IBM Developer
SKILLS NETWORK

Winning Space Race with Data Science

<Nik Mohamad Azam>
<10th March 2024>

Outline

Executive Summary

- ## Summary of methodologies

  - Data has been collected by using get request / beautiful soup API
  - Data is processed by using several API such as Python Pandas, NumPy, Seaborn, SQL query, etc.
  - Folium API has been used to show location of launch site
  - Logistic Regression, Support Vector Machine, Tree Classifier, K-Nearest Neighbors has been used to predict succes rate of Falcon 9 rocket landing

- ## Summary of all results

  - Correlation between several variable such as, Pay load mass (kg), launch site, orbit, success rate are available in graph figures
  - Location of launch site is shown in map figures
  - Prediction of success landing is shown in simple numeric value and confusion matrix plot

Introduction

- Project background and context

SpaceX advertises on their website about Falcon 9 rocket that launches with its cost of only 62 million dollars, while other providers’ rocket could cost up around 165 million dollars each. The reason why SpaceX rocket cost incredibly low is due to they are able to reuse the first stage of their rockets.

If we can determine the first stage success landing rate, we could determine the cost of their product. This information can be used if an alternate company wants to bid against SpaceX for rocket launch

- Problems you want to find answers

The objective of this project is to find the probability of the Falcon 9 first stage successful landing.

Section 1

Methodology

Methodology

Data collection methodology

※Data has been collected by using get request / beautiful soup API

Perform data wrangling

※Data is processed by using several API such as Python Pandas, NumPy, Seaborn, SQL query, etc.

Perform exploratory data analysis (EDA) using visualization and SQL

Perform interactive visual analytics using Folium and Plotly Dash

Perform predictive analysis using classification model

※Logistic Regression, Support Vector Machine, Tree Classifier, K-Nearest Neighbors has been used

Data Collection

- Data sets are mainly collected by using the python functions.
- Data is collected from various source, such as SpaceX API, Wikipedia, and other website links.
- The data are then read and converted into Python Pandas API for further analyzing.
- The data is also saved into csv file format .

Data Collection – SpaceX API

Used Python requests.get() to below SpaceX API links to collect information:

- https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN-SkillsNetwork/datasets/API_call_spacex_api.json
- https://api.spacexdata.com/v4/rockets/
- https://api.spacexdata.com/v4/launchpads
- https://api.spacexdata.com/v4/payloads/
- https://api.spacexdata.com/v4/cores/
- https://api.spacexdata.com/v4/launches/past

All collected data is then converted in Python Pandas API Data frame and CSV file format .
Then, filter out other rocket launches and only keep Falcon 9 rocket information.

	FlightNumber	Date	BoosterVersion	PayloadMass	Orbit	LaunchSite	Outcome	Flights	GridFins	Reused	Legs	LandingPad	Block	ReusedCount	Serial	Longitude	Latitude
4	6	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	8	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	10	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	11	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	12	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
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
89	102	2020-09-03	Falcon 9	15600.0	VLEO	KSC LC 39A	True ASDS	2	True	True	True	5e9e3032383ecb6bb234e7ca	5.0	12	B1060	-80.603956	28.608058
90	103	2020-10-06	Falcon 9	15600.0	VLEO	KSC LC 39A	True ASDS	3	True	True	True	5e9e3032383ecb6bb234e7ca	5.0	13	B1058	-80.603956	28.608058
91	104	2020-10-18	Falcon 9	15600.0	VLEO	KSC LC 39A	True ASDS	6	True	True	True	5e9e3032383ecb6bb234e7ca	5.0	12	B1051	-80.603956	28.608058
92	105	2020-10-24	Falcon 9	15600.0	VLEO	CCSFS SLC 40	True ASDS	3	True	True	True	5e9e3033383ecbb9e534e7cc	5.0	12	B1060	-80.577366	28.561857
93	106	2020-11-05	Falcon 9	3681.0	MEO	CCSFS SLC 40	True ASDS	1	True	False	True	5e9e3032383ecb6bb234e7ca	5.0	8	B1062	-80.577366	28.561857

GitHub URL:

https://github.com/NikAzam422/SpaceX.git

File name:

jupyter-labs-spacex-data-collection-api.ipynb

File name:

jupyter-labs-webscraping.ipynb

Data Collection - Scraping

Used Python requests.get() and Beautiful Soup API function to below Wikipedia links to collect information from a table in the website:

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

The Webscrapping processed also including extracting column/header, parsing HTML table, and converting into Python Pandas API Data frame format and CSV file format.

	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
1	FlightNumber	Date	BoosterVersion	PayloadMass	Orbit	LaunchSite	Outcome	Flights	GridFins	Reused	Legs	LandingPad	Block	ReusedCount	Serial	Longitude	Latitude
2	1	2010-06-04	Falcon 9	6123.54764705882	LEO	CCSFS SLC 40	None None	1	False	False	False		1	0	B0003	-80.577366	28.5618571
3	2	2012-05-22	Falcon 9	525	LEO	CCSFS SLC 40	None None	1	False	False	False		1	0	B0005	-80.577366	28.5618571
4	3	2013-03-01	Falcon 9	677	ISS	CCSFS SLC 40	None None	1	False	False	False		1	0	B0007	-80.577366	28.5618571
5	4	2013-09-29	Falcon 9	500	PO	VAFB SLC 4E	False Ocean	1	False	False	False		1	0	B1003	-120.610829	34.632093
6	5	2013-12-03	Falcon 9	3170	GTO	CCSFS SLC 40	None None	1	False	False	False		1	0	B1004	-80.577366	28.5618571
84	83	2020-07-20	Falcon 9	[illegible]	[illegible]	CCSFS SLC 40	True ASDS	2	True	True	True	[illegible]	5	[illegible]	[illegible]	[illegible]	[illegible]
85	84	2020-08-18	Falcon 9	15600	VLEO	CCSFS SLC 40	True ASDS	6	True	True	True	5e9e3032383ecb6bb234e7ca	5	9	B1049	-80.577366	28.5618571
86	85	2020-08-30	Falcon 9	1600	SSO	CCSFS SLC 40	True RTLS	4	True	True	True	5e9e3032383ecb267a34e7c7	5	5	B1059	-80.577366	28.5618571
87	86	2020-09-03	Falcon 9	15600	VLEO	KSC LC 39A	True ASDS	2	True	True	True	5e9e3032383ecb6bb234e7ca	5	12	B1060	-80.6039558	28.6080585
88	87	2020-10-06	Falcon 9	15600	VLEO	KSC LC 39A	True ASDS	3	True	True	True	5e9e3032383ecb6bb234e7ca	5	13	B1058	-80.6039558	28.6080585
89	88	2020-10-18	Falcon 9	15600	VLEO	KSC LC 39A	True ASDS	6	True	True	True	5e9e3032383ecb6bb234e7ca	5	12	B1051	-80.6039558	28.6080585
90	89	2020-10-24	Falcon 9	15600	VLEO	CCSFS SLC 40	True ASDS	3	True	True	True	5e9e3033383ecbb9e534e7cc	5	12	B1060	-80.577366	28.5618571
91	90	2020-11-05	Falcon 9	3681	MEO	CCSFS SLC 40	True ASDS	1	True	False	True	5e9e3032383ecb6bb234e7ca	5	8	B1062	-80.577366	28.5618571

GitHub URL:

https://github.com/NikAzam422/SpaceX.git

File name:

jupyter-labs-webscraping.ipynb

Data Wrangling

The collected data is further processed to further understand it:

1) Replace missing values with mean of other values
2) Calculate number of rocket launch site, orbit, launches count, and landing outcomes
3) Simplify landing outcome into numeric data (Fail : 0 and Success : 1)
4) Calculate landing success rate based on past data

Launch site:

CCAFS SLC 40	55
KSC LC 39A	22
VAFB SLC 4E	13

*[CCAFS LC 40]
also come out on later data

Orbit:

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

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

Landing success rate:

0.6666666666666666

GitHub URL:

https://github.com/NikAzam422/SpaceX.git

File name:

labs-jupyter-spacex-Data wrangling.ipynb

EDA with Data Visualization

For Exploratory Data Analysis (EDA), several graph/chart have been created :

- Scatter plot (Flight Number vs. PayloadMass, and Landing outcomes)
- Scatter plot (Flight Number vs. Launch site, and Landing outcomes)
- Scatter plot (Payload Mass vs. Launch site, and Landing outcomes)
- Bar chart (Orbit and Landing outcomes)
- Scatter plot(Flight Number vs. Orbit, and Landing outcomes)
- Scatter plot(Payload Mass vs. Orbit, and Landing outcomes)
- Line chart (Year vs. Success rate)

Each plot and chart figure will be shown on later slides.

GitHub URL:

https://github.com/NikAzam422/SpaceX.git

File name:

jupyter-labs-eda-dataviz.ipynb.jupyterlite.ipynb

EDA with SQL

For Exploratory Data Analysis (EDA) with SQL :

1) For this analysis, another data is extracted and analyzed
2) https://cf-courses-data.s3.us.cloud-object-storage.appdomain.cloud/IBM-DS0321EN-SkillsNetwork/labs/module_2/data/Spacex.csv
3) Check all launch site list (4 Launch sites are available in the data)
4) Check Total Payload Mass (kg) : 45,596kg
5) Check Average Payload Mass (kg) : 2,928kg
6) Check dates of launch (Such as the earliest date launch record is on 4th Oct 2010)
7) Check list of Booster Version
8) Check count of the recorded mission’s success and failure
9) Check the which Booster Version has the max Payload Mass
10) Check failure measure in within certain date

GitHub URL:

https://github.com/NikAzam422/SpaceX.git

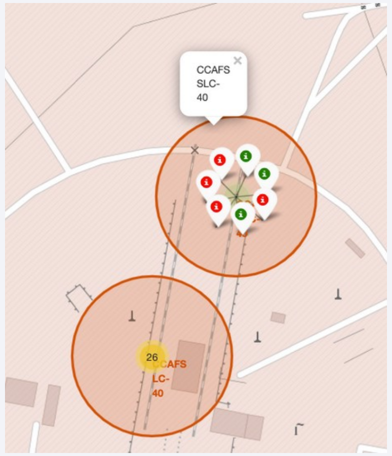
File name:

jupyter-labs-eda-sql-coursera_sqllite.ipynb

Build an Interactive Map with Folium

With Interactive Map Folium, some important location can be visualized on map :

1) Location of launch sites
2) Location of NASA Johnson Space Center at Houston Texas
3) Colored markers and label popups are also inserted into the map
4) Count of success and failure of launch also insert into the map
5) Nearest coastline from some launch site also marked, line drawn and its distance is calculated



GitHub URL:

https://github.com/NikAzam422/SpaceX.git

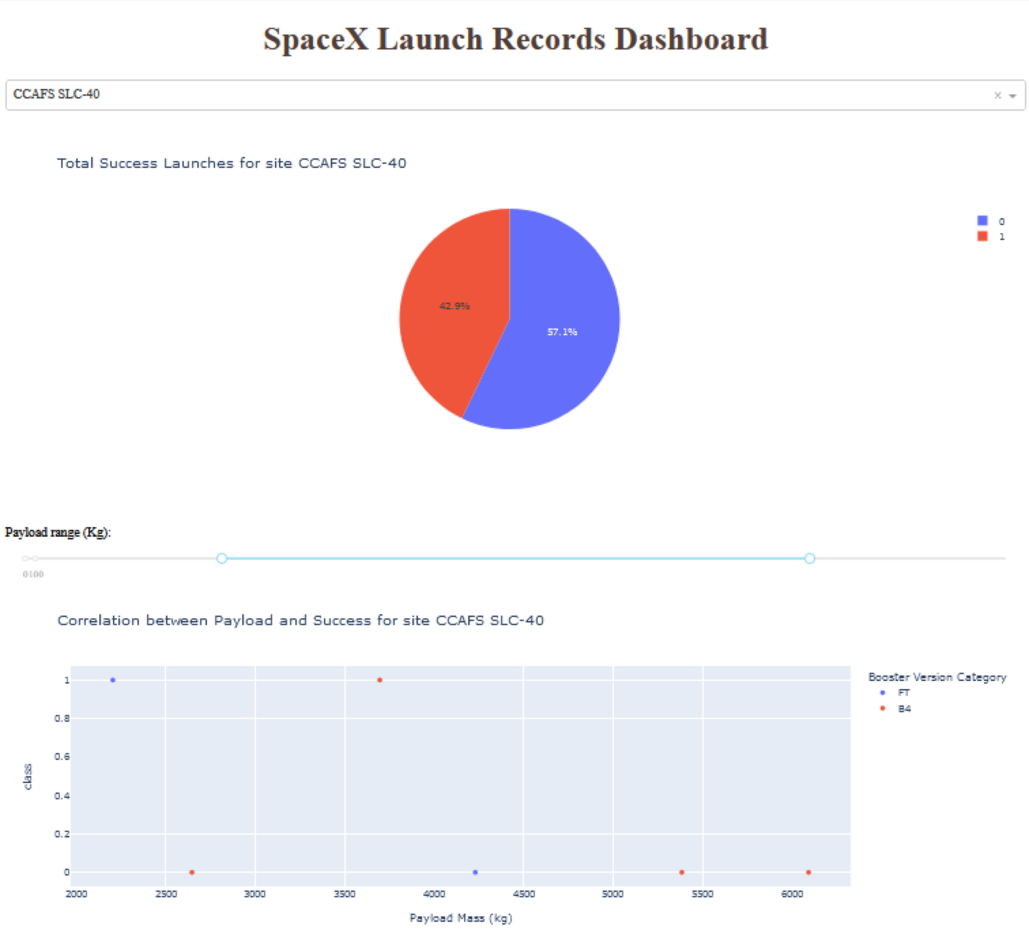
File name:

jupyter-labs-eda-sql-coursera_sqllite.ipynb

Build a Dashboard with Plotly Dash

A dashboard is created with below function:

- Pie chart and Scatter plot as output
- Pull down and numerical slider as input
- Depends on input, the output will change in real time
- If pull down input ‘ALL’ is selected, Pie Chart will show the ratio of successful landing on all launch site.
- If pull down input launch site name is selected, Pie Chart will show the ratio of success and failure of landing of the site
- The numerical slider is for Pay load Mass.
- The Scatter plot, which is Pay load mass vs. Success rate, will change in real time according to input of numerical slider and selected launch site.



GitHub URL:

https://github.com/NikAzam422/SpaceX.git

File name:

spacex_dash_app.py

Predictive Analysis (Classification)

- For predictive analysis, machine learning method is used
- For machine learning, 80% of data is assigned for Training, 20% is assigned for Test.
- Predictive analysis has been conducted by using GridSearchCV function from Scikit-Learn API
- Logistic Regression, Support Vector Machine, Tree Classifier, K-Nearest Neighbour objects have been used on GridSearchCV function.
- Each result is then compared to find the best prediction method.

GitHub URL:

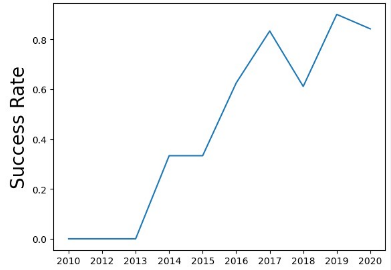
https://github.com/NikAzam422/SpaceX.git

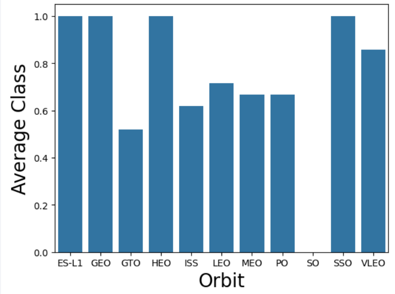
File name:

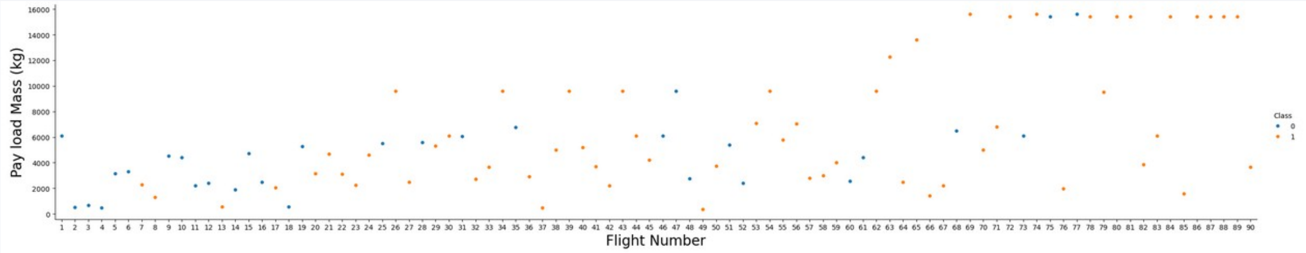
spacex_dash_app.py

Results

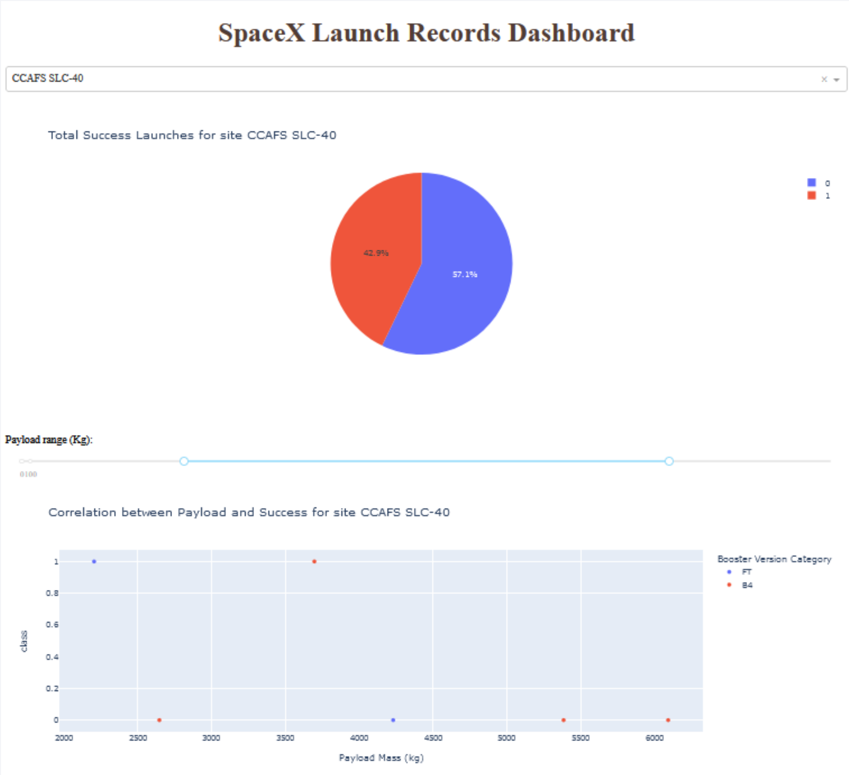
- **Exploratory data analysis results**







- **Interactive analytics demo in screenshots**



- **Predictive analysis results**

Tree classifier method has shown higher evaluation for Predictive analysis

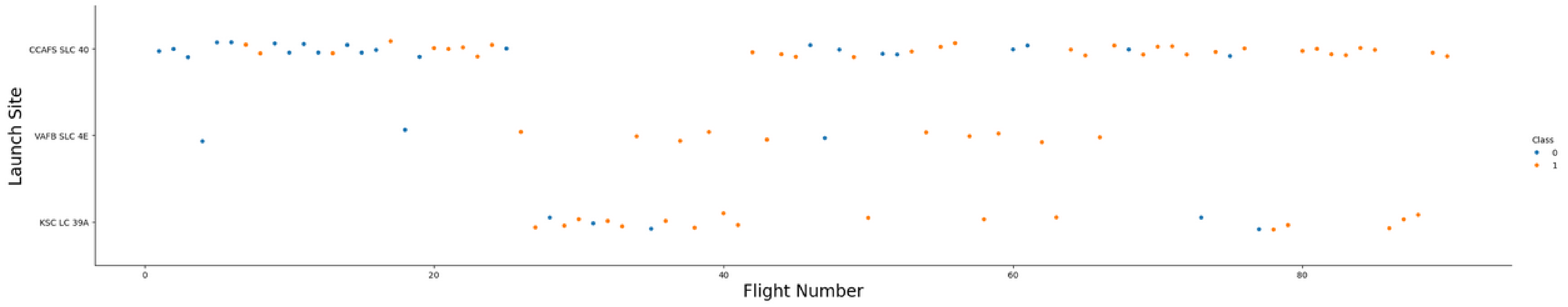
Its score is 0.9027777777777778 (Nearer to 1 is best result)

Section 2

Insights drawn from EDA

Flight Number vs. Launch Site

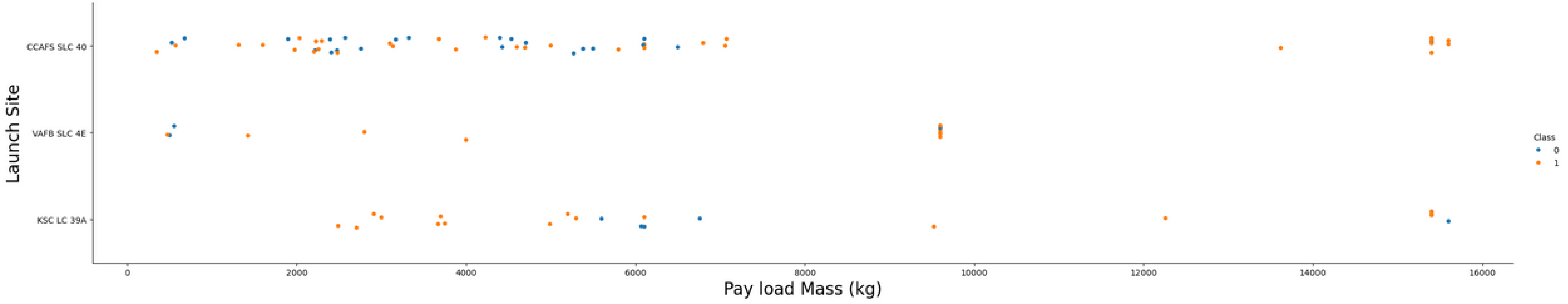
```
# Plot a scatter point chart with x axis to be Flight Number and y axis to be the launch site, and hue to be the class value
sns.catplot(x="FlightNumber", y="LaunchSite", hue="Class", data=df, aspect = 5)
plt.xlabel("Flight Number",fontsize=20)
plt.ylabel("Launch Site",fontsize=20)
plt.show()
```



- CCAFS-SLC-40 Launch site has the most launch count
- CCAFS-SLC-40 Launch site has lots of failed landing, but more successful latter
- KSC-LC-39A only has most launch while other site do not

Payload vs. Launch Site

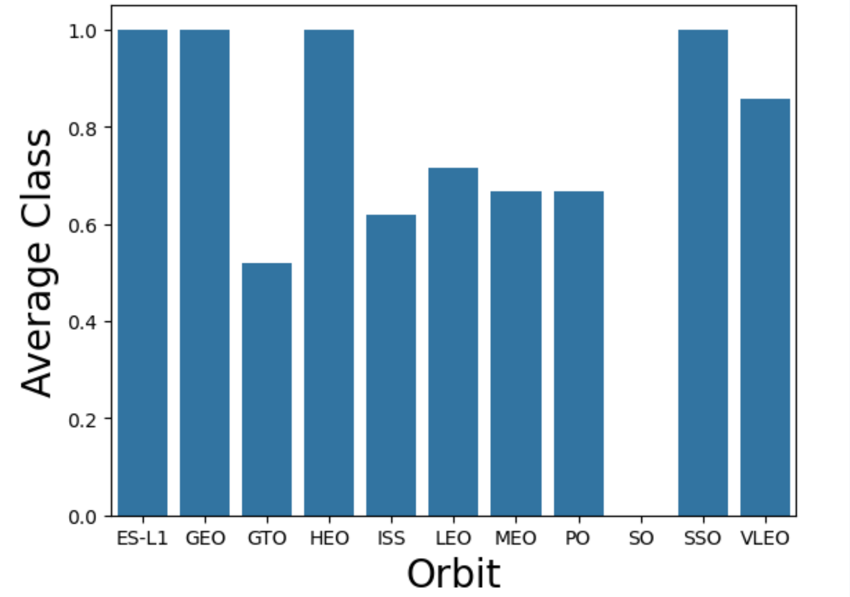
```
# Plot a scatter point chart with x axis to be Pay Load Mass (kg) and y axis to be the launch site, and hue to be the class value
sns.catplot(x="PayloadMass", y="LaunchSite", hue="Class", data=df, aspect = 5)
plt.xlabel("Pay load Mass (kg)",fontsize=20)
plt.ylabel("Launch Site",fontsize=20)
plt.show()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



- Most launch has less than 8,000 kg pay load
- CCAFS-SLC-40 and KSC-LC-39A has high pay load mass launch, but VAFB-SLC-4E do not.
- VAFB-SLC-4E mostly launch middle pay load around 9,000 ~ 10,000 kg

Success Rate vs. Orbit Type

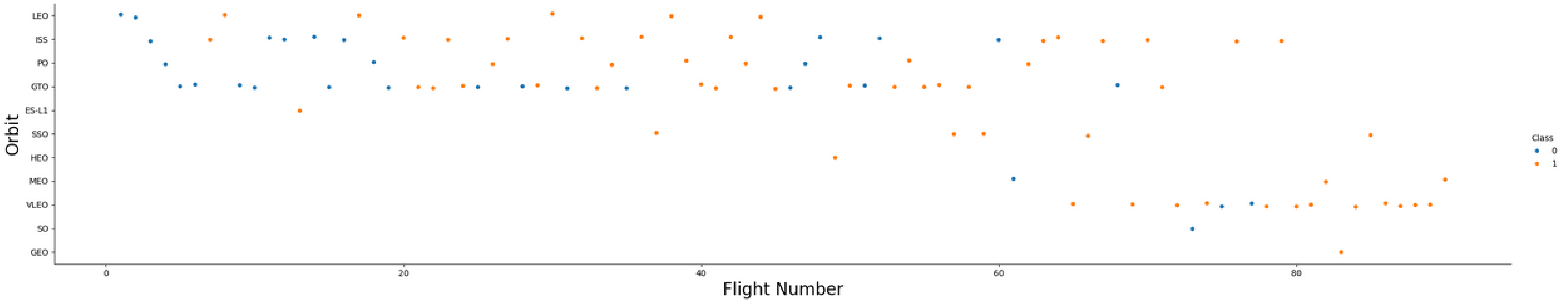
```
# HINT use groupby method on Orbit column and get the mean of Class column
df_orbit = df.groupby(['Orbit'])['Class'].mean().reset_index()
#df_orbit

sns.barplot(x='Orbit', y='Class', data=df_orbit)
plt.xlabel("Orbit",fontsize=20)
plt.ylabel("Average Class",fontsize=20)
plt.show()
```



- ES-L1, GEO, HEO, and SSO orbit has very high success rate
- SO orbit has lowest success rate
- Other orbits have more than 50% of success rate

Flight Number vs. Orbit Type

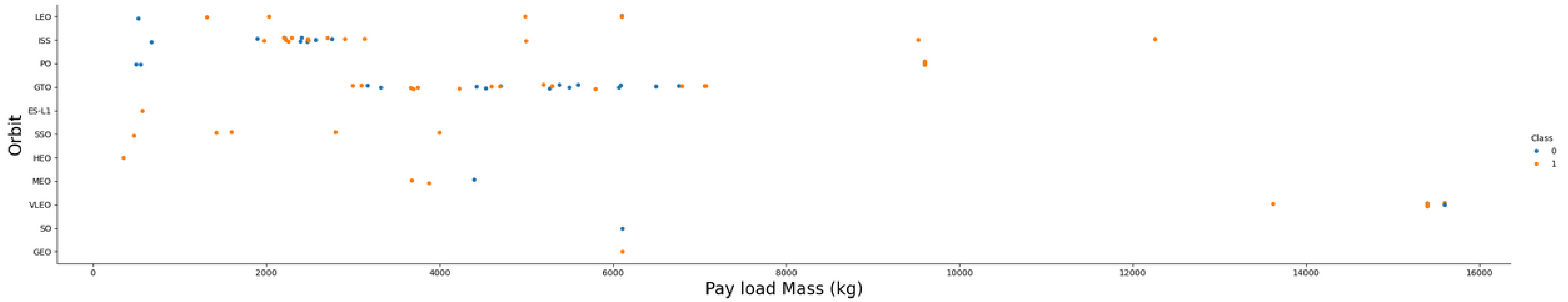
```
# Plot a scatter point chart with x axis to be FlightNumber and y axis to be the Orbit, and hue to be the class value
sns.catplot(x="FlightNumber", y="Orbit", hue="Class", data=df, aspect = 5)
plt.xlabel("Flight Number",fontsize=20)
plt.ylabel("Orbit",fontsize=20)
plt.show()
```



- Most of launch are to LEO, ISS, PO, GTO orbit
- Not much launches for EL-L1, SSO, HEO, VLEO, SO, GEO orbit
- Lots of launch for VLEO orbit later on

Payload vs. Orbit Type

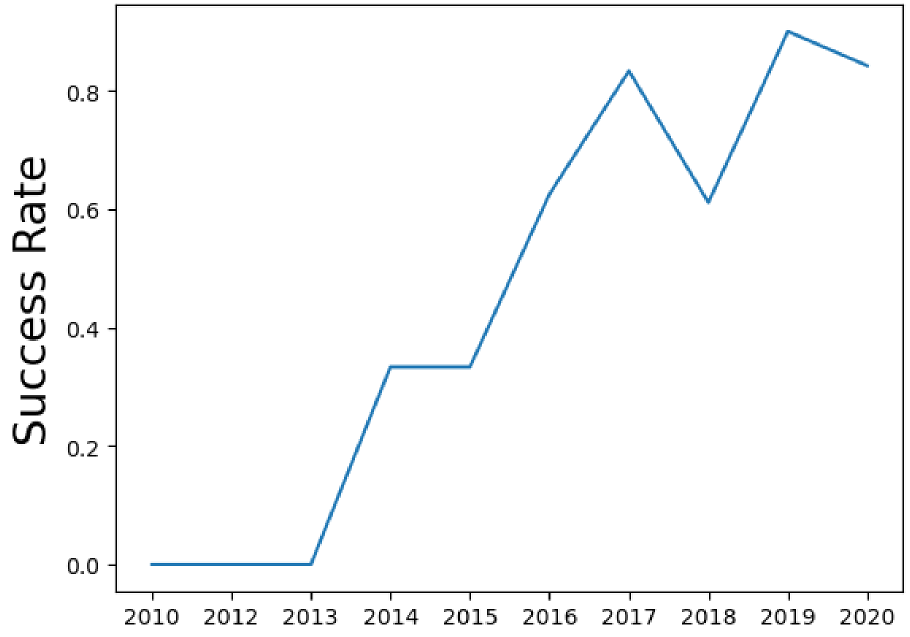
```
# Plot a scatter point chart with x axis to be Payload and y axis to be the Orbit, and hue to be the class value
sns.catplot(x="PayloadMass", y="Orbit", hue="Class", data=df, aspect = 5)
plt.xlabel("Pay load Mass (kg)",fontsize=20)
plt.ylabel("Orbit",fontsize=20)
plt.show()
```



- GTO orbit mostly has launch with around 3,000 ~ 8,000 kg payload mass
- ISS orbit mostly has payload mass around 2,000 ~ 4,000 kg
- VLEO orbit only have high payload mass, which is around 13,000 ~ 16,000 kg

Launch Success Yearly Trend

```python
# Plot a line chart with x axis to be the extracted year and y axis to be the success rate
df_orbit_class = df.groupby(['Date'])['Class'].mean().reset_index()
sns.lineplot(x="Date", y="Class", data=df_orbit_class)
plt.xlabel("Year",fontsize=20)
plt.ylabel("Success Rate",fontsize=20)
plt.show()
```



- Success rate since 2013 kept increasing till 2020
- There is only small flop around 2019

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

There are 4 launch sites:

- CCAFS LC-40
- VAB SLC-4E
- KSC LC-39A
- CCAFS SLC-40

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

Total Payload Mass

```
%sql select sum(PAYLOAD_MASS__KG_) from SPACEXTABLE where Customer =='NASA (CRS)'
```

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

sum(PAYLOAD_MASS__KG_)
45596

Average Payload Mass by F9 v1.1

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

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

avg(PAYLOAD_MASS__KG_)
2928.4

First Successful Ground Landing Date

```
%sql select min(Date) from SPACEXTABLE where Mission_Outcome=='Success'
```

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

min(Date)
2010-06-04

Successful Drone Ship Landing with Payload between 4000 and 6000

```
%sql select distinct Booster_Version from SPACEXTABLE where PAYLOAD_MASS__KG_ between 4000 and 6000
```

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

Booster_Version
F9 v1.1
F9 v1.1 B1011
F9 v1.1 B1014
F9 v1.1 B1016
F9 FT B1020
F9 FT B1022
F9 FT B1026
F9 FT B1030
F9 FT B1021.2
F9 FT B1032.1
F9 B4 B1040.1
F9 FT B1031.2
F9 B4 B1043.1
F9 FT B1032.2
F9 B4 B1040.2
F9 B5 B1046.2
F9 B5 B1047.2
F9 B5 B1046.3
F9 B5B1054
F9 B5 B1048.3
F9 B5 B1051.2
F9 B5B1060.1
F9 B5 B1058.2
F9 B5B1062.1

Total Number of Successful and Failure Mission Outcomes

```
%sql select Mission_Outcome, count(*) as Total from SPACEXTBL group by Mission_Outcome
```

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

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

Success : 100

Fail : 1

*Mission outcomes is not same as failure or success of landing

Boosters Carried Maximum Payload

```
%sql select Booster_Version, max(PAYLOAD_MASS__KG_) from SPACEXTBL
```

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

Booster_Version	max(PAYLOAD_MASS__KG_)
F9 B5 B1048.4	15600

2015 Launch Records

```
%sql select substr(Date,6,2), Landing_Outcome, Booster_Version, Launch_Site from SPACEXTBL where Landing_Outcome like 'Failure%' and substr(Date,0,5)=='2015'
```

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

substr(Date,6,2)	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

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

```
%sql select Landing_Outcome, Date from SPACEXTBL where Date between '2010-06-04' and '2017-03-20' order by Date desc
```

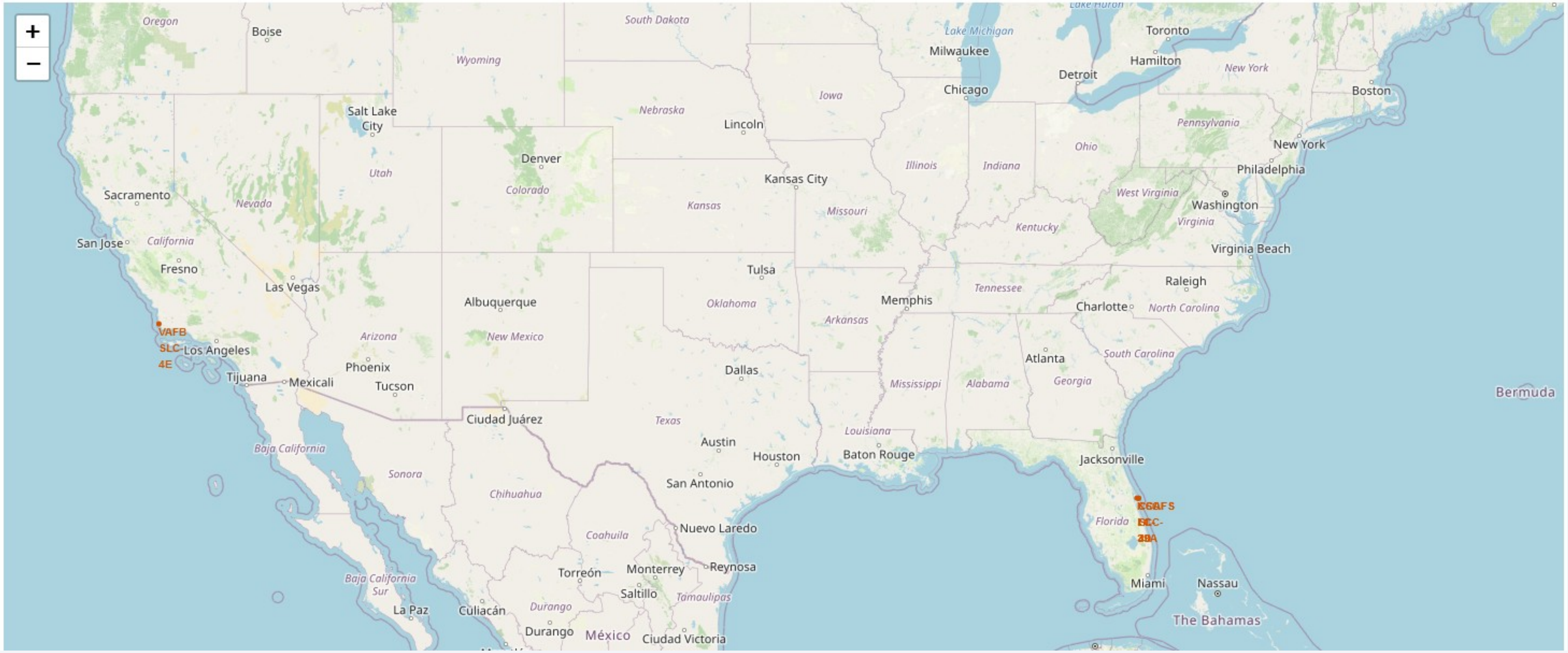
```
 * sqlite:///my_data1.db
Done.
```

Landing_Outcome	Date
No attempt	2017-03-16
Success (ground pad)	2017-02-19
Success (drone ship)	2017-01-14
Success (drone ship)	2016-08-14
Success (ground pad)	2016-07-18
Failure (drone ship)	2016-06-15
Success (drone ship)	2016-05-27
Success (drone ship)	2016-05-06
Success (drone ship)	2016-04-08
Failure (drone ship)	2016-03-04
Failure (drone ship)	2016-01-17
Success (ground pad)	2015-12-22
Precluded (drone ship)	2015-06-28
No attempt	2015-04-27
Failure (drone ship)	2015-04-14
No attempt	2015-03-02
Controlled (ocean)	2015-02-11
Failure (drone ship)	2015-01-10
Uncontrolled (ocean)	2014-09-21
No attempt	2014-09-07
No attempt	2014-08-05
Controlled (ocean)	2014-07-14
Controlled (ocean)	2014-04-18
No attempt	2014-01-06
No attempt	2013-12-03
Uncontrolled (ocean)	2013-09-29
No attempt	2013-03-01
No attempt	2012-10-08
No attempt	2012-05-22
Failure (parachute)	2010-12-08
Failure (parachute)	2010-06-04

Section 3

Launch Sites Proximities Analysis

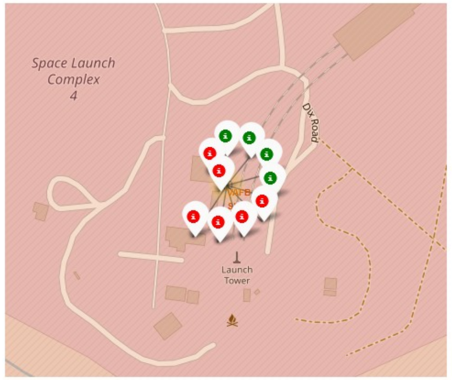
<Folium Map Screenshot 1>

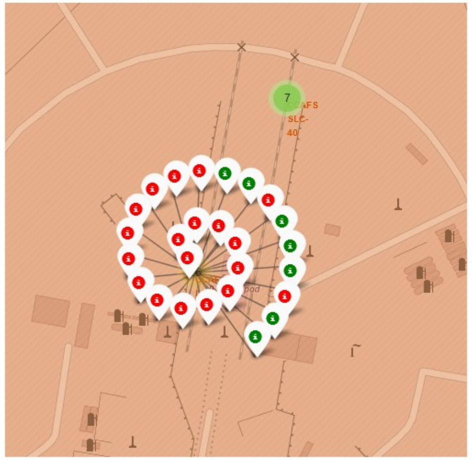


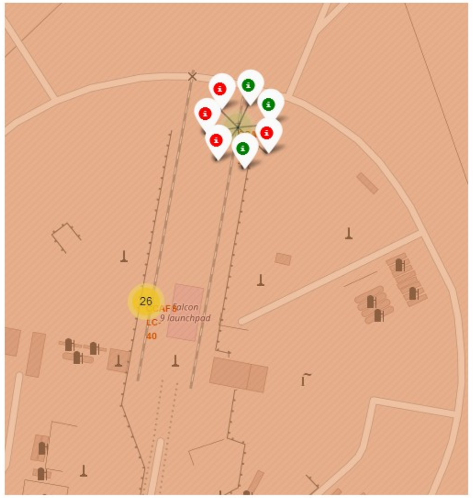
SpaceX Launch Site Location

- All launch site location are near coastline
- Only VAFB SLC-4E at the westcoast. Other launch site are at eastcoast

<Folium Map Screenshot 2>







VAB SLC-4E

KSC LC-39A

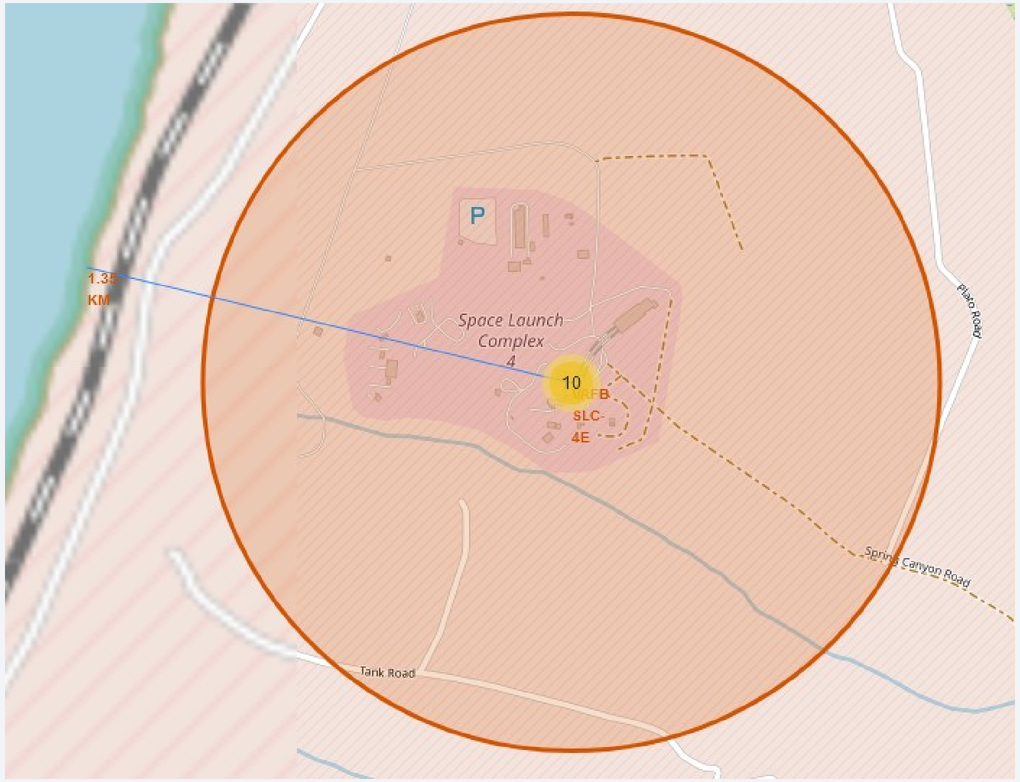
CCAFS LC-40

CCAFS SLC-40

Green : Success

Red : Failure

<Folium Map Screenshot 3>

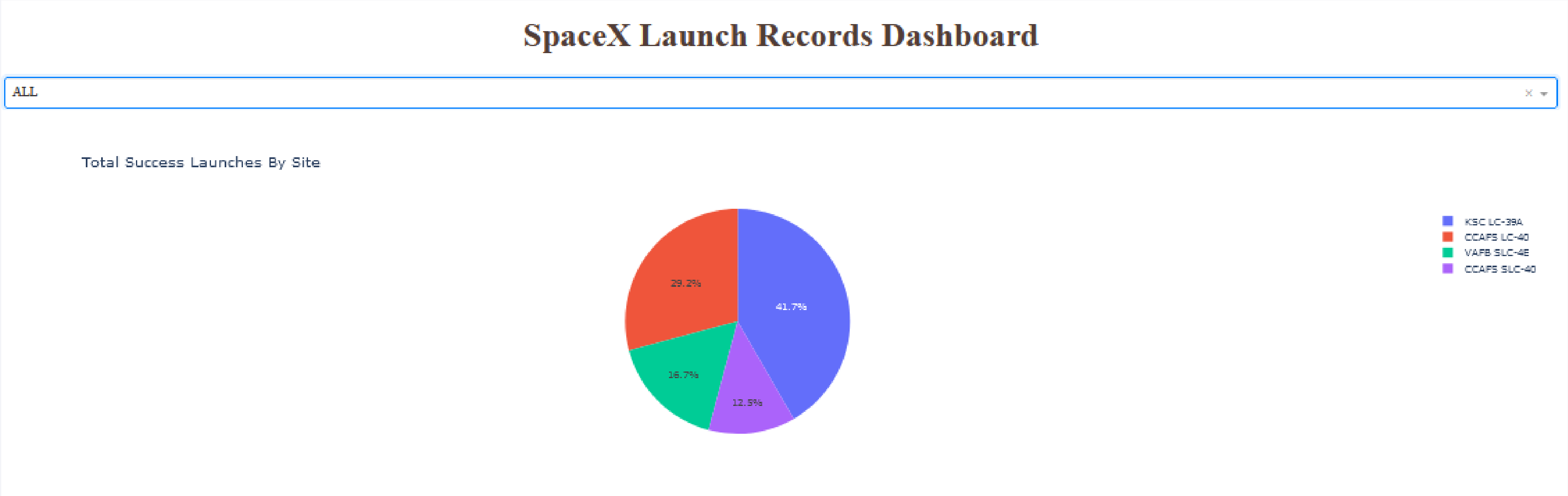


VAB SLC-4E launch site is around 1.35 km away from nearest coast line

Section 4

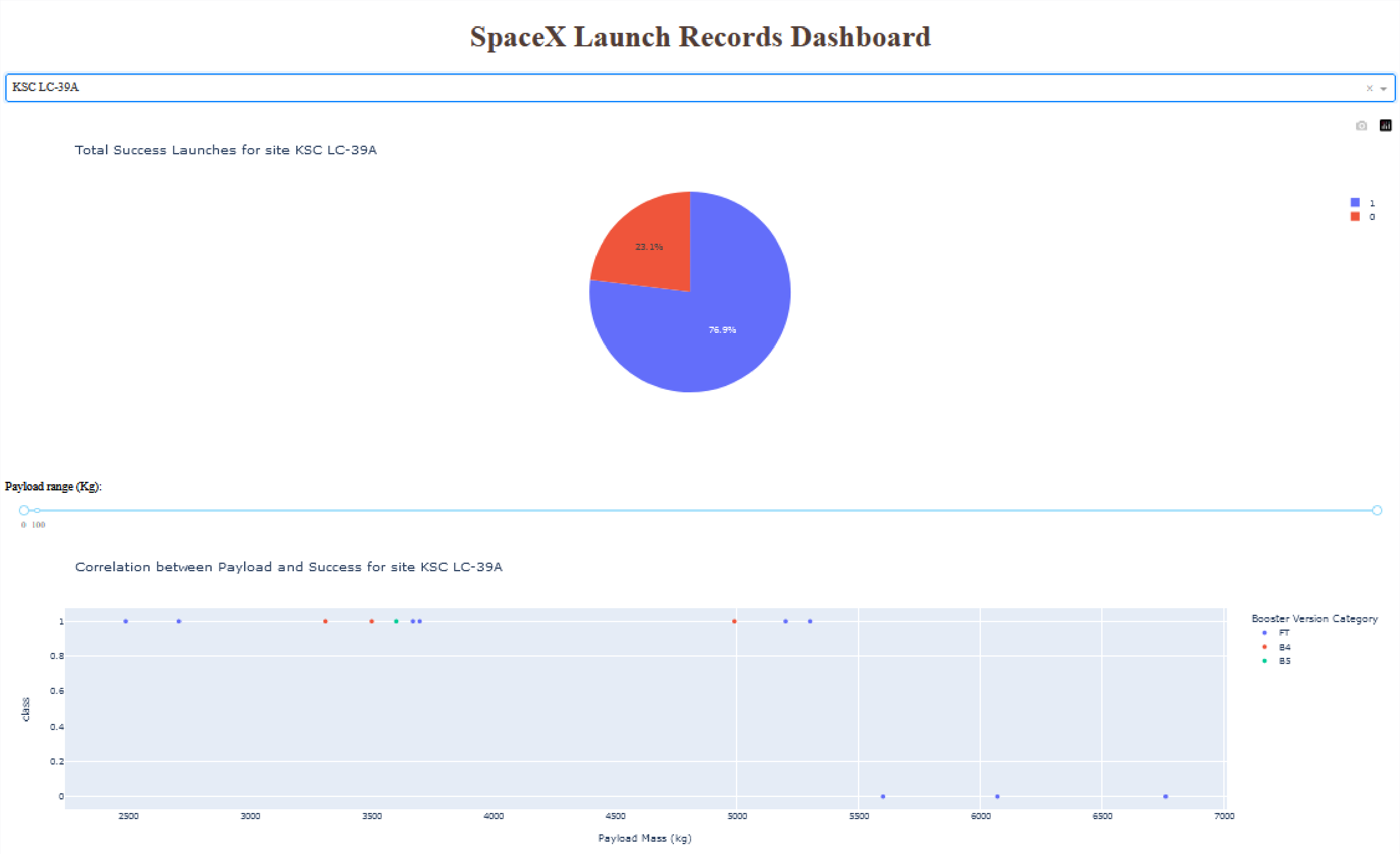
Build a Dashboard with Plotly Dash

<Dashboard Screenshot 1>



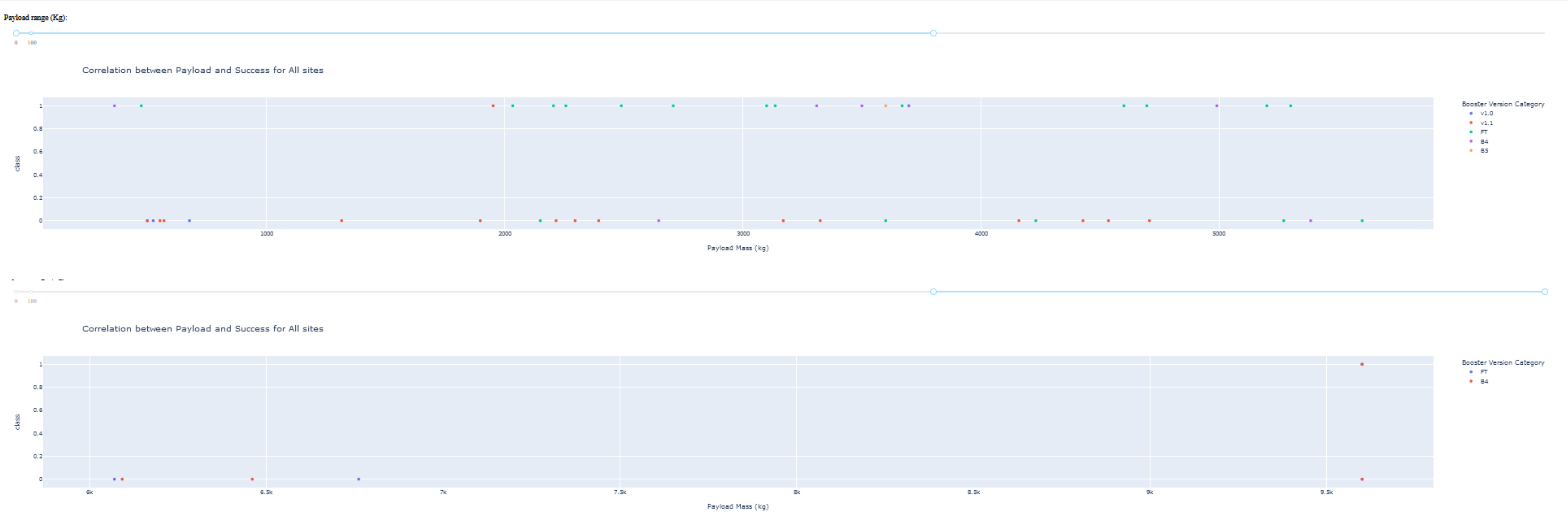
- Most success launch are from KSC LC-39A site
- Least success launch are from CCAFS SLC-40 site

<Dashboard Screenshot 2>



- KSC LC-39A success rate is around 76.9%
- Higher pay load mass has tendency to be failed

<Dashboard Screenshot 3>

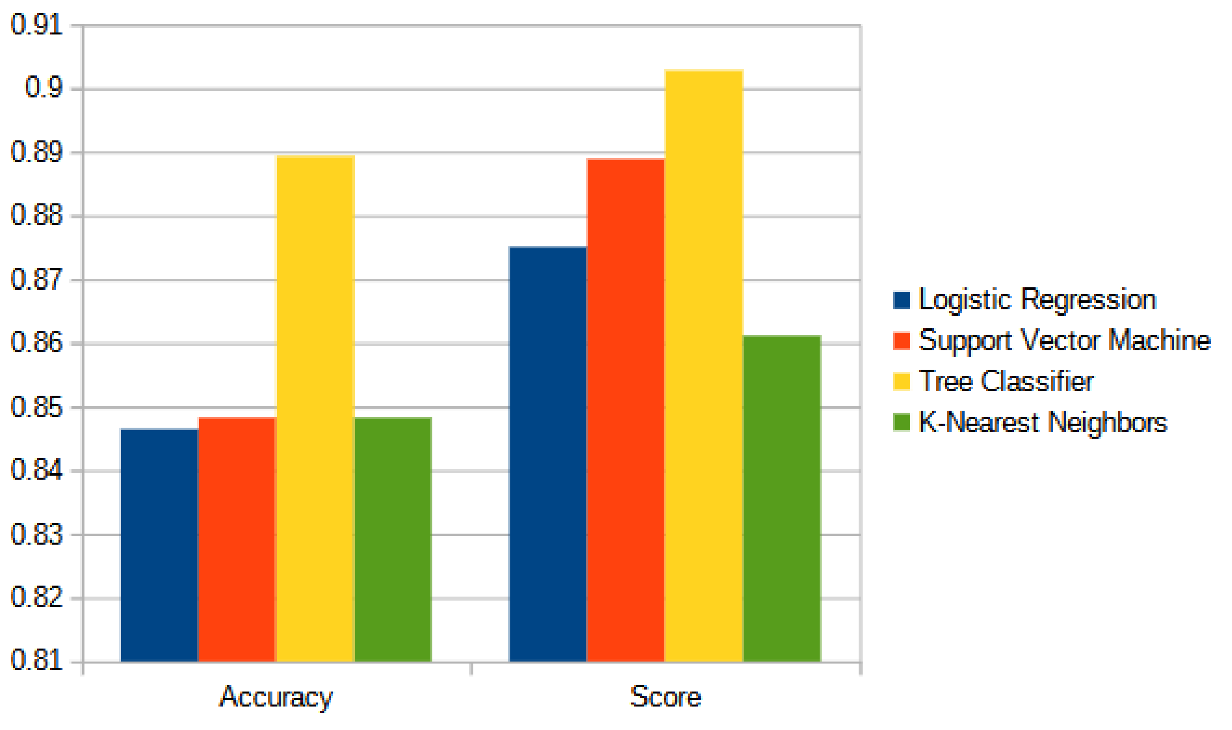


- There is not clear success or fail within lower pay load mass
- But it become clear success rate is lower with higher pay load mass

Section 5

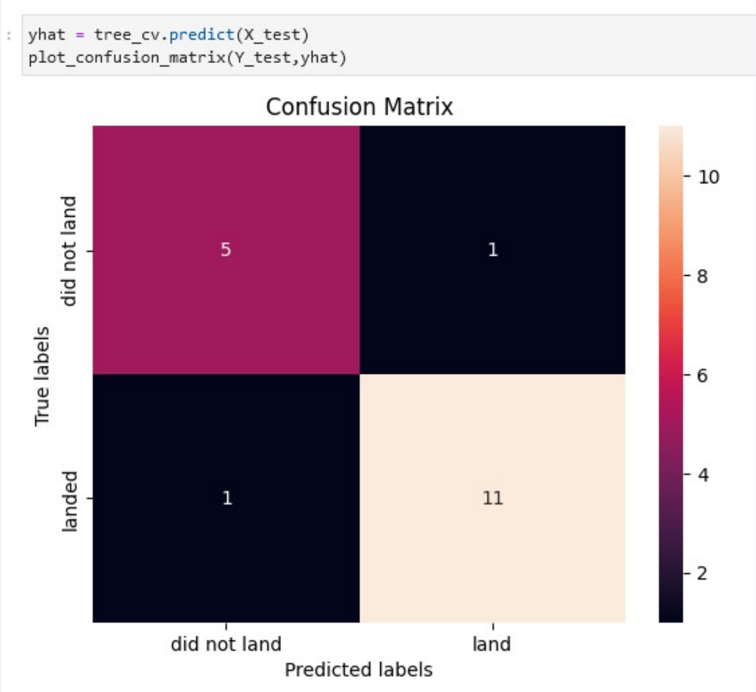
Predictive Analysis (Classification)

Classification Accuracy



- Tree Classifier has most highest accuracy and score

Confusion Matrix



- Tree Classifier method predicted 5 launches doesn’t land successfully correctly, while predict wrong 1
- Tree Classifier method predicted 11 launches land successfully correctly, while predict wrong 1

Conclusions

Point 1 :

ES-L1, GEO, HEO, and SSO orbit has very high success rate. While SO has very low success rate

Point 2 :

Success rate has been increasing yearly

Point 3 :

Higher pay load mass has tendency to fail.

Point 4 :

Tree Classifier method has the best accuracy when predicting outcome of rocket landing.

Appendix

Include any relevant assets like Python code snippets, SQL queries, charts, Notebook outputs, or data sets that you may have created during this project

Thank you!