ipynb

EDA with SQL

```
%%sql
SELECT MIN(DATE("Date")) FROM SPACEXTBL WHERE "Landing_Outcome" = 'Success (ground pad)'
```

```
 * sqlite:///my_data1.db
Done.
```

MIN(DATE("Date"))
2015-12-22

```
%%sql
SELECT "YYYY" AS 'Year', STRFTIME('%m', DATE("Date")) AS 'Month', "Landing_Outcome",
    "Booster_Version", "Launch_Site", "YYYY" AS 'Year' FROM
        (SELECT *, STRFTIME('%Y', DATE("Date")) AS 'YYYY' FROM SPACEXTBL)
        WHERE "YYYY" = '2015' AND "Landing_Outcome" = 'Failure (drone ship)'
```

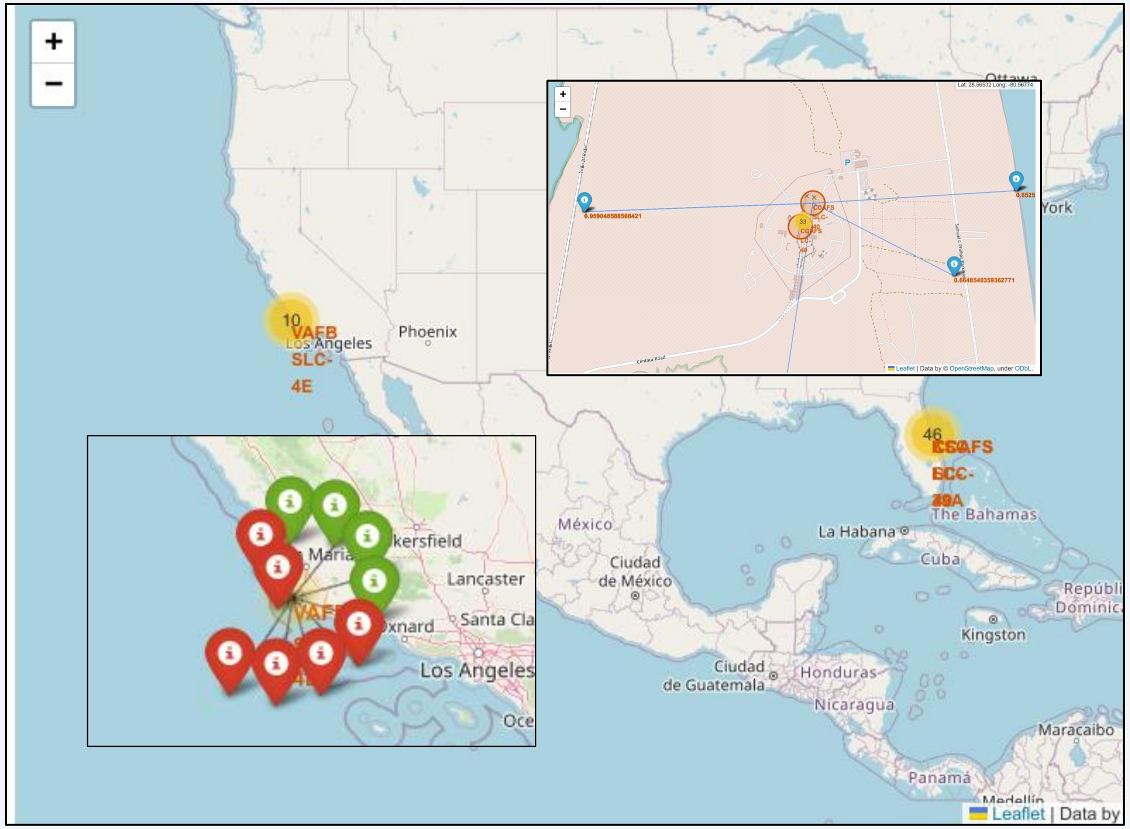
```
 * sqlite:///my_data1.db
Done.
```

Year	Month	Landing_Outcome	Booster_Version	Launch_Site	Year_1
2015	10	Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40	2015
2015	04	Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40	2015

- We used SQL to query and explore different aspects of the data
- We used simple queries (top picture) as well as nested queries (bottom picture) to explore questions like
    - When was the first time that Falcon 9 successfully landed on a ground pad?
    - When in 2015 did Falcon experienced failed landings on drone ships? Which versions were involved?
    - etc.
- SQL is fast and useful for very large data sets

https://github.com/memology73/Assignments_IBM_DS/blob/main/Course_10-Capstone/Notebooks/4_jupyter-labs-eda-sql-coursera_sqllite_AM.ipynb

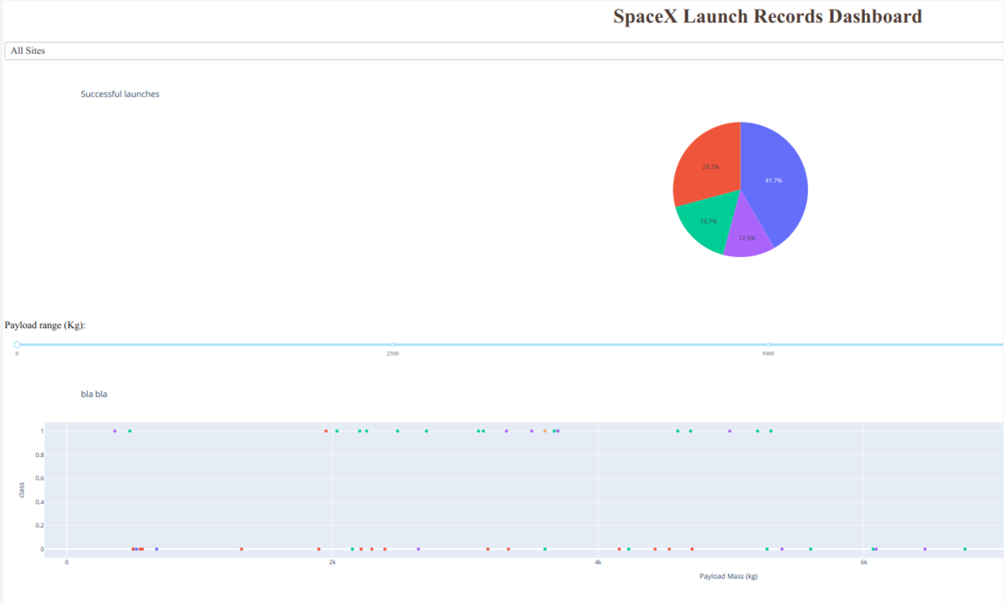
Build an Interactive Map with Folium



- We used Folium to locate in a map the launch sites based on their coordinates
- We used simple and clustered markers to point out relevant landmarks, e.g., site location, how many launches from a site, how many successful vs. failed launches
- We also used MousePosition to explore the coordinates of infrastructure (e.g., roads) in a launch site's vicinity and draw conclusions

https://github.com/memology73/Assignments_IBM_DS/blob/main/Course_10-Capstone/Notebooks/6_lab_jupyter_launch_site_location_AM.ipynb

Build a Dashboard with Plotly Dash



- Using Dash and Plotly, we created an interactive dashboard that enables the user to explore different aspects of SpaceX's launches
- With this specific dashboard, the user can understand
  - The overall success rate as well as of a specific launch site
  - How the payload correlates (or not) with launch success

https://github.com/memology73/Assignments_IBM_DS/tree/main/Course_10-Capstone/Dashboard_code

Predictive Analysis (Classification)

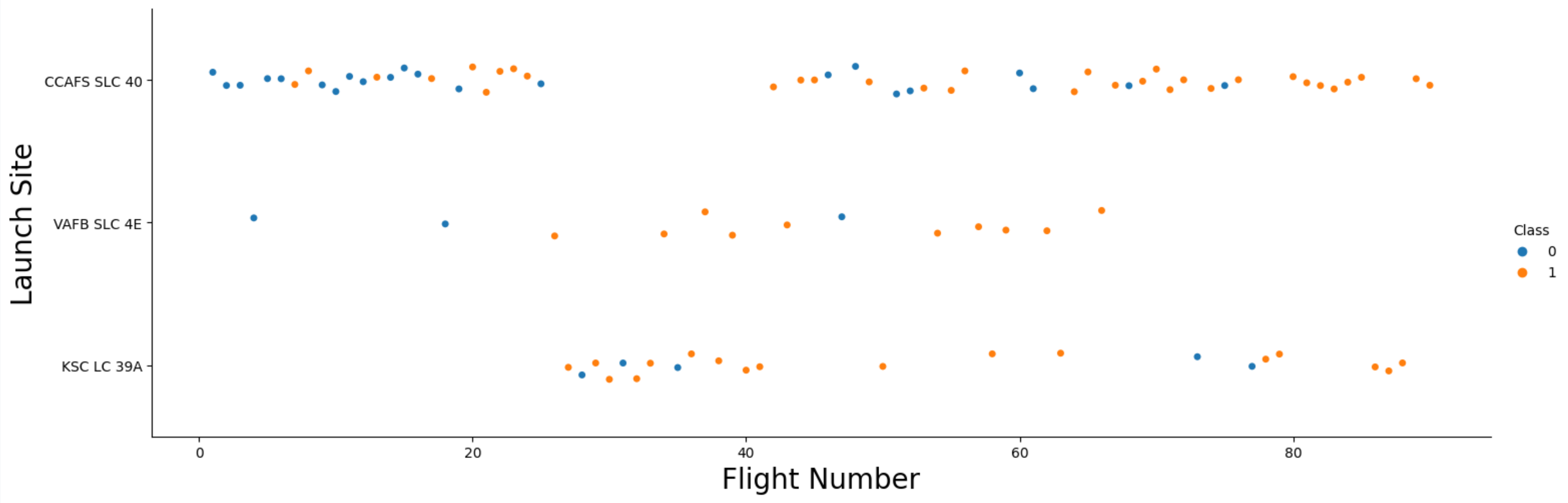
- We used different Machine Learning algorithms to build and test predictive models
- In this case, we used supervised learning to build classification models, translating the input variables into a binary outcome for a launch (success / fail)
- We divided the dataset into train and test parts, and looked for optimized model parameters using a grid search
- Finally, we used the best model to gain further insights into SpaceX's success factors

https://github.com/memology73/Assignments_IBM_DS/blob/main/Course_10-Capstone/Notebooks/7_SpaceX_Machine_Learning_Prediction_Part_5.jupyterlite_AM.ipynb

Section 2

Insights drawn from EDA

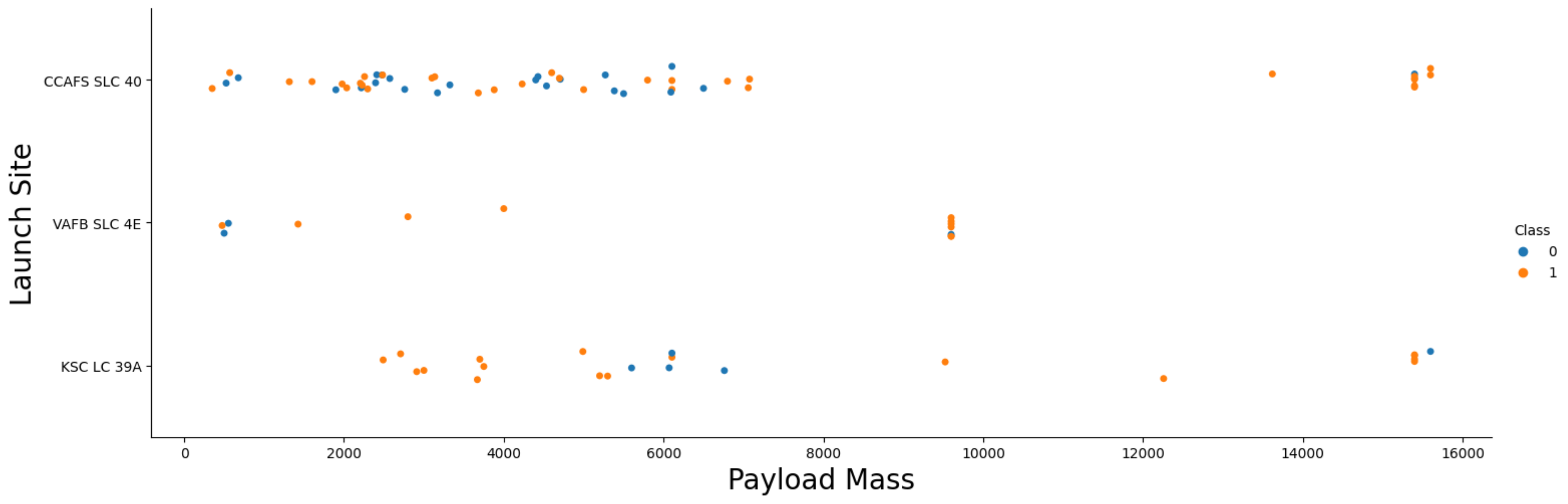
Flight Number vs. Launch Site



- Most Falcon 9 flights were launched from its sites in the East Coast (CCAFS and KSC), with some flights from California (VAFB)
- The color code provides some insight on SpaceX's learning curve: there are more blue dots (= failed launches) to the left, corresponding to the earlier flights
- More about these locations later in this presentation

Source: https://github.com/memology73/Assignments_IBM_DS/blob/main/Course_10-Capstone/Notebooks/5_jupyter-labs-eda-dataviz_AM.ipynb

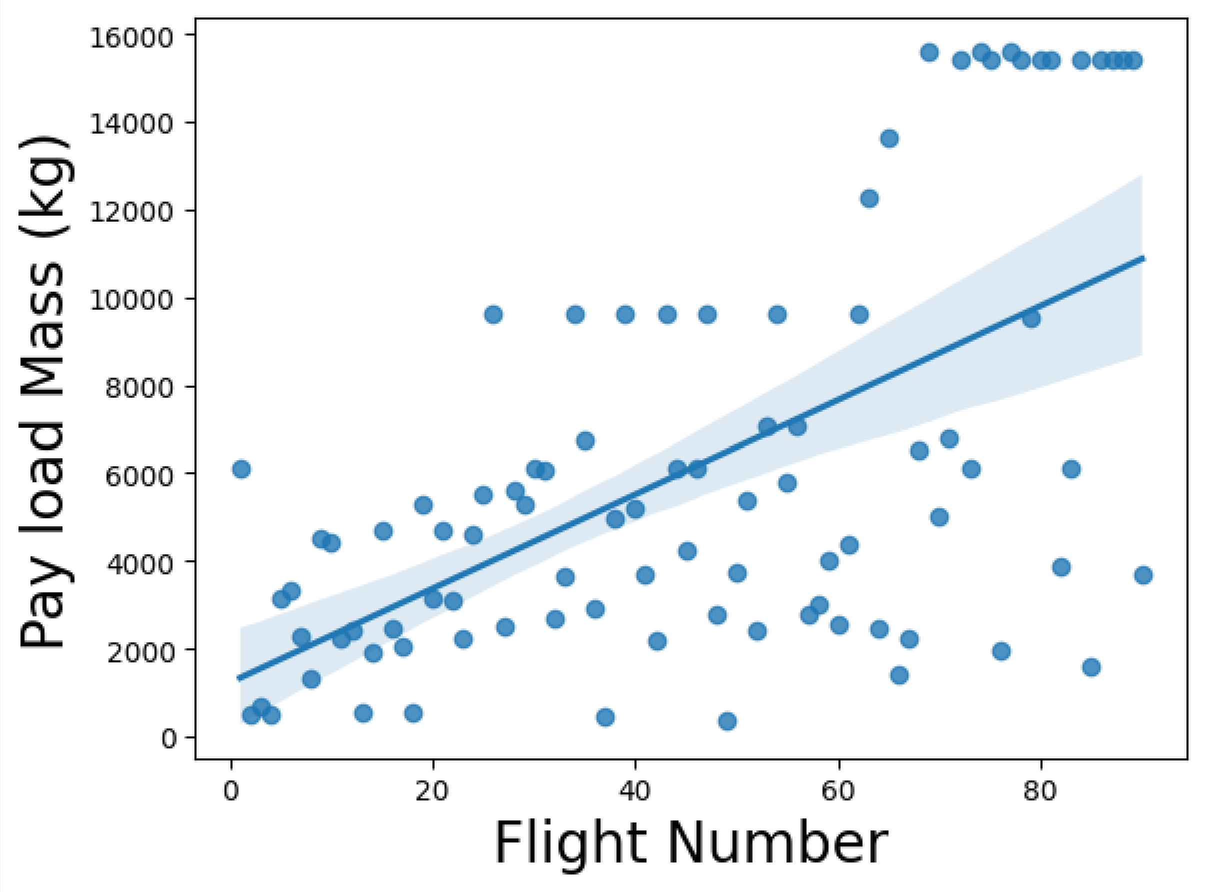
Payload vs. Launch Site



- Flights from VAFB (California) have payloads up to ca. 10 metric tons
- The sites in the East Coast (CCASF and KSC) are used for all payloads
- No visibly obvious correlation between site, payload and success rate of a launch

Source: https://github.com/memology73/Assignments_IBM_DS/blob/main/Course_10-Capstone/Notebooks/5_jupyter-labs-eda-dataviz_AM.ipynb

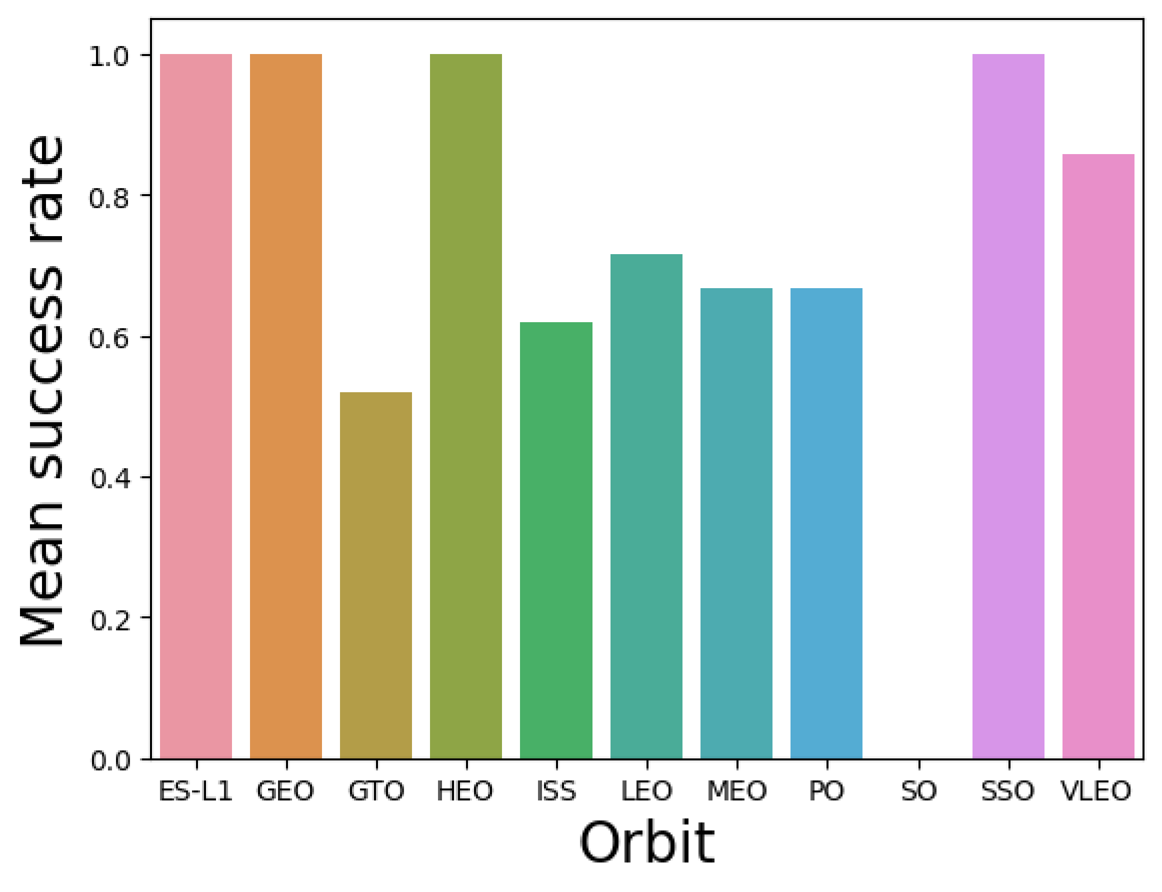
Flight Number vs. Payload (Extra)



- The graph clearly shows an upward trend for the payload mass with increasing flight number
  - Within 10 years, the average payload per launch grew more than 5x, from less than 2000 kg to over 10000 kg
  - In other words: SpaceX learned over the years how to transport more payload per launch
  - This is a key factor to reduce specific launch costs (cost per kg of launched material), making them more competitive

https://github.com/memology73/Assignments_IBM_DS/blob/main/Course_10-Capstone/Notebooks/8_Extra_notebook-EDA-ML_AM.ipynb

Success Rate vs. Orbit Type

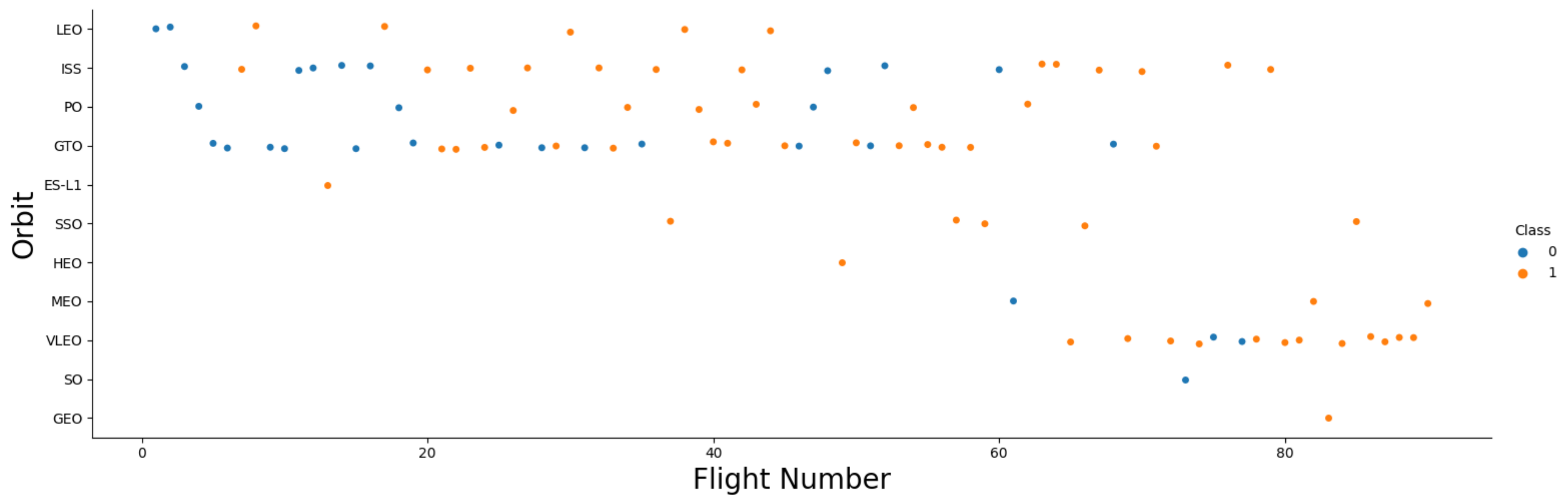


- ES-L1: Lagrange point 1
- GEO and GTO: geosynchronous orbits
- HEO: highly elliptical orbit
- ISS: International Space Station
- LEO: Low Earth orbit
- MEO: Geocentric intermediate circular orbit.
- PO: orbit that passes above or nearly above both poles
- SO and SSO: Sun-synchronous orbit
- VLEO: Very Low Earth Orbits

- Falcon 9 launches were more successful to some orbits than others
- The graph does not show any obvious correlation between a successful launch and the altitude /type of orbit
- NB: Success in this context means that the mission was completed, and the booster was safely brought back for re-use

Source: https://github.com/memology73/Assignments_IBM_DS/blob/main/Course_10-Capstone/Notebooks/5_jupyter-labs-eda-dataviz_AM.ipynb

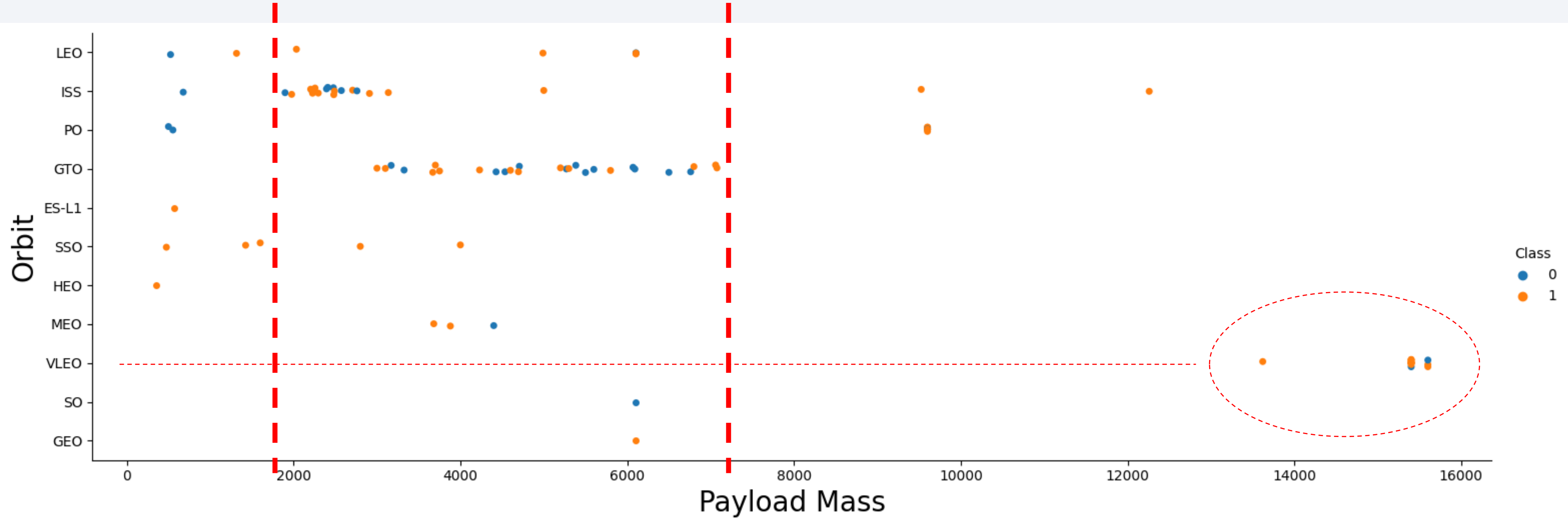
Flight Number vs. Orbit Type



- Most Falcon 9 flights in its early days were focused on LEO, ISS, PO and GTO orbits
- As Falcon 9 launches got more successful, it diversified its flights to other orbits
- In later years, most flights were focused on Very Low Earth Orbits (VLEO)
  - Likely connected to launching more internet service satellites (to be investigated)

Source: https://github.com/memology73/Assignments_IBM_DS/blob/main/Course_10-Capstone/Notebooks/5_jupyter-labs-eda-dataviz_AM.ipynb

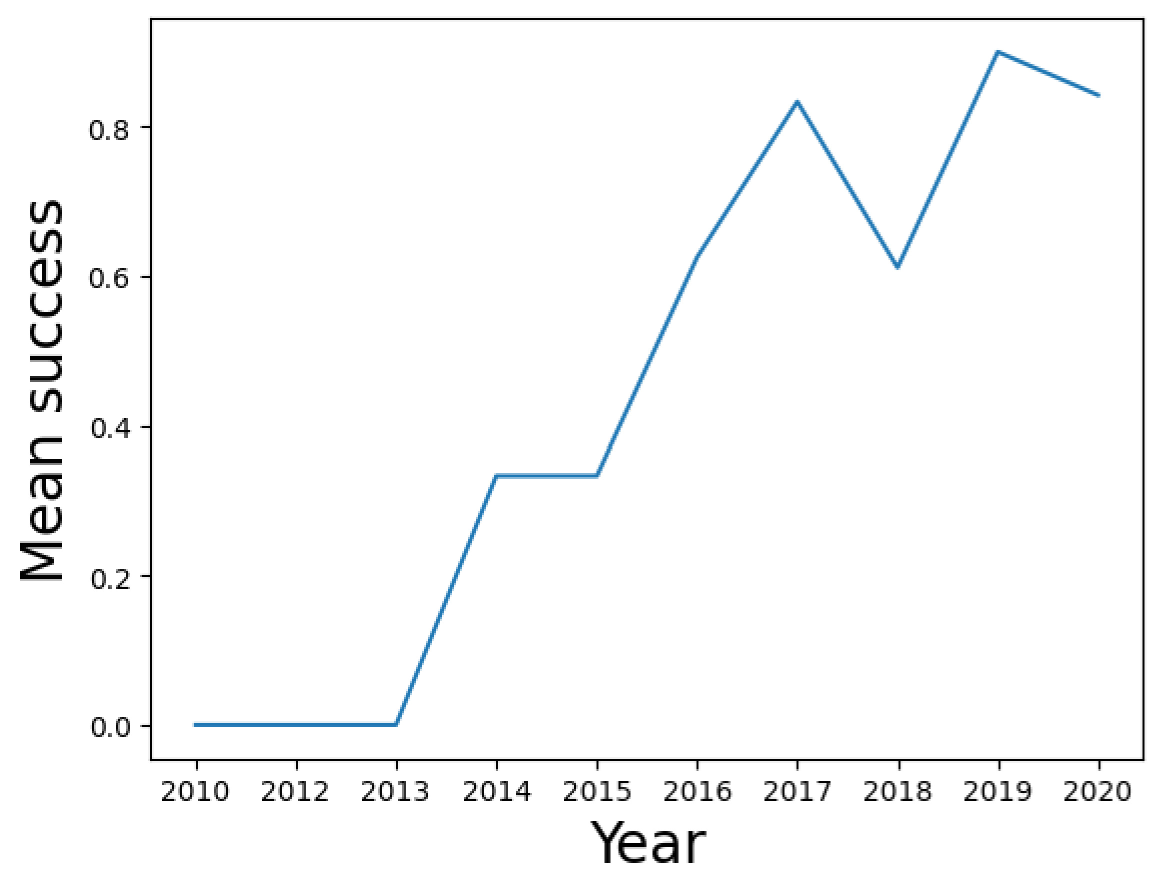
Payload vs. Orbit Type



- There is a wide distribution of payloads, however the majority remains in the range between 2000 and 7000 kg per launch
- Payloads to the ISS were mostly between ~2000 and ~3000 kg
- Payloads to GTO had a wider range, between ~3000 and ~7000 kg
- The heaviest payloads were delivered at VLEO

Source: https://github.com/memology73/Assignments_IBM_DS/blob/main/Course_10-Capstone/Notebooks/5_jupyter-labs-eda-dataviz_AM.ipynb

Launch Success Yearly Trend



- Looking at the mean Falcon 9 launch success on a yearly basis shows a clear upward trend
- This is what one calls a "learning curve": as time passes and SpaceX does more launches, there is a cumulative learning on past errors that are avoided in future launches, increasing the chance of future success
- More comments on SpaceX's learning curve at the end of this report
- NB: Success in this context means that the mission was completed, and the booster was safely brought back for re-use

Source: https://github.com/memology73/Assignments_IBM_DS/blob/main/Course_10-Capstone/Notebooks/5_jupyter-labs-eda-dataviz_AM.ipynb

Short Note on the Data Frame for SQL Queries

```
import pandas as pd
df = pd.read_csv("https://cf-courses-data.s3.
df.to_sql("SPACEXTBL", con, if_exists='replac
```

999

```
df.head()
```

	Date	Time (UTC)	Booster_Version	Launch_S
0	06/04/2010	:45:00	F9 v1.0 B0003	CCAFS

In order to make the SQL queries involving dates easier to work with, I used pandas to change the date formats of the column "Date"

```
df['Date'] = pd.to_datetime(df['Date'], dayfirst=True)
```

```
df.head()
```

	Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing_Outcor
0	2010-04-06	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0.0	LEO	SpaceX	Success	Failure (parachu
1	2010-08-12	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight C1, two CubeSats, barrel of...	0.0	LEO (ISS)	NASA (COTS) NRO	Success	Failure (parachu
2	2012-05-22	7:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525.0	LEO (ISS)	NASA (COTS)	Success	No attem
3	2012-08-10	0:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500.0	LEO (ISS)	NASA (CRS)	Success	No attem
4	2013-01-03	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677.0	LEO (ISS)	NASA (CRS)	Success	No attem

```
df.to_sql("SPACEXTBL", con, if_exists='replace', index=False,method="multi")
```

999

https://github.com/memology73/Assignments_IBM_DS/blob/main/Course_10-Capstone/Notebooks/4_jupyter-labs-eda-sql-coursera_sqllite_AM.ipynb

All Launch Site Names

```
%%sql
SELECT DISTINCT "Launch_Site" FROM SPACEXTBL
```

```
 * sqlite:///my_data1.db
Done.
```

Launch_Site
CCAFS LC-40
VAFB SLC-4E
KSC LC-39A
CCAFS SLC-40
None

Based on our database query:

- Falcon 9 was launched from 4 distinct sites (see names above)

https://github.com/memology73/Assignments_IBM_DS/blob/main/Course_10-Capstone/Notebooks/4_jupyter-labs-eda-sql-coursera_sqllite_AM.ipynb

Launch Site Names Begin with 'CCA'

```
%%sql
SELECT * FROM SPACEXTBL WHERE "Launch_Site" LIKE 'CCA%' LIMIT 5
```

Here showing only the top 5 results

```
* sqlite:///my_data1.db
Done.
```

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing_Outc
06/04/2010	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0.0	LEO	SpaceX	Success	Failure (paracl
12/08/2010	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight C1, two CubeSats, barrel of Brouere cheese	0.0	LEO (ISS)	NASA (COTS) NRO	Success	Failure (paracl
22/05/2012	7:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525.0	LEO (ISS)	NASA (COTS)	Success	No att
10/08/2012	0:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500.0	LEO (ISS)	NASA (CRS)	Success	No att
03/01/2013	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677.0	LEO (ISS)	NASA (CRS)	Success	No att

Total Payload Mass

```
%%sql
SELECT SUM("PAYLOAD_MASS__KG_") FROM SPACEXTBL WHERE "Customer"='NASA (CRS)'
```

```
 * sqlite:///my_data1.db
Done.
```

SUM("PAYLOAD_MASS__KG_")
45596.0

Based on our database query:

- Falcon 9 transported a total payload of ~46 metric tons of material to the space from NASA

https://github.com/memology73/Assignments_IBM_DS/blob/main/Course_10-Capstone/Notebooks/4_jupyter-labs-eda-sql-coursera_sqllite_AM.ipynb

Average Payload Mass by F9 v1.1

```
%%sql
SELECT AVG("PAYLOAD_MASS__KG_") FROM SPACEXTBL WHERE "Booster_Version" LIKE 'F9 v1.1%'
```

```
 * sqlite:///my_data1.db
Done.
```

AVG("PAYLOAD_MASS__KG_")
2534.6666666666665

Based on our database query:

- Falcon 9 v1.1 transported on average a payload of ~2.5 metric tons per launch

First Successful Ground Landing Date

```
%%sql
SELECT MIN(DATE("Date")) FROM SPACEXTBL WHERE "Landing_Outcome" = 'Success (ground pad)'
```

```
 * sqlite:///my_data1.db
Done.
```

MIN(DATE("Date"))
2015-12-22

Based on our database query:

- Falcon 9 had its first successful landing on a ground pad on December 22, 2015

https://github.com/memology73/Assignments_IBM_DS/blob/main/Course_10-Capstone/Notebooks/4_jupyter-labs-eda-sql-coursera_sqllite_AM.ipynb

Successful Drone Ship Landing with Payload between 4000 and 6000

```
%%sql
SELECT "Booster_Version" FROM
    (SELECT * FROM SPACEXTBL WHERE "PAYLOAD_MASS__KG_" BETWEEN 4000 AND 6000)
    WHERE "Landing_Outcome"='Success (drone ship)'
```

```
 * sqlite:///my_data1.db
Done.
```

Booster_Version
F9 FT B1022
F9 FT B1026
F9 FT B1021.2
F9 FT B1031.2

Based on our database query:

- There were 4 different Falcon 9 boosters (see names above) that successfully launched a payload between 4 and 6 metric tons and returned successfully, landing on a drone ship

https://github.com/memology73/Assignments_IBM_DS/blob/main/Course_10-Capstone/Notebooks/4_jupyter-labs-eda-sql-coursera_sqllite_AM.ipynb

Total Number of Successful and Failure Mission Outcomes

```
%%sql
SELECT COUNT(*) FROM SPACEXTBL WHERE "Mission_Outcome" LIKE 'Success%'
```

```
 * sqlite:///my_data1.db
Done.
```

COUNT(*)
100

```
%%sql
SELECT COUNT(*) FROM SPACEXTBL WHERE "Mission_Outcome" LIKE 'Failure%'
```

```
 * sqlite:///my_data1.db
Done.
```

COUNT(*)
1

- Note that in this database, these are two columns related to outcome: “mission outcome” and “landing outcome”. This query only looks at the success of the mission.
- In other parts of this report the measure of success is a combination of mission and landing outcomes.
- This explains why here the success rate is much higher than, for instance, what we previously saw under the topic “learning curve”

Based on our database query:

- There were 100 successful missions versus 1 failed mission

https://github.com/memology73/Assignments_IBM_DS/blob/main/Course_10-Capstone/Notebooks/4_jupyter-labs-eda-sql-coursera_sqllite_AM.ipynb

Boosters Carried Maximum Payload

```
%%sql
SELECT DISTINCT("Booster_Version") FROM SPACEXTBL WHERE
    "PAYLOAD_MASS__KG_" = (SELECT MAX("PAYLOAD_MASS__KG_") FROM SPACEXTBL)
```

```
 * sqlite:///my_data1.db
Done.
```

Booster_Version
F9 B5 B1048.4
F9 B5 B1049.4
F9 B5 B1051.3
F9 B5 B1056.4
F9 B5 B1048.5
F9 B5 B1051.4
F9 B5 B1049.5
F9 B5 B1060.2
F9 B5 B1058.3
F9 B5 B1051.6
F9 B5 B1060.3
F9 B5 B1049.7

Based on our database query:

- The listed Falcon 9 booster versions transported the maximum payload at some point in the past

https://github.com/memology73/Assignments_IBM_DS/blob/main/Course_10-Capstone/Notebooks/4_jupyter-labs-eda-sql-coursera_sqllite_AM.ipynb

2015 Launch Records

```
%%sql
SELECT "YYYY" AS 'Year', STRFTIME('%m', DATE("Date")) AS 'Month', "Landing_Outcome",
    "Booster_Version", "Launch_Site", "YYYY" AS 'Year' FROM
        (SELECT *, STRFTIME('%Y', DATE("Date")) AS 'YYYY' FROM SPACEXTBL)
        WHERE "YYYY" = '2015' AND "Landing_Outcome" = 'Failure (drone ship)'
```

```
 * sqlite:///my_data1.db
Done.
```

Year	Month	Landing_Outcome	Booster_Version	Launch_Site	Year_1
2015	10	Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40	2015
2015	04	Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40	2015

Based on our database query:

- In 2015 there were 2 failed landing outcomes, one in April and another in October. In both case the launch site was CCAFS LC-40

https://github.com/memology73/Assignments_IBM_DS/blob/main/Course_10-Capstone/Notebooks/4_jupyter-labs-eda-sql-coursera_sqllite_AM.ipynb

Landing Successful Outcomes Between 2010-06-04 and 2017-03-20

```
%%sql
SELECT * FROM
    (SELECT * FROM SPACEXTBL WHERE "Date" BETWEEN '2010-06-04' AND '2017-03-20')
        WHERE "Landing_Outcome" LIKE 'Succ%'
        ORDER BY "Date" DESC
```

```
* sqlite:///my_data1.db
Done.
```

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing_Ou
2017-03-06 00:00:00	21:07:00	F9 FT B1035.1	KSC LC-39A	SpaceX CRS-11	2708.0	LEO (ISS)	NASA (CRS)	Success	Success (g
2017-02-19 00:00:00	14:39:00	F9 FT B1031.1	KSC LC-39A	SpaceX CRS-10	2490.0	LEO (ISS)	NASA (CRS)	Success	Success (g
2017-01-14 00:00:00	17:54:00	F9 FT B1029.1	VAFB SLC-4E	Iridium NEXT 1	9600.0	Polar LEO	Iridium Communications	Success	Success
2017-01-05 00:00:00	11:15:00	F9 FT B1032.1	KSC LC-39A	NROL-76	5300.0	LEO	NRO	Success	Success (g
2016-08-14 00:00:00	5:26:00	F9 FT B1026	CCAFS LC-40	JCSAT-16	4600.0	GTO	SKY Perfect JSAT Group	Success	Success
2016-08-04 00:00:00	20:43:00	F9 FT B1021.1	CCAFS LC-40	SpaceX CRS-8	3136.0	LEO (ISS)	NASA (CRS)	Success	Success
2016-07-18 00:00:00	4:45:00	F9 FT B1025.1	CCAFS LC-40	SpaceX CRS-9	2257.0	LEO (ISS)	NASA (CRS)	Success	Success (g
2016-06-05 00:00:00	5:21:00	F9 FT B1022	CCAFS LC-40	JCSAT-14	4696.0	GTO	SKY Perfect JSAT Group	Success	Success
2016-05-27 00:00:00	21:39:00	F9 FT B1023.1	CCAFS LC-40	Thaicom 8	3100.0	GTO	Thaicom	Success	Success
2015-12-22 00:00:00	1:29:00	F9 FT B1019	CCAFS LC-40	OG2 Mission 2 11 Orbcomm-OG2 satellites	2034.0	LEO	Orbcomm	Success	Success (g


```
%%sql
SELECT * FROM
    (SELECT * FROM SPACEXTBL WHERE "Date" BETWEEN '2010-06-04' AND '2017-03-20')
        WHERE "Landing_Outcome" LIKE 'Succ%'
        ORDER BY "Date" DESC
```


Based on our database query:

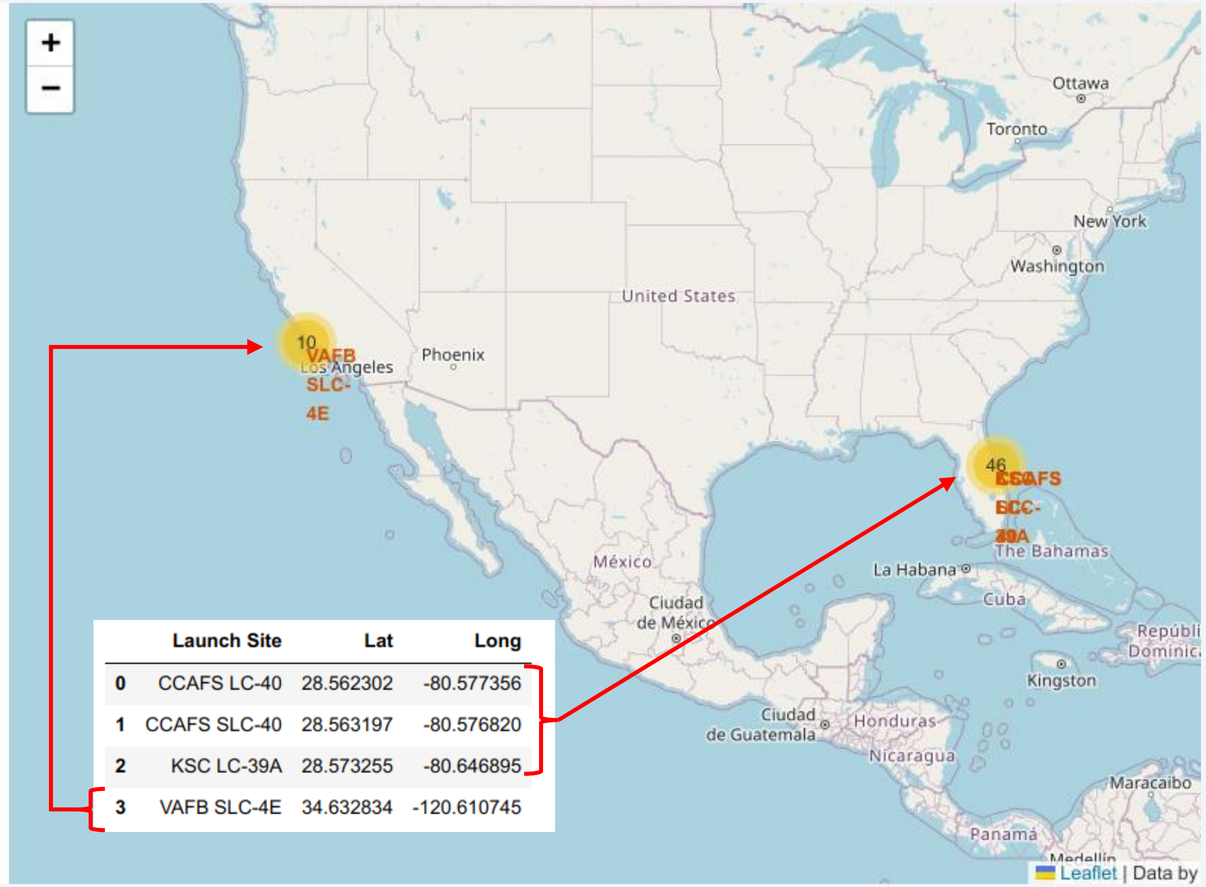
- In the period between June 4, 2010, and March 20, 2017, there were 10 successful landing outcomes
    - Thereof 5 were on gound pads, the 5 on drone ships (not visible in the screenshot)

https://github.com/memology73/Assignments_IBM_DS/blob/main/Course_10-Capstone/Notebooks/4_jupyter-labs-eda-sql-coursera_sqllite_AM.ipynb

Section 3

Launch Sites Proximities Analysis

Launch site locations



	Launch Site	Lat	Long
0	CCAFS LC-40	28.562302	-80.577356
1	CCAFS SLC-40	28.563197	-80.576820
2	KSC LC-39A	28.573255	-80.646895
3	VAFB SLC-4E	34.632834	-120.610745

- Falcon 9 was launched from 4 sites: 3 are located on the East Coast (Florida) and 1 in the West Coast (California)
- Being close to the equator is advantageous for launching: all sites are located as close as possible to the equator, while remaining within US territory
- This is a data subset containing 56 Falcon 9 launches
    - 46 launches from Florida (CCAFS LC40, CCAFS SLC-40 and KSC LC-39A)
    - 10 launches from California (VAFB)
- The company uses existing infrastructure, which reduces overall upfront investment

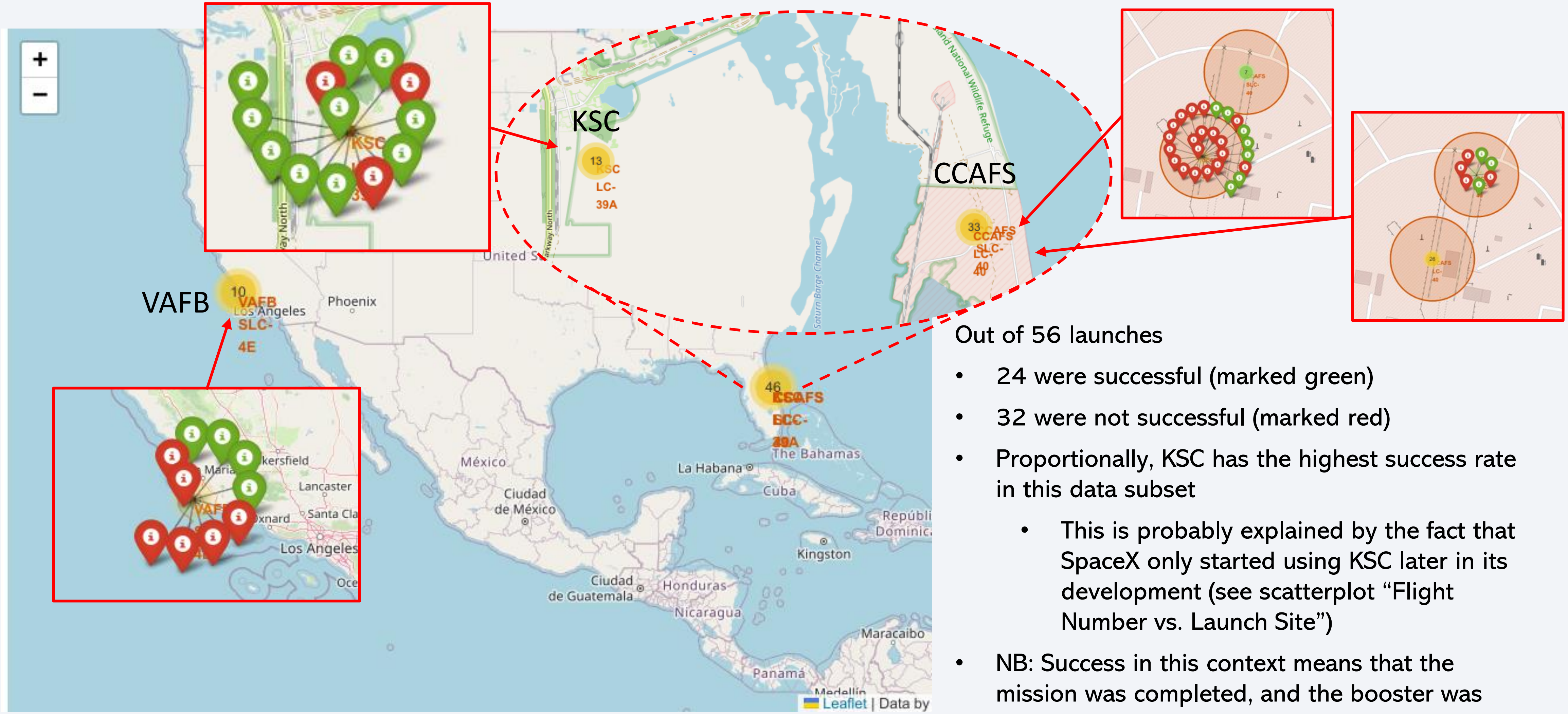
https://github.com/memology73/Assignments_IBM_DS/blob/main/Course_10-Capstone/Notebooks/6_lab_jupyter_launch_site_location_AM.ipynb

Zooming in – Successful vs. Failed launches

- Zooming into the map shows that the launches were distributed as follows
  - 33 launches from CCAFS (LC40 or SLC-40)
  - 13 launches from KSC
  - 10 launches from VAFB

https://github.com/memology73/Assignments_IBM_DS/blob/main/Course_10-Capstone/Notebooks/6_lab_jupyter_launch_site_location_AM.ipynb

Zooming in – Successful vs. Failed launches

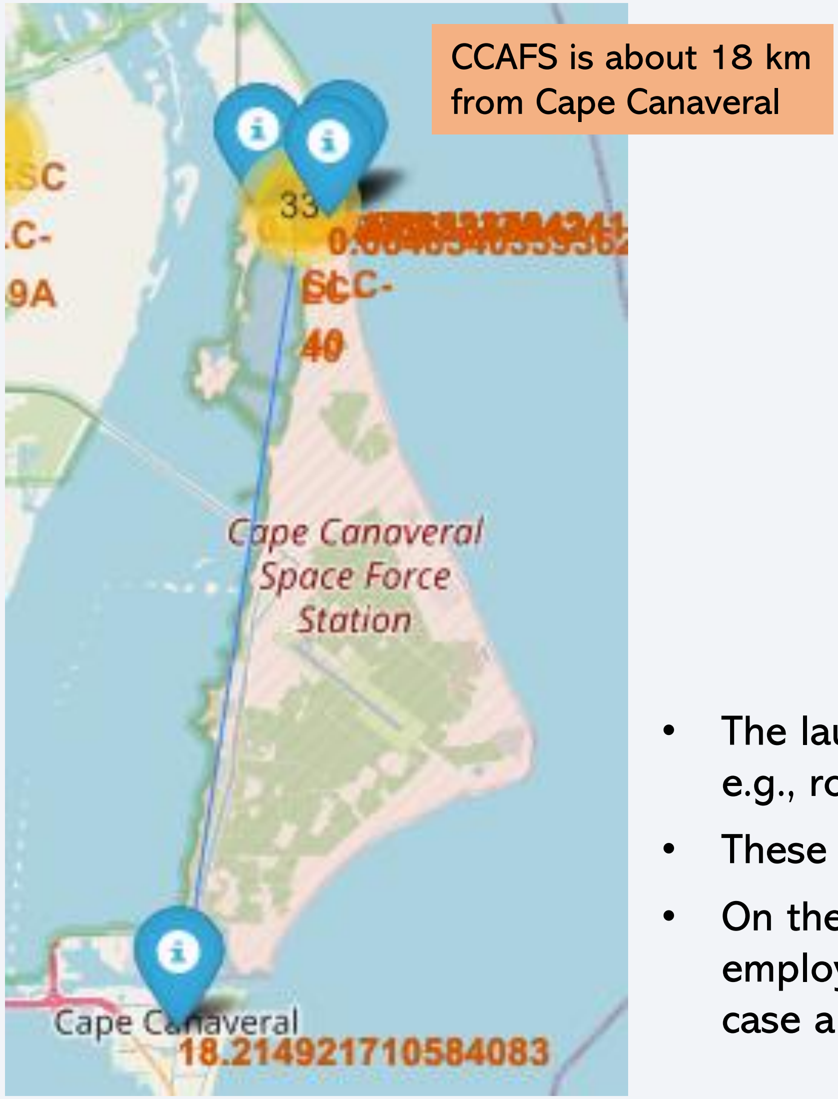


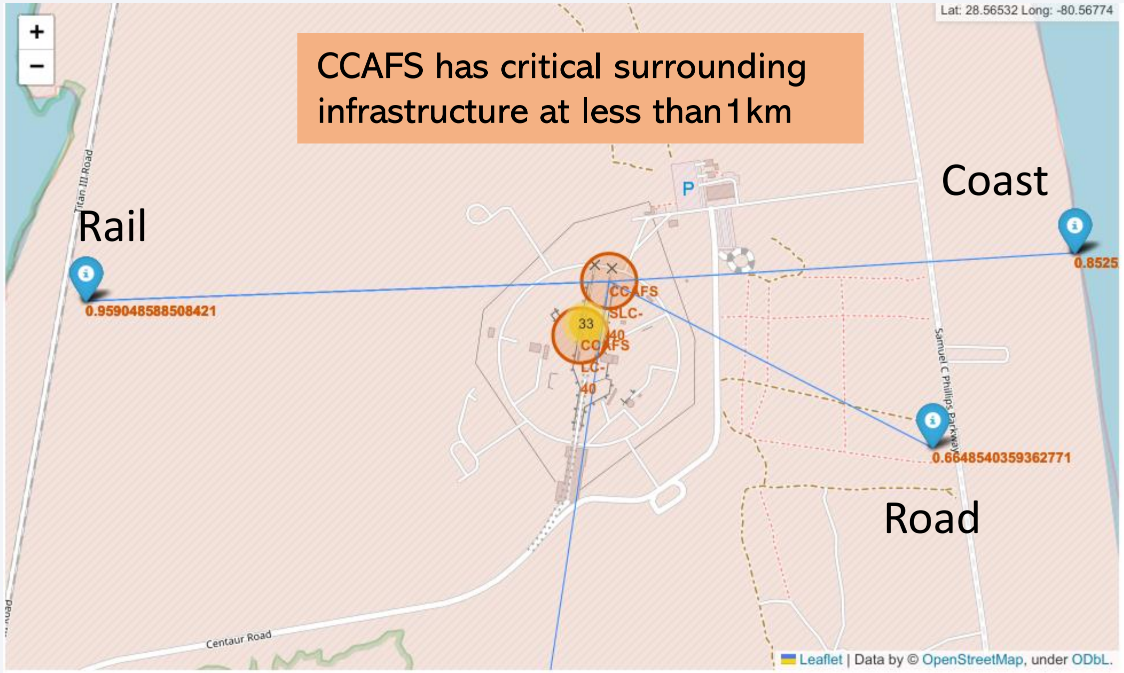
Out of 56 launches

- 24 were successful (marked green)
- 32 were not successful (marked red)
- Proportionally, KSC has the highest success rate in this data subset
  - This is probably explained by the fact that SpaceX only started using KSC later in its development (see scatterplot “Flight Number vs. Launch Site”)
- NB: Success in this context means that the mission was completed, and the booster was safely brought back for re-use

https://github.com/memology73/Assignments_IBM_DS/blob/main/Course_10-Capstone/Notebooks/6_lab_jupyter_launch_site_location_AM.ipynb

Launch Site: Local Infrastructure (Example)





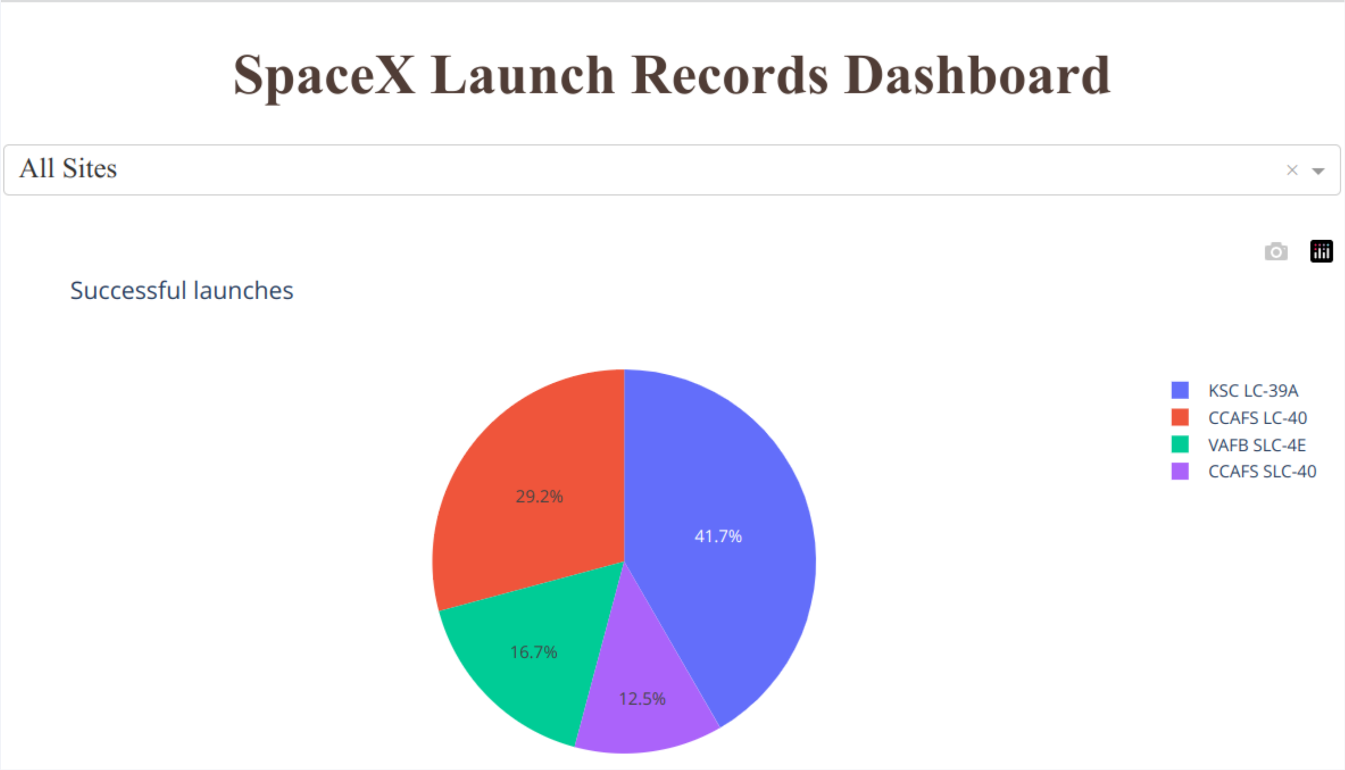
- The launch sites are close to critical infrastructure (here CCAFS as an example), e.g., roads and railway at less than 1 km from the launch site
- These are needed to transport heavy equipment
- On the other hand, the sites are close, but not too close, to towns –site employees need to live close, but the towns need also to be at a safe distance in case a launch catastrophically fails

https://github.com/memology73/Assignments_IBM_DS/blob/main/Course_10-Capstone/Notebooks/6_lab_jupyter_launch_site_location_AM.ipynb

Section 4

Build a Dashboard with Plotly Dash

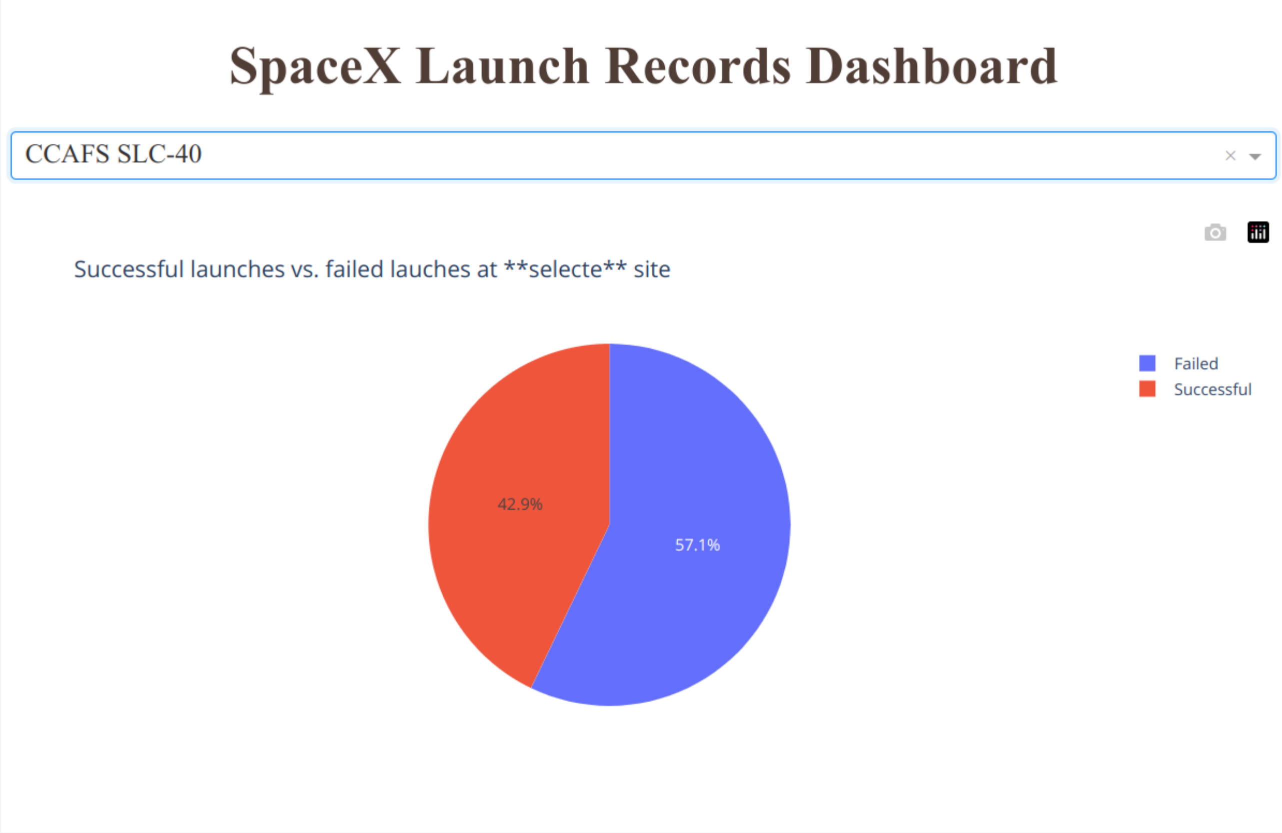
Interactive Dashboard App



- The interactive dashboard shows SpaceX's launch record success for all sites (total) as well as for a specific site
- The dashboard has 2 parts
  - A piechart on success launches (shown here)
  - A scatterplot showing payload and the success/failure (not visible in this screenshot)
- NB: Success in this context means that the mission was completed, and the booster was safely brought back for re-use

https://github.com/memology73/Assignments_IBM_DS/tree/main/Course_10-Capstone/Dashboard_code

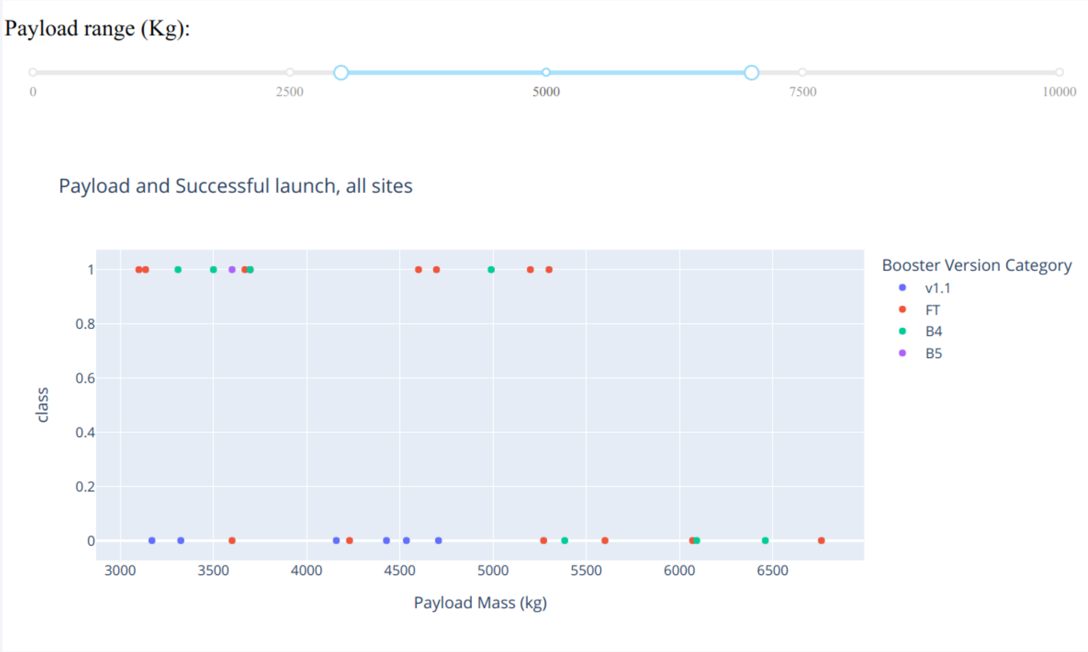
Interactive Dashboard App: Exploring SpaceX's Data



- The interactive dashboard can also show SpaceX's launch record success for a specific site
- Here as screenshot showing the site with the highest launch success rate in this dataset (CCAFS SLC-40)
- This is in line with the data shown at the scatterplot "Flight Number vs. Launch Site"

https://github.com/memology73/Assignments_IBM_DS/tree/main/Course_10-Capstone/Dashboard_code

Interactive Dashboard App: Exploring SpaceX's Data



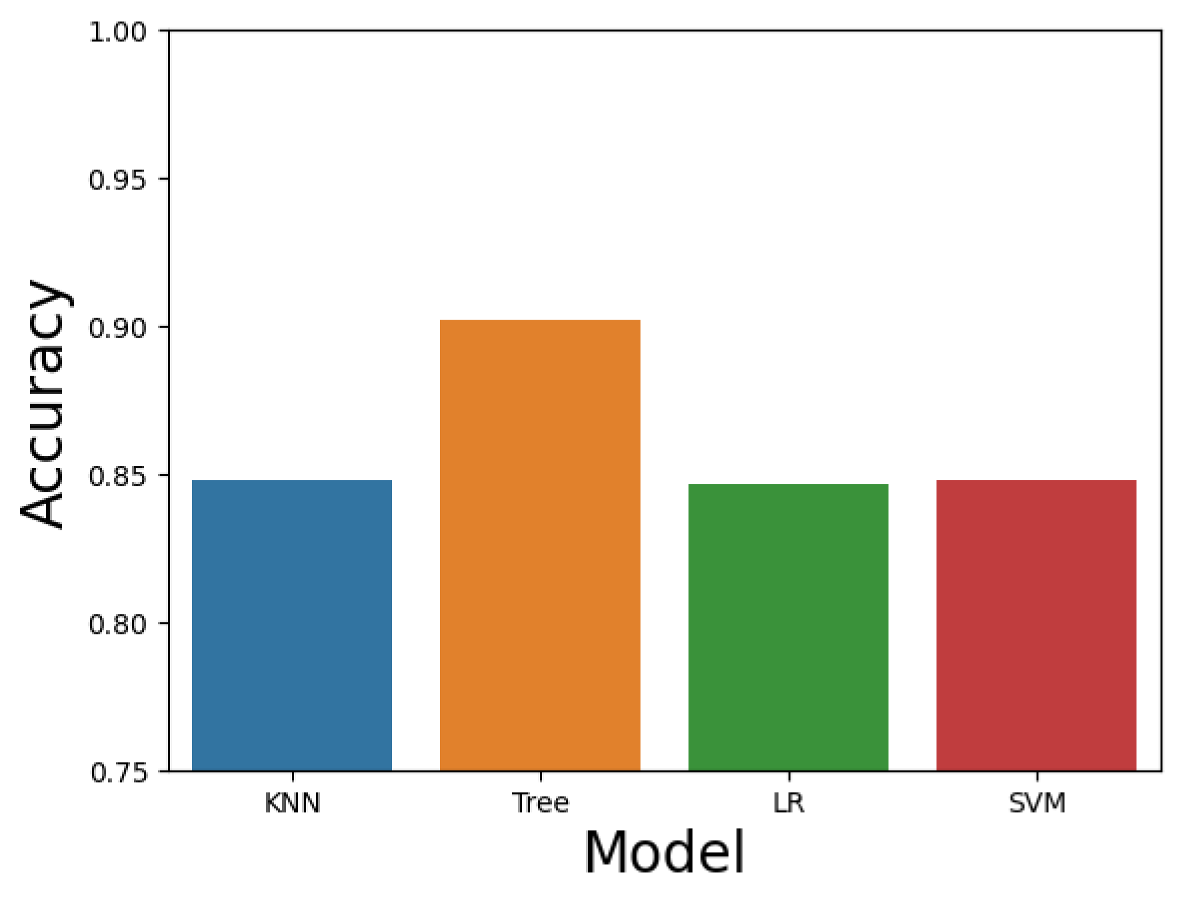
- At the bottom of the interactive dashboard one can see SpaceX's launch record success different payload ranges
- Here as screenshot showing the launch outcome for all payloads launched (left) and for a specific range, between 3000 and 7000 kg (right)
- There is no obvious visible correlation between success and payload, or success and booster version used

https://github.com/memology73/Assignments_IBM_DS/tree/main/Course_10-Capstone/Dashboard_code

Section 5

Predictive Analysis (Classification)

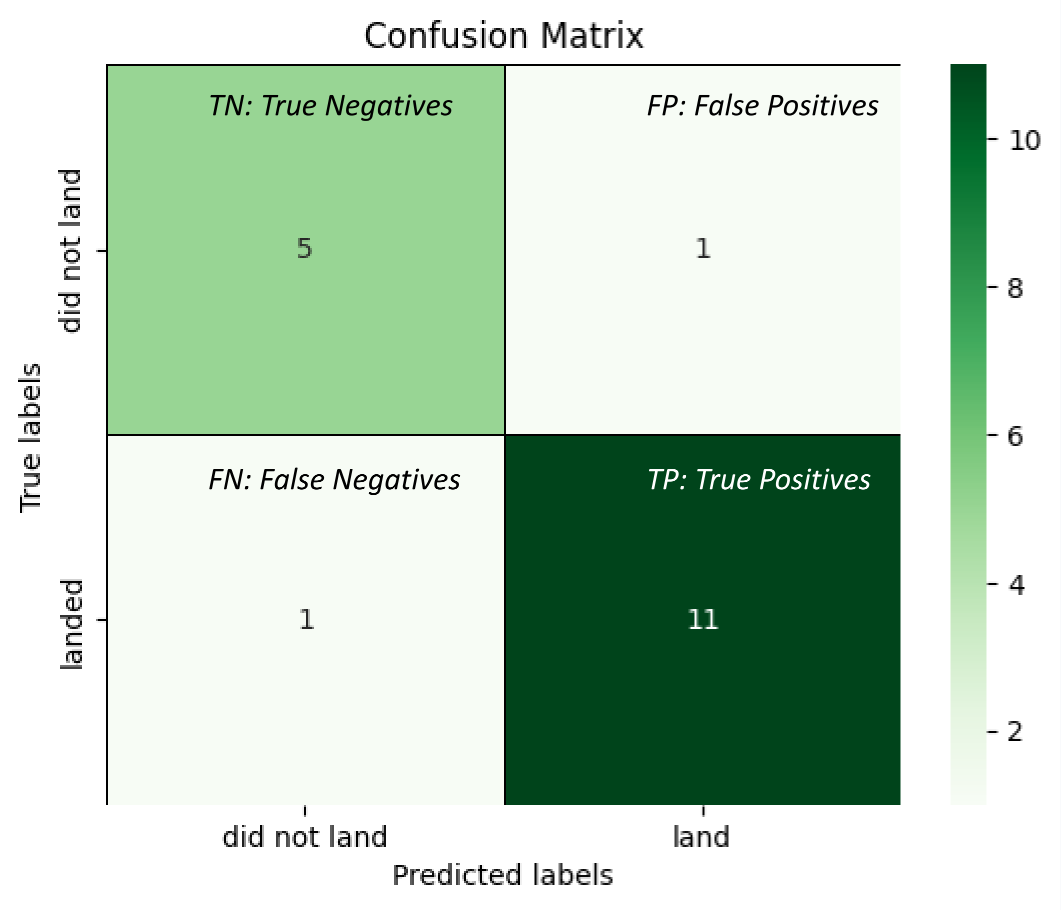
Classification Accuracy



- We used different models to build a predictive tools of a successful outcome
  - Models: (1) KNN, (2) Decision Tree, (3) Logistic Regression and (4) Support Vector Machine
- We divided the dataset into training and a testing sets, at a proportion of 80:20, meaning we used 80% of the data for training and 20% for testing the model
- All optimized models had similar accuracies
- However, the Decision Tree model had a slightly higher accuracy… and a key advantage (see next slide)

https://github.com/memology73/Assignments_IBM_DS/blob/main/Course_10-Capstone/Notebooks/7_SpaceX_Machine_Learning_Prediction_Part_5.jupyterlite_AM.ipynb

Tree Model: Confusion Matrix

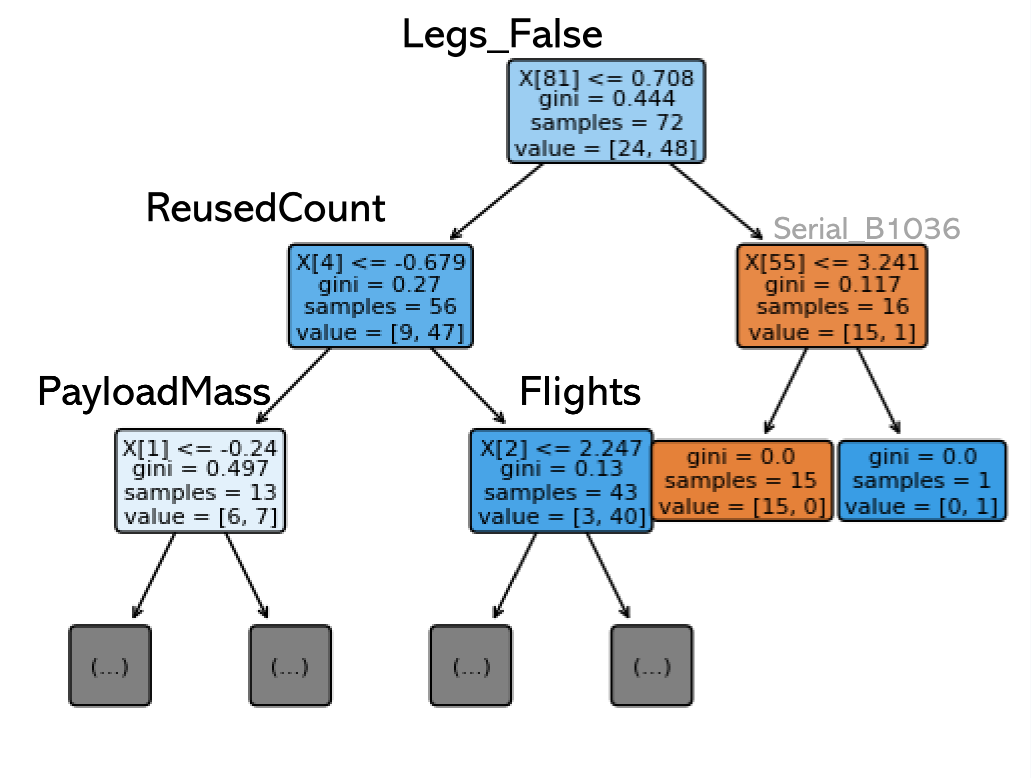



```
print("tuned hpyerparameters :(best parameters) ",tree_cv.best_params_)
print("accuracy :",tree_cv.best_score_)
```

```
tuned hpyerparameters :(best parameters)  {'criterion': 'gini', 'max_depth': 4, 'max_features': 'sqrt', 'min_samples_leaf': 1,
'min_samples_split': 2, 'splitter': 'random'}
accuracy : 0.9017857142857144
```

- The critical point: the Decision Tree had the lowest False Positives (FP), with 1 FP vs 3 FPs in each of the other models
- FP in this context means that the model predicted “land,” but the actual data says “did not land”
- Minimizing FP is a key success factor for a predictive model
  - Predicting a successful landing while in reality it fails is a very costly mistake
  - FP minimization is thus a key economical factor
- Minimizing FN is also important (minimize missing opportunities to launch), but not as critical as minimizing FP

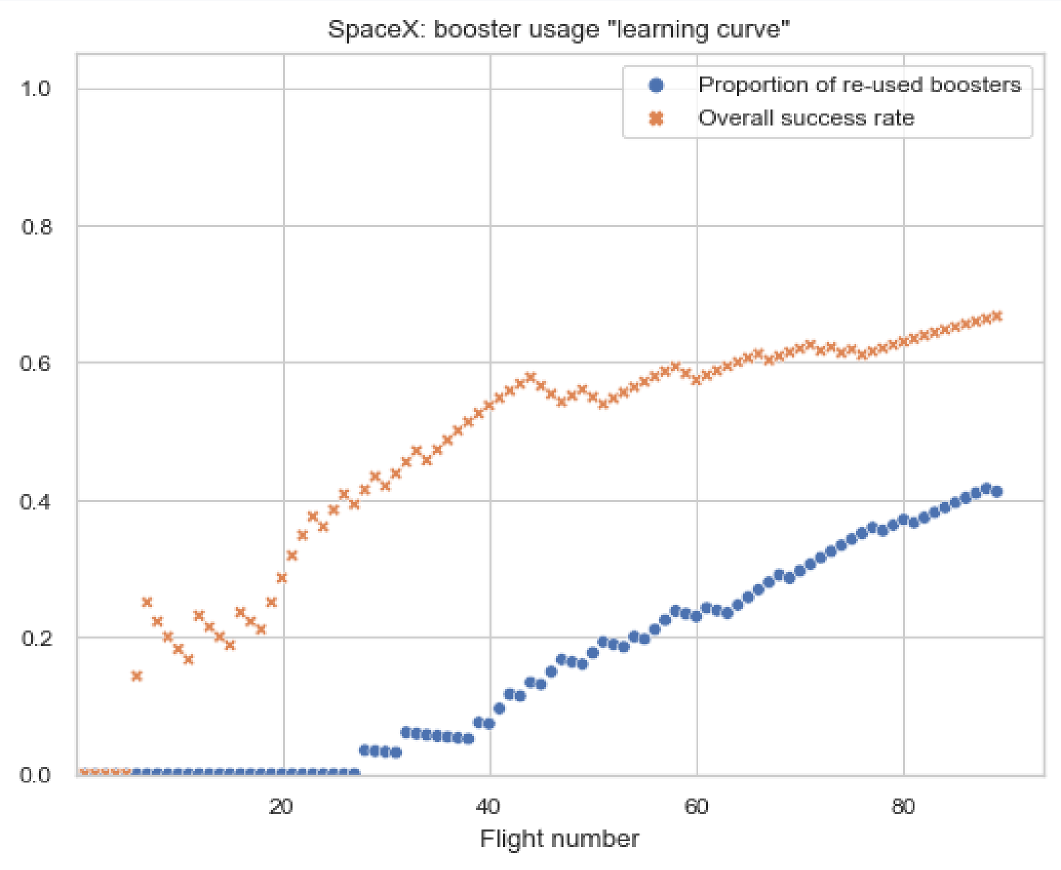
Tree Model: Extra Points



- We explored the optimized Decision Tree model a bit further, see graph
- Key top factors influencing the prediction outcome are
    - Does the booster have legs? (if no legs, cannot expect it to land safely, so it is a trivial parameter)
    - Was it re-used? Re-using a booster is unlikely to directly influence the outcome, but it is related to SpaceX's learning curve – see next 2 slides
    - Payload mass and flights (how many flights a booster had prior to the mission) are also top factors in predicting success, but again related to SpaceX's learning curve

https://github.com/memology73/Assignments_IBM_DS/blob/main/Course_10-Capstone/Notebooks/8_Extra_notebook-EDA-ML_AM.ipynb

Top Parameters and Learning Curve



- We built the curve for success rate as a function of flight number (learning curve, as mentioned earlier)
- It is visible that over time, the success rate (launch and recovery) goes up, and at some point, the proportion of re-used booster being in service also goes up
- Therefore, it should not be a surprise that the parameter “Reused-Count” (and the related “flight” parameter) are top predictive in the Decision Tree – after all, as the overall Falcon 9 success goes up at the same time
    - Classical case of “correlation” versus “causation”: because the launches are more successful – meaning, more booster landings – there are more boosters that can be successfully re-used, and so on

https://github.com/memology73/Assignments_IBM_DS/blob/main/Course_10-Capstone/Notebooks/8_Extra_notebook-EDA-ML_AM.ipynb

Conclusions

- SpaceX is a very successful space transportation company
  - Between 2010 and 2020, Falcon 9 transported ~550 metric tons of material to the space
  - Falcon 9 booster landing and re-use is a key success factor → reduces overall launch costs
  - The company uses existing facilities to launch their vehicles, which reduced upfront investment level
- In the analyzed period (2010-2020), there is a clear upward learning curve trend, with increasing rate of successful missions and booster recovery over time
  - The overall success rate for the analyzed period is at ~66% at the end of the learning curve, and about 40% of all launches are performed using recovered boosters
- The steady increase of payload per launch is another important commercial factor → reduces specific ($/kg) launch cost
  - Steady increase of payload per launch is part of SpaceX's learning curve and another important commercial factor → expect it to reduce specific ($/kg) launch costs
- Predictive model for launch outcomes: optimized Decision Tree model performed best
  - It showed the highest accuracy, and importantly, the least number of False Positives (FP), which is a costly prediction mistake (model predicts booster landing, but outcome is a failed landing)

Recommendations to SpaceY

- “Copy with pride”: do it like SpaceX
  - Focus on using reusable boosters
  - Maximize payload per launch
  - Use existing sites / infrastructure
- Use Machine Learning models to predict launch outcomes
  - Minimize failed launches

Thank you!

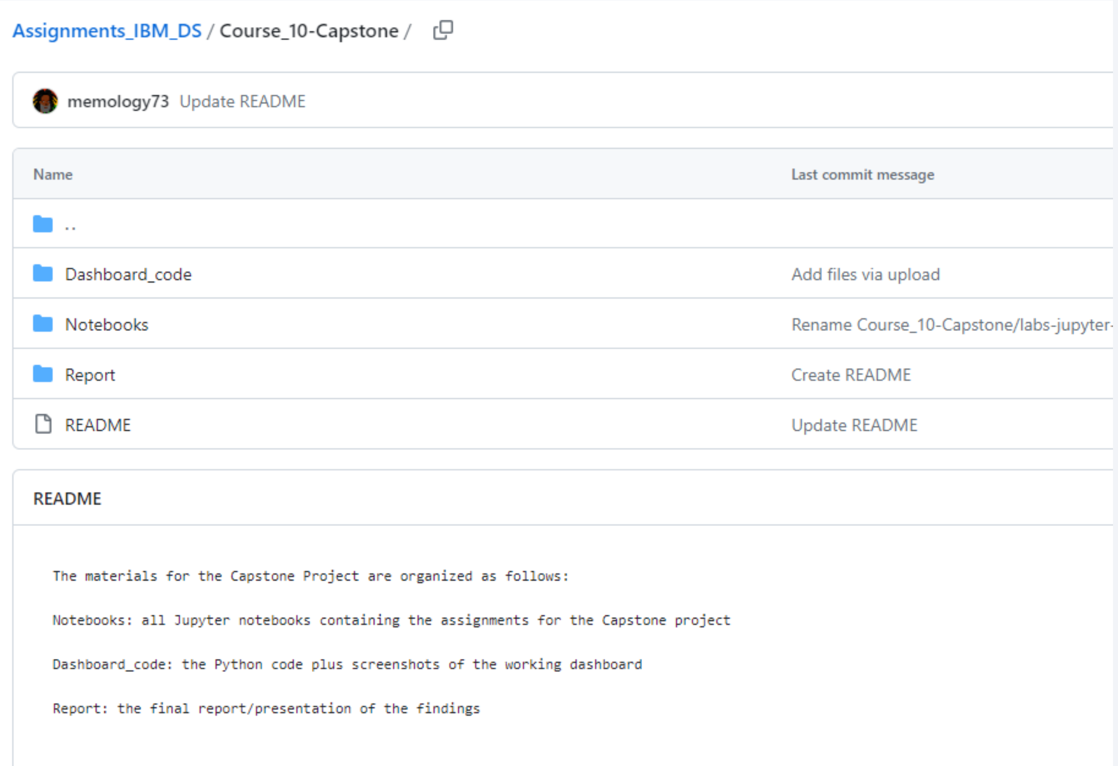
Appendix: GitHub Links (Reminder)

All materials related to the Capstone Project are under the following link

https://github.com/memology73/Assignments_IBM_DS/tree/main/Course_10-Capstone

- Notebooks: all Jupyter notebooks containing the assignments for the Capstone project
- Dashboard_code: the Python code plus screenshots of the working dashboard
- Report: the final report/presentation of the findings

GitHub links are also available at several slides, corresponding the slide's content



Appendix

Assignments_IBM_DS / Course_10-Capstone / Dashboard_code / spacex_dash_app_completed.py

memology73 Rename Course_10-Capstone/spacex_dash_app_completed.py to Course_10-C...

Code Blame 124 lines (102 loc) · 5.49 KB

```
# Import required libraries
#import pandas as pd
#import dash
#import dash_html_components as html
#import dash_core_components as dcc
#from dash.dependencies import Input, Output
#import plotly.express as px


import pandas as pd
import dash
from dash import html
from dash import dcc
from dash.dependencies import Input, Output, State
import plotly.express as px


# Read the airline data into pandas dataframe
spacex_df = pd.read_csv("spacex_launch_dash.csv")
max_payload = spacex_df['Payload Mass (kg)'].max()
min_payload = spacex_df['Payload Mass (kg)'].min()

# Create a dash application
app = dash.Dash(__name__)
```

Screenshot of dashboard code (Python)