SpaceX Falcon 9 First Stage Landing Prediction

Dr. Mohsen Mesgar

13.02.2022

IBM Developer

SKILLS NETWORK

OUTLINE

- Executive Summary
- Introduction
- Methodology
- Results
  - Visualization – Charts
  - Dashboard
- Discussion
  - Findings & Implications
- Conclusion
- Appendix

EXECUTIVE SUMMARY (Abstract)

- The task is to predict if Falcon 9 first stage lands successfully.
- What we do
  - Collect data from the wiki page of SpacX
  - Conduct EDA on the data to predict some trend
  - Train machine learning models (support vector machines (SVM), LogisticRegression, and Decsision Tree).
- Findings
  - The sucess rate since 2013 kept increasing till 2020
  - Different orbits have different success rates.
  - DecisionTree achieves 90% accuracy and outperforms the other examined ML model.

INTRODUCTION

- We aim at predicting if the Falcon 9 first stage lands successfully.
- SpaceX advertises Falcon 9 rocket launches on its website with a cost of 62 million dollars; other providers cost upward of 165 million dollars each, much of the savings is because SpaceX can reuse the first stage. Therefore if we can determine if the first stage will land, we can determine the cost of a launch. This information can be used if an alternate company wants to bid against SpaceX for a rocket launch.

METHODOLOGY

- Gathering the data including a class label to say if a landing is successful or not.
  - We do webscraping methods to collect such data from Wikipedia.
  - We deal with missing values of an attribute by replacing its NaN values with the mean of its values for other examples.
- We explore the data using some Exploratory Data Analysis (EDA) methods to find some patterns in the data.
  - We study these patterns using data visualization methods
- We conduct a feature engineering to understand what attribute in the data is distinctive between successful and unsuccessful landings.
- We build different types of machine learning methods such as (decision trees, logistic regression, and SVM) to predict if a landing is successful or not.

Data collection-SpaceX API

- Request and parse the SpaceX launch data using the GET request *(source: SpaceX API)*
- Filter the Dataframe to only include Falcon 9 launches
- Dealing with Missing Values
- URL: https://github.com/MMesgar/tutorial-data-science/blob/main/capstone/1.jupyter-labs-spacex-data-collection-api.ipynb

Data collection-Web scraping

- Request the Falcon9 Launch Wiki page from its URL using *Python's BeautifulSoup library*

- Extract all column/variable names from the HTML table header

- Create a data frame by parsing the launch HTML tables

- URL: https://github.com/MMesgar/tutorial-data-science/blob/main/capstone/2.jupyter-labs-webscraping.ipynb

Data wrangling methodology

- Calculate the number of launches on each site
- Calculate the number and occurrence of each orbit
- Calculate the number and occurence of mission outcome per orbit type
- Create a landing outcome label from Outcome column

- URL: https://github.com/MMesgar/tutorial-data-science/blob/main/capstone/1.jupyter-labs-spacex-data-collection-api.ipynb

EDA with data visualization

- Visualize the relationship between Flight Number and Launch Site
- Visualize the relationship between Payload and Launch Site
- Visualize the relationship between success rate of each orbit type
- Visualize the relationship between FlightNumber and Orbit type
- Visualize the relationship between Payload and Orbit type
- Visualize the launch success yearly trend
- Create dummy variables to categorical columns
- Cast all numeric columns to float64

- URL: https://github.com/MMesgar/tutorial-data-science/blob/main/capstone/5.jupyter-labs-eda-dataviz.ipynb

EDA with SQL

- Display the names of the unique launch sites in the space mission
- Display 5 records where launch sites begin with the string 'CCA'
- Display the 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 acheived.
- 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
- URL: https://github.com/MMesgar/tutorial-data-science/blob/main/capstone/4.jupyter-labs-eda-sql-coursera.ipynb

Interactive visual analytics

- Mark all launch sites on a map
- Mark the success/failed launches for each site on the map
- Calculate the distances between a launch site to its proximities

- URL: https://github.com/MMesgar/tutorial-data-science/blob/main/capstone/5.jupyter-labs-eda-dataviz.ipynb

Predictive analysis (classification)

- Standardize the data
- Use the function train_test_split to split the data X and Y into training and test data. Set the parameter test_size to 0.2 and random_state to 2.
- Create a logistic regression object then create a GridSearchCV object
- Create a support vector machine object then create a GridSearchCV object svm_cv with cv – 10
- Create a decision tree classifier object then create a GridSearchCV object tree_cv with cv = 10
- Create a k nearest neighbors object then create a GridSearchCV object knn_cv with cv = 10
- URL: https://github.com/MMesgar/tutorial-data-science/blob/main/capstone/7.SpaceX_Machine%20Learning%20Prediction_Part_5.ipynb

RESULTS

- We collected examples of landings represented with 16 different attributes and 1 class label that shows if a landing is successful or not
- The following slides summarize the results.

Results: Data collection-SpaceX API

Request and parse the SpaceX launch data using the GET request

In [8]:

```
print(response.content)
```


```
b'[{"fairings":{"reused":false,"recovery_attempt":false,"recovered":false,"ships":[]},"links":{"patch":{"small":"https://ima
ges2.imgbox.com/3c/0e/T8iJcSN3_o.png","large":"https://images2.imgbox.com/40/e3/GypSkayF_o.png"},"reddit":{"campaign":null,"
launch":null,"media":null,"recovery":null},"flickr":{"small":[],"original":[]},"presskit":null,"webcast":"https://www.youtub
e.com/watch?v=0a_00nJ_Y88","youtube_id":"0a_00nJ_Y88","article":"https://www.space.com/2196-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","s
tatic_fire_date_unix":1142553600,"net":false,"window":0,"rocket":"5e9d0d95eda69955f709d1eb","success":false,"failures":[{"ti
me":33,"altitude":null,"reason":"merlin engine failure"}],"details":"Engine failure at 33 seconds and loss of vehicle","crew
":[],"ships":[],"capsules":[],"payloads":["5eb0e4b5b6c3bb0006eeb1e1"],"launchpad":"5e9e4502f5090995de566f86","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":"5e9e289df35918033d3b2623","flight":1,"gridfins":false,"legs":fals
e,"reused":false,"landing_attempt":false,"landing_success":null,"landing_type":null,"landpad":null}],"auto_update":true,"tbd
":false,"launch_library_id":null,"id":"5eb87cd9ffd86e000604b32a"},{"fairings":{"reused":false,"recovery_attempt":false,"reco
vered":false,"ships":[]},"links":{"patch":{"small":"https://images2.imgbox.com/4f/e3/I0lkuJ2e_o.png","large":"https://images
2.imgbox.com/be/e7/iNqsqVYM_o.png"},"reddit":{"campaign":null,"launch":null,"media":null,"recovery":null},"flickr":{"small":
[],"original":[]},"presskit":null,"webcast":"https://www.youtube.com/watch?v=Lk4zQ2wP-Nc","youtube_id":"Lk4zQ2wP-Nc","articl
e":"https://www.space.com/3590-spacex-falcon-1-rocket-fails-reach-orbit.html","wikipedia":"https://en.wikipedia.org/wiki/Dem
oSat"},"static_fire_date_utc":null,"static_fire_date_unix":null,"net":false,"window":0,"rocket":"5e9d0d95eda69955f709d1eb","
success":false,"failures":[{"time":301,"altitude":289,"reason":"harmonic oscillation leading to premature engine shutdown"}]
,"details":"Successful first stage burn and transition to second stage, maximum altitude 289 km, Premature engine shutdown a
t T+7 min 30 s, Failed to reach orbit, Failed to recover first stage","crew":[],"ships":[],"capsules":[],"payloads":["5eb0e4
b6b6c3bb0006eeb1e2"],"launchpad":"5e9e4502f5090995de566f86","flight_number":2,"name":"DemoSat","date_utc":"2007-03-21T01:10:
```

Results: Data collection-SpaceX API

Filter the dataframe to only include Falcon 9 launches

In [12]:
```python
# Get the head of the dataframe
data.head()
```

Out[12]:

	static_fire_date_utc	static_fire_date_unix	net	window	rocket	success	failures	details	crew	ships	capsules	
0	2006-03-17T00:00:00.000Z	1.142554e+09	False	0.0	5e9d0d95eda69955f709d1eb	False	[{'time': 33, 'altitude': None, 'reason': 'merlin engine failure'}]	Engine failure at 33 seconds and loss of vehicle	[]	[]	[]	[5eb0e4b5b6c3bb00(
1	None	NaN	False	0.0	5e9d0d95eda69955f709d1eb	False	[{'time': 301, 'altitude': 289, 'reason': 'harmonic oscillation leading to premature engine shutdown'}]	Successful first stage burn and transition to second stage, maximum altitude 289 km, Premature engine shutdown at T+7 min 30 s, Failed to reach orbit, Failed to recover first stage	[]	[]	[]	[5eb0e4b6b6c3bb00(
2	None	NaN	False	0.0	5e9d0d95eda69955f709d1eb	False	[{'time': 140, 'altitude': 35, 'reason': 'residual stage-1 thrust led	Residual stage 1 thrust led to collision between stage 1 and	[]	[]	[]	[5eb0e4b6b6c3bb00(5eb0e4b6b6c3bb00(

Results: Data collection-SpaceX API

A summary of data

In [23]:

```
# Show the head of the dataframe
df.describe()
```

Out[23]:

	FlightNumber	PayloadMass	Flights	Block	ReusedCount	Longitude	Latitude
count	94.000000	88.000000	94.000000	90.000000	94.000000	94.000000	94.000000
mean	54.202128	5919.165341	1.755319	3.500000	2.670213	-75.553302	28.581782
std	30.589048	4909.689575	1.197544	1.595288	3.412149	53.391880	4.639981
min	1.000000	20.000000	1.000000	1.000000	0.000000	-120.610829	9.047721
25%	28.250000	2406.250000	1.000000	2.000000	0.000000	-80.603956	28.561857
50%	52.500000	4414.000000	1.000000	4.000000	1.000000	-80.577366	28.561857
75%	81.500000	9543.750000	2.000000	5.000000	4.000000	-80.577366	28.608058
max	106.000000	15600.000000	6.000000	5.000000	10.000000	167.743129	34.632093

Results: Data collection-SpaceX API

Dealing with Missing Values

In [26]:
```
data_falcon9.isnull().sum()
```

Out[26]:
```
FlightNumber       0
Date               0
BoosterVersion     0
PayloadMass        5
Orbit              0
LaunchSite         0
Outcome            0
Flights            0
GridFins           0
Reused             0
Legs               0
LandingPad        26
Block              0
ReusedCount        0
Serial             0
Longitude          0
Latitude           0
dtype: int64
```

In [28]:
```
data_falcon9.isnull().sum()
```

Out[28]:
```
FlightNumber       0
Date               0
BoosterVersion     0
PayloadMass        0
Orbit              0
LaunchSite         0
Outcome            0
Flights            0
GridFins           0
Reused             0
Legs               0
LandingPad        26
Block              0
ReusedCount        0
Serial             0
Longitude          0
Latitude           0
dtype: int64
```

Results: Data collection-Web scraping

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.

In [9]:

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

Create a `BeautifulSoup` object from the HTML `response`

In [14]:

```
# Use BeautifulSoup() to create a BeautifulSoup object from a response text content
soup = BeautifulSoup(content, 'html.parser')
```

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

In [15]:

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

```
<title>List of Falcon 9 and Falcon Heavy launches - Wikipedia</title>
```

Results: Data collection-Web scraping

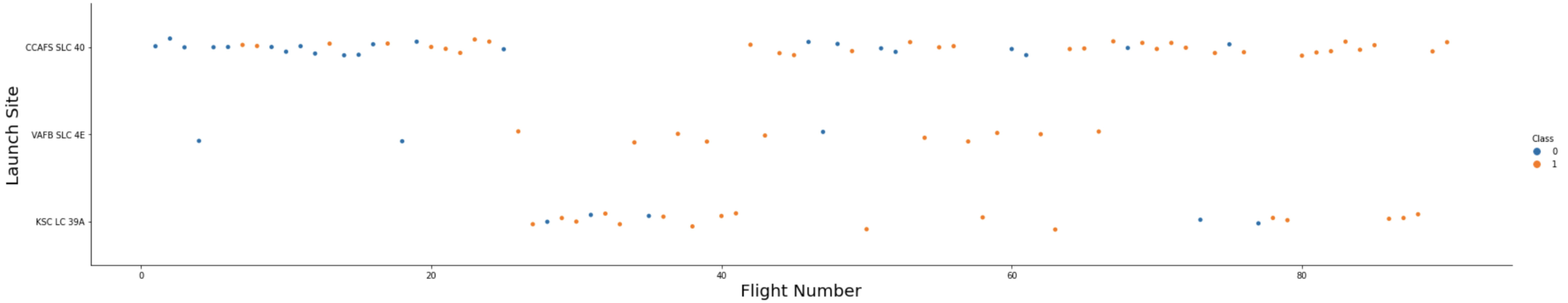
In [24]:

```
print(column_names)
```

```
['Flight No.', 'Date and time ( )', 'Launch site', 'Payload', 'Payload mass', 'Orbit', 'Customer', 'Launch outcome']
```

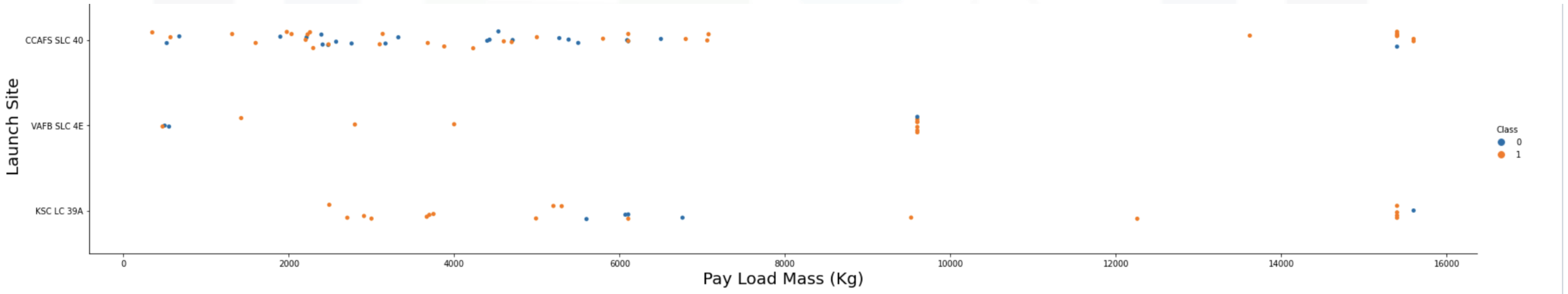
Results: EDA with data visualization

Results: Flight Number vs. Launch site scatter chart

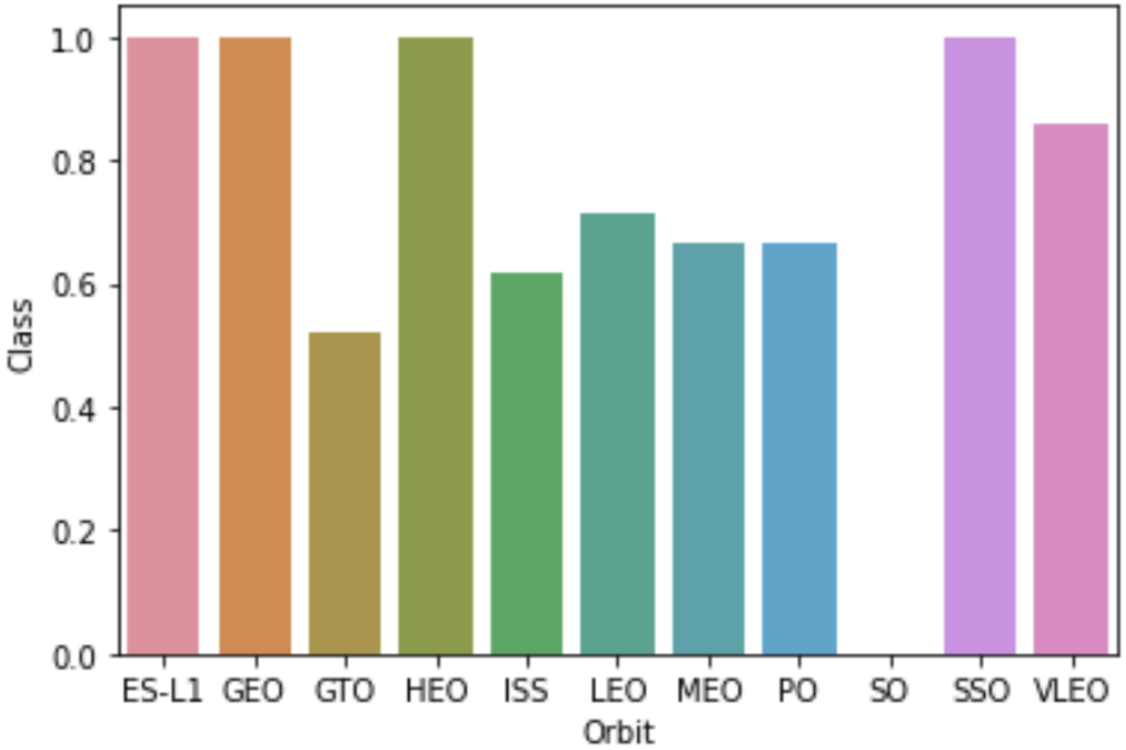


For CCAFS SLC 40, the higher the payload, the better the success rate.
For KSC LC39A, the opposite seems to be correct.

Results:
Payload vs. Launch site scatter chart

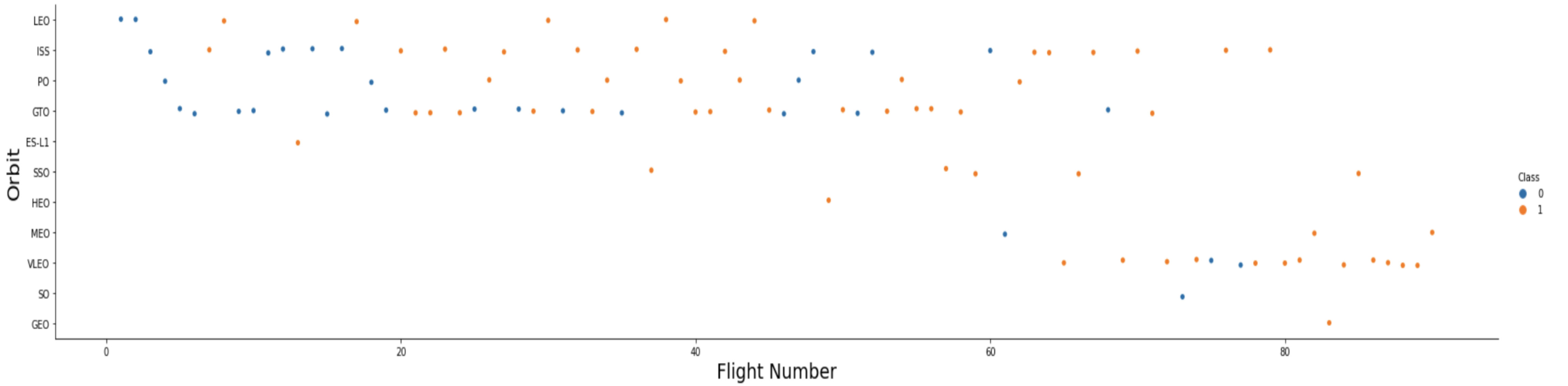


Results: Success rate vs. Orbit type bar chart

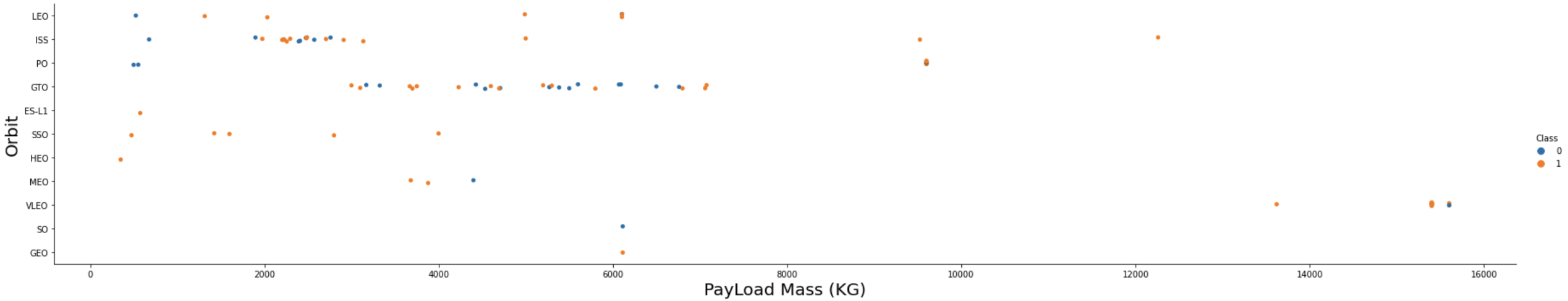


ESL1, GEO, HEO, SSO have not failed.

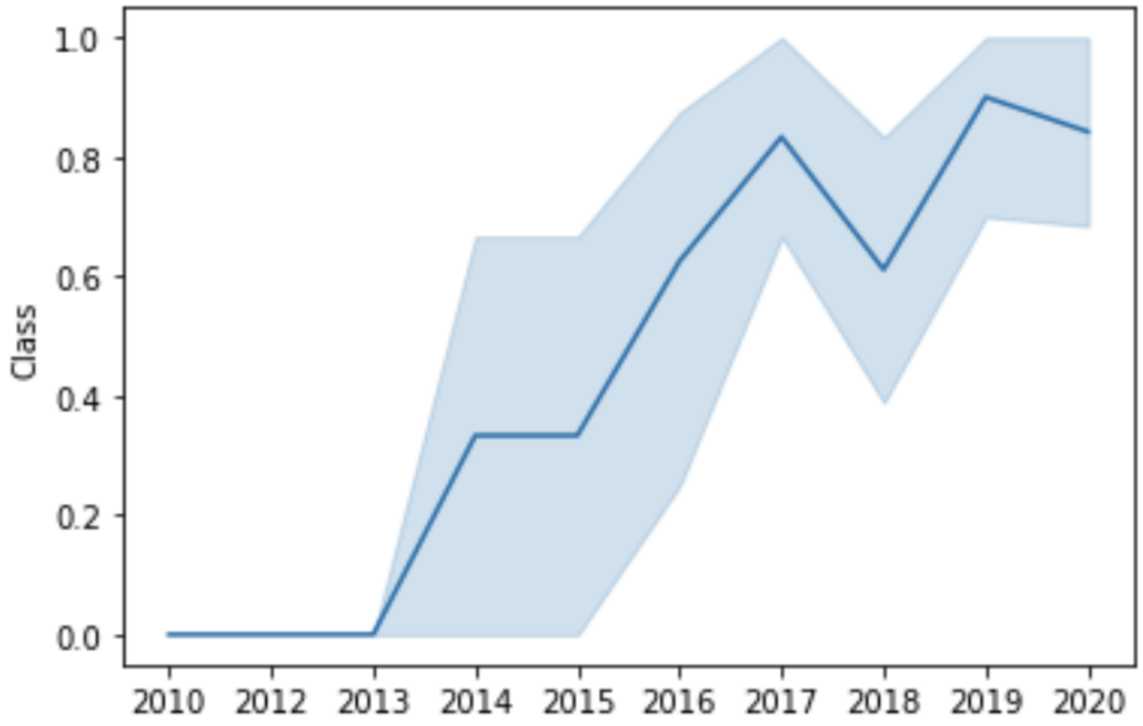
Results: Flight Number vs. Orbit type scatter chart



Results: Payload vs. Orbit type scatter chart



Results: Launch success yearly trend line



Results: EDA with SQL

Results: All launch site names: Find the names of the unique sites

Out[10]:

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

Results: Launch site names begin with `CCA`

Out[12]:

DATE	Time (UTC)	booster_version	launch_site	payload	payload_mass__kg_	orbit	customer	mission_outcome	Landing _Outcome
2010-04-06	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
2010-08-12	15:43:00	F9 v1.0 B0004	CCAFS LC-40	Dragon demo flight C1, two CubeSats, barrel of Brouere cheese	0	LEO (ISS)	NASA (COTS) NRO	Success	Failure (parachute)
2012-08-10	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
2013-01-03	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt
2013-03-12	22:41:00	F9 v1.1	CCAFS LC-40	SES-8	3170	GTO	SES	Success	No attempt

Results: Total payload mass: Calculate the total payload carried by booster from NASA

```
Done.
```

Out[25]:	1
	37249

Results: Average payload mass by F9 v1.1

```
Out[32]:     1
          3226
```

Results: First successful ground landing date

Out[44]:

	1
	2017-01-05

Results: Successful drone ship landing with payload between 4000 and 6000

Done.

Out[56]:

booster_version	Landing _Outcome	payload_mass__kg_
F9 FT B1022	Success (drone ship)	4696
F9 FT B1031.2	Success (drone ship)	5200

Results: Total number of successful and failure mission outcomes

1

45

Results: Boosters carried maximum payload

booster_version
F9 B5 B1048.4
F9 B5 B1049.4
F9 B5 B1049.5
F9 B5 B1060.2
F9 B5 B1058.3

Results: 2015 launch records

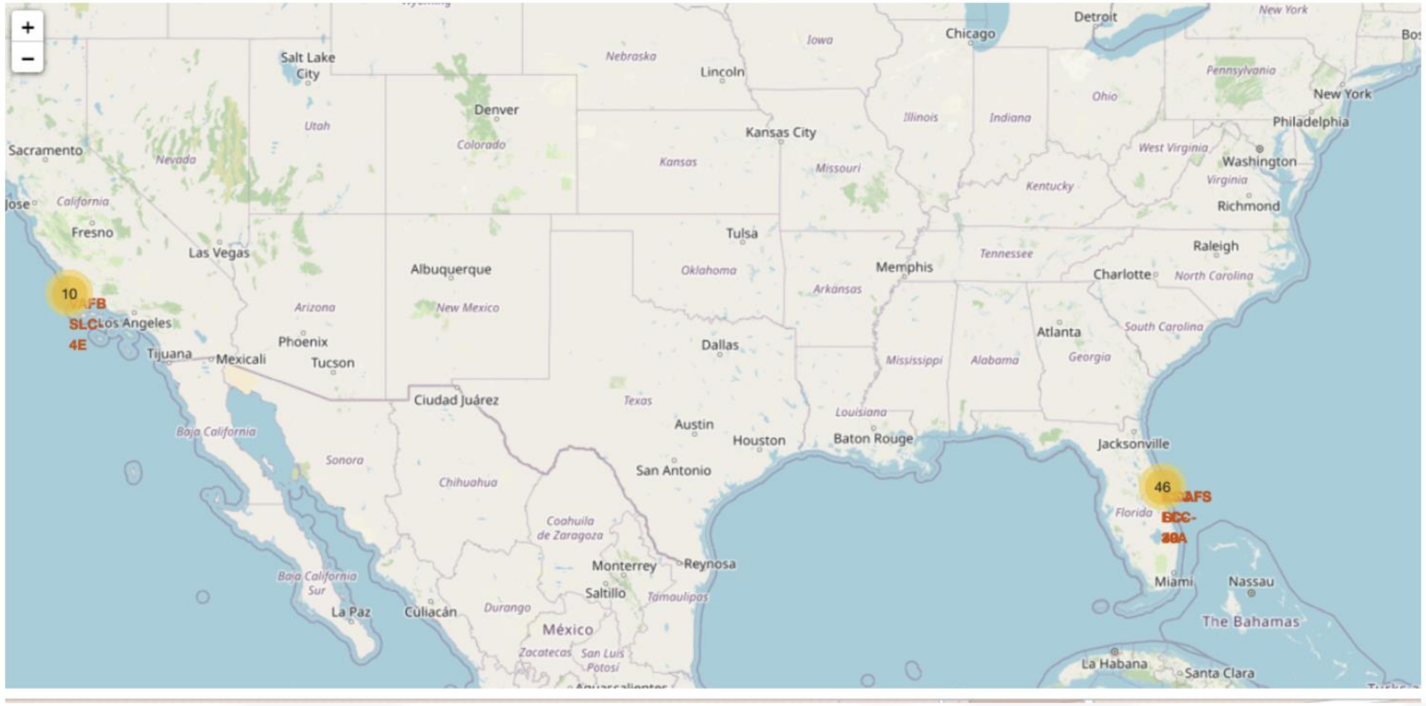
booster_version	launch_site
F9 v1.1 B1012	CCAFS LC-40

Results: Rank success count between 2010-06-04 and 2017-03-20

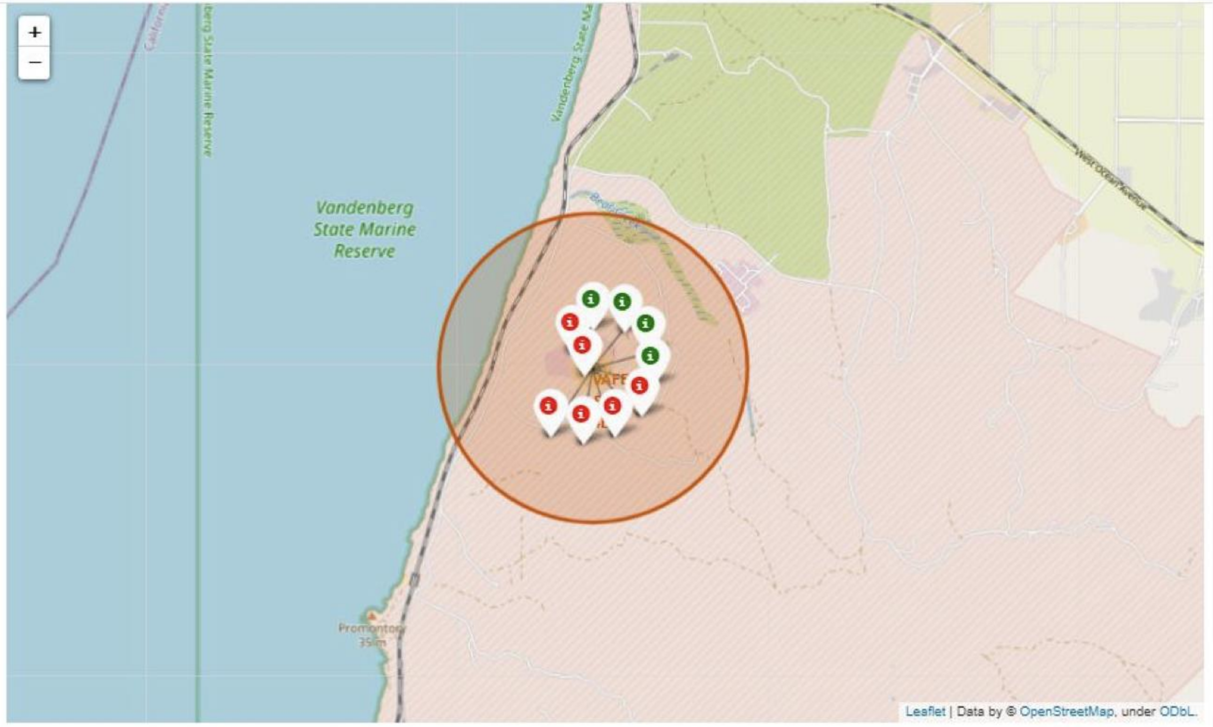
Landing _Outcome	cnt
No attempt	7
Failure (drone ship)	2
Success (drone ship)	2
Success (ground pad)	2
Controlled (ocean)	1
Failure (parachute)	1

Results: Interactive visual analytics

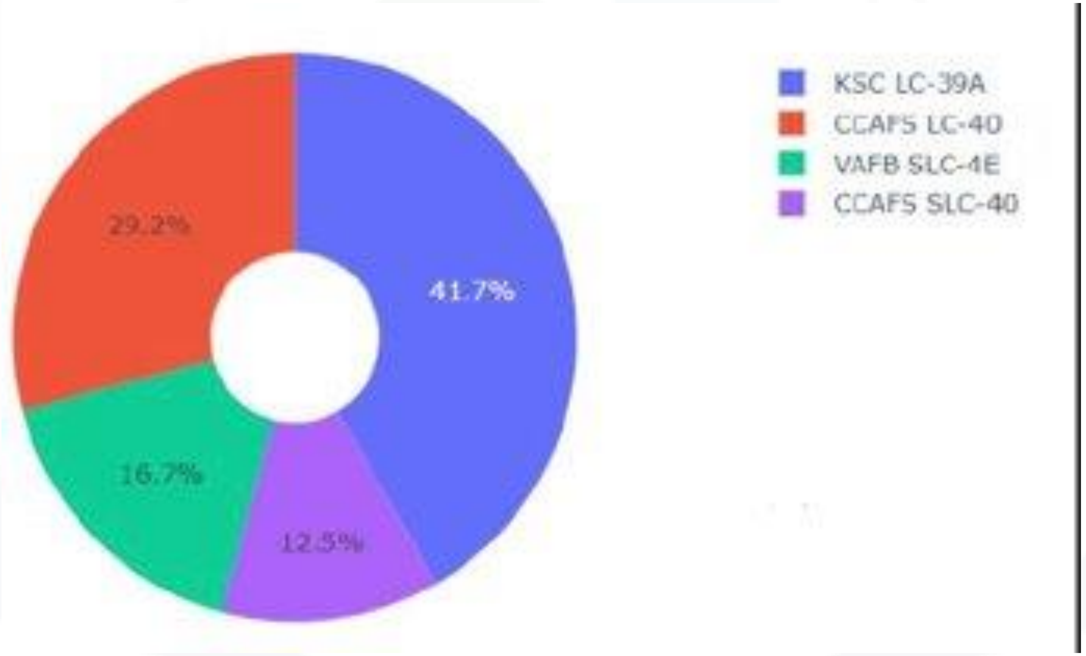
Results: Screenshot of all launch sites' markers on a global map



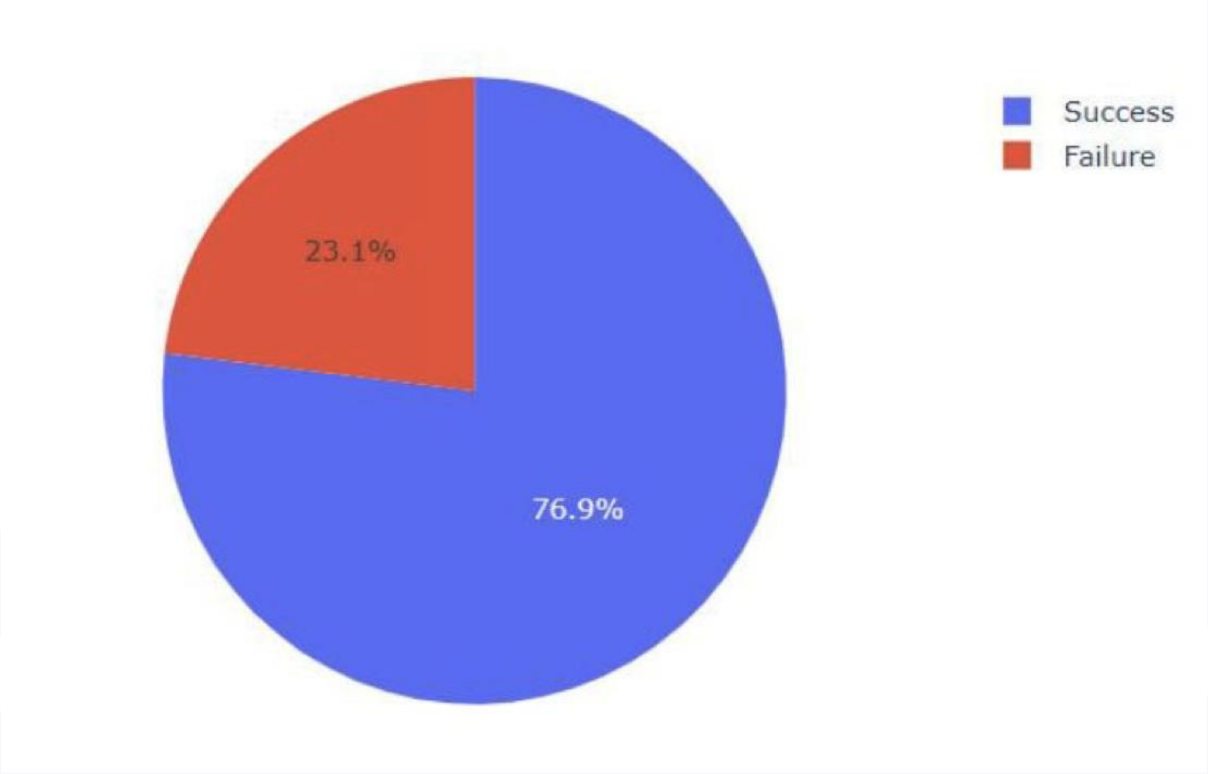
Results: Screenshot of all launch records per site on the map



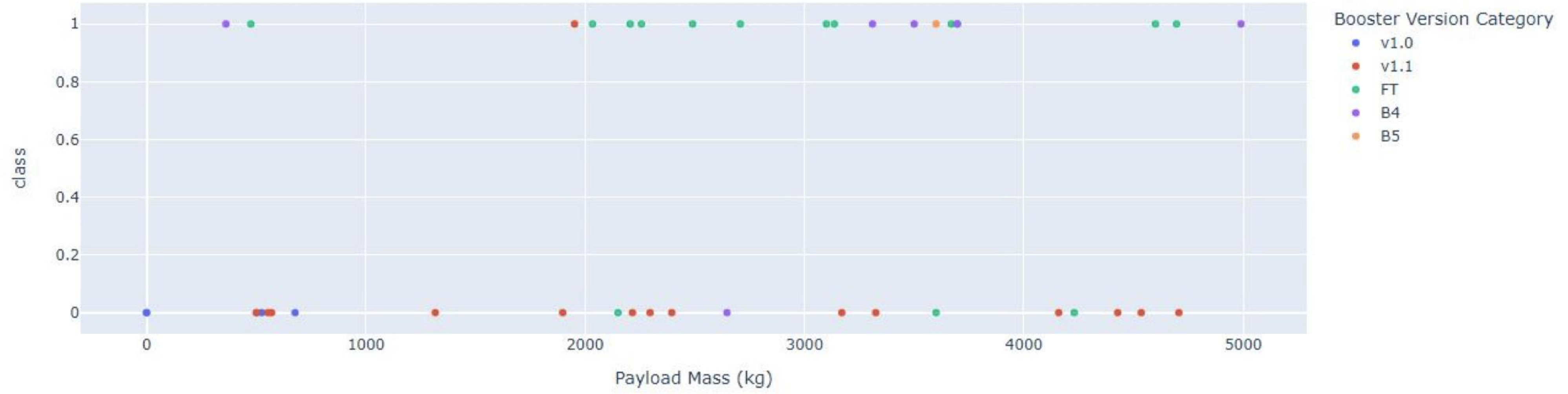
Results: Screenshot of launch success count for all sites, in a pie chart

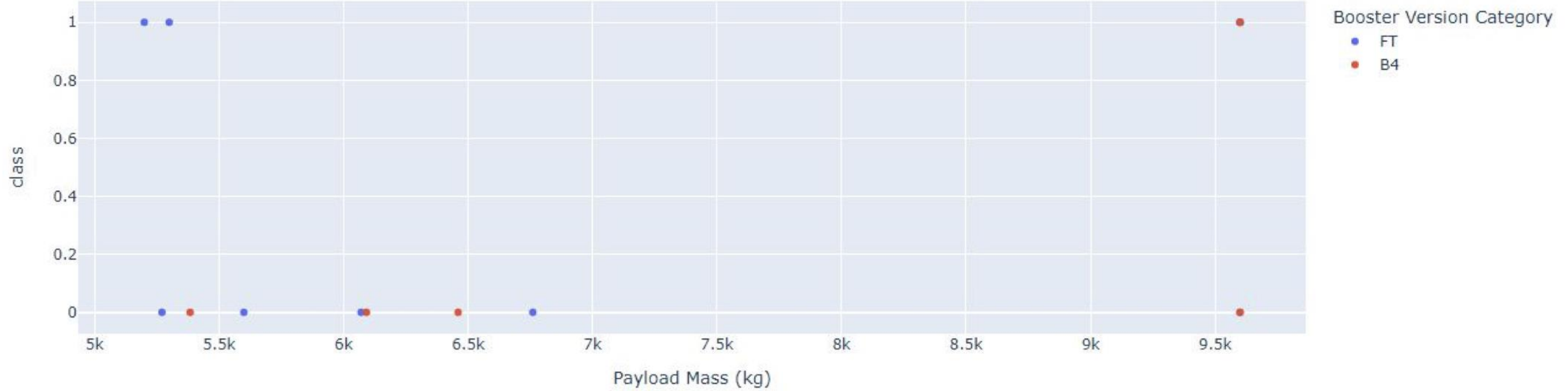


Results: Screenshot of the pie chart for the launch site with the highest launch success ratio



Results: Screenshot of Payload vs. Launch outcome scatter plot for all sites



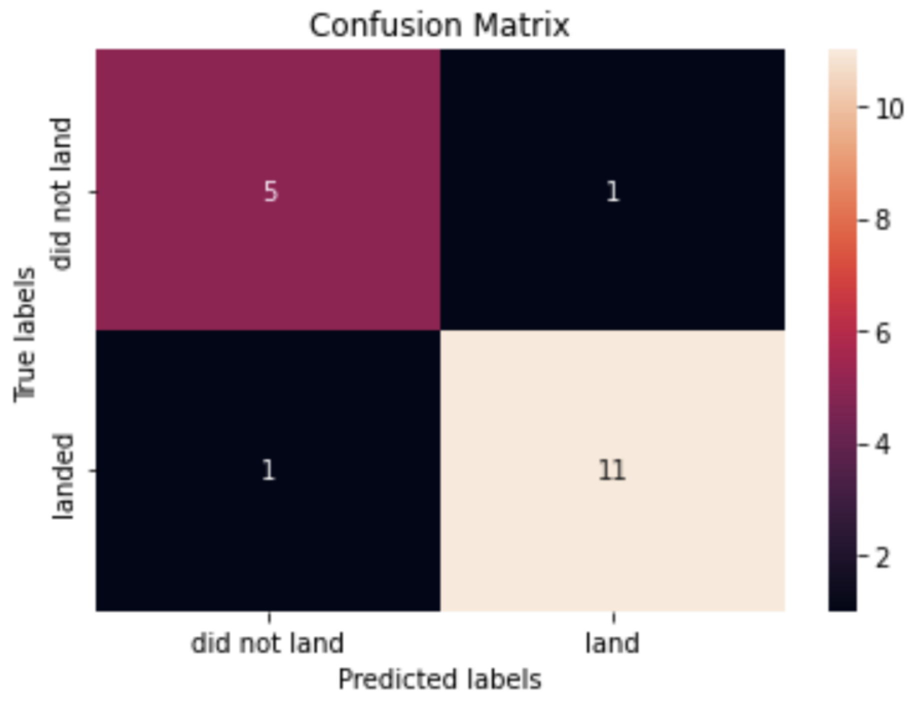


Results: Predictive analysis (classification)

Classification. KNN vs logisticRegression vs SVM vs DecisionTree

Model	Accuracy (%)
LogisticRegression	84.64
Support Vector Machines (SVM)	84.82
KNN	84.82
Decision Tree	**90.18**

Displayed the confusion matrix of the best model



DISCUSSION

- The DecisionTree model achieves the best accuracy compared with LogisticRegression and Support Vector Machines.
- So if we use our DecsisionTree model, we can predict if a future landing can be successful with about 10% error rate.

CONCLUSION

- Our goal was to predict if the Falcon 9 first stage lands successfully.
- We collected data from SpaceX wiki page.
- We then extracted several distinctive features.
- We find out the decision tree is the best performing model for achieving our goal.

APPENDIX

- The repository of this project is here:
- https://github.com/MMesgar/tutorial-data-science/tree/main/capstone