IBM Data Science Capstone Project

Ali Haider Qureshi

12 December 2021

GitHub URL

Outline

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

GitHub URL

EXECUTIVE SUMMARY

Summary of methodologies

- Data collection
- Data wrangling
- Exploratory data analysis (EDA)
- Interactive map with folium
- Dashboard with Plotly Dash
- Predictive analysis with machine learning models

Summary of results

- Exploratory data analysis results
- Interactive visualization
- Best model for predictive analysis

GitHub URL

Introduction

We will predict if the Falcon 9 first stage will land 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.

- Can we predict if a rocket will land successfully?

- What are the variables that influence a successful landing of the first stage?

- Therefore what condition offer the best results?

GitHub URL

Methodology

- Data collection
  - SpaceX REST API
  - Web scrapping (Wikipedia)
- EDA with data visualization and SQL
  - Graphs to show the correlation between between different variables
- Visually interactive analysis with Folium and Plotly Dash (Dashboard)
- Data Wrangling
  - One Hot Encoding for Machine Learning
- Predictive analysis with 3 different ML models
  - Logistic Regression
  - SVM
  - Decision tree classifier

GitHub URL

Methodology

Data Collection – SpaceX

Notebook Link

Collecting data from the SpaceX REST API containing all the information we need about their rocket launches, like date, rocket type, payload, launch specifications, landing specification etc.

Alternatively we can also use web scrapping to get the needed data, for example for this project we gathered data from Wikipedia and we filtered the data to contain only information about Falcon 9.

DataFrame

	FlightNumber	Date	BoosterVersion	PayloadMass	Orbit	LaunchSite	Outcome	Flights	GridFins	Reused	Legs	LandingPad	Block	ReusedCount	Serial	Longitude	Latitude
4	1	2010-06-04	Falcon 9	NaN	LEO	CCSFS SLC 40	None None	1	False	False	False	None	1.0	0	B0003	-80.577366	28.561857
5	2	2012-05-22	Falcon 9	525.0	LEO	CCSFS SLC 40	None None	1	False	False	False	None	1.0	0	B0005	-80.577366	28.561857
6	3	2013-03-01	Falcon 9	677.0	ISS	CCSFS SLC 40	None None	1	False	False	False	None	1.0	0	B0007	-80.577366	28.561857
7	4	2013-09-29	Falcon 9	500.0	PO	VAFB SLC 4E	False Ocean	1	False	False	False	None	1.0	0	B1003	-120.610829	34.632093
8	5	2013-12-03	Falcon 9	3170.0	GTO	CCSFS SLC 40	None None	1	False	False	False	None	1.0	0	B1004	-80.577366	28.561857

Data Collection – SpaceX REST API

[Notebook Link]

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

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

response = requests.get(spacex_url)
```

requesting rocket launch data from SpaceX API → decode the response content as a Json using `.json()` and turn it into a Pandas dataframe using `.json_normalize()` → Custom function to retrieve specific data from API → construct our dataset using the data we have obtained.

```
# Use json_normalize meethod to convert the json result into a dataframe
data = pd.json_normalize(response.json())
```

```
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}
```

This is what our data looks like after the collection process and some cleaning to make it more usable

	FlightNumber	Date	BoosterVersion	PayloadMass	Orbit	LaunchSite	Outcome	Flights	GridFins	Reused	Legs	LandingPad	Block	ReusedCount	Serial	Longitude	Latitude
4	1	2010-06-04	Falcon 9	NaN	LEO	CCSFS SLC 40	None None	1	False	False	False	None	1.0	0	B0003	-80.577366	28.561857
5	2	2012-05-22	Falcon 9	525.0	LEO	CCSFS SLC 40	None None	1	False	False	False	None	1.0	0	B0005	-80.577366	28.561857
6	3	2013-03-01	Falcon 9	677.0	ISS	CCSFS SLC 40	None None	1	False	False	False	None	1.0	0	B0007	-80.577366	28.561857
7	4	2013-09-29	Falcon 9	500.0	PO	VAFB SLC 4E	False Ocean	1	False	False	False	None	1.0	0	B1003	-120.610829	34.632093
8	5	2013-12-03	Falcon 9	3170.0	GTO	CCSFS SLC 40	None None	1	False	False	False	None	1.0	0	B1004	-80.577366	28.561857

Data Collection – Web Scraping

GET method to request the Falcon9 Launches

```
static_url = "https://en.wikipedia.org/w/index.php?title=List_of_Falcon_9_and_Falcon_Heavy_launches&oldid=1027686922"
requests.get(static_url)
response = requests.get(static_url)
```

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

```
soup = BeautifulSoup(response.text, 'html.parser')
```

find all tables on the wiki page

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

extract column name one by one

```
column_names = []


first_launch_table.find_all('th')

for row in first_launch_table('th'):
      name = extract_column_from_header(row)
      if name is not None and len(name) > 0:
                  column_names.append(name)
```

create an empty dictionary with keys from the extracted column names

```
launch_dict= dict.fromkeys(column_names)

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

# Let's initial the launch_dict with each value to be an empty list
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']=[]
```

After you fill in the parsed launch record values into `launch_dict`, we can create a dataframe from it.

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

Notebook Link

	FlightNumber	Date	BoosterVersion	PayloadMass	Orbit	LaunchSite	Outcome	Flights	GridFins	Reused	Legs	LandingPad	Block	ReusedCount	Serial	Longitude	Latitude
4	1	2010-06-04	Falcon 9	NaN	LEO	CCSFS SLC 40	None None	1	False	False	False	None	1.0	0	B0003	-80.577366	28.561857
5	2	2012-05-22	Falcon 9	525.0	LEO	CCSFS SLC 40	None None	1	False	False	False	None	1.0	0	B0005	-80.577366	28.561857
6	3	2013-03-01	Falcon 9	677.0	ISS	CCSFS SLC 40	None None	1	False	False	False	None	1.0	0	B0007	-80.577366	28.561857
7	4	2013-09-29	Falcon 9	500.0	PO	VAFB SLC 4E	False Ocean	1	False	False	False	None	1.0	0	B1003	-120.610829	34.632093
8	5	2013-12-03	Falcon 9	3170.0	GTO	CCSFS SLC 40	None None	1	False	False	False	None	1.0	0	B1004	-80.577366	28.561857

Data Wrangling

- In the data set, there are several different cases where the booster did not land successfully. Sometimes a landing was attempted but failed due to an accident; we will mainly convert those outcomes into Training Labels with `1` means the booster successfully landed `0` means it was unsuccessful.
- Some of the rows are missing values in our dataset, to deal with this problem we calculate the mean for the column with the missing values using `.mean()`. Then use the mean and the `.replace()` function to replace `np.nan` values in the data with the mean you calculated.
- Identify and calculate the percentage of the missing values in each attribute
- Calculate the number of launches on each site
- Calculate the number and occurrence of each orbit
- Calculate the number and occurrence of mission outcome per orbit type
- Create a landing outcome label from Outcome column

Notebook Link

EDA with Data Visualization

Notebook Link

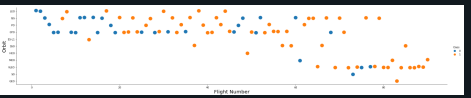
Scatterplot drawn:

1) `Flight Number` vs `Launch Site`
2) Flight Number and Orbit type
3) Payload Vs. Launch Site
4) `Flight Number` vs. `Payload Mass`
5) Payload vs. Orbit

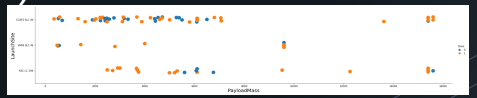
Scatter plots are used to observe relationships between variables, helping us in finding patterns and meaningful factors

1)

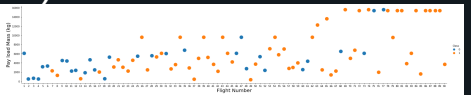
2)



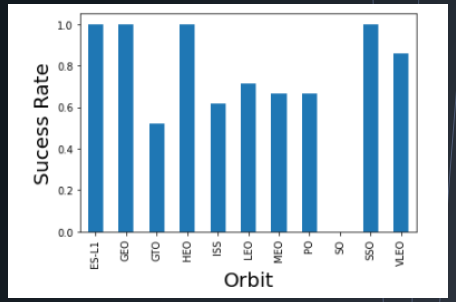
3)



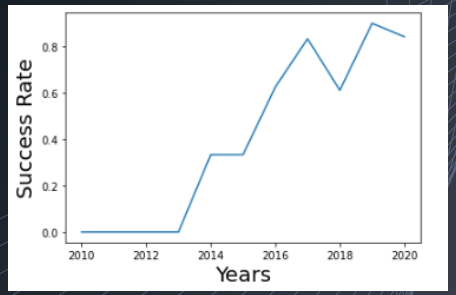
4)



5)



A bar chart or bar graph is a chart or graph that presents categorical data. Helpful in finding which orbit has the highest success.



Line chart is a graphical representation of an attribute's value change through time. It shows trends and meaningful changes with time.

EDA with SQL

Performed SQL quaries to gain further insight on the data

1) Names of the unique launch sites in the space mission

2) Total payload mass carried by boosters launched by NASA (CRS)

3) Average payload mass carried by booster version F9 v1.1

4) Total number of successful and failure mission outcomes

5) Display 5 records where launch sites begin with the string 'CCA'

6) Date when the first successful landing outcome in ground pad was achieved

7) Names of the boosters which have success in drone ship and have payload mass greater than 4000 but less than 6000

8) Names of the booster versions which have carried the maximum payload mass

9) Failed landing outcomes in drone ship, their booster versions, and launch site names for in year 2015

10) Ranking the count of landing outcomes between the date 2010-06-04 and 2017-03-20

Notebook Link

Build an Interactive Map with Folium

We first create a folium `Map` object, use `folium.Circle` `folium.Marker` to add a highlighted circle around all the launch sites using their latitude and longitude.
Let's add the launch outcomes for each site, and see which sites have high success rates. Next, let's create markers for all launch records. If a launch was successful `(class=1)`, then we use a green marker and if a launch was failed, we use a red marker `(class=0)`
Next we create lines on the map that measure the distance between launch sites and landmarks such as cities railways, highways and coastline , we need to explore and analyze the proximities of launch sites to see whether Launch Sites follow any pattern.

Notebook Link

Build a Dashboard with Plotly Dash

Build a Dashboard using Flask and Dash.
The dashboard includes :

- Pie Chart showing success rate of all launch sites
- It shows us a comparison between all launch sites and their success rate(%), but also of a single launch site.

- Scatter Graph showing the success rate of the launches based on the payload mass (Kg) for different booster versions
- We can change the range of the payload from 0 to 10000 kg. Choose a single launch sites or all launch sites, it also shows what booster was used.

Predictive Analysis (Classification)

Notebook Link

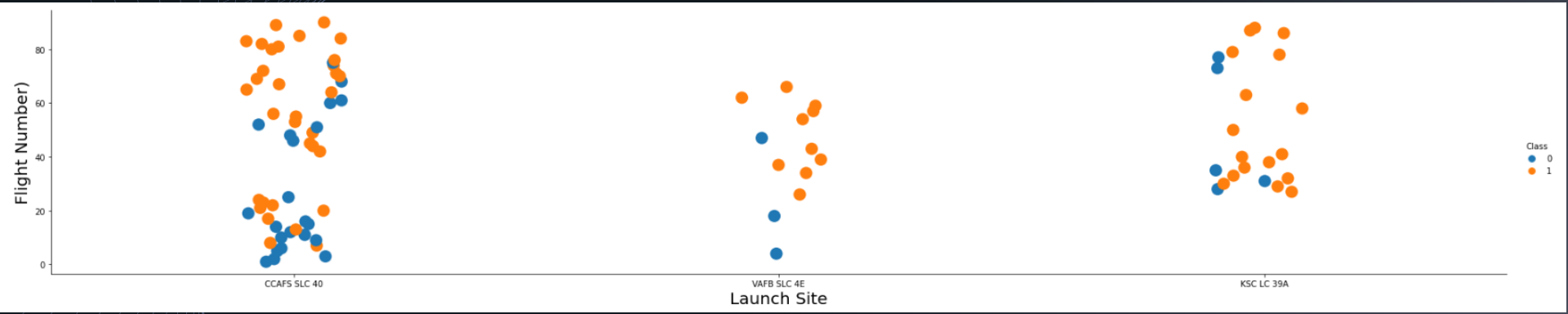
Results

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

Insight drawn from EDA

Flight Number vs Launch Site

Class 1 = success
Class 0 = failure



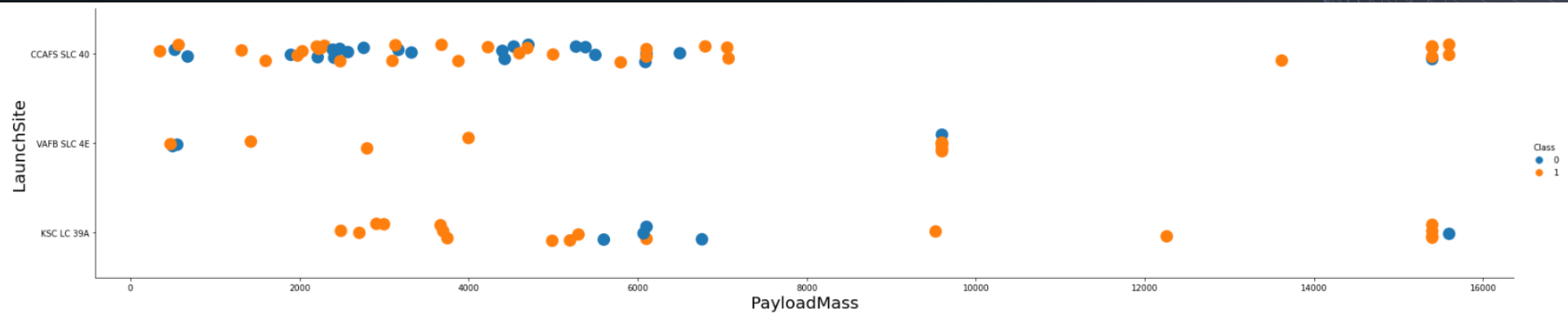
We see that different launch sites have different success rates. **CCAFS LC-40**, has a success rate of **60 %**, while KSC **LC-39A** and **VAFB SLC 4E** has a success rate of **77%**.

Notebook Link

Payload vs. Launch Site

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

Class 1 = success
Class 0 = failure

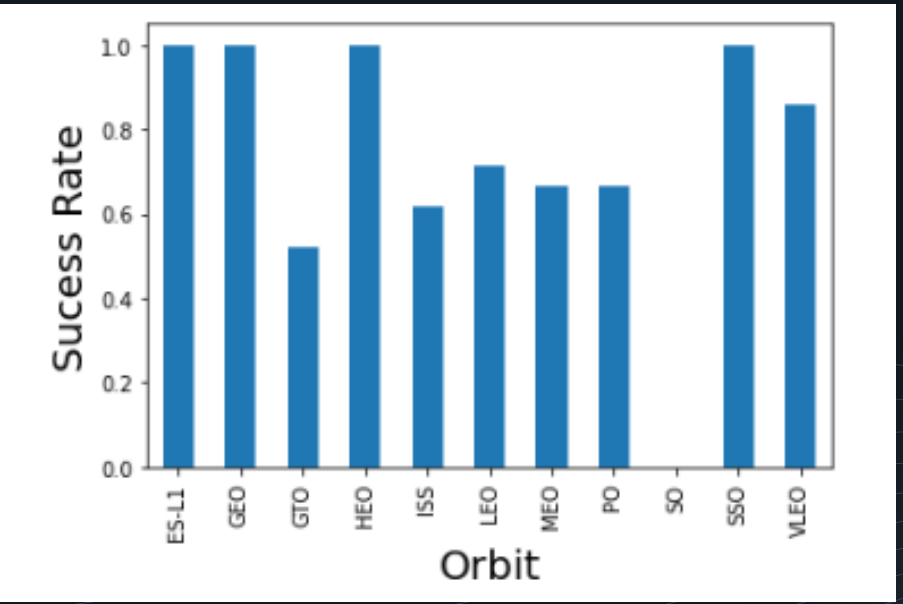


Now if you observe Payload Vs. Launch Site scatter point chart you will find for the VAFB-SLC launch site there are no rockets launched for heavy payload mass(greater than 10000).

Notebook Link

Success Rate vs. Orbit Type

We want to visually check if there are any relationship between success rate and orbit type.



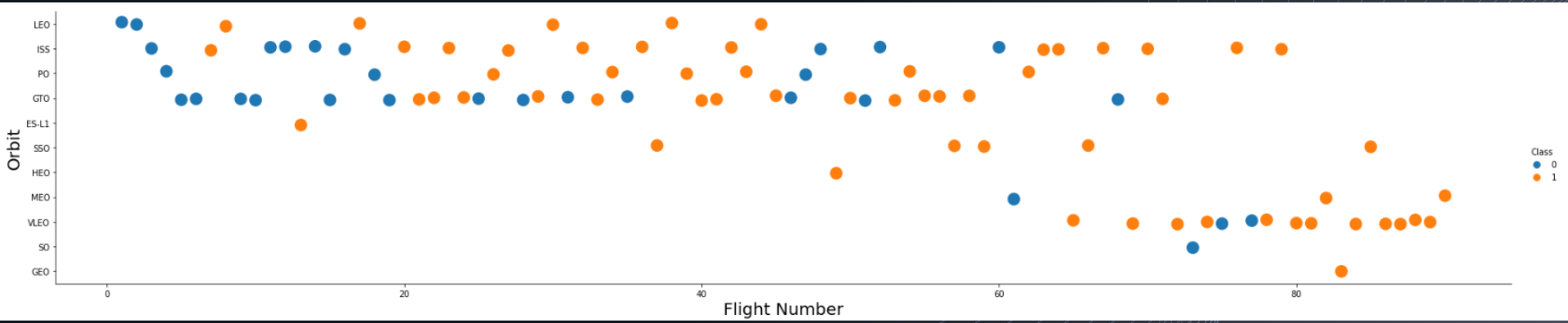
We can see that ES-L1, GEO, HEO and SSO all have a 100% success rate.

Notebook Link

Flight Number vs. Orbit Type

For each orbit, we want to see if there is any relationship between Flight Number and Orbit type.

Class 1 = success
Class 0 = failure



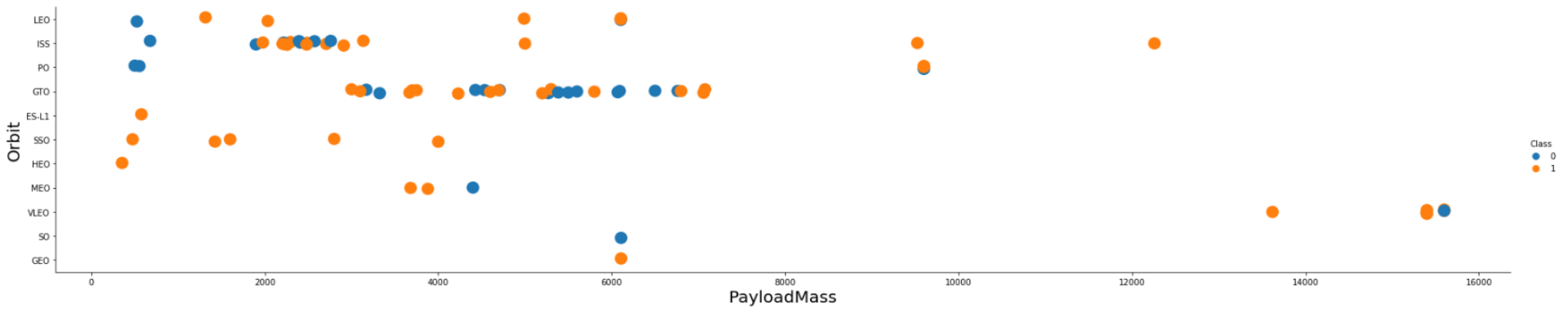
You should see that in the LEO orbit the Success appears related to the number of flights; on the other hand, there seems to be no relationship between flight number when in GTO orbit.

Notebook Link

Payload vs. Orbit Type

We can plot the Payload vs. Orbit scatter point charts to reveal the relationship between Payload and Orbit type

Class 1 = success
Class 0 = failure

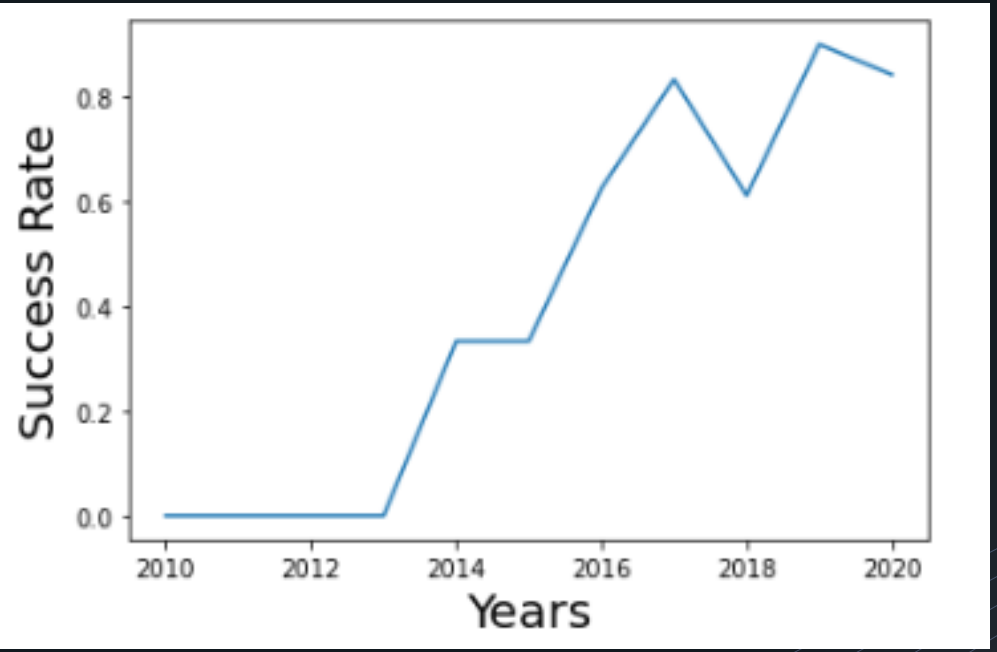


With heavy payloads the successful landing or positive landing rate are more for Polar, LEO and ISS.
However for GTO we cannot distinguish this well as both positive landing rate and negative landing(unsuccessful mission) are both there here.

Notebook Link

Launch Success Yearly Trend

Let's see how the success rate changes throughout the years.



You can observe that the sucess rate since 2013 kept increasing till 2020

Notebook Link

EDA with SQL

All Launch Site Names

SLQ QUERY

SELECT DISTINCT LAUNCH_SITE FROM SPACEXTBL;

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

Used the DISTINCT command to retrieve unique values of the LAUNCH_SITE column from our table SPACEXTBL.

Notebook Link

Launch Site Names Begin with 'CCA'

SLQ QUERY

```
SELECT* FROM SPACEXTBL
WHERE LAUNCH_SITE LIKE
'CCA%' LIMIT 5;
```

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	07:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	No attempt
2012-10-08	00:35:00	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

Used the WHERE command to specify the column LAUNCH_SITE in combination with the LIKE 'CCA%' command to narrow down to only rows containing those letters, and finally a LIMIT command to return only 5 elements of the table.

Notebook Link

Total Payload Mass

SLQ QUERY

```
SELECT
SUM(PAYLOAD_MASS__KG_)
FROM SPACEXTBL WHERE
CUSTOMER LIKE 'NASA
(CRS)';
```

Use the SUM function to calculate the total in the PAYLOAD_MASS__KG_ columns followed by a WHERE command to filter the data to only fetch values containing 'NASA (CRS) from the CUSTOMER columns.

Notebook Link

Average Payload Mass by F9 v1.1

SLQ QUERY

```
SELECT
AVG(PAYLOAD_MASS__KG_)
FROM SPACEXTBL WHERE
BOOSTER_VERSION = 'F9
v1.1'
```

Used the AVG function to calculate the average in the PAYLOAD_MASS__KG_ columns followed by a WHERE command to count only rows that have 'F9 v1.1' in their BOOSTER_VERSION column.

First Successful Ground Landing Date

SLQ QUERY

```
SELECT* FROM SPACEXTBL
WHERE DATE = (SELECT
MIN(DATE) FROM
SPACEXTBL WHERE
LANDING__OUTCOME LIKE
'Success%');
```

DATE	time__utc_	booster_version	launch_site	payload	payload_mass__kg_	orbit	customer	mission_outcome	landing__outcome
2015-12-22	01:29:00	F9 FT B1019	CCAFS LC-40	OG2 Mission 2 11 Orbcomm-OG2 satellites	2034	LEO	Orbcomm	Success	Success (ground pad)

Used the WHERE command to specify the column DATE from which we select the minimum value by using the MIN function followed by a WHERE command so that it only fetches rows that contain the strings LIKE 'Success%' from the LANDING_OUTCOME

Notebook Link

Successful Drone Ship Landing with Payload between 4000 and 6000

SLQ QUERY

```
SELECT* FROM SPACEXTBL WHERE
(PAYLOAD_MASS__KG_ BETWEEN 4000 AND 6000) AND
LANDING__OUTCOME = 'Success (drone ship)';
```

DATE	time__utc_	booster_version	launch_site	payload	payload_mass__kg_	orbit	customer	mission_outcome	landing__outcome
2016-05-06	05:21:00	F9 FT B1022	CCAFS LC-40	JCSAT-14	4696	GTO	SKY Perfect JSAT Group	Success	Success (drone ship)
2016-08-14	05:26:00	F9 FT B1026	CCAFS LC-40	JCSAT-16	4600	GTO	SKY Perfect JSAT Group	Success	Success (drone ship)
2017-03-30	22:27:00	F9 FT B1021.2	KSC LC-39A	SES-10	5300	GTO	SES	Success	Success (drone ship)
2017-10-11	22:53:00	F9 FT B1031.2	KSC LC-39A	SES-11 / EchoStar 105	5200	GTO	SES EchoStar	Success	Success (drone ship)

Used the WHERE command to specify the column PAYLOAD_MASS__KG_ and the BETWEEN statement to consider only values between 4000 and 6000, followed by and AND command that adds an extra filter so that it only fetches rows that have 'Success (drone ship)' in their LANDING_OUTCOME column.

Notebook Link

Total Number of Successful and Failure Mission Outcomes

SLQ QUERY

```
SELECT MISSION_OUTCOME,
COUNT(MISSION_OUTCOME) AS TOTAL_NUMBER FROM
SPACEXTBL GROUP BY MISSION_OUTCOME;
```

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

Used the COUNT function on the MISSION_OUTCOME to count values in the table followed by a GROUP BY statement.

Notebook Link

Boosters Carried Maximum Payload

SLQ QUERY

```
SELECT* FROM SPACEXTBL WHERE
PAYLOAD_MASS__KG_ = (SELECT
MAX(PAYLOAD_MASS__KG_) FROM SPACEXTBL);
```

DATE	time__utc_	booster_version	launch_site	payload	payload_mass__kg_	orbit	customer	mission_outcome	landing__outcome
2019-11-11	14:56:00	F9 B5 B1048.4	CCAFS SLC-40	Starlink 1 v1.0, SpaceX CRS-19	15600	LEO	SpaceX	Success	Success
2020-01-07	02:33:00	F9 B5 B1049.4	CCAFS SLC-40	Starlink 2 v1.0, Crew Dragon in-flight abort test	15600	LEO	SpaceX	Success	Success
2020-01-29	14:07:00	F9 B5 B1051.3	CCAFS SLC-40	Starlink 3 v1.0, Starlink 4 v1.0	15600	LEO	SpaceX	Success	Success
2020-02-17	15:05:00	F9 B5 B1056.4	CCAFS SLC-40	Starlink 4 v1.0, SpaceX CRS-20	15600	LEO	SpaceX	Success	Failure
2020-03-18	12:16:00	F9 B5 B1048.5	KSC LC-39A	Starlink 5 v1.0, Starlink 6 v1.0	15600	LEO	SpaceX	Success	Failure
2020-04-22	19:30:00	F9 B5 B1051.4	KSC LC-39A	Starlink 6 v1.0, Crew Dragon Demo-2	15600	LEO	SpaceX	Success	Success
2020-06-04	01:25:00	F9 B5 B1049.5	CCAFS SLC-40	Starlink 7 v1.0, Starlink 8 v1.0	15600	LEO	SpaceX, Planet Labs	Success	Success
2020-09-03	12:46:14	F9 B5 B1060.2	KSC LC-39A	Starlink 11 v1.0, Starlink 12 v1.0	15600	LEO	SpaceX	Success	Success
2020-10-06	11:29:34	F9 B5 B1058.3	KSC LC-39A	Starlink 12 v1.0, Starlink 13 v1.0	15600	LEO	SpaceX	Success	Success
2020-10-18	12:25:57	F9 B5 B1051.6	KSC LC-39A	Starlink 13 v1.0, Starlink 14 v1.0	15600	LEO	SpaceX	Success	Success
2020-10-24	15:31:34	F9 B5 B1060.3	CCAFS SLC-40	Starlink 14 v1.0, GPS III-04	15600	LEO	SpaceX	Success	Success
2020-11-25	02:13:00	F9 B5 B1049.7	CCAFS SLC-40	Starlink 15 v1.0, SpaceX CRS-21	15600	LEO	SpaceX	Success	Success

Used the WHERE command to fetch only values that are maximum in the table by using a subquery and MAX function

Notebook Link

2015 Launch Records

SLQ QUERY

```
SELECT* FROM SPACEXTBL WHERE
LANDING__OUTCOME = 'Failure (drone ship)' AND DATE
LIKE '2015%';
```

DATE	time__utc_	booster_version	launch_site	payload	payload_mass__kg_	orbit	customer	mission_outcome	landing__outcome
2015-01-10	09:47:00	F9 v1.1 B1012	CCAFS LC-40	SpaceX CRS-5	2395	LEO (ISS)	NASA (CRS)	Success	Failure (drone ship)
2015-04-14	20:10:00	F9 v1.1 B1015	CCAFS LC-40	SpaceX CRS-6	1898	LEO (ISS)	NASA (CRS)	Success	Failure (drone ship)

Used the WHERE command to specify so that we only fetch rows containing 'Failure (drone ship)' from the LANDING_OUTCOME followed by an AND statement to further filter the result by DATE using the LIKE '2015%' clause.

Notebook Link

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

SLQ QUERY

```
SELECT LANDING__OUTCOME, COUNT(LANDING__OUTCOME) AS TOTAL FROM
SPACEXTBL WHERE DATE BETWEEN '2010-06-04' AND '2017-03-20' GROUP BY
LANDING__OUTCOME ORDER BY TOTAL DESC;
```

landing__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

Used the COUNT function on the LANDING_OUTCOME column to count only the values between 2010-06-04 and 2017-03-20 by using the WHERE statement applied to the DATE columns, a BETWEEN command to filter dates, a GROUP BY statement to arrange similar data and finally a DESC function to the total values so that the values are ranked in descending order.

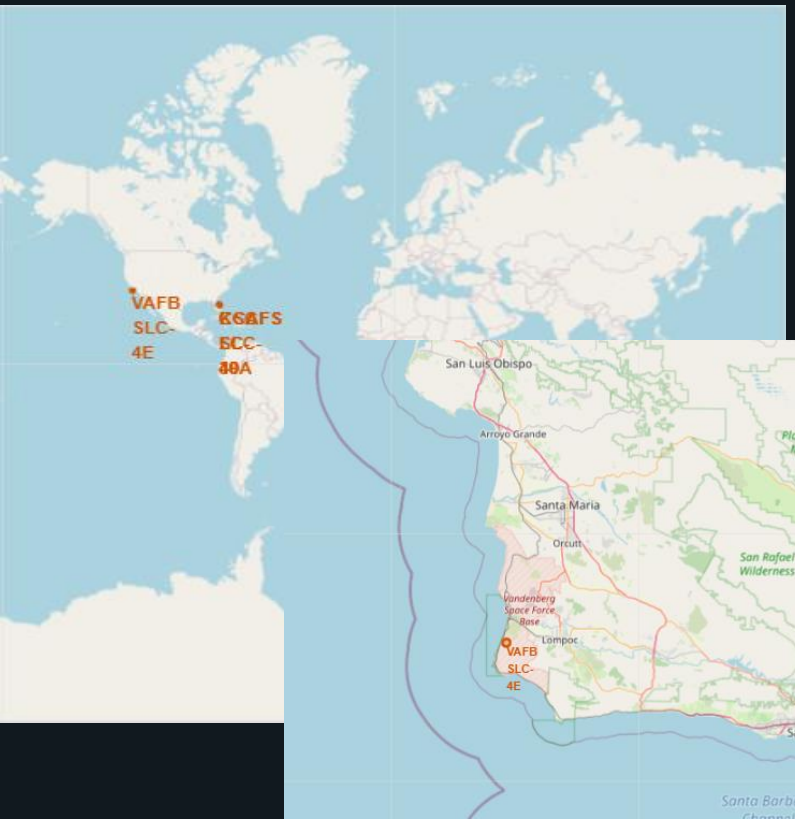
Notebook Link

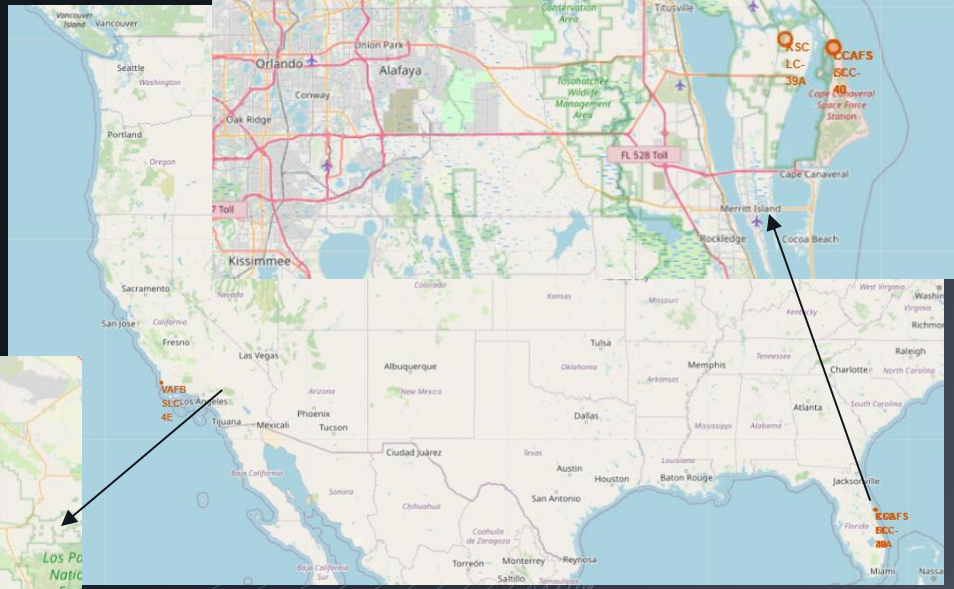
Launch Site Proximities Analysis

Interactive map with Folium

Notebook Link

Let’s find some geographical patterns about launch sites





All Launch sites are located in the USA. CCAFS LC-40 and CCAFS SLC-40 are very close to each other, KSC LC-39A is also nearby; all in Florida.
VAFB SLC-4E on the other hand is located at the opposite side in California.

Launch Records

Green Markers represents successful launches.

Red Markers represent failure in saving the first stage.

CCAFS SLC-40 has the most number of overall launch attempts

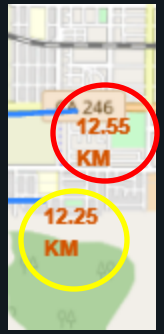
Notebook Link

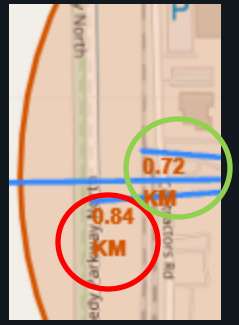
We can clearly see that KSC LC-39A has had the most success

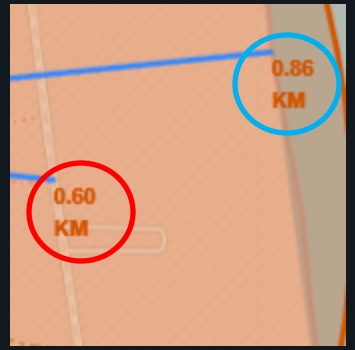
Notebook Link

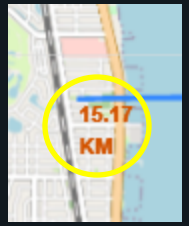
Launch Site Proximities

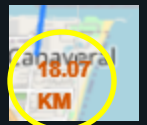
Let's explore and analyze the proximities of launch sites.

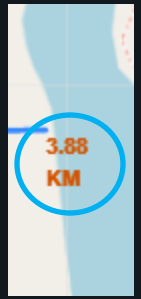


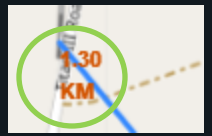


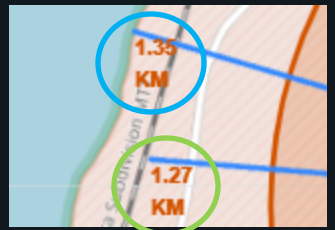


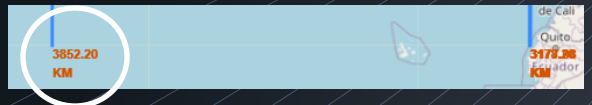












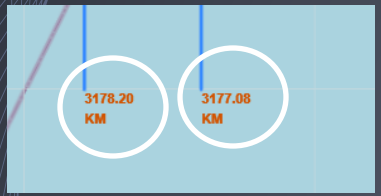
Launch sites can be in close proximities of highways.
600 m < Distance < 12 550 m.

Launch sites keep a distance from Cities.
12 250 m < Distance < 18 070 m.

Launch site can be present near railways.
720 m < Distance < 1 300 m.

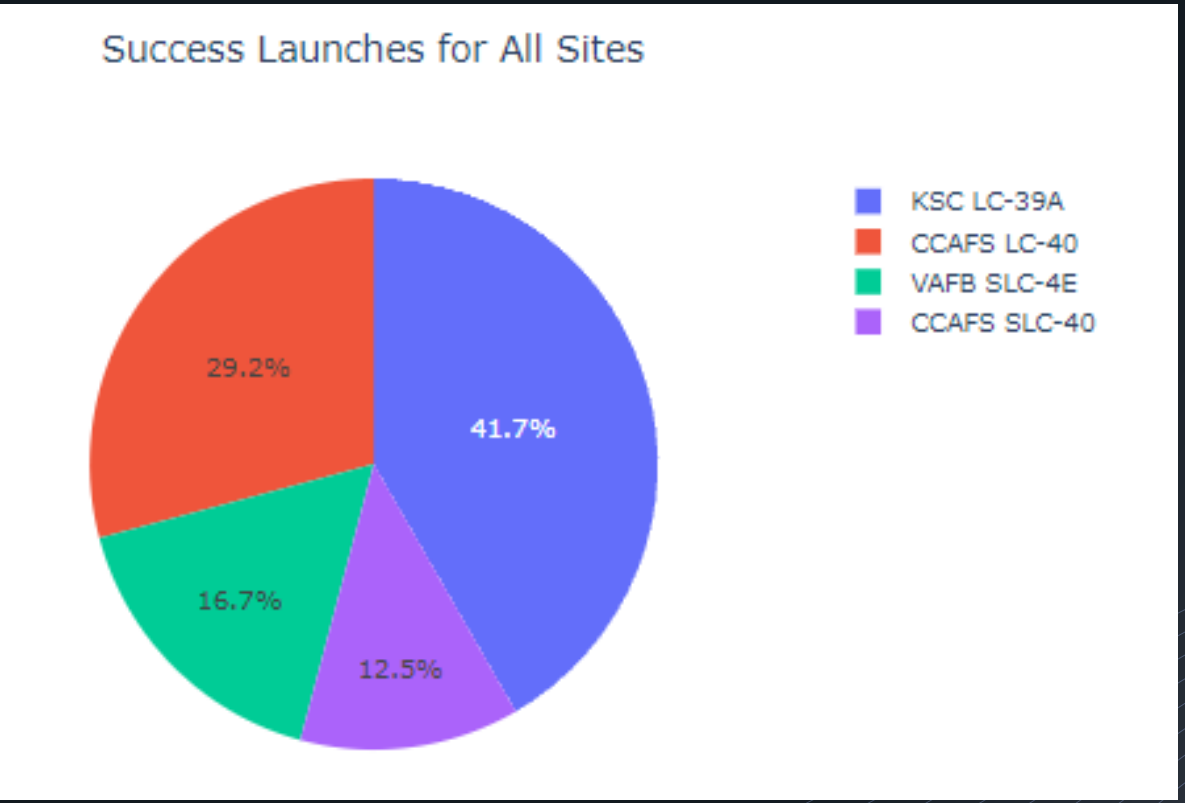
Launch site are close to the coast.
860 m < Distance < 3 880 m.

Launch sites are really far from the Equator.



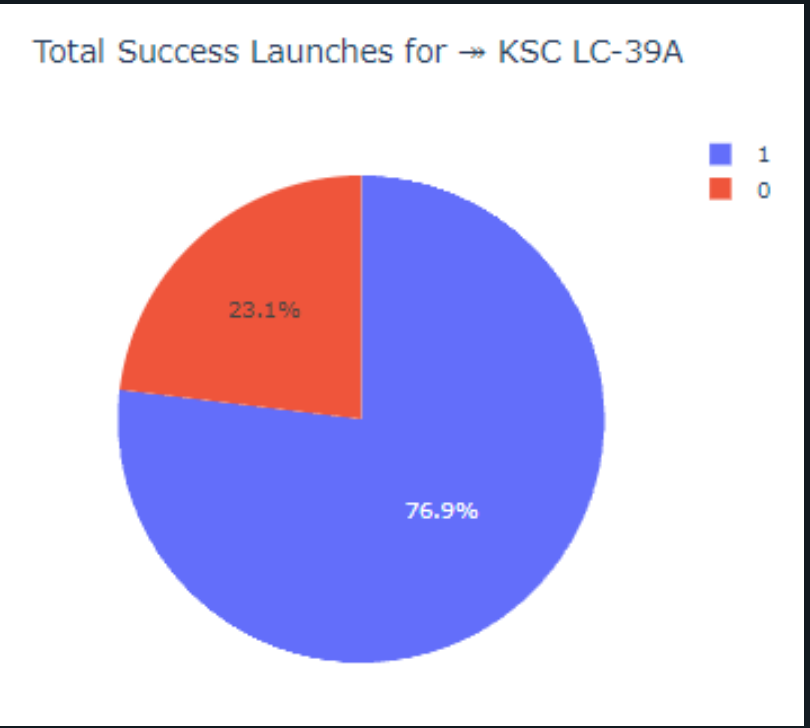
Build a Dashboard with Plotly Dash

Dashboard



KSC LC-39A is the launch site with the most successful launches with a 41.7%

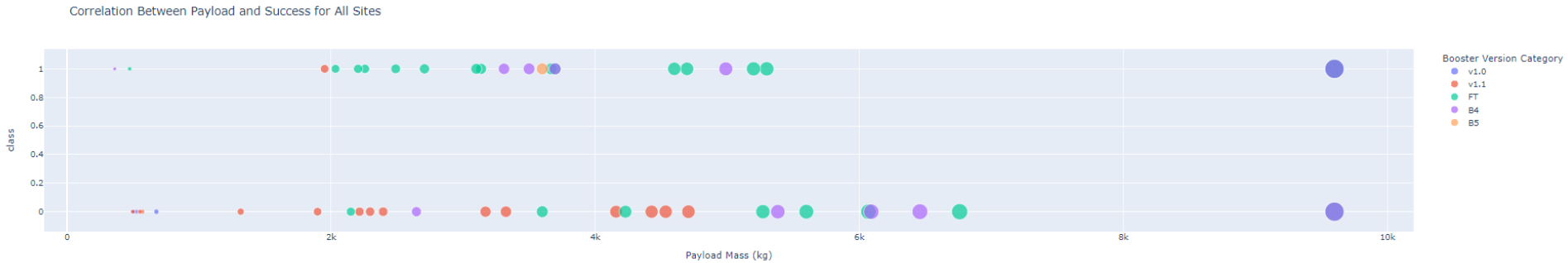
Dashboard with Plotly Dash



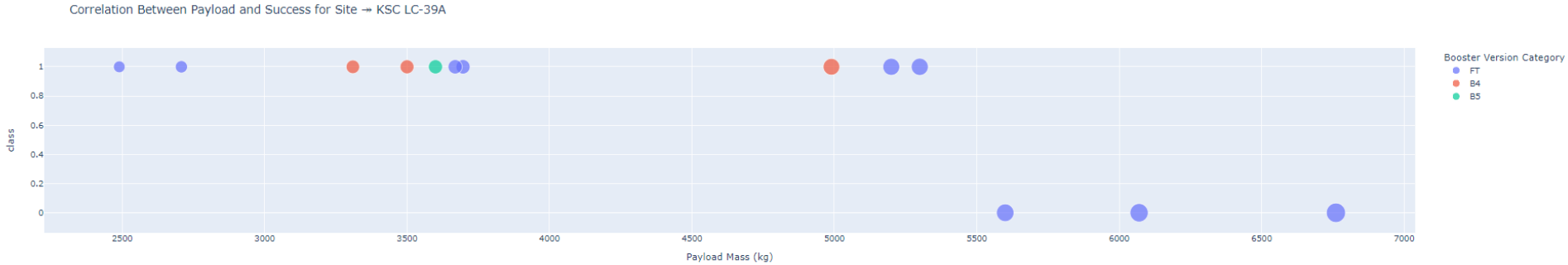
The KSC LC-39A launch site has a success rate of 76.9, and only a 23.1% chance to fail.

1 is success
0 is failure

Payload and Success Rate



We can see that the booster with most success is the FT version. Also that between 0 and 2000 kg the success rate is fairly low. On the other hand the most success is observed in payloads that range from 2000 kg to 4000 kg.

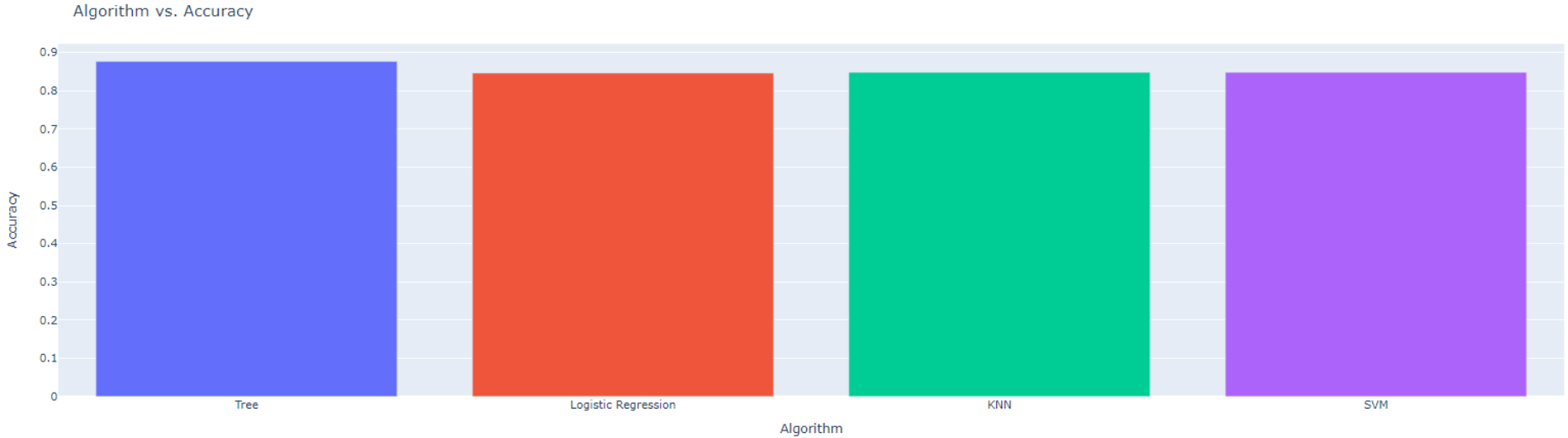


For payloads above 5500 kg the success rate is 0. So it is clear for payload below 5500 kg we have a 100% success rate.

Predictive Analysis (Classification)

Classification Accuracy

Notebook Link

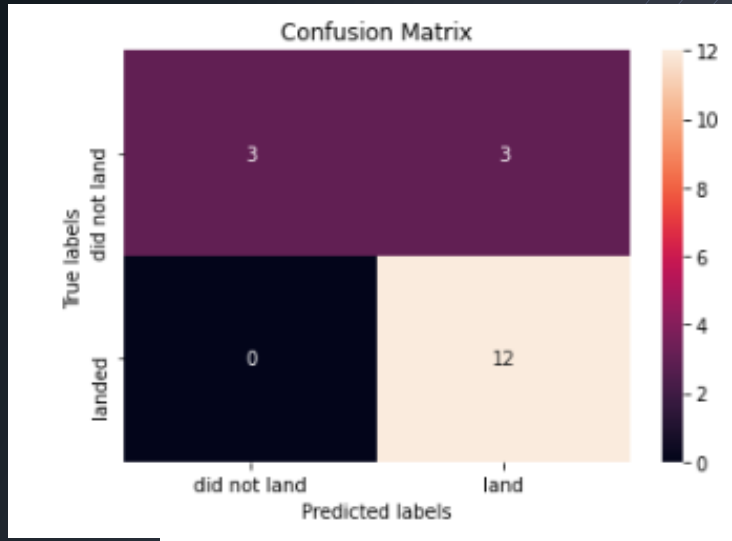


Looking at the graph we can see that the Tree(Decision Tree) model is slightly better than the others.

Model	Accuracy
Tree	0.876786
Logistic Regression	0.846429
KNN	0.848214
SVM	0.848214

Confusion Matrix

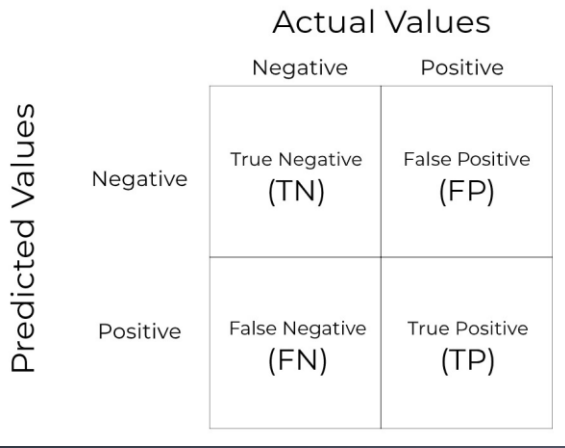
We have the same Matrix for all the 4 models.



Let's now define the most basic terms, which are whole numbers (not rates):

- **true positives (TP):** These are cases in which we predicted yes (they have the disease), and they do have the disease.
- **true negatives (TN):** We predicted no, and they don't have the disease.
- **false positives (FP):** We predicted yes, but they don't actually have the disease. (Also known as a "Type I error.")
- **false negatives (FN):** We predicted no, but they actually do have the disease. (Also known as a "Type II error.")

Accuracy: Overall, how often is the classifier correct? (TP+TN)/total = (3+12)/18 = 0.83333
Misclassification Rate: Overall, how often is it wrong?(FP+FN)/total = (3+0)/18 = 0.167
True Positive Rate: When it's actually yes, how often does it predict yes? TP/actual yes = 12/12 = 1
False Positive Rate: When it's actually no, how often does it predict yes? FP/actual no = 3/6 = 0.5
True Negative Rate: When it's actually no, how often does it predict no? TN/actual no = 3/6 = 0.5
Precision: When it predicts yes, how often is it correct? TP/predicted yes = 12/15 = 0.8
Prevalence: How often does the yes condition actually occur in our sample? actual yes/total = 12/18 = 0.6667



Notebook Link

Conclusions

Success rate of SpaceX Falcon 1 has been increasing with time.

Higher the flight number, higher the success rate.
Majority of the launches have been done at CCAFS LC-40 and also contains most of the first 25 flight numbers, that may be a reason for the lower success rates.

KSC LC-39A launch site has the most success, but a higher payload mass has a negative affect on the success rate.

Between all the orbits ES-L1, GEO, HEO and SSO have a 100% success rate.

The Decision Tree machine learning model gave us the better Accuracy, compared to the other 3.

GitHub URL

Appendix

Exploratory Data Analysis (EDA)

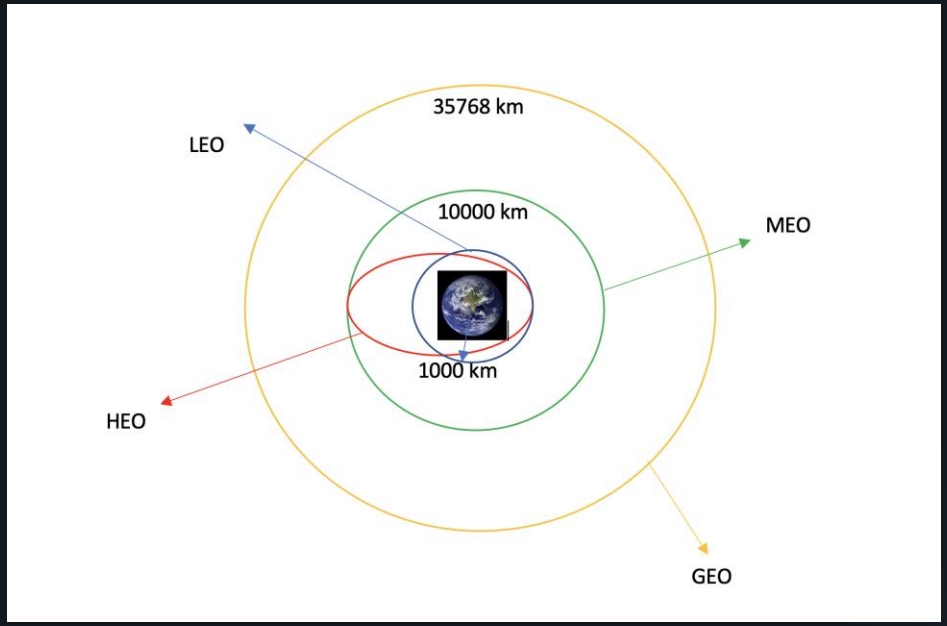
We will perform some Exploratory Data Analysis (EDA) to find some patterns in the data and determine what would be the label for training supervised models.
In the data set, there are several different cases where the booster did not land successfully. Sometimes a landing was attempted but failed due to an accident.
We will mainly convert those outcomes into Training Labels with `1` means the booster successfully landed `0` means it was unsuccessful.

Launch Site	N. of Launches
CCAFS SLC 40	55
KSC LC 39A	22
VAFB SLC 4E	13

The data contains several Space X launch facilities: Cape Canaveral Space Launch Complex 40 **VAFB SLC 4E** , Vandenberg Air Force Base Space Launch Complex 4E **(SLC-4E)**, Kennedy Space Center Launch Complex 39A **KSC LC 39A**

Notebook Link

Each launch aims to a dedicated orbit, and here are some common orbit types



- **LEO**: Low Earth orbit (LEO)is an Earth-centred orbit w ith an altitude of 2,000 km. Most of the manmade objects in outer space are in LEO
- **VLEO**: Very Low Earth Orbits (VLEO) can be defined as the orbits w ith a mean altitude below 450 km. Operating in these orbits can provide a number of benefits to Earth observation spacecraft as the spacecraft operates closer to the observation
- **GTO** A geosynchronous orbit is a high Earth orbit that allow s satellites to match Earth's rotation.
- **SSO (or SO)**: It is a Sun-synchronous orbit also called a heliosynchronous orbit is a nearly polar orbit around a planet, in w hich the satellite passes over any given point of the planet's surface at the same local mean solar time
- **ES-L1** :At the Lagrange points the gravitational forces of the tw o large bodies cancel out in such a w ay that a small object placed in orbit there is in equilibrium relative to the center of mass of the large bodies. L1 is one such point betw een the sun and the earth
- **HEO** A highly elliptical orbit, is an elliptic orbit w ith high eccentricity, usually referring to one around Earth
- **ISS** A modular space station (habitable artificial satellite) in low Earth orbit. It is a multinational collaborative project betw een five participating space agencies
- **MEO** Geocentric orbits ranging in altitude from 2,000 km (1,200 mi) to just below geosynchronous orbit at 35,786 kilometers (22,236 mi).**HEO** Geocentric orbits above the altitude of geosynchronous orbit (35,786 km or 22,236 mi)
- **GEO** It is a circular geosynchronous orbit 35,786 kilometres (22,236 miles) above Earth's equator and follow ing the direction of Earth's rotation
- **PO** It is one type of satellites in w hich a satellite passes above or nearly above both poles of the body being orbited (usually a planet such as the Earth

The table represent the number of launches performed per Orbit

LEO	VLEO	GTO	SSO	ES-L1	HEO	ISS	MEO	SSO	GEO	PO
7	14	27	5	1	1	21	3	1	1	9

Notebook Link

Outcome Of The Mission

Table containing the number of landing outcomes

True ASDS	41
None None	19
True RTLS	14
False ASDS	6
True Ocean	5
None ASDS	2
False Ocean	2
False RTLS	1

- `True Ocean` means the mission outcome was successfully landed to a specific region of the ocean
- `False Ocean` means the mission outcome was unsuccessfully landed to a specific region of the ocean
- `True RTLS` means the mission outcome was successfully landed to a ground pad
- `False RTLS` means the mission outcome was unsuccessfully landed to a ground pad
- `True ASDS` means the mission outcome was successfully landed to a drone ship
- `False ASDS` means the mission outcome was unsuccessfully landed to a drone ship.
- `None ASDS` and `None None` these represent a failure to land.

Overall Success Rate is 67%

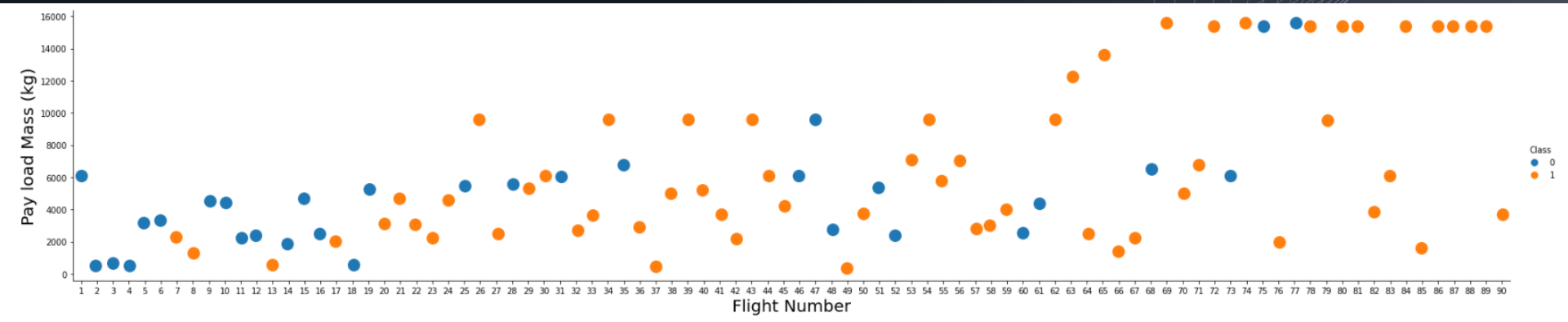
Notebook Link

Payload vs. Flight number

Let's try to see how the `Flight Number` (indicating the continuous launch attempts.) and `Payload` variables would affect the launch outcome.

We see that as the flight number increases, the first stage is more likely to land successfully. The payload mass is also important; it seems the more massive the payload, the less likely the first stage will return.

Class 1 = success
Class 0 = failure



Notebook Link

Extra Predictive Analysis

Parameters used for all the machine learning models:

- Logistic Regression parameters = {'C':[0.01,0.1,1], 'penalty':['l2'], 'solver' ['lbfgs']}
- SVM parameters = {'kernel':('linear', 'rbf','poly','rbf', 'sigmoid'), 'C': np.logspace(-3, 3, 5), 'gamma':np.logspace(-3, 3, 5)}
- Decision Tree parameters = {'criterion': ['gini', 'entropy'], 'splitter': ['best', 'random'], 'max_depth': [2*n for n in range(1,10)], 'max_features': ['auto', 'sqrt'], 'min_samples_leaf': [1, 2, 4], 'min_samples_split': [2, 5, 10]}
- KNN parameters = {'n_neighbors': [1, 2, 3, 4, 5, 6, 7, 8, 9, 10], 'algorithm': ['auto', 'ball_tree', 'kd_tree', 'brute'], 'p': [1,2]}
-

Thank You