IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Watcharachai Jiramahawitthayakul

Oct 13, 2022

Outline

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

Executive Summary

- Summary of methodologies
- Summary of all results

Introduction

- Project background and context
- Problems you want to find answers

Section 1

Methodology

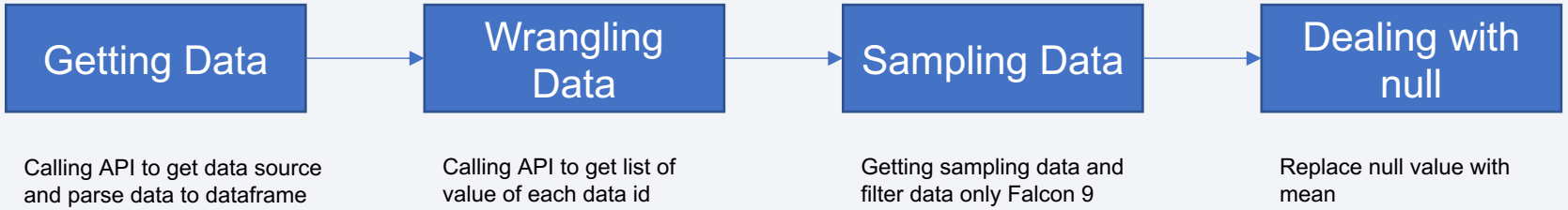
Methodology

Executive Summary

- Data collection
- Data wrangling
- Exploratory data analysis
- Data visualization
- Model Development
- Reporting

Data Collection

- Describe how data sets were collected.
- You need to present your data collection process use key phrases and flowcharts

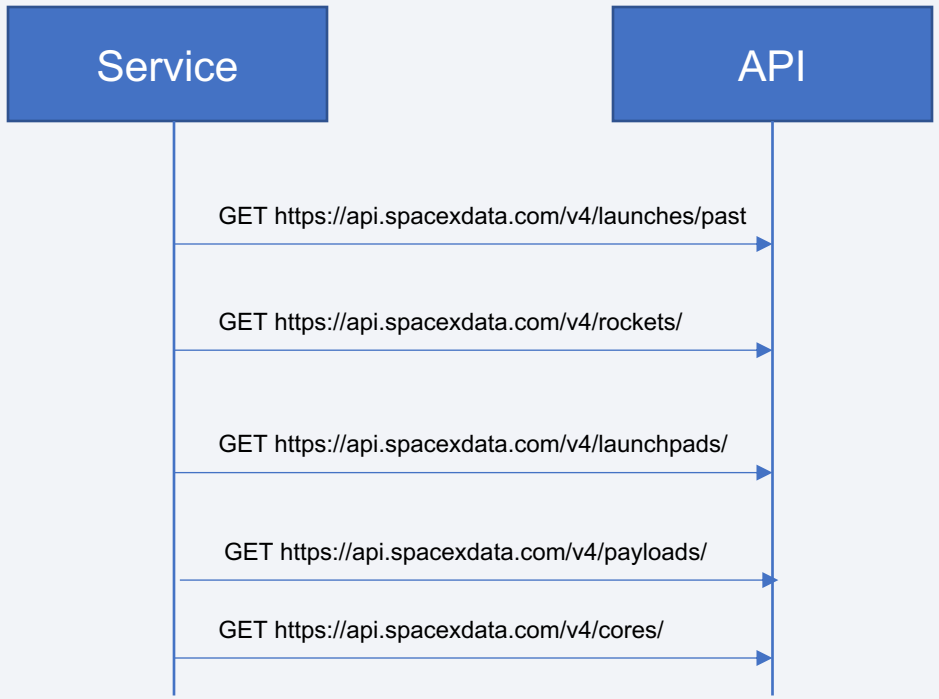


Data Collection – SpaceX API

- Present your data collection with SpaceX REST calls using key phrases and flowcharts

- Add the GitHub URL of the completed SpaceX API calls notebook (must include completed code cell and outcome cell), as an external reference and peer-review purpose

- https://github.com/watcharachaiji/capstone/blob/master/01-jupyter-labs-spacex-data-collection-api.ipynb

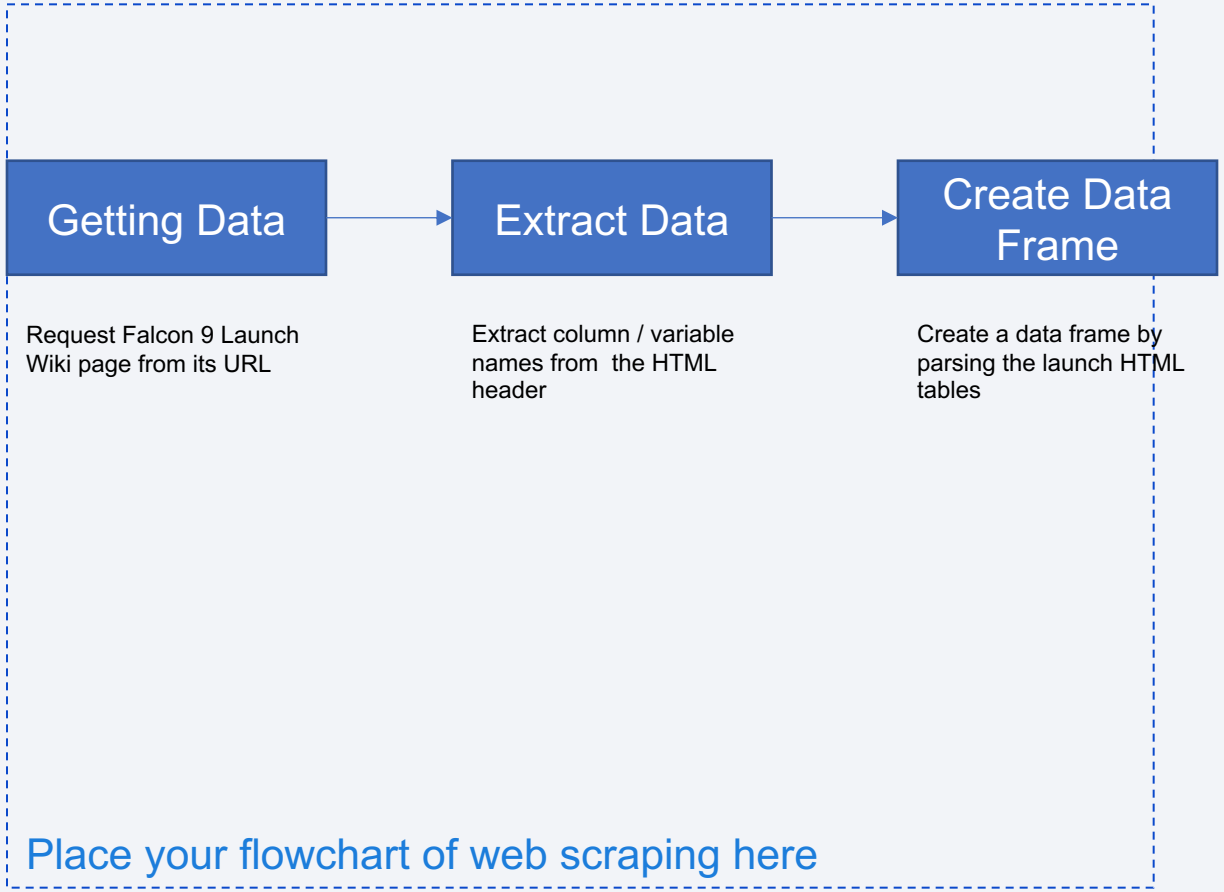


Place your flowchart of SpaceX API calls here

Data Collection - Scraping

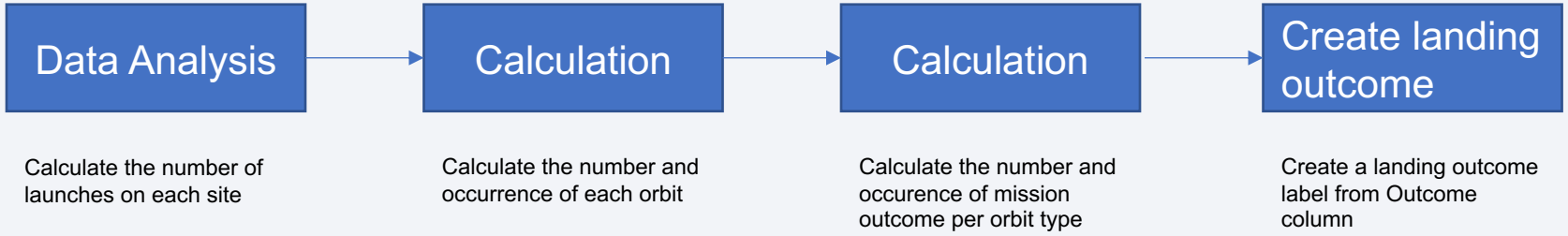
- Present your web scraping process using key phrases and flowcharts

- Add the GitHub URL of the completed web scraping notebook, as an external reference and peer-review purpose
- https://github.com/watcharachaiji/capstone/blob/master/02-jupyter-labs-webscraping.ipynb



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/watcharachaiji/capstone/blob/master/03-labs-jupyter-spacex-Data-wrangling.ipynb



EDA with Data Visualization

- Summarize what charts were plotted and why you used those charts
- Flight no. vs Launch Site
- Payload vs Launch Site
- Success rate of each Orbit type
- Flight no. vs Orbit type
- Payload vs Orbit type
- Launch success yearly trend
- Add the GitHub URL of your completed EDA with data visualization notebook, as an external reference and peer-review purpose
- https://github.com/watcharachaiji/capstone/blob/master/05-jupyter-labs-eda-dataviz.ipynb

EDA with SQL

- Using bullet point format, summarize the SQL queries you performed
- Get list of launch site, 5 records where launch sites begin with the string ‘CCA’
- Calculate total payload mass carried by boosters launched by NASA (CRS)
- Display average payload mass carried by booster version F9 v1.1
- List the date when the first successful landing outcome in ground pad was achieved.
- List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000
- List the total number of successful and failure mission outcomes
- List the names of the booster_versions which have carried the maximum payload mass
- List the failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015
- 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
- Add the GitHub URL of your completed EDA with SQL notebook, as an external reference and peer-review purpose
- https://github.com/watcharachaiji/capstone/blob/master/04-jupyter-labs-eda-sql-coursera.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
- Explain why you added those objects
- Plot all launch sites on map with the success / fail launches for each site on map
- Calculate the distances between a launch site to its proximities
- Add the GitHub URL of your completed interactive map with Folium map, as an external reference and peer-review purpose
- https://github.com/watcharachaiji/capstone/blob/master/06-lab_jupyter_launch_site_location.ipynb

Build a Dashboard with Plotly Dash

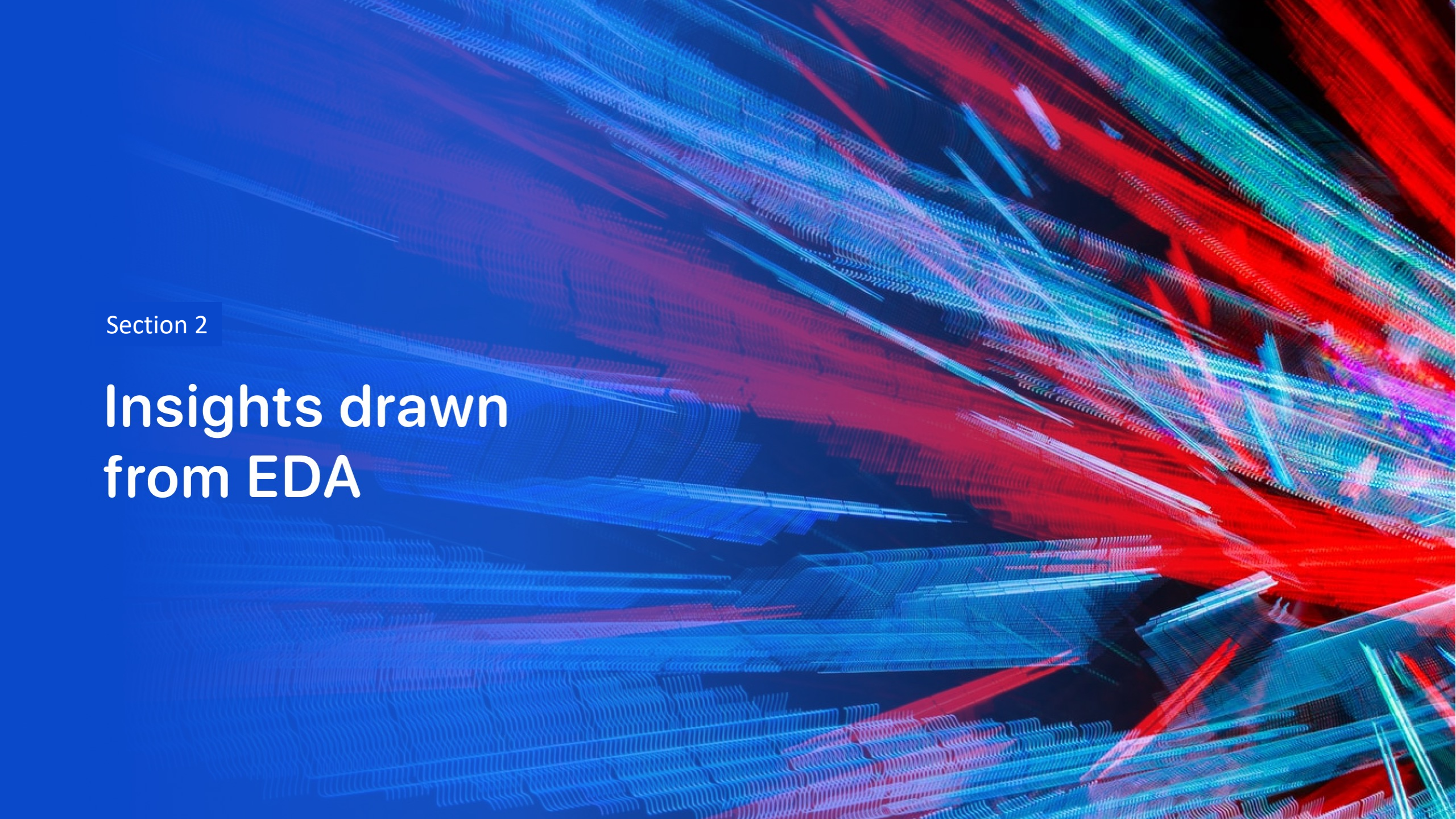
- Summarize what plots/graphs and interactions you have added to a dashboard
- Explain why you added those plots and interactions
- Add the GitHub URL of your completed Plotly Dash lab, as an external reference and peer-review purpose

Predictive Analysis (Classification)

- Summarize how you built, evaluated, improved, and found the best performing classification model
- You need present your model development process using key phrases and flowchart
- Split data for test and train
- Logistic regression - 0.8333333333333334
- Vector machine - 0.8333333333333334
- Decision tree classifier - 0.8333333333333334
- KNN 0.8333333333333334
- Conclusion – all the same
- Add the GitHub URL of your completed predictive analysis lab, as an external reference and peer-review purpose
- https://github.com/watcharachaiji/capstone/blob/master/08-SpaceX_Machine%20Learning%20Prediction_Part_5.ipynb

Results

- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results

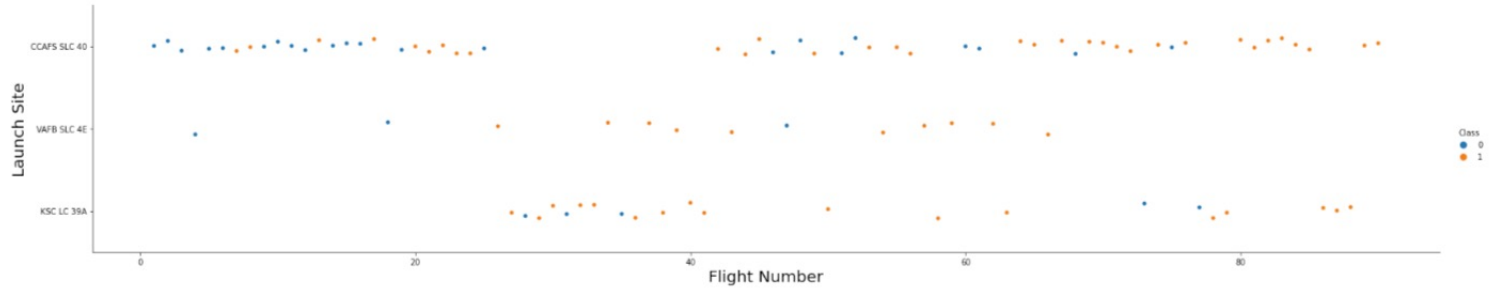

Section 2
Insights drawn from EDA

Flight Number vs. Launch Site

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

- Show the screenshot of the scatter plot with explanations
- KSC LC-39A and VAFB SLC 4E have more success rate than CCAFS LC-40

In [4]:
```
# Plot a scatter point chart with x axis to be Flight Number and y axis to be the launch site, and hue to be the class value
sns.catplot(y="LaunchSite", x="FlightNumber", hue="Class", data=df, aspect = 5)
plt.xlabel("Flight Number",fontsize=20)
plt.ylabel("Launch Site",fontsize=20)
plt.show()
```

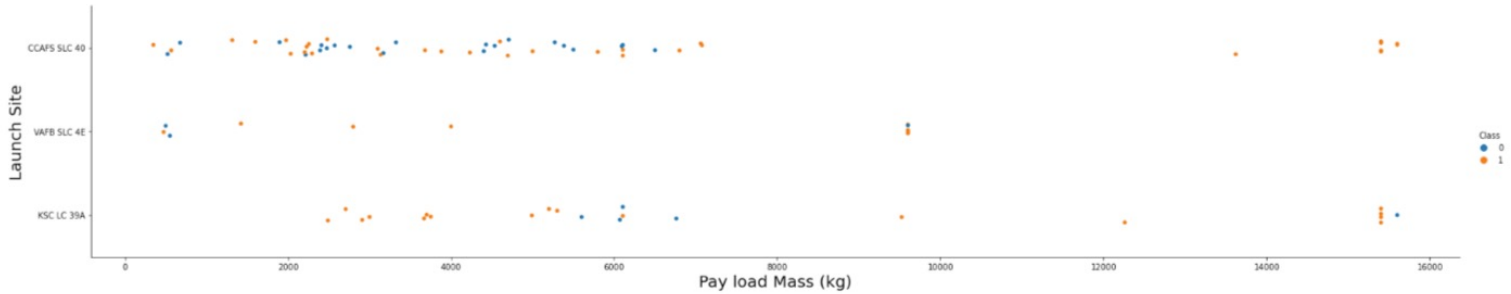


Payload vs. Launch Site

- Show a scatter plot of Payload vs. Launch Site

- Show the screenshot of the scatter plot with explanations
- CCAFS LC-40 and KSC LC-39A have more success rate when carry heavier pay load
- While VAFB SLC 4E and KSC LC-39A have more success rate when carry lighter pay load

In [5]:
```
# Plot a scatter point chart with x axis to be Pay Load Mass (kg) and y axis to be the launch site, and hue to be the class value
sns.catplot(y="LaunchSite", x="PayloadMass", hue="Class", data=df, aspect = 5)
plt.xlabel("Pay load Mass (kg)",fontsize=20)
plt.ylabel("Launch Site",fontsize=20)
plt.show()
```



Success Rate vs. Orbit Type

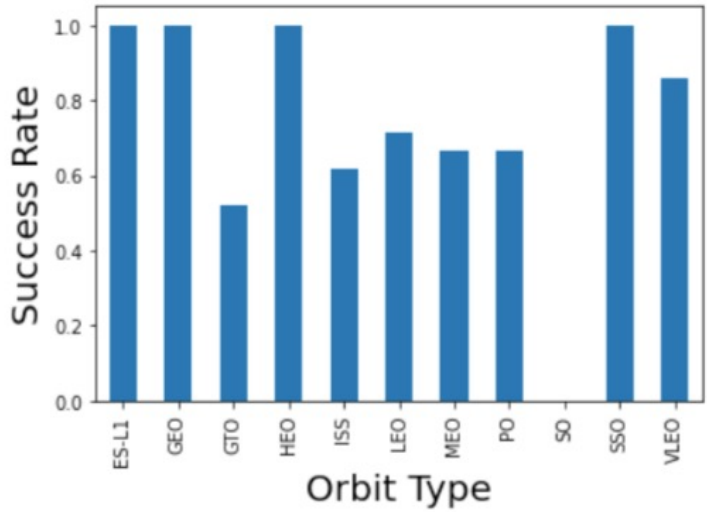
- Show a bar chart for the success rate of each orbit type

- Show the screenshot of the scatter plot with explanations

- ES-L1, GEO, HEO, SSO, VLEO have high success rate which mean far orbit might have more success rate

In [6]:

```
# HINT use groupby method on Orbit column and get the mean of Class column
df.groupby("Orbit").mean()['Class'].plot(kind='bar')
plt.xlabel("Orbit Type",fontsize=20)
plt.ylabel("Success Rate",fontsize=20)
plt.show()
```

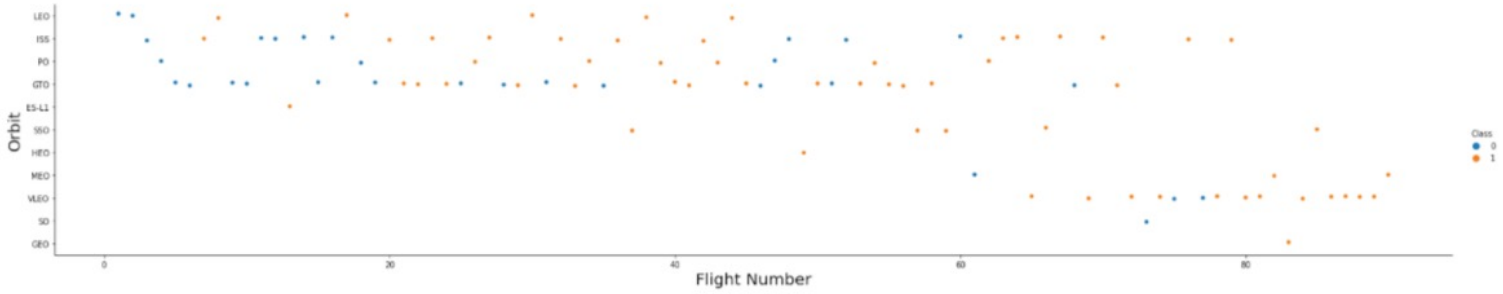


Flight Number vs. Orbit Type

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

- Show the screenshot of the scatter plot with explanations
- the LEO orbit and other orbit the Success appears related to the number of flights; on the other hand, there seems to be no relationship between flight number when in GTO orbit.

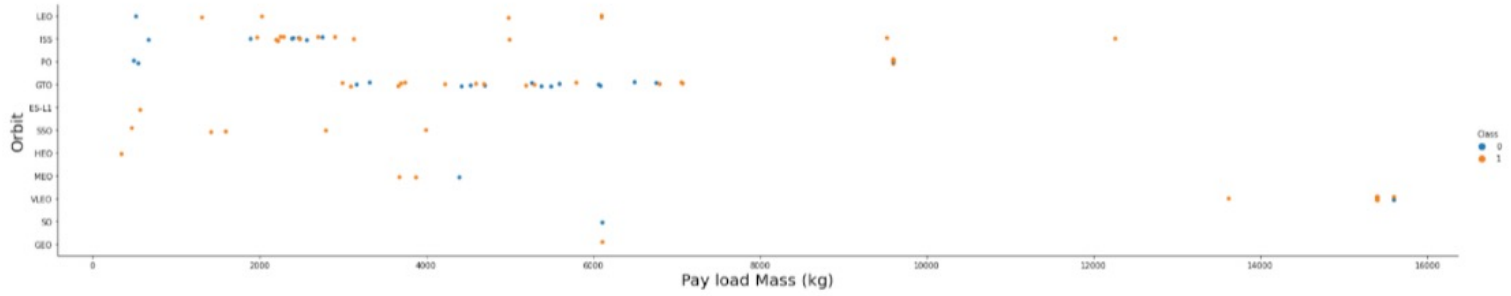
In [7]:
```
# Plot a scatter point chart with x axis to be FlightNumber and y axis to be the Orbit, and hue to be the class value

sns.catplot(y="Orbit", x="FlightNumber", hue="Class", data=df, aspect = 5)
plt.xlabel("Flight Number",fontsize=20)
plt.ylabel("Orbit",fontsize=20)
plt.show()
```



Payload vs. Orbit Type

- Show a scatter point of payload vs. orbit type
- Show the screenshot of the scatter plot with explanations
- With heavy payloads the successful landing or positive landing rate are more for Polar,LEO and ISS.
- However for GTO we cannot distinguish this well as both positive landing rate and negative landing(unsuccessful mission) are both there here.

In [8]:
```
# Plot a scatter point chart with x axis to be Payload and y axis to be the Orbit, and hue to be the class value
sns.catplot(y="Orbit", x="PayloadMass", hue="Class", data=df, aspect = 5)
plt.xlabel("Pay load Mass (kg)",fontsize=20)
plt.ylabel("Orbit",fontsize=20)
plt.show()
```



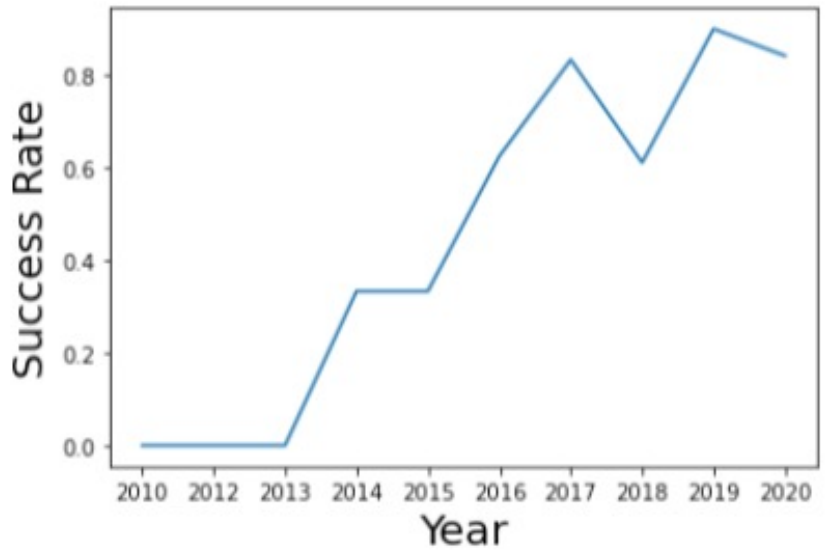
Launch Success Yearly Trend

- Show a line chart of yearly average success rate

- Show the screenshot of the scatter plot with explanations

- you can observe that the sucess rate since 2013 kept increasing till 2020

```
In [11]: sns.lineplot(data=df1.groupby('year')['Class'].mean())
         plt.xlabel("Year", fontsize=20)
         plt.ylabel("Success Rate", fontsize=20)
         plt.show()
```



you can observe that the sucess rate since 2013 kept increasing till 2020

All Launch Site Names

- Find the names of the unique launch sites
- Present your query result with a short explanation here

Task 1

Display the names of the unique launch sites in the space mission

In [5]:
```
%sql SELECT DISTINCT LAUNCH_SITE FROM spacex
```

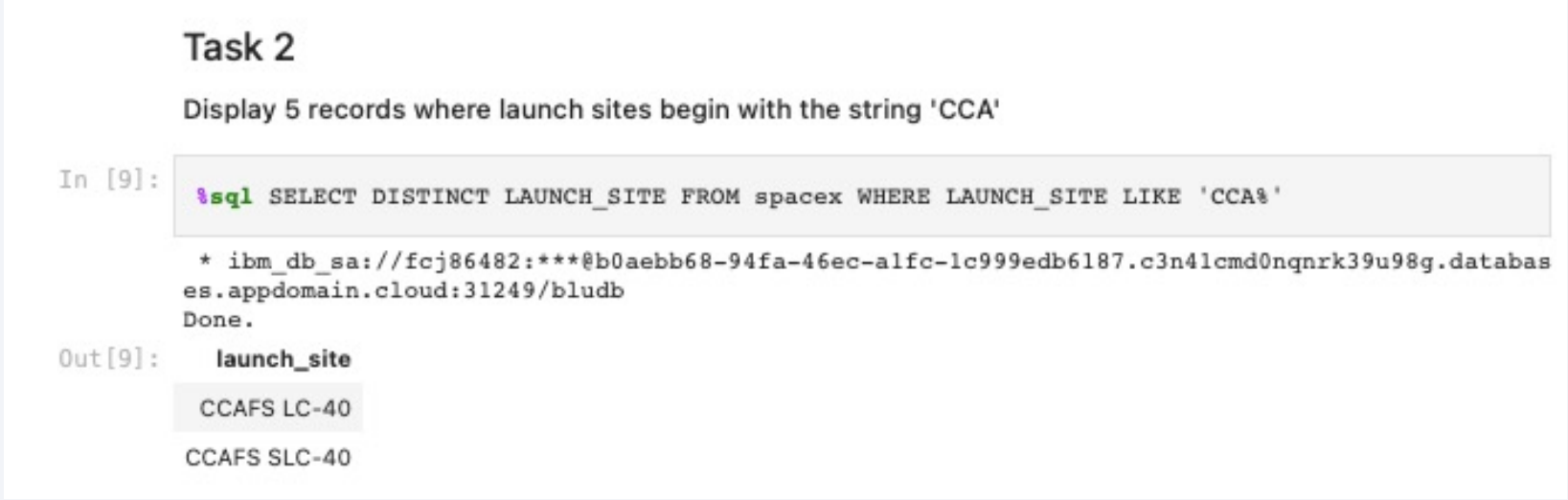
```
 * ibm_db_sa://fcj86482:***@b0aebb68-94fa-46ec-a1fc-1c999edb6187.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud:31249/bludb
Done.
```

Out[5]:

launch_site
CCAFS LC-40
CCAFS SLC-40
KSC LC-39A
VAFB SLC-4E

Launch Site Names Begin with 'CCA'

- Find 5 records where launch sites begin with `CCA`
- Present your query result with a short explanation here



Task 2

Display 5 records where launch sites begin with the string 'CCA'

In [9]:

```
%sql SELECT DISTINCT LAUNCH_SITE FROM spacex WHERE LAUNCH_SITE LIKE 'CCA%'
```

```
 * ibm_db_sa://fcj86482:***@b0aebb68-94fa-46ec-a1fc-1c999edb6187.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud:31249/bludb
Done.
```

Out[9]:

launch_site
CCAFS LC-40
CCAFS SLC-40

Total Payload Mass

- Calculate the total payload carried by boosters from NASA
- Present your query result with a short explanation here

Task 3

Display the total payload mass carried by boosters launched by NASA (CRS)

In [10]:
```
%sql SELECT SUM(payload_mass__kg_) FROM spacex WHERE customer = 'NASA (CRS)'
```

```
 * ibm_db_sa://fcj86482:***@b0aebb68-94fa-46ec-a1fc-1c999edb6187.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud:31249/bludb
Done.
```

Out[10]:

1
45596

Average Payload Mass by F9 v1.1

- Calculate the average payload mass carried by booster version F9 v1.1
- Present your query result with a short explanation here

Task 4

Display average payload mass carried by booster version F9 v1.1

In [11]:
```
%sql SELECT AVG(payload_mass__kg_) FROM spacex WHERE BOOSTER_VERSION LIKE 'F9 v1.1%'
```

```
 * ibm_db_sa://fcj86482:***@b0aebb68-94fa-46ec-a1fc-1c999edb6187.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud:31249/bludb
Done.
```

Out[11]:

1
2534

First Successful Ground Landing Date

- Find the dates of the first successful landing outcome on ground pad
- Present your query result with a short explanation here

Task 5

List the date when the first successful landing outcome in ground pad was acheived.

Hint:Use min function

In [12]:
```
%sql SELECT MIN(DATE) FROM spacex WHERE LANDING__OUTCOME = 'Success (ground pad)'
```

```
 * ibm_db_sa://fcj86482:***@b0aebb68-94fa-46ec-a1fc-1c999edb6187.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud:31249/bludb
Done.
```

Out[12]:

1
2015-12-22

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

- Present your query result with a short explanation here

Task 6

List the names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000

In [13]:
```
%sql SELECT DISTINCT customer FROM spacex WHERE ( LANDING__OUTCOME = 'Success (drone ship)'
```

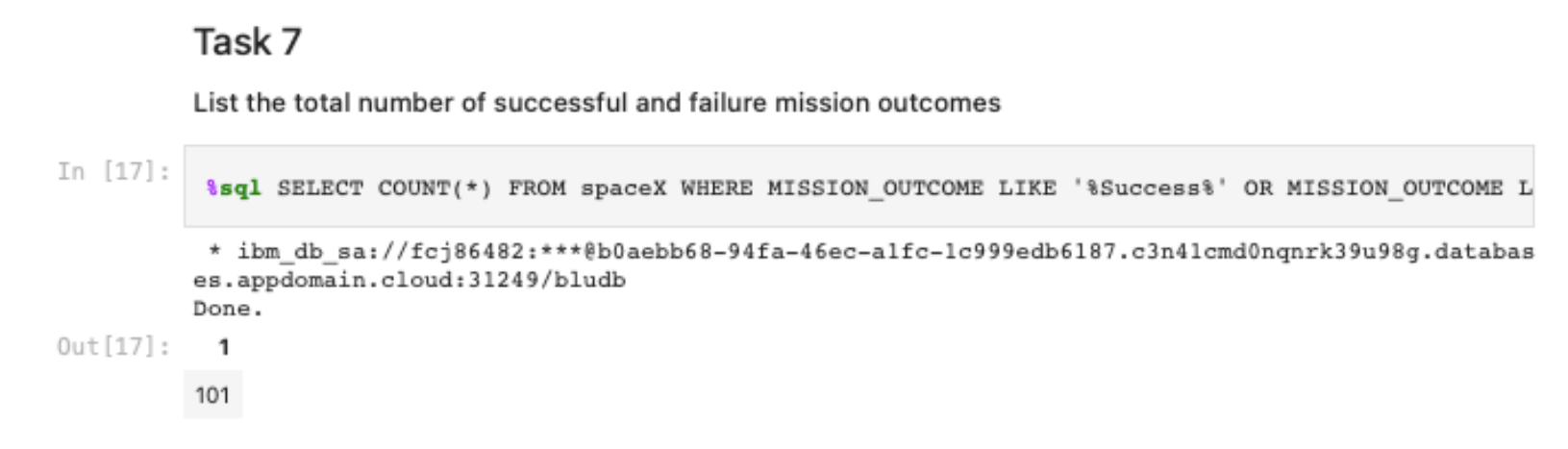
```
 * ibm_db_sa://fcj86482:***@b0aebb68-94fa-46ec-a1fc-1c999edb6187.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud:31249/bludb
Done.
```

Out[13]:

customer
SES
SES EchoStar
SKY Perfect JSAT Group

Total Number of Successful and Failure Mission Outcomes

- Calculate the total number of successful and failure mission outcomes
- Present your query result with a short explanation here



Task 7

List the total number of successful and failure mission outcomes

In [17]:
```
%sql SELECT COUNT(*) FROM spaceX WHERE MISSION_OUTCOME LIKE '%Success%' OR MISSION_OUTCOME L
```

```
 * ibm_db_sa://fcj86482:***@b0aebb68-94fa-46ec-a1fc-1c999edb6187.c3n41cmd0nqnrk39u98g.databas
es.appdomain.cloud:31249/bludb
Done.
```

Out[17]:

1
101

Boosters Carried Maximum Payload

- List the names of the booster which have carried the maximum payload mass
- Present your query result with a short explanation here

Task 8

List the names of the booster_versions which have carried the maximum payload mass. Use a subquery

In [16]:
```
%sql SELECT BOOSTER_VERSION FROM spacex WHERE PAYLOAD_MASS__KG_ = (SELECT MAX(PAYLOAD_MASS__
```

```
 * ibm_db_sa://fcj86482:***@b0aebb68-94fa-46ec-a1fc-1c999edb6187.c3n41cmd0nqnrk39u98g.databas
es.appdomain.cloud:31249/bludb
Done.
```

Out[16]:

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

2015 Launch Records

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

- Present your query result with a short explanation here

Task 9

List the failed landing_outcomes in drone ship, their booster versions, and launch site names for in year 2015

In [23]:
```
%sql SELECT date, LANDING__OUTCOME, BOOSTER_VERSION, launch_site FROM spaceX WHERE LANDING__
```

```
 * ibm_db_sa://fcj86482:***@b0aebb68-94fa-46ec-a1fc-1c999edb6187.c3n41cmd0nqnrk39u98g.databases.appdomain.cloud:31249/bludb
Done.
```

Out[23]:

DATE	landing__outcome	booster_version	launch_site
2015-01-10	Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
2015-04-14	Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40

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

- Present your query result with a short explanation here

Task 10

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

In [25]:
```
%sql SELECT LANDING__OUTCOME, COUNT(*) FROM spaceX WHERE date BETWEEN '2010-06-04' AND '2017
```

```
 * ibm_db_sa://fcj86482:***@b0aebb68-94fa-46ec-a1fc-1c999edb6187.c3n41cmd0nqnrk39u98g.databas
es.appdomain.cloud:31249/bludb
Done.
```

Out[25]:

landing__outcome	2
Controlled (ocean)	3
Failure (drone ship)	5
Failure (parachute)	2
No attempt	10
Precluded (drone ship)	1
Success (drone ship)	5
Success (ground pad)	3
Uncontrolled (ocean)	2

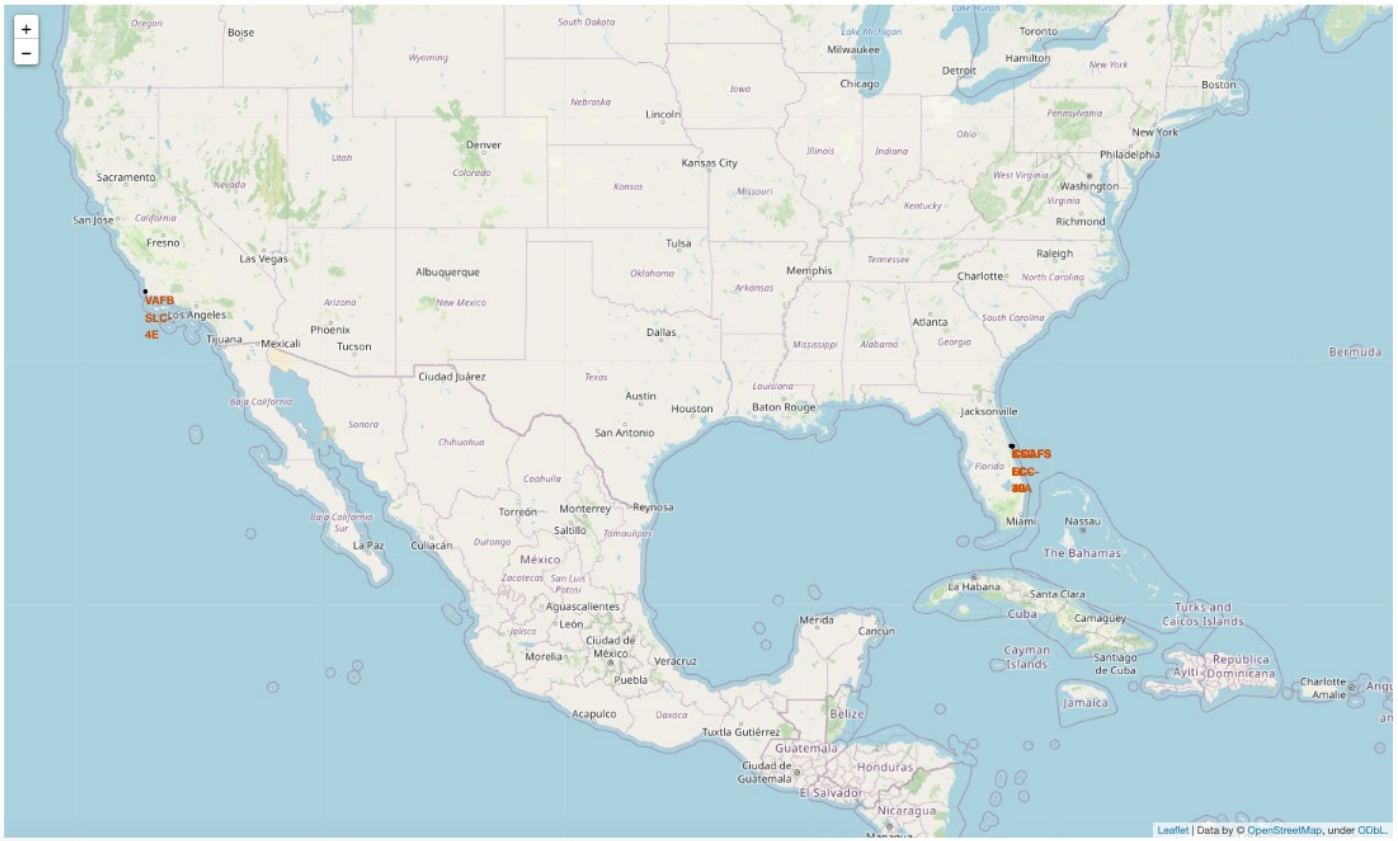

Section 3
Launch Sites
Proximities Analysis

All launch site on map

- Replace <Folium map screenshot 1> title with an appropriate title
- 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
- This map display all launch site on map, we can see that most of the launch site are located near by the ocean and equator

In [8]:
```
# Initial the map
site_map = folium.Map(location=nasa_coordinate, zoom_start=5)
# For each launch site, add a Circle object based on its coordinate (Lat, Long) values. In addition, add Launch site name as a popup label
for index, row in launch_sites_df.iterrows():
    coordinate = [row['Lat'], row['Long']]
    folium.Circle(coordinate, radius=1000, color='#000000', fill=True).add_child(folium.Popup(row['Launch Site'])).add_to(site_map)
    folium.map.Marker(coordinate, icon=DivIcon(icon_size=(20,20),icon_anchor=(0,0), html='<div style="font-size: 12; color:#d35400;"><b>%s</b></div>' % row['Launch Site'], )).add_to(site_m
site_map
```

Out[8]:



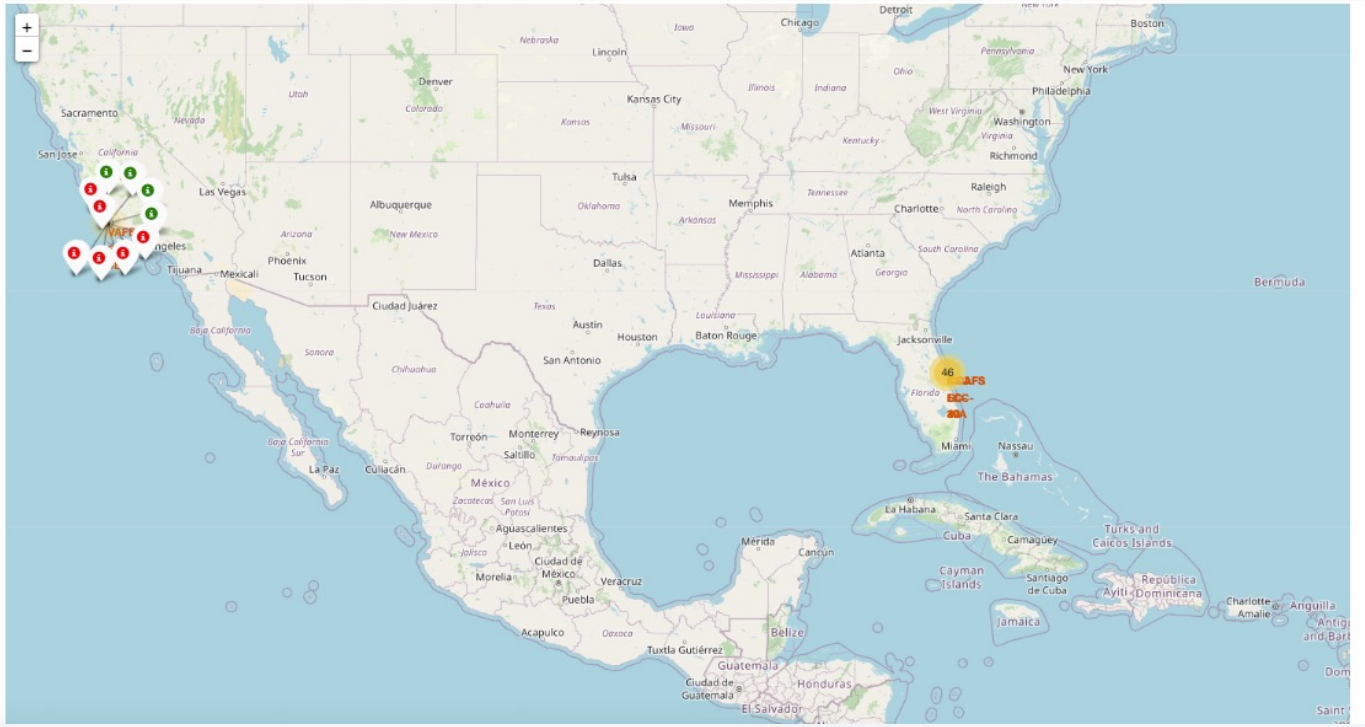
Outcome launches for each site on the map

- Replace <Folium map screenshot 2> title with an appropriate title
- 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
- This picture shown the outcome launches for each site

TODO: For each launch result in `spacex_df` data frame, add a `folium.Marker` to `marker_cluster`

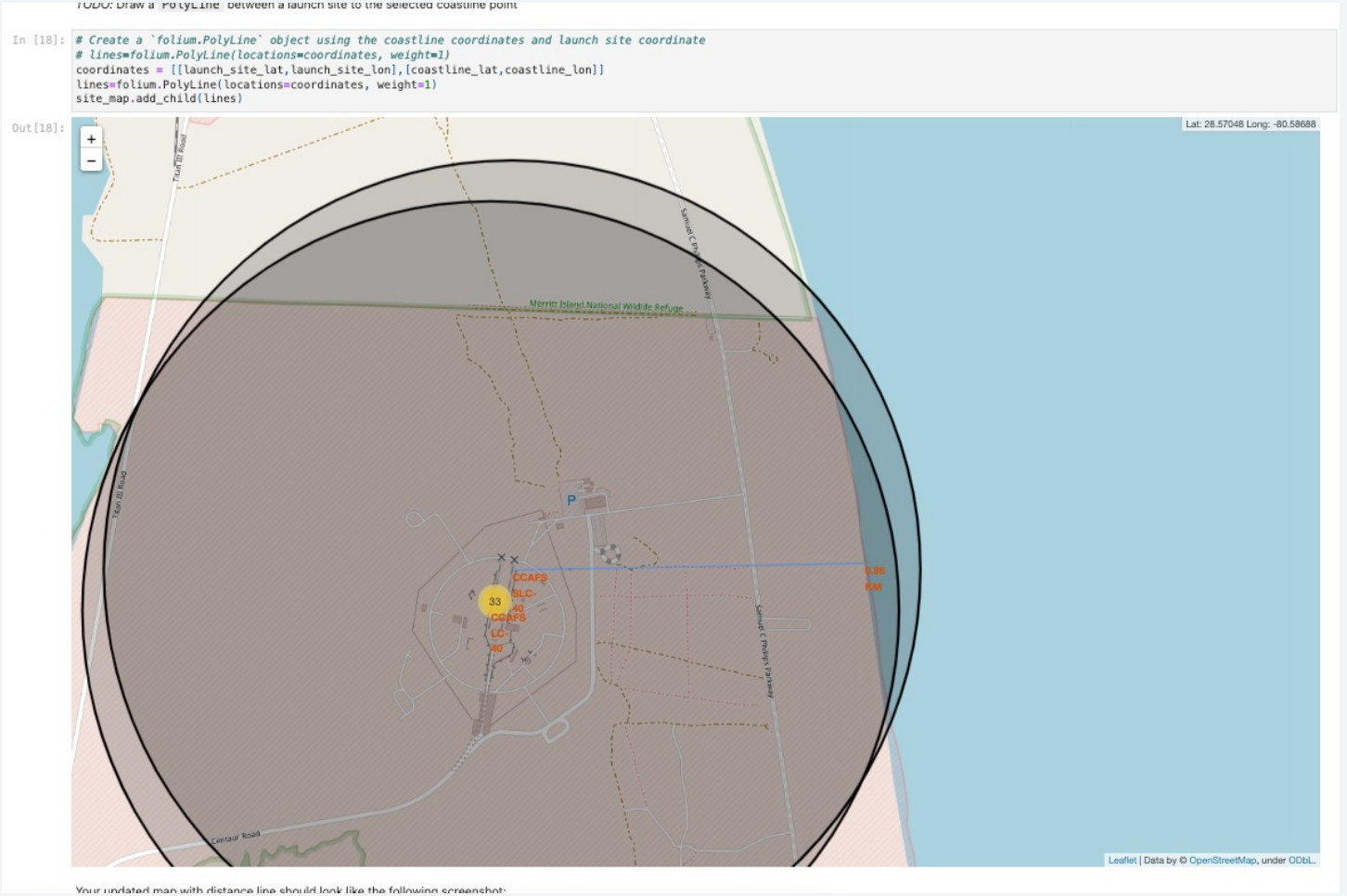
In [13]:
```
# Add marker_cluster to current site_map
site_map.add_child(marker_cluster)

# for each row in spacex_df data frame
# create a Marker object with its coordinate
# and customize the Marker's icon property to indicate if this launch was successed or failed,
# e.g., icon=folium.Icon(color='white', icon_color=row['marker_color']
for index, row in spacex_df.iterrows():
    # create and add a Marker cluster to the site map
    coordinate = [row['Lat'], row['Long']]
    folium.map.Marker(coordinate, icon=folium.Icon(color='white',icon_color=row['marker_color'])).add_to(marker_cluster)
site_map
```

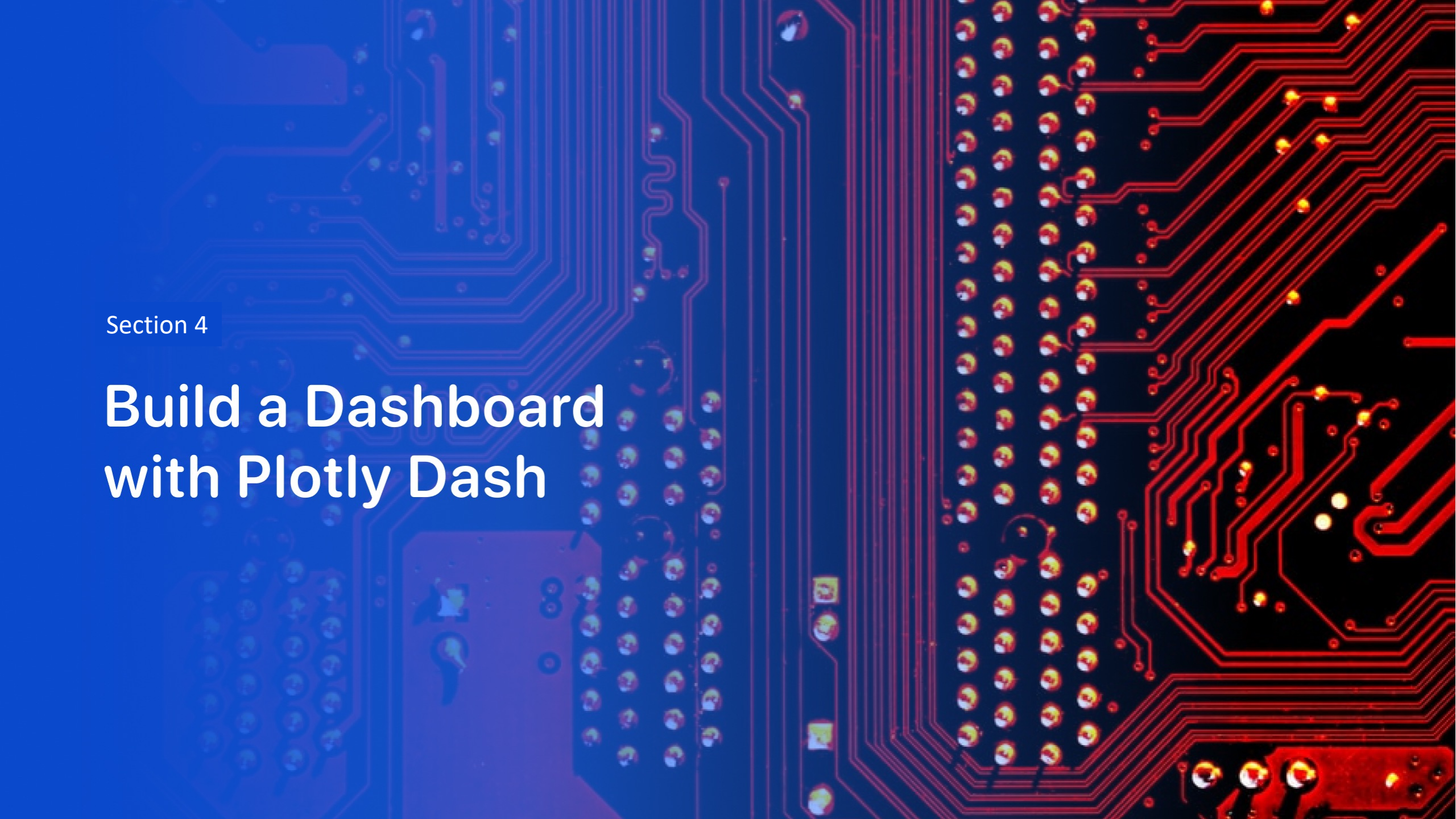
Out[13]:



Distances between a launch site to its proximities

- Replace <Folium map screenshot 3> title with an appropriate title
- 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
- This picture shown distance between launch site CCAFS SLC-40 and coastline is around 0.9 km





Section 4

Build a Dashboard with Plotly Dash

<Dashboard Screenshot 1>

- Replace <Dashboard screenshot 1> title with an appropriate title
- Show the screenshot of launch success count for all sites, in a piechart
- Explain the important elements and findings on the screenshot

<Dashboard Screenshot 2>

- Replace <Dashboard screenshot 2> title with an appropriate title
- Show the screenshot of the piechart for the launch site with highest launch success ratio
- Explain the important elements and findings on the screenshot

<Dashboard Screenshot 3>

- Replace <Dashboard screenshot 3> title with an appropriate title
- 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.

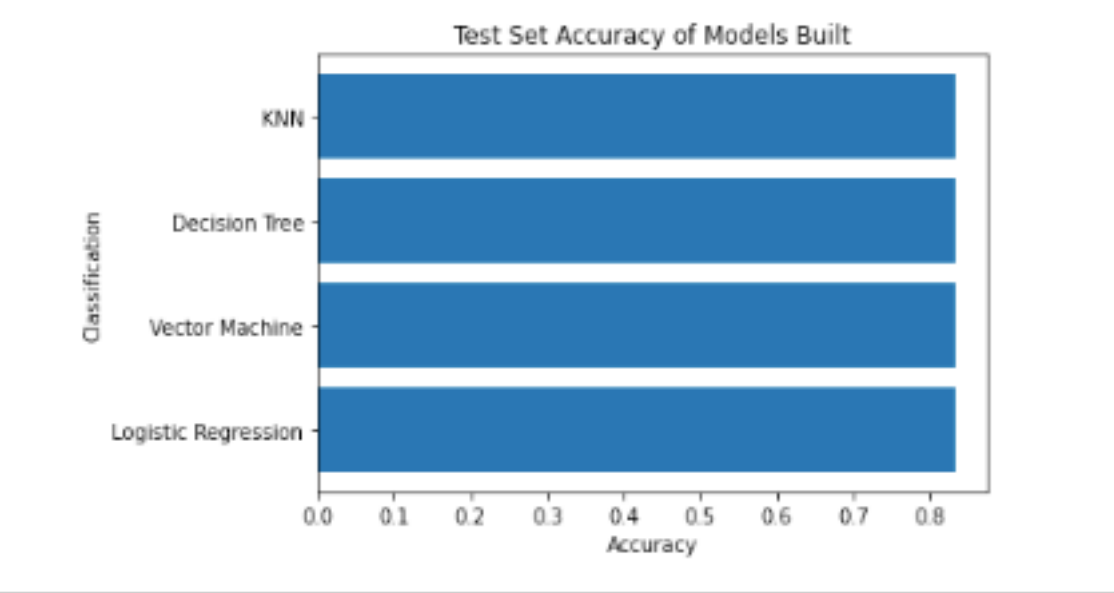
Section 5

Predictive Analysis (Classification)

Classification Accuracy

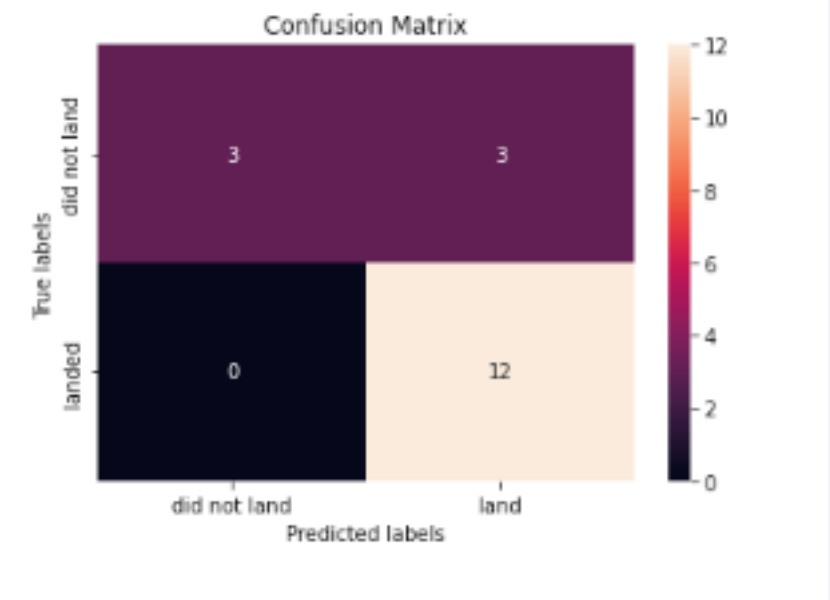
- Visualize the built model accuracy for all built classification models, in a bar chart

- Find which model has the highest classification accuracy
- All models came back with the same accuracy – 83.33%



Confusion Matrix

- Show the confusion matrix of the best performing model with an explanation
- All models came back with the same confusion matrix
-



Conclusions

- Point 1
- Point 2
- Point 3
- Point 4
- …

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!