



IBM Developer  
SKILLS NETWORK

# Winning Space Race with Data Science

Li Tang  
11/16/2022



# Outline

---

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

# Executive Summary

---

- Summary of methodologies
- 1. from problem to approach
- 2. working with the data
- 3. Deriving the answer
- Summary of all results

# Introduction

---

- Project background and context
- The commercial space age is here, companies are making space travel affordable for everyone. The most successful is SpaceX. Now SpaceY want to compete with it.
- Problems you want to find answers
- You will also determine if SpaceX will reuse the first stage.
- Instead of using rocket science to determine if the first stage will land successfully,
- you will train a machine learning model and use public information to predict if SpaceX
- will reuse the first stage.



Section 1

# Methodology

# Methodology

---

## Executive Summary

- Data collection methodology:
  - [Take a ride around in SpaceX AR](#)
- Perform data wrangling
  - Using Mean value of PayloadMass to replace it's the missing value
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - How to build, tune, evaluate classification models

# Data Collection

---

- Describe how data sets were collected.

1. Request and parse the SpaceX launch data using the GET request

2. Filter the DataFrame to only include Falcon 9 launches

3. Dealing with missing value (using mean value to replace the missing)

# Data Collection – SpaceX API

- Present your data collection with SpaceX REST calls using key phrases and flowcharts
- Add the GitHub URL of the completed SpaceX API calls notebook , as an external reference and peer-review purpose
- [https://github.com/tanglily08/IBM-data-science-final-project/blob/c8eb934d06a503b6ea7fb050e491a887669f15bd/spacex-collecting-date-ds-final1%20\(2\).ipynb](https://github.com/tanglily08/IBM-data-science-final-project/blob/c8eb934d06a503b6ea7fb050e491a887669f15bd/spacex-collecting-date-ds-final1%20(2).ipynb)

```
[7]: spacex_url="https://api.spacexdata.com/v4/launches/past"

[8]: response = requests.get(spacex_url)

Check the content of the response

[11]: print(response.content)

b[{"fairings":{"reused":false,"recovery_attempt":false,"recovered":false,"ships":[]},"links":{"patch":{"small":"https://images2.imgbox.com/94/f2/W6P4H5r_o.png","large":"https://images2.imgbox.com/5b/02/cwH0b5V_o.png"},"reddit":{"campaign":null,"launch":null,"media":null,"recovery":null},"flickr":{"small":"","original":[]},"presskit":null,"webcast":"https://www.youtube.com/watch?v=0a_00n1_Y80","youtube_id":"0a_00n1_Y80","article":"https://www.space.com/2236-spacex-inaugural-falcon-1-rocket-lost-launch.html","wikipedia":"https://en.wikipedia.org/wiki/DemoSat"},"static_fire_date_utc":"2006-03-17T00:00:00.000Z","static_fire_date_unix":1143535000,"net":false,"window":0,"rocket":"5e90b095e6d69959f708d1eb","success":false,"failures":[{"time":33,"altitude":null,"reason":"merlin engine failure"}],"details":{"engine_failure_at_33_seconds_and_loss_of_vehicle":"crew":[],"ships":[]},"capsules":{"payloads":[{"core":"5e90b095e6d69959f708d1eb","launchpad":"5e94502f5080995de566f708","flight_number":1,"name":"FalconSat","date_utc":"2006-03-24T22:30:00.000Z","date_unix":1143239400,"date_local":"2006-03-25T10:30:00+12:00","date_precision":"hour","upcoming":false,"cores":[{"core":"5e90b095e6d69959f708d1eb","flight":1,"griffins":false,"legs":false,"reused":false,"landing_attempt":false,"landing_success":null,"landing_type":null,"landpad":null},"auto_update":true,"tbody":false,"launch_library_id":null,"id":"5e807cd0ff08e000040431b"}],"fairings":{"reused":false,"recovery_attempt":false,"recovered":false,"ships":[]},"links":{"patch":{"small":"https://images2.imgbox.com/f9/4a/2boX0e0_o.png","large":"https://images2.imgbox.com/80/a2/bk0tCIS_o.png"},"reddit":{"campaign":null,"launch":null,"media":null,"recovery":null},"flickr":{"small":"","original":[]},"presskit":null,"webcast":"https://www.youtube.com/watch?v=Lk4Q2uP-Mc","youtube_id":"Lk4Q2uP-Mc","article":"https://www.space.com/3590-spacex-falcon-1-rocket-fails-reach-orbit.html","wikipedia":"https://en.wikipedia.org/wiki/DemoSat"},"static_fire_date_utc":null,"static_fire_date_unix":null,"net":false,"window":0,"rocket":"5e90b095e6d69959f708d1eb","success":false,"failures":[{"time":301,"altitude":209,"reason":"harmonic oscillation leading to premature engine shutdown"}],"details":{"Successful first stage burn and transition to second stage, maximum altitude 209 km, premature engine shutdown at T+7 min 30 s, failed to reach orbit, failed to recover first stage","crew":[],"ships":[]},"capsules":{"payloads":[{"core":"5e90b095e6d69959f708d1eb","launchpad":"5e94502f5080995de566f708","flight_number":2,"name":"DemoSat","date_utc":"2007-03-17T01:10:00.000Z","date_unix":1174439400,"date_local":"2007-03-17T13:10:00+12:00","date_precision":"hour","upcoming":false,"cores":[{"core":"5e90b095e6d69959f708d1eb","flight":1,"griffins":false,"legs":false,"reused":false,"landing_attempt":false,"landing_success":null,"landing_type":null,"landpad":null},"auto_update":true,"tbody":false,"launch_library_id":null,"id":"5e807cd0ff08e000040431b"}],"fairings":{"reused":false,"recovery_attempt":false,"recovered":false,"ships":[]},"links":{"patch":{"small":"https://images2.imgbox.com/4a/00/Ks4K4V9k_o.png","large":"https://images2.imgbox.com/4a/00/Ks4K4V9k_o.png"},"reddit":{"campaign":null,"launch":null,"media":null,"recovery":null},"flickr":{"small":"","original":[]},"presskit":null,"webcast":"https://www.youtube.com/watch?v=0a_00n1_Y80","youtube_id":"0a_00n1_Y80","article":"https://www.space.com/2013/02/11/falcon-1-flight-3-mission-summary","wikipedia":"https://en.wikipedia.org/wiki/Trailblazer_(satellite)","static_fire_date_utc":null,"static_fire_date_unix":null,"net":false,"window":0,"rocket":"5e90b095e6d69959f708d1eb","success":false,"failures":[{"time":140,"altitude":35,"reason":"residual stage-1 thrust led to collision between stage 1 and stage 2"}],"details":{"Residual stage 1 thrust led to collision between stage 1 and stage 2","crew":[],"ships":[]},"capsules":{"payloads":[{"core":"5e90b095e6d69959f708d1eb","launchpad":"5e94502f5080995de566f708","flight_number":3,"name":"Trailblazer","date_utc":"2008-08-03T03:34:00.000Z","date_unix":1217394400,"date_local":"2008-08-03T15:34:00+12:00","date_precision":"hour","upcoming":false,"cores":[{"core":"5e90b095e6d69959f708d1eb","flight":1,"griffins":false,"legs":false,"reused":false,"landing_attempt":false,"landing_success":null,"landing_type":null,"landpad":null},"auto_update":true,"tbody":false,"launch_library_id":null,"id":"5e807cd0ff08e000040431b"}],"fairings":{"reused":false,"recovery_attempt":false,"recovered":false,"ships":[]},"links":{"patch":{"small":"https://images2.imgbox.com/95/39/skq07rsV_o.png","large":"https://images2.imgbox.com/a3/99/qswRtE3_o.png"},"reddit":{"campaign":null,"launch":null,"media":null,"recovery":null},"flickr":{"small":"","original":[]},"presskit":null,"webcast":"https://www.youtube.com/watch?v=4QzT2EH6G0","article":
```



# Data Collection - Scraping

- Present your web scraping process using key phrases and flowcharts
- Add the GitHub URL of the completed web scraping notebook, as an external reference and peer-review purpose
- <https://github.com/tanglily08/IBM-data-science-final-project/blob/c8eb934d06a503b6ea7fb050e491a887669f15bd/webscraping-datascience-final2.ipynb>

## TASK 1: Request the Falcon9 Launch Wiki page from its URL

First, let's perform an HTTP GET method to request the Falcon9 Launch HTML page, as an HTTP response.

```
# use requests.get() method with the provided static_url
# assign the response to a object
data = requests.get(static_url)
```

Create a BeautifulSoup object from the HTML response

```
# Use BeautifulSoup() to create a BeautifulSoup object from a response text content
soup = data BeautifulSoup()
```

```
AttributeError                                Traceback (most recent call last)
~/tmp/ipykernel_37/958580267.py in <module>
      1 # Use BeautifulSoup() to create a BeautifulSoup object from a response text content
----> 2 soup = data BeautifulSoup()

AttributeError: 'Response' object has no attribute 'BeautifulSoup'
```

Print the page title to verify if the BeautifulSoup object was created properly

```
# Use soup.title attribute
print(soup.title)
```

## TASK 2: Extract all column/variable names from the HTML table header

# Data Wrangling

---

- Describe how data were processed
- You need to present your data wrangling process using key phrases and flowcharts
- Add the GitHub URL of your completed data wrangling related notebooks, as an external reference and peer-review purpose
- <https://github.com/tanglily08/IBM-data-science-final-project/blob/c8eb934d06a503b6ea7fb050e491a887669f15bd/labs-jupyter-spacex-Data%20wrangling.ipynb>

# EDA with Data Visualization

---

- Summarize what charts were plotted and why you used those charts
  1. Catplot
  2. Barplot
  3. Scatterplot
  4. Lineplot
- Add the GitHub URL of your completed EDA with data visualization notebook, as an external reference and peer-review purpose
- <https://github.com/tanglily08/IBM-data-science-final-project/blob/c8eb934d06a503b6ea7fb050e491a887669f15bd/spacex-explor-prepare-data-datascience-final4.ipynb>

# EDA with SQL

---

- Using bullet point format, summarize the SQL queries you performed

1. `selectQuery = "select launchsite from spacex"`
2. `selectQuery2 = "select launchsite from spacex where launchsite='CCA%' limit 5"`
3. `selectQuery3 = "select sum(PAYLOAD_MASS_KG ) as sum_of_payload_mass from spacex_new "`
4. `selectQuery4 = "select avg(payload_mass_kg ) as average_payload_mass_by_F9v1_1 from spacex_new where booster_version = 'F9v1.1'"`
5. `selectQuery5 = "select min(date) as first_success_landing from spacex_new where landingoutcome='Success'"`
6. `selectQuery6 = "select booster_version from spacex_new where landingoutcome='Success (drone ship)' and (payload_mass_kg >4000 and payload_mass_kg <6000)"`
7. `selectQuery7 = "select mission_outcome, count(mission_outcome) from spacex_new group by mission_outcome"`
8. `selectQuery8 = "select booster_version from spacex_new where payload_mass_kg =(select max(payload_mass_kg ) from spacex_new)"`
9. `selectQuery9 = "select booster_version, launch_site from spacex_new where YEAR(date)=2015 and landingoutcome= 'Failure (drone ship)'"`
10. `selectQuery10 = "select landingoutcome as landing_outcome, count(landingoutcome) as count_landing_outcome from spacex_new where date between '2010-06-04' and '2017-03-20' group by landingoutcome order by count(landingoutcome) desc"`

- Add the GitHub URL of your completed EDA with SQL notebook, as an external reference and peer-review purpose

- <https://github.com/tanglily08/IBM-data-science-final-project/blob/c8eb934d06a503b6ea7fb050e491a887669f15bd/sql-notebook-datascience-final.ipynb>

# Build an Interactive Map with Folium

---

- Summarize what map objects such as markers, circles, lines, etc. you created and added to a folium map
- I added markers and circles.
- Explain why you added those objects
- Add the GitHub URL of your completed interactive map with Folium map, as an external reference and peer-review purpose
- <https://github.com/tanglily08/IBM-data-science-final-project/blob/c8eb934d06a503b6ea7fb050e491a887669f15bd/interactive-visual-analytics-datascience-final5.ipynb>



# Build a Dashboard with Plotly Dash

---

- Summarize what plots/graphs and interactions you have added to a dashboard
- Explain why you added those plots and interactions
- Add the GitHub URL of your completed Plotly Dash lab, as an external reference and peer-review purpose
- [https://github.com/tanglily08/IBM-data-science-final-project/blob/c8eb934d06a503b6ea7fb050e491a887669f15bd/spacex\\_dash\\_app.py](https://github.com/tanglily08/IBM-data-science-final-project/blob/c8eb934d06a503b6ea7fb050e491a887669f15bd/spacex_dash_app.py)

# Predictive Analysis (Classification)

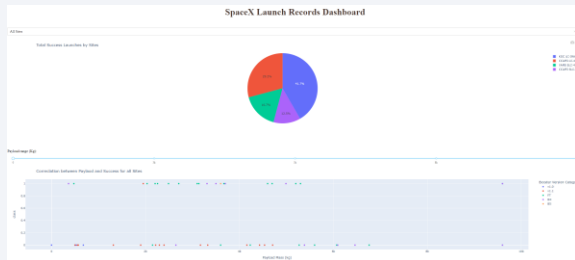
---

- Summarize how you built, evaluated, improved, and found the best performing classification model
- KNN Regression performs best
- You need present your model development process using key phrases and flowchart
- Add the GitHub URL of your completed predictive analysis lab, as an external reference and peer-review purpose
- <https://github.com/tanglily08/IBM-data-science-final-project/blob/040bb1c94f9eb0da532011ded55c65da144a573f/spacex-ml-prediction-datascience-final7.ipynb>

# Results

---

- Exploratory data analysis results
- [Logistic Regression performs best, because it got highest score, 1.](#)
- Interactive analytics demo in screenshots



- Predictive analysis results



The background of the slide is an abstract composition. It features a dark blue field on the left side, which transitions into a complex pattern of diagonal streaks in shades of blue, red, and teal on the right. These streaks have a textured, almost woven appearance. Overlaid on this pattern is a faint, light blue grid that recedes into the distance, creating a sense of depth and perspective.

Section 2

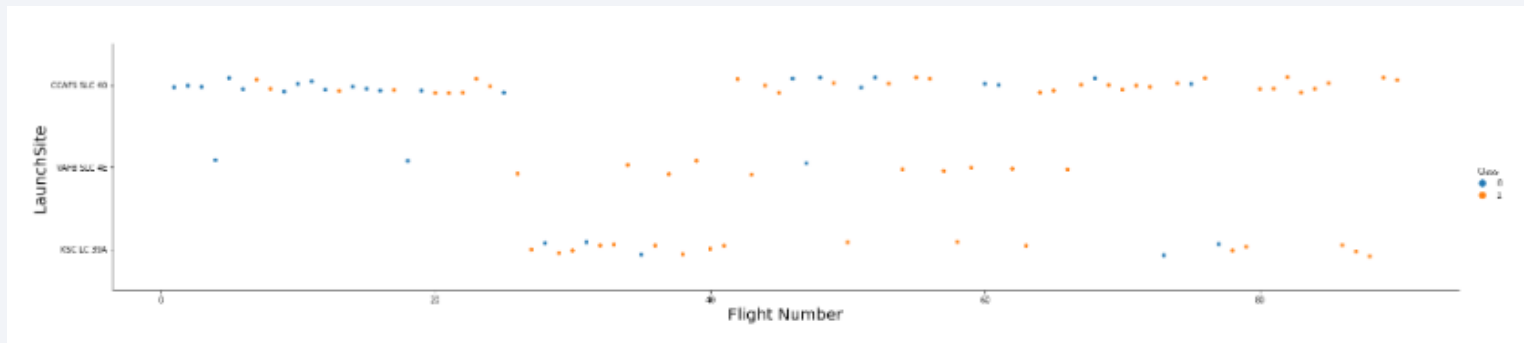
# Insights drawn from EDA



# Flight Number vs. Launch Site

---

- Show a scatter plot of Flight Number vs. Launch Site



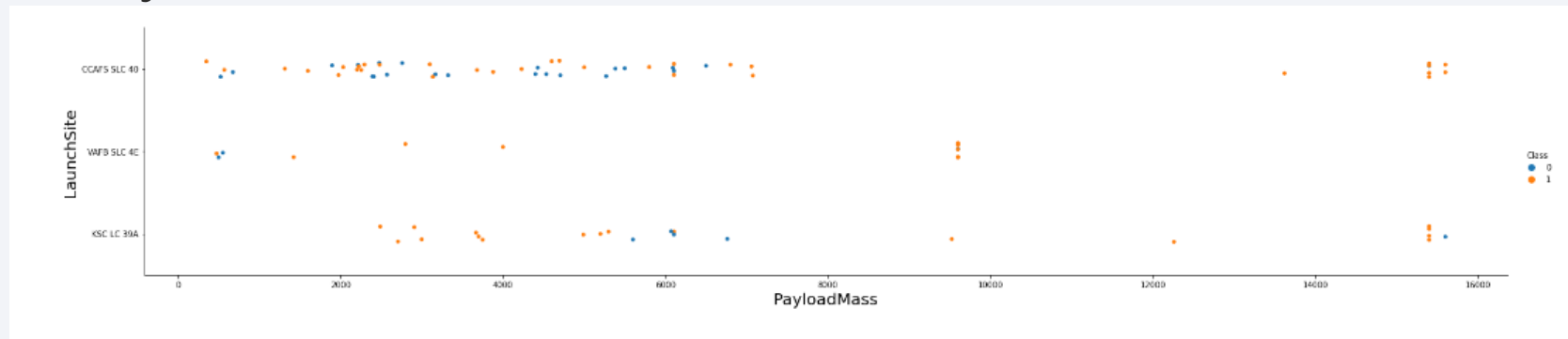
- Show the screenshot of the scatter plot with explanations



# Payload vs. Launch Site

---

- Show a scatter plot of Payload vs. Launch Site

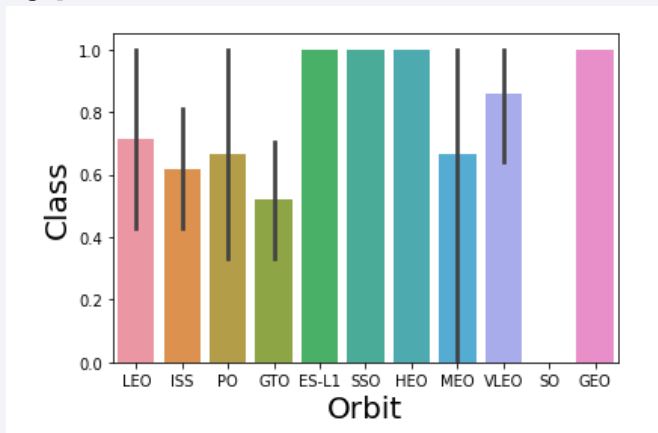


- Show the screenshot of the scatter plot with explanations

# Success Rate vs. Orbit Type

---

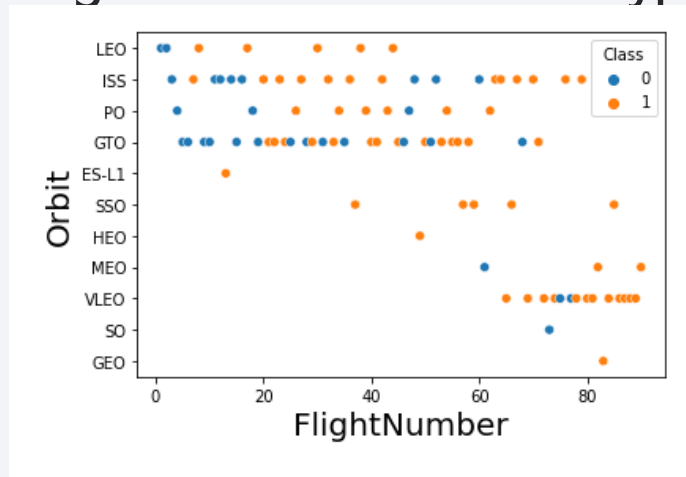
- Show a bar chart for the success rate of each orbit type



- Show the screenshot of the scatter plot with explanations

# Flight Number vs. Orbit Type

- Show a scatter point of Flight number vs. Orbit type

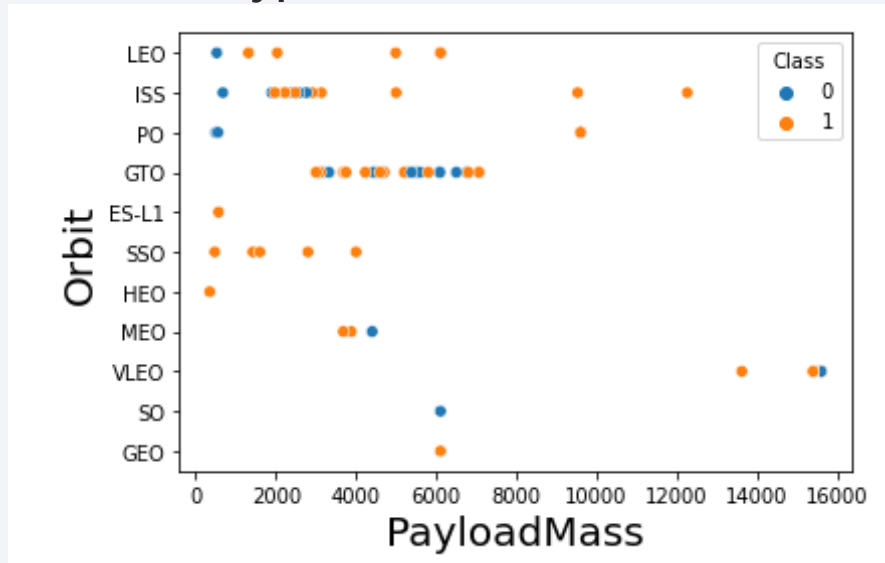


- Show the screenshot of the scatter plot with explanations

# Payload vs. Orbit Type

---

- Show a scatter point of payload vs. orbit type

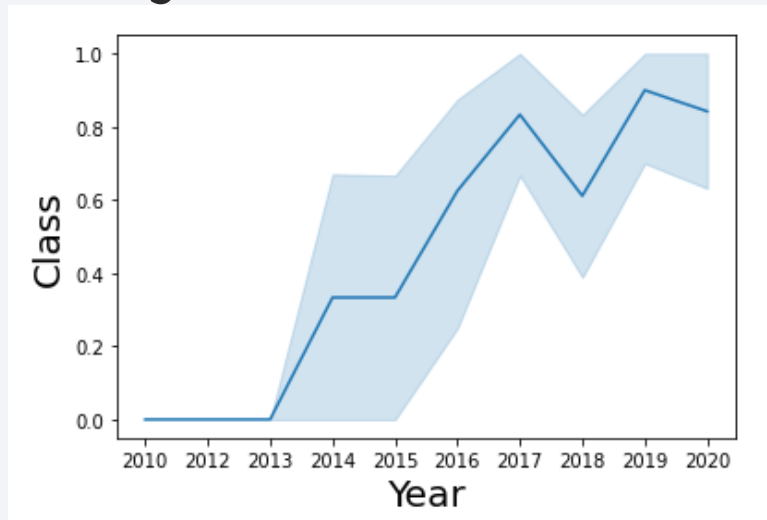


- Show the screenshot of the scatter plot with explanations

# Launch Success Yearly Trend

---

- Show a line chart of yearly average success rate



- Show the screenshot of the scatter plot with explanations



# All Launch Site Names

---

- Find the names of the unique launch sites

	LAUNCH_SITE
0	CCAFS LC-40
1	CCAFS SLC-40
2	KSC LC-39A
3	VAFB SLC-4E

- Present your query result with a short explanation here

# Launch Site Names Begin with 'CCA'

- Find 5 records where launch sites begin with `CCA`

```
[17]:
```

	DATE	TIME_UTC_	BOOSTER_VERSION	LAUNCH_SITE	PAYLOAD	PAYLOAD_MASS_KG_	ORBIT	CUSTOMER	MISSION_OUTCOME	LANDINGOUTCOME
0	1974-04-02	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
1	2002-04-04	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight C1, two CubeSats, barrel of...	0	LEO (ISS)	NASA (COTS) NRO	Success	Failure (parachute)
2	2012-05-22	7:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
3	1931-04-11	0:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
4	1977-04-12	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt

- Present your query result with a short explanation here

# Total Payload Mass

---

- Calculate the total payload carried by boosters from NASA

SUM_OF_PAYLOAD_MASS
619967

- Present your query result with a short explanation here

# Average Payload Mass by F9 v1.1

---

- Calculate the average payload mass carried by booster version F9 v1.1

AVERAGE_PAYLOAD_MASS_BY_F9V1_1
--------------------------------

2928
------

- Present your query result with a short explanation here

# First Successful Ground Landing Date

---

- Find the dates of the first successful landing outcome on ground pad

FIRST_SUCCESS_LANDING
2018-03-12

- Present your query result with a short explanation here



## Successful Drone Ship Landing with Payload between 4000 and 6000

---

- List the names of boosters which have successfully landed on drone ship and had payload mass greater than 4000 but less than 6000

BOOSTER_VERSION	
0	F9 FT B1022
1	F9 FT B1026
2	F9 FT B1021.2
3	F9 FT B1031.2

- Present your query result with a short explanation here

# Total Number of Successful and Failure Mission Outcomes

---

- Calculate the total number of successful and failure mission outcomes

MISSION_OUTCOME		2
0	Failure (in flight)	1
1	Success	99
2	Success (payload status unclear)	1

- Present your query result with a short explanation here

# Boosters Carried Maximum Payload

---

- List the names of the booster which have carried the maximum payload mass

BOOSTER_VERSION	
0	F9 B5 B1048.4
1	F9 B5 B1049.4
2	F9 B5 B1051.3
3	F9 B5 B1056.4
4	F9 B5 B1048.5
5	F9 B5 B1051.4
6	F9 B5 B1049.5
7	F9 B5 B1060.2
8	F9 B5 B1058.3
9	F9 B5 B1051.6
10	F9 B5 B1060.3
11	F9 B5 B1049.7

- Present your query result with a short explanation here

# 2015 Launch Records

---

- List the failed landing\_outcomes in drone ship, their booster versions, and launch site names for in year 2015

BOOSTER_VERSION	LAUNCH_SITE
F9 v1.1 B1015	CCAFS LC-40

- Present your query result with a short explanation here

## Rank Landing Outcomes Between 2010-06-04 and 2017-03-20

---

- Rank the count of landing outcomes (such as Failure (drone ship) or Success (ground pad)) between the date 2010-06-04 and 2017-03-20, in descending order

	LANDING_OUTCOME	COUNT_LANDING_OUTCOME
0	Failure (drone ship)	3
1	No attempt	3
2	Success (drone ship)	3
3	Success (ground pad)	3
4	Controlled (ocean)	2
5	Uncontrolled (ocean)	2
6	Precluded (drone ship)	1

- Present your query result with a short explanation here

A satellite view of Earth from space, showing the curvature of the planet and the glowing lights of cities at night. The background is a deep blue gradient.

Section 3

# Launch Sites Proximities Analysis

# <All US Launchsite>

---

- Explore the generated folium map and make a proper screenshot to include all launch sites' location markers on a global map

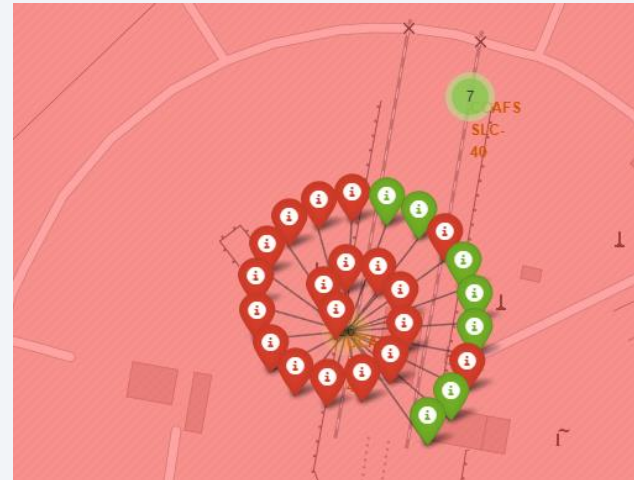
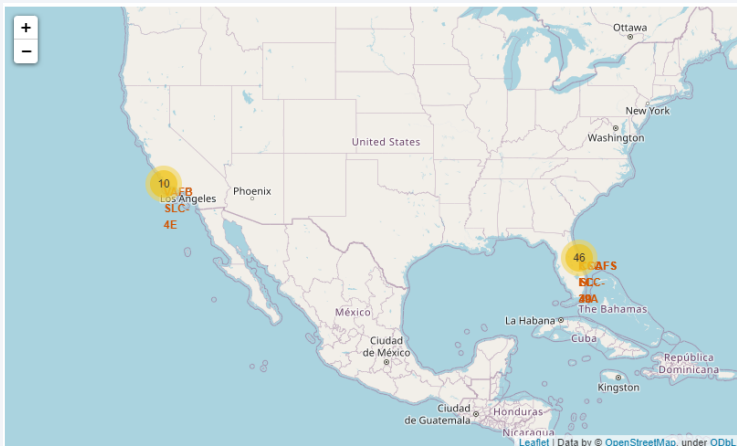


- Explain the important elements and findings on the screenshot
- All launch sites are close to ocean.

# <Launch sites with success marker>

---

- Explore the folium map and make a proper screenshot to show the color-labeled launch outcomes on the map



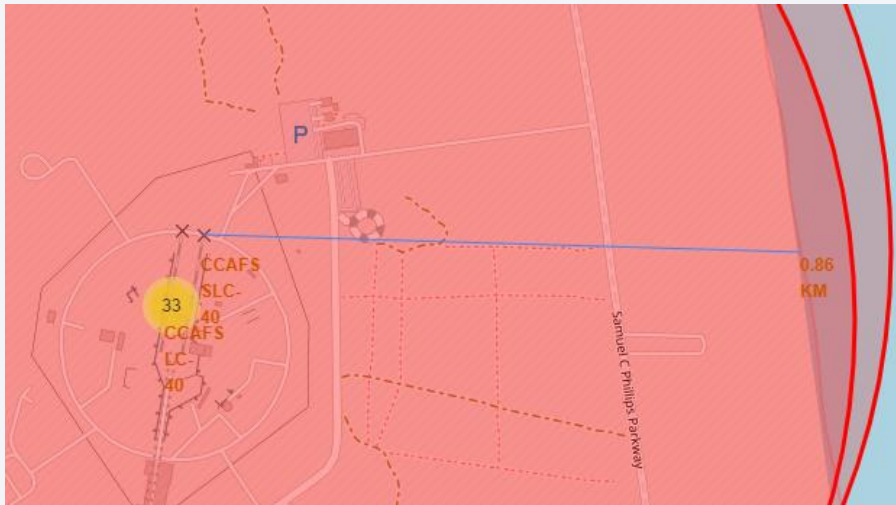
- Explain the important elements and findings on the screenshot
- KCFS-39 site has the most success rate



# <Launch site distance from ocean>

---

- Explore the generated folium map and show the screenshot of a selected launch site to its proximities such as railway, highway, coastline, with distance calculated and displayed



- Explain the important elements and findings on the screenshot
- All launch sites are close to ocean, railway and highway. I think it is for safety and easy to transport.



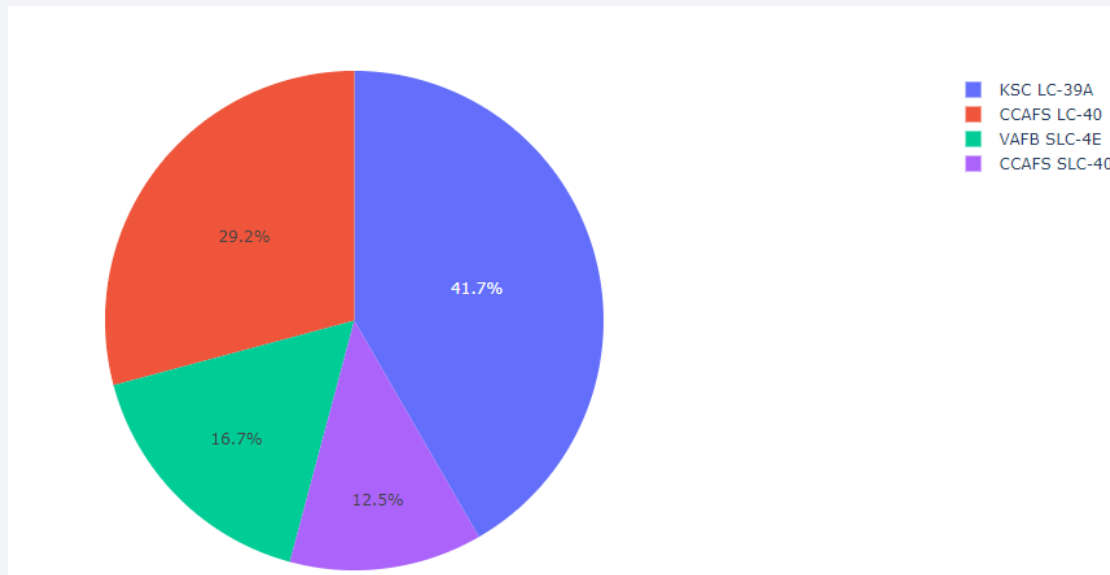
Section 4

# Build a Dashboard with Plotly Dash

# <Launch success count for all sites>

---

- Show the screenshot of launch success count for all sites, in a piechart

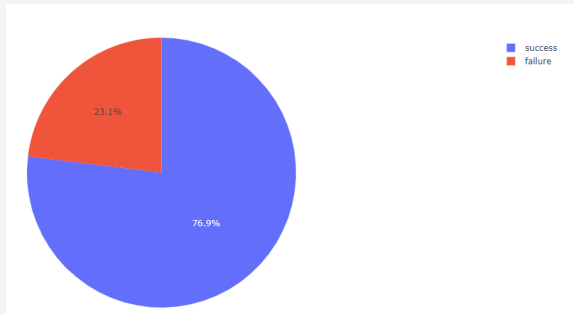


- Explain the important elements and findings on the screenshot
- KSC LX-39A got the most success rate.

## <success rate for KSC LC-39A>

---

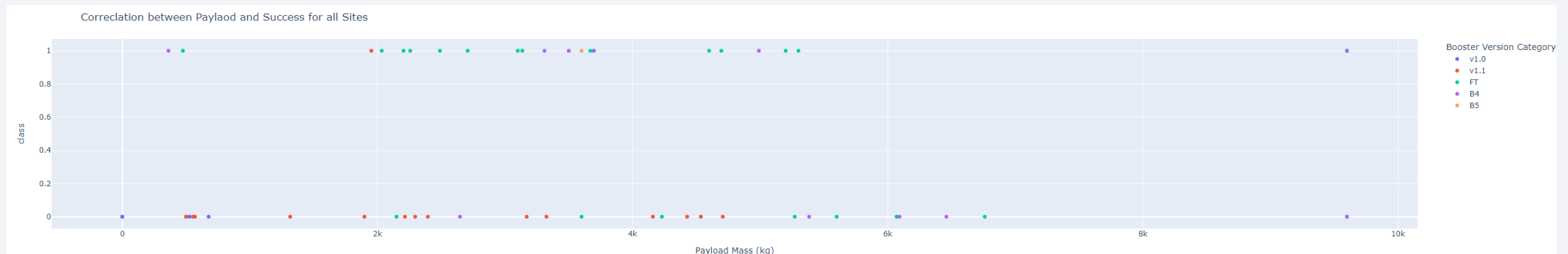
- Show the screenshot of the piechart for the launch site with highest launch success ratio



- Explain the important elements and findings on the screenshot

# <Payload vs launch Outcome>

- Show screenshots of Payload vs. Launch Outcome scatter plot for all sites, with different payload selected in the range slider



- Explain the important elements and findings on the screenshot, such as which payload range or booster version have the largest success rate, etc.
- Between 2K-5K payload, FT have the most success rate.





Section 5

# Predictive Analysis (Classification)

# Classification Accuracy

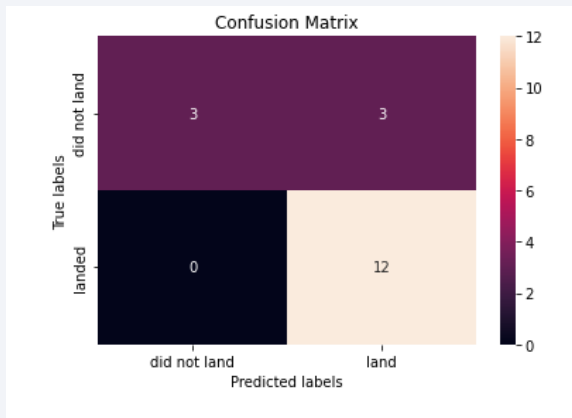
---

- Visualize the built model accuracy for all built classification models, in a bar chart
- Find which model has the highest classification accuracy

# Confusion Matrix

---

- Show the confusion matrix of the best performing model with an explanation
- K Means Clustering is the best model.





# Conclusions

---

- 1. ES-L1, SSO, HEO, GEO orbits have the most success rate.
- 2. All launch sites are close to ocean. KCFS-39 has the most success rate.
- 3. Between 2K-5K payload mass, Booster version FT has the most success rate
- 4. K means clustering provide the best accuracy.
- ...

# Appendix

---

- Include any relevant assets like Python code snippets, SQL queries, charts, Notebook outputs, or data sets that you may have created during this project

Thank you!

