



IBM Developer  
SKILLS NETWORK

# Winning Space Race with Data Science

Pipitton Sanseea  
30/7/2022



# Outline

---

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

# Executive Summary

---

In this study we try to predict the outcome of a first stage launch of rocket similar to Falcon 9 type. First, we collect the data from 1. SpaceX API and 2. web scraping. Then, we clean the data (replace null value and create outcome Boolean from data). We visualize the data to find out which method do we will use for analyze the outcome of each launch. We also display the outcome with dash and plotly before analyze. Finally, we analyze with classification supervised machine learning method. We found that log-regression and decision tree have the most accuracy of 0.83 for the data.

# Introduction

---

In the space traveling era, the commercial space company are making affordable rocket for everyone. The most successful company is SpaceX. Because they can reuse the first stage rocket. This reduce the cost for each launch, SpaceX cost 62 MUSD, where other cost around 165 MUSD.

Falcon 9 rocket is the model that SpaceX use in the first stage. We want to determine the price of each launch by observing Falcon 9 data. First, we visualize the data to find the correlation between factors (e.g., launch site, payload mass and booster version) and outcomes (class). Then we predict the outcome by using machine learning approaches.



Section 1

# Methodology

# Methodology

---

## Executive Summary

- Data collection methodology:
  - Describe how data was collected
- Perform data wrangling
  - Describe how data was processed
- 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

---

We collect SpaceX data with 2 method, SpaceX API and web scrapping.

- For API, we use request library to get a responses object from SpaceX API. Then we create a dictionary from data. We see missing data and call API to fill data into dictionary. Final, we store it to data frame.
- For web scraping method, we get response from Falcon 9 Wikipedia page. Then we create object from BeautifulSoup. We extract all table from object and select third table, which contain the information that we want. We create dictionary from tag `<th>`, which are columns name. Then we add data to dictionary from tag `<td>` if the head column is correct. Final, we store it to data frame.

# Data Collection – SpaceX API

1. Request to API
2. Normalize
3. Fill table with data from API
4. We get a table with 90 rows × 17 columns
5. Replace payload mass null value with mean
6. Store the data into data frame using pandas

## Result

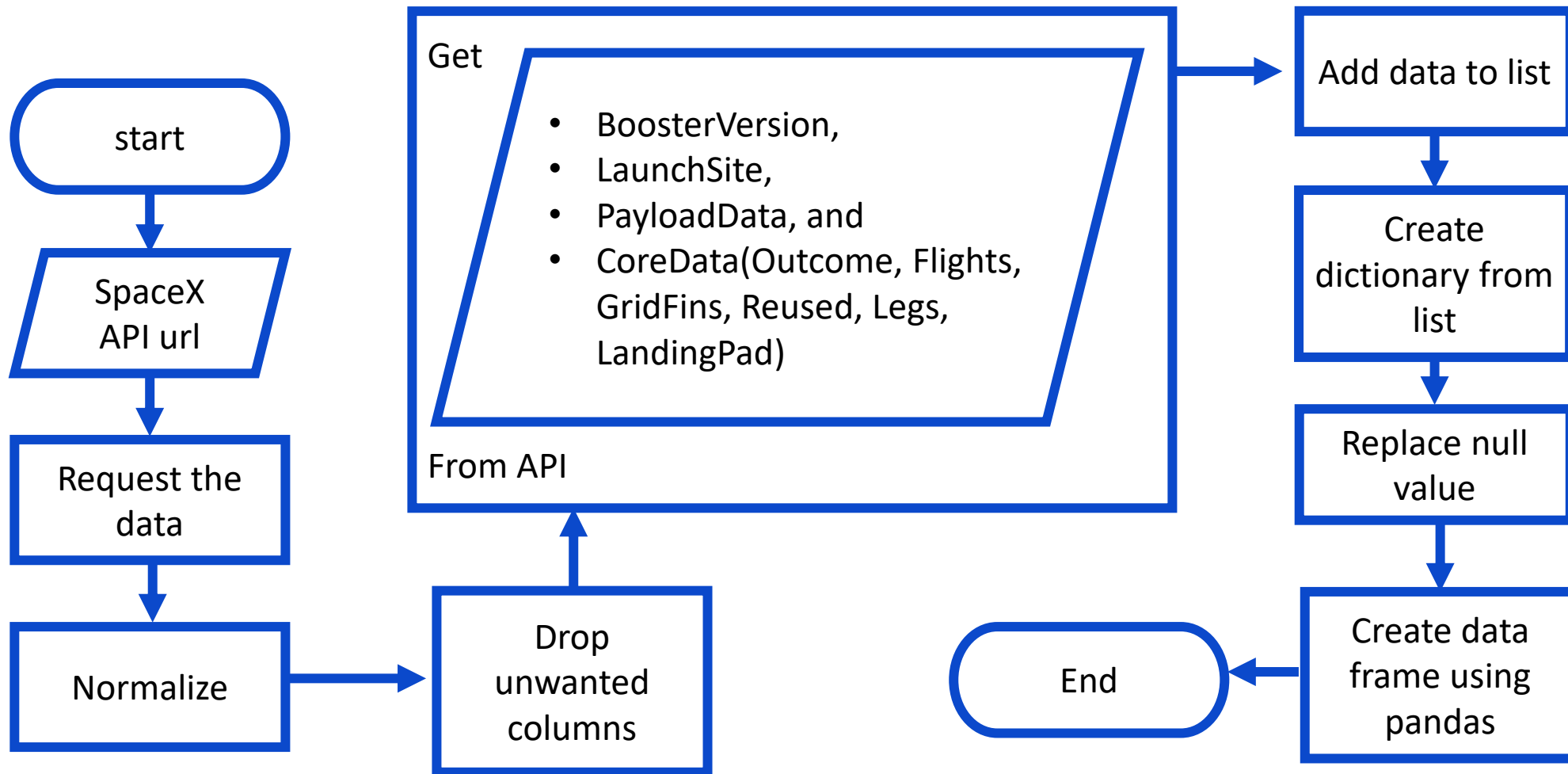
[ 34 ]:																	
	Index	FlightNumber	Date	BoosterVersion	PayloadMass	Orbit	LaunchSite	Outcome	Flights	GridFins	Reused	Legs		LandingPad	Block	ReusedCount	Ser
	0	4	1	2010-06-04	Falcon 9	6123.547647	LEO	CCSFS SLC 40	None	None	1	False	False	False	None	1.0	0 B00
	1	5	2	2012-05-22	Falcon 9	525.000000	LEO	CCSFS SLC 40	None	None	1	False	False	False	None	1.0	0 B00
	2	6	3	2013-03-01	Falcon 9	677.000000	ISS	CCSFS SLC 40	None	None	1	False	False	False	None	1.0	0 B00
	3	7	4	2013-09-29	Falcon 9	500.000000	PO	VAFB SLC 4E	False	Ocean	1	False	False	False	None	1.0	0 B10
	4	8	5	2013-12-03	Falcon 9	3170.000000	GTO	CCSFS SLC 40	None	None	1	False	False	False	None	1.0	0 B10
	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
	85	89	86	2020-09-03	Falcon 9	15600.000000	VLEO	KSC LC 39A	True	ASDS	2	True	True	True	Se9e3032383ecb6bb234e7ca	5.0	12 B10
	86	90	87	2020-10-06	Falcon 9	15600.000000	VLEO	KSC LC 39A	True	ASDS	3	True	True	True	Se9e3032383ecb6bb234e7ca	5.0	12 B10
	87	91	88	2020-10-18	Falcon 9	15600.000000	VLEO	KSC LC 39A	True	ASDS	6	True	True	True	Se9e3032383ecb6bb234e7ca	5.0	12 B10
	88	92	89	2020-10-24	Falcon 9	15600.000000	VLEO	CCSFS SLC 40	True	ASDS	3	True	True	True	Se9e3032383ecb6bb234e7cc	5.0	12 B10
	89	93	90	2020-11-05	Falcon 9	3681.000000	MEO	CCSFS SLC 40	True	ASDS	1	True	False	True	Se9e3032383ecb6bb234e7ca	5.0	7 B10

90 rows × 18 columns

[https://github.com/pipitton-s/SpaceX\\_assignment/blob/main/1\\_data\\_collection.ipynb](https://github.com/pipitton-s/SpaceX_assignment/blob/main/1_data_collection.ipynb)



# Data Collection – SpaceX API

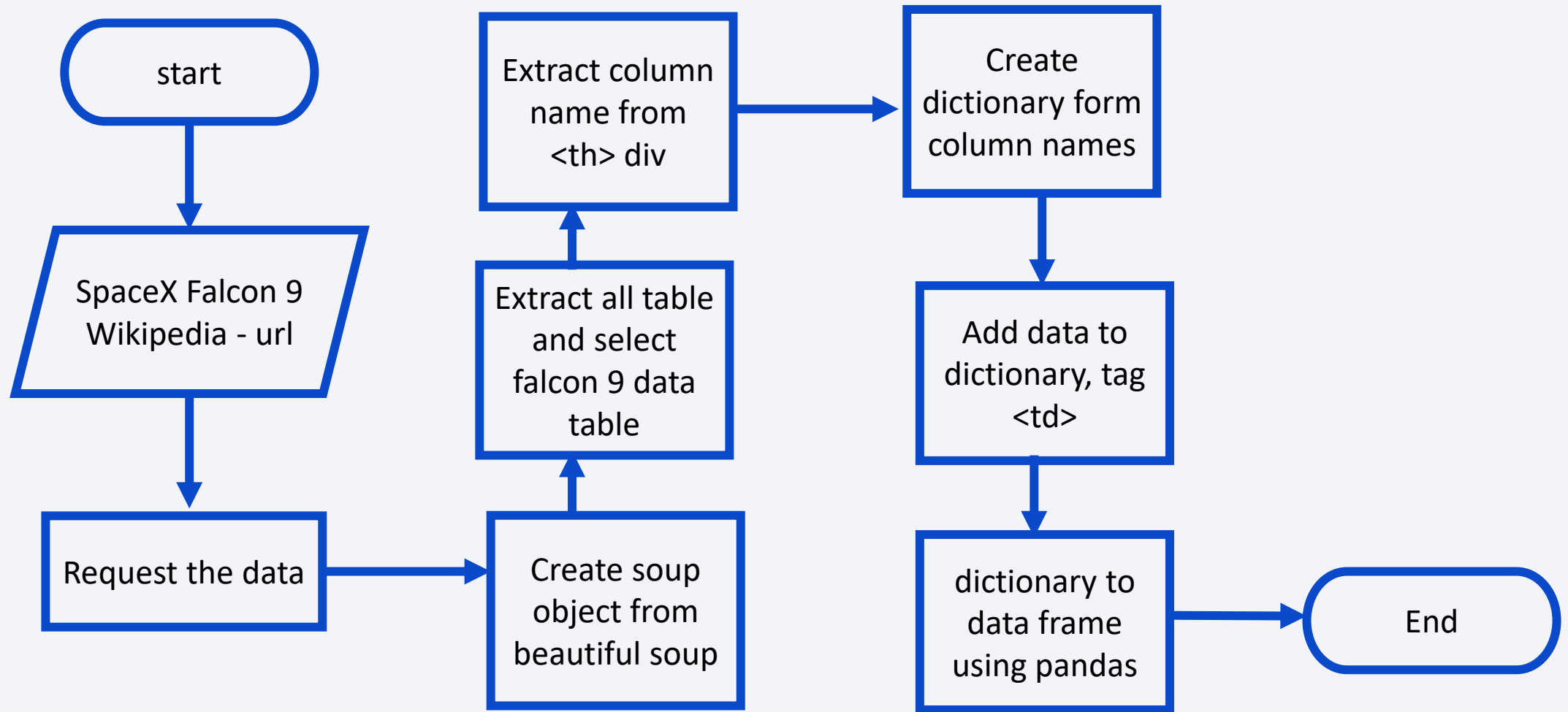


# Data Collection - Scraping

1. Request to Falcon 9 Wikipedia page
2. Create soup object from response
3. Find the table using find\_all
4. Get column names from tag <th>
5. Get data from tag <td> if columns name is correct
6. Store the data into data frame using pandas

[https://github.com/pipitton-s/SpaceX\\_assignment/blob/main/2\\_web\\_scraping.ipynb](https://github.com/pipitton-s/SpaceX_assignment/blob/main/2_web_scraping.ipynb)

# Data Collection - Scraping



# Data Wrangling

- We use data from SpaceX API lab.
- Then, we clean outcome column. Only outcomes with successfully landed are 1 other 0, then call it class.
- Store it to **"dataset\_part\_2.csv"**
- We want to find the **correlation** between each variable, to determine which is **features** before we analyze with supervised machine learning.

landing_outcomes	output
True ASDS	1
None None	0
True RTLS	1
False ASDS	0
True Ocean	1
False Ocean	0
None ASDS	0
False RTLS	0

[https://github.com/pipitton-s/SpaceX\\_assignment/blob/main/3\\_data\\_wrangling.ipynb](https://github.com/pipitton-s/SpaceX_assignment/blob/main/3_data_wrangling.ipynb)

# EDA with Data Visualization

---

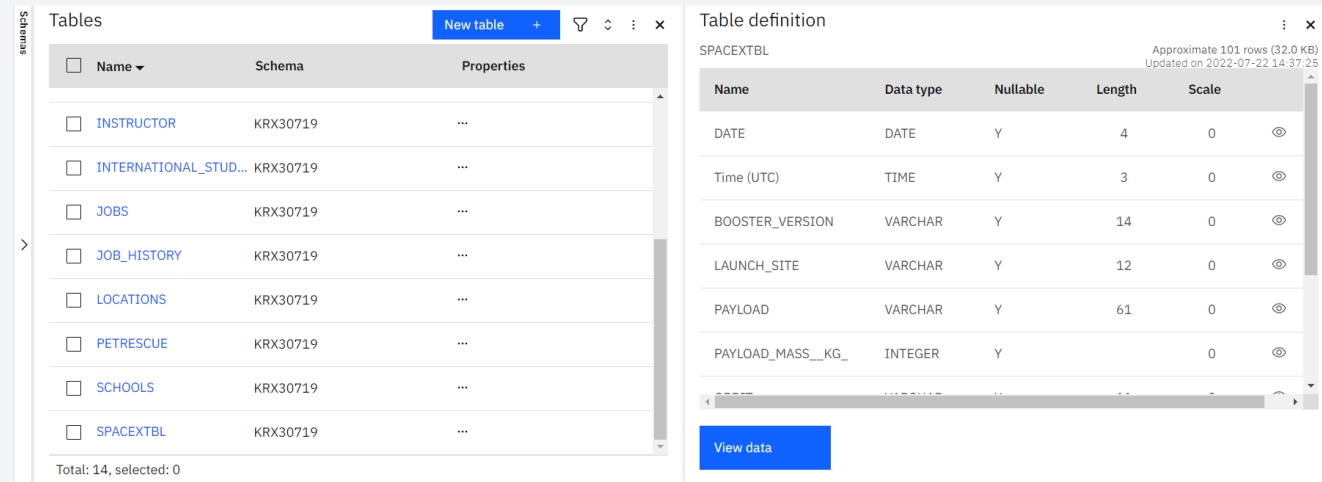
- We use scatter plot to plot between variables to see if they have correlation.
- Bar plot between launch site and success rate to determine success rate for each site.
- Line chart between year and success rate to determine success rate for each year.
- We create dummy variables to categorical columns to have data frame with 90 rows  $\times$  80 columns. Convert 'GridFins', 'Reused' and 'Legs' from Boolean to float type.

[https://github.com/pipitton-s/SpaceX\\_assignment/blob/main/4\\_EDA\\_data\\_visualize.ipynb](https://github.com/pipitton-s/SpaceX_assignment/blob/main/4_EDA_data_visualize.ipynb)



# EDA with SQL

- We upload data frame to data base and use SQL to overview the data.
- Execute SQL queries to answer assignment questions



The screenshot displays a database management interface. On the left, a 'Tables' panel lists several tables under the 'KRX30719' schema: INSTRUCTOR, INTERNATIONAL\_STUD..., JOBS, JOB\_HISTORY, LOCATIONS, PETRESCUE, SCHOOLS, and SPACEXTBL. The 'SPACEXTBL' table is selected. On the right, the 'Table definition' panel shows the schema for 'SPACEXTBL', which has approximately 101 rows (32.0 KB) and was last updated on 2022-07-22 14:37:25. The table has five columns: DATE (DATE, nullable), Time (UTC) (TIME, nullable), BOOSTER\_VERSION (VARCHAR, nullable), LAUNCH\_SITE (VARCHAR, nullable), and PAYLOAD (VARCHAR, nullable). The PAYLOAD column has a length of 61. The PAYLOAD\_MASS\_KG\_ column is an INTEGER, nullable, with a scale of 0. A 'View data' button is visible at the bottom of the table definition panel.

Name	Data type	Nullable	Length	Scale
DATE	DATE	Y	4	0
Time (UTC)	TIME	Y	3	0
BOOSTER_VERSION	VARCHAR	Y	14	0
LAUNCH_SITE	VARCHAR	Y	12	0
PAYLOAD	VARCHAR	Y	61	0
PAYLOAD_MASS_KG_	INTEGER	Y		0

[https://github.com/pipitton-s/SpaceX\\_assignment/blob/main/5\\_EDA\\_SQL.ipynb](https://github.com/pipitton-s/SpaceX_assignment/blob/main/5_EDA_SQL.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
- Add the GitHub URL of your completed interactive map with Folium map, as an external reference and peer-review purpose

[https://github.com/pipitton-s/SpaceX\\_assignment/blob/main/6\\_folium.ipynb](https://github.com/pipitton-s/SpaceX_assignment/blob/main/6_folium.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/pipitton-s/SpaceX\\_assignment/blob/main/7\\_spacex\\_dash.py](https://github.com/pipitton-s/SpaceX_assignment/blob/main/7_spacex_dash.py)

# Predictive Analysis (Classification)

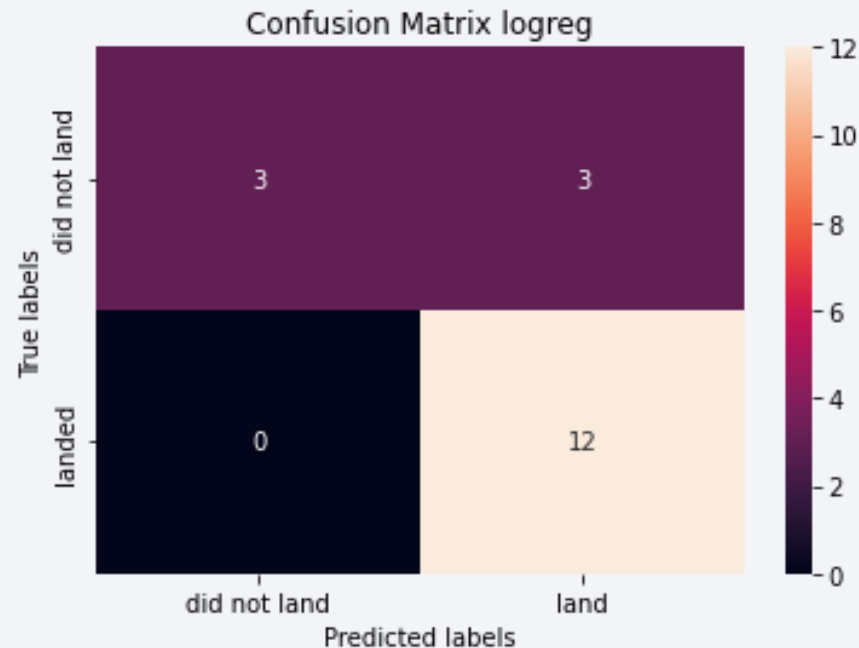
---

- we analyze with classification supervised machine learning method.
- Log-regression has accuracy 0.83
- Support vector machine has accuracy 0.67
- Decision tree has accuracy 0.83
- K-nearest neighbor has accuracy 0.61
- We found that log-regression and decision tree have the most accuracy of 0.83 for the data.

[https://github.com/pipitton-s/SpaceX\\_assignment/blob/main/8\\_analyze.ipynb](https://github.com/pipitton-s/SpaceX_assignment/blob/main/8_analyze.ipynb)

# Results

- The problem is classification.
- Log-regression and decision tree have the most accuracy of 0.83.
- Interactive analytics demo in screenshots
- Predictive analysis results





The background of the slide is an abstract composition. It features a dark blue base color. Overlaid on this are numerous diagonal streaks in shades of red and cyan. A faint, light blue grid pattern is also visible, particularly in the lower half of the image. The overall effect is dynamic and technological.

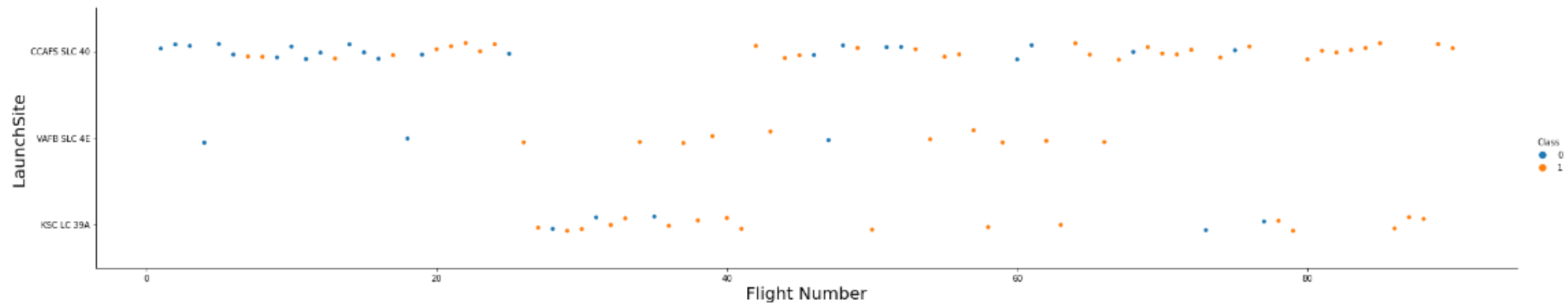
Section 2

# Insights drawn from EDA



# Flight Number vs. Launch Site

```
[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("LaunchSite",fontsize=20)
plt.show()
```

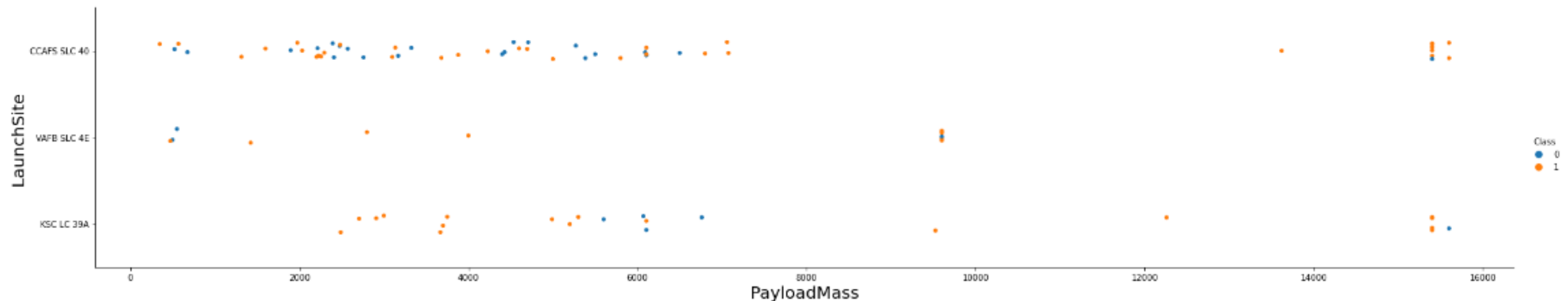


- There is no correlation between CCAFS LC-40 and flight number.
- VAFB SLC and KSC LC-39A tend to have a good outcome when flight number is high

# Payload vs. Launch Site

We also want to observe if there is any relationship between launch sites and their payload mass.

```
[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 val  
sns.catplot(y="LaunchSite", x="PayloadMass", hue="Class", data=df, aspect = 5)  
plt.xlabel("PayloadMass", fontsize=20)  
plt.ylabel("LaunchSite", fontsize=20)  
plt.show()
```



- Higher payload mass has more success rate

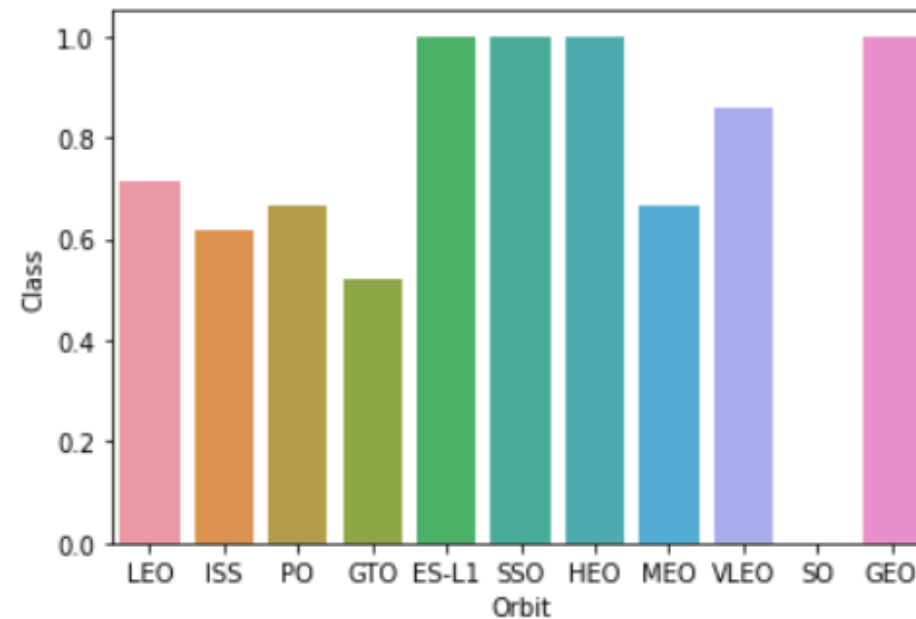
# Success Rate vs. Orbit Type

- ES-L1, SSO, HEO and GEO have the most success rate.

```
[4]: # HINT use groupby method on Orbit column and get the mean of Class column
Orbit_rate = df.groupby(['Orbit'], as_index=False, sort=False)['Class'].mean()

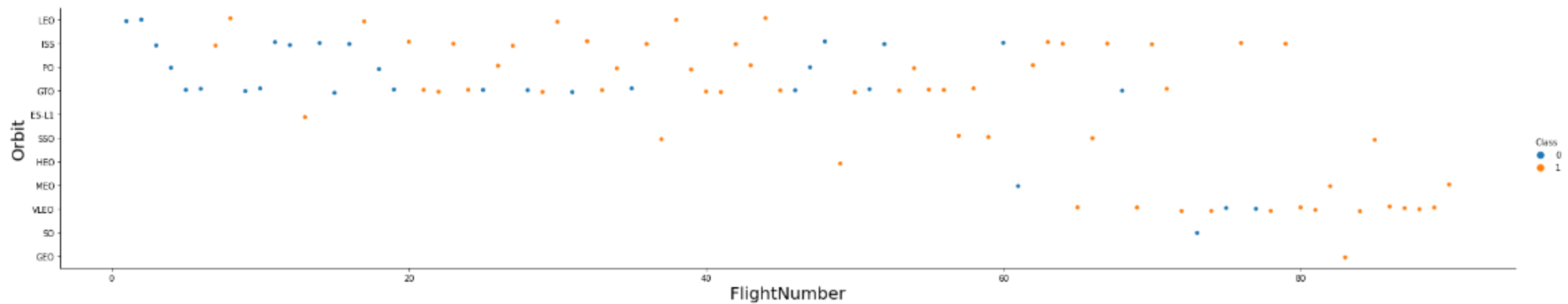
sns.barplot(x="Orbit", y="Class", data=Orbit_rate)
```

```
[4]: <AxesSubplot:xlabel='Orbit', ylabel='Class'>
```



# Flight Number vs. Orbit Type

```
[73]: # 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("FlightNumber",fontsize=20)
plt.ylabel("Orbit",fontsize=20)
plt.show()
```



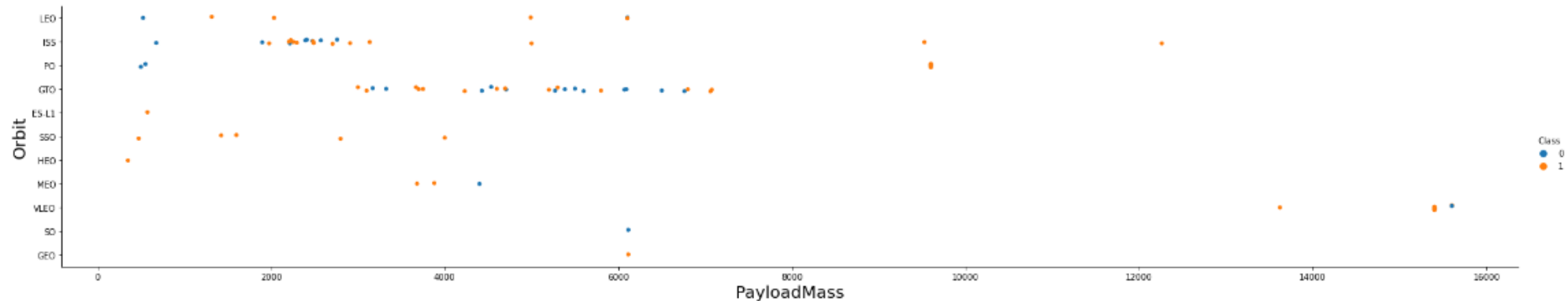
- The LEO and VLEO success relate to flight number.



# Payload vs. Orbit Type

```
[75]: # 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("PayloadMass",fontsize=20)  
plt.ylabel("Orbit",fontsize=20)  
plt.show()
```

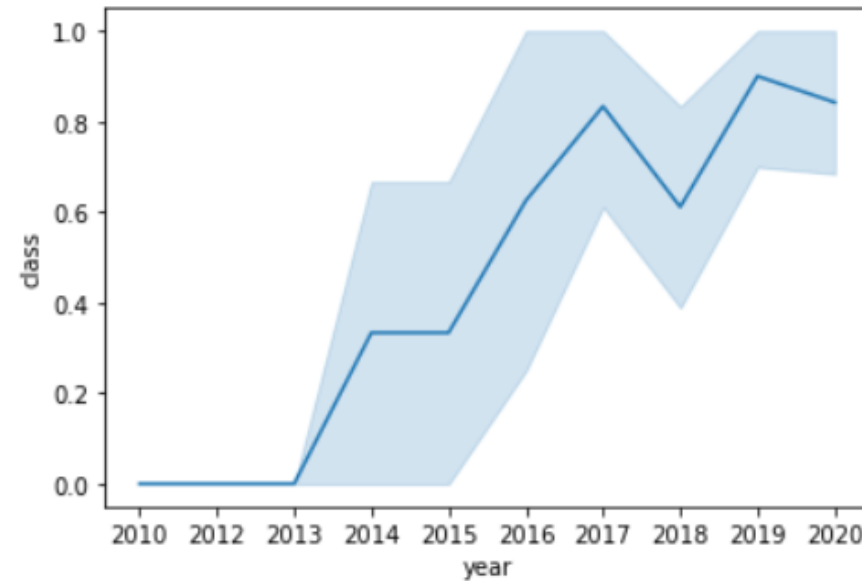


- With heavy payloads the successful landing or positive landing rate are more for Po, LEO and ISS.

# Launch Success Yearly Trend

```
[142]: # Plot a line chart with x axis to be the extracted year and y axis to be the success rate  
sns.lineplot(x='year',y='class',data=year_df)
```

```
[142]: <AxesSubplot:xlabel='year', ylabel='class'>
```



- Success rate stop increasing since 2019.

# All Launch Site Names

---

- Find the names of the unique launch sites

```
[14]: %sql select distinct(launch_site) from SPACEXTBL
* ibm_db_sa://krx30719:***@ea286ace-86c7-4d5b-8580-3fbfa46b1c
sqlite:///my_data1.db
Done.
```

```
[14]: 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'

```
[24]: %sql select launch_site from SPACEXTBL where launch_site like '%CCA%' LIMIT 5

* ibm_db_sa://krx30719:***@ea286ace-86c7-4d5b-8580-3fbfa46b1c66.bs2io90l08kqb1oc
1505/bludb
sqlite:///my_data1.db
Done.
```

```
[24]: launch_site
```

CCAFS LC-40
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40
CCAFS LC-40

# Total Payload Mass

---

- Calculate the total payload carried by boosters from NASA

```
[31]: %sql select sum(payload_mass__kg_) from SPACEXTBL where CUSTOMER = 'NASA (CRS)'  
      * ibm_db_sa://krx30719:***@ea286ace-86c7-4d5b-8580-3fbfa46b1c66.bs2io90l08kqb1od8  
      1505/bludb  
      sqlite:///my_data1.db  
      Done.  
[31]: 1  
      45596
```



# Average Payload Mass by F9 v1.1

---

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

```
[38]: %sql select CAST(sum(payload_mass__kg_ )as float)/CAST(count(payload_mass__kg_ )as float) as avg_payload_by_f9\
      from SPACEXTBL where BOOSTER_VERSION like '%F9 v1.1%'

* ibm_db_sa://krx30719:***@ea286ace-86c7-4d5b-8580-3fbfa46b1c66.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:3
1505/bludb
      sqlite:///my_data1.db
Done.

[38]:  avg_payload_by_f9
      2534.6666666666665
```

# First Successful Ground Landing Date

---

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

```
[51]: %sql select date,landing_outcome from spacextbl \
      where date = (select min(date) from spacextbl where landing_outcome='Success (ground pad)')

* ibm_db_sa://krx30719:***@ea286ace-86c7-4d5b-8580-3fbfa46b1c66.bs2io90l08kqb1od8l1cg.database:
1505/bludb
sqlite:///my_data1.db
Done.
```

DATE	landing_outcome
2015-12-22	Success (ground pad)

## 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

```
[57]: %sql select booster_version,landing_outcome,payload_mass__kg_ from spacextbl \
      where landing_outcome = 'Success (drone ship)' and payload_mass__kg_ between 4000 and 6000

* ibm_db_sa://krx30719:***@ea286ace-86c7-4d5b-8580-3fbfa46b1c66.bs2io90108kqb1od8lcg.databases.ap
1505/bludb
sqlite:///my_data1.db
Done.
```

```
[57]:
```

booster_version	landing_outcome	payload_mass__kg_
F9 FT B1022	Success (drone ship)	4696
F9 FT B1026	Success (drone ship)	4600
F9 FT B1021.2	Success (drone ship)	5300
F9 FT B1031.2	Success (drone ship)	5200

# Total Number of Successful and Failure Mission Outcomes

---

- Calculate the total number of successful and failure mission outcomes

## Task 7

List the total number of successful and failure mission outcomes

```
[21]: %sql select distinct(mission_outcome),count(mission_outcome) from SPACEXTBL group by mission_outcome
```

```
* ibm_db_sa://krx30719:***@ea286ace-86c7-4d5b-8580-3fbfa46b1c66.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:31505/bludb  
sqlite:///my_data1.db
```

Done.

```
[21]:
```

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

# Boosters Carried Maximum Payload

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

```
[66]: %sql select booster_version,payload_mass_kg_ from spacextbl \
      where payload_mass_kg_ = (select max(payload_mass_kg_) from spacextbl)

* ibm_db_sa://krx30719:***@ea286ace-86c7-4d5b-8580-3fbfa46b1c66.bs2io90l08kqb1od8l1cg.databases.appdomain.cl
1505/bludb
sqlite:///my_data1.db
Done.
```

```
[66]: booster_version  payload_mass_kg_
```

F9 B5 B1048.4	15600
F9 B5 B1049.4	15600
F9 B5 B1051.3	15600
F9 B5 B1056.4	15600
F9 B5 B1048.5	15600
F9 B5 B1051.4	15600
F9 B5 B1049.5	15600
F9 B5 B1060.2	15600
F9 B5 B1058.3	15600
F9 B5 B1051.6	15600
F9 B5 B1060.3	15600
F9 B5 B1049.7	15600

# 2015 Launch Records

---

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

```
[132]: %sql select substr(Date, 6, 2) as Month,\
landing_outcome,booster_version,launch_site,substr(Date,1,4) as Year\
from spacextbl where landing_outcome = 'Failure (drone ship)' and substr(Date,1,4)=2015

* ibm_db_sa://krx30719:***@ea286ace-86c7-4d5b-8580-3fbfa46b1c66.bs2io90l08kqb1od8lcg.databases.appdomain.cloud
1505/bludb
sqlite:///my_data1.db
Done.
```

```
[132]:
```

	MONTH	landing_outcome	booster_version	launch_site	YEAR
	01	Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40	2015
	04	Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40	2015

# 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

```
[25]: %sql select distinct(LANDING_OUTCOME), count(LANDING_OUTCOME) as number\
      from spacextbl \
      where date between '2010-06-04' and '2017-03-20'\
      group by LANDING_OUTCOME order by count(LANDING_OUTCOME) desc

* ibm_db_sa://krx30719:***@ea286ace-86c7-4d5b-8580-3fbfa46b1c66.bs2io90l08kqb1od8lce
sqlite:///my_data1.db
Done.
```

```
[25]:
```

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



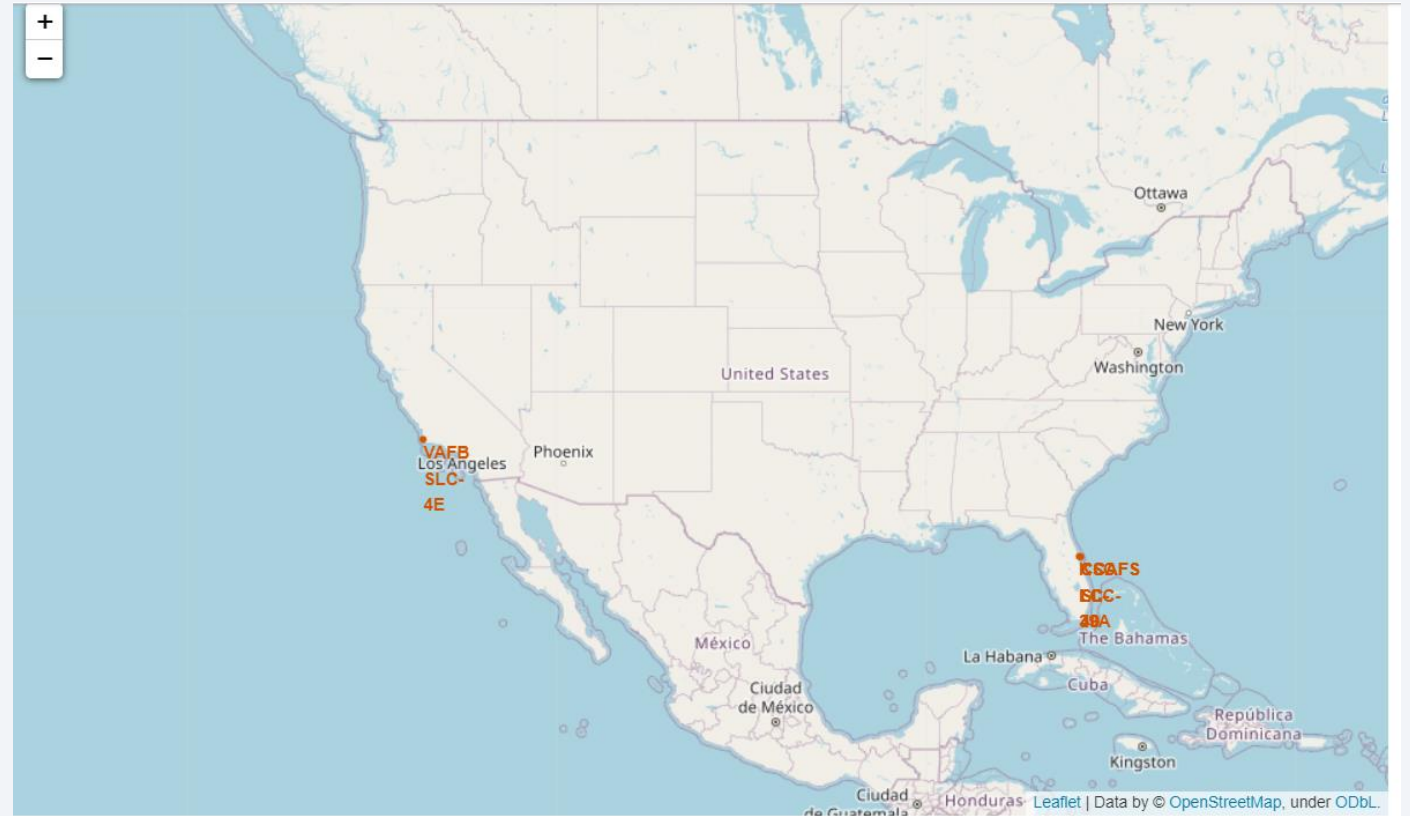
A satellite view of Earth from space, showing the curvature of the planet and city lights at night. The background is a deep blue gradient.

Section 3

# Launch Sites Proximities Analysis

# <Folium Map Screenshot 1>

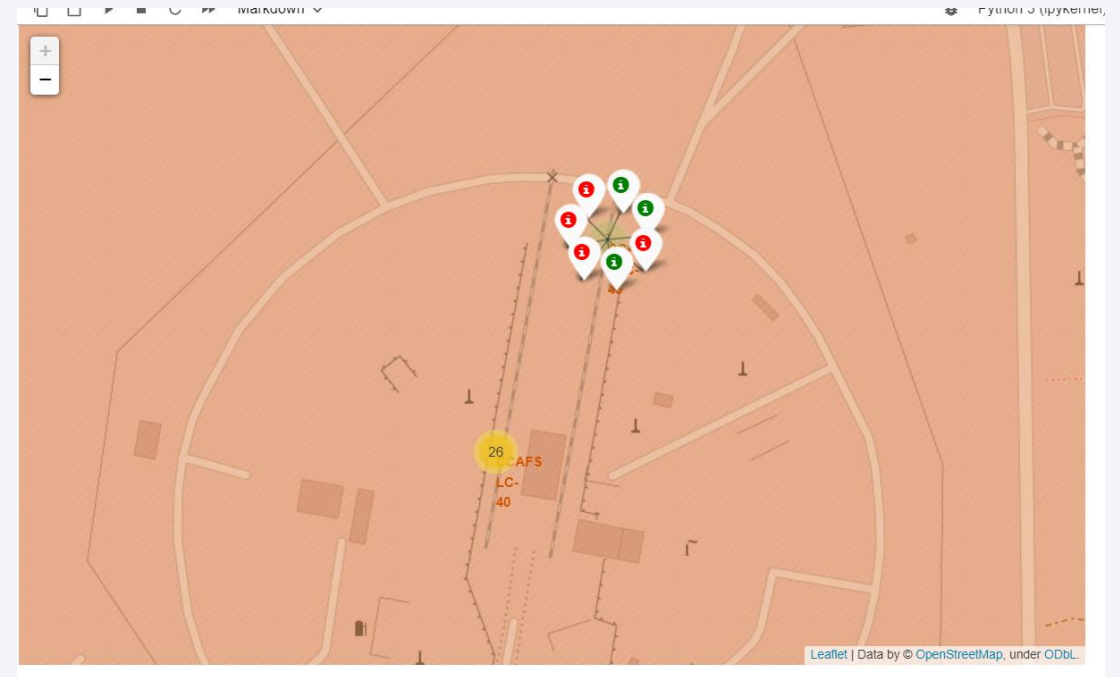
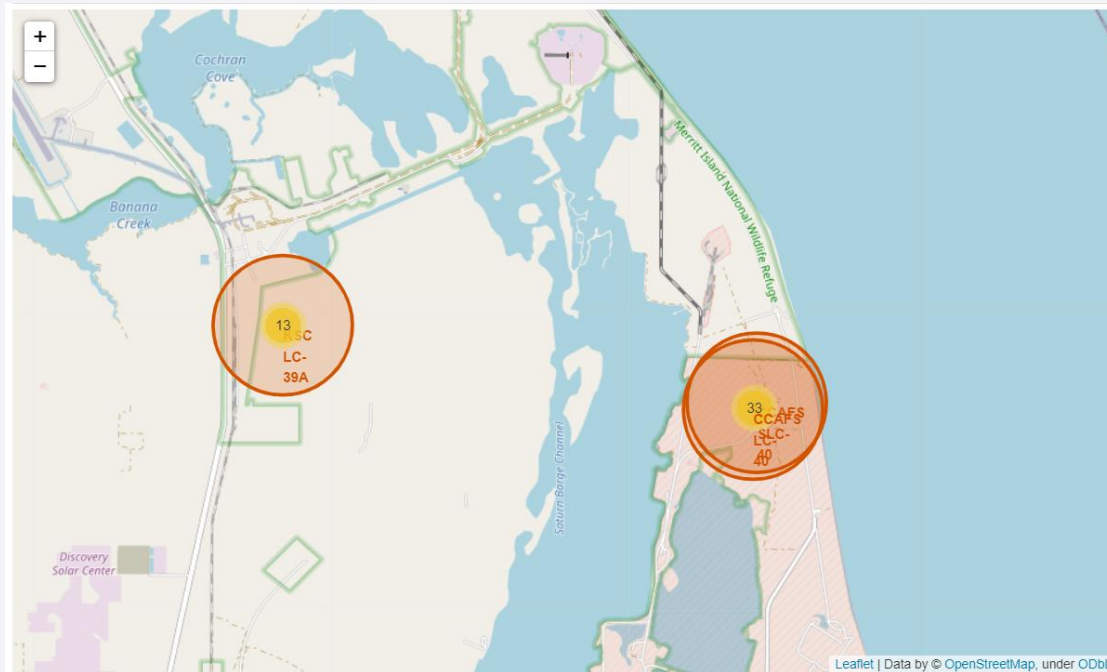
- Launch site on USA map.



- Explain the important elements and findings on the screenshot

## <Folium Map Screenshot 2>

- Mark the success/failed launches for each site on the map.



- Green marker is success outcome and red is failed.

# <Folium Map Screenshot 3>

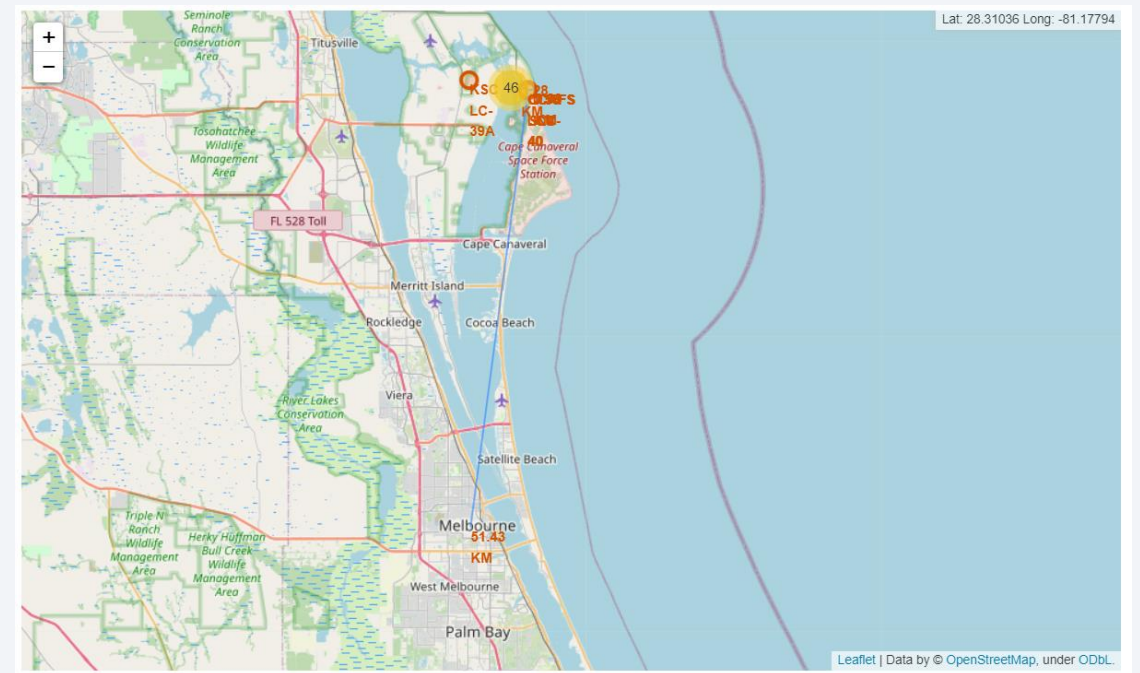
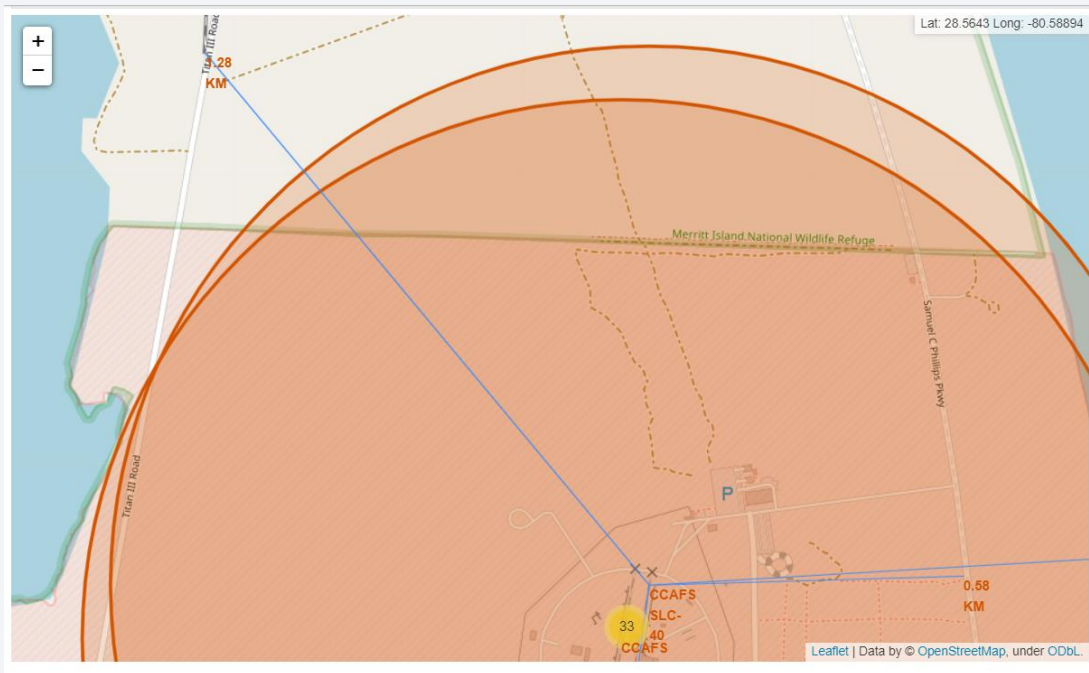
- Closest coastline and highway from CCAFS SLC-40





# <Folium Map Screenshot 3>

- Closest railway and city from CCAFS SLC-40





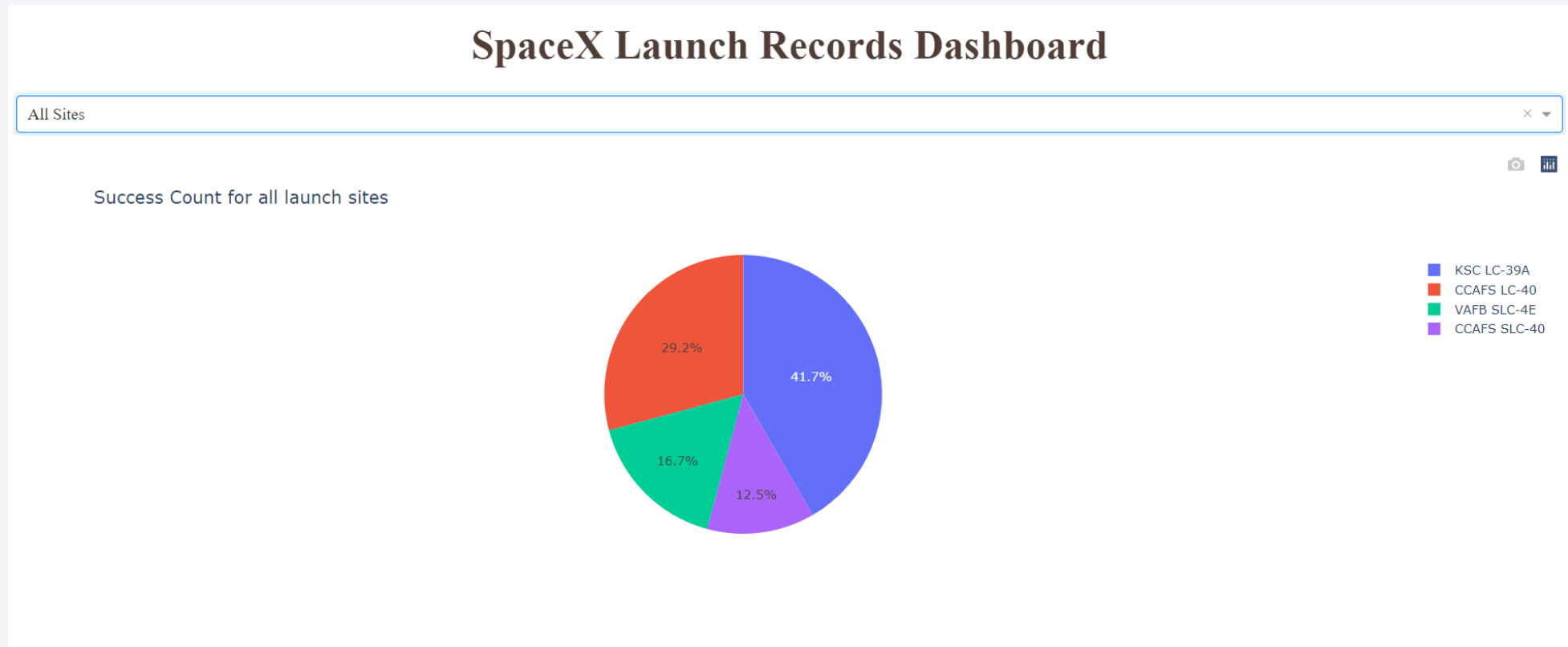
Section 4

# Build a Dashboard with Plotly Dash

# <Dashboard Screenshot 1>

---

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

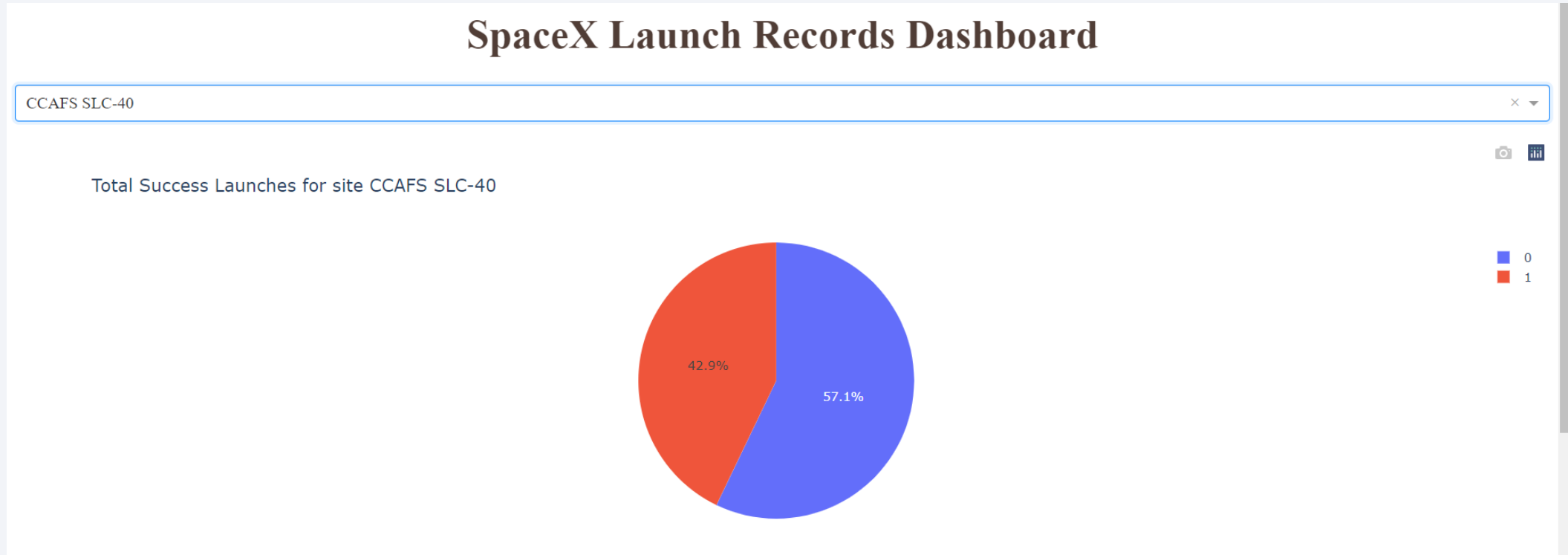




## <Dashboard Screenshot 2>

---

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



## <Dashboard Screenshot 3>

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



- Display all value when choose ALL Sites.

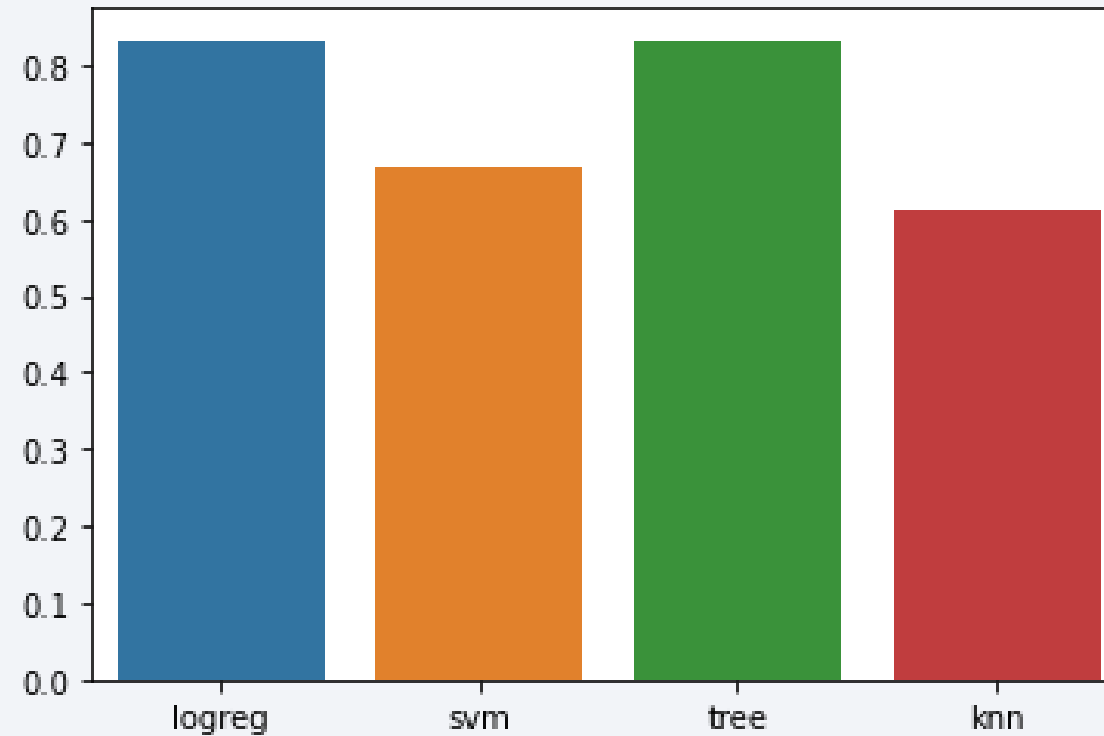


Section 5

# Predictive Analysis (Classification)

# Classification Accuracy

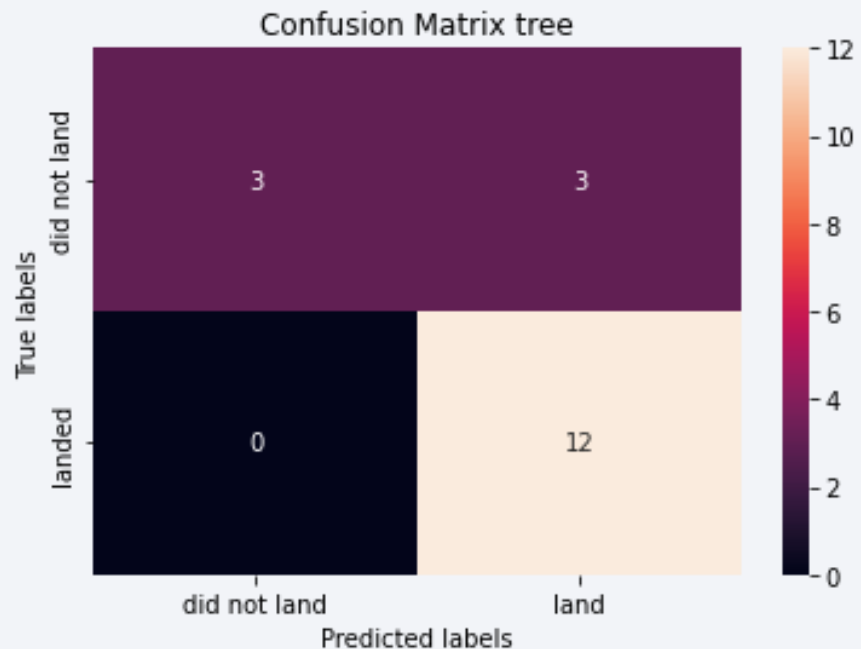
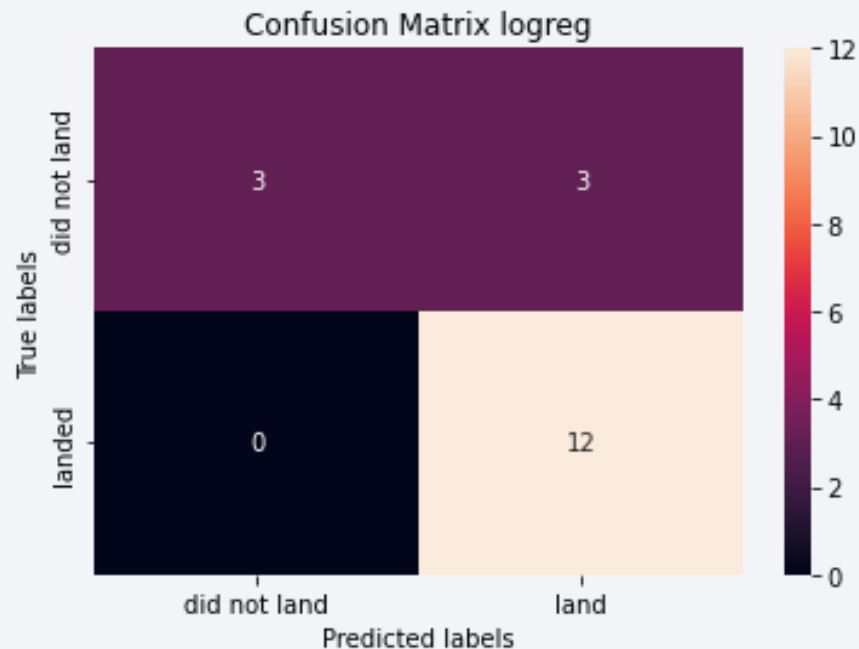
---



- The highest classification accuracy are **logistic regression** and **decision tree**
- The accuracy is 0.8333333333333334

# Confusion Matrix

- The confusion matrices show how our model predict vs actual label.
- Our test data contain 18 samples with 12 landed(1) and 6 not land(0).
- All landed label are correct.



# Conclusions

---

- We can collect the data from 1. SpaceX API and 2. web scraping
- The visualize of data suggest that our problem is to categorize the outcomes (0 or 1)
- We use 4 classification models, e.g., log regression, support vector machine, decision tree and k-nearest neighbor, we found that log regression and decision tree have the highest accuracy score of 0.83
- This study works well with the SpaceX data
- For the fast scan to the data, we think **Orbit type, Launch site, Grid fins and Legs** have significant contribution to launch outcome of the rocket.



Thank you!

