IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Alexander Griffiths
20th Nov 2023

Outline

Executive Summary

Methodology

- Use of Beautiful Soup to web-scrape Wikipedia and obtain Falcon 9 data
- Data cleaned and results explored using Pandas functionality and SQL queries
- Data plotted for further analysis using Seaborn and Folium functionality
- Dashboard creation for interactive analysis of results
- One-hot-encoding applied to allow for Machine Learning analysis with 4 different models

Results

- We can supply a ML model with Accuracy of 0.85 for predicting the ability to re-use Stage 1
- The biggest influence on success was the launch being more recent, suggesting experience is required
- Launch sites should be near the sea and transport links, far from cities. If able to use it, KSC LC-39A launch site was most successful
- Lighter payloads were more successful overall, with optimal being 3,000-4,000kg
- Most successful orbit types were ES-L1, GEO, HEO and SSO

Introduction

The Problem:

- SpaceY, a new space exploration business would like to compete with SpaceX so wishes to analyse their launch data to estimate how much a launch may cost
- SpaceX is competitive in the space exploration field in that it has much lower costs than competitors. Space Y believes this is due to SpaceX's re-use of the expensive Stage 1 of their rockets.
- As such, we aim to determine if it is possible to predict the success of a Stage 1 landing (meaning it can be reused).

Required Data:

- SpaceX launch data is available publicly on Wikipedia, so we need to analyse this data to work out what parameters are most likely to indicate a Stage 1 landing will be successful. For the sake of this IBM course, historical data is being used.

Section 1

Methodology

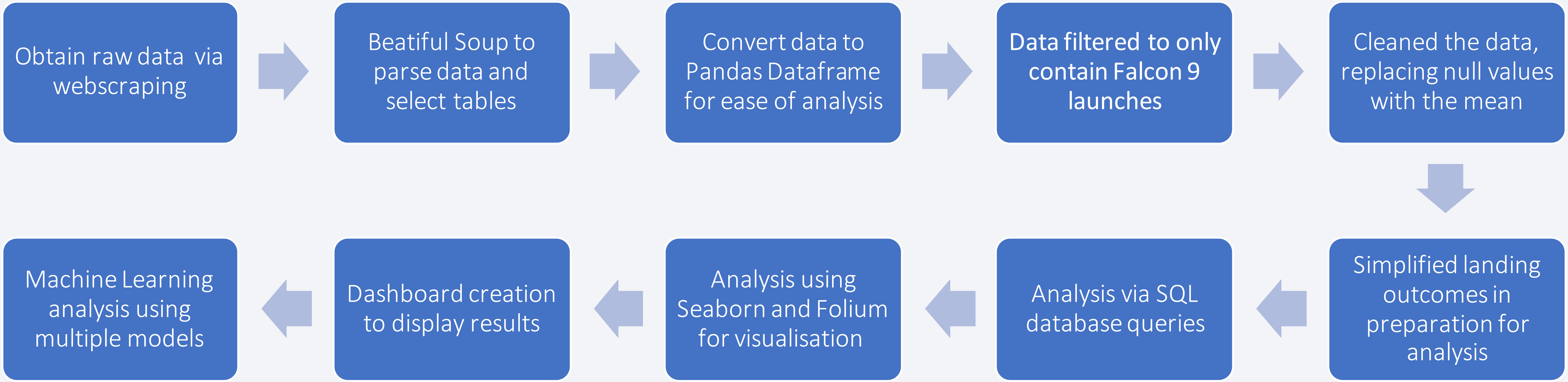
Methodology

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

- SpaceX launch data is freely available online via Wikipedia and their own website
- The below flowchart shows how the data was collected then processed:



Data Collection – SpaceX API

- Github Link:

https://github.com/AHGriffiths/IBM-Final-Project/blob/main/1.IBM-capstone-data-collection-api.ipynb

Using Requests API to obtain data:

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

```
response = requests.get(spacex_url)
```

Convert to .json and normalise data

```
response = requests.get(static_json_url)
response.json()
data = pd.json_normalize(response.json())
```

Use custom functions to clean data

```
getLaunchSite(data)
getPayloadData(data)
getCoreData(data)
```

Assign to Dataframe

```
launch_dict = {'FlightNumber': list(data['flight_number']),
'Date': list(data['date']),
'BoosterVersion':BoosterVersion,
'PayloadMass':PayloadMass,
'Orbit':Orbit,
'LaunchSite':LaunchSite,
'Outcome':Outcome,
'Flights':Flights,
'GridFins':GridFins,
'Reused':Reused,
'Legs':Legs,
'LandingPad':LandingPad,
'Block':Block,
'ReusedCount':ReusedCount,
'Serial':Serial,
'Longitude': Longitude,
'Latitude': Latitude}

df = pd.DataFrame.from_dict(launch_dict)
```

Filter results

```
data_falcon9 = df[df['BoosterVersion']!='Falcon 1']
```

```
meanPayload = data_falcon9['PayloadMass'].mean()
data_falcon9['PayloadMass'].replace(to_replace=np.nan, value=meanPayload, inplace=True)
data_falcon9.isnull().sum()
```

Export file for analysis

```
data_falcon9.to_csv('dataset_part_1.csv', index=False)
```

Data Collection - Scraping

Requests API to scrape text from Wikipedia page and Beautiful Soup to parse it

```
wikihtml = requests.get(static_url).text
soup = BeautifulSoup(wikihtml, "html.parser")
```

Use Beautiful Soup to further search data and select relevant table

```
tablelist = soup.find_all(name='table')
html_tables = tablelist
first_launch_table = html_tables[2]
```

Extract column headers

```
column_names = []
allth = first_launch_table.find_all(name='th')
for i,row in enumerate(allth):
    name = extract_column_from_header(row)
    if name is not None and len(name) > 0:
        column_names.append(name)
```

Parse remaining data into dictionary then Dataframe (please see Notebook for full code)

```
df= pd.DataFrame({ key:pd.Series(value) for key, value in launch_dict.items() })
```

Export to csv for future analysis

```
df.to_csv('spacex_web_scraped.csv', index=False)
```

GitHub Link:

https://github.com/AHGriffiths/IBM-Final-Project/blob/main/2.IBM-capstone-webscraping.ipynb

Data Wrangling

Load Dataset for analysis

```
url = "https://cf-courses-data.s3.us.cloud-object-storage\
.appdomain.cloud/IBM-DS0321EN-SkillsNetwork/datasets/dataset_part_1.csv"
df=pd.read_csv(url)
df.head(10)
```

Identify the frequency of different orbits and outcomes

```
landing_outcomes = df['Outcome'].value_counts()
landing_outcomes
```

```
True ASDS      41
None None      19
True RTLS      14
False ASDS      6
True Ocean      5
False Ocean     2
None ASDS       2
False RTLS      1
Name: Outcome, dtype: int64
```

Process the outcomes data to create a binary 'Class' result and add this to the Dataframe

```
landing_class = []
for count, Outcome in enumerate(df['Outcome']):
    if Outcome in bad_outcomes:
        landing_class.append(0)
    else:
        landing_class.append(1)
df['Class']=landing_class
```

Identify the average success rate

```
df["Class"].mean()
```

Export to csv for future processing

```
df.to_csv("dataset_part_2.csv", index=False)
```

GitHub Link:

https://github.com/AHGriffiths/IBM-Final-Project/blob/main/3.IBM-capstone-data%20wrangling.ipynb

EDA with Data Visualization

The following graphs were used to visualise the data:

- Categorical scatter plots to determine if there was a relationship between Flight Number, Payload Mass, Launch Site, Orbit type and Success Rate in various different combinations
- Bar chart to determine the overall success rate for launches to each orbit type
- Line graph to determine any relationship with Success Rate over time

- GitHub Link: https://github.com/AHGriffiths/IBM-Final-Project/blob/main/4.IBM-capstone-eda-visualisation.ipynb

EDA with SQL

The following SQL queries were performed to explore the data:

- Using "Distinct" to select unique Launch Sites and "Where-Like" to search for specific strings in the Launch Sites
- Using "Sum" and "Where" to determine total Payload Mass for a customer, and "Average" to determine average Payload Mass for a Booster type
- Using "Min" to determine earliest successful launch using a Ground Pad and "Where" with "And" for successful drone ship outcomes with 4k-6k Payload
- Using "Count" and "Group By" to show frequency of mission outcomes, and a subquery to show which Booster versions had the maximum Payload Mass
- Using "Substr" and "Where" to determine months in a year with failed drone ship outcomes
- Using "Between", "Group By", "Order by", "Desc" and "Where" to rank the frequency of landing outcomes between two dates

GitHub link: https://github.com/AHGriffiths/IBM-Final-Project/blob/main/5.IBM-capstone-sql-eda-coursera.ipynb

Build an Interactive Map with Folium

The following Folium features were used:

- Circles to identify Launch Sites
- Markers via Marker Clusters to show individual Launches, colour coded for launch outcome
- Lines to show the distance from a launch site to various points of interest

- This was done to explore possible reasons for the chosen sites (eg being closer to/further away various features) and more quickly identify which site was most successful

- GitHub Link: https://github.com/AHGriffiths/IBM-Final-Project/blob/main/6.IBM-capstone-folium.jupyterlite.ipynb

Build a Dashboard with Plotly Dash

The following interactive plots were implemented:

- Pie chart to show proportion of successful launches from each site
- Pie charts to show the success rate for each individual site
- Scatter graph to show the correlation between Payload Mass and Success Rate, which could be filtered by site via a dropdown and Payload Mass via an interactive slider

- These plots were chosen to explore the correlation between Payload Mass and Launch Site on the Success Rate of a mission, in a way that is intuitive to understand

- GitHub Link: https://github.com/AHGriffiths/IBM-Final-Project/blob/main/7.IBM-capstone-dash.py

Predictive Analysis (Classification)

1. Standardise data using StandardScalar transformation

```
transform = preprocessing.StandardScaler()
X = preprocessing.StandardScaler().fit(X).transform(X.astype(float))
```

2. Split data into train and test data via SciKitLearn

```
X_train, X_test, Y_train, Y_test = train_test_split(X,y,test_size=0.2,random_state=2)
```

3. Train a Logistic Regression model using Cross Validation and Grid Search to identify best parameters

```
parameters ={'C':[0.01,0.1,1],
             'penalty':['l2'],
             'solver':['lbfgs']}
```

```
parameters ={"C":[0.01,0.1,1],'penalty':['l2'], 'solver':['lbfgs']}
lr=LogisticRegression()
logreg_cv = GridSearchCV(lr,parameters,cv=10)

logreg_cv.fit(X_train,Y_train)
```

4. Repeat for SVM, Decision Tree and KNN models

5. Compare confusion matrices, accuracy and R^2 score of all models to identify most suitable

GitHub Link: https://github.com/AHGriffiths/IBM-Final-Project/blob/main/8.IBM-capstone-machine-learning-predictions.jupyterlite.ipynb

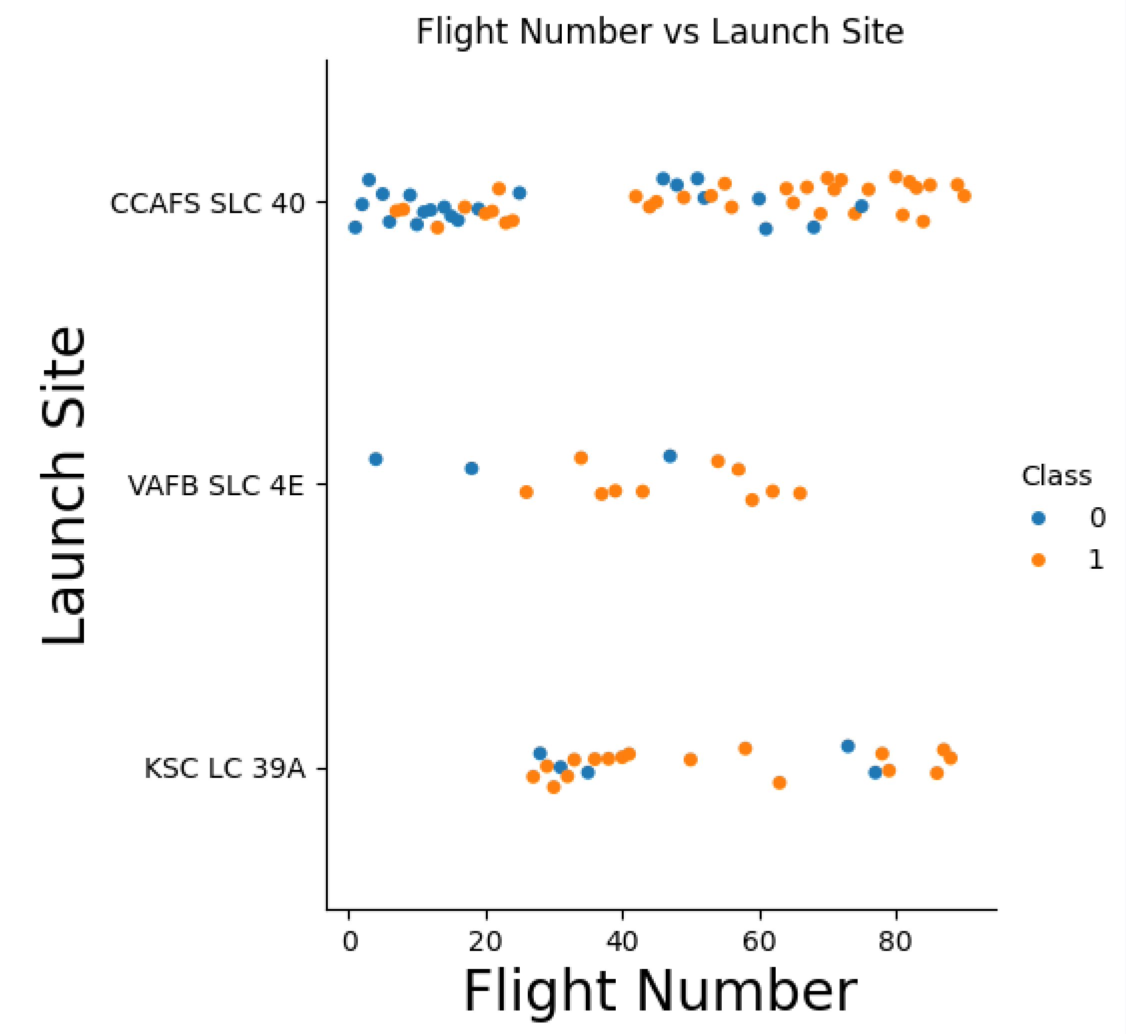
Results – Exploratory Data Analysis

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

Section 2

Insights drawn from EDA

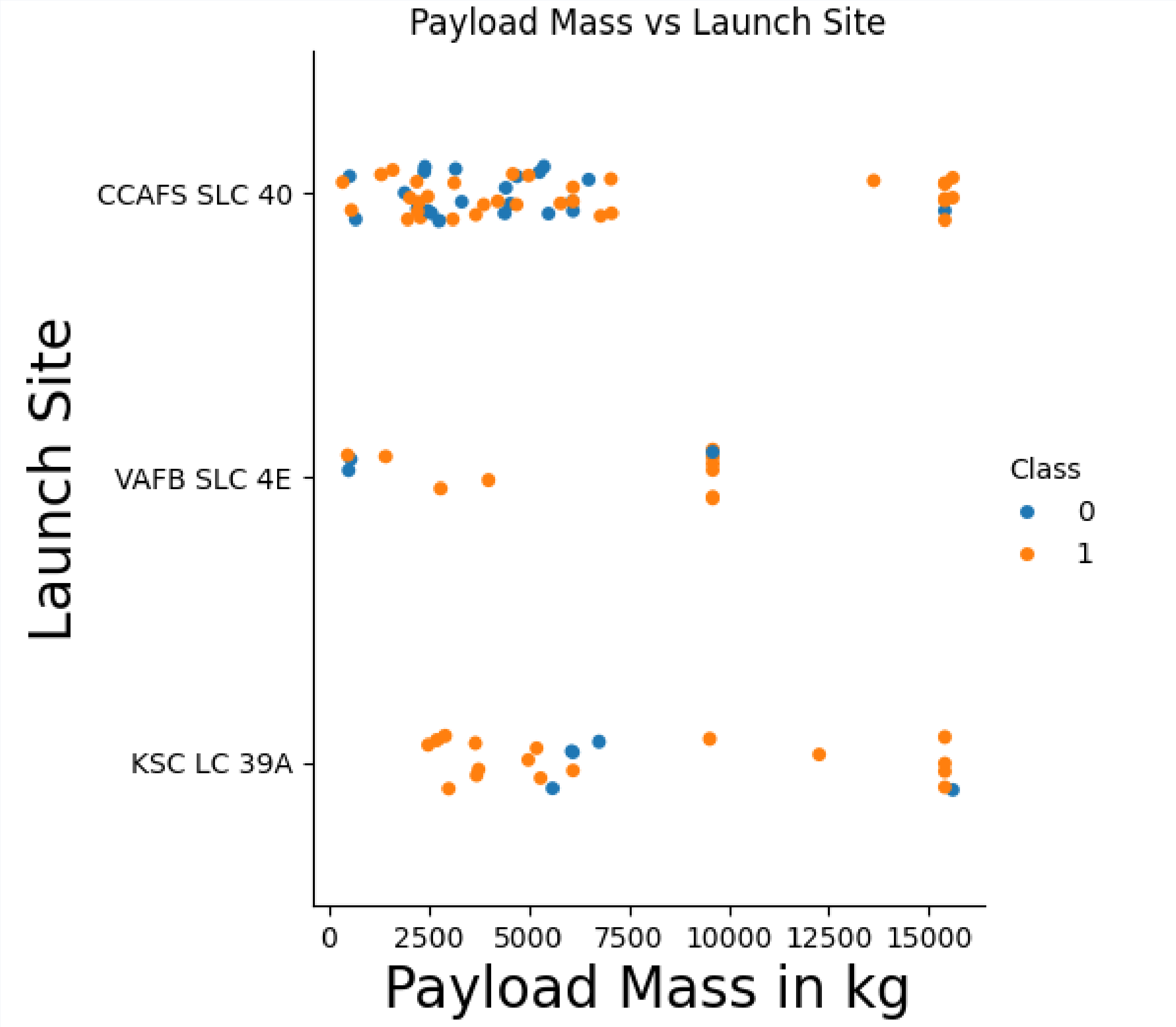
Flight Number vs. Launch Site



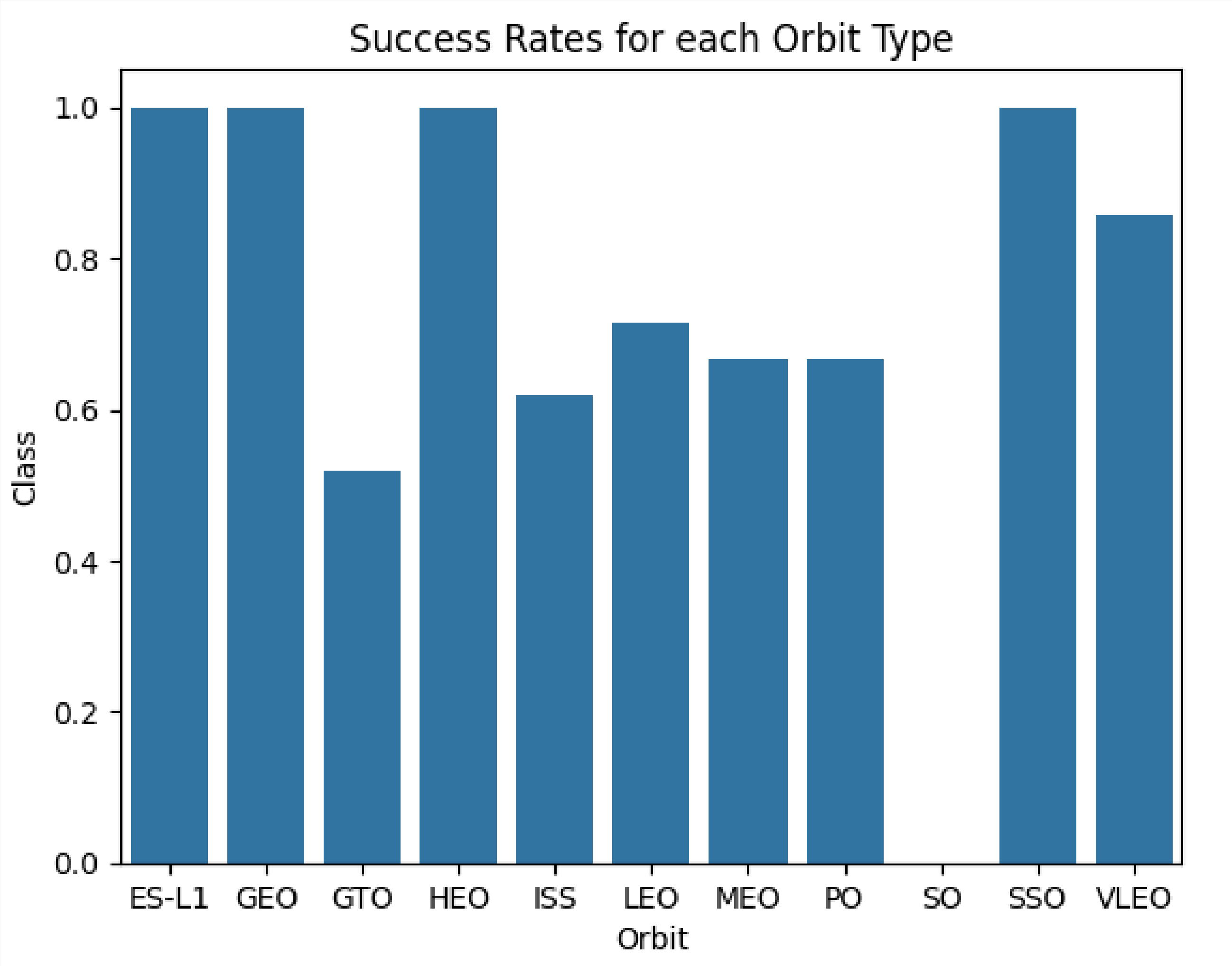
- Overall higher success rate with later launches
- VAFB shows the clearest trend of more success based on later launches
- CCAFS and KSC also show this trend, but it is slightly less clear
- This suggests that significant investment of time and money may be required to get a more reliable launch success rate

Payload vs. Launch Site

- While VAFB has no Payloads higher than 10,000kg, CCAFS and KSC both show a trend of much higher success rates with Mass > 10,000kg
- Lighter Payloads (<7500kg) had mixed success at CCAFS, but much higher success at KSC (especially <5000kg) and VAFB
- This suggests that lighter Payloads might be better to launch from KSC or VAFB, and heavier from CCAFS or KSC



Success Rate vs. Orbit Type

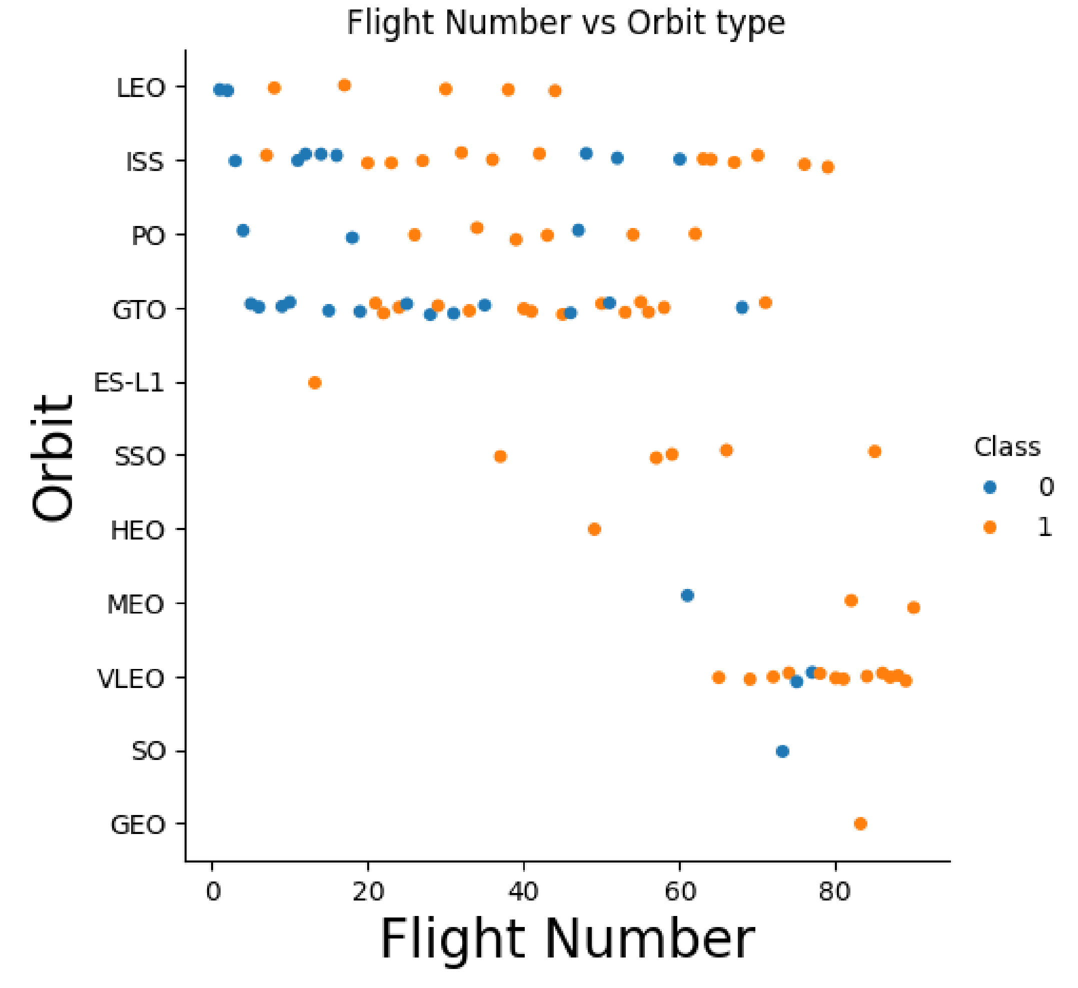


- ES-L1, GEO, HEO and SSO all have a 100% success rate
- VLEO is also highly successful
- SO is extremely unsuccessful, with 0% success
- The rest of the orbit types were of mixed success
- This suggests ES-L1, GEO, HEO, SSO are the best Orbits to attempt

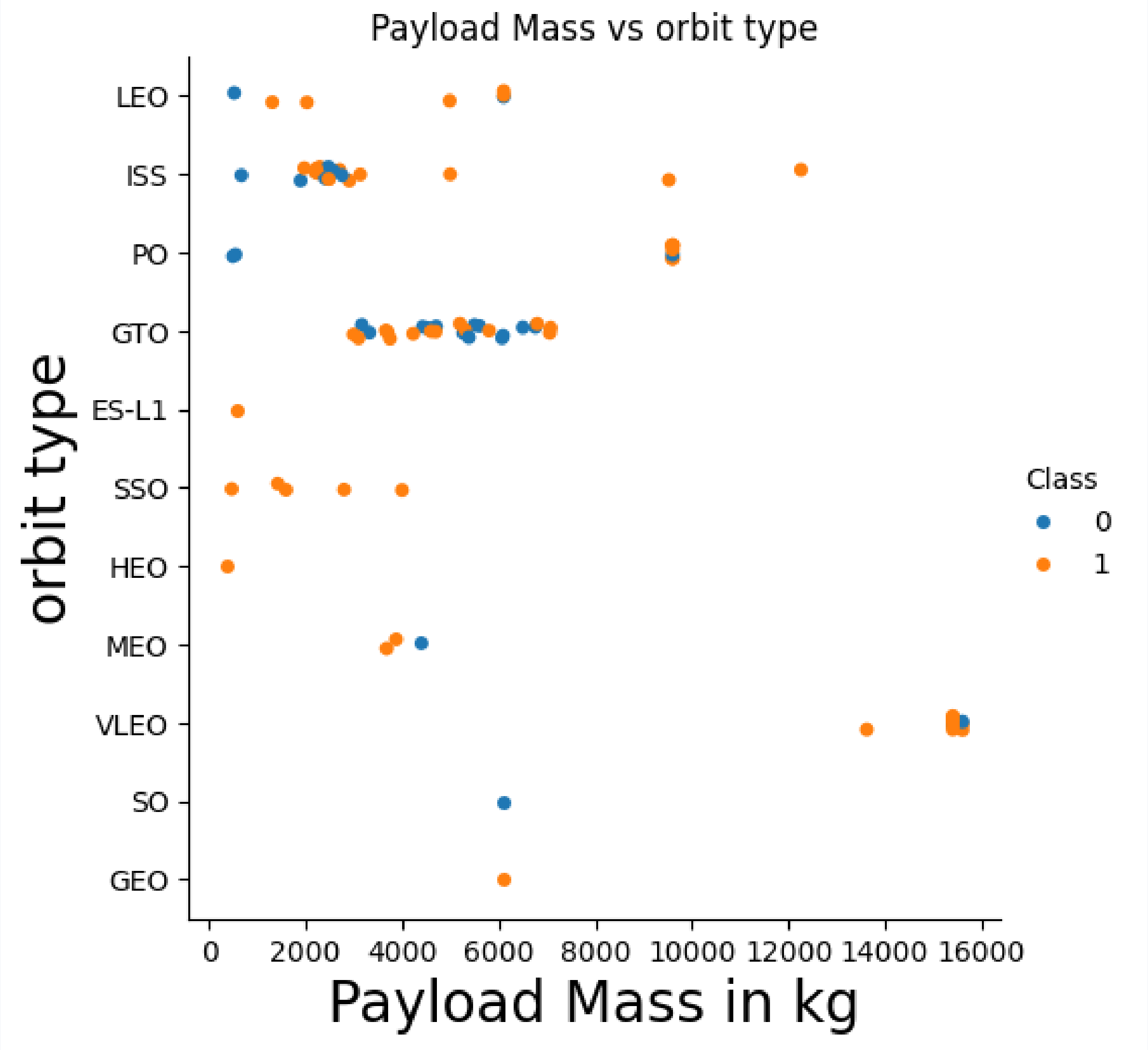
Flight Number vs. Orbit Type

- LEO/MEO show significant improvement in success rate at later launches. Similar, lesser trends for ISS/PO

- This suggests an improvement in success rate based on launch attempt and experience



Payload vs. Orbit Type

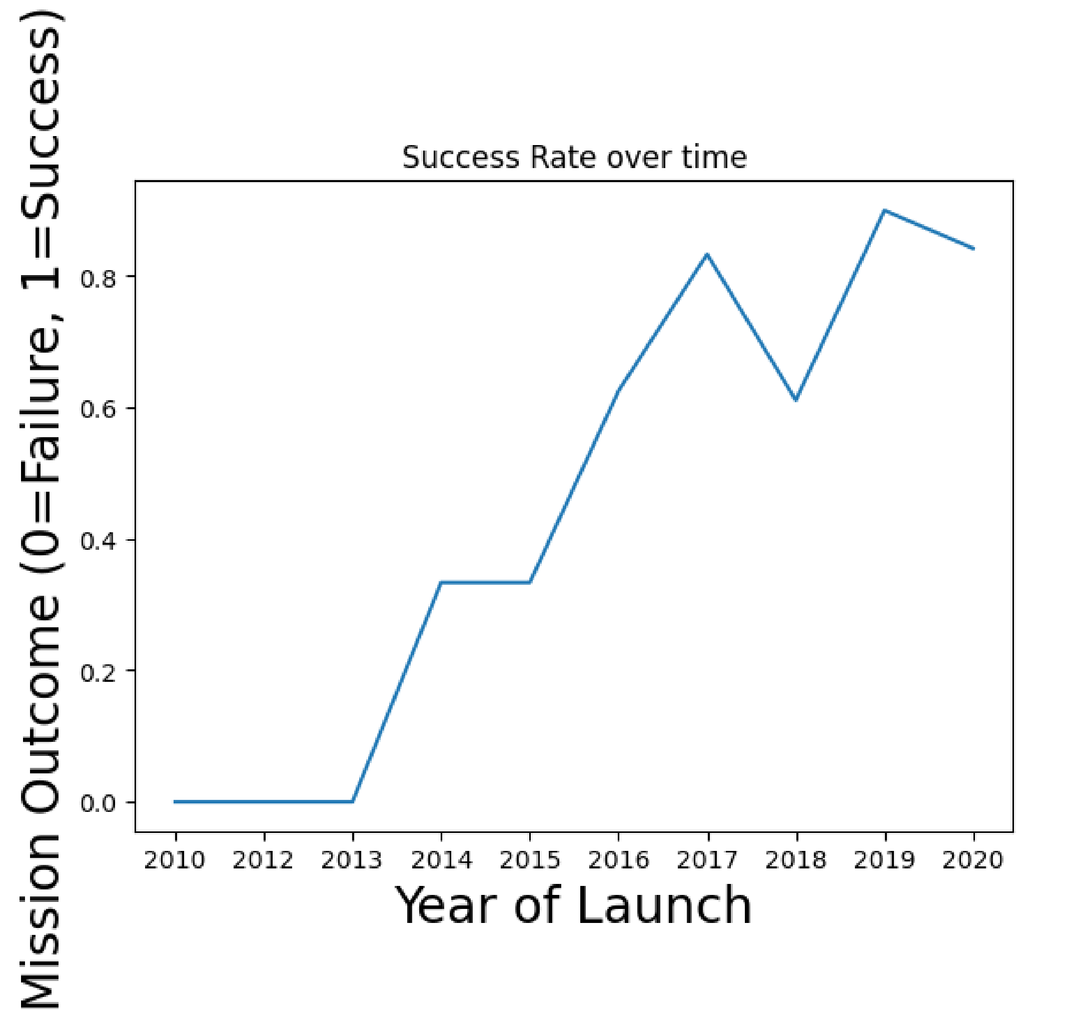


- VLEO is the only orbit type with exclusively high Payload launches
- ISS, LEO and PO have better success with higher payloads
- For the most successful orbit types (ES-L1, GEO, HEO, SSO and VLEO from 2 slides prior), we see that apart from VLEO, they all only used lighter payloads
- VLEO on the other hand may require a very heavy payload due to the distance of Orbit

Launch Success Yearly Trend

- This shows a clear upward trend of an increase in success rate over time, confirming the suspicions from earlier slides

- Interestingly, there was a slight dip in 2018 and 2020, but more research would be needed to determine a possible cause



All Launch Site Names

```
%%sql

select distinct("Launch_Site") from spacextable;
```

```
 * sqlite:///my_data1.db
Done.
```

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

- Use of the "DISTINCT" argument to return each Launch Site that shows in the database once, irrespective of how many entries it has
- As we know from previous slides, there are only 4 launch sites

Launch Site Names Begin with 'CCA'

```
%%sql

select * from spacextable
where "Launch_Site" like 'CCA%'
limit 5;
```

```
 * sqlite:///my_data1.db
Done.
```

Date	Time (UTC)	Booster_Version	Launch_Site	Payload	PAYLOAD_MASS__KG_	Orbit	Customer	Mission_Outcome	Landing_Outcome
2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Failure (parachute)
2010-12-08	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)
2012-05-22	7:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
2012-10-08	0:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	No attempt
2013-03-01	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	No attempt

- Use of the "WHERE-LIKE" argument to filter results based on an input string
- The "LIMIT" clause has also been used to reduce the output to the first 5 rows
- This shows that the first 5 launches from the two sites with "CCA" in their name were all from the same launch site, indicating an early preference for this site

Total Payload Mass

```
%%sql
select sum(payload_mass__kg_) as Total_Payload from spacextable
where customer = 'NASA (CRS)';
```

```
 * sqlite:///my_data1.db
Done.
```

Total_Payload
45596

- Use of "SUM" to obtain the aggregate values in a column, filtered by the results of the "WHERE" clause and labeled using "AS"
- This gives the result of 45,596kg total Payload for NASA launches
- Considering some Payloads are over 10,000kg each, this suggests either few launches or a low average Payload for NASA launches

Average Payload Mass by F9 v1.1

```
%%sql
select avg(payload_mass__kg_) as "Average_F9_v1.1_Payload" from spacextable
where "Booster_Version" like 'F9 v1.1%';
```

```
 * sqlite:///my_data1.db
Done.
```

Average_F9_v1.1_Payload
2534.6666666666665

- Use of "AVG" argument to calculate the mean of supplied values, using "WHERE-LIKE" to again filter results and "AS" to rename the result
- This returns an average Payload for the F9 v1.1 of 2534.67kg, which is relatively light

First Successful Ground Landing Date

```
%%sql

select min("Date") as EARLIEST_GROUND_PAD from spacextable
where "Landing_Outcome" = "Success (ground pad)";
```

```
 * sqlite:///my_data1.db
Done.
```

EARLIEST_GROUND_PAD
2015-12-22

- Use of "MIN" to get the earliest Date meeting a condition in the "WHERE" clause. This shows the first time SpaceX made a successful ground pad landing
- Considering that the Date of the first launch was in 2010, this suggests it took 5 years to get a successful Ground Pad landing

Successful Drone Ship Landing with Payload between 4000 and 6000

```
%%sql

select "Booster_Version", payload_mass__kg_, "Landing_Outcome" from spacextable
where "Landing_Outcome" = "Success (drone ship)"
    and payload_mass__kg_ > 4000 and payload_mass__kg_ < 6000;
```

```
 * sqlite:///my_data1.db
Done.
```

Booster_Version	PAYLOAD_MASS__KG_	Landing_Outcome
F9 FT B1022	4696	Success (drone ship)
F9 FT B1026	4600	Success (drone ship)
F9 FT B1021.2	5300	Success (drone ship)
F9 FT B1031.2	5200	Success (drone ship)

- Use of multiple conditions in the "WHERE" clause using "AND" to further filter results and show the Booster versions with a Payload Mass between 4,000 and 6,000kg with a successful Drone Ship landing
- Notably, only the FT Booster version is returned, suggesting it might be the Booster version designed for this Payload range and landing type

Total Number of Successful and Failure Mission Outcomes

- Use of "COUNT" and "GROUP BY" to return the frequency of each Mission Outcome
- Interestingly, there is a duplicate 'Success' entry here, potentially due to incorrect formatting of this entry in the data
- However, what is most notable is that there is only a single Mission Outcome recorded as a failure – suggesting that even if Stage 1 cannot be reused, it is highly likely that a rocket will deliver its payload to orbit

```
%%sql

select "Mission_Outcome", count(*) from spacextable
group by "Mission_Outcome";
```

```
 * sqlite:///my_data1.db
Done.
```

Mission_Outcome	count(*)
Failure (in flight)	1
Success	98
Success	1
Success (payload status unclear)	1

Boosters Carried Maximum Payload

```
%%sql

select "Booster_Version", payload_mass__kg_ from spacextable
where payload_mass__kg_ =
    (select max(payload_mass__kg_) from spacextable);
```

```
 * sqlite:///my_data1.db
Done.
```

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

- Use of a subquery to show all Booster Versions used with the maximum Payload Mass (15,600kg)
- Notably, every result is a B5 B10* model, suggesting it is the model designed for the heaviest payloads

2015 Launch Records

```
%%sql
select distinct("Landing_Outcome") from spacextable;


select substr(Date, 6,2) as Month, "Landing_Outcome", "Booster_Version", "Launch_Site" from spacextable
where "Landing_Outcome" = "Failure (drone ship)" and substr(Date,0,5) = '2015';
```

```
 * sqlite:///my_data1.db
Done.
Done.
```

Month	Landing_Outcome	Booster_Version	Launch_Site
01	Failure (drone ship)	F9 v1.1 B1012	CCAFS LC-40
04	Failure (drone ship)	F9 v1.1 B1015	CCAFS LC-40

- Use of "DISTINCT", "SUBSTR" and "WHERE" to use parts of a string (the year in Date) as part of a condition, and return only Failed Drone Ship landings in the year 2015, alongside the month it was in
- We can see that both failures were early in the year, the F9 v1.1 Booster and from CCAF SLC-40 site

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

```
%%sql
select "Landing_Outcome", count(*) as Freq from spacextable
where (substr(Date,1,4) || substr(Date,6,2) || substr(Date,9,2))
between '20100604' and '20170320'
group by "Landing_Outcome"
order by Freq desc;
```

```
 * sqlite:///my_data1.db
Done.
```

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

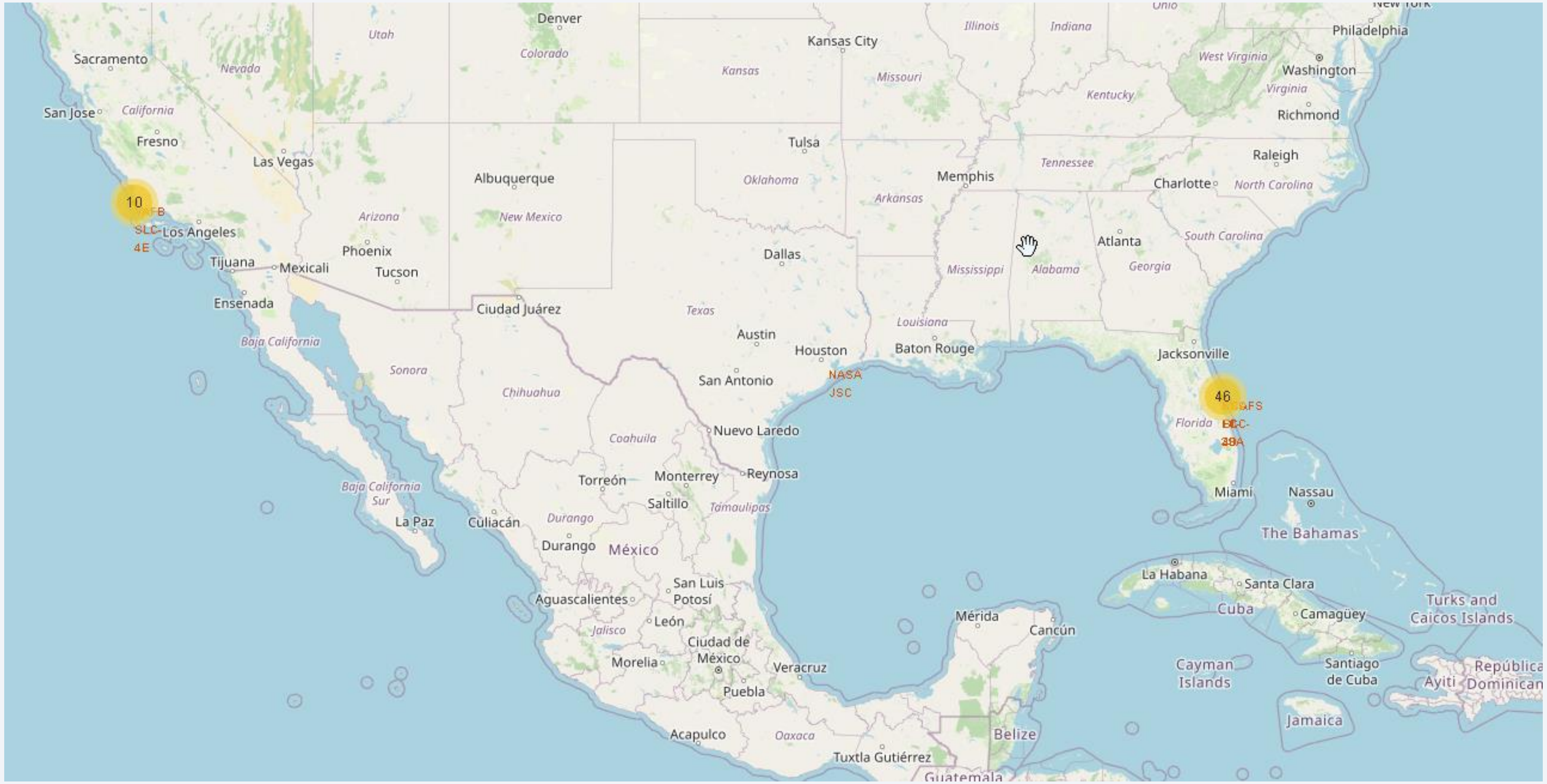
- Use of "COUNT", "WHERE", "SUBSTR", "BETWEEN", "GROUP BY", "ORDER BY" and "DESC" to return a list of Landing outcomes between specific dates in 2010 and 2017, ordered by frequency

- It is interesting that in this early stage of SpaceX launches, almost a third of launches did not even attempt to reuse Stage 1, and the remaining success rate was still low

Section 3

Launch Sites Proximities Analysis

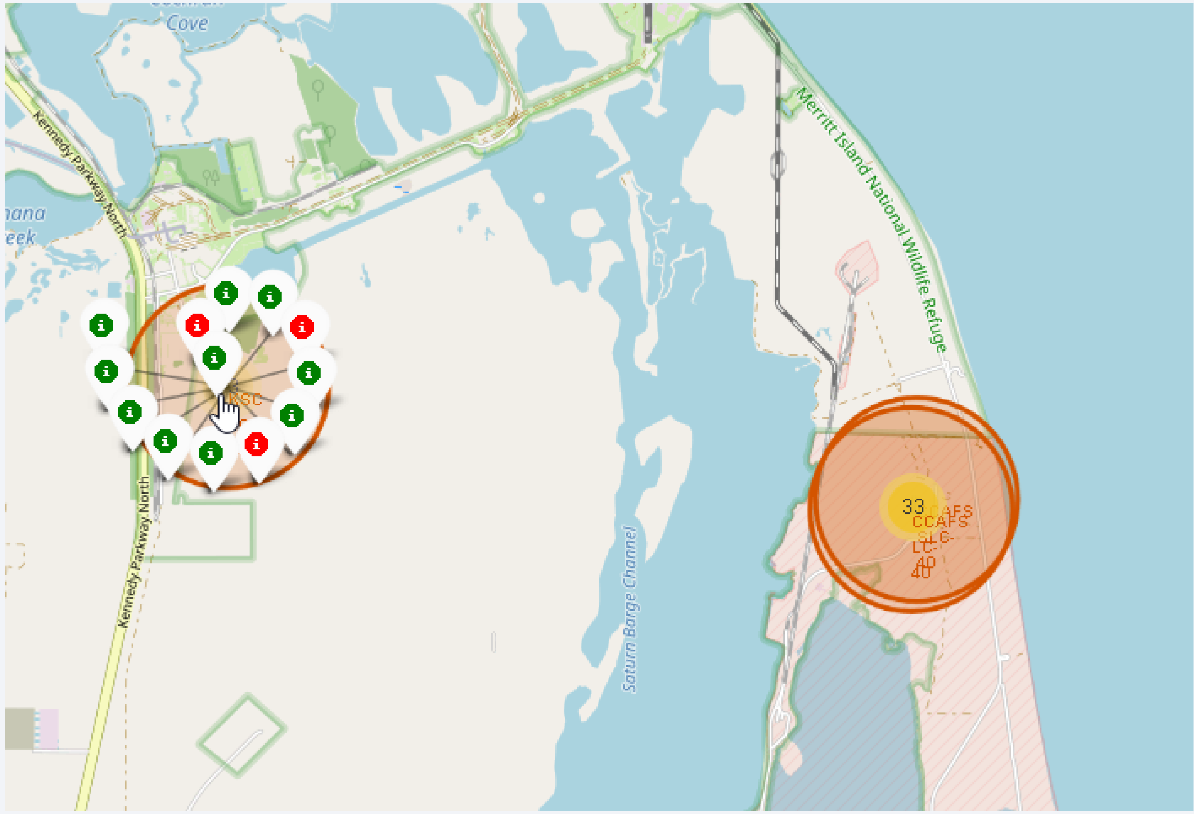
Distribution of Launch Sites



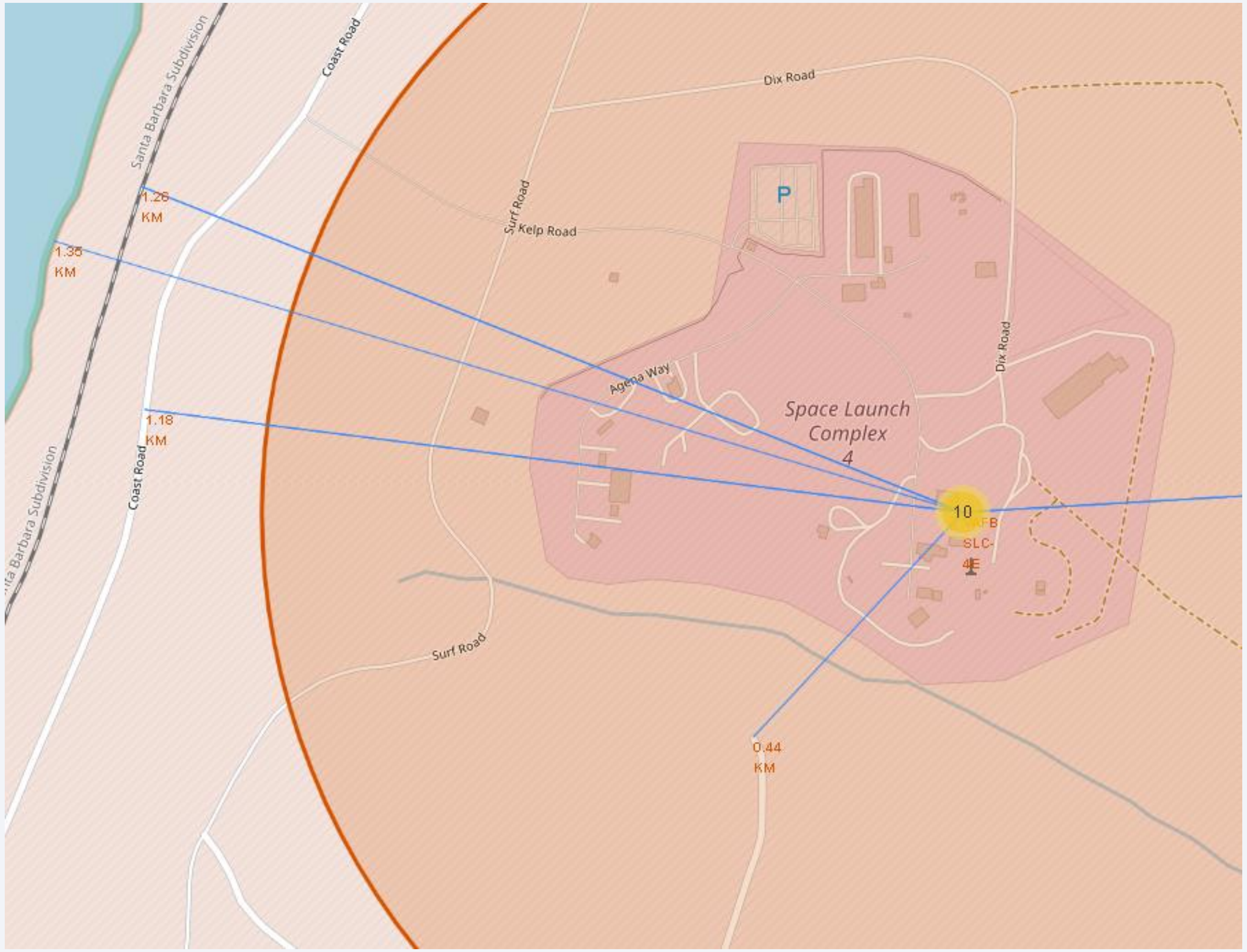
- Three of the Launch Sites are close together in Florida and overlapping
- All launch sites are on the coast
- All launch sites appear to NOT be near the major cities

Markers to show Success Rate at each Site

- To enhance interactivity, clicking on a Site's circle brings up a Cluster of markers showing the number of successes and failures at the site
- This allows easy visualisation to see that KAFB was the most successful launch site
- CCAFS SLC-40 however performed below average compared to the other sites



Distance Markers from Sites to Map Features



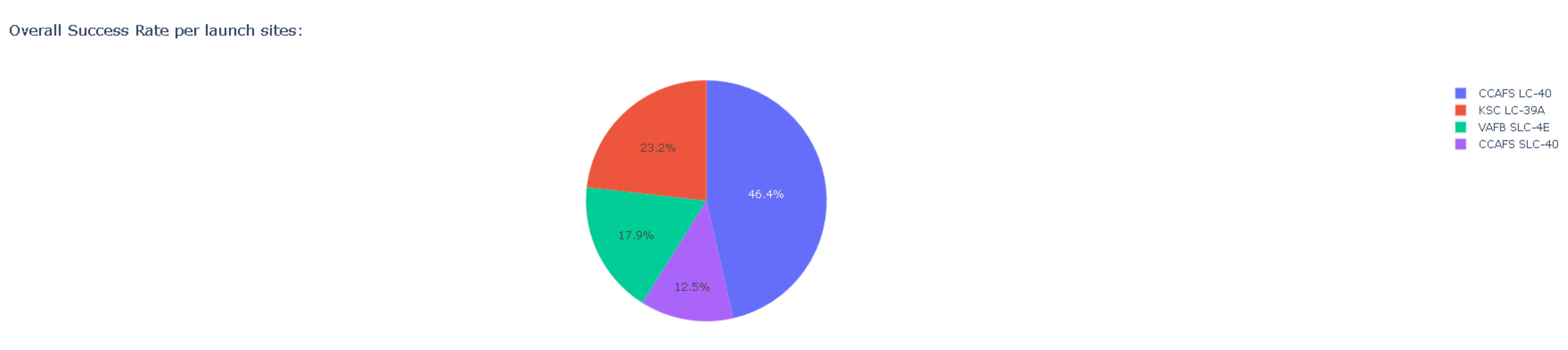
Please note that the distance to Lompoc, the nearest city to this site is 12km, but showing this makes all distances impossible to read in a screenshot

- These distance markers allow us to see that all sites are very close to roads, rail lines and the coast, but far from the nearest city
- This likely helps maximise efficiency for the delivery of resources and components, allows a safe place for a controlled crash (the sea) and minimises the risk to human life and city infrastructure in case of a catastrophe

Section 4

Build a Dashboard with Plotly Dash

Success Rate for each Site as a proportion of overall Success

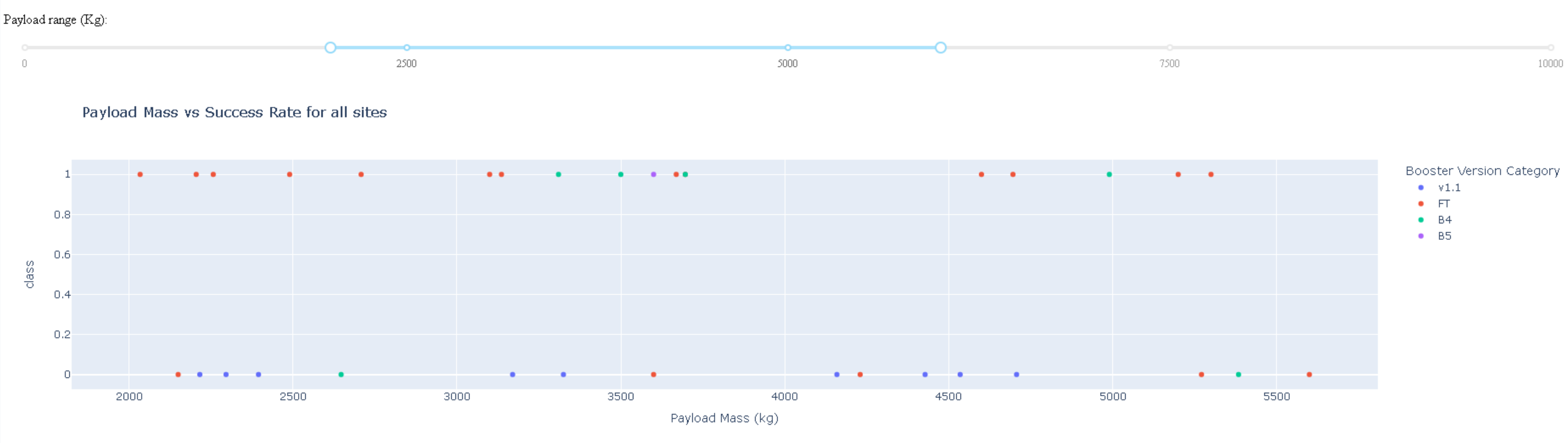


- Selecting "All Sites" from the Dashboard allows us to see a pie chart showing the proportion of successful launches by site
- This shows us that CCAFS LC-40 had the largest proportion of successes, but CCAFS SLC-40 the lowest

Individual Success Rate Pie Charts for each Launch Site

- Using the Launch Site drop-down menu, we can see the success rates for all of the sites individually
- This shows that KSC has the highest success rate, with 76.9%
- Note that CCAFS LC-40, while having the highest proportion of all successes, is slightly less successful, with 73.1% - this discrepancy is due to the larger number of launches from the site

Using the Interactive Payload Mass Slider



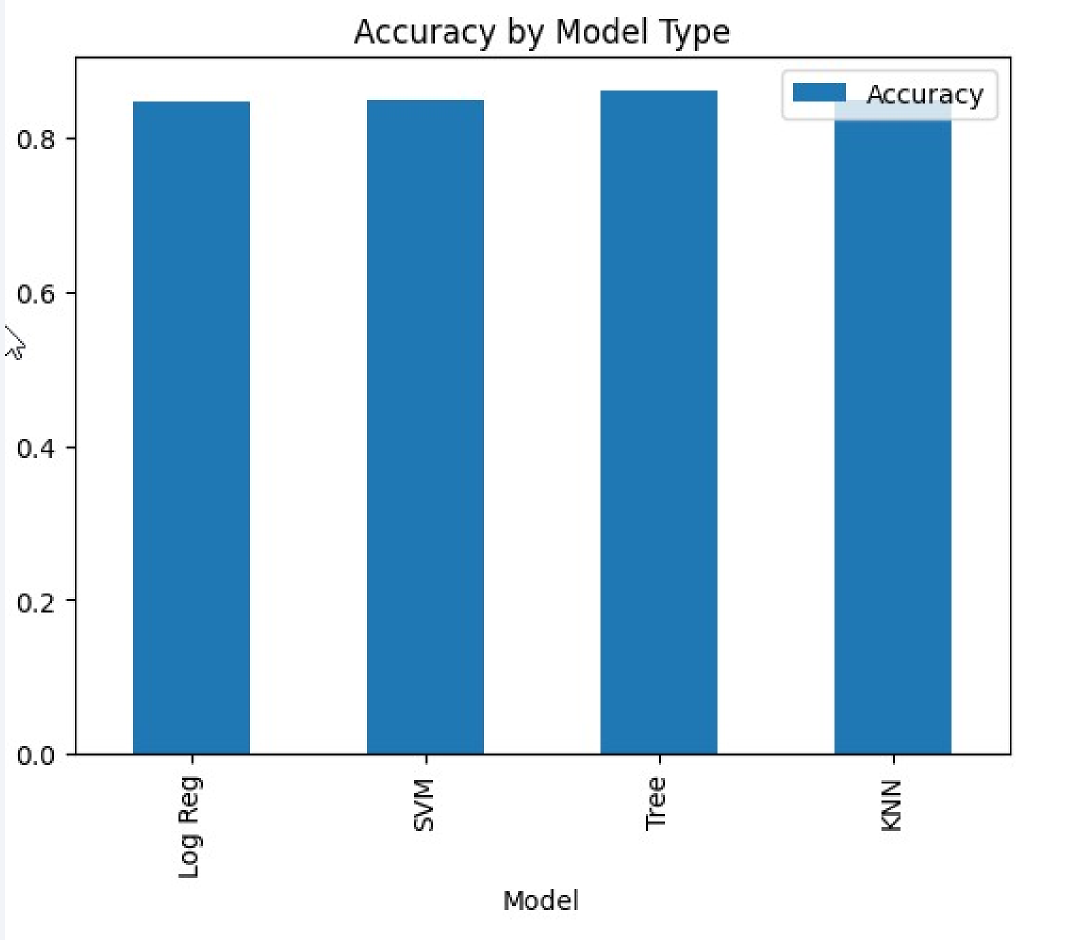
- The Payload range slider allows you to choose a custom range of data points to display on the graph, also filtered according to the Launch Site drop-down menu
- We can see that 3,000-4,000kg has a success rate of about 70% and about 80% at 3,500-4,000kg, making this the optimal light payload
- 4,000-4,500kg and 5,500-7,000kg ranges have a 0% success rate
- The FT Booster version is most successful, with a 60-70% success rate

Section 5

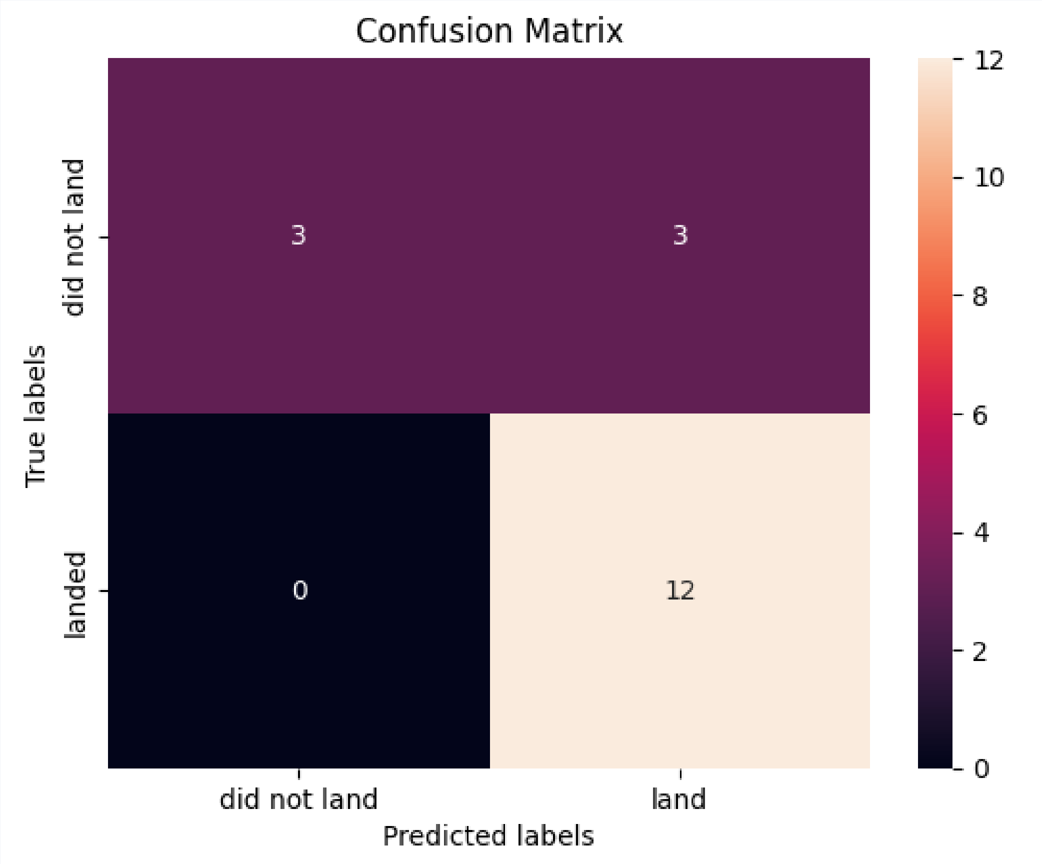
Predictive Analysis (Classification)

Classification Accuracy

- Surprisingly, all the 4 models showed the same accuracy when run
- This will be due to the limited dataset – for more accurate predictions, we would need a larger set of launch data, giving better data to train on (and more to test on)



Confusion Matrix



- The Confusion Matrices for all 4 models showed the same results – 3 True Negatives, 12 True Positives and only 3 False Positives
- The reason for the identical results will mostly be due to the small sample size – there are only 18 test samples to predict outcomes for

Conclusions – how to optimise Stage 1 reuse

- We can supply a ML model with Accuracy of 0.85 for predicting the ability to re-use Stage 1
  - However, it may not work optimally on real-world data due to the small data-set used to train it. Ideally, we would wait for a bigger dataset and re-train the models. At present, all of our models trained have the same accuracy.
- The biggest influence on success was the launch being more recent, suggesting experience is required, but also that initial costs will be high until you can successfully reuse Stage 1
  - Is it possible to collaborate with SpaceX engineers, hire some of their workers or buy the schematics for their Boosters?
- Launch sites should be near the sea and transport links, far from cities. If able to use it, KSC LC-39A launch site was most successful
  - The KSC launch site had the highest success rate – even with relatively few launches. Perhaps due to the fewer launches, it would be easier to hire the launch site, which would work especially well given the high success rate
- The following launch parameters were more successful:
  - Lighter payloads were more successful overall, with 3,000-4,000kg being the optimal range
  - The most successful orbit types were ES-L1, GEO, HEO and SSO
  - FT Booster was the most successful, with 70-80% success rate – can schematics be obtained to try and replicate its success?

Appendix

- All Notebooks and code used for the generation of supplied screenshots can be found in the following repository:
- https://github.com/AHGriffiths/IBM-Final-Project/tree/main

- Thanks to the IBM team for providing the data and skeleton Notebooks used in this project.

Thank you!