IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Dr. Stefan Coetzee
8 April 2023

Outline

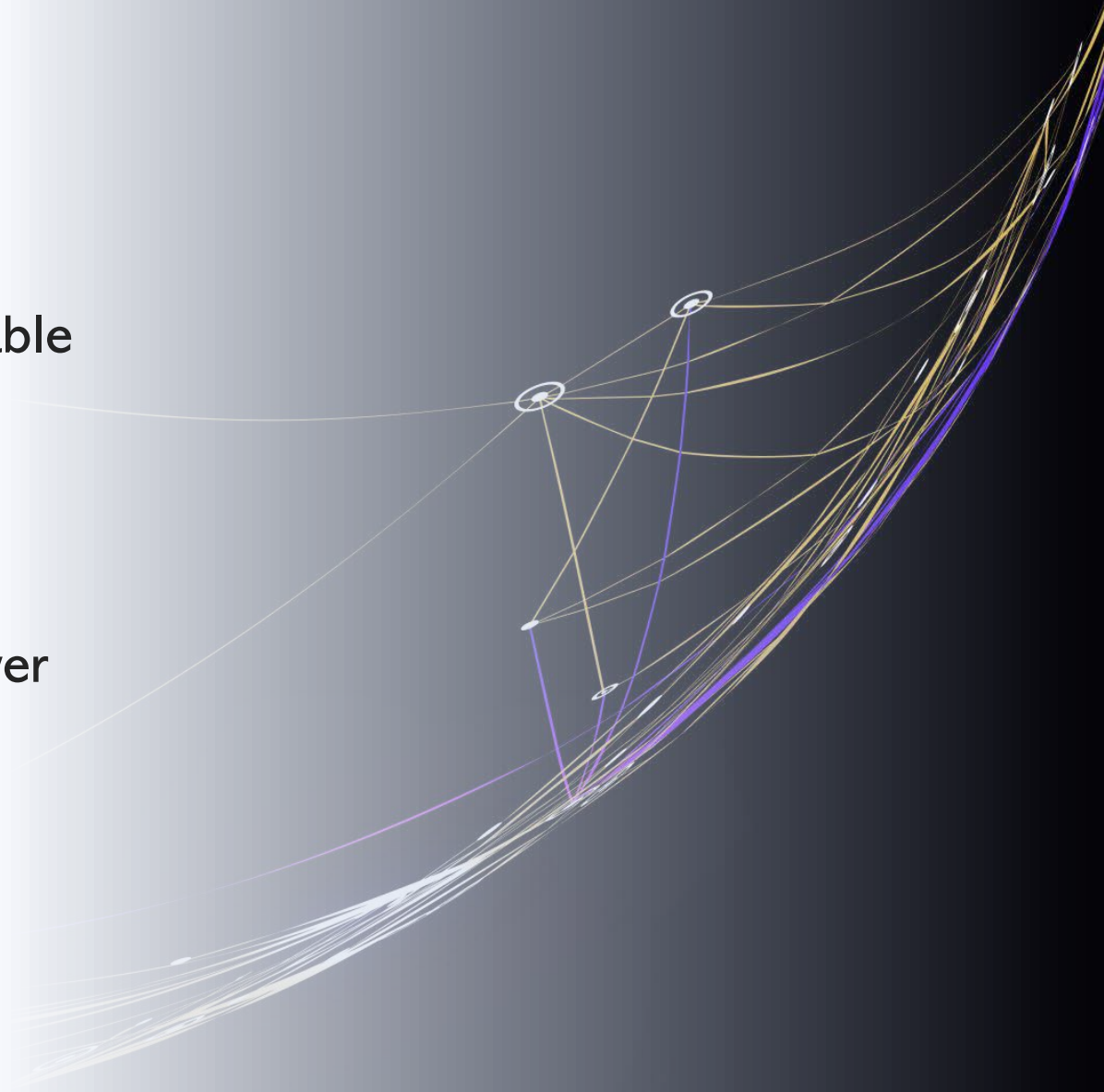
- Executive Summary
- Introduction
- Methodology
- Results
- Conclusion
- Appendix

Executive Summary

- Data Analysis:
  - API, Webscraping used for Data Collection
- Data Processing:
  - Null removal, Target/Feature Engineering, Encoding Categoricals, Standardization
- Predictive Analysis
  - KNN, SWM, Dec. Tree, Log. Reg.
- EDA:
  - Launch from KSC LC 39
  - Take a large payload
  - Launch into ES-L1, GEO, HEO, SSO

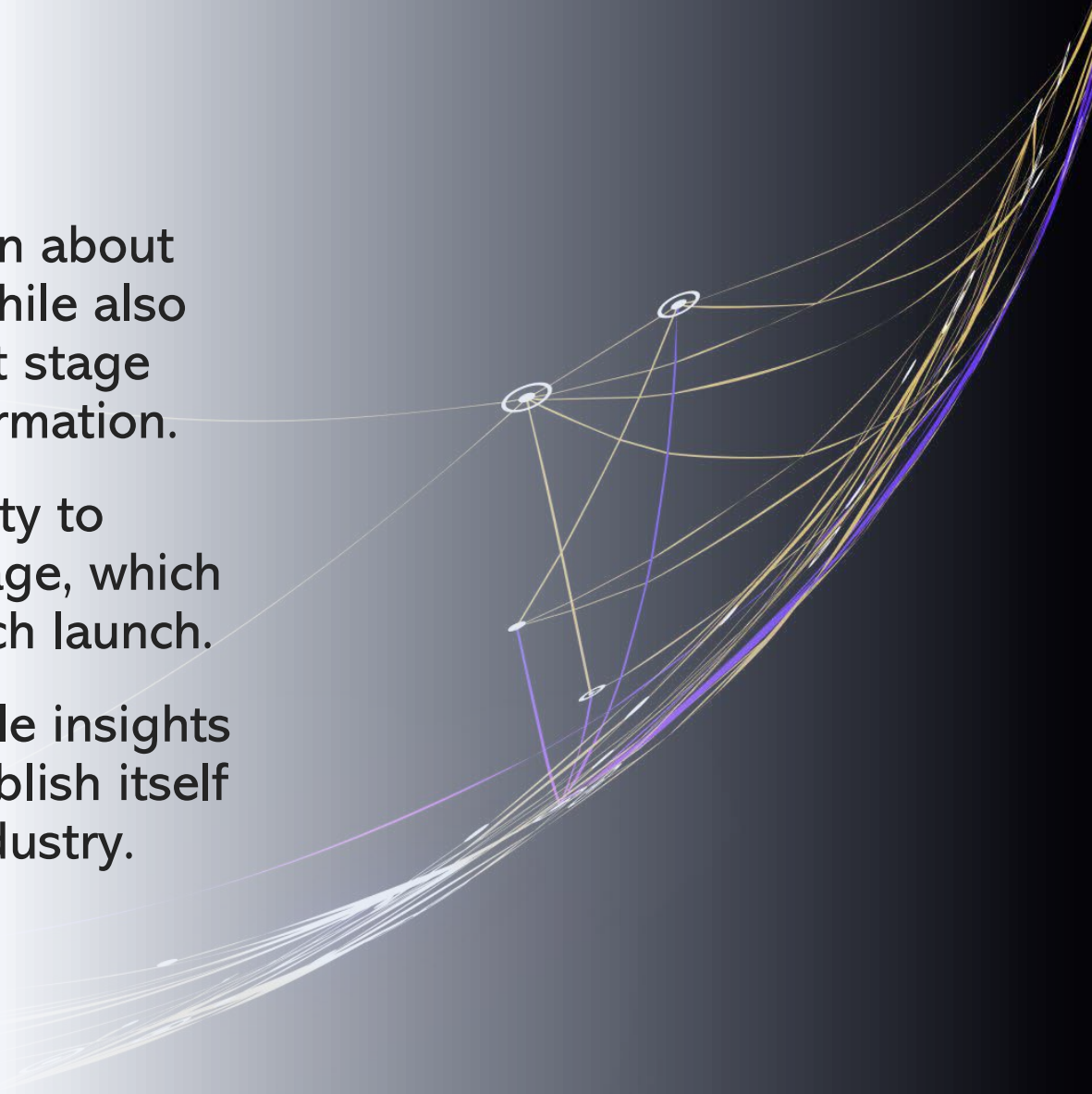
Introduction

- SpaceY to enter the market as an affordable space travel company
  - Competition with Virgin Galactic, Rocket Lab, Blue Origin, and SpaceX making space travel affordable and accessible to everyone.
- Use SpaceX's idea to reuse stages to lower cost



Introduction

- Our main objective will be to gather information about SpaceX and create dashboards for our team, while also predicting if SpaceX will be able to reuse a first stage using machine learning models and public information.
- The success of this project will rely on our ability to accurately predict the reusability of the first stage, which will have a significant impact on the cost of each launch.
- Through this project, we aim to provide valuable insights and data-driven solutions to help Space Y establish itself as a leading player in the commercial space industry.



Section 1

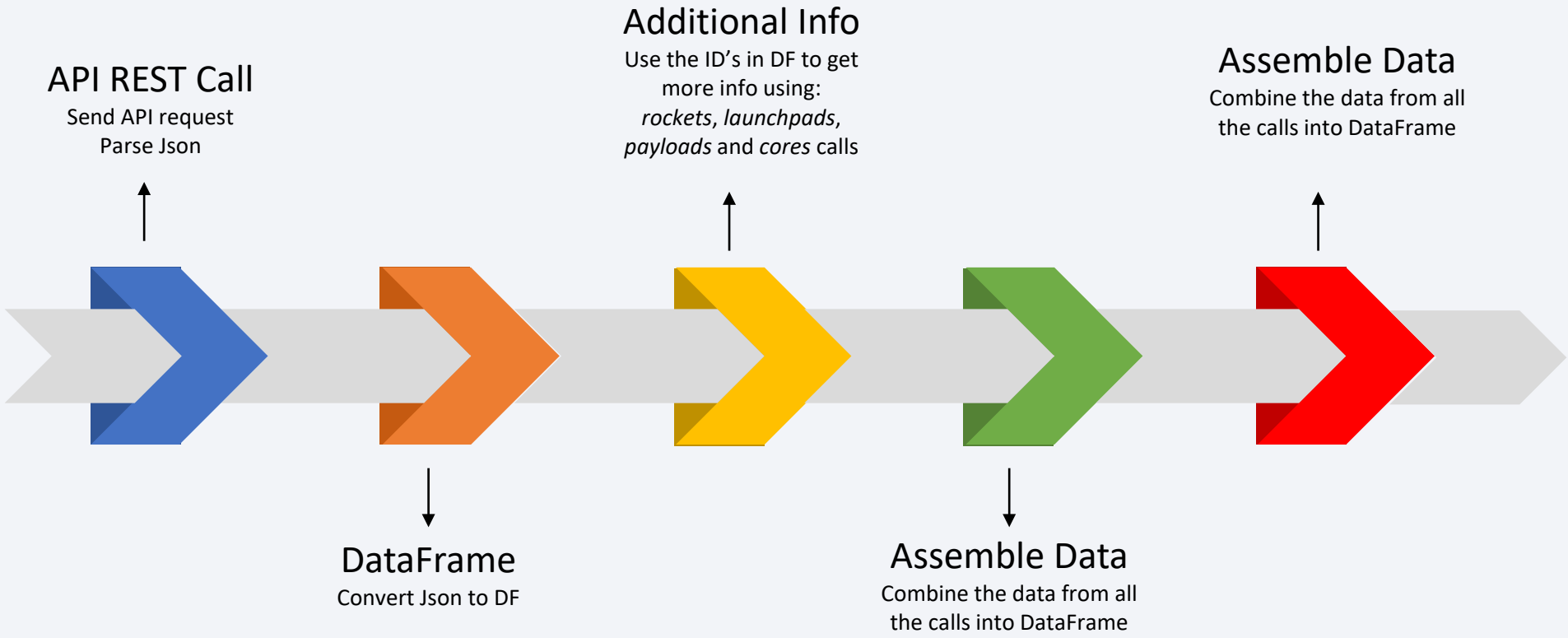
Methodology

Methodology

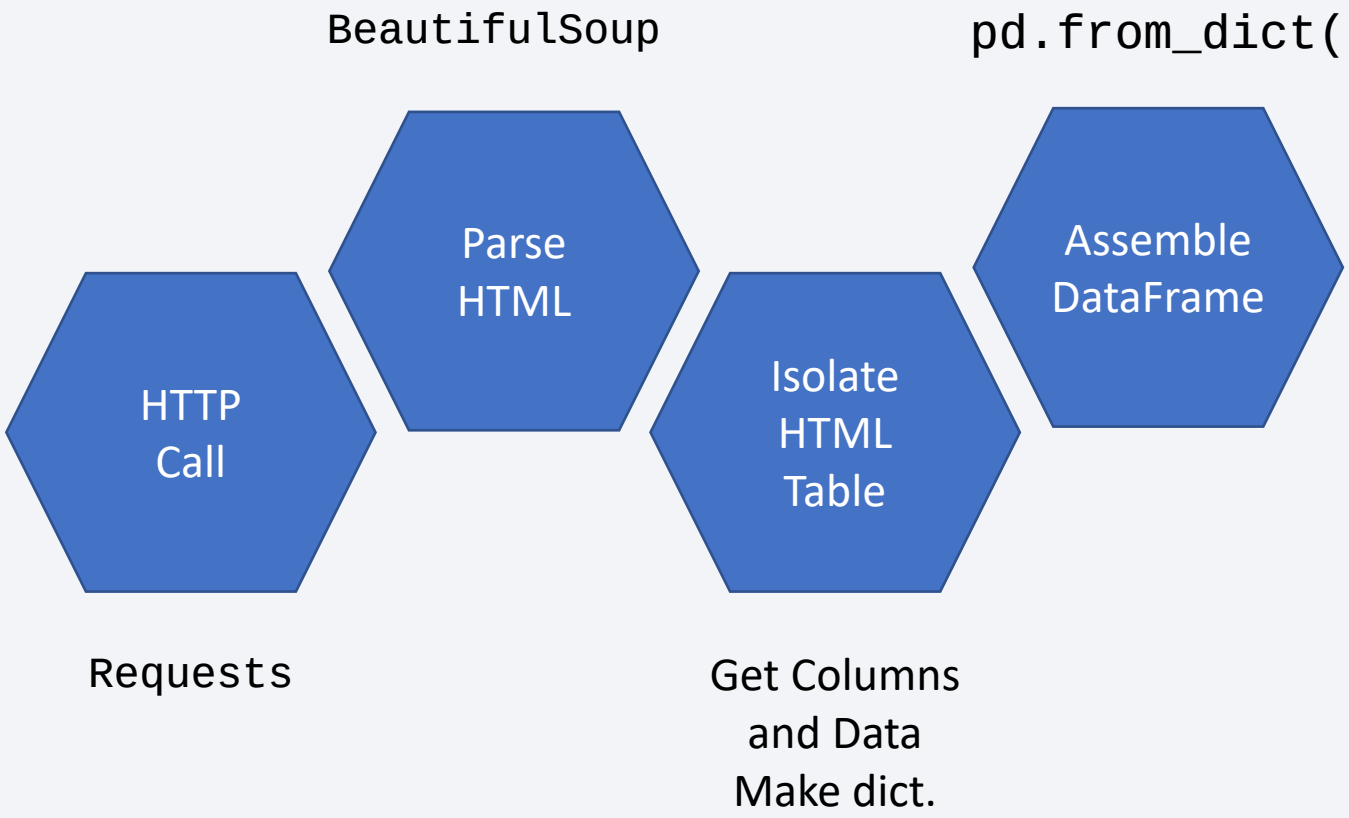
Executive Summary

- Data collection methodology:
  - SpaceX API, Webscraping Wikipedia
- Perform data wrangling
  - Null removal, Target/Feature Engineering, Encoding Categoricals, Standardization
- Perform exploratory data analysis (EDA) using visualization,Pandas and SQL
- Perform interactive visual analytics using Seaborn, Folium and Plotly Dash
- Perform predictive analysis using classification models
  - SVM, KNN, **Decision Tree**, Logistic Regression

Data Collection – SpaceX API



Data Collection - Scraping



Data Wrangling

1  NaN

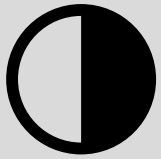
LandingPad column removed, 41 NaN. 5 missing Payload mass replaced with Mean(Payload mass).

2  Target/Features

Outcome is target variable, mapped to 0 and 1 in col. “Class”. Relevant feature data was selected.

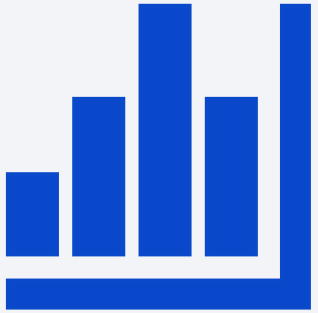
3  One Hot Encoding

Categorical Cols. Encoded to 0 or 1. Total encoded features = 83

4  Std. Test/Train Split

Z-score standardization, TSS at 80/20%

EDA with Data Visualization

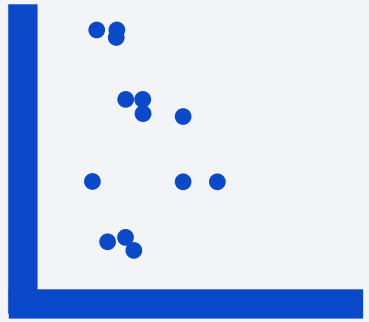


Barplot
Visually Compare values

Ex:
Comparing models

Time Series Plot
See a variable's trend over time

Ex:
Success Rate over time



Categorical Plot
Explore distribution of variable

Ex:
Mission Outcome per Payload

EDA with SQL

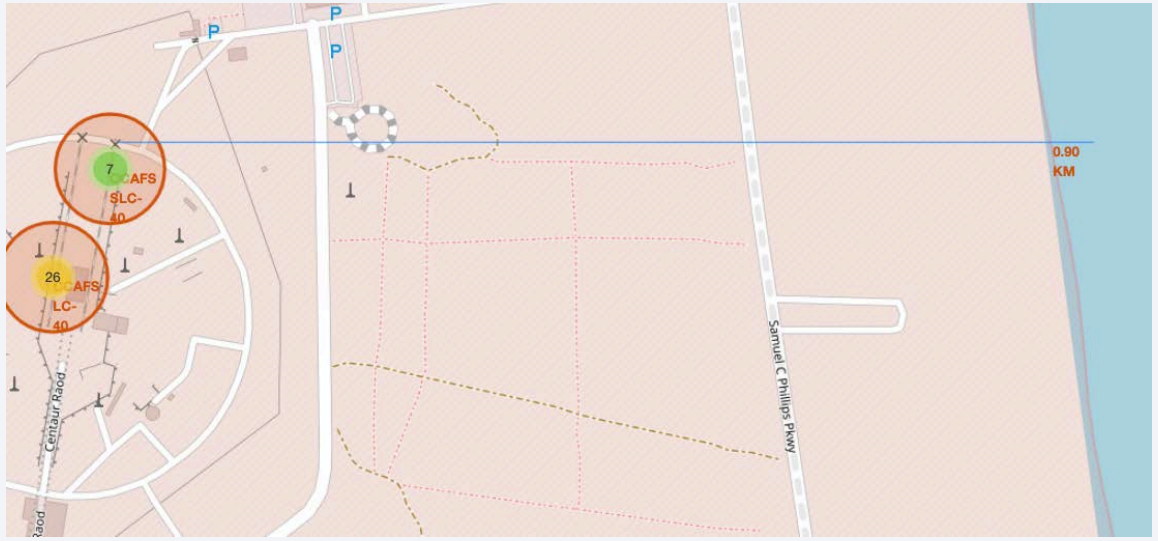
- SQL queries performed
  - Display the names of the unique launch sites in the space mission¶
  - Display 5 records where launch sites begin with the string 'CCA’
  - Total payload mass carried by boosters launched by NASA (CRS)
  - Average payload mass carried by booster version F9 v1.1
  - Date when the first successful landing outcome in ground pad was achieved
  - Names of the boosters success in drone ship and payload mass between 4000 and 6000

EDA with SQL

- SQL queries performed
  - Total number of successful and failure mission outcomes
  - Booster versions which have carried the maximum payload mass.
  - 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

Build an Interactive Map with Folium

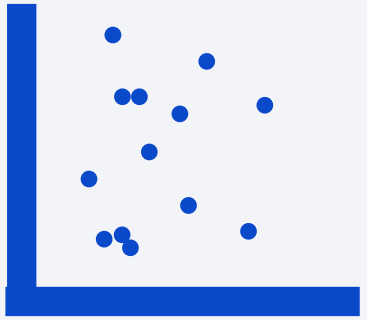
- Circles – Show the location of launch sites
- Marker clusters – Cluster outcomes in a location, such as outcome per site.
- Lines – Clearly show distances and directions of points of interest.
- Markers – Show the positions and distances of points of interest.



Build a Dashboard with Plotly Dash

Pie Chart
Compare relative values

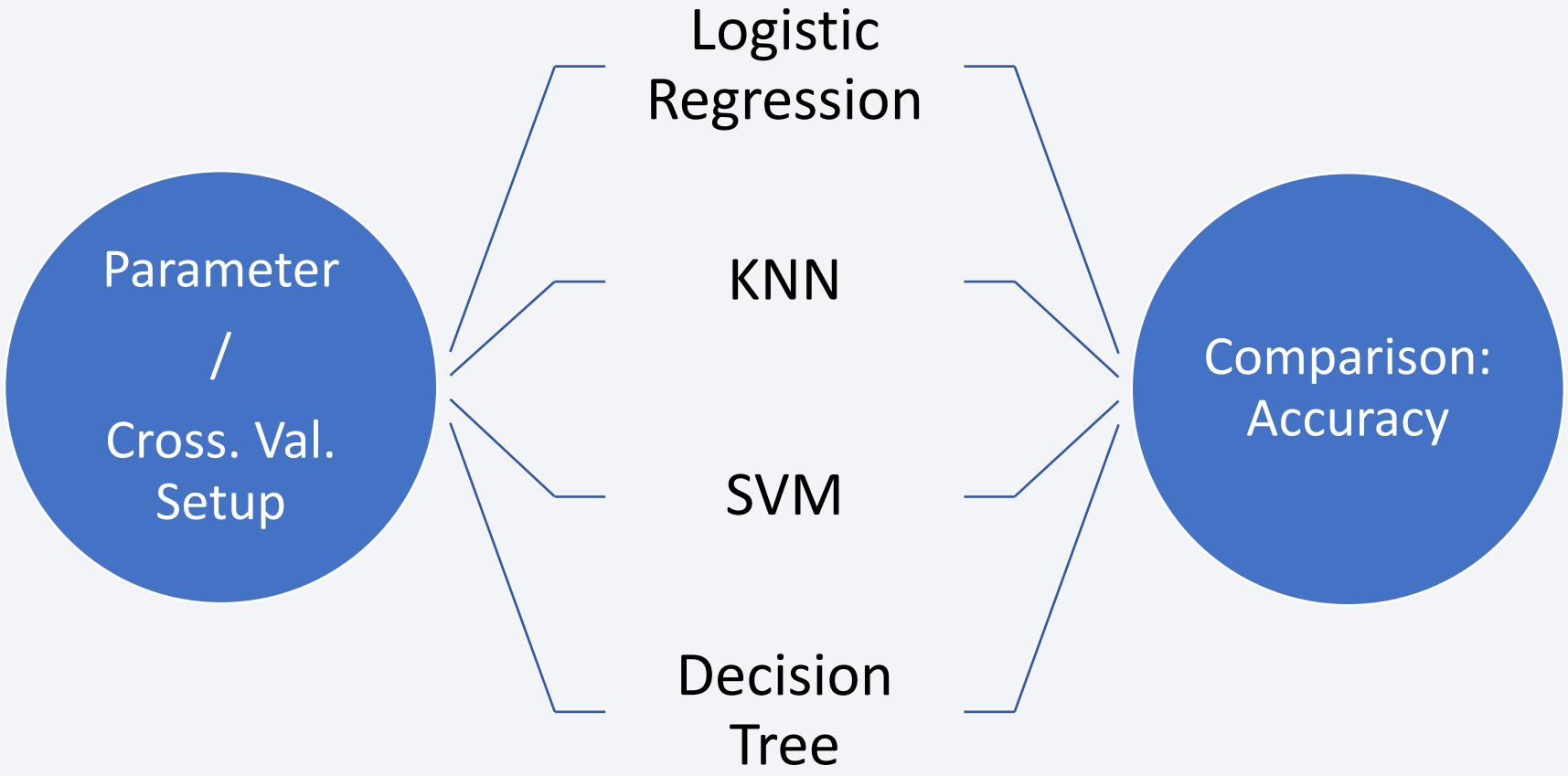
Ex:
Highest Success rates



Interactive Scatter Plot
Explore distribution of variable

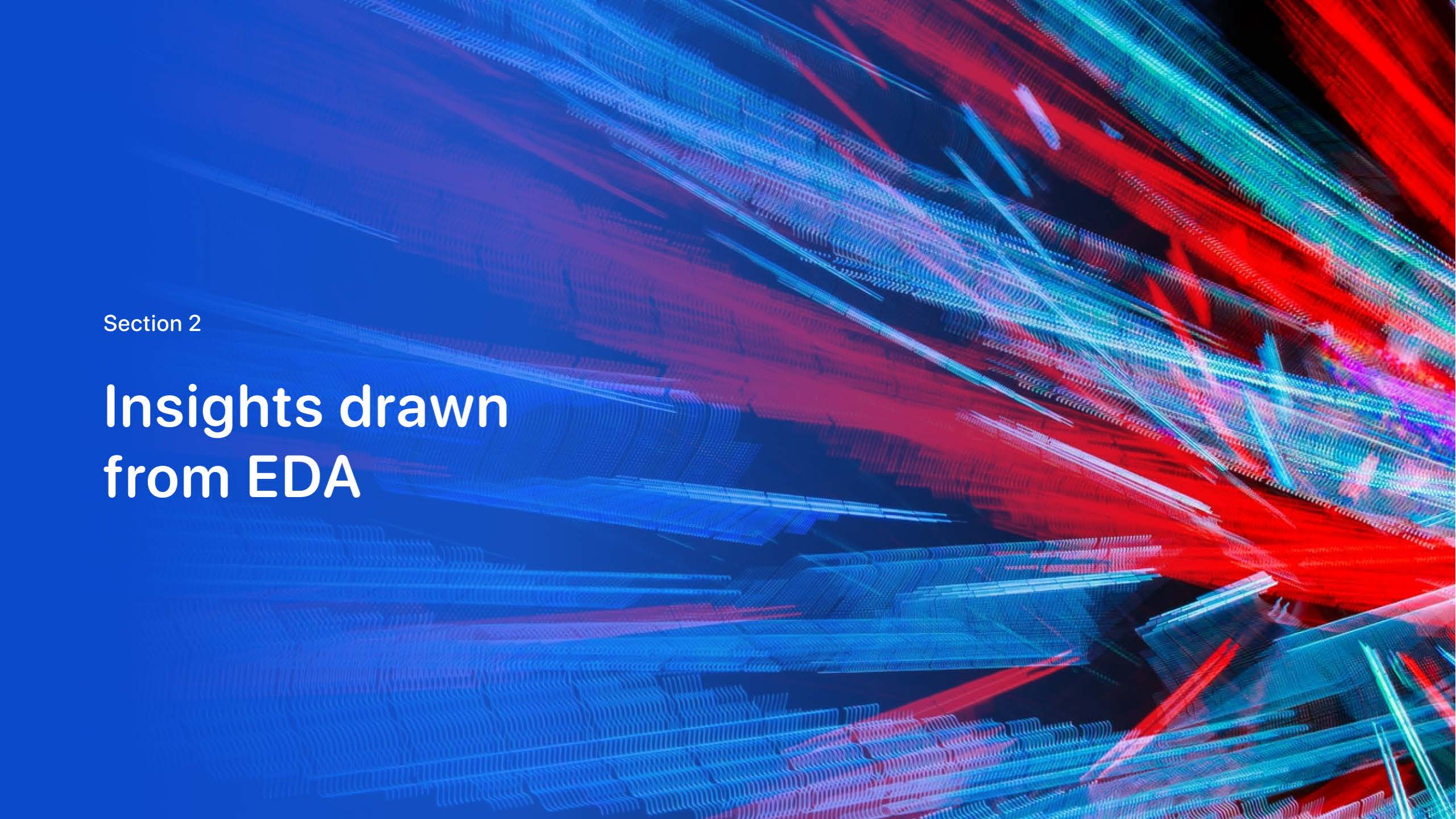
Ex:
Mission Outcome per Payload

Predictive Analysis (Models)



Predictive Analysis (Parameters)

Logistic Regression	SVM		KNN		Decision Tree
C = 0.1 - 1	C = 0.001 to 10000		N = 1-10		criterion: gini, entropy
L2 penalty	Kernels:		Algo=		splitter: best random
lbfgs solver	rbf	linear	ball_tree	auto	max_features: auto, sqrt
	poly	sigmoid	kd_tree	brute	max_depth: 2*n for n in range(1,10)
	Gamma: 0.001 to 10000		P = 1, 2		min_samples_leaf: 1, 2, 4 min_samples_split: 2, 5, 10



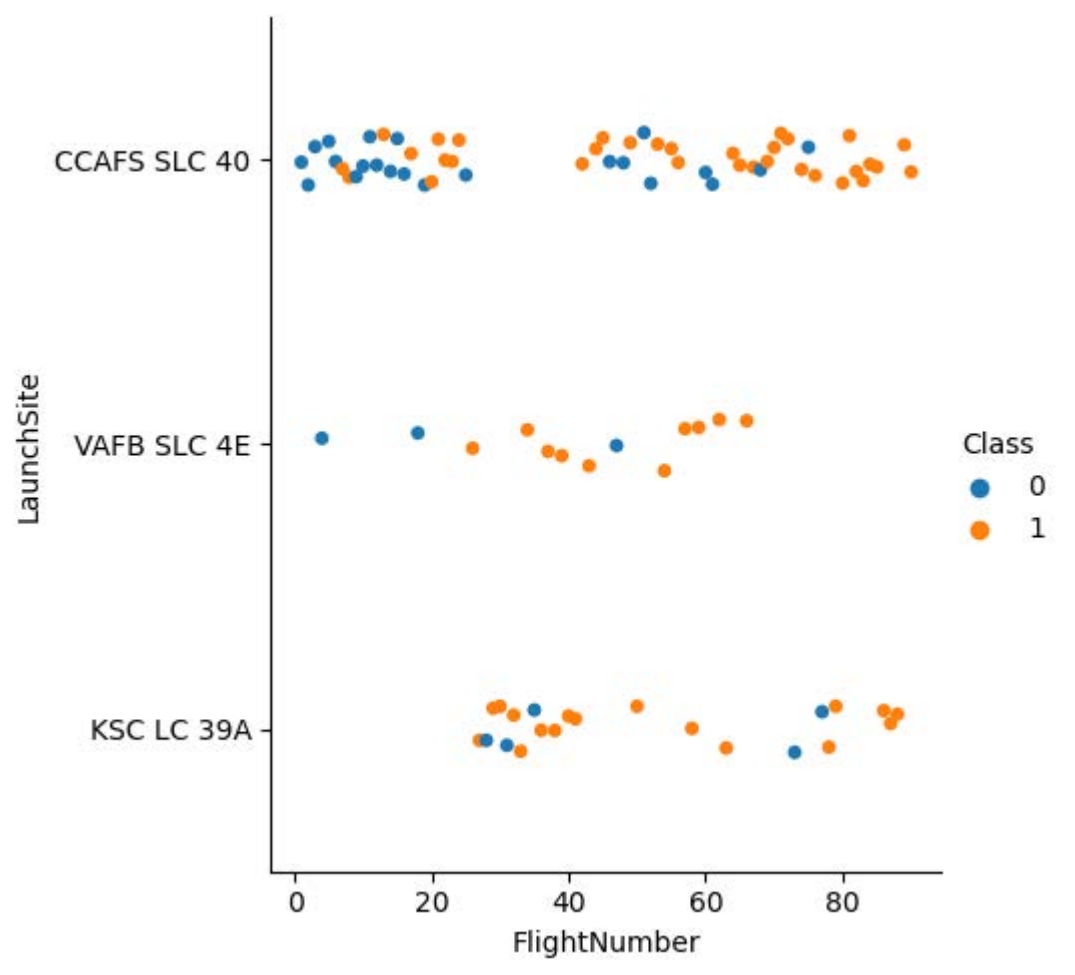
Section 2

Insights drawn from EDA

Results

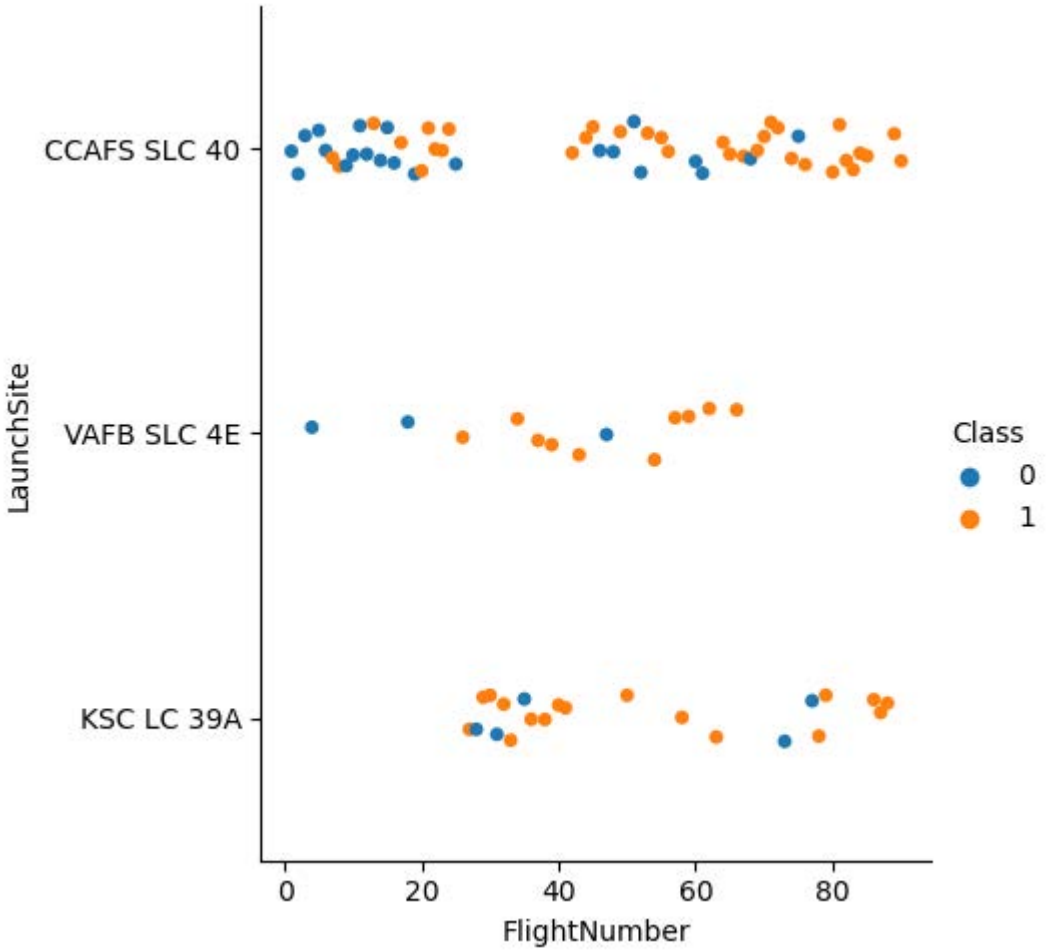
- Exploratory data analysis
- Interactive analytics demo in screenshots
- Predictive analysis

Flight Number vs. Launch Site



- Initial Launches @ CCAFS SLC 40, and were mostly **Unsuccessful (0)**
- Most Launches @ CCAFS SLC 40
- KSC LC 39A most **Successful (0)** launches
- No failed Launches after 80

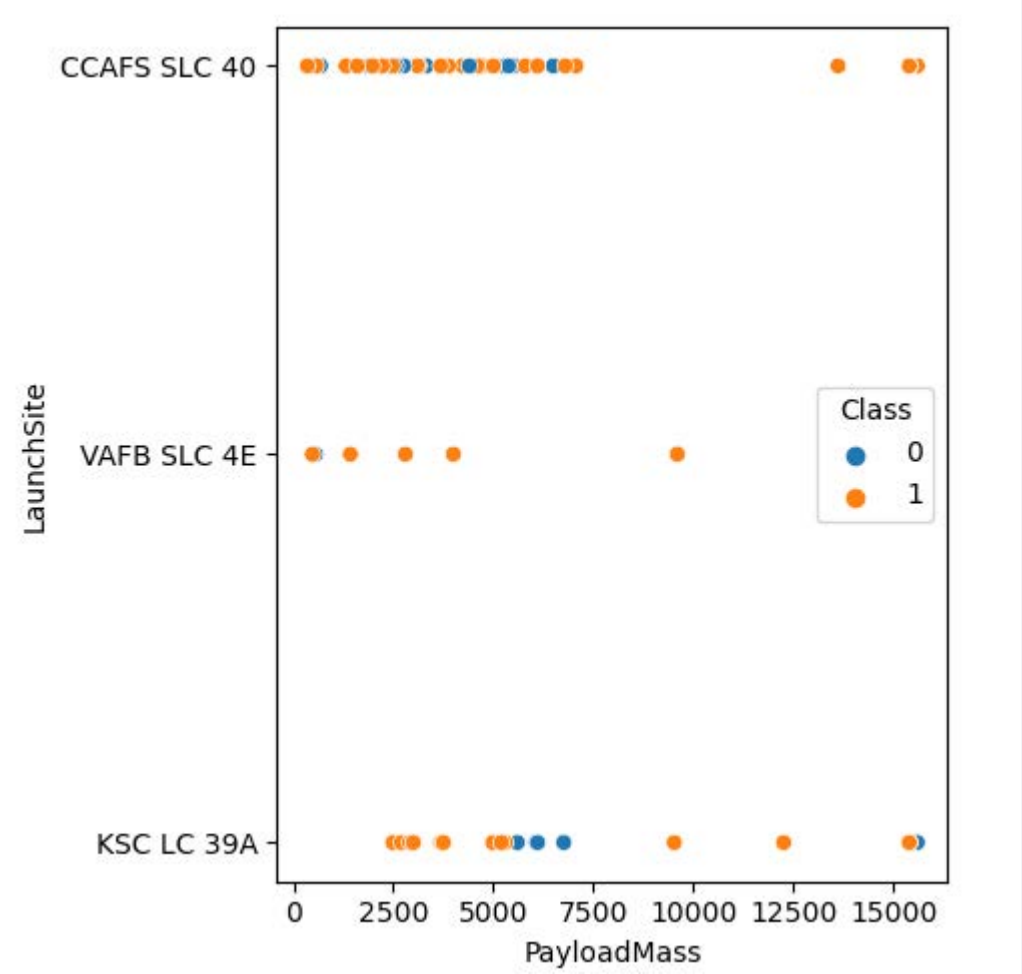
Flight Number vs. Launch Site



- Model based on CCAFS SLC – 40 may be biased.
  - Early launch failures are not indicative of future launches.
  - CCAFS SLC – 40 will predict more failures on average.

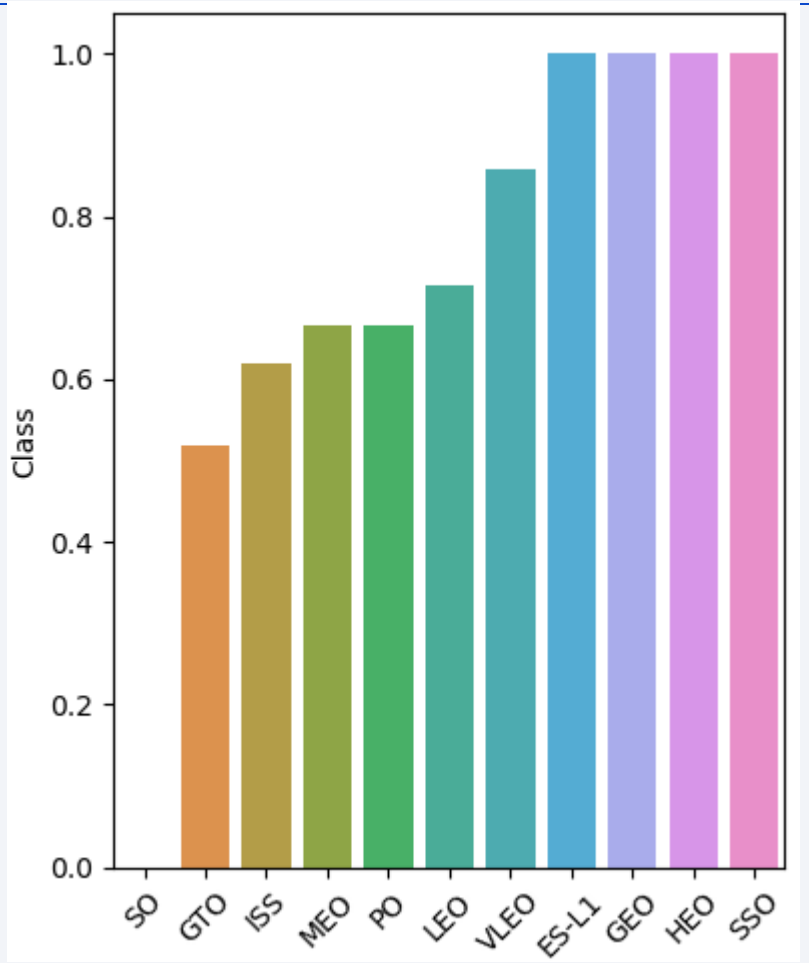
Payload vs. Launch Site

- Light Rockets launched from CCAFS SLC 40 and VAFB SLC 4E.
- Most failures with Payloads < 7500 Kg.



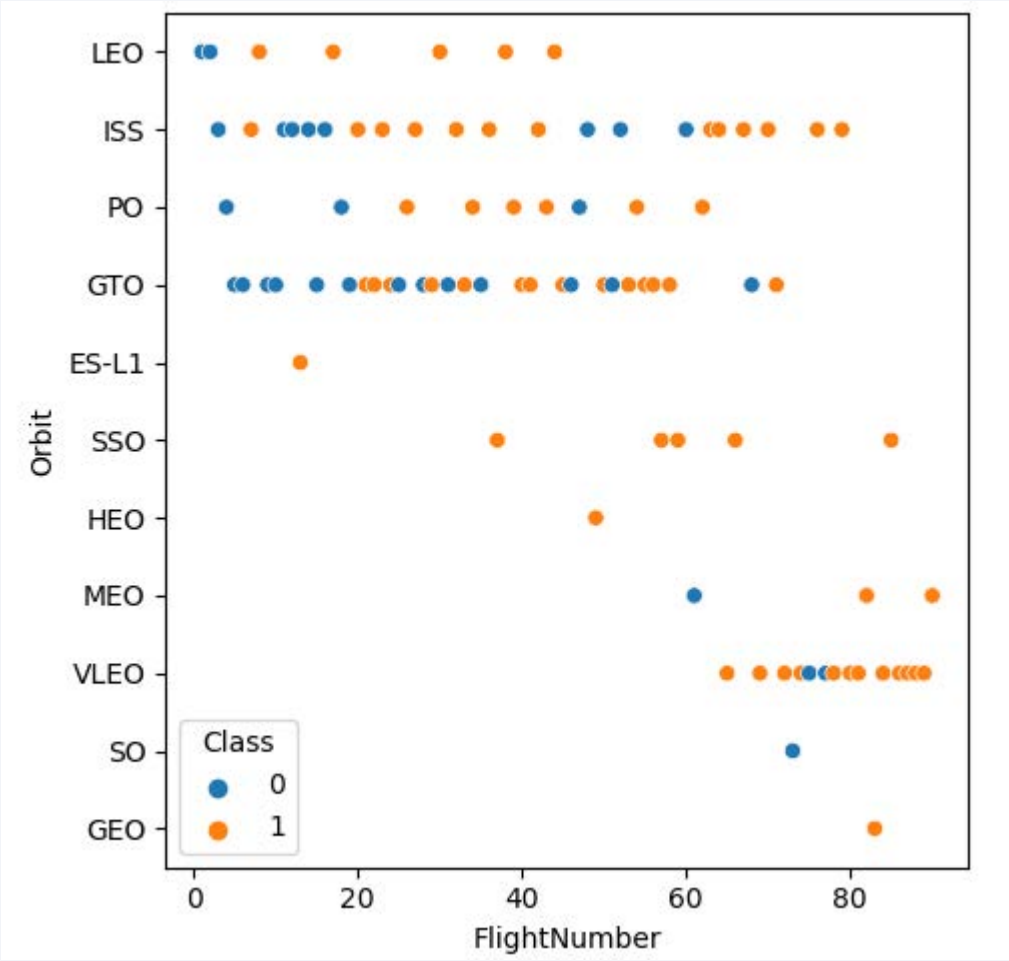
Success Rate vs. Orbit Type

- ES-L1, GEO, HEO, SSO 100% success rate. Some are difficult, so impressive.
- SO, GTO, and ISS worst.



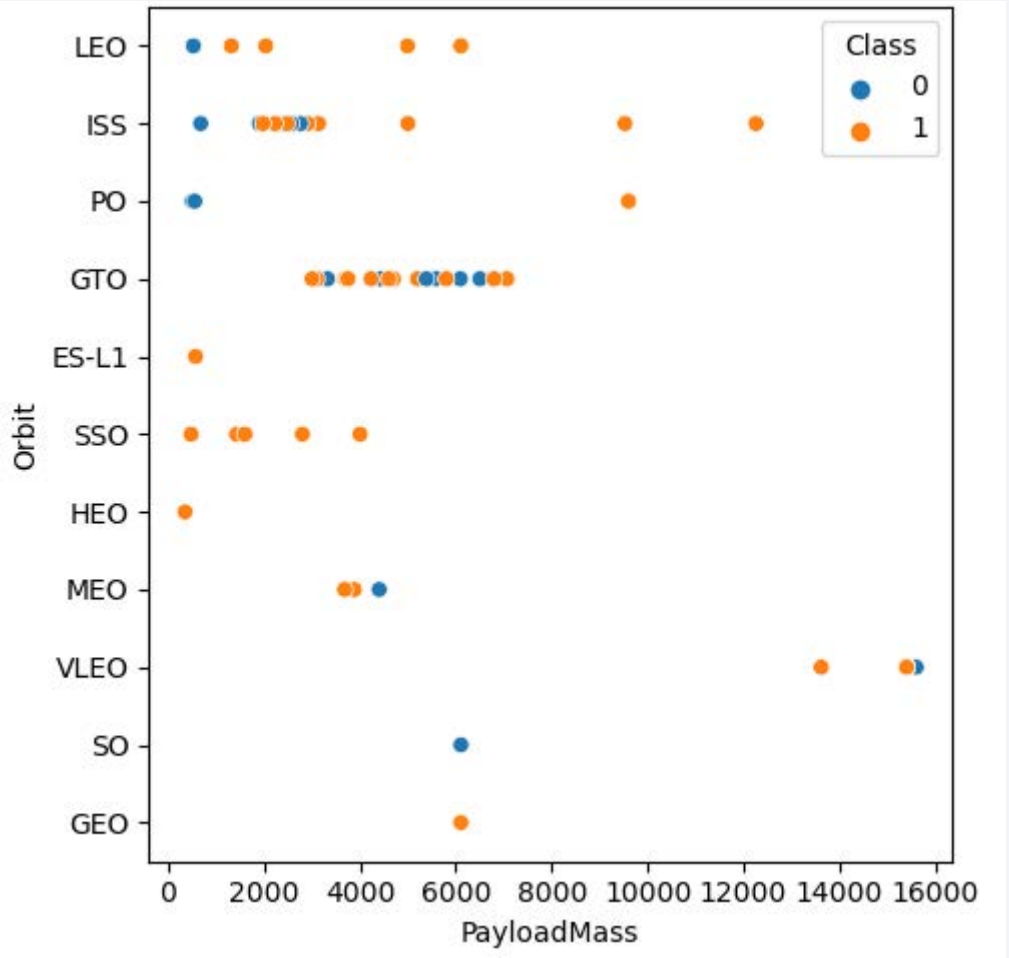
Flight Number vs. Orbit Type

- Most First launches were LEO, ISS, PO and GTO orbits which are mostly low/easy.
- Many VLEO which are Starlink launches



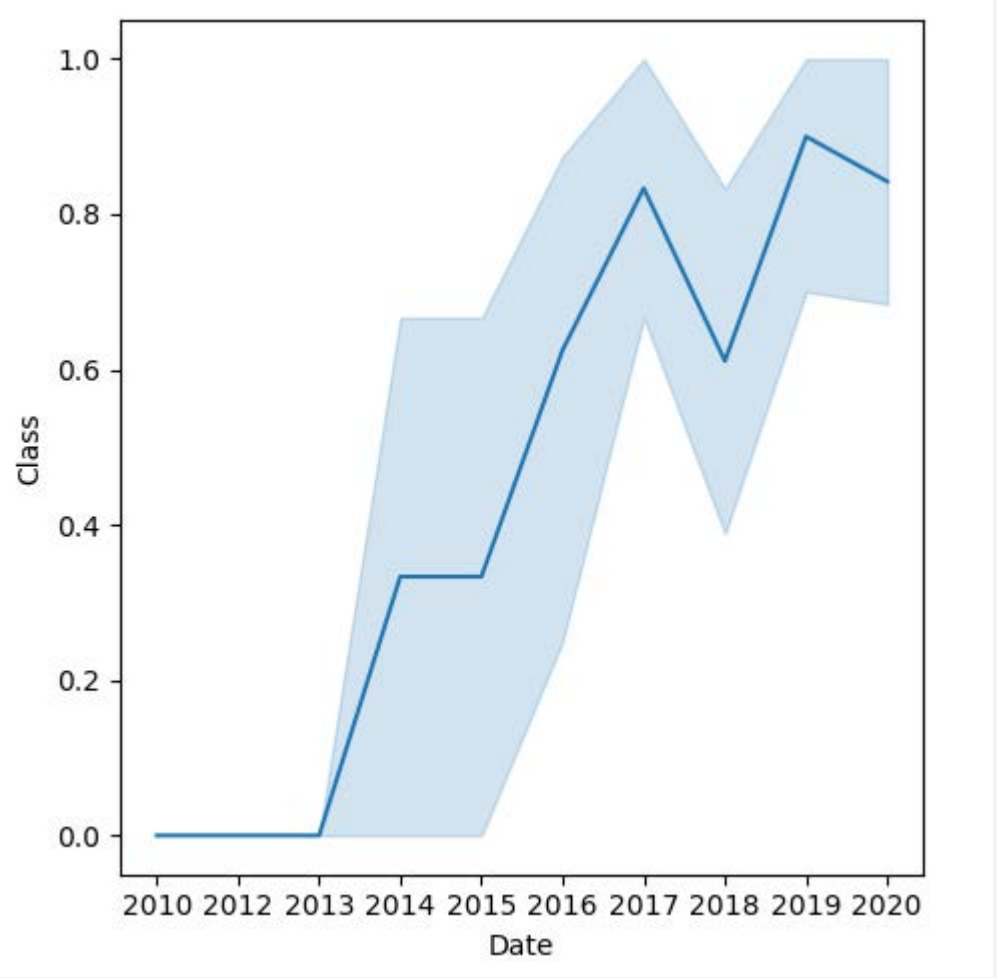
Payload vs. Orbit Type

- First launches were < 2000 kg, with many failures.
- 16 000 kg is the largest payloads so far, Starlink which has a payload of 15-17 tons



Launch Success Yearly Trend

- Low success at start
- 50% success rate in 2015, so year window.
- Current success rate > 80% after 10 years



All Launch Site Names

```
%sql SELECT UNIQUE(LAUNCH_SITE) FROM SPACEX_EDA;
 * ibm_db_sa://rrg16207:***@764264db-9824-4b7c-82c
ain.cloud:32536/bludb
Done.
```

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

SELECT all of the unique values in the LAUNCH_SITE column

Launch Site Names Begin with 'CCA'

```
%sql SELECT LAUNCH_SITE FROM SPACEX_EDA WHERE LAUNCH_SITE LIKE 'CCA%' LIMIT 5
```

```
 * ibm_db_sa://rrg16207:***@764264db-9824-4b7c-82df-40d1b13897c2.bs2io90l08kqb1o
ain.cloud:32536/bludb
Done.
```

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

SELECT the LAUNCH_SITE column where the LAUNCH_SITE value starts with 'CCA', and returns a maximum of 5 rows.

•This SQL statement calculates the total payload mass of all SpaceX launches where the customer was 'NASA (CRS)', from the SPACEX_EDA table.
•The SUM() function is used to add up all values in the PAYLOAD_MASS__KG_ column, and the result is renamed as TOTAL_PAYLOAD_MASS.
•The SUM() function is used to add up all values in the PAYLOAD_MASS__KG_ column, and the result is renamed as TOTAL_PAYLOAD_MASS.

Total Payload Mass

Task 3

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

```
%%sql SELECT SUM(PAYLOAD_MASS__KG_) AS TOTAL_PAYLOAD_MASS
    FROM SPACEX_EDA
    WHERE CUSTOMER = 'NASA (CRS)'
```

```
 * ibm_db_sa://rrg16207:***@764264db-9824-4b7c-82df-40d1b13897c2.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:32536/bludb
Done.
```

total_payload_mass
45596

Total payload mass of all SpaceX launches where the customer was 'NASA (CRS)'.
The SUM() function is used to add up all values in the PAYLOAD_MASS__KG_ column, and the result is renamed as TOTAL_PAYLOAD_MASS.

Average Payload Mass by F9 v1.1

```
%%sql SELECT SUM(PAYLOAD_MASS__KG_) AS TOTAL_PAYLOAD_MASS
    FROM SPACEX_EDA
    WHERE CUSTOMER = 'NASA (CRS)'
```

```
 * ibm_db_sa://rrg16207:***@764264db-9824-4b7c-82df-40d1b13
ain.cloud:32536/bludb
Done.
```

total_payload_mass
45596

This SQL statement calculates the average payload mass of all SpaceX launches where the BOOSTER_VERSION was 'F9 v1.1', from the SPACEX_EDA table, and renames the result as AVE_PAYLOAD_MASS.

First Successful Ground Landing Date

```
%%sql SELECT min(DATE) AS First_GP_Landing
    FROM SPACEX_EDA
    where LANDING__OUTCOME = 'Success (ground pad)'
```

```
 * ibm_db_sa://rrg16207:***@764264db-9824-4b7c-82df-40d1b13897c2.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:32536/bludb
Done.
```

first_gp_landing
2015-12-22

This SQL statement finds the earliest **DATE** of a SpaceX launch with a successful landing on a ground pad, and renames the result as **First_GP_Landing**.

Successful Drone Ship Landing with Payload between 4000 and 6000

```
%%sql SELECT BOOSTER_VERSION AS First_DS_Landing
    FROM SPACEX_EDA
    where LANDING__OUTCOME = 'Success (drone ship)' AND (PAYLOAD_MASS__KG_ between 4000 AND 6000)
```

```
 * ibm_db_sa://rrg16207:***@764264db-9824-4b7c-82df-40d1b13897c2.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:32536/bludb
Done.
```

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

This SQL statement selects the **BOOSTER_VERSION** of all launches where the landing outcome was a success on a drone ship, and the payload mass was between 4000 and 6000 kg and renames the result as First_DS_Landing.

Total Number of Successful and Failure Mission Outcomes

```
%sql SELECT MISSION_OUTCOME as Outcome, Count(MISSION_OUTCOME)as Total from SPACEX_EDA group b
```

```
 * ibm_db_sa://rrg16207:***@764264db-9824-4b7c-82df-40d1b13897c2.bs2io90l08kqb1od8lcg.database
s.appdomain.cloud:32536/bludb
Done.
```

outcome	total
Failure (in flight)	1
Success	99
Success (payload status unclear)	1

Counts the total number of missions for each MISSION_OUTCOME and returns the two columns ‘Outcome’ and ‘Total’. The COUNT() function is used to count the number of occurrences of each MISSION_OUTCOME and the GROUP BY clause is used to group the results by MISSION_OUTCOME.

Boosters Carried Maximum Payload

SELECT all of the unique BOOSTER_VERSION with a maximum payload mass, and renames the result as Heaviest_Payload_Boosters. The subquery is used to find the maximum payload mass and the outer query is used to select the BOOSTER_VERSION associated with that maximum payload mass.

```
%%sql SELECT DISTINCT(BOOSTER_VERSION) AS Heaviest_Payload_Boosters
    FROM SPACEX_EDA
    where PAYLOAD_MASS__KG_ = (Select MAX(PAYLOAD_MASS__KG_) from SPACEX_EDA)
```

```
 * ibm_db_sa://rrg16207:***@764264db-9824-4b7c-82df-40d1b13897c2.bs2io90l08kqb1
s.appdomain.cloud:32536/bludb
Done.
```

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

2015 Launch Records

```
%%sql SELECT BOOSTER_VERSION, LAUNCH_SITE FROM SPACEX_EDA
    WHERE (LANDING__OUTCOME = 'Failure (drone ship)') AND (EXTRACT(YEAR FROM DATE) = '2015');
```

```
 * ibm_db_sa://rrg16207:***@764264db-9824-4b7c-82df-40d1b13897c2.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:32536/bludb
Done.
```

booster_version	launch_site
F9 v1.1 B1012	CCAFS LC-40
F9 v1.1 B1015	CCAFS LC-40

SELECT the BOOSTER_VERSION and LAUNCH_SITE of all failed drone ship landing in 2015. The EXTRACT() function is used to extract the year from the DATE column.

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

```
%%sql SELECT LANDING__OUTCOME as Outcome, Count(LANDING__OUTCOME) as Total from SPACEX_EDA
WHERE DATE BETWEEN '2010-06-04' AND '2017-03-20'
group by (LANDING__OUTCOME)
order by total DESC
```

Counts the total number of successful and unsuccessful landing outcomes for launches that occurred between the dates and returns the results: Outcome and Total. The COUNT() function is used to count the number of occurrences of each LANDING__OUTCOME and the GROUP BY clause is used to group the results by LANDING__OUTCOME. Finally, the ORDER BY clause is used to sort the results in descending order of total count.

outcome	total
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

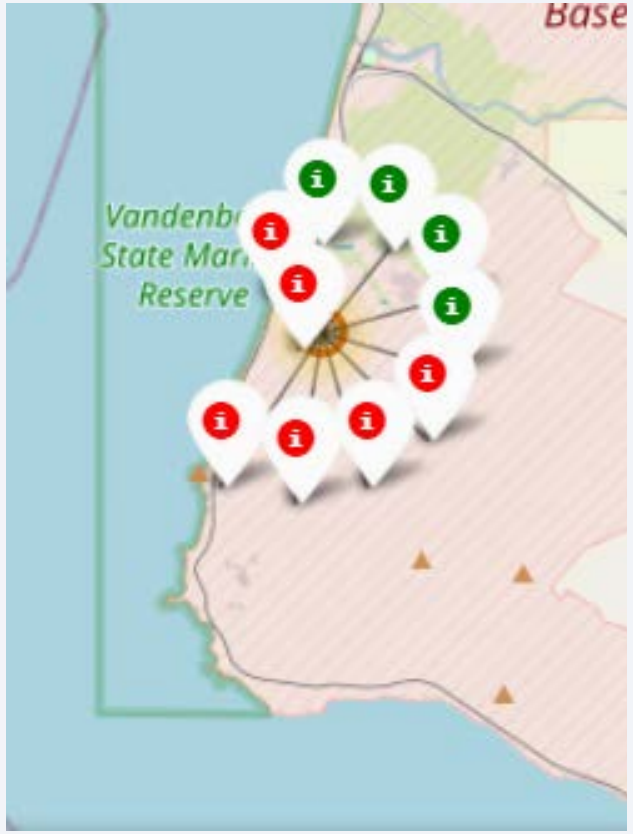

Section 3
Launch Sites
Proximities Analysis

<Folium Map Screenshot 1>

- KSC is on the west coast of the USA, and the rest on the east coast of Florida
- All sites significantly above equator, so all launches are inclined.

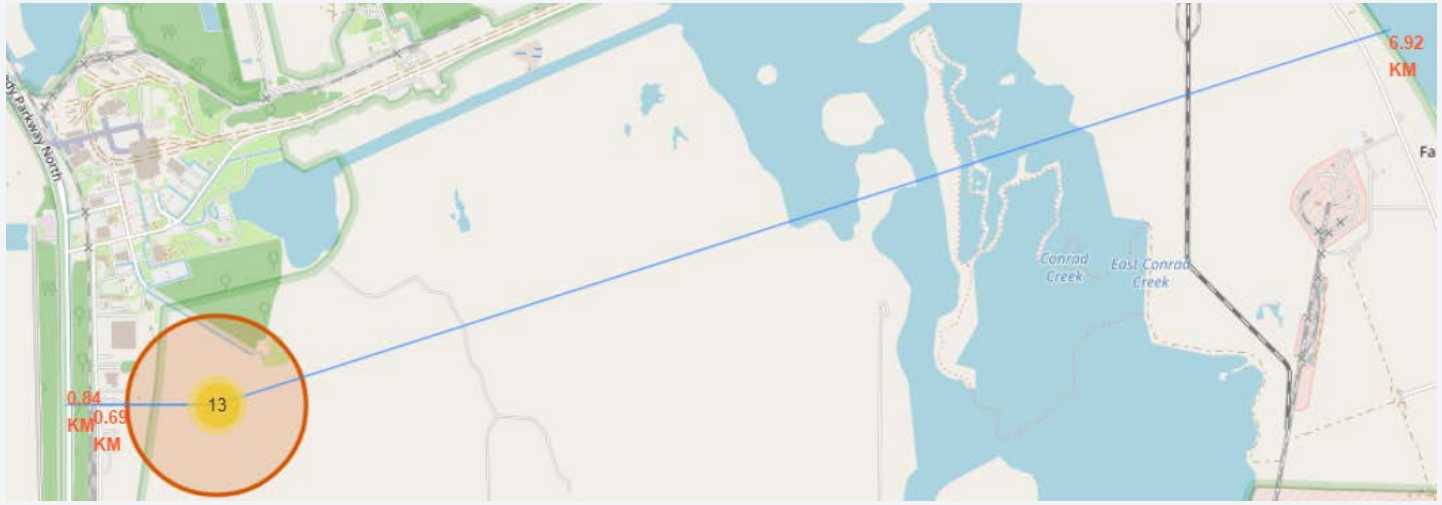
<Folium Map Screenshot 2>

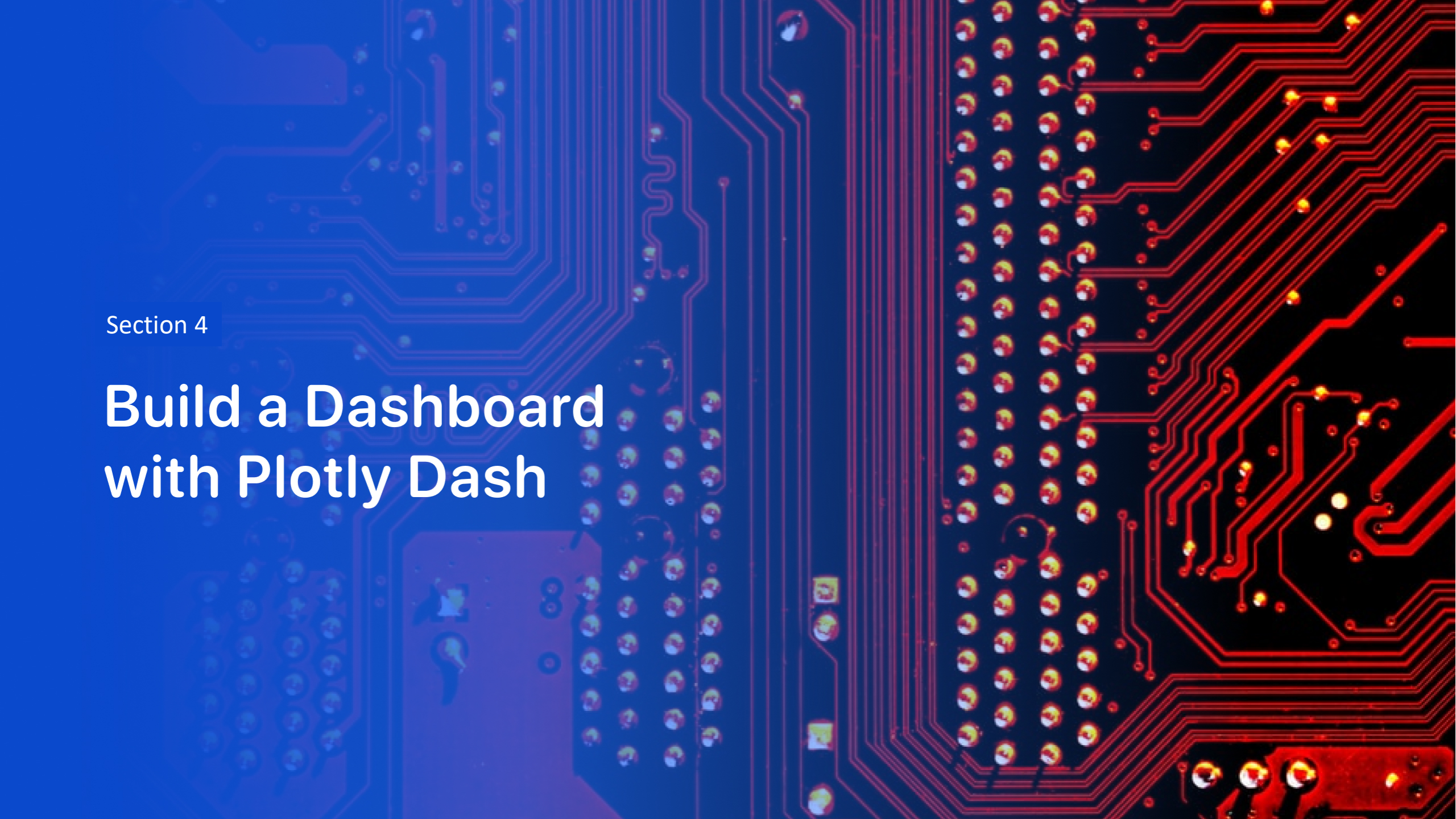
- The Vandanburg Site has some failed and successful launches



<Folium Map Screenshot 3>

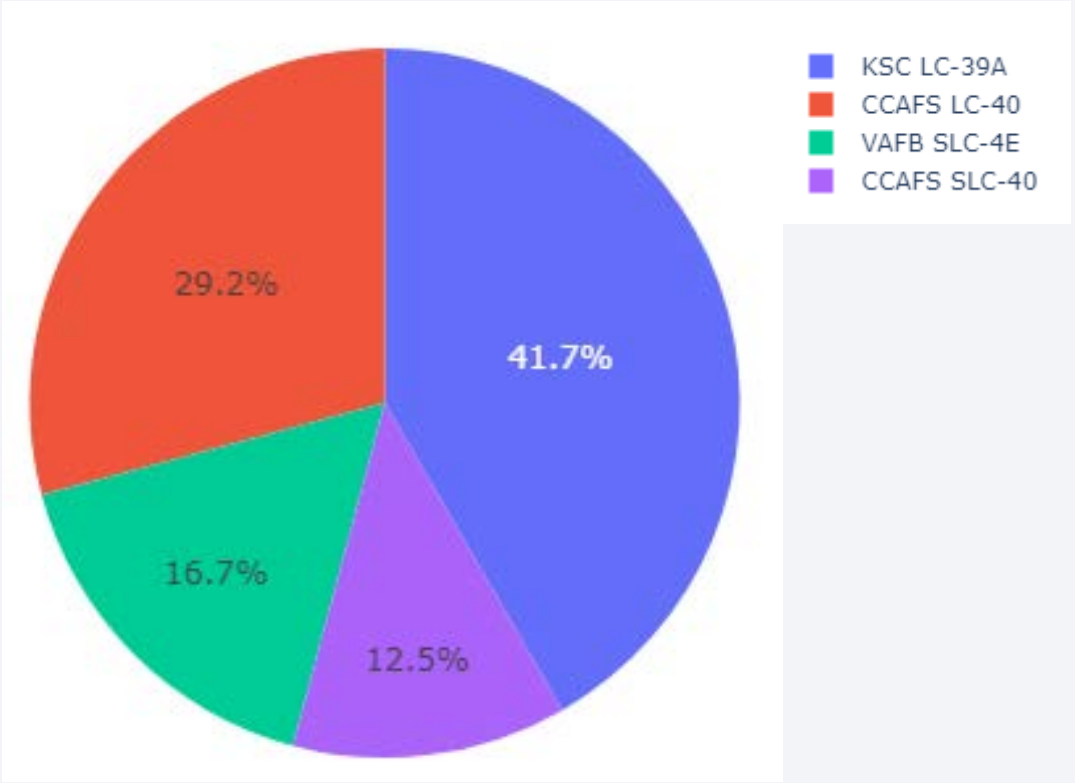
- The launch sites normally have infrastructure close to the site.
  - Rail is used to deliver parts, fuel, as shown – 0.65 km away
  - Road is mostly used for staff and vehicle/payload delivery 0.84 km away
  - Coasts are a good place to have crashes – 6.92 km away




Section 4
Build a Dashboard with Plotly Dash

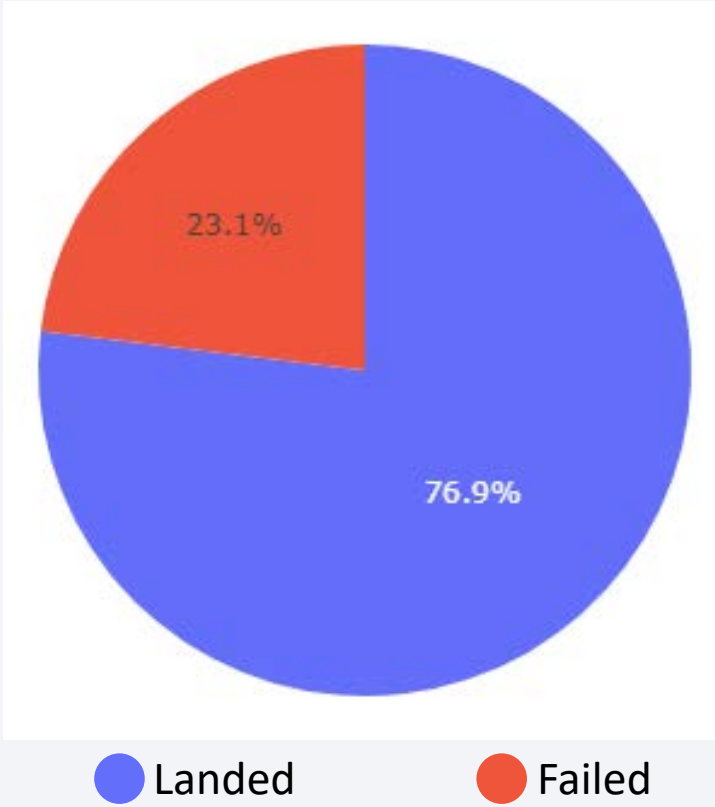
Total Successful Launches per Site

- Highest success from KSC LC-39A, followed by CCAFS LC-40, with CCAFS SLC-40 the lowest success rate



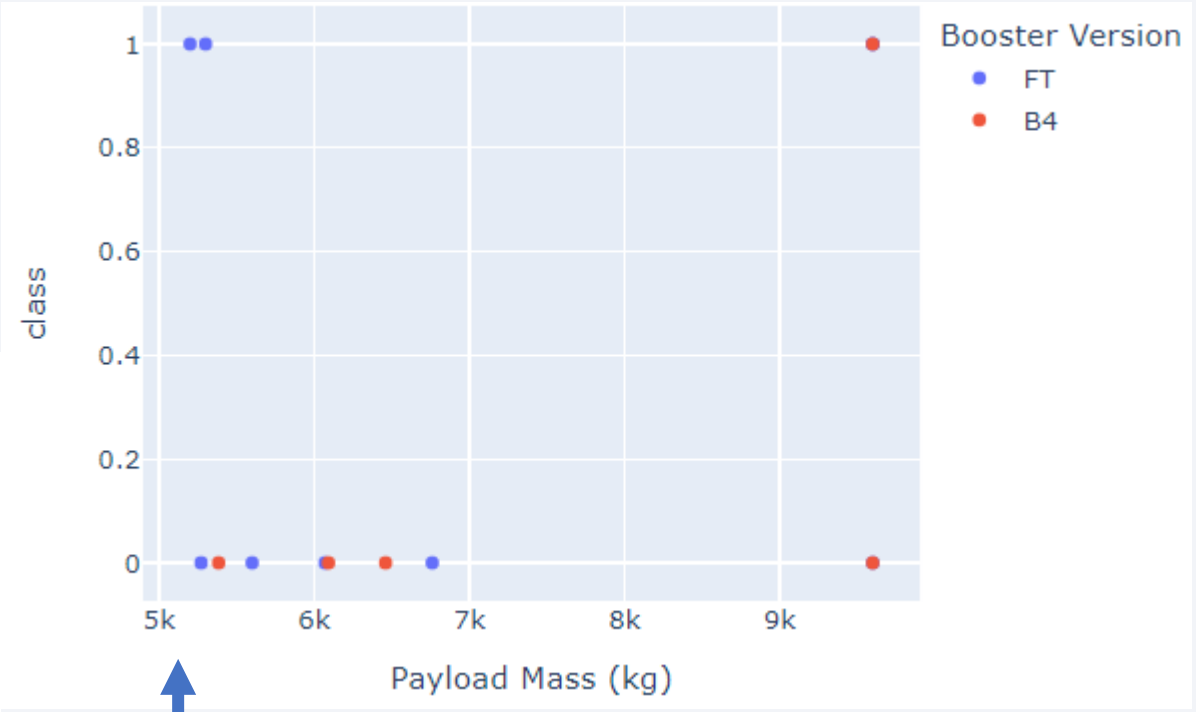
Closer look at KSC LC-39A…

- KSC LC-39A has a 77% success rate.



Payload vs. Success Rate

- FT version most failures
- V1.1 most successful



Most success when Mass > 5000 kg

Least success when Mass < 5000 kg

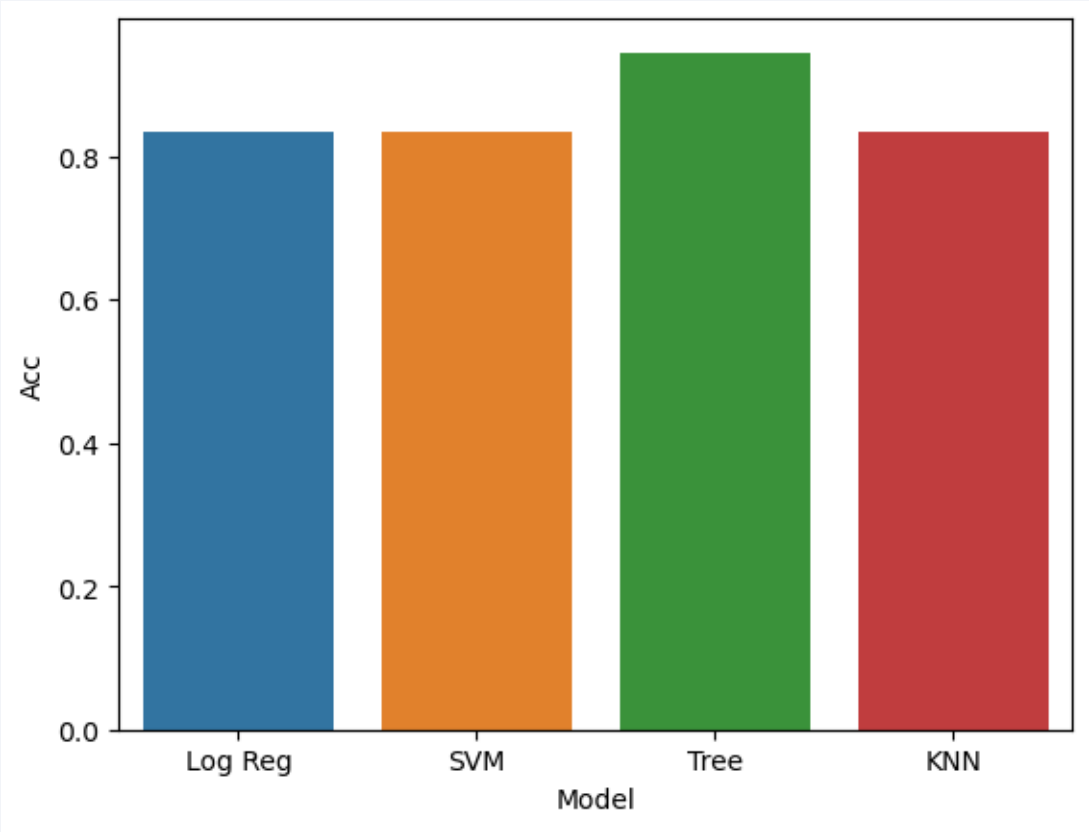
Section 5

Predictive Analysis (Classification)

Classification Accuracy

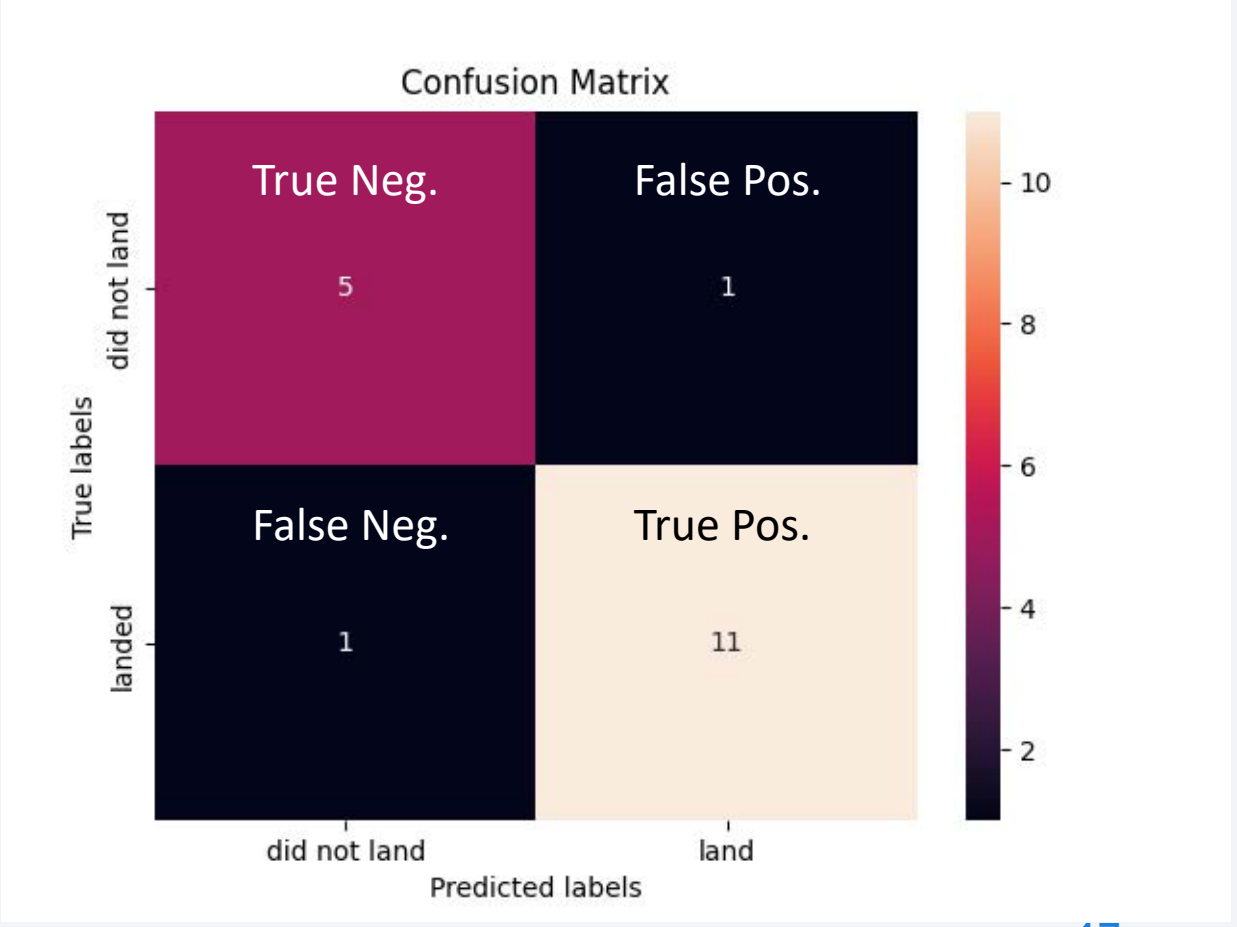
- Visualize the built model accuracy for all built classification models, in a bar chart

- Find which model has the highest classification accuracy

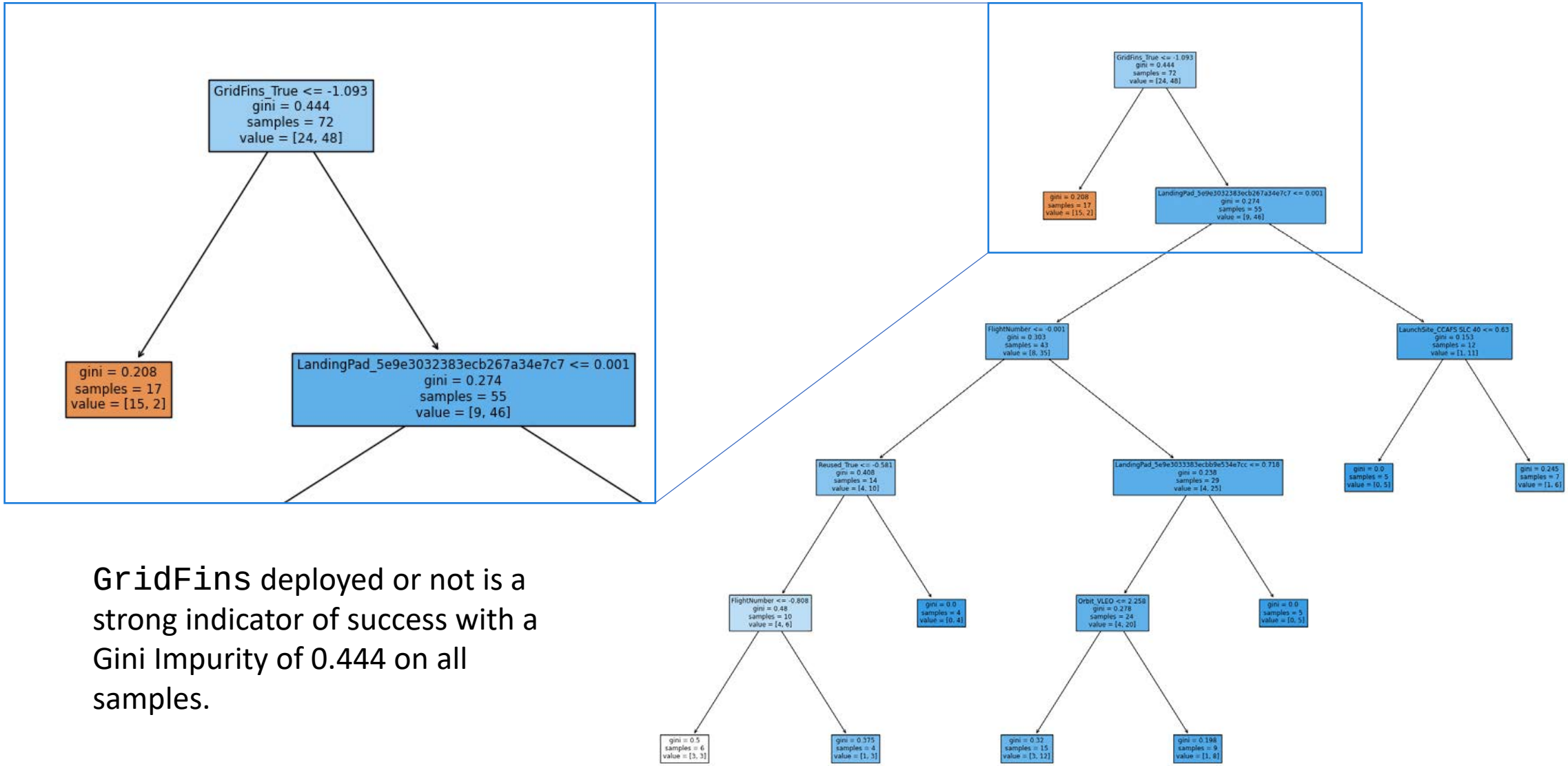


Confusion Matrix

- False Negative result in significant loss of asset, so should try to improve.
  - Larger data set may fix
- False Positives will require some further inspections, so not that serious.

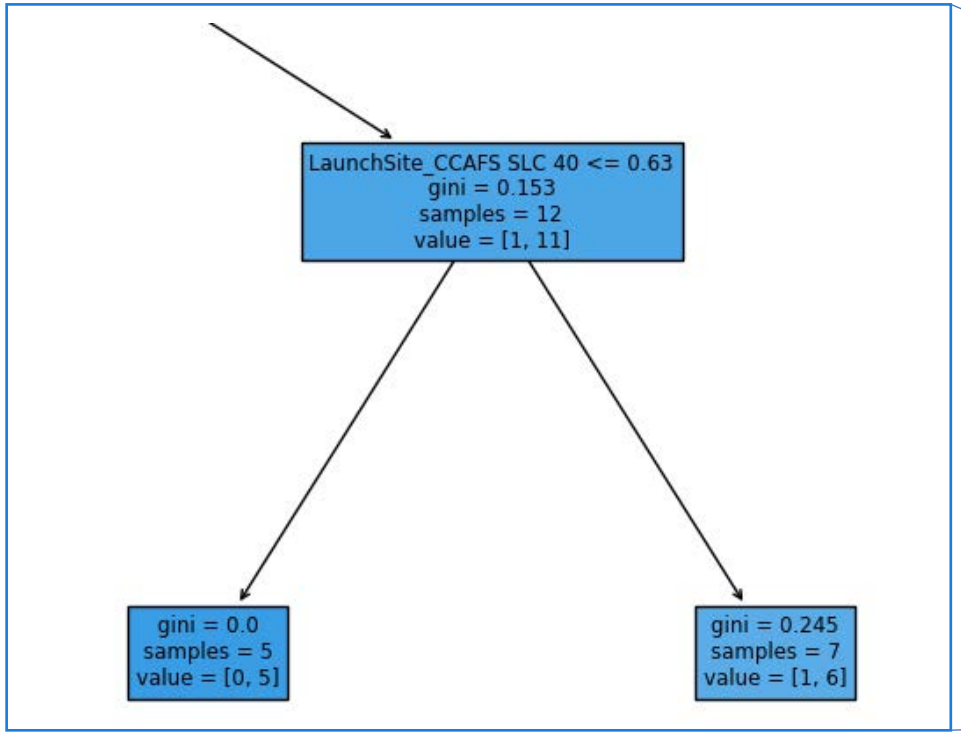


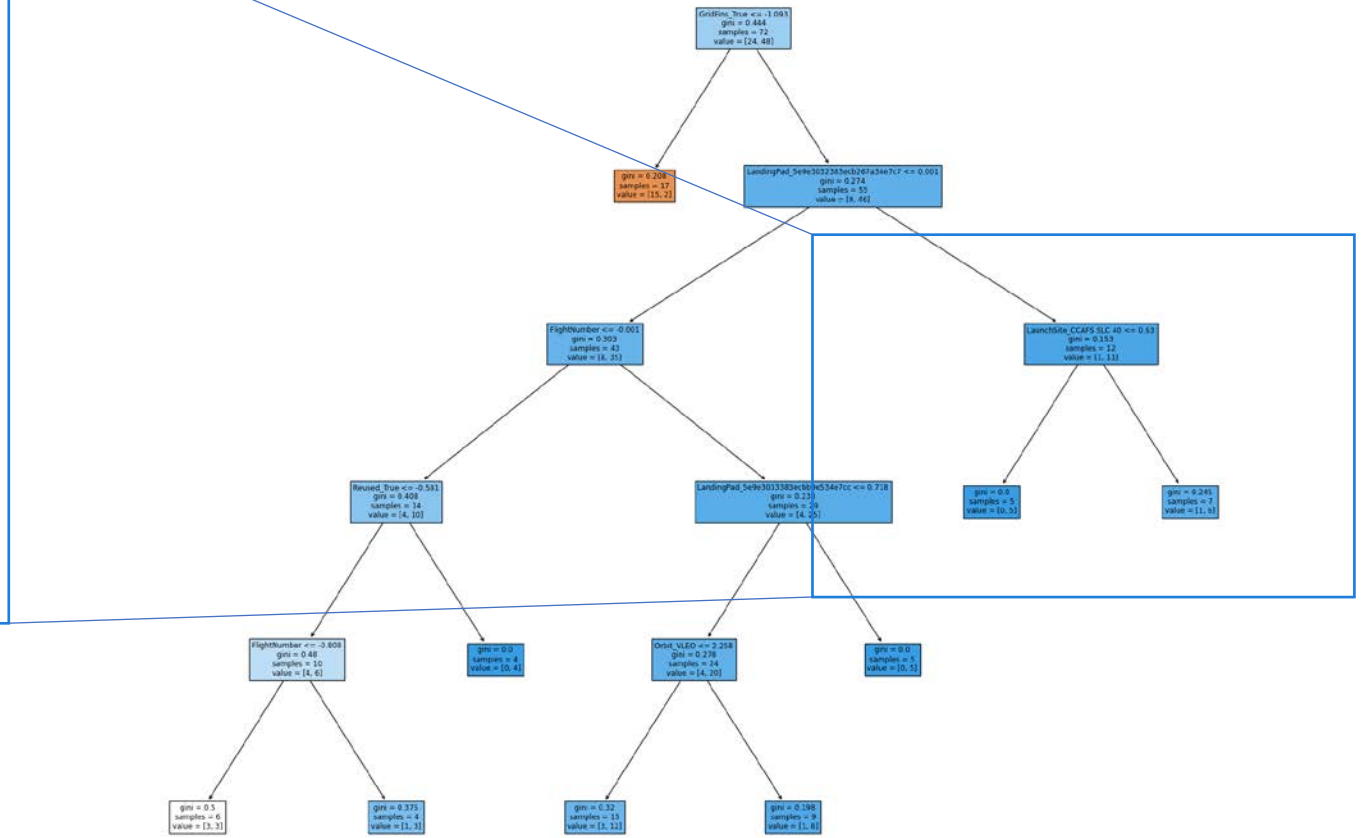
Deeper look into the Tree…



`GridFins` deployed or not is a strong indicator of success with a Gini Impurity of 0.444 on all samples.

Deeper look into the Tree…





The CCAFS SLC – 40 is high on the Decision Tree, but has a relatively low impurity. While the bias is there, its effect is limited.

A note on Trees…

- Decision Trees are dependent on the first branches chosen.
    - Many model fits resulted in Out of Sample Accuracies of 0.83, 0.88, 0.77. Perhaps a random forest may be more applicable.
    - Different Environments such as Jupyterlab(local), Skills Network Labs, and Watson studio resulted in different results. This is likely due to the version of sklearn. Version 1.0 brought some changes to Trees, which may lead to different results.

Model Improvement

	FlightNumber	Date	BoosterVersion	PayloadMass	Orbit	LaunchSite	Outcome	Flights	GridFins	Reused	Legs	LandingPad	Block	ReusedCount	Seria
49	50	2018-05-11	Falcon 9	3750.000000	GTO	KSC LC 39A	True ASDS	1	True	False	True	5e9e3032383ecb6bb234e7ca	5.0	3	B104
54	55	2018-08-07	Falcon 9	5800.000000	GTO	CCAFS SLC 40	True ASDS	2	True	True	True	5e9e3032383ecb6bb234e7ca	5.0	3	B104
58	59	2018-12-03	Falcon 9	4000.000000	SSO	VAFB SLC 4E	True ASDS	3	True	True	True	5e9e3033383ecbb9e534e7cc	5.0	3	B104
72	73	2020-01-19	Falcon 9	6104.959412	SO	KSC LC 39A	None None	4	False	True	False	NaN	5.0	3	B104

- From this view, Flight 73 is an example of a failed flight.
  - The view shows that booster B1046 launched four times, and the last time, exploded at T+00:01:36, and was unable to deploy its fins, so `GridFins` = False. Video.
- `GridFins` is a dependent variable. A flight that doesn't get an opportunity deploy the Grid fins before failing is reported as False, but it should be None.

Conclusions

High success:

Launch from KSC LC 39

Take a large payload

Launch into ES-L1, GEO, HEO, SSO

Best Model:

Decision Tree

- Gave insight into most important variables

Model can be improved:

Mine failure causes (unstructured) from API data and categorize

Mine web searches of launch on a specific date to get more failure data.

Test/Remove Landing site variables

Reclassify GridFins to True(deployed), False(not deployed) and None (when not tried to deploy, but failed mission)

Appendix

- Github Project Page:
  - https://github.com/Spookperd/IBMCertificate/tree/main/Capstone%20Project
- Flight 73
  - https://www.youtube.com/watch?v=ARIZnaMXTEU&ab_channel=NASA

Thank you!