IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

Joshua Wheelhouse
21st October 2022

Outline

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

Executive Summary

Methodologies Employed

- Data Collection

Through API

Through Web Scraping

- Data Wrangling

- Exploratory Data Analysis (EDA)

EDA With SQL

EDA With Seaborn and Matlplotlib Visualisation

- Interactive Visual Analytics

With Folium

With Plotly Dash

- Predictive Modelling

Summary of Results

•EDA Results

•Folium Output

•Plotly Dash Output

•Predictive Modelling Outcome

Introduction

SpaceX are a private space faring organisation, who have pioneered re-usable rocket booster technology with the development of Falcon 9.

The use of re-usable boosters has made SpaceX an extremely affordable option for rocket launches. Reducing the cost of said launches by over half when compared to other providers – $62 million and $165 million, respectively.

SpaceX launch data, or more accurately, landing data for their re-usable rockets is freely available online and accessible through APIs and web scraping.

Therefore, if this data can be leveraged by a competitor with similar goals and technology; it may be possible to identify how to successfully land these rockets in a consistent manner.

So, what needs to be understood to be able to land these rockets successfully?

- What factors affect a successful landing?
- How do these factors paint a picture of success i.e., how do these factors interact
- How can a competitor exploit this information for a positive outcome?

Section 1

Methodology

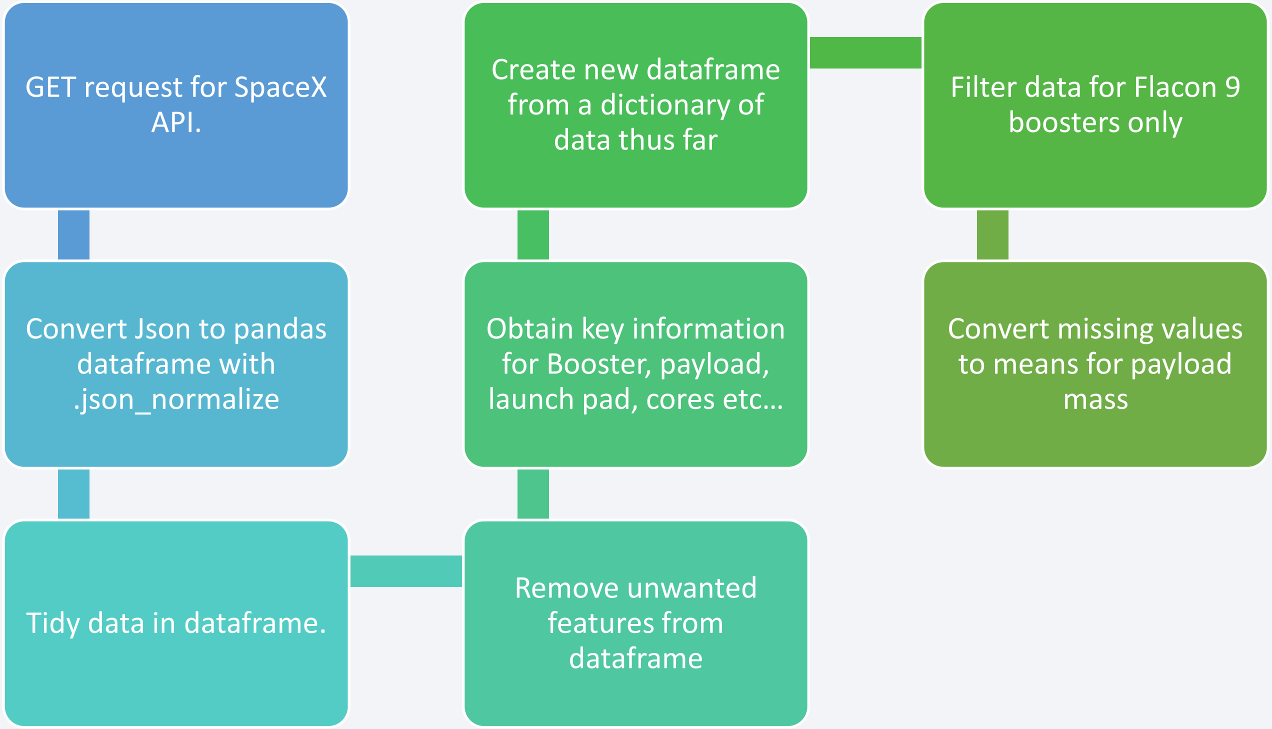
Methodology

Executive Summary

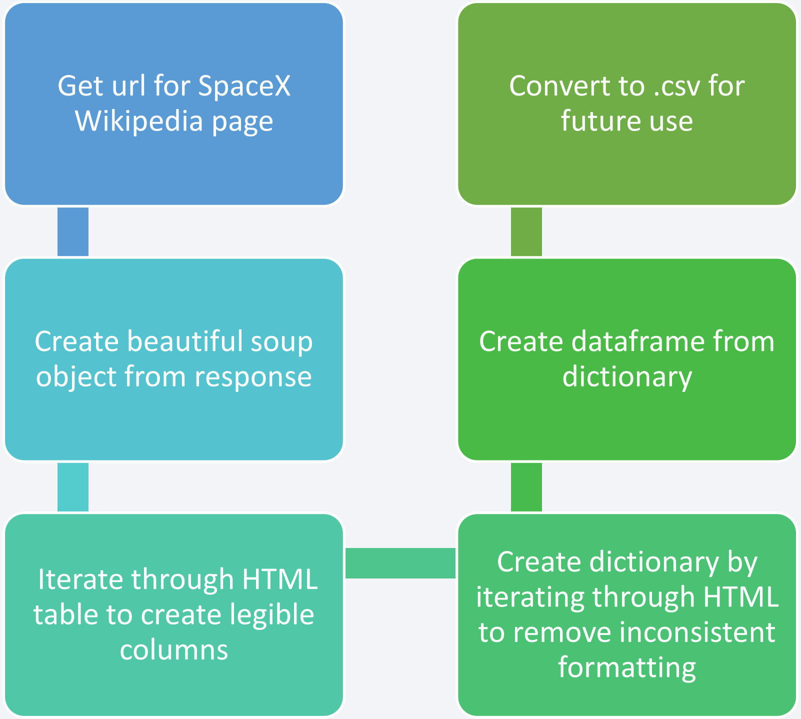
- Data collection methodology:
  - Use of python to call the SpaceX API and ‘web scrape’ Wikipedia information
- Perform data wrangling
  - Data was then standardized using one-hot encoding for landing outcomes
- Perform exploratory data analysis (EDA) using visualization and SQL
- Perform interactive visual analytics using Folium and Plotly Dash
- Perform predictive analysis using classification models
  - Multiple models were then built and assessed using scores and confusion matrices. Ultimately, the best form of modelling was identified for the landing outcome prediction

Data Collection

API Data Collection



Data collection from web scrape



Data Collection – SpaceX API

GitHub URL for Jupyter notebook:

https://github.com/Superfly634/IBMCapstone/blob/master/Space%20X%20API.ipynb

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

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

Get request

Read .json into dataframe

```
# Use json_normalize method to convert the json result into a dataframe
response = requests.get(static_json_url)
dictr = response.json()
data = pd.json_normalize(dictr)
```

Data Clean

```
# Call getLaunchSite
getLaunchSite(data)
```

```
# Call getPayloadData
getPayloadData(data)
```

```
# Call getCoreData
getCoreData(data)
```

Pass to dictionary

```
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}
```

Update dataframe

```
# Create a data from launch_dict
df = pd.DataFrame(launch_dict)
```

Filter data for falcon 9 booster & update dataframe

```
# Hint data['BoosterVersion']!='Falcon 1'
data_falcon9 = df[df.BoosterVersion !='Falcon 1']
data_falcon9.head
```

```
data_falcon9.loc[:,'FlightNumber'] = list(range(1, data_falcon9.shape[0]+1))
data_falcon9
```

Data Collection - Scraping

GitHub URL for Jupyter notebook:

[https://github.com/Superfly634/IBM Capstone/blob/master/jupyter-labs-webscraping.ipynb](https://github.com/Superfly634/IBMCapstone/blob/master/jupyter-labs-webscraping.ipynb)

```
object = requests.get(static_url)
object
```

Get wiki data and parse html with beautiful soup

```
soup = BeautifulSoup(object.content, 'html.parser')
```

```
html_tables = soup.find_all('table')
```

Extract columns from html

```
column_names = []
temp = soup.find_all('th')
for x in range(len(temp)):
    try:
     name = extract_column_from_header(temp[x])
     if (name is not None and len(name) > 0):
        column_names.append(name)
    except:
     pass
```

Create dictionary to form dataframe

```
launch_dict= dict.fromkeys(column_names)

# Remove an irrelvant column
del launch_dict['Date and time ( )']

# Let's initial the launch_dict with each
launch_dict['Flight No.'] = []
launch_dict['Launch site'] = []
launch_dict['Payload'] = []
launch_dict['Payload mass'] = []
launch_dict['Orbit'] = []
launch_dict['Customer'] = []
launch_dict['Launch outcome'] = []
# Added some new columns
launch_dict['Version Booster']=[]
launch_dict['Booster landing']=[]
launch_dict['Date']=[]
launch_dict['Time']=[]
```

Iterate through html to add the data for dataframe

```
extracted_row = 0
for table_number,table in enumerate(soup.find_all('table',"wikitable plainrowheaders collapsible")):
   for rows in table.find_all("tr"):  # get table row
      if rows.th:  #check to see if first table heading is a number corresponding to launch a number
         if rows.th.string:
            flight_number=rows.th.string.strip()
            flag=flight_number.isdigit()
      else:
         flag=False
      row=rows.find_all('td')  # #get table element
      if flag:  #if it is number save cells in a dictonary
         extracted_row += 1
         # Flight Number value
         # TODO: Append the flight_number into launch_dict with key `Flight No.`
         launch_dict['Flight No.'].append(flight_number)
         # print(flight_number)
         datatimelist=date_time(row[0])

         # Date value
         # TODO: Append the date into launch_dict with key `Date`
         date = datatimelist[0].strip(',')
         launch_dict['Date'].append(date)
         # print(date)

         # Time value
         # TODO: Append the time into launch_dict with key `Time`
         time = datatimelist[1]
         launch_dict['Time'].append(time)
         # print(time)

         # Booster version
         # TODO: Append the bv into launch_dict with key `Version Booster`
         bv=booster_version(row[1])
         if not(bv):
            bv=row[1].a.string
         launch_dict['Version Booster'].append(bv)
         # print(bv)

         # Launch Site
         # TODO: Append the bv into launch_dict with key `Launch Site`
         launch_site = row[2].a.string
         launch_dict['Launch site'].append(launch_site)
         # print(launch_site)
```

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

Create dataframe

Static_url defined as:
https://en.wikipedia.org/w/index.php?title=List_of_Falcon_9_and_Falcon_Heavy_launches&oldid=1027686922

Data Wrangling

Data collated, and then assigned keys to assess whether a 'good' or 'bad' landing had occurred.

Data then used to calculate success rate of launches.

GitHub URL for Jupyter notebook:

https://github.com/Superfly634/IBMCapstone/blob/master/Data%20Wrangling.ipynb

```
df['LaunchSite'].value_counts()

CCAFS SLC 40    55
KSC LC 39A      22
VAFB SLC 4E     13
Name: LaunchSite, dtype: int64
```

Summation of launches by launch site

```
df['Orbit'].value_counts()

GTO      27
ISS      21
VLEO     14
PO        9
LEO       7
SSO       5
MEO       3
ES-L1     1
HEO       1
SO        1
GEO       1
Name: Orbit, dtype: int64
```

Summation of launches by orbit

```
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
```

Types of landing outcome

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

Encoding of successful landing (1) and unsuccessful landing (0)

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

0.6666666666666666
```

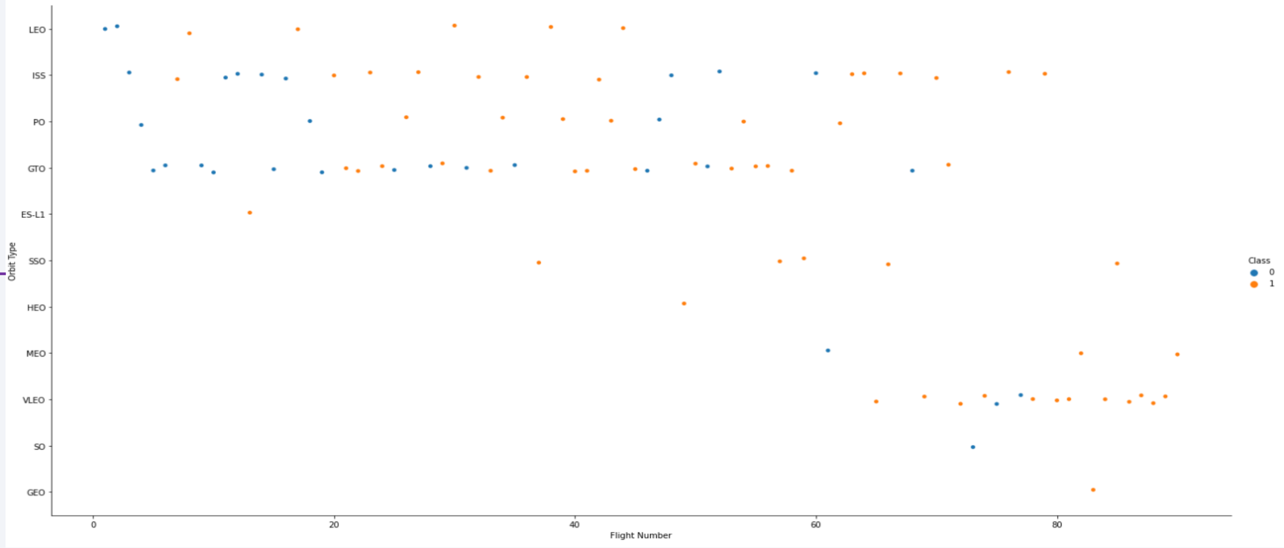
Success rate

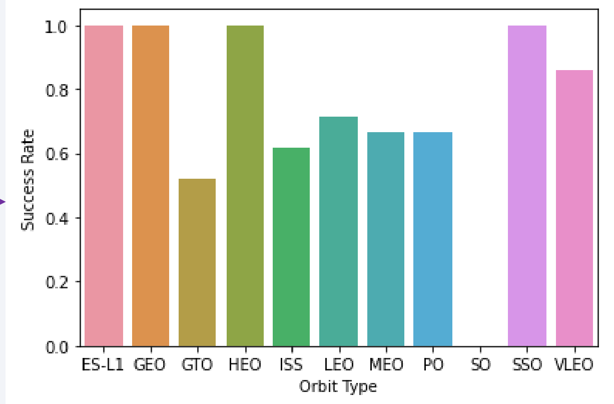
EDA with Data Visualization

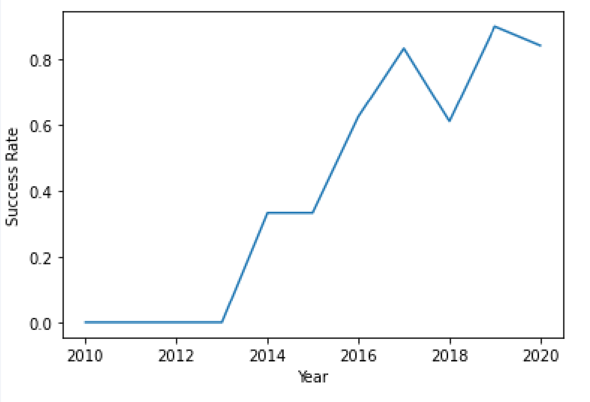
Many charts were plotted to visualise the success of launces by SpaceX. These included scatter plots as initial indicators, and then the use of bar chart and line plot to obtain some key insights.

GitHub URL for Jupyter notebook:

https://github.com/Superfly634/IBM Capstone/blob/master/EDA%20with %20vis.ipynb







EDA with SQL

SQL Used to visualise the following info:

1. Unique launch sites
```
%%sql
select distinct Launch_Site from spacex
```
2. Launch sites based on string
```
%%sql
select * from spacex where Launch_Site like 'KSC%' limit 5
```
3. Total payload mass used by Nasa
```
%%sql
select sum(PAYLOAD_MASS__KG_) from spacex where CUSTOMER='NASA (CRS)'
```
4. Total payload for F9 v1.1 Booster
```
%%sql
select avg(PAYLOAD_MASS__KG_) from spacex where BOOSTER_VERSION='F9 v1.1'
```
5. Date of first successful landing by drone ship
```
%%sql
Select min(DATE) from spacex where LANDING__OUTCOME = 'Success (drone ship)'
```
6. Successful groundpad landings, 4000 – 6000kg
```
%%sql
Select BOOSTER_VERSION from spacex where LANDING__OUTCOME = 'Success (ground pad)' and PAYLOAD_MASS__KG_ between 4000 and 6000
```
7. Total number of successes and failures
```
%%sql
Select MISSION_OUTCOME, Count(MISSION_OUTCOME) as outcomes from spacex group by MISSION_OUTCOME
```
8. Booster versions which carried the maximum payload
```
%%sql
Select BOOSTER_VERSION from spacex where PAYLOAD_MASS__KG_ = (select max(PAYLOAD_MASS__KG_ ) from spacex)
```
9. Successful groundpad launches, booster version & launch site
```
%%sql
select month(date), LANDING__OUTCOME, BOOSTER_VERSION, LAUNCH_SITE from spacex where LANDING__OUTCOME = 'Success (ground pad)'
```
10. Ranking landing outcomes 2016 – 2017
```
%%sql
select LANDING__OUTCOME, COUNT(*) as outcome from spacex where DATE between '2010-06-04' and '2017-02-20' group by LANDING__OUTCOME order by outcome desc;
```

GitHub URL for Jupyter notebook:

https://github.com/Superfly634/IBMCapstone/blob/master/EDA%20with%20SQL%20(1).ipynb

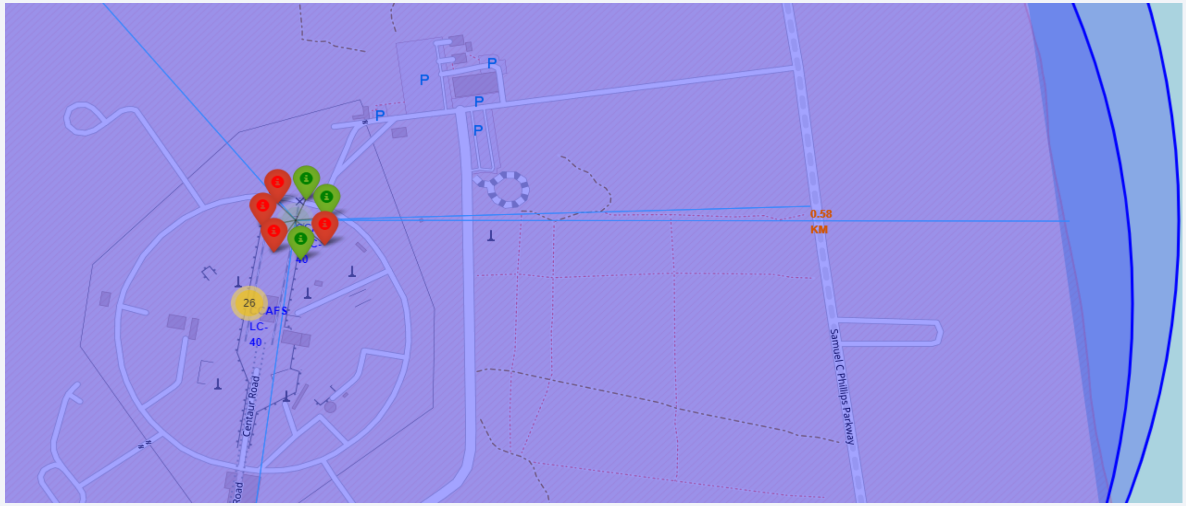
Build an Interactive Map with Folium

Interactive folium maps were generated to visualise the SpaceX launches

Circles were used to outline where the launch sites can be found in the U.S.A.

Markers were then used to indicate, within their respective launch site, which launches were successful and unsuccessful. Giving an easy indication of high-performing launchpads.

Lines were then drawn on the map to show the distance of such things as coastlines and infrastructure were from the launch sites. To know if they were a suitable distance away.



GitHub URL for Jupyter notebook:

https://github.com/Superfly634/IBM Capstone/blob/master/fin_%20lab_jupyter_launch_site_location.ipynb

Build a Dashboard with Plotly Dash

Interactive charts were created within a Plotly Dash app.

A pie chart was created to demonstrate the success rate of all the launch sites. This allows for a quick and intuitive way to understand which site(s) have the greatest success rate, and conversely perform the worst.

A scatter graph was also created showing success rate (‘Class’), for payload mass and booster version; where-by the payload mass has an interactive slider, which automatically updates the graph. This allows for an understanding of how payload mass effects launch success, and how each booster version performs at different payload masses.

GitHub URL for Jupyter notebook:

https://github.com/Superfly634/IBMCapstone/blob/master/Plotly%20Dash%20App%20code.ipynb

Predictive Analysis (Classification)

4 Models were created:

Logistic Regression
Support Vector Machine (SVM)
Decision Tree
K-Nearest Neighbour (KNN)

Each model was trained on a numpy array (Y) of landing success vs. The factors which affect the landing success such as payload mass and flight number (X). X was standardised using a scalar function from scikitlearn.

The data was then split into test and training sets, which then allow for the tuning of the models to produce the best accuracy based upon the best parameters (.best_param)

GitHub URL for Jupyter notebook:

https://github.com/Superfly634/IBMCapstone/blob/master/SpaceX_Machine%20Learning%20Prediction_Part_5.ipynb

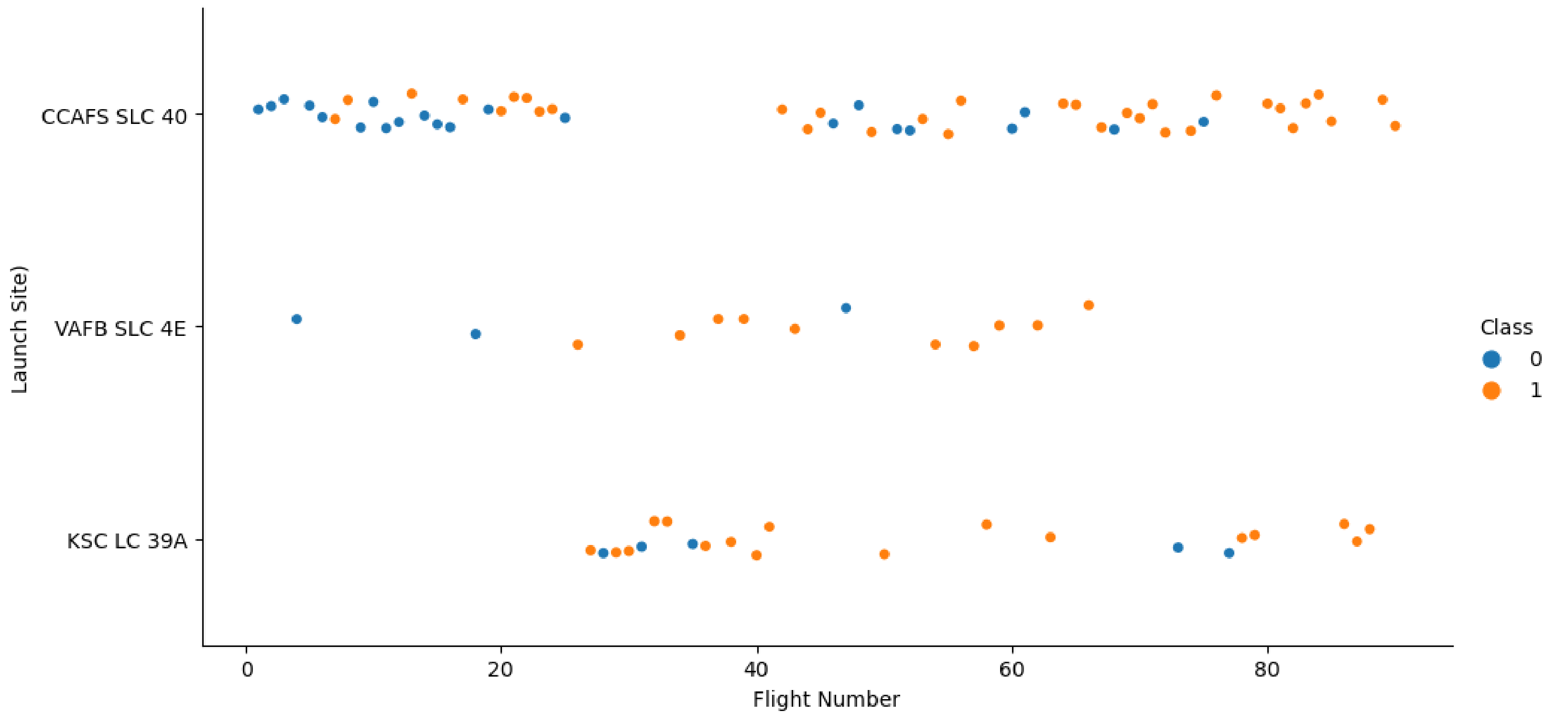
Results

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

Section 2

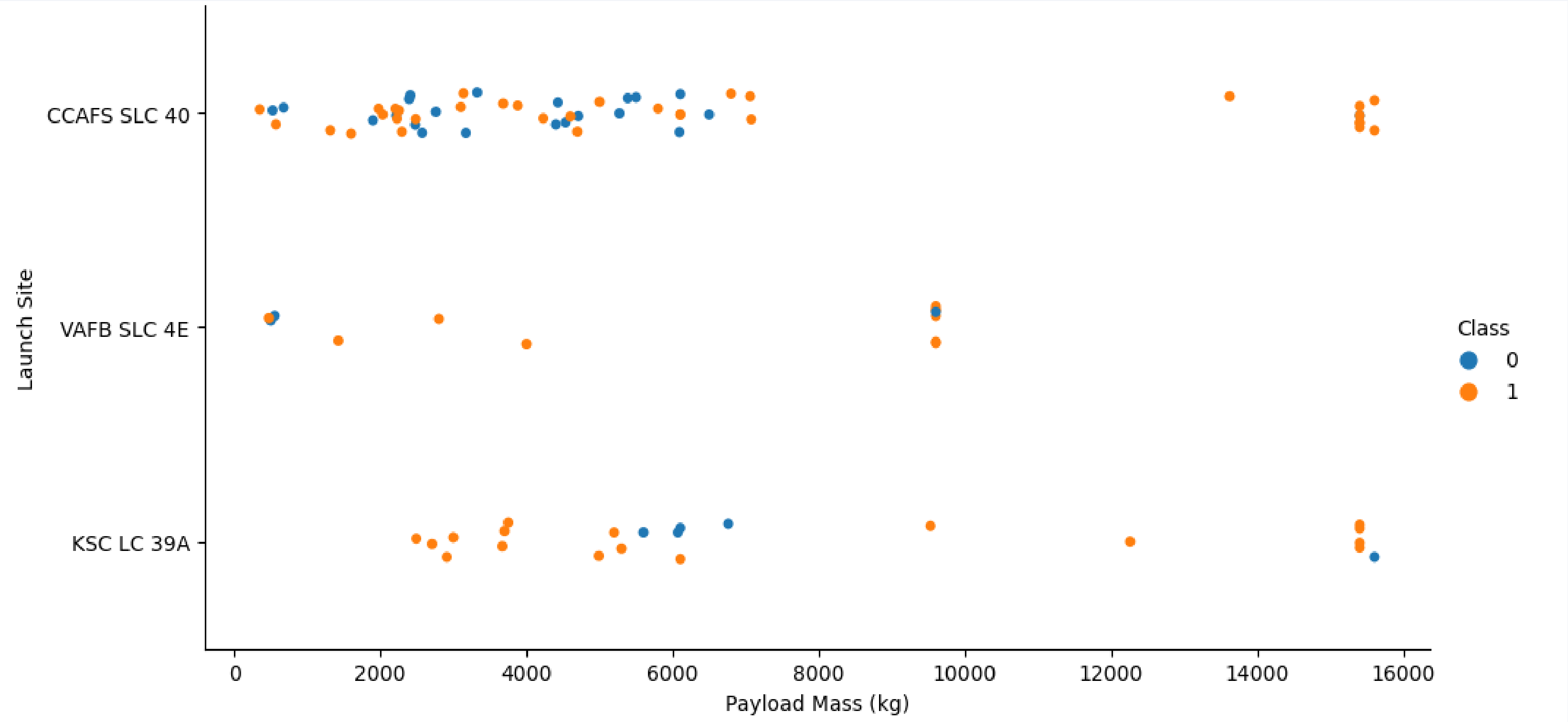
Insights drawn from EDA

Flight Number vs. Launch Site



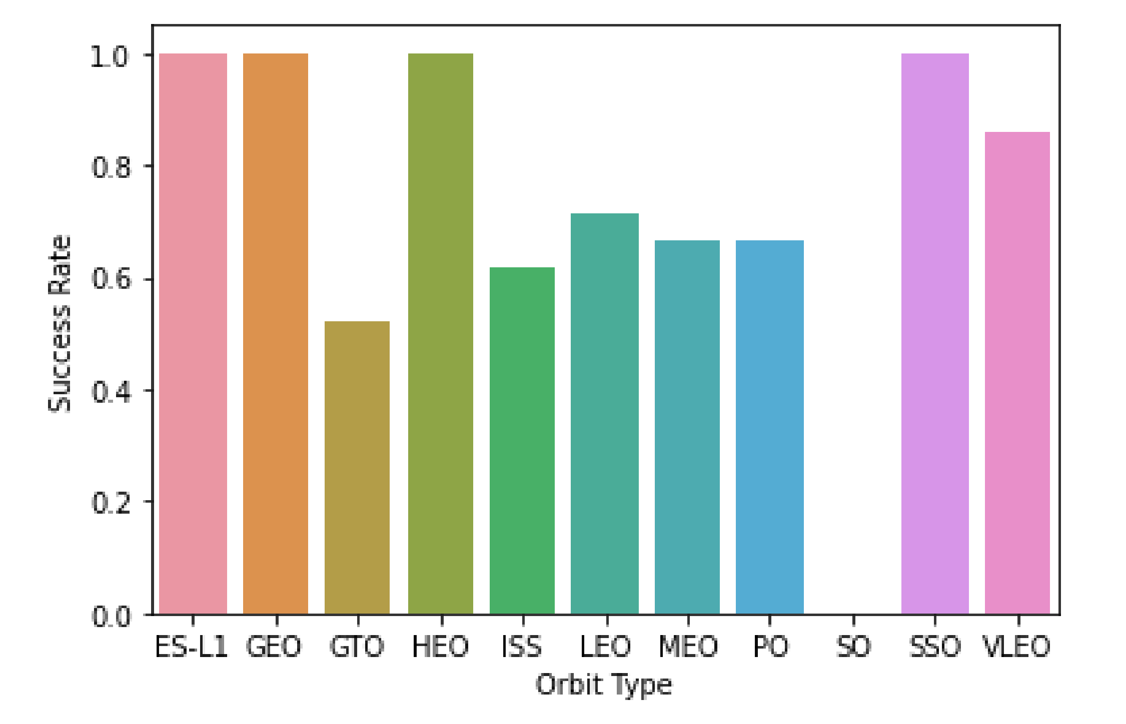
Here we see a general trend of success increasing with flight number, showing improved experience. We also see that 'CCAFS SLC 40' is the preferred launch site with the most launches, with 'KSC LC 39A' and 'VAFB SLC 4E' being second and third respectively.

Payload vs. Launch Site



Here we see a higher success rate for larger payloads, regardless of launch site. However, the sample size is small. CCAFS SLC 40 has the most samples, showing a mix of success and failure at lower payloads. KSC LC 39A seems to perform particularly well across the board, with its worst success rate at around ~6000kg of payload.

Success Rate vs. Orbit Type



The orbit-types with the greatest success rate are:

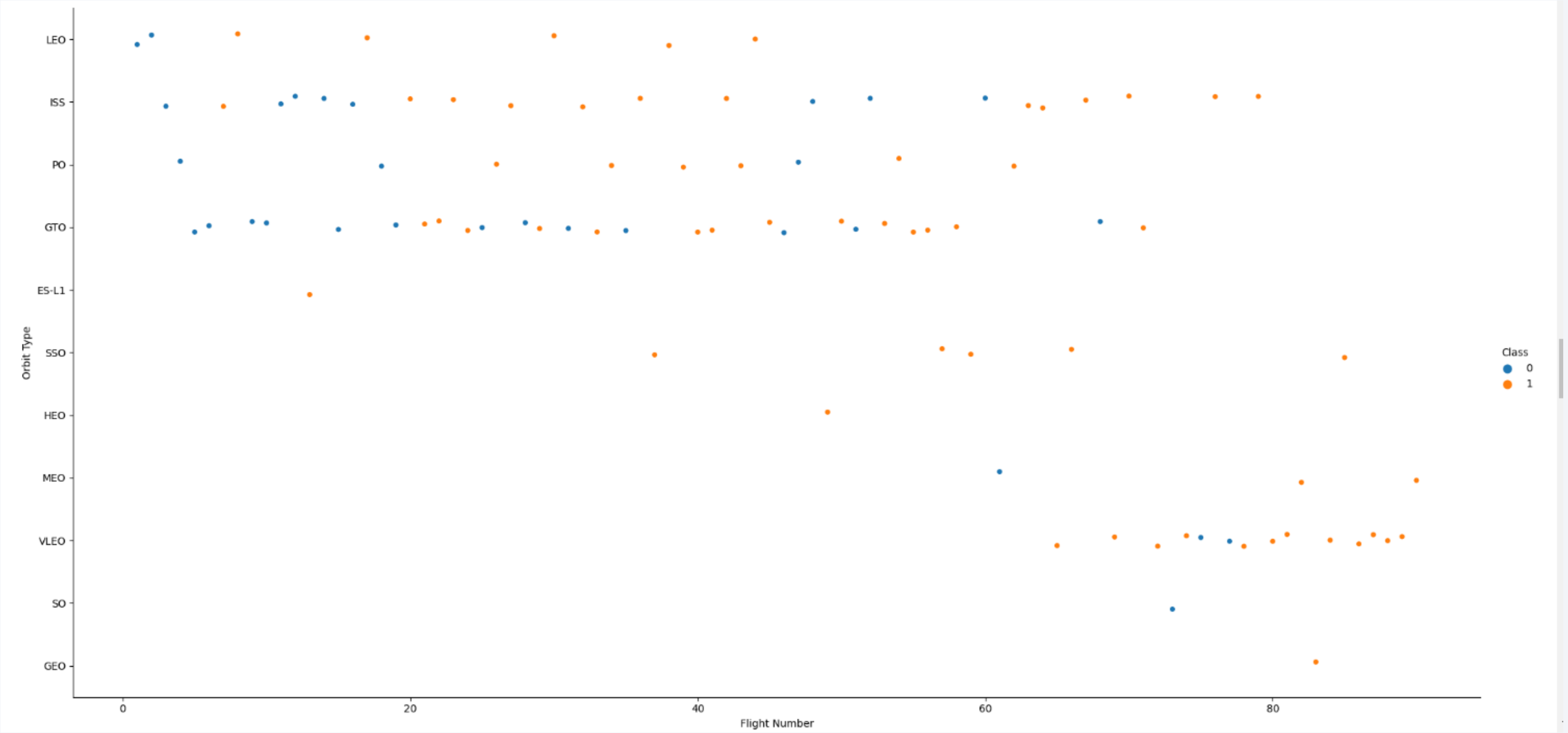
ES-L1, GEO, HEO, SSO

However, GEO and HEO only have one launch each. Which is not a representative data set.

SSO is the most successful orbit type with more than one launch

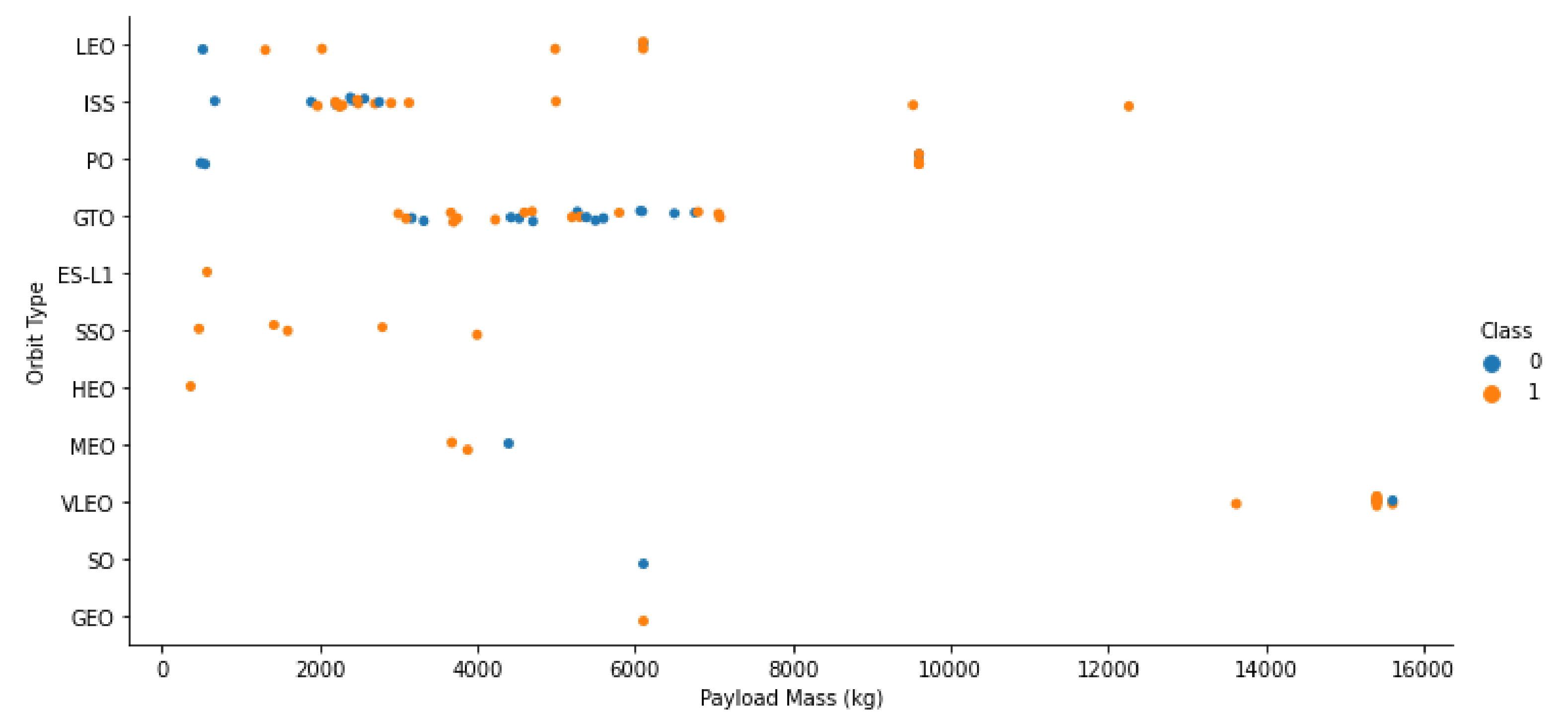
GTO also has multiple launch attempts and is evidently the most poorly performing launch site.

Flight Number vs. Orbit Type



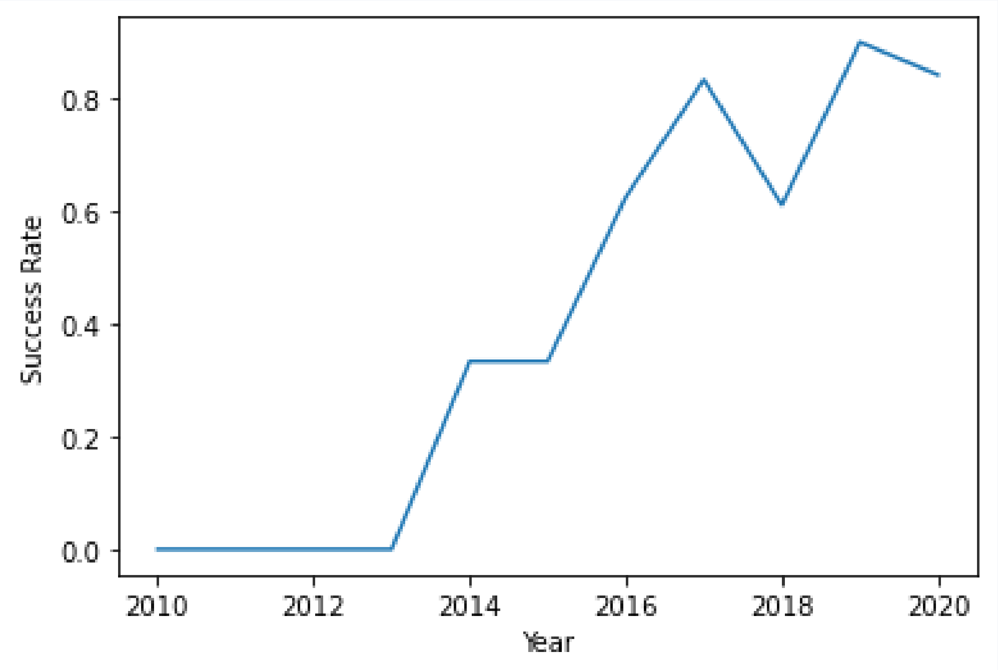
Much like the launch site data, we see increased success over time for all the orbit types. We also see a preference form for the VLEO orbit type – Which is a very low earth orbit.

Payload vs. Orbit Type



There are much more launches across the orbit types for lower payload masses. However, the higher payloads seem to be more successful across the orbit types. This data also ratifies the SSO orbit to be the most successful but has not been tested for heavier payloads.

Launch Success Yearly Trend



Here, we see one of the strongest indicators for success which is time, otherwise classified as experience.

All Launch Site Names

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

```
%%sql
select distinct Launch_Site from spacex
```

Although there are many rows of data within the SQL database, all with an entry for ‘launch_site’, there are only 4 unique launch sites.

Launch Site Names Begin with 'KSC'

DATE	time__utc_	booster_version	launch_site	payload	payload_mass__kg_	orbit	customer	mission_outcome	landing__outcome
2017-02-19	14:39:00	F9 FT B1031.1	KSC LC-39A	SpaceX CRS-10	2490	LEO (ISS)	NASA (CRS)	Success	Success (ground pad)
2017-03-16	06:00:00	F9 FT B1030	KSC LC-39A	EchoStar 23	5600	GTO	EchoStar	Success	No attempt
2017-03-30	22:27:00	F9 FT B1021.2	KSC LC-39A	SES-10	5300	GTO	SES	Success	Success (drone ship)
2017-05-01	11:15:00	F9 FT B1032.1	KSC LC-39A	NROL-76	5300	LEO	NRO	Success	Success (ground pad)
2017-05-15	23:21:00	F9 FT B1034	KSC LC-39A	Inmarsat-5 F4	6070	GTO	Inmarsat	Success	No attempt

```
%%sql
select * from spacex where Launch_Site like 'KSC%' limit 5
```

A limit of 5 responses for site names beginning with 'KSC'

Total Payload Mass

1
45596

Total payload carried for launches contracted for Nasa

```
%%sql
select sum(PAYLOAD_MASS__KG_) from spacex where CUSTOMER='NASA (CRS)'
```

Average Payload Mass by F9 v1.1

1
2928

Total payload carried F9 v1.1 Booster

```
%%sql
select avg(PAYLOAD_MASS__KG_) from spacex where BOOSTER_VERSION='F9 v1.1'
```

First Successful Ground Landing Date

1
2016-04-08

First successful landing on a drone ship

```
%%sql
Select min(DATE) from spacex where LANDING__OUTCOME = 'Success (drone ship)'
```

Successful Ground Pad Landing with Payload between 4000 and 6000

booster_version
F9 FT B1032.1
F9 B4 B1040.1
F9 B4 B1043.1

Booster versions with successful ground pad landing for payload mass between 4000 - 6000

```
%%sql
Select BOOSTER_VERSION from spacex where LANDING__OUTCOME = 'Success (ground pad)' and PAYLOAD_MASS__KG_ between 4000 and 6000
```

Total Number of Successful and Failure Mission Outcomes

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

Count of all mission outcomes

```
%%sql
Select MISSION_OUTCOME, Count(MISSION_OUTCOME) as outcomes from spacex group by MISSION_OUTCOME
```

Boosters Carried Maximum Payload

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

Booster versions which have carried the maximum payload mass

```
%%sql
Select BOOSTER_VERSION from spacex where PAYLOAD_MASS__KG_ = (select max(PAYLOAD_MASS__KG_ ) from spacex)
```

2017 Launch Records

1	landing__outcome	booster_version	launch_site
12	Success (ground pad)	F9 FT B1019	CCAFS LC-40
7	Success (ground pad)	F9 FT B1025.1	CCAFS LC-40
2	Success (ground pad)	F9 FT B1031.1	KSC LC-39A
5	Success (ground pad)	F9 FT B1032.1	KSC LC-39A
6	Success (ground pad)	F9 FT B1035.1	KSC LC-39A
8	Success (ground pad)	F9 B4 B1039.1	KSC LC-39A
9	Success (ground pad)	F9 B4 B1040.1	KSC LC-39A
12	Success (ground pad)	F9 FT B1035.2	CCAFS SLC-40
1	Success (ground pad)	F9 B4 B1043.1	CCAFS SLC-40

Landing outcome, booster version and launch site with successful ground pad launch in 2017

```
%%sql
select month(date), LANDING__OUTCOME, BOOSTER_VERSION, LAUNCH_SITE from spacex where LANDING__OUTCOME = 'Success (ground pad)'
```

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

landing__outcome	outcome
No attempt	9
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

Landing outcomes during 04-06-2010 and 20-03-2017

```
%%sql
select LANDING__OUTCOME, COUNT(*) as outcome from spacex where DATE between '2010-06-04' and '2017-02-20' group by LANDING__OUTCOME order by outcome desc;
```

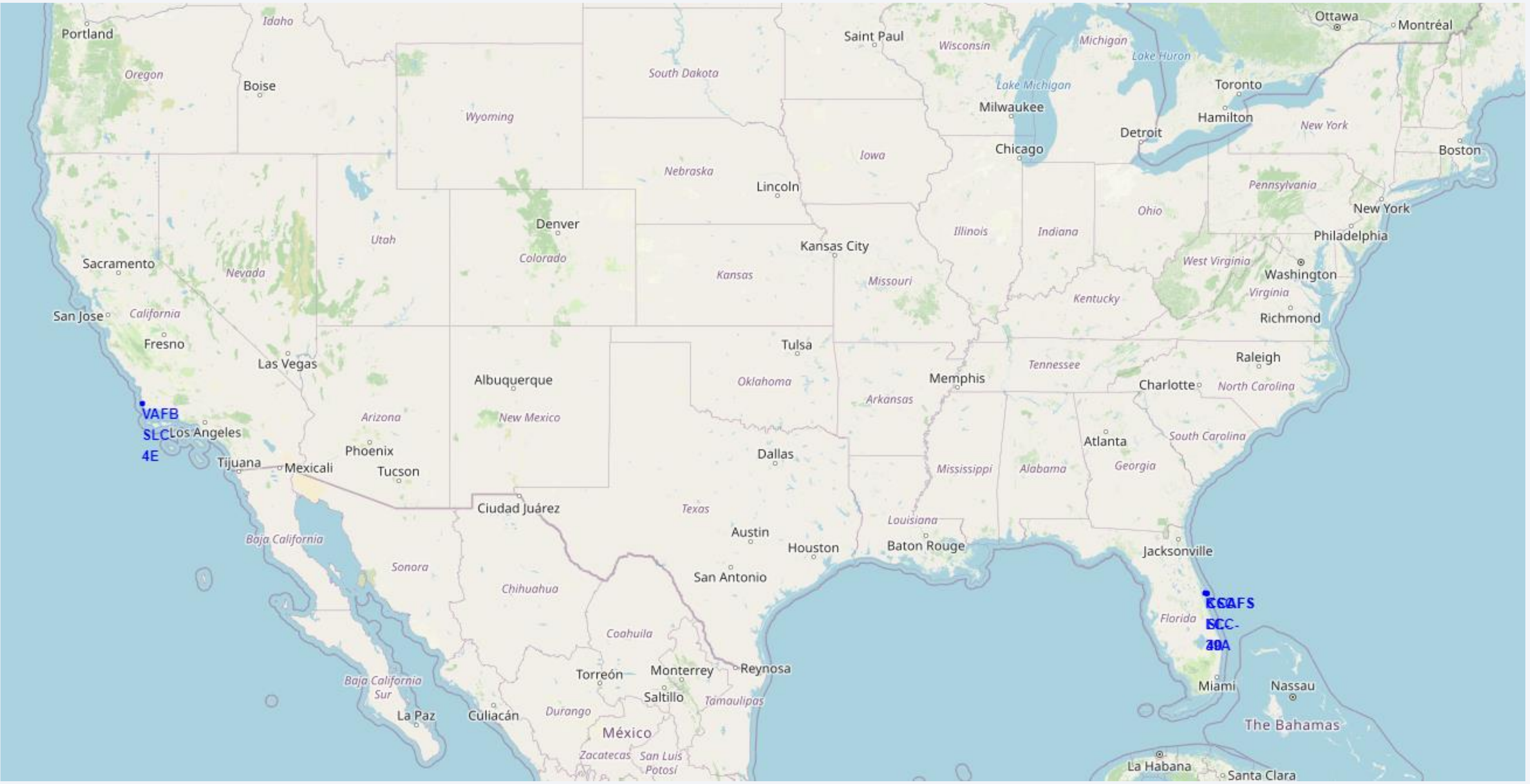
Section 3

Launch Sites Proximities Analysis

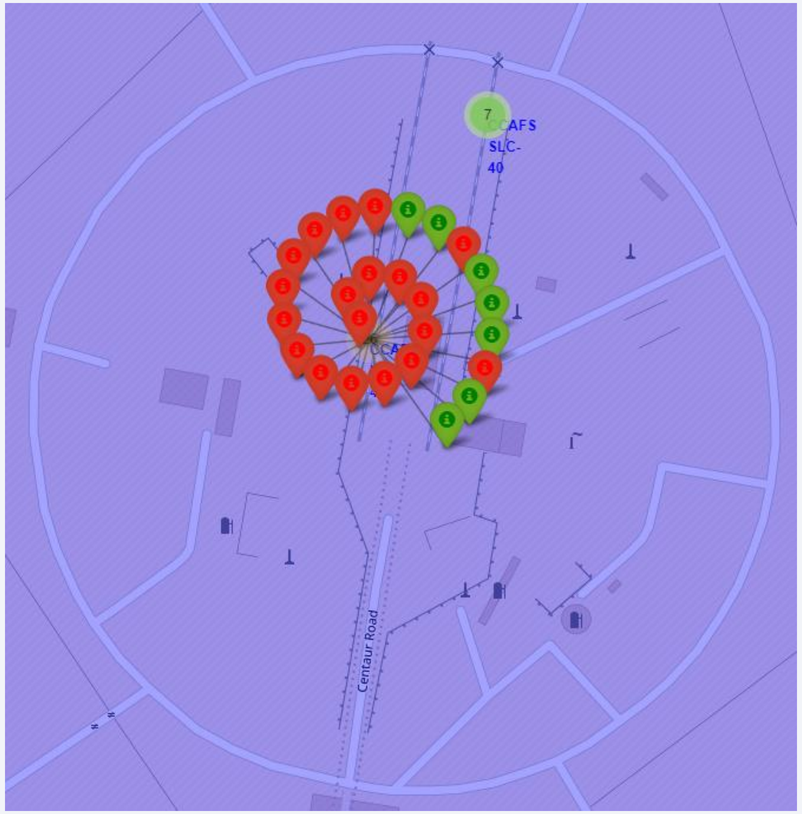
SpaceX Launch Sites

SpaceX launch sites are wholly based within the United states. Three on the east coast, at the same location at Nasa's cape Canaveral. And one at the Vandenberg air force base on the west coast.

CCAFS SLC-40 & CCAFS LC-40 launch from the same pad.



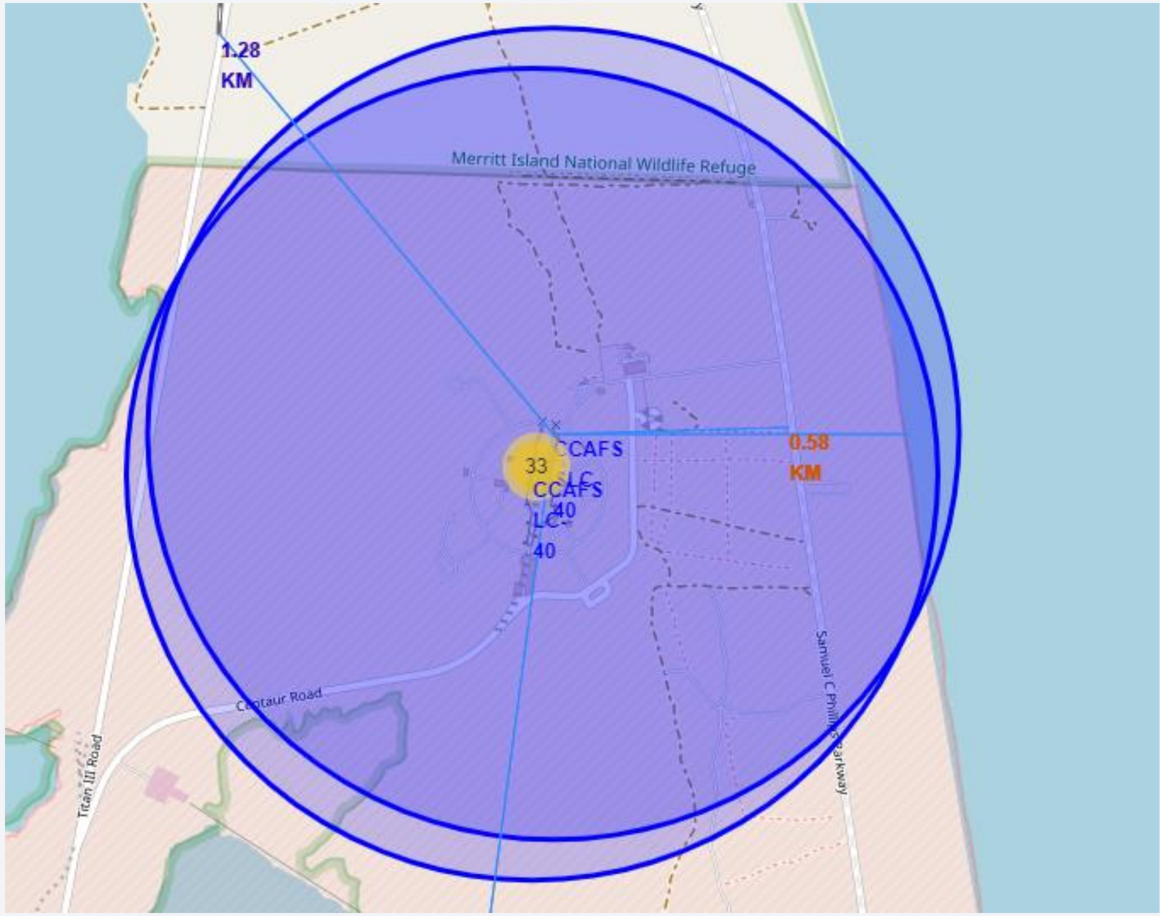
CCAFS LC-40 Launch Outcome Markers



An example of launch outcome, coloured for success (green) and failure (red).

This is for CCAFS LC-40.

Distance to infrastructure from CCAFS SLC-40

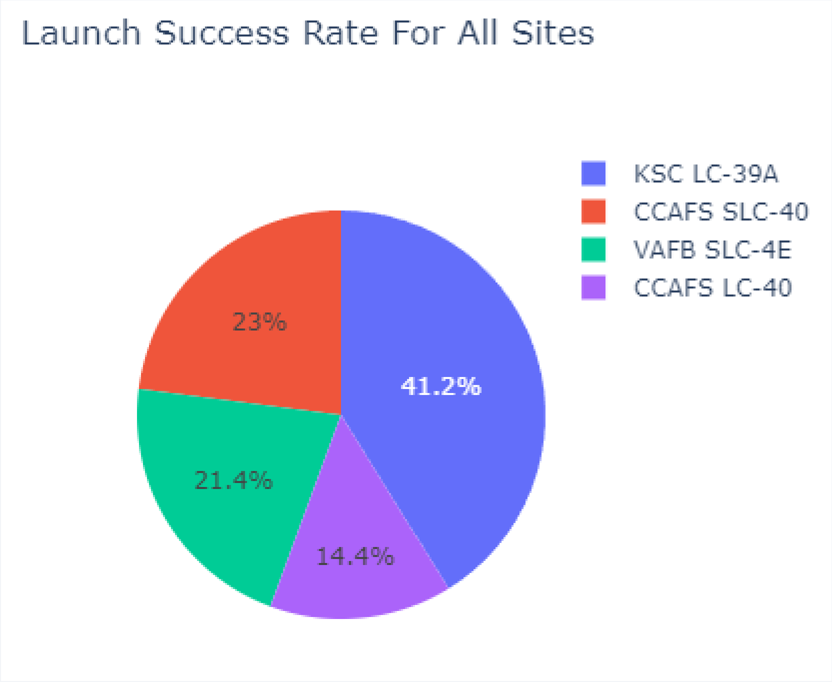


An example of lines with distance from a particular launch site to infrastructure such as train line – Top left, 1.28km.

Section 4

Build a Dashboard with Plotly Dash

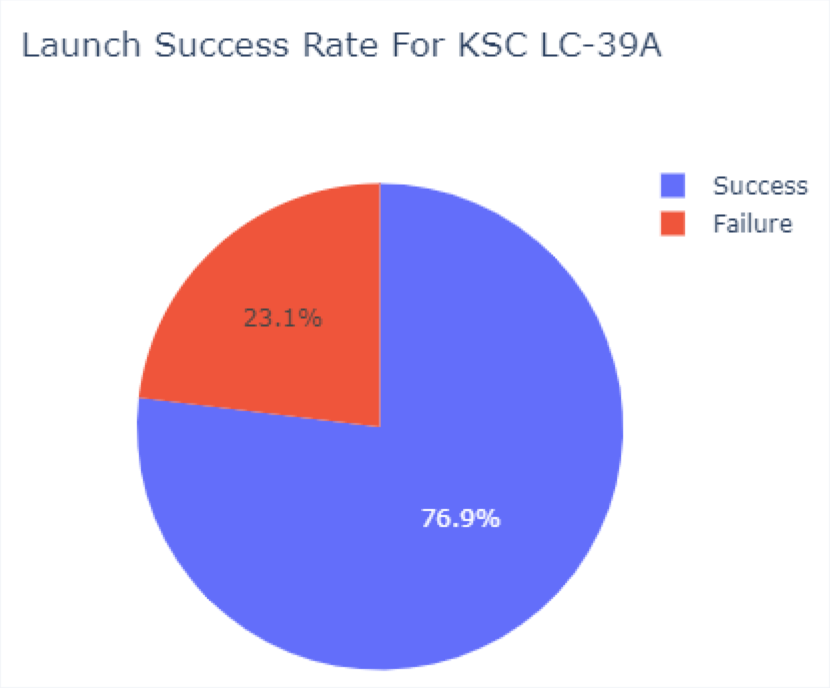
Plotly Dashboard – Success Rate by Launch Site



Pie chart of launch site by success rate.

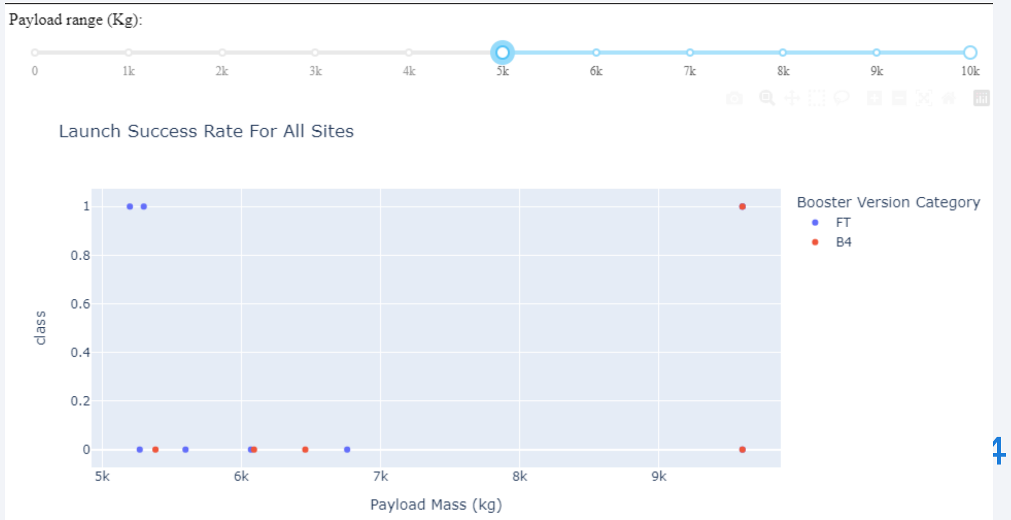
KSC LC-39A Being the most successful at 41.2%

Plotly Dashboard – Most Successful Launch Site



Pie chart breakdown for the success rate of the most successful launch site

Plotly Dashboard – Booster and Success by Payload



Here we see most of the launches happening at the lower payloads, with most the launches conducted with the 'FT' booster version.

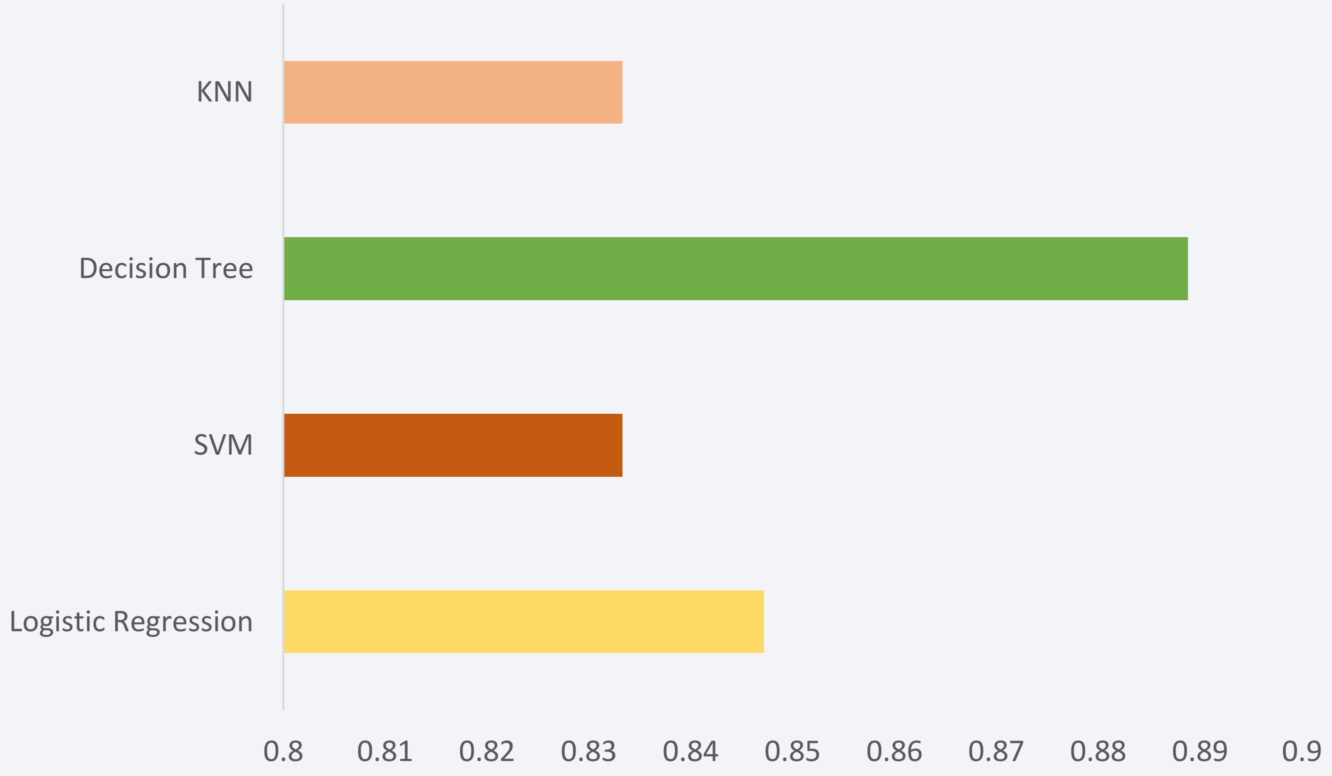
Only two boosters launch with the heavier payloads, and with not much success.

The general trend is also one of unsuccessful launches

Section 5

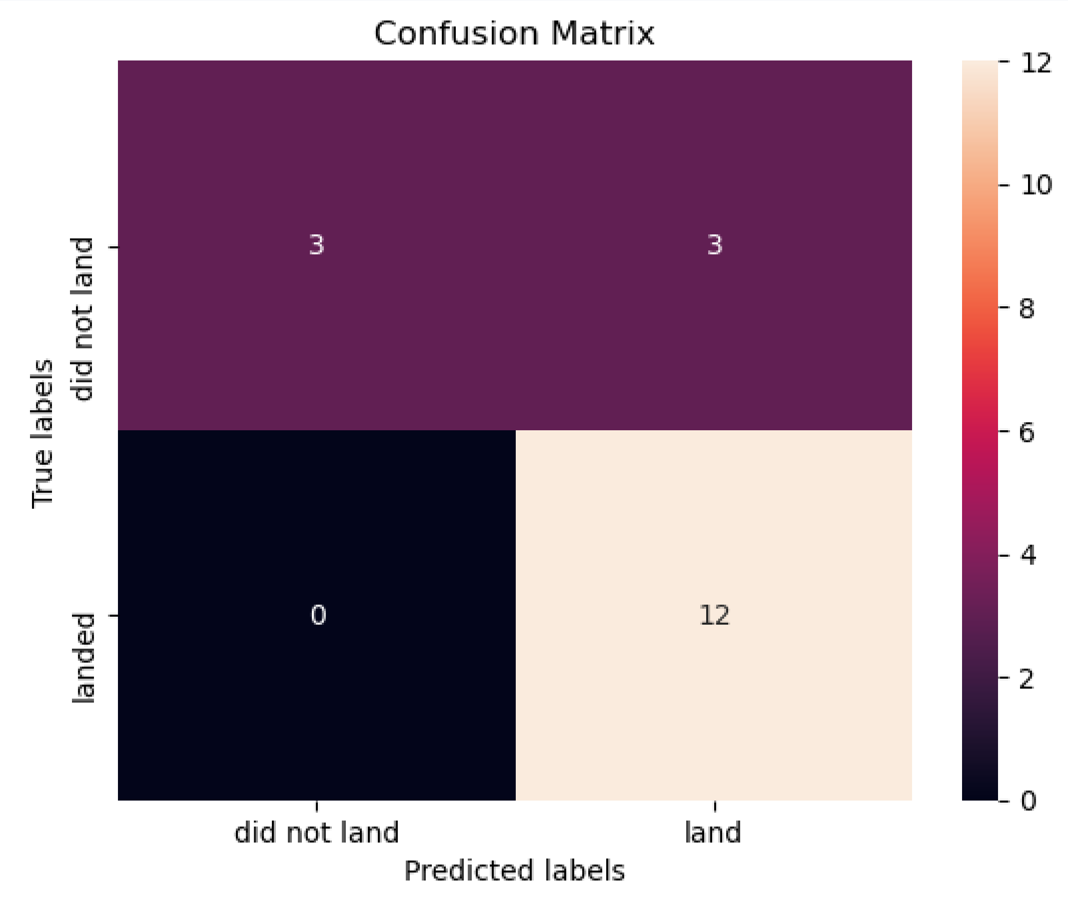
Predictive Analysis (Classification)

Classification Accuracy



The most accurate model for the prediction of launch success is the 'Decision Tree' at 0.8888.

Confusion Matrix



Only 3 predictions from the test set were miss-identified as ‘land’ for a true result of ‘did not land’

Conclusions

- It is possible to predict whether a launch will be successful, with a relatively high degree of accuracy – Decision Tree Modelling

Factors contributing towards a successful landing are:

- Launch Site
- Orbit type
- Payload & Booster version

However, one of the leading indicators which is not immediately useful to the competitor SpaceY, is that of experience as launch success drastically increased over time with only failures observed in the first three years of operation

Thank you!