Winning Space Race with Data Science

<Alberto Terranova>
<31.05.2022>

Outline

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

Executive Summary

- **Methodology**:
  - **Data Collection**: SpaceX API, and Web-Scraping from SpaceX Wikipedia page.
  - **Data Wrangling**:
    - Missing Data were imputed with mean values
    - Binary Classification: 0-1 unsuccesful-succesful landing
  - **Exploratory Data Analysis (EDA)**:
    - SQL – for Database analytics
    - Folium – for visual analysis
  - **Predictive Analysis**:
    - Logistic Regression
    - Support Vector Machine
    - Decision Tree
    - K-Nearest Neighbors

Executive Summary

- **Results predictive analysis**:
  - All four models in the previous slide achieved similar testing accuracy (83.3%) with a prediction accuracy varying in the range of 84-87%
  - However more data is required for more comprehensive studies.

Introduction

- SpaceX’s costs for launching Falcon9 rockets is roughly $62m, which is almost 3x cheaper than NASA’s. This is partially due to the fact SpaceX’s technology allows for re-usage of landed rockets.
- The aim of this project was to determine whether SpaceX Falcon9 first stage would land successfully or not.
- Business-wise if a rocket will land, then one can determine the cost of a launch and use this information to infer whether or not a company should or shouldn’t enter the market.

Section 1

Methodology

Data Collection REST API

Two methods were used for data collection:

1. Using the **SpaceX REST API** one retrieves important information about the rocket, payload, landing outcome etc.

2. The main function used to retrieve data and convert then to a json file is:

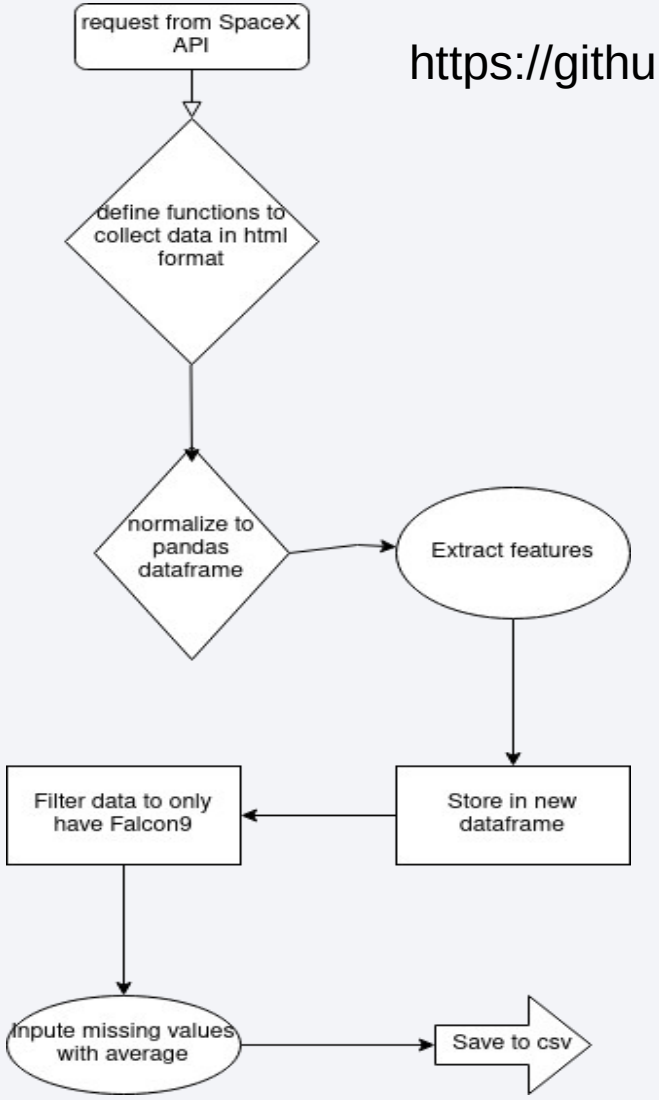
- requests.get("https://api.spacexdata.com/v4/*/"+str()).json()

Where * is replaced by the attribute e.g. rockets, launchpads, payloads, cores, launches

3. After retrieving information in a json format, they get converted into a panda dataframe calling the method “pandas.json_normalize()”

4. Data is then filtered to include only Falcon9 launches and Payload missing values inputed with the mean

Data Collection – SpaceX API



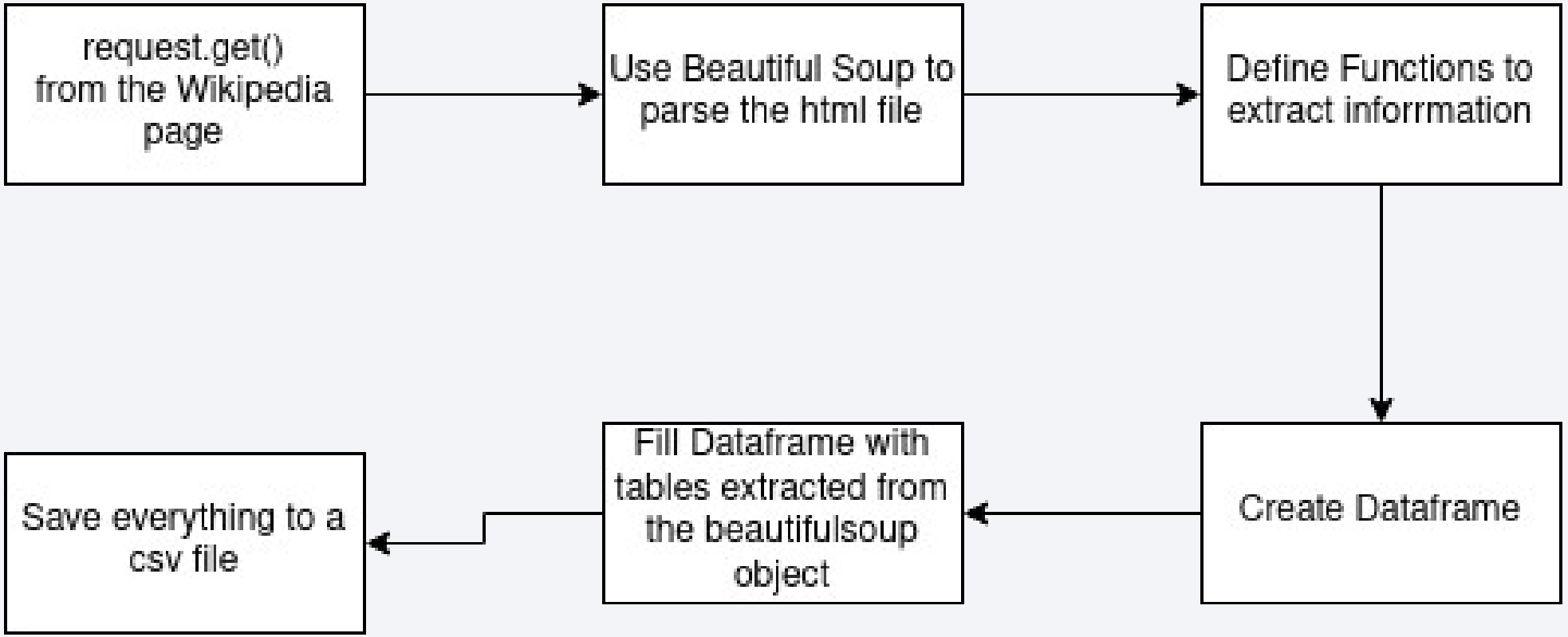
https://github.com/albertoterranova/IBM/blob/main/1%20-%20Data_collection_API.ipynb

Here you can find the notebook

Data Collection - Scraping

Here is the link to the github notebook

https://github.com/albertoterranova/IBM/blob/main/2%20-%20Data_Collection_Webscraping.ipynb



EDA with Data Visualization

https://github.com/albertoterranova/IBM/blob/main/4%20-%20EDA.ipynb

Scatter Plot	Bar Chart	Line Chart
Flight Number vs Launch Site	Success Rate vs Orbit Type	Success Rate vs Year
Payload vs Launch Site		
Orbit Type vs Flight Number		
Payload and Orbit Type		

Scatter plots are useful to observe dependence or correlations between numerical variables

Bar Charts are used to compare numerical values to categorical ones

Line charts contain numerical variables on both axis. In this case we used them to show the change of a variable over time

EDA with SQL

https://github.com/albertoterranova/IBM/blob/main/5%20-%20EDA_with_SQL.ipynb

1. Show names of unique launch sites
2. Show 5 launch sites whose name begins with ‘CCA’
3. Show the total payload mass carried by NASA’s boosters (CRS)
4. Show the average payload carried by Falcon9 v.1.1
5. List dates of first successful landing on ground pad
6. List the names of boosters landed on droneships and with payload mass between 4000 and 6000 kg
7. List the total number of successful and failed missions
8. List the name of boosters carrying the maximum payload mass
9. List the failed landing on drone ships, boosters name and launch site names for 2015
10. Rank the count of landings between 04/06/2010 and 20/03/2017

Build an Interactive Map with Folium

https://github.com/albertoterranova/IBM/blob/main/5%20-%20Folium.ipynb

All launch sites were displayed on a map using a folium map object by adding folium.Circle and folium.Marker for each site.

Many launches were performed on the same site, therefore we classified the successful and failed launches respectively with 1 and 0.

We proceeded by adding a folium.Marker to a MarkerCluster() object for each site. Colors were then assigned using the icon_color.

Great-circle distance was calculated between points using latitude and longitude coordinates extracted from the dataframe. Distances were drawn using the folium.Polyline method.

Build a Dashboard with Plotly Dash

https://github.com/albertoterranova/IBM/blob/main/7%20-%20spacex_dash_app.py

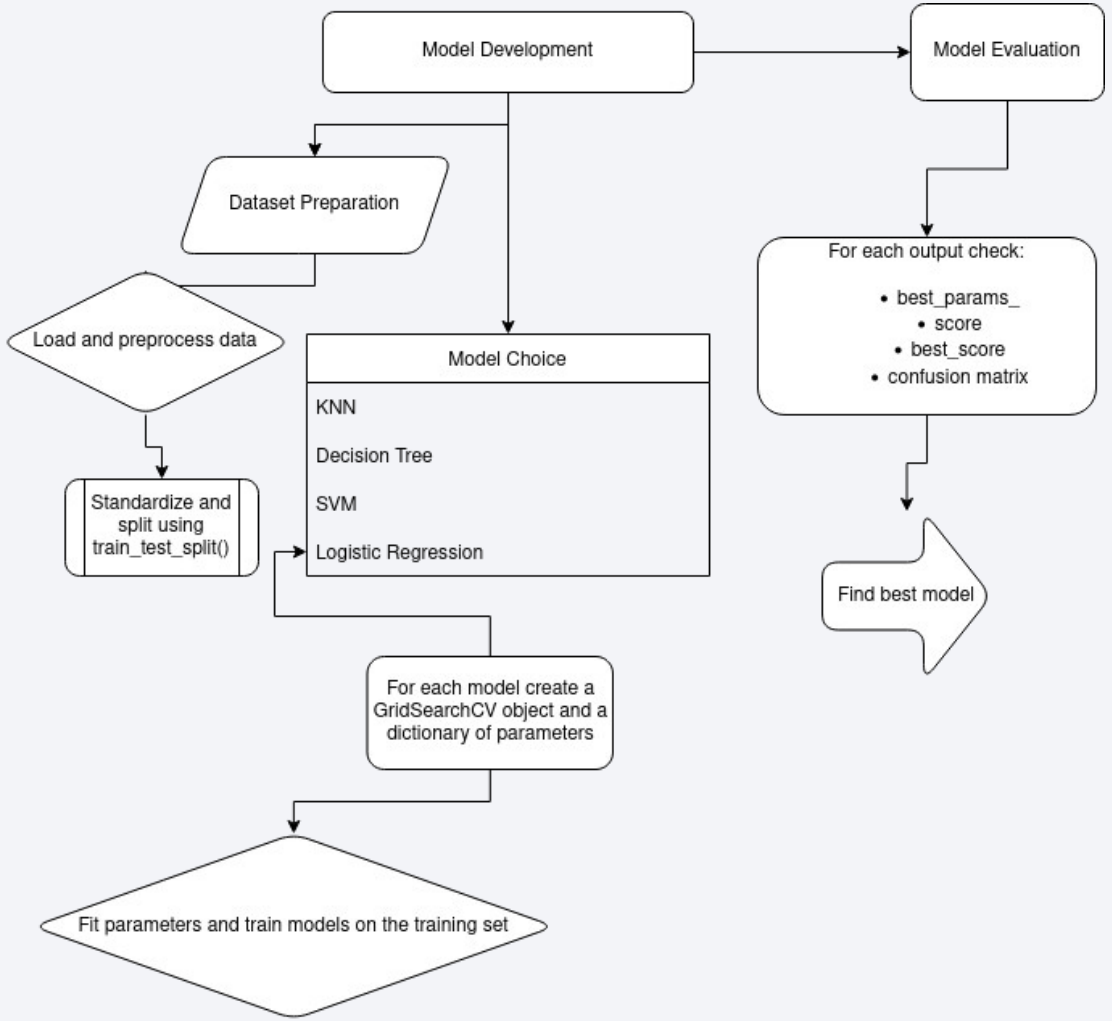
On the Potly Dash dashboard we displayed

1. **Pie chart** using the method px.pie(). This method allows the user to easily see the percentage of successful and unsuccessful launches. Using the dcc.Dropdown() method one manges to check for each individual site.

2. **Scatter plot** using the method pc.scatter(). This graph showed the dependence between payload mass in kg and the outcome of the landing. Using the RangeSlider() method, the user can extract infographics of the selected payload mass range.

Predictive Analysis (Classification)

https://github.com/albertoterranova/IBM/blob/main/6%20-%20ML_Prediction.ipynb



Results

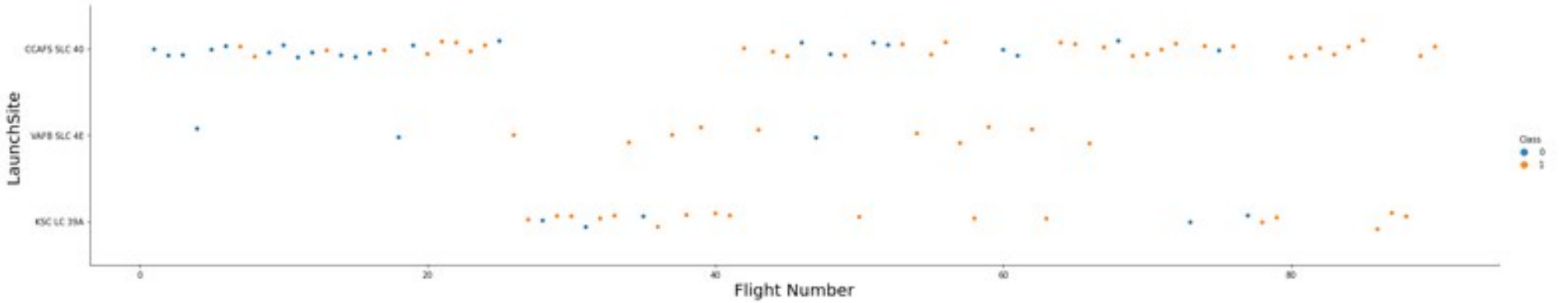
- Exploratory data analysis results
- Interactive analytics demo in screenshots
- Predictive analysis results

Section 2

Insights drawn from EDA

Flight Number vs. Launch Site

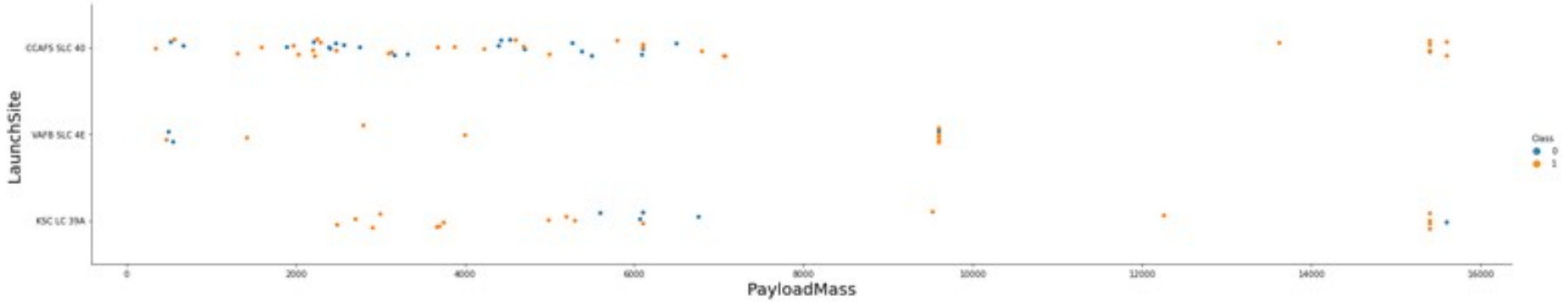
```
# Plot a scatter point chart with x axis to be Flight Number and y axis to be the launch site, and
sns.catplot(y="LaunchSite", x="FlightNumber", hue="Class", data=df, aspect = 5)
plt.xlabel("Flight Number",fontsize=20)
plt.ylabel("LaunchSite",fontsize=20)
plt.show()
```



In general one can infer that with more launches the success rate increases.
For the first 30 launches almost all were deployed from the CCAFS SLC 40 station and most of them were unsuccessful.
The most successful site was KSC LC 39A, from where no early launches were performed.
Above the threshold of 30 launches, the success was significantly high.

Payload vs. Launch Site

```
# Plot a scatter point chart with x axis to be Pay Load Mass (kg) and y axis to be the launch site
sns.catplot(y="LaunchSite", x="PayloadMass", hue="Class", data=df, aspect = 5)
plt.xlabel("PayloadMass",fontsize=20)
plt.ylabel("LaunchSite",fontsize=20)
plt.show()
```



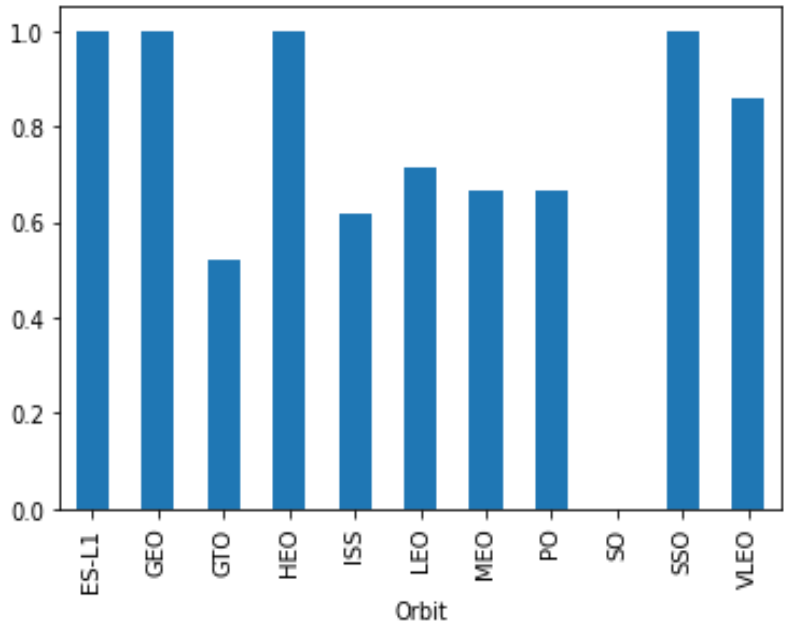
In general one cannot infer much from this scatterplot as it shows no clear correlation/dependence between payload mass and success rate.
Above 8000 kg of payload there are little unsuccessful launches yet the number of launches is too small to infer robust results.

Success Rate vs. Orbit Type

- ES-L1,GEO,HEO,SSO have 100% success rate
- VLEO comes in second place with roughly 90%
- GTO,ISS,LEO,MEO,PO range between 50% and 70%
- SO performs poorly with a staggering 0% of success rate

```
# HINT use groupby method on Orbit column and get the mean of Class column
df.groupby('Orbit')['Class'].mean().plot.bar()
```

```
<AxesSubplot:xlabel='Orbit'>
```



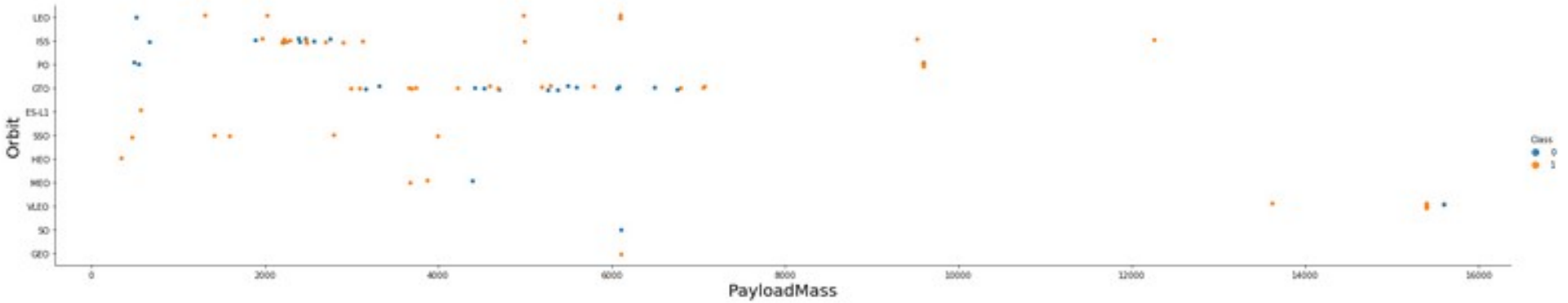
Flight Number vs. Orbit Type

```
# Plot a scatter point chart with x axis to be Payload and y axis to be the Orbit, and hue to be t
sns.catplot(y="Orbit", x="PayloadMass", hue="Class", data=df, aspect = 5)
plt.xlabel("PayloadMass",fontsize=20)
plt.ylabel("Orbit",fontsize=20)
plt.show()
```

From this scatterplot one can infer results regarding also the previous slide.
- The 100% success rate of GEO, HEO and ES-L1 is explained by having only one launch into their respective orbits.
- SSO has instead 100% on 5 flights
- We can easily see a positive correlation occurring between the number of flights and their success.

Payload vs. Orbit Type

```
# Plot a scatter point chart with x axis to be Payload and y axis to be the Orbit, and hue to be t
sns.catplot(y="Orbit", x="PayloadMass", hue="Class", data=df, aspect = 5)
plt.xlabel("PayloadMass",fontsize=20)
plt.ylabel("Orbit",fontsize=20)
plt.show()
```



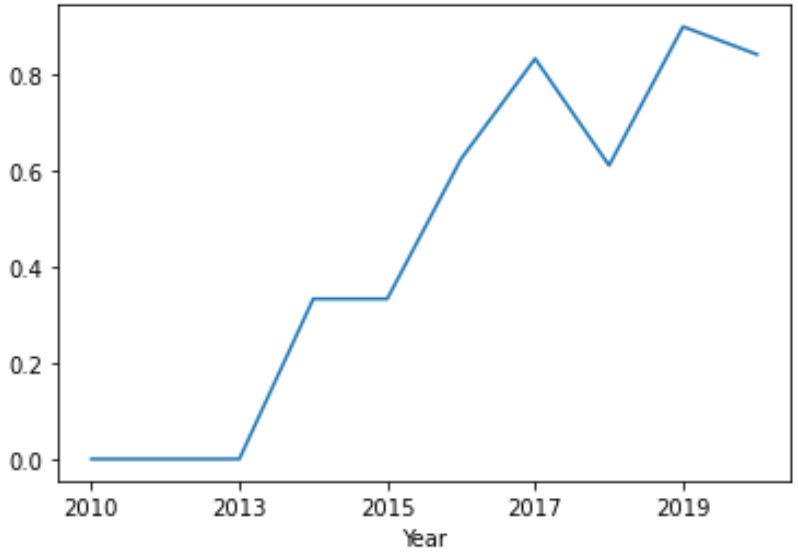
- Almost all orbits performed well with payloads < 8000kg except SO and MEO
- PO,ISS and LEO show however few unsuccessful launches on light payloads
- GTO does not have a clear trend
- Above 8000kg only PO,ISS and VLEO were tested with positive results

Launch Success Yearly Trend

This graph shows the exponential increase in successful launches.
- 2010-2013 all landings were unsuccessful
- After 2013 the success rate started increasing
- 2017 and 2019 marked two all time high in success topping almost 90% of all launches.

```
temp_df = df.copy()
temp_df['Year'] = year
temp_df.groupby('Year')['Class'].mean().plot()
```

```
<AxesSubplot:xlabel='Year'>
```



All Launch Site Names

- %sql is used to call the SQL extension on python

- DISTINCT is a builtin method of SQL that selects the unique values from a database

```
%sql SELECT DISTINCT LAUNCH_SITE FROM SPACEXBL
```

```
 * ibm_db_sa://ggx76781:***@764264db-9824-4b7c-82df
pdomain.cloud:32536/bludb
Done.
```

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

Launch Site Names Begin with 'CCA'

- Here we take all values SELECT * FROM …

- We specify the column LAUNCH_SITE

- Filter using LIKE (case sensitive)

- in ‘CCA%’ the value % is used to take any string at its place

- LIMIT 5 is to take 5 rows

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

```
 * ibm_db_sa://ggx76781:***@764264db-9824-4b7c-82df-40d1b13897c2.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:32536/bludb
Done.
```

DATE	time__utc_	booster_version	launch_site	payload	payload_mass__kg_	orbit	customer	mission_outcome	lan
2010-06-04	18:45:00	F9 v1.0 B0003	CCAFS LC-40	Dragon Spacecraft Qualification Unit	0	LEO	SpaceX	Success	Fa
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	Fa
2012-05-22	07:44:00	F9 v1.0 B0005	CCAFS LC-40	Dragon demo flight C2	525	LEO (ISS)	NASA (COTS)	Success	
2012-10-08	00:35:00	F9 v1.0 B0006	CCAFS LC-40	SpaceX CRS-1	500	LEO (ISS)	NASA (CRS)	Success	
2013-03-01	15:10:00	F9 v1.0 B0007	CCAFS LC-40	SpaceX CRS-2	677	LEO (ISS)	NASA (CRS)	Success	

Total Payload Mass

```
%sql SELECT SUM(PAYLOAD_MASS__KG_) AS TOTAL_PAYLOAD FROM SPACEXBL WHERE PAYLOAD  LIKE '%CRS%';
```

```
 * ibm_db_sa://ggx76781:***@764264db-9824-4b7c-82df-40d1b13897c2.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:32536/bludb
Done.
```

total_payload
111268

- SUM() takes the sum of all rows in the chosen column, in this case PAYLOAD_MASS_KG
- Store the result in TOTAL_PAYLOAD
- ‘%CRS%’ is used to take all the payload whose name contain CRS

Average Payload Mass by F9 v1.1

```
%sql SELECT AVG(PAYLOAD_MASS__KG_) AS AVG_PAYLOAD FROM SPACEXBL WHERE BOOSTER_VERSION = 'F9 v1.1';
```

```
 * ibm_db_sa://ggx76781:***@764264db-9824-4b7c-82df-40d1b13897c2.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:32536/bludb
Done.
```

avg_payload
2928

- AVG() takes the average
- We select only the booster version ‘F9 v.1.1’

First Successful Ground Landing Date

```
%sql SELECT MIN(DATE) AS FIRST_SUCCESS_GP FROM SPACEXBL WHERE LANDING__OUTCOME = 'Success (ground pad)'
```

```
 * ibm_db_sa://ggx76781:***@764264db-9824-4b7c-82df-40d1b13897c2.bs2io90l08kqb1od8lcg.databases.appdomai
n.cloud:32536/bludb
Done.
```

first_success_gp
2015-12-22

- MIN(DATE) to select the first date
- Filter the landing on the ground using LANDING_OUTCOME = ‘Success (ground pad)’

Successful Drone Ship Landing with Payload between 4000 and 6000

```
%sql SELECT DISTINCT BOOSTER_VERSION FROM SPACEXBL WHERE PAYLOAD_MASS__KG_ BETWEEN 4000 AND 6000 AND LANDING__OUTCOME = 'Succe
```

```
 * ibm_db_sa://ggx76781:***@764264db-9824-4b7c-82df-40d1b13897c2.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:32536/bludb
Done.
```

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

Here we select unique values of boosters versions and filter the mass using the function BETWEEN.
We further filter the landing outcome on the successful drone ship

Total Number of Successful and Failure Mission Outcomes

```
%sql SELECT MISSION_OUTCOME, COUNT(*) AS QTY FROM SPACEXBL GROUP BY MISSION_OUTCOME ORDER BY MISSION_OUTCOME;
```

```
 * ibm_db_sa://ggx76781:***@764264db-9824-4b7c-82df-40d1b13897c2.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:32536/bludb
Done.
```

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

- Extract mission outcome
- COUNT(*) counts the number of rows
- Group by mission_outcome intends to split the count into the subcategories of mission outcome
- Order by is used to rule the order of the dataset

2015 Launch Records

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

```
 * ibm_db_sa://ggx76781:***@764264db-9824-4b7c-82df-40d1b13897c2.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:32536/bludb
Done.
```

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

Here we use the function DATE_PART('YEAR',DATE) = 2015 to filter only for the year 2015 in the database.
Note that the function DATE_PART() returns a numeric value, e.g. DATE_PART('yy',01/01/1999) yields as a result 1999.

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

```
%sql SELECT LANDING__OUTCOME, COUNT(*) AS QTY FROM SPACEXBL WHERE DATE BETWEEN '2010-06-04' AND '2017-03-20' GROUP BY LANDING__OUTCOME OR
```

```
 * ibm_db_sa://ggx76781:***@764264db-9824-4b7c-82df-40d1b13897c2.bs2io90l08kqb1od8lcg.databases.appdomain.cloud:32536/bludb
Done.
```

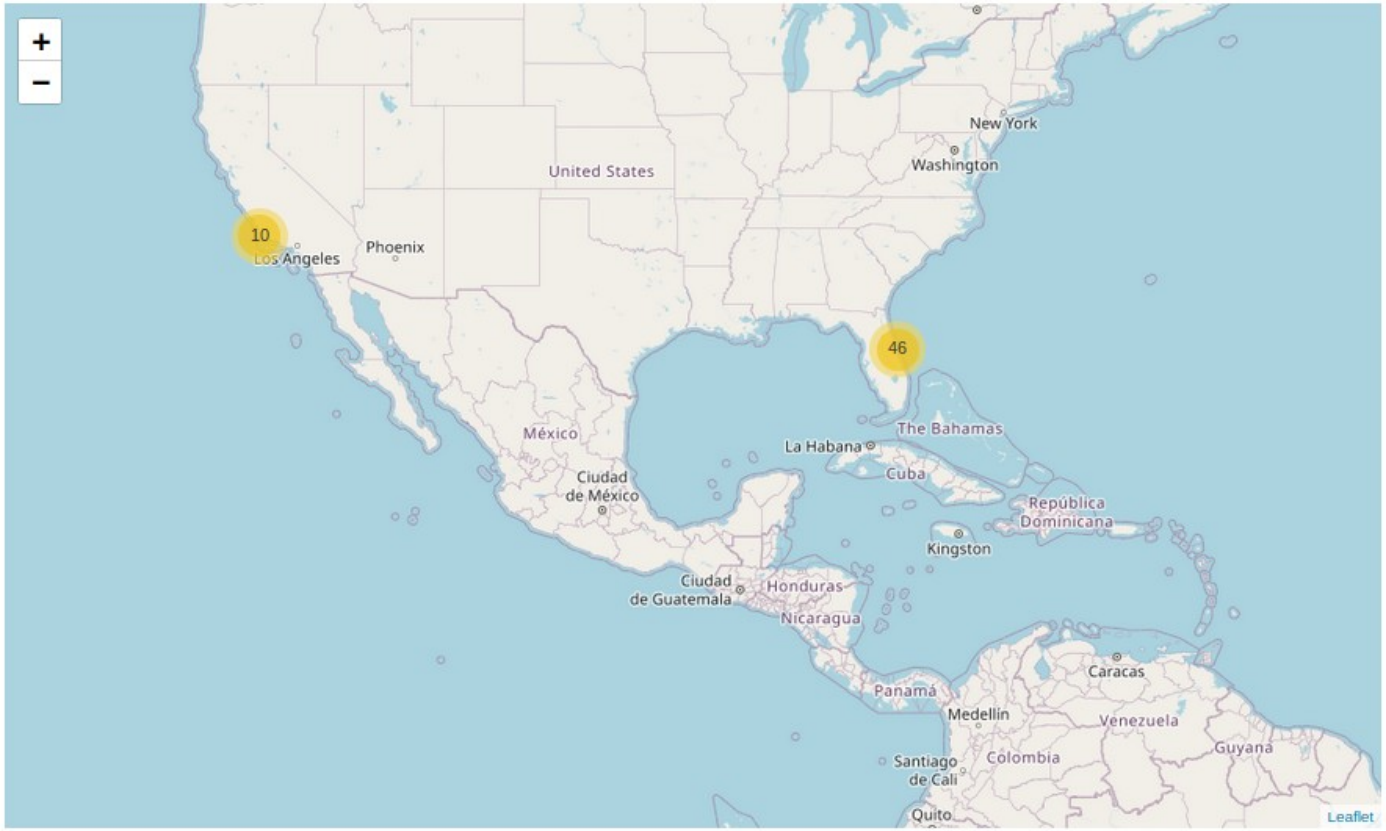
landing__outcome	qty
No attempt	10
Failure (drone ship)	5
Success (drone ship)	5
Controlled (ocean)	3
Success (ground pad)	3
Failure (parachute)	2
Uncontrolled (ocean)	2
Precluded (drone ship)	1

Section 3

Launch Sites Proximities Analysis

All Launch Sites

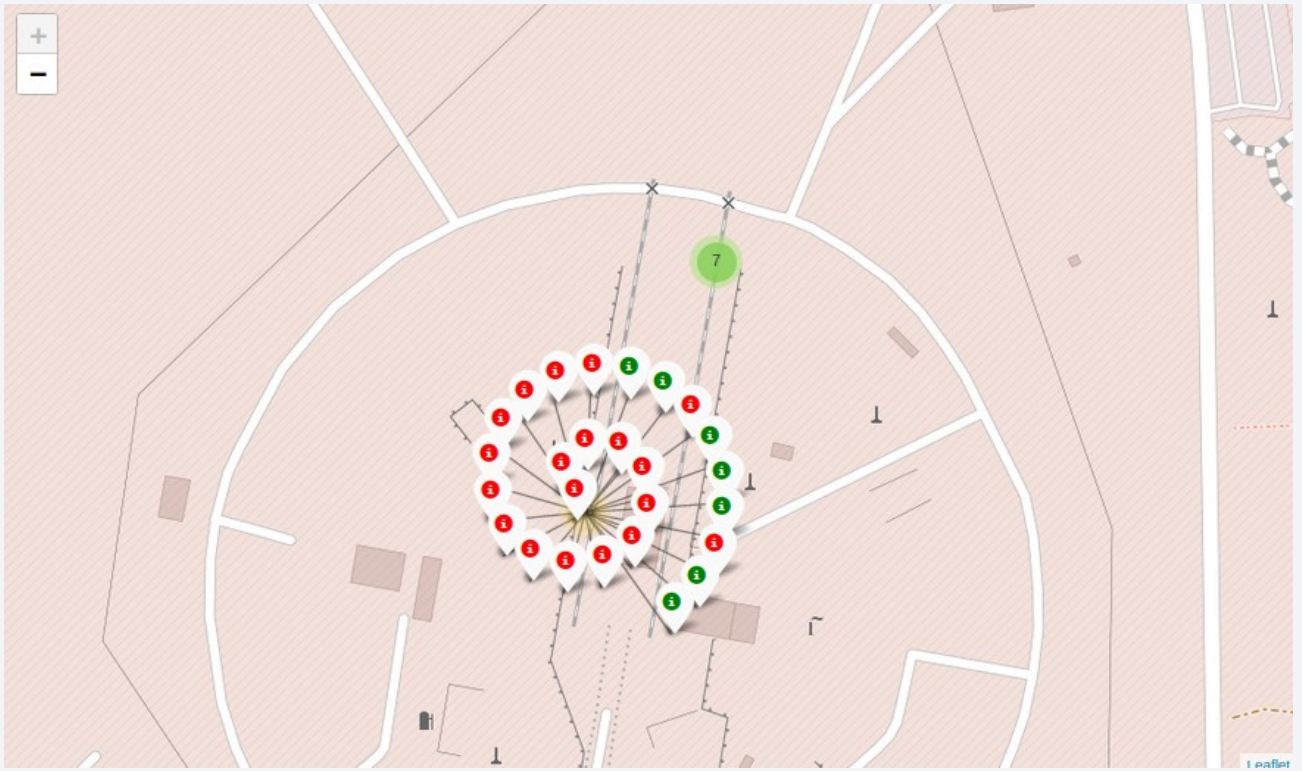
In the map we can see two launching sites with the yellow circle listing the number of launches



Successful/Failed launch for each site

If one clicks on the yellow marker, it will show the number of failed (red) and successful (green) launches for each site.

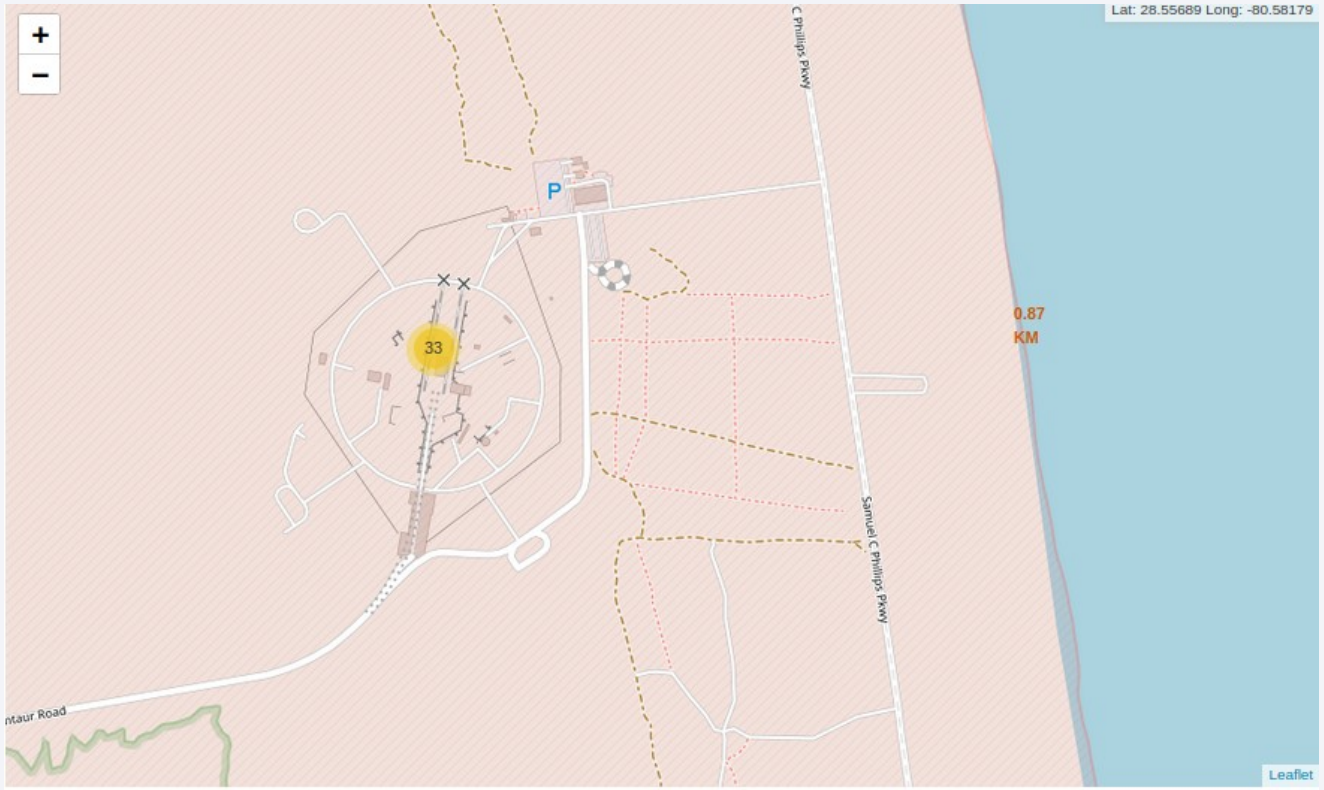
The marker was added on the map using the folium.Marker().
The white Icon was added using the folium.Icon(color='white',…) function

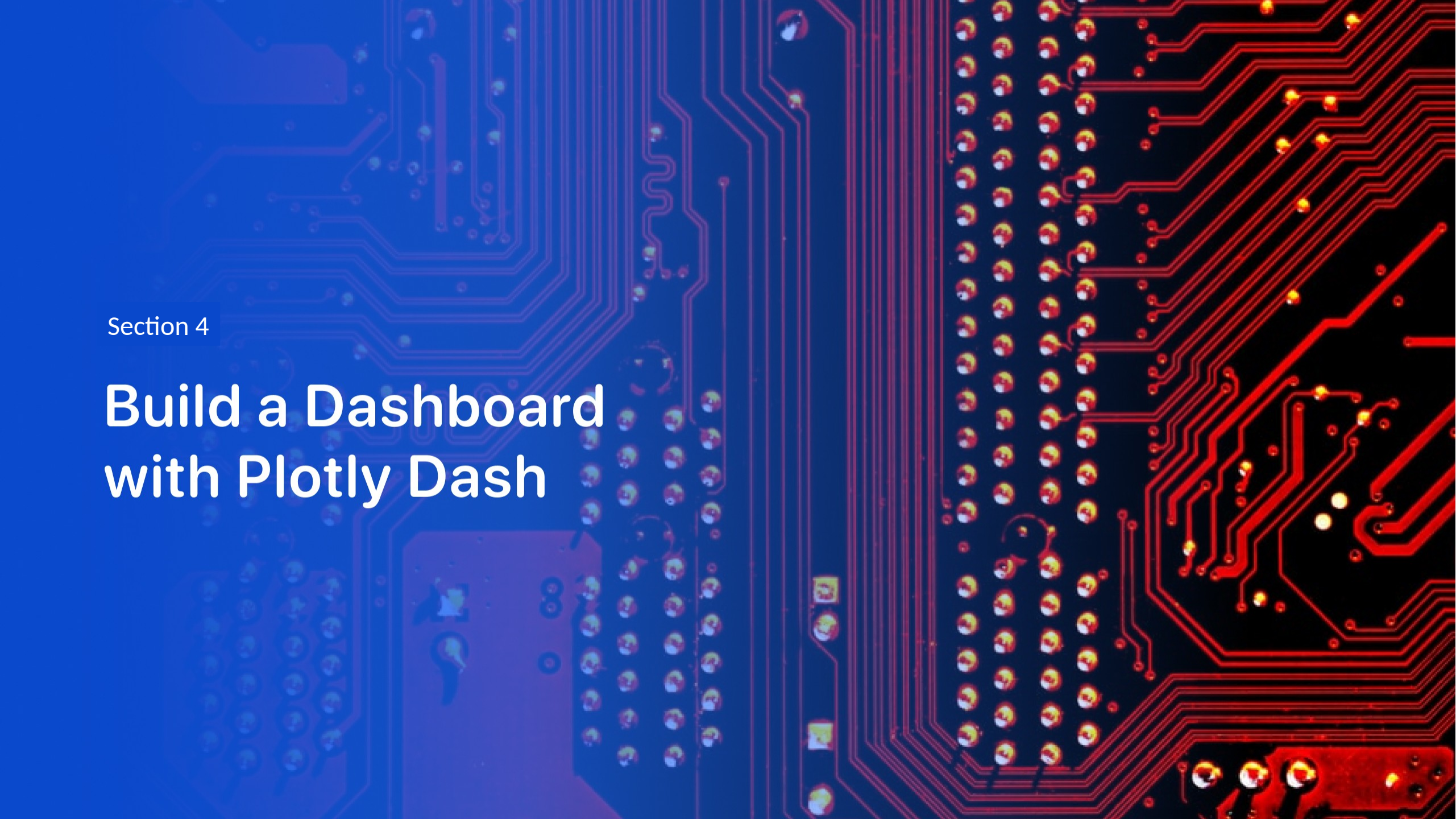


Distance to coastline

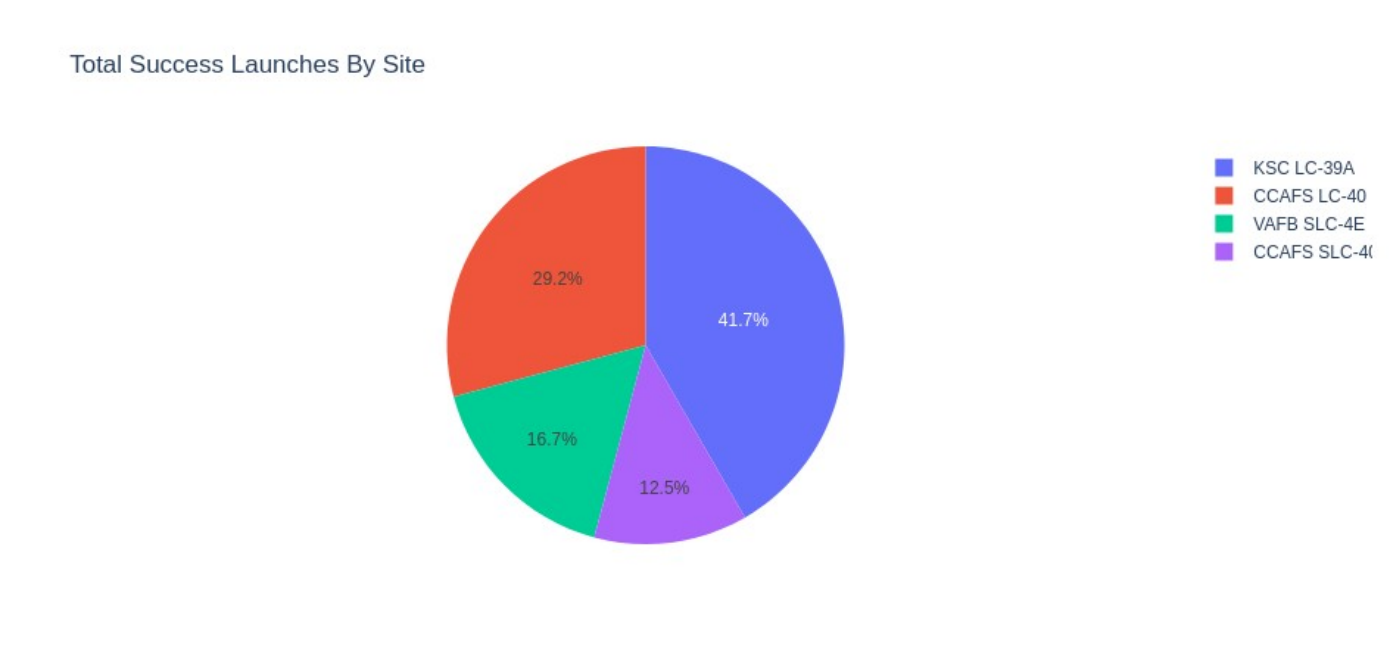
The red number 0.87 km is the shortest path to the coast line from the launch site.

It was calculated using the great-circle distance which is indeed the shortest path between two points.




Section 4
Build a Dashboard with Plotly Dash

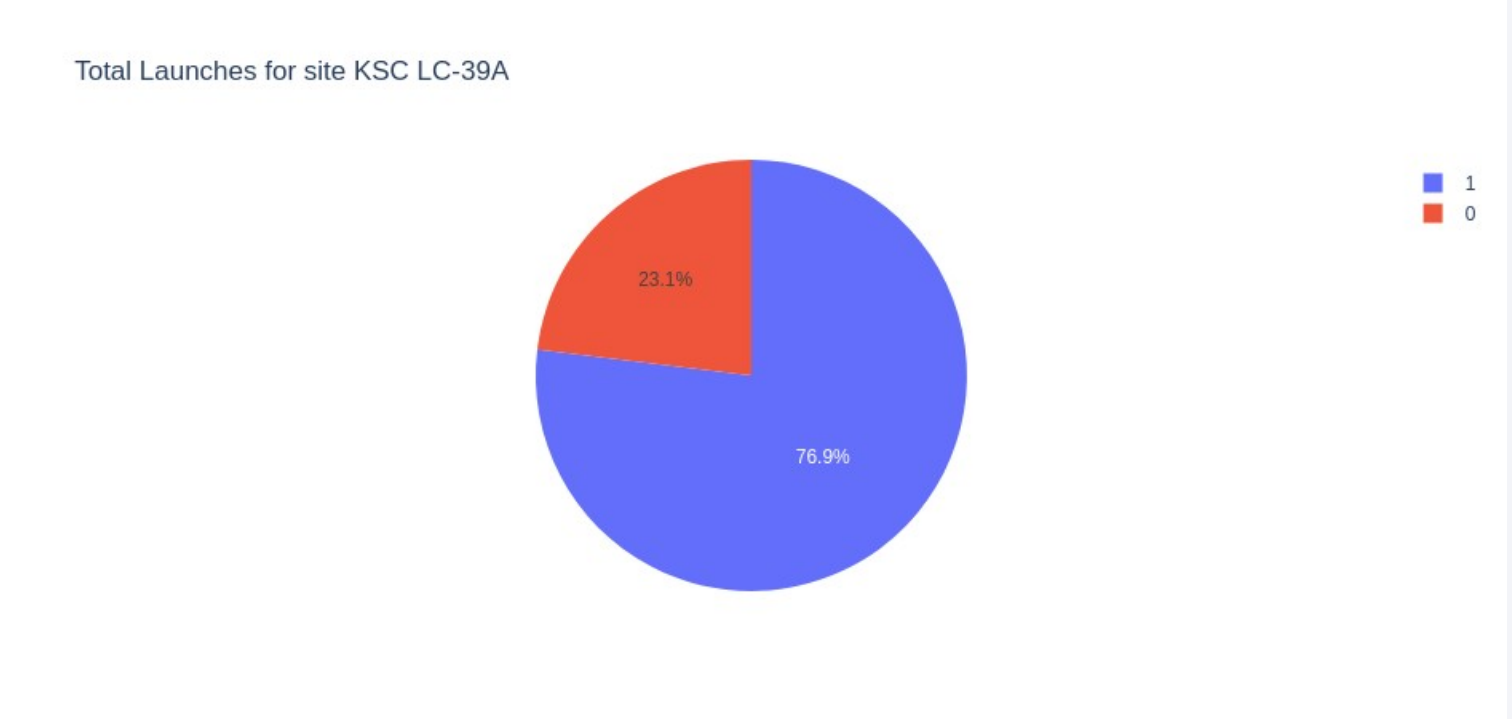
PIE CHART SPACEX – Counting Success



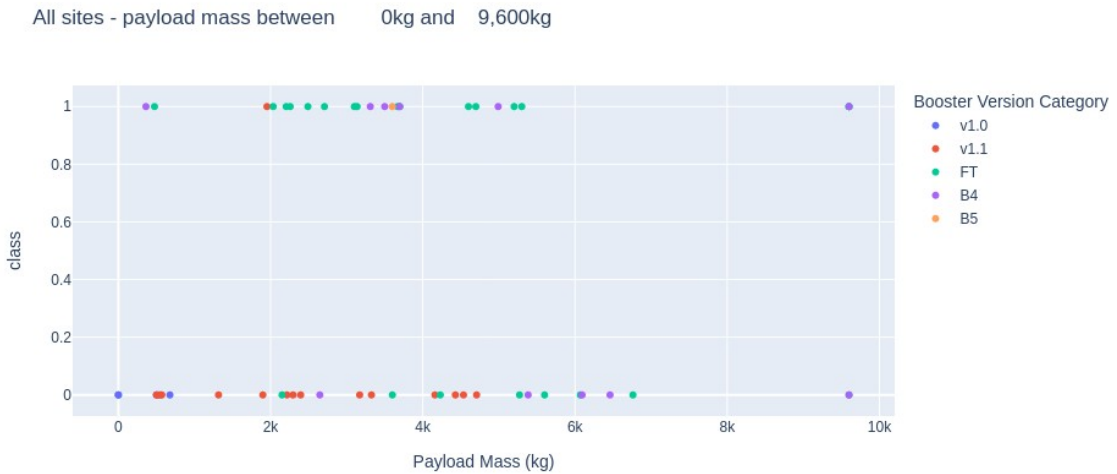
From the hovering menu one can see that this is the pie chart for all sites.
The site with most successes is the KSC LC 39 A with 41.7% overall successful launches coming from this site.
The plot was download using the widget download plot at the top right corner identified with a small camera.

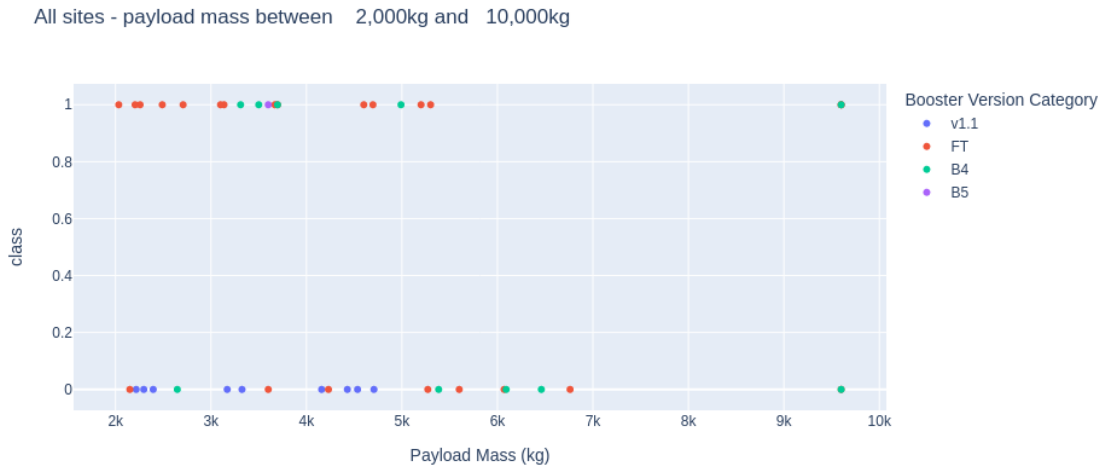
KSC LC-39A PIE CHART

In this figure we can see the pie chart with the highest success rate (76.9).



Payload vs Launch Outcome





In these plots we can see the outcome for each payload range in successful launches.
The figures show also the different booster versions.

Section 5

Predictive Analysis (Classification)

Classification Accuracy

Model	Accuracy	TestAccuracy
LogReg	0.84643	0.83333
SVM	0.84821	0.83333
Tree	0.87679	0.83333
KNN	0.84821	0.83333

The decision tree has the highest classification accuracy

Conclusions

More flights → More Accuracy

Launches seem to be independent of Payload Mass, however the threshold seem to be around 4000kg for successful launches

More Data on orbits are needed in order to infer conclusions

KSC LC-39 A is the most successful site

All classification model seem to perform equally well on the data, more models need to be tested though.

Thank you!